User Guide

Arista Networks

www.arista.com

Arista EOS version Version 4.19.0F
9 November 2017

Headquarters	Support	Sales
5453 Great America Parkway Santa Clara, CA 95054 USA		
(408) 547-5500	(408) 547-5502 (866) 476-0000	(408) 547-5501 (866) 497-0000
www.arista.com	support@arista.com	sales@arista.com

© Copyright 2017 Arista Networks, Inc. The information contained herein is subject to change without notice. Arista Networks and the Arista logo are trademarks of Arista Networks, Inc., in the United States and other countries. Other product or service names may be trademarks or service marks of others.

Table of Contents

Chapter 1 Overview 1

Chapter 2 Initial Configuration and Recovery 3

Initial Switch Access 3
Connection Management 9
Configure Session 10
Recovery Procedures 15
Session Management Commands 19

Chapter 3 Command-Line Interface 47

Accessing the EOS CLI 47
Processing Commands 48
Kernel-based Virtual Machine Commands and Configuration 55
Switch Platforms 61
Command Modes 73
Managing Switch Configuration Settings 76
Other Command-Line Interfaces 79
Common Criteria (CC) 80
Directory Structure 81
Command-Line Interface Commands 82

Chapter 4 AAA Configuration 155

Authorization, Authentication, and Accounting Overview 155
Configuring the Security Services 157
Server Groups 165
Role Based Authorization 166
Activating Security Services 175
TACACS+ Configuration Examples 178
AAA Commands 180

Chapter 5 Administering the Switch 237

Managing the Switch Name 237
Managing the System Clock 240
Synchronizing the Time Settings 242
Managing Display Attributes 252
Event Monitor 254
Switch Administration Commands 258

Chapter 6 Booting the Switch 335

Boot Loader – Aboot 335
Configuration Files 336
Supervisor Redundancy 340
System Reset 342
Aboot Shell 347
Aboot Configuration Commands 351
Switch Booting Commands 356

Chapter 7 Upgrades and Downgrades 375

Upgrade/Downgrade Overview 375
Accelerated Software Upgrade (ASU) 376
Leaf Smart System Upgrade (Leaf SSU) 380
Standard Upgrades and Downgrades 387

Upgrade/Downgrade Commands 395
Chapter 8 Switch Environment Control 399
Environment Control Introduction 399
Environment Control Overview 399
Configuring and Viewing Environment Settings 401
Environment Commands 406
Chapter 9 Maintenance Mode 419
Overview 420
Maintenance Mode Elements 421
Maintenance Mode Features 426
Maintenance Mode Configuration 427
Maintenance Mode Commands 443
Chapter 10 Ethernet Ports 495
Ethernet Ports Introduction 495
Ethernet Standards 495
Ethernet Physical Layer 498
Interfaces 501
Ethernet Configuration Procedures 504
Ethernet Configuration Commands 525
Chapter 11 Port Channels and LACP 559
Port Channel Introduction 559
Port Channel Conceptual Overview 559
Port Channel Configuration Procedures 562
Load Balancing Hash Algorithms 567
Port Channel and LACP Configuration Commands 573
Chapter 12 Multi-Chassis Link Aggregation 623
MLAG Introduction 623
MLAG Conceptual Overview 624
MLAG Maintenance 626
Configuring MLAG 629
MLAG Implementation Example 636
MLAG Commands 645
Chapter 13 802.1x Port Security 663
802.1x Port Security Introduction 663
802.1x Port Security Description 664
Configuring 802.1x Port Security 669
Displaying 802.1x information 672
IEEE 802.1x Configuration Commands 674
Chapter 14 DCBX and Flow Control 687
Introduction 687
DCBX and Priority-Based Flow Control Overview 688
DCBX and PFC Configuration and Verification Procedures 689
DCBX and Flow Control Configuration Commands 691
Chapter 15 LLDP 705
LLDP Introduction 705
LLDP Overview 705
LLDP Configuration Procedures 707

LLDP Configuration Commands 713

Chapter 16 Data Transfer 731

Data Transfer Introduction 731
Data Transfer Methods 732
MAC Address Table 736
Configuring Ports 740
Monitoring Links 750
Data Transfer Command Descriptions 762

Chapter 17 Tap Aggregation 845

Tap Aggregation Introduction 845
Tap Aggregation Description 846
Tap Aggregation Configuration 850
Tap Aggregation Traffic Steering 857
Tap Aggregation GUI 860
Keyframe and Timestamp Configuration 862
Tap Aggregation Command Descriptions 864

Chapter 18 VLANs 901

VLAN Introduction 901
VLAN Conceptual Overview 901
VLAN Configuration Procedures 904
VLAN Configuration Commands 912

Chapter 19 VXLAN 943

VXLAN Introduction 943
VXLAN Description 944
VXLAN Configuration 950
VXLAN Command Descriptions 964

Chapter 20 ACLs and Route Maps 983

ACL, Service ACL, Route Map, and Prefix List Introduction 983
Access Control Lists 985
Service ACLs 1000
Route Maps 1004
Prefix Lists 1009
ACL, Route Map, and Prefix List Commands 1013

Chapter 21 VRRP and VARP 1073

VRRP and VARP Conceptual Overview 1073
VRRP and VARP Implementation Procedures 1075
VRRP and VARP Implementation Examples 1083
VRRP and VARP Configuration Commands 1088

Chapter 22 Spanning Tree Protocol 1115

Introduction to Spanning Tree Protocols 1115
Spanning Tree Overview 1115
Configuring a Spanning Tree 1122
STP Commands 1137

Chapter 23 Quality of Service 1205

Quality of Service Conceptual Overview 1205
QoS Configuration: Arad Platform Switches 1211
QoS Configuration: Jericho Platform Switches 1222

QoS Configuration: FM6000 Platform Switches 1233
QoS Configuration: Petra Platform Switches 1241
QoS Configuration: Trident Platform Switches 1250
QoS Configuration: Trident-II and Helix Platform Switches 1261
Quality of Service Configuration Commands 1269

Chapter 24 IPv4 1345

IPv4 Addressing 1345
IPv4 Routing 1353
IPv4 Multicast Counters 1358
Route Management 1362
IPv4 Route Scale 1367
IP Source Guard 1371
DHCP Relay Across VRF 1373
IP NAT 1376
IPv4 Command Descriptions 1384

Chapter 25 IPv6 1477

Introduction 1477
IPv6 Description 1478
Configuring IPv6 1480
IPv6 Command Descriptions 1489

Chapter 26 Traffic Management 1535

Traffic Management Conceptual Overview 1536
Traffic Management Configuration – Arad Platform Switches 1539
Traffic Management Configuration – FM6000 Platform Switches 1545
Traffic Management Configuration – Petra Platform Switches 1553
Traffic Management Configuration – Trident Platform Switches 1556
Traffic Management Configuration – Trident-II Platform Switches 1565
Traffic Management Configuration Commands 1568

Chapter 27 Open Shortest Path First – Version 2 1653

OSPFv2 Introduction 1653
OSPFv2 Conceptual Overview 1654
Configuring OSPFv2 1657
OSPFv2 Examples 1673
OSPFv2 Commands 1682

Chapter 28 Open Shortest Path First – Version 3 1747

OSPFv3 Introduction 1747
OSPFv3 Conceptual Overview 1748
Configuring OSPFv3 1751
OSPFv3 Examples 1762
OSPFv3 Commands 1770

Chapter 29 Border Gateway Protocol (BGP) 1817

BGP Conceptual Overview 1818
Configuring BGP 1821
BGP Examples 1840
BGP Commands 1843

Chapter 30 Routing Information Protocol 1959

RIP Conceptual Overview 1959
Running RIP on the Switch 1960

Configuring RIP on Multiple VRFs 1963
RIP Commands 1964

Chapter 31 IS-IS 1981

IS-IS Introduction 1981
IS-IS 1981
IS-IS Command Descriptions 1991

Chapter 32 Multiprotocol Label Switching (MPLS) 2031

MPLS 2031
Decap Groups 2034
Nexthop Groups 2035
MPLS Command Descriptions 2040

Chapter 33 Bidirectional Forwarding Detection 2057

Introduction 2057
BFD Configuration 2059
BFD Command Descriptions 2063

Chapter 34 Multicast Architecture 2077

Introduction 2077
Multicast Architecture Description 2078
Multicast Configuration 2080
Multicast Commands 2085

Chapter 35 IGMP and IGMP Snooping 2103

Introduction 2103
IGMP Protocols 2104
Configuring IGMP 2106
Configuring IGMP Snooping 2109
IGMP Host Proxy 2118
IGMP and IGMP Snooping Commands 2121

Chapter 36 Protocol Independent Multicast 2203

Introduction 2203
Configuring PIM 2206
Multicast Example 2210
PIM Commands 2214

Chapter 37 Multicast Source Discovery Protocol 2247

MSDP Introduction 2247
MSDP Description 2248
MSDP Configuration 2251
MSDP Commands 2259

Chapter 38 Audio Video Bridging (AVB) 2283

AVB Overview 2283
AVB Protocols 2284
AVB Configuration 2287
AVB Command Descriptions 2291

Chapter 39 SNMP 2305

SNMP Introduction 2305
SNMP Conceptual Overview 2305
Configuring SNMP 2306

SNMP Commands 2313

Chapter 40 Latency Analyzer (LANZ) 2345

Introduction to LANZ 2345
LANZ Overview 2345
Configuring LANZ 2347
LANZ Commands 2357

Chapter 41 VM Tracer 2387

VM Tracer Introduction 2387
VM Tracer Description 2388
VM Tracer Configuration Procedures 2390
VM Tracer Configuration Commands 2394

Chapter 42 Path Tracer 2417

Path Tracer Description 2417
Path Tracer Configuration 2422
Path Tracer Command Descriptions 2431

Chapter 43 MapReduce Tracer 2453

MapReduce Tracer Introduction 2453
MapReduce Tracer Configuration 2455
Displaying MapReduce Tracer Results 2460
MapReduce Tracer Command Descriptions 2465

Chapter 44 sFlow 2513

sFlow Conceptual Overview 2513
sFlow Configuration Procedures 2516
sFlow Configuration Commands 2518

Chapter 45 OpenFlow 2533

OpenFlow Introduction 2533
OpenFlow Description 2533
OpenFlow Configuration 2541
OpenFlow Command Descriptions 2544

Chapter 46 DirectFlow 2569

Introduction 2569
DirectFlow Configuration 2572
DirectFlow Feature Interactions 2575
DirectFlow Command Descriptions 2578

Command Reference

Chapter 1 Overview 1

Chapter 2 Initial Configuration and Recovery 3

domain (XMPP Management) 20
idle-timeout (Console Management) 21
idle-timeout (SSH Management) 22
idle-timeout (Telnet Management) 23
management api http-commands 24
management console 25
management ssh 26
management telnet 27
management xmpp 28
protocol http (API Management) 29
protocol https (API Management) 30
protocol https certificate (API Management) 31
server (XMPP Management) 32
session privilege (XMPP Management) 33
show inventory 34
show xmpp neighbors 35
show xmpp status 36
show xmpp switch-group 37
shutdown (API Management) 38
shutdown (Telnet Management) 39
shutdown (XMPP Management) 40
switch-group (XMPP Management) 41
username (XMPP Management) 42
vrf (API Management) 43
vrf (XMPP Management) 44
xmpp send 45
xmpp session 46

Chapter 3 Command-Line Interface 47

action bash 84
alias 85
bash 86
boot test memory 87
comment (various configuration modes) 88
configure (configure terminal) 89

configure network 90
copy running-config 91
daemon 92
delay 93
dir 94
disable 95
enable 96
end 97
entropy source hardware 98
event-handler 99
event-handler DropCountersHandler 100
exit 101
fips restrictions (SSH Management) 102
hostkey client strict-checking (SSH Management) 103
ip ftp client source-interface 104
ip http client source-interface 105
ip ssh client source-interface 106
ip tftp client source-interface 107
known-hosts (SSH Management) 108
local (SSH Management-Tunnel) 109
logging host 110
logging source-interface 111
logging trap system 112
log-level (SSH Management) 114
management security 115
platform arad lag mode 116
platform sand fabric mode (7500 and 7500E Series) 117
platform sand forwarding mode (7500 and 7500E Series) 119
platform sand lag hardware-only 121
pwd 122
remote (SSH Management-Tunnel) 123
schedule 124
schedule config max-concurrent-jobs 126
secret hash 127
send log message 128
server-alive count-max (SSH Management-Tunnel) 129
server-alive interval (SSH Management-Tunnel) 130
show (various configuration modes) 131
show event-handler 132
show event-handler DropCountersHandler 133
show management ssh hostkey 134
show module 135
show platform sand compatibility 138
show platform sand lag hardware-only 139
show schedule 140
show schedule summary 141

show version 142
shutdown (SSH Management-Tunnel) 143
ssh 144
tcpdump 146
tcpdump session 148
terminal length 150
terminal monitor 151
trigger 152
tunnel (SSH Management) 153

Chapter 4 AAA Configuration 155

aaa accounting 182
aaa accounting dot1x 183
aaa accounting system 184
aaa authentication enable 185
aaa authentication login 186
aaa authentication policy local 188
aaa authentication policy log 189
aaa authorization commands 190
aaa authorization config-commands 191
aaa authorization console 192
aaa authorization exec 193
aaa authorization policy local default-role 194
aaa group server radius 195
aaa group server tacacs+ 196
aaa root 197
clear aaa counters 198
clear aaa counters radius 199
clear aaa counters tacacs+ 200
deny (Role) 201
enable secret 203
ip radius source-interface 204
ip tacacs source-interface 205
no <sequence number> (Role) 206
permit (Role) 207
radius-server deadtime 209
radius-server host 210
radius-server key 212
radius-server retransmit 213
radius-server timeout 214
resequence (Role) 215
role 216
server (server-group-RADIUS configuration mode) 217
server (server-group-TACACS+ configuration mode) 218
show aaa 219
show aaa counters 220

show aaa method-lists 221
show aaa sessions 222
show privilege 223
show radius 224
show role 225
show tacacs 226
show user-account 227
show users 228
tacacs-server host 229
tacacs-server key 231
tacacs-server policy 232
tacacs-server timeout 233
username 234
username sshkey 236

Chapter 5 Administering the Switch 237

banner login 260
banner motd 261
clear ptp interface counters 262
clock set 263
clock timezone 264
email 265
no event-monitor 266
event-monitor <log enable> 267
event-monitor backup max-size 269
event-monitor backup path 270
event-monitor buffer max-size 271
event-monitor clear 272
event-monitor interact 273
event-monitor sync 274
hostname 275
ip domain-list 276
ip domain lookup 277
ip domain-name 278
ip host 279
ip name-server 280
ipv6 host 281
ntp authenticate 282
ntp authentication-key 283
ntp serve 284
ntp serve all 285
ntp server 286
ntp source 289
ntp trusted-key 290
prompt 291
ptp announce interval 293

ptp announce timeout 294
ptp delay-mechanism 295
ptp delay-req interval 296
ptp domain 297
ptp enable 298
ptp forward-v1 299
ptp hold-ptp-time 300
ptp mode 301
ptp pdelay-neighbor-threshold 302
ptp pdelay-req interval 303
ptp priority1 304
ptp priority2 305
ptp source ip 306
ptp sync interval 307
ptp sync timeout 308
ptp transport 309
ptp ttl 310
show banner 311
show clock 312
show event-monitor arp 313
show event-monitor mac 315
show event-monitor route 317
show event-monitor sqlite 319
show hostname 320
show hosts 321
show ip domain-name 322
show ip name-server 323
show ntp associations 324
show ntp status 325
show ptp 326
show ptp clock 327
show ptp foreign-master-record 328
show ptp interface 329
show ptp interface counters 330
show ptp parent 331
show ptp source ip 332
show ptp time-property 333

Chapter 6 Booting the Switch 335

CONSOLESPEED 352
NET commands 353
PASSWORD (ABOOT) 354
SWI 355
boot console 357
boot secret 358
boot system 360

erase startup-config 361
protocol 362
redundancy 363
redundancy force-switchover 364
reload 365
reload <scheduled> 367
service sequence-numbers 368
show redundancy file-replication 369
show redundancy states 370
show redundancy switchover sso 371
show reload 372
show reload cause 373

Chapter 7 Upgrades and Downgrades 375
install 396
reload fast-boot 397
reload hitless 398

Chapter 8 Switch Environment Control 399
environment fan-speed 407
environment insufficient-fans action 408
environment overheat action 409
locator-led 410
show environment all 411
show environment cooling 412
show environment power 414
show environment temperature 415
show locator-led 417

Chapter 9 Maintenance Mode 419
group bgp 445
group interface 446
maintenance 447
interface 448
neighbor 449
maintenance profile interface 450
maintenance profile bgp 451
vrf 452
profile interface 453
profile bgp 454
profile unit 455
unit 456
interface 457
bgp <peer> [vrf <vrf_name>] 458
profile interface <profile_name> default 459
profile bgp <profile_name> default 460
profile unit <profile_name> default 461

group bgp <group_name> ... 462
group interface <group_name> ... 463
profile unit <profile_name> ... 464
quiesce ... 466
rate-monitoring load-interval ... 468
rate-monitoring threshold ... 469
shutdown max-delay ... 470
initiator route-map <route-map-name> inout ... 471
on-boot duration ... 472
trigger on-maintenance ... 473
show maintenance ... 476
show maintenance summary ... 477
show maintenance units ... 478
show maintenance interface ... 479
show maintenance interface status ... 481
show maintenance bgp ... 482
show maintenance groups ... 483
show maintenance profiles ... 484
show maintenance stages ... 485
show maintenance bgp receiver route-map ... 486
show maintenance debug ... 487
show interface ... 489
show interface <intf_name> status ... 490
show ip | ipv6 bgp summary [vrf <vrf_name>] ... 491
show ip | ipv6 bgp neighbors <peer_addr> [vrf <vrf_name>] ... 492

Chapter 10 Ethernet Ports ... 495

flowcontrol receive ... 526
flowcontrol send ... 527
hardware port-group ... 528
interface ethernet ... 530
interface ethernet create ... 531
interface management ... 532
link-debounce ... 533
mac-address ... 534
show flowcontrol ... 535
show hardware counter ... 536
show hardware port-group ... 537
show interfaces capabilities ... 538
show interfaces counters ... 539
show interfaces counters bins ... 540
show interfaces counters errors ... 541
show interfaces counters queue ... 542
show interfaces counters rates ... 543
show interfaces negotiation ... 544
show interfaces phy ... 545

show interfaces status 547
show interfaces status errdisabled 548
show interfaces transceiver 549
show interfaces transceiver channels 550
show interfaces transceiver hardware 551
show interfaces transceiver properties 552
show platform fm6000 agileport map 553
speed 554
transceiver qsfp default-mode 556
transceiver channel 557

Chapter 11 Port Channels and LACP 559

channel-group 574
distribution random 576
distribution symmetric-hash 577
fields ip 578
fields mac 580
ingress load-balance profile 581
interface port-channel 582
lacp port-priority 583
lacp rate 584
lacp system-priority 585
load-balance fm6000 profile 586
load-balance policies 588
port-channel hash-seed 589
port-channel lacp fallback 590
port-channel lacp fallback timeout 592
port-channel load-balance 593
port-channel load-balance arad fields ip 595
port-channel load-balance fm6000 fields ip 596
port-channel load-balance fm6000 fields mac 597
port-channel load-balance module 598
port-channel load-balance petraA fields ip 599
port-channel load-balance trident fields ip 600
port-channel load-balance trident fields ipv6 602
port-channel load-balance trident fields mac 604
port-channel min-links 605
show etherchannel 606
show lacp aggregates 607
show lacp counters 609
show lacp interface 610
show lacp internal 612
show lacp neighbor 613
show lacp sys-id 615
show load-balance profile 616
show port-channel 617

show port-channel limits 619
show port-channel load-balance fields 620
show port-channel summary 621
show port-channel traffic 622

Chapter 12 Multi-Chassis Link Aggregation 623
domain-id 646
heartbeat-interval (MLAG) 647
local-interface 648
mlag (port-channel interface configuration) 649
mlag configuration (global configuration) 650
peer-address 651
peer-link 652
reload-delay mlag 653
reload-delay mode 654
reload-delay non-mlag 655
show mlag 656
show mlag interfaces 657
show mlag interfaces members 658
show mlag interfaces states 659
show mlag issu warnings 660
shutdown (MLAG) 661

Chapter 13 802.1x Port Security 663
clear dot1x statistics 675
dot1x system-auth-control 676
dot1x max-reauth-req 677
dot1x pae authenticator 678
dot1x port-control 679
dot1x reauthentication 680
dot1x timeout quiet-period 681
dot1x timeout reauth-period 682
dot1x timeout tx-period 683
show dot1x 684
show dot1x statistics 685
show dot1x all summary 686

Chapter 14 DCBX and Flow Control 687
dcbx application priority 692
dcbx mode 693
no priority-flow-control 694
platform fm6000 pfc-wm 695
priority-flow-control mode 696
priority-flow-control priority 697
show dcbx 698
show dcbx application-priority-configuration 699
show dcbx priority-flow-control-configuration 700

show dcbx status 701
show interfaces priority-flow-control 702
show platform fm6000 pfc-wm 703
show priority-flow-control 704

Chapter 15 LLDP 705

clear lldp counters 714
clear lldp table 715
lldp holdtime 716
lldp management-address 717
lldp management-address vrf 718
lldp receive 719
lldp reinit 720
lldp run 721
lldp timer 722
lldp tlv-select 723
lldp transmit 724
show lldp 725
show lldp local-info 727
show lldp neighbors 728
show lldp traffic 730

Chapter 16 Data Transfer 731

clear counters 764
clear mac address-table dynamic 767
clear server-failure servers inactive 768
control-plane 769
default-profiles 770
description 771
errdisable detect cause link-flap 772
errdisable flap-setting cause link-flap 773
errdisable recovery cause 774
errdisable recovery interval 776
interface loopback 777
ip access-group (Control Plane mode) 778
link state group 779
link state track 780
links minimum 781
load interval 782
mac address-table aging-time 783
mac address-table static 784
monitor link-flap policy 787
monitor link-flap profiles 788
monitor server-failure 789
monitor server-failure link 790
monitor session destination 791
monitor session destination cpu 793

monitor session ip access-group 795
monitor session source 796
monitor session source ip access-group 798
monitor session truncate 799
mtu 800
network (server-failure configuration mode) 801
no monitor session 802
platform sand monitor serdes error log 803
platform sand monitor serdes error threshold 804
platform sand monitor serdes poll period 805
platform sand monitor serdes poll threshold isolation 806
platform sand monitor serdes poll threshold recovery 807
profile max-flaps (Link Flap Configuration) 808
proxy (server-failure configuration mode) 810
show fabric monitoring health 811
show interfaces 812
show interfaces description 814
show link state group 815
show mac address-table 816
show mac address-table aging time 818
show mac address-table count 819
show mac address-table mlag-peer 820
show mac address-table multicast 821
show mac address-table multicast brief 822
show monitor server-failure 823
show monitor server-failure history 824
show monitor server-failure servers 825
show monitor session 827
show port-security 828
show port-security address 829
show port-security interface 830
show storm-control 831
show switch forwarding-mode 832
show track 833
shutdown (server-failure configuration mode) 834
storm-control 835
switch forwarding-mode 836
switchport 837
switchport default mode access 838
switchport default mode routed 839
switchport mac address learning 840
switchport port-security 841
switchport port-security maximum 842
switchport port-security violation protect 843
track 844

Chapter 17 Tap Aggregation 845

class (policy-map (tapagg)) ... 865
class-map type tapagg ... 867
mac timestamp ... 868
match (class-map (tapagg)) ... 869
match (policy-map (tapagg)) ... 870
mode (tap-agg configuration mode) ... 873
mode exclusive no-errdisable (tap-agg configuration mode) ... 874
platform fm6000 keyframe ... 875
platform fm6000 keyframe device ... 876
platform fm6000 keyframe fields skew ... 877
platform fm6000 keyframe rate ... 878
platform fm6000 keyframe source ... 879
policy-map type tapagg ... 880
resequence (class-map (tapagg)) ... 881
resequence (policy-map (tapagg)) ... 882
service-policy type tapagg (Interface mode) ... 883
set (policy-map-class (tapagg)) ... 884
show interfaces tap ... 885
show interfaces tool ... 886
show platform fm6000 keyframe ... 887
show tap aggregation groups ... 888
switchport tap allowed vlan ... 889
switchport tap default group ... 890
switchport tap identity ... 891
switchport tap native vlan ... 892
switchport tap truncation ... 893
switchport tool allowed vlan ... 894
switchport tool group ... 895
switchport tool identity ... 897
switchport tool truncation ... 898
tap aggregation ... 899

Chapter 18 VLANs ... 901

autostate ... 913
encapsulation dot1q vlan ... 914
interface vlan ... 915
l2-protocol encapsulation dot1q vlan ... 916
name (VLAN configuration mode) ... 917
show dot1q-tunnel ... 918
show interfaces switchport ... 919
show interfaces switchport backup ... 921
show interfaces trunk ... 922
show interfaces vlans ... 923
show vlan ... 924
show vlan dynamic ... 925
show vlan internal allocation policy ... 926

show vlan internal usage 927
show vlan summary 928
show vlan trunk group 929
state 930
switchport dot1q ethertype 931
switchport access vlan 932
switchport mode 933
switchport trunk allowed vlan 935
switchport trunk group 936
switchport trunk native vlan 937
switchport vlan mapping 938
trunk group 939
vlan 940
vlan internal allocation policy 941

Chapter 19 VXLAN 943

clear vxlan counters 965
interface vxlan 966
ip address virtual 967
show service vxlan 968
show vxlan address-table 969
show vxlan counters 971
show vxlan flood vtep 973
show vxlan vtep 974
vxlan flood vtep 975
vxlan multicast-group 977
vxlan multicast-group decap 978
vxlan source-interface 979
vxlan udp-port 980
vxlan vlan vni 981
vxlan vni notation dotted 982

Chapter 20 ACLs and Route Maps 983

clear ip access-lists counters 1015
clear ipv6 access-lists counters 1016
continue (route-map) 1017
deny (IPv4 ACL) 1018
deny (IPv6 ACL) 1020
deny (IPv6 Prefix List) 1022
deny (MAC ACL) 1023
deny (Standard IPv4 ACL) 1025
deny (Standard IPv6 ACL) 1026
description (route-map) 1027
hardware access-list update default-result permit 1028
hardware counter feature acl out 1029
ip access-group 1030
ip access-group (Service ACLs) 1031

ip access-list 1032
ip access-list standard 1033
ip prefix-list 1034
ipv6 access-group 1035
ipv6 access-group (Service ACLs) 1036
ipv6 access-list 1037
ipv6 access-list standard 1038
ipv6 prefix-list 1039
mac access-group 1040
mac access-list 1041
match (route-map) 1042
no <sequence number> (ACLs) 1044
permit (IPv4 ACL) 1045
permit (IPv6 ACL) 1047
permit (IPv6 Prefix List) 1049
permit (MAC ACL) 1050
permit (Standard IPv4 ACL) 1052
permit (Standard IPv6 ACL) 1053
remark 1054
resequence (ACLs) 1055
route-map 1056
seq (IPv6 Prefix Lists) 1058
set (route-map) 1059
set community (route-map) 1061
set extcommunity (route-map) 1062
show (ACL configuration modes) 1063
show ip access-lists 1065
show ip prefix-list 1066
show ipv6 access-lists 1067
show ipv6 prefix-list 1068
show mac access-lists 1069
show route-map 1070
statistics per-entry (ACL configuration modes) 1071

Chapter 21 VRRP and VARP 1073

ip fhrp accept-mode 1089
ip virtual-router address 1090
ip virtual-router mac-address 1091
ip virtual-router mac-address advertisement-interval 1092
ipv6 virtual-router address 1093
no vrrp 1094
show ip virtual-router 1095
show ipv6 virtual-router 1096
show vrrp 1097
show vrrp internal 1099
vrrp authentication 1100

vrrp delay reload 1101
vrrp description 1102
vrrp ip 1103
vrrp ip secondary 1104
vrrp ip version 1105
vrrp ipv6 1106
vrrp mac-address advertisement-interval 1107
vrrp preempt 1108
vrrp preempt delay 1109
vrrp priority 1111
vrrp shutdown 1112
vrrp timers advertise 1113
vrrp track 1114

Chapter 22 Spanning Tree Protocol 1115

abort (mst-configuration mode) 1139
clear spanning-tree counters 1140
clear spanning-tree counters session 1141
clear spanning-tree detected-protocols 1142
disabled-time 1143
exit (mst-configuration mode) 1144
instance 1145
loop-protection 1146
monitor loop-protection 1147
name (mst-configuration mode) 1148
protect vlan 1149
rate-limit 1150
revision (mst-configuration mode) 1151
show (mst-configuration mode) 1152
show loop-protection 1154
show spanning-tree 1156
show spanning-tree blockedports 1159
show spanning-tree bridge 1160
show spanning-tree bridge assurance 1161
show spanning-tree counters 1162
show spanning-tree interface 1163
show spanning-tree mst 1164
show spanning-tree mst configuration 1166
show spanning-tree mst interface 1167
show spanning-tree mst test information 1168
show spanning-tree root 1169
show spanning-tree topology status 1170
shutdown (Loop-protection) 1171
spanning-tree bpdufilter 1172
spanning-tree bpduguard 1173
spanning-tree bpduguard rate-limit count (global) 1174

spanning-tree bpduguard rate-limit count (interface) 1175
spanning-tree bpduguard rate-limit default 1176
spanning-tree bpduguard rate-limit enable / disable 1177
spanning-tree bridge assurance 1178
spanning-tree cost 1179
spanning-tree forward-time 1181
spanning-tree guard 1182
spanning-tree hello-time 1183
spanning-tree link-type 1184
spanning-tree loopguard default 1185
spanning-tree max-age 1186
spanning-tree max-hops 1187
spanning-tree mode 1188
spanning-tree mst configuration 1189
spanning-tree portchannel guard misconfig 1190
spanning-tree portfast 1191
spanning-tree portfast auto 1192
spanning-tree portfast bpdufilter default 1193
spanning-tree portfast bpduguard default 1194
spanning-tree portfast <port type> 1195
spanning-tree port-priority 1196
spanning-tree priority 1197
spanning-tree root 1198
spanning-tree transmit hold-count 1200
spanning-tree vlan 1201
switchport backup interface 1202
transmit-interval 1204

Chapter 23 Quality of Service 1205

bandwidth guaranteed (Helix) 1271
bandwidth guaranteed (Trident-II) 1272
bandwidth percent (Arad/Jericho) 1273
bandwidth percent (FM6000) 1275
bandwidth percent (Petra) 1277
bandwidth percent (Trident) 1279
mc-tx-queue 1282
platform petraA traffic-class 1283
priority (Arad/Jericho) 1285
priority (FM6000) 1287
priority (Petra) 1289
priority (Trident) 1291
qos cos 1293
qos dscp 1294
qos trust 1295
qos map cos 1296
qos map dscp 1297

qos map traffic-class to cos 1298
qos map traffic-class to dscp 1299
qos map traffic-class to mc-tx-queue 1300
qos map traffic-class to tx-queue 1301
qos map traffic-class to uc-tx-queue 1302
qos random-detect ecn global-buffer (Helix) 1303
qos random-detect ecn global-buffer (Trident) 1305
qos rewrite cos 1307
qos rewrite dscp 1308
random-detect ecn (Arad/Jericho) 1309
random-detect ecn (Helix) 1311
random-detect ecn (Trident) 1313
shape rate (Interface – Arad/Jericho) 1315
shape rate (Interface – FM6000) 1316
shape rate (Interface – Helix) 1317
shape rate (Interface – Petra) 1318
shape rate (Interface – Trident) 1319
shape rate (Interface – Trident-II) 1320
shape rate (Tx-queue – Arad/Jericho) 1321
shape rate (Tx-queue – FM6000) 1322
shape rate (Tx-queue – Helix) 1323
shape rate (Tx-queue – Petra) 1325
shape rate (Tx-queues – Trident) 1327
shape rate (Tx-queue – Trident-II) 1329
show platform petraA traffic-class 1331
show qos interfaces 1333
show qos interfaces random-detect ecn 1334
show qos maps 1335
show qos random-detect ecn 1336
show qos interfaces trust 1337
tx-queue (Arad/Jericho) 1338
tx-queue (FM6000) 1339
tx-queue (Helix) 1340
tx-queue (Petra) 1341
tx-queue (Trident-II) 1342
uc-tx-queue 1343

Chapter 24 IPv4 1345

agent SandL3Unicast terminate 1386
arp 1387
arp cache persistent 1388
arp timeout 1389
clear arp-cache 1390
clear ip arp 1391
clear ip arp inspection statistics 1392
clear ip dhcp relay counters 1393

clear ip dhcp snooping counters 1395
clear ip nat translation 1396
description (VRF) 1397
ip address 1398
ip arp inspection limit 1399
ip arp inspection logging 1400
ip arp inspection trust 1401
ip arp inspection vlan 1402
ip dhcp relay always-on 1403
ip dhcp relay information option (Global) 1404
ip dhcp relay information option circuit-id 1405
ip dhcp smart-relay 1406
ip dhcp smart-relay global 1408
ip dhcp snooping 1409
ip dhcp snooping information option 1410
ip dhcp snooping vlan 1411
ip hardware fib ecmp resilience 1413
ip hardware fib optimize 1414
ip helper-address 1416
ip icmp redirect 1417
ip load-sharing 1418
ip local-proxy-arp 1419
ip nat destination static 1420
ip nat pool 1422
ip nat source dynamic 1423
ip nat source static 1424
ip nat translation low-mark 1426
ip nat translation max-entries 1427
ip nat translation tcp-timeout 1428
ip nat translation udp-timeout 1429
ip proxy-arp 1430
ip route 1431
ip routing 1433
ip source binding 1434
ip verify 1435
ip verify source 1436
platform trident forwarding-table partition 1437
platform trident routing-table partition 1438
rd (VRF configuration mode) 1439
routing-context vrf 1440
show arp 1441
show ip 1443
show ip arp 1444
show ip arp inspection vlan 1446
show ip arp inspection statistics 1447
show ip dhcp relay 1449

show ip dhcp relay counters 1450
show ip dhcp snooping 1451
show ip dhcp snooping counters 1452
show ip dhcp snooping hardware 1453
show ip helper-address 1454
show ip interface 1455
show ip interface brief 1456
show ip nat access-list interface 1457
show ip nat pool 1458
show ip nat translations 1459
show ip route 1461
show ip route age 1463
show ip route gateway 1464
show ip route host 1465
show ip route summary 1466
show ip route tag 1467
show ip verify source 1468
show platform arad ip route 1469
show platform arad ip route summary 1471
show platform trident forwarding-table partition 1472
show routing-context vrf 1473
show vrf 1474
vrf definition 1475
vrf forwarding 1476

Chapter 25 IPv6 1477

clear ipv6 dhcp relay counters 1490
clear ipv6 neighbors 1492
ipv6 address 1493
ipv6 dhcp relay always-on 1494
ipv6 dhcp relay destination 1495
ipv6 enable 1496
ipv6 hardware fib aggregate-address 1497
ipv6 hardware fib ecmp resilience 1498
ipv6 hardware fib nexthop-index 1499
ipv6 helper-address 1500
ipv6 nd managed-config-flag 1501
ipv6 nd ns-interval 1502
ipv6 nd other-config-flag 1503
ipv6 nd prefix 1504
ipv6 nd ra dns-server 1506
ipv6 nd ra dns-servers lifetime 1507
ipv6 nd ra dns-suffix 1508
ipv6 nd ra dns-suffixes lifetime 1509
ipv6 nd ra hop-limit 1510
ipv6 nd ra interval 1511

ipv6 nd ra lifetime 1512
ipv6 nd ra mtu suppress 1513
ipv6 nd ra suppress 1514
ipv6 nd reachable-time 1515
ipv6 nd router-preference 1516
ipv6 neighbor 1517
ipv6 neighbor cache persistent 1518
ipv6 route 1519
ipv6 unicast-routing 1521
ipv6 verify 1522
show ipv6 dhcp relay counters 1523
show ipv6 hardware fib aggregate-address 1524
show ipv6 helper-address 1525
show ipv6 interface 1526
show ipv6 nd ra internal state 1527
show ipv6 neighbors 1528
show ipv6 route 1529
show ipv6 route age 1530
show ipv6 route host 1531
show ipv6 route interface 1532
show ipv6 route summary 1533
show ipv6 route tag 1534

Chapter 26 Traffic Management 1535

bandwidth (policy-map-class (control-plane) – Arad) 1570
bandwidth (policy-map-class (control-plane) – FM6000) 1572
bandwidth (policy-map-class (control-plane) – Helix) 1573
bandwidth (policy-map-class (control-plane) – Petra) 1575
bandwidth (policy-map-class (control-plane) – Trident) 1577
bandwidth (policy-map-class (control-plane) – Trident-II) 1578
class (policy-map (control-plane) – Arad) 1580
class (policy-map (control-plane) – FM6000) 1582
class (policy-map (control-plane) – Helix) 1584
class (policy-map (control-plane) – Petra) 1586
class (policy-map (control-plane) – Trident) 1588
class (policy-map (control-plane) – Trident-II) 1590
class (policy-map (pbr)) 1592
class (policy-map (qos) – FM6000) 1594
class (policy-map (qos) – Helix) 1596
class (policy-map (qos) – Trident) 1598
class (policy-map (qos) – Trident II) 1600
class-map type control-plane 1602
class-map type pbr 1603
class-map type qos 1604
clear policy-map counters 1605
match (class-map (control-plane) – Helix) 1606

match (class-map (control-plane) – Trident) 1607
match (class-map (control-plane) – Trident-II) 1608
match (class-map (pbr)) 1609
match (class-map (qos) – FM6000) 1610
match (class-map (qos) – Helix) 1611
match (class-map (qos) – Trident) 1612
match (class-map (qos) – Trident II) 1613
match (policy-map (pbr)) 1614
platform arad tcam counters feature 1615
policy-map type control-plane 1616
policy-map type pbr 1617
policy-map type pdp 1618
policy-map type qos 1619
resequence (class-map (pbr)) 1620
resequence (policy-map (pbr)) 1621
service-policy type pbr (Interface mode) 1622
service-policy type pdp (Interface mode) 1623
service-policy type qos (Interface mode) 1624
set (policy-map-class (qos) – FM6000) 1626
set (policy-map-class (qos) – Helix) 1627
set (policy-map-class (qos) – Trident) 1628
set (policy-map-class (qos) – Trident II) 1629
set nexthop (policy-map-class – pbr) 1630
set nexthop-group (policy-map-class(pbr) – Arad) 1631
shape (policy-map-class (control-plane) – Arad) 1632
shape (policy-map-class (control-plane) – FM6000) 1634
shape (policy-map-class (control-plane) – Helix) 1635
shape (policy-map-class (control-plane) – Petra) 1637
shape (policy-map-class (control-plane) – Trident) 1639
shape (policy-map-class (control-plane) – Trident-II) 1640
show class-map type control-plane 1642
show class-map type pbr 1643
show class-map type qos 1644
show policy-map type control-plane 1645
show policy-map type pbr 1646
show policy-map type qos 1647
show policy-map type qos counters 1648
show policy-map interface control-plane 1649
show policy-map interface type qos 1650
show policy-map interface type qos counters 1651

Chapter 27 Open Shortest Path First – Version 2 1653

adjacency exchange-start threshold (OSPFv2) 1684
area default-cost (OSPFv2) 1685
area filter (OSPFv2) 1686
area nssa (OSPFv2) 1687

area nssa default-information-originate (OSPFv2) 1688
area nssa no-summary (OSPFv2) 1690
area nssa translate type7 always (OSPFv2) 1691
area range (OSPFv2) 1692
area stub (OSPFv2) 1693
auto-cost reference-bandwidth (OSPFv2) 1694
clear ip ospf neighbor 1695
compatible (OSPFv2) 1696
default-information originate (OSPFv2) 1697
distance ospf (OSPFv2) 1698
distribute-list in 1699
ip ospf authentication 1700
ip ospf authentication-key 1701
ip ospf cost 1702
ip ospf dead-interval 1703
ip ospf hello-interval 1704
ip ospf message-digest-key 1705
ip ospf name-lookup 1706
ip ospf network point-to-point 1707
ip ospf priority 1708
ip ospf retransmit-interval 1709
ip ospf shutdown 1710
ip ospf transmit-delay 1711
log-adjacency-changes (OSPFv2) 1712
max-lsa (OSPFv2) 1713
max-metric router-lsa (OSPFv2) 1714
maximum-paths (OSPFv2) 1715
network area (OSPFv2) 1716
no area (OSPFv2) 1717
passive-interface default (OSPFv2) 1718
passive-interface <interface> (OSPFv2) 1719
point-to-point routes (OSPFv2) 1720
redistribute (OSPFv2) 1721
router-id (OSPFv2) 1722
router ospf 1723
show ip ospf 1724
show ip ospf border-routers 1726
show ip ospf database database-summary 1727
show ip ospf database <link state list> 1728
show ip ospf database <link-state details> 1729
show ip ospf interface 1732
show ip ospf interface brief 1733
show ip ospf lsa-log 1734
show ip ospf neighbor 1735
show ip ospf neighbor adjacency-changes 1736
show ip ospf neighbor state 1737

show ip ospf neighbor summary 1738
show ip ospf request-list 1739
show ip ospf retransmission-list 1740
show ip ospf spf-log 1741
shutdown (OSPFv2) 1742
timers lsa arrival (OSPFv2) 1743
timers throttle lsa all (OSPFv2) 1744
timers throttle spf (OSPFv2) 1745

Chapter 28 Open Shortest Path First – Version 3 1747

adjacency exchange-start threshold (OSPFv3) 1771
area default-cost (OSPFv3) 1772
area nssa (OSPFv3) 1773
area nssa default-information-originate (OSPFv3) 1774
area nssa translate type7 always (OSPFv3) 1775
area range (OSPFv3) 1776
area stub (OSPFv3) 1777
clear ipv6 ospf force-spf 1778
default-information originate (OSPFv3) 1779
default-metric (OSPFv3) 1780
distance ospf intra-area (OSPFv3) 1781
ipv6 ospf area 1782
ipv6 ospf cost 1783
ipv6 ospf dead-interval 1784
ipv6 ospf hello-interval 1785
ipv6 ospf network 1786
ipv6 ospf priority 1787
ipv6 ospf retransmit-interval 1788
ipv6 ospf transmit-delay 1789
ipv6 router ospf 1790
log-adjacency-changes (OSPFv3) 1791
max-metric router-lsa (OSPFv3) 1792
maximum-paths (OSPFv3) 1793
no area (OSPFv3) 1794
passive-interface (OSPFv3) 1795
redistribute (OSPFv3) 1796
router-id (OSPFv3) 1797
show ipv6 ospf 1798
show ipv6 ospf border-routers 1799
show ipv6 ospf database 1800
show ipv6 ospf database<link-state details> 1801
show ipv6 ospf database <link state list> 1804
show ipv6 ospf database link 1806
show ipv6 ospf database link if-name 1807
show ipv6 ospf database link if-type 1808
show ipv6 ospf interface 1810

show ipv6 ospf lsa-log 1811
show ipv6 ospf neighbor 1812
show ipv6 ospf neighbor state 1813
show ipv6 ospf neighbor summary 1814
show ipv6 ospf spf-log 1815
shutdown (OSPFv3) 1816

Chapter 29 Border Gateway Protocol (BGP) 1817

address-family 1846
aggregate-address 1847
bgp advertise-inactive 1849
bgp always-compare-med 1850
bgp bestpath as-path ignore 1851
bgp bestpath as-path multipath-relax 1852
bgp bestpath ecmp-fast 1853
bgp bestpath med confed 1854
bgp bestpath med missing-as-worst 1855
bgp bestpath tie-break cluster-list-length 1856
bgp bestpath tie-break router-id 1857
bgp client-to-client reflection 1858
bgp cluster-id 1859
bgp confederation identifier 1860
bgp confederation peers 1861
bgp default 1862
bgp enforce-first-as 1864
bgp listen limit 1865
bgp listen range 1866
bgp log-neighbor-changes 1867
bgp redistribute-internal (BGP) 1868
clear ip bgp 1869
clear ip bgp neighbor * 1871
clear ipv6 bgp 1872
clear ipv6 bgp neighbor * 1874
distance bgp 1875
graceful-restart stalepath-time 1876
graceful-restart-helper 1877
ip as-path access-list 1878
ip as-path regex-mode 1879
ip community-list expanded 1880
ip community-list standard 1881
ip extcommunity-list expanded 1882
ip extcommunity-list standard 1883
maximum paths (BGP) 1884
neighbor activate 1885
neighbor allowas-in 1886
neighbor auto-local-addr 1887

neighbor default-originate 1888
neighbor description 1889
neighbor ebgp-multihop 1890
neighbor enforce-first-as 1891
neighbor export-localpref 1892
neighbor graceful-restart-helper 1893
neighbor import-localpref 1894
neighbor local-as 1895
neighbor local-v4-addr 1896
neighbor local-v6-addr 1897
neighbor maximum-routes 1898
neighbor next-hop-peer 1899
neighbor next-hop-self 1900
neighbor out-delay 1901
neighbor password 1902
neighbor peer-group (create) 1903
neighbor peer-group (neighbor assignment) 1905
neighbor remote-as 1906
neighbor remove-private-as 1907
neighbor route-map (BGP) 1908
neighbor route-reflector-client 1909
neighbor send-community 1910
neighbor shutdown 1911
neighbor soft-reconfiguration 1912
neighbor timers 1913
neighbor transport connection-mode 1914
neighbor update-source 1915
neighbor weight 1916
network (BGP) 1917
no neighbor 1918
redistribute (BGP) 1919
router-id (BGP) 1920
router bgp 1921
show bgp instance 1922
show ip as-path access-list 1923
show ip bgp 1924
show ip bgp community 1928
show ip bgp neighbors 1929
show ip bgp neighbors (route type) 1932
show ip bgp neighbors (route-type) community 1934
show ip bgp neighbors regexp 1935
show ip bgp paths 1936
show ip bgp peer-group 1937
show ip bgp regexp 1939
show ip bgp summary 1940
show ip community-list 1942

show ip extcommunity-list 1943
show ipv6 bgp 1944
show ipv6 bgp community 1946
show ipv6 bgp neighbors 1947
show ipv6 bgp neighbors (route type) 1949
show ipv6 bgp neighbors (route type) community 1951
show ipv6 bgp neighbors regexp 1953
show ipv6 bgp regexp 1954
show ipv6 bgp summary 1955
shutdown (BGP) 1956
timers bgp 1957
vrf 1958

Chapter 30 Routing Information Protocol 1959
default-metric 1965
distance (RIP) 1966
distribute-list (RIP) 1967
ip rip v2-broadcast 1969
network (RIP) 1970
redistribute (RIP) 1971
router rip 1972
router rip vrf 1973
show ip rip database 1974
show ip rip neighbors 1975
shutdown (RIP) 1976
timers basic (RIP) 1977
vrf definition 1978
vrf forwarding 1979

Chapter 31 IS-IS 1981
address-family 1992
bfd all-interfaces 1993
authentication mode 1994
authentication key 1995
isis authentication mode 1996
isis authentication key 1997
isis enable 1998
isis bfd 1999
isis hello-interval 2000
isis hello-multiplier 2001
isis ipv6 metric 2002
isis lsp-interval 2003
isis metric 2004
isis multi-topology 2005
isis network 2006
isis passive 2007
isis priority 2008

is-hostname 2009
is-type 2010
log-adjacency-changes (IS-IS) 2011
multi-topology 2012
net 2013
passive-interface (IS-IS) 2014
redistribute (IS-IS) 2015
router isis 2016
set-overload-bit 2017
show isis database 2018
show isis hostname 2020
show isis interface 2021
show isis neighbors 2024
show isis summary 2026
show isis topology 2027
shutdown (IS-IS) 2028
spf-interval 2029

Chapter 32 Multiprotocol Label Switching (MPLS) 2031

entry (Nexthop Group) 2041
ip decap-group 2042
ip route nexthop-group 2043
mpls ip 2045
mpls static 2046
nexthop-group 2048
show mpls route 2049
show mpls route summary 2050
show nexthop-group 2051
size (Nexthop Group) 2052
ttl (Nexthop Group) 2053
tunnel decap-ip (Decap Group) 2054
tunnel-source (Nexthop Group) 2055
tunnel type (Decap Group) 2056

Chapter 33 Bidirectional Forwarding Detection 2057

bfd all-interfaces 2064
bfd echo 2065
bfd interval 2066
bfd per-link 2067
bfd slow-timer 2068
ip ospf bfd 2069
ip pim bfd 2070
ip pim bfd-instance 2071
neighbor fall-over bfd 2072
show bfd neighbors 2073
vrrp bfd ip 2075

Chapter 34 Multicast Architecture 2077
clear ip mfib fastdrop 2086
clear ip mroute 2087
ip mfib activity polling-interval 2088
ip mfib cache-entries unresolved max 2089
ip mfib fastdrop 2090
ip mfib max-fastdrops 2091
ip mfib packet-buffers unresolved max 2092
ip multicast boundary 2093
ip mroute 2095
ip multicast multipath none 2096
ip multicast-routing 2097
show ip mfib 2098
show ip mfib software 2099
show ip mroute 2100
show ip mroute count 2101

Chapter 35 IGMP and IGMP Snooping 2103
clear ip igmp group 2123
clear ip igmp snooping counters 2124
clear ip igmp statistics 2125
ip igmp host-proxy 2126
ip igmp host-proxy report-interval 2129
ip igmp host-proxy version 2130
ip igmp last-member-query-count 2131
ip igmp last-member-query-interval 2132
ip igmp profile 2133
ip igmp query-interval 2135
ip igmp query-max-response-time 2136
ip igmp router-alert 2137
ip igmp snooping 2138
ip igmp snooping filter 2139
ip igmp snooping interface-restart-query 2140
ip igmp snooping querier 2141
ip igmp snooping querier address 2142
ip igmp snooping querier last-member-query-count 2143
ip igmp snooping querier last-member-query-interval 2144
ip igmp snooping querier max-response-time 2145
ip igmp snooping querier query-interval 2146
ip igmp snooping querier startup-query-count 2147
ip igmp snooping querier startup-query-interval 2148
ip igmp snooping querier version 2149
ip igmp snooping report-flooding 2150
ip igmp snooping report-flooding switch-port 2151
ip igmp snooping restart query-interval 2152
ip igmp snooping robustness-variable 2153

ip igmp snooping vlan 2154
ip igmp snooping vlan immediate-leave 2155
ip igmp snooping vlan max-groups 2156
ip igmp snooping vlan mrouter 2157
ip igmp snooping vlan querier 2158
ip igmp snooping vlan querier address 2160
ip igmp snooping vlan querier last-member-query-count 2161
ip igmp snooping vlan querier last-member-query-interval 2162
ip igmp snooping vlan querier max-response-time 2163
ip igmp snooping vlan querier query-interval 2164
ip igmp snooping vlan querier startup-query-count 2165
ip igmp snooping vlan querier startup-query-interval 2166
ip igmp snooping vlan querier version 2167
ip igmp snooping vlan report-flooding 2168
ip igmp snooping vlan report-flooding switch-port 2169
ip igmp snooping vlan static 2170
ip igmp startup-query-count 2171
ip igmp startup-query-interval 2172
ip igmp static-group 2173
ip igmp static-group acl 2174
ip igmp static-group range 2175
ip igmp version 2177
permit / deny 2178
range 2179
show ip igmp groups 2180
show ip igmp groups count 2181
show ip igmp host-proxy config-sanity 2182
show ip igmp host-proxy interface 2183
show ip igmp interface 2184
show ip igmp profile 2185
show ip igmp snooping 2186
show ip igmp snooping counters 2187
show ip igmp snooping counters ethdev-pams 2188
show ip igmp snooping groups 2189
show ip igmp snooping groups count 2192
show ip igmp snooping mrouter 2193
show ip igmp snooping querier 2194
show ip igmp snooping querier counters 2195
show ip igmp snooping querier membership 2196
show ip igmp snooping report-flooding 2197
show ip igmp static-groups 2198
show ip igmp static-groups acl 2199
show ip igmp static-groups group 2200
show ip igmp statistics 2201

Chapter 36 Protocol Independent Multicast 2203

ip pim anycast-rp 2215
ip pim border-router 2216
ip pim bsr-border 2217
ip pim bsr-candidate 2218
ip pim bsr-holdtime 2220
ip pim dr-priority 2221
ip pim join-prune-interval 2222
ip pim log-neighbor-changes 2223
ip pim neighbor-filter 2224
ip pim query-interval 2225
ip pim register-source 2226
ip pim rp-address 2227
ip pim rp-candidate 2229
ip pim sparse-mode 2231
ip pim sparse-mode sg-expiry-timer 2232
ip pim spt-threshold 2233
ip pim spt-threshold group-list 2234
ip pim ssm range 2235
show ip pim bsr 2236
show ip pim config-sanity 2237
show ip pim interface 2238
show ip pim neighbor 2240
show ip pim protocol counters 2241
show ip pim register-source 2242
show ip pim rp 2243
show ip pim rp-candidate 2244
show ip pim rp-hash 2245
show ip pim upstream joins 2246

Chapter 37 Multicast Source Discovery Protocol 2247

clear ip msdp sa-cache 2260
ip msdp cache-sa-state 2261
ip msdp default-peer 2262
ip msdp description 2263
ip msdp group-limit 2264
ip msdp keepalive 2265
ip msdp mesh-group 2266
ip msdp originator-id 2267
ip msdp peer 2268
ip msdp rejected-limit 2269
ip msdp sa-filter in 2270
ip msdp sa-filter out 2271
ip msdp sa-limit 2272
ip msdp shutdown 2273
ip msdp timer 2274
show ip msdp mesh-group 2275

show ip msdp peer 2276
show ip msdp pim sa-cache 2277
show ip msdp rpf-peer 2278
show ip msdp sa-cache 2279
show ip msdp sanity 2280
show ip msdp summary 2281

Chapter 38 Audio Video Bridging (AVB) 2283

msrp 2292
mrp leave-all-timer 2293
mrp leave-timer 2294
msrp streams load-file 2295
mvrp 2296
show msrp 2297
show msrp interfaces 2298
show msrp streams 2300
show mvrp 2303

Chapter 39 SNMP 2305

no snmp-server 2314
show snmp 2315
show snmp chassis 2316
show snmp community 2317
show snmp contact 2318
show snmp engineID 2319
show snmp group 2320
show snmp host 2321
show snmp location 2322
show snmp mib 2323
show snmp source-interface 2324
show snmp trap 2325
show snmp user 2326
show snmp view 2327
snmp-server chassis-id 2328
snmp-server community 2329
snmp-server contact 2330
snmp-server enable traps 2331
snmp-server engineID local 2332
snmp-server engineID remote 2333
snmp-server extension 2334
snmp-server group 2335
snmp-server host 2336
snmp-server location 2338
snmp-server source-interface 2339
snmp-server user 2340
snmp-server view 2341
snmp-server vrf 2342

snmp trap link-status 2343

Chapter 40 Latency Analyzer (LANZ) 2345

clear queue-monitor length statistics 2358
max-connections 2359
queue-monitor length (global configuration mode) 2360
queue-monitor length threshold (Arad and Petra) 2361
queue-monitor length thresholds (FM6000, Trident II, and Tomahawk) 2362
queue-monitor length global-buffer 2364
queue-monitor length global-buffer log 2365
queue-monitor length global-buffer thresholds 2366
queue-monitor length log 2367
queue-monitor length mirror 2368
queue-monitor length mirror destination 2369
queue-monitor streaming 2370
show queue-monitor length 2371
show queue-monitor length all 2372
show queue-monitor length cpu 2373
show queue-monitor length csv 2375
show queue-monitor length drops 2376
show queue-monitor length ethernet 2377
show queue-monitor length global-buffer 2379
show queue-monitor length limit 2380
show queue-monitor length statistics 2381
show queue-monitor length status 2382
show queue-monitor length tx-latency 2384
shutdown (queue-monitor-streaming configuration) 2385
tcpdump queue-monitor 2386

Chapter 41 VM Tracer 2387

allowed-vlan 2395
autovlan disable 2396
password (vmtracer mode) 2397
password (vmtracer-vxlan mode) 2398
show vmtracer all 2399
show vmtracer interface 2400
show vmtracer session 2401
show vmtracer session vcenter 2402
show vmtracer session vsm 2403
show vmtracer vm 2404
show vmtracer vm detail 2405
show vmtracer vnic counters 2406
show vmtracer vxlan segment 2407
show vmtracer vxlan vm 2409
url (vmtracer mode) 2410
url (vmtracer-vxlan mode) 2411
username (vmtracer mode) 2412

username (vmtracer-vxlan mode) 2413
vmtracer 2414
vmtracer session 2415
vxlan (vmtracer mode) 2416

Chapter 42 Path Tracer 2417

clear monitor reachability probe-statistics 2432
destination ip (Monitor Reachability Probe Transmitter) 2433
destination port (Monitor Reachability) 2434
hops (Monitor Reachability Probe Transmitter) 2435
ip protocol (Monitor Reachability Probe Transmitter) 2436
monitor reachability 2437
preserve streams 2438
probe receiver max-streams 2439
probe transmitter 2440
show monitor reachability 2441
show monitor reachability egress-streams 2442
show monitor reachability ingress-streams 2444
show monitor reachability interface-ttl-statistics 2445
show monitor reachability probe-statistics 2446
show monitor reachability probe-transmitter 2447
shutdown (Monitor Reachability) 2448
shutdown (Monitor Reachability Probe Transmitter) 2449
source interface (Monitor Reachability Probe Transmitter) 2450
source port (Monitor Reachability Probe Transmitter) 2451

Chapter 43 MapReduce Tracer 2453

clear monitor hadoop burst-counters 2466
clear monitor hadoop job-history 2467
cluster (Monitor Hadoop) 2468
description (Monitor Hadoop Cluster) 2470
interval (Monitor Hadoop Cluster) 2471
jobtracker (Monitor Hadoop Cluster) 2472
monitor hadoop 2474
show monitor hadoop 2475
show monitor hadoop cluster all 2476
show monitor hadoop cluster counters 2477
show monitor hadoop cluster history 2478
show monitor hadoop cluster history jobs 2479
show monitor hadoop cluster jobs 2480
show monitor hadoop cluster jobs <job number> 2481
show monitor hadoop cluster jobs counter 2482
show monitor hadoop cluster status 2483
show monitor hadoop cluster tasktracker 2484
show monitor hadoop counters 2485
show monitor hadoop history 2486
show monitor hadoop status 2487

show monitor hadoop tasktracker all 2488
show monitor hadoop tasktracker all counters 2489
show monitor hadoop tasktracker counters 2490
show monitor hadoop tasktracker jobs 2492
show monitor hadoop tasktracker running-tasks 2495
show monitor hadoop tasktracker running-tasks cluster job task 2499
show monitor hadoop tasktracker status 2501
show monitor hadoop traffic burst 2505
shutdown (Monitor-Hadoop) 2509
shutdown (Monitor Hadoop Cluster) 2510
tasktracker (Monitor Hadoop Cluster) 2511

Chapter 44 sFlow 2513

clear sflow counters 2519
platform petraA buffers mini-multicast 2520
sflow destination 2521
sflow enable 2522
sflow polling-interval 2523
sflow run 2524
sflow sample 2525
[no] sflow sample output interface 2526
sflow source 2527
sflow source-interface 2528
show sflow 2529
show sflow interfaces 2531

Chapter 45 OpenFlow 2533

bind interface (OpenFlow) 2545
bind mode (OpenFlow) 2546
bind vlan (OpenFlow) 2547
clear openflow statistics 2548
controller (OpenFlow) 2549
default-action (OpenFlow) 2550
description (OpenFlow) 2551
keepalive (OpenFlow) 2552
openflow 2553
profile (OpenFlow) 2554
routing recirculation-interface (OpenFlow) 2555
routing vlan (OpenFlow) 2556
shell-command allowed (OpenFlow) 2557
show openflow 2558
show openflow flows 2559
show openflow ports 2560
show openflow profiles 2561
show openflow queues 2564
show openflow statistics 2565
shutdown (Openflow) 2567

Chapter 46 DirectFlow 2569
action drop (DirectFlow-flow mode) 2579
action mirror (DirectFlow-flow mode) 2580
action output (DirectFlow-flow mode) 2581
action output interface cpu (DirectFlow-flow mode) 2582
action set (DirectFlow-flow mode) 2583
directflow 2584
flow (DirectFlow) 2585
match (DirectFlow-flow mode) 2586
priority (DirectFlow-flow mode) 2588
show directflow 2589
show directflow flows 2590
shutdown (DirectFlow) 2591
timeout (DirectFlow-flow mode) 2592

Chapter 1

Overview

Arista Networks features switches with high-density, non-blocking Ethernet ports that are controlled through an extensible, Linux-based, modular network operating system. The intended audience for this manual is network administrators who configure Arista switches. A working knowledge of network administration is assumed.

New Features

This guide may not describe the features added in the most recent EOS version. For information on undocumented features, consult the TOI documents here: https://eos.arista.com/toi/

Switch Platforms

A list of Arista switches and detailed information about each is available online here: http://www.arista.com/products/switches

Recently released switches may not appear in the list, but can be found in the most recent Release Notes (found under Active Releases here: https://www.arista.com/support/software-download).

Supported Features

For the complete supported-features list in the latest EOS release, see https://www.arista.com/en/support/product-documentation/supported-features.

For details on a specific release, please see the Release Notes (found under Active Releases here: https://www.arista.com/en/support/software-download).

Chapter 2

Initial Configuration and Recovery

This chapter describes initial configuration and recovery tasks. Subsequent chapters provide details about features introduced in this chapter.

This chapter contains these sections:

- Section 2.1: Initial Switch Access
- Section 2.2: Connection Management
- Section 2.3: Configure Session
- Section 2.4: Recovery Procedures
- Section 2.5: Session Management Commands

2.1 Initial Switch Access

Arista network switches provide two initial configuration methods:

- Zero Touch Provisioning (ZTP) configures the switch without user interaction (Section 2.1.1).
- Manual provisioning configures the switch through commands entered by a user through the CLI (Section 2.1.2).

2.1.1 Zero Touch Provisioning

Zero Touch Provisioning (ZTP) configures a switch without user intervention by downloading a startup configuration file (***startup-config***) or a boot script from a location specified by a DHCP server. Section 6.5.2 describes network tasks required to set up ZTP.

The switch enters ZTP mode when it boots if flash memory does not contain ***startup-config***. It remains in ZTP mode until a user cancels ZTP mode, or until the switch retrieves a ***startup-config*** or a boot script. After downloading a file through ZTP, the switch reboots again, using the retrieved file.

Security Considerations

The ZTP process cannot distinguish an approved DHCP server from a rogue DHCP server. For secure provisioning, you must ensure that only approved DHCP servers are able to communicate with the switch until after the ZTP process is complete. Arista also recommends validating the EOS image on your ZTP server by confirming that its MD5 checksum matches the MD5 checksum that can be found on the EOS download page of the Arista website.

On a UNIX server, the **md5sum** command calculates this checksum:

```
% md5sum EOS.swi
3bac45b96bc820eb1d10c9ee33108a25  EOS.swi
```

This command is also available on Arista switches from the CLI or from within the Bash shell.

```
switch#bash md5sum /mnt/flash/EOS-4.18.0F.swi
73435f0db3af785011f88743f4c01abd  /mnt/flash/EOS-4.18.0F.swi
switch#

[admin@switch ~]$ md5sum /mnt/flash/EOS-4.18.0F.swi
73435f0db3af785011f88743f4c01abd /mnt/flash/EOS-4.18.0F.swi
[admin@switch ~]$
```

To provision the switch through Zero Touch Provisioning:

Step 1 Mount the switch in its permanent location.

Step 2 Connect at least one management or Ethernet port to a network that can access the DHCP server and the configuration file.

Step 3 Provide power to the switch.

ZTP provisioning progress can be monitored through the console port. Section 2.1.2.1 provides information for setting up the console port. Section 2.1.2.2 provides information for monitoring ZTP progress and canceling ZTP mode.

2.1.2 Manual Provisioning

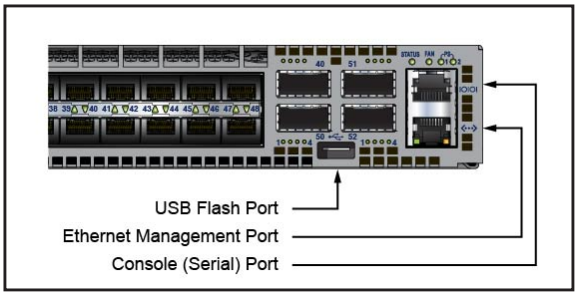
Initial manual switch provisioning requires the cancellation of ZTP mode, the assignment of an IP address to a network port, and the establishment of an IP route to a gateway. Initial provisioning is performed through the serial console and Ethernet management ports.

- The console port is used for serial access to the switch. These conditions may require serial access:
  - management ports are not assigned IP addresses
  - the network is inoperable
  - the password for the user's log on is not available
  - the password to access the enable mode is not available
- The Ethernet management ports are used for out-of-band network management tasks. Before using a management port for the first time, an IP address must be assigned to that port.

2.1.2.1 Console Port

The console port is a serial port located on the front of the switch. Figure 2-1 shows the console port on the DCS-7050T-64 switch. Use a serial or RS-232 cable to connect to the console port. The accessory kit also includes an RJ-45 to DB-9 adapter cable for connecting to the switch.

Figure 2-1: Switch Ports



Port Settings

Use these settings when connecting the console port:

- 9600 baud
- no flow control
- 1 stop bit
- no parity bits
- 8 data bits

Admin Username

The initial configuration provides one username, **admin,** that is not assigned a password. When using the **admin** username without a password, you can only log into the switch through the console port. After a password is assigned to the **admin** username, it can log into the switch through any port.

The **username** command assigns a password to the specified username.

Example

- This command assigns the password **pxq123** to the **admin** username:

```
switch(config)#username admin secret pxq123
switch(config)#
```

New and altered passwords that are not saved to the startup configuration file are lost when the switch is rebooted.

2.1.2.2 Canceling Zero Touch Provisioning

Zero Touch Provisioning (ZTP) installs a ***startup-config*** file from a network location if flash memory does not contain a ***startup-config*** when the switch reboots. Canceling ZTP is required if the switch cannot download a ***startup-config*** or boot script file.

When the switch boots without a ***startup-config*** file, it displays the following message through the console port:

```
No startup-config was found.

The device is in Zero Touch Provisioning mode and is attempting to
download the startup-config from a remote system. The device will not
be fully functional until either a valid startup-config is downloaded
from a remote system or Zero Touch Provisioning is cancelled. To cancel
Zero Touch Provisioning, login as admin and type 'zerotouch cancel'
at the CLI.

localhost login:
```

To cancel ZTP mode, log into the switch with the **admin** password, then enter the **zerotouch cancel** command. The switch immediately boots without installing a ***startup-config*** file.

```
localhost login: admin
admin
localhost>Apr 15 21:28:21 localhost ZeroTouch: %ZTP-5-DHCP_QUERY: Sending DHCP
request on  [ Ethernet10, Ethernet13, Ethernet14, Ethernet17, Ethernet18,
Ethernet21, E-thernet22, Ethernet23, Ethernet24, Ethernet7, Ethernet8,
Ethernet9, Management1, Management2 ]
Apr 15 21:28:51 localhost ZeroTouch: %ZTP-5-DHCP_QUERY_FAIL: Failed to get a
valid DHCP response
Apr 15 21:28:51 localhost ZeroTouch: %ZTP-5-RETRY: Retrying Zero Touch
Provisioning from the beginning (attempt 1)
Apr 15 21:29:22 localhost ZeroTouch: %ZTP-5-DHCP_QUERY: Sending DHCP request on
[ Ethernet10, Ethernet13, Ethernet14, Ethernet17, Ethernet18, Ethernet21,
Ethernet22, Ethernet23, Ethernet24, Ethernet7, Ethernet8, Ethernet9,
Management1, Management2 ]


localhost>zerotouch cancel
zerotouch cancel
localhost>Apr 15 21:29:39 localhost ZeroTouch: %ZTP-5-CANCEL: Canceling Zero
Touch Provisioning
Apr 15 21:29:39 localhost ZeroTouch: %ZTP-5-RELOAD: Rebooting the system
Broadcast messagStopping sshd: [  OK  ]
watchdog is not running
SysRq : Remount R/O
Restarting system
ø

Aboot 1.9.0-52504.EOS2.0
Press Control-C now to enter Aboot shell
```

To avoid entering ZTP mode on subsequent reboots, create a ***startup-config*** file as described in step 8 of Section 2.1.2.3.

2.1.2.3 Ethernet Management Port

Arista switches provide one or more Ethernet management ports for configuring the switch and managing the network out of band. Figure 2-1 shows the location of the Ethernet management ports on a DCS-7050T-64 switch. Only one port is required to manage the switch.

You can access the Ethernet management port(s) remotely over a common network or locally through a directly connected PC. Before you can access the switch through a remote connection, an IP address and a static route to the default gateway are required. On a modular switch with dual supervisors, a virtual IP address can also be configured to access the management port on whichever supervisor is active.

Assigning a Virtual IP Address to Access the Active Ethernet Management Port

On modular switches with dual supervisors, this procedure assigns a virtual IP address which will connect to the Ethernet management port of the active supervisor. (To assign a physical IP address to an individual Ethernet management port, see Assigning an IP Address to a Specific Ethernet Management Port below.)

Step 1 Connect a PC or terminal server to the console port. Use the settings listed in Section 2.1.2.1 under Port Settings.

Step 2 Type **admin** at the login prompt to log into the switch. Initial login through the console port does not require a password.

```
Arista EOS
switch login:admin
Last login: Fri Apr 9 14:22:18 on Console

switch>
```

Step 3 Type **enable** at the command prompt to enter Privileged EXEC mode.

```
switch>enable
switch#
```

Step 4 Type **configure terminal** (or **config**) to enter global configuration mode.

```
switch#configure terminal
switch(config)#
```

Step 5 Type **interface management 0** to enter interface configuration mode for the virtual interface which accesses management port 1 on the currently active supervisor.

```
switch(config)#interface management 0
switch(config-if-Ma0)#
```

Step 6 Type **ip address**, followed by the desired address, to assign a virtual IP address for access to the active management port.

This command assigns IP address 10.0.2.5 to management port 0.

```
switch(config-if-Ma0)#ip address 10.0.2.5/24
```

Step 7 Type **exit** at both the interface configuration and global configuration prompts to return to Privileged EXEC mode.

```
switch(config-if-Ma0)#exit
switch(config)#exit
switch#
```

Step 8 Type **write** (or **copy running-config startup-config)** to save the new configuration to the ***startup-config*** file.

```
switch# write
switch#
```

Assigning an IP Address to a Specific Ethernet Management Port

This procedure assigns an IP address to a specific Ethernet management port:

Step 1 Connect a PC or terminal server to the console port. Use the settings listed in Section 2.1.2.1 under Port Settings.

Step 2 Type **admin** at the login prompt to log into the switch. The initial login does not require a password.

```
Arista EOS
switch login:admin
Last login: Fri Apr 9 14:22:18 on Console

switch>
```

Step 3 Type **enable** at the command prompt to enter Privileged EXEC mode.

```
switch>enable
switch#
```

Step 4 Type **configure terminal** (or **config**) to enter global configuration mode.

```
switch#configure terminal
```

Step 5 Type **interface management 1** to enter interface configuration mode. (Any available management port can be used in place of management port 1.)

```
switch(config)#interface management 1
switch(config-if-Ma1)#
```

Step 6 Type **ip address**, followed by the desired address, to assign an IP address to the port.

This command assigns the IP address 10.0.2.8 to management port 1.

```
switch(config-if-Ma1)#ip address 10.0.2.8/24
```

Step 7 Type **exit** at both the interface configuration and global configuration prompts to return to Privileged EXEC mode.

```
switch(config-if-Ma1)#exit
switch(config)#exit
switch#
```

Step 8 Type **write** (or **copy running-config startup-config)** to save the new configuration to the ***startup-config*** file.

```
switch# write
switch#
```

Configuring a Default Route to the Gateway

This procedure configures a default route to a gateway located at 10.0.2.1.

Step 1 Enter global configuration mode.

```
switch>enable
switch#configure terminal
```

Step 2 Create a static route to the gateway with the IP route command.

```
switch(config)#ip route 0.0.0.0/0 10.0.2.1
```

Step 3 Save the new configuration.

```
switch#write
switch#
```

2.2 Connection Management

The switch supports three connection methods:

- console
- SSH
- Telnet

The switch always enables console and SSH. Telnet is disabled by default.

Management commands place the switch in a configuration mode for changing session connection parameters.

Examples

- The **management console** command places the switch in console management mode:

```
switch(config)#management console
switch(config-mgmt-console)#
```

- The **management ssh** command places the switch in SSH management mode:

```
switch(config)#management ssh
switch(config-mgmt-ssh)#
```

- The **management telnet** command places the switch in Telnet management mode:

```
switch(config)#management telnet
switch(config-mgmt-telnet)#
```

- The **exit** command returns the switch to global configuration mode:

```
switch(config-mgmt-ssh)#exit
switch(config)#
```

The **idle-timeout** commands shown below configure the idle-timeout period for the connection type being configured. The idle timeout is the interval that the connection waits after a user's most recent command before shutting down the connection. Automatic connection timeout is disabled by setting the idle-timeout to zero, which is the default setting.

Examples

- This **idle-timeout (SSH Management)** command configures an SSH idle-timeout period of three hours.

```
switch(config)#management ssh
switch(config-mgmt-ssh)#idle-timeout 180
```

- This **idle-timeout (Telnet Management)** command disables automatic connection timeout for telnet connections.

```
switch(config)#management telnet
switch(config-mgmt-telnet)#idle-timeout 0
```

The **shutdown (Telnet Management)** command enables and disables Telnet on the switch.

Examples

- These commands enable Telnet.

```
switch(config)#management telnet
switch(config-mgmt-telnet)#no shutdown
```

- These commands disable Telnet.

```
switch(config)#management telnet
switch(config-mgmt-telnet)#shutdown
```

2.3 Configure Session

The command **configure session** allows users to issue configuration sessions as CLI commands that do not take effect immediately. Each **configure session** is saved with a unique name. A session canbe entered, modified and exited at any time without impacting the currently running system configuration.

A session is defined as a collection of configuration changes that are grouped together.

When a session is committed, the configuration that was modified during the session is copied into ***running-config***. A session can be aborted or removed, thereby removing the session completely and freeing up memory used by the session. The user must explicitly request that the changes in a deferred session be applied to the configuration of the router by entering a **commit** command and exiting the mode. Alternately, the user may abandon the changes by entering an **abort** command.

Configuration sessions are used to make sets of changes, after verifying that there are no CLI errors. Configuration sessions allow the administrator to pre-provision a group of CLI commands in a named session, then execute each configuration session at specified times.

This chapter contains the following sections:

- Section 2.3.1: Configuration Session
- Section 2.3.2: Configure Replace

2.3.1 Configuration Session

The **configure session** command allows users to make a series of configuration changes in a temporary location and commit them to ***running-config*** at once by issuing the **commit** command.

- **configure session <name of session>** and **running-config** — The user enters a session (versus **configure terminal** in the case where configuration sessions are not used). If a session name is not specified, a system named session is created. A snapshot of the current **running-config** is copied into the session's data structure as the basis of further configuration changes.
- CLI configuration commands — User can run any configuration commands inside the session.
- **rollback clean-config** — User can run **rollback** command to revert the session's configuration to the factory-default configuration (or clean configuration).
- **show session-config** — User can run **show session-config** to show the session's configuration, which will be the future **running-config** once committed.
- **commit** — User issues **commit** to commit the changes, which will replace the current **running-config**.
- **abort** — To abort the session and throw away all changes.
- **exit** — User can **exit** from the session, and later return to the same session by running **configure session <name>** again.
- For named session — More than one CLI instance can enter the same session and make changes to the session configuration. Once the session is committed in any of the CLIs, no other CLI can commit or make any other changes in that session.

2.3.2 Configure Replace

The command **configure replace <URL>** replaces the current ***running-config*** with the configuration saved in **<URL>**. By default, **configure replace <URL>** will replace ***running-config*** only if the configuration in **<URL>** loads without errors. The command **configure replace <URL> ignore-errors** forces the operation in spite of errors.

Note The command **copy <URL> running-config** was typically used to apply a saved configuration file to the system, and append that configuration to the current ***running-config*** (in lieu of replacing it). However, it is recommended the user uses the CLI command **configure replace <URL>** to streamline the process of deterministically restoring the system back to a known good configuration.

The normal workflow internally uses a configuration session to perform the replace.

2.3.3 Configuration CLI

In the CLI, execute the following configuration steps to create a configuration session.

Step 1 configure session [*name of session***]**

Create or enter a session. If a name is not specified, it is automatically generated. The user is put in the session configuration mode and the prompt will change to show the first six characters of the session name. Designating the name of a session is optional. When *name of session* is not specified, a unique name is assigned.

no configure session *name of session*

Delete the specified configuration session. Designating the name of a session is required.

Step 2 commit

Commit the changes made in the session. This command must be issued from within the session configuration mode.

abort

Abort the session, which is the same as deleting it. This command must be issued from within the session configuration mode.

Step 3 rollback clean-config

Revert configuration in the session to the clean, factory-default configuration. This command must be issued from within the session configuration mode.

Step 4 service configuration session max completed *number*

Set a limit on the maximum number of committed sessions that are saved.

Step 5 service configuration session max pending *number*

Set a limit on the maximum number of uncommitted sessions that can be outstanding.

2.3.4 Show Commands

2.3.4.1 show configuration sessions [detail]

This command displays the following information about the sessions that exist in the system:

- The name of each session and its state (completed, pending, aborted, etc.) are displayed.
- If a user has currently entered the session, the user name and the associated terminal are also shown.
- With the detail flag, the process ID of the CLI process that is using the session is also displayed.
- Commit Time Left shows the countdown until the configuration is reverted back. It shows data only when the configuration session command **commit timer** *hh:mm:ss* has been used.

Note An asterisk (*) indicates that the user running the show command is currently in the marked session.

Example

```
switch(config-s-s2)#show configuration sessions detail
Maximum number of completed sessions: 1
Maximum number of pending sessions: 5


  Name       State                User      Terminal      PID    Commit Time Left

  --------- ----------------------- ---------- -------------- ---------
----------------

  session10    completed

  session20   pendingCommitTimer                                              2m20s
```

In order for the commit to persist, confirm the **commit** before the timer expires.This is used to prevent configurations that would lock you out of the device due to loss of connectivity or when performing security changes such as user account, password, RADIUS, or TACACS changes.

Example

This example displays how to set the commit timer.

```
switch(config)#configure session routingChanges
switch(config-s-routin)#ip route 10.2.5.0/24 10.0.0.1
switch(config-s-routin)#commit timer 00:02:00
```

This show command displays the configuration sessions.

```
switch#show configuration sessions
Maximum number of completed sessions: 1
Maximum number of pending sessions: 5

  Name              State                     User    Terminal

  -------------- ----------------------- ---------- --------

  routingChanges    pendingCommitTimer
```

This show command displays detailed information for the configuration sessions.

```
switch#show configuration sessions detail

Maximum number of completed sessions: 1

Maximum number of pending sessions: 5


  Name            State                 User      Terminal      PID      Commit Time Left    Description

  -------------- ----------------------- ---------- -------------- --------- ----------------------
-------------------------------------------

  cfg0             completed                                                                  config
replace of commitTimerCheckPointConfig

  routingChanges    pendingCommitTimer                                                    1m49s
```

Note You cannot enter session routingChanges, when the switch is in pendingCommitTimer state.

```
Arista(config)#configure session routingChanges commit
Arista(config)#show configuration sessions detail
Maximum number of completed sessions: 1
Maximum number of pending sessions: 5

Name             State          User      Terminal      PID    Commit Time Left

  -------------- -------------- ---------- -------------- ---------
---------------

  routingChanges    completed
switch(config)#
```

2.3.4.2 show session-config [diff]

This command must be issued from within a session. It shows the following:

- The session configuration, including the changes made in the session.
- The diff flag shows the differences with the ***running-config***, which helps highlight the changes made in the session.

Example 1

```
Arista(config-s-s2)#show session-config
! Command: show session-configuration named s2
ip dhcp smart-relay global
!
transceiver qsfp default-mode 4x10G
!
ip pim bsr-candidate Loopback0 224.0.0.0/4 priority 64 hashmask 30 interval 60
!
hostname Arista
ip host one 1.1.1.1
!
no aaa root
!
spanning-tree mode mstp
!
interface Ethernet1
!
interface Ethernet2
!
interface Ethernet3
!
interface Ethernet4
!
interface Ethernet5
!
interface Ethernet6
!
no ip routing
!
!
end
```

Example 2

```
Arista(config-s-s2)#show session-config diff
--- system:/running-config
+++ session:/s2
@@ -5,6 +5,7 @@
ip pim bsr-candidate Loopback0 224.0.0.0/4 priority 64 hashmask 30 interval 60
!
hostname Arista
+ip host one 1.1.1.1
!
no aaa root
!
```

2.3.4.3 show session-config name *<name of session>*

Show the session configuration of the named session.

Example

```
Arista#show session-config named s1
! Command: show session-configuration named s1
ip dhcp smart-relay global
!
transceiver qsfp default-mode 4x10G
!
ip pim bsr-candidate Loopback0 224.0.0.0/4 priority 64 hashmask 30 interval 60
!
hostname Arista
!
no aaa root
!
spanning-tree mode mstp
!
interface Ethernet1
!
interface Ethernet2
!
interface Ethernet3
!
interface Ethernet4
!
interface Ethernet5
!
interface Ethernet6
!
no ip routing
!
!
end
```

2.4 Recovery Procedures

These sections describe switch recovery procedures:

- Section 2.4.1: Removing the Enable Password from the Startup Configuration
- Section 2.4.2: Reverting the Switch to the Factory Default Startup Configuration
- Section 2.4.3: Restoring the Factory Default EOS Image and Startup Configuration
- Section 2.4.4: Restoring the Configuration and Image from a USB Flash Drive

The first three procedures require Aboot Shell access through the console port. If the console port is not accessible, use the last procedure in the list to replace the configuration file through the USB Flash Drive.

Boot Loader – Aboot describes the switch booting process and includes descriptions of the Aboot shell, Aboot boot loader, and required configuration files.

2.4.1 Removing the Enable Password from the Startup Configuration

The **enable password** controls access to Privileged EXEC mode. To prevent unauthorized disclosure, the switch stores the **enable password** as an encrypted string that it generates from the clear-text password. When the switch authentication mode is local and an **enable password** is configured, the CLI prompts the user to enter the clear-text password after the user types **enable** at the EXEC prompt.

The ***startup-config*** file stores the encrypted **enable password** to ensure that the switch loads it when rebooting. If the text version of the **enable password** is lost or forgotten, access to enable mode is restored by removing the encrypted **enable password** from the startup configuration file.

This procedure restores access to enable mode without changing any other configuration settings.

Step 1 Access the Aboot shell:

a Power cycle the switch by successively removing and restoring access to its power source.

b Type **Ctrl-C** when prompted, early in the boot process.

c Enter the Aboot password, if prompted.
If the Aboot password is unknown, refer to Section 2.4.3: Restoring the Factory Default EOS Image and Startup Configuration for instructions on reverting all flash directory contents to the factory default, including the startup configuration and EOS image.

Step 2 Change the active directory to /mnt/flash directory.

```
Aboot#cd /mnt/flash
```

Step 3 Open the startup-config file in vi.

```
Aboot#vi startup-config
```

Step 4 Remove the enable password line.

This is an example of an enable password line:

```
enable secret 5 $1$dBXo2KpF$Pd4XYLpI0ap1ZaU7glG1w/
```

Step 5 Save the changes and exit vi.

Step 6 Exit Aboot. This boots the switch.

```
Aboot#exit
```

2.4.2 Reverting the Switch to the Factory Default Startup Configuration

The ***startup-config*** file contains configuration parameters that the switch uses during a boot. Parameters that do not appear in ***startup-config*** are set to their factory defaults when the switch reloads. The process requires the Aboot password if Aboot is password protected.

This procedure reverts EOS configuration settings to the default state through bypassing the ***startup-config*** file during a switch boot.

Step 1 Access the Aboot shell through the console port:

a Type **reload** at the Privileged EXEC prompt.

b Type **Ctrl-C** when prompted, early in the boot process.

c Enter the Aboot password, if prompted.
If the Aboot password is unknown, refer to Section 2.4.3: Restoring the Factory Default EOS Image and Startup Configuration for instructions on reverting all flash directory contents to the factory default, including ***startup-config*** and EOS image.

Step 2 Change the active directory to **/mnt/flash** directory.

```
Aboot#cd /mnt/flash
```

Step 3 Rename the startup configuration file.

```
Aboot#mv startup-config startup-config.old
```

Step 4 Exit Aboot. This boots the switch

```
Aboot#exit
```

Step 5 Cancel Zero Touch Provisioning (ZTP). Refer to Section 2.1.2.2: Canceling Zero Touch Provisioning for instructions.

If ZTP is not canceled, the switch either:

- boots, using the ***startup-config*** file or boot script that it obtains from the network, or
- remains in ZTP mode if the switch is unable to download a ***startup-config*** file or boot script.

Step 6 Configure the **admin** and **enable** passwords.

```
switch>enable
switch#configure terminal
switch(config)#enable secret xyz1
switch(config)#username admin secret abc41
```

Step 7 Save the new ***running-config*** to the startup configuration file.

```
switch#write
```

Step 8 (Optional) Delete the old startup configuration file.

```
switch#delete startup-config.old
```

After ZTP is canceled, the switch reboots, using the factory default settings. To avoid entering ZTP mode on subsequent reboots, create a ***startup-config*** file before the next switch reboot.

2.4.3 Restoring the Factory Default EOS Image and Startup Configuration

A **fullrecover** command removes all internal flash contents (including configuration files, EOS image files, and user files), then restores the factory default EOS image and ***startup-config***. A subsequent installation of the current EOS image may be required if the default image is outdated. This process requires Aboot shell access through the console port.

This procedure restores the factory default EOS image and startup configuration.

Step 1 Access the Aboot shell through the console port:

a Type **reload** at the Privileged EXEC prompt.

b Type **Ctrl-C** when prompted, early in the boot process.

c Enter the Aboot password, if prompted.
If the Aboot password is not known, enter an empty password three times, after which the CLI displays:

```
Type "fullrecover" and press Enter to revert /mnt/flash to factory default
state, or just press Enter to reboot:
```

d Type **fullrecover** and go to step 4.

Step 2 Type **fullrecover** at the Aboot prompt.

```
Aboot#fullrecover
```

Aboot displays this warning:

```
All data on /mnt/flash will be erased; type "yes" and press Enter to proceed, or
just press Enter to cancel:
```

Step 3 Type **yes** and press **Enter**.

The switch performs these actions:

- erases the contents of /mnt/flash
- writes new boot-config, startup-config, and EOS.swi files to /mnt/flash
- returns to the Aboot prompt

Step 4 Exit Aboot. This boots the switch.

```
Aboot#exit
```

The serial console settings are restored to their default values (9600/N/8/1/N).

Step 5 Reconfigure the console port if non-default settings are required.

Step 6 Cancel Zero Touch Provisioning (ZTP). Refer to Section 2.1.2.2: Canceling Zero Touch Provisioning for instructions.

If ZTP is not canceled, the switch either:

- boots, using the ***startup-config*** file or boot script that it obtains from the network, or
- remains in ZTP mode if the switch is unable to download a ***startup-config*** file or boot script.

After ZTP is canceled, the switch reboots, using the factory default settings. To avoid entering ZTP mode on subsequent reboots, create a ***startup-config*** file before the next switch reboot.

2.4.4 Restoring the Configuration and Image from a USB Flash Drive

The USB flash drive port can be used to restore an original configuration when you cannot establish a connection to the console port. This process removes the contents of the internal flash drive, restores the factory default configuration, and installs a new EOS image from the USB flash drive.

This procedure restores the factory default configuration and installs an EOS image stored on a USB flash drive.

Step 1 Prepare the USB flash drive:

a Verify the drive is formatted with MS-DOS or FAT file system.

Most USB drives are pre-formatted with a compatible file system.

b Create a text file named **fullrecover** on the USB flash drive.

The filename does not have an extension. The file may be empty.

c Create a text file named **boot-config**.

The last modified timestamp of the **boot-config** file on the USB flash must differ from the timestamp of the **boot-config** file on the switch.

d Enter this line in the new **boot-config** file on the USB flash:

```
SWI=flash:EOS.swi
```

e Copy an EOS image file to the flash drive. Rename it **EOS.swi** if it has a different file name.

For best results, the flash drive should contain only these three files, because the procedure copies all files and directories on the USB flash drive to the switch.

- fullrecover
- boot-config
- EOS.swi

Step 2 Insert the USB flash drive into the USB flash port on the switch, as shown in Figure 2-1.

Step 3 Connect a terminal to the console port and configure it with the default terminal settings (9600/N/8/1) to monitor progress messages on the console.

Step 4 Power up or **reload** the switch.

The switch erases internal flash contents and copies the files from the USB flash drive to internal flash. The switch then boots automatically.

Step 5 Cancel Zero Touch Provisioning (ZTP). Refer to Section 2.1.2.2: Canceling Zero Touch Provisioning for instructions.

If ZTP is not canceled, the switch either:

- boots, using the ***startup-config*** file or boot script that it obtains from the network, or
- remains in ZTP mode if the switch is unable to download a ***startup-config*** file or boot script.

After ZTP is canceled, the switch reboots using the factory default settings. To avoid entering ZTP mode on subsequent reboots, create a ***startup-config*** file before the next switch reboot.

2.5 Session Management Commands

Global Configuration Commands

- management api http-commands
- management console
- management ssh
- management telnet
- management xmpp

Management Configuration Commands

- domain (XMPP Management)
- idle-timeout (Console Management)
- idle-timeout (SSH Management)
- idle-timeout (Telnet Management)
- protocol http (API Management)
- protocol https (API Management)
- protocol https certificate (API Management)
- server (XMPP Management)
- session privilege (XMPP Management)
- shutdown (API Management)
- shutdown (Telnet Management)
- shutdown (XMPP Management)
- switch-group (XMPP Management)
- username (XMPP Management)
- vrf (API Management)
- vrf (XMPP Management)
- xmpp send
- xmpp session

Display Commands

- show inventory
- show xmpp neighbors
- show xmpp status
- show xmpp switch-group

domain (XMPP Management)

The domain command configures the switch's XMPP domain name. Only messages using a domain matching the locally configured one are accepted by the XMPP client. The switch's domain name is used if none is specified.

Management over XMPP is disabled by default. To enable it, you must provide the location of the server along with the domain, username and password for the switch.

Arista recommends configuring the XMPP domain before the username, because it will provide shortcuts for the **switch-group** and **username** so they can be configured without the domain attached to it (e.g., USERNAME instead of USERNAME@DOMAIN).

The **no domain** and **default domain** commands delete the domain name by removing the **domain** command from ***running-config***.

Command Mode

Mgmt-xmpp Configuration

Command Syntax

```
domain string
no domain
default domain
```

Parameters

- *string* domain name (text string)

Example

- This command configures *test.aristanetworks.com* as the switch's domain name.

```
switch(config)#management xmpp
test1(config-mgmt-xmpp)#server arista-xmpp
test1(config-mgmt-xmpp)#domain test.aristanetworks.com
test1(config-mgmt-xmpp)#username test1@test.aristanetworks.com password 0 arista
test1(config-mgmt-xmpp)#no shutdown
```

- This command removes the domain name from the XMPP *configuration*.

```
switch(config-mgmt-xmpp)#no domain
switch(config-mgmt-xmpp)#
```

idle-timeout (Console Management)

The **idle-timeout (Console Management)** command configures the idle-timeout period for console connection sessions. The idle timeout is the interval that the connection waits after a user's most recent command before shutting down the connection. Automatic connection timeout is disabled by setting the idle-timeout to zero, which is the default setting.

The **no idle-timeout** and **default idle-timeout** commands disables the automatic connection timeout by removing the **idle-timeout** statement from ***running-config***.

Command Mode

Mgmt-console

Command Syntax

```
idle-timeout idle_period
no idle-timeout
default idle-timeout
```

Parameters

- *idle_period* session idle-timeout length. Options include:
  - 0 Automatic connection timeout is disabled
  - <***1*** to ***86400***> Automatic timeout period (minutes).

Example

- These commands configure a console idle-timeout period of three hours, then return the switch to global configuration mode.

```
switch(config)#management console
switch(config-mgmt-console)#idle-timeout 180
switch(config-mgmt-console)#exit
switch(config)#
```

- These commands disable automatic connection timeout.

```
switch(config)#management console
switch(config-mgmt-console)#idle-timeout 0
switch(config-mgmt-console)#
```

idle-timeout (SSH Management)

The **idle-timeout (SSH Management)** command configures the idle-timeout period for SSH connection sessions. The idle timeout is the interval that the connection waits after a user's most recent command before shutting down the connection. Automatic connection timeout is disabled by setting the idle-timeout to zero, which is the default setting.

The **no idle-timeout** and **default idle-timeout** commands disables the automatic connection timeout by removing the **idle-timeout** statement from ***running-config***.

Command Mode

Mgmt-ssh Configuration

Command Syntax

```
idle-timeout idle_period
no idle-timeout
default idle-timeout
```

Parameters

- *idle_period* session idle-timeout length. Options include:
  - 0 Automatic connection timeout is disabled
  - <***1*** to ***86400***> Automatic timeout period (minutes).

Example

- These commands configure an ssh idle-timeout period of three hours, then return the switch to global configuration mode.

```
switch(config)#management ssh
switch(config-mgmt-ssh)#idle-timeout 180
switch(config-mgmt-ssh)#exit
switch(config)#
```

- These commands disable automatic connection timeout.

```
switch(config)#management ssh
switch(config-mgmt-ssh)#idle-timeout 0
switch(config-mgmt-ssh)#
```

idle-timeout (Telnet Management)

The **idle-timeout (Telnet Management)** command configures the idle-timeout period for Telnet connection sessions. The idle timeout is the interval that the connection waits after a user's most recent command before shutting down the connection. Automatic connection timeout is disabled by setting the idle-timeout to zero, which is the default setting.

The **no idle-timeout** and **default idle-timeout** commands disables the automatic connection timeout by removing the **idle-timeout** statement from ***running-config***.

Command Mode

Mgmt-telnet

Command Syntax

```
idle-timeout idle_period
no idle-timeout
default idle-timeout
```

Parameters

- *idle_period* session idle-timeout length. Options include:
  - 0 Automatic connection timeout is disabled
  - <***1*** to ***86400***> Automatic timeout period (minutes).

Example

- These commands configure a telnet idle-timeout period of three hours, then return the switch to global configuration mode.

```
switch(config)#management telnet
switch(config-mgmt-telnet)#idle-timeout 180
switch(config-mgmt-telnet)#exit
switch(config)#
```

- These commands disable automatic connection timeout.

```
switch(config)#management telnet
switch(config-mgmt-telnet)#idle-timeout 0
switch(config-mgmt-telnet)#
```

management api http-commands

The **management api http-commands** command places the switch in mgmt-API configuration mode.

The **no management api http-commands** and **default management api http-commands** commands delete mgmt-api-http-command configuration mode statements from ***running-config***.

Mgmt-api-http-cmds configuration mode is not a group change mode; ***running-config*** is changed immediately upon entering commands. Exiting mgmt-api-http-cmds configuration mode does not affect ***running-config***. The **exit** command returns the switch to global configuration mode.

Command Mode

Global Configuration

Command Syntax

```
management api http-commands
no management api http-commands
default management api http-commands
```

Commands Available in Mgmt-API Configuration Mode

- **protocol http (API Management)**
- **protocol https (API Management)**
- **protocol https certificate (API Management)**
- **shutdown (API Management)**
- **vrf (API Management)**

Example

- This command places the switch in mgmt-api-http-cmds configuration mode.

```
switch(config)#management api http-commands
switch(config-mgmt-api-http-cmds)#
```

- This command returns the switch to global management mode.

```
switch(config-mgmt-api-http-cmds)#exit
switch(config)#
```

management console

The **management console** command places the switch in mgmt-console configuration mode to adjust the idle-timeout period for console connection sessions. The idle-timeout period determines the inactivity interval that terminates a connection session.

The **no management console** and **default management console** commands delete mgmt-console configuration mode statements from ***running-config***.

Mgmt-console configuration mode is not a group change mode; ***running-config*** is changed immediately upon entering commands. Exiting mgmt-console configuration mode does not affect ***running-config***. The **exit** command returns the switch to global configuration mode.

Command Mode

Global Configuration

Command Syntax

```
management console
no management console
default management console
```

Commands Available in mgmt-console Configuration Mode

- **idle-timeout (Console Management)**

Example

- This command places the switch in mgmt-console configuration mode:

```
switch(config)#management console
switch(config-mgmt-console)#
```

- This command returns the switch to global management mode:

```
switch(config-mgmt-console)#exit
switch(config)#
```

management ssh

The **management ssh** command places the switch in mgmt-ssh configuration mode to adjust SSH session connection parameters.

The **no management ssh** and **default management ssh** commands delete the mgmt-ssh configuration mode statements from ***running-config***.

Mgmt-ssh configuration mode is not a group change mode; ***running-config*** is changed immediately upon entering commands. Exiting mgmt-ssh configuration mode does not affect ***running-config***. The **exit** command returns the switch to global configuration mode.

Command Mode

Global Configuration

Command Syntax

```
management ssh
no management ssh
default management ssh
```

Commands Available in Mgmt-ssh Configuration Mode

- authentication mode (Management-SSH)
- cipher (Management-SSH)
- fips restrictions (Management-SSH)
- hostkey (Management-SSH)
- idle-timeout (Management-SSH)
- ip access group (Management-SSH)
- ipv6 access group (Management-SSH)
- key-exchange (Management-SSH)
- login timeout (Management-SSH)
- mac hmac (Management-SSH)
- server-port (Management-SSH)
- shutdown (Management-SSH)
- vrf (Management-SSH)

Example

- This command places the switch in mgmt-ssh configuration mode:

```
switch(config)#management ssh
switch(config-mgmt-ssh)#
```

- This command returns the switch to global management mode:

```
switch(config-mgmt-ssh)#exit
switch(config)#
```

management telnet

The **management telnet** command places the switch in mgmt-telnet configuration mode to adjust telnet session connection parameters.

The **no management telnet** and **default management telnet** commands delete the mgmt-telnet configuration mode statements from ***running-config***.

Mgmt-telnet configuration mode is not a group change mode; ***running-config*** is changed immediately upon entering commands. Exiting mgmt-telnet configuration mode does not affect ***running-config***. The **exit** command returns the switch to global configuration mode.

Command Mode

Global Configuration

Command Syntax

```
management telnet
no management telnet
default management telnet
```

Commands Available in mgmt-telnet Configuration Mode

- idle-timeout (Management-Telnet)
- ip access group (Management-Telnet)
- ipv6 access group (Management-Telnet)
- shutdown (Management-Telnet)
- vrf (Management-Telnet)

Example

- This command places the switch in mgmt-telnet configuration mode:

```
switch(config)#management telnet
switch(config-mgmt-telnet)#
```

- This command returns the switch to global management mode:

```
switch(config-mgmt-telnet)#exit
switch(config)#
```

management xmpp

The **management xmpp** command places the switch in mgmt-xmpp configuration mode. Management over XMPP is disabled by default. To enable XMPP, you must provide the location of the XMPP server along with the username and password for the switch.

The **no management xmpp** and **default management xmpp** commands delete the mgmt-xmpp configuration mode statements from ***running-config***.

Mgmt-xmpp configuration mode is not a group change mode; ***running-config*** is changed immediately upon entering commands. Exiting mgmt-xmpp configuration mode does not affect ***running-config***. The **exit** command returns the switch to global configuration mode.

Command Mode

Global Configuration

Command Syntax

```
management xmpp
no management xmpp
default management xmpp
```

Commands Available in Mgmt-xmpp Configuration Mode

- domain (Management-xmpp)
- server (Management-xmpp)
- session (Management-xmpp)
- shutdown (Management-xmpp)
- switch-group (Management-xmpp)
- username (Management-xmpp)
- vrf (Management-xmpp)

Example

- This command places the switch in mgmt-xmpp configuration mode:

```
switch(config)#management xmpp
switch(config-mgmt-xmpp)#
```

- This command returns the switch to global management mode:

```
switch(config-mgmt-xmpp)#exit
switch(config-mgmt-xmpp)#
```

protocol http (API Management)

The **protocol http** command enables the hypertext transfer protocol (HTTP) server.

The **no protocol http** and **default protocol http** commands disable the HTTP server by removing the **protocol http** statement from ***running-config*.**

Command Mode

Mgmt-API Configuration

Command Syntax

```
protocol http [TCP_PORT]
no protocol http
default protocol http
```

Parameters

- ***TCP_PORT*** *Port number to be used for the HTTP server*. Options include:
  - <no parameter> Specifies default port number 80.
  - **port** <***1*** to ***65535***> Specifies HTTP server port number. Value ranges from ***1*** to ***65535***.
- **localhost** *The name of the server bound on the localhost.*
- **port** *The number of the TCP port to serve on.*

Related Commands

- **management api http-commands** places the switch in mgmt-api configuration mode.

Examples

- These commands enables the management API for the HTTP server.

```
switch(config)#management api http-commands
switch(config-mgmt-api-http-cmds)#
```

protocol https (API Management)

The **protocol https** command enables the HTTP secure server. The HTTP secure server is active by default.

The **default protocol https** command restores the default setting by removing the **no protocol https** statement from ***running-config***. The **no protocol https** command disables the HTTP secure server.

Command Mode

Mgmt-API Configuration

Command Syntax

```
protocol https [TCP_PORT]
no protocol https
default protocol https
```

Parameters

- ***TCP_PORT*** *Port number to be used for the HTTPS server*. Options include:
  - <no parameter> Specifies default port number 443.
  - **port** <***1*** to ***65535***> Specifies HTTP server port number. Value ranges from ***1*** to ***65535***.
- **certificate** *The HTTPS key and certificate to use.*
- **cipher** *Exclusive list of cryptographic ciphers.*
- **key-exchange** *Exclusive list of key-exchange algorithms.*
- **mac** *Exclusive list of MAC algorithms.*
- **port** *The number of the TCP port to serve on.*
- **ssl** *Configure SSL options.*

Related Commands

- **management api http-commands** places the switch in mgmt-api configuration mode.

Examples

- These commands enables service to the HTTP server. The **no shutdown** command allows access to the service.

```
switch(config)#management api http-commands
switch(config-mgmt-api-http-cmds)#protocol https
switch(config-mgmt-api-http-cmds)# no shutdown
```

- These commands specifies the port number that should be used for the HTTPS server. The **no shutdown** command allows access to the service.

```
switch(config)#management api http-commands
switch(config-mgmt-api-http-cmds)#protocol https port 52
switch(config-mgmt-api-http-cmds)#no shutdown
```

protocol https certificate (API Management)

The **protocol https certificate** command configures the HTTP secure server to request an X.509 certificate from the client. The client then authenticates the certificate with a public key.

The **no protocol https certificate** and **default protocol https certificate** commands restore default behavior by removing the **protocol https certificate** statement from ***running-config*.**

Command Mode

Mgmt-API Configuration

Command Syntax

```
protocol https certificate
no protocol https certificate
default protocol https certificate
```

Related Commands

- **management api http-commands** places the switch in mgmt-api configuration mode.

Examples

- These commands configure the HTTP secure server to request an X.509 certificate from the client for authentication.

```
switch(config)#management api http-commands
switch(config-mgmt-api-http-cmds)#protocol https certificate
switch(config-mgmt-api-http-cmds)#
```

server (XMPP Management)

The **server** command adds a XMPP server to ***running-config***. Multiple XMPP servers can be set up for redundancy. For redundant configurations, the XMPP server location should be a DNS name and not a raw IP address. The DNS server is responsible for returning the list of available XMPP servers, which the client can go through until an accessible server is found.

User authentication is provided by the XMPP server. Command authorization can be provided by EOS local configuration or TACACS+. The XMPP server should use the same authentication source as the switches. RADIUS is not supported as an XMPP authorization mechanism.

The **no server** and **default server** commands remove the specified XMPP server from ***running-config***.

Command Mode

Mgmt-xmpp Configuration

Command Syntax

```
server SERVER_NAME [SERVER_PORT]
no server
default server
```

Parameters

- ***SERVER_NAME*** XMPP server location. Options include:
  - *IP address* in dotted decimal notation.
  - a host name for the XMPP server.
- ***SERVER_PORT*** Server port. Options include:
  - **port <*1* to *65535*>** where *number* ranges from 1 to 65535. If no port is specified, the default port 5222 is used.

Examples

- This command configures the server hostname **arista-xmpp to server port 1**.

```
switch(config)#management xmpp
switch(config-mgmt-xmpp)#server arista-xmpp port 1
```

- This command removes the XMPP server.

```
switch(config-mgmt-xmpp)# no server
```

session privilege (XMPP Management)

The **session privilege** command will place the user in EXEC mode. The initial privilege level is meaningless by default. However, with the configuration of roles, users can add meaning to the different privilege levels. By default, XMPP does not limit access to any command.

Level 1-15: Commands accessible from EXEC Mode.

If AAA is not configured and the switch is configured to connect to the XMPP client, any message received is executed with privilege level 1 by default.

The **no session privilege** and **default session privilege** commands revert the list contents to ***none*** for the specified privilege levels.

Command Mode

Mgmt-xmpp Configuration

Command Syntax

```
session privilege PRIV_LEVEL
no session privilege
default session privilege
```

Parameters

- ***PRIV_LEVEL*** Privilege levels of the commands. Value ranges from **0** and **15.**

Examples

- These commands authorizes configuration commands (privilege level config 5) for XMPP.

```
switch(config)#(config)#management xmpp
switch(config-mgmt-xmpp)#session privilege 5
switch(config-mgmt-xmpp)#
```

- This command removes the privilege levels set for the XMPP session.

```
switch(config)#management xmpp
switch(config-mgmt-xmpp)#no session privilege
```

show inventory

The **show inventory** command displays the hardware components installed in the switch. Serial numbers and a description is also provided for each component.

Command Mode

EXEC

Command Syntax

show inventory

Examples

- This command displays the hardware installed in a DCS-7150S-52 switch.

```
switch>show inventory
System information
  Model                     Description
  ------------------------ ----------------------------------------------------
  DCS-7150S-52-CL           52-port SFP+ 10GigE 1RU + Clock

  HW Version  Serial Number  Mfg Date
  ----------- -------------- ----------
  02.00       JPE13120702    2013-03-27

System has 2 power supply slots
  Slot Model            Serial Number
  ---- ---------------- ----------------
  1    PWR-460AC-F      K192KU00241CZ
  2    PWR-460AC-F      K192L200751CZ

System has 4 fan modules
  Module  Number of Fans  Model            Serial Number
  ------- -------------- ---------------- ----------------
  1       1              FAN-7000-F       N/A
  2       1              FAN-7000-F       N/A
  3       1              FAN-7000-F       N/A
  4       1              FAN-7000-F       N/A

System has 53 ports
  Type             Count
  ---------------- ----
  Management       1
  Switched         52

System has 52 transceiver slots
  Port Manufacturer     Model            Serial Number    Rev
  ---- ---------------- ---------------- ---------------- ----
  1    Arista Networks  SFP-10G-SR       XCW1225FD753     0002
  2    Arista Networks  SFP-10G-SR       XCW1225FD753     0002

  51   Arista Networks  SFP-10G-SR       XCW1225FD753     0002
  52   Arista Networks  SFP-10G-SR       XCW1225FD753     0002

switch>
```

show xmpp neighbors

The **show xmpp neighbors** command displays all neighbors and their connection status. The XMPP server keeps track of all relationships between its users.

Command Mode

EXEC

Command Syntax

```
show xmpp neighbors
```

Example

- This command displays all the XMPP neighbors and their connection status.

```
switch#show xmpp neighbors
Neighbor                         State           Last Seen Login Time
-------------------------------  --------------- -------------------------
admin@test.aristanetworks.com    present         0:01:40 ago
test1@test.aristanetworks.com    present         20:29:39 ago

Neighbor                         Status Message
-------------------------------  ----------------------------------------
admin@test.aristanetworks.com
test1@test.aristanetworks.com    Arista Networks DCS-7048T-4S
switch#
```

show xmpp status

The **show xmpp status** command displays the current XMPP connection status to the server.

The XMPP server keeps track of all relationships between its users. In order for two users to directly communicate, this relationship must first be established and confirmed by the other party.

Switches automatically confirm requests from outside parties as long as they are a user from the same domain name, for example when you chat with your switch from your own XMPP chat client.

Command Mode

EXEC

Command Syntax

```
show xmpp status
```

Example

- This command displays the current XMPP connection status to the server.

```
switch# show xmpp status
XMPP Server:  port 5222
Client username: test@test.aristanetworks.com
Default domain: test.aristanetworks.com
Connection status: connected
switch#
```

show xmpp switch-group

The **show xmpp switch-group** command displays the configured and active switch groups for the switch.

Command Mode

EXEC

Command Syntax

```
show xmpp switch-group
```

Example

- This command displays the configured and active switch groups.

```
switch#show xmpp switch-group
testroom@conference.test.aristanetworks.com
switch#
```

shutdown (API Management)

The **shutdown** command, in mgmt-api configuration mode, disables management over API on the switch. API is disabled by default.

The **no shutdown** command, in mgmt-api configuration mode, enables the management API access**.**

The **default shutdown** command, in mgmt-api configuration mode, disables the management API access.

Command Mode

Mgmt-API Configuration

Command Syntax

```
shutdown
no shutdown
default shutdown
```

Related Commands

- **management api http-commands** places the switch in mgmt-api configuration mode.

Example

- These commands disables API access to the HTTP server.

```
switch(config)#management api http-commands
switch(config-mgmt-api-http-cmds)# shutdown
switch(config-mgmt-api-http-cmds)#
```

- These commands enables API access to the HTTP server.

```
switch(config)#management api http-commands
switch(config-mgmt-api-http-cmds)# no shutdown
switch(config-mgmt-api-http-cmds)#
```

shutdown (Telnet Management)

The **shutdown** command, in management-telnet mode, disables or enables Telnet on the switch. Telnet is disabled by default. The **management telnet** command places the switch in management-telnet mode.

- To enable Telnet, enter **no shutdown** at the management-telnet prompt.
- To disable Telnet, enter **shutdown** at the management-telnet prompt.

Command Mode

Management-Telnet Configuration

Command Syntax

```
shutdown
no shutdown
```

Example

- These commands enable Telnet, then return the switch to global configuration mode.

```
switch(config)#management telnet
switch(config-mgmt-telnet)#no shutdown
switch(config-mgmt-telnet)#exit
switch(config)#
```

- This command disables Telnet.

```
switch(config-mgmt-telnet)#shutdown
```

shutdown (XMPP Management)

The **shutdown** command, in mgmt-xmpp mode, disables or enables management over XMPP on the switch. XMPP is disabled by default.

The **no shutdown** and **default shutdown** commands re-enable XMPP by removing the **shutdown** command from ***running-config***.

Command Mode

Mgmt-xmpp Configuration

Command Syntax

```
shutdown
no shutdown
default shutdown
```

Example

- These commands enable management over XMPP, then return the switch to global configuration mode.

```
switch(config-mgmt-xmpp)#no shutdown
switch(config-mgmt-xmpp)#exit
switch(config)#
```

- This command disables management over XMPP.

```
switch(config-mgmt-xmpp)#shutdown
switch(config-mgmt-xmpp)#
```

switch-group (XMPP Management)

The **switch-group** command allows you to configure each switch to join specified chat rooms on startup. In order for the switch to participate in a chat group, the switch has to be configured to belong to the specified chat room.

The **no username** and **default username** commands delete the specified username by removing the corresponding **username** statement from ***running-config***.

Command Mode

Mgmt-xmpp Configuration

Command Syntax

```
switch-group name SECURITY
no switch-group
default switch-group
```

Parameters

- *name* Group name text that the user enters at the login prompt to access the CLI.

  Valid usernames begin with A-Z, a-z, or 0-9 and may also contain any of these characters:

  ```
  @   #   $   %   ^   &   *   -   _
  =   +   ;   <   >   ,   .   ~   |
  ```

- ***SECURITY*** password assignment.
  - **password** *pwd_txt* *name* is protected by specified password. *pwd_txt* is a clear-text string.
  - **password 0** *pwd_txt* *name* is protected by specified password. *pwd_txt* is a clear-text string.
  - **password 7** *pwd_txt* *name* is protected by specified password. *pwd_txt* is encrypted string.

Guidelines

- A switch group is an arbitrary grouping of switches within the network which belong to one chat group.
- In order to belong to one or more switch groups, the switch has to be manually assigned to it.
- Switch groups are defined dynamically based on the configuration of all of the switches in the network.
- As per the multi-user chat XMPP standard (XEP-0045), switch groups have a full name of GROUPNAME@conference.DOMAIN
- All CLI commands allow either the full group name or the short name, which are appended the @conference.DOMAIN
- If the switch belongs to multiple chat rooms, you must configure each group with a separate command.

Examples

- These commands configures the switch-group to be part of the chat room.

  ```
  switch(config)#management xmpp
  switch(config-mgmt-xmpp)# switch-group
  testroom@conference.test.aristanetworks.com password 0 arista
  ```

- Use the **show xmpp switch-group** to verify the active switch-group for the switch.

  ```
  switch# show xmpp switch-group
  testroom@conference.test.aristanetworks.com
  ```

username (XMPP Management)

The **username** command configures the switch's username and password on the XMPP server.

The **no username** and **default username** commands delete the specified username by removing the corresponding **username** statement from ***running-config***.

Command Mode

Mgmt-xmpp Configuration

Command Syntax

```
username name SECURITY
no username
default username
```

Parameters

- *name* username text that defines the XMPP username and password.

  Valid usernames begin with A-Z, a-z, or 0-9 and may also contain any of these characters:

  ```
  @   #   $   %   ^   &   *   (   )   -   _   =
  +   {   }   [   ]   ;   <   >   ,   .   ~   |
  ```

- ***SECURITY*** password assignment.
  - **password** *pwd_txt* *name* specifies and unencrypted shared key. *pwd_txt* is a clear-text string.
  - **password 0** *pwd_txt* *name* specifies and unencrypted key. *pwd_txt* is a clear-text string.
  - **password 7** *pwd_txt* *name* specifies a hidden key. *pwd_txt* is encrypted string.

Guidelines

Encrypted strings entered through this parameter are generated elsewhere. The **password 7** option (***SECURITY***) is typically used to enter a list of username-passwords from a script.

Examples

- These commands create the username and assigns it a password. The password is entered in clear text because the parameter is set to 0.

  ```
  switch(config)#management xmpp
  switch(config-mgmt-xmpp)#server arista-xmpp
  switch(config-mgmt-xmpp)#domain test.aristanetworks.com
  switch(config-mgmt-xmpp)#username test1@test.aristanetworks.com password 0
  arista
  switch(config-mgmt-xmpp)#no shutdown
  ```

- This command removes all usernames from the XMPP server.

  ```
  switch(config-mgmt-xmpp)#no username
  switch(config-mgmt-xmpp)#
  ```

vrf (API Management)

The **vrf** command places the switch in VRF configuration mode for the server. If the named VRF does not already exist, this command creates it.

Command Mode

Mgmt-API Configuration

Command Syntax

```
vrf VRF_INSTANCE
```

Parameters

- ***VRF_INSTANCE*** specifies the VRF instance.
  - **default** Instance is created in the default VRF.
  - *vrf_name* Instance is created in the specified user-defined VRF.

Related Commands

- **management api http-commands** places the switch in mgmt-api configuration mode.

Example

- This command creates a VRF named *management-vrf* and places the switch in VRF configuration mode for the new VRF.

```
switch(config)#management api http-commands
switch(config-mgmt-api-http-cmds)#vrf management-vrf
switch(config-mgmt-api-http-cmds-vrf-management-vrf)#
```

vrf (XMPP Management)

The **vrf** command places the switch in VRF configuration mode for the XMPP server. If the named VRF does not already exist, this command creates it.

The VRF configuration for the client is for the entire XMPP service, rather than per server. All servers resolving on a particular hostname must be reachable in the same VRF.

Command Mode

Mgmt-xmpp Configuration

Command Syntax

`vrf [VRF_INSTANCE]`

Parameters

- ***VRF_INSTANCE*** specifies the VRF instance.
  - **default** Instance is created in the default VRF.
  - *vrf_name* Instance is created in the specified user-defined VRF.

Example

- This command creates a VRF named *management-vrf* and places the switch in VRF configuration mode for the server.

```
switch(config)#management xmpp
switch(config-mgmt-xmpp)#vrf management-vrf
switch(config-mgmt-xmpp)
```

xmpp send

The **xmpp send** command can be used to connect to the XMPP server and send messages to switches or switch groups within the network.

Before switches can send messages to each other, they must friend each other. An easy way to have them auto friend each other is to have them join the same chat room. The friendship between switches can be verified by using the **show xmpp neighbor** command.

Command Mode

Privileged EXEC

Command Syntax

xmpp send to *neighbor* ***XMIT_TYPE*** *content*

Parameters

- *neighbor* Options include switches or switch groups within the network that are connected as friends in a chat room.
- ***XMIT_TYPE*** Transmission type. Valid options include:
  - **command** Sends an XMPP command.
  - **message** Sends an XMPP message.
- *content* The command you want the friends within the chat room to display or execute.

Configuration Restrictions

- Only enable-mode commands are allowed within the multi-switch CLI.
- Changing into a different CLI mode and running several commands in that mode is not supported (e.g., into configuration mode)
- An external XMPP client (for example Adium) can be used to send multiple lines within a single message. By sending multiple lines, it is possible to change into another CLI mode. After the message is processed, the switch automatically return to the enable mode.
- Commands that prompt for a response (like reload) are not supported.
- Long commands, such as image file copies, may cause the switch XMPP client to momentarily stop responding and disconnect. The switch should reconnect and the long command should complete.
- Many command outputs display in a specific table format. To achieve the same visual feel as through a terminal, use a monospaced font, such as Courier, for the incoming messages.

Example

- This command sends the switch in the chat room the request to execute the **show version** command.

```
switch# xmpp send test2 command show version
message from user: test2@test.aristanetworks.com
-----------------------------------------------
Hardware version:    04.40
Serial number:       JFL08432083
System MAC address:  001c.7301.7d69
Software image version: 4.12.3
Architecture:           i386
Internal build version: 4.12.3
Internal build ID:      f5ab5f57-9c26-4fe4-acaa-fb60fa55d01d
Uptime:                 2 hours and 38 minutes
Total memory:           1197548 kB
Free memory:            182452 kB
```

xmpp session

The **xmpp session** command is similar to running SSH from the switch. The user is required to input their username (default is to USER@DEFAULTDOMAIN) and password in order to connect to the XMPP server. This command allows you to interact in the enable mode with a switch or switch group over XMPP using the standard CLI, with access to help and tab completion. All commands are then executed remotely and only the non-empty results are displayed on the screen.

Command Mode

Privileged EXEC

Command Syntax

xmpp session *switchgroup*

Parameters

- *switchgroup* The option includes the switch group within the network that is connected as friends in a chat room.

Configuration Restrictions

- Only enable-mode commands are allowed within the multi-switch CLI.
- Changing into a different CLI mode and running several commands in that mode is not supported (e.g., into configuration mode)
- An external XMPP client (for example Adium) can be used to send multiple lines within a single message. By sending multiple lines, it is possible to change into another CLI mode. After the message is processed, the switch automatically return to the enable mode.
- Commands that prompt for a response (like reload) are not supported.
- Long commands, such as image file copies, may cause the switch XMPP client to momentarily stop responding and disconnect. The switch should reconnect and the long command should complete.
- Many command outputs display in a specific table format. To achieve the same visual feel as through a terminal, use a monospaced font, such as Courier, for the incoming messages.

Example

- This command displays the status of Ethernet 3 from *test1*, which is a member of the switch group chat room.

```
switch# xmpp session all@test.aristanetworks.com
xmpp-all# show int Eth3 status

response from: test1@test.aristanetworks.com

------------------------------------------------

Port  Name  Status     Vlan     Duplex  Speed   Type
Et3   bs3   connected  in Po3   a-full  a-1000  10GBASE-SR
switch#
```

Chapter 3

Command-Line Interface

The command-line interface (CLI) is one tool for controlling the switch and displaying information about its status and configuration. This chapter describes the use of the CLI.

This chapter includes these sections:

- Section 3.1: Accessing the EOS CLI
- Section 3.2: Processing Commands
- Section 3.3: Kernel-based Virtual Machine Commands and Configuration
- Section 3.4: Switch Platforms
- Section 3.5: Command Modes
- Section 3.6: Managing Switch Configuration Settings
- Section 3.7: Other Command-Line Interfaces
- Section 3.8: Common Criteria (CC)
- Section 3.9: Directory Structure
- Section 3.10: Command-Line Interface Commands

3.1 Accessing the EOS CLI

You can open an EOS CLI session through these connections:

- Ethernet management ports
- console port
- Telnet connections
- Secure Shell (SSH)

Figure 3-1 displays the EOS CLI in a Secure Shell connection.

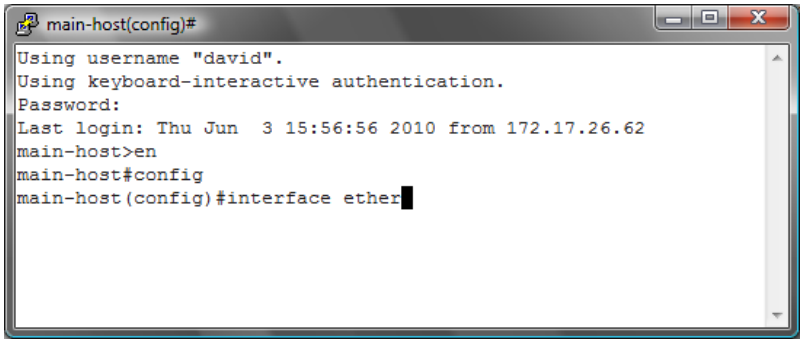


Figure 3-1: EOS Command-Line Interface

3.2 Processing Commands

3.2.1 Command Execution

Command keywords are not case-sensitive. The CLI also accepts truncated keywords that uniquely correspond to one command.

- The command abbreviation **con** does not execute a command in Privileged EXEC mode because the names of two commands begin with these letters: **configure** and **connect**.

  ```
  switch#con
  % Ambiguous command
  ```

- The command abbreviation **conf** executes **configure** in Privileged EXEC mode because no other command name begins with **conf**.

  ```
  switch#conf
  switch(config)#
  ```

3.2.2 Alias

The **alias** command creates an alias for a CLI command. Entering the alias in the CLI executes the corresponding command.

Example

- This command makes **srie** an alias for the command **show running-config interface ethernet 1-5.**

```
switch(config)#alias srie show running-config interface ethernet 1-5
switch(config)#srie
interface Ethernet1
   switchport access vlan 33
   storm-control broadcast level 1
   spanning-tree portfast
   spanning-tree bpduguard enable
interface Ethernet2
   switchport access vlan 33
   spanning-tree portfast
interface Ethernet3
   switchport access vlan 33
   spanning-tree portfast
   spanning-tree bpduguard enable
interface Ethernet4
interface Ethernet5
   shutdown
```

3.2.3 Cursor Movement Keystrokes

EOS supports these cursor movement keystrokes:

- **Ctrl-B** or the **Left Arrow** key: moves cursor to the left.
- **Ctrl-F** or the **Right Arrow** key: moves cursor to the right.
- **Ctrl-A**: moves cursor to beginning of line.
- **Ctrl-E**: moves cursor to end of line.
- **Esc-B**: moves cursor left one word.
- **Esc-F**: moves cursor right one word.

3.2.4 History Substitution Keystrokes

The history buffer retains the last 20 commands entered. History substitution keystrokes that access previously entered commands include:

- **Ctrl-P** or the **Up Arrow** key: Recalls the most recent buffered commands. Repeat to recall older commands.
- **Ctrl-N** or the **Down Arrow** key: Recalls more recent commands after using the **Ctrl-P** or the **Up Arrow**. Repeat to recall newer commands.

The **show history** command in Privileged EXEC mode displays the history buffer contents.

```
switch#show history
  en
  config
  exit
  show history
```

3.2.5 Command Lists and Syntax Assistance

EOS CLI uses widely followed conventions for providing command lists and syntax assistance. These conventions are available in all command modes.

- To display all commands available at this level, type a question mark (**?**):

```
switchName>?
  clear       Reset functions
  connect     Open a terminal connection
  disable     Turn off privileged commands
  enable      Turn on privileged commands
  exit        Exit from the EXEC
  help        Description of the interactive help system
  logout      Exit from the EXEC
  no          Negate a command or set its defaults
  ping        Send echo messages
  show        Show running system information
  telnet      Open a telnet connection
  terminal    Configure the terminal
  traceroute  Trace route to destination
```

- To display a list of commands beginning with a specific character sequence, type the sequence followed by a question mark.

```
switch#di?
diagnostic  diff  dir  disable
```

- To display a command's keywords or arguments, type a question mark as an argument.

```
switch>ping ?
  WORD  Ping destination address or hostname
```

- The switch accepts an address-mask or CIDR notation (address-prefix) in commands that require an IP address and mask. For example, these commands are processed identically:

```
switch(config)#ip route 0.0.0.0 255.255.255.255 10.1.1.254

switch(config)#ip route 0.0.0.0/32 10.1.1.254
```

- The switch accepts an address-wildcard or CIDR notation in commands requiring an IP address and wildcard. Wildcards use zeros to mask portions of the IP address and are found in some protocol configuration statements, including OSPF. The switch processes these commands identically:

```
switch:network 10.255.255.1 0.0.0.255 area 15

switch:network 10.255.255.1/24 area 15
```

3.2.6 Regular Expressions

A regular expression is pattern of symbols, letters, and numbers that represent an input string for matching an input string entered as a CLI parameter. The switch uses regular expression pattern matching in several BGP commands.

Regular expressions use the following operands:

. (period) matches any single character.

Example **1.3** matches 123, 133, and 1c3.

\ (backslash) matches character or special character following the backslash.

Example **15\.5\..** matches 15.5.10.10 it does not match 15.52.10.10

Example **\.** matches . (period)

^ (caret) matches the character or null string at the beginning of a string.

Example **^read** matches reader **^read** does not match bread.

*** (asterisk)** matches zero or more sequences of characters preceding the asterisk.

Example 12* matches 167, 1267, or 12267 it does not match 267

+ (plus sign) matches one or more sequences of characters preceding the plus sign.

Example **46+** matches 2467 or 24667 it does not match 247

$ (dollar sign) dollar sign matches the character or null string at the end of an input string.

Example **read$** matches bread but not reads

[ ] (brackets) matches characters or a character range separated by a hyphen.

Example **[0137abcr-y]** matches 0, 1, 3, v it does not match 2, 9, m, z

? (question mark) pattern matches zero or one instance. Entering Ctrl-V prior to the question mark prevents the CLI from interpreting **?** as a help command.

Example **x1?x** matches ***xx*** and ***x1x***

| (pipe) pattern matches character patterns on either side of bar.

Example **B(E|A)D** matches ***BED*** and ***BAD***. It does not match BD, BEAD, BEED, or EAD

()(parenthesis) nests characters for matching. Endpoints of a range are separated with a dash (-).

Example **6(45)+** matches 645454523 it does not match 6443

Example **([A-Za-z][0-9])+** matches ***C4*** or ***x9***

_ (underscore) Pattern replaces a long regular expression list by matching a comma (,) a space, or the beginning or end of the input string.

Example **_rxy_** matches any of the following:

```
^rxy$
^rxy  23
21  rxy
,rxy,
rxy
,rxy.
```

3.2.7 Scheduling CLI Commands

The **schedule** command facilitates the periodic execution of a specified CLI command. Command parameters configure the interval between consecutive execution instances and the maximum number of files that can be created when the command requires log files. By default, periodic execution of the following ***show tech-support*** command is enabled:

```
schedule tech-support interval 60 max-log-files 100 command show tech-support
```

The **schedule config max-concurrent-jobs** command specifies the number of scheduled commands which can be concurrently executed on the switch. By default, concurrency is set to 1.

The **show schedule summary** command displays commands scheduled for periodic execution.

Examples

- This command schedules the execution of a script once every 12 hours. The log file option is set to zero because the command does not generate output to std-out.

```
switch#schedule ms_1 interval 720 max-log-files 0 command bash
/mnt/flash/myscript.sh
```

- This command stores ***running-config*** contents to a log file once each hour, creating up to 24 log files.

```
switch#schedule backup-test interval 60 max-log-files 24 command show
running-config
```

- This command allows the switch to concurrently execute up to 2 scheduled commands.

```
switch(config)#schedule config max-concurrent-jobs 2
switch(config)#
```

- This command displays the commands that are scheduled for periodic execution.

```
switch(config)#show schedule summary
Maximum concurrent jobs  2
Name         At time Last  Inter\  Max   Logfile Location              Status
                     time   val    log
                            (mins)  files
------------ ------- ----- ------- ----- ----------------------------- ------
tech-support   now   09:41   60    100   flash:/schedule/tech-support/  Success
ms_1           now   16:28  720    10    flash:/schedule/ms_1
```

3.2.8 Running Bash Shell Commands Automatically with Event Handlers

Event handlers execute a Linux Bash shell command in response to a specific system event. An event handler consists of a Bash command, a trigger and a delay; when the trigger event occurs, the action is scheduled to run after ***delay*** seconds.

To create an event handler, use the **event-handler** command. This creates a new event handler and places the CLI in event handler configuration mode for that handler. Use the **action bash** command to configure a Bash command to run when the handler is triggered, and the **trigger** command to specify the trigger. Event handlers can be triggered by various events, including:

- system booting
- a change in a specified interface's operational status or IP address
- a change in the ***startup-config*** file
- a state change in a virtual machine monitored by VM Tracer

To change the delay period between the trigger and the action, use the **delay** command.

When an action is run, certain information is passed to it through environment variables. For the ***boot*** trigger, no variables are set. For the ***interface*** triggers, the following variables are set and passed to the action:

- ***$INTF*** interface name
- ***$OPERSTATE*** current operational status of the specified interface
- ***$IP-PRIMARY*** current primary IP address of the specified interface

To execute more than one Bash command in response to a trigger, create a script containing the desired commands and enter the file path to the script as the argument of the **action bash** command.

To display information about all event handlers or about a specific event handler, use the **show event-handler** command.

The **no event-handler** command deletes an event handler.

Examples

- These commands create an event handler named “eth_4” which will send email to a specified address when there is a change in the operational status of Ethernet interface 4:

```
switch(config)#event-handler eth_4
switch(config-event-eth_4)#action bash email x@yz.com -s "Et4 $OPERSTATE"
switch(config-event-eth_4)#trigger onintf ethernet 4 operstatus
switch(config-event-eth_4)#delay 60
switch(config-event-eth_4)#exit
switch(config)#
```

  The above handler uses the ***$OPERSTATE*** variable to include the current operational state (“linkup” or “linkdown”) in the subject of the email. Note that the action will only function if email has been configured on the switch.

- These commands create an event handler named “onStartup” which will execute a user-defined script 60 seconds after the system boots.

```
switch(config)#event-handler onStartup
switch(config-event-onStartup)#action bash /mnt/flash/startupScript1
switch(config-event-onStartup)#trigger onboot
switch(config-event-onStartup)#delay 60
switch(config-event-onStartup)#exit
switch(config)#
```

  The above handler will also be executed on exiting from event-handler configuration mode.

- This command displays information about all event handlers configured on the system.

```
switch#show event-handler
Event-handler onStartup
Trigger: onBoot delay 60 seconds
Action: /mnt/flash/startupScript1
Last Trigger Activation Time: 1 minutes 51 seconds ago
Total Trigger Activations: 1
Last Action Time: 51 seconds ago
Total Actions: 1

switch#
```

- This command deletes the event handler named “onStartup”.

```
switch(config)#no event-handler onStartup
switch(config)#
```

3.2.9 Running Adverse Drop Counters Monitor with Event Handlers

A monitoring capability for adverse drop counters can be used as a warning that the system is encountering an abnormal condition. The adverse drop counter monitor runs periodically (with a default of 60 seconds) and performs the following actions:

- Reads the values of adverse drop counters.
- Compares each value to the value read in the previous run.
- If counter values increase more than a certain threshold (with a default of 100), it is considered as a threshold violation.
- If any counter has more than a certain number of threshold violations within a specific time window (with a default of 3 violations within 15 minutes) a syslog message is logged.

No configuration is required to enable adverse drop counters monitor with event handlers. It is enabled by default and can be disabled, and can be customized for duration of time window and threshold levels. To customize the delay, polling interval, and condition for width, violation count, and threshold of this event handler, use the **event-handler DropCountersHandler** command. To display details of this event handler, use the **show event-handler DropCountersHandler** command.

Examples

- These commands customize the delay, polling interval, and condition for width, violation count, and threshold of this event handler. Each parameter may be customized separately, with all other parameters remaining unchanged.

```
switch(config)#event-handler DropCountersHandler
switch(config-DropCountersHandler)#action bash DropCounterLog.py -l
switch(config-DropCountersHandler)#delay 0
switch(config-DropCountersHandler)#trigger on-counters
switch(config-DropCountersHandler-counters)#poll interval 60
switch(config-DropCountersHandler-counters)#condition
bashCmd."DropCounterMonitor.py" -w 800" > 0
switch(config-DropCountersHandler-counters)#condition
bashCmd."DropCounterMonitor.py" -c 5" > 0
switch(config-DropCountersHandler-counters)#condition
bashCmd."DropCounterMonitor.py" -t 200" > 0
switch(config-DropCountersHandler-counters)#
```

- This command disables this event handler.

```
switch(config)#no event-handler DropCountersHandler
switch(config)#
```

- This command displays details of this event-handler.

```
switch(config)#show event-handler DropCountersHandler
Event-handler DropCountersHandler (BUILT-IN)
Trigger: on-counters delay 0 seconds
  Polling Interval: 60 seconds
  Condition: bashCmd."DropCounterMonitor.py" > 0
Threshold Time Window: 0 Seconds, Event Count: 1 times
Action: DropCounterLog.py -l
Action expected to finish in less than 20 seconds
Total Polls: 39
Last Trigger Detection Time: 38 minutes 22 seconds ago
Total Trigger Detections: 1
Last Trigger Activation Time: 38 minutes 22 seconds ago
Total Trigger Activations: 1
Last Action Time: Never
Total Actions: 1

switch(config)#
```

3.3 Kernel-based Virtual Machine Commands and Configuration

Arista's EOS has leveraged its unmodified Linux kernel, and embraced open source standards-based technology that has brought operating system virtualization to Ethernet switching, utilizing the kernel-based virtual machine (KVM) as follows:

- The hypervisor is the Linux kernel.
- The core virtualization infrastructure is provided by the kernel module.
- The CPU-specific implementation is provided by the processor-specific module (Intel or AMD).
- The generic machine emulator and virtualizer KVM is provided by a Modified Quick Emulator (QEMU), which transforms the Linux kernel into the hypervisor.

The standard Linux kernel is the hypervisor, resulting in changes to the standard kernel (such as memory support and scheduler). Optimizations to these Linux components (such as a new scheduler in the 2.6 kernel) benefit both the hypervisor (host operating system) and Linux guest operating systems. With the kernel acting as the hypervisor, the switch can run other operating systems, such as Windows or Linux.

All components required are pre-installed with the Arista EOS software image, requiring only the download of the image. A few additional configuration steps get the KVM fully operational.

This chapter contains the following sections:

- Section 3.3.1: KVM Commands
- Section 3.3.2: KVM Configuration

3.3.1 KVM Commands

The following table covers KVM commands used throughout the configuration.

Table 3-1 KVM Commands

Command	Description
comment	Up to 240 character comment for this mode.
config-file	VM's `libvirt` configuration file (overrides other settings).
default	Set a command to its defaults.
disk-image	Add Virtual Machine disk image.
enable	Enable VM.
exit	Exit from Virtual Machine configuration mode.
help	Description of the interactive help system.
memory-size	Set memory size.
no	Negate a command or set its defaults.
show	Show running system information.
virtual-nic	Add virtual NIC.
vnc-port	Set VNC server port.
! !	Append to comment

3.3.1.1 CLI Commands

The following KVN CLI commands are used throughout the configuration.

vm

In `config` mode, the `vm` CLI command creates or deletes a KVM configuration, or enters `config-vm` mode. A newly created KVM will have an empty config file path and is disabled.

The CLI command syntax is as follows:

```
[no] vm NAME
```

Note Deleting an enabled KVM first disables it, using the same process as the `no enabled` command in the `config-vm` mode.

config-file

In `config-vm` mode, the `config-file` CLI command sets the path of the `libvirt` config file, using standard file syntax (e.g. `flash:vm/NetscalerVPX.xml or sata1:vm/NetscalerVPX.xml` or `/mnt/sata1/vm/NetscalerVPX.xml`). Changing this value does not affect the state of a currently enabled KVM. To use the new file, the user must disable and then reenable the KVM.

The CLI command syntax is as follows:

```
config-file [PATH]
```

Note If the file does not exist, a warning is printed and the new value is stored.

enabled

In `config-vm` mode, the `enabled` CLI command allows enabling a currently disabled VM, causing it to start up immediately. If a VM is enabled in the `startup-config`, it starts up automatically when EOS boots (or when `VirtAgent` starts).

The CLI command syntax is as follows:

```
[no] enabled
```

Disabling a currently enabled VM initiates a shutdown process in the following sequence:

- Attempt to shut down the VM politely if the guest OS supports ACPI.
- If the VM is still running after 30 seconds, terminate it.

show vm

In `enable` mode, the `show vm` CLI command prints information about the configuration and status of a KVM, or of all KVMs if `NAME` is omitted, as follows:

- Configuration:
  Name, config file path, and enabled.
- Status:
  PID, log file path, and serial console pty path.
- Current resource usage:
  RES, CPU%
- (Detailed only) contents of the config file.
- (Detailed only) contents of the log file.

The CLI command syntax is as follows:

```
show vm [detailed] [NAME]
```

attach vm

In `enable` mode, the `attach vm` CLI command connects to a KVM's serial console pty (using virsh console).

Note Press `Ctrl-]` to exit to the CLI.

The CLI command syntax is as follows:

```
attach vm [NAME]
```

show tech-support

The CLI command syntax is as follows:

```
show tech-support [detailed] [NAME]
```

reload

In `enable` mode, the `reload` CLI command is executed before restarting the system, and will shut down currently enabled KVMs using the same process as the `no enabled` command in `config-vm` mode.

The CLI command syntax is as follows:

```
reload
```

3.3.2 KVM Configuration

Arista EOS enables kernel-based virtual machine (KVM) instances by running KVM on the control-plane CPU of the switch. KVM instances can be defined from the CLI.

To configure a KVM, you must download the virtual machine image and configure the EOS.

This section contains the following topics:

- Section 3.3.2.1: Configuring a KVM
- Section 3.3.2.2: Configuring a Guest KVM

3.3.2.1 Configuring a KVM

To configure a KVM, perform the following steps:

Step 1 Download the Virtual Machine Image to `/mnt/flash`

Step 2 Name the virtual machine:
```
switch(config)#virtual-machine [kvm_name]
```
Example:
```
switch(config)#virtual-machine foo
```

Step 3 Provide a pointer to the image:
```
switch(config-vm-foo)#disk-image [file:[path] image-format [format]
```
Example:
```
disk-image file:/mnt/flash/fedora.img image-format qcow2
```

Step 4 Define the amount of memory allocated:
```
switch(config-vm-foo)#memory-size [size in bytes]
```

Step 5 Bind the virtual NIC to an SVI (or management interface):
```
switch(config-vm-foo)#virtual-nic 1 vlan [1-4]
switch(config-vm-foo)#virtual-nic 1 management [1-4]
```

Step 6 Create the VNC server's tcp port (display):
`switch(config-vm-foo)#vnc-port [vnc-port number]`

Step 7 Enable the virtual machine:
`switch(config-vm-foo)#enable`

Optionally attach to the virtual machine via VNC client pointed to the switch's IP address. However, if Kernel `hair-pinning` is currently not enabled, preventing communication directly with the local switch, all traffic must have a destination on another networked device (such as a router, switch, or server).

For specifics about KVM please visit http://www.linux-kvm.org/.

Note In the Real VNC Viewer for **`Options`**, **`Expert`**, and **`ColorLevel`**, if the default value is `pal8`, establishing a session may fail. If this occurs, set this value to `full` and reconnect.

Example

```
switch#copy http://berrange.fedorapeople.org/images/2012-02-
29/f16-x86_64-openstack-sda.qcow2
(http://berrange.fedorapeople.org/images/2012-02-29/f16-x86_64-
openstack-sda.qcow2) flash:
...
switch(config)#virtual-machine foo
switch(config-vm-foo)#disk-image file:/mnt/flash/fedora.img image-format qcow2
switch(config-vm-foo)#memory-size 512
switch(config-vm-foo)#virtual-nic 1 vlan 1
switch(config-vm-foo)#virtual-nic 2 management 1
switch(config-vm-foo)#vnc-port 5900
switch(config-vm-foo)#enable
```

3.3.2.2 Configuring a Guest KVM

To configure a guest KVM, perform the following steps:

Step 1 Download the Virtual Machine Image to `/mnt/flash`

Step 2 Name the virtual machine:
`switch(config)#virtual-machine [guest_name]`
Example:
`switch(config)#virtual-machine guest123`

Step 3 Provide a pointer to the image:
`switch(config-vm-guest123)#disk-image [file:[path] image-format [format]`
Example:

```
switch(config-vm-guest123)#disk-image flash:f16-x86_64-openstack-sda.qcow2
image-format ?
iso iso image format
  qcow qcow image format
  qcow2 qcow2 image format
  raw raw image format
  vmdk vmdk image format
switch(config-vm-guest123)#disk-image flash:f16-x86_64-openstack-sda.qcow2
image-format qcow2
```

Step 4 Define the amount of memory allocated:
`switch(config-vm-guest123)#memory-size [size in bytes]`

Step 5 Bind the virtual NIC to an SVI (or management interface):

```
switch(config-vm-guest123)#virtual-nic 1 vlan [1-4]
switch(config-vm-guest123)#virtual-nic 2 management [1-4]
```

Step 6 Create the VNC server's tcp port (display):

```
switch(config-vm-guest123)#vnc-port [vnc-port number]
```

Step 7 Enable the virtual machine:

```
switch(config-vm-guest123)#enable
```

Example

```
switch#copy http://berrange.fedorapeople.org/images/2012-02-
29/f16-x86_64-openstack-sda.qcow2
(http://berrange.fedorapeople.org/images/2012-02-29/f16-x86_64-
openstack-sda.qcow2) flash:
...
switch#config terminal
switch(config)#virtual-machine ?
    WORD Virtual Machine name
switch(config)#virtual-machine foo
switch(config-vm-foo)#disk-image flash:f16-x86_64-openstack-sda.qcow2
image-format ?
    iso           iso image format
    qcow          qcow image format
    qcow2         qcow2 image format
    raw           raw image format
    vmdk          vmdk image format
switch(config-vm-foo)#disk-image flash:f16-x86_64-openstack-sda.qcow2
image-format qcow2
switch(config-vm-foo)#memory-size 1024
switch(config-vm-foo)#virtual-nic ?
    <1-4>         Virtual NIC Id
switch(config-vm-foo)#virtual-nic 1 ?
    Management    Management interface
    Vlan          Vlan interface
switch(config-vm-foo)#virtual-nic 1 vlan 1
switch(config-vm-foo)#virtual-nic 2 management 1
switch(config-vm-foo)#enable
switch(config-vm-foo)#
switch(config-vm-foo)#^Z
switch#write mem
switch#
switch#show virtual-machine detail
Virtual Machine: foo
    Enabled:           Yes
    State:             Running
    Disk Image:        /mnt/flash/f16-x86_64-openstack-sda.qcow2
    Disk Image Format: qcow2
    Memory Size: 1024MB
    VNC port: 5900
    Virtual Nic: vnic1
        Mac Address:   52:54:00:ee:11:c9
        Device:        Vlan1
        Model Type:    e1000
    Virtual Nic:       vnic2
        Mac Address:   52:54:00:df:2a:e1
        Device:        Management1
        Model Type:    e1000
switch#
```

Note Once a Guest KVM has its configuration setup correctly, it can have virtual NIC connections in VLANs (inband), or on out-of-band management interfaces.

3.4 Switch Platforms

Features and CLI commands vary by switch platform. CLI options may also vary by switch platform for commands that are available on all platforms. Command descriptions in this manual describe feature availability and command parameters on the basis of switch platform, noting exceptions that exist among models that use a common platform.

- https://www.arista.com/en/products/switches lists the Arista switches and platforms upon which they operate.
- https://www.arista.com/en/support/product-documentation/supported-features lists Arista switch feature availability by switch platform. For the latest features, also consult the Release Notes, available at https://www.arista.com/en/support/software-download.

These sections describe the following topics:

- Section 3.4.1: Viewing the Model Number
- Section 3.4.2: Determining a Switch's Operating Platform
- Section 3.4.3: Modular System Platforms – 7500 and 7500E Series Switches
- Section 3.4.4: Viewing Modules on 7300 Series Modular Switches
- Section 3.4.5: Multi-Chip Devices

3.4.1 Viewing the Model Number

To view the switch's model number through the CLI, enter **show version**.

Example

- This command displays the model number, serial number, system MAC address, and manufacturing information of a DCS-7150S-64 switch.

```
switch>show version
Arista DCS-7150S-64-CL-F
Hardware version:    01.01
Serial number:       JPE13120819
System MAC address:  001c.7326.fd0c

Software image version: 4.13.2F
Architecture:           i386
Internal build version: 4.13.2F-1649184.4132F.2
Internal build ID:      eeb3c212-b4bd-4c19-ba34-1b0aa36e43f1

Uptime:                 16 hours and 39 minutes
Total memory:           4017088 kB
Free memory:            1348228 kB

switch>
```

3.4.2 Determining a Switch's Operating Platform

FM6000 Platforms

To determine the operating platform on switch, display **platform** command options from Global Configuration command mode.

- This command displays the operating platform of a switch operating on the FM6000 platform (7150 Series switches).

```
switch(config)#platform ?
  fm6000  FM6000 chip

switch(config)#platform
```

Arad and Petra Platforms

The **platform ? command** displays the same options on Arad and Petra platform switches. Refer to Section 3.4.1 to determine the switch's model number.

- Fixed system switches (DCS-7048 Series) operate on the Petra platform.
- Modular switches (DCS-7500 Series) operate on Arad and Petra platforms. Section 3.4.3: Modular System Platforms – 7500 and 7500E Series Switches describe platform usage on these switches.

Arad and Petra platform switch typically utilize multiple chips. Section 3.4.5 describe methods of determining the port distribution on multi-chip platforms.

Example

- These commands display platform options of a switch operating on either Petra or Arad platforms.

```
switch(config)#platform ?
  arad    Arad switch chip
  fe1600  Fe1600 chip
  fe600   Fe600 fabric chip
  petraA  PetraA switch chip
  ptp     Precision Time Protocol
  sand    Sand platform

switch(config)#platform
```

Trident and Trident-II Platforms

The **platform ? command** returns ***trident*** on switches that operate on Trident or Trident-II platforms. Trident-II platform switches include options that configure the forwarding and routing tables. To determine the Trident platform that a switch uses, display **platform trident** options.

- These commands indicate that the switch is operating on the Trident-II platform:

```
switch(config)#platform ?
  ptp      Precision Time Protocol
  trident  Trident chip

switch(config)#platform trident ?
  fabric            Fabric configuration
  forwarding-table  Forwarding table configuration
  mmu               Trident MMU configuration
  routing-table     Routing table configuration

switch(config)#platform trident
```

Fixed and Modular switches are available that operate on the Trident-II platform. Refer to Section 3.4.1 to determine the switch's model number. Section 3.4.4: Viewing Modules on 7300 Series Modular Switches displays the modules on a Trident-II platform modular switch.

Trident-II platform switches typically utilize multiple chips. Section 3.4.5 describe methods of determining port distribution on multi-chip platforms.

3.4.3 Modular System Platforms – 7500 and 7500E Series Switches

Modular switch platforms depend on their installed modules along with the fabric and forwarding software modes. The **show module** command displays the fabric modules in the switch. System performance in switches containing both module types is based on first-generation fabric capabilities. Best practice is to avoid switch configurations with mixed fabric modules.

These sections describe modular switch components and software modes that program their capacities.

3.4.3.1 Fabric Modules and Fabric Mode – 7500 and 7500E Series Switches

Each modular switch fabric module is categorized as first-generation or E-Series:

- First-generation fabric modules support all basic switch functions.
- E-Series fabric modules support faster fabric link speeds, greater internal table capacities, and advanced encoding formatting.

Fabric mode determines the switch's fabric performance capabilities. This mode must match the fabric modules in the switch. Fabric mode settings include:

- fe600: Supports first-generation fabric modules.
- fe1600: Supports E-Series fabric modules.

E-series fabric modules can operate in ***fe600*** mode, but are limited to first-generation fabric performance. First-generation modules cannot operate in ***fe1600*** mode. Switches containing both types of modules must be set to ***fe600*** mode. Best practice is to avoid switch configurations with mixed fabric modules.

When a switch reloads, fabric mode is determined by the following (in order of precedence):

Step 1 Switches reloading in ***petraA*** forwarding compability mode (Section 3.4.3.2) also reload in ***fe600*** fabric mode.

Step 2 As specified by the **platform sand fabric mode (7500 and 7500E Series)** statement in ***running-config***.

Step 3 The first fabric module that becomes operational as the switch reloads.

In switches with a homogeneous module set, the fabric mode matches its fabric modules. Switches with a mixed set of modules are typically reloaded in ***fe600*** mode because first generation modules are usually operational before E-Series modules. However, the fabric mode in mixed module switches that are reloading cannot be guaranteed in the absence of the first two conditions.

Example

- This command configures the switch to reload in ***fe1600*** fabric mode to support E-series fabric modules. After issuing this command, the switch should be reset only after exchanging all switch fabric modules to E-series modules.

```
switch(config)#platform sand fabric mode fe1600
switch(config)#exit
switch#show platform sand compatibility
                    Configuration        Status
Forwarding mode     None                 Arad
Fabric mode         Fe1600               Fe600
switch#
```

3.4.3.2 Linecard Modules and Forwarding Compatibility Mode – 7500 and 7500E Series

Each modular switch linecard module is categorized as first-generation or E-Series:

- First-generation linecard modules support all basic switch functions.
- E-Series linecard modules support provide faster data processing, greater internal table capacities, and advanced encoding formatting.

The forwarding compatibility mode determines the switch's performance capabilities when forwarding data between linecard interfaces. Forwarding compatibility mode settings include:

- ***PetraA:*** Supports first-generation linecard modules.
- ***Arad:*** Supports E-Series linecard modules.

Forwarding compatibility mode determines the operational capacity of installed linecards. Table 3-2 lists the affect of the forwarding compatibility mode on linecard module types.

Table 3-2 Linecard Module and Forwarding Mode Performance

Linecard Module Type	Forwarding Compatibility Mode	Linecard Operating Capacity
First-generation	petraA	First-generation performance capacity.
First-generation	arad	Linecard is powered-down.
E-Series	petraA	First-generation performance capacity.
E-Series	arad	E-series performance capacity.

Important! Switches must contain E-Series fabric modules to operate at E-Series performance capacities.

The forwarding compatibility mode is configured by the **platform sand forwarding mode (7500 and 7500E Series)** command. This command may be required after exchanging a linecard for a different module type or in switches containing first-generation and E-series linecards.

Without a **platform sand forwarding mode** command, forwarding compatibility mode is determined by the first linecard that is operational after reloading the switch. In a switch that is reloaded with a homogeneous module set, forwarding compatibility mode matches its linecards. Switches with a mixed set of modules are typically reloaded in ***petraA*** mode because first generation modules are usually operational before E-Series modules. However, forwarding compatibility mode in mixed module switches that are reloading is not guaranteed without a **platform sand forwarding mode** command.

Example

- This command changes the forwarding software mode to support E-series linecard modules. This command should be run only after exchanging all linecards to E-series modules.

```
switch(config)#platform sand forwarding mode arad
switch(config)#
```

3.4.3.3 Viewing Modules – 7500 and 7500E Series

The **show module** command displays the model number of all installed modules.

- This command displays the modules of a 7504 switch that contains first-generation modules.

```
switch>show module
Module    Ports Card Type                              Model           Serial No.
--------- ----- -------------------------------------- --------------- -----------
1         2     DCS-7500 Series Supervisor Module      7500-SUP        JSH11440327
2         1      Standby supervisor                     Unknown         Unknown
3         48    48-port SFP+ 10GigE Linecard           7548S-LC        JSH10449938
4         48    48-port SFP+ 10GigE Linecard           7548S-LC        JSH11091247
5         48    48-port SFP+ 10GigE Linecard           7548S-LC        JSH11211614
6         48    48-port SFP+ 10GigE Linecard           7548S-LC        JSH11520288
Fabric1   0     DCS-7504 Fabric Module                 7504-FM         JSH11451230
Fabric2   0     DCS-7504 Fabric Module                 7504-FM         JSH11451210
Fabric3   0     DCS-7504 Fabric Module                 7504-FM         JSH11410115
Fabric4   0     DCS-7504 Fabric Module                 7504-FM         JSH11380318
Fabric5   0     DCS-7504 Fabric Module                 7504-FM         JSH11340955
Fabric6   0     DCS-7504 Fabric Module                 7504-FM         JSH11410128

Module    MAC addresses                          Hw      Sw      Status
--------- -------------------------------------- ------- ------- -------
1         00:1c:73:03:06:ac - 00:1c:73:03:06:ac  07.06   4.12.1  Active
2                                                        4.12.1  Standby
3         00:1c:73:03:80:44 - 00:1c:73:03:80:73  06.00           Ok
4         00:1c:73:03:e4:34 - 00:1c:73:03:e4:63  07.10           Ok
5         00:1c:73:12:0b:3f - 00:1c:73:12:0b:6e  07.30           Ok
6         00:1c:73:12:b6:3f - 00:1c:73:12:b6:6e  08.00           Ok
Fabric1                                          05.03           Ok
Fabric2                                          05.03           Ok
Fabric3                                          05.02           Ok
Fabric4                                          05.02           Ok
Fabric5                                          05.02           Ok
Fabric6                                          05.02           Ok
switch>
```

- This command displays modules of a 7504 switch that contains E-Series modules.

```
switch>show module
Module    Ports Card Type                              Model           Serial No.
--------- ----- ------------------------------------ --------------- -----------
1         3     DCS-7500E-SUP Supervisor Module      7500E-SUP       JAS13060306
3         72    48 port 10GbE SFP+ & 2x100G Linecard 7500E-72S-LC    JAS12410019
4         72    48 port 10GbE SFP+ & 2x100G Linecard 7500E-72S-LC    JPE13041458
5         72    48 port 10GbE SFP+ & 2x100G Linecard 7500S-72S-LC    JAS12380089
Fabric1   0     DCS-7504-E Fabric Module             7504E-FM        JAS12370008
Fabric2   0     DCS-7504-E Fabric Module             7504E-FM        JAS12380012
Fabric3   0     DCS-7504-E Fabric Module             7504E-FM        JAS12370014
Fabric4   0     DCS-7504-E Fabric Module             7504E-FM        JAS12380008
Fabric5   0     DCS-7504-E Fabric Module             7504E-FM        JAS12380017
Fabric6   0     DCS-7504-E Fabric Module             7504E-FM        JAS12370009

Module    MAC addresses                             Hw      Sw      Status
--------- ----------------------------------------- ------- ------- -------
1         00:1c:73:00:f4:cd - 00:1c:73:00:f4:ce  00.00   4.12.3  Active
3         00:1c:73:00:9c:7b - 00:1c:73:00:9c:c2  00.00           Ok
4         00:1c:73:28:a0:57 - 00:1c:73:28:a0:9e  00.00           Ok
5         00:1c:73:00:9a:cb - 00:1c:73:00:9b:12  02.07           Ok
Fabric1                                          00.00           Ok
Fabric2                                          00.00           Ok
Fabric3                                          00.00           Ok
Fabric4                                          00.00           Ok
Fabric5                                          00.00           Ok
Fabric6                                          00.00           Ok
switch>
```

3.4.4 Viewing Modules on 7300 Series Modular Switches

7300 Series Modular switches operate on Trident-II platform. The **show module** command displays the model number of all installed modules.

```
switch>show module
Module    Ports Card Type                              Model           Serial No.
--------- ----- ------------------------------------ --------------- -----------
1         3     Supervisor 7300X SSD                 DCS-7300-SUP-D  JAS13340024
3         128   32 port 40GbE QSFP+ LC               7300X-32Q-LC    JPE13440416
4         64    48 port 10GbE SFP+ & 4 port QSFP+ LC 7300X-64S-LC    JAS13310113
5         64    48 port 10GbE SFP+ & 4 port QSFP+ LC 7300X-64S-LC    JAS13340033
6         64    48 port 10GbE SFP+ & 4 port QSFP+ LC 7300X-64S-LC    JAS13310103
Fabric1   0     7304X Fabric Module                  7304X-FM        JAS13320077
Fabric2   0     7304X Fabric Module                  7304X-FM        JAS13350043
Fabric3   0     7304X Fabric Module                  7304X-FM        JAS13350050
Fabric4   0     7304X Fabric Module                  7304X-FM        JAS13350056

Module    MAC addresses                          Hw      Sw      Status
--------- -------------------------------------- ------- ------- -------
1         00:1c:73:36:4b:71 - 00:1c:73:36:4b:72  01.01   4.13.3F Active
3         00:1c:73:58:d4:68 - 00:1c:73:58:d4:87  03.04           Ok
4         00:1c:73:36:05:61 - 00:1c:73:36:05:94  02.02           Ok
5         00:1c:73:36:0a:e1 - 00:1c:73:36:0b:14  02.03           Ok
6         00:1c:73:36:02:e1 - 00:1c:73:36:03:14  02.02           Ok
Fabric1                                          00.00           Ok
Fabric2                                          00.00           Ok
Fabric3                                          00.00           Ok
Fabric4                                          00.00           Ok
switch>
```

3.4.5 Multi-Chip Devices

Trident-II, Petra, and Arad platform switches and linecards utilize multiple chips, with Ethernet ports evenly distributed among the chips. Creating multi-port data structures (including port channels) that include ports from multiple chips protects against the failure of an individual chip on a device.

The following sections describe methods of determining port distribution on various switch platforms

Petra Fixed Switches

7048-Series switches are Petra platform devices that distribute ports among two PetraA chips. The **show platform petraA port-info routing** command displays the ports that are controlled by each chip.

Example

- This command displays the following Ethernet port distribution on a DCS-7048-T switch:
  - Petra0 chip controls Ethernet 1 through Ethernet 32

- Petra1 chip controls Ethernet 33 through Ethernt 52

```
switch#show platform petraA port-info routing
Petra0 Port Routing Information:
================================================================================
                          sys     fap                          routing
 intfName               port-id port-id intfType  portType   v4 v6
================================================================================
 CpuTm                     2       0    Cpu       Tm          1  1
            <-------OUTPUT OMITTED FROM EXAMPLE-------->
 Ethernet1                29       2    Nif       Ethernet    1  1
 Ethernet2                30       3    Nif       Ethernet    1  1
            <-------OUTPUT OMITTED FROM EXAMPLE-------->
 Ethernet31               59      32    Nif       Ethernet    1  1
 Ethernet32               60      33    Nif       Ethernet    1  1
            <-------OUTPUT OMITTED FROM EXAMPLE-------->
 RawPetra0/70            2118      70    Recycling Raw         1  1
Petra1 Port Routing Information:
================================================================================
                          sys     fap                          routing
 intfName               port-id port-id intfType  portType   v4 v6
================================================================================
 CpuTm                     2       0    Cpu       Tm          1  1
            <-------OUTPUT OMITTED FROM EXAMPLE-------->
 Ethernet33               66       2    Nif       Ethernet    1  1
            <-------OUTPUT OMITTED FROM EXAMPLE-------->
 Ethernet52               85      21    Nif       Ethernet    1  1
 L3SecondHop1Petra1       86      22    Recycling Ethernet    1  1
            <-------OUTPUT OMITTED FROM EXAMPLE-------->
 RawPetra1/70            2118      70    Recycling Raw         1  1
switch#
```

Petra Modular Switches

Linecards on 7500-Series modular switches distribute Ethernet ports among multiple petraA chips. The **show platform petraA port-info routing** command displays the ports that are controlled by each chip on all PetraA linecards or on a single linecard.

Example

- This command displays the following Ethernet port distribution on linecard 4 of a DCS-7504 switch:
  - Petra4/0 chip controls Ethernet 4/1 through Ethernet 4/8
  - Petra4/1 chip controls Ethernet 4/9 through Ethernet 4/16
  - Petra4/2 chip controls Ethernet 4/17 through Ethernet 4/24
  - Petra4/3 chip controls Ethernet 4/25 through Ethernet 4/32
  - Petra4/4 chip controls Ethernet 4/33 through Ethernet 4/40

- Petra4/5 chip controls Ethernet 4/41 through Ethernet 4/48

```
switch(s1)#show platform petra module 4 port-info routing
Petra4/0 Port Routing Information:
================================================================================
                          sys     fap                          routing
 intfName               port-id port-id intfType  portType    v4 v6
================================================================================
 CpuTm                      2       0    Cpu       Tm           1  0
             <-------OUTPUT OMITTED FROM EXAMPLE-------->
 Ethernet4/1              221       2    Nif       Ethernet     1  0
 Ethernet4/2              222       3    Nif       Ethernet     1  0
 Ethernet4/3              223       4    Nif       Ethernet     1  0
 Ethernet4/4              224       5    Nif       Ethernet     1  0
 Ethernet4/5              225       6    Nif       Ethernet     1  0
 Ethernet4/6              226       7    Nif       Ethernet     1  0
 Ethernet4/7              227       8    Nif       Ethernet     1  0
 Ethernet4/8              228       9    Nif       Ethernet     1  0
             <-------OUTPUT OMITTED FROM EXAMPLE-------->
 RawPetra4/0/70          2118      70    Recycling Raw          1  0
Petra4/1 Port Routing Information:
================================================================================
                          sys     fap                          routing
 intfName               port-id port-id intfType  portType    v4 v6
================================================================================
 CpuTm                      2       0    Cpu       Tm           1  0
             <-------OUTPUT OMITTED FROM EXAMPLE-------->
 Ethernet4/9              253       2    Nif       Ethernet     1  0
             <-------OUTPUT OMITTED FROM EXAMPLE-------->
Petra4/5 Port Routing Information:
================================================================================
                          sys     fap                          routing
 intfName               port-id port-id intfType  portType    v4 v6
================================================================================
             <-------OUTPUT OMITTED FROM EXAMPLE-------->
 Ethernet4/41             381       2    Nif       Ethernet     1  0
 Ethernet4/42             382       3    Nif       Ethernet     1  0
 Ethernet4/43             383       4    Nif       Ethernet     1  0
 Ethernet4/44             384       5    Nif       Ethernet     1  0
 Ethernet4/45             385       6    Nif       Ethernet     1  0
 Ethernet4/46             386       7    Nif       Ethernet     1  0
 Ethernet4/47             387       8    Nif       Ethernet     1  0
 Ethernet4/48             388       9    Nif       Ethernet     1  0
             <-------OUTPUT OMITTED FROM EXAMPLE-------->
switch(s1)#
```

Arad Modular Switches

7500-E Series linecards distribute Ethernet ports among multiple Arad chips. The **show platform arad port-info routing** command displays the ports that are controlled by each chip on all Arad linecards.

Example

- This command displays the following Ethernet port distribution on the 7500E-72S-LC linecard that is inserted as module 3 in a DCS-7508E switch:
  - Arad3/0 chip: Ethernet 3/1– Ethernet 3/20
  - Arad3/1 chip: Ethernet 3/21 – Ethernet 3/34 and Ethernet 3/49/1 – Ethernet 3/49/12

- Arad3/2 chip: Ethernet 3/35 – Ethernet 3/48 and Ethernet 3/50/1 – Ethernet 3/50/12

```
switch#show platform arad mapping

Arad3/0           Port                          SysPhyPort Voq  (Fap,FapPort)
Xlge  Serdes
-------------------------------------------------------------------------------
-----
                CpuTm                            2    32    (0  ,  0)    n/a   n/a

               Ethernet3/1                      28   240    (0  ,  2)    n/a   (16)
               Ethernet3/2                      29   248    (0  ,  3)    n/a   (17)
               Ethernet3/3                      30   256    (0  ,  4)    n/a   (18)
               Ethernet3/4                      31   264    (0  ,  5)    n/a   (19)
               Ethernet3/5                      32   272    (0  ,  6)    n/a   (20)
               Ethernet3/6                      33   280    (0  ,  7)    n/a   (21)
               Ethernet3/7                      34   288    (0  ,  8)    n/a   (22)
               Ethernet3/8                      35   296    (0  ,  9)    n/a   (23)
               Ethernet3/9                      36   304    (0  , 10)    n/a   (24)
              Ethernet3/10                      37   312    (0  , 11)    n/a   (25)
              Ethernet3/11                      38   320    (0  , 12)    n/a   (26)
              Ethernet3/12                      39   328    (0  , 13)    n/a   (27)
               Ethernet3/13                      40   336    (0  , 14)    n/a   (4)
               Ethernet3/14                      41   344    (0  , 15)    n/a   (5)
               Ethernet3/15                      42   352    (0  , 16)    n/a   (6)
               Ethernet3/16                      43   360    (0  , 17)    n/a   (7)
               Ethernet3/17                      44   368    (0  , 18)    n/a   (0)
               Ethernet3/18                      45   376    (0  , 19)    n/a   (1)
               Ethernet3/19                      46   384    (0  , 20)    n/a   (2)
               Ethernet3/20                      47   392    (0  , 21)    n/a   (3)

               RawArad3/0/56                   2104 16848    (0  , 56)    n/a   n/a

Arad3/1           Port                          SysPhyPort Voq  (Fap,FapPort)
Xlge  Serdes
-------------------------------------------------------------------------------
-----

               Ethernet3/21                     60   496    (1  ,  2)    n/a   (16)

               Ethernet3/34                     73   600    (1  , 15)    n/a   (13)
               Ethernet3/49/1                    74   608    (1  , 16)    n/a   (0)

              Ethernet3/49/12                   85   696    (1  , 27)    n/a   (11)


Arad3/2           Port                          SysPhyPort Voq  (Fap,FapPort)
Xlge  Serdes
-------------------------------------------------------------------------------
-----

               Ethernet3/35                     92   752    (2  ,  2)    n/a   (16)

               Ethernet3/48                    105   856    (2  , 15)    n/a   (13)
               Ethernet3/50/1                   106   864    (2  , 16)    n/a   (0)

              Ethernet3/50/12                  117   952    (2  , 27)    n/a   (11)
```

```
switch#
```

Trident-II Fixed Switches

Trident-II platform devices distribute their ports among multiple Trident II chips. The **show platform trident system port** command displays the ports that are controlled by each chip.

Example

- This command displays the following Ethernet port distribution on a DCS-7250QX-64-F switch:
  - Trident 0 chip controls Ethernet 1/1 through Ethernet 16/4
  - Trident 1 chip controls Ethernet 17/1 through Ethernet 32/4
  - Trident 2 chip controls Ethernet 33/1 through Ethernet 48/4
  - Trident 3 chip controls Ethernet 49/1 through Ethernet 64/4

```
switch#show platform trident system port

                                                                  Port
             Intf             Chip       ModId       Logical       Physical    MMU
--------------------- ----------------- ----------- ------------- ---------- ---
       Ethernet1/1      Linecard0/0          1            1            17      9
       Ethernet1/2      Linecard0/0          1            2            18     10

      Ethernet16/3      Linecard0/0          1           60           107     98
      Ethernet16/4      Linecard0/0          1           61           108     99

      Ethernet64/2      Linecard0/3          4           62           106     97
      Ethernet64/3      Linecard0/3          4           63           107     98
      Ethernet64/4      Linecard0/3          4           64           108     99

--------------------------------------------------------------------------------
switch#
```

Trident-II Modular Switches

Linecards on 7300-Series modular switches distribute Ethernet ports among multiple Trident II chips. The **show platform trident system port** command can display the ports that are controlled by each chip on all linecards or on a single chip.

- This command displays the following Ethernet port distribution on DCS-7304-F switch that contains a 7300X-32Q-LC linecard as module 3:
  - Trident 0 chip controls Ethernet 1/1 through Ethernet 16/4 (on module 3)

- Trident 1 chip controls Ethernet 17/1 through Ethernet 32/4 (on module 3)

```
switch#show platform trident system port
          <-------OUTPUT OMITTED FROM EXAMPLE-------->
--------------------------------------------------------------------------------
                                                                      Port
             Intf             Chip       ModId      Logical       Physical    MMU
-------------------- ----------------- ----------- ------------- -------------
     Ethernet3/1/1      Linecard3/0          5            1            17      4
     Ethernet3/2/1      Linecard3/0          5            2            21      5
          <-------OUTPUT OMITTED FROM EXAMPLE-------->
    Ethernet3/16/3      Linecard3/0          5           51           111    102
    Ethernet3/16/4      Linecard3/0          5           52           112    103
          <-------OUTPUT OMITTED FROM EXAMPLE-------->
    Ethernet3/32/3      Linecard3/1          6           63           111    102
    Ethernet3/32/4      Linecard3/1          6           64           112    103
          <-------OUTPUT OMITTED FROM EXAMPLE-------->
--------------------------------------------------------------------------------
          <-------OUTPUT OMITTED FROM EXAMPLE-------->
switch#
```

3.5 Command Modes

Command modes define the user interface state. Each mode is associated with commands that perform a specific set of network configuration and monitoring tasks.

- Section 3.5.1: Mode Types lists the available modes.
- Section 3.5.2: Navigating Through Command Modes lists mode entry and exit commands.
- Section 3.5.3: Command Mode Hierarchy describes the mode structure.
- Section 3.5.4: Group-Change Configuration Modes describes editing aspects of these modes.

3.5.1 Mode Types

The switch includes these command modes:

- **EXEC**: EXEC mode commands display system information, perform basic tests, connect to remote devices, and change terminal settings. When logging into EOS, you enter EXEC mode.

  EXEC mode prompt: `switch>`

- **Privileged EXEC**: Privileged EXEC mode commands configure operating and global parameters. The list of Privileged EXEC commands is a superset of the EXEC command set. You can configure EOS to require password access to enter Privileged EXEC from EXEC mode.

  Privileged EXEC mode prompt: `switch#`

- **Global Configuration**: Global Configuration mode commands configure features that affect the entire system, such as system time or the switch name.

  Global Configuration mode prompt: `switch(config)#`

- **Interface Configuration**: Interface configuration mode commands configure or enable Ethernet, VLAN, and Port-Channel interface features.

  Interface Configuration mode prompt: `switch(config-if-Et24)#`

- *Protocol specific mode*: Protocol specific mode commands modify global protocol settings. Protocol specific mode examples include **ACL Configuration** and **Router BGP Configuration**.

  The prompt indicates the active command mode. For example, the Router BGP command prompt is `switch(config-router-bgp)#`

3.5.2 Navigating Through Command Modes

To change the active command mode, perform one of these actions:

- To enter EXEC mode, log into the switch.
- To enter Privileged EXEC mode from EXEC, type **enable** (or **en**) followed, if prompted, by the enable password:

  ```
  switch>en
  Password:
  switch#
  ```

- To enter Global Configuration mode from Privileged EXEC, type **configure** (or **config**):

  ```
  switch#config
  switch(config)#
  ```

Note EOS supports **copy <url>** ***running-config*** in place of the **configure network** command.

- To enter Interface Configuration mode from Global Configuration, type **interface** and the name of the interface to be modified:

```
switch(config)#interface Et24
switch(config-if-Et24)#
```

- To enter a protocol specific configuration mode from Global Configuration, type the required command for the desired mode.

```
switch(config)#router bgp 100
switch(config-router-bgp)#
```

- To return one level from any configuration mode, type **exit**.

```
switch(config)#exit
switch#
```

- To return to Privileged EXEC mode from any configuration mode, type **end** or **Ctrl-Z**.

```
switch(config-if-Et24)#<Ctrl-z>
switch#
```

- To return to EXEC mode from Privileged EXEC mode, type **disable** (or **dis**).

```
switch#dis
switch>
```

- To exit EOS and log out of the CLI, type **exit** from EXEC mode or Privileged EXEC mode.

```
switch#exit

login:
```

3.5.3 Command Mode Hierarchy

Command modes are hierarchical. The parent mode of a specified command mode is the mode that contains the command that enters the specified mode.

Example

- EXEC mode contains the **enable** command, which enters Privileged EXEC mode. Therefore, EXEC is the parent mode of Privileged EXEC.

Commands that are executable in a specified command mode include all commands available in the specified mode plus all commands executable from its parent mode.

Example

- EXEC mode includes the **ping** command. EXEC mode is the parent mode of Privileged EXEC mode. Therefore, Privileged EXEC mode includes **ping**.

  Additionally, Privileged EXEC is the parent mode of Global Configuration mode. Therefore, Global Configuration mode also includes **ping**.

Executing a configuration mode command from a child mode may change the active command mode.

Example

- Global Configuration mode contains **interface ethernet** and **ip access-list** commands, which enter Interface Configuration and Access Control List (ACL) Configuration modes, respectively. When the switch is in Interface Configuration mode, the **ip access-list** command is available and changes the active mode to ACL Configuration.

```
switch(config)#interface ethernet 1
switch(config-if-Et1)#ip access-list master-list
switch(config-acl-master-list)#
```

The **exit** command changes the active command mode to its parent mode. When executed from Privileged EXEC or EXEC modes, the exit command terminates the session.

Example

- This command exits Global Configuration mode to Privileged EXEC mode.

```
switch(config)#exit
switch#
```

- This command terminates the user session.

```
switch#exit
```

3.5.4 Group-Change Configuration Modes

Group-change modes apply all changes made during an edit session only after exiting the mode. Changes are stored when the user exits the mode, either through an **exit** or **end** command or through a command that enters a different configuration mode.

The **abort** command discards all changes not previously applied.

Access Control List (ACL) and Multiple Spanning Tree (MST) configuration modes are examples of group-change modes.

3.6 Managing Switch Configuration Settings

3.6.1 Verifying the Running Configuration Settings

running-config is the virtual file that stores the operating configuration. The **show running-config** command displays the ***running-config***. The command is supported in Privileged EXEC mode.

Example

- Type **show running-config** in Privileged EXEC mode. The response in the example is truncated to display only the ip route configured.

```
switch#show running-config
! Command: show running-config
             <-------OUTPUT OMITTED FROM EXAMPLE-------->
!
ip route 0.0.0.0/0 192.0.2.1
!
             <-------OUTPUT OMITTED FROM EXAMPLE-------->
end
switch#
```

3.6.2 Verifying Settings for the Current Mode

To display only the lines of ***running-config*** that affect the current mode, use the **active** option of the **show (various configuration modes)** command. This command option is available in all configuration modes except global configuration.

Example

- Type **show active** to display the content of ***running-config*** that affects the current mode. To include default settings in the display, type **show active all**.

```
switch(config-router-ospf3)#show active all
ipv6 router ospf 9
   router-id 0.0.0.0
   default-metric 10
   distance ospf intra-area 10
   area 0.0.0.200 default-cost 10
   area 0.0.0.200
   no log-adjacency-changes
   timers spf 5
switch(config-router-ospf3)#
```

To display any comments associated with the current mode, use the **comment** option of the **show (various configuration modes)** command.

Example

- Type **show comment** to display any comments attached to the current mode.

```
switch(config-router-ospf3)#show comment
Comment for router-ospf3:
    Consult Thomas Morton before making changes to the OSPF configuration.
switch(config-router-ospf3)#
```

3.6.3 Adding a Comment to a Configuration Mode

To add a comment to most switch configuration modes, use the **comment (various configuration modes)** command. Comments cannot be modified, but can be replaced by entering the **comment** command again and entering new text. Comments cannot be added to global configuration mode

To append to an existing comment, enter ! followed by additional comment text. To display comments for the active mode, use the **comment** option of the **show (various configuration modes)** command. The **no comment** and **default comment** commands remove the comment from ***running-config***.

Examples

- To add a comment to the active configuration mode, enter **comment**, then type the comment text. To end comment editing, type EOF on a separate line (case sensitive) and press **enter**.

```
switch(config-router-ospf3)#comment
Enter TEXT message. Type 'EOF' on its own line to end.
Consult Thomas Morton before making changes to the OSPF configuration.
EOF
switch(config-router-ospf3)#
```

- To append to an existing comment, enter **!** followed by additional comment text.

```
switch(config-router-ospf3)#!x2735
switch(config-router-ospf3)#show comment
Comment for router-ospf3:
    Consult Thomas Morton before making changes to the OSPF configuration.
    x2735
switch(config-router-ospf3)#
```

3.6.4 Saving the Running Configuration Settings

startup-config is the file, stored in internal flash memory, that the switch loads when it boots. Configuration changes that are not saved to ***startup-config*** are lost the next time the switch is booted.

The **write** and **copy running-config startup-config** commands store the operating configuration to ***startup-config.*** Both commands are supported in Privileged EXEC mode.

Example

- These equivalent commands save the current operating configure to the startup-config file.

```
switch#write

switch#copy running-config startup-config
```

The **show startup-config** command displays the startup configuration file. The command is supported in Privileged EXEC mode.

Example

- Type **show startup-config** to display the startup configuration file. The response in the example is truncated to display only the ip route configured in Admin Username.

```
switch#show startup-config
! Command: show startup-config
! Startup-config last modified at  Wed Feb 19 08:34:31 2014 by admin
!
              <-------OUTPUT OMITTED FROM EXAMPLE-------->
!
ip route 0.0.0.0/0 192.0.2.1
!
              <-------OUTPUT OMITTED FROM EXAMPLE-------->
end
switch#
```

3.7 Other Command-Line Interfaces

EOS can access other CLIs that provide switch commands, files, and services. .

- Section 3.7.1: Aboot Command-Line Interface describes the boot-loader CLI
- Section 3.7.2: Bash Shell describes the Bash shell CLI.

3.7.1 Aboot Command-Line Interface

Aboot is the switch boot loader. It reads a configuration file from the internal flash or a USB flash drive and attempts to boot a software image. The switch opens an Aboot shell if the switch does not find a software image, the configuration is corrupted, or the user terminates the boot process. The Aboot shell provides a CLI for manually booting a software image, recovering the internal flash to its default factory state, running hardware diagnostics, and managing files.

3.7.2 Bash Shell

The switch provides a Linux Bash shell for accessing the underlying Linux operating system and extensions. The Bash shell is accessible in all command modes except EXEC. Section 3.5.1: Mode Types describes EOC command modes.

- To enter the Bash, type **bash** at the prompt.

```
switch#bash

Arista Networks EOS shell

[admin@Switch ~]$
```

- To exit the Bash, type **logout, exit,** or **Ctrl-D** at the Bash prompt.

```
[admin@Switch ~]$ logout
switch#
```

3.8 Common Criteria (CC)

EOS firmware supports U.S. Federal Information Processing Standards (FIPS) and Common Criteria (CC) security requirements. These are enhanced security options for some Arista models.

CC consists of specifications and guidelines for the evaluation of information security products. It is used internationally to ensure that these products meet the security standards necessary for government deployments.

The primary elements of CC are Protection Profiles and Evaluation Assurance Levels. Protection Profiles define the standard security requirements for a product type. Evaluation Assurance Levels describe the thoroughness of the testing done to evaluate the product on a scale of 1-7, with one being the least thorough evaluation and seven being the most thorough. A higher Evaluation Assurance Level indicates that the product has undergone more testing, but does not necessarily correlate to a higher level of security.

United States non-military governmental security requirements for computer systems are detailed in Federal Information Processing Standards (FIPS).

Refer to the Arista Networks website for additional information at:

http://www.arista.com

3.9 Directory Structure

EOS operates from a flash drive root mounted as the **/mnt/flash** directory on the switch. The EOS CLI supports these file and directory commands:

- **delete:** Delete a file or directory tree.
- **copy:** Copy a file.
- **more:** Display the file contents.
- **diff:** Compares the contents of files located at specified URLs.
- **rename:** Rename a file
- **cd:** Change the current working directory.
- **dir:** Lists directory contents, including files and subdirectories.
- **mkdir:** Create a directory.
- **rmdir:** Remove a directory.
- **pwd:** Display the current working directory.

Verify flash memory space before copying a file. When a file is copied to flash, it is first written to a temporary file and then renamed to the destination rather than directly overwriting the destination file. This protects the integrity of the existing file if the **copy** command is interrupted, but requires more free space to complete the process.

Switch directory files are accessible through the Bash shell and Aboot. When entering the Bash shell from the switch, the working directory is located in **/home** and has the name of the user name from which Bash was entered.

Example

- These commands were entered from the user name john:

```
switch#bash
[john@switch ~]$ pwd
/home/john
[john@switch ~]$
```

  In this instance, the working directory is **/home/john**

When a flash drive is inserted in the USB flash port, flash drive contents are accessible through **/mnt/usb1**.

When entering Aboot, the working directory is the root directory of the boot.

3.10 Command-Line Interface Commands

Mode Navigation Commands

- alias
- bash
- configure (configure terminal)
- configure network
- daemon
- disable
- enable
- end
- exit

File Transfer Commands

- ip ftp client source-interface
- ip http client source-interface
- ip ssh client source-interface
- ip tftp client source-interface

File Management Commands

- copy running-config
- dir
- pwd

Modular Switch Platform Commands

- platform arad lag mode
- platform arad lag mode
- platform sand fabric mode (7500 and 7500E Series)
- platform sand forwarding mode (7500 and 7500E Series)
- platform sand lag hardware-only
- show platform sand compatibility
- show platform sand lag hardware-only

CLI Scheduling Commands

- schedule
- schedule config max-concurrent-jobs
- show schedule
- show schedule summary

Common Criteria Commands

- boot test memory
- entropy source hardware
- fips restrictions (SSH Management)
- hostkey client strict-checking (SSH Management)
- known-hosts (SSH Management)
- local (SSH Management-Tunnel)
- logging host
- logging source-interface
- logging trap system
- log-level (SSH Management)
- management security

- remote (SSH Management-Tunnel)
- secret hash
- send log message
- server-alive count-max (SSH Management-Tunnel)
- server-alive interval (SSH Management-Tunnel)
- show management ssh hostkey
- shutdown (SSH Management-Tunnel)
- ssh
- tcpdump
- tcpdump session
- tunnel (SSH Management)

Event Handler Commands

- action bash
- delay
- event-handler
- event-handler DropCountersHandler
- show event-handler
- show event-handler DropCountersHandler
- trigger

Terminal Parameter Commands

- terminal length
- terminal monitor

Display and Comment Commands

- comment (various configuration modes)
- show (various configuration modes)
- show module
- show version

action bash

The **action bash** command specifies a Bash shell command to be run when an event handler is triggered. When an event handler is triggered, execution of the associated shell command is delayed by a configurable period set by the **delay** command. Only a single Bash command may be configured for an event handler, but the command may have multiple arguments. If more than one Bash command must be executed in response to a trigger, create a script containing the desired commands and enter the file path to the script as the argument of the **action bash** command.

To specify the event that will trigger the action, use the **trigger** command.

If the event handler uses an ***onIntf*** trigger, the following environment variables are passed to the action and can be used as arguments to the Bash command:

- ***$INTF*** interface name.
- ***$OPERSTATE*** current operational status of the specified interface.
- ***$IP-PRIMARY*** current primary IP address of the specified interface.

Command Mode

Event-Handler Configuration

Command Syntax

action bash *command*

Parameters

- *command* Bash shell command to be executed when the event handler is triggered.

Example

- This command configures the event handler “onStartup” to run a script on the flash drive.

```
switch(config-handler-onStartup)#action bash /mnt/flash/myScript1
switch(config-handler-onStartup)#
```

- This command configures the event handler “eth_4” to send email to the specified address when there is a change in the operational status of Ethernet interface 4.

```
switch(config-event-eth_4)#action bash email x@yz.com -s "Et4 $OPERSTATE"
switch(config-event-eth_4)#
```

  The above action uses the ***$OPERSTATE*** variable to include the current operational state (“linkup” or “linkdown”) in the subject of the email. Note that the action will only function if email has been configured on the switch.

alias

The **alias** command creates an alias for a CLI command. Entering the alias in the CLI executes the corresponding command. Once created, an alias is accessible in all modes and all user sessions, but is subject to all the restrictions of the original command.

When using a command alias, no tokens may precede the alias except the **no** and **default** keywords. However, an alias can incorporate positional parameters.

In online help, aliases are preceded by an asterisk (*) in this format:

**alias_name=command_name*

The **no alias** and **default alias** commands remove the specified alias.

Command Mode

Global Configuration

Command Syntax

```
alias alias_name command_name
no alias alias_name
default alias alias_name
```

Parameters

- *alias_name* the string which is to be substituted for the original command. The string can include letters, numbers, and punctuation, but no spaces. If the ***alias_name*** string is identical to an existing command, the alias will supercede the original command.
- *command_name* the command which is to be executed when the alias is entered in the CLI. If the original command requires additional parameters, they must be included in the ***command_name*** string in the following manner:

  Positional parameters are of the form “%n” and must be whitespace-delimited. The first parameter is represented by “%1” and any additional parameters must be numbered sequentially. When executing the alias a value must be entered for each parameter or the CLI will display the error “% incomplete command”.

Examples

- This command makes **e** an alias for the command **enable.**

  ```
  switch(config)#alias e enable
  ```

- This command makes **srie** an alias for the command **show running-config interface ethernet 1-6.**

  ```
  switch(config)#alias srie show running-config interface ethernet 1-6
  ```

- These commands make **ss** an alias for the command **show interfaces ethernet <range> status with a positional parameter for the port range, then use the alias to display the status of ports 4/1-4/5.**

  ```
  switch(config)#alias ss show interfaces ethernet %1 status
  switch(config)#ss 4/1-4/5
  Port      Name            Status       Vlan       Duplex  Speed Type
  Et4/1                     connected    in Po1       full  10000 10GBASE-SRL
  Et4/2                     notconnect   in Po1       full  10000 10GBASE-SRL
  Et4/3                     notconnect   1            full  10000 10GBASE-SRL
  Et4/4                     notconnect   1            full  10000 10GBASE-SRL
  Et4/5                     notconnect   1            full  10000 10GBASE-SRL
  ```

bash

The **bash** command starts the Linux Bash shell. The Bash shell gives you access to the underlying Linux operating system and system extensions.

To exit the Bash, type **logout**, **exit**, or Ctrl-D at the Bash prompt.

Command Mode

Privileged EXEC

Command Syntax

bash

Examples

- This command starts the Bash shell.

```
switch#bash

Arista Networks EOS shell

[admin@switch ~]$
```

- This command, executed within Bash, exits the Bash shell.

```
[admin@switch ~]$ logout
switch#
```

boot test memory

The **boot test memory** command enables the user to set the number of iterations for a boot memory test.

In order to ensure proper operation of the switch, software and hardware checks are run continuously on and by the switch. By default the cryptographic libraries run tests to verify that they are operating correctly, EOS software processes are continually monitored and hardware health is monitored for proper functionality. Additionally, a check must be enabled for an in-kernel memory check of the Linux subsystem.

The **no boot test memory** and **default boot test memory** commands revert to the default setting by removing the corresponding **boot test memory** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
boot test memory iterations
no boot test memory
default boot test memory
```

Parameters

- *iterations* Specified number of times until a condition is met. Two iterations of the memory test are the minimum for Common Criteria mode operation. Value ranges from ***1*** to ***17***.

Examples

- This command enables two iterations of the memory test that are required for common criteria mode operation.

```
switch(config)#boot test memory 2
switch(config)#
```

comment (various configuration modes)

The **comment** command adds a comment for the active configuration mode to ***running-config***. Comments cannot be modified, but can be replaced by entering the **comment** command again and entering new text. To append to an existing comment, enter **!** followed by additional comment text. To display comments, use the **comment** option of the **show (various configuration modes)** command.

The **no comment** and **default comment** commands remove the comment from ***running-config***.

Comments cannot be added to the global configuration mode through the EOS..

Command Mode

All configuration modes except Global Configuration

Command Syntax

```
comment comment_text EOF
no comment
default comment
! comment_text
```

Parameters

- *comment_text* To create a comment, enter a message when prompted. The message may span multiple lines.
- **EOF** To append to an existing comment, enter **!** followed by additional comment text. To end comment editing, type EOF on its own line (case sensitive) and press **enter**.

Example

- This command adds a comment to the active configuration mode.

```
switch(config-sg-radius-RAD-SV1)#comment
Enter TEXT message. Type 'EOF' on its own line to end.
Consult Thomas Morton before making changes to the RADIUS configuration.
EOF
switch(config-sg-radius-RAD-SV1)#
```

- This command appends a line to the comment for the active configuration mode.

```
switch(config-sg-radius-RAD-SV1)#! x3452
switch(config-sg-radius-RAD-SV1)#
```

configure (configure terminal)

The **configure** command places the switch in Global Configuration mode to configure features that affect the entire system. This mode also provides access to Interface Configuration mode and protocol-specific modes. The command may also be entered as **configure terminal**.

The **configure network** command refers the user to Arista's **copy** <*url*> **running-config** command for configuring the switch from a local file or network location.

Command Mode

Privileged EXEC

Command Syntax

```
configure
configure terminal
```

Example

- These commands place the switch in Global Configuration mode.

```
switch>enable
switch#configure
switch(config)#
```

configure network

The **configure network** command refers the user to Arista's **copy** <*url*> **running-config** command for configuring the switch from a local file or network location.

Command Mode

Privileged EXEC

Command Syntax

configure network

Example

- This is the output of the **configure network** command.

```
switch#configure network
%% Please use copy <url> running-config
switch#
```

copy running-config

The current operating configuration of the switch is stored in a virtual file called ***running-config***. The **copy running-config** command saves the contents of the ***running-config*** virtual file to a new location.

Command Mode

Privileged EXEC

Command Syntax

copy running-config DESTINATION

Parameters

- *DESTINATION* destination for the contents of the ***running-config*** file. Values include:
  - **startup-config** the configuration file that the switch loads when it boots.

    The **copy running-config startup-config** and **write** commands are equivalent.
  - file: a file in the switch file directory.
  - flash: a file in flash memory.
  - url any valid URL.

    The **copy running-config url** and **write network url** commands are equivalent.

Examples

- This command copies ***running-config*** to the ***startup-config*** file.

```
switch#copy running-config startup-config
switch#
```

- This command copies ***running-config*** to a file called rc20110617 in the dev subdirectory of the switch directory.

```
switch#copy running-config file:dev/rc20110617
switch#
```

daemon

The **daemon** command accesses daemon configuration mode for adding or removing external daemons and scripts, which are then managed by ProcMgr.

The **no daemon** and **default daemon** commands delete the deamon by removing the corresponding **daemon** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

daemon *daemon_name*
no daemon *daemon_name*
default daemon *daemon_name*

Parameters

- *daemon_name* label that references the daemon configuration mode.

Examples

- These commands enters daemon configuration mode and initiates the daemon script.

```
switch(config)#daemon process1
switch(config-daemon-process1)#command process-script -i -m
switch(config-daemon-process1)#
```

delay

The **delay** command specifies the time in seconds the system will delay between a triggering event and the execution of an event handler action. The default delay is 20 seconds.

Command Mode

Event-Handler Configuration

Command Syntax

delay *seconds*

Parameters

- *seconds* number of seconds to delay before executing the action. The default is 20.

Example

- This command configures the event handler Eth5 to delay 10 seconds before executing.

```
switch(config-handler-Eth5)#delay 10
switch(config-handler-Eth5)#
```

dir

The **dir** command displays a list of files on a file system.

Command Mode

Privileged EXEC

Command Syntax

dir [*SCOPE*][*FILE TYPE*]

Parameters

- ***SCOPE*** the files to display. Options include
  - <no parameter> lists normal files in current directory.
  - **/all** list all files, including hidden files
  - **/recursive** list files recusively
- ***FILE TYPE*** The options include:
  - <no parameter> lists undeleted files
  - **all_filesystems** list files on all filesystems including deleted files, undeleted files, and files with errors
  - **extensions** directory or file name
  - **file** directory or file name
  - **flash** directory or file name
  - **supervisor-peer** directory or file name
  - system directory or file name
  - **usb1** directory or file name

Example

- This command displays the flash directory.

```
switch# dir flash:
Directory of flash:/

       -rwx   293409892           Oct 23 08:55  EOS-4.11.0.swi
       -rwx   221274543            Sep 6 13:37  EOS-4.7.5.swi
       -rwx   271453650            Sep 4 19:13  EOS_4.10.1-SSO.swi
       -rwx      135168           Dec 31  1979  FSCK0000.REC
       -rwx          26           Oct 23 13:51  boot-config
       -rwx        8570           Sep 10 12:22  cfg_sso_mst
       -rwx        5642           Sep 20 10:35  config.reset
       drwx        4096           Oct 23 13:59  debug
       -rwx          12           Oct 23 13:56  kernel-params
       drwx        4096           Oct 23 14:59  persist
       drwx        4096            Sep 6 14:50  schedule
       -rwx        5970           Oct 23 13:53  startup-config
switch#
```

disable

The **disable** command exchanges the session's current command mode with the specified privilege level.

Command Mode

Privileged EXEC

Command Syntax

disable [*PRIVILEGE_LEVEL*]

Parameters

- ***PRIVILEGE_LEVEL*** Session's new privilege level. Value ranges from 0 to 15. Levels 2 through 15 place the switch in Privileged EXEC mode. Values of 0 or 1 leave the switch in EXEC mode.
  - <no parameter> Session is assigned default level of 1.
  - <***0*** to ***15***> Specifies session level.

Restrictions

New privilege level must be less than the session's current level.

Examples

- This command exits Privileged EXEC mode level of 15 to enter EXEC mode level 1.

```
switch# disable
switch>
```

enable

The **enable** command places the switch in Privileged EXEC mode. If an ***enable*** password is set, the CLI displays a password prompt when a user enters the **enable** command. If the user enters an incorrect password three times, the CLI displays the EXEC mode prompt.

To set a local ***enable*** password, use the **enable secret** command.

Command Mode

EXEC

Command Syntax

enable [*PRIVILEGE_LEVEL*]

Parameters

- ***PRIVILEGE_LEVEL*** Session's privilege level. Values range from 0 to 15. Values of 0 or 1 places the switch in EXEC mode. Any level above 1 leaves the switch in Privileged EXEC mode.
  - <no parameter> Session is assigned default level of 15.
  - <***0*** to ***15***> Specifies session level.

Example

- This command places the switch in Privileged EXEC mode with the default privilege level of 15.

```
switch>enable
switch#
```

end

The **end** command exits to Privileged Exec mode from any Configuration mode. If the switch is in a group-change mode (such as ACL-Configuration mode or MST-Configuration mode), the **end** command also saves all pending changes made in that mode to ***running-config***.

Command Mode

All configuration modes

Command Syntax

end

Example

- This command exits to Privileged Exec mode.

```
switch(config-if-Et25)#end
switch#
```

entropy source hardware

The **entropy source hardware** command specifies that the switch must use its hardware-based random number generator from a physical process.

Entropy is a measure of randomness in a system. An entropy source is a device that gathers quantum randomness from a physical system.

The **no entropy source hardware** and **default entropy source hardware** commands disable the hardware-based random number generator.

Command Mode

Mgmt-security Configuration

Command Syntax

```
entropy source hardware
no entropy source hardware
default entropy source hardware
```

Examples

- This command enables the hardware random number generator.

```
switch(config)#management security
switch(config-mgmt-security)#entropy source hardware
```

- Use the following command to verify that entropy generation is enabled.

```
switch#show management security
CPU Version: 03.02
Hardware Version: 04.00
Security Chip Version: R5H30211
Hardware Entropy Generation is enabled
```

event-handler

An event handler executes a Linux Bash shell command in response to a specific system event. An event handler consists of a Bash command, a trigger and a delay; when the trigger event occurs, the action is scheduled to run after ***delay*** seconds.

The **event-handler** command places the switch in event-handler configuration mode for the specified event handler. If the named event handler does not already exist, this command creates it. Event-handler configuration mode is a group change mode that configures event handlers.

Changes made in a group change mode are saved by leaving the mode through the **exit** command or by entering another configuration mode.

These commands are available in event-handler configuration mode:

- **action bash**
- **delay**
- **trigger**

The **no event-handler** and **default event-handler** commands delete the specified event handler by removing it from ***running config***.

Command Mode

Global Configuration

Command Syntax

```
event-handler name
no event-handler name
default event-handler name
```

Parameters

- *name* name of the event handler to be configured. If the named event handler does not already exist, this command will create it.

Example

- This command places the switch in event-handler configuration mode for an event handler called “Eth_5”.

```
switch(config)#event-handler Eth_5
switch(config-handler-Eth_5)#
```

event-handler DropCountersHandler

The **event-handler DropCountersHandler** command enables the adverse drop counters monitor with event handlers. The DropCountersHandler event handler is enabled by default, and can be customized for duration of time window and threshold levels.

The **no event-handler DropCountersHandler** command disables the adverse drop counters monitor with event handlers. The **default event-handler DropCountersHandler** command resets the DropCountersHandler event handler to the system default.

Command Mode

Global Configuration

Command Syntax

```
event-handler DropCountersHandler
no event-handler DropCountersHandler
default event-handler DropCountersHandler
```

Examples

- These commands customize the delay, polling interval, and condition for width (-w), violation count (-c), and threshold (-t) of this event handler. Each parameter may be customized separately, with all other parameters remaining unchanged.

```
switch(config)#event-handler DropCountersHandler
switch(config-DropCountersHandler)#action bash DropCounterLog.py -l
switch(config-DropCountersHandler)#delay 0
switch(config-DropCountersHandler)#trigger on-counters
switch(config-DropCountersHandler-counters)#poll interval 60
switch(config-DropCountersHandler-counters)#condition
bashCmd."DropCounterMonitor.py" -w 800" > 0
switch(config-DropCountersHandler-counters)#condition
bashCmd."DropCounterMonitor.py" -c 5" > 0
switch(config-DropCountersHandler-counters)#condition
bashCmd."DropCounterMonitor.py" -t 200" > 0
switch(config-DropCountersHandler-counters)#
```

- This command disables this event handler.

```
switch(config)#no event-handler DropCountersHandler
switch(config)#
```

exit

The **exit** command places the switch in the parent of the command mode from which the exit command was entered.

- When used in Global configuration, the switch enters Privileged EXEC mode.
- When used in EXEC or Privileged EXEC mode, the **exit** command terminates the user session.
- When the command is used in a group-change mode (such as ACL-Configuration mode or MST-Configuration mode), the **exit** command also applies all pending changes made in that mode.

Command Mode

All modes

Command Syntax

```
exit
```

Example

- This command exits Global Configuration mode to Privileged EXEC mode.

```
switch(config)#exit
switch#
```

- This command terminates the user session.

```
switch#exit
```

fips restrictions (SSH Management)

The **fips restrictions** command enables the switch to use FIPS-validated encryption algorithms to fulfill Common Criteria requirements.

The **no fips restrictions** and **default fips restrictions** commands restore default behavior by removing the **fips restrictions** statement from ***running-config*.**

Command Mode

Mgmt-ssh Configuration

Command Syntax

```
fips restrictions
no fips restrictions
default fips restrictions
```

Examples

- These commands configure the switch to use FIPS-validated encryption algorithms to fulfill the Common Criteria requirements.

```
switch(config)#management ssh
switch(config-mgmt-ssh)#fips restrictions
```

hostkey client strict-checking (SSH Management)

The **hostkey client strict-checking** command specifies how host keys are checked during the connection and authentication phase. By default strict host key checking is disabled. When disabled the SSH client verifies the incoming host key against the keys in the known hosts list. If the host key does not match an existing known host entry for the remote server, the connection is rejected. If the known host list does not contain a host key for the for the remote server, the SSH client automatically accepts the host and adds its host key to the known host list.

When strict host key checking is enabled, the SSH client connects only to known hosts with valid SSH host keys that are stored in the known hosts list. Host keys not listed in the known host list are rejected.

The **no hostkey client strict-checking** and **default hostkey client strict-checking** commands revert to its default by removing the corresponding **hostkey client strict-checking** command from ***running-config***.

Command Mode

Mgmt-ssh Configuration

Command Syntax

```
hostkey client strict-checking
no hostkey client strict-checking
default hostkey client strict-checking
```

Examples

- These commands specify how host keys are checked during the connection and authentication phase.

```
switch(config)#management ssh
switch(config-mgmt-ssh)# hostkey client strict-checking
```

ip ftp client source-interface

By default, the FTP (File Transfer Protocol) source IP address is selected by the switch (the IP address of the source interface if one is assigned). The **ip ftp client source-interface** command allows the user to override the default FTP source address.

The **ip ftp client source-interface** and **ip ftp source-interface** commands are functionally equivalent. In each case, **ip ftp client source-interface** is stored in ***running-config***

The **no ip ftp client source-interface** and **default ip ftp client source-interface** commands restore default behavior by removing the **ip ftp client source-interface** statement from ***running-config*.**

Command Mode

Global Configuration

Command Syntax

```
ip ftp [client] source-interface INTERFACE
no ip ftp [client] source-interface
default ip ftp [client] source-interface
```

Parameters

- **client** Parameter has no functional effect.
- ***INTERFACE*** Interface providing the IP address. Options include:
  - **ethernet** *e_num* Ethernet interface specified by *e_num*.
  - **loopback** *l_num* Loopback interface specified by *l_num*.
  - **management** *m_num* Management interface specified by *m_num*.
  - **port-channel** *p_num* Port-channel interface specified by *p_num*.
  - **vlan** *v_num* VLAN interface specified by *v_num*.

Examples

- These commands configure the 10.10.121.15 as the source IP address the switch uses when communicating with FTP servers.

```
switch(config)#interface ethernet 17
switch(config-if-Et17)#ip address 10.10.121.15/24
! IP configuration will be ignored while interface Ethernet17 is not a routed
port.
switch(config-if-Et17)#ip ftp client source-interface ethernet 17
switch(config-if-Et17)#
```

ip http client source-interface

The **ip http client source-interface** command specifies the source IP address for hypertext transfer protocol (HTTP) connections. By default, the source IP address is selected by the switch when this command is not configured or when the specified interface is not assigned an IP address.

The **no ip http client source-interface** and **default ip http client source-interface** commands restore default behavior by removing the **ip http client source-interface** statement from ***running-config*.**

Command Mode

Global Configuration

Command Syntax

```
ip http client source-interface INTERFACE
no ip http client source-interface
default ip http client source-interface
```

Parameters

- ***INTERFACE*** Interface providing the IP address. Options include:
  - **ethernet** *e_num* Ethernet interface specified by *e_num*.
  - **loopback** *l_num* Loopback interface specified by *l_num*.
  - **management** *m_num* Management interface specified by *m_num*.
  - **port-channel** *p_num* Port-channel interface specified by *p_num*.
  - **vlan** *v_num* VLAN interface specified by *v_num*.

Examples

- These commands configure the 10.15.17.9 as the source IP address the switch uses when communicating with http servers.

```
switch(config)#interface vlan 10
switch(config-if-Vl10)#ip address 10.15.17.9/24
switch(config-if-Vl10)#ip http client source-interface vlan 10
switch(config)#
```

ip ssh client source-interface

The **ip ssh client source-interface** command specifies the source IP address for secure shell (SSH) connections. By default, the source IP address is selected by the switch when this command is not configured or when the specified interface is not assigned an IP address.

The **ip ssh client source-interface** and **ip ssh source-interface** commands are functionally equivalent. In each case, **ip ssh client source-interface** is stored in ***running-config***

The **no ip ssh client source-interface** and **default ip ssh client source-interface** commands restore default behavior by removing the **ip ssh client source-interface** statement from ***running-config*.**

Command Mode

Global Configuration

Command Syntax

```
ip ssh [client] source-interface INTERFACE
no ip ssh [client] source-interface
default ip ssh [client] source-interface
```

Parameters

- **client** Parameter has no functional effect.
- ***INTERFACE*** Interface providing the IP address. Options include:
  - **ethernet** *e_num* Ethernet interface specified by *e_num*.
  - **loopback** *l_num* Loopback interface specified by *l_num*.
  - **management** *m_num* Management interface specified by *m_num*.
  - **port-channel** *p_num* Port-channel interface specified by *p_num*.
  - **vlan** *v_num* VLAN interface specified by *v_num*.

Examples

- These commands configure the 10.17.17.9 as the source IP address the switch uses when communicating with ssh servers.

```
switch(config)#interface vlan 10
switch(config-if-Vl10)#ip address 10.15.17.9/24
switch(config-if-Vl10)#ip ssh client source-interface vlan 10
switch(config)#
```

ip tftp client source-interface

The **ip tftp client source-interface** command specifies the source IP address for Trivial File Transfer Protocol (TFTP) connections. By default, the source IP address is selected by the switch when this command is not configured or when the specified interface is not assigned an IP address.

The **ip tftp client source-interface** and **ip tftp source-interface** commands are functionally equivalent. In each case, **ip tftp client source-interface** is stored in ***running-config***

The **no ip tftp client source-interface** and **default ip tftp client source-interface** commands restore default behavior by removing the **ip tftp client source-interface** statement from ***running-config*.**

Command Mode

Global Configuration

Command Syntax

```
ip tftp [client] source-interface INTERFACE
no ip tftp [client] source-interface
default ip tftp [client] source-interface
```

Parameters

- **client** Parameter has no functional effect.
- ***INTERFACE*** Interface providing the IP address. Options include:
  - **ethernet** *e_num* Ethernet interface specified by *e_num*.
  - **loopback** *l_num* Loopback interface specified by *l_num*.
  - **management** *m_num* Management interface specified by *m_num*.
  - **port-channel** *p_num* Port-channel interface specified by *p_num*.
  - **vlan** *v_num* VLAN interface specified by *v_num*.

Examples

- These commands configure the 10.15.17.9 as the source IP address the switch uses when communicating with tftp servers.

```
switch(config)#interface vlan 10
switch(config-if-Vl10)#ip address 10.15.17.9/24
switch(config-if-Vl10)#ip tftp client source-interface vlan 10
switch(config)#
```

known-hosts (SSH Management)

The **known-hosts** command configures the MD5 fingerprint of the SSH server's host key. The local SSH client uses this fingerprint to authenticate the server, which should return a matching fingerprint.

Note Note: the fingerprint must be re-entered after a system reboot.

The **no known-hosts** and **default known-hosts** commands revert the specified list configuration to its default by removing the corresponding **known-hosts** command from ***running-config***. By default, there is no fingerprint configured and the SSH server will not be authenticated.

Command Mode

Mgmt-ssh Configuration

Command Syntax

```
known-hosts SERVER ALGRORITHM key_string
no known-hosts
default known-hosts
```

Parameters

- ***SERVER*** Location of SSH server. Options include:
  - *ip_address* IP address of the SSH server
  - *hostname* Hostname of the SSH server
- ***ALGORITHM*** connection type of sessions for which authentication list is used
  - **dsa** Fingerprint was created using DSA (Digital Signature Algorithm).
  - **rsa** Fingerprint was created using RSA encryption.
- *key_string* MD5 fingerprint string (Base-64).

Examples

- This command configures an entry for a server called "tacplus" that is connected to by an SSH tunnel:

```
switch(config-mgmt-ssh)#known-hosts tacplus rsa
AAAAB3NzaC1yc2EAAAADAQABAAABAQDxBBRbatWBM7ubC7TQMECNcHVqxhTEo4kRl
8cbw8dAtVUtQUlhBJJBRul8cqTv6lVnw7Rl5+05kuROGHDVNx3mbXwnyWBVhgorw7
RvNEwW46SW280XaD1NcCCJM42sJk+xO6qAIRj3L7WhobDU05HR9JrrmKZcBxR5VTK
L0a7zOOM2NrVOi/Uf6fJ1m0NktxDlJQJnoqdxupc5fkxqOdAURLtuP5H+pRyPEXrB
zTk1O07EaNG6ZhMgTBjxISSNKR48dM0WRXjc+6loVpAqDvdkDa4kDmsRb9QRuzNpI
XYrhzOQXf+nDIbdrVta77oPuwwb3M35OStFFaUB4/nilCs3
switch(config-mgmt-ssh)#
```

local (SSH Management-Tunnel)

The **local** command specifies the local binding on the switch for an SSH Tunnel.

The **no local** and **default local** commands remove the **local** command from the configuration**.**

Command Mode

Mgmt-SSH-Tunnel

Command Syntax

```
local port port_number
no local
default local
```

Parameters

- *port_number* port number. Value ranges from ***1*** to ***32767***.

Example

- This command enables a tunnel named “test”. This tunnel will bind to local port 49 on the switch.

```
switch(config)#management ssh
switch(config-mgmt-ssh)#
switch(config-mgmt-ssh)#tunnel test
switch(config-mgmt-ssh-tunnel-Test)#local port 49
```

logging host

The **logging host** command specifies a remote host to receive syslog messages generated by the switch. ***Running-config*** can contain multiple logging host statements.

A Common Criteria compliant switch must connect to a syslog audit server to record syslog messages generated. The syslog server being connected to must implement RFC 5424 to be considered capable of using the syslog protocol.

The default port for syslog is 514. The host must be localhost and the protocol TCP for SSH Tunneling.

The **no logging host** and **default logging host** commands clear the specified method list by removing the corresponding **logging host** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
logging [VRF_INSTANCE] host syslog_host [PORT] [PROT_TYPE]
no logging [VRF_INSTANCE] host syslog_host
default logging [VRF_INSTANCE] host syslog_host
```

Parameters

- ***VRF_INSTANCE*** specifies the VRF instance being modified.
  - <no parameter> changes are made to the default VRF.
  - **vrf** *vrf_name* changes are made to the specified user-defined VRF.
- *syslog_host* remote syslog server location. Valid formats include hostname or IPv4 address.
- ***PORT*** Remote syslog server port that handles syslog traffic. Options include:
  - <no parameter> Default port number 514.
  - <***1*** to ***65535***> Port number.
- ***PROT_TYPE*** Specifies the transport protocol for packets. Options include:
  - <no parameter> Packets transported by User Datagram Protocol (UDP).
  - **protocol tcp** Packets transported by TCP.
  - **protocol udp** Packets transported by User Datagram Protocol (UDP).

Examples

- This command logs system messages to a host with an IP address of 172.1.1.63.

```
switch(config)#logging host 172.1.1.63
switch(config)#
```

- A Common Criteria compliant switch must connect to a syslog audit server to record syslog messages generated. The syslog server on the switch should be configured as follows:

```
switch(config)#logging host localhost protocol tcp
switch(config)#logging trap informational
switch(config)#logging trap system tag ntpd contain clock_step
switch(config)#logging trap system tag sshd
switch(config)#
```

logging source-interface

The **logging source-interface** command specifies a local interface as the source for UDP packets sent to a syslog server (a process known as “source spoofing”).

Important! Source spoofing cannot be used with TCP, and is therefore incompatible with all TCP syslog usage including Common Criteria remote logging.

The **no logging source-interface** and **default logging source-interface** commands restore the default source address for syslog packets by removing the corresponding **logging source-interface** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
logging source-interface INTERFACE
no logging source-interface
default logging source-interface
```

Parameters

- ***INTERFACE*** Interface providing the source address of outgoing UDP packets sent to the logging server. Values include:
  - **ethernet** *interface* Ethernet interface specified by *interface*.
  - **loopback** *interface* Loopback interface specified by *interface*.
  - **management** *interface* Management interface specified by *interface*.
  - **port-channel** *interface* Port-channel interface specified by *interface*.
  - **vlan** *interface* VLAN interface specified by *interface*.

Related Commands

- The **logging host** command specifies the transport protocol used to communicate with the remote syslog server. Source spoofing is only supported when communicating with the syslog server using UDP; to use source spoofing, use the **logging host** command to specify UDP as the transport protocol.

Example

- These commands configure packets sent to a syslog server at 198.l62.3.5 to use the IP address of Ethernet port 5 as their source address.

```
switch(config)#logging host 198.162.3.5 protocol udp
switch(config)#logging source-interface ethernet 5
switch(config)#
```

logging trap system

The **logging trap system** command configures remote logging of system messages. Specifying a severity level logs only those messages with a severity at or above that level to the remote server. To configure the IP address of the remote syslog server, use the **logging host** command; to enable logging, use the **logging on** command.

The **no logging trap system** and **default logging trap system** commands restore remote logging defaults by removing the corresponding **logging trap system** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
logging trap system [FACILITY] [SEVERITY] [PROGRAM] [TEXT]
no logging trap system [FACILITY] [SEVERITY] [PROGRAM] [TEXT]
default logging trap system [FACILITY] [SEVERITY] [PROGRAM] [TEXT]
```

The ***TEXT*** parameter, when present, is always last. All other parameters can be placed in any order.

Parameters

- ***FACILITY*** Defines the appropriate facility.
  - <no parameter> Specifies default facility.
  - **facility** <*facility-name*> Specifies named facility.
- ***SEVERITY*** Specifies minimum severity level to be logged. Options include:
  - <no parameter> Specifies default severity level.
  - **severity** <*level*> Minimum severity level for remote logging.

    Valid *level* options include:
    - **0** or **emergencies** System is unusable
    - **1** or **alerts** Immediate action needed
    - **2** or **critical** Critical conditions
    - **3** or **errors** Error conditions
    - **4** or **warnings** Warning conditions
    - **5** or **notifications** Normal but significant conditions
    - **6** or **informational** Informational messages
    - **7** or **debugging** Debugging messages
- ***PROGRAM*** Filters packets based on program name. Options include:
  - <no parameter> All tags or program names.
  - **tag** *program-name* Specific tag or program name.
- ***TEXT*** Specifies log message text. Options include:
  - <no parameter> Specify text contained in log message.
  - **contain** *reg-expression* Specify text contained in log message.

Examples

- This command enables the logging of system informational messages to a remote server.

```
switch(config)#logging trap informational
switch(config)#
```

log-level (SSH Management)

The **log-level** command configures the verbosity level that is used when logging messages from SSH.

The **no log-level** and **default log-level** commands revert the verbosity level to its default by removing the corresponding **log-level** command from ***running-config***.

Command Mode

Mgmt-ssh Configuration

Command Syntax

```
log-level MESSAGE_LEVEL
no log-level
default log-level
```

Parameters

- ***MESSAGE_LEVEL*** Specifies level of detail in debug messages. The higher the number, the more detail that is logged. Higher levels include all lower levels.
  - **debug** production logging levels
  - **debug1** informational logging levels
  - **debug2** higher logging levels
  - **debug3** extended higher logging levels
  - **error** non-fatal errors
  - **fatal** severe errors likely to terminate SSH
  - **info** informational messages
  - **quiet** only fatal errors
  - **verbose** debugging messages about the progress of SSH

Examples

- This command configures the switch to log SSH messages of level “error.”

```
switch(config)#management ssh
switch(config-mgmt-ssh)#log-level error
switch(config-mgmt-ssh)#
```

management security

The **management security** command places the switch in mgmt-security configuration mode.

The **no management security** and **default management security** commands delete all mgmt-security configuration mode statements from ***running-config***.

Mgmt-security configuration mode is not a group change mode; ***running-config*** is changed immediately upon entering commands. Exiting mgmt-security configuration mode does not affect ***running-config***. The **exit** command returns the switch to global configuration mode.

Command Mode

Global Configuration

Command Syntax

```
management security
no management security
default management security
```

Example

- This command places the switch in mgmt-security configuration mode:

```
switch(config)#management security
switch(config-mgmt-security)#
```

- This command returns the switch to global management mode:

```
switch(config-mgmt-security)#exit
switch(config)#
```

platform arad lag mode

The **platform arad lag mode** command allows configuration of LAGs with more than 16 members.

Command Mode

Global Configuration

Command Syntax

platform arad lag mode [1024x16 | 256x64 | 512x32]

Examples

- This command configures 1024 LAGs with 16 members each.

```
switch(config)# platform arad lag mode 1024x16
! Change will take effect only after switch reboot.
switch(config)#
```

- This command configures 256 LAGs with 64 members each.

```
switch(config)# platform arad lag mode 256x64
! Change will take effect only after switch reboot.
switch(config)#
```

- This command configures 512 LAGs with 32 members each.

```
switch(config)# platform arad lag mode 512x32
! Change will take effect only after switch reboot.
switch(config)#
```

platform sand fabric mode (7500 and 7500E Series)

The **platform sand fabric mode** command specifies the fabric mode under which the switch operates after the next system reload. The command has no operational effect until the switch reloads.

The fabric mode determines the modular switch's fabric performance capabilities and must be compatibile with the installed fabric modules. Fabric mode settings include:

- ***fe600:*** Supports first-generation fabric modules.
- ***fe1600:*** Supports E-Series fabric modules.

Important! Switches that reload in ***petraA*** forwarding compatibility mode (**platform sand forwarding mode (7500 and 7500E Series)**) also reload in ***fe600*** fabric mode regardless of the presence of a **platform sand fabric mode** statement in ***running-config***.

The switch's fabric mode setting must match the capabilities of its installed fabric modules. Reloading the switch in a different mode may be required after exchanging fabric modules for a different module type. The **show module** command displays the fabric modules in the switch.

Each fabric module is categorized as first-generation or E-Series:

- First-generation fabric modules support all basic switch functions.
- E-Series fabric modules support faster fabric link speeds, greater internal table capacities, and advanced encoding formatting.

E-series fabric modules can operate in ***fe600*** mode, but are limited to first-generation fabric performance. First-generation modules cannot operate in ***fe1600*** mode. Switches containing both types of modules must be set to **fe600** mode. Best practice is to avoid switch configurations with mixed fabric modules.

When a switch reloads, fabric mode is determined by the following (in order of precedence):

Step 1 Switches reloading in ***petraA*** forwarding compability mode also reload in ***fe600*** fabric mode .

Step 2 As specified by the **platform sand fabric mode** statement in ***running-config***.

Step 3 The first fabric module that becomes operational as the switch reloads.

In switches with a homogeneous module set, the fabric mode matches its fabric modules. Switches with a mixed set of modules are typically reloaded in ***fe600*** mode because first generation modules are usually operational before E-Series modules. However, the fabric mode in mixed module switches that are reloading cannot be guaranteed in the absence of the first two conditions.

The **no platform sand fabric mode** and **default platform sand fabric mode** commands remove the **platform sand fabric mode** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
platform sand fabric mode [MODE_SETTING]
no platform sand fabric mode
default platform sand fabric mode
```

Parameters

- ***MODE_SETTING*** Specifies the switch's fabric mode. Options include:
  - **fe16000** E-Series fabric mode.
  - **fe600** First-generation fabric mode.

Examples

- This command configures the switch to reload in ***fe1600*** fabric mode to support E-series fabric modules. After issuing this command, the switch should be reset only after exchanging all switch fabric modules to E-series modules.

```
switch(config)#platform sand fabric mode fe1600
switch(config)#exit
switch#show platform sand compatibility
                    Configuration        Status
Forwarding mode     None                 Arad
Fabric mode         Fe1600               Fe600
switch#
```

platform sand forwarding mode (7500 and 7500E Series)

The **platform sand forwarding mode** command specifies the forwarding compatibility mode under which the switch operates after the next system reload. The command has no operational effect until the switch reloads.

Forwarding compatibility mode specifies switch forwarding capabilities and configures performance capacity of installed linecards. Forwarding compatibility modes settings include:

- ***petraA:*** Supports first-generation fabric modules.
- ***arad:*** Supports E-Series fabric modules.

Important! Switches that reload in ***petraA*** forwarding compatibility mode also reload in ***fe600*** fabric mode regardless of the presence of a **platform sand fabric mode (7500 and 7500E Series)** statement in ***running-config***.

This command may be required after exchanging a linecard for a different module type or in switches containing first-generation and E-series linecards. The **show module** command displays the linecard modules in the switch.

Each modular switch linecard module is categorized as first-generation or E-Series:

- First-generation linecards support all basic switch functions.
- E-Series linecards support provide faster data processing, greater internal table capacities, and advanced encoding formatting.

The forwarding compatibility mode determines the operational capacity of installed linecards. Table 3-3 lists the affect of the forwarding compatibility mode on all linecard module types.

Table 3-3 Linecard Module and Forwarding Mode Performance

Linecard Module Type	Forwarding Software Mode	Linecard Operating Capacity
First-generation	petraA	Linecard performs at first-generation performance capacity.
First-generation	arad	Linecard is powered-down.
E-Series	petraA	Linecard performs at first-generation performance capacity.
E-Series	arad	Linecard performs at E-series performance capacity.

Important! Linecards operate at E-Series performance capacities only on switches that contain E-Series fabric modules and have a fabric mode seting of ***fe1600*** fabric mode (**platform sand fabric mode (7500 and 7500E Series)**).

Without a **platform sand forwarding mode** command, forward compatibility mode is determined by the first linecard that becomes operational after reloading the switch. In a switch that is reloaded with a homogeneous module set, forwarding compatibility mode matches its linecards. Switches with a mixed set of modules are typically reloaded in ***petraA*** mode because first generation modules are usually operational before E-Series modules. However, forwarding compatibility mode in mixed module switches that are reloading is not guaranteed without a **platform sand forwarding mode** command.

The **no platform sand forwarding mode** and **default platform sand forwarding mode** commands restore the **platform sand forwarding mode** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
platform sand forwarding mode [MODE_SETTING]
no platform sand forwarding mode
default platform sand forwarding mode
```

Parameters

- ***MODE_SETTING*** Specifies the switch's software forwarding mode. Options include:
  - **arad** the switch supports E-Series linecard capabilities.
  - **petraA** the switch supports first-generation linecard capabilities.

Examples

- This command changes the forwarding software mode to support E-series linecard modules. This command should be run only after exchanging all linecards to E-series modules.

```
switch(config)#platform sand forwarding mode arad
switch(config)#
```

platform sand lag hardware-only

The **platform sand lag hardware-only** command specifies that all LAGs will use hardware resources including single member LAGs. Hardware resource allocation and deallocation traffic disruption occurs on the first member addition or deletion, rather than the second member addition or deletion.

The **no platform sand lag hardware-only** and **default platform sand lag hardware-only** commands specify that LAGs are not required to be implemented in hardware, and therefore some LAGs may be implemented in software. Permitting both hardware and software LAGs may increase the total number of port-channels because we have no resource limit on the number of software LAGs.

Command Mode

Global Configuration

Command Syntax

```
platform sand lag hardware-only
no platform sand lag hardware-only
default platform sand lag hardware-only
```

Examples

- This command configures all LAGs to use hardware resources. All existing one member LAGs will be allocated hardware resources, when available.

```
switch(config)#platform sand lag hardware-only
switch(config)#
```

- This command allows certain LAGs (single member LAGs) to not consume hardware resources. All existing one member LAGs will release their hardware resources.

```
switch(config)#no platform sand lag hardware-only
switch(config)#
```

pwd

The **pwd** command displays the working directory.

Command Mode

Privileged EXEC

Command Syntax

pwd

Examples

- This command shows that the working is Flash.

```
switch# pwd
flash:/
switch#
```

remote (SSH Management-Tunnel)

The **remote** command provides a SSH tunneling feature to transparently secure TCP connections to remote servers. This feature can accept any TCP connection that goes to the CPU on the switch and establish a standard SSH tunnel to a remote host. This has a twofold advantage:

- TCP based services do not need to perform any encryption within their own protocol.
- The connection is secure against replay attacks, manipulation and eavesdropping via SSHv2 Common Criteria compliant encryption.

In order to support this the server the switch will connect on will need to support the following:

- The SSH Server will need to have TCP forwarding allowed for the user account the SSH Tunnel will connect to. On OpenSSH's sshd implementation this is the "AllowTcpForwarding" option.
- Public key login for users. The SSH Tunneling feature does not support password based login and uses the switch's SSH keys instead to perform authentication.
- As an additional Common Criteria requirement the switch needs to know the hostkey it is connecting to in advance to prevent attacks where the connection is intercepted. To do this, enter the hostkey in either the management ssh mode for the main vrf or a vrf submode. (Refer to **hostkey client strict-checking (SSH Management)** and **known-hosts (SSH Management)** commands for further information.)

The **no remote** and **default remote** commands remove the **remote** command from the configuration**.**

Command Mode

Mgmt-SSH-Tunnel

Command Syntax

```
remote host_addr host_port
no remote
default remote
```

Parameters

- *host_addr* IP address or host name.
- *host_port* port number. Value ranges from ***1*** to ***32767***.

Example

- The following commands will configure and enable a tunnel named "bar". This tunnel will bind to local port 49 on the switch. The tunnel will then connect to a server named "tacplus" with a SSH server listening on the standard port 22. The user account on tacplus that the tunnel connects to will be called "authuser". From that connection the tunnel will bind to port 49 on the tacplus server.

```
switch(config-mgmt-ssh)#tunnel bar
switch(config-mgmt-ssh-tunnel-bar)#local port 49
switch(config-mgmt-ssh-tunnel-bar)#ssh-server tacplus user authuser port 22
switch(config-mgmt-ssh-tunnel-bar)#remote host localhost port 49
```

schedule

The **schedule** command facilitates the periodic execution of a specified CLI command. Command parameters configure the start time of periodic execution, the interval between consecutive execution instances, and the maximum number of files that can be created. By default, periodic execution of the following ***show tech-support*** command is enabled:

```
schedule tech-support interval 60 max-log-files 100 command show tech-support
```

Text that the CLI normally displays as a result of executing the scheduled command through the CLI is stored in log files at ***flash:/schedule/<sched_name>***. Empty log files are created for commands that do not generate CLI text.

The **no schedule** and **default schedule** commands disable execution of the specified command by removing the corresponding **schedule** statement from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
schedule name interval PERIOD max-log-files files [logging verbose] command cmd
no schedule sched_name
default schedule sched_name
```

Parameters

- *name* label associated with the scheduled command.
- ***PERIOD*** start time for execution and interval between consecutive execution instances. Options include:
  - **at *hh:mm:ss* interval** <***1*** to ***1440***> The command is executed at the next *hh:mm:ss* and repeated every *interval* minutes.
  - **at *hh:mm:ss* once** The command is executed at the next *hh:mm:ss* and not repeated.
  - **at *hh:mm:ss mm/dd/yyyy* interval** <***1*** to ***1440***> The command is executed at *hh:mm:ss* on *mm/dd/yyyy* and repeated every *interval* minutes.
  - **at *hh:mm:ss mm/dd/yyyy* once** The command is executed at *hh:mm:ss* on *mm/dd/yyyy* and not repeated.
  - **at *hh:mm:ss yyyy-mm-dd* interval** <***1*** to ***1440***> The command is executed at *hh:mm:ss* on *yyyy-mm-dd* and repeated every *interval* minutes.
  - **at *hh:mm:ss yyyy-mm-dd* once** The command is executed at *hh:mm:ss* on *yyyy-mm-dd* and not repeated.
  - **interval** <***1*** to ***1440***> The command is executed immediately and repeated every *interval* minutes.
  - **now interval** <***1*** to ***1440***> The command is executed immediately and repeated every *interval* minutes.
- *files* maximum number of log files command generates for command output. Range is 1 to 10000.
- **logging verbose** Sets the logging level to “verbose.” A syslog entry is added after the execution of the scheduled command, regardless of whether the scheduled command has succeeded or failed. In the absence of **logging verbose**, the syslog entry is added *only* if the execution of the scheduled command fails with an error.
- *cmd* name of the CLI command.

Guidelines

Log files created by the command are stored in the **flash:/schedule/<*sched_name*>/** directory.

Examples

- This command schedules the execution of a script file once every 12 hours, beginning at noon. The log file option is set to the option minimum of one because the command does not generate output to the CLI.

  ```
  switch(config)#schedule ms_1 at 12:00:00 interval 720 max-log-files 1 command
  bash /mnt/flash/myscript.sh
  ```

  The **show schedule summary** command displays the commands that are scheduled for periodic execution.

  ```
  switch(config)#show schedule summary
  Maximum concurrent jobs  2
  Name         At time Last  Inter\  Max   Logfile Location              Status
                       time   val    log
                             (mins)  files
  ------------ ------- ----- ------- ----- ----------------------------- ------
  tech-support   now   09:41   60    100   flash:/schedule/tech-support/  Success
  ms_1           now   16:28  720    10    flash:/schedule/ms_1
  switch#
  ```

- This command stores ***running-config*** contents to a log file once each hour, beginning immediately, and creating up to 24 log files.

  ```
  switch(config)#schedule backup-test interval 60 max-log-files 24 command show
  running-config
  ```

schedule config max-concurrent-jobs

The **schedule config max-concurrent-jobs** command specifies the number of scheduled commands that can run concurrently on the switch. The **no schedule config max-concurrent-jobs** and **default schedule config max-concurrent-jobs** commands reset the limit to the default value of 1 by removing the corresponding **schedule config max-concurrent-jobs** statement from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
schedule config max-concurrent-jobs limit
no schedule config max-concurrent-jobs
default schedule config max-concurrent-jobs
```

Parameters

- *limit* maximum number of scheduled commands which can run concurrently on the switch. Values range from 1-4, default value is 1.

Example

- This command sets the number of scheduled commands which can run concurrently on the switch to a maximum of 2.

```
switch(config)#schedule config max-concurrent-jobs 2
switch(config)#
```

secret hash

The **secret hash** command enables the default hash function used for encrypting passwords.

The **no secret hash** and **default secret hash** commands reverts the default hash setting to its default value of MD5 by deleting the **secret hash** command from ***running-config***.

Command Mode

Mgmt-Defaults

Command Syntax

```
secret hash ENCRYPT_TYPE
no secret hash
default secret hash
```

Parameters

- ***ENCRYPT_TYPE*** encryption level of the *password* parameter. Settings include:
  - **md5** the password is entered as an MD5-encrypted string.
  - **sha512** the password is entered as an SHA-512-encrypted string.

Example

- These commands enable SHA-512 as the default hash function used for encrypting passwords.

```
switch(config)# management defaults
switch(config-mgmt-defaults)# secret hash sha512
```

send log message

The **send log message** command allows the user to manually send a syslog message with an optional severity level attribute. This feature is usually used for debugging purposes.

Command Mode

Privileged EXEC

Command Syntax

send log [*CONDITION*] message *message_text*

Parameters

- ***CONDITION*** The severity of level value. Options include:
  - **level *condition_name*** Severity level to be included in the message. Values include:
    - **alerts** Immediate action needed (severity level = 1)
    - **critical** Critical conditions (severity level = 2)
    - **debugging** Debugging messages (severity level = 7)
    - **emergencies** System is unusable (severity level = 0)
    - **errors** Error conditions (severity level = 3)
    - **informational** Informational messages (severity level = 6)
    - **notifications** Normal but significant conditions (severity level = 5)
    - **warnings** Warning conditions (severity level = 4)
    - <**0** ***to*** **7**> Severity level value
- *message_text* The description of the event log message.

Example

- This command generates an ***alerts***-level syslog message reading “test message from console.”

```
switch# send log level alerts message test message from console
switch# show logging alerts
Dec 23 16:52:56 switch Cli: %SYS-1-LOGMSG_ALERT: Message from admin on con0
(0.0.0.0): test message from console
switch#
```

server-alive count-max (SSH Management-Tunnel)

The **server-alive count-max** command sets the maximum number of server-alive messages that can be lost before the server is declared dead. Note: These packets are sent inside the tunnel and have the same properties of not being replayable or readable.

The **no server-alive count-max** and **default server-alive count-max** commands remove the **server-alive count-max** command from ***running-config***.

Command Mode

Mgmt-SSH-Tunnel

Command Syntax

```
server-alive count-max max_packet_lost
no server-alive count-max
default server-alive count-max
```

Parameters

- *max_packet_lost* the maximum number of keep-alive messages that are sent to the Secure Shell server. Value ranges from ***1*** to ***1000***; default value is ***3***.

Example

- This command sets the rate to 600 keep-alive packets that can be lost before the connection is declared dead.

```
switch(config)#management ssh
switch(config-mgmt-ssh-tunnel)test
switch(config-mgmt-ssh-tunnel-test)#server-alive count-max 10
switch(config-mgmt-ssh-tunnel-test)#
```

server-alive interval (SSH Management-Tunnel)

The **server-alive interval** command specifies an interval for sending keepalive messages to the Secure Shell server. The time value is given in seconds.

The **no server-alive interval** and **default server-alive interval** commands remove the **server-alive interval** command from ***running-config***.

Command Mode

Mgmt-SSH-Tunnel

Command Syntax

```
server-alive interval keep_alive_period
no server-alive interval
default server-alive interval
```

Parameters

- *keep_alive_period* keepalive period (seconds). Value ranges from 1 to 1000. Default value is ***10***.

Example

- These commands set the ***server-alive interval*** to 15 and the server-alive maximum count to 3. If the server becomes unresponsive, SSH will disconnect after approximately 45 seconds.

```
switch(config)#management ssh
switch(config-mgmt-ssh-tunnel)test
switch(config-mgmt-ssh-tunnel-test)#server-alive count-max 3
switch(config-mgmt-ssh-tunnel-test)#server-alive interval 15
switch(config-mgmt-ssh-tunnel-test)#
```

show (various configuration modes)

The **show** command, when executed within a configuration mode, can display data in ***running-config*** for the active configuration mode.

Command Mode

All configuration modes except Global Configuration

Command Syntax

`show [DATA_TYPE]`

Parameters

- ***DATA_TYPE*** Specifies display contents. Values include:
  - **active** Displays ***running-config*** settings for the configuration mode.
  - **active all** Displays ***running-config*** plus defaults for the configuration mode.
  - **active all detail** Displays ***running-config*** plus defaults for the configuration mode.
  - **comment** Displays comment entered for the configuration mode.

Related Commands

The **show** commands in ACL-configuration mode and MST-configuration mode include the **active** and **comment** options along with additional mode-specific options.

Example

- This command shows the server-group-TACACS+ configuration commands in ***running-config.***

```
switch(config-sg-tacacs+-TAC-GR)#show active
   server TAC-1
   server 10.1.4.14
switch(config-sg-tacacs+-TAC-GR)#
```

show event-handler

The **show event-handler** command displays the contents and activation history of a specified event handler or all event handlers.

Command Mode

Privileged EXEC

Command Syntax

show event-handler [*handler_name***]**

Parameters

- *handler_name* optional name of an event handler to display. If no parameter is entered, the command displays information for all event handlers configured on the system.

Example

- This command displays information about an event handler called “eth_5”.

```
switch#show event-handler eth_5
Event-handler eth_5
Trigger: onIntf Ethernet5 on operstatus delay 20 seconds
Action: /mnt/flash/myScript1
Last Trigger Activation Time: Never
Total Trigger Activations: 0
Last Action Time: Never
Total Actions: 0
switch#
```

show event-handler DropCountersHandler

The **show event-handler** command displays details of the DropCountersHandler event handler.

Command Mode

Privileged EXEC

Command Syntax

show event-handler DropCountersHandler

Example

- This command displays details of this event handler.

```
switch(config)#show event-handler DropCountersHandler
Event-handler DropCountersHandler (BUILT-IN)
Trigger: on-counters delay 0 seconds
  Polling Interval: 60 seconds
  Condition: bashCmd."DropCounterMonitor.py" > 0
Threshold Time Window: 0 Seconds, Event Count: 1 times
Action: DropCounterLog.py -l
Action expected to finish in less than 20 seconds
Total Polls: 39
Last Trigger Detection Time: 38 minutes 22 seconds ago
Total Trigger Detections: 1
Last Trigger Activation Time: 38 minutes 22 seconds ago
Total Trigger Activations: 1
Last Action Time: Never
Total Actions: 1

switch(config)#
```

show management ssh hostkey

The **show management ssh hostkey** command to display the public key authentication.

Command Mode

EXEC

Command Syntax

show management ssh hostkey *ALGORITHM*

Parameters

- ***ALGORITHM*** the public key in a public/private keypair is used.
  - **dsa public** the default authentication list. Note: It is a violation of Common Criteria policy to use the DSA algorithm.
  - **rsa public** the authentication list for SSH logins.

Example

- After a switch is configured to a hardware source of entropy and FIPS algorithms, the RSA key pair used on the switch for SSH access must be regenerated. Regeneration of the key will securely zeroize the old key pair and generate a new one. This command regenerates the key pair.

  ```
  switch#reset ssh hostkey rsa
  ```

  This command displays the new public key.

  ```
  switch#show management ssh hostkey rsa public
  ssh-rsa AAAAB3NzaC1yc2EAAAABIwAAAQEAwGXvEQ40NMSGpNqQ44mzFc6STteWp3FmAK+l9TJYsK9
  crekmuwkar8ImLdgc9U/XQvkCZ4RiVQM3yJ+RP21S8azN90OTa2EReCgi2og0oUMGbjDlUFzwBnr5ne
  eORdOE1CDZ4d/9DwI5dCVcrQtnwL6TLO/aEPNBg/iL+maBKf00HgIoFow6xeFE1EigQBixrsyW3jau1
  87oI+bSAhzzHjKRT+3Wd4jT9OVc57EfH3dNmB2RPAvVqGv/x9nea1v+N2dlGk7jzpUjkx76p6LtZbNR
  Z/xZqFLbawLfRK4fmnqS9eNJ/4iBkS9FSrglcbj2yP96YKAv5Zky4rp8yNrVgw==chassisAddr=00:
  1c:73:1b:c1:e0
  switch#
  ```

show module

The **show module** command displays information that identifies the supervisor, fabric, and linecard modules in a modular switch, including model number, serial number, hardware version number, software version (supervisors only), MAC address (supervisors and linecards), and operational status.

Command Mode

EXEC

Command Syntax

show module [*MODULE_NAME*]

Parameters

- ***MODULE_NAME*** Specifies modules for which data is displayed. Options include:
  - <no parameter> All modules (identical to **all** option).
  - **fabric** *fab_num* Specified fabric module. Number range varies with switch model.
  - **linecard** *line_num* Linecard module. Number range varies with switch model.
  - **supervisor** *super_num* Supervisor module. Number range varies with switch model.
  - *mod_num* Supervisor (1 to 2) or linecard (3 to 18) module.
  - **all** All modules.

Related Commands

- **show version** displays model and serial numbers of modular system components.

Example

This command displays information about all installed modules on a DCS-7504 switch.

```
switch#show module
Module    Ports Card Type                              Model           Serial No.
--------- ----- ------------------------------------ --------------- -----------
1         2     DCS-7500 Series Supervisor Module    7500-SUP        JSH11440327
2         1      Standby supervisor                   Unknown         Unknown
3         48    48-port SFP+ 10GigE Linecard         7548S-LC        JSH10315938
4         48    48-port SFP+ 10GigE Linecard         7548S-LC        JSH11665247
5         48    48-port SFP+ 10GigE Linecard         7548S-LC        JSH11834614
6         48    48-port SFP+ 10GigE Linecard         7548S-LC        JSH11060688
Fabric1   0     DCS-7504 Fabric Module               7504-FM         JSH11244430
Fabric2   0     DCS-7504 Fabric Module               7504-FM         JSH11892120
Fabric3   0     DCS-7504 Fabric Module               7504-FM         JSH11941115
Fabric4   0     DCS-7504 Fabric Module               7504-FM         JSH11661618
Fabric5   0     DCS-7504 Fabric Module               7504-FM         JSH11757555
Fabric6   0     DCS-7504 Fabric Module               7504-FM         JSH11847728

Module    MAC addresses                             Hw      Sw      Status
--------- ----------------------------------------- ------- ------- -------
1         00:1c:23:03:06:ac - 00:1c:23:03:06:ac  07.06   4.12.1  Active
2                                                        4.12.1  Standby
3         00:1c:23:03:80:44 - 00:1c:23:03:80:73  06.00           Ok
4         00:1c:23:03:e4:34 - 00:1c:23:03:e4:63  07.10           Ok
5         00:1c:23:12:0b:3f - 00:1c:23:12:0b:6e  07.30           Ok
6         00:1c:23:12:b6:3f - 00:1c:23:12:b6:6e  08.00           Ok
Fabric1                                             05.03           Ok
Fabric2                                             05.03           Ok
Fabric3                                             05.02           Ok
Fabric4                                             05.02           Ok
Fabric5                                             05.02           Ok
Fabric6                                             05.02           Ok
switch#
```

- This command displays information about all installed modules on a DCS-7304 switch.

```
switch#show module
Module    Ports Card Type                              Model           Serial No.
--------- ----- ------------------------------------ --------------- -----------
1         3     Supervisor 7300X SSD                 DCS-7300-SUP-D  JAS13340024
3         128   32 port 40GbE QSFP+ LC               7300X-32Q-LC    JPE13440416
4         64    48 port 10GbE SFP+ & 4 port QSFP+ LC 7300X-64S-LC    JAS13310113
5         64    48 port 10GbE SFP+ & 4 port QSFP+ LC 7300X-64S-LC    JAS13340033
6         64    48 port 10GbE SFP+ & 4 port QSFP+ LC 7300X-64S-LC    JAS13310103
Fabric1   0     7304X Fabric Module                  7304X-FM        JAS13320077
Fabric2   0     7304X Fabric Module                  7304X-FM        JAS13350043
Fabric3   0     7304X Fabric Module                  7304X-FM        JAS13350050
Fabric4   0     7304X Fabric Module                  7304X-FM        JAS13350056

Module    MAC addresses                          Hw      Sw      Status
--------- -------------------------------------- ------- ------- -------
1         00:1c:73:36:4b:71 - 00:1c:73:36:4b:72  01.01   4.13.3F Active
3         00:1c:73:58:d4:68 - 00:1c:73:58:d4:87  03.04           Ok
4         00:1c:73:36:05:61 - 00:1c:73:36:05:94  02.02           Ok
5         00:1c:73:36:0a:e1 - 00:1c:73:36:0b:14  02.03           Ok
6         00:1c:73:36:02:e1 - 00:1c:73:36:03:14  02.02           Ok
Fabric1                                          00.00           Ok
Fabric2                                          00.00           Ok
Fabric3                                          00.00           Ok
Fabric4                                          00.00           Ok
switch#
```

show platform sand compatibility

The **show sand platform compatibility** command displays the fabric and forwarding modes. These modes determine switch forwarding capabilities and programs performance capacity of installed linecards

sinformation that identifies the supervisor, fabric, and linecard modules in the modular switch, including model number, serial number, hardware version number, software version (supervisors only), MAC address (supervisors and linecards), and operational status.

Command Mode

Privileged EXEC

Command Syntax

```
show platform sand compatibility
```

Related Commands

- **platform sand fabric mode (7500 and 7500E Series)** specifes the fabric software mode.
- **platform sand forwarding mode (7500 and 7500E Series)** specifes the forwarding software mode.

Example

- This command indicates that the switch is in Fe600 fabric mode and PetraA forwarding mode.

```
switch#show platform sand compatibility
                    Configuration        Status
Forwarding mode     None                 PetraA
Fabric mode         None                 Fe600
switch#
```

show platform sand lag hardware-only

The **show platform sand lag hardware-only** command displays whether or not LAGs are hardware-only.

Command Mode

Privileged EXEC

Command Syntax

show platform sand lag hardware-only

Examples

- This command indicates that LAGs are hardware-only.

```
switch(config)#platform sand lag hardware-only
switch(config)#exit
switch#show platform sand lag hardware-only
Hardware resources are used for all LAGs: True
switch#
```

- This command indicates that LAGs are not hardware-only.

```
switch(config)#no platform sand lag hardware-only
switch(config)#exit
switch#show platform sand lag hardware-only
Hardware resources are used for all LAGs: False
switch#
```

show schedule

The **show schedule** command displays logging output on the terminal during the current terminal session. This command affects only the local monitor. The **no terminal monitor** command disables direct monitor display of logging output for the current terminal session.

Command Mode

Privileged EXEC

Command Syntax

show schedule *schedule_name*

Parameters

- *schedule_name* label associated with the scheduled command.

Example

- This command displays logging to the local monitor during the current terminal session.

```
switch#show schedule tech-support
The last CLI command execution was successful
CLI command "show tech-support" is scheduled next at "10:41:37 04/04/2017",
interval is 60 minutes
Maximum of 100 log files will be stored
Verbose logging is off
1 log files currently stored in flash:/schedule/tech-support

Start time                Size         Filename
------------------------ ------------- ----------------------------------
Apr 04 2017 09:40       115.3 KB       tech-support_2017-04-04.0940.log.gz
switch#
```

show schedule summary

The **show schedule summary** command displays the list of active scheduled commands.

Command Mode

Privileged EXEC

Command Syntax

show schedule summary

Example

- This command displays the list of active scheduled commands.

```
switch#show schedule summary
Maximum concurrent jobs  2
Name         At time Last  Inter\  Max   Logfile Location               Status
                     time   val    log
                           (mins)  files
------------ ------- ----- ------- ----- ------------------------------ ------
tech-support   now   09:41   60    100   flash:/schedule/tech-support/  Success
ms_1           now   16:28  720    10    flash:/schedule/ms_1
switch#
```

show version

The **show version** command displays information that identifies the switch, including its model number, serial number, and system MAC address. The command also provides hardware and software manufacturing information, along with the available memory and elapsed time from the most recent reload procedure.

Command Mode

EXEC

Command Syntax

show version [*INFO_LEVEL*]

Parameters

- ***INFO_LEVEL*** Specifies information the command displays. Options include
  - <no parameter> Model and serial numbers, manufacturing data, uptime, and memory.
  - **detail** Data listed *<no parameter>* option plus version numbers of internal components.

Related Commands

- **show module** displays model and serial numbers of modular system components.

Examples

- This command displays the switch's model number, serial number, hardware and software manufacturing information, uptime, and memory capacity,

```
switch>show version
Arista DCS-7150S-64-CL-F
Hardware version:    01.01
Serial number:       JPE13120819
System MAC address:  001c.7326.fd0c

Software image version: 4.13.2F
Architecture:           i386
Internal build version: 4.13.2F-1649184.4132F.2
Internal build ID:      eeb3c212-b4bd-4c19-ba34-1b0aa36e43f1

Uptime:                 1 hour and 36 minutes
Total memory:           4017088 kB
Free memory:            1473280 kB

switch>
```

shutdown (SSH Management-Tunnel)

The **shutdown** command, in Mgmt-SSH-Tunnel mode, disables or enables management SSH on the switch. Management SSH is disabled by default.

The **no shutdown** command, in Mgmt-SSH-Tunnel mode, re-enables the management SSH access**.**

The default shutdown command, in Mgmt-SSH-Tunnel mode, disables the management SSH access and removes the corresponding **no shutdown** command from the from ***running-config***.

Command Mode

Mgmt-SSH-Tunnel

Command Syntax

```
shutdown
no shutdown
default shutdown
```

Example

- These commands disables the SSH management access.

```
switch(config)#management ssh
switch(config-mgmt-ssh)# shutdown
switch(config-mgmt-ssh)#
```

- These commands enables the SSH management access.

```
switch(config)#management ssh
switch(config-mgmt-ssh)# no shutdown
switch(config-mgmt-ssh)#
```

ssh

The **ssh** command establishes an SSH connection to an IPv4 server and optionally specifies additional parameters for the connection.

Command Mode

EXEC

Command Syntax

ssh [*VRF_INST*] [*CIPHER*] [*LOG_NAME*] [*MAC_CRYPT*] [*KEX*] [*KEY*] [*PORT*] [*VERSION*] *SERVER*

The ***VRF_INST*** parameter, when present, is always first. The ***SERVER*** parameter is always last. All other parameters can be placed in any order.

Parameters

- ***VRF_INST*** specifies the VRF instance.
  - <no parameter> changes are made to the default VRF.
  - **vrf** *vrf_name* changes are made to the specified user-defined VRF.
- ***CIPHER*** Cipher specification for encryption. Options include:
  - **-c** *cipher_1 cipher_2* **...** *cipher_n*
    - **3des** Specifies the encryption algorithm 3des(v1)
    - **3des-cbc** Specifies triple DES (112 bit)
    - **aes128-cbc** Specifies advanced Encryption Standard (128 bit, CBC mode)
    - **aes128-ctr** Specifies advanced Encryption Standard (128 bit, counter mode)
    - **aes192-cbc** Specifies advanced Encryption Standard (192 bit, CBC mode)
    - **aes192-ctr** Specifies advanced Encryption Standard (192 bit, counter mode)
    - **aes256-cbc** Specifies advanced Encryption Standard (256 bit, CBC mode)
    - **aes256-ctr** Specifies advanced Encryption Standard (256 bit, counter mode)
    - **arcfour** Specifies arcfour stream cipher (RC4 like)
    - **arcfour128** Specifies arcfour stream cipher (RFC 4345, 128 bit)
    - **arcfour256** Specifies arcfour stream cipher (RFC 4345, 256 bit)
    - **blowfish** Specifies blowfish block cipher (64 bit)
    - **blowfish-cbc** Specifies blowfish block cipher (128 bit, CBC mode)
    - **cast128-cbc** Specifies CAST-128 (RFC 2144, 128 bit, CBC mode)
    - **des** Specifies the encryption algorithm des(v1)
- ***LOG_NAME*** Login name. Options include:
  - **-l** *login* **Name of a user during login**
- ***MAC_CRYPT*** MAC specification for encryption. Options include:
  - **-m** *mac_crypt_1 mac_crypt_2* **...** *crypt_n*
    - **hmac-md5** Hash Message Authentication Code MD5
    - **hmac-md5-96** Hash Message Authentication Code MD5 for use in ESP and AH
    - **hmac-ripemd160** Hash Message Authentication Code RIPEMD-160
    - **hmac-sha1** Hash Message Authentication Code SHA-1
    - **hmac-sha1-96** Hash Message Authentication Code SHA-1 for use in ESP and AH

- ***KEX*** Limits the message authentication codes from all of the available options to the set specified.
  - **-o KexAlgorithms** *kex_1 kex_2* **...** *kex_n*
    - **diffie-hellman-group-exchange-sha1** Negotiated Group Exchange with SHA-1
    - **diffie-hellman-group-exchange-sha256** Negotiated Group Exchange with SHA-256
    - **diffie-hellman-group1-sha1** Oakley Group 1 with SHA-1
    - **diffie-hellman-group14-sha1** Oakley Group 14 with SHA-1
- ***KEY*** Limits the key-exchange methods from all of the available options to the set specified. Options include:
  - **-o StrictHostKeyChecking yes Specifies that SSH will automatically check remote servers hostkey before connecting.**
  - **-o StrictHostKeyChecking no Specifies that SSH will ignore remote servers hostkey.**
- ***PORT*** Options include:
  - **-p <*1* to *65535*> Specifies the port number of the server.**
- ***VERSION*** Protocol version to force. Options include:
  - **-v <*1* to *2*> When the SSH supports SSH1, the protocol version is 1.99. Otherwise, the protocol version is 2.**
- ***SERVER*** Options include:
  - *ip_address* Specifies the IPv4 address of the server.
  - *hostname* Specifies the host name of the server

Example

- This command instructs the server to use the Oakley Group 14 Diffie-Hellman method with an SHA-1 hash for key exchange.

```
switch(config)#management ssh
switch(config-mgmt-ssh)#key-exchange diffie-hellman-group14-sha1
```

For Common Criteria

- The following set of commands will put the SSH server and SSH Tunnels on the switch into a Common Criteria approved mode:

```
switch(config)#management ssh
switch(config-mgmt-ssh)#cipher aes128-cbc aes256-cbc
switch(config-mgmt-ssh)#key-exchange diffie-hellman-group14-sha1
switch(config-mgmt-ssh)#mac hmac-sha1
switch(config-mgmt-ssh)#hostkey server rsa
switch(config-mgmt-ssh)#hostkey client strict-checking
switch(config-mgmt-ssh)#log-level verbose
```

tcpdump

The **tcpdump** command is a packet sniffer used for analyzing network traffic, capturing traffic on a specified **tcpdump** interface, and assisting in troubleshooting network problems. Traffic coming to or from the control plane of the switch is visible when running **tcpdump**, and allows users to instantly analyze important traffic such as spanning tree and routing protocols, as well as any other traffic to the switch (via an SVI or management IP address). If no interface is provided, **tcpdump** defaults to capturing on the fabric interface. The fabric interface is always in the main VRF.

Command Mode

Privileged EXEC

Command Syntax

```
tcpdump [OPTIONS]
```

Parameters

- ***OPTIONS*** Specify **tcpdump** options which include:
  - **interface** tcpdump interface. Options include:
    - **ethernet** *number* ethernet interface number.
    - **loopback** *number* loopback interface number <1-1000>.
    - **management** *number* management interface. Options include:
      - **management interface** *number* management interface number on the active supervisor.
      - **slot** *number* slot number on the management interface.
    - **port-channel** *number* port-channel interface. Options include:
      - **port-channel Interface** *number* port-channel interface number <1-2000>.
      - **port-channel sub interface** *number* port-channel sub interface number <1-2000>,<1-4094>.
    - **tunnel** *number* tunnel interface <0-63>.
    - **vlan** *number* vlan interface number <1-4094>.
    - **vxlan** *number* vxlan interface number.
    - **any** monitor any interface. Options include:
      - **vrf** specify a VRF.
  - **file** set the output file.
  - **filecount** specify the number of output files.
  - **filter** set the filtering expression.
  - **lookup-names** enable reverse DNS lookup.
  - **max-file-size** *number* specify the maximum size of output file in millions of bytes <0-100>.
  - **packet-count** limit the number of packets to capture.
  - **size** *number* set the maximum number of bytes to dump per packet <1-65536>.
  - **verbose** enable verbose mode.

Examples

- This command will ensure that capture files are continuously rotated once they reach the *file count* and *max-file-size*.

```
switch#tcpdump interface Ethernet1 file flash:/icmp.pcap filecount 2
max-file-size 1 filter icmp
switch#
```

Note It is recommended that captures are explicitly limited to avoid accidentally filling the filesystem with the options *file count* and *max-file-size*.

- This command runs the **tcpdump** command for *any* interface and captures a specific number of packets to a file.

```
switch#tcpdump interface any packet-count 20 file flash:foo.pcap
tcpdump: listening on any, link-type LINUX_SLL (Linux cooked), capture size
262144 bytes
20 packets captured
21 packets received by filter
0 packets dropped by kernel
switch#
```

Note Specifying the packet-count will result in the command completing and returning the user to the command prompt once the count has completed. Therefore, it is not necessary to press CTRL-C to return to the switch prompt.

- This command runs the **tcpdump** command for *any* interface, filters the input for *ip proto ospf*, and writes the results to a file.

```
switch#tcpdump interface any file flash:foo.pcap filter ip proto ospf
tcpdump: listening on any, link-type LINUX_SLL (Linux cooked), capture size
262144 bytes
switch#
```

- This command runs the **tcpdump** command for *any* interface for LLDP frames.

```
switch#tcpdump interface any filter ether proto 0x88cc
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on any, link-type LINUX_SLL (Linux cooked), capture size 262144 bytes
switch#
```

- This command runs the **tcpdump** command for the file flash:tcpdumptest.pcap on interface PO26.

```
switch#tcpdump file flash:tcpdumptest.pcap interface PO26
tcpdump: WARNING: po26: no IPv4 address assigned
tcpdump: listening on po26, link-type EN10MB (Ethernet), capture size 65535 bytes
switch#
```

Note To stop **tcpdump** from listening on the interface, press CTRL-C to return to the switch prompt.

Note It is important that you do not fill up the switch filesystem with capture files. Always remove capture files as soon as possible, and monitor the space available while carrying out captures.

tcpdump session

The **tcpdump session** command configures a session for analyzing network traffic, capturing traffic on a VRF interface, and assisting in troubleshooting network problems. Traffic coming to or from the control plane of the switch is visible when running **tcpdump**, and allows users to instantly analyze important traffic such as spanning tree and routing protocols, as well as any other traffic to the switch (via an SVI or management IP address).

The **no tcpdump session** command ends the **tcpdump session**, and the **default tcpdump session** command restores the default **tcpdump session**.

Command Mode

Global Configuration

Command Syntax

```
tcpdump session [SESSION_NAME]
no tcpdump session [SESSION_NAME]
default tcpdump session [SESSION_NAME]
```

Parameters

- ***SESSION_NAME*** Define the name of a session using a word with the following options:
  - **duration** *number* limit duration <1-10000> seconds.
  - **file** set the output file.
  - **filecount** *number* specify the number of output files <0-100>.
  - **filter** set the filtering expression to select which packets will be dumped.
  - **interface** select an interface to monitor. (The default is fabric.) Options include:
    - **ethernet** *number* ethernet interface number.
    - **loopback** *number* loopback interface number <1-1000>.
    - **management** *number* management interface. Options include:
      - **management interface** *number* management interface number on the active supervisor.
      - **slot** *number* slot number on the management interface.
    - **port-channel** *number* port-channel interface. Options include:
      - **port-channel interface** *number* port-channel interface number <1-2000>.
      - **port-channel sub interface** *number* port-channel sub interface number <1-2000>,<1-4094>.
    - **tunnel** *number* tunnel interface <0-63>.
    - **vlan** *number* vlan interface number <1-4094>. Options include:
    - **vxlan** *number* vxlan interface number. Options include:
    - **any** monitor any interface. Options include:
      - **vrf** specify a VRF.
  - **max-file-size** *number* specify the maximum size of output file in millions of bytes <0-100>.
  - **monitor** specify a monitor session name with a word.
  - **packet-count** *number* specify a limit on the number of packets to capture <1-10000>.
  - **queue-monitor** monitor queue length. Options include:
  - **shutdown** shutdown the tcpdump session.

- **size** *number* set the maximum number of bytes to dump per packet <1-65536>.

Examples

- This command will ensure that capture files are continuously rotated once they reach the *file count* and *max-file-size*.

```
switch(config)#tcpdump session icmp interface Ethernet1 file flash:/icmp.pcap
filecount 2 max-file-size 1 filter icmp
switch(config)#
```

Note It is recommended that captures are explicitly limited to avoid accidentally filling the filesystem with the options *file count* and *max-file-size*.

- This command saves a tcpdump configuration session named ospf-capture that filters packets matching the pcap expression 'ip proto ospf' on interface vlan 1, appending the output to foo.pcap on the flash disk. The session is immediately started.

```
switch(config)#tcpdump session ospf-capture interface vlan 1 file flash:foo.pcap
filter ip proto ospf
switch(config)#
```

- These commands stop and restart the ospf-capture session. When restarted, the session overwrites the previous buffer.

```
switch(config)#tcpdump session ospf-capture shutdown
switch(config)#no tcpdump session ospf-capture shutdown
```

- This command displays existing configurations.

```
switch(config)#show tcpdump
Session      Enabled Target VRF     File            Filter        c/C/s/t On
------------ ------- ------ ------- --------------- ------------- ------- -----
ospf-capture True    Vlan1  default flash:/foo.pcap ip proto ospf 0/1/0/0 False

switch(config)#no tcpdump session ospf-capture
switch(config)#
```

- This command removes the ospf-capture session.

```
switch(config)#show tcpdump
Session      Enabled Target VRF     File            Filter        c/C/s/t On
------------ ------- ------ ------- --------------- ------------- ------- -----
ospf-capture True    Vlan1  default flash:/foo.pcap ip proto ospf 0/1/0/0 False

switch(config)#no tcpdump session ospf-capture
switch(config)#
```

Note It is important that you do not fill up the switch filesystem with capture files. Always remove capture files as soon as possible, and monitor the space available while carrying out captures.

terminal length

The **terminal length** command overrides automatic pagination and sets pagination length for all show commands on a terminal. If the output of a show command is longer than the configured terminal length, the output will be paused after each screenful of output, prompting the user to continue.

To disable pagination for an SSH session, set ***terminal length*** to 0. By default, all console sessions have pagination disabled.

The **no terminal length** and **default terminal length** commands restore automatic pagination by removing the **terminal length** command from ***running-config***.

The pagination setting is persistent if configured from Global Configuration mode. If configured from EXEC mode, the setting applies only to the current CLI session. Pagination settings may also be overridden when you adjust the size of the SSH terminal window, but can be reconfigured by running the **terminal length** command again.

Command Mode

EXEC

Command Syntax

```
terminal length lines
no terminal length
default terminal length
```

Parameters

- *lines* number of lines to be displayed at a time. Values range from 0 through 32767. A value of 0 disables pagination.

Example

- This command sets the pagination length for the current terminal session to 10 lines.

```
switch#terminal length 10
Pagination set to 10 lines.
```

- This command configures the switch to paginate terminal output automatically based on screen size for the current terminal session.

```
switch#no terminal length
```

- These commands disable pagination globally.

```
switch#configure
switch(config)#terminal length 0
Pagination disabled.
```

terminal monitor

The **terminal monitor** command enables the display of logging output on the terminal during the current terminal session. This command affects only the local monitor. The **no terminal monitor** command disables direct monitor display of logging output for the current terminal session.

Command Mode

Privileged EXEC

Command Syntax

```
terminal monitor
no terminal monitor
default terminal monitor
```

Example

- This command enables the display of logging to the local monitor during the current terminal session.

```
switch#terminal monitor
switch#
```

trigger

The **trigger** command specifies what event will trigger the event handler. Handlers can be triggered either by the system booting or by a change in a specified interface's IP address or operational status.

To specify the action to be taken when the handler is triggered, use the **action bash** command.

Command Mode

Event-Handler Configuration

Command Syntax

`trigger` ***EVENT***

Parameters

- ***EVENT*** event which will trigger the configuration mode event handler. Values include:
  - **onboot** triggers when the system reboots, or when you exit event-handler configuration mode. This option takes no further arguments, and passes no environment variables to the action triggered.
  - **onintf *INTERFACE CHANGE*** triggers when a change is made to the specified interface.
  - **on-startup-config** triggers when a change is made to the ***startup-config*** file.
  - **vm-tracer vm** triggers when a virtual machine monitored by VM Tracer changes state.
- ***INTERFACE*** the triggering interface. Values include:
  - **ethernet** *number* Ethernet interface specified by *number*.
  - **loopback** *number* loopback interface specified by *number*.
  - **management** *number* management interface specified by *number*.
  - **port-channel** *number* channel group interface specified by *number*.
  - **vlan** *numver* VLAN interface specified by *number*.
- ***CHANGE*** the change being watched for in the triggering interface. Values include:
  - **ip** triggers when the IPv4 address of the specified interface is changed.
  - **ip6** triggers when the IPv6 address of the specified interface is changed.
  - **operstatus** triggers when the operational status of the specified interface changes.

Examples

- This command configures the event handler "Eth5" to be triggered when there is a change in the operational status or IP address of Ethernet interface 5.

```
switch(config-handler-Eth5)#trigger onIntf Ethernet 5 operstatus ip
switch(config-handler-Eth5)#
```

- This command configures the event handler "onStartup" to be triggered when the system boots, or on exiting event-handler configuration mode.

```
switch(config-handler-onStartup)#trigger onboot
switch(config-handler-onStartup)#
```

tunnel (SSH Management)

The **tunnel** command places the switch in SSH tunnel configuration mode. EOS provides a SSH Tunneling feature to transparently secure TCP connections to remote servers. This feature can accept any TCP connection that goes to the CPU on the switch and establish a standard SSH tunnel to a remote host. This has a twofold advantage:

- TCP based services do not need to perform any encryption within their own protocol.
- The connection is secure against replay attacks, manipulation and eavesdropping via SSHv2 Common Criteria compliant encryption.

The **no tunnel** and **default tunnel** commands disable Management-ssh-tunnel mode on the switch by removing all Management-ssh-tunnel configuration mode commands from ***running-config***.

Management-ssh-tunnel configuration mode is not a group change mode; ***running-config*** is changed immediately upon entering commands. Exiting SSH configuration mode does not affect ***running-config***. The **exit** command returns the switch to global configuration mode.

Command Mode

Mgmt-ssh Configuration

Command Syntax

```
tunnel tunnel_name
no tunnel
default tunnel
```

Parameters

- *tunnel_name* SSH Tunnel or SSH VRF name.

Commands Available in Management-ssh-tunnel Configuration Mode

- **local (SSH Management-Tunnel)**
- **remote (SSH Management-Tunnel)**
- **server-alive count-max (SSH Management-Tunnel)**
- **server-alive interval (SSH Management-Tunnel)**
- **shutdown (SSH Management-Tunnel)**
- **ssh-server user (SSH Management-Tunnel)**

Example

- These commands place the switch in management-ssh-tunnel mode and create a management SSH tunnel called “foo”.

```
switch(config)#management ssh
switch(config-mgmt-ssh)#tunnel foo
switch(config-mgmt-ssh)#
```

Chapter 4

AAA Configuration

This chapter describes authentication, authorization, and accounting configuration tasks and contains these sections:

- Section 4.1: Authorization, Authentication, and Accounting Overview
- Section 4.2: Configuring the Security Services
- Section 4.3: Server Groups
- Section 4.4: Role Based Authorization
- Section 4.5: Activating Security Services
- Section 4.6: TACACS+ Configuration Examples
- Section 4.7: AAA Commands

4.1 Authorization, Authentication, and Accounting Overview

4.1.1 Methods

The switch controls access to EOS commands by authenticating user identity and verifying user authorization. Authentication, authorization, and accounting activities are conducted through three data services – a local security database, TACACS+ servers, and RADIUS servers. Section 4.2: Configuring the Security Services describes these services.

4.1.2 Configuration Statements

Switch security requires two steps:

Step 1 Configuring security service parameters.

The switch provides configuration commands for each security service:

- A local file supports authentication through **username** and **enable secret** commands.
- TACACS+ servers provide security services through **tacacs-server** commands.
- RADIUS servers provide security services through **radius-server** commands.

Section 4.2: Configuring the Security Services describes security service configuration commands.

Step 2 Activating authentication, authorization, and accounting services.

EOS provides **aaa authorization**, **aaa authentication**, and **aaa accounting** commands to select the primary and backup services. Section 4.5: Activating Security Services provides information on implementing a security environment.

4.1.3 Encryption

The switch uses clear-text passwords and server access keys to authenticate users and communicate with security systems. To prevent accidental disclosure of passwords and keys, ***running-config*** stores their corresponding encrypted strings. The encryption method depends on the type of password or key.

Commands that configure passwords or keys can accept the clear-text password or an encrypted string that was generated by the specified encryption algorithm with the clear-text password as the seed.

4.2 Configuring the Security Services

The switch can access three security data services to authenticate users and authorize switch tasks: a local file, TACACS+ servers, and RADIUS Servers.

4.2.1 Local Security File

The local file uses passwords to provide these authentication services:

- authenticate users as they log into the switch
- control access to configuration commands
- control access to the switch root login

The local file contains username-password combinations to authenticate users. Passwords also authorize access to configuration commands and the switch root login.

4.2.1.1 Passwords

The switch recognizes passwords as clear text and encrypted strings.

- **clear-text** passwords are the text that a user enters to access the CLI, configuration commands, or the switch root login.
- **Encrypted strings** are MD5-encrypted strings generated with the **clear text** as the seed. The local file stores passwords in this format to avoid unauthorized disclosure. When a user enters the clear-text password, the switch generates the corresponding secure hash and compares it to the stored version. **The switch cannot recover the clear text from which an encrypted string is generated**.

Valid passwords contain the characters A-Z, a-z, 0-9 and any of these punctuation characters:

```
!   @   #   $   %   ^   &   *   (   )   -   _   =   +
{   }   [   ]   ;   :   <   >   ,   .   ?   /   ~   \
```

4.2.1.2 Usernames

Usernames control access to the EOS and all switch commands. The switch is typically accessed through an SSH login, using a previously defined username-password combination. To create a new username or modify an existing username, use the **username** command.

Valid usernames begin with A-Z, a-z, or 0-9 and may also contain any of these characters:

```
@   #   $   %   ^   &   *   -   _   =
+   ;   <   >   ,   .   ~   |
```

The default username is **admin,** which is described in Admin Username.

Examples

- These equivalent commands create the username *john* and assign it the password *x245*. The password is entered in clear text because the encrypt-type parameter is omitted or zero.

```
switch(config)#username john secret x245
switch(config)#username john secret 0 x245
```

- This command creates the username *john* and assigns it to the text password that corresponds to the encrypted string *1sU.7hptc$TsJ1qslCL7ZYVbyXNG1wg1.* The string was generated by an MD5-encryption program using x245 as the seed.

```
switch(config)#username john secret 5
$1$sU.7hptc$TsJ1qslCL7ZYVbyXNG1wg1
```

The username is authenticated by entering ***x245*** when the CLI prompts for a password.

- This command creates the username *jane* without securing it with a password. It also removes a password if the *jane* username exists.

```
switch(config)#username jane nopassword
```

- This command removes the username **william** from the local file.

```
switch(config)#no username william
```

4.2.1.3 Logins by Unprotected Usernames

The default switch configuration allows usernames that are not password protected to log in only from the console. The **aaa authentication policy local** command configures the switch to allow unprotected usernames to log in from any port. To reverse this setting to the default state, use **no aaa authentication policy local allow-nopassword-remote-login.**

Warning Allowing remote access to accounts without passwords is a severe security risk. Arista Networks recommends assigning strong passwords to all usernames.

Examples

- This command configures the switch to allow unprotected usernames to log in from any port.

```
switch(config)#aaa authentication policy local
allow-nopassword-remote-login
```

- This command configures the switch to allow unprotected usernames to log in only from the console port.

```
switch(config)#no aaa authentication policy local
allow-nopassword-remote-login
```

4.2.1.4 Enable Command Authorization

The **enable** command controls access to Privileged EXEC and all configuration command modes. The enable password authorizes users to execute the **enable** command. When the enable password is set, the CLI displays a password prompt when a user attempts to enter Privileged EXEC mode.

```
main-host>enable
Password:
main-host#
```

If an incorrect password is entered three times in a row, the CLI displays the EXEC mode prompt.

If no enable password is set, the CLI does not prompt for a password when a user attempts to enter Privileged EXEC mode.

To set the enable password, use the **enable secret** command.

Examples

- These equivalent commands assign *xyrt1* as the enable password.

```
switch(config)#enable secret xyrt1
switch(config)#enable secret 0 xyrt1
```

- This command assigns the enable password to the clear text (*12345*) corresponding to the encrypted string *$1$8bPBrJnd$Z8wbKLHpJEd7d4tc5Z/6h/*. The string was generated by an MD5-encryption program using *12345* as the seed.

```
switch(config)#enable secret 5 $1$8bPBrJnd$Z8wbKLHpJEd7d4tc5Z/6h/
```

- This command deletes the enable password.

```
switch(config)#no enable secret
```

4.2.1.5 Root Account Password

The root account accesses the root directory in the underlying Linux shell. When it is not password protected, you can log into the root account only through the console port. After you assign a password to the root account, you can log into it through any port.

To set the password for the root account, use the **aaa root** command.

Examples

- These equivalent commands assign *f4980* as the root account password.

```
switch(config)#aaa root secret f4980
switch(config)#aaa root secret 0 f4980
```

- This command assigns the text (*ab234*) that corresponds to the encrypted string of *1HW05LEY8$QEVw6JqjD9VqDfh.O8r.b.* as the root password.

```
switch(config)#aaa root secret 5 $1$HW05LEY8$QEVw6JqjD9VqDfh.O8r.b
```

- This command removes the password from the root account.

```
switch(config)#aaa root nopassword
```

- This command disables the root login.

```
switch(config)#no aaa root
```

4.2.2 TACACS+

Terminal Access Controller Access-Control System Plus (TACACS+), derived from the TACACS protocol defined in RFC 1492, is a network protocol that provides centralized user validation services. TACACS+ information is maintained on a remote database. EOS support of TACACS+ services requires access to a TACACS+ server.

TACACS+ manages multiple network access points from a single server. The switch defines a TACACS+ server connection by its address and port, allowing the switch to conduct multiple data streams to a single server by addressing different ports on the server.

These sections describe steps that configure access to TACACS+ servers. Configuring TACACS+ access is most efficiently performed when TACACS+ is functioning prior to configuring switch parameters.

4.2.2.1 Configuring TACACS+ Parameters

TACACS+ parameters define settings for the switch to communicate with TACACS+ servers. A set of values can be configured for individual TACACS+ servers that the switch accesses. Global parameters define settings for communicating with servers for which parameters are not individually configured.

The switch supports these TACACS+ parameters:

Encryption key

The encryption key is code that the switch and the TACACS+ server share to facilitate communications.

- The **tacacs-server host** command defines the encryption key for a specified server.
- The **tacacs-server key** command defines the global encryption key.

Examples

- This command configures the switch to communicate with the TACACS+ server assigned the host name *TAC_1* using the encryption key *rp31E2v*.

  ```
  switch(config)#tacacs-server host TAC_1 key rp31E2v
  ```

- This command configures *cv90jr1* as the global encryption key.

  ```
  switch(config)#tacacs-server key 0 cv90jr1
  ```

- This command assigns *cv90jr1* as the global key, using the corresponding encrypted string.

  ```
  switch(config)#tacacs-server key 7 020512025B0C1D70
  ```

Session Multiplexing

The switch supports multiplexing sessions on a single TCP connection.

- The **tacacs-server host** command configures the multiplexing option for a specified server.
- There is no global multiplexing setting.

Example

- This command configures the switch to communicate with the TACACS+ server at *10.12.7.9* and indicates the server supports session multiplexing on a TCP connection.

  ```
  switch(config)#tacacs-server host 10.12.7.9 single-connection
  ```

Timeout

The timeout is the period the switch waits for a successful connection to, or response from, the TACACS+ server. The default is 5 seconds.

- The **tacacs-server host** command defines the timeout for a specified server.
- The **tacacs-server timeout** command defines the global timeout.

Examples

- This command configures the switch to communicate with the TACACS+ server assigned the host name *TAC_1* and configures the timeout period as *20 seconds*.

  ```
  switch(config)#tacacs-server host TAC_1 timeout 20
  ```

- This command configures *40 seconds* as the period that the server waits for a response from a TACACS+ server before issuing an error.

  ```
  switch(config)#tacacs-server timeout 40
  ```

Port

The port specifies the port number through which the switch and the servers send information. The TACACS+ default port is 49.

- The **tacacs-server host** command specifies the port number for an individual TACACS+ server.
- The global TACACS+ port number cannot be changed from the default value of 49.

Example

- This command configures the switch to communicate with the TACACS+ server at *10.12.7.9* through port 54.

  ```
  switch(config)#tacacs-server host 10.12.7.9 port 54
  ```

4.2.2.2 TACACS+ Status

To display the TACACS+ servers and their interactions with the switch, use the **show tacacs** command.

Example

- This command lists the configured TACACS+ servers.

```
switch(config)#show tacacs

server1: 10.1.1.45
Connection opens: 15
Connection closes: 6
Connection disconnects: 6
Connection failures: 0
Connection timeouts: 2
Messages sent: 45
Messages received: 14
Receive errors: 2
Receive timeouts: 2
Send timeouts: 3

Last time counters were cleared: 0:07:02 ago
```

To reset the TACACS+ status counters, use the **clear aaa counters tacacs+** command.

Example

- This command clears all TACACS+ status counters.

```
switch(config)#clear aaa counters tacacs
```

4.2.3 RADIUS

Remote Authentication Dial In User Service (RADIUS) is a networking protocol that provides centralized authentication, authorization, and accounting services for computers connecting to and using network resources. RADIUS is used to manage access to the Internet, internal networks, wireless networks, and integrated email services.

These sections describe steps that configure RADIUS server access. Configuring RADIUS parameters is most efficiently performed when RADIUS is functioning prior to configuring switch parameters.

4.2.3.1 RADIUS Vendor-Specific Attribute-Value Pairs

RADIUS servers and client companies extend basic RADIUS functionality through vendor specific attributes. A dictionary file includes a list of RADIUS attribute-value pairs that Arista switches use to perform AAA operations through the RADIUS server.

Arista switches use the following attribute values:

- Arista Vendor number: 30065
- Attribute: Arista-AVPair 1 string

Acceptable string values for Arista-AVPair include:

- "shell:priv-lvl=<privilege level of a user, 0-15>"
- "shell:roles=<list of roles for a user>"

Example

- This is a sample dictionary file that identifies Arista RADIUS vendor-specific attribute value pairs.

```
#
# dictionary.arista
#
VENDOR          Arista    30065
#   Standard Attribute
BEGIN-VENDOR    Arista
ATTRIBUTE       Arista-AVPair    1    string
END-VENDOR      Arista
```

4.2.3.2 Configuring RADIUS Defaults

RADIUS policies specify settings for the switch to communicate with RADIUS servers. A set of values can be configured for individual RADIUS servers that the switch accesses. Global parameters define settings for communicating with servers for which parameters are not individually configured.

The switch defines these RADIUS parameters:

Encryption key

The encryption key is the key shared by the switch and RADIUS servers to facilitate communications.

- The **radius-server host** command defines the encryption key for a specified server.
- The **radius-server key** command specifies the global encryption key.

Examples

- This command configures the switch to communicate with the RADIUS server assigned the host name *RAD-1* using the encryption key *rp31E2v*.

```
switch(config)#radius-server host RAD-1 key rp31E2v
```

- This command configures *cv90jr1* as the global encryption key.

```
switch(config)#radius-server key 0 cv90jr1
```

- This command assigns *cv90jr1* as the key by specifying the corresponding encrypted string.

```
switch(config)#radius-server key 7 020512025B0C1D70
```

Timeout

The timeout is the period that the switch waits for a successful connection to, or response from, a RADIUS server. The default period is 5 seconds.

- The **radius-server host** command defines the timeout for a specified server.
- The **radius-server timeout** command defines the global timeout.

Examples

- This command configures the switch to communicate with the RADIUS server assigned the host name *RAD-1* and configures the timeout period as *20 seconds*.

```
switch(config)#radius-server host RAD-1 timeout 20
```

- This command configures *50 seconds* as the period that the server waits for a response from a RADIUS server before issuing an error.

```
switch(config)#radius-server timeout 50
```

Retransmit

Retransmit is the number of times the switch attempts to access the RADIUS server after the first server timeout expiry. The default value is 3 times.

- The **radius-server host** command defines the retransmit for a specified server.
- The **radius-server retransmit** command defines the global retransmit value.

Examples

- This command configures the switch to communicate with the RADIUS server assigned the host name *RAD-1* and configures the retransmit value as 2.

  ```
  switch(config)#radius-server host RAD-1 retransmit 2
  ```

- This command configures the switch to attempt five RADIUS server contacts after the initial timeout. If the timeout parameter is set to *50 seconds,* then the total period that the switch waits for a response is ((5+1)*50) = 300 seconds.

  ```
  switch(config)#radius-server retransmit 5
  ```

Deadtime

Deadtime is the period when the switch ignores a non-responsive RADIUS server, or a server that does not answer retransmit attempts after timeout expiry. Deadtime is disabled if a value is not specified.

- The **radius-server host** command defines the deadtime for a specified server.
- The **radius-server deadtime** command defines the global deadtime setting.

Examples

- This command configures the switch to communicate with the RADIUS server assigned the host name *RAD-1* and configures the deadtime period as 90 minutes.

  ```
  switch(config)#radius-server host RAD-1 deadtime 90
  ```

- This command programs the switch to ignore a server for two hours if the server does not respond to a request during the timeout-retransmit period.

  ```
  switch(config)#radius-server deadtime 120
  ```

Port

The port specifies the port number through which the switch and servers send information.

- The **radius-server host** command specifies the port numbers for an individual RADIUS server.
- The global RADIUS port numbers cannot be changed from the default values of 1812 for an authorization port and 1813 for an accounting port.

Example

- These commands configure the switch to communicate with the RADIUS server named *RAD-1* through port number 1850 for authorization and port number 1851 for accounting.

  ```
  switch(config)#radius-server host RAD-1 auth-port 1850
  switch(config)#radius-server host RAD-1 acct-port 1851
  ```

  To remove the configuration for this server, use **no radius-server host** command and specify the hostname or IP address with both the authorization and accounting port numbers.

4.2.3.3 RADIUS Status

The **show radius** command displays configured RADIUS servers and their interactions with the switch.

Example

- This command lists the configured RADIUS servers.

```
switch(config)#show radius

server1: 10.1.1.45
Messages sent: 24
Messages received: 20
Requests accepted: 14
Requests rejected: 8
Requests timeout: 2
Requests retransmitted: 1
Bad responses: 1
Last time counters were cleared: 0:07:02 ago
```

To reset the RADIUS status counters, use the **clear aaa counters radius** command.

Example

- This command clears all RADIUS status counters.

```
switch(config)#clear aaa counters radius
```

4.3 Server Groups

A server group is a collection of servers that are associated with a single label. Subsequent authorization and authentication commands access all servers in a group by invoking the group name. The switch supports TACACS+ and RADIUS server groups.

The **aaa group server** commands create server groups and place the switch in server group mode to assign servers to the group. Commands that reference an existing group place the switch in group server mode to modify the group.

These commands create named server groups and enter the appropriate command mode for the specified group:

- **aaa group server radius**
- **aaa group server tacacs+**

The **server (server-group-RADIUS configuration mode)** and **server (server-group-RADIUS configuration mode)** commands add servers to the configuration mode server group. Servers must be previously configured with a **radius-server host** or **tacacs-server host** command before they are added to a group.

Examples

- This command creates the TACACS+ server group named TAC-GR and enters server group configuration mode for the new group.

```
switch(config)#aaa group server tacacs+ TAC-GR
switch(config-sg-tacacs+-TAC-GR)#
```

- These commands add two servers to the TAC-GR server group. To add servers to this group, the switch must be in sg-tacacs+-TAC-GR configuration mode.

  The CLI remains in server group configuration mode after adding the TAC-1 server (port 49) and the server located at 10.1.4.14 (port 151) to the group.

```
switch(config-sg-tacacs+-TAC-GR)#server TAC-1
switch(config-sg-tacacs+-TAC-GR)#server 10.1.4.14 port 151
switch(config-sg-tacacs+-TAC-GR)#
```

- This command exits server group mode.

```
switch(config-sg-tacacs+-TAC-GR)#exit
switch(config)#
```

- This command creates the RADIUS server group named RAD-SV1 and enters server group configuration mode for the new group.

```
switch(config)#aaa group server radius RAD-SV1
switch(config-sg-radius-RAD-SV1)#
```

- These commands add two servers to the RAD-SV1 server group. To add servers to this group, the switch must be in sg-radius-RAD-SV1 configuration mode.

  The CLI remains in server group configuration mode after adding the RAC-1 server (authorization port 1812, accounting port 1813) and the server located at 10.1.5.14 (authorization port 1812, accounting port 1850) to the group.

```
switch(config-sg-radius-RAD-SV1)#server RAC-1
switch(config-sg-radius-RAD-SV1)#server 10.1.5.14 acct-port 1850
switch(config-sg-radius-RAD-SV1)#
```

4.4 Role Based Authorization

Role based authorization is a method of restricting access to CLI command through the assignment of profiles, called ***roles***, to user accounts. Each role consists of rules that permit or deny access to a set of commands within specified command modes.

All roles are accessible to the local security file through a **username** parameter and to remote users through RADIUS servers. Each role can be applied to multiple user accounts. Only one role may be applied to a user.

4.4.1 Role Types

The switch defines two types of roles: user-defined and built-in:

- User-defined roles are created and edited through CLI commands.
- Built-in roles are supplied with the switch and are not user-editable.

Built-in roles supplied by the switch are ***network-operator*** and ***network-admin***.

4.4.2 Role Structure

A role is an ordered list of rules that restricts access to specified commands from users on whom it is applied. Roles consist of deny and permit rules. Each rule references a set of command modes and contains a regular expression that specifies one or more CLI commands. Commands are compared sequentially to the rules within a role until a rule's regular expression matches the command.

- Commands that match a regular expression in a permit rule are executed.
- Commands that match a regular expression in a deny rule are disregarded.
- Commands that do not match a regular expression are evaluated against the next rule in the role.

Upon its entry on the CLI, a command is compared to the first rule of the role. Commands that match the rule are executed (permit rule) or disregarded (deny rule). Commands that do not match the rule are compared to the next rule. This process continues until the command either matches a rule or the rule list is exhausted. The switch disregards commands not matching any rule.

4.4.3 Role Rules

Role rules consist of four components: sequence number, filter type, mode expression, and command expression:

Sequence number

The sequence number designates a rule's placement in the role. Sequence numbers range in value from 1 to 256. Rule commands that do not include a sequence number append the rule at the end of the list, deriving its sequence number by adding 10 to the sequence number of the last rule in the list.

Example

- These rules have sequence numbers 10 and 20.

```
10 deny mode exec command reload
20 deny mode config command (no |default )?router
```

Filter type

The filter type specifies the disposition of matching commands. Filter types are permit and deny. Commands matching permit rules are executed. Commands matching deny rules are disregarded.

Example

- These rules are deny and permit rules, respectively.

```
10 deny mode exec command reload
20 permit mode config command interface
```

Mode expression

The mode expression specifies the command mode under which the command expression is effective. The mode expression may be a regular expression or a designated keyword. Rules support the following mode expressions:

- **exec** EXEC and Privileged EXEC modes.
- **config** Global configuration mode.
- **config-all** All configuration modes, including global configuration mode.
- *short_name* Short key name of a command mode (exact match).
- *long_name* Long key name of a command mode (regular expression match of one or more modes).
- <no parameter> All command modes.

The **prompt** command parameters configures the CLI to display a configuration mode's key name:

- %P long key name.
- %p short key name.

Example

- These commands use the prompt command to display short key name (***if***) and long key name (***if-Et1***) for interface-ethernet 1.

```
switch(config)#prompt switch%p
switch(config)#interface ethernet 1
switch(config-if)#exit
switch(config)#prompt switch%P
switch(config)#interface ethernet 1
switch(config-if-Et1)#
```

The command supports the use of regular expressions to reference multiple command modes.

Example

These regular expressions correspond to the listed command modes:

- ***if-Vlan(1|2)*** matches interface-Vlan 1 or interface-Vlan 2.
- ***if*** matches all interface modes.
- ***acl-text1*** matches ACL configuration mode for text1 ACL.

Command Expression

The command expression is a regular expression that corresponds to one or more CLI commands.

Examples

These regular expressions correspond to the specified commands:

- ***reload*** reload command
- ***(no |default)?router*** commands that enter routing protocol configuration modes.
- ***(no |default)?(ip|mac) access-list*** commands that enter ACL configuration modes

- ***(no |default)?(ip|mac) access-group*** commands that bind ACLs to interfaces.
- ***lacp|spanning-tree*** LACP and STP commands
- ***.**** all commands

4.4.4 Creating and Modifying Roles

4.4.4.1 Built-in Role

The switch provides the following two built-in roles:

- **network-operator** Allows all commands in EXEC (Privileged) modes. Commands in all other modes are denied.
- **network-admin** Allows all CLI commands in all modes.

  The **network-admin is** typically assigned to the **admin** user to allow it to run any command.

Built-in roles are not editable.

Examples

- These show role commands display the contents of the built-in roles.

```
switch(config)#show role network-operator
The default role is network-operator

role: network-operator
        10 deny mode exec command bash|\|
        20 permit mode exec command .*
switch(config)#show role network-admin
The default role is network-operator

role: network-admin
        10 permit command .*
switch(config)#
```

4.4.4.2 Managing Roles

Creating and Opening a Role

Roles are created and modified in role configuration mode. To create a role, enter the **role** command with the role's name. The switch enters role configuration mode. If the command is followed by the name of an existing role, subsequent commands edit that role.

Example

- This command places the switch in role configuration mode to create a role named ***sysuser***.

```
switch(config)#role sysuser
switch(config-role-sysuser)#
```

Saving Role Changes

Role configuration mode is a group-change mode; changes are saved by exiting the mode.

Examples

- These commands create a role, then adds a deny rule to the role. Because the changes are not yet saved, the role remains empty, as shown by **show role**.

```
switch(config)#role sysuser
switch(config-role-sysuser)#deny mode exec command reload
switch(config-role-sysuser)#show role sysuser
The default role is network-operator

switch(config)#
To save all current changes to the role and exit role configuration mode, type
exit.
switch(config-role-sysuser)#exit
switch(config)#show role sysuser
The default role is network-operator

role: sysuser
        10 deny mode exec command reload
switch(config)#
```

Important! After exiting role mode, ***running-config*** must be saved to ***startup-config*** to preserve role changes past system restarts.

Discarding Role Changes

The **abort** command exits role configuration mode without saving pending changes.

Example

- These commands enter role configuration mode to add deny rules, but discard the changes before saving them to the role.

```
switch(config)#role sysuser
switch(config-role-sysuser)#deny mode exec command reload
switch(config-role-sysuser)#abort
switch(config)#show role sysuser
The default role is network-operator

switch(config)#
```

4.4.4.3 Modifying Roles

Adding Rules to a Role

The **deny (Role)** command adds a deny rule to the configuration mode role. The **permit (Role)** command adds a permit rule to the configuration mode role.

To append a rule to the end of a role, enter the rule without a sequence number while in role configuration mode. The new rule's sequence number is derived by adding 10 to the last rule's sequence number.

Examples

- These commands enter the first three rules into a new role.

```
switch(config)#role sysuser
switch(config-role-sysuser)#deny mode exec command reload
switch(config-role-sysuser)#deny mode config command (no |default )?router
switch(config-role-sysuser)#permit command .*
switch(config-role-sysuser)#exit
switch(config)#show role sysuser
The default role is network-operator

role: sysuser
        10 deny mode exec command reload
        20 deny mode config command (no |default )?router
        30 permit command .*
switch(config)#
```

Inserting a Rule

To insert a rule into a role, enter the rule with a sequence number between the existing rules' numbers.

Example

- This command inserts a rule between the first two rules by assigning it the sequence number 15.

```
switch(config)#role sysuser
switch(config-role-sysuser)#15 deny mode config-all command lacp
switch(config-role-sysuser)#exit
switch(config)#show role sysuser
The default role is network-operator

role: sysuser
        10 deny mode exec command reload
        15 deny mode config-all command lacp|spanning-tree
        20 deny mode config command (no |default )router
        30 permit command .*
switch(config)#
```

Deleting a Rule

To remove a rule from the current role, perform one of these commands:

- Enter **no**, followed by the sequence number of the rule to be deleted.
- Enter **no**, followed by the rule be deleted.
- Enter **default**, followed by the sequence number of the rule to be deleted.
- Enter **default**, followed by the rule to be deleted.

Example

- These equivalent commands remove rule 30 from the list.

```
switch(config-role-sysuser)#no 30

switch(config-role-sysuser)#default 30

switch(config-role-sysuser)#no permit command .*

switch(config-role-sysuser)#default permit command .*
```

This role results from entering one of the preceding commands.

```
switch(config)#show role sysuser
The default role is network-operator

role: sysuser
        10 deny mode exec command reload
        15 deny mode config-all command lacp|spanning-tree
        20 deny mode config command (no |default )router
switch(config)#
```

Redistributing Sequence Numbers

Sequence numbers determine the order of the rules in a role. After a list editing session where existing rules are deleted and new rules are inserted between existing rules, the sequence number distribution may not be uniform. Redistributing rule numbers changes adjusts the sequence number of rules to provide a constant difference between adjacent rules. The **resequence (Role)** command adjusts the sequence numbers of role rules.

Example

- The **resequence** command renumbers rules in the sysuser role. The sequence number of the first rule is 100; subsequent rules numbers are incremented by 20.

```
switch(config)#show role sysuser
The default role is network-operator

role: sysuser
        10 deny mode exec command reload
        20 deny mode config-all command lacp|spanning-tree
        25 deny mode config command (no |default )?router
        30 permit command .*
switch(config)#role sysuser
switch(config-role-sysuser)#resequence 100 20
switch(config-role-sysuser)#exit
switch(config)#show role sysuser
The default role is network-operator

role: sysuser
        100 deny mode exec command reload
        120 deny mode config-all command lacp|spanning-tree
        140 deny mode config command (no |default )?router
        160 permit command .*
switch(config)#
```

4.4.5 Assigning a Role to a User Name

Roles are assigned to local users through the username command and to remote users through RADIUS servers. Each user is assigned one role. Each role can be assigned to multiple local and remote users.

4.4.5.1 Default Roles

Users that are not explicitly assigned a role are assigned the default role. The **aaa authorization policy local default-role** command designates the default role. The network-operator built-in role is the default role when the **aaa authorization policy local default-role** is not configured.

Examples

- These commands assign ***sysuser*** as the default role, then displays the name of the default role.

```
switch(config)#aaa authorization policy local default-role sysuser
switch(config)#show role
The default role is sysuser

             <-------OUTPUT OMITTED FROM EXAMPLE-------->
switch(config)#
```

- These commands restore ***network-operator*** as the default role by deleting the **aaa authorization policy local default-role** statement from ***running-config***, then displays the default role name.

```
switch(config)#no aaa authorization policy local default-role
switch(config)#show role
The default role is network-operator

             <-------OUTPUT OMITTED FROM EXAMPLE-------->
switch(config)#
```

4.4.5.2 Local Security File (Username command)

Roles are assigned to users with the **username** command's ***role*** parameter. A user name whose ***running-config*** **username** statement does not include a ***role*** parameter is assigned the ***default*** role.

The ***role*** parameter function in a command creating a user name is different from its function in a command editing an existing name.

Assigning a Role to a New User Name

A username command creating a user name explicitly assigns a role to the user name by including the ***role*** parameter; commands without a ***role*** parameter assigns the default role to the user name.

Example

- These commands create two user names. The first user is assigned a role; the second user assumes the default role.

```
switch(config)#username FRED secret 0 axced role sysuser1
switch(config)#username JANE nopassword
switch(config)#show running-config
             <-------OUTPUT OMITTED FROM EXAMPLE-------->
!
username FRED role sysuser1 secret 5 $1$dhJ6vrPV$PFOvJCX/vcqyIHV.vd.l20
username JANE nopassword
!
             <-------OUTPUT OMITTED FROM EXAMPLE-------->
switch(config)#
```

Editing the Role of an Existing User Name

The role of a previously configured user name may be edited by a **username** command without altering its password. The role assignment of a user name is not changed by **username** commands that do not include a ***role*** parameter.

Examples

- These commands assign a role to a previously configured user name.

```
switch(config)#username JANE role sysuser2
switch(config)#show running-config
             <-------OUTPUT OMITTED FROM EXAMPLE-------->
!
username FRED role sysuser1 secret 5 $1$dhJ6vrPV$PFOvJCX/vcqyIHV.vd.l20
username JANE role sysuser2 nopassword
!
             <-------OUTPUT OMITTED FROM EXAMPLE-------->
switch(config)#
```

- These commands reverts a user name to the default role by removing its role assignment.

```
switch(config)#no username FRED role
switch(config)#show running-config
             <-------OUTPUT OMITTED FROM EXAMPLE-------->
!
username FRED secret 5 $1$dhJ6vrPV$PFOvJCX/vcqyIHV.vd.l20
username JANE role sysuser2 nopassword
!
             <-------OUTPUT OMITTED FROM EXAMPLE-------->
switch(config)#
```

Displaying the Role Assignments

The **show user-account** command displays role assignment of the configured users. The **show aaa sessions** command displays roles of users that are currently logged into the switch.

Example

- This command displays the configured users and their role assignments.

```
switch(config)#show user-account
user: FRED
       role: <unknown>
       privilege level: 1
user: JANE
       role: sysuser2
       privilege level: 1
user: admin
       role: network-admin
       privilege level: 1
switch(config)#
```

- This command displays information about the active AAA login sessions.

```
switch(config)# show aaa session
Session  Username Roles           TTY    State Duration  Auth        Remote Host
-------   ----------
------------ ------ ----- --------  ------------ ------------
2      admin     network-operator ttyS0  E     0:01:21  local
4      Fred      sysadmin         telnet E     0:02:01  local
        sf.example.com
6      Jane      sysuser2         ssh    E     0:00:52  group radius  ny.exampl
e.com
9      admin     network-admin    ssh    E     0:00:07  local         bj.exampl
e.com
10     max       network-admin    telnet E     0:00:07  local         sf.exampl
e.com
```

4.4.5.3 Radius Servers

A role can be assigned to a remote user authenticated through a RADIUS server. Roles are assigned through the vendor specific attribute-value (AV) pair named ***Arista-AVPair***. The switch extracts the remote user's role upon a successful authentication when RADIUS authentication is enabled.

Example

- This file extract is sample FreeRadius server code that includes the AV pair that assigns roles to three remote users.

```
# Sample RADIUS server users file
"Jane"          Cleartext-Password := "Abc1235"
                Arista-AVPair = "shell:roles=sysuser2",
                Service-Type = NAS-Prompt-User
"Mary"          Cleartext-Password := "xYz$2469"
                Arista-AVPair = "shell:roles=sysadmin",
                Service-Type = NAS-Prompt-User
"Fred"          Cleartext-Password := "rjx4#222"
                Arista-AVPair = "shell:roles=network-operator",
                Service-Type = NAS-Prompt-User
```

The **aaa authentication login** command selects the user authentication service (Section 4.5.1.2).

Example

- This command configures the switch to authenticate users through all RADIUS servers.

```
switch(config)#aaa authentication login default group radius
switch(config)#
```

4.4.5.4 Enable Role Based Access Control

To enable Role Based Access Control on the switch, apply the following configuration:

```
switch(config)#aaa authorization commands all default local
switch(config)#
```

4.5 Activating Security Services

After configuring the access databases, **aaa authentication, aaa authorization**, and **aaa accounting** commands designate active and backup services for handling access requests.

These sections describe the methods of selecting the database that the switch uses to authenticate users and authorize access to network resources.

4.5.1 Authenticating Usernames and the Enable Password

Service lists specify the services the switch uses to authenticates usernames and the enable password.

4.5.1.1 Service List Description

Service list elements are service options, ordered by their priority.

Important! When the local file is one of the service list elements, any attempts to locally authenticate a username that is not included in the local file will result in the switch continuing to the next service list element.

Example

- This is an example service list for username authentication:
  1. Location_1 server group – specifies a server group (Section 4.3: Server Groups).
  2. Location_2 server group – specifies a server group (Section 4.3: Server Groups).
  3. TACACS+ servers – specifies all hosts for which a **tacacs-server host** command exists.
  4. Local file – specifies the local file.
  5. None – specifies that no authentication is required – all access attempts succeed.

To authenticate a username, the switch checks Location_1 server group. If a server in the group is available, the switch authenticates the username through that group. Otherwise, it continues through the list until it finds an available service or utilizes option 5, which allows the access attempt to succeed without authentication.

4.5.1.2 Configuring Service Lists

Service lists are incorporated into these **aaa authentication** commands to specify services the switch uses to authenticate usernames and the enable password.

- **aaa authentication login** specifies services the switch uses to authenticates usernames.
- **aaa authentication enable** specifies services the switch uses to authenticates the enable password.

Examples

- This command configures the switch to authenticate usernames through the TAC-1 server group. The local database is the backup method if TAC-1 servers are unavailable.

  ```
  switch(config)#aaa authentication login default group TAC-1 local
  ```

- This command configures the switch to authenticate usernames through all TACACS+ servers, then all RADIUS servers if the TACACS+ servers are not available. If the RADIUS servers are unavailable, the switch does not authenticate any login attempts.

  ```
  switch(config)#aaa authentication login default group tacacs+ group radius none
  ```

- This command configures the switch to authenticate the enable password through all TACACS+ servers, then through the local database if the TACACS+ servers are unavailable.

```
switch(config)#aaa authentication enable default group TACACS+ local
```

4.5.2 Authorization

Authorization commands control EOS shell access, CLI command access, and configuration access through the console port. The switch also supports role based authorization, which allows access to specified CLI commands by assigning command profiles (or roles) to usernames. Section 4.4 describes role based authorization.

During the exec authorization process, TACACS+ server responses may include attribute-value (AV) pairs. The switch recognizes the mandatory AV pair named **priv-lvl=x** (where **x** is between 0 and 15).

By default, a TACACS+ server that sends any other mandatory AV pair is denied access to the switch. The receipt of optional AV pairs by the switch has no affect on decisions to permit or deny access to the TACACS+ server. The **tacacs-server policy** command programs the switch to allow access to TACACS+ servers that send unrecognized mandatory AV pairs.

Authorization to switch services is configured by these **aaa authorization** commands

- To specify the method of authorizing the opening of an EOS shell, enter **aaa authorization exec**.
- To specify the method of authorizing CLI commands, enter **aaa authorization commands**.

Examples

- This command specifies that TACACS+ servers authorize users attempting to open a CLI shell.

```
switch(config)#aaa authorization exec default group tacacs+
switch(config)#
```

- This command programs the switch to authorize configuration commands (privilege level 15) through the local file and to deny command access to users not listed in the local file.

```
switch(config)#aaa authorization commands all default local
switch(config)#
```

- This command programs the switch to permit all commands entered on the CLI.

```
switch(config)#aaa authorization commands all default none
switch(config)#
```

- This command configures the switch to permit access to TACACS+ servers that send unrecognized mandatory AV pairs.

```
switch(config)#tacacs-server policy unknown-mandatory-attribute ignore
switch(config)#
```

All commands are typically authorized through **aaa authorization commands**. However, the **no aaa authorization config-commands** command disables the authorization of configuration commands. In this state, authorization to execute configuration commands can be managed by controlling access to Global Configuration commands. The default setting authorizes configuration commands through the policy specified for all other commands.

- To enable the authorization of configuration commands with the policy specified for all other commands, enter **aaa authorization config-commands.**
- To require authorization of commands entered on the console, enter **aaa authorization console**.

By default, EOS does not verify authorization of commands entered on the console port.

Examples

- This command disables the authorization of configuration commands.

```
switch(config)#no aaa authorization config-commands
switch(config)#
```

- This command enables the authorization of configuration commands.

```
switch(config)#aaa authorization config-commands
switch(config)#
```

- This command configures the switch to authorize commands entered on the console, using the method specified through a previously executed **aaa authorization command**.

```
switch(config)#aaa authorization console
switch(config)#
```

4.5.3 Accounting

The accounting service collects information for billing, auditing, and reporting. The switch supports TACACS+ and RADIUS accounting by reporting user activity to either the TACACS+ server or RADIUS server in the form of accounting records.

The switch supports two types of accounting:

- EXEC: Provides information about user CLI sessions.
- Commands: Command authorization for all commands, including configuration commands that are associated with a privilege level.

The accounting mode determines when accounting notices are sent. Mode options include:

- start-stop: a *start* notice is sent when a process begins; a *stop* notice is sent when it ends.
- stop-only: a *stop* accounting record is generated after a process successfully completes.

Accounting is enabled by the **aaa accounting** command.

Examples

- This command configures the switch to maintain start-stop accounting records for all commands executed by switch users and submits them to all TACACS+ hosts.

```
switch(config)#aaa accounting commands all default start-stop group tacacs+
switch(config)#
```

- This command configures the switch to maintain stop accounting records for all user EXEC sessions performed through the console and submits them to all TACACS+ hosts.

```
switch(config)#aaa accounting exec console stop group tacacs+
switch(config)#
```

4.6 TACACS+ Configuration Examples

These sections describe two sample TACACS+ host configurations.

4.6.1 Single Host Configuration

The single host configuration consists of a TACACS+ server with these attributes:

- IP address: 10.1.1.10
- encryption key: example_1
- port number: 49 (global default)
- timeout: 5 seconds (global default)

The switch authenticates the username and enable command against all TACACS+ servers which, in this case, is one host. If the TACACS+ server is unavailable, the switch authenticates with the local file.

Step 1 This step configures TACACS+ server settings – port number and timeout are global defaults.

```
switch(config)#tacacs-server host 10.1.1.10 key example_1
```

Step 2 This step configures the login authentication service.

```
switch(config)#aaa authentication login default group tacacs+ local
```

Step 3 This step configures the enable command password authentication service.

```
switch(config)#aaa authentication enable default group tacacs+ local
```

4.6.2 Multiple Host Configuration

The multiple host configuration consists of three TACACS+ servers at these locations:

- IP address 10.1.1.2 – port 49
- IP address 13.21.4.12 – port 4900
- IP address – 16.1.2.10 – port 49

The configuration combines the servers into these server groups:

- Bldg_1 group consists of the servers at 10.1.1.2 and 13.21.4.12
- Bldg_2 group consists of the servers at 16.1.2.10

All servers use these global TACACS+ defaults:

- encryption key – example_2
- timeout – 10 seconds

The switch authenticates these access methods:

- username access against Bldg_1 group then, if they are not available, against the local file.
- enable command against Bldg_2 group, then Bldg_1 group, then against the local file.

Step 1 TACACS+ Host commands:

These commands configure the IP address and ports for the three TACACS+ servers. The port for the first and third server is default 49.

```
switch(config)#tacacs-server host 10.1.1.12
switch(config)#tacacs-server host 13.21.4.12 port 4900
switch(config)#tacacs-server host 16.1.2.10
```

Step 2 Global Configuration Commands:

These commands configure the global encryption key and timeout values.

```
switch(config)#tacacs-server key example_2
switch(config)#tacacs-server timeout 10
```

Step 3 Group Server Commands:

The **aaa group server** commands create the server groups and place the CLI in server group configuration mode, during which the servers are placed in the group. The port number must be included if it is not the default port, as in the line that adds 192.168.1.1.

```
switch(config)#aaa group server tacacs+ Bldg_1
switch(config-sg-tacacs+-Bldg_1)#server 10.1.1.2
switch(config-sg-tacacs+-Bldg_1)#server 192.168.1.1 port 4900
switch(config-sg-tacacs+-Bldg_1)#exit
switch(config)#aaa group server tacacs+ Bldg_2
switch(config-sg-tacacs+-Bldg_2)#server 192.168.2.2
switch(config-sg-tacacs+-Bldg_2)#exit
switch(config)#
```

Step 4 Login and enable configuration authentication responsibility commands:

These commands configure the username and enable command password authentication services.

```
switch(config)#aaa authentication login default group Bldg_1 local
switch(config)#aaa authentication enable default group Bldg_1 group Bldg_2
local
```

4.7 AAA Commands

Local Security File Commands

- aaa root
- enable secret
- username
- username sshkey
- show privilege
- show user-account
- show users

Accounting, Authentication, and Authorization Commands

- aaa accounting
- aaa accounting dot1x
- aaa accounting system
- aaa authentication enable
- aaa authentication login
- aaa authentication policy local
- aaa authentication policy log
- aaa authorization commands
- aaa authorization config-commands
- aaa authorization console
- aaa authorization exec
- aaa authorization policy local default-role
- clear aaa counters
- clear aaa counters radius
- clear aaa counters tacacs+
- show aaa
- show aaa counters
- show aaa method-lists
- show aaa sessions

Server (RADIUS and TACACS+) Configuration Commands

- ip radius source-interface
- radius-server deadtime
- radius-server host
- radius-server key
- radius-server retransmit
- radius-server timeout
- show radius
- ip tacacs source-interface
- tacacs-server host
- tacacs-server key
- tacacs-server policy
- tacacs-server timeout
- show tacacs

Server Group Configuration Commands

- aaa group server radius
- aaa group server tacacs+
- server (server-group-RADIUS configuration mode)

- server (server-group-TACACS+ configuration mode)

Role Based Authorization Configuration Commands

- role
- deny (Role)
- permit (Role)
- no <sequence number> (Role)
- resequence (Role)
- show role

aaa accounting

The **aaa accounting** command configures accounting method lists for a specified authorization type. Each list consists of a prioritized list of methods. The accounting module uses the first available listed method for the authorization type.

The **no aaa accounting** and **default aaa accounting** commands clear the specified method list by removing the corresponding **aaa accounting** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
aaa accounting TYPE CONNECTION MODE [METHOD_1] [METHOD_2] ... [METHOD_N]
no aaa accounting TYPE CONNECTION
default aaa accounting TYPE CONNECTION
```

Parameters

- ***TYPE*** authorization type for which the command specifies a method list. Options include:
  - **EXEC** records user authentication events.
  - **COMMANDS ALL** records all entered commands.
  - **COMMANDS** *level* records entered commands of the specified *level* (ranges from 0 to 15).
- ***CONNECTION*** connection type of sessions for which method lists are reported. Options include:
  - **console** console connection.
  - **default** all connections not covered by other command options.
- ***MODE*** accounting mode that defines when accounting notices are sent. Options include:
  - **none** no notices are sent.
  - **start-stop** a *start* notice is sent when a process begins; a *stop* notice is sent when it ends.
  - **stop-only** a *stop* accounting record is generated after a process successfully completes.
- ***METHOD_X*** server groups (methods) to which the switch can send accounting records. The switch sends the method list to the first listed group that is available.

  Parameter value is not specified if ***MODE*** is set to ***none***. If ***MODE*** is not set to ***none,*** the command must provide at least one method. Each method is composed of one of the following:
  - **group** *name* the server group identified by *name.*
  - **group radius** server group that includes all defined RADIUS hosts.
  - **group tacacs+** server group that includes all defined TACACS+ hosts.
  - **logging** log all accounting messages to syslog.

Examples

- This command configures the switch to maintain start-stop accounting records for all commands executed by switch users and submits them to all TACACS+ hosts.

  ```
  switch(config)#aaa accounting commands all default start-stop group tacacs+
  switch(config)#
  ```

- This command configures the switch to maintain stop accounting records for all user EXEC sessions performed through the console and submits them to all TACACS+ hosts.

  ```
  switch(config)#aaa accounting exec console stop group tacacs+
  switch(config)#
  ```

aaa accounting dot1x

The **aaa accounting system** command enables the accounting of requested 802.1X services for network access.

The **no aaa accounting system** and **default aaa accounting system** commands disable the specified method list by removing the corresponding **aaa accounting system** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
aaa accounting dot1x default MODE [METHOD_1] [METHOD_2] ... [METHOD_N]
no aaa accounting dot1x default
default aaa accounting dot1z default
```

Parameters

- ***MODE*** accounting mode that defines when accounting notices are sent. Options include:
  - **start-stop** a *start* notice is sent when a process begins; a *stop* notice is sent when it ends.
- ***METHOD_X*** server groups (methods) to which the switch can send accounting records. The switch sends the method list to the first listed group that is available.

  Parameter value is not specified if ***MODE*** is set to ***none***. If ***MODE*** is not set to ***none,*** the command must provide at least one method. Each method is composed of one of the following:

  - **group** *name* the server group identified by *name.*
  - **group radius** server group that includes all defined RADIUS hosts.
  - **logging** server group that includes all defined TACACS+ hosts.

Examples

- This example configures IEEE 802.1x accounting on the switch.

```
switch(config)#aaa accounting dot1x default start-stop group radius
switch(config)#
```

- This example disables IEEE 802.1x accounting on the switch.

```
switch(config)#no aaa accounting dot1x default
switch(config)#
```

aaa accounting system

The **aaa accounting system** command performs accounting for all system-level events.

The **no aaa accounting system** and **default aaa accounting system** commands clear the specified method list by removing the corresponding **aaa accounting system** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
aaa accounting system default MODE [METHOD_1] [METHOD_2] ... [METHOD_N]
no aaa accounting system default
default aaa accounting system default
```

Parameters

- ***MODE*** accounting mode that defines when accounting notices are sent. Options include:
  - **none** no notices are sent.
  - **start-stop** a *start* notice is sent when a process begins; a *stop* notice is sent when it ends.
  - **stop-only** a *stop* accounting record is generated after a process successfully completes.
- ***METHOD_X*** server groups (methods) to which the switch can send accounting records. The switch sends the method list to the first listed group that is available.

  Parameter value is not specified if ***MODE*** is set to ***none***. If ***MODE*** is not set to ***none,*** the command must provide at least one method. Each method is composed of one of the following:
  - **group** *name* the server group identified by *name.*
  - **group radius** server group that includes all defined RADIUS hosts.
  - **group tacacs+** server group that includes all defined TACACS+ hosts.
  - **logging** server group that includes all defined TACACS+ hosts.

Examples

- This command configures AAA accounting to not use any accounting methods.

```
switch(config)#aaa accounting system default none
switch(config)#
```

- This command configures the switch to maintain stop accounting records for all user EXEC sessions performed through the console and submits them to all TACACS+ hosts.

```
switch(config)#aaa accounting exec console stop group tacacs+
switch(config)#
```

aaa authentication enable

The **aaa authentication enable** command configures the service list that the switch references to authorize access to Privileged EXEC command mode.

The list consists of a prioritized list of service options. Available service options include:

- a named server group
- all defined TACACS+ hosts
- all defined RADIUS hosts
- local authentication
- no authentication

The switch authorizes access by using the first listed service option that is available. When the local file is a service list element, attempts to locally authenticate a username that is not in the local file result in the switch continuing to the next service list element.

When the list is not configured, it is set to **local**.

The **no aaa authentication enable** and **default aaa authentication enable** commands revert the list configuration as **local** by removing the **aaa authentication enable** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
aaa authentication enable default METHOD_1 [METHOD_2] ... [METHOD_N]
no aaa authentication enable default
default aaa authentication enable default
```

Parameters

- ***METHOD_X*** authentication service method list. The command must provide at least one method. Each method is composed of one of the following:
  - **group** *name* the server group identified by *name.*
  - **group radius** a server group that consists of all defined RADIUS hosts.
  - **group tacacs+** a server group that consists of all defined TACACS+ hosts.
  - **local** local authentication.
  - **none** users are not authenticated; all access attempts succeed.

Example

- This command configures the switch to authenticate the enable password through all configured TACACS+ servers. Local authentication is the backup if TACACS+ servers are unavailable.

```
switch(config)#aaa authentication default enable group TACACS+ local
switch(config)#
```

aaa authentication login

The **aaa authentication login** command configures service lists the switch references to authenticate usernames. Service lists consist of service options ordered by usage priority. The switch authenticates usernames through the first available service option. Supported service options include:

- a named server group
- all defined TACACS+ hosts
- all defined RADIUS hosts
- local authentication
- no authentication

When the local file is a service list element, attempts to locally authenticate a username that is not in the local file result in the switch continuing to the next service list element.

The switch supports a ***console*** list for authenticating usernames through the console and a default list for authenticating usernames through all other connections.

- When the ***console*** list is not configured, the console connection uses the ***default*** list.
- When the ***default*** list is not configured, it is set to *local*.

The **no aaa authentication login** and **default aaa authentication login** commands revert the specified list configuration to its default by removing the corresponding **aaa authentication** login command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
aaa authentication login CONNECTION SERVICE_1 [SERVICE_2] ... [SERVICE_N]
no aaa authentication login CONNECTION
default aaa authentication login CONNECTION
```

Parameters

- ***CONNECTION*** connection type of sessions for which authentication list is used
  - **default** the default authentication list.
  - **console** the authentication list for console logins.
- ***SERVICE_X*** an authentication service. Settings include:
  - **group** *name* identifies a previously defined server group.
  - **group radius** a server group that consists of all defined RADIUS hosts.
  - **group tacacs+** a server group that consists of all defined TACACS+ hosts.
  - **local** local authentication.
  - **none** the switch does not perform authentication. All access attempts succeed.

Examples

- This command configures the switch to authenticate usernames through the TAC-1 server group. The local database is the backup method if TAC-1 servers are unavailable.

```
switch(config)#aaa authentication login default group TAC-1 local
switch(config)#
```

- This command configures the switch to authenticate usernames through all TACACS+ servers, then all RADIUS servers if the TACACS+ servers are not available. If the RADIUS servers are also unavailable, the switch allows access to all login attempts without authentication.

```
switch(config)#aaa authentication login default group tacacs+ group radius none
switch(config)#
```

aaa authentication policy local

The **aaa authentication policy local allow-nopassword-remote-login** command permits usernames without passwords to log in from any port. The default switch setting only allows unprotected usernames to log in from the console.

The **no aaa authentication policy local allow-nopassword-remote-login** and **default aaa authentication policy local allow-nopassword-remote-login** commands return the switch to the default setting of denying unprotected usernames to log in except from the console.

Command Mode

Global Configuration

Command Syntax

```
aaa authentication policy local allow-nopassword-remote-login
no aaa authentication policy local allow-nopassword-remote-login
default aaa authentication policy local allow-nopassword-remote-login
```

Examples

- This command configures the switch to allow unprotected usernames to log in from any port.

```
switch(config)#aaa authentication policy local allow-nopassword-remote-login
switch(config)#
```

- This command configures the switch to allow unprotected usernames to log in only from the console port.

```
switch(config)#no aaa authentication policy local allow-nopassword-remote-login
switch(config)#
```

aaa authentication policy log

The **aaa authentication policy log** command configure TACACS+ for remote AAA services.

A Common Criteria compliant AAA setup with EOS requires the use of TACACS+ as the AAA solution. For security it must be run inside of an SSH Tunnel. The remote TACACS+ server being connected to must implement TACACS+ protocol version 1.78 or greater to be considered Common Criteria compliant.

The **no aaa authentication policy log** and **default aaa authentication policy log** commands return the switch to the default setting of denying unprotected usernames to log in except from the console.

Command Mode

Global Configuration

Command Syntax

```
aaa authentication policy LOGIN_TYPE log
no aaa authentication policy LOGIN_TYPE log
default aaa authentication policy LOGIN_TYPE log
```

Parameters

- ***MODE*** accounting mode that defines when accounting notices are sent. Options include:
  - **on-failure** a notice is sent when a process begins; a *stop* notice is sent when it ends.
  - **on-success** a record is generated after a process successfully completes.

Examples

To configure TACACS+ on the switch run all of the following commands.

- Configure the server and keys:

```
switch(config)#tacacs-server host HOST key TACACS_KEY
```

- Configure user authentication:

```
switch(config)#aaa authentication login default group tacacs+ local
switch(config)#aaa authentication enable default group tacacs+ local
```

- Configure authentication policy to log successful and failed login attempts:

```
switch(config)#aaa authentication policy on-success log
switch(config)#aaa authentication policy on-failure log
```

aaa authorization commands

The **aaa authorization commands** command configures the service list that authorizes CLI command access. All switch commands are assigned a privilege level that corresponds to the lowest level command mode from which it can be executed:

- Level 1: Commands accessible from EXEC mode.
- Level 15: Commands accessible from any mode except EXEC.

Command usage is authorized for each privilege level specified in the command.

The list consists of a prioritized list of service options. The switch authorizes access by using the first listed service option that is available. The available service options include:

- a named server group
- all defined TACACS+ hosts
- all defined RADIUS hosts
- local authorization
- no authorization

The list is set to **none** for all unconfigured privilege levels, allowing all CLI access attempts to succeed.

The **no aaa authorization commands** and **default aaa authorization commands** commands revert the list contents to ***none*** for the specified privilege levels.

Command Mode

Global Configuration

Command Syntax

```
aaa authorization commands PRIV default SERVICE_1 [SERVICE_2] ... [SERVICE_N]
no aaa authorization commands PRIV default
default aaa authorization commands PRIV default
```

Parameters

- ***PRIV*** Privilege levels of the commands. Options include:
  - *level* numbers from **0** and **15.** Number, range, comma-delimited list of numbers and ranges.
  - **all** commands of all levels.
- ***SERVICE_X*** Authorization service. Command must list at least one service. Options include:
  - **group** *name* the server group identified by *name.*
  - **group tacacs+** a server group that consists of all defined TACACS+ hosts.
  - **local** local authorization.
  - **none** the switch does not perform authorization. All access attempts succeed.

Examples

- This command authorizes configuration commands (privilege level 15) through the local file. The switch denies command access to users not listed in the local file.

```
switch(config)#aaa authorization commands all default local
switch(config)#
```

- This command authorizes all commands entered on the CLI.

```
switch(config)#aaa authorization commands all default none
switch(config)#
```

aaa authorization config-commands

The **aaa authorization config-commands** command enables authorization of commands in any configuration mode, such as global Configuration and all interface configuration modes. Commands are authorized through the policy specified by the **aaa authorization commands** setting. Authorization is enabled by default, so issuing this command has no effect unless ***running-config*** contains the **no aaa authorization config-commands** command.

The **no aaa authorization config-commands** command disables configuration command authorization. When configuration command authorization is disabled, ***running-config*** contains the **no aaa authorization config-commands** command. The **default aaa authorization config-commands** command restores the default setting by removing the **no aaa authorization config-commands** from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
aaa authorization config-commands
no aaa authorization config-commands
default aaa authorization config-commands
```

Example

- This command disables the authorization of configuration commands.

```
switch(config)#no aaa authorization config-commands
switch(config)#
```

- This command enables the authorization of configuration commands.

```
switch(config)#aaa authorization config-commands
switch(config)#
```

aaa authorization console

The **aaa authorization console** command configures the switch to authorize commands entered through the console. By default, commands entered through the console do not require authorization.

The **no aaa authorization console** and **default aaa authorization console** commands restore the default setting.

Command Mode

Global Configuration

Command Syntax

```
aaa authorization console
no aaa authorization console
default aaa authorization console
```

Example

- This command configures the switch to authorize commands entered on the console, using the method specified through a previously executed **aaa authorization command**.

```
switch(config)#aaa authorization console
switch(config)#
```

aaa authorization exec

The **aaa authorization exec** command configures the service list that the switch references to authorize access to open an EOS CLI shell.

The list consists of a prioritized list of service options. The switch authorizes access by using the first listed service option to which the switch can connect. When the switch cannot communicate with an entity that provides a specified service option, it attempts to use the next option in the list.

The available service options include:

- a named server group
- all defined TACACS+ hosts
- all defined RADIUS hosts
- local authentication
- no authentication

When the list is not configured, it is set to **none**, allowing all CLI access attempts to succeed.

The **no aaa authorization exec** and **default aaa authorization exec** commands set the list contents to **none**.

Command Mode

Global Configuration

Command Syntax

```
aaa authorization exec default METHOD_1 [METHOD_2] ... [METHOD_N]
no aaa authorization exec default
default aaa authorization exec default
```

Parameters

- ***METHOD_X*** authorization service (method). The switch uses the first listed available method.

  The command must provide at least one method. Each method is composed of one of the following:

  - **group** *name* the server group identified by *name.*
  - **group radius** a server group that consists of all defined RADIUS hosts.
  - **group tacacs+** a server group that consists of all defined TACACS+ hosts.
  - **local** local authentication.
  - **none** the switch does not perform authorization. All access attempts succeed.

Guidelines

During the exec authorization process, the TACACS+ server response may include attribute-value (AV) pairs. The switch recognizes **priv-lvl=x** (where **x** is an integer between 0 and 15), which is a mandatory AV pair. A TACACS+ server that sends any other mandatory AV pair is denied access to the switch. The receipt of optional AV pairs by the switch has no affect on decisions to permit or deny access to the TACACS+ server.

Example

- This command specifies that the TACACS+ servers authorize users that attempt to open an EOS CLI shell.

```
switch(config)#aaa authorization exec default group tacacs+
switch(config)#
```

aaa authorization policy local default-role

The **aaa authorization policy local** command specifies the name of the default role. A role is a data structure that supports local command authorization through its assignment to user accounts. Roles consist of permit and deny rules that define authorization levels for specified commands. Applying a role to a username authorizes the user to execute commands specified by the role.

The default role is assigned to the following users:

- local or remote users assigned to a role that is not configured.
- local users to whom a role is not assigned.

When the default-role is not specified, **network-operator** is assigned to qualified users as the default role. The network-operator role authorizes assigned users access to all CLI commands in EXEC and Privileged EXEC modes.

The **no aaa authentication policy local default-role** and **default aaa authentication policy local default-role** commands remove the **authentication policy local default-role** statement from ***running-config***. Removing this statement restores **network-operator** as the default role.

Command Mode

Global Configuration

Command Syntax

```
aaa authorization policy local default-role role_name
no aaa authorization policy local default-role
default aaa authorization policy local default-role
```

Parameters

- *role_name* Name of the default role.

Related Commands

The **role** command places the switch in role configuration mode for creating and editing roles.

Examples

- This command configures the sysuser as the default role.

```
switch(config)#aaa authorization policy local default-role sysuser
switch(config)#
```

- This command restores **network-operator** as the default role.

```
switch(config)#no aaa authorization policy local default-role
switch(config)#
```

- This command displays the contents of the **network-operator** role.

```
switch#show role network-operator
The default role is network-operator

role: network-operator
        10 deny mode exec command bash|\|
        20 permit mode exec command .*
switch#
```

aaa group server radius

The **aaa group server radius** command enters the server-group-radius configuration mode for the specified group name. The command creates the specified group if it was not previously created. Commands are available to add servers to the group.

A server group is a collection of servers that are associated with a single label. Subsequent authorization and authentication commands access all servers in a group by invoking the group name. Server group members must be previously configured with a **radius-server host** command.

The **no aaa group server radius** and **default aaa group server radius** commands delete the specified server group from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
aaa group server radius group_name
no aaa group server radius group_name
default aaa group server radius group_name
```

Parameters

- *group_name* name (text string) assigned to the group. Cannot be identical to a name already assigned to a TACACS+ server group.

Commands Available in server-group-radius Configuration Mode

- **server (server-group-RADIUS configuration mode)**

Related Commands

- **aaa group server tacacs+**

Example

- This command creates the RADIUS server group named RAD-SV1 and enters server group configuration mode for the new group.

```
switch(config)#aaa group server radius RAD-SV1
switch(config-sg-radius-RAD-SV1)#
```

aaa group server tacacs+

The **aaa group server tacacs+** command enters server-group-tacacs+ configuration mode for the specified group name. The command creates the specified group if it was not previously created. Commands are available to add servers to the group.

A server group is a collection of servers that are associated with a single label. Subsequent authorization and authentication commands access all servers in a group by invoking the group name. Server group members must be previously configured with a **tacacs-server host** command.

The **no aaa group server tacacs+** and **default aaa group server tacacs+** commands delete the specified server group from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
aaa group server tacacs+ group_name
no aaa group server tacacs+ group_name
default aaa group server tacacs+ group_name
```

Parameters

- *group_name* name (text string) assigned to the group. Cannot be identical to a name already assigned to a RADIUS server group.

Commands Available in server-group-tacacs+ Configuration Mode

- **server (server-group-TACACS+ configuration mode)**

Related Commands

- **aaa group server radius**

Example

- This command creates the TACACS+ server group named TAC-GR and enters server group configuration mode for the new group.

```
switch(config)#aaa group server tacacs+ TAC-GR
switch(config-sg-tacacs+-TAC-GR)#
```

aaa root

The **aaa root** command specifies the password security level for the root account and can assign a password to the account.

The **no aaa root** and **default aaa root** commands disable the root account by removing the **aaa root** command from ***running-config***. The root account is disabled by default.

Command Mode

Global Configuration

Command Syntax

```
aaa root SECURITY_LEVEL [ENCRYPT_TYPE] [password]
no aaa root
default aaa root
```

Parameters

- ***SECURITY_LEVEL*** password assignment level. Settings include
  - **secret** the root account is assigned to the password.
  - **nopassword** the root account is not password protected.
- ***ENCRYPT_TYPE*** encryption level of the *password* parameter. This parameter is present only when ***SECURITY_LEVEL*** is **secret**. Settings include:
  - <no parameter> the password is entered as clear text.
  - **0** the password is entered as clear text. Equivalent to <no parameter>.
  - **5** the password is entered as an md5 encrypted string.
  - **sha512** the password is entered as an sha512 encrypted string.
- *password* text that authenticates the username. The command includes this parameter only if ***SECURITY_LEVEL*** is **secret**.
  - *password* must be in clear text if ***ENCRYPT_TYPE*** specifies clear text.
  - *password* must be an appropriately encrypted string if ***ENCRYPT_TYPE*** specifies encryption.

  Encrypted strings entered through this parameter are generated elsewhere.

Examples

- These equivalent commands assign **f4980** as the root account password.

```
switch(config)#aaa root secret f4980
switch(config)#aaa root secret 0 f4980
```

- This command assigns the text (ab234) that corresponds to the encrypted string of 1HW05LEY8$QEVw6JqjD9VqDfh.O8r.b. as the root password.

```
switch(config)#aaa root secret 5 $1$HW05LEY8$QEVw6JqjD9VqDfh.O8r.b
switch(config)#
```

- This command removes the password from the root account.

```
switch(config)#aaa root nopassword
switch(config)#
```

- This command disables the root login.

```
switch(config)#no aaa root
switch(config)#
```

clear aaa counters

The **clear aaa counters** command resets the counters that track the number of service transactions performed by the switch since the last time the counters were reset. The **show aaa counters** command displays the counters reset by the **clear aaa counters** command.

Command Mode

Privileged EXEC

Command Syntax

clear aaa counters [*SERVICE_TYPE*]

Example

- These commands display the effect of the **clear aaa counters** command on the aaa counters.

```
switch#clear aaa counters
switch#show aaa counters
Authentication
          Successful:          0
              Failed:          0
 Service unavailable:          0

Authorization
             Allowed:          1
              Denied:          0
 Service unavailable:          0

Accounting
          Successful:          0
               Error:          0
             Pending:          0

Last time counters were cleared: 0:00:44 ago
```

clear aaa counters radius

The **clear aaa counters radius** command resets the counters that track the statistics for the RADIUS servers that the switch access. The **show radius** command displays the counters reset by the **clear aaa counters radius** command.

Command Mode

Privileged EXEC

Command Syntax

clear aaa counters radius

Example

- These commands display the effect of the **clear aaa counters radius** command on the RADIUS counters.

```
switch#show radius
RADIUS server            : radius/10
          Connection opens:        204
         Connection closes:          0
    Connection disconnects:        199
       Connection failures:         10
       Connection timeouts:          2
             Messages sent:       1490
         Messages received:       1490
            Receive errors:          0
          Receive timeouts:          0
             Send timeouts:          0

Last time counters were cleared: never
switch#clear aaa counters radius
switch#show radius
RADIUS server            : radius/10
          Connection opens:          0
         Connection closes:          0
    Connection disconnects:          0
       Connection failures:          0
       Connection timeouts:          0
             Messages sent:          0
         Messages received:          0
            Receive errors:          0
          Receive timeouts:          0
             Send timeouts:          0

Last time counters were cleared: 0:00:03 ago
switch#
```

clear aaa counters tacacs+

The **clear aaa counters tacacs+** command resets the counters that track the statistics for the TACACS+ servers that the switch access. The **show tacacs** command displays the counters reset by the **clear aaa counters tacacs+** command.

Command Mode

Privileged EXEC

Command Syntax

```
clear aaa counters tacacs+
```

Example

- These commands display the effect of the **clear aaa counters tacacs+** command on the tacacs+ counters.

```
switch#show tacacs
TACACS+ server            : tacacs/49
          Connection opens:      15942
         Connection closes:          7
    Connection disconnects:       1362
       Connection failures:          0
       Connection timeouts:          0
             Messages sent:      34395
         Messages received:      34392
            Receive errors:          0
          Receive timeouts:          2
             Send timeouts:          0

Last time counters were cleared: never

TACACS+ source-interface: Enabled
  TACACS+ outgoing packets will be sourced with an IP address associated with the
Loopback0 interface
switch#clear aaa counters tacacs+
switch#show tacacs
TACACS+ server            : tacacs/49
          Connection opens:          0
         Connection closes:          0
    Connection disconnects:          0
       Connection failures:          0
       Connection timeouts:          0
             Messages sent:          0
         Messages received:          0
            Receive errors:          0
          Receive timeouts:          0
             Send timeouts:          0

Last time counters were cleared: 0:00:03 ago
switch#

TACACS+ source-interface: Enabled
  TACACS+ outgoing packets will be sourced with an IP address associated with the
Loopback0 interface
switch#
```

deny (Role)

The **deny** command adds a deny rule to the configuration mode role. Deny rules prohibit access of specified commands from usernames to which the role is applied. Sequence numbers determine rule placement in the role. Commands are compared sequentially to rules within a role until it matches a rule. A command's authorization is determined by the first rule it matches. Sequence numbers for commands without numbers are derived by adding 10 to the number of the role's last rule.

Deny rules use regular expression to denote commands. A ***mode*** parameter specifies command modes from which commands are restricted. Modes are denoted either by predefined keywords, a command mode's short key, or a regular expression that specifies the long key of one or more command modes.

The **no deny** and **default deny** commands remove the specified rule from the configuration mode role. The **no <sequence number> (Role)** command also removes the specified rule from the role.

Command Mode

Role Configuration

Command Syntax

```
[SEQ_NUM] deny [MODE_NAME] command command_name
no deny [MODE_NAME] command command_name
default deny [MODE_NAME] command command_name
```

Parameters

- ***SEQ_NUM*** Sequence number assigned to the rule. Options include:
  - <no parameter> Number is derived by adding 10 to the number of the role's last rule.
  - <***1 – 256***> Number assigned to entry.
- ***MODE_NAME*** Command mode from which command access is prohibited. Values include:
  - <no parameter> All command modes
  - **mode** *short_name* Exact match of a mode's short key name.
  - **mode** *long_name* Regular expression matching long key name of one or more modes.
  - **mode config** Global configuration mode.
  - **mode config-all** All configuration modes, including global configuration mode.
  - **mode exec** EXEC and Privileged EXEC modes.
- *command_name* Regular expression that denotes the name of one or more commands.

Guidelines

These CLI **prompt** format commands program the prompt to display the following mode keys:

- %p short mode key.
- %P long mode key.

Deny statements are saved to ***running-config*** only upon exiting role configuration mode.

Related Commands

The **role** command places the switch in role configuration mode.

Example

- These commands append a **deny** rule at the end of the ***sysuser*** role that restricts access to the **reload** command from EXEC and Privileged EXEC mode.

```
switch(config)#role sysuser
switch(config-mode-sysuser)#deny mode exec command reload
switch(config-mode-sysuser)#
```

enable secret

The **enable secret** command creates a new enable password or changes an existing password.

The **no enable secret** and **default enable secret** commands delete the **enable password** by removing the **enable secret** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
enable secret [ENCRYPT_TYPE] password
no enable secret
default enable secret
```

Parameters

- ***ENCRYPT_TYPE*** encryption level of the *password* parameter. Settings include:
  - <no parameter> the password is entered as clear text.
  - **0** the password is entered as clear text. Equivalent to <no parameter>.
  - **5** the password is entered as an md5 encrypted string.
  - **sha512** the password is entered as an sha512 encrypted string.
- *password* text that authenticates the username.
  - *password* must be in clear text if ***ENCRYPT_TYPE*** specifies clear text.
  - *password* must be an appropriately encrypted string if ***ENCRYPT_TYPE*** specifies encryption.

  Encrypted strings entered through this parameter are generated elsewhere.

Examples

- These equivalent commands assign *xyrt1* as the enable password.

```
switch(config)#enable secret xyrt1
switch(config)#enable secret 0 xyrt1
```

- This command assigns the enable password to the clear text (12345) that corresponds to the encrypted string *$1$8bPBrJnd$Z8wbKLHpJEd7d4tc5Z/6h/*. The string was generated by an MD5-encryption program using 12345 as the seed.

```
switch(config)#enable secret 5 $1$8bPBrJnd$Z8wbKLHpJEd7d4tc5Z/6h/
switch(config)#
```

- This command deletes the enable password.

```
switch(config)#no enable secret
switch(config)#
```

ip radius source-interface

The **ip radius source-interface** command specifies the interface from which the IPv4 address is derived for use as the source for outbound RADIUS packets. When a source interface is not specified, the switch selects an interface.

The **no ip radius source-interface** and **default ip radius source-interface** commands remove the **ip radius source-interface** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
ip radius [VRF_INST] source-interface INT_NAME
no ip radius [VRF_INST] source-interface
default ip radius [VRF_INST] source-interface
```

Parameters

- ***VRF_INST*** specifies the VRF instance used to communicate with the specified server.
  - <no parameter> switch communicates with the server using the default VRF.
  - **vrf** *vrf_name* switch communicates with the server using the specified user-defined VRF.
- ***INT_NAME*** Interface type and number. Options include:
  - **interface ethernet** *e_num* Ethernet interface specified by *e_num*.
  - **interface loopback** *l_num* Loopback interface specified by *l_num*.
  - **interface management** *m_num* Management interface specified by *m_num*.
  - **interface port-channel** *p_num* Port-Channel Interface specified by *p_num*.
  - **interface vlan** *v_num* VLAN interface specified by *v_num*.

Example

- This command configures the source address for outbound RADIUS packets as the IPv4 address assigned to the loopback interface.

```
switch(config)#ip radius source-interface loopback 0
switch(config)#
```

ip tacacs source-interface

The **ip tacacs source-interface** command specifies the interface from which the IPv4 address is derived for use as the source for outbound TACACS+ packets. When a source interface is not specified, the switch selects an interface.

The **no ip tacacs source-interface** and **default ip tacacs source-interface** commands remove the **ip tacacs source-interface** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
ip tacacs [VRF_INST] source-interface INT_NAME
no ip tacacs [VRF_INST] source-interface
default ip tacacs [VRF_INST] source-interface
```

Parameters

- ***VRF_INST*** specifies the VRF instance used to communicate with the specified server.
  - <no parameter> switch communicates with the server using the default VRF.
  - **vrf** *vrf_name* switch communicates with the server using the specified user-defined VRF.
- ***INT_NAME*** Interface type and number. Options include:
  - **interface ethernet** *e_num* Ethernet interface specified by *e_num*.
  - **interface loopback** *l_num* Loopback interface specified by *l_num*.
  - **interface management** *m_num* Management interface specified by *m_num*.
  - **interface port-channel** *p_num* Port-Channel Interface specified by *p_num*.
  - **interface vlan** *v_num* VLAN interface specified by *v_num*.

Example

- This command configures the source address for outbound TACACS+ packets as the IPv4 address assigned to the loopback interface.

```
switch(config)#ip tacacs source-interface loopback 0
switch(config)#
```

no <sequence number> (Role)

The **no <sequence number>** command removes the rule with the specified sequence number from the configuration mode role. The **default <sequence number>** command also removes the specified rule.

Command Mode

Role Configuration

Command Syntax

```
no sequence_num
default sequence_num
```

Parameters

- *sequence_num* sequence number of rule to be deleted. Values range from ***1 to 256***.

Guidelines

Role statement changes are saved to ***running-config*** only upon exiting role configuration mode.

Related Commands

The **role** command places the switch in role configuration mode.

Example

- These commands display the rules in the sysuser role, removes rule 30 from the role, then displays the edited role.

```
switch(config)#show role sysuser
The default role is network-operator

role: sysuser
        10 deny mode exec command reload
        20 deny mode config command (no |default )?router
        30 deny mode config command (no |default )?(ip|mac) access-list
        40 deny mode if command (no |default )?(ip|mac) access-group
        50 deny mode config-all command lacp|spanning-tree
        60 permit command .*
switch(config)#role sysuser
switch(config-role-sysuser)#no 30
switch(config-role-sysuser)#exit
switch(config)#show role sysuser
The default role is network-operator

role: sysuser
        10 deny mode exec command reload
        20 deny mode config command (no |default )?router
        40 deny mode if command (no |default )?(ip|mac) access-group
        50 deny mode config-all command lacp|spanning-tree
        60 permit command .*
switch(config)#
```

permit (Role)

The **permit** command adds a permit rule to the configuration mode role. Permit rules authorize access to specified commands for usernames to which the role is applied. Sequence numbers determine rule placement in the role. Commands are compared sequentially to rules within a role until it matches a rule. A command's authorization is determined by the first rule it matches. Sequence numbers for commands without numbers are derived by adding 10 to the number of the role's last rule.

Permit rules use regular expression to denote commands. A ***mode*** parameter specifies command modes in which commands are authorized. Modes are denoted either by predefined keywords, a command mode's short key, or a regular expression that specifies the long key of one or more command modes.

The **no deny** and **default deny** commands remove the specified rule from the configuration mode role. The **no <sequence number> (Role)** command also removes the specified rule from the role.

Command Mode

Role Configuration

Command Syntax

```
[SEQ_NUM] permit [MODE_NAME] command command_name
no permit [MODE_NAME] command command_name
default permit [MODE_NAME] command command_name
```

Parameters

- ***SEQ_NUM*** Sequence number assigned to the rule. Options include:
  - <no parameter> Number is derived by adding 10 to the number of the role's last rule.
  - <***1 – 256***> Number assigned to entry.
- ***MODE_NAME*** Command mode in which command access is authorized. Values include:
  - <no parameter> All command modes
  - **mode** *short_name* Exact match of a mode's short key name.
  - **mode** *long_name* Regular expression matching long key name of one or more modes.
  - **mode config** Global configuration mode.
  - **mode config-all** All configuration modes, including global configuration mode.
  - **mode exec** EXEC and Privileged EXEC modes.
- *command_name* Regular expression that denotes the name of one or more commands.

Guidelines

These CLI **prompt** format commands program the prompt to display the following mode keys:

- %p short mode key.
- %P long mode key.

Permit statements are saved to ***running-config*** only upon exiting role configuration mode.

Related Commands

The **role** command places the switch in role configuration mode.

Example

- These commands append a **permit** rule at the end of the sysuser role that authorizes all commands from VLAN 1 or VLAN 2 interface configuration modes.

```
switch(config)#role sysuser
switch(config-mode-sysuser)#permit mode if-Vl(1|2) command .*
switch(config-mode-sysuser)#
```

radius-server deadtime

The **radius-server deadtime** command defines global deadtime period, when the switch ignores a non-responsive RADIUS server. A non-responsive server is one that failed to answer any attempt to retransmit after a timeout expiry. Deadtime is disabled if a value is not configured.

The **no radius-server deadtime** and **default radius-server deadtime** commands restore the default global deadtime period of three minutes by removing the **radius-server deadtime** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
radius-server deadtime dead_interval
no radius-server deadtime
default radius-server deadtime
```

Parameters

- *dead_interval* period that the switch ignores non-responsive servers (minutes). Value ranges from 1 to 1000. Default is 3.

Related Commands

- **radius-server host**

Example

- This command programs the switch to ignore a server for two hours if it fails to respond to a request during the period defined by timeout and retransmit parameters.

```
switch(config)#radius-server deadtime 120
switch(config)#
```

radius-server host

The **radius-server host** command sets parameters for communicating with a specific RADIUS server. These values override global settings when the switch communicates with the specified server.

A RADIUS server is defined by its server address, authorization port, and accounting port. Servers with different address-authorization port-accounting port combinations have separate configurations.

The **no radius-server host** and **default radius-server** commands remove settings for the RADIUS server configuration at the specified address-authorization port-accounting port location by deleting the corresponding **radius-server host** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
radius-server host ADDR [VRF_INST][AUTH][ACCT][TIMEOUT][DEAD][RETRAN][ENCRYPT]
no radius-server host [ADDR][VRF_INST][AUTH][ACCT]
default radius-server host [ADDR][VRF_INST][AUTH][ACCT]
```

Parameters

- ***ADDR*** RADIUS server location. Options include:
  - *ipv4_addr* server's IPv4 address.
  - *host_name* server's DNS host name (FQDN).
- ***VRF_INST*** specifies the VRF instance used to communicate with the specified server.
  - <no parameter> switch communicates with the server using the default VRF.
  - **vrf** *vrf_name* switch communicates with the server using the specified user-defined VRF.
- ***AUTH*** Authorization port number.
  - <no parameter> default port of 1812.
  - **auth-port** *number* *number* ranges from 1 to 65535.
- ***ACCT*** Accounting port number.
  - <no parameter> default port of 1813.
  - **acct-port** *number* *number* ranges from 1 to 65535.
- ***TIMEOUT*** timeout period (seconds). Ranges from 1 to 1000.
  - <no parameter> assigns global timeout value (see **radius-server timeout**).
  - **timeout** *number* assigns *number* as the timeout period. Ranges from 1 to 1000.
- ***DEAD*** period (minutes) when the switch ignores a non-responsive RADIUS server.
  - <no parameter> assigns global deadtime value (see **radius-server deadtime**).
  - **deadtime** *number* specifies deadtime, where *number* ranges from 1 to 1000.
- ***RETRAN*** attempts to access RADIUS server after the first timeout expiry.
  - <no parameter> assigns global retransmit value (see **radius-server retransmit**).
  - **retransmit** *number* specifies number of attempts, where *number* ranges from 1 to 100.
- ***ENCRYPT*** encryption key that switch and server use to communicate.
  - <no parameter> assigns global encryption key (see **radius-server key**).
  - **key** *key_text* where *key_text* is in clear text.
  - **key 5** *key_text* where *key_text* is in clear text.
  - **key 7** *key_text* where *key_text* is provide in an encrypted string.

Examples

- This command configures the switch to communicate with the RADIUS server located at *10.1.1.5*. The switch uses the global timeout, deadtime, retransmit, and key settings to communicate with this server, and communicates through port 1812 for authorization and 1813 for accounting.

```
switch(config)#radius-server host 10.1.1.5
switch(config)#
```

- This command configures the switch to communicate with the RADIUS server assigned the host name *RAD-1*. Communication for authorization is through port 1850; communication for accounting is through port 1813 (the default).

```
switch(config)#radius-server host RAD-1 auth-port 1850
switch(config)#
```

radius-server key

The **radius-server key** command defines the global encryption key the switch uses when communicating with any RADIUS server for which a key is not defined.

The **no radius-server key** and **default radius-server key** commands remove the global key from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
radius-server key [ENCRYPT_TYPE] encrypt_key
no radius-server key
default radius-server key
```

Parameters

- ***ENCRYPT_TYPE*** encryption level of *encrypt_key*.
  - <no parameter> encryption key is entered as clear text.
  - **0** encryption key is entered as clear text. Equivalent to <no parameter>.
  - **7** *encrypt_key* is an encrypted string.
- *encrypt_key* shared key that authenticates the username.
  - *encrypt_key* must be in clear text if ***ENCRYPT_TYPE*** specifies clear text.
  - *encrypt_key* must be an encrypted string if ***ENCRYPT_TYPE*** specifies an encrypted string.

  Encrypted strings entered through this parameter are generated elsewhere.

Related Commands

- **radius-server host**

Examples

- This command configures *cv90jr1* as the global encryption key.

```
switch(config)#radius-server key 0 cv90jr1
switch(config)#
```

- This command assigns *cv90jr1* as the key by specifying the corresponding encrypted string.

```
switch(config)#radius-server key 7 020512025B0C1D70
switch(config)#
```

radius-server retransmit

The **radius-server retransmit** command defines the global retransmit count, which specifies the number of times the switch attempts to access the RADIUS server after the first timeout expiry.

The **no radius-server retransmit** and **default radius-server retransmit** commands restore the global retransmit count to its default value of three by deleting the **radius-server retransmit** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
radius-server retransmit count
no radius-server retransmit
default radius-server retransmit
```

Parameters

- *count* retransmit attempts after first timeout expiry. Settings range from 1 to 100. Default is 3.

Related Commands

- **radius-server host**

Example

- This command configures the switch to attempt five RADIUS server contacts after the initial timeout. If the timeout parameter is set to 50 seconds, then the total period that the switch waits for a response is ((5+1)*50) = 300 seconds.

```
switch(config)#radius-server retransmit 5
switch(config)#
```

radius-server timeout

The **radius-server timeout** command defines the global timeout the switch uses when communicating with any RADIUS server for which a timeout is not defined.

The **no radius-server timeout** and **default radius-server timeout** commands restore the global timeout default period of five seconds by removing the **radius-server timeout** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
radius-server timeout time_period
no radius-server timeout
default radius-server timeout
```

Parameters

- *time_period* timeout period (seconds). Range from 1 to 1000. Default is 5.

Related Commands

- **radius-server host**
- **radius-server key**
- **radius-server deadtime**
- **radius-server retransmit**

Example

- This command configures the switch to wait 50 seconds for a RADIUS server response before issuing an error.

```
switch(config)#radius-server timeout 50
switch(config)#
```

resequence (Role)

The **resequence** command assigns sequence numbers to rules in the configuration mode role. Command parameters specify the number of the first rule and the numeric interval between consecutive rules.

The maximum sequence number is 256.

Command Mode

Role Configuration

Command Syntax

resequence *start_num inc_num*

Parameters

- *start_num* sequence number assigned to the first rule. Value ranges from 1 to 256. Default is 10.
- *inc_num* numeric interval between consecutive rules. Value ranges from 1 to 256. Default is 10.

Guidelines

Role statement changes are saved to ***running-config*** only upon exiting role configuration mode.

Related Commands

The **role** command places the switch in role configuration mode.

Example

- The **resequence** command renumbers the rules in the sysuser role, starting the first rule at 15 and incrementing subsequent lines by 5.

```
switch(config)#show role sysuser
The default role is network-operator

role: sysuser
        10 deny mode exec command reload
        20 deny mode config command (no |default )?router
        40 deny mode if command (no |default )?(ip|mac) access-group
        50 deny mode config-all command lacp|spanning-tree
        60 permit command .*
switch(config)#role sysuser
switch(config-role-sysuser)#resequence 15 5
switch(config-role-sysuser)#exit
switch(config)#show role sysuser
The default role is network-operator

role: sysuser
        15 deny mode exec command reload
        20 deny mode config command (no |default )?router
        25 deny mode if command (no |default )?(ip|mac) access-group
        30 deny mode config-all command lacp|spanning-tree
        35 permit command .*
switch(config)#role sysuser
```

role

The **role** command places the switch in role configuration mode, which is a group change mode that modifies a role. A role is a data structure that supports local command authorization through its assignment to user accounts. Roles consist of permit and deny rules that define authorization levels for specified commands. Applying a role to a username authorizes the user to execute commands specified by the role.

The **role** command specifies the name of the role that subsequent commands modify and creates a role if it references a nonexistent role. All changes in a group change mode edit session are pending until the session ends:

- The **exit** command saves pending changes to ***running-config*** and returns the switch to global configuration mode. Changes are also saved by entering a different configuration mode.
- The **abort** command discards pending changes, returning the switch to global configuration mode.

The **no role** and **default role** commands delete the specified role by removing the role and its statements from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
role role_name
no role role_name
default role role_name
```

Parameters

- *role_name* Name of role.

Commands Available in Role configuration mode:

- **deny (Role)**
- **permit (Role)**
- **no <sequence number> (Role)**
- **resequence (Role)**

Related Commands

- **show role**

Examples

- This command places the switch in role configuration mode to modify the speaker role.

```
switch(config)#role speaker
switch(config-role-speaker)#
```

- This command saves changes to ***speaker*** role, then returns the switch to global configuration mode.

```
switch(config-role-speaker)#exit
switch(config)#
```

- This command discards changes to ***speaker***, then returns the switch to global configuration mode.

```
switch(config-role-speaker)#abort
switch(config)#
```

server (server-group-RADIUS configuration mode)

The **server (server-group-RADIUS configuration mode)** command adds the specified RADIUS server to the configuration mode group. Servers must be configured with the **radius-server host** command before adding them to the server group.

A RADIUS server is defined by its server address, authorization port, and accounting port. A group can contain multiple servers with the same IP address that have different authorization or accounting ports.

The **no server** and **default server** commands remove the specified server from the group.

Command Mode

Server-Group-RADIUS Configuration

Command Syntax

```
server LOCATION [VRF_INST][AUTH][ACCT]
no server LOCATION [VRF_INST][AUTH][ACCT]
default server LOCATION [VRF_INST][AUTH][ACCT]
```

Parameters

- ***LOCATION*** RADIUS server location. Options include:
  - *ipv4_addr* server's IPv4 address.
  - *host_name* server's DNS host name (FQDN).
- ***VRF_INST*** specifies the VRF instance used to communicate with the specified server.
  - <no parameter> switch communicates with the server using the default VRF.
  - **vrf** *vrf_name* switch communicates with the server using the specified user-defined VRF.
- ***AUTH*** Authorization port number.
  - <no parameter> default port of 1812.
  - **auth-port** *number* *number* ranges from 1 to 65535.
- ***ACCT*** Accounting port number.
  - <no parameter> default port of 1813.
  - **acct-port** *number* *number* ranges from 1 to 65535.

Related Commands

The **aaa group server radius** command places the switch in server-group-radius configuration mode.

Example

- These commands add two servers to the RAD-SV1 server group.

```
switch(config)#aaa group server radius RAD-SV1
switch(config-sg-radius-RAD-SV1)#server RAC-1
switch(config-sg-radius-RAD-SV1)#server 10.1.5.14 acct-port 1851
switch(config-sg-radius-RAD-SV1)#
```

server (server-group-TACACS+ configuration mode)

The **server (server-group-TACACS+ configuration mode)** command adds the specified TACACS+ server to the configuration mode group. Servers must be configured with the **tacacs-server host** command before adding them to the server group.

A TACACS+ server is defined by its server address and port number. Servers with different address-port combinations have separate statements in ***running-config***.

The **no server** and **default server** commands remove the specified server from the group.

Command Mode

Server-Group-TACACS+ Configuration

Command Syntax

```
server LOCATION [VRF_INST] [PORT]
no server LOCATION [VRF_INST] [PORT]
default server LOCATION [VRF_INST] [PORT]
```

Parameters

- ***LOCATION*** TACACS+ server location. Options include:
  - *ipv4_addr* server's IPv4 address.
  - *ipv6_addr* server's IPv6 address.
  - *host_name* server's DNS host name (FQDN).
- ***VRF_INST*** specifies the VRF instance used to communicate with the specified server.
  - <no parameter> switch communicates with the server using the default VRF.
  - **vrf** *vrf_name* switch communicates with the server using the specified user-defined VRF.
- ***PORT*** TCP connection port number.
  - <no parameter> default port of 49.
  - **port** *number* *number* ranges from 1 to 65535.

Related Commands

The **aaa group server tacacs+** command places the switch in server-group-radius configuration mode.

Example

- These commands add two servers to the TAC-GR server group with default port number 49.

```
switch(config)#aaa group server tacacs+ TAC-GR
switch(config-sg-tacacs+-TAC-GR)#server TAC-1
switch(config-sg-tacacs+-TAC-GR)#server 10.1.4.14
switch(config-sg-tacacs+-TAC-GR)#
```

show aaa

The **show aaa** command displays the user database. The command displays the encrypted enable password first, followed by a table of usernames and their corresponding encrypted password.

The command does not display unencrypted passwords.

Command Mode

Privileged EXEC

Command Syntax

show aaa

Example

- This command configures the switch to authenticate the enable password through all configured TACACS+ servers. Local authentication is the backup if TACACS+ servers are unavailable.

```
switch#show aaa
Enable password (encrypted): $1$UL4gDWy6$3KqCPYPGRvxDxUq3qA/Hs/
Username  Encrypted passwd
--------  --------------------------------
admin
janis     $1$VVnDH/Ea$iwsfnrGNO8nbDsf0tazp9/
thomas    $1$/MmXTUil$.fJxLfcumzppNSEDVDWq9.
switch#
```

show aaa counters

The **show aaa counters** command displays the number of service transactions performed by the switch since the last time the counters were reset.

Command Mode

Privileged EXEC

Command Syntax

show aaa counters

Example

- This command displays the number of authentication, authorization, and accounting transactions.

```
switch#show aaa counters
Authentication
          Successful:         30
              Failed:          0
 Service unavailable:          0

Authorization
             Allowed:        188
              Denied:          0
 Service unavailable:          0

Accounting
          Successful:          0
               Error:          0
             Pending:          0

Last time counters were cleared: never
switch#
```

show aaa method-lists

The **show aaa method-lists** command displays all the named method lists defined in the specified authentication, authorization, and accounting (AAA) service.

Command Mode

Privileged EXEC

Command Syntax

show aaa method-lists *SERVICE_TYPE*

Parameters

- ***SERVICE_TYPE*** the service type of the method lists that the command displays.
  - **accounting** accounting services.
  - **authentication** authentication services.
  - **authorization** authorization services.
  - **all** accounting, authentication, and authorization services.

Example

- This command configures the named method lists for all AAA services.

```
switch#show aaa method-lists all
Authentication method lists for LOGIN:
  name=default methods=group tacacs+, local
Authentication method list for ENABLE:
  name=default methods=local
Authorization method lists for COMMANDS:
  name=privilege0-15 methods=group tacacs+, local
Authentication method list for EXEC:
  name=exec methods=group tacacs+, local
Accounting method lists for COMMANDS:
  name=privilege0-15 default-action=none
Accounting method list for EXEC:
  name=exec default-action=none
switch#
```

show aaa sessions

The **show aaa sessions** command displays information about active AAA login sessions. Information includes username, roles, TTY, state of the session (pending or established), duration, authentication method, and if available, remote host and remote username.

Command Mode

Privileged EXEC

Command Syntax

show aaa sessions

Example

- This command displays information about the active AAA login sessions.

```
switch# show aaa session
Session  Username Roles           TTY    State Duration  Auth        Remote Host
-------   ----------
------------ ------ ----- --------  ------------ ------------
2      admin     network-admin    ttyS0  E     0:01:21  local
4      joe       sysadmin         telnet E     0:02:01  local
        sf.example.com
6      alice     sysadmin         ssh    E     0:00:52  group radius  ny.exampl
e.com
7      bob       sysadmin         ssh    E     0:00:48  group radius  la.exampl
e.com
8      kim       network-admin1   ssh    E     0:00:55  group radius  de.exampl
e.com
9      admin     network-admin    ssh    E     0:00:07  local         bj.exampl
e.com
10     max       network-admin    telnet E     0:00:07  local         sf.exampl
e.com
```

show privilege

The **show privilege** command displays the current privilege level for the CLI session.

Command Mode

EXEC

Command Syntax

show privilege

Example

- This command displays the current privilege level.

```
switch>show privilege
Current privilege level is 15
switch>
```

show radius

The **show radius** command displays statistics for the RADIUS servers that the switch accesses.

Command Mode

EXEC

Command Syntax

show radius

Example

- This command displays statistics for connected TACACS+ servers.

```
switch>show radius
RADIUS server            : radius/10
          Connection opens:        204
         Connection closes:          0
    Connection disconnects:        199
       Connection failures:         10
       Connection timeouts:          2
             Messages sent:       1490
         Messages received:       1490
            Receive errors:          0
          Receive timeouts:          0
             Send timeouts:          0

Last time counters were cleared: never
switch>
```

show role

The **show role** command displays the name of the default role and the contents of the specified roles. Commands that do not specify a role display the rules in all built-in and configured roles.

Command Mode

Privileged EXEC

Command Syntax

show role [*ROLE_LIST*]

Parameters

- ***ROLE_LIST*** Roles that the command displays. Options include:
  - <no parameter> Command displays all roles.
  - *role_name* Name of role displayed by command.

Related Commands

The **role** command places the switch in role configuration mode, which is used to create new roles or modify existing roles.

Example

- This command displays the contents of all user-defined and built-in roles.

```
switch#show role
The default role is network-operator

role: network-admin
        10 permit command .*
role: network-operator
        10 deny mode exec command bash|\|
        20 permit mode exec command .*
role: sysuser
        15 deny mode exec command reload
        20 deny mode config command (no |default )?router
        25 deny mode if command (no |default )?(ip|mac) access-group
        30 deny mode config-all command lacp|spanning-tree
        35 permit command .*
        40 deny mode exec command .*
        50 permit mode exec command show|clear (counters|platform)|configure
switch#
```

show tacacs

The **show tacacs** command displays statistics for the TACACS+ servers that the switch accesses.

Command Mode

EXEC

Command Syntax

show tacacs

Example

- This command displays statistics for connected TACACS+ servers.

```
switch>show tacacs
TACACS+ server            : tacacs/49
          Connection opens:      15942
         Connection closes:          7
    Connection disconnects:       1362
       Connection failures:          0
       Connection timeouts:          0
             Messages sent:      34395
         Messages received:      34392
            Receive errors:          0
          Receive timeouts:          2
             Send timeouts:          0

Last time counters were cleared: never

TACACS+ source-interface: Enabled
  TACACS+ outgoing packets will be sourced with an IP address associated with the
Loopback0 interface
switch>
```

show user-account

The **show user-account** command displays the names, roles, and privilege levels of users that are listed in ***running-config***. The ssh public-key is also listed for names for which an SSH key is configured.

Command Mode

Privileged EXEC

Command Syntax

show user-account

Example

- This command displays the usernames that are configured on the switch.

```
switch#show user-account
user: FRED
       role: <unknown>
       privilege level: 1
       ssh public key: ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQDjUg2VDiBX7In0q
HtN5PyHOWtYvIoeZsxF5YmesQ/rh++mbpT504dL7So+Bpr9T/0qIj+zilat8fX/JlO42+3pjfkHY/+l
sT2EPNjGTK7uJv1wSGmhc3+90dNmJtr5YVlJFjjQ5m+5Pa+PGe3z4JIV1lY2NhLrV2fXtbciLdjnj6F
AlhXjiLt51DJhG13uUxGBJe0+NlGvpEsTJVJvMdJuS6weMi+xSXc9yQimVD2weJBHsYFnghST2j0pAy
F2S7/EOU13pY42RztDSs42nMNNrutPT0q5Z17aAKvhpd0dDlc+qIwrCrXbeIChHem7+0N8/zA3alBK4
eKSFSZBd3Pb admin@switch
switch#
user: JANE
       role: sysuser2
       privilege level: 1
user: admin
       role: network-admin
       privilege level: 1
```

show users

The **show users** command displays the usernames that are currently logged into the switch.

Command Mode

Privileged EXEC

Command Syntax

show users

Example

- This command displays the users that are logged into the switch.

```
switch#show users
    Line       User       Host(s)              Idle       Location
   1 vty 2     john       idle                    1d  10.22.6.113
   2 vty 4     jane       idle                21:33:00  10.22.26.26
*  3 vty 6     ted        idle                00:00:01  10.17.18.71

switch#
```

tacacs-server host

The **tacacs-server host** command sets communication parameters for communicating with a specific TACACS+ server. These values override global settings when the switch communicates with the specified server.

A TACACS+ server is defined by its server address and port number. Servers with different combinations of address-port-VRF-multiplex settings have separate statements in ***running-config***.

The **no tacacs-server host** and **default tacacs-server host** commands remove settings for the TACACS+ server configuration at the specified address-port-VRF combination by deleting the corresponding **tacacs-server host** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
tacacs-server host SERVER_ADDR [MULTIPLEX][VRF_INST][PORT][TIMEOUT][ENCRYPT]
no tacacs-server host [SERVER_ADDR][MULTIPLEX][VRF_INST][PORT]
default tacacs-server host [SERVER_ADDR][MULTIPLEX][VRF_INST][PORT]
```

Parameters

- ***SERVER_ADDR*** TACACS+ server location. Options include:
  - *ipv4_addr* server's IPv4 address.
  - *ipv6_addr* server's IPv6 address.
  - *host_name* server's DNS host name (FQDN).
- ***MULTIPLEX*** TACACS+ server support of multiplex sessions on a TCP connection.
  - <no parameter> server does not support multiplexing.
  - **single-connection** server supports session multiplexing.
- ***VRF_INST*** specifies the VRF instance used to communicate with the specified server.
  - <no parameter> switch communicates with the server using the default VRF.
  - **vrf** *vrf_name* switch communicates with the server using the specified user-defined VRF.
- ***PORT*** port number of the TCP connection.
  - <no parameter> default port of 49.
  - **port** *number* port *number* ranges from 1 to 65535.
- ***TIMEOUT*** timeout period (seconds).
  - <no parameter> assigns the globally configured timeout value (see **tacacs-server timeout**).
  - **timeout** *number* timeout period (seconds). *number* ranges from 1 to 1000.
- ***ENCRYPT*** encryption key the switch and server use to communicate. Settings include
  - <no parameter> assigns the globally configured encryption key (see **tacacs-server key**).
  - **key** *key_text* where *key_text* is in clear text.
  - **key 5** *key_text* where *key_text* is in clear text.
  - **key 7** *key_text* where *key_text* is an encrypted string.

Examples

- This command configures the switch to communicate with the TACACS+ server located at *10.1.1.5*. The switch uses the global timeout, encryption key, and port settings.

```
switch(config)#tacacs-server host 10.1.1.5
switch(config)#
```

- This command configures the switch to communicate with the TACACS+ server assigned the host name *TAC_1*. The switch defines the timeout period as *20 seconds* and the encryption key as *rp31E2v*.

```
switch(config)#tacacs-server host TAC_1 timeout 20 key rp31E2v
switch(config)#
```

- This command configures the switch to communicate with the TACACS+ server located at *10.12.7.9,* indicates that the server supports multiplexing sessions on the same TCP connection, and that access is through port 54.

```
switch(config)#tacacs-server host 10.12.7.9 single-connection port 54
switch(config)#
```

tacacs-server key

The **tacacs-server key** command defines the global encryption key the switch uses when communicating with any TACACS+ server for which a key is not defined.

The **no tacacs-server key** and **default tacacs-server key** commands remove the global key from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
tacacs-server key [ENCRYPT_TYPE] encrypt_key
no tacacs-server key
default tacacs-server key
```

Parameters

- ***ENCRYPT_TYPE*** encryption level of *encrypt_key*.
  - <no parameter> encryption key is entered as clear text.
  - **0** encryption key is entered as clear text. Equivalent to <no parameter>.
  - **7** *encrypt_key* is an encrypted string.
- *encrypt_key* shared key that authenticates the username.
  - *encrypt_key* must be in clear text if ***ENCRYPT_TYPE*** specifies clear text.
  - *encrypt_key* must be an encrypted string if ***ENCRYPT_TYPE*** specifies an encrypted string.

  Encrypted strings entered through this parameter are generated elsewhere.

Related Commands

- **tacacs-server host**

Examples

- This command configures *cv90jr1* as the encryption key.

```
switch(config)#tacacs-server key 0 cv90jr1
switch(config)#
```

- This command assigns *cv90jr1* as the key by specifying the corresponding encrypted string.

```
switch(config)#tacacs-server key 7 020512025B0C1D70
switch(config)#
```

tacacs-server policy

The **tacacs-server policy** command programs the switch to permit access to TACACS+ servers that send mandatory attribute-value (AV) pairs that the switch does not recognize. By default, the switch denies access to TACACS+ servers when it received unrecognized AV pairs from the server.

The switch recognizes the following mandatory AV pairs:

- **priv-lvl=x** where x is an integer between 0 and 15.

The **no tacacs-server policy** and **default tacacs-server policy** commands restore the switch default of denying access to servers from which it receives unrecognized mandatory AV pair by deleting the ***tacacs-server policy*** statement from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
tacacs-server policy unknown-mandatory-attribute ignore
no tacacs-server policy unknown-mandatory-attribute ignore
default tacacs-server policy unknown-mandatory-attribute ignore
```

Example

- This command configures the switch to permit access to TACACS+ servers that send unrecognized mandatory AV pairs.

```
switch(config)#tacacs-server policy unknown-mandatory-attribute ignore
switch(config)#
```

tacacs-server timeout

The **tacacs-server timeout** command defines the global timeout the switch uses when communicating with any TACACS+ server for which a timeout is not defined.

The **no tacacs-server timeout** and **default tacacs-server timeout** commands restore the global timeout default period of five seconds by removing the **tacacs-server timeout** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
tacacs-server timeout time_period
no tacacs-server timeout
default tacacs-server timeout
```

Parameters

- *time_period* timeout period (seconds). Settings range from 1 to 1000. Default is 5.

Related Commands

- **tacacs-server host**

Example

- This command configures the switch to wait 20 seconds for a TACACS+ server response before issuing an error.

```
switch(config)#tacacs-server timeout 20
switch(config)#
```

username

The **username** command adds a username to the local file and assigns a password to a username. If the command specifies an existing username, the command replaces the password in the local file. The command can define a username without a password or remove the password from a username.

The **no username** and **default username** commands delete the specified username by removing the corresponding **username** statement from ***running-config***.

The **no username role** command assigns the default role assignment to the specified **username** statement by editing the corresponding **username** statement in ***running-config***. The **default username role** command reverts the specified username to its default role by editing the corresponding **username** statement in ***running-config***. For the admin username, this restores **network-admin** as its role.

Command Mode

Global Configuration

Command Syntax

```
username name [PRIVILEGE_LEVEL] SECURITY [ROLE_USER]
no username name [role]
default username name [role]
```

All parameters except *name* can be placed in any order.

Parameters

- *name* username text that the user enters at the login prompt to access the CLI.

  Valid usernames begin with A-Z, a-z, or 0-9 and may also contain any of these characters:

  ```
  @   #   $   %   ^   &   *   -   _   =
  +   ;   <   >   ,   .   ~   |
  ```

- ***PRIVILEGE_LEVEL*** user's initial session privilege level. This parameter is used when an authorization command includes the local option.
  - <no parameter> the privilege level is set to 1.
  - **privilege** *rank* where *rank* is an integer between 0 and 15.
- ***SECURITY*** password assignment option.
  - **nopassword** *name* is not password protected.
  - **secret** *password* *name* is protected by specified password (clear-text string).
  - **secret 0** *password* *name* is protected by specified password (clear-text string).
  - **secret 5** *password* *name* is protected by specified password. (md5 encrypted string).
  - **secret sha5** *password* *name* is protected by specified password (sha512 encrypted string).
- ***ROLE_USER*** specifies the role for performing command authorization. Options include:
  - <no parameter> user is assigned default role (**aaa authorization policy local default-role**).
  - **role** *role_name* specifies role assigned to the user.

Guidelines

Encrypted strings entered through this parameter are generated elsewhere. The **secret 5** option (***SECURITY***) is typically used to enter a list of username-passwords from a script.

The ***SECURITY*** parameter is mandatory for unconfigured usernames. For previously configured users, the command can specify a ***PRIVILEGE_LEVEL*** or ***ROLE*** without a ***SECURITY*** setting.

admin is a reserved username that is provided by the initial configuration. The **admin** username cannot be deleted, but its parameters are editable. The initial **admin** configuration is:

```
username admin privilege 1 role network-admin nopassword
```

Examples

- These equivalent commands create the username *john* and assigns it the password *x245*. The password is entered in clear text because the ***ENCRYPTION*** parameter is either omitted or zero.

  ```
  switch(config)#username john secret x245
  switch(config)#username john secret 0 x245
  ```

- This command creates the username *john* and assigns it to the text password that corresponds to the encrypted string *1sU.7hptc$TsJ1qslCL7ZYVbyXNG1wg1.* The string was generated by an MD5-encryption program using x245 as the seed.

  ```
  switch(config)#username john secret 5 $1$sU.7hptc$TsJ1qslCL7ZYVbyXNG1wg1
  switch(config)#
  ```

  A user authenticates the username *john* by entering ***x245*** when the CLI prompts for a password.

- This command creates the username *jane* without securing it with a password. It also removes a password if the *jane* username exists.

  ```
  switch(config)#username jane nopassword
  switch(config)#
  ```

- This command removes the username **william** from the local file.

  ```
  switch(config)#no username william
  switch(config)#
  ```

username sshkey

The **username sshkey** command configures an ssh key for the specified username. Command options allow the key to be entered directly into the CLI or referenced from a file.

The specified username must be previously configured through a **username** command.

The **no username sshkey** and **default username sshkey** commands delete the sshkey for the specified username by removing the corresponding **username sshkey** command from ***running-config***.

The **no username sshkey role** and **default username sshkey role** commands perform the following:

- delete the sshkey for the specified username by removing the corresponding **username sshkey** command from ***running-config***.
- delete the role assignment from the specified username by editing the corresponding **username** statement in ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
username name sshkey KEY
no username name sshkey [role]
default username name sshkey [role]
```

Parameters

- *name* username text that the user enters at the login prompt to access the CLI.

  Valid usernames begin with A-Z, a-z, or 0-9 and may also contain any of these characters:

  ```
  @   #   $   %   ^   &   *   -   _   =
  +   ;   <   >   ,   .   ~   |
  ```

- ***KEY*** SSH key. Options include:
  - *key_text* username is associated with ssh key specified by *key_text* string.
  - **file** *key_file* username is associated with ssh key in the specified file.

Example

- These commands create the username *john*, assign it the password *x245,* then associate it to the SSH key listed in the file named ***john-ssh***.

  ```
  switch(config)#username john secret x245
  switch(config)#username john sshkey file john-ssh
  switch(config)#
  ```

Chapter 5

Administering the Switch

This chapter describes administrative tasks that are typically performed only after initially configuring the switch or after recovery procedures.

This chapter includes these sections:

- Section 5.1: Managing the Switch Name
- Section 5.2: Managing the System Clock
- Section 5.3: Synchronizing the Time Settings
- Section 5.4: Managing Display Attributes
- Section 5.5: Event Monitor
- Section 5.6: Switch Administration Commands

5.1 Managing the Switch Name

These sections describe how to configure the switch's domain and host name.

- Section 5.1.1: Assigning a Name to the Switch describes the assigning of an FQDN to the switch.
- Section 5.1.2: Specifying DNS Addresses describes the adding of name servers to the configuration.

5.1.1 Assigning a Name to the Switch

A fully qualified domain name (FQDN) labels the switch and defines its organization ID in the Domain Name System hierarchy. The switch's FQDN consists of a host name and domain name.

The host name is uniquely associated with one device within an IP-domain. The default host name is ***localhost***. You can configure the prompt to display the host name, as described in Section 5.4.2: Prompt.

- To assign a host name to the switch, use the **hostname** command. To return the switch's host name to the default value of ***localhost***, use the **no hostname** command.
- To specify the domain location of the switch, use the **ip domain-name** command.

Example

- This command assigns the string *main-host* as the switch's host name.

```
switch(config)#hostname main-host
main-host(config)#
```

- This command configures *aristanetworks.com* as the switch's domain name.

```
switch(config)#ip domain-name aristanetworks.com
switch(config)#
```

- This procedure configures ***sales1.samplecorp.org*** as the switch's FQDN.

```
switch(config)#ip domain-name samplecorp.org
switch(config)#
```

- This ***running-config*** extract contains the switch's host name and IP-domain name.

```
switch#show running-config
! Command: show running-config
! device: switch (DCS-7150S-64-CL, EOS-4.13.2F)
!
              <-------OUTPUT OMITTED FROM EXAMPLE-------->
vlan 3-4
!
username john secret 5 $1$a7Hjept9$TIKRX6ytkg8o.ENja.na50
!
hostname sales1
ip name-server 172.17.0.22
ip domain-name samplecorp.org
!
              <-------OUTPUT OMITTED FROM EXAMPLE-------->
end
switch#
```

5.1.2 Specifying DNS Addresses

The Domain Name Server (DNS) maps FQDN labels to IP addresses and provides addresses for network devices. Each network requires at least one server to resolve addresses. The configuration file can list a maximum of three server addresses.

To add name servers to the configuration, use the **ip name-server** command. Each command can add one to three servers. The switch disregards any attempt to add a fourth server to the configuration. All server addresses must be in a single VRF. If servers have been previously configured in a different VRF they must be removed before adding a new server to the configuration.

Example

- This code performs these actions:
  - adds three names servers to the configuration in the default VRF
  - attempts to add a fourth server, resulting in an error message
  - displays the configuration file.

```
switch(config)#ip name-server 10.1.1.24 10.1.1.25 172.17.0.22
switch(config)#ip name-server 10.15.3.28
% Maximum number of nameservers reached. '10.15.3.28' not added
switch(config)#show running-config
! device: Switch (EOS-4.11.2-1056939.EOS4112)
!
username david secret 5 $1$a7Hjept9$TIKRX6ytkg8o.ENja.na50
!
hostname Switch
ip name-server 10.1.1.24
ip name-server 10.1.1.25
ip name-server 172.17.0.22
ip domain-name aristanetworks.com
              <-------OUTPUT OMITTED FROM EXAMPLE-------->
```

The switch assigns source IP addresses to outgoing DNS requests. To force the switch to use a single, user-defined source interface for all requests, use the **ip domain lookup** command.

Example

- This command forces the switch to use VLAN 5 as the source interface for DNS requests originating from the default VRF.

```
switch(config)#ip domain lookup source-interface Vlan5
switch(config)#
```

- This command forces the switch to use VLAN 10 as the source interface for DNS requests originating from VRF “purple.”

```
switch(config)#ip domain lookup vrf purple source-interface Vlan10
switch(config)#
```

5.2 Managing the System Clock

The switch uses the system clock for displaying the time and time-stamping messages. The system clock is set to Coordinated Universal Time (UTC). The switch calculates local time based on the time zone setting. Time-stamps and time displays are in local time.

5.2.1 Configuring the Time Zone

The time zone setting is used by the switch to convert the system time (UTC) to local time. To specify the time zone, use the **clock timezone** command.

Examples

- These commands configure the switch for the United States Central Time Zone.

```
switch(config)#clock timezone US/Central
switch(config)#show clock
Mon Jan 14 18:42:49 2013
timezone is US/Central
switch(config)#
```

- To view the predefined time zone labels, enter **clock timezone** with a question mark.

```
switch(config)#clock timezone ?
  Africa/Abidjan                    Africa/Accra
             <-------OUTPUT OMITTED FROM EXAMPLE-------->
  WET                               WET timezone
  Zulu                              Zulu timezone

switch(config)#clock timezone
```

- This command displays all time zone labels that start with **America**.

```
switch(config)#clock timezone AMERICA?
America/Adak                      America/Anchorage
             <-------OUTPUT OMITTED FROM EXAMPLE-------->
America/Yellowknife

switch(config)#clock timezone AMERICA
```

5.2.2 Setting the System Clock Manually

The **clock set** command manually configures the system clock time and date, in local time. Any NTP servers properly configured on the switch override time that is manually entered.

Example

- This command manually sets the switch time.

```
switch#clock set 08:15:24 14 Jan 2013
Mon Jan 14 08:15:25 2013
timezone is US/Central
```

5.2.3 Displaying the Time

To display the local time and configured time zone, enter the **show clock** command.

Example

- This command displays the switch time.

```
switch(config)>show clock
Mon Jan 14 16:32:46 2013
timezone is America/Los_Angeles
```

5.3 Synchronizing the Time Settings

Time settings are synchronized through Network Time Protocol (NTP).

5.3.1 Network Time Protocol (NTP)

Network Time Protocol (NTP) servers synchronize time settings of systems running an NTP client. The switch supports NTP versions 1 through 4. The default is version 4.

After configuring the switch to synchronize with an NTP server, it may take up to ten minutes for the switch to set its clock. The ***running-config*** lists NTP servers that the switch is configured to use.

5.3.1.1 Configuring the NTP Server

The **ntp server** command adds a server to the list or modifies the parameters of a previously listed address. When the system contains multiple NTP servers, the ***prefer*** keyword can be used to specify a preferred NTP server, which will be used as the NTP server if not discarded by NTP.

Note that all NTP servers must be in the same VRF, and that they are added in the default VRF if no VRF is specified.

Example

- These commands add three NTP servers, designating the second server as preferred.

```
switch(config)#ntp server local-NTP
switch(config)#ntp server 172.16.0.23 Prefer
switch(config)#ntp server 172.16.0.25
```

5.3.1.2 Configuring the NTP Source

The **ntp source** command configures an interface as the source of NTP packets. That interface's IP address is then used as the source address for all NTP packets unless a server-specific source is configured using the **source** option of the **ntp server** command. For an **ntp source** command to take effect, the specified interface and the NTP server must both belong to the same VRF.

Example

- This command configures VLAN interface 25 as the source of NTP update packets.

```
switch(config)#ntp source vlan 25
switch(config)#
```

5.3.1.3 Configuring the Switch as an NTP Server

To configure the switch to accept NTP requests on all interfaces, use the **ntp serve all** command to enable NTP server mode globally on the switch. To configure an individual interface to accept or deny NTP requests, use the **ntp serve** command. Interface level settings override the global settings, and changing the settings at either the global or interface level also causes the switch to re-synchronize with its upstream NTP server. NTP server mode is disabled by default.

Example

- This command configures the switch to act as an NTP server, accepting NTP requests.

```
switch(config)# ntp serve all
switch(config)#
```

- These commands configure Ethernet interface 5 to accept NTP requests regardless of global settings.

```
switch(config)#interface ethernet 5
switch(config-if-Et5)#ntp serve
switch(config-if-Et5)#
```

5.3.1.4 Configuring NTP Authentication

The switch can be configured to synchronize its clock using NTP packets only from an authenticated source. NTP authentication is disabled by default.

To configure the switch to authenticate NTP packets, create one or more authentication keys using the **ntp authentication-key** command, specify which keys are trusted by using the **ntp trusted-key** command, and use the **ntp authenticate** command to enable NTP authentication. The NTP server must be configured to use the same authentication key and key ID number.

Example

- These commands configure the switch to authenticate NTP packets using key 328 with the plaintext password “timeSync.”

```
switch(config)# ntp authentication-key 328 md5 timeSync
switch(config)# ntp trusted key 328
switch(config)# ntp authenticate
switch(config)#
```

5.3.1.5 Viewing NTP Settings and Status

To display the status of Network Time Protocol (NTP) on the switch, use the **show ntp status** command. To display the status of connections to NTP servers, use the **show ntp associations** command. Note that for IPv4 addresses, the reference ID is the IPv4 address of the NTP server. For IPv6 addresses, the reference ID is the first four octets of the MD5 hash of the NTP server’s IP address.

Example

- This command displays the status of the switch’s NTP connection.

```
switch#show ntp status
unsynchronised
  time server re-starting
   polling server every 64 s
switch #
```

- This command displays data about the NTP servers in the configuration.

```
switch#show ntp associations
     remote           refid      st t when poll reach   delay   offset  jitter
==============================================================================
 moose.aristanet 66.187.233.4     2 u    9   64  377    0.118  9440498   0.017
 172.17.2.6      .INIT.          16 u    - 1024    0    0.000    0.000   0.000
*LOCAL(0)        .LOCL.          10 l   41   64  377    0.000    0.000   0.000
switch#
```

5.3.2 Precision Time Protocol (PTP)

The Precision Time Protocol (PTP) enhances the accuracy of real-time clocks in networked devices by providing sub-microsecond clock synchronization. Inbound clock signals are organized into a master-slave hierarchy. PTP identifies the switch port that is connected to the device with the most precise clock. This clock is referred to as the master clock. All the other devices on the network synchronize their clocks with the master and are referred to as slaves.

The master clock sends out a sync message every second. The slave clock sends a delay request message to the master clock noting the time it was sent in order to measure and eliminate packet delays. The master clock then replies with the time stamp the delay message was received. The slave clock then computes the master clock time compensated for delays and finalizes synchronization. Constantly exchanged timing messages ensure continued synchronization.

5.3.2.1 Enable PTP

The following PTP commands are required to enable PTP on a device:

- Set the PTP Mode
- Enable PTP on An Interface

Set the PTP Mode

To specify the Precision Time Protocol (PTP), use the **ptp mode** command. PTP mode options include:

- **boundary** The device acts as a boundary clock, and both runs and participates in the best master clock algorithm.
- **disabled** PTP is disabled, and the device forwards all PTP packets as normal traffic.
- **end-to-end transparent** The device acts as an end-to-end transparent clock, synchronizing all ports to a connected master clock and updating the time interval field of forwarded PTP packets using switch residence time.
- **peer-to-peer transparent** The device acts as a peer-to-peer transparent clock, synchronizing all ports to a connected master clock and updating the time interval field of forwarded PTP packets using switch residence time and inbound path delays.
- **generalized Precision Time Protocol (gPTP)** The device runs generalized Precision Time Protocol (gPTP), participating in the best master clock algorithm but also updating the interval field of forwarded PTP packets using switch residence time and inbound path delays.

Example

- This command configures the device as a PTP boundary clock.

```
switch(config)# ptp mode boundary
switch(config)#
```

Enable PTP on An Interface

To enable PTP on a specific interface on the device, use the **ptp enable** command.

Example

- This command enables PTP on Ethernet interface 5.

```
switch(config)# interface ethernet 5
switch(config-if-Et5)# ptp enable
```

5.3.2.2 Configuring PTP Global Options

The following PTP global commands are optional:

- Configure the PTP Domain
- Configure the Offset Hold Time
- Set the PTP Priority 1
- Set the PTP Priority 2

- Configure the Source IP
- Configure the TTL for the PTP Packets

Configure the PTP Domain

To set the domain number to use for the clock, use the **ptp domain** command.

- **The ptp domain** command configures the domain 1 to use with a clock.

```
switch(config)# ptp domain 1
switch(config)#
```

Configure the Offset Hold Time

To set the PTP offset hold time, use the **ptp hold-ptp-time** command.

- The **ptp hold-ptp-time** command configures the PTP offset hold time to 600 seconds.

```
switch(config)# ptp hold-ptp-time 600
switch(config)#
```

Set the PTP Priority 1

To set the priority 1 value, use the **ptp priority1** command. Lower values take precedence.

- The **ptp priority1** command configures the priority 1 value of 120 to use when advertising the clock.

```
switch(config)# ptp priority1 120
switch(config)#
```

Set the PTP Priority 2

To set the priority 2 value for the clock, use the **ptp priority2** command.

- The **ptp priority2** command configures the priority 2 value of 128.

```
switch(config)# ptp priority2 128
switch(config)#
```

Configure the Source IP

To set the source IP address for all PTP packets, use the **ptp source ip command.**

- The **ptp source ip** command configures the source IP address of 10.0.2.1 for all PTP packets.

```
switch(config)# ptp source ip 10.0.2.1
switch(config)#
```

Configure the TTL for the PTP Packets

To set the time to live (TTL) of the PTP packets, use the **ptp ttl** command. Time to live is the maximum number of hops that a PTP packet may make.

- The **ptp ttl** command configures the time to live (TTL) of 64 hops for PTP packets.

```
switch(config)# ptp ttl 64
switch(config)#
```

5.3.2.3 Configuring PTP Interface Options

The following PTP interface commands are optional:

- Set the PTP Announcement Interval
- Set the PTP Timeout Interval

- Configure the PTP Delay Mechanism
- Set the Delay Request Interval
- Set the Peer Delay Request Interval
- Set the Peer Link Propagation Threshold
- Set the Interval for Sending Synchronization Messages
- Set the PTP Transport Type

Set the PTP Announcement Interval

To set the interval between PTP announcement messages before a timeout occurs, use the **ptp announce interval** command.

- The **ptp announce interval** command configures the interval between PTP announcement messages before a timeout occurs.

```
switch(config)# interface ethernet 5
switch(config-if-Et5)# ptp announce interval 1
switch(config-if-Et5)#
```

Set the PTP Timeout Interval

To set the time for sending timeout messages, use the **ptp announce timeout** command. The range is 2 to 10 seconds. The default is 3 (8 seconds).

- The **ptp announce timeout** command specifies the time for announcing timeout messages.

```
switch(config-if-Et5)# ptp announce timeout 5
switch(config-if-Et5)#
```

Configure the PTP Delay Mechanism

To set the delay in the boundary clock, use the **ptp delay-mechanism command.**

- The **ptp delay-mechanism** command configures the delay in boundary clock mode.

```
switch(config-if-Et5)# ptp delay-mechanism p2p
switch(config-if-Et5)#
```

Set the Delay Request Interval

To set the time for the slave devices to send delay request messages, use the **ptp delay-req interval** command.

- The **ptp delay-req interval** command sets the time the slave devices to send delay request messages to the master state to 3.

```
switch(config-if-Et5)# ptp delay-request interval 3
switch(config-if-Et5)#
```

Set the Peer Delay Request Interval

To set the minimum interval between the PTP peer delay-request messages, use the **ptp pdelay-req interval** command.

- The **ptp pdelay-req interval** command configures the interval between Precision Time Protocol (PTP) peer delay-request messages to 3.

```
switch(config-if-Et5)# ptp pdelay-request interval 3
switch(config-if-Et5)#
```

Set the Peer Link Propagation Threshold

To set the delay threshold for which the peer will be considered unable to run generalized Precision Time Protocol (gPTP), use the **ptp pdelay-neighbor-threshold** command.

- The **ptp pdelay-neighbor-threshold** command sets the link propagation delay threshold on Ethernet interface 5 to 200000 nanoseconds..

```
switch(config-if-Et5)# ptp pdelay-neighbor-threshold 200000
switch(config-if-Et5)#
```

Set the Interval for Sending Synchronization Messages

To set the interval for sending synchronization messages, use the **ptp sync interval** command.

- The **ptp sync interval** command configures the time for sending synchronization messages to 3.

```
switch(config-if-Et5)# ptp sync interval 3
switch(config-if-Et5)#
```

Set the PTP Transport Type

To set the PTP transport type, use the **ptp transport** command.

- The **ptp transport** command configures the PTP transport type for a specific interface.

```
switch(config-if-Et5)# ptp transport ipv4
switch(config-if-Et5)#
```

5.3.2.4 Viewing PTP Settings and Status

The following commands display the status of the switch PTP server connections:

- Show General PTP Information
- Show PTP Clock and Offset
- Show PTP Parent Information
- Show PTP Clock Properties
- Show PTP Information for all Interfaces
- Show PTP Interface Counters
- Show PTP Foreign Master
- Show PTP Source IP

Show General PTP Information

To display general Precision Time Protocol (PTP) information, use the **show ptp** command.

The **show ptp** command displays PTP summary and port status information.

```
switch#show ptp
PTP Mode: gptp - Generalized PTP Clock
Clock Identity: 2001:0DB8:73:ff:ff:26:fd:90
Grandmaster Clock Identity: 2001:0DB8:96:ff:fe:6c:ed:02
Number of slave ports: 1
Number of master ports: 6
Slave port: Ethernet33
Mean Path Delay (nanoseconds): 718
Steps Removed: 1
Neighbor Rate Ratio: 1.00000007883
Rate Ratio: 1.00000007883
Interface State    AS      Time Since Last    Neighbor    Mean Path   Residence
                   Capable Changed            Rate Ratio  Delay (ns)  Time (ms)
--------- -------- ------- ------------------ ----------- ----------- ---------
Et1       Disabled No      Never              1.0         0           0
Et2       Disabled No      Never              1.0         0           0
Et3       Disabled No      Never              1.0         0           0
Et4       Disabled No      Never              1.0         0           0
Et5       Disabled No      Never              1.0         0           0
Et6       Disabled No      Never              1.0         0           0
Et7       Master   Yes     0:21:08            1.00000009  420         0
             <-------OUTPUT OMITTED FROM EXAMPLE-------->
```

Show PTP Clock and Offset

The **show ptp clock** command displays the local PTP clock and offset.

```
switch#show ptp clock
PTP Mode: Boundary Clock
Clock Identity: 0x00:1c:73:ff:ff:1e:83:24
Clock Domain: 1
Number of PTP ports: 24
Priority1: 128
Priority2: 128
Clock Quality:
   Class: 248
   Accuracy: 0x30
   OffsetScaledLogVariance: 0xffff
Offset From Master: 0
Mean Path Delay: 0
Steps Removed: 0
switch#
```

Show PTP Parent Information

The **show ptp parent** command displays the PTP clock's parent and grandmaster identity and configuration.

```
switch# show ptp parent
Parent Clock:
Parent Clock Identity: 0x00:1c:73:ff:ff:00:72:40
Parent Port Number: 0
Parent IP Address: N/A
Observed Parent Offset (log variance): N/A
Observed Parent Clock Phase Change Rate: N/A

Grandmaster Clock:
Grandmaster Clock Identity: 0x00:1c:73:ff:ff:00:72:40
Grandmaster Clock Quality:
   Class: 248
   Accuracy: 0x30
   OffsetScaledLogVariance: 0xffff
   Priority1: 128
   Priority2: 128
switch#
```

Show PTP Clock Properties

The **show ptp time-property** command displays PTP clock properties.

```
switch# show ptp time-property
Current UTC offset valid: False
Current UTC offset: 0
Leap 59: False
Leap 61: False
Time Traceable: False
Frequency Traceable: False
PTP Timescale: False
Time Source: 0x0
switch#
```

Show PTP Information for all Interfaces

The **show ptp interface** command displays PTP information for specified interfaces.

```
switch# show ptp interface
Interface Ethernet1
PTP: Disabled
Port state: Disabled
Sync interval: 1.0 seconds
Announce interval: 2.0 seconds
Announce interval timeout multiplier: 3
Delay mechanism: end to end
Delay request message interval: 32.0 seconds
Transport mode: ipv4

Interface Ethernet5
PTP: Disabled
Port state: Disabled
Sync interval: 8.0 seconds
Announce interval: 2.0 seconds
Announce interval timeout multiplier: 5
Delay mechanism: peer to peer
Peer delay request message interval: 8.0 seconds
Peer Mean Path Delay: 0
Transport mode: ipv4
             <-------OUTPUT OMITTED FROM EXAMPLE-------->
switch#
```

Show PTP Interface Counters

The **show ptp interface counters** command displays PTP interface counters for specified interfaces.

```
switch# show ptp interface ethernet 5 counters
Interface Ethernet5
Announce messages sent: 0
Announce messages received: 0
Sync messages sent: 0
Sync messages received: 0
Follow up messages sent: 0
Follow up messages received: 0
Delay request messages sent: 0
Delay request messages received: 0
Delay response messages sent: 0
Delay response messages received: 0
Peer delay request messages sent: 0
Peer delay request messages received: 0
Peer delay response messages sent: 0
Peer delay response messages received: 0
Peer delay response follow up messages sent: 0
Peer delay response follow up messages received: 0
switch#
```

Show PTP Foreign Master

The **show ptp foreign-master-record** command displays information about foreign masters (PTP sources not designated as the switch's master from which the switch has received sync packets).

```
switch# show ptp clocks foreign-masters-record
No Foreign Master Records
switch#
```

Show PTP Source IP

The **show ptp source ip** command displays PTP IP source information.

```
switch#show ptp source ip
PTP source IP: 10.0.2.1
switch#
```

5.4 Managing Display Attributes

Display commands control the content of the banner and the command line prompt.

5.4.1 Banners

The switch can display two banners:

- **Login banner**: The login banner precedes the login prompt. One common use for a login banner is to warn against unauthorized network access attempts.
- **motd banner**: The message of the day (motd) banner is displayed after a user logs into the switch.

This output displays both banners in bold:

```
This is a login banner
switch login: john
Password:
Last login: Mon Jan 14 09:24:36 2013 from adobe-wrks.aristanetworks.com
This is an motd banner
switch>
```

These commands create the login and motd banner shown earlier in this section.

```
switch(config)#banner login
Enter TEXT message. Type 'EOF' on its own line to end.
This is a login banner
EOF
switch(config)#banner motd
Enter TEXT message. Type 'EOF' on its own line to end.
This is an motd banner
EOF
switch(config)#
```

To create a banner:

Step 1 Enter global configuration mode.

```
switch#config
switch(config)#
```

Step 2 Enter banner edit mode by typing the desired command:

- To create a login banner, type **banner login**.
- To create a motd banner, type **banner motd**.

The switch responds with instructions on entering the banner text.

```
switch(config)#banner login
Enter TEXT message. Type 'EOF' on its own line to end.
```

Step 3 Enter the banner text.

```
This is the first line of banner text.
This is the second line of banner text.
```

Step 4 Press **Enter** to place the cursor on a blank line after completing the banner text.

Step 5 Exit banner edit mode by typing EOF.

```
EOF
switch(config)#
```

5.4.2 Prompt

The prompt provides an entry point for EOS commands. The **prompt** command configures the contents of the prompt. The **no prompt** command returns the prompt to the default of %H%P.

Characters allowed in the prompt include A-Z, a-z, 0-9, and these punctuation marks:

! @ # $ % ^ & * ( ) - = + f g [ ] ; : < > , . ? / ~ n

The prompt supports these control sequences:

- %s – space character
- %t – tab character
- %% – percent character
- %H – host name
- %D – time and date
- %D{*f_char*} – time and date, format specified by the BSD **strftime (*f_char*)** time conversion function.
- %h – host name up to the first '.'
- %P – extended command mode
- %p – command mode
- %r – redundancy status on modular systems (has no effect on a fixed system)
- %R – extended redundancy status on modular systems – includes status and slot number (has no effect on a fixed system)

Example

- This command creates a prompt that displays **system 1** and the command mode.

```
host-name.dut103(config)#prompt system%s1%P
system 1(config) #
```

- This command creates a prompt that displays the command mode.

```
host-name.dut103(config)#prompt %p
(config)#
```

- These equivalent commands create the default prompt.

```
% prompt %H%P
host-name.dut103(config)#


% no prompt
host-name.dut103(config)#
```

5.5 Event Monitor

The event monitor writes system event records to local files for access by SQLite database commands.

5.5.1 Description

The event monitor receives notifications for changes to the mac table, route table, and arp table. These changes are logged to a fixed-size circular buffer. The size of this buffer is configurable, but it does not grow dynamically. Buffer contents can be stored to permanent files to increase the event monitor effective capacity. The permanent file size and the number of permanent files is configurable. The buffer is stored at a fixed location on the switch. The location of the permanent files is configurable and can be in any switch file directory, including flash (**/mnt/flash**).

Specific event monitor queries are available through CLI commands. For queries not available through specific commands, manual queries are supported through other CLI commands. When the user issues a query command, the relevant events from the circular buffer and permanent files are written to and accessed from a temporary SQLite database file. The database keeps a separate table for each logging type (mac, arp, route). When the monitor receives notification of a new event, the database file is deleted, then recreated.

5.5.2 Configuring the Event Monitor

Enabling the Event Monitor

The **event-monitor <log enable>** command enables the event monitor and specifies the types of events that are logged. The event monitor is an event logging service that records system events to a local database. The event monitor records these events:

- arp changes to the ARP table (IP address to MAC address mappings).
- IGMP snooping changes to the IGMP snooping table.
- mac changes to the MAC address table (MAC address to port mappings).
- mroute changes to the IP multicast routing table.
- route changes to the IP routing table.

By default, the event monitor is enabled and records each type of event. The **no event-monitor all** disables the event monitor. The **no event-monitor** command, followed by a log type parameter, disables event recording for the specified type.

Example

- This command disables the event monitor for all types of events.

```
switch(config)#no event-monitor all
```

- This command enables the event monitor for routing table changes.

```
switch(config)#event-monitor route
```

The **event-monitor clear** command removes the contents of the event monitor buffer. If event monitor backup is enabled, this command removes the contents from all event monitor backup files.

Example

- This command clears the contents of the event monitor buffer.

```
switch#event-monitor clear
switch(config)#
```

Configuring the Buffer

The **event-monitor buffer max-size** command specifies the size of the event monitor buffer. The event monitor buffer is a fixed-size circular data structure that receives event records from the event monitor. When event monitor backup is enabled, the buffer is copied to a backup file before each rollover. Buffer size ranges from 6 Kb to 50 Kb. The default size is 32 Kb.

Example

- This command configures a buffer size of 48 Kb.

```
switch(config)#event-monitor buffer max-size 48
switch(config)#
```

Configuring Permanent Files

The **event-monitor backup path** command enables storage of the event monitor buffer to permanent switch files and specifies the path/name of these files. The command references file location either from the flash drive root directory where the CLI operates (***/mnt/flash***) or from the switch root directory (*/*).

The event monitor buffer is circular – after the buffer is filled, new data replaces older data at the beginning of the buffer. The buffer is copied into a new backup file after each buffer writing cycle before the switch starts re-writing the buffer.

Example

- These commands configure the switch to store the event monitor buffer in sw-event.log, then display the new file in the flash directory.

```
switch(config)#event-monitor backup path sw-event.log
switch(config)#dir
Directory of flash:/

       -rwx   245761935           Jan 18 04:18  EOS-4.9.0.swi
       -rwx   245729161           Jan 17 06:57  EOS-4.9.0f.swi
       -rwx          25            Jan 5 08:59  boot-config
       -rwx          14           Jun 20  2011  boot-extensions
       -rwx        2749           Nov 22  2011  startup-config
       -rwx      418884           Jan 18 13:55  sw-event.log.0
       -rwx          13            Nov 9  2011  zerotouch-config

931745792 bytes total (190517248 bytes free)
switch(config)#
```

The **event-monitor backup max-size** command specifies the quantity of event monitor backup files the switch maintains. The switch appends a extension number to the file name when it creates a new file. After every 500 events, the switch deletes the oldest backup file if the file limit is exceeded.

Example

- These commands configure the switch to back up the event buffer to a series of files named sw-event.log. The switch can store a maximum of four files.

```
switch(config)#event-monitor backup path sw-event.log
switch(config)#event-monitor backup max-size 4
switch(config)#
```

The first five files that the switch creates to store event monitor buffer contents are:

sw-event.log.0
sw-event.log.1
sw-event.log.2
sw-event.log.3
sw-event.log.4

The switch deletes ***sw-event.log.0*** the first time it verifies the number of existing backup files after the creation of ***sw-event.log.4***.

5.5.3 Querying the Event Monitor

These CLI commands perform SQL-style queries on the event monitor database:

- The **show event-monitor arp** command displays ARP table events.
- The **show event-monitor mac** command displays MAC address table events.
- The **show event-monitor route** command displays routing table events.

Example

- This command displays all events triggered by MAC address table events.

```
switch#show event-monitor mac
% Writing 0 Arp, 0 Route, 1 Mac events to the database
2012-01-19 13:57:55|1|08:08:08:08:08:08|Ethernet1|configuredStaticMac|added|0
```

For other database queries, the **show event-monitor sqlite** command performs an SQL-style query on the database, using the statement specified in the command.

Example

- This command displays all entries from the route table.

```
switch#show event-monitor sqlite select * from route;
2012-01-19 13:53:01|16.16.16.0/24||||removed|0
2012-01-19 13:53:01|16.16.16.17/32||||removed|1
2012-01-19 13:53:01|16.16.16.18/32||||removed|2
2012-01-19 13:53:01|16.16.16.240/32||||removed|5
2012-01-19 13:53:01|16.16.16.0/32||||removed|6
2012-01-19 13:53:01|16.16.16.255/32||||removed|7
2012-01-19 13:53:01|192.168.1.0/24||||removed|8
2012-01-19 13:53:01|192.168.1.5/32||||removed|9
2012-01-19 13:53:01|192.168.1.6/32||||removed|10
```

5.5.4 Accessing Event Monitor Database Records

The **event-monitor interact** command replaces the CLI prompt with an SQLite prompt. The event monitor buffer and all backup logs are synchronized into a single SQLite file and loaded for access from the prompt.

- To access help from the SQLite prompt, enter ***.help***
- To exit SQLite and return to the CLI prompt, enter ***.quit*** or ***.exit***

The **event-monitor sync** command combines the event monitor buffer and all backup logs and synchronizes them into a single SQLite file. The data can be accessed through SQLite or by using the **show event-monitor** commands described above.

Examples

- This command replaces the EOS CLI prompt with an SQLite prompt.

```
switch#event-monitor interact
sqlite>
```

- This command exits SQLite and returns to the EOS CLI prompt.

```
sqlite> .quit
switch#
```

- This command synchronizes the buffer and backup logs into a single SQLite file.

```
switch(config)#event-monitor sync
switch(config)#
```

5.6 Switch Administration Commands

Switch Name Configuration Commands

- hostname
- ip domain-list
- ip domain lookup
- ip domain-name
- ip host
- ip name-server
- ipv6 host
- show hostname
- show hosts
- show ip domain-name
- show ip name-server

Banner Configuration Commands

- banner login
- banner motd
- show banner

Prompt Configuration Command

- prompt

Event Manager Commands

- no event-monitor
- event-monitor <log enable>
- event-monitor backup max-size
- event-monitor backup path
- event-monitor buffer max-size
- event-monitor clear
- event-monitor interact
- event-monitor sync
- show event-monitor arp
- show event-monitor mac
- show event-monitor route
- show event-monitor sqlite

Email Configuration Command

- email

System Clock Commands

- clock set
- clock timezone
- show clock

NTP Configuration Commands

- ntp authenticate
- ntp authentication-key
- ntp serve
- ntp serve all
- ntp server
- ntp source

- ntp trusted-key
- show ntp associations
- show ntp status

PTP Configuration Commands

- clear ptp interface counters
- ptp announce interval
- ptp announce timeout
- ptp delay-mechanism
- ptp delay-req interval
- ptp domain
- ptp enable
- ptp forward-v1
- ptp hold-ptp-time
- ptp mode
- ptp pdelay-neighbor-threshold
- ptp pdelay-req interval
- ptp priority1
- ptp priority2
- ptp source ip
- ptp sync interval
- ptp sync timeout
- ptp transport
- ptp ttl
- show ptp
- show ptp clock
- show ptp foreign-master-record
- show ptp interface
- show ptp interface counters
- show ptp parent
- show ptp source ip
- show ptp time-property

banner login

The **banner login** command configures a message that the switch displays before login and password prompts. The login banner is available on console, telnet, and ssh connections.

The **no banner login** and **default banner login** commands delete the login banner.

Command Mode

Global Configuration

Command Syntax

```
banner login
no banner login
default banner login
```

Parameters

- *banner_text* To configure the banner, enter a message when prompted. The message may span multiple lines. Banner text supports the following keywords:
  - **$(hostname)** displays the switch's host name.
- **EOF** To end the banner editing session, type EOF on its own line and press **enter**.

Examples

- These commands create a two-line login banner.

```
switch(config)#banner login
Enter TEXT message. Type 'EOF' on its own line to end.
This is a login banner for $(hostname).
Enter your login name at the prompt.
EOF
switch(config)#
```

This output displays the login banner.

```
This is a login banner for switch.
Enter your login name at the prompt.
switch login: john
Password:
Last login: Mon Jan 14 09:05:23 2013 from adobe-wrks.aristanetworks.com
switch>
```

banner motd

The **banner motd** command configures a “message of the day” (motd) that the switch displays after a user logs in. The motd banner is available on console, telnet, and ssh connections.

The **no banner motd** and **default banner motd** commands delete the motd banner.

Command Mode

Global Configuration

Command Syntax

```
banner motd
no banner motd
default banner motd
```

Parameters

- *banner_text* To configure the banner, enter a message when prompted. The message may span multiple lines. Banner text supports this keyword:
  - **$(hostname)** displays the switch’s host name.
- **EOF** To end the banner editing session, type EOF on its own line and press **enter**.

Examples

- These commands create an motd banner.

```
switch(config)#banner motd
Enter TEXT message. Type 'EOF' on its own line to end.
This is an motd banner for $(hostname)
EOF
switch(config)#
```

This output displays the motd banner.

```
switch login: john
Password:
Last login: Mon Jan 14 09:17:09 2013 from adobe-wrks.aristanetworks.com
This is an motd banner for Switch
switch>
```

clear ptp interface counters

The **clear ptp interface counters** command resets the Precision Time Protocol (PTP) packet counters.

Command Mode

Privileged EXEC

Command Syntax

```
clear ptp interface [INTERFACE_NAME] counters
```

Parameters

- ***INTERFACE_NAME*** Interface type and numbers. Options include:
  - <no parameter> Display information for all interfaces.
  - **ethernet** *e_range* Ethernet interface range specified by *e_range*.
  - **loopback** *l_range* Loopback interface specified by *l_range*.
  - **management** *m_range* Management interface range specified by *m_range*.
  - **port-channel** *p_range* Port-Channel Interface range specified by *p_range*.
  - **vlan** *v_range* VLAN interface range specified by *v_range*.
  - **vxlan** *vx_range* VXLAN interface range specified by *vx_range*.

  Valid parameter formats include number, number range, or comma-delimited list of numbers and ranges.

Example

- This command clears all PTP counters.

```
switch# clear ptp counters
switch#
```

clock set

The **clock set** command sets the system clock time and date. If the switch is configured with an NTP server, NTP time synchronizations override manually entered time settings.

Time entered by this command is local, as configured by the **clock timezone** command.

Command Mode

Privileged EXEC

Command Syntax

clock set *hh:mm:ss date*

Parameters

- *hh:mm:ss* is the current time (24-hour notation).
- *date* is the current date. Date formats include:
  - ***mm/dd/yy*** example: 05/15/2012
  - ***Month day year*** example: May 15 2012
  - ***day month year*** example: 15 May 2012

Example

- This command manually sets the switch time.

```
switch#clock set 08:15:24 14 Jan 2013
Mon Jan 14 08:15:25 2013
timezone is US/Central
```

clock timezone

The **clock timezone** command specifies the UTC offset that converts system time to local time. The switch uses local time for time displays and to time-stamp system logs and messages.

The **no clock timezone** and **default clock timezone** commands delete the **timezone** statement from ***running-config***, setting local time to UTC.

Command Mode

Global Configuration

Command Syntax

```
clock timezone zone_name
no clock timezone
default clock timezone
```

Parameters

- *zone_name* the time zone. Settings include a list of predefined time zone labels.

Examples

- This command configures the switch for the United States Central Time Zone.

```
switch(config)#clock timezone US/Central
switch(config)#show clock
Fri Jan 11 18:42:49 2013
timezone is US/Central
switch(config)#
```

- To view the predefined time zone labels, enter **clock timezone** with a question mark.

```
switch(config)#clock timezone ?
  Africa/Abidjan                    Africa/Accra
  Africa/Addis_Ababa                Africa/Algiers
  Africa/Asmara                     Africa/Asmera
  Africa/Bamako                     Africa/Bangui
             <-------OUTPUT OMITTED FROM EXAMPLE-------->
  W-SU                              W-SU timezone
  WET                               WET timezone
  Zulu                              Zulu timezone

switch(config)#clock timezone
```

- This command displays all time zone labels that start with **America**.

```
switch(config)#clock timezone AMERICA?
America/Adak                    America/Anchorage
America/Anguilla                America/Antigua
America/Araguaina               America/Argentina/Buenos_Aires
             <-------OUTPUT OMITTED FROM EXAMPLE-------->
America/Virgin                  America/Whitehorse
America/Winnipeg                America/Yakutat
America/Yellowknife

switch(config)#clock timezone AMERICA
```

email

The **email** command places the switch in email client configuration mode. If you configure a ***from-user*** and an outgoing SMTP server on the switch, you can then use an email address as an output modifier to a **show** command and receive the output as email.

Command Mode

Global Configuration

Command Syntax

```
email
```

Example

- This command places the switch in email client configuration mode.

```
switch(config)#email
switch(config)#
```

no event-monitor

The **no event-monitor** and **default event-monitor** commands remove the specified **event-monitor** configuration statements from ***running-config***, returning the switch to the specified default state.

- **no event-monitor <**with no parameters> restores all default setting states:
  - event monitor is enabled.
  - buffer backup is disabled.
- **no event-monitor backup** disables the backup.

To disable the event monitor, enter the **no event-monitor all** command (**event-monitor <log enable>**).

Command Mode

Global Configuration

Command Syntax

```
no event-monitor [PARAMETER]
default event-monitor [PARAMETER]
```

Parameters

- ***PARAMETER*** the event monitor property that is returned to the default state.
  - <no parameter> all event monitor properties.
  - **backup** event monitor buffer backup is disabled.

Example

- This command removes all event monitor configuration statements from ***running-config***.

```
switch(config)#no event-monitor
switch(config)#
```

event-monitor <log enable>

The **event-monitor <log enable>** command enables the event monitor and specifies the types of events that are logged. The event monitor is an event logging service that records system events to a local database. The event monitor records these events:

- arp changes to the ARP table (IP address to MAC address mappings).
- IGMP snooping changes to the IGMP snooping table.
- mac changes to the MAC address table (MAC address to port mappings).
- mroute changes to the IP multicast routing table.
- route changes to the IP routing table.

The database maintains a separate table for each event type.

By default, the event monitor is enabled and records each type of event.

- The **no event-monitor all** command disables the event monitor.
- The **no event-monitor** command, followed by a log type parameter, disables event recording for the specified type.
- The **event-monitor** and **default event-monitor** commands enable the specified event logging type by removing the corresponding **no event-monitor** command from ***running-config***.

The **no event-monitor** and **default event-monitor** commands, without a ***LOG_TYPE*** parameter, restore the default event monitor settings by deleting all event monitor related commands from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
event-monitor LOG_TYPE
no event-monitor LOG_TYPE
default event-monitor LOG_TYPE
```

Parameters

- ***LOG_TYPE*** specifies the event logging type. Options include:
  - **all** all event logging types.
  - **arp** changes to ARP table.
  - **igmpsnooping** changes to IGMP snooping table.
  - **mac** changes to MAC address table.
  - **mroute** changes to multicast routing table.
  - **route** changes to IP routing table.

Related Commands

- **no event-monitor**

Examples

- This command disables the event monitor for all types of events.

```
switch(config)#no event-monitor all
switch(config)#
```

- This command enables the event monitor for routing table changes.

```
switch(config)#event-monitor route
switch(config)#
```

event-monitor backup max-size

The **event-monitor backup max-size** command specifies the quantity of event monitor backup files the switch maintains. Values range from 1 to 200 files with a default of ten files.

The **event-monitor backup path** command specifies the path/name of these files. The switch appends an extension to the file name that tracks the creation order of backup files. When the quantity of files exceeds the configured limit, the switch deletes the oldest file.

The **no event-monitor backup max-size** and **default event-monitor backup max-size** command restores the default maximum number of backup files the switch can store to ten by removing the corresponding **event-monitor backup max-size** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
event-monitor backup max-size file_quantity
no event-monitor backup max-size
default event-monitor backup max-size
```

Parameters

- *file_quantity* maximum number of backup files. Value ranges from 1 to 200. Default is 10.

Examples

- These commands configure the switch to back up the event buffer to a series of files named sw-event.log. The switch can store a maximum of four files.

```
switch(config)#event-monitor backup path sw-event.log
switch(config)#event-monitor backup max-size 4
switch(config)#
```

The first five files that the switch creates to store event monitor buffer contents are:

sw-event.log.0
sw-event.log.1
sw-event.log.2
sw-event.log.3
sw-event.log.4

The switch deletes ***sw-event.log.0*** the first time it verifies the number of existing backup files after the creation of ***sw-event.log.4***.

event-monitor backup path

The **event-monitor backup path** command enables the storage of the event monitor buffer to switch files and specifies the path/name of these files. The command references the file location either from the flash drive root directory (/mnt/flash) where the CLI operates or from the switch root directory (/).

The event monitor buffer is circular – after the buffer is filled, new data is written to the beginning of the buffer, replacing old data. At the conclusion of each buffer writing cycle, it is copied into a new backup file before the switch starts re-writing the buffer. The switch appends a extension number to the file name when it creates a new file. After every 500 events, the switch deletes the oldest backup file if the file limit specified by the **event-monitor backup max-size** command is exceeded.

running-config can contain a maximum of one **event-monitor backup path** statement**.** Subsequent **event-monitor backup path** commands replace the existing statement in ***running-config***, changing the name of the file where event monitor backup files are stored.

The **no event-monitor backup path** and **default event-monitor backup path** commands disable the storage of the event monitor buffer to switch files by deleting the **event-monitor backup path** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
event-monitor backup path URL_FILE
no event-monitor backup path
default event-monitor backup path
```

Parameters

- ***URL_FILE*** path and file name of the backup file
  - *path_string* specified path is appended to /mnt/flash/
  - **file:** *path_string* specified path is appended to /
  - **flash:** *path_string* specified path is appended to /mnt/flash/

Examples

- These commands configure the switch to store the event monitor buffer in sw-event.log, then display the new file in the flash directory.

```
switch(config)#event-monitor backup path sw-event.log
switch(config)#dir
Directory of flash:/

       -rwx   245761935           Jan 18 04:18  EOS-4.9.0.swi
       -rwx   245729161           Jan 17 06:57  EOS-4.9.0f.swi
       -rwx          25            Jan 5 08:59  boot-config
       -rwx          14           Jun 20  2011  boot-extensions
       -rwx        2749           Nov 22  2011  startup-config
       -rwx      418884           Jan 18 13:55  sw-event.log.0
       -rwx          13            Nov 9  2011  zerotouch-config

931745792 bytes total (190517248 bytes free)
switch(config)#
```

event-monitor buffer max-size

The **event-monitor buffer max-size** command specifies the size of the event monitor buffer. The event monitor buffer is a fixed-size circular data structure that receives event records from the event monitor. When event monitor backup is enabled (**event-monitor backup path**), the buffer is copied to a backup file before each rollover.

Buffer size ranges from 6 Kb to 50 Kb. The default size is 32 Kb.

The **no event-monitor buffer max-size** and **default event-monitor buffer max-size** commands restore the default buffer size of 32 Kb by removing the **event-monitor buffer max-size** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
event-monitor buffer max-size buffer_size
no event-monitor buffer max-size
default event-monitor buffer max-size
```

Parameters

- *buffer_size* buffer capacity (Kb). Values range from 6 to 50. Default value is 32.

Example

- This command configures a buffer size of 48 Kb.

```
switch(config)#event-monitor buffer max-size 48
switch(config)#
```

event-monitor clear

The **event-monitor clear** command removes the contents of the event monitor buffer. If event monitor backup is enabled, this command removes the contents from all event monitor backup files.

Command Mode

Privileged EXEC

Command Syntax

```
event-monitor clear
```

Example

- This command clears the contents of the event monitor buffer.

```
switch#event-monitor clear
switch#
```

event-monitor interact

The **event-monitor interact** command replaces the CLI prompt with an SQLite prompt. The event monitor buffer and all backup logs are synchronized into a single SQLite file and loaded for access from the prompt.

- To access help from the SQLite prompt, enter ***.help***
- To exit SQLite and return to the CLI prompt, enter ***.quit*** or ***.exit***

Command Mode

Privileged EXEC

Command Syntax

```
event-monitor interact
```

Examples

- This command replaces the EOS CLI prompt with an SQLite prompt.

```
switch#event-monitor interact
sqlite>
```

- This command exits SQLite and returns to the EOS CLI prompt.

```
sqlite> .quit
switch#
```

event-monitor sync

The **event-monitor buffer sync** command combines the event monitor buffer and all backup logs and synchronizes them into a single SQLite file, which is stored at ***/var/log/eventMon.db***

Command Mode

Privileged EXEC

Command Syntax

```
event-monitor sync
```

Example

- This command synchronizes the buffer and backup logs into a single SQLite file.

```
switch(config)#event-monitor sync
switch(config)#
```

hostname

The **hostname** command assigns a text string as the switch's host name. The default host name is ***localhost***.

The prompt displays the host name when appropriately configured through the **prompt** command.

The **no hostname** and **default hostname** commands return the switch's host name to the default value of ***localhost***.

Command Mode

Global Configuration

Command Syntax

hostname *string*
no hostname
default hostname

Parameters

- *string* host name assigned to the switch.

Example

- This command assigns the string *main-host* as the switch's host name.

```
switch(config)#hostname main-host
main-host(config)#
```

The prompt was previously configured to display the host name.

ip domain-list

The **ip domain-list** command specifies a domain name to add to the IP domain list.

The **no ip domain-list** and **default ip domain-list** commands return the IP domain list to its default state, in which the switch selects source IP addresses for each DNS request from the specified VRF.

Command Mode

Global Configuration

Command Syntax

```
ip domain-list [IP_DOMAIN_NAME]
no ip domain-list [IP_DOMAIN_NAME]
default ip domain-list [IP_DOMAIN_NAME]
```

Parameters

- ***IP_DOMAIN_NAME*** specifies the IP domain name.

Examples

- This command specifies foo.com as the IP domain name to add to the IP domain list.

```
switch(config)#ip domain-list foo.com
switch(config)#
```

- This command removes foo.com and returns the IP domain list to its default state.

```
switch(config)#no ip domain-list foo.com
switch(config)#
```

ip domain lookup

The **ip domain lookup** command specifies the source interface for all DNS requests sent from the specified VRF.

The **no ip domain lookup** and **default ip domain lookup** commands return the switch to its default state, in which the switch selects source IP addresses for each DNS request from the specified VRF.

Command Mode

Global Configuration

Command Syntax

```
ip domain lookup [VRF_INSTANCE] source-interface INTF_NAME
no ip domain lookup [VRF_INSTANCE] source-interface
default ip domain lookup [VRF_INSTANCE] source-interface
```

Parameters

- ***VRF_INSTANCE*** specifies the VRF instance being modified.
  - <no parameter> changes are made to the default VRF.
  - **vrf** *vrf_name* changes are made to the specified VRF.
- ***INTF_NAME*** name of source interface to be used for DNS requests. Options include:
  - **ethernet** *e_num* Ethernet interface specified by *e_num*.
  - **loopback** *l_num* Loopback interface specified by *l_num*.
  - **management** *m_num* Management interface specified by *m_num*.
  - **port-channel** *p_num* Port-channel interface specified by *p_num*.
  - **vlan** *v_num* VLAN interface specified by *v_num*.

Examples

- This command specifies VLAN 5 as the source interface for DNS requests originating from the default VRF.

```
switch(config)#ip domain lookup source-interface Vlan5
switch(config)#
```

- This command specifies VLAN 10 as the source interface for DNS requests originating from VRF “purple.”

```
switch(config)#ip domain lookup vrf purple source-interface Vlan10
switch(config)#
```

ip domain-name

The **ip domain-name** command configures the switch's domain name. The switch uses this name to complete unqualified host names.

The **no ip domain-name** and **default ip domain-name** commands delete the domain name by removing the **ip domain-name** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
ip domain-name string
no ip domain-name
default ip domain-name
```

Parameters

- *string* domain name (text string)

Example

- This command configures *aristanetworks.com* as the switch's domain name.

```
switch(config)#ip domain-name aristanetworks.com
switch(config)#
```

ip host

The **ip host** command associates a hostname to an IPv4 address. This command supports local hostname resolution based on local hostname-IP address maps. Multiple hostnames can be mapped to an IP address. IPv4 and IPv6 addresses can be mapped to the same hostname (to map an IPv6 address to a hostname, use the **ipv6 host** command). The **show hosts** command displays the local hostname-IP address mappings.

The **no ip host** and **default ip host** commands removes hostname-IP address maps by deleting the corresponding **ip host** command from ***running-config***, as specified by command parameters:

- no parameters: command removes all hostname-IP address maps.
- ***hostname*** parameter: command removes all IP address maps for the specified hostname.
- ***hostname*** and ***IP address*** parameters: command removes specified hostname-IP address maps.

Command Mode

Global Configuration

Command Syntax

```
ip host hostname hostadd_1 [hostadd_2] ... [hostadd_X]
no ip host [hostname] [hostadd_1] [hostadd_2] [hostadd_X]
default ip host [hostname] [hostadd_1] [hostadd_2] [hostadd_X]
```

Parameters

- *hostname* hostname (text).
- *hostadd_N* IPv4 address associated with hostname (dotted decimal notation).

Related Commands

- **ipv6 host**
- **show hosts**

Examples

- This command associates the hostname ***test_lab*** with the IP addresses ***10.24.18.5*** and ***10.24.16.3.***

```
switch(config)#ip host test_lab 10.24.18.5 10.24.16.3
```

- This command removes all IP address maps for the hostname ***production_lab***.

```
switch(config)#no ip host production_lab
switch(config)#
```

ip name-server

The **ip name-server** command adds name server addresses to ***running_config***. The switch uses name servers for name and address resolution. The switch can be configured with up to three name servers. Although a command can specify multiple name server addresses, ***running_config*** stores each address in a separate statement. Name server addresses can be IPv4 and IPv6; each command can specify both address types.

Attempts to add a fourth server generate an error message. All name server addresses must be configured in the same VRF. When name servers were previously configured in a VRF, they must all be removed before adding new name server entries.

The **no ip name-server** and **default ip name-server** commands remove specified name servers from ***running_config***. Commands that do not list an address remove all name servers.

Command Mode

Global Configuration

Command Syntax

```
ip name-server [VRF_INSTANCE] SERVER_1 [SERVER_2] [SERVER_3]
no ip name-server [VRF_INSTANCE] [SERVER_1] [SERVER_2] [SERVER_3]
default ip name-server [VRF_INSTANCE] [SERVER_1] [SERVER_2] [SERVER_3]
```

Parameters

- ***VRF_INSTANCE*** specifies the VRF instance containing the addresses.
  - <no parameter> default VRF.
  - **vrf** *vrf_name* a user-defined VRF.
- ***SERVER_X*** IP address of the name server (dotted decimal notation). Options include:
  - *ipv4_addr* (A.B.C.D)
  - *ipv6_addr* (A:B:C:D:E:F:G:H)

  A command can contain both (IPv4 and IPv6) address types.

Guidelines

All configured name server addresses must come from the same VRF. To use a user defined VRF for connection to a name server, first remove any name servers configured in the default VRF.

Examples

- This command adds two name servers to the configuration.

```
switch(config)#ip name-server 172.0.14.21 3:4F21:1902::
switch(config)#
```

- This command attempts to add a name server when the configuration already lists three servers.

```
switch(config)#ip name-server 172.1.10.22
% Maximum number of nameservers reached. '172.1.10.22' not added
switch(config)#
```

ipv6 host

The **ipv6 host** command associates a hostname to an IPv6 address. This command supports local hostname resolution based on local hostname-IP address maps. Multiple hostnames can be mapped to an IPv6 address. IPv4 and IPv6 addresses can be mapped to the same hostname (to map IPv4 addresses to a hostname, use the **ip host** command). The **show hosts** command displays the local hostname-IP address mappings.

The **no ipv6 host** and **default ipv6 host** commands remove hostname-IP address maps by deleting the corresponding **ipv6 host** command from ***running-config***, as specified by command parameters:

- no parameters: command removes all hostname-IPv6 address maps.
- ***hostname*** parameter: command removes all IPv6 address maps for the specified hostname.
- ***hostname*** and ***IP address*** parameters: command removes specified hostname-IP address maps.

Command Mode

Global Configuration

Command Syntax

```
ipv6 host hostname hostadd_1 [hostadd_2] ... [hostadd_X]
no ipv6 host [hostname] [hostadd_1] [hostadd_2] [hostadd_X]
default ipv6 host [hostname] [hostadd_1] [hostadd_2] [hostadd_X]
```

Parameters

- *hostname* hostname (text).
- *hostadd_N* IPv6 addresses associated with hostname (dotted decimal notation).

Related Commands

- **ip host**
- **show hosts**

Example

- This command associates the hostname ***support_lab*** with the IPv6 address 2001:0DB8:73:ff:ff:26:fd:90.

```
switch(config)#ipv6 host support_lab 2001:0DB8:73:ff:ff:26:fd:90
switch(config)#
```

ntp authenticate

The **ntp authenticate** command enables the authentication of incoming NTP packets. When authentication is enabled, NTP packets will be used to synchronize time on the switch only if they include a trusted authentication key. Authentication keys are created on the switch using the **ntp authentication-key** command, and the **ntp trusted-key** command is used to specify which keys are trusted. NTP authentication is disabled by default.

The **no ntp authenticate** and **default ntp authenticate** commands disable NTP authentication on the switch by removing the corresponding **ntp authenticate** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
ntp authenticate
no ntp authenticate
default ntp authenticate
```

Examples

- This command enables NTP authentication on the switch.

```
switch(config)#ntp authenticate
switch(config)#
```

- This command disables NTP authentication on the switch.

```
switch(config)#no ntp authenticate
switch(config)#
```

ntp authentication-key

The **ntp authentication-key** command creates an authentication key for use in authenticating incoming NTP packets. For the key to be used in authentication:

- It must be configured as a trusted key using the **ntp trusted-key** command.
- NTP authentication must be enabled on the switch using the **ntp authenticate** command.
- The same key must be configured on the NTP server.

The **no ntp authentication-key** and **default ntp authentication-key** commands remove the specified authentication key by removing the corresponding **ntp authentication-key** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
ntp authentication-key key_id ENCRYPT_TYPE password_text
no ntp authentication-key key_id
default ntp authentication-key key_id
```

Parameters

- *key_id* key ID number. Value ranges from 1 to 65534.
- ***ENCRYPT_TYPE*** encryption method. Values include:
  - **md5** *key_text* is MD5 encrypted.
  - **sha1** *key_text* is SHA-1 encrypted.
- *password_text* the authentication-key password.

Example

- This command creates an NTP authentication key with ID 234 and password “timeSync” using MD5 encryption.

  ```
  switch(config)#ntp authentication-key 234 md5 timeSync
  ```

  Running-config stores the password as plain text.

- This command removes NTP authentication key 234.

  ```
  switch(config)#no ntp authentication-key 234
  ```

ntp serve

The **ntp serve** command configures the command mode interface to accept incoming NTP requests regardless of the global setting.

The **no ntp serve** command configures the command mode interface to refuse incoming NTP requests regardless of the global setting. The **default ntp serve** command configures the command mode interface to follow the global setting.

Using this command also causes the switch to re-synchronize with its upstream NTP server.

Command Modes

Interface-Ethernet Configuration
Interface-Loopback Configuration
Interface-Management Configuration
Interface-Port-channel Configuration
Interface-VLAN Configuration
Interface-VXLAN Configuration

Command Syntax

```
ntp serve
no ntp serve
          default ntp serve
```

Example

- These commands configure Ethernet interface 5 to accept incoming NTP requests regardless of global settings.

```
switch(config)#interface ethernet 5
switch(config-if-Et5)#ntp serve
switch(config-if-Et5)#
```

- These commands configure Ethernet interface 5 to deny incoming NTP requests regardless of global settings.

```
switch(config)#interface ethernet 5
switch(config-if-Et5)#no ntp serve
switch(config-if-Et5)#
```

- These commands configure Ethernet interface 5 to use global settings in responding to incoming NTP requests.

```
switch(config)#interface ethernet 5
switch(config-if-Et5)#default ntp serve
switch(config-if-Et5)#
```

ntp serve all

The **ntp serve all** command configures the switch to act as an NTP server by accepting incoming NTP requests.

Using this command also causes the switch to re-synchronize with its upstream NTP server.

Individual interfaces can be configured separately to accept or deny NTP requests by using the **ntp serve** command, and these settings override the global setting.

Command Mode

Global Configuration

Command Syntax

```
ntp serve all
no ntp serve all
default ntp serve all
```

Example

- This command configures the switch to accept incoming NTP requests.

```
switch(config)#ntp serve all
switch(config)#
```

- This command configures the switch to deny incoming NTP requests.

```
switch(config)#no ntp serve all
switch(config)#
```

ntp server

The **ntp server** command adds a Network Time Protocol (NTP) server to ***running-config***. If the command specifies a server that already exists in ***running-config***, it will modify the server settings. The switch synchronizes the system clock with an NTP server when ***running-config*** contains at least one valid NTP server.

The switch supports NTP versions 1 through 4. The default is version 4.

The ***prefer*** option specifies a preferred NTP server, which will be used as the NTP server if not discarded by NTP.

The **no ntp server** and **default ntp server** commands remove the specified NTP server from ***running-config***. To remove an NTP server configured in a user-defined VRF, include the VRF name in the **no ntp server** command.

Command Mode

Global Configuration

Command Syntax

```
ntp server [VRF_INSTANCE] SERVER_NAME
[PREFERENCE][NTP_VERSION][IP_SOURCE][burst] [iburst][AUTH_KEY][MAX_POLL_INT]
[MIN_POLL_INT]
no ntp [server [VRF_INSTANCE] SERVER_NAME]
default ntp [server [VRF_INSTANCE] SERVER_NAME]
```

All parameters except ***VRF_INSTANCE*** and ***SERVER_NAME*** can be placed in any order.

Parameters

- ***VRF_INSTANCE*** the VRF instance to be used for connection to the specified server.
  - <no parameter> connects using the default VRF.
  - **vrf** *vrf_name* connects using the specified user-defined VRF.
- ***SERVER_NAME*** NTP server location. Options include:
  - *IP address* in dotted decimal notation
  - an FQDN host name
- ***PREFERENCE*** indicates priority of this server when the switch selects a synchronizing server.
  - <no parameter> server has no special priority.
  - **prefer** server has priority when the switch selects a synchronizing server.
- ***NTP_VERSION*** specifies the NTP version. Settings include:
  - <no parameter> sets NTP version to 4 (default).
  - **version** *number*, where *number* ranges from 1 to 4.
- ***IP_SOURCE*** specifies the source interface for NTP updates for the specified NTP server. This option overrides global settings created by the **ntp source** command. Options include:
  - <no parameter> sets the source interface to the global default.
  - **source ethernet** *e_num* Ethernet interface specified by *e_num*.
  - **source loopback** *l_num* loopback interface specified by *l_num*.
  - **source management** *m_num* management interface specified by *m_num*.
  - **source port-channel** *p_num* port-channel interface specified by *p_num*.
  - **source vlan** *v_num* VLAN interface specified by *v_num*.

- **burst** indicates that when the NTP server is reached, the switch sends packets to the server in bursts of eight instead of the usual one. Recommended only for local servers. Off by default.
- **iburst** indicates that the switch sends packets to the server in bursts of eight instead of the usual one until the server is reached. Recommended for general use to speed synchronization. Off by default.
- ***AUTH_KEY*** the authentication key to use in authenticating NTP packets from the server.
  - <no parameter> no authentication key is specified.
  - **key <1 to 65534>** switch will use the specified key to authenticate NTP packets from the server.
- ***MAX_POLL_INT*** specifies the maximum polling interval for the server (as the base-2 logarithm of the interval in seconds). Settings include:
  - <no parameter> sets the maximum polling interval to 10 (1,024 seconds, the default).
  - **maxpoll** *number*, where *number* is the base-2 logarithm of the interval in seconds. Values range from 3 (8 seconds) to 17 (131,072 seconds, approximately 36 hours).
- ***MIN_POLL_INT*** specifies the minimum polling interval for the server (as the base-2 logarithm of the interval in seconds). Settings include:
  - <no parameter> sets the minimum polling interval to 6 (64 seconds, the default).
  - **minpoll** *number* where *number* is the base-2 logarithm of the interval in seconds. Values range from 3 (8 seconds) to 17 (131,072 seconds, approximately 36 hours).

Guidelines

To configure multiple parameters for a single server, include them all in a single **ntp server** command. Using the command again for the same server overwrites parameters previously configured in ***running-config***.

All NTP servers must use the same VRF. If no VRF is specified, the server is configured in the default VRF. To use a user-defined VRF for connection to an NTP server, first use the **no ntp server** command to remove any NTP servers configured in the default VRF.

When specifying a source interface, choose an interface in the same VRF as the server. If the source interface is not in the same VRF, the source data will be included in ***running-config*** but will not be added to NTP packets.

An NTP server may be configured using an invalid or inactive VRF, but the status of the NTP server will remain inactive until the VRF is active.

Examples

- This command configures the switch to update its time with the NTP server at address 172.16.0.23 and designates it as a preferred NTP server.

  ```
  switch(config)#ntp server 172.16.0.23 prefer
  ```

- This command configures the switch to update its time through an NTP server named *local-nettime*.

  ```
  switch(config)#ntp server local-nettime
  ```

- This command configures the switch to update its time through a version 3 NTP server.

  ```
  switch(config)#ntp server 171.18.1.22 version 3
  ```

- These commands reconfigure the switch to access the above NTP servers through VRF “magenta.”

```
switch(config)#no ntp server 172.16.0.23
switch(config)#no ntp server local-nettime
switch(config)#no ntp server 171.18.1.22
switch(config)#ntp server vrf magenta 172.16.0.23 prefer
switch(config)#ntp server vrf magenta local-nettime
switch(config)#ntp server vrf magenta 171.18.1.22 version 3
switch(config)#
```

ntp source

The **ntp source** command configures an interface as the source of NTP updates. That interface's IP address is then used as the source address for all NTP packets sent to all destinations unless a server-specific source interface has been specified using the **source** option of the **ntp server** command.

The **no ntp source** and **default ntp source** commands remove the **ntp source** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
ntp source [VRF_INSTANCE] INT_PORT
no ntp source
default ntp source
```

Parameters

- ***VRF_INSTANCE*** the VRF instance to be used for connection to the specified server.
  - <no parameter> connects using the default VRF.
  - **vrf** *vrf_name* connects using the specified user-defined VRF.
- ***INT_PORT*** the interface port that specifies the NTP source. Settings include:
  - **ethernet** *e_range* Ethernet interface list.
  - **loopback** *l_range* loopback interface list.
  - **management** *m_range* management interface list.
  - **port-channel** *c_range* port channel interface list.
  - **vlan** *v_range* VLAN interface list.

Examples

- This command configures VLAN interface 25 as the source of NTP update packets.

```
switch(config)#ntp source vlan 25
switch(config)#
```

- This command removes the NTP source command from the configuration.

```
switch(config)#no ntp source
switch(config)#
```

ntp trusted-key

The **ntp trusted-key** command specifies which authentication keys will be trusted for authentication of NTP packets. A packet with a trusted key will be used to update the local time if authenticated.

The **no ntp trusted-key** and **default ntp trusted-key** commands remove the specified authentication keys from the trusted key list by removing the corresponding **ntp trusted-key** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
ntp trusted-key key_list
no ntp trusted-key
default ntp trusted-key
```

Parameters

- ***key_list*** specified one or more keys. Formats include a number (1 to 65534), number range, or comma-delimited list of numbers and ranges.

Examples

- This command configures the switch to trust authentication keys 234 and 237 for authentication of NTP packets.

```
switch(config)#ntp trusted-key 234,237
switch(config)#
```

prompt

The **prompt** command specifies the contents of the CLI prompt. Characters allowed in the prompt include A-Z, a-z, 0-9, and these punctuation marks:

! @ # $ % ^ & * ( ) - = + f g [ ] ; : < > , . ? / ~ n

The prompt supports these control sequences:

- %s – space character
- %t – tab character
- %% – percent character
- %D – time and date
- %D{*f_char*} – time and date, format specified by the BSD **strftime (*f_char*)** time conversion function.
- %H – host name
- %h – host name up to the first '.'
- %P – extended command mode
- %p – command mode
- %r[1] – redundancy status on modular systems
- %R[2] – extended redundancy status on modular systems – includes status and slot number

Table 5-1 displays Command Mode and Extended Command Mode prompts for various modes.

Table 5-1 Command Mode Prompt Examples

Command Mode	Command Mode Prompt	Extended Command Mode Prompt
Exec	>	>
Privileged Exec	#	#
Global Configuration	(config)#	(config)#
Ethernet Interface Configuration	(config-if)#	(config-if-ET15)#
VLAN Interface Configuration	(config-if)#	(config-if-Vl24)#
Port Channel Interface Configuration	(config-if)#	(config-if-Po4)#
Management Interface Configuration	(config-if)#	(config-if-Ma1)
Access List Configuration	(config-acl)#	(config-acl-listname)#
OSPF Configuration	(config-router)#	(config-router-ospf)#
BGP Configuration	(config-router)#	(config-router-bgp)#

The **no prompt** and **default prompt** commands return the prompt to the default of %H%R%P.

Command Mode

Global Configuration

Command Syntax

```
prompt p_string
no prompt
default prompt
```

1. When logged into a fixed system or a supervisor on a modular system, this option has no effect.
2. When logged into a fixed system, this option has no effect.

Parameters

- *p_string* prompt text (character string). Elements include letters, numbers, and control sequences.

Examples

- This command creates a prompt that displays **system 1** and the command mode.

```
host-name.dut103(config)#prompt system%s1%P
system 1(config) #
```

- This command creates a prompt that displays the command mode.

```
host-name.dut103(config)#prompt %p
(config)#
```

- These equivalent commands create the default prompt.

```
% prompt %H%P
host-name.dut103(config)#


% no prompt
host-name.dut103(config)#
```

ptp announce interval

The **ptp announce interval** command configures the interval between PTP announcement messages before a timeout occurs on the configuration mode interface. The **no ptp announce interval command resets the timeout interval to its default of 1**.

Command Mode

Interface-Ethernet Configuration
Interface-Port Channel Configuration

Command Syntax

```
ptp announce interval log_interval
no ptp announce interval
default ptp announce interval
```

Parameters

- *log_interval* The number of log seconds between PTP announcement message (base 2 log (seconds)). Value ranges from 0 to 4; default value is 1.

Examples

- These commands set the interval between PTP announcements on interface Ethernet 5 to 2.

```
switch(config)# interface ethernet 5
switch(config-if-Et5)# ptp announce interval 1
switch(config-if-Et5)#
```

- These commands reset the PTP announcement interval on interface Ethernet 5 to the default value of 1.

```
switch(config)# interface ethernet 5
switch(config-if-Et5)# no ptp announce interval
switch(config-if-Et5)#
```

ptp announce timeout

The **ptp announce timeout** command specifies the time for announcing timeout messages. The range is 2 to 10 log seconds. The default is 3 (8 seconds). The **no ptp announce timeout** command disables the feature.

Command Mode

Interface-Ethernet Configuration
Interface-Port Channel Configuration

Command Syntax

```
ptp announce timeout log_interval
no ptp announce timeout
default ptp announce timeout
```

Parameters

- *log_interval* The range is 2 to 10 log seconds (base 2 log (seconds)). The default is 3 (8 seconds).

Examples

- These commands set the timeout interval for PTP announcements on interface Ethernet 5 to 5 log seconds.

```
switch(config)# interface ethernet 5
switch(config-if-Et5)# ptp announce timeout 5
switch(config-if-Et5)#
```

- These commands reset the PTP timeout interval on interface Ethernet 5 to the default value of 3 (8 seconds).

```
switch(config)# interface ethernet 5
switch(config-if-Et5)# no ptp announce timeout
switch(config-if-Et5)#
```

ptp delay-mechanism

The **ptp delay-mechanism** command configures the delay mechanism in boundary clock mode. The **no ptp delay-mechanism** command disables the feature.

Command Mode

Interface-Ethernet Configuration
Interface-Port Channel Configuration

Command Syntax

```
ptp delay-mechanism MECH_TYPE
no ptp delay-mechanism
default ptp delay-mechanism
```

Parameters

- ***MECH_TYPE*** The delay mechanism. Options include:
  - **e2e** The delay request or response mechanism used in the boundary clock mode.
  - **p2p** The peer-to-peer mechanism used in the boundary clock mode.

Examples

- This command sets the delay mechanism to p2p in the boundary clock mode.

```
switch(config)# interface ethernet 5
switch(config-if-Et5)# ptp delay-mechanism p2p
switch(config-if-Et5)#
```

- This command sets the delay mechanism to e2e in the boundary clock mode.

```
switch(config)# interface ethernet 5
switch(config-if-Et5)# ptp delay-mechanism e2e
switch(config-if-Et5)#
```

- This command removes the delay mechanism configuration from Ethernet 5.

```
switch(config)# interface ethernet 5
switch(config-if-Et5)# no ptp delay-mechanism e2e
switch(config-if-Et5)#
```

ptp delay-req interval

The **ptp delay-req interval** command specifies the time in log seconds recommended to the slave devices to send delay request messages. You must enable PTP on the switch first and configure the source IP address for PTP communication. The **no ptp delay-req interval** command resets the interval to its default of 5 (32 seconds).

Command Mode

Interface-Ethernet Configuration
Interface-Port Channel Configuration

Command Syntax

```
ptp delay-req interval log_interval
no ptp delay-req interval
default ptp delay-req interval
```

Parameters

- *log_interval* The range is -1 to 8 log seconds (base 2 log (seconds)). The default is 5 (32 seconds).

Examples

- These commands set the minimum interval allowed between PTP delay request messages on Ethernet interface 5 to 3 (8 seconds).

```
switch(config)# interface ethernet 5
switch(config-if-Et5)# ptp delay-request interval 3
switch(config-if-Et5)#
```

- These commands reset the minimum interval allowed between PTP delay-request messages to the default of 5 (32 seconds).

```
switch(config)# interface ethernet 5
switch(config-if-Et5)# no ptp delay-request interval
switch(config-if-Et5)#
```

ptp domain

The **ptp domain** command sets the domain number to use for the clock. The **no ptp domain command** disables the feature.

Command Mode

Global Configuration

Command Syntax

```
ptp domain domain_number
no ptp domain
default ptp domain
```

Parameters

- *domain_number* Value ranges from 0 to 255.

Examples

- This command shows how to configure domain 1 for use with a clock.

```
switch(config)# ptp domain 1
switch(config)#
```

- This command removes the configured domain 1 for use with a clock.

```
switch(config)# no ptp domain 1
switch(config)#
```

ptp enable

The **ptp enable** command enables PTP on the interface. The **no ptp enable** command disables PTP on the interface.

Command Mode

Interface-Ethernet Configuration
Interface-Port Channel Configuration

Command Syntax

```
ptp enable
no ptp enable
default ptp enable
```

Examples

- This command enables PTP on Ethernet interface 5.

```
switch(config)# interface ethernet 5
switch(config-if-Et5)# ptp enable
```

- This command disables PTP on Ethernet interface 5.

```
switch(config)# interface ethernet 5
switch(config-if-Et5)# no ptp enable
```

ptp forward-v1

The **ptp forward-v1** command configures the switch to forward Precision Time Protocol version packets as regular multicast traffic. By default, PTP v1 packets are trapped by the CPU, logged and discarded.

The **no ptp forward-v1** and **default ptp forward-v1** commands restore the default forwarding behavior by removing the corresponding **ptp forward-v1** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
ptp forward-v1
no ptp forward-v1
default ptp forward-v1
```

Examples

- This command configures the switch to forward PTP v1 packets as regular multicast traffic.

```
switch(config)#ptp forward-v1
switch(config)#
```

- This command configures the switch to log and discard PTP v1 packets.

```
switch(config)#no ptp forward-v1
switch(config)#
```

ptp hold-ptp-time

The **ptp hold-ptp-time** command configures the PTP offset hold time in seconds. The **no ptp hold-ptp-time** command disables the feature.

Command Mode

Global Configuration

Command Syntax

```
ptp hold-ptp-time offset
no ptp hold-ptp-time
default ptp hold-ptp-time
```

Parameters

- *offset* Value ranges from 0 to 86400.

Examples

- This command shows how to configure the PTP offset hold time.

```
switch(config)# ptp hold-ptp-time 600
switch(config)#
```

- This command removes the configured PTP offset hold time.

```
switch(config)# no ptp hold-ptp-time
switch(config)#
```

ptp mode

The **ptp mode** command configures the Precision Time Protocol (PTP) packet forwarding mode for the switch. The default **ptp mode** is ***disabled***.

The **no ptp mode** and **default ptp mode** commands return the forwarding mode to **disabled** by removing the **ptp mode** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
ptp mode MODE_NAME
no ptp mode
default ptp mode
```

Parameters

- ***MODE_NAME*** Options include:
  - **boundary**
  - **disabled**
  - **e2etransparent**
  - **p2ptransparent**
  - **gptp**

Examples

- This command configures the boundary mode for PTP.

```
switch(config)# ptp mode boundary
switch(config)#
```

- This command restores PTP to disabled mode.

```
switch(config)# no ptp mode
switch(config)#
```

ptp pdelay-neighbor-threshold

The **ptp pdelay-neighbor-threshold** command configures the propagation delay threshold above which the switch will consider the neighbor connected to this port to be incapable of participating in generalized Precision Time Protocol (gPTP).

The **no ptp pdelay-neighbor-threshold** and **default ptp pdelay-neighbor-threshold** commands restore the threshold to 100000 nanoseconds by removing the corresponding **ptp pdelay-neighbor-threshold** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration
Interface-Port Channel Configuration

Command Syntax

```
ptp pdelay-neighbor-threshold link_prop
no ptp pdelay-neighbor-threshold
default ptp pdelay-neighbor-threshold
```

Parameters

- *link_prop* Threshold in nanoseconds. Value ranges from 0 to 10000000000 (ten billion). Default is 100000.

Examples

- These commands set the link propagation delay threshold on Ethernet interface 5 to 200000 nanoseconds.

```
switch(config)# interface ethernet 5
switch(config-if-Et5)# ptp pdelay-neighbor-threshold 200000
switch(config-if-Et5)#
```

- These commands restore the link propagation delay threshold on Ethernet interface 5 to its default value of 100000 nanoseconds.

```
switch(config)# interface ethernet 5
switch(config-if-Et5)# no ptp pdelay-neighbor-threshold
switch(config-if-Et5)#
```

ptp pdelay-req interval

The **ptp pdelay-req interval** command configures the interval between Precision Time Protocol peer delay-request messages. The **no ptp pdelay-req interval** command removes the configuration.

Command Mode

Interface-Ethernet Configuration
Interface-Port Channel Configuration

Command Syntax

```
ptp pdelay-req interval log_interval
no ptp pdelay-req interval
default ptp pdelay-req interval
```

Parameters

- *log_interval* The log interval in seconds (base 2 log (seconds)). Value ranges from 0 to 5.

Examples

- This command shows how to configure the interval allowed between PTP peer delay request messages on interface Ethernet 5.

```
switch(config)# interface ethernet 5
switch(config-if-Et5)# ptp pdelay-request interval 3
switch(config-if-Et5)#
```

- This command removes the configure the interval allowed between PTP peer delay request messages on interface Ethernet 5.

```
switch(config)# interface ethernet 5
switch(config-if-Et5)# no ptp pdelay-request interval
switch(config-if-Et5)#
```

ptp priority1

The **ptp priority1** command configures the priority 1 value for advertising the switch's PTP clock. Priority 1 is the most significant of the six factors used by devices in the selection of a master clock. Lower values indicate higher priority.

The **no ptp priority1** and **default ptp priority1** commands restore the priority 1 default setting of 128.

Command Mode

Global Configuration

Command Syntax

```
ptp priority1 priority_rate
no ptp priority1
default ptp priority1
```

Parameters

- *priority_rate* Value ranges from 0 to 255. Default is 128.

Examples

- This command sets the priority 1 level for the switch's PTP clock to 120.

```
switch(config)# ptp priority1 120
switch(config)#
```

- This command restores the default priority 1 level of 128.

```
switch(config)# no ptp priority1
switch(config)#
```

ptp priority2

The **ptp priority2** command sets the priority 2 value for the clock. The range is from 0 to 255. Priority 2 is the fifth most significant of the six factors used by devices in the selection of a master clock. Lower values indicate higher priority.

The **no ptp priority2** and **default ptp priority2** commands restore the priority 2 default setting of 128.

Command Mode

Global Configuration

Command Syntax

```
ptp priority2 priority_rate
no ptp priority2
default ptp priority2
```

Parameters

- *priority_rate* Specifies the priority 2 level for the PTP clock. Value ranges from 0 to 255; default value is 128.

Examples

- This command sets the priority 2 level for the switch's PTP clock to 120.

```
switch(config)# ptp priority2 120
switch(config)#
```

- This command restores the default priority 2 level of 128.

```
switch(config)# no ptp priority2
switch(config)#
```

ptp source ip

The **ptp source ip** command configures the source IP address for all PTP packets. The IP address can be in IPv4 format. The **no ptp source ip** command removes this configuration.

Command Mode

Global Configuration

Command Syntax

```
ptp source ip ipv4_addr
no ptp source ip
default ptp source ip
```

Parameters

- *ipv4_addr* IPv4 address

Examples

- This command configures the source IP address **10.0.2.1** for all PTP packets.

```
switch(config)# ptp source ip 10.0.2.1
switch(config)#
```

- This command removes the source IP address **10.0.2.1** for all PTP packets.

```
switch(config)# no ptp source ip
switch(config)#
```

ptp sync interval

The **ptp sync interval** command configures the time for sending synchronization messages. The command configures the interval by specifying its $\log_2$ value. Parameter value ranges from -1 (1/2 second) to 3 (eight seconds). The default value is 0 (one second).

The **no ptp sync interval** and **default ptp sync interval** commands restore the default sync interval setting of 0 by removing the corresponding **ptp ptp sync interval** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration
Interface-Port Channel Configuration

Command Syntax

```
ptp sync interval log_interval
no ptp sync interval
default ptp sync interval
```

Parameters

- *log_interval* The interval between PTP synchronization messages sent from the master to the slave (base 2 log(seconds)). Values range from -1 to 3; default value is 0 (1 second).

Examples

- These commands set the interval for PTP synchronization messages on Ethernet interface 5 to 3 (8 seconds).

```
switch(config)# interface ethernet 5
switch(config-if-Et5)# ptp sync interval 3
switch(config-if-Et5)#
```

- These commands restore the interval for PTP synchronization messages on Ethernet interface 5 to its default of 0 (1 second).

```
switch(config)# interface ethernet 5
switch(config-if-Et5)# no ptp sync interval
switch(config-if-Et5)#
```

ptp sync timeout

A PTP synchronization timeout occurs if a sync message is not received for a specified period of time, calculated as a multiple of the PTP sync interval. The **ptp sync timeout** command configures the sync timeout multiplier. The range is 2 to 255, with a default of 20 (20 times the sync interval). To configure the sync interval, use the **ptp sync interval** command.

The **no ptp sync timeout** and **default ptp sync timeout** commands restore the PTP sync timeout multiplier to its default value of 20.

Command Mode

Interface-Ethernet Configuration

Command Syntax

```
ptp sync timeout interval_multiplier
no ptp sync timeout
default ptp sync timeout
```

Parameters

- *interval_multiplier* The number of sync intervals that must pass without the configuration mode interface receiving a PTP sync message before a timeout occurs. Value ranges from 2 to 255. Default value is 20.

Examples

- These commands configure the sync timeout on Ethernet interface 5 to ten times the configured sync interval.

```
switch(config)# interface ethernet 5
switch(config-if-Et5)# ptp sync timeout 10
switch(config-if-Et5)#
```

ptp transport

The **ptp transport** command configures the PTP transport type for a specific interface. Any values set in interface PTP configuration mode override the settings in the PTP configuration profile associated with the interface. The **no ptp transport** command removes the stetting from the running configuration.

Command Mode

Interface-Ethernet Configuration
Interface-Port Channel Configuration

Command Syntax

```
ptp transport TRANSPORT_TYPE
no ptp transport
default ptp transport
```

Parameters

- ***TRANSPORT_TYPE*** The transport mode in boundary clock mode. Options include:
  - **ipv4** The IPv4 address used as the transport type on the interface.
  - **layer2** The Layer 2 protocol used as the transport type on the interface.

Examples

- This command overrides the transport type in the profile and sets it to be IPv4 for the interface.

```
switch(config)# interface ethernet 5
switch(config-if-Et5)# ptp transport ipv4
switch(config-if-Et5)#
```

- This command removes the interval for PTP synchronization messages on interface Ethernet 5.

```
switch(config)# interface ethernet 5
switch(config-if-Et5)# no ptp transport
switch(config-if-Et5)#
```

ptp ttl

The **ptp ttl** command configures the time-to-live value of the PTP packets. The **no ptp ttl** resets the time to live to the default value of 1 hop by removing the **ptp ttl** command from the running configuration.

Command Mode

Global Configuration

Command Syntax

```
ptp ttl hop_count
no ptp ttl
default ptp ttl
```

Parameters

- *hop_count* The time to live measured in hops. Value ranges from 1 to 255, default is 1.

Example

- This command sets the time to live of the PTP packets to 60 hops.

```
switch(config)# ptp ttl 60
switch(config)#
```

- This command resets the time to live of the PTP packets to the default value of 1 hop.

```
switch(config)# no ptp ttl
switch(config)#
```

show banner

The **show banner** command displays the specified banner.

Command Mode

Privileged EXEC

Command Syntax

show banner ***BANNER_TYPE***

Parameters

- ***BANNER_TYPE*** banner that the command displays. Options include
  - **login** command displays login banner.
  - **motd** command displays message of the day banner.

Example

- These commands configure and display the message of the day banner.

```
switch(config)#banner motd
Enter TEXT message. Type 'EOF' on its own line to end.
This is an motd banner for $(hostname)
EOF
switch(config)#show banner motd
This is an motd banner for $(hostname)

switch(config)#
```

show clock

The **show clock** command displays the current system clock time and configured time zone. The switch uses the system clock for system log messages and debugging traces.

Command Mode

EXEC

Command Syntax

show clock

Example

- This command displays the current system clock time and configured time zone.

```
switch>show clock
Wed Nov  2 10:29:32 2011
timezone is America/Los_Angeles
switch>
```

show event-monitor arp

The **show event-monitor arp** command performs an SQL-style query on the event monitor database and displays ARP table events as specified by command parameters. The event monitor buffer and all backup logs are synchronized into a single SQLite file.

Command Mode

Privileged EXEC

Command Syntax

```
show event-monitor arp [GROUP] [MESSAGES] [INTERFACE] [IP] [MAC] [TIME]
```

Optional parameters can be placed in any order.

Parameters

- ***GROUP*** used with aggregate functions to group results. Analogous to SQL ***group by*** command.
  - <no parameter> results are not grouped.
  - **group-by** *ip* results are grouped by IP address.
  - **group-by** *mac* results are grouped by MAC address.
- ***MESSAGES*** number of message returned from query. Analogous to SQL ***limit*** command.
  - <no parameter> result-set size is not limited.
  - **limit** *msg_quantity* number of results that are displayed. Values range from 1 to 15,000.
- ***INTERFACE*** restricts result-set to events that include specified interface (SQL Like command).
  - <no parameter> result-set not restricted by interface.
  - **match-interface ethernet** *e_range* Ethernet interface list.
  - **match-interface loopback** *l_range* loopback interface list.
  - **match-interface management** *m_range* management interface list.
  - **match-interface port-channel** *c_range* port channel interface list.
  - **match-interface vlan** *v_range* VLAN interface list.
- ***IP*** restricts result-set to events that include specified IP address (SQL Like command).
  - <no parameter> result-set not restricted to specific IP addresses.
  - **match-ip** *ip_address_rex* IP address, as represented by regular expression.
- ***MAC*** restricts result-set to events that include specified MAC address (SQL Like command).
  - <no parameter> result-set not restricted to specific MAC addresses.
  - **match-mac** *mac_address_rex* MAC address, as represented by regular expression.
- ***TIME*** restricts result-set to events generated during specified period.
  - <no parameter> result-set not restricted by time of event.
  - **match-time last-minute** includes events generated during last minute.
  - **match-time last-day** includes events generated during last day.
  - **match-time last-hour** includes events generated during last hour.
  - **match-time last-week** includes events generated during last week.

Example

- This command displays ARP table events listed in the event monitor database.

```
switch#show event-monitor arp
% Writing 220017 Arp, 234204 Route, 1732559 Mac events to the database
2012-11-06 12:36:10|10.33.6.159|Vlan1417|00:00:00:dc:cc:0d|0|added|2186271
2012-11-06 12:38:20|10.33.7.150|Vlan1417|00:00:00:f7:e2:5f|0|added|2186292
2012-11-06 12:38:34|10.33.6.62|Vlan1417|00:00:00:01:c2:ac|0|added|2186295
2012-11-06 12:39:13|10.33.7.162|Vlan1417|00:00:00:45:c2:79|0|added|2186299
2012-11-06 12:39:50|10.33.12.54|Vlan1417|||removed|2186303
2012-11-06 12:39:51|10.33.6.218|Vlan1417|00:00:00:e9:36:46|0|added|2186305
2012-11-06 12:40:00|10.33.6.140|Vlan1417|00:00:00:4a:36:c3|0|added|2186308
2012-11-06 12:40:02|10.33.6.239|Vlan1417|00:00:00:5b:a7:21|0|added|2186312
2012-11-06 12:41:16|10.33.7.11|Vlan1417|00:00:00:3f:94:59|0|added|2186320
2012-11-06 12:41:50|10.33.7.60|Vlan1417|00:00:00:1f:3c:8e|0|added|2186346
2012-11-06 12:43:34|10.33.7.81|Vlan1417|00:00:00:e3:0d:9c|0|added|2186762
2012-11-06 12:43:42|10.33.6.214|Vlan1417|00:00:00:7b:09:7d|0|added|2186765
2012-11-06 12:43:59|10.33.7.149|Vlan1417|00:00:00:8d:a6:d8|0|added|2186768
switch#
```

show event-monitor mac

The **show event-monitor mac** command performs an SQL-style query on the event monitor database and displays MAC address table events as specified by command parameters. The event monitor buffer and all backup logs are synchronized into a single SQLite file.

Command Mode

Privileged EXEC

Command Syntax

show event-monitor mac [GROUP] [MESSAGES] [INTERFACE] [MAC] [TIME]

Optional parameters can be placed in any order.

Parameters

- ***GROUP*** used with aggregate functions to group results. Analogous to SQL ***group by*** command.
  - <no parameter> results are not grouped.
  - **group-by** *interface* results are grouped by interface.
  - **group-by** *mac* results are grouped by MAC address.
- ***MESSAGES*** number of message returned from query. Analogous to SQL ***limit*** command.
  - <no parameter> result-set size is not limited.
  - **limit** *msg_quantity* number of results that are displayed. Values range from 1 to 15,000.
- ***INTERFACE*** restricts result-set to events that include specified interface (SQL Like command).
  - <no parameter> result-set not restricted by interface.
  - **match-interface ethernet** *e_range* Ethernet interface list.
  - **match-interface loopback** *l_range* loopback interface list.
  - **match-interface management** *m_range* management interface list.
  - **match-interface port-channel** *c_range* port channel interface list.
  - **match-interface vlan** *v_range* VLAN interface list.
- ***MAC*** restricts result-set to events that include specified MAC address (SQL Like command).
  - <no parameter> result-set not restricted to specific MAC addresses.
  - **match-mac** *mac_address_rex* MAC address, as represented by regular expression.
- ***TIME*** restricts result-set to events with specified period.
  - <no parameter> result-set not restricted by time of event.
  - **match-time last-minute** includes events generated during last minute.
  - **match-time last-day** includes events generated during last day.
  - **match-time last-hour** includes events generated during last hour.
  - **match-time last-week** includes events generated during last week.

Examples

- This command displays all events triggered by MAC address table events.

```
switch#show event-monitor mac
% Writing 0 Arp, 0 Route, 1 Mac events to the database
2012-01-19 13:57:55|1|08:08:08:08:08:08|Ethernet1|configuredStaticMac|added|0
```

- This command displays events triggered by MAC address table changes.

```
switch#show event-monitor mac match-mac 08:08:08:%
2012-01-19 13:57:55|1|08:08:08:08:08:08|Ethernet1|configuredStaticMac|added|0
```

show event-monitor route

The **show event-monitor route** command performs an SQL-style query on the event monitor database and displays routing table events as specified by command parameters. The event monitor buffer and all backup logs are synchronized into a single SQLite file.

Command Mode

Privileged EXEC

Command Syntax

```
show event-monitor route [GROUP] [MESSAGES] [IP] [TIME]
```

Optional parameters can be placed in any order.

Parameters

- ***GROUP*** used with aggregate functions to group results. Analogous to SQL ***group by*** command.
  - <no parameter> results are not grouped.
  - **group-by** *ip* results are grouped by IP address.
- ***MESSAGES*** number of message returned from query. Analogous to SQL ***limit*** command.
  - <no parameter> result-set size is not limited.
  - **limit** *msg_quantity* number of results that are displayed. Values range from 1 to 15,000.
- ***INTERFACE*** restricts result-set to events that include specified interface (SQL Like command).
  - <no parameter> result-set not restricted by interface.
  - **match-interface ethernet** *e_range* Ethernet interface list.
  - **match-interface loopback** *l_range* loopback interface list.
  - **match-interface management** *m_range* management interface list.
  - **match-interface port-channel** *c_range* port channel interface list.
  - **match-interface vlan** *v_range* VLAN interface list.
- ***IP*** restricts result-set to events that include specified IP address (SQL Like command).
  - <no parameter> result-set not restricted to specific IP addresses.
  - **match-ip** *ip_address_rex* IP address, as represented by regular expression.
- ***TIME*** restricts result-set to events with specified period.
  - <no parameter> result-set not restricted by time of event.
  - **match-time last-minute** includes events generated during last minute.
  - **match-time last-day** includes events generated during last day.
  - **match-time last-hour** includes events generated during last hour.
  - **match-time last-week** includes events generated during last week.

Example

- This command displays 10 routing table events listed in the event monitor database.

```
switch#show event-monitor route limit 10
% Writing 0 Arp, 2 Route, 0 Mac events to the database
2012-11-07 12:48:02|10.44.54.0/23|ospfAseE2|30|110|changed|2186957
2012-11-07 12:48:02|10.44.254.172/30|ospfAseE2|20|110|added|2186958
2012-11-07 12:48:02|10.44.254.112/30|ospfAseE2|30|110|changed|2186959
2012-11-07 12:48:02|10.44.48.0/23|ospfAseE2|30|110|changed|2186960
2012-11-07 12:48:02|10.52.0.35/32|ospfAseE2|30|110|changed|2186961
2012-11-07 12:48:02|10.44.50.0/23|ospfAseE2|30|110|changed|2186962
2012-11-07 12:48:02|10.44.254.172/30||||removed|2186963
2012-11-07 12:48:07|10.44.254.148/30|ospfInterArea|50|110|changed|2186964
2012-11-07 12:48:07|10.44.32.0/23|ospfInterArea|50|110|changed|2186965
2012-11-07 12:48:07|10.44.254.128/30|ospfInterArea|40|110|changed|2186966
switch#
```

show event-monitor sqlite

The **show event-monitor sqlite** command performs an SQL-style query on the event monitor database, using the statement specified in the command.

Command Mode

Privileged EXEC

Command Syntax

show event-monitor sqlite *statement*

Parameters

- *statement* SQLite statement.

Example

- This command displays all entries from the route table.

```
switch#show event-monitor sqlite select * from route;
2012-01-19 13:53:01|16.16.16.0/24||||removed|0
2012-01-19 13:53:01|16.16.16.17/32||||removed|1
2012-01-19 13:53:01|16.16.16.18/32||||removed|2
2012-01-19 13:53:01|16.16.16.240/32||||removed|5
2012-01-19 13:53:01|16.16.16.0/32||||removed|6
2012-01-19 13:53:01|16.16.16.255/32||||removed|7
2012-01-19 13:53:01|192.168.1.0/24||||removed|8
2012-01-19 13:53:01|192.168.1.5/32||||removed|9
2012-01-19 13:53:01|192.168.1.6/32||||removed|10
switch#
```

show hostname

The **show hostname** command displays the hostname and the fully qualified domain name (FQDN) of the switch.

Command Mode

EXEC

Command Syntax

show hostname

Example

- This command displays the hostname and FQDN of the switch.

```
switch>show hostname
Hostname: switch_1
FQDN:     switch_1.aristanetworks.com

switch>
```

show hosts

The **show hosts** command displays the default domain name, name lookup service style, a list of name server hosts, and the static hostname-IP address maps.

Command Mode

EXEC

Command Syntax

show hosts

Example

- This command displays the switch's IP domain name:

```
switch>show hosts

Default domain is: aristanetworks.com
Name/address lookup uses domain service
Name servers are: 172.22.22.40, 172.22.22.10

Static Mappings:

Hostname                        IP     Addresses

TEST_LAB                       IPV4    10.24.18.6
PRODUCTION_LAB                 IPV4    10.24.18.7
SUPPORT_LAB                    IPV6    2001:0DB8:73:ff:ff:26:fd:90
switch>
```

show ip domain-name

The **show ip domain-name** command displays the switch's IP domain name that is configured with the ip domain name command.

Command Mode

EXEC

Command Syntax

show ip domain-name

Example

- This command displays the switch's IP domain name:

```
switch>show ip domain-name
aristanetworks.com
switch>
```

show ip name-server

The **ip name-server** command displays the ip addresses of name-servers in ***running-config***. The name servers are configured by the **ip name-server** command.

Command Mode

EXEC

Command Syntax

show ip name-server

Example

- This command displays the IP address of name servers that the switch is configured to access.

```
switch>show ip name-server
172.22.22.10
172.22.22.40
switch>
```

show ntp associations

The **show ntp associations** command displays the status of connections to NTP servers.

Command Mode

EXEC

Command Syntax

show ntp associations

Display Values

- st (stratum): number of steps between the switch and the reference clock.
- t (transmission type): u – unicast; b – broadcast; l – local.
- when: interval since reception of last packet (seconds unless unit is provided).
- poll: interval between NTP poll packets. Maximum (1024) reached as server and client syncs.
- reach: octal number that displays status of last eight NTP messages (377 - all messages received).
- delay: round trip delay of packets to selected reference clock.
- offset: difference between local clock and reference clock.
- jitter: maximum error of local clock relative to reference clock.

Example

- This command displays the status of the switch's NTP associations.

```
switch>show ntp associations
     remote           refid      st t when poll reach   delay   offset
jitter
==============================================================================
=========
 172.1.1.1        .INIT.         16 u    - 1024    0    0.000    0.000
0.000
 moose.aristanet 192.187.233.4   2 u     9   64  377    0.118  9440498
0.017
 172.17.2.6     .INIT.         16 u    - 1024   0    0.000   0.000   0.000
*LOCAL(0)       .LOCL.         10 l   41   64  377    0.000   0.000   0.000
```

show ntp status

The **show ntp status** command displays the status of NTP on the switch. If the switch clock is not synchronized to an NTP server, the status reads “unsynchronised” and shows the server polling interval. If the clock is synchronized to an NTP server, the status shows the reference ID and stratum of the server, the precision of the synchronization, and the polling interval.

Important! As specified in RFC5905, for servers with IPv4 addresses the reference ID is the four-octet IPv4 address, but for servers with IPv6 addresses the reference ID is the first four octets of the MD5 hash of the IPv6 address.

Command Mode

EXEC

Command Syntax

show ntp status

Example

- This command displays the switch’s NTP status.

```
switch>show ntp status
synchronised to NTP server (172.16.1.50) at stratum 4
   time correct to within 77 ms
   polling server every 1024 s
switch>
```

show ptp

The **show ptp** command displays summary Precision Time Protocol (PTP) information and PTP status of switch ports.

Command Mode

EXEC

Command Syntax

show ptp

Example

- This command displays summary PTP information.

```
switch#show ptp
PTP Mode: gptp - Generalized PTP Clock
Clock Identity: 2001:0DB8:73:ff:ff:26:fd:90
Grandmaster Clock Identity: 2001:0DB8:96:ff:fe:6c:ed:02
Number of slave ports: 1
Number of master ports: 6
Slave port: Ethernet33
Mean Path Delay (nanoseconds): 718
Steps Removed: 1
Neighbor Rate Ratio: 1.00000007883
Rate Ratio: 1.00000007883
Interface State    AS      Time Since Last    Neighbor    Mean Path   Residence
                   Capable Changed            Rate Ratio  Delay (ns)  Time (ms)
--------- -------- ------- ------------------ ----------- ----------- ---------
Et1       Disabled No      Never              1.0         0           0
Et2       Disabled No      Never              1.0         0           0
Et3       Disabled No      Never              1.0         0           0
Et4       Disabled No      Never              1.0         0           0
Et5       Disabled No      Never              1.0         0           0
Et6       Disabled No      Never              1.0         0           0
Et7       Master   Yes     0:21:08            1.00000009  420         0
            <-------OUTPUT OMITTED FROM EXAMPLE-------->
```

show ptp clock

The **show ptp clock** command displays the Precision Time Protocol (PTP) clock information.

Command Mode

EXEC

Command Syntax

show ptp clock

Example

- This command shows how to display the PTP local clock and offset.

```
switch#show ptp clock
PTP Mode: Boundary Clock
Clock Identity: 0x00:1c:73:ff:ff:1e:83:24
Clock Domain: 1
Number of PTP ports: 24
Priority1: 128
Priority2: 128
Clock Quality:
   Class: 248
   Accuracy: 0x30
   OffsetScaledLogVariance: 0xffff
Offset From Master: 0
Mean Path Delay: 0
Steps Removed: 0
switch#
```

show ptp foreign-master-record

The **show ptp foreign-master-record** command displays information about foreign masters (PTP sources not designated as the switch's master from which the switch has received sync packets).

Command Mode

EXEC

Command Syntax

show ptp foreign-master-record

Examples

- This command displays information about PTP foreign masters.

```
switch# show ptp clocks foreign-masters-record
No Foreign Master Records
switch#
```

show ptp interface

The **show ptp interface** command displays PTP information for all the interfaces on the device.

Command Mode

EXEC

Command Syntax

show ptp [*INTERFACE_NAME*][*STATUS_FILTER*]

Parameters

- ***INTERFACE_NAME*** Interface type and numbers. Options include:
  - <no parameter> Display information for all interfaces.
  - **ethernet** *e_range* Ethernet interface range specified by *e_range*.
  - **loopback** *l_range* Loopback interface specified by *l_range*.
  - **management** *m_range* Management interface range specified by *m_range*.
  - **port-channel** *p_range* Port-Channel Interface range specified by *p_range*.
  - **vlan** *v_range* VLAN interface range specified by *v_range*.

  Valid range formats include number, number range, or comma-delimited list of numbers and ranges.
- ***STATUS_FILTER*** Filters interfaces by their configuration status. Options include:
  - <no parameter> all interfaces.
  - **enabled** PTP configured interfaces.

Examples

This command displays PTP information for all the interfaces on the device.

```
switch# show ptp interface
Interface Ethernet1
PTP: Disabled
Port state: Disabled
Sync interval: 1.0 seconds
Announce interval: 2.0 seconds
Announce interval timeout multiplier: 3
Delay mechanism: end to end
Delay request message interval: 32.0 seconds
Transport mode: ipv4
             <-------OUTPUT OMITTED FROM EXAMPLE-------->
Interface Ethernet5
PTP: Disabled
Port state: Disabled
Sync interval: 8.0 seconds
Announce interval: 2.0 seconds
Announce interval timeout multiplier: 5
Delay mechanism: peer to peer
Peer delay request message interval: 8.0 seconds
Peer Mean Path Delay: 0
Transport mode: ipv4
             <-------OUTPUT OMITTED FROM EXAMPLE-------->
switch#
```

show ptp interface counters

The **show ptp interface counters** command displays PTP interface counters for all interfaces.

Command Mode

EXEC

Command Syntax

show ptp [*INTERFACE_NAME*] counters

Parameters

- ***INTERFACE_NAME*** Interface type and numbers. Options include:
  - <no parameter> Display information for all interfaces.
  - **ethernet** *e_range* Ethernet interface range specified by *e_range*.
  - **loopback** *l_range* Loopback interface specified by *l_range*.
  - **management** *m_range* Management interface range specified by *m_range*.
  - **port-channel** *p_range* Port-Channel Interface range specified by *p_range*.
  - **vlan** *v_range* VLAN interface range specified by *v_range*.
  - **vxlan** *vx_range* VXLAN interface range specified by *vx_range*.

  Valid range formats include number, number range, or comma-delimited list of numbers and ranges.

Examples

- This command displays the PTP interface counters.

```
switch# show ptp interface ethernet 5 counters
Interface Ethernet5
Announce messages sent: 0
Announce messages received: 0
Sync messages sent: 0
Sync messages received: 0
Follow up messages sent: 0
Follow up messages received: 0
Delay request messages sent: 0
Delay request messages received: 0
Delay response messages sent: 0
Delay response messages received: 0
Peer delay request messages sent: 0
Peer delay request messages received: 0
Peer delay response messages sent: 0
Peer delay response messages received: 0
Peer delay response follow up messages sent: 0
Peer delay response follow up messages received: 0
switch#
```

show ptp parent

The **show ptp parent** command displays information about the switch's PTP parent and grand master clocks.

Command Mode

Privileged EXEC

Command Syntax

show ptp parent

Examples

- This command displays information about the switch's PTP parent and grand master clocks.

```
switch# show ptp parent
Parent Clock:
Parent Clock Identity: 0x00:1c:73:ff:ff:00:72:40
Parent Port Number: 0
Parent IP Address: N/A
Observed Parent Offset (log variance): N/A
Observed Parent Clock Phase Change Rate: N/A

Grandmaster Clock:
Grandmaster Clock Identity: 0x00:1c:73:ff:ff:00:72:40
Grandmaster Clock Quality:
   Class: 248
   Accuracy: 0x30
   OffsetScaledLogVariance: 0xffff
   Priority1: 128
   Priority2: 128
switch#
```

show ptp source ip

The **show ptp source ip** command displays the PTP source IP for the device.

Command Mode

Privileged EXEC

Command Syntax

show ptp source ip

Examples

- This command shows the PTP source IP to be 10.0.2.1.

```
switch#show ptp source ip
PTP source IP: 10.0.2.1
switch#
```

show ptp time-property

The **show ptp time-property** command displays the Precision Time Protocol (PTP) clock properties.

Command Mode

Privileged EXEC

Command Syntax

show ptp time-property

Examples

- This command shows the PTP clock properties.

```
switch# show ptp time-property
Current UTC offset valid: False
Current UTC offset: 0
Leap 59: False
Leap 61: False
Time Traceable: False
Frequency Traceable: False
PTP Timescale: False
Time Source: 0x0
switch#
```

Chapter 6

Booting the Switch

This chapter describes the switch boot process, describes configuration options, and lists the components it requires, including the boot loader, the boot loader shell, and other configuration files.

This chapter includes the following sections:

- Section 6.1: Boot Loader – Aboot
- Section 6.2: Configuration Files
- Section 6.3: Supervisor Redundancy
- Section 6.4: System Reset
- Section 6.5: Aboot Shell
- Section 6.6: Aboot Configuration Commands
- Section 6.7: Switch Booting Commands

6.1 Boot Loader – Aboot

Aboot is the boot loader for Arista switches. In addition to booting the switch EOS, Aboot provides a shell for changing boot parameters, restoring default switch settings, diagnosing hardware problems, and managing switch files. Section 6.5: Aboot Shell describes the Aboot shell.

The boot process loads an EOS image file, initiates switch processes, performs self tests, restores interface settings, and configures other network parameters. The replacement image file can be in the switch's flash or on a device in the flash drive port. Configuration files stored in flash memory specify boot parameters.

Aboot supports most available USB flash drive models. The flash drive must be formatted with the FAT or VFAT file system. Windows NT File System (NTFS) is not supported.

Aboot initiates a system reboot upon a **reload** command or by restoring power to the switch. Before loading the EOS image file, Aboot provides an option to enter the Aboot shell. The user can either enter the shell to modify boot parameters or allow the switch to boot.

The boot process can be monitored through a terminal connected to the console port. The console port is configured to interact with the terminal by configuration file settings.

6.2 Configuration Files

Three files define boot and running configuration parameters.

- **boot-config:** Contains the location and name of the image to be loaded.
- **running-config**: Contains the current switch configuration.
- **startup-config**: Contains the switch configuration that is loaded when the switch boots.

The ***running-config*** and ***startup-config*** are different when configuration changes have not been saved since the last boot.

6.2.1 boot-config

The ***boot-config*** file is an ASCII file that Aboot uses to configure console communication settings, locate the EOS flash image, and specify initial network configuration settings.

Aboot attempts to boot the EOS flash software image (with the extension .swi) referenced by ***boot-config*** if the user does not interrupt the boot process. See Section 6.5: Aboot Shell describes how Aboot uses ***boot-config***.

You can view and edit the ***boot-config*** file contents. Viewing and editing options include:

- View ***boot-config*** file contents with the **more boot-config** command:

```
switch(config)#more boot-config
SWI=flash:/EOS.swi
CONSOLESPEED=2400
Aboot password (encrypted): $1$A8dZ3GLZ$knKrBpTyg5dhmtGdCdwNM.
switch(config)#
```

- View ***boot-config*** settings with the **show boot-config** command:

```
switch(config)#show boot-config
Software image: flash:/EOS.swi
Console speed: 2400
Aboot password (encrypted): $1$A8dZ3GLZ$knKrBpTyg5dhmtGdCdwNM.
Memory test iterations: (not set)
switch(config)#
```

- Modify file settings from the command line with EOS **boot** commands.

  See Section 6.2.1.3: Programming boot-config from the CLI for a list of **boot** commands.

- Edit the file directly by using vi from the Bash shell.

  See Section 6.2.1.2: boot-config Command Line Content for a list of ***boot-config*** parameters.

6.2.1.1 boot-config File Structure

Each line in the ***boot-config*** file specifies a configuration setting and has this format:

NAME=VALUE

- NAME is the parameter label.
- VALUE indicates the parameter's bootup setting.

The NAME and VALUE fields cannot contain spaces.

Aboot ignores blank lines and lines that begin with a # character.

6.2.1.2 boot-config Command Line Content

Aboot configuration commands that ***boot-config*** files can contain include:

- **SWI** specifies the location and file name of the EOS image file that Aboot loads when booting, using the same format as the boot command to designate a local or network path.

Example

- `SWI=flash:EOS.swi` (flash drive location)
- `SWI=usb1:/EOS1.swi` (USB drive location)
- `SWI=file:/tmp/EOSexp.swi` (switch directory location)
- `SWI=/mnt/flash/EOS.swi`
- `SWI=http://foo.com/images/EOS.swi`
- `SWI=ftp://foo.com/images/EOS.swi`
- `SWI=tftp://foo.com/EOS.swi`
- `SWI=nfs://foo.com/images/EOS.swi`

- **CONSOLESPEED** specifies the console baud rate. To communicate with the switch, the connected terminal must match the specified rate. Baud rates are 1200, 2400, 4800, 9600, 19200, or 38400. The default baud rate is 9600.

Example

```
CONSOLESPEED=2400
CONSOLESPEED=19200
```

- **PASSWORD (ABOOT)** specifies the Aboot password, as described in Section 6.5.2: Accessing the Aboot Shell. If ***boot-config*** does not contain a PASSWORD line, the Aboot shell does not require a password.

Example

```
PASSWORD=$1$CdWp5wfe$pzNtE3ujBoFEL8vjcq7jo/
```

- **NET commands** NET commands in the ***boot-config*** file are used by Aboot during switch booting to configure the network interface that will be used for switch configuration. These commands can also be entered manually in Aboot.

  NETDEV indicates which network interface is being configured. If ***boot-config*** does not contain a NETDEV setting, the booting process does not attempt to configure a network interface. Other NET commands specify settings that Aboot uses to configure the interface.

Examples

- This NETDEV command specifies management port 1 as the network interface to be configured by ***boot-config***.

```
NETDEV=ma1
This NETAUTO command instructs the switch to configure the network interface
through a DHCP server, ignoring other NET settings.
NETAUTO=dhcp
These NET commands configure the network interface.
NETIP=10.12.15.10
NETMASK=255.255.255.0
NETGW=10.12.15.24
NETDOMAIN=mycompany.com
NETDNS=10.12.15.13
```

6.2.1.3 Programming boot-config from the CLI

The switch CLI provides **boot** commands for editing ***boot-config*** contents. The **boot** commands are not accessible from a console port CLI. Parameters not configurable from a boot command can be modified by directly editing the ***boot-config*** file.

Commands that configure boot parameters include **boot system**, **boot secret**, and **boot console**.

boot system

The **boot system** command provides the EOS image file location to Aboot.

Example

- This command specifies EOS1.swi on USB flash memory as the software image load file.

  ```
  switch(config)#boot system usb1:EOS1.swi
  ```

  The **boot system** command above adds this line to ***boot-config***.

  ```
  SWI=usb1:/EOS1.swi
  ```

- This command designates EOS.swi, on the switch flash, as the EOS software image load file.

  ```
  switch(config)#boot system flash:EOS.swi
  ```

  The **boot system** command above adds this line to ***boot-config***.

  ```
  SWI=flash:/EOS.swi
  ```

boot secret

The **boot secret** command sets the Aboot password.

Example

- These equivalent commands set the Aboot password to xr19v.

  ```
  switch(config)#boot secret xr19v

  switch(config)#boot secret 0 xr19v
  ```

  This command shows the password that has been set.

  ```
  switch(config)#show boot-config
  Software image: flash:/EOS.swi
  Console speed: (not set)
  Aboot password (encrypted): $1$k9YHFW8D$cgM8DSN.e/yY0p3k3RUvk.
  ```

  The **boot secret** commands above add this line to ***boot-config***.

  ```
  PASSWORD=$1$k9YHFW8D$cgM8DSN.e/yY0p3k3RUvk.
  ```

  The user must enter xr19v at the login prompt to access the Aboot shell.

- This command sets the Aboot password to ***xr123***. The encrypted string was previously generated with ***xr123*** as the clear-text seed.

  ```
  switch(config)#boot secret 5 $1$QfbYkVWb$PIXG0udEquW0wOSiZBN3D/
  ```

  This command shows the password that has been set.

  ```
  switch(config)#show boot-config
  Software image: flash:/EOS.swi
  Console speed: (not set)
  Aboot password (encrypted): $1$QfbYkVWb$PIXG0udEquW0wOSiZBN3D/
  ```

The **boot secret** command above adds this line to ***boot-config***.

```
PASSWORD=$1$QfbYkVWb$PIXG0udEquW0wOSiZBN3D/
```

The user must enter ***xr123*** at the login prompt to access the Aboot shell.

- This command removes the Aboot password; subsequent Aboot access is not authenticated.

```
switch(config)#no boot secret
```

This command shows that there is now no Aboot password.

```
switch(config)#show boot-config
Software image: flash:/EOS.swi
Console speed: (not set)
Aboot password (encrypted): (not set)
```

boot console

The **boot console** command sets console settings for attaching devices.

Example

- This command sets the console speed to 4800 baud:

```
switch(config)#boot console speed 4800
```

This command shows the console speed.

```
switch(config)#show boot-config
Software image: flash:/EOS.swi
Console speed: 4800
Aboot password (encrypted): (not set)
```

The **boot console** command above adds this line to ***boot-config***.

```
CONSOLESPEED=4800
```

6.2.2 Running-Config

running-config is a virtual file that contains the system's operating configuration, formatted as a command sequence. Commands entered from the CLI modify ***running-config***. Copying a file to ***running-config*** updates the operating configuration by executing the commands in the copied file.

running-config commands include:

- **show running-config** displays ***running-config***.
- **copy running-config startup-config** copies ***running-config*** contents to the ***startup-config***.
- **write** copies ***running-config*** contents to the ***startup-config*** file.

6.2.3 Startup-Config

The ***startup-config*** file is stored in flash memory and contains the configuration that the switch loads when booting. During a switch boot, ***running-config*** is replaced by ***startup-config***. Changes to ***running-config*** that are not copied to ***startup-config*** are lost when the system reboots.

startup-config commands include:

- **show startup-config** displays ***startup-config***.
- **copy** <filename> **startup-config** copies contents of the specified file to ***startup-config***.
- **erase startup-config** deletes the ***startup-config*** file.

6.3 Supervisor Redundancy

On modular switches with redundant supervisor modules, control of the switch can be transferred to the standby supervisor to minimize downtime and data loss in the case of a reset, reload, or failure of the active supervisor. How the switchover takes place is determined by the redundancy protocol on the active supervisor.

To display the state and the current redundancy protocol of both supervisors, use the **show redundancy states** command. To display the state of configuration file synchronization between the supervisors, use the **show redundancy file-replication** command.

6.3.1 Redundancy Supervisor Protocols

There are three available supervisor redundancy protocols.

Route Processor Redundancy (RPR)

The default redundancy protocol is route processor redundancy (RPR), which synchronizes ***startup-config*** files between the supervisor modules and partially boots the standby supervisor to a "standby warm" state, but does not synchronize ***running-config***. If the active supervisor fails, or a manual switchover is initiated with the **redundancy force-switchover** command, the standby supervisor will become active. Running state, including spanning tree, is lost, and all links are temporarily brought down.

Under RPR, the CLI of the standby supervisor can be accessed by SSH or through the console port, but the available command set is limited. Any configuration changes made to the standby supervisor will be lost when the supervisor reboots.

Stateful Switchover (SSO)

In stateful switchover (SSO) protocol, the switch synchronizes both ***startup-config*** and ***running-config*** files between the supervisor modules and fully boots the standby module to a "standby hot" state to speed the switchover process and minimize packet loss. If the active supervisor fails, or a manual switchover is initiated, the standby supervisor immediately becomes active, and L2 running state is maintained. An SSO switchover is largely transparent from the outside, but because L3 state is not synchronized the switchover can result in traffic loss for traffic forwarded on routes learned by a dynamic routing protocol. Enabling nonstop forwarding can eliminate most packet loss for BGP and OSPF.

Under SSO, the CLI of the standby supervisor can be accessed only through the console port, and the command set is limited. Any configuration changes made on the standby supervisor will be lost when the supervisor reboots.

Important! When upgrading the EOS on a dual-supervisor switch to an SSO-capable version (4.11.0 or higher) from a version that does not support SSO, both supervisors will reset simultaneously, causing several seconds of system downtime.

Simplex

When the switch is set to simplex protocol, the standby supervisor is disabled and switchover will not occur even if the active supervisor fails. Reloading the active supervisor results in system downtime while the supervisor reboots, and the standby supervisor remains disabled. To transfer control of the switch to the standby supervisor, the redundancy protocol must be changed to **RPR** or **SSO.**

Under simplex protocol, the CLI of the disabled supervisor can be accessed only through the console port, and the command set is limited. Any configuration changes made on the standby supervisor will be lost when the supervisor reboots.

6.3.2 Configuring Supervisor Redundancy

The supervisor redundancy protocol is configured using the **protocol** command in redundancy configuration mode (accessed with the **redundancy** command).

Changing the redundancy protocol on the active supervisor resets the standby supervisor regardless of redundancy protocol, and executing the **write** command on the active supervisor synchronizes the ***startup-config*** files between supervisors in RPR and SSO modes.

Examples

- These commands display the current redundancy state of the switch and the most recent file synchronization information.

```
switch#show redundancy state
  my state = ACTIVE
peer state = STANDBY WARM
      Unit = Primary
   Unit ID = 1

Redundancy Protocol (Operational) = Route Processor Redundancy
Redundancy Protocol (Configured) = Route Processor Redundancy
Communications = Up
Ready for switchover

  Last switchover time = 7:23:56 ago
Last switchover reason = Supervisor has control of the active supervisor lock
Switch#show redundancy file-replication
0 files unsynchronized, 2 files synchronized, 0 files failed, 2 files total.

File                   Status         Last Synchronized
---------------------- -------------- -------------------
file:persist/sys       Synchronized   0:10:04 ago
flash:startup-config   Synchronized   0:10:04 ago
```

- These commands set the redundancy protocol for the active supervisor to stateful switchover (SSO).

```
switch#config
switch(config)#redundancy
switch(config-redundancy)#protocol sso
Peer supervisor will be restarted.
switch(config-redundancy)#
```

6.4 System Reset

When a reset condition exists, Aboot can either reset the switch without user intervention or facilitate a manual reset through the Aboot shell. A reset operation clears the switch, including memory states and other hardware logic

- Fixed systems: The power supply remains powered up through the reset. Power is removed from all other switch components for two to five seconds.
- Modular systems: The power supply on the active supervisor remains powered up through the reset. Power is removed from all other supervisor components for at least one second. In stateful switchover (SSO) and route processor redundancy (RPR) modes, resetting the standby supervisor has no effect on the active supervisor, but resetting the active supervisor causes the standby supervisor to immediately become active. After the supervisor becomes functional, it manages the power-cycling of all line cards.

The **reload** command initiates an immediate reset, terminating all CLI instances not running through the console port. The console port CLI displays messages that the switch generates during a reset. On modular switches with redundant supervisors, CLI sessions on the standby supervisor are not terminated.

The **reload <scheduled>** command schedules a reset operation to initiate at a specific time or after a specified period.

6.4.1 Typical Reset Sequence

The **reload** command power cycles the switch, then resets it under Aboot control. The hard reset clears the switch, including memory states and other hardware logic.

By default, the **reload** command triggers a request to store unsaved ***running-config*** commands and an option to open the Aboot shell before starting the reboot when accessing the CLI through the console port. The switch then begins the reboot process controlled by Aboot.

This procedure is an example of a typical restart.

Step 1 Begin the reboot process by typing the **reload** command:

```
switch#reload
```

The switch sends a message to confirm the reload request:

```
Proceed with reload? [confirm]
```

Step 2 Press **enter** or type **y** to confirm the requested reload. Pressing any other key terminates the reload operation.

The switch sends a series of messages, including a notification that a message was broadcast to all open CLI instances, informing them that the system is being rebooted. The reload pauses when the CLI displays the Aboot shell notification line.

```
Broadcast message from root@mainStopping sshd: [  OK  ]
SysRq : Remount R/O
Restarting system

Aboot 1.9.0-52504.EOS2.0

Press Control-C now to enter Aboot shell
```

Step 3 To continue the reload process, do nothing. Typing **Ctrl-C** opens the Aboot shell; see Section 6.5.5: Commands for Aboot editing instructions.

The switch continues the reset process, displaying messages to indicate the completion of individual tasks. The reboot is complete when the CLI displays a login prompt.

```
Booting flash:/EOS.swi
Unpacking new kernel
Starting new kernel
Switching to rooWelcome to Arista Networks EOS 4.4.0
Mounting filesystems:  [  OK  ]
Entering non-interactive startup
Starting EOS initialization stage 1: [  OK  ]
ip6tables: Applying firewall rules: [  OK  ]
iptables: Applying firewall rules: [  OK  ]
iptables: Loading additional modules: nf_conntrack_tftp [  OK  ]
Starting system logger: [  OK  ]
Starting system message bus: [  OK  ]
Starting NorCal initialization: [  OK  ]
Starting EOS initialization stage 2: [  OK  ]
Starting ProcMgr: [  OK  ]
Completing EOS initialization: [  OK  ]
Starting Power On Self Test (POST): [  OK  ]
Generating SSH2 RSA host key: [  OK  ]
Starting isshd: [  OK  ]
Starting sshd: [  OK  ]
Starting xinetd: [  OK  ]
[  OK  ] crond: [  OK  ]

switch login:
```

Step 4 Log into the switch to resume configuration tasks.

6.4.2 Switch Recovery

Aboot can automatically erase the internal flash and copy the contents of a USB drive that has been inserted before powering up or rebooting the switch. This recovery method does not require access to the switch console or Aboot password entry, even if the ***boot-config*** file lists one.

Aboot invokes the recovery mechanism only if each of these two conditions is met:

- The USB drive must contain a file called **fullrecover**

  The file's contents are ignored; an empty text file is sufficient.

- If the USB drive contains a file named ***boot-config***, its timestamp must differ from the timestamp of the ***boot-config*** file on the internal flash.

  This prevents Aboot from invoking the recovery mechanism again on every boot if you leave the flash key inserted.

To use this recovery mechanism, set up a USB drive with the files to be installed on the internal flash – for example, a current EOS software image, and a customized or empty ***boot-config*** – plus an empty file named fullrecover.

Check that the timestamp of ***boot-config*** is current to ensure that the above conditions are met.

6.4.3 Display Reload Cause

The **show reload cause** command displays the cause of the most recent system reset and lists recommended actions, if any exist, to avoid future spontaneous resets or resolve other issues that may have cause the reset.

Example

- To display the reset cause, type **show reload cause** at the prompt.

```
switch# show reload cause
Reload Cause 1:
------------------
Reload requested by the user.

Recommended Action:
------------------
No action necessary.

Debugging Information:
----------------------
None available.
switch#
```

6.4.4 Configuring Zero Touch Provisioning

Zero Touch Provisioning (ZTP) is a switch configuration method that uses files referenced by a DCHP server to initially provision the switch without user intervention. A switch enters ZTP mode when it is reloaded if flash memory does not contain a ***startup-config***.

Canceling ZTP boots the switch without using a ***startup-config*** file. When ZTP mode is canceled, a ***startup-config*** file is not stored to flash memory. Until a ***startup-config*** file is stored to flash, the switch returns to ZTP mode on subsequent reboots. This section describes steps required to implement, monitor, and cancel ZTP.

6.4.4.1 Configuring the Network for ZTP

A switch performs the following after booting in ZTP mode:

- Configures each physical interface to ***no switchport*** mode.
- Sends a DHCP query packet on all Ethernet and management interfaces.

After the switch receives a DHCP offer, it responds with a DHCP request for Option 66 (TFTP server name), Option 67 (bootfile name), and dynamic network configuration settings. When the switch receives a valid DHCP response, it configures the network settings, then fetches the file from the location listed in Option 67. If Option 67 returns a network URL (http:// or ftp://), the switch obtains the file from the network. If Option 67 returns a file name, the switch retrieves the file from the TFTP server listed in Option 66.

The Option 67 file can be a ***startup-config*** file or a boot script. The switch distinguishes between a ***startup-config*** file and a boot script by examining the first line in the file:

- The first line of a boot file must consist of the #! characters followed by the interpreter path. The switch executes the code in the script, then reboots. The boot script may fetch an EOS software image or perform required customization tasks.

  The following boot file fetches an EOS software image and stores a startup configuration file to flash.

  ```
  #!/usr/bin/Cli -p2
  copy http://company.com/startup-config flash:startup-config
  copy http://company.com/EOS-2.swi flash:EOS-2.swi
  config
  boot system flash:EOS-2.swi
  ```

- The switch identifies any other file as a ***startup-config*** file. The switch copies the ***startup-config*** file into flash as ***mnt/flash/startup-config***, then reboots.

The switch uses its system MAC address as the DHCP client identifier and ***Arista*** as the Vendor Class Identifier (Option 60). When the switch receives an http URL through Option 67, it sends the following http headers in the GET request:

```
X-Arista-SystemMAC:
X-Arista-HardwareVersion:
X-Arista-SKU:
X-Arista-Serial:
X-Arista-Architecture:
```

6.4.4.2 Monitoring ZTP Progress

A switch displays the following message after rebooting when it does not contain a ***startup-config*** file:

```
No startup-config was found.

The device is in Zero Touch Provisioning mode and is attempting to
download the startup-config from a remote system. The device will not
be fully functional until either a valid startup-config is downloaded
from a remote system or Zero Touch Provisioning is cancelled. To cancel
Zero Touch Provisioning, login as admin and type 'zerotouch cancel'
at the CLI.

switch login:
```

The switch displays a CONFIG_DOWNLOAD_SUCCESS message after it successfully downloads a ***startup-config*** file, then continues the reload process as described in Section 6.4.1.

```
===============================================================================

Successful download
--------------------


Apr 15 21:36:46 switch ZeroTouch: %ZTP-5-DHCP_QUERY: Sending DHCP request on  [
Ethernet10, Ethernet13, Ethernet14, Ethernet17, Ethernet18, Ethernet21,
Ethernet22, Ethernet23, Ethernet24, Ethernet7, Ethernet8, Ethernet9,
Management1, Management2 ]
Apr 15 21:36:56 switch ZeroTouch: %ZTP-5-DHCP_SUCCESS: DHCP response received on
Ethernet24  [ Mtu: 1500; Ip Address: 10.10.0.4/16; Nameserver: 10.10.0.1; Domain:
aristanetworks.com; Gateway: 10.10.0.1; Boot File:
http://10.10.0.2:8080/tmp/172.17.11.196-startup-config.1 ]
Apr 15 21:37:01 switch ZeroTouch: %ZTP-5-CONFIG_DOWNLOAD: Attempting to download
the startup-config from http://10.10.0.2:8080/tmp/172.17.11.196-startup-config.1
Apr 15 21:37:02 switch ZeroTouch: %ZTP-5-CONFIG_DOWNLOAD_SUCCESS: Successfully
downloaded startup-config from
http://10.10.0.2:8080/tmp/172.17.11.196-startup-config.1
Apr 15 21:37:02 switch ZeroTouch: %ZTP-5-RELOAD: Rebooting the system
Broadcast messagStopping sshd: [  OK  ]
watchdog is not running
SysRq : Remount R/O
Restarting system
ø

Aboot 1.9.0-52504.EOS2.0


Press Control-C now to enter Aboot shell
```

6.4.4.3 ZTP Failure Notification

The switch displays a DHCP_QUERY_FAIL message when it does not receive a valid DHCP response within 30 seconds of sending the query. The switch then sends a new DHCP query and waits for a response. The switch continues sending queries until it receives a valid response or until ZTP mode is canceled.

```
switch login:admin
admin
switch>Apr 15 21:28:21 localhost ZeroTouch: %ZTP-5-DHCP_QUERY: Sending DHCP
request on  [ Ethernet10, Ethernet13, Ethernet14, Ethernet17, Ethernet18,
Ethernet21, E-thernet22, Ethernet23, Ethernet24, Ethernet7, Ethernet8,
Ethernet9, Management1, Management2 ]
Apr 15 21:28:51 localhost ZeroTouch: %ZTP-5-DHCP_QUERY_FAIL: Failed to get a
valid DHCP response
Apr 15 21:28:51 localhost ZeroTouch: %ZTP-5-RETRY: Retrying Zero Touch
Provisioning from the begining (attempt 1)
Apr 15 21:29:22 localhost ZeroTouch: %ZTP-5-DHCP_QUERY: Sending DHCP request on
[ Ethernet10, Ethernet13, Ethernet14, Ethernet17, Ethernet18, Ethernet21,
Ethernet22, Ethernet23, Ethernet24, Ethernet7, Ethernet8, Ethernet9,
Management1, Management2 ]
```

6.4.4.4 Canceling ZTP Mode

To boot the switch without a ***startup-config*** file, log into the console, then cancel ZTP mode. After the switch boots, it uses all factory default settings. A ***startup-config*** file must be saved to flash memory to prevent the switch from entering ZTP mode on subsequent boots.

6.4.5 Configuring the Networks

If the ***boot-config*** file contains a NETDEV statement, Aboot attempts to configure the network interface, as specified by Network configuration commands. See Section 6.2.1.2: boot-config Command Line Content for a list of commands that define the network configuration.

6.5 Aboot Shell

The Aboot shell is an interactive command-line interface used to manually boot a switch, restore the internal flash to its factory-default state, run hardware diagnostics, and manage files. The Aboot shell is similar to the Linux Bourne Again Shell (Bash).

The Aboot shell provides commands for restoring the state of the internal flash to factory defaults or a customized default state. You can use these recovery methods to:

- restore the factory-default flash contents before transferring the switch to another owner.
- restore Aboot shell access if the Aboot password is lost or forgotten.
- restore console access if baud rate or other settings are incompatible with the terminal.
- replace the internal flash contents with configuration or image files stored on a USB flash drive.

6.5.1 Operation

When the switch is powered on or rebooted, Aboot reads its configuration from ***boot-config*** on the internal flash and attempts to boot an EOS software image (with the extension .swi) automatically if one is configured.

You can monitor the automatic boot process or enter the Aboot shell only from the console port. You can connect a PC or terminal directly to the port and run a terminal emulator to interact with the serial port or access it through a serial concentrator device.

Console settings are stored in ***boot-config***; the factory-default settings for Arista switches are 9600 baud, no parity, 8 character bits, and 1 stop bit. If you do not know the current settings, perform a full flash recovery to restore the factory-default settings. When the console port is connected and the terminal settings are configured properly, the terminal displays a message similar to the following a few seconds after powering up the switch:

```
Aboot 1.0.0

Press Control-C now to enter the Aboot shell
```

To abort the automatic boot process and enter the Aboot shell, press **Ctrl-C** (ASCII 3 in the terminal emulator) after the *Press Control-C now to enter Aboot shell* message appears. Pressing **Ctrl-C** can interrupt the boot process up through the starting of the new kernel.

If the ***boot-config*** file does not contain a password command, the Aboot shell starts immediately. Otherwise, you must enter the correct password at the password prompt to start the shell. If you enter the wrong password three times, Aboot displays this message:

```
Type "fullrecover" and press Enter to revert /mnt/flash to factory default
state, or just press Enter to reboot:
```

- Pressing **Enter** continues a normal soft reset without entering the Aboot shell.
- Typing **fullrecover** and pressing **Enter** performs a full flash recovery to restore the factory-default settings, removing all previous contents of the flash drive.

The Aboot shell starts by printing:

```
Welcome to Aboot.
```

Aboot then displays the Aboot# prompt.

Aboot reads its configuration from ***boot-config*** on the internal flash.

6.5.2 Accessing the Aboot Shell

This procedure accesses the Aboot Shell:

Step 1 Reload the switch and press **enter** or type **y** when prompted, as described by step 1 and step 2 in Section 6.4.1: Typical Reset Sequence.

The command line displays this Aboot entry prompt.

```
Press Control-C now to enter Aboot shell
```

Step 2 Type **Ctrl-C**.

If the ***boot-config*** file does not contain a PASSWORD command, the CLI displays an Aboot welcome banner and prompt.

```
^CWelcome to Aboot.
Aboot#
```

If the ***boot-config*** file contains a PASSWORD command, the CLI displays a password prompt. In this case, proceed to step 3. Otherwise, the CLI displays the Aboot prompt.

Step 3 If prompted, enter the Aboot password.

```
Press Control-C now to enter Aboot shell
^CAboot password:
Welcome to Aboot.
Aboot#
```

Aboot allows three attempts to enter the correct password. After the third attempt, the CLI prompts the user to either continue the reboot process without entering the Aboot shell or to restore the flash drive to the factory default state.

```
Press Control-C now to enter Aboot shell
^CAboot password:
incorrect password
Aboot password:
incorrect password
Aboot password:
incorrect password
Type "fullrecover" and press Enter to revert /mnt/flash to factory default
state, or just press Enter to reboot: fullrecover
All data on /mnt/flash will be erased; type "yes" and press Enter to proceed,
or just press Enter to cancel:
```

The **fullrecover** operation replaces the flash contents with a factory default configuration. The CLI displays text similar to the following when performing a fullrecover, finishing with another entry option into the Aboot shell.

```
Erasing /mnt/flash
Writing recovery data to /mnt/flash
boot-config
startup-config
EOS.swi
210770 blocks
Restarting system.


Aboot 1.9.0-52504.EOS2.0


Press Control-C now to enter Aboot shell
```

6.5.3 File Structure

When you enter the Aboot CLI, the current working directory is the root directory on the switch. Switch image and configuration files are at **/mnt/flash.** When exiting the Aboot shell, only the contents of **/mnt/flash** are preserved. The **/mnt** directory contains the file systems of storage devices. Aboot mounts the internal flash device at **/mnt/flash**.

When a USB flash drive is inserted in one of the flash ports, Aboot mounts its file system on **/mnt/usb1**. The file system is unmounted when the USB flash drive is removed from the port. Most USB drives contain an LED that flashes when the system is accessing it; do not remove the drive from the flash port until the LED stops flashing.

6.5.4 Booting From the Aboot Shell

Aboot attempts to boot the EOS software image (with the extension .swi) configured in ***boot-config*** automatically if you take no action during the boot process. If the boot process fails for any reason, such as an incorrectly configured software image, Aboot enters the shell, allowing you to correct the configuration or boot a software image manually. The **boot** command loads and boots an EOS software image file.

The **boot** command syntax is

boot *SWI*

where *SWI* lists the location of the EOS image that the command loads. *SWI* settings include:

- *DEVICE:PATH*Loads the image file from the specified storage device. The default *DEVICE* value is **flash;** other values include **file** and **usb1.**
- */PATH*Loads the image file from the specified path in the switch directory.
- **http://***SERVER/PATH*Loads the image file from the HTTP server on the host server.
- **ftp://***SERVER/PATH*Loads the image file from the FTP server on the host server.
- **tftp:**//*SERVER/PATH*Loads the image file from the TFTP server on the host server
- **nfs:**//*SERVER/PATH*Mounts the path's parent directory from the host server and loads the image file from the loaded directory.

The accepts the same commands as the SWI variable in the ***boot-config*** file. See Section 6.2.1.2: boot-config Command Line Content for a list of boot command formats.

If an image file is not specified in ***boot-config***, or if booting the image results in an error condition (for example, an incorrect path or unavailable HTTP server), Aboot halts the boot process and drops into the shell.

Example

- To boot EOS.swi from internal flash, enter one of these commands on the Aboot command line:

```
boot flash:EOS.swi
boot /mnt/flash/EOS.swi.
```

6.5.5 Commands

To list the contents of the internal flash, enter **ls /mnt/flash** at the Aboot# prompt.

Example

```
Aboot# ls /mnt/flash
EOS.swi boot-config startup-config
```

Commonly used commands include:

- `ls` Prints a list of the files in the current working directory.
- `cd` Changes the current working directory.
- `cp` Copies a file.
- `more` Prints the contents of a file one page at a time.
- `vi` Edits a text file.
- `boot` Boots a software image file.
- `swiinfo` Prints information about a software image.
- `recover` Recovers the factory-default configuration.
- `reboot` Reboots the switch.
- `udhcpc` Configures a network interface automatically via DHCP.
- `ifconfig` Prints or alters network interface settings.
- `wget` Downloads a file from an HTTP or FTP server.

Many Aboot shell commands are provided by Busybox, an open-source implementation of UNIX utilities. Busybox command help is found at http://www.busybox.net/downloads/BusyBox.html. Aboot provides access to only a subset of the documented commands.

Aboot can access networks through the Ethernet management ports. Aboot provides network interfaces mgmt1 and mgmt2. These ports are unconfigured by default; you can configure management port settings using Aboot shell commands like **ifconfig** and **udhcpc**. When a management interface is configured, use **wget** to transfer files from an HTTP or FTP server, **tftp** to transfer files from a TFTP server, or **mount** to mount an NFS filesystem.

6.6 Aboot Configuration Commands

This section describes the Aboot configuration commands that a ***boot-config*** file can contain.

- SWI
- CONSOLESPEED
- PASSWORD (ABOOT)
- NET commands

CONSOLESPEED

CONSOLESPEED specifies the console baud rate. To communicate with the switch, the connected terminal must match the specified rate. Baud rates are 1200, 2400, 4800, 9600, 19200, or 38400.

The default baud rate is 9600.

Command Syntax

CONSOLESPEED=*baud_rate*

Parameters

- *baud_rate* specifies the console speed. Values include **1200**, **2400**, **4800**, **9600**, **19200**, or **38400**

Examples

- These lines are CONSOLESPEED command examples:

```
CONSOLESPEED=2400
CONSOLESPEED=19200
```

NET commands

NET commands in the ***boot-config*** file are used by Aboot during switch booting to configure the network interface that will be used for switch configuration. These commands can also be entered manually in Aboot.

NETDEV indicates which network interface is being configured. If ***boot-config*** does not contain a NETDEV setting, the booting process does not attempt to configure a network interface. Other NET commands specify settings that Aboot uses to configure the interface.

Command Syntax

```
NETDEV=interface
NETAUTO=auto_setting
NETIP=interface_address
NETMASK=interface_mask
NETGW=gateway_address
NETDOMAIN=domain_name
NETDNS=dns_address
```

Parameters

- *interface* the network interface. Settings include:
  - **NETDEV=ma1** management port 1.
  - **NETDEV=ma2** management port 2.
- *auto_setting* the configuration method. Settings include
  - **NETAUTO=dhcp** interface is configured through a DHCP server; other NET commands are ignored.
  - **NETAUTO** command is omitted interface is configured with other NET commands,
- *interface_address* interface IP address, in dotted-decimal notation.
- *interface_mask* interface subnet mask, in dotted-decimal notation.
- *gateway_address* default gateway IP address, in dotted decimal notation.
- *domain_name* interface domain name.
- *dns_address* IP address of the Domain Name Server, in dotted decimal notation.

Examples

- This NETDEV command specifies management port 1 as the network interface to be configured for management traffic.

  ```
  NETDEV=ma1
  ```

- This NETAUTO command instructs the switch to configure the network interface through a DHCP server, ignoring other NET settings.

  ```
  NETAUTO=dhcp
  ```

- These NET commands configure the network interface.

  ```
  NETIP=10.12.15.10
  NETMASK=255.255.255.0
  NETGW=10.12.15.24
  NETDOMAIN=mycompany.com
  NETDNS=10.12.15.13
  ```

PASSWORD (ABOOT)

PASSWORD specifies the Aboot password, as described in Section 6.5.2: Accessing the Aboot Shell. If ***boot-config*** does not contain a PASSWORD line, the Aboot shell does not require a password.

boot-config stores the password as an MD5-encrypted string as generated by the UNIX passwd program or the crypt library function from a clear-text seed. When entering the Aboot password, the user types the clear-text seed.

There is no method of recovering the password from the encrypted string. If the clear-text password is lost, delete the corresponding PASSWORD command line from the ***boot-config*** file.

The EOS **boot secret** command is the recommended method of adding or modifying the PASSWORD configuration line.

Command Syntax

PASSWORD=*encrypted_string*

Parameters

- *encrypted_string* the encrypted string that corresponds to the clear-text Aboot password.

Example

- This line is a PASSWORD command example where the encrypted string corresponds with the clear-text password **abcde**.

```
PASSWORD=$1$CdWp5wfe$pzNtE3ujBoFEL8vjcq7jo/
```

SWI

SWI specifies the location and file name of the EOS image file that Aboot loads when booting, using the same format as ***boot-config*** to designate a local or network path.

Command Syntax

SWI=*FILE_LOCATION*

Parameters

- ***FILE_LOCATION*** specifies the location of the EOS image file. Formats include:
  - *device*:*path* storage device location:

    device denotes a storage device. Settings include **flash**, **file** and **usb1**. Default is **flash**.
    path denotes a file location.

Examples

flash drive location `SWI=flash:EOS.swi`
`usb drive location.` `SWI=usb1:/EOS1.swi`
`switch directory location` `SWI=file:/tmp/EOSexp.swi`

- */path* switch directory location.

Example `SWI=/mnt/flash/EOS.swi`

- **http://***server/path* HTTP server location.

Example `SWI=http://foo.com/images/EOS.swi`

- **ftp://***server/path* FTP server location.

Example `SWI=ftp://foo.com/images/EOS.swi`

- **tftp://***server/path* TFTP server location.

Example `SWI=tftp://foo.com/EOS.swi`

- **nfs:***//server/path* imports *path* from *server*, then mounts parent directory of the *path*

Example `SWI=nfs://foo.com/images/EOS.swi`

6.7 Switch Booting Commands

- boot console
- boot secret
- boot system
- erase startup-config
- protocol
- redundancy
- redundancy force-switchover
- reload
- reload <scheduled>
- service sequence-numbers
- show redundancy file-replication
- show redundancy states
- show redundancy switchover sso
- show reload
- show reload cause

boot console

The **boot console** command configures terminal settings for serial devices connecting to the console port. Console settings that you can specify from ***boot-config*** include:

- speed

Factory-default console settings are 9600 baud, no parity, 8 character bits, and 1 stop bit. If you do not know the current settings, restore the factory-default settings as described in Section 2.4.3: Restoring the Factory Default EOS Image and Startup Configuration.

The **no boot console** and **default boot console** commands restore the factory default settings on the switch and remove the corresponding CONSOLESPEED command from the ***boot-config*** file.

Command Mode

Global Configuration

Command Syntax

```
boot console speed baud
no boot console speed
default boot console speed
```

Parameters

- *baud* console baud rate. Settings include 1200, 2400, 4800, 9600, 19200, and 38400.

Example

- This command sets the console speed to 4800 baud

```
switch(config)#boot console speed 4800
```

This code displays the result of the command:

```
switch(config)#show boot-config
Software image: flash:/EOS.swi
Console speed: 4800
Aboot password (encrypted): (not set)
```

The above **boot console** command adds the following line to ***boot-config***.

```
CONSOLESPEED=4800
```

boot secret

The **boot secret** command creates or edits the Aboot shell password and stores the encrypted string in the PASSWORD command line of the ***boot-config*** file.

The **no boot secret** and **default boot secret** commands remove the Aboot password from the ***boot-config*** file. When the Aboot password does not exist, entering Aboot shell does not require a password.

Command Mode

Global Configuration

Command Syntax

```
boot secret [ENCRYPT_TYPE] password
no boot secret
default boot secret
```

Parameters

- ***ENCRYPT_TYPE*** indicates the encryption level of the password parameter. Settings include:
  - <no parameter> the *password* is clear text.
  - **0** the *password* is clear text. Equivalent to the <no parameter> case.
  - **5** the *password* is an md5 encrypted string.
  - **sha512** the password is entered as an sha512 encrypted string.
- *password* specifies the boot password.
  - *password* must be in clear text if ***ENCRYPT_TYPE*** specifies clear text.
  - *password* must be an appropriately encrypted string if ***ENCRYPT_TYPE*** specifies encryption.

Restrictions

The **sha512** encryption option is not available on Trident platform switches.

Examples

- These equivalent commands set the Aboot password to xr19v:

```
switch(config)#boot secret xr19v

switch(config)#boot secret 0 xr19v
```

This CLI code displays the result:

```
switch(config)#show boot-config
Software image: flash:/EOS.swi
Console speed: (not set)
Aboot password (encrypted): $1$k9YHFW8D$cgM8DSN.e/yY0p3k3RUvk.
```

The **boot secret** commands above add this line to ***boot-config.***

```
PASSWORD=$1$k9YHFW8D$cgM8DSN.e/yY0p3k3RUvk.
```

The user must enter ***xr19v*** at the login prompt to access the Aboot shell.

- These commands set the Aboot password to xr123, then displays the resulting ***boot-config*** code. The encrypted string was previously generated with xr123 as the clear-text seed.

```
switch(config)#boot secret 5 $1$QfbYkVWb$PIXG0udEquW0wOSiZBN3D/
switch(config)#show boot-config
Software image: flash:/EOS.swi
Console speed: (not set)
Aboot password (encrypted): $1$QfbYkVWb$PIXG0udEquW0wOSiZBN3D/
```

The **boot secret** command above adds this line to ***boot-config***.

```
PASSWORD=$1$QfbYkVWb$PIXG0udEquW0wOSiZBN3D/
```

The user must enter xr123 at the login prompt to access the Aboot shell.

- This command removes the Aboot password, allowing access to the Aboot shell without a password.

```
switch(config)#no boot secret
```

boot system

The **boot system** command specifies the location of the EOS software image that Aboot loads when the switch boots. The command can refer to files on flash or on a module in the USB flash port.

Command Mode

Global Configuration

Command Syntax

boot system *DEVICE* *file_path*

Parameters

- ***DEVICE*** Location of the image file. Options include
  - **file:** file is located in the switch file directory.
  - **flash:** file is located in flash memory.
  - **usb1:** file is located on a drive inserted in the USB flash port. Available if a drive is in the port.
- *file_path* Path and name of the file.

Examples

- This command designates EOS1.swi, on USB flash memory, as the EOS software image load file.

  ```
  switch(config)#boot system usb1:EOS1.swi
  ```

  The **boot system** command above adds this line to ***boot-config***.

  ```
  SWI=usb1:/EOS1.swi
  ```

- This command designates EOS.swi, on the switch flash, as the EOS software image load file.

  ```
  switch(config)#boot system flash:EOS.swi
  ```

  The **boot system** command above adds this line to ***boot-config***.

  ```
  SWI=flash:/EOS.swi
  ```

erase startup-config

The **erase startup-config** command erases or deletes the startup configuration.

Command Mode

Privileged EXEC

Command Syntax

erase startup-config [CONFIRMATION]

Parameters

- CONFIRMATION
  - <no parameter> the switch requires a confirmation before starting the erase.
  - **now** the erase begins immediately without prompting the user to confirm the request.

Examples

- This command deletes the startup configuration from the switch. When the **erase startup-config** command is entered, the switch sends a message prompting the user to confirm the **erase startup-config** request.

```
switch# erase startup-config
Proceed with erasing startup configuration? [confirm]y
switch#
```

- This command deletes the startup configuration from the switch immediately without prompting.

```
switch# erase startup-config now
switch#
```

protocol

The **protocol** command configures how the supervisors on a modular switch will handle switchover events. By default, the switch is set to route processor redundancy (RPR), which synchronizes ***startup-config*** files between the supervisor modules and partially boots the standby supervisor. The mode can also be set to simplex (manual switchover only) or to stateful switchover (SSO) which synchronizes both ***startup-config*** and ***running-config*** files between the supervisor modules and fully boots the standby module to speed the switchover process and minimize packet loss. Note that SSO synchronizes L2 state between the supervisors, but that L3 state is not synchronized. This can result in traffic loss for traffic forwarded on routes learned by a dynamic routing protocol. Enabling nonstop forwarding can eliminate most packet loss for BGP and OSPF.

The **no protocol** and **default protocol** commands set the redundancy protocol to the default value (**rpr**) by removing the **protocol** command from ***running-config***.

Command Mode

Redundancy Configuration

Command Syntax

```
protocol PROTOCOL_NAME
no protocol
default protocol
```

Parameters

- ***PROTOCOL_NAME*** specifies the location of the image file. Settings include
  - **rpr** route processor redundancy protocol (the default).
  - **simplex** no redundancy. Switchover must be initiated manually.
  - **sso** stateful switchover.

Related Commands

- **redundancy** Places switch in redundancy configuration mode.

Example

- These commands enter redundancy configuration mode and set the redundancy protocol to stateful switchover.

```
switch(config)#redundancy
switch(config-redundancy)#protocol sso
switch(config-redundancy)#
```

redundancy

The **redundancy** command places the switch in redundancy configuration mode.

Command Mode

Global Configuration

Command Syntax

redundancy

Commands Available in Redundancy Configuration Mode

- **protocol**

Related Commands

- **redundancy force-switchover** Manually initiates a switchover.

Example

- These commands enter redundancy configuration mode and set the redundancy protocol to stateful switchover.

```
switch(config)#redundancy
switch(config-redundancy)#protocol sso
switch(config-redundancy)#
```

redundancy force-switchover

The **redundancy force-switchover** command immediately switches control of the switch to the standby supervisor. If the redundancy mode is set to simplex or the standby supervisor is unavailable for any other reason, this command will not function.

Command Mode

Privileged EXEC

Command Syntax

redundancy force-switchover

Related Commands

- **redundancy** Places the switch in redundancy configuration mode.

Example

- This command forces a switchover to the standby supervisor. The switchover is executed immediately without further confirmation from the user.

```
switch#redundancy force-switchover
This supervisor will be restarted.
```

reload

The **reload** command power cycles the switch, then resets it under Aboot control. The hard reset clears the switch, including memory states and other hardware logic.

Important! The **reload** commands are used to initiate Accelerated Switch Update (ASU) and Smart Switch Update (SSU); for descriptions of these features and the appropriate command syntax, please refer to the Accelerated Software Upgrade (ASU) and Leaf Smart System Upgrade (Leaf SSU) sections.

- Fixed 1-RU systems: The power supply remains powered up through the reset. Power is removed from all other switch components for two to five seconds.
- Modular systems: The power supply on the active supervisor remains powered up through the reset. Power is removed from all other supervisor components for at least one second. After the supervisor becomes functional, it manages the power-cycling of all line cards.

Command Mode

Privileged EXEC

Command Syntax

`reload [TARGET] [CONFIRMATION]`

Parameters

- ***TARGET*** specifies which supervisor(s) will be reset. Some options are available only on dual-supervisor switches.
  - <no parameter> the active supervisor is reset.
  - **all** both supervisors are reset.
  - **peer** the peer supervisor is reset.
  - **power** the active supervisor is reset.
- ***CONFIRMATION*** specifies when the switch resets.
  - <no parameter> the switch requires a confirmation before starting the reset.
  - **now** the reset begins immediately without prompting the user to confirm the request.

Related Commands

- **reload <scheduled>** Schedules a pending reload operation.
- **show reload cause** Displays cause of most recent reload.

Example

- Begin the reboot process by typing the **reload** command:

```
switch#reload
```

When the **reload** command is entered, the switch sends a message prompting the user to save the configuration if it contains unsaved modifications, then asks the user to confirm the reload request.

```
System configuration has been modified. Save? [yes/no/cancel/diff]:n
Proceed with reload? [confirm]
```

The switch responds by broadcasting a series of messages, including a notification that the system is being rebooted, to all open CLI instances. The reload pauses to provide an option for the user to enter Aboot shell; the Aboot shell supports commands that restore the state of the internal flash to factory defaults or create a customized default state.

```
Broadcast message from root@mainStopping sshd: [  OK  ]
SysRq : Remount R/O
Restarting system

Aboot 1.9.0-52504.EOS2.0

Press Control-C now to enter Aboot shell
```

No action is required to continue the reset process. The switch displays messages to indicate the completion of individual tasks. The reboot is complete when the CLI displays a login prompt.

```
Booting flash:/EOS.swi
Unpacking new kernel
Starting new kernel
Switching to rooWelcome to Arista Networks EOS 4.4.0
Mounting filesystems:  [  OK  ]
Entering non-interactive startup
Starting EOS initialization stage 1: [  OK  ]
ip6tables: Applying firewall rules: [  OK  ]
iptables: Applying firewall rules: [  OK  ]
iptables: Loading additional modules: nf_conntrack_tftp [  OK  ]
Starting system logger: [  OK  ]
Starting system message bus: [  OK  ]
Starting NorCal initialization: [  OK  ]
Starting EOS initialization stage 2: [  OK  ]
Starting ProcMgr: [  OK  ]
Completing EOS initialization: [  OK  ]
Starting Power On Self Test (POST): [  OK  ]
Generating SSH2 RSA host key: [  OK  ]
Starting isshd: [  OK  ]
Starting sshd: [  OK  ]
Starting xinetd: [  OK  ]
[  OK  ] crond: [  OK  ]

switch login:
```

reload <scheduled>

The **reload <scheduled>** command configures the switch to reset at a specified time or after a specified interval. Refer to **reload** for details on the functional details of the reset operation.

The switch prompts to save the configuration and confirm the reload request. After the request is confirmed, the switch resumes normal operation until the reload initiates.

The **reload cancel**, **no reload**, and **default reload** commands cancel the pending reload operation.

Command Mode

Privileged EXEC

Command Syntax

```
reload [power] TIMEFRAME [COMMENT]
reload cancel
no reload
default reload
```

Parameters

- **power** Parameter has no functional affect.
- ***TIMEFRAME*** specifies when the switch resets.
  - **at** *absolute* specifies the time when a reset begins.
  - **in** *relative* specifies the period until the reset begins.

    absolute denotes time-date (24-hour notation): ***hh***:***mm*** [***month date***] or ***hh***:***mm*** [***date month***]
    relative designates a time period: ***hhh***:***mm***

- ***COMMENT*** descriptive text for denoting the reset reason. This option has no functional effect on the reset operation.
  - <no parameter> reason for system reset is not stated.
  - **reason** *comment_string* text that describes the reset.

Related Commands

- **reload** Initiate an immediate reload operation.
- **show reload** Displays time and reason of any pending reload operation.

Examples

- This command schedules a switch reset to begin in twelve hours.

```
switch#reload in 12:00
System configuration has been modified. Save? [yes/no/cancel/diff]:y
Proceed with reload? [confirm]
Reload scheduled for Tue Mar 27 05:57:25 2012 ( in 11 hours 59 minutes )
```

- This command cancels a scheduled switch reset.

```
switch#no reload
Scheduled reload has been cancelled
switch#
```

service sequence-numbers

The **service sequence-numbers** command causes the sequence numbers of syslog messages to be visible when the messages are displayed.

The **no service sequence-numbers** and **default service sequence-numbers** commands remove the **service sequence-numbers** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
service sequence-numbers
no service sequence-numbers
default service sequence-numbers
```

Examples

- This command enables sequence numbering that can been seen when syslog messages are displayed.

```
switch(config)#service sequence-numbers
switch(config)#
```

- To display the service sequence number, issue the **show logging** command.

```
switch#show logging
Syslog logging: enabled
    Buffer logging: level debugging
    Console logging: level informational
    Synchronous logging: disabled
    Trap logging: level informational
    Sequence numbers: enabled
    Syslog facility: local4
    Hostname format: Hostname only
    Repeat logging interval: disabled
             <-------OUTPUT OMITTED FROM EXAMPLE-------->
Log Buffer:
             <-------OUTPUT OMITTED FROM EXAMPLE-------->
Nov 12 14:03:34 switch1 SuperServer: 1: %SYS-7-CLI_SCHEDULER_LOG_STORED: Logfile
for scheduled CLI execution job 'tech-support' is stored in
flash:/schedule/tech-support/tech-support_2012-11-12.1402.log.gz
Nov 12 14:06:52 switch1 Cli: 2: %SYS-5-CONFIG_I: Configured from console by admin
on con0 (0.0.0.0)
Nov 12 14:07:26 switch1 Cli: 3: %SYS-5-CONFIG_E: Enter configuration mode from
console by admin on con0 (0.0.0.0)
Nov 12 14:14:29 switch1 Cli: 4: %SYS-5-CONFIG_I: Configured from console by admin
on con0 (0.0.0.0)
Nov 12 14:15:55 switch1 Cli: 5: %SYS-5-CONFIG_E: Enter configuration mode from
console by admin on con0 (0.0.0.0)
Nov 12 14:33:05 switch1 Cli: 6: %SYS-5-CONFIG_I: Configured from console by admin
on con0 (0.0.0.0)
Nov 12 14:45:13 switch1 Cli: 7: %SYS-5-CONFIG_E: Enter configuration mode from
console by admin on con0 (0.0.0.0)
switch#
```

show redundancy file-replication

The **show redundancy file-replication** command displays the status and last synchronization date of file replication between the supervisors on the switch.

Command Mode

EXEC

Command Syntax

show redundancy file-replication

Related Commands

- **show redundancy states** Displays status and redundancy protocol of supervisors.
- **show redundancy switchover sso** Displays stateful switchover information since last reload.

Example

- This command displays the current file replication status of the supervisors.

```
switch#show redundancy file-replication
0 files unsynchronized, 2 files synchronized, 0 files failed, 2 files total.

File                   Status         Last Synchronized
---------------------- -------------- ----------------------
file:persist/sys       Synchronized   25 days, 19:48:26 ago
flash:startup-config   Synchronized   25 days, 19:48:26 ago
switch#
```

show redundancy states

The **show redundancy states** command displays the current status (active or standby) and the configured redundancy protocol of both supervisors, as well as summary information about the latest switchover event.

Command Mode

EXEC

Command Syntax

show redundancy states

Related Commands

- **show redundancy file-replication** Displays status of file replication between supervisors.
- **show redundancy switchover sso** Displays stateful switchover information since last reload.

Example

- This command displays redundancy information for both supervisors and a summary of the latest switchover.

```
switch#show redundancy states
  my state = ACTIVE
peer state = STANDBY HOT
      Unit = Secondary
   Unit ID = 2

Redundancy Protocol (Operational) = Stateful Switchover
Redundancy Protocol (Configured) = Stateful Switchover
Communications = Up
Ready for switchover

  Last switchover time = 25 days, 19:51:34 ago
Last switchover reason = Other supervisor stopped sending heartbeats
```

show redundancy switchover sso

The **show redundancy switchover sso** command displays the number of stateful switchovers since the last reload and a log of the events in the latest stateful switchover.

Command Mode

EXEC

Command Syntax

show redundancy switchover sso

Related Commands

- **show redundancy file-replication** Displays status of file replication between supervisors.
- **show redundancy states** Displays status and redundancy protocol of supervisors.

Example

- This command displays stateful switchover information.

```
switch#show redundancy switchover sso
Total number of Stateful Switchover completed since reload: 4

Latest Stateful Switchover occured 29 days, 12:48:22 ago @ 2012-06-09 19:47:50
(completed)
  0.000000: switchover started
  0.000235: stage PCIEAcquired started
  0.000349:   event PCIEAcquired:__dummyInternal1__ completed
  0.000394:   event PCIEAcquired:PlxPcie-system started
  0.027738:   event PCIEAcquired:PlxPcie-system completed
  0.027829: stage PCIEAcquired is complete
  0.027935: stage DmaReady started
  0.028042:   event DmaReady:ForwardingAgent started
  0.079620:   event DmaReady:ForwardingAgent completed
  0.079699: stage DmaReady is complete
  0.079781: stage TimeCriticalServices started
  0.079887:   event TimeCriticalServices:__dummyInternal1__ completed
  0.079928:   event TimeCriticalServices:Stp started
  0.208035:   event TimeCriticalServices:Stp completed
  0.208120: stage TimeCriticalServices is complete
             <-------OUTPUT OMITTED FROM EXAMPLE-------->
 39.675076: stage NonCriticalServices started
 39.675145:   event NonCriticalServices:__dummyInternal1__ completed
 39.675183: stage NonCriticalServices is complete
 39.675399: switchover is complete
```

show reload

The **show reload** command displays the time and reason of any pending reload operation. The **reload <scheduled>** command schedules a reload operation and can be used to cancel a pending reload.

Command Mode

EXEC

Command Syntax

show reload

Related Commands

- **reload <scheduled>** Schedules a pending reload operation.
- **show reload cause** Displays cause of most recent reload.

Example

- These commands schedule a reload for 2:45 pm, display the time of the pending reload, then cancel the scheduled reload.

```
switch>reload at 14:45
Proceed with reload? [confirm]
Reload scheduled for Tue Mar 27 14:45:00 2012 ( in 4 hours 11 minutes )
switch#show reload
Reload scheduled for Tue Mar 27 14:45:00 2012 ( in 4 hours 11 minutes )
switch#reload cancel
Scheduled reload has been cancelled
switch>
```

show reload cause

The **show reload cause** command displays the reason of the most recent reload operation. The command displays recommended actions and debug information related to the executed reload.

Command Mode

EXEC

Command Syntax

show reload cause

Related Commands

- **reload** Initiates an immediate reload operation.
- **show reload** Displays time and reason of all pending reload operations.

Example

- This command displays the cause of the recent reload operation.

```
switch>show reload cause
Reload Cause 1:
------------------
Reload requested by the user.

Recommended Action:
------------------
No action necessary.

Debugging Information:
----------------------
None available.
switch>
```

Chapter 7

Upgrades and Downgrades

This chapter describes the procedures for upgrading or downgrading the switch software.

This chapter contains these sections:

- Section 7.1: Upgrade/Downgrade Overview
- Section 7.2: Accelerated Software Upgrade (ASU)
- Section 7.3: Leaf Smart System Upgrade (Leaf SSU)
- Section 7.4: Standard Upgrades and Downgrades
- Section 7.5: Upgrade/Downgrade Commands

7.1 Upgrade/Downgrade Overview

Upgrading or downgrading Arista switch software is accomplished by replacing the EOS image and reloading the switch. Depending on the switch model and the software change being made, there are different options for minimizing (or potentially eliminating) downtime and packet loss during the upgrade/downgrade.

Accelerated Software Upgrade (ASU): ASU is available on the 7050SX-64, 7050SX-128, 7050Q-32, and 7050Q-32S and can be used on both leaf and spine switches. It significantly reduces reload time by streamlining and optimizing the reload procedure for upgrades, and continues sending LACP PDUs while the CPU is rebooting, keeping port channels operational during the reload. Downtime during the upgrade is reduced to 30 seconds. Note: ASU does not support software downgrades.

Leaf Smart System Upgrade (Leaf SSU): SSU is available only on 7050X platforms (excluding 7050SX-72 and 7050SX-96), and can only be used on leaf switches. It includes the core functionality of ASU, plus additional elements that permit a hitless restart of several features. SSU does not support software downgrades, and is incompatible with VRRP.

Standard Upgrades and Downgrades: In those cases where an accelerated upgrade is not an option (such as software downgrades and unsupported platforms), performing a standard upgrade or downgrade using the steps described here will minimize downtime and packet loss.

Important! To upgrade the software on switches participating in an MLAG, see Section 12.3.3: Upgrading MLAG Peers.

7.2 Accelerated Software Upgrade (ASU)

The Accelerated Software Upgrade (ASU) process significantly decreases downtime and packet loss during a software upgrade in three ways:

- performing time-intensive tasks (including copying the EOS image) before rebooting the control plane
- forwarding packets in hardware (based on the last known state) while the control-plane is offline
- optimizing the boot process by performing only tasks essential for software upgrade

After the control plane has fully loaded, the data plane is restarted, causing approximately 30 seconds of downtime.

7.2.1 Upgrading the EOS image with Accelerated Software Upgrade

Using ASU to upgrade the active EOS image is a five-step process:

Step 1 Prepare switch for upgrade (Section 7.2.1.1).

Step 2 Transfer image file to the switch (Section 7.2.1.2). (Not required if desired file is on the switch).

Step 3 Modify ***boot-config*** file to point to the desired image file (Section 7.2.1.3).

Step 4 Start the ASU process (Section 7.2.1.4).

Step 5 Verify that switch is running the new image (Section 7.2.1.5).

7.2.1.1 Prepare the Switch

Before upgrading the EOS image, ensure that backup copies of the currently running EOS version and the ***running-config*** file are available in case of corruption during the upgrade process. To copy the ***running-config*** file, use the **copy running-config** command. In this example, ***running-config*** is copied to a file in the flash drive on the switch.

```
switch#copy running-config flash:/cfg_06162014
Copy completed successfully.
switch#
```

Determine the size of the new EOS image. Then verify that there is enough space available on the flash drive for *two* copies of this image (use the **dir** command to check the “bytes free” figure).

```
switch#dir flash:
Directory of flash:/
       -rwx   293168526            Nov 4 22:17   EOS4.11.0.swi
       -rwx          36            Nov 8 10:24   boot-config
       -rwx       37339            Jun 16 14:18  cfg_06162014



606638080 bytes total (602841088 bytes free)
```

Ensure that the switch has a management interface configured with an IP addresses and default gateway (see Assigning an IP Address to a Specific Ethernet Management Port and Configuring a Default Route to the Gateway), and confirm that it can be reached through the network by using the **show interfaces status** command and pinging the default gateway.

```
switch#show interfaces status
Port       Name             Status       Vlan        Duplex  Speed Type
Et3/1                       notconnect   1             auto   auto 1000BASE-T



Ma1/1                       connected    routed       unconf unconf Unknown

switch#ping 1.1.1.10
PING 172.22.26.1 (172.22.26.1) 72(100) bytes of data.
80 bytes from 1.1.1.10: icmp_seq=1 ttl=64 time=0.180 ms
80 bytes from 1.1.1.10: icmp_seq=2 ttl=64 time=0.076 ms
80 bytes from 1.1.1.10: icmp_seq=3 ttl=64 time=0.084 ms
80 bytes from 1.1.1.10: icmp_seq=4 ttl=64 time=0.073 ms
80 bytes from 1.1.1.10: icmp_seq=5 ttl=64 time=0.071 ms
```

7.2.1.2 Transfer the Image File

The target image must be copied to the file system on the switch, typically onto the flash drive. After verifying that there is space for two copies of the image, use the **copy** command to copy the image to the flash drive, then confirm that the new image file has been correctly transferred.

These command examples transfer an image file to the flash drive from various locations.

USB Memory

Command

`copy usb1:/sourcefile flash:/destfile`

Example

```
Sch#copy usb1:/EOS-4.14.4.swi flash:/EOS-4.14.4.swi
```

FTP Server

Command

`copy ftp:/ftp-source/sourcefile flash:/destfile`

Example

```
Sch#copy ftp:/user:password@10.0.0.3/EOS-4.14.4.swi flash:/EOS-4.14.4.swi
```

SCP

Command

`copy scp://scp-source/sourcefile flash:/destfile`

Example

```
sch#copy scp://user@10.1.1.8/user/EOS-4.14.4.swi flash:/EOS-4.14.4.swi
```

HTTP

Command

`copy http://http-source/sourcefile flash:/destfile`

Example

```
sch#copy http://10.0.0.10/EOS-4.14.4.swi flash:/EOS-4.14.4.swi
```

Once the file has been transferred, verify that it is present in the directory, then confirm the MD5 checksum using the **verify** command. The MD5 checksum is available from the EOS download page of the Arista website.

```
switch#dir flash:
Directory of flash:/
       -rwx   293168526            Nov 4 22:17   EOS4.14.2.swi
       -rwx          36            Nov 8 10:24   boot-config
       -rwx       37339            Jun 16 14:18  cfg_06162014
       -rwx   394559902            May 30 02:57  EOS-4.13.1.swi




606638080 bytes total (208281186 bytes free)
switch#53#verify /md5 flash:EOS-4.14.4.swi
verify /md5 (flash:EOS-4.14.4.swi) =c277a965d0ed48534de6647b12a86991
```

7.2.1.3 Modify boot-config

After transferring and confirming the desired image file, use the **boot system** command to update the ***boot-config*** file to point to the new EOS image.

This command changes the ***boot-config*** file to point to the image file located in flash memory at EOS-4.14.4.swi.

```
switch#configure terminal
switch(config)#boot system flash:/EOS-4.14.4.swi
```

Use the **show boot-config** command to verify that the boot-config file is correct:

```
switch(config)#show boot-config
Software image: flash:/EOS-4.14.4.swi
Console speed: (not set)
Aboot password (encrypted): $1$ap1QMbmz$DTqsFYeauuMSa7/Qxbi2l1
```

Save the configuration to the ***startup-config*** file with the **write** command.

```
switch#write
```

7.2.1.4 Start the ASU Process

After updating the ***boot-config*** file, start the ASU process using the **reload fast-boot** command to reload the switch and activate the new image. If ***running-config*** has not been saved, the CLI will prompt to save any modifications to the system configuration; failure to save modifications will abort the reload.

Note Once the system configuration is saved, there is a significant delay before the user is prompted to confirm the reload.

```
switch#reload fast-boot
System configuration has been modified. Save? [yes/no/cancel/diff]:y
Proceed with reload? [confirm]y
Proceeding with reload
```

7.2.1.5 Verify

After the switch finishes reloading, log into the switch and use the **show version** command to confirm the correct image is loaded. The ***Software image version*** line displays the version of the active image file.

```
switch#show version
Arista DCS-7150S-64-CL-F
Hardware version:    01.01
Serial number:       JPE13120819
System MAC address:  001c.7326.fd0c

Software image version: 4.14.4F
Architecture:           i386
Internal build version: 4.14.4F-1649184.4144F.2
Internal build ID:      eeb3c212-b4bd-4c19-ba34-1b0aa36e43f1

Uptime:                 14 hours and 48 minutes
Total memory:           4017088 kB
Free memory:            1569760 kB

switch>
```

7.3 Leaf Smart System Upgrade (Leaf SSU)

The Smart System Upgrade (SSU) process includes the core functionality of Accelerated Software Upgrade, plus additional optimizations that permit a hitless restart of several features. SSU leverages protocols capable of graceful restart to minimize traffic loss during upgrade. For protocols not capable of graceful restart, SSU generates control plane messages and buffers them in hardware to be slowly released when the control plane is offline. Additionally, under SSU, the forwarding ASIC does not get reset and ports do not flap.

Features capable of hitless restart under SSU include:

- QinQ
- 802.3ad Link Aggregation/LACP
- 802.3x flow control
- BGP (BGP graceful restart must be enabled: see Configuring BGP)
- MP-BGP (BGP graceful restart must be enabled: see Configuring BGP)
- 128-way Equal Cost Multipath Routing (ECMP)
- VRF
- route maps
- L2 MTU
- QoS

Important! SSU is not compatible with VRRP. If VRRP is configured on the switch, another upgrade method must be used.

7.3.1 Upgrading the EOS image with Smart System Upgrade

Using SSU to upgrade the active EOS image is a five-step process:

Step 1 Prepare switch for upgrade (Section 7.3.1.1).

Step 2 Transfer image file to the switch (Section 7.3.1.2). (Not required if desired file is on the switch).

Step 3 Modify ***boot-config*** file to point to the desired image file (Section 7.3.1.3).

Step 4 Start the SSU process (Section 7.3.1.4).

Step 5 Verify that the upgrade was successful (Section 7.3.1.5).

7.3.1.1 Prepare the Switch

Preparation of the switch for SSU includes:

- Backing Up Critical Software
- Making Room on the Flash Drive
- Verifying Connectivity
- Verifying Configuration
- Configuring BGP

Backing Up Critical Software

Before upgrading the EOS image, ensure that copies of the currently running EOS version and the ***running-config*** file are available in case of corruption during the upgrade process. To copy the ***running-config*** file, use the **copy running-config** command. In this example, ***running-config*** is copied to a file in the flash drive on the switch.

```
switch#copy running-config flash:/cfg_06162014
Copy completed successfully.
switch#
```

Making Room on the Flash Drive

Determine the size of the new EOS image. Then verify that there is enough space available on the flash drive for two copies of this image, plus a recommended 240MB (if available) for diagnostic information in case of a fatal error. Use the dir command to check the “bytes free” figure.

```
switch#dir flash:
Directory of flash:/
       -rwx   293168526            Nov 4 22:17   EOS4.11.0.swi
       -rwx          36            Nov 8 10:24   boot-config
       -rwx       37339            Jun 16 14:18  cfg_06162014



606638080 bytes total (602841088 bytes free)
```

Verifying Connectivity

Ensure that the switch has a management interface configured with an IP addresses and default gateway (see Assigning an IP Address to a Specific Ethernet Management Port and Configuring a Default Route to the Gateway), and confirm that it can be reached through the network by using the **show interfaces status** command and pinging the default gateway.

```
switch#show interfaces status
Port       Name             Status       Vlan        Duplex  Speed Type
Et3/1                       notconnect   1             auto   auto 1000BASE-T

          <-------OUTPUT OMITTED FROM EXAMPLE-------->

Ma1/1                       connected    routed       unconf unconf Unknown

switch#ping 1.1.1.10
PING 172.22.26.1 (172.22.26.1) 72(100) bytes of data.
80 bytes from 1.1.1.10: icmp_seq=1 ttl=64 time=0.180 ms
80 bytes from 1.1.1.10: icmp_seq=2 ttl=64 time=0.076 ms
80 bytes from 1.1.1.10: icmp_seq=3 ttl=64 time=0.084 ms
80 bytes from 1.1.1.10: icmp_seq=4 ttl=64 time=0.073 ms
80 bytes from 1.1.1.10: icmp_seq=5 ttl=64 time=0.071 ms
```

Verifying Configuration

Verify that the switch configuration is valid for SSU by using the **show reload hitless** command. If parts of the configuration are blocking execution of SSU, an error message will be displayed explaining what they are. For SSU to proceed, the configuration conflicts must be corrected before issuing the **reload hitless** command.

```
switch#show reload hitless
switch#'reload hitless' cannot proceed due to the following:
  Spanning-tree portfast is not enabled for one or more ports
  Spanning-tree BPDU guard is not enabled for one or more ports
switch#
```

Configuring BGP

For hitless restart of BGP and MP-BGP, BGP graceful restart must first be enabled using the **graceful-restart** command. The default restart time value (300 seconds) is appropriate for most configurations.

The BGP configuration mode in which the **graceful-restart** command is issued determines which BGP connections will restart gracefully.

- For all BGP connections, use the **graceful-restart** command in BGP configuration mode:

  ```
  switch#config
  switch(config)#router bgp 64496
  switch(config-router-bgp)#graceful-restart
  switch(config-router-bgp)#
  ```

- For all BGP connections in a specific VRF, use the **graceful-restart** command in BGP VRF configuration mode:

  ```
  switch#config
  switch(config)#router bgp 64496
  switch(config-router-bgp)#vrf purple
  switch(config-router-bgp-vrf-purple)#graceful-restart
  switch(config-router-bgp-vrf-purple)#exit
  switch(config-router-bgp)#
  ```

- For all BGP connections in a specific BGP address family, use the **graceful-restart** command in BGP address-family configuration mode:

  ```
  switch#config
  switch(config)#router bgp 64496
  switch(config-router-bgp)#address-family ipv6
  switch(config-router-bgp-af)#graceful-restart
  switch(config-router-bgp-af)#exit
  switch(config-router-bgp)#
  ```

BGP graceful restart can also be configured for a specific interface.

7.3.1.2 Transfer the Image File

The target image must be copied to the file system on the switch, typically onto the flash drive. After verifying that there is space for two copies of the image plus an optional 240MB for diagnostic information, use the **copy** command to copy the image to the flash drive, then confirm that the new image file has been correctly transferred.

These command examples transfer an image file to the flash drive from various locations.

USB Memory

Command

```
copy usb1:/sourcefile flash:/destfile
```

Example

```
Sch#copy usb1:/EOS-4.14.4.swi flash:/EOS-4.14.4.swi
```

FTP Server

Command

```
copy ftp:/ftp-source/sourcefile flash:/destfile
```

Example

```
Sch#copy ftp:/user:password@10.0.0.3/EOS-4.14.4.swi flash:/EOS-4.14.4.swi
```

SCP

Command

```
copy scp://scp-source/sourcefile flash:/destfile
```

Example

```
sch#copy scp://user@10.1.1.8/user/EOS-4.14.4.swi flash:/EOS-4.14.4.swi
```

HTTP

Command

```
copy http://http-source/sourcefile flash:/destfile
```

Example

```
sch#copy http://10.0.0.10/EOS-4.14.4.swi flash:/EOS-4.14.4.swi
```

Once the file has been transferred, verify that it is present in the directory, then confirm the MD5 checksum using the **verify** command. The MD5 checksum is available from the EOS download page of the Arista website.

```
switch#dir flash:
Directory of flash:/
       -rwx   293168526           Nov 4 22:17   EOS4.14.2.swi
       -rwx          36           Nov 8 10:24   boot-config
       -rwx       37339           Jun 16 14:18  cfg_06162014
       -rwx   394559902           May 30 02:57  EOS-4.13.1.swi


       <-------OUTPUT OMITTED FROM EXAMPLE-------->

606638080 bytes total (208281186 bytes free)
switch#53#verify /md5 flash:EOS-4.14.4.swi
verify /md5 (flash:EOS-4.14.4.swi) =c277a965d0ed48534de6647b12a86991
```

7.3.1.3 Modify boot-config

After transferring and confirming the desired image file, use the **boot system** command to update the ***boot-config*** file to point to the new EOS image.

This command changes the ***boot-config*** file to point to the image file located in flash memory at EOS-4.14.4.swi.

```
switch#configure terminal
switch(config)#boot system flash:/EOS-4.14.4.swi
```

Use the **show boot-config** command to verify that the boot-config file is correct:

```
switch(config)#show boot-config
Software image: flash:/EOS-4.14.4.swi
Console speed: (not set)
Aboot password (encrypted): $1$ap1QMbmz$DTqsFYeauuMSa7/Qxbi2l1
```

Save the configuration to the ***startup-config*** file with the **write** command.

```
switch#write
```

7.3.1.4 Start the SSU Process

After updating the ***boot-config*** file, verify that your configuration supports SSU (if you have not already done so) by using the **show reload hitless** command. If parts of the configuration are blocking execution of SSU, an error message will be displayed explaining what they are.

```
switch#show reload hitless
switch#'reload hitless' cannot proceed due to the following:
  Spanning-tree portfast is not enabled for one or more ports
  Spanning-tree BPDU guard is not enabled for one or more ports
```

Then start the SSU process using the **reload hitless** command to reload the switch and activate the new image. The CLI will identify any changes that must be made to the configuration before starting SSU, prompt to save any modifications to the system configuration, and request confirmation before reloading.

```
switch#reload hitless
System configuration has been modified. Save? [yes/no/cancel/diff]:y
Copy completed successfully.
   Proceed with reload? [confirm]y
```

Important! Any configuration changes must be saved for SSU to continue. However, once the upgrade has begun, no changes should be made to the configuration until the “LAUNCHER-6-BOOT_STATUS: 'reload hitless' reconciliation complete.” syslog message has been generated by the switch.

7.3.1.5 Verify Success of the Upgrade

Before making any configuration changes to the switch after reload, verify that the SSU process is complete using the command **show boot stages log.** If the process is complete, the last message should be “Asu Hitless boot stages complete.”

```
switch#show boot stages log
Timestamp           Delta Begin Msg
2015-03-28 15:18:30 000.000000 Asu Hitless boot stages started
2015-03-28 15:18:30 000.069732 stage CriticalAgent started
2015-03-28 15:18:30 000.069811   event CriticalAgent:SuperServer completed

2015-03-28 15:20:20 110.224504 stage BootSanityCheck is complete
2015-03-28 15:20:20 110.225439 Asu Hitless boot stages complete
switch#
```

Completion of the SSU process may also be verified by checking the syslog for the following message:

```
LAUNCHER-6-BOOT_STATUS: 'reload hitless' reconciliation complete
```

To verify whether the SSU upgrade was successful, use the **show reload cause** command. If a fatal error occurred during the upgrade process, the switch will have completely rebooted and the fatal error will be displayed along with the directory in which diagnostic information can be found. If the SSU upgrade succeeded, it will read “Hitless reload requested by the user.”

Fatal Error Display

```
switch#show reload cause
Reload Cause 1:
------------------
Reload requested by the user.

Reload Time:
------------
Reload occurred at Sat Feb 28 02:34:26 2015 PST.

Recommended Action:
-------------------
No action necessary.

Debugging Information:
----------------------
None available.

Reload Cause 2:
-------------------
Fatal error during 'reload hitless'. (stageMgr - LinkStatusUpdate timed out)

Reload Time:
------------
Reload occurred at Sat Feb 28 02:33:54 2015 PST.

Recommended Action:
-------------------
A fatal error occurred during hitless reload.
If the problem persists, contact your customer support representative.

Debugging Information:
----------------------
/mnt/flash/persist/fatalError-2015-02-28_023355
switch#
```

Successful Upgrade Display

```
switch#show reload cause
Reload Cause 1:
------------------
Hitless reload requested by the user.

Reload Time:
------------
Reload occurred at Wed Mar 25 14:49:04 2015 PDT.

Recommended Action:
-------------------
No action necessary.

Debugging Information:
----------------------
None available.
switch#
```

The **show version** command will confirm whether the correct image is loaded. The ***Software image version*** line displays the version of the active image file.

```
switch#show version
Arista DCS-7050QX-32-F
Hardware version:    02.00
Serial number:       JPE14071098
System MAC address:  001c.7355.556f

Software image version: 4.14.5F-2353054.EOS4145F
Architecture:           i386
Internal build version: 4.14.5F-2353054.EOS4145F
Internal build ID:      e8748ea7-916d-4217-878f-4bfe2adc7122

Uptime:                 4 minutes
Total memory:           3981328 kB
Free memory:            1342408 kB


switch#
```

Important! If a fatal error occurs during the SSU process, the new EOS image will still be loaded and booted.

7.4 Standard Upgrades and Downgrades

Standard software upgrades and downgrades on Arista switches are accomplished by installing a different EOS image and reloading the switch. On switches with redundant supervisors, the EOS image must be installed on both supervisors. Using the procedure described below will minimize packet loss during a standard upgrade or downgrade.

These sections describe standard switch upgrade and downgrade procedures

- Section 7.4.1: Upgrading or Downgrading the EOS on a Single-Supervisor Switch
- Section 7.4.2: Upgrading or Downgrading the EOS on a Dual-Supervisor Switch

7.4.1 Upgrading or Downgrading the EOS on a Single-Supervisor Switch

Modifying the active EOS image is a five-step process:

Step 1 Prepare switch for upgrade (Section 7.4.1.1).

Step 2 Transfer image file to the switch (Section 7.4.1.2). (Not required if desired file is on the switch).

Step 3 Modify ***boot-config*** file to point to the desired image file (Section 7.4.1.3).

Step 4 Reload switch (Section 7.4.1.4).

Step 5 Verify that switch is running the new image (Section 7.4.1.5).

7.4.1.1 Prepare the Switch

Before upgrading the EOS image, ensure that backup copies of the currently running EOS version and the ***running-config*** file are available in case of corruption during the upgrade process. To copy the ***running-config*** file, use the **copy running-config** command. In this example, ***running-config*** is copied to a file in the flash drive on the switch.

```
switch#copy running-config flash:/cfg_06162014
Copy completed successfully.
switch#
```

Determine the size of the new EOS image and verify that there is enough space available for ***two copies*** of it on the flash drive, using the **dir** command to check the “bytes free” figure. The EOS boot process makes a copy of the .swi image file to the internal flash, and the switch will boot to the Aboot prompt if there is insufficient room for both copies.

```
switch#dir flash:
Directory of flash:/
       -rwx   293168526           Nov 4 22:17   EOS4.11.0.swi
       -rwx          36           Nov 8 10:24   boot-config
       -rwx       37339           Jun 16 14:18  cfg_06162014

          <-------OUTPUT OMITTED FROM EXAMPLE-------->

606638080 bytes total (602841088 bytes free)
```

Ensure that the switch has a management interface configured with an IP addresses and default gateway (see Assigning an IP Address to a Specific Ethernet Management Port and Configuring a Default Route to the Gateway), and confirm that it can be reached through the network by using the

show interfaces status command and pinging the default gateway. To configure a virtual IP address to access the active supervisor on a modular switch, see also Assigning a Virtual IP Address to Access the Active Ethernet Management Port.

```
switch#show interfaces status
Port       Name              Status       Vlan        Duplex  Speed Type
Et3/1                        notconnect   1             auto   auto 1000BASE-T

          <-------OUTPUT OMITTED FROM EXAMPLE-------->

Ma1/1                        connected    routed       unconf unconf Unknown

switch#ping 1.1.1.10
PING 172.22.26.1 (172.22.26.1) 72(100) bytes of data.
80 bytes from 1.1.1.10: icmp_seq=1 ttl=64 time=0.180 ms
80 bytes from 1.1.1.10: icmp_seq=2 ttl=64 time=0.076 ms
80 bytes from 1.1.1.10: icmp_seq=3 ttl=64 time=0.084 ms
80 bytes from 1.1.1.10: icmp_seq=4 ttl=64 time=0.073 ms
80 bytes from 1.1.1.10: icmp_seq=5 ttl=64 time=0.071 ms
```

7.4.1.2 Transfer the Image File

The target image must be copied to the file system on the switch, typically onto the flash drive. After verifying that there is space for the image, use the CLI **copy** command to copy the image to the flash drive, then confirm that the new image file has been correctly transferred.

These command examples transfer an image file to the flash drive from various locations.

USB Memory

Command

`copy usb1:/sourcefile flash:/destfile`

Example

```
Sch#copy usb1:/EOS-4.13.2.swi flash:/EOS-4.13.2.swi
```

FTP Server

Command

`copy ftp:/ftp-source/sourcefile flash:/destfile`

Example

```
sch#copy ftp:/user:password@10.0.0.3/EOS-4.13.2.swi flash:/EOS-4.13.2.swi
```

SCP

Command

`copy scp://scp-source/sourcefile flash:/destfile`

Example

```
sch#copy scp://user:password@10.1.1.8/user/EOS-4.13.2.swi flash:/EOS-4.13.2.swi
```

HTTP

Command

`copy http://http-source/sourcefile flash:/destfile`

Example

```
sch#copy http://10.0.0.10/EOS-4.13.2.swi flash:/EOS-4.13.2.swi
```

Once the file has been transferred, verify that it is present in the directory, then confirm the MD5 checksum using the **verify** command. The MD5 checksum is available from the EOS download page of the Arista website.

```
switch#dir flash:
Directory of flash:/
       -rwx   293168526            Nov 4 22:17   EOS4.11.0.swi
       -rwx          36            Nov 8 10:24   boot-config
       -rwx       37339            Jun 16 14:18  cfg_06162014
       -rwx   394559902            May 30 02:57  EOS-4.12.2.swi


          <-------OUTPUT OMITTED FROM EXAMPLE-------->

606638080 bytes total (208281186 bytes free)
switch#53#verify /md5 flash:EOS-4.13.2.swi
verify /md5 (flash:EOS-4.13.2.swi) =c277a965d0ed48534de6647b12a86991
```

7.4.1.3 Modify boot-config

After transferring and confirming the desired image file, use the **boot system** command to update the ***boot-config*** file to point to the new EOS image.

This command changes the ***boot-config*** file to point to the image file located in flash memory at EOS-4.12.2.swi.

```
switch#configure terminal
switch(config)#boot system flash:/EOS-4.13.2.swi
```

Use the **show boot-config** command to verify that the boot-config file is correct:

```
switch(config)#show boot-config
Software image: flash:/EOS-4.13.2.swi
Console speed: (not set)
Aboot password (encrypted): $1$ap1QMbmz$DTqsFYeauuMSa7/Qxbi2l1
```

Save the configuration to the ***startup-config*** file with the **write** command.

```
switch#write
```

7.4.1.4 Reload

After updating the ***boot-config*** file, reset the switch to activate the new image. The **reload** command resets the switch, resulting in temporary downtime and packet loss on single supervisor switches.

When reloading from the console port, all rebooting messages are displayed on the terminal. From any port except the console, the CLI displays this text:

```
switch#reload
The system is going down for reboot NOW!
```

Important! The EOS boot process makes a copy of the .swi image file in the internal flash while booting, so sufficient space for ***two copies*** must be present when loading the new EOS image. If the switch is reloaded without adequate space on the flash drive, it will boot to the Aboot prompt from which you can delete files from /mnt/flash to free up additional space. Exiting Aboot will begin the boot process again.

7.4.1.5 Verify

After the switch finishes reloading, log into the switch and use the **show version** command to confirm the correct image is loaded. The ***Software image version*** line displays the version of the active image file.

```
switch#show version
Arista DCS-7150S-64-CL-F
Hardware version:    01.01
Serial number:       JPE13120819
System MAC address:  001c.7326.fd0c

Software image version: 4.13.2F
Architecture:           i386
Internal build version: 4.13.2F-1649184.4132F.2
Internal build ID:      eeb3c212-b4bd-4c19-ba34-1b0aa36e43f1

Uptime:                 14 hours and 48 minutes
Total memory:           4017088 kB
Free memory:            1569760 kB

switch>
```

7.4.2 Upgrading or Downgrading the EOS on a Dual-Supervisor Switch

Modifying the active EOS image is a four-step process:

Step 1 Prepare switch for upgrade (Section 7.4.2.1).

Step 2 Transfer image file to primary supervisor (Section 7.4.2.2). (Not required if desired file is on switch)

Step 3 Use the **install** command to install the new EOS image and update ***boot-config*** (Section 7.4.2.3).

Step 4 Verify that the switch is running the new image (Section 7.4.2.4).

Important! Due to a change in the supervisor heartbeat timeout, booting one supervisor with a post-SSO image (version 4.10.0-SSO, 4.11.X and later) while the other supervisor is running a pre-SSO image will cause the supervisor running the pre-SSO image to reload. This will cause a disruption as both supervisors will be inactive for a short time. To minimize downtime, upgrade the images on both supervisors and reload the entire chassis using the **install** command.

7.4.2.1 Prepare the Switch

To prepare the switch for an EOS upgrade, take the following steps:

- Back up essential files.
- Ensure that you are logged in to the primary supervisor.
- Ensure that both supervisors are reachable and have management interfaces configured.
- Ensure that there is enough room on both supervisors for the new image file.
- Ensure that any extensions running on the active supervisor are also available on the standby.

Before upgrading the EOS image, ensure that backup copies of the currently running EOS version and the ***running-config*** file are available in case of corruption during the upgrade process. To copy the ***running-config*** file, use the **copy running-config** command. In this example, ***running-config*** is being copied to a file called “backup2” on the flash drive.

```
switch#copy running-config backup2
switch#
```

Ensure that you are logged in to the primary supervisor, not the standby. Use the **show redundancy states** command, and verify that ***my state*** reads “ACTIVE” and not “STANDBY.”

```
switch#show redundancy states
  my state = ACTIVE
peer state = STANDBY HOT
      Unit = Secondary
   Unit ID = 1

Redundancy Protocol (Operational) = Stateful Switchover
Redundancy Protocol (Configured) = Stateful Switchover
Communications = Up
Ready for switchover

  Last switchover time = 25 days, 19:51:34 ago
Last switchover reason = Other supervisor stopped sending heartbeats
```

Ensure that each supervisor has a management interface configured with an IP addresses and default gateway (see Assigning a Virtual IP Address to Access the Active Ethernet Management Port and Configuring a Default Route to the Gateway), and confirm that both management interfaces can be reached through the network by using the **show interfaces status** command and pinging the default gateway.

```
switch#show interfaces status
Port       Name              Status       Vlan        Duplex  Speed Type
Et3/1                        notconnect   1             auto   auto 1000BASE-T



Ma1/1                        connected    routed       unconf unconf Unknown
Ma2/1                        connected    routed     a-full  a-100M 10/100/1000

switch#ping 1.1.1.10
PING 172.22.26.1 (172.22.26.1) 72(100) bytes of data.
80 bytes from 1.1.1.10: icmp_seq=1 ttl=64 time=0.180 ms
80 bytes from 1.1.1.10: icmp_seq=2 ttl=64 time=0.076 ms
80 bytes from 1.1.1.10: icmp_seq=3 ttl=64 time=0.084 ms
80 bytes from 1.1.1.10: icmp_seq=4 ttl=64 time=0.073 ms
80 bytes from 1.1.1.10: icmp_seq=5 ttl=64 time=0.071 ms
```

Determine the size of the new EOS image and verify that there is space available for it on the flash drive of both supervisors, using the **dir** command to check the “bytes free” figure.

Primary supervisor:

```
switch#dir flash:
Directory of flash:/
       -rwx   293168526           Nov 4 22:17   EOS4.11.0.swi
       -rwx          36           Nov 8 10:24   boot-config
       -rwx       37339           Jun 16 14:18  cfg_06162014


          <-------OUTPUT OMITTED FROM EXAMPLE-------->

606638080 bytes total (602841088 bytes free)
```

Standby supervisor:

```
switch#dir supervisor-peer:mnt/flash/
Directory of flash:/
       -rwx   293168526           Nov 4 22:17   EOS4.11.0.swi
       -rwx          36           Nov 8 10:24   boot-config
       -rwx       37339           Jun 16 14:18  cfg_06162014


          <-------OUTPUT OMITTED FROM EXAMPLE-------->

606638080 bytes total (602841088 bytes free)
```

And, finally, ensure that any extensions running on the primary supervisor are also available on the secondary supervisor.

7.4.2.2 Transfer the Image File to the Primary Supervisor

Load the desired image to the file system on the primary supervisor, typically into the flash. Use the CLI **copy** command to load files to the flash on the primary supervisor, then confirm that the new image file has been correctly transferred.

These command examples transfer an image file to flash from various locations.

USB Memory

Command

`copy usb1:/sourcefile flash:/destfile`

Example

```
Sch#copy usb1:/EOS-4.13.2.swi flash:/EOS-4.13.2.swi
```

FTP Server

Command

`copy ftp:/ftp-source/sourcefile flash:/destfile`

Example

```
Sch#copy ftp:/user:password@10.0.0.3/EOS-4.13.2.swi flash:/EOS-4.13.2.swi
```

SCP

Command

`copy scp://scp-source/sourcefile flash:/destfile`

Example

```
sch#copy scp://user:password@10.1.1.8/user/EOS-4.13.2.swi flash:/EOS-4.13.2.swi
```

HTTP

Command

```
copy http://http-source/sourcefile flash:/destfile
```

Example

```
sch#copy http://10.0.0.10/EOS-4.13.2.swi flash:/EOS-4.13.2.swi
```

Once the file has been transferred, verify that it is present in the directory, then confirm the MD5 checksum using the **verify** command. The MD5 checksum for each available image can be found on the EOS download page of the Arista website.

```
switch#dir flash:
Directory of flash:/
       -rwx   293168526            Nov 4 22:17   EOS4.11.0.swi
       -rwx          36            Nov 8 10:24   boot-config
       -rwx       37339            Jun 16 14:18  cfg_06162014
       -rwx   394559902            May 30 02:57  EOS-4.12.2.swi


          <-------OUTPUT OMITTED FROM EXAMPLE-------->

606638080 bytes total (208281186 bytes free)
switch#53#verify /md5 flash:EOS-4.13.2.swi
verify /md5 (flash:EOS-4.13.2.swi) =c277a965d0ed48534de6647b12a86991
```

7.4.2.3 Install the New EOS Image

Once the EOS image has been copied to the flash drive of the primary supervisor, use the **install** command to update the ***boot-config***, copy the new image to the secondary supervisor and reload both supervisors. When upgrading to a new image, both supervisors will briefly be unavailable; using the **install** command minimizes packet loss during reload.

```
switch(config)#install source EOS-4.13.2.swi reload
Preparing new boot-config... done.
Copying new software image to standby supervisor... done.
Copying new boot-config to standby supervisor... done.
Committing changes on standby supervisor... done.
Reloading standby supervisor... done.
Committing changes on this supervisor... done.
Reloading this supervisor...
```

7.4.2.4 Verify the New Image

After the switch finishes reloading, log into the switch and use the **show version** command to confirm the correct image is loaded. The ***Software image version*** line displays the version of the active image file.

```
switch#show version
Arista DCS-7504
Hardware version:    01.01
Serial number:       JPE13120819
System MAC address:  001c.7326.fd0c

Software image version: 4.13.2F
Architecture:           i386
Internal build version: 4.13.2F-1649184.4132F.2
Internal build ID:      eeb3c212-b4bd-4c19-ba34-1b0aa36e43f1

Uptime:                 1 hour and 36 minutes
Total memory:           4017088 kB
Free memory:            1473280 kB

switch#
```

7.5 Upgrade/Downgrade Commands

- install
- reload fast-boot
- reload hitless

install

The **install** command copies the specified EOS image onto the switch (if the source is external), configures the ***boot-config*** file to point to the specified EOS image, copies the image to the standby supervisor (on dual-supervisor switches), and optionally reloads the switch to run the new EOS.

Command Mode

Privileged EXEC

Command Syntax

install source *source_path* **[destination** *destination_path***] [now] [reload]**

Parameters

- *source_path* file path and name of EOS image. If no file path is specified, the switch will look for the image on the flash drive of the primary supervisor.
- **destination** *destination_path* destination file path and name of the EOS image. If no destination or name is specified, the EOS image will be stored on the flash drive with its original file name.
- **now** command is executed immediately without further prompts.
- **reload** supervisor is reloaded after the image and updated ***boot-config*** file are installed. On dual-supervisor switches, reloads both supervisors, after which control is returned to the primary supervisor.

Example

- This command updates the ***boot-config*** file to point to the EOS.swi file on the primary supervisor's flash drive, copies the image and ***boot-config*** file to the secondary supervisor, and reboots both.

```
switch(config)#install source EOS.swi reload
Preparing new boot-config... done.
Copying new software image to standby supervisor... done.
Copying new boot-config to standby supervisor... done.
Committing changes on standby supervisor... done.
Reloading standby supervisor... done.
Committing changes on this supervisor... done.
Reloading this supervisor...
```

reload fast-boot

The **reload fast-boot** command starts the Accelerated Software Upgrade (ASU) process using the EOS image specified by the ***boot-config*** file (configured by the **boot system** command).

ASU significantly decreases downtime and packet loss during a software upgrade, but the data plane is still restarted after the control plane has loaded, resulting in approximately 30 seconds of downtime. If available, Arista recommends using Smart System Upgrade (SSU) instead.

ASU shortens downtime and minimizes packet loss during EOS upgrades in three ways:

- performing time-intensive tasks (including copying the EOS image) before rebooting the control plane
- forwarding packets in hardware (based on the last known good state) while the control-plane is offline
- optimizing the boot process by performing only tasks essential for software upgrade

Command Mode

Privileged EXEC

Command Syntax

```
reload fast-boot
```

Guidelines

- ASU is supported only for upgrades (not downgrades).
- ASU is not supported if the EOS upgrade requires an FPGA upgrade.
- Enough free space must be available on the flash drive to store two copies of the target EOS image.

Example

- This command starts the Accelerated Software Upgrade process.

```
switch#reload fast-boot
Proceed with reload? [confirm]
```

When the **reload fast-boot** command is entered, the switch sends a message prompting the user to save the configuration if it contains unsaved modifications, then asks the user to confirm the reload request.

reload hitless

The **reload hitless** command starts the Smart System Upgrade (SSU) process using the EOS image specified by the ***boot-config*** file (configured by the **boot system** command).

Command Mode

Privileged EXEC

Command Syntax

```
reload hitless
```

Guidelines

- SSU is supported only for upgrades (not downgrades).
- SSU is not supported if the EOS upgrade requires an FPGA upgrade.
- Enough free space must be available on the flash drive to store two copies of the target EOS image. It is also recommended that an *additional* 240MB be available to store diagnostic information.

Example

- This command starts the SSU process.

```
switch#reload hitless
Proceed with reload? [confirm]
```

If there are issues with the current switch configuration that will prevent SSU from being performed, the switch lists the changes that must be made before SSU can begin.

```
switch#reload hitless
switch#'reload hitless' cannot proceed due to the following:
  Spanning-tree portfast is not enabled for one or more ports
  Spanning-tree BPDU guard is not enabled for one or more ports
switch#
```

When the **reload hitless** command is entered, the switch sends a message prompting the user to save the configuration if it contains unsaved modifications, then asks the user to confirm the reload request.

```
switch#reload hitless
System configuration has been modified. Save? [yes/no/cancel/diff]:y
Copy completed successfully.
Proceed with reload? [confirm]y
```

Chapter 8

Switch Environment Control

The following sections describe the commands that display temperature, fan, and power supply status:

- Section 8.1: Environment Control Introduction
- Section 8.2: Environment Control Overview
- Section 8.3: Configuring and Viewing Environment Settings
- Section 8.4: Environment Commands

The switch chassis, fans, power supplies, line cards, and supervisors also provide LEDs that signal status and conditions that require attention. The Quick Start Guide for the individual switches provides information about their LEDs.

8.1 Environment Control Introduction

Arista Networks switching platforms are designed to work reliably in common data center environments. To ensure their reliable operation and to monitor or diagnose the switch's health, Arista provides a set of monitoring capabilities available through the CLI or SNMP entity MIBs to monitor and diagnose potential problems with the switching platform.

8.2 Environment Control Overview

8.2.1 Temperature

Arista switches include internal temperature sensors. The number and location of the sensors vary with each switch model. Each sensor is assigned temperature thresholds that denote alert and critical conditions. Temperatures that exceed the threshold trigger the following:

- **Alert Threshold**: All fans run at maximum speed and a warning message is logged.
- **Critical Threshold**: The component is shut down immediately and its Status LED flashes orange.

In modular systems, cards are shut down when their temperatures exceed the critical threshold. The switch is shut down if the temperature remains above the critical threshold for three minutes.

8.2.2 Fans

Arista switches include fan modules that maintain internal components at proper operating temperatures. The number and type of fans vary with switch chassis type:

- **Fixed configuration switches** contain hot-swappable independent fans. Fan models with different airflow directions are available. All fans within a switch must have the same airflow direction.
- **Modular switches** contain independent fans that circulate air from front-to-rear panel. Power supplies for modular switches also include fans that cool the power supply and supervisors.

The switch operates normally when one fan is not operating. Non-functioning modules should not be removed from the switch unless they are immediately replaced; adequate switch cooling requires the installation of all components, including a non-functional fan.

Two non-operational fans trigger an ***insufficient fan shutdown*** condition. Under normal operations, this condition initiates a switch power down procedure.

Fans are accessible from the rear panel.

8.2.2.1 Power

Arista switches contain power supplies which provide power to internal components.

- **Fixed configuration switches** contain two power supplies, providing 1+1 redundancy.
- **Modular switches** contain four power supplies, providing a minimum of 2+2 redundancy.

Power supply LED indicators are visible from the rear panel.

8.3 Configuring and Viewing Environment Settings

8.3.1 Overriding Automatic Shutdown

8.3.1.1 Overheating

The switch can be configured to continue operating during temperature shutdown conditions. Ignoring a temperature shutdown condition is strongly discouraged because operating at high temperatures can damage the switch and void the warranty.

Temperature shutdown condition actions are specified by the **environment overheat action** command. The switch displays this warning when configured to ignore shutdown temperature conditions.

```
Switch(config)#environment overheat action ignore
======================================================================
WARNING: Overriding the system shutdown behavior when the system
is overheating is unsupported and should only be done under
the direction of an Arista Networks engineer. You risk damaging
hardware by not shutting down the system in this situation, and doing
so without direction from Arista Networks can be grounds for voiding
your warranty. To re-enable the shutdown-on-overheat behavior, use
the 'environment overheat action shutdown' command.
======================================================================
Switch(config)#
```

The ***running-config*** contains the **environment overheat action** command when it is set to **ignore**. When the command is not in ***running-config***, the switch shuts down when an overheating condition exists.

The following ***running-config*** file lists the **environment overheat action** command.

```
switch#show running-config
! Command: show running-config
! device: switch (DCS-7150S-64-CL, EOS-4.13.2F)


ip route 0.0.0.0/0 10.255.255.1
!
environment overheat action ignore
!
!
end
switch#
```

8.3.1.2 Insufficient Fans

The switch can be configured to ignore the ***insufficient fan shutdown*** condition. This is strongly discouraged because continued operation without sufficient cooling may lead to a critical temperature condition that can damage the switch and void the warranty.

Insufficient-fans shutdown override is configured by the **environment insufficient-fans action** command. The switch displays this warning when configured to ignore insufficient-fan conditions.

```
Switch(config)#environment insufficient-fans action ignore
======================================================================
WARNING: Overriding the system shutdown behavior when the system
has insufficient fans inserted is unsupported and should only be done
under
the direction of an Arista Networks engineer. You risk damaging
hardware by not shutting down the system in this situation, and doing
so without direction from Arista Networks can be grounds for voiding
your warranty. To re-enable the shutdown-on-overheat behavior, use
the 'environment insufficient-fans action shutdown' command.
======================================================================
Switch(config)#
```

The ***running-config*** contains the **environment insufficient-fans action** command when it is set to ***ignore***. When ***running-config*** does not contain this command, the switch shuts down when it detects an insufficient-fans condition.

8.3.1.3 Fan Speed

The switch can be configured to override the automatic fan speed. The switch normally controls the fan speed to maintain optimal operating temperatures. The fans can be configured to operate at a constant speed regardless of the switch temperature conditions.

Fan speed override is configured by the **environment fan-speed** command. The switch displays this warning when its control of fan speed is overridden.

```
switch(config)#environment fan-speed override 50
======================================================================
WARNING: Overriding the system fan speed is unsupported and should only
be done under the direction of an Arista Networks engineer.
You can risk damaging hardware by setting the fan speed too low
and doing so without direction from Arista Networks can be grounds
for voiding your warranty.
To set the fan speed back to automatic mode, use the
'environment fan-speed auto' command
======================================================================
switch(config)#
```

The ***running-config*** contains the **environment fan-speed override** command if it is set to override. When ***running-config*** does not contain this command, the switch controls the fan speed.

8.3.2 Viewing Environment Status

8.3.2.1 Temperature Status

To display internal temperature sensor status, enter **show environment temperature**.

```
switch>show environment temperature
System temperature status is: Ok
                                                                    Alert   Critical
Sensor  Description                          Temperature  Threshold  Threshold
------- ------------------------------------ ------------ ---------- ----------
1       Front-panel temp sensor                 22.000C        65C        75C
2       Fan controller 1 sensor                 23.000C        75C        85C
3       Fan controller 2 sensor                 28.000C        75C        85C
4       Switch chip 1 sensor                    40.000C       105C       115C
5       VRM 1 temp sensor                       48.000C       105C       110C
switch>
```

System temperature status is the first line that the command displays. **System temperature status** values indicate the following:

- ***Ok:*** All sensors report temperatures below the alert threshold.
- ***Overheating:*** At least one sensor reports a temperature above its alert threshold.
- ***Critical:*** At least one sensor reports a temperature above its critical threshold.
- ***Unknown:*** The switch is initializing.
- ***Sensor Failed:*** At least one sensor is not functioning.

8.3.2.2 Fans

The **show environment cooling** command displays the cooling and fan status.

Example

This command displays the fan and cooling status.

```
switch>show environment cooling
System cooling status is: Ok
Ambient temperature: 22C
Airflow: front-to-back
Fan Tray  Status           Speed
--------- ---------------- ------
1         Ok                 35%
2         Ok                 35%
3         Ok                 35%
4         Ok                 35%
5         Ok                 35%
switch>
```

8.3.2.3 Power

The **show environment power** command displays the status of the power supplies.

Example

- This command displays the status of the power supplies:

```
switch>show environment power
Power                                    Input    Output   Output
Supply  Model                Capacity  Current  Current  Power    Status
------- -------------------- --------- -------- -------- -------- -------------
1       PWR-650AC                 650W    0.44A   10.50A   124.0W Ok
Switch>
```

8.3.2.4 System Status

The **show environment all** command lists the temperature, cooling, fan, and power supply information that the individual show environment commands display, as described in Section 8.3.2.1, Section 8.3.2.2, and Section 8.3.2.3.

Example

- This command displays the temperature, cooling, fan, and power supply status:

```
switch>show environment all
System temperature status is: Ok
                                                                    Alert   Critical
Sensor  Description                          Temperature  Threshold  Threshold
------- ------------------------------------ ------------ ---------- ----------
1       Front-panel temp sensor                   22.750C        65C        75C
2       Fan controller 1 sensor                   24.000C        75C        85C
3       Fan controller 2 sensor                   29.000C        75C        85C
4       Switch chip 1 sensor                      41.000C       105C       115C
5       VRM 1 temp sensor                         49.000C       105C       110C

System cooling status is: Ok
Ambient temperature: 22C
Airflow: front-to-back
Fan Tray  Status           Speed
--------- ---------------  ------
1         Ok                 35%
2         Ok                 35%
3         Ok                 35%
4         Ok                 35%
5         Ok                 35%

Power                                    Input    Output   Output
Supply  Model                Capacity  Current  Current  Power    Status
------- -------------------- --------- -------- -------- -------- -------------
1       PWR-650AC                 650W    0.44A   10.50A   124.0W Ok
```

8.3.3 Locating Components on the Switch

When a component requires service, the switch administrator may use the **locator-led** command to assist a technician in finding the component. The command causes the status LED on the specified component to flash, and also displays a “service requested” message on the LCD panel of modular switches or lights the blue locator light on the front of fixed switches. Use the **show locator-led** command to display all locator LEDs currently enabled on the switch.

Examples

- This command enables the locator LED on fan tray 3:

```
switch#locator-led fantray 3
Enabling locator led for FanTray3
switch#
```

- This command displays all locator LEDs enabled on the switch:

```
switch#show locator-led
There are no locator LED enabled
switch#
```

8.4 Environment Commands

Environment Control Configuration Commands

- environment fan-speed
- environment insufficient-fans action
- environment overheat action
- locator-led

Environment Display Commands

- show environment all
- show environment cooling
- show environment power
- show environment temperature
- show locator-led

environment fan-speed

The **environment fan-speed** command determines the method of controlling the speed of the switch fans. The switch automatically controls the fan speed by default.

The switch normally controls the fan speed to maintain optimal operating temperatures. The fans can be configured to operate at a constant speed regardless of the switch temperature conditions.

The **no environment fan-speed** and **default environment fan-speed** commands restore the default action of automatic fan-speed control by removing the **environment fan-speed override** statement from ***running-config***.

Important! Overriding the system fan speed is unsupported and should only be done under the direction of an Arista Networks engineer. You can risk damaging hardware by setting the fan speed too low. Doing so without direction from Arista Networks can be grounds for voiding your warranty.

Command Mode

Global Configuration

Command Syntax

```
environment fan-speed ACTION
no environment fan-speed
default environment fan-speed
```

Parameters

- ***ACTION*** fan speed control method. Valid settings include:
  - **auto** fan speed is controlled by the switch.

    This option restores the default setting by removing the **environment fan-speed override** command from ***running-config***.
  - **override** *percent* fan speed is set to the specified percentage of the maximum. Valid *percent* settings range from 30 to 100.

Examples

- This command overrides the automatic fan speed control and configures the fans to operate at 50% of maximum speed.

```
switch(config)#environment fan-speed override 50
==================================================================
WARNING: Overriding the system fan speed is unsupported and should only
be done under the direction of an Arista Networks engineer.
You can risk damaging hardware by setting the fan speed too low
and doing so without direction from Arista Networks can be grounds
for voiding your warranty.
To set the fan speed back to automatic mode, use the
'environment fan-speed auto' command
==================================================================
switch(config)#
```

- This command restores control of the fan speed to the switch.

```
switch(config)#environment fan-speed auto
switch(config)#
```

environment insufficient-fans action

The **environment insufficient-fans** command controls the switch response to the insufficient fan condition. By default, the switch initiates a shutdown procedure when it senses insufficient fans.

The switch operates normally when one fan is not operating. Non-functioning modules should not be removed from the switch unless they are immediately replaced; adequate switch cooling requires the installation of all components, including a non-functional fan.

Two non-operational fans trigger an ***insufficient fan shutdown*** condition. This condition normally initiates a power down procedure.

The **no environment insufficient-fans** and **default environment insufficient-fans** commands restore the default shutdown response to the insufficient-fans condition by removing the **environment insufficient-fans action ignore** statement from ***running-config***.

Important! Overriding the system shutdown behavior when the system has insufficient fans inserted is unsupported and should only be done under the direction of an Arista Networks engineer. You risk damaging hardware by not shutting down the system in this situation, and doing so without direction from Arista Networks can be grounds for voiding your warranty.

Command Mode

Global Configuration

Command Syntax

```
environment insufficient-fans action REMEDY
no environment insufficient-fans action
default environment insufficient-fans action
```

Parameters

- ***REMEDY*** configures action when switch senses an insufficient fan condition. Settings include:
  - **ignore** switch continues operating when insufficient fans are operating.
  - **shutdown** switch shuts power down when insufficient fans are operating.

  The ***shutdown*** parameter restores default behavior by removing the **environment insufficient-fans** command from ***running-config***.

Examples

- This command configures the switch to continue operating after it senses insufficient fan condition.

```
switch(config)#environment insufficient-fans action ignore
================================================================
WARNING: Overriding the system shutdown behavior when the system
has insufficient fans inserted is unsupported and should only be done under
the direction of an Arista Networks engineer. You risk damaging
hardware by not shutting down the system in this situation, and doing
so without direction from Arista Networks can be grounds for voiding
your warranty. To re-enable the shutdown-on-overheat behavior, use
the 'environment insufficient-fans action shutdown' command.
================================================================
```

- This command configures the switch to shut down when it senses an insufficient fan condition.

```
switch(config)#environment insufficient-fans action shutdown
switch(config)#
```

environment overheat action

The **environment overheat** command controls the switch response to an overheat condition. By default, the switch shuts down when it senses an overheat condition.

Important! Overriding the system shutdown behavior when the system is overheating is unsupported and should only be done under the direction of an Arista Networks engineer. You risk damaging hardware by not shutting down the system in this situation, and doing so without direction from Arista Networks can be grounds for voiding your warranty.

Arista switches include internal temperature sensors. The number and location of the sensors vary with each switch model. Each sensor is assigned temperature thresholds that denote alert and critical conditions. Temperatures that exceed the threshold trigger the following:

- **Alert Threshold**: All fans run at maximum speed and a warning message is logged.
- **Critical Threshold**: The component is shut down immediately and its Status LED flashes orange.

In modular systems, cards are shut down when their temperatures exceed the critical threshold. The switch normally shuts down if the temperature remains above the critical threshold for three minutes.

The **no environment overheat action** and **default environment overheat action** commands restore the default shutdown response to the environment overheat condition by removing the **environment overheat action ignore** statement from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
environment overheat action REMEDY
no environment overheat action
default environment overheat action
```

Parameters

- ***REMEDY*** reaction to an overheat condition. Default value is ***shutdown***.
  - **shutdown** switch shuts power down by an overheat condition.
  - **ignore** switch continues operating during an overheat condition.

Examples

- This command configures the switch to continue operating after it senses an overheat condition.

```
switch(config)#environment overheat action ignore
======================================================================
WARNING: Overriding the system shutdown behavior when the system
is overheating is unsupported and should only be done under
the direction of an Arista Networks engineer. You risk damaging
hardware by not shutting down the system in this situation, and doing
so without direction from Arista Networks can be grounds for voiding
your warranty. To re-enable the shutdown-on-overheat behavior, use
the 'environment overheat action shutdown' command.
======================================================================
switch(config)#
```

- This command configures the switch to shut down when it senses an overheat condition.

```
switch(config)#environment overheat action shutdown
switch(config)#
```

locator-led

When a component requires service, the **locator-led** command activates a locator to assist a technician in finding the component. The command causes the status LED on the specified component to flash, and also displays a “service requested” message on the LCD panel of modular switches or lights the blue locator light on the front of fixed switches. The available locators vary by platform; to see a list of the locator LEDs available on the switch, use the **locator-led ?** command. To disable the locator LED, use the **no locator-led** command.

Command Mode

Privileged EXEC

Command Syntax

locator-led {**fantray** *tray_num* | **interface** *interface* | **module** *module_num* | **powersupply** *supply_num*}
no locator-led {**fantray** *tray_num* | **interface** *interface* | **module** *module_num* | **powersupply** *supply_num*}

Parameters

- **fantray** ***tray_num*** activates locator on specified fan tray.
- **interface** ***interface*** activates locator on specified interface.
- **module** ***module_num*** activates locator on specified module.
- **powersupply** ***supply_num*** activates locator on specified power supply.

Examples

- This command enables the locator LED on fan tray 3.

```
switch#locator-led fantray 3
Enabling locator led for FanTray3
switch#
```

- This command disables the locator LED on fan tray 3.

```
switch#no locator-led fantray 3
Disabling locator led for FanTray3
switch#
```

- This command displays the locator LEDs available on the switch.

```
switch#locator-led ?
  fantray      Fan tray LED
  interface    Interface LED
  module       Module LED
  powersupply  Power supply LED
switch#
```

show environment all

The **show environment all** command displays temperature, cooling, and power supply status.

Command Mode

EXEC

Command Syntax

show environment all

Examples

- This command displays the switch's temperature, cooling, and power supply status

```
switch>show environment all
System temperature status is: Ok
                                                                  Alert   Critical
Sensor  Description                          Temperature  Threshold  Threshold
------- ------------------------------------ ------------ ---------- ----------
1       Front-panel temp sensor                   31.000C        65C        75C
2       Fan controller 1 sensor                   32.000C        75C        85C
3       Fan controller 2 sensor                   38.000C        75C        85C
4       Switch chip 1 sensor                      50.000C       105C       115C
5       VRM 1 temp sensor                         60.000C       105C       110C

System cooling status is: Ok
Ambient temperature: 31C
Airflow: front-to-back
Fan Tray  Status           Speed
--------- ---------------- ------
1         Ok                 52%
2         Ok                 52%
3         Ok                 52%
4         Ok                 52%
5         Ok                 52%

Power                                    Input    Output   Output
Supply  Model                Capacity  Current  Current  Power    Status
------- -------------------- --------- -------- -------- -------- -------------
1       PWR-760AC                 760W    0.81A   11.00A   132.6W Ok
2       PWR-760AC                 760W    0.00A    0.00A     0.0W AC Loss

switch>
```

show environment cooling

The **show environment cooling** command displays fan status, air flow direction, and ambient temperature on the switch.

Command Mode

EXEC

Command Syntax

```
show environment cooling [INFO_LEVEL]
```

Parameters

- ***INFO_LEVEL*** specifies level of detail that the command displays. Options include:
  - <no parameter> displays the fan status, air flow direction, and ambient switch temperature.
  - **detail** also displays actual and configured fan speed of each fan.

Display Values

- **System cooling status**:
  - ***Ok*** no more than one fan has failed or is not inserted.
  - ***Insufficient fans*** more than one fan has failed or is not inserted. This status is also displayed if fans with different airflow directions are installed. The switch shuts down if the error is not resolved.
- **Ambient temperature** temperature of the surrounding area.
- **Airflow** indicates the direction of the installed fans:
  - ***front-to-back*** all fans flow air from the front to the rear of the chassis.
  - ***back-to-front*** all fans flow air from the rear to the front of the chassis.
  - ***incompatible fans*** fans with different airflow directions are inserted.
  - ***Unknown*** The switch is initializing.
- **Fan Tray Status** table displays the status and operating speed of each fan. Status values indicate the following conditions:
  - ***OK*** The fan is operating normally.
  - ***Failed*** The fan is not operating normally.
  - ***Unknown*** The system is initializing.
  - ***Not Inserted*** The system is unable to detect the specified fan.
  - ***Unsupported*** The system detects a fan that the current software version does not support.

Example

- This command displays the fan status, air flow direction, and ambient switch temperature.

```
switch>show environment cooling
System cooling status is: Ok
Ambient temperature: 30C
Airflow: front-to-back
Fan Tray  Status           Speed
--------- ---------------- ------
1         Ok                 51%
2         Ok                 51%
3         Ok                 51%
4         Ok                 51%
5         Ok                 51%
switch>
```

show environment power

The **show environment power** command displays the status of all power supplies in the switch.

Command Mode

EXEC

Command Syntax

show environment power [*INFO_LEVEL*]

Parameters

- ***INFO_LEVEL*** specifies level of detail that the command displays. Options include:
  - <no parameter> displays current and power levels for each supply.
  - **detail** also includes status codes that can report error conditions.

Example

- This command displays the status of power supplies on the switch.

```
switch>show environment power
Power                                    Input    Output   Output
Supply  Model                Capacity  Current  Current  Power    Status
------- -------------------- --------- -------- -------- -------- -------------
1       PWR-760AC                 760W    0.81A   11.00A   132.8W Ok
2       PWR-760AC                 760W    0.00A    0.00A     0.0W AC Loss
switch>
```

show environment temperature

The **show environment temperature** command displays the operating temperature of all sensors on the switch.

Command Mode

EXEC

Command Syntax

show environment temperature [*MODULE_NAME*][*INFO_LEVEL*]

Parameters

- ***MODULE_NAME*** Specifies modules for which data is displayed. This parameter is only available on modular switches. Options include:
  - <no parameter> All modules (identical to **all** option).
  - **fabric** *fab_num* Specified fabric module. Number range varies with switch model.
  - **linecard** *line_num* Line card module. Number range varies with switch model.
  - **supervisor** *super_num* Supervisor module. Number range varies with switch model.
  - *mod_num* Supervisor (1 to 2) or line card (3 to 18) module.
  - **all** All modules.
- ***INFO_LEVEL*** specifies level of detail that the command displays. Options include:
  - <no parameter> displays table that lists the temperature and thresholds of each sensor.
  - **detail** displays data block for each sensor listing the current temperature and historic data.

Display Values

- **System temperature status** is the first line that the command displays. Values report the following:
  - ***Ok*** All sensors report temperatures below the alert threshold.
  - ***Overheating*** At least one sensor reports a temperature above its alert threshold.
  - ***Critical*** At least one sensor reports a temperature above its critical threshold.
  - ***Unknown*** The switch is initializing.
  - ***Sensor Failed*** At least one sensor is not functioning.

Examples

- This command displays a table that lists the temperature measured by each sensor.

```
switch>show environment temperature
System temperature status is: Ok
                                                                  Alert   Critical
Sensor  Description                          Temperature  Threshold  Threshold
------- ------------------------------------ ------------ ---------- ----------
1       Front-panel temp sensor                  30.750C        65C        75C
2       Fan controller 1 sensor                  32.000C        75C        85C
3       Fan controller 2 sensor                  38.000C        75C        85C
4       Switch chip 1 sensor                     50.000C       105C       115C
5       VRM 1 temp sensor                        60.000C       105C       110C
switch>
```

- This command lists the temperature detected by each sensor, and includes the number of previous alerts, the time of the last alert, and the time of the last temperature change.

```
switch>show environment temperature detail
TempSensor1 - Front-panel temp sensor
                              Current State      Count            Last Change
  Temperature                        30.750C
  Max Temperature                    35.000C             4 days, 23:35:24 ago
  Alert                                False          0                 never

TempSensor2 - Fan controller 1 sensor
                              Current State      Count            Last Change
  Temperature                        32.000C
  Max Temperature                    36.000C             4 days, 23:32:46 ago
  Alert                                False          0                 never

TempSensor3 - Fan controller 2 sensor
                              Current State      Count            Last Change
  Temperature                        38.000C
  Max Temperature                    41.000C             4 days, 23:37:56 ago
  Alert                                False          0                 never

TempSensor4 - Switch chip 1 sensor
                              Current State      Count            Last Change
  Temperature                        51.000C
  Max Temperature                    53.000C             4 days, 23:35:16 ago
  Alert                                False          0                 never

TempSensor5 - VRM 1 temp sensor
                              Current State      Count            Last Change
  Temperature                        60.000C
  Max Temperature                    62.000C             4 days, 22:54:51 ago
  Alert                                False          0                 never

switch>
```

show locator-led

The **show locator-led** command displays the status of locator LEDs enabled on the switch.

Command Mode

Privileged EXEC

Command Syntax

show locator-led

Example

- This command displays all locator LEDs enabled on the switch.

```
switch#show locator-led
There are no locator LED enabled
switch#
```

Chapter 9

Maintenance Mode

This chapter describes configuration for performing maintenance of switch elements.

This chapter contains these sections:

- Section 9.1: Overview
- Section 9.2: Maintenance Mode Elements
- Section 9.3: Maintenance Mode Features
- Section 9.4: Maintenance Mode Configuration
- Section 9.5: Maintenance Mode Commands

9.1 Overview

Using maintenance mode, you can perform several maintenance activities such as:

- EOS image upgrade
- Initial configuration or reconfiguration of a production system
- Replacement of hardware
- Changing linecards or transceiver modules
- Replace, reattach, and reroute cables

Maintenance mode uses BGP to divert traffic away from the switch on which the maintenance tasks need to be performed, minimizing traffic impact. You can set the traffic thresholds and time limits at which the switch, or parts of the switch, is considered to be available for maintenance tasks.

Maintenance mode can be activated on a switch at boot-up or during operation. The mode provides the following benefits:

- Rerouting of traffic when the mode is activated during operation and other routes are present
- Replacement of hardware in modular systems or systems with redundant hardware

The switch is placed into maintenance mode, serviced, and then returned to normal operation.

9.2 Maintenance Mode Elements

Maintenance mode elements include Units, Groups of Interfaces and BGP Peers, and Profiles. Arista Network switches provide maintenance mode operations performed on a fundamental, configurable element, referred to as a Unit. Maintenance mode will quiesce a unit, which places the unit into maintenance mode by gracefully transitioning traffic away from it.

The most common maintenance mode operations such as removing from service an entire switch system or individual components of the switch, including a single linecard, interface, or BGP peer, can be achieved using minimal configuration.

9.2.1 Units

Units are configurable maintenance mode elements that comprise a collection of various groups. In addition, units contain policies which decide whether the member groups should be put into maintenance mode automatically upon boot. Built-in units are configured by default, such as the System unit representing the entire system. All maintenance mode operations are executed at the unit level.

An interface, interface range, and BGP peer (or peer-group) can be directly put under maintenance.

9.2.1.1 Built-in Units

There are various built-in units such as **System** and **Linecardn**. Fixed systems contain only one built-in unit called **System,** which comprises the interface group containing all the Ethernet interfaces and BGP groups per VRF containing all the peers in the respective VRF.

Modular Systems have both **System** and **Linecardn** units. **Linecardn** units are present for each linecard which comprises the **linecardn** groups containing all Ethernet interfaces of that linecard.

9.2.1.2 User-configured Units

You can also configure customized units containing user-defined groups and policies as shown in the following example. A custom group called BG1 with a custom interface IG1 and a unit profile UP1 is created. The show command displays the details.

```
switch(config)#maintenance
switch(config-maintenance)# unit UNIT1
switch(config-unit-UNIT1)# group bgp BG1
switch(config-unit-UNIT1)# group interface IG1
switch(config-unit-UNIT1)# profile unit UP1
switch(config-unit-UNIT1)# exit
switch(config-maintenance)# show maintenance units
Unit Name: System
Origin: Built-in
Status: Not Under Maintenance
Unit Profile: Default
Time Since Last State Change: never
Bgp Groups:
AllBgpNeighborVrf-default
Interface Groups:
AllEthernetInterface
Unit Name: UNIT1
Origin: User Configured
Status: Under Maintenance
Unit Profile: UP1
Time Since Last State Change: 0:00:08 ago
Bgp Groups:
BG1
Interface Groups:
IG1
```

9.2.2 Groups of Interfaces and BGP Peers

Maintenance mode group types include the groups for interfaces and BGP peers. Groups are identified by a group name unique to a particular group type.

By default, several built-in groups are available on the device such as **linecard** groups containing physical interfaces.

9.2.2.1 Built-in Groups

There are several built-in groups such as **AllEthernetInterface**, **Linecard1**, **Linecard2**, etc., **AllBgpNeighborVrf-<vrf_name>**. On fixed-system, **AllEthernetInterface** is the built-in interface group which contains all the physical Ethernet interfaces on the switch whereas on modulars **Linercard1**, **Linecard2**, etc., are the built-in groups which contain respective linecard interfaces and are part of the **System** unit. **AllBgpNeighborVrf-<vrf_name>** is the built-in BGP group which contains all the BGP peers in that particular VRF.

9.2.2.2 User-defined Groups

The following set of commands sets up a custom group (IG1) of interfaces, which includes physical ports, port-channels and SVIs.

```
switch(config)#group interface IG1
switch(config-group-if-IG1)#interface Ethernet1
switch(config-group-if-IG1)#interface Port-Channel1,20
switch(config-group-if-IG1)#interface Vlan1-20
switch(config-group-if-IG1)#exit
switch(config)#
```

The following set of commands sets up a custom group (BG1) of BGP peers.

```
switch(config)#group bgp BG1
switch(config-group-bgp-BG1)#neighbor 10.0.0.1
switch(config-group-bgp-BG1)#neighbor BGP_PG1
switch(config-group-bgp-BG1)#vrf vrf1
switch(config-group-bgp-BG1)#exit
switch(config)#
```

Note BGP groups are specific to VRF.

9.2.3 Profiles

Profiles are configurable maintenance mode elements that define policies for related software or hardware components to carry out maintenance mode operations.

9.2.3.1 Default Profiles

Default profiles are the built-in policies which are applied to groups interface/BGP and unit.

The default profile is used in the absence of an explicit interface/BGP profile associated with the group, or explicit unit profile associated with the unit.

- **Interface Profile**

  Default interface profile has rate-monitoring load-interval set to 60 seconds, threshold set to 100 kbps, and shutdown disabled as shown. The max-delay parameter is set to 300 seconds but is not enabled.

```
   switch(config-maintenance)#show maintenance profile interface default
   Interface Profile: Default
Rate Monitoring:
          load-interval: 60 seconds
          threshold (in/out): 100 kbps
shutdown:
          enabled: no
          max-delay: 300 seconds
```

- **BGP Profile**

  Default BGP profile has route-map with set clauses—set community GSHUT additive and set local-preference 0.

```
   switch(config-maintenance)#show maintenance profile bgp default
   Bgp Profile: Default
Initiator route-map: SystemGenerated
   route-map SystemGenerated permit 10
     Description:
       description System generated initiator route-map
     Match clauses:
     SubRouteMap:
Set clauses:
set local-preference 0
set community GSHUT additive
```

- **Unit Profile**

  Default unit profile has on-boot setting disabled.

```
   switch(config-maintenance)#show maintenance profiles unit default
Unit Profile: Default
          On-boot:
enabled: no
duration: 300 seconds
```

9.2.3.2 User-defined Profiles

You can define your own profiles which can be associated to groups or set as default profiles.

Interface Profile: The following set of commands sets up an Interface Profile(IP1) with load interval set to 10 seconds, rate-monitoring threshold set to 100kbps and the maximum delay for shutting down the interface set to 100 seconds. The interface will be shutdown with cause **maint-down** if traffic does not drain below the threshold even after the specified maximum delay period of 100 seconds.

```
switch(config)#maintenance
switch(config-maintenance)#profile interface IP1
switch(config-profile-intf-IP1)#rate-monitoring load-interval 10
switch(config-profile-intf-IP1)#rate-monitoring threshold 100
switch(config-profile-intf-IP1)#shutdown max-delay 100
switch(config-profile-intf-IP1)#exit
switch(config-maintenance)#
```

An interface profile can be associated to only interface groups using the following set of commands.

```
switch(config)#group interface IG1
switch(config-group-if-IG1)#maintenance profile interface IP1
switch(config-group-if-IG1)#exit
switch(config)#
```

You can set the interface profile as the default interface profile using the following set of commands.

```
switch(config)#maintenance
switch(config-maintenance)# profile interface IP1 default
switch(config-maintenance)# exit
switch(config)#
```

Bgp Profile: The following set of commands sets up a BGP profile(BP1) with initiator route-map called RM which will be applied for both inbound and outbound directions.

```
switch(config)#maintenance
switch(config-maintenance)#profile bgp BP1
switch(config-profile-bgp-BP1)#initiator route-map RM inout
switch(config-profile-bgp-BP1)#exit
switch(config-maintenance)#
```

A BGP profile can be associated to both interface and bgp groups using the following commands.

```
switch(config)#group interface IG1
switch(config-group-if-IG1)#maintenance profile bgp BP1
switch(config-group-if-IG1)#exit
switch(config)# group bgp BG1
switch(config-group-bgp-BG1)# maintenance profile bgp BP1
switch(config-group-bgp-BG1)# exit
switch(config)#
```

You can set the bgp profile as the default bgp profile using the following set of commands.

```
switch(config)# maintenance
switch(config-maintenance)# profile bgp BP1 default
switch(config-maintenance)# exit
switch(config)#
```

Unit Profile: The following set of commands sets up a Unit profile(UP1) with on-boot duration of 300 seconds. The unit will enter into maintenance mode at boot-up and exit maintenance mode at the end of 5 minutes (300sec) after boot-up.

```
switch(config-maintenance)#profile unit UP1
switch(config-profile-unit-UP1)#on-boot duration 300
switch(config-profile-unit-UP1)#exit
switch(config-maintenance)#
```

A Unit profile can be associated to a Unit using the following commands.

```
switch(config)# maintenance
switch(config-maintenance)# unit UNIT1
switch(config-unit-UNIT1)# profile unit UP1
switch(config-unit-UNIT1)# exit
switch(config-maintenance)#
```

You can set the Unit profile as the default Unit profile using the following set of commands.

```
switch(config)# maintenance
switch(config-maintenance)# profile unit UP1 default
switch(config-maintenance)# exit
switch(config)#
```

9.3 Maintenance Mode Features

Arista Network switches provide maintenance mode features including rate monitoring, BGP maintenance route map, on-boot maintenance, and EventMgr integration.

9.3.1 Rate Monitoring

Rate monitoring provides a mechanism to monitor traffic on interfaces identified for maintenance. You can set the traffic threshold and a time limit for the interface to be shutdown for maintenance tasks.

A shutdown parameter can be configured in the interface profile that signals the interface to be shutdown after it has entered maintenance mode.

The max-delay parameter specifies the maximum number of seconds to allow for traffic to dissipate from the interface before the interface is shutdown. The default interface profile settings are shown in the output of the **show maintenance profile interface default** command.

9.3.2 BGP Maintenance Route Map

Route-maps are used within a BGP maintenance profile to tag the inbound and outbound routes in order to direct traffic away from the unit. The default profile tags the inbound and outbound routes with the GSHUT community. Other methods can be configured under the route-map such as alternate communities, or by using as-path prepend operations.

9.3.3 On-boot Maintenance

There are two ways of placing a unit in maintenance mode on switch boot-up:

- The unit is placed into maintenance mode prior to the switch reboot, and the running-config is saved prior to switch boot-up.
- The on-boot property in the unit maintenance profile specifies that the unit will be placed into maintenance mode as part of boot-up, and remains so for the specified duration.

Note The duration value in the on-boot unit maintenance profile starts as soon as the unit is put into maintenance mode on boot-up.

9.4 Maintenance Mode Configuration

You can configure maintenance mode for the entire device, specific linecards, or any other Unit. You can set up configuration for maintenance mode for the device at boot-up or while it is running.

9.4.1 Unit (System, Linecard*n*, etc.) Configuration

Arista Network switches provide the ability to place the switch in maintenance mode, and configuration options for groups, profiles, associating profiles with groups, units, and maintenance mode operations. **System** is a predefined (built-in) unit on all switches. Built-in groups include **AllEthernetInterface**, **AllBgpNeighborVRF-<vrf_name>, and Linecard*n*. Linecard*n*** can also be a built-in unit and can be differentiated depending on the command being used as shown.

- switch(config-maintenance)# unit Linecardn
- switch(config)# group interface Linecardn

Built-in unit **System** comprises the following groups:

- **AllEthernetInterface** - a built-in interface group which contains all physical Ethernet interfaces on the switch on a fixed system
- **Linecard*n*** - a built-in interface group which contains all interfaces for the linecard numbered ‘n’ for modular systems
- **AllBgpNeighborVRF-<vrf_name>** - a built-in BGP group which contains all the BGP peers in the named VRF.

For each Linecard ‘n’, there is a built-in unit which consists of all the **Linecard*n*** groups.

By default, the default interface and BGP profiles are applied to the built-in interface and BGP groups and the default built-in unit profile is applied to the built-in unit. You can also configure your own profiles and choose a default.

In the following example, traffic is flowing through multiple switches in the spine to and from one switch to another, when you elect to put one of the Units (entire switch or parts thereof) in the spine switch in maintenance mode. The traffic is then gracefully steered away from the Unit, provided other paths are available. Traffic will continue to flow through the Unit placed into maintenance mode, if no other path is available.

Example

Note The illustration shows an entire switch as the Unit. You can replace switch with Linecard*n* or another relevant Unit as appropriate.

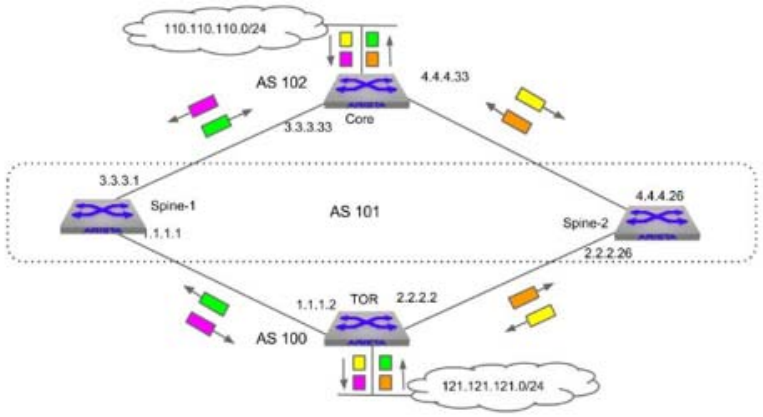


Figure 9-1: Traffic flow pattern between TOR and Core – Before Maintenance

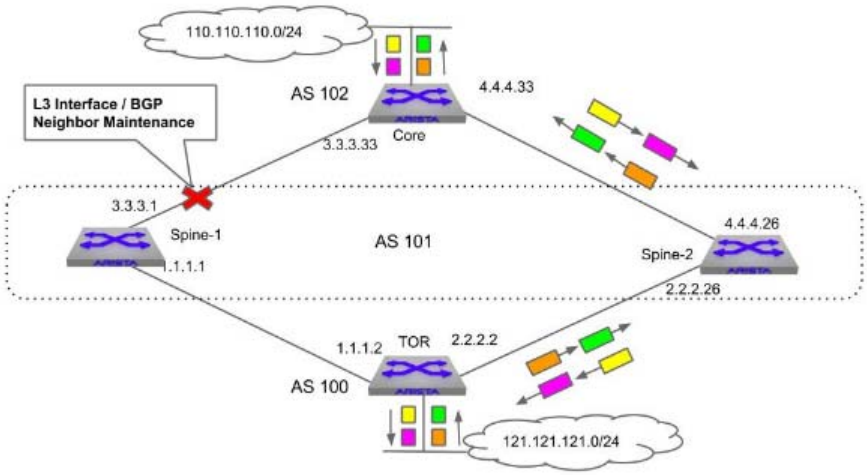


Figure 9-2: Traffic flow pattern between TOR and Core – After unit (System) on Spine-1 is put into Maintenance

You can see the status of the Unit (System) using the **show maintenance units System** command for the example above before the system is placed into maintenance mode. If the device being placed into maintenance mode is modular and the Unit is a linecard, replace the argument **System** with **Linecard*n*** to see the status of the Unit (Linecard*n*).

```
switch(config)# show maintenance units System
Unit Name: System
   Origin: Built-in
   Status: Not Under Maintenance
   Unit Profile: Default
   Time Since Last State Change: never
   Bgp Groups:
     AllBgpNeighborVrf-default
   Interface Groups:
     AllEthernetInterface
```

You can then place the Unit (System) into maintenance mode and recheck the status using the sequence of commands shown.

```
switch(config-maintenance)# unit System
switch(config-builtin-unit-System)# quiesce
switch(config-builtin-unit-System)# exit
switch(config-maintenance)# show maintenance
Flags:
o - On-boot maintenance
v - Violating traffic threshold


Unit Name              Status                   Time since last change    Flags
---------------------- ----------------------- ------------------------- -----
System                  Under Maintenance              0:02:03 ago
switch(config-maintenance)# show ip bgp summary
BGP summary information for VRF default
Router identifier 1.1.1.1, local AS number 101
Neighbor Status Codes: m - Under maintenance
  Neighbor         V  AS           MsgRcvd   MsgSent  InQ OutQ  Up/Down State
PfxRcd PfxAcc
m 1.1.1.2        4  100             24       17    0    0 00:00:40 Estab  5     5
m 3.3.3.33       4  102             15       16    0    0 00:06:23 Estab  1     1
```

Note The ‘o’ flag is shown for on-boot maintenance in the **show maintenance** command.and the ‘m’ neighbor status flag in the **show ip bgp summary** command indicates that the peer is in maintenance mode.

9.4.2 On-boot Maintenance Mode Configuration

To configure on-boot maintenance, you can use one of two methods:

- Use **quiesce config** or
- Use on-boot profile

9.4.2.1 **Using quiesce config:** You must perform the following tasks to place the Unit in maintenance mode on boot-up using the **quiesce** command.

Step 1 Place the unit into maintenance mode prior to switch reboot using the following commands.

```
switch(config)#maintenance
switch(config-maintenance)# unit System
switch(config-unit-System)# quiesce
switch(config-unit-System)# exit
switch(config-maintenance)# show maintenance

Flags:
o - On-boot maintenance
v - Violating traffic threshold
Unit Name              Status                   Time since last change     Flags
---------------------- ----------------------- ------------------------- -----
System                  Under Maintenance             00:01:10 ago
```

Step 2 Save the running-config using the following command.

```
switch(config)# copy running-config startup-config
Copy completed successfully
switch(config)#
```

Step 3 Reload the device.

```
switch(config)# reload
Proceed with reload? [Confirm] Yes
Connection to switch closed.
```

After the device comes up, you must execute the **no quiesce** command for the Unit to come out of maintenance mode. You can check the status of the device after it comes up using the **show maintenance** command.

```
switch# show maintenance
Flags:
o - On-boot maintenance
v - Violating traffic threshold
Unit Name               Status                  Time since last change    Flags
---------------------- ---------------------- -------------------------- -----
System                  Under Maintenance           00:03:10 ago
```

9.4.2.2 **Using on-boot profile:** The on-boot property in the Unit maintenance profile specifies that the Unit will be placed into maintenance mode as part of boot-up for the specified duration. You must perform the following tasks to use this method.

Step 1 Check to see if the on-boot maintenance mode is enabled using the **show maintenance profiles unit default**.

```
switch# show maintenance profiles unit default
Unit Profile: Default
   On-boot:
     enabled: no
     duration: 300 seconds
```

Step 2 Configure an on-boot profile with on-boot enabled and a duration specified. Make this the default Unit profile. The following code example shows the creation of an on-boot duration of 300 seconds in the profile unit UP1

```
switch(config)#maintenance
switch(config-maintenance)#profile unit UP1
switch(config-profile-unit-UP1)#on-boot duration 300
switch(config-profile-unit-UP1)#exit
switch(config-maintenance)#profile unit UP1 default
switch(config-maintenance)#show maintenance profiles unit default
Unit Profile: UP1
    On-boot:
         enabled: yes
         duration: 300 seconds
switch(config-maintenance)#
```

Step 3 Save the running-config and reload the device.

```
switch(config)# copy running-config startup-config
Copy completed successfully
switch(config)# reload
Connection to switch closed.
```

Step 4 After the device comes up, execute the **show maintenance** and **show maintenance units System** commands.

```
switch(config)# show maintenance
Flags:
o - On-boot maintenance
v - Violating traffic threshold
Unit Name Status Time since last change Flags
--------------------- ---------------------- ------------------------- -----
System Under Maintenance 00:00:08 ago o

switch(config)# show maintenance units System
Unit Name: System
   Origin: Built-in
   Status: Under Maintenance (on-boot)
   Unit Profile: UP1
   Time Since Last State Change: 0:00:16 ago
   Will come out of on-boot Maintenance after 0:04:43
   Interface Groups:
     AllEthernetInterface
   History:
    2017-01-18 00:44:39  old state: 'maintenanceModeEnter' to new state:
'underMaintenance' 0:00:16 ago
    2017-01-18 00:43:54  old state: 'active' to new state: 'maintenanceModeEnter'
0:01:01 ago
```

The ‘o’ - flag shows that unit System is under maintenance due to on-boot profile. Also, **show maintenance units System**’ output shows the following - ‘Will come out of on-boot Maintenance after 0:04:43’, which is the time remaining of the specified duration of 5 minutes.

The Unit will come up in maintenance mode when the device boots up and will exit maintenance mode once the specified duration of 300 seconds in the default profile is completed. The BGP sessions will remain under maintenance for the duration and will resume after the specified duration is over.

9.4.3 Interface-level Maintenance Mode Configuration

To configure interface-level maintenance, you must perform the following tasks.

Step 1 Configure an interface-level profile (or use a pre-configured one). The following code example creates a user-defined interface profile IP1 with a rate-monitoring load-interval of 100 seconds, a rate-monitoring threshold of 500 kbps and a maximum shutdown delay of 100 seconds.

```
switch(config)#maintenance
switch(config-maintenance)#profile interface IP1
switch(config-profile-intf-IP1)#rate-monitoring load-interval 100
switch(config-profile-intf-IP1)#rate-monitoring threshold 500
switch(config-profile-intf-IP1)#shutdown max-delay 100
```

Step 2 Place the unit into maintenance mode.

```
switch(config-maintenance)#interface IP1
switch(config-maint-intf-IP1)#quiesce
```

Step 3 Remove the unit from maintenance mode once the service has been performed.

```
switch(config-maintenance)#interface IP1
switch(config-maint-intf-IP1)#no quiesce
```

The unit will come out of maintenance mode and resume normal operation.

9.4.4 Entering Maintenance Mode

Enter configuration commands **unit** and **quiesce** using the **neighbor** mode command to place the switch into maintenance mode. The following code sequence places unit foo, the interface 3/3, and BGP 1.1.1.1 in maintenance mode.

Example

```
switch(config)#maintenance
switch(config-maintenance)#unit foo
switch(config-unit-foo)#quiesce
switch(config-unit-foo)#exit
switch(config-maintenance)#interface ethernet 3/3
switch(config-maint-if-Et3/3)#quiesce
switch(config-unit-if-Et3/3)#exit
switch(config-maintenance)#bgp 1.1.1.1
switch(config-maint-bgp-1.1.1.1)#quiesce
switch(config-maint-bgp-1.1.1.1)#exit
switch(config-maintenance)#
```

9.4.5 Exiting Maintenance Mode

Enter configuration commands **unit** and **no quiesce** using the **neighbor** mode command for the switch to exit maintenance mode. The following code sequence causes unit foo, the interface 3/3, and BGP 1.1.1.1 to exit maintenance mode.

Example

```
switch(config)#maintenance
switch(config-maintenance)#unit foo
switch(config-unit-foo)#no quiesce
switch(config-unit-foo)#exit
switch(config-maintenance)#interface ethernet 3/3
switch(config-maint-if-Et3/3)#no quiesce
switch(config-unit-if-Et3/3)#exit
switch(config-maintenance)#bgp 1.1.1.1
switch(config-maint-bgp-1.1.1.1)#no quiesce
switch(config-maint-bgp-1.1.1.1)#exit
switch(config-maintenance)#
```

9.4.6 Configuring Event Handlers

Enter configuration options for the **trigger on-maintenance** command to fire at different stages while entering or exiting maintenance mode.

Example for Maintenance Mode Event Handler for all Stages

```
switch(config)#event-handler foo
switch(config-handler-foo)#trigger on-maintenance enter unit unit-foo all
switch(config-handler-foo)#action bash /mnt/flash/mm-event-handler-script
switch(config-handler-foo)#timeout 20
switch(config-handler-foo)#exit
switch(config)#
```

Note The user is expected to configure the timeout value. This is time within which the script should complete execution and exit. If the script has not exited by the end of this period, then the following will occur:
1. Send the SIGUSR1 signal to the script.
2. Wait for a GRACE-PERIOD of 10 seconds for the script to exit.
3. If the script does not exit even after that GRACE-PERIOD, then send a SIGKILL to the script.
4. The maintenance operation progresses to the next stage.
GRACE-PERIOD is not configurable.

```
switch(config)#event-handler bar
switch(config-handler-bar)#trigger on-maintenance exit unit unit-foo before
stage ratemon
switch(config-handler-bar)#action bash /mnt/flash/mm-event-handler-script
switch(config-handler-bar)#exit
switch(config)#
```

9.4.7 Configuring Groups

Enter the maintenance mode configuration options for groups with the **maintenance** and **group bgp** commands.

Example for group interface IG1

```
switch(config)#group interface IG1
switch(config-group-if-IG1)#interface Ethernet1
switch(config-group-if-IG1)#interface Port-Channel1,20
switch(config-group-if-IG1)#interface Vlan1-20
switch(config-group-if-IG1)#exit
switch(config)#
```

Example for group bgp BG1

```
switch(config)#group bgp BG1
switch(config-group-bgp-BG1)#neighbor 10.0.0.1
switch(config-group-bgp-BG1)#neighbor BGP_PG1
switch(config-group-bgp-BG1)#vrf vrf1
switch(config-group-bgp-BG1)#exit
switch(config)#
```

Note BGP groups are specific to VRF.

9.4.8 Configuring Profiles

Enter the maintenance mode configuration options for profiles with the **profile interface**, **rate-monitoring threshold**, **profile bgp**, and **profile unit <profile_name>** commands.

These command examples assign a user configured profile as the **default** profile.

Example for profile interface IP1

```
switch(config)#maintenance
switch(config-maintenance)#profile interface IP1
switch(config-profile-intf-IP1)#rate-monitoring load-interval 10
switch(config-profile-intf-IP1)#rate-monitoring threshold 100
switch(config-profile-intf-IP1)#shutdown max-delay 100
switch(config-profile-intf-IP1)#profile interface IP1 default
switch(config-profile-intf-IP1)#exit
switch(config-maintenance)#
```

Example for profile bgp BP1

```
switch(config-maintenance)#profile bgp BP1
switch(config-profile-bgp-BP1)#initiator route-map rmap inout
switch(config-profile-bgp-BP1)#profile bgp BP1 default
switch(config-profile-bgp-BP1)#exit
switch(config-maintenance)#
```

Example for profile unit UP1

```
switch(config-maintenance)#profile unit UP1
switch(config-profile-unit-UP1)#on-boot duration 300
switch(config-profile-unit-UP1)#profile unit UP1 default
switch(config-profile-unit-UP1)#exit
switch(config-maintenance)#
```

9.4.9 Associating Profiles with Groups

Enter the maintenance mode configuration options for associating profiles with groups using the **maintenance** and **group bgp** command.

Example

```
switch(config)#group interface IG1
switch(config-group-if-IG1)#maintenance profile bgp BP1
switch(config-group-if-IG1)#maintenance profile interface IP1
switch(config-group-if-IG1)#
```

Note An interface/BGP profile can be associated with the interface group, and a BGP profile can be associated with the BGP group.

9.4.10 Configuring Units

Enter the maintenance mode configuration options for units using the **unit**, **group bgp**, and **maintenance** commands.

Example

```
switch(config)#maintenance
switch(config-maintenance)#unit foo
switch(config-unit-foo)#group bgp BG1
switch(config-unit-foo)#group interface IG1
switch(config-unit-foo)#profile unit UP1
```

9.4.11 Show Commands

Maintenance mode show commands display general and detailed information associated with maintenance mode.

9.4.11.1 show maintenance

This example of the **show maintenance** command displays maintenance mode details.

Example

```
switch(config)#show maintenance
Flags:
o - On-boot maintenance
v - Violating traffic threshold
Unit Name              Status                   Time since last change    Flags
---------------------- ------------------------ ------------------------- -----
System                 Not Under Maintenance            never
Foo                    Under Maintenance             0:00:14 ago         ov

Interface Name         Status                   Time since last change    Flags
---------------------- ------------------------ ------------------------- -----
Ethernet4              Entering Maintenance           0:00:24 ago
Bgp Neighbor(vrf: defa Status                   Time since last change    Flags

---------------------- ------------------------ ------------------------- -----
12.12.12.12            Under Maintenance              0:00:04 ago
Bgp Neighbor(vrf: red) Status                   Time since last change    Flags

---------------------- ------------------------ ------------------------- -----
12.12.12.13            Under Maintenance              0:00:34 ago

switch(config)#
```

9.4.11.2 show maintenance summary

This example of the **show maintenance summary** command displays a summary of maintenance mode information.

Example

```
switch(config)#show maintenance summary
Number of Units configured: 3
Number of Units not under maintenance: 2
Number of Units entering maintenance: 1
Number of Units under maintenance: 0
Number of Units exiting maintenance: 0
Directly Put Under Maintenance:
  Number of interfaces entering maintenance: 0
  Number of interfaces under maintenance: 2
  Number of bgp peers entering maintenance: 0
  Number of bgp peers under maintenance: 3
Rate Monitoring:
  Number of interfaces entering maintenance: 0
  Number of interfaces under maintenance: 4
  Number of interfaces under maintenance with threshold violation: 0
  Number of interfaces shutdown for maintenance: 0

switch(config)#
```

9.4.11.3 show maintenance units

This example of the **show maintenance units** command displays maintenance mode units details.

Example

```
switch(config)#show maintenance units
Unit Name: Linecard3
   Origin: User Configured
   Status: Under Maintenance
   Unit Profile: Default
   Time Since Last State Change: 0:12:07 ago
   Interface Groups:
     IG1
   Interface Traffic Threshold violations:
     Current violations: 1
       Et1
     Total violations, during maintenance: 5
   History:
    2016-04-27 04:00:42 old state: 'maintenanceModeEnter' to new state:
'underMaintenance' 0:12:07 ago
    2016-04-27 03:59:31 old state: 'active' to new state: 'maintenanceModeEnter'
0:13:18 ago
Unit Name: System
   Origin: Built-in
   Status: Not Under Maintenance
   Unit Profile: Default
   Time Since Last State Change: never
   Interface Groups:
     AllEthernetInterface

switch(config)#
```

9.4.11.4 show maintenance bgp

This example of the **show maintenance bgp** command displays maintenance mode BGP details for all IPs and VRFs.

Example

```
switch(config)#show maintenance bgp ip all vrf all
BGP peer maintenance information for VRF default
Router identifier 2.2.2.1, local AS number 1
  Neighbor: 2.2.2.2
    Maintenance State: Not Under Maintenance
BGP peer maintenance information for VRF red
Router identifier 6.6.6.1, local AS number 1
  Neighbor: 1.1.1.2
    Maintenance State: Not Under Maintenance
Router identifier 2.2.2.1, local AS number 1
  Neighbor: 2.2.2.2
    Maintenance State: Not Under Maintenance
     Maintenance route-map: SystemGenerated
      route-map SystemGenerated permit 10
        Description:
        Match clauses:
        Set clauses:
         set community GSHUT additive
         set local-preference 0
  Selected profile from BGP groups: Default

switch(config)#
```

9.4.11.5 show maintenance interface

This example of the **show maintenance interface status** command displays maintenance mode interface details.

Example

```
switch(config)#show maintenance interface
Flags:
v - Violating traffic threshold
s - Shutdown for maintenance
                                                    Rate (Mbps)
Interface                Status                      In         Out   Flags
------------------------ ----------------------- ---------- ---------- -----
 Ethernet1               Under Maintenance          0.4         0.0   v
 Ethernet2               Under Maintenance          0.0         0.0
 Ethernet3               Not Under Maintenance        -           -
 Ethernet4               Under Maintenance          0.0         0.0
 Ethernet5               Not Under Maintenance        -           -

switch(config)#
```

9.4.11.6 show maintenance interface status quiesced

This example of the **show maintenance interface status quisced** command displays maintenance mode interface status details for quiesced interfaces.

Example

```
switch(config)#show maintenance interface status quiesced
Flags:
v - Violating traffic threshold
s - Shutdown for maintenance
                                                        Rate (Mbps)
Interface                Status                          In         Out   Flags
------------------------ ------------------------ ----------- ----------- -----
 Ethernet1               Under Maintenance            0.3         0.0   v
 Ethernet2               Under Maintenance            0.0         0.0
 Ethernet4               Under Maintenance            0.0         0.0

switch(config)#
```

9.4.11.7 show maintenance groups

This example of the **show maintenance groups** command displays maintenance mode group details.

Example

```
switch(config)#show maintenance groups
Interface Group: IG1
  Interfaces:
    Et4-6
  Profiles:
    Interface Profile: IP1
  Units: newEt
Bgp Group: BG
  Neighbors:
    IPv4 Peers: 4.4.4.2, 1.1.1.2, 3.3.3.2
    IPv6 Peers: 3::3
  Bgp Profile: prepend
  Units: newBG

switch(config)#
```

9.4.11.8 show maintenance profiles

This example of the command displays maintenance mode profile details.

Example

```
switch(config)#show maintenance profiles
Interface Profile: INTFPROFILE
   Rate Monitoring:
     load-interval: 444 seconds
     threshold (in/out): 4000 Kbps
   shutdown:
     enabled: yes
     max-delay: 399 seconds
Bgp Profile: BGPPROFILE
   Initiator route-map:
     name: rm
Unit Profile: UNITPROFILE
   On-boot:
     enabled: yes
     duration: 340 seconds

switch(config)#
```

9.4.11.9 show interface status

This example of the **show interface <intf_name> status** command displays maintenance mode information for interfaces.

Example

```
switch(config)#show interface status
Port       Name          Status       Vlan      Duplex   Speed Type           Flags
Et1                      connected    1         full     10G   EbraTestPhyP   mv
Et2                      connected    1         full     10G   EbraTestPhyP   m
Et3                      maint-down   1         full     10G   EbraTestPhyP   m
Et4                      maint-down   1         full     10G   EbraTestPhyP   m
Et5                      connected    1         full     10G   EbraTestPhyP
Et6                      connected    1         full     10G   EbraTestPhyP

switch(config)#
```

9.4.11.10 show interface ethernet

This example of the **show interface ethernet** command displays maintenance mode information for an ethernet interface.

Example

```
switch(config)#show interface ethernet 4
Ethernet4 is down, line protocol is down (maint-down)
  Hardware is Ethernet, address is 0000.0101.0004 (bia 0000.0101.0004)
  Ethernet MTU 9214 bytes , BW 10000000 kbit
  Full-duplex, 10Gb/s, auto negotiation: off, uni-link: unknown
  Down 18 minutes, 39 seconds
  Under maintenance for 18 minutes, 42 seconds
  2 link status changes since last clear
  Last clearing of "show interface" counters never
  5 minutes input rate 0 bps (0.0% with framing overhead), 0 packets/sec
  5 minutes output rate 0 bps (0.0% with framing overhead), 0 packets/sec
     0 packets input, 0 bytes
     Received 0 broadcasts, 0 multicast
     0 runts, 0 giants
     0 input errors, 0 CRC, 0 alignment, 0 symbol, 0 input discards
     0 PAUSE input
     94 packets output, 11562 bytes
     Sent 0 broadcasts, 94 multicast
     0 output errors, 0 collisions
     0 late collision, 0 deferred, 0 output discards
     0 PAUSE output

switch(config)#
```

9.4.11.11 show ip bgp neighbors

This example of the **show ip bgp neighbors** command displays IP BGP neighbors maintenance mode details.

Example

```
switch(config)#show ip bgp neighbors 1.1.1.2
BGP neighbor is 1.1.1.2, remote AS 1, external link
...
  Prefix statistics:
                        Sent      Rcvd
    IPv4 prefixes:         0         0
    IPv6 prefixes:         0         0
  Inbound route map is foo
  Outbound route map is foo
  Session is under maintenance
  Maintenance-mode:
    Inbound and Outbound policy
      Route map is SystemGenerated

switch(config)#
```

9.4.11.12 show ip bgp summary

This example of the **show ip bgp summary** command displays maintenance mode information for IP BGP.

Example

```
switch(config)#show ip bgp summary
 BGP summary information for VRF default
 Router identifier 192.168.201.13, local AS number 100
 Neighbor Status Codes: m - Under maintenance
  Neighbor      V  AS        MsgRcvd   MsgSent  InQ OutQ  Up/Down State  PfxRcd
PfxAcc
m 1.0.0.1       4  300           983       988    0    0 16:16:03 Estab  1      1
  1.0.1.1       4  300           983       983    0    0 16:15:58 Estab  1      1

switch(config)#
```

9.4.11.13 show maintenance stages

These examples of the **show maintenance stages** command display maintenance mode stages details.

Example

```
switch(config)#show maintenance stages
Maintenance Enter Stage Sequence
   No.       Stage      Description
--------- ------------- --------------------------
   1         bgp        BGP Maintenance processing
   2         ratemon    Interface Rate Monitoring

Maintenance Exit Stage Sequence
   No.       Stage      Description
--------- ------------- --------------------------
   1         ratemon    Interface Rate Monitoring
   2         bgp        BGP Maintenance processing

switch(config)#
```

Example

```
switch(config)#show maintenance bgp receiver route-map
route-map SystemGenerated permit 10
  Description:
    description System generated receiver route-map
  Match clauses:
    match community GSHUT-LIST
  SubRouteMap:
  Set clauses:
route-map SystemGenerated permit 50
  Description:
    description System generated receiver route-map
  Match clauses:
  SubRouteMap:
  Set clauses:
tg232(s1)(config)#show maintenance profiles interface
tg232(s1)(config)#show maintenance profiles bgp
tg232(s1)(config)#show maintenance profiles unit
tg232(s1)(config)#show maintenance profiles unit default
Unit Profile: Default
   On-boot:
     enabled: no
     duration: 300 seconds

switch(config)#
```

Example

```
switch(config)#show maintenance profiles interface default
Interface Profile: Default
   Rate Monitoring:
     load-interval: 60 seconds
     threshold (in/out): 100 Kbps
   shutdown:
     enabled: no
     max-delay: 300 seconds

switch(config)#
```

Example

```
switch(config)#show maintenance profiles bgp default
Bgp Profile: Default
  Initiator route-map: SystemGenerated
   route-map SystemGenerated permit 10
     Description:
       description System generated initiator route-map
     Match clauses:
     SubRouteMap:
     Set clauses:
       set local-preference 0
       set community GSHUT additive

switch(config)#
```

Example

```
switch(config)#show maintenance profiles unit default
Unit Profile: Default
   On-boot:
     enabled: no
     duration: 300 seconds

switch(config)#
```

9.4.12 Syslog Messages

Maintenance mode syslog messages are as follows:

- `MaintenanceMode: %MMODE-4-MAINT_OP_WARNING: Unit config is deleted for unit foo. The unit is still undergoing maintenance operation.`
- `MaintenanceMode: %MMODE-5-MAINT_UNIT_STATE_CHANGE: Maintenance unit state changed for unit <Dynamic Unit><SPINE-V6><vrf-default>. Old State maintenanceModeEnter, New State underMaintenance.`
- `MaintenanceMode: %ETH-6-MAINTENANCE_DOWN: Interface Et1 has been shutdown for maintenance.`
- `MaintenanceMode: %MMODE-5-INTF_PROFILE_CHANGE: For interface Et1 interface profile changed to IP1.`
- `Rib: %BGP-6-MAINTENANCE-MODE: peer 1.1.1.1 is placed under maintenance.`
- `Rib: %BGP-6-MAINTENANCE-MODE: peer 1.1.1.1 is taken out of maintenance.`

9.5 Maintenance Mode Commands

Global Configuration Commands

- group bgp
- group interface
- maintenance

Group Configuration Commands

- interface
- neighbor
- maintenance profile interface
- maintenance profile bgp
- vrf

Maintenance Configuration Commands

- profile interface
- profile bgp
- profile unit
- unit
- interface
- bgp <peer> [vrf <vrf_name>]
- profile interface <profile_name> default
- profile bgp <profile_name> default
- profile unit <profile_name> default

Unit Configuration Commands

- group bgp <group_name>
- group interface <group_name>
- profile unit
- quiesce

Interface Profile Configuration Commands

- rate-monitoring load-interval
- rate-monitoring threshold
- shutdown max-delay

BGP Profile Configuration Commands

- initiator route-map <route-map-name> inout

Unit Profile Configuration Commands

- on-boot duration

EventMgr Configuration Commands

- trigger on-maintenance

Display Commands

- show maintenance
- show maintenance summary
- show maintenance units
- show maintenance interface
- show maintenance interface status
- show maintenance bgp

- show maintenance groups
- show maintenance profiles
- show maintenance stages
- show maintenance bgp receiver route-map
- show maintenance debug

Enhanced Commands to show Maintenance Status

- show interface
- show interface <intf_name> status
- show ip | ipv6 bgp summary [vrf <vrf_name>]
- show ip | ipv6 bgp neighbors <peer_addr> [vrf <vrf_name>]

group bgp

The **group bgp <group_name> command** places the switch in group-BGP configuration mode for configuring the members of a BGP group in a particular VRF and associating a BGP maintenance profile for these members.

The command creates the group if the specified group does not exist prior to issuing the command.

The **no group bgp <group_name>** and **default group bgp <group_name>** removes the BGP group.

Command Mode

Global Configuration

Command Syntax

```
group bgp group_name
no group bgp group_name
default group bgp group_name
```

Parameters

- *group_name* name of the BGP group

Commands available in group-BGP configuration mode:

- neighbor (ipv4 address | ipv6 address | peer-group)
- vrf (vrf-name)
- maintenance profile bgp

Note Built-in BGP groups like **AllBgpNeighborVrf-default** and **AllBgpNeighborVrf-<vrf_name>** do not allow neighbor configuration. Only BGP maintenance profile can be associated to them.

Example

- This command creates a BGP group BG1 and enters into group BGP BG1 configuration mode.

```
switch(config)#group bgp BG1
switch(config-group-bgp-BG1)# show active
group bgp BG1
    exit
switch(config-group-bgp-BG1)#
```

- This command enters into BGP built-in configuration mode for AllBgpNeighborVrf-default.

```
switch(config)#group bgp AllBgpNeighborVrf-default
switch(config-builtin-group-bgp-AllBgpNeighborVrf-default)#show active
group bgp AllBgpNeighborVrf-default
   exit
switch(config-builtin-group-bgp-AllBgpNeighborVrf-default)#exit
switch(config)#show maintenance groups bgp AllBgpNeighborVrf-default
BGP Group: AllBgpNeighborVrf-default
  Origin: Built-in
  Neighbors:
    Ipv4 Peers: 1.0.0.1, 1.0.1.2
  Bgp Profile: Default
  Vrf: default
  Units: System
switch(config)#
```

group interface

The **group interface command** places the switch in group-intf configuration mode for configuring the members of interface group and associating a BGP/interface maintenance profile for these members.

The command creates the group if the specified group does not exist prior to issuing the command.

The **no group interface <group_name>** and **default group interface <group_name>** removes the interface group.

Command Mode

Global Configuration

Command Syntax

```
group interface group_name
no group interface group_name
default group interface group_name
```

Parameters

- *group_name* name of the interface group

Commands available in group-BGP configuration mode:

- interface
- maintenance profile bgp
- maintenance profile interface

Note Built-in Interface groups like **AllEthernetInterface**, **Linecard3**, **Linecard4**, etc. do not allow interface configurations. Only BGP/interface maintenance profiles can be associated to them.

Example

- This command creates an interface group IG1 and enters into group interface IG1 configuration mode.

```
switch(config)#group interface IG1
switch(config-group-if-IG1)# show active
group interface IG1
    exit
switch(config-group-if-IG1)#
```

- This command enters into built-in interface group **AllEthernetInterface**.

```
switch(config)#group interface AllEthernetIntetrface
switch(config-builtin-group-if-AllEthernetInterface)#show active
group interface AllEthernetInterface
   exit
switch(config-builtin-group-if-AllEthernetInterface)#exit
switch(config)# show maintenance groups interface AllEthernetInterface
Interface Group: AllEthernetInterface
  Origin: Built-in
  Interfaces:
    Et1, Et2, Et3, Et4, Et5/1, … Et34, Et35, Et36
  Profiles:
    Interface Profile: Default
    Bgp Profile: Default
  Units: System#
```

maintenance

The **maintenance** command allows you to enter maintenance configuration mode and specify maintenance configuration options.

The **no maintenance** and **default maintenance** command removes the maintenance configuration from the *running-config*.

Command Mode

Global Configuration

Command Syntax

```
maintenance
no maintenance
default maintenance
```

Commands available in maintenance configuration mode:

- **unit**
- **bgp**
- **interface**
- **profile bgp**
- **profile interface**
- **profile unit**
- **profile interface <profile-name> default**
- **profile bgp <profile-name> default**
- **profile unit <profile-name> default**

Examples

- This example shows the commands to enter maintenance configuration mode and configure maintenance related parameters.

```
switch(config)#maintenance
switch(config-maintenance)#profile unit foo
switch(config-profile-unit-foo)#on-boot duration 300
switch(config-profile-unit-foo)#exit
switch(config-maintenance)#unit U1
switch(config-unit-U1)#group interface IG1
switch(config-unit-U1)#group bgp BG1
switch(config-unit-U1)#profile unit foo
switch(config-unit-U1)#exit
switch(config-maintenance)#show active
maintenance
    profile unit foo
        on-boot duration 300
    unit U1
        group interface IG1
        group bgp BG1
        profile unit foo
switch(config-maintenance)#
```

interface

The **interface** command adds interfaces to interface group.

The **no interface <intf-name>** and **default interface <intf-name>** removes the interface from the group.

Command Mode

Group-Interface Configuration

Command Syntax

```
interface interface-name
no interface interface-name
default interface interface-name
```

Parameters

- *interface-name* name of the interface
  - ethernet *e_range* Ethernet interfaces specified by e_range
  - port-channel *p_range* port channel interfaces specified by p_range
  - vlan *v_range* vlans specified by v_range.

Valid *e_range*, *p_range* and *v_range* formats include number, range, or comma-delimited list of numbers and ranges. Valid Ethernet numbers depend on the Ethernet interfaces available on the switch.

Example

- This command adds **Ethernet8**, **Ethernet9**, and **port-channel10** to the interface group **IG1**.

```
switch(config)#group interface IG1
switch(config-group-if-IG1)#interface Ethernet8-9
switch(config-group-if-IG1)#interface port-channel10
switch(config-group-if-IG1)#show active
group interface IG1
    interface Et8-9
    interface Po10
switch(config-group-if-IG1)#exit
switch(config)#
```

neighbor

The **neighbor** command adds BGP peer(s) to a BGP group. The neighbors can be IPv4, IPv6 or a peer-group. The **no neighbor <peer>** and **default neighbor <peer>** removes the BGP peer from the group.

Command Mode

Group-BGP Configuration

Command Syntax

neighbor *ipv4_addr*
no neighbor *ipv4_addr*
default neighbor *ipv4_addr*
neighbor *ipv6_addr*
no neighbor *ipv6_addr*
default neighbor *ipv6_addr*
neighbor *peer-group-name*
no neighbor *peer-group-name*
default neighbor *peer-group-name*

Parameters

- *ipv4_addr* BGP neighbor ipv4 address
- *ipv6_addr* BGP neighbor ipv6 address
- *peer-group-name* BGP peer group name

Example

- This command adds ipv4 peer 1.0.1.1, ipv6 peer 1::1 and peer-group PG to the BGP group **BG1**.

```
switch(config)#group bgp BG1
switch(config-group-bgp-BG1)#neighbor 1.0.1.1
switch(config-group-bgp-BG1)#neighbor 1::1
switch(config-group-bgp-BG1)#neighbor PG
switch(config-group-bgp-BG1)#group bgp BG1
switch(config-group-bgp-BG1)#neighbor 1.0.1.1
switch(config-group-bgp-BG1)#neighbor 1::1
switch(config-group-bgp-BG1)#neighbor PG
switch(config-group-bgp-BG1)#exit
switch(config)#
```

maintenance profile interface

The **maintenance profile interface <profile-name>** command associates interface profile to interface group.

The **no maintenance profile interface <profile-name>** and **default maintenance profile interface <profile-name>** removes the interface profile from interface group.

Command Mode

Group-Interface Configuration
Built-in-Group-Interface Configuration

Command Syntax

```
maintenance profile interface profile-name
no maintenance profile interface profile-name
default maintenance profile interface profile-name
```

Parameters

- *profile-name* name of the interface profile

Example

- This command adds interface profile **IP1** to interface group **IG1**.

```
switch(config)#group interface IG1
switch(config-group-if-IG1)#interface Ethernet8-9
switch(config-group-if-IG1)#maintenance profile interface IP1
switch(config-group-if-IG1)#show active
group interface IG1
    interface Et8-9
    maintenance profile interface IP1

switch(config-group-if-IG1)#
```

- This command adds interface profile **IP1** to built-in interface group **AllEthernetInterface**.

```
switch(config)#group interface AllEthernetInterface
switch(config-builtin-group-if-AllEtherentInterface)#maintenance profile
interface IP1
switch(config-builtin-group-if-AllEtherentInterface)#show active
group interface AllEthernetInterface
    maintenance profile interface IP1

switch(config-builtin-group-if-AllEtherentInterface)#
```

maintenance profile bgp

The **maintenance profile bgp <profile-name>** command associates a BGP maintenance profile to an interface/BGP group. A BGP profile can be associated to both the interface and BGP group.

The **no maintenance profile bgp <profile-name>** and **default maintenance profile bgp <profile-name>** removes the profile from the interface/BGP group.

Command Mode

Group-Interface Configuration
Group-BGP Configuration
Built-in-Group-Interface Configuration
Built-in-Group-BGP Configuration

Command Syntax

```
maintenance profile bgp profile-name
no maintenance profile bgp profile-name
default maintenance profile profile-name
```

Parameters

- *profile name* name of the BGP profile

Example

- This command adds BGP profile **BP1** to a BGP group **BG1**.

```
switch(config)#group bgp BG1
switch(config-group-bgp-BG1)#neighbor 1.0.1.1
switch(config-group-bgp-BG1)#neighbor 1::1
switch(config-group-bgp-BG1)#neighbor PG
switch(config-group-bgp-BG1)#maintenance profile bgp BP1
switch(config-group-bgp-BG1)#show active
group bgp BG1
    neighbor 1.0.1.1
    neighbor 1::1
    neighbor PG
    maintenance profile bgp BP1
switch(config-group-bgp-BG1)#exit
switch(config)#
```

- This command adds BGP profile **BP1** to interface group **IG1**.

```
switch(config)#group interface IG1
switch(config-group-if-IG1)#interface Ethernet8-9
switch(config-group-if-IG1)#maintenance profile bgp BP1
switch(config-group-if-IG1)#show active
group interface IG1
    interface Et8-9
    maintenance profile bgp BP1
switch(config-group-if-IG1)#exit
switch(config)#
```

- This command adds BGP profile **BP1** to built-in interface group **AllEthernetInterface**.

```
switch(config)#group interface AllEthernetInterface
switch(config-builtin-group-if-AllEtherentInterface)#maintenance profile bgp BP1
switch(config-builtin-group-if-AllEtherentInterface)#show active
group interface AllEthernetInterface
    maintenance profile bgp BP1

switch(config-builtin-group-if-AllEtherentInterface)#
```

vrf

The **vrf** command specifies the VRF for BGP group. All the neighbors configured in the BGP group are considered to be members of the BGP group in the particular VRF context.

The **no vrf <vrf-name>** and **default vrf <vrf-name>** removes the VRF configuration from the BGP group and sets the VRF context to “default”.

Command Mode

Group-BGP Configuration

Command Syntax

```
vrf vrf_name
no vrf vrf_name
default vrf vrf_name
```

Parameters

- *vrf_name* name of the VRF in a group belonging to neighbors in that group

Example

- This command specifies VRF **VRF1** for the neighbors in the BGP group **BGP1**.

```
switch(config)#group bgp BG1
switch(config-group-bgp-BG1)#neighbor 1.0.1.1
switch(config-group-bgp-BG1)#neighbor 1::1
switch(config-group-bgp-BG1)#neighbor PG
switch(config-group-bgp-BG1)#vrf VRF1
switch(config-group-bgp-BG1)#show active
group bgp BG1
    neighbor 1.0.1.1
    neighbor 1::1
    neighbor PG
    vrf VRF1
switch(config-group-bgp-BG1)#exit
switch(config)#
```

profile interface

The **profile interface** command places the switch in maintenance profile interface configuration mode for configuring rate-monitoring threshold, load-interval, and shutdown max-delay.

The command creates the profile if the specified interface profile does not exist prior to issuing the command.

The **no profile interface <profile-name>** and **default profile interface <profile-name>** removes the profile from running-config.

Command Mode

Maintenance Configuration

Command Syntax

```
profile interface profile-name
no profile interface profile-name
default profile interface profile-name
```

Parameters

- *profile-name* name of the interface profile

Commands available in maintenance profile interface configuration mode:

- **rate-monitoring load-interval**
- **rate-monitoring threshold**
- **shutdown max-delay**

Example

- This command creates interface profile **IP1**.

```
switch(config)#maintenance
switch(config-maintenance)#profile interface IP1
switch(config-profile-intf-IP1)#show active
maintenance
    profile interface IP1

switch(config-profile-intf-IP1)#
```

profile bgp

The **profile bgp** command places the switch in maintenance profile BGP configuration mode for configuring initiator route-map.

The command creates the profile if the specified BGP profile does not exist prior to issuing the command.

The **no profile bgp <profile-name>** and **default profile bgp <profile-name>** removes the profile from *running-config*.

Command Mode

Maintenance Configuration

Command Syntax

```
profile bgp profile-name
no profile bgp profile-name
default profile bgp profile-name
```

Parameters

- *profile-name* name of the BGP profile

Commands available in maintenance profile BGP configuration mode:

- **initiator route-map (route-map name) inout**

Example

- This command creates BGP profile **BP1**.

```
switch(config)#maintenance
switch(config-maintenance)#profile bgp BP1
switch(config-profile-bgp-BP1)#show active
maintenance
    profile bgp BP1

switch(config-profile-bgp-BP1)#
```

profile unit

The **profile unit** command places the switch in maintenance profile unit configuration mode for configuring on-boot duration.

The command creates the profile if the specified BGP profile does not exist prior to issuing the command.

The **no profile unit <profile-name>** and **default profile unit <profile-name>** removes the profile from *running-config*.

Command Mode

Maintenance Configuration

Command Syntax

profile unit *profile-name*
no profile unit *profile-name*
default profile unit *profile-name*

Parameters

- *profile-name* name of the unit profile

Commands available in maintenance profile unit configuration mode:

- **on-boot duration**

Example

- This command creates unit profile **UP1**.

```
switch(config)#maintenance
switch(config-maintenance)#profile unit UP1
switch(config-profile-unit-UP1)#show active
maintenance
    profile unit UP1

switch(config-profile-unit-UP1)#
```

unit

The **unit <unit_name>** command places the switch in maintenance unit configuration mode for configuring BGP/interface groups in the unit.

The command creates the unit if the specified unit profile does not exist prior to issuing the command.

The **no unit <unit-name>** and **default unit <unit-name>** removes the unit from *running-config*.

Command Mode

Maintenance Configuration

Command Syntax

```
unit linecard l_range | unit_name
no unit linecard l_range | unit_name
default unit linecard l_range | unit_name
```

Parameters

- *Linecard l_range* name of the Linecard built-in unit
- *0 l_range* linecards available on the switch
- *unit_name* name of the user-configured unit

Commands available in maintenance unit configuration mode:

- **group interface**
- group bgp
- profile unit
- quiesce

Note Built-in units like **System**, **Linecard3**, **Linecard4,** etc. do not allow group configuration but unit profile can be associated to these units.

Examples

- This command creates maintenance unit UNIT1.

```
switch(config)#maintenance
switch(config-maintenance)#unit UNIT1
switch(config-unit-UNIT1)#show active
maintenance
          unit UNIT1
switch(config-unit-UNIT1)#
```

- This command enters the built-in **Linecard1** unit configuration mode.

```
switch(config)#maintenance
switch(config-maintenance)#unit Linecard1
switch(config-builtin-unit-Linecard1)#show active
maintenance
          unit Linecard1
switch(config-builtin-unit-Linecard1)#
```

interface

The **interface <intf-name>** command places the switch in maintenance dynamic interface unit configuration mode.

The command creates the dynamic interface unit if the specified dynamic interface unit does not exist prior to issuing the command.

The **no interface <intf-name>** and **default interface <intf-name>** removes the dynamic interface unit from *running-config*.

Command Mode

Maintenance Configuration

Command Syntax

```
interface interface-name
no interface interface-name
default interface interface-name
```

Parameters

- *interface-name* name of the interface
- ethernet *e_range* Ethernet interfaces specified by e_range
- port-channel *p_range* port channel interfaces specified by p_range
- vlan *v_range* vlans specified by v_range.

Valid *e_range*, *p_range* and *v_range* formats include number, range, or comma-delimited list of numbers and ranges.

Note Different dynamic interface units are created for each interface in the range.

Commands available in maintenance dynamic interface unit configuration mode:

- quiesce

Example

- This command creates two dynamic interface units for interfaces Ethernet1-2 under maintenance configuration.

```
switch(config)#maintenance
switch(config-maintenance)#interface Ethernet1-2
switch(config-maint-if-Et1-2)#exit
switch(config-maintenance)#show active
maintenance
    interface Ethernet1
    !
    interface Ethernet2
switch(config-maintenance)#
```

bgp <peer> [vrf <vrf_name>]

The **bgp <peer> [vrf <vrf-name>]** command places the switch in maintenance dynamic BGP unit configuration mode. If no VRF is specified, the BGP peer is considered to be in the DEFAULT VRF, otherwise, in the specified VRF.

The command creates the dynamic BGP unit if the specified dynamic BGP unit does not exist prior to issuing the command.

The **no bgp <peer> [vrf <vrf_name>]** and **default bgp <peer> [vrf <vrf_name>]** removes the dynamic BGP unit from *running-config*.

Command Mode

Maintenance Configuration

Command Syntax

bgp *ipv4_addr [vrf <vrf_name>]*
bgp *ipv6_addr [vrf <vrf_name>]*
bgp *peer_group_name [vrf <vrf_name>]*
<no | default> bgp ***ipv4_addr|ipv6_addr|peer_group_name*** **[vrf <***vrf_name***>]**

Parameters

- *ipv4_addr* BGP neighbor IPv4 address
- *ipv6_addr* BGP neighbor IPv6 address
- *peer_group_name* BGP peer group name
- *vrf_name* name of the VRF to which the BGP peer belongs

Commands available in maintenance dynamic interface unit configuration mode:

- quiesce

Example

- This command creates dynamic BGP unit for IPv4 addr 1.0.1.1, IPv6 addr 1::1 with quiesce and peer-group PG in VRF VRF1 under maintenance configuration.

```
switch(config)#maintenance
switch(config-maintenance)#bgp 1.0.1.1
switch(config-maint-bgp-1.0.1.1)#exit
switch(config-maintenance)#bgp 1::1
switch(config-maint-bgp-1::1)#quiesce
switch(config-maint-bgp-1::1)#exit
switch(config-maintenance)#bgp PG vrf VRF1
switch(config-maint-bgp-PG)#exit
switch(config-maint-bgp-PG)#show active
maintenance
    bgp 1.0.1.1
    !
    bgp 1::1
        quiesce
    !
    bgp PG vrf VRF1
switch(config-maintenance)#
```

profile interface <profile_name> default

The **profile interface <profile_name> default** command configures a user-configured interface profile as default interface profile.

The **no profile interface <profile_name> default** and **default profile interface <profile_name> default** removes the user-configured interface profile as default interface profile.

Command Mode

Maintenance Configuration

Command Syntax

```
profile interface profile_name default
no profile interface profile_name default
default profile interface profile_name default
```

Parameters

- *profile_name* name of the interface profile

Example

- This command configures user configured interface profile **IP1** as default interface profile.

```
switch(config)#maintenance
switch(config-maintenance)#profile interface IP1
switch(config-profile-intf-IP1)#rate-monitoring load-interval 100
switch(config-profile-intf-IP1)#rate-monitoring threshold 500
switch(config-profile-intf-IP1)#shutdown max-delay 100
switch(config-profile-intf-IP1)#exit
switch(config-maintenance)#
switch(config-maintenance)#show maintenance profile interface default
Interface Profile: Default
  Rate Monitoring:
    load-interval: 60 seconds
    threshold (in/out): 100 kbps
  shutdown:
    enabled: no
    max-delay: 300 seconds

switch(config-maintenance)#
switch(config-maintenance)#profile interface IP1 default
switch(config-maintenance)#show maintenance profile interface default
Interface Profile: IP1
  Rate Monitoring:
    load-interval: 100 seconds
    threshold (in/out): 500 kbps
  shutdown:
    enabled: yes
    max-delay: 100 seconds
switch(config-maintenance)#
switch(config-maintenance)#show active
maintenance
    profile interface IP1 default
    profile interface IP1
        rate-monitoring load-interval 100
        rate-monitoring threshold 500
        shutdown max-delay 100

switch(config-maintenance)#
```

profile bgp <profile_name> default

The **profile bgp <profile_name> default** command configures a user-configured BGP profile as default BGP profile.

The **no profile bgp <profile_name> default** and **default profile bgp <profile_name> default** removes the user-configured BGP profile as default BGP profile.

Command Mode

Maintenance Configuration

Command Syntax

```
profile bgp profile_name default
no profile bgp profile_name default
default profile bgp profile_name default
```

Parameters

- *profile_name* name of the BGP profile

Example

- This command configures user configured BGP profile **BP1** as default BGP profile.

```
switch(config)#maintenance
switch(config-maintenance)#profile bgp BP1
switch(config-profile-bgp-BP1)#initiator route-map RM1 inout
switch(config-profile-bgp-BP1)#exit
switch(config-maintenance)#
switch(config-maintenance)#show maintenance profile bgp default
Bgp Profile: Default
  Initiator route-map: SystemGenerated
   route-map SystemGenerated permit 10
     Description:
       description System generated initiator route-map
     Match clauses:
     Set clauses:
       set community GSHUT additive
       set local-preference 0

switch(config-maintenance)#
switch(config-maintenance)#profile bgp BP1 default
switch(config-maintenance)#show maintenance profile bgp default
Bgp Profile: BP1
  Initiator route-map: RM1
switch(config-maintenance)#
switch(config-maintenance)#show active
maintenance
    profile bgp BP1
        initiator route-map RM1 inout
    profile bgp BP1 default

switch(config-maintenance)#
```

profile unit <profile_name> default

The **profile unit <profile_name> default** command configures a user-configured unit profile as default unit profile.

The **no profile unit <profile_name> default** and **default profile unit <profile_name> default** removes the user-configured unit profile as default unit profile.

Command Mode

Maintenance Configuration

Command Syntax

profile unit *profile_name* **default**
no profile unit *profile_name* **default**
default profile unit *profile_name* **default**

Parameters

- *profile_name* name of the interface profile

Example

- This command configures user-configured unit profile **UP1** as the default unit profile.

```
switch(config)#maintenance
switch(config-maintenance)#profile unit UP1
switch(config-profile-unit-UP1)#on-boot duration 1000
switch(config-profile-unit-UP1)#exit
switch(config-maintenance)#
switch(config-maintenance)#show maintenance profiles unit default
Unit Profile: Default
    On-boot:
         enabled: no
         duration: 300 seconds

switch(config-maintenance)#profile unit UP1 default
switch(config-maintenance)#show maintenance profile unit default
Unit Profile: UP1
    On-boot:
         enabled: yes
         duration: 1000 seconds
switch(config-maintenance)#
switch(config-maintenance)#show active
maintenance
    profile unit UP1 default
    profile unit UP1
         on-boot duration 1000

switch(config-maintenance)#
```

group bgp <group_name>

The **group bgp <group_name>** command adds a BGP group to a unit.

The **no group bgp <group_name>** and **default group bgp <group_name>** removes the BGP group from a unit.

Command Mode

Maintenance Unit Configuration

Command Syntax

group bgp *group_name*
no group bgp *group_name*
default group bgp *group_name*

Parameters

- *group_name* name of the BGP group

Example

- This command adds a BGP group **BG1** to unit UNIT1.

```
switch(config)#maintenance
switch(config-maintenance)# unit UNIT1
switch(config-unit-UNIT1)# group bgp BG1
switch(config-unit-UNIT1)# show active
maintenance
       unit UNIT1
          group bgp BG1
switch(config-unit-UNIT1)
```

group interface <group_name>

The **group interface <group_name>** command adds an interface to a unit.

The **no group interface <group_name>** and **default group interface <group_name>** removes the interface group from a unit.

Command Mode

Maintenance Unit Configuration

Command Syntax

group interface *group_name*
no group interface *group_name*
default group interface *group_name*

Parameters

- *group_name* name of the interface group

Example

- This command adds an interface group IG1 to unit UNIT1.

```
switch(config)#maintenance
switch(config-maintenance)# unit UNIT1
switch(config-unit-UNIT1)# group interface IG1
switch(config-unit-UNIT1)# show active
maintenance
       unit UNIT1
          group interface IG1
switch(config-unit-UNIT1)
```

profile unit <profile_name>

The **profile unit <profile_name>** command associates unit profile to a particular unit.

The **no profile unit <profile_name>** and **default profile unit <profile_name>** removes the unit profile from a unit.

Command Mode

Maintenance-Unit Configuration
Maintenance-Built-in-Unit Configuration

Command Syntax

profile unit *profile-name*
no profile unit
default profile unit

Parameters

- *profile-name* name of the unit profile

Example

- This command adds unit profile UP1 to **UNIT1**.

```
switch(config)#maintenance
switch(config-maintenance)#unit UNIT1
switch(config-unit-UNIT1)#group interface IG1
switch(config-unit-UNIT1)#exit
switch(config-maintenance)#show maintenance units UNIT1
Unit Name: UNIT1
   Origin: User Configured
   Status: Not Under Maintenance
   Unit Profile: Default
   Time Since Last State Change: never
   Interface Groups:
     IG1

switch(config-maintenance)#unit UNIT1
switch(config-unit-UNIT1)#profile unit UP1
switch(config-unit-UNIT1)#show maintenance units UNIT1
Unit Name: UNIT1
   Origin: User Configured
   Status: Not Under Maintenance
   Unit Profile: UP1
   Time Since Last State Change: never
   Interface Groups:
     IG1
switch(config-unit-UNIT1)#show active
maintenance
    unit UNIT1
        group interface IG1
        profile unit UP1

switch(config-unit-UNIT1)#
```

- This command adds unit profile **UP2** to built-in unit **System**.

```
switch(config)#maintenance
switch(config-maintenance)#profile unit UP2
switch(config-profile-unit-UP2)#on-boot duration 600
switch(config-profile-unit-UP2)#exit
switch(config-maintenance)#
switch(config-maintenance)#unit System
switch(config-builtin-unit-System)#show active
maintenance
    unit System
switch(config-builtin-unit-System)#exit
switch(config-maintenance)#show maintenance units System
Unit Name: System
   Origin: Built-in
   Status: Not Under Maintenance
   Unit Profile: Default
   Time Since Last State Change: never
   Interface Groups:
     AllEthernetInterface

switch(config-maintenance)#
switch(config-maintenance)#unit System
switch(config-builtin-unit-System)#profile unit UP2
switch(config-builtin-unit-System)#show active
maintenance
    unit System
        profile unit UP2
switch(config-builtin-unit-System)#exit
switch(config-maintenance)#show maintenance units System
Unit Name: System
   Origin: Built-in
   Status: Not Under Maintenance
   Unit Profile: UP2
   Time Since Last State Change: never
   Interface Groups:
     AllEthernetInterface

switch(config-maintenance)#
```

quiesce

The **quiesce** command places a unit or dynamic interface/BGP unit into maintenance mode, gracefully transitioning traffic away from it.

The **no quiesce** and **default quiesce** exits the unit from maintenance.

Command Mode

Maintenance-Unit Configuration
Maintenance-Built-in-Unit Configuration
Maintenance Dynamic-Interface Unit Configuration
Maintenance Dynamic-Bgp Unit Configuration

Command Syntax

```
quiesce
no quiesce
default quiesce
```

Examples

- This command places unit **UNIT1**, interface **Et1**, BGP peer 1.0.1.1 in VRF default, BGP peer 1::1 in VRF **VRF1** into maintenance.

```
switch(config)#group interface IG1
switch(config-group-if-IG1)#interface Ethernet3-6
switch(config-group-if-IG1)#maintenance profile interface IP1
switch(config-group-if-IG1)#exit
switch(config)#maintenance
switch(config-maintenance)#unit UNIT1
switch(config-unit-UNIT1)#group interface IG1
switch(config-unit-UNIT1)#quiesce
switch(config-unit-UNIT1)#exit
switch(config-maintenance)#interface Ethernet1
switch(config-maint-if-Et1)#quiesce
switch(config-maint-if-Et1)#exit
switch(config-maintenance)#bgp 1.0.1.1
switch(config-maint-bgp-1.0.1.1)#quiesce
switch(config-maint-bgp-1.0.1.1)#exit
switch(config-maintenance)#bgp 1::1 vrf VRF1
switch(config-maint-bgp-1::1)#quiesce
switch(config-maint-bgp-1::1)#exit
switch(config-maintenance)#show active
maintenance
    bgp 1.0.1.1
        quiesce
    !
    bgp 1::1 vrf VRF1
        quiesce
    interface Et1
        quiesce
    unit UNIT1
        quiesce

switch(config-maintenance)#show maintenance
Flags:
o - On-boot maintenance
v - Violating traffic threshold

Unit Name              Status                   Time since last change    Flags
---------------------- ------------------------ ------------------------- -----
System                 Not Under Maintenance            never
UNIT1                  Under Maintenance             0:00:06 ago

Interface Name         Status                   Time since last change    Flags
---------------------- ------------------------ ------------------------- -----
Ethernet1              Entering Maintenance          0:00:06 ago

Bgp Neighbor(vrf: defa Status                   Time since last change    Flags
---------------------- ------------------------ ------------------------- -----
1.0.1.1                Under Maintenance             0:00:06 ago

Bgp Neighbor(vrf: VRF1 Status                   Time since last change    Flags
---------------------- ------------------------ ------------------------- -----
1::1                   Under Maintenance             0:00:06 ago

switch(config-maintenance)#
```

rate-monitoring load-interval

The **rate-monitoring load-interval** command is a maintenance interface profile configuration option for configuring the interface's rate monitoring load interval with a load interval value between 5 and 600 seconds.

Command Mode

Maintenance-Profile-Interface Configuration

Command Syntax

```
rate-monitoring load-interval load_interval
no rate-monitoring load-interval
default rate-monitoring load-interval
```

Parameters

- *load_interval* load interval value between 5 and 600 seconds

Example

- This command configures the rate monitoring load interval for the profile interface **IP1** to a load interval of 10 seconds.

```
switch(config)#maintenance
switch(config-maintenance)#profile interface IP1
switch(config-profile-intf-IP1)#rate-monitoring load-interval 10
switch(config-profile-intf-IP1)#show active
maintenance
    profile interface IP1
        rate-monitoring load-interval 10

switch(config-profile-intf-IP1)#
```

rate-monitoring threshold

The **rate-monitoring threshold** command is a maintenance interface profile configuration option for configuring the interface's rate monitoring threshold with a threshold value between 1 and 4294967295 kilobytes.

The **no rate-monitoring threshold** and **default rate-monitoring threshold** removes this configuration from the interface profile.

Command Mode

Maintenance-Profile-Interface Configuration

Command Syntax

rate-monitoring threshold *threshold_in_kbps*
no rate-monitoring threshold
default rate-monitoring threshold

Parameters

- *threshold_in_kbps* threshold in kilobytes per second (kbps) between 1 and 4294967295 kilobytes

Example

- This command configures the rate monitoring threshold for the profile interface **IP1** to a threshold of 1000 kilobytes per second (kbps).

```
switch(config)#maintenance
switch(config-maintenance)#profile interface IP1
switch(config-profile-intf-IP1)#rate-monitoring threshold 1000
switch(config-profile-intf-IP1)# show active
maintenance
    profile interface IP1
        rate-monitoring threshold 1000

switch(config-profile-intf-IP1)#
```

shutdown max-delay

The **shutdown max-delay** command is a maintenance interface profile configuration option for configuring the maximum duration after which the interface is shutdown with a value between 1 and 4294967295 seconds.

The **no shutdown** and **default shutdown** removes this configuration from the interface profile.

Command Mode

Maintenance-Profile-Interface Configuration

Command Syntax

shutdown max-delay *delay*
no shutdown max-delay *delay*
default shutdown max-delay *delay*

Parameters

- *delay* maximum shutdown delay between 1 and 4294967295 seconds

Example

- This command configures the shutdown max-delay for the profile interface IP1 to 500 seconds or one hour.

```
switch(config)#maintenance
switch(config-maintenance)#profile interface IP1
switch(config-profile-intf-IP1)#shutdown max-delay 500
switch(config-profile-intf-IP1)#show active
maintenance
    profile interface IP1
        shutdown max-delay 500

switch(config-profile-intf-IP1)#
```

initiator route-map <route-map-name> inout

The **initiator route-map <route-map-name> inout** command is a maintenance BGP profile configuration option for assigning the initiator route-map, which will be applied to inout (inbound and outbound).

The **no initiator route-map <route-map-name> inout** and **default initiator route-map <route-map-name> inout** removes this configuration from the BGP profile.

Command Mode

Maintenance-Profile-BGP Configuration

Command Syntax

```
initiator route-map route-map-name inout
no initiator route-map
default initiator route-map
```

Parameters

- *route-map-name* initiator route-map name

Example

- This command configures initiator route-map **RM1** within a BGP profile **BP1**.

```
switch(config)#maintenance
switch(config-maintenance)#profile bgp BP1
switch(config-profile-bgp-BP1)#initiator route-map RM1 inout
switch(config-profile-bgp-BP1)#show active
maintenance
    profile bgp BP1
        initiator route-map RM1 inout

switch(config-profile-bgp-BP1)#
```

on-boot duration

The **on-boot duration** command is a maintenance unit profile configuration option for specifying the duration after which the associated unit will be brought out of maintenance after reboot. The on-boot property in the maintenance unit profile specifies that the unit will be placed into maintenance mode as part of boot-up, and remain so for the specified duration.

The **no on-boot** and **default on-boot** removes this configuration from the unit profile.

Command Mode

Maintenance-Profile-Unit Configuration

Command Syntax

on-boot duration *duration*
no on-boot
default on-boot

Parameters

- *duration* number of seconds for which unit will remain under maintenance after reboot (from 300 to 3600 seconds)

Example

- This command configures on-boot duration of 1000 seconds in profile unit **UP1**.

```
switch(config)#maintenance
switch(config-maintenance)#profile unit UP1
switch(config-profile-unit-UP1)#on-boot duration 1000
switch(config-profile-unit-UP1)#show active
maintenance
    profile unit UP1
        on-boot duration 1000

switch(config-profile-unit-UP1)#
```

trigger on-maintenance

The **trigger on-maintenance** command is an event handler configuration for triggering actions during the maintenance operation of a unit, interface and BGP peer at specified stages.

The event-handler configuration takes effect only after exiting the event-handler configuration mode.

Command Mode

Event-handler Configuration

Command Syntax

```
trigger on-maintenance <enter | exit> <unit <unit_name> | bgp
<ipv4_addr|ipv6_addr| peer_group> [vrf <vrf_name>] | interface <intf_name>>
<begin | end | all |<before | after> stage <stage_name>>
```

Parameters

- *enter* trigger on-maintenance event-handler on maintenance enter operation
- *exit* trigger on-maintenance event-handler on maintenance exit operation
- *bgp* trigger event-handler on dynamic BGP unit maintenance operation
  - *pv4_addr* BGP neighbor ipv4 address
  - *pv6_addr* BGP neighbor ipv6 address
  - *peer_group* BGP peer group name
- *vrf vrf_name* name of the VRF to which BGP peer belongs
- *interface* trigger event-handler on dynamic interface unit maintenance operation
  - *intf_name* name of the interface
    - *ethernet* trigger event-handler on specified Ethernet interface
    - *port-channel* trigger event-handler on specified port channel interface
    - *vlan* trigger event-handler on specified vlan

Note Comma-delimited list, ranges are not supported.

- *unit* trigger event-handler on maintenance operation of unit
- *begin* action is triggered in the beginning of maintenance operation
- *end* action is triggered at the end of maintenance operation
- *stage_name* action is triggered at specified stage
  - bgp and ratemon are the two stages
- *all* action is triggered at all the stages
- *before* action is triggered before the specified stage
- *after* action is triggered after the specified stage

Examples

- This command configures event-handler **E1**, which triggers on maintenance an enter operation of unit **UNIT1** at all the stages.

```
switch(config)#event-handler E1
switch(config-handler-E1)#trigger on-maintenance enter unit UNIT1 all
switch(config-handler-E1)#action bash FastCli -c "show maintenance"
switch(config-handler-E1)# exit
switch(config)# show event-handler E1
Event-handler E1
Trigger: Asynchronous on-maintenance enter unit UNIT1 all delay 0 seconds
Threshold Time Window: 0 Seconds, Event Count: 1 times
Action: FastCli -c "show maintenance"
Action expected to finish in less than 10 seconds
Last Trigger Detection Time: Never
Total Trigger Detections: 0
Last Trigger Activation Time: Never
Total Trigger Activations: 0
Last Action Time: Never
Total Actions: 0

switch(config)#
```

- This command configures event-handler **E2**, which triggers on maintenance an exit operation of dynamic interface unit **Ethernet1** before stage bgp.

```
switch(config)#event-handler E2
switch(config-handler-E2)#trigger on-maintenance exit interface Ethernet1 before
stage bgp
switch(config-handler-E2)#action bash FastCli -c "show maintenance summary"
switch(config-handler-E2)# exit
switch(config)# show event-handler E2
Event-handler E2
Trigger: Asynchronous on-maintenance exit interface Ethernet1 before stage bgp
delay 0 seconds
Threshold Time Window: 0 Seconds, Event Count: 1 times
Action: FastCli -c "show maintenance summary"
Action expected to finish in less than 10 seconds
Last Trigger Detection Time: Never
Total Trigger Detections: 0
Last Trigger Activation Time: Never
Total Trigger Activations: 0
Last Action Time: Never
Total Actions: 0

switch(config)#
```

- This command configures event-handler **E3**, which triggers on maintenance an enter operation of dynamic BGP unit 1::1 in VRF **VRF1** at the last stage end.

```
switch(config)#event-handler E3
switch(config-handler-E3)#trigger on-maintenance enter bgp 1::1 vrf VRF1 end
switch(config-handler-E3)#action bash FastCli -c "show maintenance bgp ip all vrf
all"
switch(config-handler-E3)# exit
switch(config)# show event-handler E3
Event-handler E3
Trigger: Asynchronous on-maintenance enter bgp 1::1 vrf VRF1 end delay 0 seconds
Threshold Time Window: 0 Seconds, Event Count: 1 times
Action: FastCli -c "show maintenance bgp ip all vrf all"
Action expected to finish in less than 10 seconds
Last Trigger Detection Time: Never
Total Trigger Detections: 0
Last Trigger Activation Time: Never
Total Trigger Activations: 0
Last Action Time: Never
Total Actions: 0

switch(config)#
```

show maintenance

The **show maintenance** command provides brief information about all units/dynamic interface unit/dynamic bgp unit and status.

'o' - flag displays that unit is undergoing or has undergone a maintenance operation because of on-boot.

'v' - flag displays that one/some of the interfaces are violating traffic, i.e. traffic for those interfaces is above threshold.

Command Mode

EXEC

Command Syntax

show maintenance

Examples

- This command displays maintenance mode details.

```
switch#show maintenance
Flags:
o - On-boot maintenance
v - Violating traffic threshold
  Unit Name              Status                   Time since last change  Flags
---------------------- ---------------------- -------------------------- -----
  System                 Not Under Maintenance             never
  Foo                    Under Maintenance              0:00:40 ago        o

  Interface Name         Status                   Time since last change    Flags
---------------------- ---------------------- -------------------------- -----
  Ethernet16/1           Entering Maintenance           0:00:02 ago        v

  Bgp Neighbor(vrf: defa Status                   Time since last change    Flags
---------------------- ---------------------- -------------------------- -----
  1.0.0.2                Not Under Maintenance             never

  Bgp Neighbor(vrf: red) Status                   Time since last change    Flags
---------------------- ---------------------- -------------------------- -----
  2.0.1.2                Under Maintenance              0:00:16 ago

switch#
```

show maintenance summary

The **show maintenance summary** command displays summarized information about the maintenance mode operations such as number of units configured, number of units Entering/Exiting maintenance etc.

Command Mode

EXEC

Command Syntax

show maintenance summary

Example

- This command displays summary of maintenance mode operations.

```
switch#show maintenance summary
Number of Units Configured: 0
Number of Units Exiting Maintenance: 0
Number of Units Entering Maintenance: 0
Number of Units Not Under Maintenance: 1
Number of Units Under Maintenance: 0
Directly Put Under Maintenance:
  Number of interfaces Entering Maintenance: 0
  Number of interfaces Under Maintenance: 1
  Number of bgp peers Entering Maintenance: 0
  Number of bgp peers Under Maintenance: 1
Rate Monitoring:
  Number of interfaces Entering Maintenance: 0
  Number of interfaces Under Maintenance: 1
  Number of interfaces Under Maintenance with threshold violation: 0
  Number of interfaces shutdown for maintenance: 0

switch#
```

show maintenance units

The **show maintenance units** command displays detailed information about the particular unit.

Command Mode

EXEC

Command Syntax

show maintenance units *[unit_name]*

Parameters

- *unit_name* name of unit

Example

- This command displays maintenance units details.

```
switch#show maintenance units
Unit Name: System
   Origin: Built-in
   Status: Not Under Maintenance
   Unit Profile: Default
   Time Since Last State Change: never
   Bgp Groups:
     AllBgpNeighborVrf-default
   Interface Groups:
     AllEthernetInterface

Unit Name: UNIT1
   Origin: User Configured
   Status: Under Maintenance
   Unit Profile: UP1
   Time Since Last State Change: 0:00:08 ago
   Bgp Groups:
     BG1
   Interface Groups:
     IG1
   History:
    2016-08-29 23:05:30  old state: 'maintenanceModeEnter' to new state:
'underMaintenance' 0:00:08 ago
   2016-08-29 23:05:30  old state: 'active' to new state: 'maintenanceModeEnter'
0:00:08 ago

switch#
```

show maintenance interface

The **show maintenance interface** command displays detailed information about interfaces and their maintenance status with traffic rates.

Command Mode

EXEC

Command Syntax

show maintenance interface *[<intf_name> [detail] | detail]*

Parameters

- *intf_name* name of the interface
  - ethernet *e_range* Ethernet interfaces specified by e_range
  - port-channel *p_range* port channel interfaces specified by p_range
  - vlan *v_range* vlans specified by v_range

Valid *e_range*, *p_range* and *v_range* formats include number, range, or comma-delimited list of numbers and ranges.

- *detail* provides the detailed rate-monitoring information

Example

- This command displays interface status and traffic rates.

```
switch#show maintenance interface
Flags:
v - Violating traffic threshold
s - Shutdown for maintenance
                                                    Rate (Mbps)
   Interface            Status                      In         Out    Flags
------------------- --------------------------- --------- ---------- ------
   Ethernet1            Not Under Maintenance        -           -
   Ethernet2            Not Under Maintenance        -           -
   Ethernet3            Under Maintenance          0.0         0.0
   Ethernet4            Not Under Maintenance        -           -
  ...
   Ethernet35           Entering Maintenance       8.7         2.9
   Ethernet36           Not Under Maintenance        -           -

switch#
```

- This command displays detailed information about the interface Ethernet16/1.

```
switch#show maintenance interface Ethernet16/1 detail
Ethernet16/1 is Under Maintenance
  Groups: AllEthernetInterface
  Selected profiles from Interface groups:
    Interface Maintenance profile: low-load-interval-profile
    Bgp Maintenance profile: Default
  Bgp:
    Maintenance State: Under Maintenance
    Vrf: default
      Neighbor: 1.0.1.2
        Maintenance routemap: SystemGenerated
  Rate Monitoring:
    Passive monitoring since 0:42:25 ago
    Total samples taken: 236
      Before Maintenance:
        Below threshold: 1
        Above threshold: 0
      After Maintenance:
        Below threshold: 235
        Above threshold: 0
    Last sample information:
      Sample taken 0:00:04 ago
      In:  0.0 Mbps
      Out: 0.0 Mbps
switch#
```

show maintenance interface status

The **show maintenance interface status** command displays maintenance status and rates for interfaces.

Command Mode

EXEC

Command Syntax

show maintenance interface status *active* | *entering* | *exiting* | *quiesced*

Parameters

- *active* interfaces which are active
- *entering* interface which are entering maintenance
- *exiting* interface which are exiting maintenance
- *quiesced* interface which are under maintenance

Example

- This command displays interface status and traffic rates of interfaces which are quiesced.

```
switch#show maintenance interface status quiesced
Flags:
v - Violating traffic threshold
s - Shutdown for maintenance
                                                  Rate (Mbps)
   Interface            Status                    In         Out    Flags
------------------- --------------------------- --------- ---------- ------
   Ethernet1            Not Under Maintenance       -           -
   Ethernet2            Not Under Maintenance       -           -
   Ethernet3            Not Under Maintenance       -           -
   Ethernet4            Not Under Maintenance       -           -
   Ethernet16/1         Under Maintenance         0.0         0.0
   Port-Channel10       Under Maintenance         100.5       50.5    v
   Port-Channel11       Entering Maintenance      15.5        10.5
   Port-Channel10       Under Maintenance           -           -

switch#
```

show maintenance bgp

The **show maintenance bgp** command displays detailed maintenance information about BGP peers.

Command Mode

EXEC

Command Syntax

show maintenance bgp <*ipv4_addr*> *[vrf* <*vrf_name*>*]* | <*ipv6_addr*> *[vrf* <*vrf_name*>*]* | <*peer_group*> *[vrf* <*vrf_name*>*]* | **ip all** *[* **vrf** <*vrf_name*> | *vrf all]* | **ipv6 all** *[* **vrf** <*vrf_name*> | **vrf all** *]*

Parameters

- *ipv4_addr* BGP neighbor ipv4 address
- *ipv6_addr* BGP neighbor ipv6 address
- *peer_group* BGP peer group name
- *vrf_name* name of the VRF to which peer belongs
- **ip all vrf** *vrf_name* all ipv4 peers in specified VRF
- **ipv6 all vrf** *vrf_name* all ipv6 peers in specified VRF
- **ip all vrf all** all ipv4 peers in all the VRFs
- **ipv6 all vrf all** all ipv6 peers in all the VRFs

Example

- This command displays maintenance information about BGP peers 1.0.0.1 and 1.0.1.1 and maintenance route-map applied.

```
switch#show maintenance bgp ip all vrf all
BGP peer maintenance information for VRF default
Router identifier 0.0.1.1, local AS number 200
  Neighbor: 1.0.0.1
    Maintenance state: Under Maintenance
    Maintenance route-map: SystemGenerated
  Neighbor: 1.0.1.2
    Maintenance state: Under Maintenance
    Maintenance route-map: SystemGenerated

switch#
```

show maintenance groups

The **show maintenance groups** command displays all the interface/BGP groups along with their members and associated profiles.

Command Mode

EXEC

Command Syntax

show maintenance groups *interface* | *bgp* *<group_name>*

Parameters

- *interface* display only interface groups
- *bgp* display only BGP groups
- *group_name* name of the group

Example

- This command displays group details for built-in interface group **AllEthernetInterface** and built-in BGP group **AllBgpNeighborVrf-default** and user-configured interface group **IG1**.

```
switch#show maintenance groups
Interface Group: AllEthernetInterface
  Origin: Built-in
  Interfaces:
    Et1, Et2, Et3, Et4, Et5/1, Et5/2, Et5/3, Et5/4, Et6/1, Et6/2, Et6/3, Et6/4,
Et7/1, Et7/2, Et7/3, Et7/4, Et8/1, Et8/2, Et8/3, Et8/4, Et9/1, Et9/2, Et9/3,
Et9/4, Et10/1, Et10/2, Et10/3, Et10/4, Et11/1, Et11/2, Et11/3, Et11/4, Et12/1,
Et12/2, Et12/3, Et12/4, Et13/1, Et13/2, Et13/3, Et13/4, Et14/1, Et14/2, Et14/3,
Et14/4, Et15/1, Et15/2, Et15/3, Et15/4, Et16/1, Et16/2, Et16/3, Et16/4, Et17/1,
Et17/2, Et17/3, Et17/4, Et18/1, Et18/2, Et18/3, Et18/4, Et19/1, Et19/2, Et19/3,
Et19/4, Et20/1, Et20/2, Et20/3, Et20/4, Et21/1, Et21/2, Et21/3, Et21/4, Et22/1,
Et22/2, Et22/3, Et22/4, Et23/1, Et23/2, Et23/3, Et23/4, Et24/1, Et24/2, Et24/3,
Et24/4, Et25/1, Et25/2, Et25/3, Et25/4, Et26/1, Et26/2, Et26/3, Et26/4, Et27/1,
Et27/2, Et27/3, Et27/4, Et28/1, Et28/2, Et28/3, Et28/4, Et29, Et30, Et31, Et32,
Et33, Et34, Et35, Et36
  Profiles:
    Interface Profile: low-load-interval-profile
    Bgp Profile: Default
  Units: System
Interface Group: IG1
  Origin: User Configured
  Interfaces:
    Et1, Et2, Et3, Et4, Po10, Po11, Po12
  Profiles:
    Interface Profile: IP1
    Bgp Profile: BP1
  Units: UNIT1
Bgp Group: AllBgpNeighborVrf-default
  Origin: Built-in
  Neighbors:
    Ipv4 Peers: 1.0.0.1, 1.0.1.2
  Bgp Profile: Default
  Vrf: default
  Units: System

switch#
```

show maintenance profiles

The **show maintenance profiles** command displays all the interface/BGP/unit profiles configuration.

Command Mode

EXEC

Command Syntax

show maintenance profiles *interface* | *bgp* | *unit* <*profile_name*>

Parameters

- *interface* display only interface profiles
- *bgp* display only BGP profiles
- *unit* display only unit profiles
- *profile_name* name of the profile

Example

- This command displays profile configuration details for interface profile **IP1**, unit profile **UP1** and BGP profile **BP1**.

```
switch #show maintenance profiles
Interface Profile: IP1
   Rate Monitoring:
     load-interval: 444 seconds
     threshold (in/out): 4000 Kbps
   shutdown:
     enabled: yes
     max-delay: 399 seconds
Bgp Profile: BP1
   Initiator route-map:
     name: RM1
Unit Profile: UP1
   On-boot:
     enabled: yes
     duration: 340 seconds

switch #
```

show maintenance stages

The **show maintenance stages** command displays stages of maintenance operation while entering/exiting maintenance.

Command Mode

EXEC

Command Syntax

show maintenance stages [enter | exit]

Parameters

- *enter* display maintenance stages during maintenance enter operation
- *exit* display maintenance stages during maintenance exit operation

Example

- This command displays maintenance mode stages details.

```
switch #show maintenance stages
   No.       Stage      Description
--------- ------------ -------------------------
   1         bgp        BGP Maintenance processing
   2         ratemon    Interface Rate Monitoring

Maintenance Exit Stage Sequence
   No.       Stage      Description
--------- ------------ -------------------------
   1         ratemon    Interface Rate Monitoring
   2         bgp        BGP Maintenance processing

switch #
```

- This command displays maintenance mode stage details during entry.

```
switch #show maintenance stages enter
   No.       Stage      Description
--------- ------------ -------------------------
   1         bgp        BGP Maintenance processing
   2         ratemon    Interface Rate Monitoring
switch#
```

show maintenance bgp receiver route-map

The **show maintenance bgp receiver route-map** command displays receiver route-map which is applied during maintenance operation.

Command Mode

EXEC

Command Syntax

show maintenance bgp receiver route-map

Example

- This command displays receiver route-map contents.

```
switch#show maintenance bgp receiver route-map
route-map SystemGenerated permit 10
  Description:
    description System generated receiver route-map
  Match clauses:
    match community GSHUT-LIST
  SubRouteMap:
  Set clauses:
route-map SystemGenerated permit 50
  Description:
    description System generated receiver route-map
  Match clauses:
  SubRouteMap:
  Set clauses:switch#
```

show maintenance debug

The **show maintenance debug** command displays the history of various maintenance operations on a unit/interface/BGP peer.

Command Mode

EXEC

Command Syntax

show maintenance debug bgp [peer_name] | interface [intf_name] | units [unit_name]

Parameters

- *bgp* display history of all dynamic BGP units which have undergone maintenance operation
- *interface* display history of all dynamic interface units which have undergone maintenance operation
- *units* display history of all units which have undergone maintenance operation
- *peer_name* name of the peer
  - **ipv4_addr** BGP neighbor IPv4 address
  - **ipv6_addr** BGP neighbor IPv6 address
  - **peer-group-name** BGP peer group name
- *intf_name* name of the interface
  - **ethernet** *e_range* Ethernet interfaces specified by e_range
  - **port-channel** *p_range* port channel interfaces specified by p_range
  - **vlan** *v_range* vlans specified by v_range

Note Valid *e_range*, *p_range* and *v_range* formats include number, range, or comma-delimited list of numbers and ranges. Valid Ethernet numbers depend on the Ethernet interfaces available on the switch.

- *unit_name* name of the unit

Example

- This command displays history of maintenance operation on Ethernet 16/1.

```
switch#show maintenance debug interface Ethernet 16/1-4
Interface Ethernet16/1
  History:
   Maintenance Enter Stage Progression started 4:07:07 ago @ 2016-08-29 22:38:54
     0.000000   maintEnter stages started
     0.000091   stage begin started
     0.000151     event begin:EventMgr started
     0.004222     event begin:EventMgr completed
     0.004256   stage begin is complete
     0.004315   stage before_bgp started
     0.004368     event before_bgp:EventMgr started
     0.005820     event before_bgp:EventMgr completed
     0.005843   stage before_bgp is complete
     0.005904   stage bgp started
     0.005947     event bgp:Rib started
     0.013821     event bgp:Rib completed
     0.013855   stage bgp is complete
     0.013921   stage after_bgp started
     0.013974     event after_bgp:EventMgr started
     0.015848     event after_bgp:EventMgr completed
     0.015878   stage after_bgp is complete
     0.015935   stage before_ratemon started
     0.015982     event before_ratemon:EventMgr started
     0.017394     event before_ratemon:EventMgr completed
     0.017423   stage before_ratemon is complete
     0.017470   stage ratemon started
     0.017506     event ratemon:MaintenanceMode started
     5.021404     event ratemon:MaintenanceMode completed
     5.021438   stage ratemon is complete
     5.021500   stage after_ratemon started
     5.021556     event after_ratemon:EventMgr started
     5.023223     event after_ratemon:EventMgr completed
     5.023247   stage after_ratemon is complete
     5.023300   stage end started
     5.023352     event end:EventMgr started
     5.024683     event end:EventMgr completed
     5.024705   stage end is complete
     5.024762   maintEnter stages complete
```

show interface

The **show interface** command displays detailed information about the interface.

It displays an extra line that reads: “Under maintenance for time in hours and minutes”.

Command Mode

EXEC

Command Syntax

show interface *intf_name*

Parameters

- *intf_name* name of the interface
  - **ethernet** *e_range* Ethernet interfaces specified by e_range
  - **port-channel** *p_range* port channel interfaces specified by p_range
  - **vlan** *v_range* vlans specified by v_range

Note Valid *e_range*, *p_range* and *v_range* formats include number, range, or comma-delimited list of numbers and ranges. Valid Ethernet numbers depend on the Ethernet interfaces available on the switch.

Example

- This command displays detailed information about Ethernet 16/1 interface.

```
switch#show interface ethernet 16/1
Ethernet16/1 is up, line protocol is up (connected)
  Hardware is Ethernet, address is 001c.7373.efc7
  Internet address is 1.0.1.1/24
  Broadcast address is 255.255.255.255
  Address determined by manual configuration
  IP MTU 1500 bytes, BW 40000000 kbit
  Full-duplex, 40Gb/s, auto negotiation: off, uni-link: n/a
  Up 4 hours, 44 minutes, 36 seconds
  Under maintenance for 4 hours, 22 minutes, 26 seconds
  Loopback Mode : None
  2 link status changes since last clear
  Last clearing of "show interface" counters 4:45:12 ago
  5 minutes input rate 20 bps (0.0% with framing overhead), 0 packets/sec
  5 minutes output rate 20 bps (0.0% with framing overhead), 0 packets/sec
     580 packets input, 46286 bytes
     Received 1 broadcasts, 0 multicast
     0 runts, 0 giants
     0 input errors, 0 CRC, 0 alignment, 0 symbol, 0 input discards
     0 PAUSE input
     601 packets output, 48954 bytes
     Sent 7 broadcasts, 15 multicast
     0 output errors, 0 collisions
     0 late collision, 0 deferred, 0 output discards
     0 PAUSE output
switch#
```

show interface <intf_name> status

The **show interface <intf_name> status** command displays an ‘m’ flag if the interface is undergoing maintenance operation.

Command Mode

EXEC

Command Syntax

show interface [*intf_name* **] status**

Parameters

- *intf_name* name of the interface
  - **ethernet** *e_range* Ethernet interfaces specified by e_range
  - **port-channel** *p_range* port channel interfaces specified by p_range
  - **vlan** *v_range* vlans specified by v_range

Note Valid *e_range*, *p_range* and *v_range* formats include number, range, or comma-delimited list of numbers and ranges. Valid Ethernet numbers depend on the Ethernet interfaces available on the switch.

Example

- This command display tabular output and shows 'm' flag for Ethernet16/1 status.

```
switch#show interface Ethernet16/1 status
Port       Name       Status       Vlan      Duplex   Speed  Type          Flags
Et1                   disabled     1         auto     auto   1000BASE-T
...
Et14/1                connected    2         full     40G    40GBASE-CR4
Et15/1                connected    2         full     40G    40GBASE-CR4
Et16/1                connected    routed    full     40G    40GBASE-CR4   m
Et17/1                notconnect   1         full     10G    Not Present
...
switch#
```

show ip | ipv6 bgp summary [vrf <vrf_name>]

The **show ip | ipv6 bgp summary** [**vrf <vrf_name>]** command displays the 'm' flag if the BGP IPv4 or IPv6 peer is undergoing maintenance operation.

Command Mode

EXEC

Command Syntax

show ip bgp summary *[vrf <vrf_name>]*
show ipv6 bgp summary [vrf <vrf_name>]

Parameters

- *vrf_name* name of the VRF

Example

- This command displays the 'm' flag in **show ip bgp summary** output for peer 1.0.1.2 which is in maintenance mode.

```
switch#show ip bgp summary
BGP summary information for VRF default
Router identifier 0.0.1.1, local AS number 200
Neighbor Status Codes: m - Under maintenance
  Neighbor    V  AS    MsgRcvd   MsgSent  InQ  OutQ  Up/Down  State  PfxRcd
 PfxAcc
  1.0.0.1     4  100      292       296    0     0 04:47:44  Estab 1       1
m 1.0.1.2     4  300      292       296    0     0 04:47:44  Estab 1       1
switch#
```

show ip | ipv6 bgp neighbors <peer_addr> [vrf <vrf_name>]

The **show ip | ipv6 bgp neighbors <peer_addr> [vrf <vrf_name>]** command displays maintenance related information when relevant.

Command Mode

EXEC

Command Syntax

show ip bgp neighbors *<peer_addr> [vrf <vrf_name>]*
show ipv6 bgp neighbors *<peer_addr> [vrf <vrf_name>]*

Parameters

- *peer_addr* name of the peer
  - *ipv4_addr* BGP neighbor IPv4 address
  - *ipv6_addr* BGP neighbor IPv6 address
  - *peer-group-name* BGP peer group name
- *vrf_name* name of the VRF

Example

- This command displays the 'm' flag in **show ip bgp summary** output for peer 1.0.1.2 which is in maintenance mode.

```
switch#show ip bgp neighbors 1.0.1.2
BGP neighbor is 1.0.1.2, remote AS 300, external link
  BGP version 4, remote router ID 0.0.2.1, VRF default
  Negotiated BGP version 4
  Last read 00:00:09, last write 00:00:11
  Hold time is 180, keepalive interval is 60 seconds
  Configured hold time is 180, keepalive interval is 60 seconds
  Connect timer is inactive
  Idle-restart timer is inactive
  Session is under maintenance
  BGP state is Established, up for 04:55:11
  Number of transitions to established: 1
  Last state was OpenConfirm
  Last event was RecvKeepAlive
  Neighbor Capabilities:
    Multiprotocol IPv4 Unicast: advertised and received and negotiated
    Four Octet ASN: advertised and received
    Route Refresh: advertised and received and negotiated
    Send End-of-RIB messages: advertised and received and negotiated
    Additional-paths Receive:
      IPv4 Unicast: advertised and received
  Restart timer is inactive
  End of rib timer is inactive
  Message statistics:
    InQ depth is 0
    OutQ depth is 0
                         Sent      Rcvd
    Opens:                  1         1
    Notifications:          0         0
    Updates:                6         2
    Keepalives:           297       297
    Route-Refresh:          0         0
    Total messages:       304       300
  Prefix statistics:
                         Sent      Rcvd
    IPv4 Unicast:           2         1
    IPv6 Unicast:           0         0
  Inbound updates dropped by reason:
    AS path loop detection: 0
    Enforced First AS: 0
    Malformed MPBGP routes: 0
    Originator ID matches local router ID: 0
    Nexthop matches local IP address: 0
    Unexpected IPv6 nexthop for IPv4 routes: 0
    Nexthop invalid for single hop eBGP: 0
  Inbound paths dropped by reason:
    IPv4 labeled-unicast NLRIs dropped due to excessive labels: 0
  Outbound paths dropped by reason:
    IPv4 local address not available: 0
    IPv6 local address not available: 0
  Maintenance-mode:
    Inbound and Outbound policy
      Route map is SystemGenerated
Local AS is 200, local router ID 0.0.1.1
TTL is 1
```

```
Local TCP address is 1.0.1.1, local port is 179
Remote TCP address is 1.0.1.2, remote port is 51936
Auto-Local-Addr is disabled
TCP Socket Information:
  TCP state is ESTABLISHED
  Recv-Q: 0/32768
  Send-Q: 0/32768
  Outgoing Maximum Segment Size (MSS): 1448
  Total Number of TCP retransmissions: 0
  Options:
    Timestamps enabled: yes
    Selective Acknowledgments enabled: yes
    Window Scale enabled: yes
    Explicit Congestion Notification (ECN) enabled: no
  Socket Statistics:
    Window Scale (wscale): 9,7
    Retransmission Timeout (rto): 204.0ms
    Round-trip Time (rtt/rtvar): 7.5ms/3.0ms
    Delayed Ack Timeout (ato): 40.0ms
    Congestion Window (cwnd): 10
    TCP Throughput: 15.45 Mbps
    Advertised Recv Window (rcv_space): 14480
switch#
```

Chapter 10

Ethernet Ports

This chapter describes Ethernet ports supported by Arista switches. Sections covered in this chapter include:

- Section 10.1: Ethernet Ports Introduction
- Section 10.2: Ethernet Standards
- Section 10.3: Ethernet Physical Layer
- Section 10.4: Interfaces
- Section 10.5: Ethernet Configuration Procedures
- Section 10.6: Ethernet Configuration Commands

10.1 Ethernet Ports Introduction

Arista switches support a variety of Ethernet network interfaces. This chapter describes the configuration and monitoring options available in Arista switching platforms.

10.2 Ethernet Standards

Ethernet, standardized in IEEE 802.3, is a group of technologies used for communication over local area networks. Ethernet communication divides data streams into frames containing addresses (source and destination), payload, and cyclical redundancy check (CRC) information.

IEEE 802.3 also describes two types of optical fiber: single-mode fiber (SMF) and multi-mode fiber (MMF).

- SMF is more expensive, but can be used over longer distances (over 300 meters).
- MMF is less expensive, but can only be used over distances of less than 300 meters.

10.2.1 100 Gigabit Ethernet

The 100 Gigabit Ethernet (100GbE) standard defines an Ethernet implementation with a nominal data rate of 100 billion bits per second over multiple 10 gigabit lanes. 100 Gigabit Ethernet implements full duplex point to point links connected by network switches. Arista switches support 100GBASE-10SR through MXP ports.

10.2.2 40 Gigabit Ethernet

The 40 Gigabit Ethernet (40GbE) standard defines an Ethernet implementation with a nominal data rate of 40 billion bits per second over multiple 10 gigabit lanes. 40 Gigabit Ethernet implements full duplex point to point links connected by network switches. 40 gigabit Ethernet standards are named ***40GBASE-xyz***, as interpreted by Table 10-1.

Table 10-1 40GBASE-xyz Interpretation

x	y	z
Non-fiber media type, or fiber wavelength	PHY encoding	Number of WWDM wavelengths or XAUI Lanes
C = Copper F = Serial SMF K = Backplane L = Long (1310 nm) S = Short (850 nm)	R = LAN PHY (64B/66B)	No value = 1 (serial) 4 = 4 WWDM wavelengths or XAUI Lanes

10.2.3 10 Gigabit Ethernet

The 10 Gigabit Ethernet (10GbE) standard defines an Ethernet implementation with a nominal data rate of 10 billion bits per second. 10 Gigabit Ethernet implements full duplex point to point links connected by network switches. Half duplex operation, hubs and CSMA/CD do not exist in 10GbE. The standard encompasses several PHY standards; a networking device may support different PHY types through pluggable PHY modules. 10GbE standards are named ***10GBASE-xyz***, as interpreted by Table 10-2.

Table 10-2 10GBASE-xyz Interpretation

x	y	z
media type or wavelength, if media type is fiber	PHY encoding type	Number of WWDM wavelengths or XAUI Lanes
C = Copper (twin axial) T = Twisted Pair S = Short (850 nm) L = Long (1310 nm) E = Extended (1550 nm) Z = Ultra extended (1550 nm)	R = LAN PHY (64B/66B) X = LAN PHY (8B/10B) W = WAN PHY(*) (64B/66B)	If omitted, value = 1 (serial) 4 = 4 WWDM wavelengths or XAUI Lanes

10.2.4 Gigabit Ethernet

The Gigabit Ethernet (GbE), defined by IEEE 802.3-2008, describes an Ethernet version with a nominal data rate of one billion bits per second. GbE cables and equipment are similar to those used in previous standards. While full-duplex links in switches is the typical implementation, the specification permits half-duplex links connected through hubs.

Gigabit Ethernet physical layer standards that Arista switches support include 1000BASE-X (optical fiber), 1000BASE-T (twisted pair cable), and 1000BASE-CX (balanced copper cable).

- 1000BASE-SX is a fiber optic standard that utilizes multi-mode fiber supporting 770 to 860 nm, near infrared (NIR) light wavelength to transmit data over distances ranging from 220 to 550 meters. 1000BASE-SX is typically used for intra-building links in large office buildings, co-location facilities and carrier neutral Internet exchanges.
- 1000BASE-LX is a fiber standard that utilizes a long wavelength laser (1,270–1,355 nm), with a RMS spectral width of 4 nm to transmit data up to 5 km. 1000BASE-LX can run on all common types of multi-mode fiber with a maximum segment length of 550 m.

- 1000BASE-T is a standard for gigabit Ethernet over copper wiring. Each 1000BASE-T network segment can be a maximum length of 100 meters.

10.2.5 10/100/1000 BASE-T

Arista switches provide 10/100/1000 BASE-T Mbps Ethernet out of band management ports. Auto-negotiation is enabled on these interfaces. Speed (10/100/1000), duplex (half/full), and flow control settings are available using the appropriate **speed forced** and **flowcontrol** commands.

10.3 Ethernet Physical Layer

The Ethernet physical layer (PHY) includes hardware components connecting a switch's MAC layer to the transceiver, cable, and ultimately a peer link partner. Data exist in digital form at the MAC layer. On the line side of the PHY, data exist as analog signals: light blips on optical fiber or voltage pulses on copper cable. Signals may be distorted while in transit and recovery may require signal processing.

Ethernet physical layer components include a PHY and a transceiver.

10.3.1 PHYs

The PHY provides translation services between the MAC layer and transceiver. It also helps to establish links between the local MAC layer and peer devices by detecting and signaling fault conditions. The PHY line-side interface receives Ethernet frames from the link partner as analog waveforms. The PHY uses signal processing to recover the encoded bits, then sends them to the MAC layer.

PHY line-side interface components and their functions include:

- Physical Medium Attachment (PMA): Framing, octet synchronization, scrambling / descrambling.
- Physical Medium Dependent (PMD): Consists of the transceiver.
- Physical Coding Sublayer (PCS): Performs auto-negotiation and coding (8B/10B or 64B/66B).

The MAC sublayer of the PHY provides a logical connection between the MAC layer and the peer device by initializing, controlling, and managing the connection with the peer.

Ethernet frames transmitted by the switch are received by the PHY system-side interface as a sequence of digital bits. The PHY encodes them into a media-specific waveform for transmission through the line-side interface and transceiver to the link peer. This encoding may include signal processing, such as signal pre-distortion and forward error correction.

PHY system-side interface components and their functions include:

- 10 Gigabit Attachment Unit Interface (XAUI): Connects an Ethernet MAC to a 10 G PHY.
- Serial Gigabit Media Independent Attachment (SGMII): Connects an Ethernet MAC to a 1G PHY.

10.3.2 Transceivers

A transceiver connects the PHY to an external cable (optical fiber or twisted-pair copper) and through a physical connector (LC jack for fiber or RJ-45 jack for copper).

- Optical transceivers convert the PHY signal into light pulses that are sent through optical fiber.
- Copper transceivers connect the PHY to twisted-pair copper cabling.

Arista Small Form-Factor Pluggable (SFP+) and Quad Small Form Factor Pluggable (QSFP+) modules and cables provide high-density, low-power Ethernet connectivity over fiber and copper media. Arista offers transceivers that span data rates, media types, and transmission distances.

Arista 10 Gigabit Ethernet SFP+ Modules:

- 10GBASE-SR (Short Reach)
  - Link length maximum 300 meters over multi-mode fiber.
  - Optical interoperability with 10GBASE-SRL.
- 10GBASE-SRL (Short Reach Lite)
  - Link length maximum 100 meters over multi-mode fiber.
  - Optical interoperability with 10GBASE-SR.

- 10GBASE-LRL (Long Reach Lite)
  - Link length maximum 1 km over single-mode fiber.
  - Optical interoperability with 10GBASE-LR (1 km maximum).
- 10GBASE-LR (Long Reach)
  - Link length maximum 10 km over single-mode fiber.
  - Optical interoperability with 10GBASE-LRL (1 km maximum).
- 10GBASE-LRM (Long Reach Multimode)
  - Link length maximum 220 meters over multi-mode fiber (50 um and 62.5 um).
- 10GBASE-ER (Extended Reach)
  - Link length maximum 40 km over single-mode fiber.
- 10GBASE-ZR (Ultra-Extended Reach)
  - Link length maximum 80 km over single-mode fiber.
- 10GBASE-DWDM (Dense Wavelength Division Multiplexing)
  - Link length maximum 80 km over single-mode fiber (40 color options).

Arista 10 Gigabit Ethernet CR Cable Modules:

- 10GBASE-CR SFP+ to SFP+ Cables
  - Link lengths of 0.5, 1, 1.5, 2, 2.5, 3, 5 and 7 meters over twinax copper cable
  - Includes SFP+ connectors on both ends
- 4 x 10GbE QSFP+ to 4 x SFP+ twinax copper cables
  - Link lengths of 0.5, 1, 2 and 3 meters over twinax copper cable

Arista 40 Gigabit Ethernet QSFP+ Cables and Optics:

- 40GBASE-SR4 QSFP+ Transceiver
  - Link length maximum 100 meters over parallel OM3 or 150 meters over OM4 MMF
  - Optical interoperability with 40GBASE-XSR4 (100/150 meter maximum)
- 40GBASE-XSR4 QSFP+ Transceiver
  - Link length maximum 300 meters over parallel OM3 or 450 meters over OM4 MMF
  - Optical interoperability with 40GBASE-SR4 (100/150 meter maximum)
- 40GBASE-LR4 QSFP+
  - Link length maximum 10 km over duplex single-mode fiber
- 40GBASE-CR4 QSFP+ to QSFP+ twinax copper cables
  - Link lengths of 1, 2, 3, 5 and 7 meters over twinax copper cable

Arista Gigabit Ethernet SFP Options:

- 1000BASE-SX (Short Haul)
  - Multi-mode fiber
  - Link length maximum 550 meter
- 1000BASE-LX (Long Haul)
  - Single-mode or multi-mode fiber
  - Link length maximum 10 km (single mode) or 550 meters (multi-mode)
- 1000BASE-T (RJ-45 Copper)

- Category 5 cabling
- Full duplex 1000Mbps connectivity

Internal ports

Several Arista switches include internal ports that connect directly to an external cable through an RJ-45 jack. Internal ports available on Arista switches include:

- 100/1000BASE-T (7048T-A)
- 100/1000/10GBASE-T (7050-T)

10.3.3 MXP Ports

MXP ports provide embedded optics that operate in one of three modes: 10GbE (12 ports), 40GbE (3 ports), and 100GbE (1 port). Each mode requires a specified cable is implemented through configuration commands. MXP ports utilize multi-mode fiber to provide support over 150 meters.

- 100GbE mode requires an MTP-24 to MTP-24 cable, which uses 20 of 24 fibers to carry 100Gbe across 10 send and 10 receive channels. When connecting two 100GbE MXP ports, the TX lanes must be crossed with the RX lanes.
- 40GbE mode requires an MTP cable that provides a split into three MTP-12 ends. The cable splits the MXP port into three MTP-12 ends, each compatible with standards based 40GBASE-SR4 ports over OM3 or OM4 fiber up to 100m or 150m.
- 10GbE mode requires an MTP cable that provides a split into 12x10G with LC connectors to adapt the MXP port into 12x10GbE. The cable splits the MXP port into twelve LC ends for using SR or SRL optics over multimode OM3/OM4 cables.

10.4 Interfaces

Arista switches provide two physical interface types that receive, process, and transmit Ethernet frames: Ethernet interfaces and Management interfaces.

Each Ethernet interface is assigned a 48-bit MAC address and communicates with other interfaces by exchanging data packets. Each packet contains the MAC address of its source and destination interface. Ethernet interfaces establish link level connections by exchanging packets. Interfaces do not typically accept packets with a destination address of a different interface.

Ethernet data packets are frames. A frame begins with preamble and start fields, followed by an Ethernet header that includes source and destination MAC addresses. The middle section contains payload data, including headers for other protocols carried in the frame. The frame ends with a 32-bit cyclic redundancy check (CRC) field that interfaces use to detect data corrupted during transmission.

10.4.1 Ethernet Interfaces

Ethernet speed and duplex configuration options depend on the media type of the interface:

- 40G QSFP+: Default operation is as four 10G ports. **Speed forced** command options support configuration as a single 40G port.
- 10GBASE-T: Mode is ***autonegotiate*** by default, offering 10G and 1G full duplex and 100M. Default setting is 10G. Half duplex and 10M are not supported. Adjustments may be made using **speed forced** commands.
- 10GBASE (SFP+): Port operates as a single 10G port. **Speed forced** commands do not affect configuration.
- 1000BASE-T (copper): Mode is ***autonegotiate*** by default, offering 1G full and 100M; default setting is 1G full. Autonegotiation that offers only 100M is available through **speed auto 100full** command. Half duplex and 10M are not supported.
- 100G CFP: Default operation is 100G. It cannot be split, and its speed cannot be changed.
- 100G MXP: Default operation is as a single 100G port on the 7500 and 7280 platforms, and as three 40G ports on the 7050 platform. On the 7500 and 7280 platforms, available speed/duplex settings are a single100G port, three 40G ports, or twelve 10G ports. On the 7050 platform, available speed/duplex settings are three 40G ports or twelve 10G ports. Adjustments are made with **speed forced** commands.
- 100G QSFP100: Available speeds are transceiver-dependent. The QSFP100 transceiver supports a single 100G port, four 25G ports, or two 50G ports; the QSFP+ transceiver supports one 40G port or four 10G ports; the CWDM transceiver supports all five configurations. Adjustments are made using **speed forced** commands. **Note**: 7500 and 7280 families do not currently support 25G or 50G speeds.

For information relating to transceivers, please see Transceivers.

10.4.2 Subinterfaces

Subinterfaces divide a single ethernet or port channel interface into multiple logical L3 interfaces based on the 802.1q tag (VLAN ID) of incoming traffic. Subinterfaces are commonly used in the L2/L3 boundary device, but they can also be used to isolate traffic with 802.1q tags between L3 peers by assigning each subinterface to a different VRF.

While subinterfaces can be configured on a port channel interface (the virtual interface associated with a port channel), the following restrictions apply:

- An L3 interface with subinterfaces configured on it should not be made a member of a port channel.
- An interface that is a member of a port channel should not have subinterfaces configured on it.

- A subinterface cannot be made a member of a port channel.

Subinterfaces on multiple ports can be assigned the same VLAN ID, but there is no bridging between subinterfaces (or between subinterfaces and SVIs), and each subinterface is considered to be in a separate bridge domain.

The following features are supported on subinterfaces:

- Unicast and multicast routing
- BGP, OSPF, ISIS, PIM
- VRF
- VRRP
- SNMP
- Subinterface counters (on some platforms)
- VXLAN (on some platforms)
- MPLS (on some platforms)
- GRE (on some platforms)
- PBR (on some platforms)
- QoS (on some platforms)
- Inheriting QoS settings (trust mode and default DSCP) from the parent interface
- Inheriting MTU setting from parent interface

The following are *not* supported on subinterfaces:

- BFD
- ACL
- Per-subinterface MTU setting
- Per-subinterface SFLOW settings
- Per-subinterface mirroring settings

10.4.3 Agile Ports

Agile Ports are a feature of the 7150S Series that allows the user to configure adjacent blocks of 4 x SFP+ interfaces as a single 40G link. The set of interfaces that can be combined to form a higher speed port is restricted by the hardware configuration. Only interfaces that pass through a common PHY component can be combined. One interface within a combinable set is designated as the primary port. When the primary interface is configured as a higher speed port, all configuration statements are performed on that interface. All other interfaces in the set are subsumed and not individually configurable when the primary interface is configured as the higher speed port. This feature allows the 7150S-24 to behave as a 4x40G switch (using 16 SFP+) and the remaining SFP+ provide 8 x 10G ports. On the 7150S-52 this allows up to 13x 40G (all 52 ports grouped as 40G) and on the 7150S-64 Agile Ports allows the switch to be deployed with up to 16 native 40G interfaces - 4 are QSFP+ and the remaining 12 as 4xSFP+ groups.

Section 10.5.11 describes the configuration of agile ports.

10.4.4 Management Interfaces

The management interface is a layer 3 host port that is typically connected to a PC for performing out of band switch management tasks. Each switch has one or two management interfaces. Only one port is needed to manage the switch; the second port, when available, provides redundancy.

Management interfaces are 10/100/1000 BASE-T interfaces. By default, auto-negotiation is enabled on management interfaces. All combinations of speed 10/100/1000 and full or half duplex is enforceable on these interfaces through **speed** commands.

Management ports are enabled by default. The switch cannot route packets between management ports and network (Ethernet interface) ports because they are in separate routing domains. When the PC is multiple hops from the management port, packet exchanges through layer 3 devices between the management port and PC may require the enabling of routing protocols.

The Ethernet management ports are accessed remotely over a common network or locally through a directly connected PC. An IP address and static route to the default gateway must be configured to access the switch through a remote connection.

10.4.5 Tunable SFP

Tuning of DWDM 10G SFP+ transceivers (10GBASE-DWDM) includes:

- Tuning transceiver wavelength/frequency by channel number
- Showing wavelengths/frequencies for specified channels supported by the transceiver
- Showing current wavelength/frequency settings of the transceiver interface

For information relating to tuning the transceiver wavelength/frequency by channel number, refer to the command transceiver channel. To show the current wavelength/frequency settings for specified channels, refer to the command show interfaces transceiver channels. To show the current wavelength/frequency settings of an interface, refer to the command show interfaces transceiver hardware.

10.5 Ethernet Configuration Procedures

These sections describe Ethernet and Management interface configuration procedures:

- Section 10.5.1: Physical Interface Configuration Modes
- Section 10.5.2: Assigning a MAC Address to an Interface
- Section 10.5.3: Port Groups (QSFP+ and SFP+ Interface Selection)
- Section 10.5.4: Referencing Modular Ports
- Section 10.5.5: Referencing Multi-lane Ports
- Section 10.5.6: QSFP+ Ethernet Port Configuration
- Section 10.5.7: QSFP100 Ethernet Port Configuration
- Section 10.5.8: CFP2 Ethernet Port Configuration
- Section 10.5.9: MXP Ethernet Port Configuration
- Section 10.5.10: Port Speed Capabilities
- Section 10.5.11: Agile Ports
- Section 10.5.12: Subinterface Configuration
- Section 10.5.13: Autonegotiated Settings
- Section 10.5.14: Displaying Ethernet Port Properties
- Section 10.5.15: Configuring Ingress Traffic-Class Counters

10.5.1 Physical Interface Configuration Modes

The switch provides two configuration modes for modifying Ethernet parameters:

- Interface-Ethernet mode configures parameters for specified Ethernet interfaces.
- Interface-Management mode configures parameters for specified management Ethernet interfaces.

Physical interfaces cannot be created or removed.

Multiple interfaces can be simultaneously configured. Commands are available for configuring Ethernet specific, layer 2, layer 3, and application layer parameters. Commands that modify protocol specific settings in Ethernet configuration mode are listed in the protocol chapters.

- The **interface ethernet** command places the switch in Ethernet-interface configuration mode.
- The **interface management** command places the switch in management configuration mode.

Examples

- This command places the switch in Ethernet-interface mode for Ethernet interfaces 5-7 and 10.

```
switch(config)#interface ethernet 5-7,10
switch(config-if-Et5-7,10)#
```

- This command places the switch in management-interface mode for management interface 1.

```
switch(config)#interface management 1
switch(config-if-Ma1)#
```

10.5.2 Assigning a MAC Address to an Interface

Ethernet and Management interfaces are assigned a MAC address when manufactured. This address is the ***burn-in address***. The **mac-address** command assigns a MAC address to the configuration mode interface in place of the burn-in address. The **no mac-address** command reverts the interface's current MAC address to its burn-in address.

Examples

- This command assigns the MAC address of ***001c.2804.17e1*** to Ethernet interface 7.

```
switch(config-if-Et7)#mac-address 001c.2804.17e1
```

- This command displays the MAC address of Ethernet interface 7. The active MAC address is ***001c.2804.17e1***. The burn-in address is ***001c.7312.02e2***.

```
switch(config-if-Et7)#show interface ethernet 7
Ethernet7 is up, line protocol is up (connected)
  Hardware is Ethernet, address is 001c.2804.17e1 (bia 001c.7312.02e2)
  Description: b.e45
             <-------OUTPUT OMITTED FROM EXAMPLE-------->

switch(config-if-Et7)#
```

10.5.3 Port Groups (QSFP+ and SFP+ Interface Selection)

Several of Arista's fixed switches limit the number of 10G data lanes in operation through the use of port groups. A port group is a set of interfaces that can be configured as four SFP+ interfaces or a single QSFP+ interface. When configured in SFP+ mode, the port group enables 4 standalone 10GbE interfaces using SFP+ optics. When configured in QSFP+ mode, the port group enables a single QSFP+ interface (in addition to the dedicated QSFP+ ports), which can operate as a single 40GbE port, or as four 10GbE ports with the appropriate breakout cabling.

Hardware port groups are used on the following systems:

- DCS-7050Q-16
- DCS-7050QX-32S

Use the **hardware port-group** command to select the interface mode for the specified port group.

Important! The **hardware port-group** command restarts the forwarding agent, which disrupts traffic on all switch ports.

Example

- These commands configure the DCS-7050-Q16 switch to enable four SFP+ interfaces and one extra QSFP+ interface by enabling the SFP+ interfaces in port group 1 and the QSFP+ interface in port group 2.

```
switch(config)#hardware port-group 1 select Et17-20
switch(config)#hardware port-group 2 select Et16/1-4
```

The **show hardware port-group** command displays the status of ports in the port groups.

Example

- This command displays the status of the flexible ports within the two port groups on a DCS-7050Q-16 switch.

```
switch#show hardware port-group

Portgroup: 1    Active Ports: Et17-20
Port            State
----------------------------------------
Ethernet17      Active
Ethernet18      Active
Ethernet19      Active
Ethernet20      Active
Ethernet15/1    ErrDisabled
Ethernet15/2    ErrDisabled
Ethernet15/3    ErrDisabled
Ethernet15/4    ErrDisabled

Portgroup: 2    Active Ports: Et16/1-4
Port            State
----------------------------------------
Ethernet16/1    Active
Ethernet16/2    Active
Ethernet16/3    Active
Ethernet16/4    Active
Ethernet21      ErrDisabled
Ethernet22      ErrDisabled
Ethernet23      ErrDisabled
Ethernet24      ErrDisabled
```

10.5.3.1 DCS-7050Q-16

The DCS-7050Q-16 has 14 dedicated QSFP+ ports, plus two port groups. The port groups support either two additional QSFP+ ports or eight SFP+ ports as shown in Table 10-3.

Table 10-3 DCS-7050Q-16 Port Groups

Port Group 1 Active Interface(s)		Port Group 2 Active Interface(s)	
In SFP+ Mode	In QSFP+ Mode (Default)	In SFP+ Mode	In QSFP+ Mode (Default)
Et17-20 (four SFP+ ports)	Et15/1-4 (one QSFP+ port)	Et21-24 (four SFP+ ports)	Et16/1-4 (one QSFP+ port)

10.5.3.2 DCS-7050QX-32S

The DCS-7050QX-32S has 31 dedicated QSFP+ ports, plus one port group. The port group supports either one additional QSFP+ port or four SFP+ ports as shown in Table 10-4.

Table 10-4 DCS-7050QX-32S Port Groups

Port Group 1 Active Interface(s)	
In SFP+ Mode	In QSFP+ Mode (Default)
Et1-4	Et5/1-4
(four SFP+ ports)	(one QSFP+ port)

10.5.4 Referencing Modular Ports

Arista modular switches provide port access through installed line cards. The maximum number of line cards on a modular switch varies with the switch series and model.

Several CLI commands modify modular parameters for all ports on a specified line card or controlled by a specified chip. This manual uses these conventions to reference modular components:

- *card_x* refers to a line card.
- *module_y* refers to a QSFP+ module.
- *port_z* refers to a line card or module port.

Commands that display Ethernet port status use the following conventions:

- SFP ports: : *card_x/port_z* to label the line card-port location of modular ports
- QSFP ports: *card_x/module_y/port_z* to label the line card-port location of modular ports

Section 10.5.6 describe QSFP+ module usage.

Example

- This command displays the status of interfaces 1 to 9 on line card 4:

```
switch>show interface ethernet 4/1-9 status
Port      Name              Status       Vlan        Duplex  Speed Type
Et4/1                       connected    1             full    10G Not Present
Et4/2                       connected    1             full    10G Not Present
Et4/3                       connected    1             full    10G Not Present
Et4/4                       connected    1             full    10G Not Present
Et4/5                       connected    1             full    10G Not Present
Et4/6                       connected    1             full    10G Not Present
Et4/7                       connected    1             full    10G Not Present
Et4/8                       connected    1             full    10G Not Present
Et4/9                       connected    1             full    10G Not Present
switch>
```

10.5.5 Referencing Multi-lane Ports

EOS supports two types of Ethernet ports:

- single-lane (also called fixed-lane)
- multi-lane (also called flexible-lane)

Single-lane (or “fixed-lane”) ports are always modeled as a single interface within EOS. While the speed of the interface may be configurable, the physical port can never be broken out into multiple lower-speed interfaces. Single-lane ports use the following naming scheme:

- Ethernet <port #> (for fixed switches)
- Ethernet <module #>/<port #> (for modular switches)

Multi-lane (or “flexible lane”) ports are made up of multiple parallel lanes, each served by its own laser. Multi-lane ports can be configured to combine the lanes and operate as a single native high-speed interface (a 40GbE or 100GbE interface), or to operate each lower-speed interface independently (four 10GbE or 25GbE interfaces). Multi-lane ports use the following naming scheme:

- Ethernet <port #>/<lane #> (for fixed switches)
- Ethernet <module #>/<port #>/<lane #> (for modular switches)

The operational state displayed for each lane of a multi-lane port is determined by the configuration applied to the primary lane(s), as shown in Table 10-5. When broken out into multiple lower-speed interfaces, all lanes will be active in parallel, and each will display its operational state as ***connected*** or ***not connected***. In high-speed mode, only the primary lane(s) will be displayed as active, with the remaining lanes showing as ***errdisabled***. The exception is the CFP2 module: when it is configured as a single 100GbE port, the primary lane is displayed as active in the CLI while the other lanes are hidden.

Table 10-5 Lane States

Parent Port Configured Mode	Primary Lane(s)	Secondary Lanes

A multi-lane port is configured as a single high-speed interface or multiple breakout interfaces by using the **speed** command on the primary lane(s) of the port. For specific configuration instructions and details regarding the primary lane(s) of a specific interface, refer to the configuration section for the appropriate interface type:

- QSFP+ Ethernet Port Configuration
- QSFP100 Ethernet Port Configuration
- CFP2 Ethernet Port Configuration
- MXP Ethernet Port Configuration

Important! Use of the **speed** command to configure a multi-lane port is hitless on the 7050X, 7060X, 7250X, 7260X, 7280SE, 7300X, 7320X and 7500E series platforms. On all other platforms, this command restarts the forwarding agent, which will result in traffic disruption.

10.5.6 QSFP+ Ethernet Port Configuration

Each QSFP+ module contains four data lanes which can be used individually or combined to form a single, higher-speed interface. This allows a QSFP+ Ethernet port to be configured as a single 40GbE interface or as four 10GbE interfaces.

When the four lanes are combined to form a 40GbE interface, display commands will show lane /1 as ***connected*** or ***not connected***, and will show lanes /2 through /4 as ***errdisabled***.

The following sections describe the configuration of QSFP+ ports.

10.5.6.1 Configuring a QSFP+ Module as a Single 40GbE Interface

To configure the port as a single 40GbE interface, combine the module's four data lanes by using the **speed** command (**speed forced 40g full**) on the port's /1 lane (the primary lane).

Important! The **speed** command is hitless on the 7050X, 7060X, 7250X, 7260X, 7280SE, 7300X, 7320X and 7500E series platforms. On all other platforms, this command restarts the forwarding agent, which will result in traffic disruption.

Step 1 Enter interface Ethernet configuration mode for lane /1 of the QSFP+ Ethernet interface.

```
switch(config)#interface ethernet 5/1/1
```

Step 2 Enter the **speed forced 40gfull** command. Depending on the platform, this command may restart the forwarding agent, disrupting traffic on all ports for 60 seconds or more.

```
switch(config-if-Et5/1/1)#speed forced 40gfull
```

Step 3 Use the **show interfaces status** command to confirm the change in configuration.

```
switch(config-if-Et5/1/1)#show interfaces status
Port       Name           Status       Vlan     Duplex Speed  Type         Flags
Et1                       connected    2        full   1G     10GBASE-T
          <-------OUTPUT OMITTED FROM EXAMPLE-------->
Et5/1/1                   connected    1        full   40G    40GBASE-SR4
Et5/1/2                   errdisabled  1        unconf unconf 40GBASE-SR4
Et5/1/3                   errdisabled  1        unconf unconf 40GBASE-SR4
Et5/1/4                   errdisabled  1        unconf unconf 40GBASE-SR4
          <-------OUTPUT OMITTED FROM EXAMPLE-------->
```

10.5.6.2 Configuring a QSFP+ Module as Four 10GbE Interfaces

To configure the port as four 10GbE interfaces, use the **speed** command (**speed forced 10000full**) on the port's /1 lane (the primary lane).

Important! The **speed** command is hitless on the 7050X, 7060X, 7250X, 7260X, 7280SE, 7300X, 7320X and 7500E series platforms. On all other platforms, this command restarts the forwarding agent, which will result in traffic disruption.

Step 1 Enter interface Ethernet configuration mode for lane /1 of the QSFP+ Ethernet interface.

```
switch(config)#interface ethernet 5/1/1
```

Step 2 Enter the **speed forced 10000full** command. Depending on the platform, this command may restart the forwarding agent, disrupting traffic on all ports for 60 seconds or more.

```
switch(config-if-Et5/1/1)#speed forced 10000full
```

Step 3 Use the **show interfaces status** command to confirm the change in configuration.

```
switch(config-if-Et5/1/1)#show interfaces status
Port       Name           Status       Vlan     Duplex Speed  Type         Flags
Et1                       connected    2        full   1G     10GBASE-T
          <-------OUTPUT OMITTED FROM EXAMPLE-------->
Et5/1/1                   connected    1        full   10G    40GBASE-SR4
Et5/1/2                   connected    1        full   10G    40GBASE-SR4
Et5/1/3                   connected    1        full   10G    40GBASE-SR4
Et5/1/4                   connected    1        full   10G    40GBASE-SR4
          <-------OUTPUT OMITTED FROM EXAMPLE-------->
```

10.5.7 QSFP100 Ethernet Port Configuration

Each QSFP100 module contains four data lanes which can be used individually or combined to form a single, higher-speed interface. This allows a QSFP100 Ethernet port to be configured as a single 100GbE interface, a single 40GbE interface, or four 10GbE interfaces. The default mode is a single 100GbE interface.

The 7060X, 7260X and 7320X platforms also allow a QSFP100 port to be configured as two 50GbE interfaces or four 25GbE interfaces.

When the lanes are combined to form a higher-speed interface, display commands will show the primary lane(s) as ***connected*** or ***not connected***, and will show the other lanes as ***errdisabled***.

The following sections describe the configuration of QSFP+ ports.

10.5.7.1 Configuring a QSFP100 Module as a Single 100GbE Interface

By default, the QSFP100 module operates as a single 100GbE interface; using the **default speed** or **no speed** command on the primary lane restores the default behavior.

To explicitly configure the port as a single 100GbE interface, combine the module's four data lanes by using the **speed** command (**speed forced 100gfull**) on the port's /1 lane (the primary lane).

Important! The **speed** command is hitless on the 7050X, 7060X, 7250X, 7260X, 7280SE, 7300X, 7320X and 7500E series platforms. On all other platforms, this command restarts the forwarding agent, which will result in traffic disruption.

Step 1 Enter interface Ethernet configuration mode for lane /1 of the QSFP100 Ethernet interface.

```
switch(config)#interface ethernet 5/1/1
```

Step 2 Enter the **speed forced 100gfull** command. Depending on the platform, this command may restart the forwarding agent, disrupting traffic on all ports for 60 seconds or more.

```
switch(config-if-Et5/1/1)#speed forced 100gfull
```

Step 3 Use the **show interfaces status** command to confirm the change in configuration.

```
switch(config-if-Et5/1/1)#show interfaces status
Port       Name           Status       Vlan     Duplex Speed  Type         Flags
Et1                       connected    2        full   1G     10GBASE-T
          <-------OUTPUT OMITTED FROM EXAMPLE-------->
Et5/1/1                   connected    1        full   100G   100GBASE-SR4
Et5/1/2                   errdisabled  1        unconf unconf 100GBASE-SR4
Et5/1/3                   errdisabled  1        unconf unconf 100GBASE-SR4
Et5/1/4                   errdisabled  1        unconf unconf 100GBASE-SR4
          <-------OUTPUT OMITTED FROM EXAMPLE-------->
```

10.5.7.2 Configuring a QSFP100 Module as Two 50GbE Interfaces

To configure the port as a two 50GbE interfaces, configure the module's four data lanes by using the **speed** command (**speed forced 50gfull**) on the port's /1 and /3 lanes. This configuration is available on 7060X, 7260X and 7320X platforms.

Important! The **speed** command is hitless on the 7050X, 7060X, 7250X, 7260X, 7280SE, 7300X, 7320X and 7500E series platforms. On all other platforms, this command restarts the forwarding agent, which will result in traffic disruption.

Step 1 Enter interface Ethernet configuration mode for lane /1 of the QSFP100 Ethernet interface.

```
switch(config)#interface ethernet 5/1/1
```

Step 2 Enter the **speed forced 50gfull** command. Depending on the platform, this command may restart the forwarding agent, disrupting traffic on all ports for 60 seconds or more.

```
switch(config-if-Et5/1/1)#speed forced 50gfull
```

Step 3 Repeat the above steps for lane /3.

```
switch(config-if-Et5/1/1)#interface ethernet 5/1/3
switch(config-if-Et5/1/3)#speed forced 50gfull
```

Step 4 Use the **show interfaces status** command to confirm the change in configuration.

```
switch(config-if-Et5/1/1)#show interfaces status
Port       Name           Status       Vlan     Duplex Speed  Type         Flags
Et1                       connected    2        full   1G     10GBASE-T
          <-------OUTPUT OMITTED FROM EXAMPLE-------->
Et5/1/1                   connected    1        full   50G    100GBASE-SR4
Et5/1/2                   errdisabled  1        unconf unconf 100GBASE-SR4
Et5/1/3                   connected    1        full   50G    100GBASE-SR4
Et5/1/4                   errdisabled  1        unconf unconf 100GBASE-SR4
          <-------OUTPUT OMITTED FROM EXAMPLE-------->
```

10.5.7.3 Configuring a QSFP100 Module as a Single 40GbE Interface

To configure the port as a single 40GbE interface, combine the module's four data lanes by using the **speed** command (**speed forced 40gfull**) on the port's /1 lane (the primary lane).

Important! The **speed** command is hitless on the 7050X, 7060X, 7250X, 7260X, 7280SE, 7300X, 7320X and 7500E series platforms. On all other platforms, this command restarts the forwarding agent, which will result in traffic disruption.

Step 1 Enter interface Ethernet configuration mode for lane /1 of the QSFP100 Ethernet interface.

```
switch(config)#interface ethernet 5/1/1
```

Step 2 Enter the **speed forced 40gfull** command. Depending on the platform, this command may restart the forwarding agent, disrupting traffic on all ports for 60 seconds or more.

```
switch(config-if-Et5/1/1)#speed forced 40gfull
```

Step 3 Use the **show interfaces status** command to confirm the change in configuration.

```
switch(config-if-Et5/1/1)#show interfaces status
Port       Name           Status       Vlan     Duplex Speed  Type         Flags
Et1                       connected    2        full   1G     10GBASE-T
          <-------OUTPUT OMITTED FROM EXAMPLE-------->
Et5/1/1                   connected    1        full   40G    100GBASE-SR4
Et5/1/2                   errdisabled  1        unconf unconf 100GBASE-SR4
Et5/1/3                   errdisabled  1        unconf unconf 100GBASE-SR4
Et5/1/4                   errdisabled  1        unconf unconf 100GBASE-SR4
          <-------OUTPUT OMITTED FROM EXAMPLE-------->
```

10.5.7.4 Configuring a QSFP100 Module as Four 25GbE Interfaces

To configure the port as four 25GbE interfaces, use the **speed** command (**speed forced 25gfull**) on the port's /1 lane (the primary lane). This configuration is available on 7060X, 7260X and 7320X platforms.

Important! The **speed** command is hitless on the 7050X, 7060X, 7250X, 7260X, 7280SE, 7300X, 7320X and 7500E series platforms. On all other platforms, this command restarts the forwarding agent, which will result in traffic disruption.

Step 1 Enter interface Ethernet configuration mode for lane /1 of the QSFP100 Ethernet interface.

```
switch(config)#interface ethernet 5/1/1
```

Step 2 Enter the **speed forced 25gfull** command. Depending on the platform, this command may restart the forwarding agent, disrupting traffic on all ports for 60 seconds or more.

```
switch(config-if-Et5/1/1)#speed forced 25gfull
```

Step 3 Use the **show interfaces status** command to confirm the change in configuration.

```
switch(config-if-Et5/1/1)#show interfaces status
Port       Name           Status       Vlan     Duplex Speed  Type         Flags
Et1                       connected    2        full   1G     10GBASE-T
          <-------OUTPUT OMITTED FROM EXAMPLE-------->
Et5/1/1                   connected    1        full   25G    100GBASE-SR4
Et5/1/2                   errdisabled  1        unconf unconf 100GBASE-SR4
Et5/1/3                   errdisabled  1        unconf unconf 100GBASE-SR4
Et5/1/4                   errdisabled  1        unconf unconf 100GBASE-SR4
          <-------OUTPUT OMITTED FROM EXAMPLE-------->
```

10.5.7.5 Configuring a QSFP100 Module as Four 10GbE Interfaces

To configure the port as four 10GbE interfaces, use the **speed** command (**speed forced 10000full**) on the port's /1 lane (the primary lane).

Important! The **speed** command is hitless on the 7050X, 7060X, 7250X, 7260X, 7280SE, 7300X, 7320X and 7500E series platforms. On all other platforms, this command restarts the forwarding agent, which will result in traffic disruption.

Step 1 Enter interface Ethernet configuration mode for lane /1 of the QSFP100 Ethernet interface.

```
switch(config)#interface ethernet 5/1/1
```

Step 2 Enter the **speed forced 10000full** command. Depending on the platform, this command may restart the forwarding agent, disrupting traffic on all ports for 60 seconds or more.

```
switch(config-if-Et5/1/1)#speed forced 10000full
```

Step 3 Use the **show interfaces status** command to confirm the change in configuration.

```
switch(config-if-Et5/1/1)#show interfaces status
Port       Name           Status       Vlan     Duplex Speed  Type         Flags
Et1                       connected    2        full   1G     10GBASE-T
          <-------OUTPUT OMITTED FROM EXAMPLE-------->
Et5/1/1                   connected    1        full   10G    100GBASE-SR4
Et5/1/2                   connected    1        full   10G    100GBASE-SR4
Et5/1/3                   connected    1        full   10G    100GBASE-SR4
Et5/1/4                   connected    1        full   10G    100GBASE-SR4
          <-------OUTPUT OMITTED FROM EXAMPLE-------->
```

10.5.8 CFP2 Ethernet Port Configuration

Each CFP2 module contains ten data lanes. The configuration options available on the port depend on the optic inserted:

- **CFP2-100G-LR4** optics operate only in 100GbE mode.
- **CF2-100G-ER4** optics operate only 100GbE mode.
- **CFP2-100G-XSR10** optics can be configured as a single 100GbE interface or as ten 10GbE interfaces.

When the port is configured as ten 10GbE interface, each lane is active and visible in CLI display commands. When the lanes are combined to form a single 100GbE interface, display commands will show the primary lane as ***connected*** or ***not connected***; all other lanes will be hidden.

The following sections describe the configuration of CFP2 ports.

10.5.8.1 Configuring a CFP2 Module a as a Single100GbE Interface

To configure the port as a single 100GbE interface (the default configuration), combine the module's ten data lanes by using the **speed** command (**speed forced 100gfull**) on the port's /1 lane (the primary lane).

This configuration is available for all pluggable optics.

Important! The **speed** command is hitless on the 7050X, 7060X, 7250X, 7260X, 7280SE, 7300X, 7320X and 7500E series platforms. On all other platforms, this command restarts the forwarding agent, which will result in traffic disruption.

Step 1 Enter interface Ethernet configuration mode for lane /1 of the CFP2 Ethernet interface.

```
switch(config)#interface ethernet 5/1/1
```

Step 2 Enter the **speed forced 100gfull** command. Depending on the platform, this command may restart the forwarding agent, disrupting traffic on all ports for 60 seconds or more.

```
switch(config-if-Et5/1/1)#speed forced 100gfull
```

Step 3 Use the **show interfaces status** command to confirm the change in configuration.

```
switch(config-if-Et5/1/1)#show interfaces status
Port       Name           Status       Vlan     Duplex Speed  Type         Flags
Et1                       connected    2        full   1G     10GBASE-T
          <-------OUTPUT OMITTED FROM EXAMPLE-------->
Et5/1/1                   connected    1        full   100G   100GBASE-SR1
Et5/2/1                   connected    1        full   100G   100GBASE-SR1
          <-------OUTPUT OMITTED FROM EXAMPLE-------->
```

10.5.8.2 Configuring a CFP2 Module as Ten 10GbE Interfaces

To configure the port as four 10GbE interfaces, use the **speed** command (**speed forced 10000full**) on the port's /1 lane (the primary lane).

This configuration is available only for CFP2-100G-XSR10 optics.

Important! The **speed** command is hitless on the 7050X, 7060X, 7250X, 7260X, 7280SE, 7300X, 7320X and 7500E series platforms. On all other platforms, this command restarts the forwarding agent, which will result in traffic disruption.

Step 1 Enter interface Ethernet configuration mode for lane /1 of the CFP2 Ethernet interface.

```
switch(config)#interface ethernet 5/1/1
```

Step 2 Enter the **speed forced 10000full** command. Depending on the platform, this command may restart the forwarding agent, disrupting traffic on all ports for 60 seconds or more.

```
switch(config-if-Et5/1/1)#speed forced 10000full
```

Step 3 Use the **show interfaces status** command to confirm the change in configuration.

```
switch(config-if-Et5/1/1)#show interfaces status
Port       Name           Status       Vlan     Duplex Speed  Type         Flags
Et1                       connected    2        full   1G     10GBASE-T
          <-------OUTPUT OMITTED FROM EXAMPLE-------->
Et5/1/1                   connected    1        full   10G    100GBASE-SR1
Et5/1/2                   connected    1        full   10G    100GBASE-SR1
Et5/1/3                   connected    1        full   10G    100GBASE-SR1
Et5/1/4                   connected    1        full   10G    100GBASE-SR1
Et5/1/5                   connected    1        full   10G    100GBASE-SR1
Et5/1/6                   connected    1        full   10G    100GBASE-SR1
Et5/1/7                   connected    1        full   10G    100GBASE-SR1
Et5/1/8                   connected    1        full   10G    100GBASE-SR1
Et5/1/9                   connected    1        full   10G    100GBASE-SR1
Et5/1/10                  connected    1        full   10G    100GBASE-SR1
          <-------OUTPUT OMITTED FROM EXAMPLE-------->
```

10.5.9 MXP Ethernet Port Configuration

Each MXP module contains twelve data lanes which can be used individually or combined to form one or more higher-speed interfaces. This allows an MXP Ethernet port to be configured as a single 100GbE interface, up to twelve 10GbE interfaces, or a mixture of 40GbE and 10GbE ports.

MXP ports do not use pluggable optics: instead, an MTP-24 ribbon is inserted directly into the port. The remote end of the MTP 24 ribbon must then be broken out using a splitter cable or cartridge based on the operational mode and speed of the MXP port.

When four lanes of an MXP interface are combined to form a 40GbE port, CLI commands will show the primary lane of that group as ***connected or not connected*** and the other three lanes as ***errdisabled***.

The following sections describe the configuration of MXP interfaces.

10.5.9.1 Configuring an MXP Module as a Single 100GbE Interface

To configure the port as a single 100GbE interface (the default configuration), enter the **speed** command (**speed forced 100gfull**) on the port's /1 lane (the primary lane). This combines lanes 1-10 and disables lanes 11 and 12.

Under this configuration, CLI display commands will show lane /1 as ***connected*** or ***not connected***, and show lanes /2-/12 as ***errdisabled***.

Important! The **speed** command is hitless on the 7050X, 7060X, 7250X, 7260X, 7280SE, 7300X, 7320X and 7500E series platforms. On all other platforms, this command restarts the forwarding agent, which will result in traffic disruption.

Step 1 Enter interface Ethernet configuration mode for lane /1 of the MXP Ethernet interface.

```
switch(config)#interface ethernet 5/49/1
```

Step 2 Enter the **speed forced 100gfull** command. Depending on the platform, this command may restart the forwarding agent, disrupting traffic on all ports for 60 seconds or more.

```
switch(config-if-Et5/49/1)#speed forced 100gfull
```

Step 3 Use the **show interfaces status** command to confirm the change in configuration.

```
switch(config-if-Et5/49/1)#show interfaces status
Port       Name           Status       Vlan     Duplex Speed  Type         Flags
Et1                       connected    2        full   1G     10GBASE-T
          <-------OUTPUT OMITTED FROM EXAMPLE-------->
Et5/49/1                  connected    1        full   100G   100GBASE-SR1
Et5/49/2                  errdisabled  1        unconf unconf 100GBASE-SR1
Et5/49/3                  errdisabled  1        unconf unconf 100GBASE-SR1
Et5/49/4                  errdisabled  1        unconf unconf 100GBASE-SR1
Et5/49/5                  errdisabled  1        unconf unconf 100GBASE-SR1
Et5/49/6                  errdisabled  1        unconf unconf 100GBASE-SR1
Et5/49/7                  errdisabled  1        unconf unconf 100GBASE-SR1
Et5/49/8                  errdisabled  1        unconf unconf 100GBASE-SR1
Et5/49/9                  errdisabled  1        unconf unconf 100GBASE-SR1
Et5/49/10                 errdisabled  1        unconf unconf 100GBASE-SR1
Et5/49/11                 errdisabled  1        unconf unconf 100GBASE-SR1
Et5/49/12                 errdisabled  1        unconf unconf 100GBASE-SR1
          <-------OUTPUT OMITTED FROM EXAMPLE-------->
```

10.5.9.2 Configuring an MXP Module With 40GbE Interfaces

Each set of four lanes on the MXP module is independently configurable as a single 40GbE interface or four 10GbE interfaces. To configure four lanes as a single 40GbE interface, enter the **speed** command (**speed forced 40gfull**) on the group's primary lane (/1, /5, or /9). To revert a group of four lanes to functioning as four independent 10GbE interfaces, enter the **speed forced 10000full** command on the primary lane of the group.

When four lanes of an MXP interface are combined to form a 40GbE port, CLI commands will show the primary lane of that group as ***connected*** or ***not connected*** and the other three lanes as ***errdisabled***. In groups of four lanes which are configured as four independent 10GbE interfaces, each lane will be displayed in the CLI as ***connected*** or ***not connected***.

Note that a **speed forced 100gfull** command entered on the /1 lane takes precedence over **speed forced 40gfull** commands on the /5 and /9 lanes.

Important! The **speed** command is hitless on the 7050X, 7060X, 7250X, 7260X, 7280SE, 7300X, 7320X and 7500E series platforms. On all other platforms, this command restarts the forwarding agent, which will result in traffic disruption.

The example below shows the steps for configuring an MXP module as three 40GbE interfaces.

Step 1 Enter interface Ethernet configuration mode for lane /1 of the MXP Ethernet interface.

```
switch(config)#interface ethernet 5/49/1
```

Step 2 Enter the **speed forced 40gfull** command. Depending on the platform, this command may restart the forwarding agent, disrupting traffic on all ports for 60 seconds or more.

```
switch(config-if-Et5/49/1)#speed forced 40gfull
```

Step 3 Repeat the above steps for lanes /5 and /9.

```
switch(config-if-Et5/49/1)#interface ethernet 5/49/5
switch(config-if-Et5/49/5)#speed forced 40gfull
switch(config-if-Et5/49/5)#interface ethernet 5/49/9
switch(config-if-Et5/49/9)#speed forced 40gfull
```

Step 4 Use the **show interfaces status** command to confirm the change in configuration.

```
switch(config-if-Et5/49/9)#show interfaces status
Port       Name           Status       Vlan     Duplex Speed  Type         Flags
Et1                       connected    2        full   1G     10GBASE-T
          <-------OUTPUT OMITTED FROM EXAMPLE-------->
Et5/49/1                  connected    1        full   40G    100GBASE-SR1
Et5/49/2                  errdisabled  1        unconf unconf 100GBASE-SR1
Et5/49/3                  errdisabled  1        unconf unconf 100GBASE-SR1
Et5/49/4                  errdisabled  1        unconf unconf 100GBASE-SR1
Et5/49/5                  connected    1        full   40G    100GBASE-SR1
Et5/49/6                  errdisabled  1        unconf unconf 100GBASE-SR1
Et5/49/7                  errdisabled  1        unconf unconf 100GBASE-SR1
Et5/49/8                  errdisabled  1        unconf unconf 100GBASE-SR1
Et5/49/9                  connected    1        full   40G    100GBASE-SR1
Et5/49/10                 errdisabled  1        unconf unconf 100GBASE-SR1
Et5/49/11                 errdisabled  1        unconf unconf 100GBASE-SR1
Et5/49/12                 errdisabled  1        unconf unconf 100GBASE-SR1
          <-------OUTPUT OMITTED FROM EXAMPLE-------->
```

10.5.9.3 Configuring an MXP Module as Twelve 10GbE Interfaces

Each lane of an MXP port functions as a 10GbE interface when it is not included in a higher-speed interface configuration (either actively or as an ***errdisabled*** port).

To explicitly configure the port as twelve 10GbE interfaces, use the **speed** command (**speed forced 10000full**) on all twelve lanes of the port.

When each lane is configured as an independent 10GbE interface, CLI display commands show each lane as ***connected*** or ***not connected***.

Important! The **speed** command is hitless on the 7050X, 7060X, 7250X, 7260X, 7280SE, 7300X, 7320X and 7500E series platforms. On all other platforms, this command restarts the forwarding agent, which will result in traffic disruption.

Step 1 Enter interface Ethernet configuration mode for all twelve lanes of the MXP Ethernet interface.

```
switch(config)#interface ethernet 5/49/1-12
```

Step 2 Enter the **speed forced 10000full** command. Depending on the platform, this command may restart the forwarding agent, disrupting traffic on all ports for 60 seconds or more.

```
switch(config-if-Et5/49/1-12)#speed forced 10000full
```

Step 3 Use the **show interfaces status** command to confirm the change in configuration.

```
switch(config-if-Et5/49/1-12)#show interfaces status
Port       Name           Status       Vlan     Duplex Speed  Type         Flags
Et1                       connected    2        full   1G     10GBASE-T
          <-------OUTPUT OMITTED FROM EXAMPLE-------->
Et5/1/1                   connected    1        full   10G    100GBASE-SR1
Et5/1/2                   connected    1        full   10G    100GBASE-SR1
Et5/1/3                   connected    1        full   10G    100GBASE-SR1
Et5/1/4                   connected    1        full   10G    100GBASE-SR1
Et5/1/5                   connected    1        full   10G    100GBASE-SR1
Et5/1/6                   connected    1        full   10G    100GBASE-SR1
Et5/1/7                   connected    1        full   10G    100GBASE-SR1
Et5/1/8                   connected    1        full   10G    100GBASE-SR1
Et5/1/9                   connected    1        full   10G    100GBASE-SR1
Et5/1/10                  connected    1        full   10G    100GBASE-SR1
          <-------OUTPUT OMITTED FROM EXAMPLE-------->
```

10.5.10 Port Speed Capabilities

The supported speeds supported on each Arista platform per interface type are described in Table 10-6.

Table 10-6 Supported Speeds (GbE)

Platform	SFP+	QSFP+	QSFP100	MXP	CFP2
7050	1, 10	1, 10, 40	N/A	N/A	N/A
7050X	1, 10	1, 10, 40	N/A	10, 40	N/A
7060X	1, 10	10, 40	10, 25, 40, 50, 100	N/A	N/A
7250X	N/A	1, 10, 40	N/A	N/A	N/A
7260X	1, 10	10, 40	10, 25, 40, 50, 100	N/A	N/A
7150S	1, 10	1, 10, 40	N/A	N/A	N/A
7048T	1, 10	N/A	N/A	N/A	N/A
7500	1, 10	1, 10, 40	N/A	N/A	N/A
7500E	1, 10	1, 10, 40	10, 40, 100	10, 40, 100	10, 100
7280SE	1, 10	1, 10, 40	10, 40, 100	10, 40, 100	N/A
7210T	1, 10	N/A	N/A	N/A	N/A

10.5.11 Agile Ports

An agile port is an interface that can function as a 10G port or can subsume a predefined set of 10G interfaces to form an interface with higher speed capabilities.

The set of interfaces that can be combined to form a higher speed port is restricted by the hardware configuration. Only interfaces that pass through a common PHY component can be combined. One interface within a combinable set is designated as the primary port.

- To view the set of available agile ports and the subsumable interfaces that comprise them, enter **show platform fm6000 agileport map**.
- To configure the primary port as a higher speed port, enter **speed forced 40gfull** or **speed auto 40gfull**.
- To revert the primary port and its subsumed ports to 10G interfaces, enter **no speed**.

Example

- These commands displays the agile port map for the switch, then configures ethernet interface 13 as a 40G port.

```
switch#show platform fm6000 agileport map

------------------------------------------------------------------
Agile Ports       |        Interfaces subsumed in 40G link
------------------------------------------------------------------
Ethernet1         |  Ethernet3      Ethernet5      Ethernet7
Ethernet2         |  Ethernet4      Ethernet6      Ethernet8
Ethernet13        |  Ethernet15     Ethernet17     Ethernet19
Ethernet14        |  Ethernet16     Ethernet18     Ethernet20
switch#config
switch(config)#interface ethernet 13
switch(config-if-Et13)#speed forced 40gfull

  WARNING!  Executing this command will cause the forwarding agent
            to be restarted. All interfaces will briefly drop links
            and forwarding on all interfaces will momentarily stop.


            Do you wish to proceed with this command? [y/N]

Ethernet17 configured for 40G.
Ethernet15, Ethernet17 and Ethernet19 are now subsumed.
switch(config-if-Et13)#
This command reverts the agile 40G port to a 10G port and frees its subsumed ports
as individual 10G ports.
switch(config-if-Et13)#no speed

  WARNING!  Executing this command will cause the forwarding agent
            to be restarted. All interfaces will briefly drop links
            and forwarding on all interfaces will momentarily stop.


            Do you wish to proceed with this command? [y/N]

Ethernet13 no longer configured for 40G.
Ethernet15, Ethernet17 and Ethernet19 are now free.
switch(config-if-Et13)#
```

10.5.12 Subinterface Configuration

For a subinterface to be operational on an Ethernet or port channel interface, the parent interface must be configured as a routed port and be administratively up, and a VLAN must be configured on the subinterface. If the parent interface goes down, all subinterfaces automatically go down as well, but will come back up with the same configuration once the parent interface is up.

Note that a port channel should not contain Ethernet interfaces with subinterfaces configured on them, and that subinterfaces cannot be members of a port channel.

Subinterfaces are named by adding a period followed by a unique subinterface number to the name of the parent interface. Note that the subinterface number has no relation to the ID of the VLAN corresponding to the subinterface.

A maximum of 750 subinterfaces can be configured on a switch, and a maximum of 250 subinterfaces can be configured under a single parent interface.

Subinterfaces are available on the following platforms:

- DCS-7050X
- DCS-7060X
- DCS-7250X
- DCS-7260X
- DCS-7280E
- DCS-7300X
- DCS-7320X
- DCS-7500E

10.5.12.1 Creating a Subinterface

To create a subinterface on an Ethernet or port channel interface:

Step 1 Bring up the parent interface and ensure that it is configured as a routed port.

```
switch(config)#interface Ethernet1/1
switch(config-if-Et1/1)#no switchport
switch(config-if-Et1/1)#no shutdown
```

Step 2 Configure a VLAN on the subinterface. The **encapsulation dot1q vlan** command is also used for VLAN translation, but in this context it associates a VLAN with the subinterface.

```
switch(config-if-Et1/1)#interface Ethernet1/1.1
switch(config-if-Et1/1.1)#encapsulation dot1q vlan 100
```

Step 3 Configure an IP address on the subinterface (optional) and ensure that it is up.

```
switch(config-if-Et1/1)#ip address 10.0.0.1/24
switch(config-if-Et1/1)#no shutdown
switch(config-if-Et1/1)#
```

10.5.12.2 Creating a Range of Subinterfaces

A range of subinterfaces can also be configured simultaneously. The following example configures subinterfaces 1 to 100 on Ethernet interface 1/1, and assigns VLANs 501 through 600 to them. Note that the range of interfaces must be the same size as the range of VLAN IDs.

Example

```
switch(config)#interface eth1/1.1-100
switch(config-if-Et1/1.1-100)no shutdown
switch(config-if-Et1/1.1-100)encapsulation dot1q vlan {501,600}
switch(config-if-Et1/1.1-100)exit
switch(config)#
```

10.5.12.3 Parent Interface Configuration

For subinterfaces to function, the parent interface must be administratively up and configured as a routed port.

Some settings are inherited by subinterfaces from the parent interface. These include QoS (trust mode and default DSCP) and MTU.

Additionally, on the DCS-7050X, DCS-7250X, and DCS-7300X platforms, the parent interface may be configured with an IP address. In this case, untagged packets are treated as incoming traffic on the parent interface

10.5.12.4 Configuring Routing Features on a Subinterface

Once a subinterface is created, the following features can be configured on it:

- Unicast and multicast routing
- BGP, OSPF, ISIS, PIM
- VRF
- VRRP
- SNMP
- Inheritance of QoS (trust mode and default DSCP) and MTU settings from the parent interface

Additionally, these features can be configured on subinterfaces on Arad (DCS-7500E and DCS-7280E) platforms:

- Subinterface counters on ingress
- VXLAN
- MPLS
- GRE
- PBR
- QoS

10.5.12.5 Displaying Subinterface Information

Subinterface information is displayed using the same show commands as for other interfaces.

Examples

This command displays summary information for all IP interfaces on the switch, including subinterfaces.

```
switch>show ip interfaces brief
 Interface      IP Address     Status    Protocol     MTU
 Ethernet1/1    10.1.1.1/24    up        up          1500
 Ethernet1/1.1  10.0.0.1/24    up        up          1500
 Ethernet1/2    unassigned     up        up          1500
```

This command displays information for subinterface Ethernet 1/1.1.

```
switch>show interface ethernet 1/1.1
Ethernet1/1.1 is down, line protocol is lowerlayerdown (notconnect)
  Hardware is Subinterface, address is 001c.735d.65dc
  Internet address is 10.0.0.1/24
  Broadcast address is 255.255.255.255
  Address determined by manual configuration
  IP MTU 1500 bytes , BW 10000000 kbit
  Down 59 seconds
switch>
```

This command displays status information for all subinterfaces configured on the switch.

```
switch>show interfaces status sub-interfaces
Port       Name    Status       Vlan     Duplex Speed  Type                Flags
Et1.1              connect      101      full   10G    dot1q-encapsulation
Et1.2              connect      102      full   10G    dot1q-encapsulation
Et1.3              connect      103      full   10G    dot1q-encapsulation
Et1.4              connect      103      full   10G    dot1q-encapsulation
switch>
```

10.5.13 Autonegotiated Settings

In autonegotiation, the transmission speed, duplex setting, and flow control parameters used for Ethernet-based communication can be automatically negotiated between connected devices to establish optimized common settings.

10.5.13.1 Speed and Duplex

The **speed** command affects the transmission speed and duplex setting for the configuration mode interface. When a **speed forced** command is in effect on an interface, autonegotiation of speed and duplex settings is disabled for the interface; to enable autonegotiation, use the **speed auto** command.

The scope and effect of the **speed** command depends on the interface type; see Ethernet Interfaces and Ethernet Configuration Procedures for detailed information on the speed settings for different interfaces.

10.5.13.2 Flow Control

Flow control is a data transmission option that temporarily stops a device from sending data because of a peer data overflow condition. If a device sends data faster than the receiver can accept it, the receiver's buffer can overflow. The receiving device then sends a PAUSE frame, instructing the sending device to halt transmission for a specified period.

Flow control commands configure administrative settings for flow control packets.

- The **flowcontrol receive** command configures the port's ability to receive flow control pause frames.
  - ***off***: port does not process pause frames that it receives.
  - ***on***: port processes pause frames that it receives.
  - ***desired***: port autonegotiates; processes pause frames if peer is set to ***send*** or ***desired***.
- The **flowcontrol send** command configures the port's ability to transmit flow control pause frames.
  - ***off***: port does not send pause frames.
  - ***on***: port sends pause frames.
  - ***desired***: port autonegotiates; sends pause frames if peer is set to ***receive*** or ***desired***.

Desired is not an available parameter option. Ethernet data ports cannot be set to ***desired***. Management ports are set to ***desired*** by default and with the **no flowcontrol receive** command.

The port linking process includes flow control negotiation. Ports must have compatible flow control settings to create a link. Table 10-7 lists the compatible flow control settings.

Table 10-7 Compatible Settings for Flow Control Negotiation

local port	peer port
receive on	send on or send desired
receive off	send off or send desired
receive desired	send on , send off, or send desired
send on	receive on or receive desired
send off	receive off or receive desired
send desired	receive on , receive off, or receive desired

Example

- These commands set the flow control receive and send to **on** on Ethernet interface 5.

```
switch(config)#interface ethernet 5
switch(config-if-Et5)#flowcontrol receive on
switch(config-if-Et5)#flowcontrol send on
switch(config-if-Et5)#
```

10.5.14 Displaying Ethernet Port Properties

Show commands are available to display various Ethernet configuration and operational status on each interface. Ethernet settings that are viewable include:

- Port Type
- PHY Status
- Negotiated Settings
- Flow Control
- Capabilities

Port Type

The port type is viewable from the output of **show interfaces status**, **show interfaces capabilities**, and **show interfaces transceiver properties** commands.

Example

- This **show interfaces status** command displays the status of Ethernet interfaces 1-5.

```
switch>show interfaces status
Port      Name              Status       Vlan        Duplex  Speed Type
Et1                         connected    1             full    10G 10GBASE-SRL
Et2                         connected    1             full    10G 10GBASE-SRL
Et3                         connected    1             full    10G 10GBASE-SRL
Et4                         connected    1             full    10G 10GBASE-SRL
Et5                         notconnect   1             full    10G Not Present
switch>
```

- This **show interfaces capabilities** command displays the speed, duplex, and flow control capabilities of Ethernet interfaces 2 and 18.

```
switch>show interfaces ethernet 2,18 capabilities
Ethernet2
  Model:        DCS-7150S-64-CL
  Type:         10GBASE-CR
  Speed/Duplex: 10G/full,40G/full,auto
  Flowcontrol:  rx-(off,on,desired),tx-(off,on,desired)
Ethernet18
  Model:        DCS-7150S-64-CL
  Type:         10GBASE-SR
  Speed/Duplex: 10G/full
  Flowcontrol:  rx-(off,on),tx-(off,on)
switch>
```

- This command displays the media type, speed, and duplex properties for Ethernet interfaces 1.

```
switch>show interfaces ethernet 1 transceiver properties
Name : Et1
Administrative Speed: 10G
Administrative Duplex: full
Operational Speed: 10G (forced)
Operational Duplex: full (forced)
Media Type: 10GBASE-SRL
```

PHY

PHY information for each Ethernet interface is viewed by entering the **show interfaces phy** command.

Example

- This command summarizes PHY information for Ethernet interfaces 1-3.

```
switch>show interfaces ethernet 1-3 phy
Key:
   U    = Link up
   D    = Link down
   R    = RX Fault
   T    = TX Fault
   B    = High BER
   L    = No Block Lock
   A    = No XAUI Lane Alignment
   0123 = No XAUI lane sync in lane N

                                  State    Reset
Port           PHY state        Changes    Count PMA/PMD PCS   XAUI
-------------- --------------- -------- -------- ------- ----- --------
Ethernet1      linkUp             14518     1750 U..     U.... U.......
Ethernet2      linkUp             13944     1704 U..     U.... U.......
Ethernet3      detectingXcvr          3        1              D..A0123
switch>
```

Negotiated Settings

Speed, duplex, and flow control settings are displayed through the show interfaces capabilities, PHY information for each Ethernet interface is viewed by entering the **show interfaces capabilities**, **show flowcontrol**, and **show interfaces status** commands.

Example

- This command displays speed/duplex and flow control settings for Ethernet interface 1.

```
switch>show interfaces ethernet 1 capabilities
Ethernet1
  Model:        DCS-7150S-64-CL
  Type:         10GBASE-SR
  Speed/Duplex: 10G/full
  Flowcontrol:  rx-(off,on),tx-(off,on)
switch>
```

- This command shows the flow control settings for Ethernet interfaces 1-2.

```
switch>show flowcontrol interface ethernet 1-2
Port       Send FlowControl  Receive FlowControl  RxPause       TxPause
           admin    oper     admin    oper
---------  -------- -------- -------- --------    ------------- -------------
Et1        off      off      off      off         0             0
Et2        off      off      off      off         0             0
switch>
```

- This command displays the speed type and duplex settings for management interfaces 1-2.

```
switch>show interfaces management 1-2 status
Port      Name              Status       Vlan        Duplex  Speed Type
Ma1                         connected    routed      a-full a-100M 10/100/1000
Ma2                         connected    routed      a-full   a-1G 10/100/1000
switch>
```

10.5.15 Configuring Ingress Traffic-Class Counters

Ingress traffic class counter support is enabled in order to display per traffic-class counters on ingress interfaces, and supported on routed-ports and sub-interfaces. Both packet and octet counts are displayed.

Example

- This command enables traffic-class counter support.

```
switch(config)#hardware counter feature traffic-class in
switch(config)#
```

- This command enables TCAM profile ‘tc-counters’ if this profile is configured.

```
switch(config)#hardware tcam profile tc-counters
switch(config)#
```

10.6 Ethernet Configuration Commands

Global Configuration Commands

- hardware port-group
- interface ethernet
- interface ethernet create
- interface management
- transceiver qsfp default-mode
- transceiver channel

Interface Configuration Commands – Ethernet and Management Interfaces

- flowcontrol receive
- flowcontrol send
- link-debounce
- mac-address
- speed

Interface Display Commands

- show flowcontrol
- show hardware counter
- show hardware port-group
- show interfaces capabilities
- show interfaces counters
- show interfaces counters bins
- show interfaces counters errors
- show interfaces counters queue
- show interfaces counters rates
- show interfaces negotiation
- show interfaces phy
- show interfaces status
- show interfaces status errdisabled
- show interfaces transceiver
- show interfaces transceiver channels
- show interfaces transceiver hardware
- show interfaces transceiver properties
- show platform fm6000 agileport map

flowcontrol receive

The **flowcontrol receive** command configures administrative settings for inbound flow control packets. Ethernet ports use flow control to delay packet transmission when port buffers run out of space. Ports transmit a pause frame when their buffers are full, signaling their peer ports to delay sending packets for a specified period.

The **flowcontrol receive** command configures the configuration mode port's ability to receive flow control pause frames.

- ***off***: port does not process pause frames that it receives.
- ***on***: port processes pause frames that it receives.
- ***desired***: port autonegotiates flow control; processes pause frames if the peer is set to ***send desired***.

  Desired is not an available parameter option. Ethernet data ports cannot be set to ***desired***. Management ports are set to ***desired*** by default and with the **no flowcontrol receive** command.

The port linking process includes flow control negotiation. Ports must have compatible flow control settings to create a link. Table 10-8 lists the compatible flow control settings.

Table 10-8 Compatible Settings for Flow Control Negotiation – Local Port Receiving

local port	peer port
receive on	send on or send desired
receive off	send off or send desired
receive desired	send on , send off, or send desired

The **no flowcontrol receive** and **default flowcontrol receive** commands restore the default flow control setting for the configuration mode interface by removing the corresponding **flowcontrol receive** command from ***running-config***. The default setting is ***off*** for Ethernet data ports and ***desired*** for Management ports.

Command Mode

Interface-Ethernet Configuration
Interface-Management Configuration

Command Syntax

```
flowcontrol receive STATE
no flowcontrol receive
default flowcontrol receive
```

Parameters

- ***STATE*** flow control pause frame processing setting. Options include:
  - **on**
  - **off**

Examples

- These commands set the flow control received on Ethernet interface 5.

```
switch(config)#interface ethernet 5
switch(config-if-Et5)#flowcontrol receive on
switch(config-if-Et5)#
```

flowcontrol send

The **flowcontrol send** command configures administrative settings for outbound flow control packets. Ethernet ports use flow control to delay packet transmission when port buffers run out of space. Ports transmit a pause frame when their buffers are full, signaling their peer ports to delay sending packets for a specified period.

The **flowcontrol send** command configures the configuration mode port's ability to transmit flow control pause frames.

- ***off***: port does not send pause frames.
- ***on***: port sends pause frames.
- ***desired***: port autonegotiates flow control; sends pause frames if the peer is set to ***receive desired***.

  Desired is not an available parameter option. Ethernet data ports cannot be set to ***desired***. Management ports are set to ***desired*** by default and with the **no flowcontrol send** command.

The port linking process includes flow control negotiation. Ports must have compatible flow control settings to create a link. Table 10-9 lists the compatible flow control settings.

Table 10-9 Compatible Settings for Flow Control Negotiation – Local Port Transmitting

local port	peer port
send on	receive on or receive desired
send off	receive off or receive desired
send desired	receive on , receive off, or receive desired

The **no flowcontrol send** and **default flowcontrol send** commands restore the default flow control setting for the configuration mode interface by removing the corresponding **flowcontrol send** command from ***running-config***. The default setting is ***off*** for Ethernet data ports and ***desired*** for Management ports.

Command Mode

Interface-Ethernet Configuration
Interface-Management Configuration

Command Syntax

```
flowcontrol send STATE
no flowcontrol send
default flowcontrol send
```

Parameters

- ***STATE*** flow control send setting. Options include
  - **on**
  - **off**

Examples

- These commands set the flow control sent on Ethernet interface 5.

```
switch(config)#interface ethernet 5
switch(config-if-Et5)#flowcontrol send on
switch(config-if-Et5)#
```

hardware port-group

The **hardware port-group** command configures a port group to activate a 40GBASE (QSFP+) interface or four 10GBASE (SFP+) interfaces, affecting QSFP+ and SFP+ availability.

The **no hardware port-group** and **default hardware port-group** commands restore a port group's default setting by removing the corresponding **hardware port-group** command from ***running-config***. The QSFP+ interface is active by default in each port group.

The **hardware port-group** command is available on DCS-7050Q-16 and DCS-7050QX-32S switches, and has different parameters on each platform.

Command Mode

Global Configuration

Command Syntax

```
hardware port-group group_number select PORT_LIST
no hardware port-group group_number
default hardware port-group group_number
```

Parameters

- *group_number* label of the port group. Valid options are **1** and **2** on the 7050Q-16; only **1** is available on the 7050QX-32S.
- ***PORT_LIST*** ports activated by command. Options vary by platform and depend on *group_number* value.

  DCS-7050Q-16

  - **Et15/1-4** activates QSFP+ port on port group 1. Available when *group_number* is 1.
  - **Et16/1-4** activates QSFP+ port on port group 2. Available when *group_number* is 2.
  - **Et17-20** activates SFP+ ports on port group 1. Available when *group_number* is 1.
  - **Et21-23** activates SFP+ ports on port group 2. Available when *group_number* is 2.

  DCS-7050QX-32S

  - **Et1-4** activates SFP+ ports on port group 1. Available when *group_number* is 1.
  - **Et5/1-4** activates QSFP+ port on port group 1. Available when *group_number* is 1.

Example

- These commands enable the QSFP+ interface in port group 1 and SFP+ interfaces in port group 2 on a DCS-7050Q-16 switch, display the port group status, and display interface status.

```
switch(config)#hardware port-group 1 select Et15/1-4
switch(config)#hardware port-group 2 select Et21-24
switch(config)#show hardware port-group

Portgroup: 1    Active Ports: Et17-20
Port            State
-----------------------------------------
Ethernet17      ErrDisabled
Ethernet18      ErrDisabled
Ethernet19      ErrDisabled
Ethernet20      ErrDisabled
Ethernet15/1    Active
Ethernet15/2    Active
Ethernet15/3    Active
Ethernet15/4    Active

Portgroup: 2    Active Ports: Et16/1-4
Port            State
-----------------------------------------
Ethernet16/1    Active
Ethernet16/2    Active
Ethernet16/3    Active
Ethernet16/4    Active
Ethernet21      ErrDisabled
Ethernet22      ErrDisabled
Ethernet23      ErrDisabled
Ethernet24      ErrDisabled
switch(config)#show interfaces status
Port      Name              Status       Vlan        Duplex  Speed Type
Et1/1                       connected    in Po621      full    40G 40GBASE-CR4
Et1/2                       errdisabled  inactive    unconf unconf 40GBASE-CR4
<-------OUTPUT OMITTED FROM EXAMPLE-------->
Et15/1                      connected    in Po711      full    40G 40GBASE-CR4
Et15/2                      errdisabled  inactive    unconf unconf Not Present
Et15/3                      errdisabled  inactive    unconf unconf Not Present
Et15/4                      errdisabled  inactive    unconf unconf Not Present
Et16/1                      errdisabled  inactive    unconf unconf Not Present
Et16/2                      errdisabled  inactive    unconf unconf Not Present
Et16/3                      errdisabled  inactive    unconf unconf Not Present
Et16/4                      errdisabled  inactive    unconf unconf Not Present
Et17                        errdisabled  inactive    unconf unconf Not Present
Et18                        errdisabled  inactive    unconf unconf Not Present
Et19                        errdisabled  inactive    unconf unconf Not Present
Et20                        errdisabled  inactive    unconf unconf Not Present
Et21                        connected    425           full    10G 10GBASE-SRL
Et22                        connected    611           full    10G 10GBASE-SRL
Et23                        connected    in Po998      full    10G 10GBASE-SLR
Et24                        connected    in Po998      full    10G 10GBASE-SLR
switch(config)#
```

interface ethernet

The **interface ethernet** command places the switch in Ethernet-interface configuration mode for the specified interfaces. The command can specify a single interface or multiple interfaces.

Ethernet interfaces are physical interfaces and are not created or removed.

Interface management commands include:

- description
- exit
- load-interval
- mtu
- shutdown (Interfaces)

Ethernet management commands include:

- flowcontrol
- mac-address
- speed

Chapters describing supported protocols and other features list additional configuration commands available from Ethernet interface configuration mode.

Command Mode

Global Configuration

Command Syntax

interface ethernet *e_range*

Parameters

- *e_range* Ethernet interfaces (number, range, or comma-delimited list of numbers and ranges).

  Valid Ethernet numbers depend on the switch's available Ethernet interfaces.

Example

- This command enters interface configuration mode for Ethernet interfaces 1 and 2:

```
switch(config)#interface ethernet 1-2
switch(config-if-Et1-2)#
```

- This command enters interface configuration mode for Ethernet interface 1:

```
switch(config)#interface ethernet 1
switch(config-if-Et1)#
```

interface ethernet create

The **interface ethernet create** command is used to configure a range of Ethernet subinterfaces. The command places the switch in Ethernet-interface configuration mode for the specified range of subinterfaces.

Command Mode

Global Configuration

Command Syntax

interface ethernet create *sub_range*

Parameters

- *sub_range* range of subinterfaces to be configured. Subinterfaces are named by adding a period followed by a unique subinterface number to the name of the parent interface. A maximum of 750 subinterfaces can be configured on a switch, and a maximum of 250 subinterfaces can be configured under a single parent interface.

Example

- This command enters interface configuration mode for Ethernet subinterfaces 1/1.1-100:

```
switch(config)#interface ethernet create 1/1.100
switch(config-if-Et1/1.1-100)#
```

interface management

The **interface management** command places the switch in management-interface configuration mode for the specified interfaces. The list can specify a single interface or multiple interfaces if the switch contains more than one management interface.

Management interfaces are physical interfaces and are not created or removed.

Interface management commands include:

- description
- exit
- load-interval
- mtu
- shutdown (Interfaces)

Ethernet management commands include:

- flowcontrol
- mac-address
- speed

Chapters describing supported protocols and other features list additional configuration commands available from management-interface configuration mode.

Command Mode

Global Configuration

Command Syntax

`interface management` *`m_range`*

Parameters

- *m_range* Management interfaces (number, range, or comma-delimited list of numbers and ranges).

  Valid management numbers depend on the switch's available management interfaces. A value of 0, where available, configures the virtual management interface on a dual-supervisor modular switch. Management interface 0 accesses management port 1 on the active supervisor of a dual-supervisor modular switch.

Examples

- This command enters interface configuration mode for management interfaces 1 and 2.

```
switch(config)#interface management 1-2
switch(config-if-Ma1-2)#
```

- This command enters interface configuration mode for management interface 1:

```
switch(config)#interface management 1
switch(config-if-Ma1)#
```

link-debounce

The **link-debounce** command configures the link debounce time for the configuration mode interface. Link debounce time is the time that advertisements for new link states are delayed after the link state is established. By default, debounce time is set to zero, disabling link debounce.

Debounce times for link-up and link-down transitions can be independently configured.

- Link-up debounce time: the delay before an interface advertises link down to link up transitions.
- Link-down debounce time: the delay before an interface advertises link up to link down transitions.

The **no link-debounce** and **default link-debounce** commands restore the default debounce setting for the configuration mode interface by removing the corresponding **link-debounce** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration
Interface-Management Configuration

Command Syntax

```
link-debounce time WAIT_TIME
no link-debounce
default link-debounce
```

Parameters

- ***WAIT_TIME*** link debounce period (milliseconds). Options include
  - ***<0 - 30000>*** One debounce value assigned as both link up and link down.
  - ***<0 - 30000> <0 - 30000>*** Two debounce values: link up is first, link down is second.

  All debounce values range from 0 (disabled) to 30000 (30 seconds).

Examples

- These commands set the link-up and link-down debounce period to 10 seconds on Ethernet interface 5.

```
switch(config)#interface ethernet 5
switch(config-if-Et5)#link-debounce time 10000
switch(config-if-Et5)#
```

- These commands set the link-up debounce to 10 seconds and the link-down debounce period to zero on Ethernet interface 5.

```
switch(config)#interface ethernet 5
switch(config-if-Et5)#link-debounce time 10000 0
switch(config-if-Et5)#
```

- These commands set the link-up debounce to zero and the link-down debounce period to 12.5 seconds on Ethernet interface 5.

```
switch(config)#interface ethernet 5
switch(config-if-Et5)#link-debounce time 0 12500
switch(config-if-Et5)#
```

mac-address

The **mac-address** command assigns a MAC address to the configuration mode interface. An interface's default MAC address is its burn-in address.

The **no mac-address** and **default mac-address** commands revert the interface to its default MAC address by removing the corresponding **mac-address** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration
Interface-Management Configuration

Command Syntax

```
mac-address address
no mac-address
default mac-address
```

Parameters

- *address* MAC address assigned to the interface. Format is dotted hex notation (H.H.H). Disallowed addresses are 0.0.0 and FFFF.FFFF.FFFF.

Example

- This command assigns the MAC address of 001c.2804.17e1 to Ethernet interface 7, then displays interface parameters, including the assigned address.

```
switch(config)#interface ethernet 7
switch(config-if-Et7)#mac-address 001c.2804.17e1
switch(config-if-Et7)#show interface ethernet 7
Ethernet3 is up, line protocol is up (connected)
  Hardware is Ethernet, address is 001c.2804.17e1 (bia 001c.7312.02e2)
  Description: b.e45
  MTU 9212 bytes, BW 10000000 Kbit
  Full-duplex, 10Gb/s, auto negotiation: off
  Last clearing of "show interface" counters never
  5 seconds input rate 7.84 kbps (0.0% with framing), 10 packets/sec
  5 seconds output rate 270 kbps (0.0% with framing), 24 packets/sec
     1363799 packets input, 222736140 bytes
     Received 0 broadcasts, 290904 multicast
     0 runts, 0 giants
     0 input errors, 0 CRC, 0 alignment, 0 symbol
     0 PAUSE input
     2264927 packets output, 2348747214 bytes
     Sent 0 broadcasts, 28573 multicast
     0 output errors, 0 collisions
     0 late collision, 0 deferred
     0 PAUSE output
switch(config-if-Et7)#
```

show flowcontrol

The **show interfaces flowcontrol** command displays administrative and operational flow control data for the specified interfaces. Administrative data is the parameter settings stored in ***running-config*** for the specified interface; the switch uses these settings to negotiate flow control with the peer switch. Operational data is the resolved flow control setting that controls the port's behavior.

Command Mode

EXEC

Command Syntax

```
show flowcontrol [INTERFACE]
show [INTERFACE] flowcontrol
```

Parameters

- ***INTERFACE*** Interface type and number for which flow control data is displayed.
  - <no parameter> all interfaces.
  - **ethernet** *e_range* Ethernet interfaces in the specified range.
  - **management** *m_range* Management interfaces in the specified range.

    Valid *e_range* and *m_range* formats include number, number range, or comma-delimited list of numbers and ranges.

Example

- This command shows the settings for Ethernet interfaces 1-10.

```
switch>show flowcontrol interface ethernet 1-10
Port       Send FlowControl  Receive FlowControl  RxPause       TxPause
           admin    oper     admin    oper
---------  -------- -------- -------- --------    ------------- -------------
Et1        off      off      off      off         0             0
Et2        off      off      off      off         0             0
Et3        off      off      off      off         0             0
Et4        off      off      off      off         0             0
Et5        off      off      off      off         0             0
Et6        off      off      off      off         0             0
Et7        off      off      off      off         0             0
Et8        off      off      off      off         0             0
Et9        off      off      off      off         0             0
Et10       off      off      off      off         0             0
switch>
```

show hardware counter

The **show hardware counter** command displays counter events across time intervals.

Command Mode

EXEC

Command Syntax

show hardware counter

Example

- This command displays counter events across all time intervals, which are currently more than one standard deviation apart from a given time interval.

```
switch(config-handler-eventHandler1-counters)#show hardware counter events
-------------------------------------------------------------------------------------------
Interval | Event Name  | Chip | First              | Last               | Count | Z-Score
         |             | Name | Occurrence         | Occurrence         |       |
-------------------------------------------------------------------------------------------
5 Min    | MacCounters |  All | 2017-01-31 09:31:35 | 2017-01-31 09:44:32 |    5 | -6.9430
10 Min   | MacCounters |  All | 2017-01-31 09:39:43 | 2017-01-31 09:44:32 |    3 | -4.8123
-------------------------------------------------------------------------------------------
switch(config-handler-eventHandler1-counters)#
```

show hardware port-group

The **show hardware port-group** command displays the status of DCS-7050Q-16 port-groups. Port groups contain one QSFP+ interface and a set of four SFP+ interfaces. In each port group, either the QSFP+ interface or the SFP+ interface set is enabled. The port groups are configured independent of each other.

- Port group 1 contains interface 15 (QSFP+) and interfaces 17-20 (SFP+).
- Port group 2 contains interface 16 (QSFP+) and interfaces 21-24 (SFP+).

Command Mode

EXEC

Command Syntax

```
show hardware port-group
```

Guidelines

The **hardware port-group** command is available on on DCS-7050Q-16 switches.

Example

- This command displays the status of ports in the two port groups on a DCS-7050Q-16 switch.

```
switch>show hardware port-group

Portgroup: 1    Active Ports: Et15/1-4
Port            State
----------------------------------------
Ethernet17      ErrDisabled
Ethernet18      ErrDisabled
Ethernet19      ErrDisabled
Ethernet20      ErrDisabled
Ethernet15/1    Active
Ethernet15/2    Active
Ethernet15/3    Active
Ethernet15/4    Active

Portgroup: 2    Active Ports: Et16/1-4
Port            State
----------------------------------------
Ethernet16/1    Active
Ethernet16/2    Active
Ethernet16/3    Active
Ethernet16/4    Active
Ethernet21      ErrDisabled
Ethernet22      ErrDisabled
Ethernet23      ErrDisabled
Ethernet24      ErrDisabled
switch>
```

show interfaces capabilities

The **show interfaces capabilities** command displays the model number, interface type, duplex mode, and flow control settings of the specified interfaces. The capabilities command is available on Ethernet and management interfaces.

Command Mode

EXEC

Command Syntax

```
show interfaces [INTERFACE] capabilities
```

Parameters

- ***INTERFACE*** Interface type and numbers. Options include:
  - <no parameter> all interfaces.
  - **ethernet** *e_range* Ethernet interface range specified by *e_range*.
  - **management** *m_range* Management interface range specified by *m_range*.

    Valid *e_range* and *m_range* formats include number, number range, or comma-delimited list of numbers and ranges.

Examples

- This command displays the model number, interface type, duplex mode and flow control settings for Ethernet interfaces 2 and 18.

```
switch>show interfaces ethernet 2,18 capabilities
Ethernet2
  Model:        DCS-7150S-64-CL
  Type:         10GBASE-CR
  Speed/Duplex: 10G/full,40G/full,auto
  Flowcontrol:  rx-(off,on,desired),tx-(off,on,desired)
Ethernet18
  Model:        DCS-7150S-64-CL
  Type:         10GBASE-SR
  Speed/Duplex: 10G/full
  Flowcontrol:  rx-(off,on),tx-(off,on)
switch>
```

show interfaces counters

The **show interfaces counters** command displays packet and byte counters for the specified interfaces. Counters displayed by the command include:

- inbound bytes
- inbound unicast packets
- inbound multicast packets
- inbound broadcast packets
- outbound bytes
- outbound unicast packets
- outbound multicast packets
- outbound broadcast packets

Command Mode

EXEC

Command Syntax

```
show interfaces [INTERFACE] counters
```

Parameters

- ***INTERFACE*** Interface type and numbers. Options include:
  - <no parameter> all interfaces.
  - **ethernet** *e_range* Ethernet interface range specified by *e_range*.
  - **management** *m_range* Management interface range specified by *m_range*.
  - **port-channel** *p_range* Port-Channel Interface range specified by *p_range*.

Related Commands

- **show interfaces counters bins**
- **show interfaces counters errors**
- **show interfaces counters queue**
- **show interfaces counters rates**

Examples

- This command displays byte and packet counters for Ethernet interfaces 1 and 2.

```
switch>show interfaces ethernet 1-2 counters
Port                 InOctets     InUcastPkts     InMcastPkts     InBcastPkts
Et1               99002845169        79116358           75557            2275
Et2               81289180585        76278345           86422              11

Port                OutOctets    OutUcastPkts    OutMcastPkts    OutBcastPkts
Et1                4347928323         6085482          356173            2276
Et2                4512762190         5791718          110498              15
switch>
```

show interfaces counters bins

The **show interfaces counters bins** command displays packet counters, categorized by packet length, for the specified interfaces. Packet length counters that the command displays include:

- 64 bytes
- 65-127 bytes
- 128-255 bytes
- 256-511 bytes
- 512-1023 bytes
- 1024-1522 bytes
- larger than 1522 bytes

Command Mode

EXEC

Command Syntax

```
show interfaces [INTERFACE] counters bins
```

Parameters

- ***INTERFACE*** Interface type and numbers. Options include:
  - <no parameter> all interfaces.
  - **ethernet** *e_range* Ethernet interface range specified by *e_range*.
  - **management** *m_range* Management interface range specified by *m_range*.
  - **port-channel** *p_range* Port-Channel Interface range specified by *p_range*.

Related Commands

- **show interfaces counters**
- **show interfaces counters errors**
- **show interfaces counters queue**
- **show interfaces counters rates**

Examples

- This command displays packet counter results for Ethernet interfaces 1 and 2.

```
switch>show interfaces ethernet 1-2 counters bins
Input
Port                64 Byte      65-127 Byte     128-255 Byte     256-511 Byte
--------------------------------------------------------------------------------
Et1                    2503         56681135          1045154          1029152
Et2                       8         50216275          1518179          1086297

Port           512-1023 Byte   1024-1522 Byte    1523-MAX Byte
------------------------------------------------------------
Et1                   625825         17157823          8246822
Et2                   631173         27059077          5755101
switch>
```

show interfaces counters errors

The **show interfaces counters errors** command displays the error counters for the specified interfaces.

Command Mode

EXEC

Command Syntax

show interfaces [*INTERFACE*] counters errors

Parameters

- ***INTERFACE*** Interface type and numbers. Options include:
  - <no parameter> all interfaces.
  - **ethernet** *e_range* Ethernet interface range specified by *e_range*.
  - **management** *m_range* Management interface range specified by *m_range*.
  - **port-channel** *p_range* Port-Channel Interface range specified by *p_range*.

Display Values

The table displays the following counters for each listed interface:

- FCS: Inbound packets with CRC error and proper size.
- Align: Inbound packets with improper size (undersized or oversized).
- Symbol: Inbound packets with symbol error and proper size.
- Rx: Total inbound error packets.
- Runts: Outbound packets that terminated early or dropped because of underflow.
- Giants: Outbound packets that overflowed the receiver and were dropped.
- Tx: Total outbound error packets.

Related Commands

- **show interfaces counters**
- **show interfaces counters bins**
- **show interfaces counters queue**
- **show interfaces counters rates**

Examples

- This command displays the error packet counters on Ethernet interfaces 1-2.

```
switch>show interfaces ethernet 1-2 counters errors
Port               FCS    Align   Symbol       Rx    Runts   Giants       Tx
Et1                  0        0        0        0        0        0        0
Et2                  0        0        0        0        0        0        0
switch>
```

show interfaces counters queue

The **show interfaces counters queue** command displays the queue drop counters for the specified interfaces.

Command Mode

EXEC

Command Syntax

```
show interfaces [INTERFACE] counters queue
```

Parameters

- ***INTERFACE*** Interface type and numbers. Options include:
  - <no parameter> all interfaces.
  - **ethernet** *e_range* Ethernet interface range specified by *e_range*.
  - **management** *m_range* Management interface range specified by *m_range*.
  - **port-channel** *p_range* Port-Channel Interface range specified by *p_range*.

Related Commands

- **show interfaces counters**
- **show interfaces counters bins**
- **show interfaces counters errors**
- **show interfaces counters rates**

Example

- This command displays the queue drop counters for Ethernet interfaces 1 and 2.

```
switch>show interfaces ethernet 1-2 counters queue
Port                  InDrops
Et1                       180
Et2                       169
switch>
```

show interfaces counters rates

The **show interfaces counters rates** command displays the received and transmitted packet rate counters for the specified interfaces. Counter rates provided include megabits per second (Mbps), kilopackets per second (Kpps) and utilization percentage.

All port rates are calculated approximations. Note that, when displaying rate information for a port channel, the rate value for the port channel will likely differ from the sum of the rates for the member ports. The discrepancy is likely to be larger for port channels with fewer ports, and will be most obvious in single-port port channels. The rate values for individual member ports are less inaccurate than the rate values for the port channel as a whole.

Command Mode

EXEC

Command Syntax

show interfaces [*INTERFACE*] counters rates

Parameters

- ***INTERFACE*** Interface type and numbers. Options include:
  - <no parameter> all interfaces.
  - **ethernet** *e_range* Ethernet interface range specified by *e_range*.
  - **management** *m_range* Management interface range specified by *m_range*.
  - **port-channel** *p_range* Port-Channel Interface range specified by *p_range*.

Related Commands

- **show interfaces counters**
- **show interfaces counters bins**
- **show interfaces counters errors**
- **show interfaces counters queue**

Example

- This command displays rate counters for Ethernet interfaces 1 and 2.

```
switch>show interfaces ethernet 1-2 counters rates
Port        Intvl   In Mbps      %  In Kpps  Out Mbps      % Out Kpps
Et1          0:05      53.3   0.5%        5      31.2   0.3%        2
Et2          0:05      43.3   0.4%        4       0.1   0.0%        0
switch>
```

show interfaces negotiation

The **show interfaces negotiation** command displays the speed, duplex, and flow control auto-negotiation status for the specified interfaces.

Command Mode

EXEC

Command Syntax

show interfaces [*INTERFACE*] negotiation [*INFO_LEVEL*]

Parameters

- ***INTERFACE*** Interface type and numbers. Options include:
  - <no parameter> Display information for all interfaces.
  - **ethernet** *e_range* Ethernet interface range specified by *e_range*.
  - **management** *m_range* Management interface range specified by *m_range*.

    Valid *e_range* and *m_range* formats include number, number range, or comma-delimited list of numbers and ranges.
- ***INFO_LEVEL*** amount of information that is displayed. Options include:
  - <no parameter> displays status and negotiated setting of local ports.
  - **detail** displays status and negotiated settings of local ports and their peers.

Examples

- This command displays the negotiated status of management 1 and 2 interfaces

```
switch>show interface management 1-2 negotiation
Port       Autoneg            Negotiated Settings
           Status   Speed     Duplex    Rx Pause  Tx Pause
---------  -------  --------  --------  --------  --------
Ma1        success  100M      full      off       off
Ma2        success  auto      auto      off       off
switch>
```

- This command displays the negotiated status of management 1 interface and its peer interface.

```
switch>show interface management 1 negotiation detail
Management1 :

Auto-Negotiation Mode     10/100/1000 BASE-T (IEEE Clause 28)
Auto-Negotiation Status   Success

  Advertisements       Speed           Duplex     Pause
                       --------------- ---------- --------------------
      Local            10M/100M/1G     half/full  Disabled
      Link Partner     None            None       None

  Resolution           100Mb/s         full       Rx=off,Tx=off


switch>
```

show interfaces phy

The **show interfaces phy** command displays physical layer characteristics for the specified interfaces.

Command Mode

EXEC

Command Syntax

show interfaces [*INTERFACE*] phy [*INFO_LEVEL*]

Parameters

- ***INTERFACE*** Interface type and numbers. Options include:
  - <no parameter> All interfaces.
  - **ethernet** *e_range* Ethernet interfaces in specified range.

    Valid *e_range* formats include number, number range, or comma-delimited list of numbers and ranges.
- ***INFO_LEVEL*** amount of information that is displayed. Options include:
  - <no parameter> command displays table that summarizes PHY data.
  - **detail** command displays data block for each specified interface.

Examples

- This command summarizes PHY information for Ethernet interfaces 1-5.

```
switch>show interfaces ethernet 1-5 phy
Key:
   U    = Link up
   D    = Link down
   R    = RX Fault
   T    = TX Fault
   B    = High BER
   L    = No Block Lock
   A    = No XAUI Lane Alignment
   0123 = No XAUI lane sync in lane N

                                    State    Reset
Port           PHY state        Changes    Count PMA/PMD PCS   XAUI
-------------- -------------- -------- -------- ------- ----- --------
Ethernet1      linkUp             14518     1750 U..     U.... U.......
Ethernet2      linkUp             13944     1704 U..     U.... U.......
Ethernet3      linkUp             13994     1694 U..     U.... U.......
Ethernet4      linkUp             13721     1604 U..     U.... U.......
Ethernet5      detectingXcvr          3        1               D..A0123
switch>
```

- This command displays detailed PHY information for Ethernet interface 1.

```
switch>show interfaces ethernet 1 phy detail
Current System Time: Mon Dec  5 11:32:57 2011
Ethernet1
                              Current State     Changes            Last Change
  PHY state                   linkUp              14523            0:02:01 ago
  HW resets                                        1751            0:02:07 ago
  Transceiver                 10GBASE-SRL          1704            0:02:06 ago
  Transceiver SN              C743UCZUD
  Oper speed                  10Gbps
  Interrupt Count                                 71142
  Diags mode                  normalOperation
  Model                       ael2005c
  Active uC image             microInit_mdio_SR_AEL2005C_28
  Loopback                    none
  PMA/PMD RX signal detect    ok                  11497            0:37:24 ago
  PMA/PMD RX link status      up                  11756            0:37:24 ago
  PMA/PMD RX fault            ok                  11756            0:37:24 ago
  PMA/PMD TX fault            ok                      0                  never
  PCS RX link status          up                   9859            0:02:03 ago
  PCS RX fault                ok                   9832            0:02:03 ago
  PCS TX fault                ok                    330            0:27:44 ago
  PCS block lock              ok                   9827            0:02:03 ago
  PCS high BER                ok                   8455            0:02:05 ago
  PCS err blocks              255                                  0:02:03 ago
  PCS BER                     16                  50092            0:02:05 ago
  XFI/XAUI TX link status     up                   1282            0:27:44 ago
  XFI/XAUI RX fault           ok                    585            0:27:44 ago
  XFI/XAUI TX fault           ok                   2142            0:02:05 ago
  XFI/XAUI alignment status   ok                   2929            0:02:05 ago
  XAUI lane 0-3 sync          (0123) = 1111        2932            0:02:05 ago
  XAUI sync w/o align HWM     0                                          never
  XAUI sync w/o align max OK  5
  XAUI excess sync w/o align  0                                          never
  Xcvr EEPROM read timeout                           46    4 days, 6:33:45 ago
  Spurious xcvr detection                             0                 never
  DOM control/status fail                             0
  I2C snoop reset             0
  I2C snoop reset (xcvr)      0
  Margin count                5                last > 0            0:00:00 ago
  EDC resets                  1                                    0:02:03 ago
  EDC FFE0 - FFE11            -4 -5 57 -6 -6 -2 1 0 -2 -1 1 -1
  EDC FBE1 - FBE4             6 -1 5 -1
  EDC TFBE1 - TFBE4           1 2 1 2
  EDC VGA1, VGA3              12 115
  TX path attenuation         3.0 dB
  TX preemphasis              (0,63,4) (pre,main,post)
switch>
```

show interfaces status

The **show interfaces status** command displays the interface name, link status, vlan, duplex, speed, and type of the specified interfaces. When the command includes a link status, the results are filtered to display only interfaces whose link status match the specified type.

Command Mode

EXEC

Command Syntax

```
show interfaces [INTERFACE] status [STATUS_TYPE]
```

Parameters

- ***INTERFACE*** Interface type and numbers. Options include:
  - <no parameter> All existing interfaces.
  - **ethernet** *e_range* Ethernet interfaces in the specified range.
  - **management** *m_range* Management interfaces in the specified range.
  - **port-channel** *p_range* All existing port-channel interfaces in the specified range.

    Valid *e_range*, *m_range*, and *p_range* formats include number, number range, or comma-delimited list of numbers and ranges.
- ***STATUS_TYPE*** interface status upon which the command filters output. Options include:
  - <no parameter> command does not filter on interface status.
  - **connected** interfaces connected to another port.
  - **notconnect** unconnected interfaces that are capable of connecting to another port.
  - **disabled** interfaces that have been powered down or disabled.
  - **sub-interfaces** L3 subinterfaces configured on the switch.

    Command may include multiple status types (**connected notconnect disabled**), which can be placed in any order.

Example

- This command displays the status of Ethernet interfaces 1-5.

```
switch>show interfaces ethernet 1-5 status
Port      Name              Status       Vlan        Duplex  Speed Type
Et1                         connected    1             full    10G 10GBASE-SRL
Et2                         connected    1             full    10G 10GBASE-SRL
Et3                         connected    1             full    10G 10GBASE-SRL
Et4                         connected    1             full    10G 10GBASE-SRL
Et5                         notconnect   1             full    10G Not Present
switch>
```

This command displays status information for all subinterfaces configured on the switch.

```
switch>show interfaces status sub-interfaces
Port       Name    Status       Vlan     Duplex Speed  Type                Flags
Et1.1              connect      101      full   10G    dot1q-encapsulation
Et1.2              connect      102      full   10G    dot1q-encapsulation
Et1.3              connect      103      full   10G    dot1q-encapsulation
Et1.4              connect      103      full   10G    dot1q-encapsulation
switch>
```

show interfaces status errdisabled

The **show interfaces status errdisabled** command displays interfaces that are in errdisabled state, including their link status and errdisable cause.

Command Mode

EXEC

Command Syntax

show interfaces [*INTERFACE*] status errdisabled

Parameters

- ***INTERFACE*** Interface type and numbers. Options include:
  - <no parameter> Display information for all interfaces.
  - **ethernet** *e_range* Ethernet interface range specified by *e_range*.
  - **management** *m_range* Management interface range specified by *m_range*.
  - **port-channel** *p_range* Port-Channel Interface range specified by *p_range*.

    Valid *e_range* and *m_range* formats include number, number range, or comma-delimited list of numbers and ranges.

Examples

- This command displays the error-disabled ports.

```
switch>show interfaces status errdisabled
   Port         Name             Status         Reason
------------ ---------------- ----------------- ------------------
   Et49/2                      errdisabled    multi-lane-intf
   Et49/3                      errdisabled    multi-lane-intf
   Et49/4                      errdisabled    multi-lane-intf
switch>
```

show interfaces transceiver

The **show interfaces transceiver** command displays operational transceiver data for the specified interfaces.

Command Mode

EXEC

Command Syntax

show interfaces [*INTERFACE*] transceiver [*DATA_FORMAT*]

Parameters

- ***INTERFACE*** Interface type and numbers. Options include:
  - <no parameter> all interfaces.
  - **ethernet** *e_range* Ethernet interface range specified by *e_range*.
  - **management** *m_range* Management interface range specified by *m_range*.

    Valid *e_range*, and *m_range* formats include number, number range, or comma-delimited list of numbers and ranges.
- ***DATA_FORMAT*** format used to display the data. Options include:
  - <no parameter> table entries separated by tabs.
  - **csv** table entries separated by commas.

Related Commands

- **show interfaces transceiver properties**

Examples

- This command displays transceiver data on Ethernet interfaces 1 through 4.

```
switch>show interfaces ethernet 1-4 transceiver
If device is externally calibrated, only calibrated values are printed.
N/A: not applicable, Tx: transmit, Rx: receive.
mA: milliamperes, dBm: decibels (milliwatts).
                              Bias      Optical   Optical
        Temp       Voltage    Current   Tx Power  Rx Power  Last Update
Port    (Celsius)  (Volts)    (mA)      (dBm)     (dBm)     (Date Time)
-----   ---------  --------   --------  --------  --------  -------------------
Et1      34.17      3.30       6.75     -2.41     -2.83     2011-12-02 16:18:48
Et2      35.08      3.30       6.75     -2.23     -2.06     2011-12-02 16:18:42
Et3      36.72      3.30       7.20     -2.02     -2.14     2011-12-02 16:18:49
Et4      35.91      3.30       6.92     -2.20     -2.23     2011-12-02 16:18:45
switch>
```

show interfaces transceiver channels

The **show interfaces transceiver channels** command displays current wavelength/frequency settings for the specified channels.

Command Mode

EXEC

Command Syntax

```
show interfaces [INTERFACE e_range] transceiver channels
```

Parameters

- ***INTERFACE*** Interface type and port numbers.
  - **ethernet** *e_range* Ethernet interface range specified by *e_range*.

Related Commands

- **transceiver channel**
- **show interfaces transceiver hardware**

Examples

- This command displays the supported wavelengths/frequencies and their corresponding channel numbers on Ethernet interface 4 to slot 3 through 4.

```
switch(config-as-if-Et4/1/3)#show interfaces ethernet 4 / 3 / 4 transceiver
channels
 Name: Et4/3/4
 100GHz- 50GHz-
 Wavelength   Frequency spacing spacing
 (nm)         (GHz)     Channel Channel
 ----------   --------- ------- -------
 1567.95      191,200         1       1
 1567.54      191,250                 2
 1567.13      191,300         2       3
 1566.72      191,350                 4
 ....
 1529.16      196,050                98
 1528.77      196,100        50      99
 1528.38      196,150               100
switch(config-as-if-Et4/1/3)#
```

show interfaces transceiver hardware

The **show interfaces transceiver hardware** command displays current wavelength/frequency settings for the specified transceiver interfaces.

Command Mode

EXEC

Command Syntax

show interfaces [*INTERFACE e_range*] transceiver hardware

Parameters

- ***INTERFACE*** Interface type and port numbers.
  - **ethernet** *e_range* Ethernet interface range specified by *e_range*.

Related Commands

- **transceiver channel**
- **show interfaces transceiver channels**

Examples

- This command displays the current wavelength/frequency settings on Ethernet interface 4 to slot 3 through 4.

```
switch(config-as-if-Et4/1/3)#show interfaces ethernet 4 / 3 / 4 transceiver
hardware
 Name: Et4/3/4
 Media Type: 10GBASE-DWDM
 Configured Channel : 39
 Configured Grid (GHz) : 50
 Computed Frequency (GHz) : 193,100
 Computed Wavelength (nm) : 1552.52
 Operational Channel : 39 (Default)
 Operational Grid (GHz) : 50 (Default)
 Operational Frequency (GHz): 193,100
 Operational Wavelength (nm): 1552.52
switch(config-as-if-Et4/1/3)#
```

show interfaces transceiver properties

The **show interfaces transceiver properties** command displays configuration information for the specified interfaces. Information provided by the command includes the media type, interface speed-duplex settings, speed-duplex operating state.

Command Mode

EXEC

Command Syntax

show interfaces [*INTERFACE*] transceiver properties

Parameters

- ***INTERFACE*** Interface type and numbers. Options include:
  - <no parameter> Display information for all interfaces.
  - **ethernet** *e_range* Ethernet interface range specified by *e_range*.
  - **management** *m_range* Management interface range specified by *m_range*.

    Valid *e_range* and *m_range* formats include number, number range, or comma-delimited list of numbers and ranges.

Related Commands

- **show interfaces transceiver**

Examples

- This command displays the media type, speed, and duplex properties for Ethernet interfaces 1-3.

```
switch>show interfaces ethernet 1-3 transceiver properties
Name : Et1
Administrative Speed: 10G
Administrative Duplex: full
Operational Speed: 10G (forced)
Operational Duplex: full (forced)
Media Type: 10GBASE-SRL

Name : Et2
Administrative Speed: 10G
Administrative Duplex: full
Operational Speed: 10G (forced)
Operational Duplex: full (forced)
Media Type: 10GBASE-SRL

Name : Et3
Administrative Speed: 10G
Administrative Duplex: full
Operational Speed: 10G (forced)
Operational Duplex: full (forced)
Media Type: 10GBASE-SRL

switch>
```

show platform fm6000 agileport map

The **show platform fm6000 agileport map** command displays the list of Ethernet interfaces that are combinable to form a higher speed port.

Command Mode

Privileged EXEC

Command Syntax

```
show platform fm6000 agileport map
```

Example

- These commands displays the agile port map for the switch, then configures Ethernet interface 13 as a 40G port, subsuming Ethernet interfaces 15, 17 and 19.

```
switch#show platform fm6000 agileport map

-----------------------------------------------------------------
Agile Ports       |        Interfaces subsumed in 40G link
-----------------------------------------------------------------
Ethernet1         |  Ethernet3      Ethernet5      Ethernet7
Ethernet2         |  Ethernet4      Ethernet6      Ethernet8
Ethernet13        |  Ethernet15     Ethernet17     Ethernet19
Ethernet14        |  Ethernet16     Ethernet18     Ethernet20
switch#config
switch(config)#interface ethernet 13
switch(config-if-Et13)#speed forced 40gfull

  WARNING!  Executing this command will cause the forwarding agent
            to be restarted. All interfaces will briefly drop links
            and forwarding on all interfaces will momentarily stop.


            Do you wish to proceed with this command? [y/N]

Ethernet13 configured for 40G.
Ethernet15, Ethernet17 and Ethernet19 are now subsumed.
switch(config-if-Et13)#
```

speed

The **speed** command configures the transmission speed and duplex setting for the configuration mode interface. The scope and effect of this command depends on the interface type. Interface types include:

- 40GBASE (QSFP+): Default is 4x10G-full. **Speed forced 40gfull** and **Speed auto 40gfull** configure interface as a 40G port.
- 10GBASE-T: Default is 10G-full. **Speed** command affects interface.
- 10GBASE (SFP+): Default is 10G-full. **Speed** command does not affect interface.
- 1000BASE (copper): Default is 1G-full. **speed auto 100full** affects interface.
- 1000BASE (fiber): Default is 1G-full. **Speed** command does not affect interface.
- 10/100/1000: Default is ***auto-negotiation***. **Speed** command (10/100/1000 options) affects interface.

The **speed forced 40gfull** and **auto 40gfull** commands configure a QSFP+ Ethernet interface as a 40G port. The **no speed** and **no auto 40gfull** commands configure a QSFP+ Ethernet interface as four 10G ports. These commands must be applied to the /1 port. These commands are hitless on the 7050X, 7060X, 7250X, 7260X, 7280SE, 7300X, 7320X and 7500E series platforms. On all other platforms, these commands restart the forwarding agent, which will result in traffic disruption.

The **no speed** and **default speed** commands restore the default setting for the configuration mode interface by removing the corresponding **speed** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration
Interface-Management Configuration

Command Syntax

```
speed MODE
no speed
default speed
```

Parameters

- ***MODE*** transmission speed and duplex setting. Options include:
  - **auto** auto negotiation mode.
  - **auto 40gfull** auto negotiation mode with clause 73 auto negotiation.

Important! Interfaces using clause 73 auto negotiation must connect to a device that runs clause 73 auto negotiation.

  - **sfp-1000baset auto** auto-negotiation mode (1000BASE-T interfaces only).
  - **forced 10000full** 10G full duplex.
  - **forced 1000full** 1G full duplex.
  - **forced 1000half** 1G half duplex.
  - **forced 100full** 100M full duplex.
  - **forced 100gfull** 100G full duplex.
  - **forced 100half** 100M half duplex.
  - **forced 10full** 10M full duplex.
  - **forced 10half** 10M half duplex.
  - **forced 40gfull** 40G full duplex.

On 40GBASE and 100GBASE interfaces, options that change the SFP+ and MXP interfaces (the **auto 40gfull**, the **forced 40gfull,** and the **no speed** options) may restart the forwarding agent on some switch platforms, disrupting traffic on all ports for more than a minute.

Examples

- This command configures a 40GBASE interface as a 40G port.

```
switch(config)#interface ethernet 49/1
switch(config-if-Et49/1)#speed forced 40gfull
switch(config-if-Et49/1)#show interface ethernet 49/1 - 49/4 status
Port      Name          Status       Vlan        Duplex  Speed Type
Et49/1                  connected    in Po999      full    40G 40GBASE-CR4
Et49/2                  errdisabled  inactive    unconf unconf 40GBASE-CR4
Et49/3                  errdisabled  inactive    unconf unconf 40GBASE-CR4
Et49/4                  errdisabled  inactive    unconf unconf 40GBASE-CR4
switch(config-if-Et49/1)#
```

- This command configures a 40GBASE interface as four 10G ports (default configuration).

```
switch(config-if-Et49/1)#no speed
switch(config-if-Et49/1)#show interface ethernet 49/1 - 49/4 status
Port      Name          Status       Vlan        Duplex  Speed Type
Et49/1                  connected    routed        full    10G 40GBASE-SR4
Et49/2                  connected    routed        full    10G 40GBASE-SR4
Et49/3                  connected    routed        full    10G 40GBASE-SR4
Et49/4                  notconnect   inactive      full    10G 40GBASE-SR4
switch(config-if-Et49/1)#
```

transceiver qsfp default-mode

The **transceiver qsfp default-mode** command specifies the transmission mode of all QSFP transceiver modules that are not explicitly configured.

Each QSFP+ module Ethernet interface is configurable as a single 40G port or as four 10G ports. The switch displays four ports for each interface. Each port's status depends on the interface configuration:

- The /1 port is active (***connected*** or ***not connected***), regardless of the interface configuration.
- The /2, /3, and /4 ports are ***error-disabled*** when the interface is configured as a single 40G port.
- all ports are active (***connected*** or ***not connected***), when the interface is configured as four 10G ports.

The only available default-mode value is 4x10G; QSFP modules that are not configured through a **speed** command are operated as four 10G ports.

The **no transceiver qsfp default-mode** and **default transceiver qsfp default-mode** commands restore the default-mode transceiver setting to its default value of 4x10G.

Command Mode

Global Configuration

Command Syntax

```
transceiver qsfp default-mode 4x10G
no transceiver qsfp default-mode
default transceiver qsfp default-mode
```

Guidelines

The **transceiver qsfp default-mode 4x10g** statement is always in ***running-config*** and cannot be modified or removed in the current release.

transceiver channel

The **transceiver channel** command displays transceiver wavelength/frequency by channel number. The channel numbering depends on the selected grid-spacing mode. The default grid-spacing mode is 50GHz-spacing.

- If the startup configuration does not specify the channel number for the interface, the transceiver will automatically tune to the default channel (i.e. channel-39 of 50GHz-spacing grid) when it is inserted.
- If the configured wavelength/frequency is not supported by the transceiver, the transceiver will be tuned to the default channel (i.e. channel-39 of 50GHz-spacing grid).

The interface is shutdown before the channel number is configured.

Command Mode

Global Configuration

Command Syntax

```
transceiver channel CHANNEL_NUMBER grid-spacing <SPACING_GRID>
no transceiver channel CHANNEL_NUMBER [GRID_SPACING <SPACING_GRID>
default transceiver channel CHANNEL_NUMBER [GRID_SPACING <SPACING_GRID>
```

Parameters

- ***CHANNEL-NUMBER*** The default channel is 39 (50GHz-spacing grid) which corresponds to a frequency of 193,100 GHz and a wavelength of 1552.52 nm.
- ***GRID_SPACING*** Grid-spacing mode (optional) depends on the selected grid-spacing mode. The default grid-spacing mode is 50GHz-spacing. For example, channel 39 of 50GHz-spacing grid is equivalent to channel 20 of 100GHz-spacing grid, which corresponds to a frequency of 193,100 GHz and a wavelength of 1552.52 nm.
  - <SPACING_GRID> default grid-spacing mode in GHz.

Related Commands

- **show interfaces transceiver channels**
- **show interfaces transceiver hardware**

Example

- This command tunes the transceiver on slot number 4 to slot 1 through 3 of 50GHz-spacing grid.

```
switch(config-as)#interface ethernet 4 / 1 / 3
switch(config-if-Et4/1/3)#transceiver channel 1 grid-spacing 50
switch(config-if-Et4/1/3)#
```

Chapter 11

Port Channels and LACP

This chapter describes channel groups, port channels, port channel interfaces, and the Link Aggregation Control Protocol (LACP). This chapter contains the following sections:

- Section 11.1: Port Channel Introduction
- Section 11.2: Port Channel Conceptual Overview
- Section 11.3: Port Channel Configuration Procedures
- Section 11.4: Load Balancing Hash Algorithms
- Section 11.5: Port Channel and LACP Configuration Commands

11.1 Port Channel Introduction

Arista's switching platforms support industry standard link aggregation protocols. Arista switches optimize traffic throughput by using MAC, IP addressing and services fields to effectively load share traffic across aggregated links. Managers can configure multiple ports into a logical port channel, either statically or dynamically through the IEEE Link Aggregation Control Protocol (LACP). Various negotiation modes are supported to accommodate any variety of configurations or peripheral requirements, including LACP fallback to support devices that need simple network connectivity to retrieve images or configurations prior to engaging port channel aggregation modes.

Arista's Multi-chassis Link Aggregation protocol (MLAG) supports LAGs across paired Arista switches to provide both link aggregation and active/active redundancy.

11.2 Port Channel Conceptual Overview

11.2.1 Channel Groups and Port Channels

A port channel is a communication link between two switches that consists of matching channel group interfaces on each switch. A port channel is also referred to as a Link Aggregation Group (LAG). Port channels combine the bandwidth of multiple Ethernet ports into a single logical link.

A channel group is a collection of Ethernet interfaces on a single switch. A port channel interface is a virtual interface that consists of a corresponding channel group and connects to a compatible interface on another switch to form a port channel. Port channel interfaces can be configured and used in a manner similar to Ethernet interfaces. Port channel interfaces are configurable as layer 2 interfaces, layer 3 (routable) interfaces, and VLAN members. Most Ethernet interface configuration options are available to port channel interfaces.

11.2.2 Port Channel Subinterfaces

Port channel subinterfaces divide a single port channel interface into multiple logical L3 interfaces based on the 802.1q tag (VLAN ID) of incoming traffic. Subinterfaces are commonly used in the L2/L3 boundary device, but they can also be used to isolate traffic with 802.1q tags between L3 peers by assigning each subinterface to a different VRF.

For further details about subinterfaces, see Subinterfaces.

11.2.3 Link Aggregation Control Protocol (LACP)

The Link Aggregation Control Protocol (LACP), described by IEEE 802.3ad, defines a method for two switches to automatically establish and maintain LAGs. When LACP is enabled, a switch can configure LACP-compatible ports into a LAG (also called a channel group); the maximum number of ports per LAG varies by platform (numbers for each platform in the latest EOS release are available here: https://www.arista.com/en/support/product-documentation/supported-features).

LACP terminology refers to the local interface as the ***actor*** and the remote interface as the ***partner***.

- In static mode, switches create port channels without awareness of their partner's port channels. Packets may drop when port channel static aggregate configurations differ between switches.

  The switch aggregates static links without LACP negotiation. The switches do not send LACP packets nor process inbound LACP packets.

- In dynamic mode, Link Aggregation Groups are aware of their partners' port channel states. Interfaces configured as dynamic LAGs are designated as ***active*** or ***passive***.
  - Active interfaces send LACP Protocol Data Units (LACP PDUs) at a rate of one per second when forming a channel with an interface on the peer switch. An aggregate forms if the peer runs LACP in active or passive mode.
  - Passive interfaces only send LACP PDUs in response to PDUs received from the partner. The partner switch must be in active mode and initiates negotiation by sending an LACP packet. The passive mode switch receives and responds to the packet to form a LAG.

An ***active*** interface can form port channels with ***passive*** or ***active*** partner interfaces. Port channels are not formed when the interface on each switch is ***passive***. Table 11-1 summarizes the valid LACP mode combinations:

Table 11-1 Valid LACP Mode Combinations

Switch 1	Switch 2	Comments
active	active	Links aggregate when LACP negotiation is successful.
active	passive	Links aggregate when LACP negotiation is successful.
passive	passive	Links aggregate without LACP.
on	—	Links aggregate without LACP.

During synchronization, interfaces transmit one LACP PDU per second. After synchronization is complete, interfaces exchange one PDU every thirty seconds, facilitated by a default timeout of 30 seconds and a failure tolerance of three. Under these parameters, when the switch does not receive an LACP PDU for an interface during a ninety second period, it records the partner interface as ***failed*** and removes the interface from the port channel.

Fallback mode allows an active LACP interface to maintain a LAG without receiving PDUs from its peer. (An active interface that is not in fallback mode does not form a LAG until it receives PDUs from and negotiates with its peer.) The fallback timer specifies the period the LAG waits to receive a peer PDU. Upon timer expiry, the port channel reverts to its configured fallback mode if one is configured.

Static fallback: the port channel maintains one active port while in fallback mode; all its other member ports are in standby mode until a LACP PDU is received by the port channel. All member ports send (and can receive) LACP PDUs, but only the active port sends or receives data.

Individual fallback: all member ports act as individual switch ports while in fallback mode. Individual port configuration (rather than port channel configuration) is active while the port channel is in fallback mode, with the exception of ACLs. This includes VLAN membership. All member ports send and receive data, and continue to send LACP PDUs. As soon as a LACP PDU is received by a member of the port channel, all ports revert to normal port channel operation.

The switch uses a link aggregation hash algorithm to determine the forwarding path within a Link Aggregation Group. The IP and MAC header fields can be selected as components of the hash algorithm.

11.3 Port Channel Configuration Procedures

These sections describe channel group and port channel configuration procedures:

- Section 11.3.1: Configuring a Channel Group
- Section 11.3.2: Configuring a Port Channel Interface
- Section 11.3.4: Configuring LACP

11.3.1 Configuring a Channel Group

Creating a Channel Group

The **channel-group** command assigns the configuration mode Ethernet interfaces to a channel group and specifies LACP attributes for the channel.

Channel groups are associated with a port channel interface immediately upon their creation. A command that creates a new channel group also creates a port channel with a matching ID. The port channel is configured in port-channel configuration mode. Configuration changes to a port channel interface propagate to all Ethernet interfaces in the corresponding channel group.

Example

- These commands assign Ethernet interfaces 1 and 2 to channel group 10, enable LACP, and place the channel group in a negotiating state:

```
switch(config)#interface ethernet 1-2
switch(config-if-Et1-2)#channel-group 10 mode active
switch(config-if-Et1-2)#
```

Adding an Interface to a Channel Group

The **channel-group** command adds the configuration mode interface to the specified channel group if the channel group exists. When adding channels to a previously created channel group, the LACP mode for the new channel must match the mode for the existing group.

Example

- These commands add Ethernet interfaces 7 through 10 to previously created channel group 10, using the LACP mode under which it was created.

```
switch(config)#interface ethernet 7-10
switch(config-if-Et7-10)#channel-group 10 mode active
switch(config-if-Et7-10)#
```

Removing an Interface from a Channel Group

The **no channel-group** command removes the configuration mode interface from the specified channel group. Deleting all members of a channel group does not remove the associated port channel interface from ***running-config***.

Example

- These commands remove add Ethernet interface 8 from previously created channel group 10.

```
switch(config)#interface ethernet 8
switch(config-if-Et8)#no channel-group
switch(config-if-Et7-10)#
```

Deleting a Channel Group

A channel group is deleted by removing all Ethernet interfaces from the channel group. A channel group's LACP mode can be changed only be deleting the channel group and then creating an equivalent group with a different LACP mode. Deleting a channel group by removing all Ethernet interfaces from the group preserves the port channel interface and its configuration settings.

View ***running-config*** to verify the deletion of all Ethernet interfaces from a channel group.

11.3.2 Configuring a Port Channel Interface

Creating a Port Channel Interface

The switch provides two methods for creating port channel interfaces:

- creating a channel group simultaneously creates an associated port channel.
- the **interface port-channel** command creates a port channel without assigning Ethernet channels to the new interface.

The **interface port-channel** command places the switch in interface-port channel configuration mode.

Example

- This command creates port channel interface 8 and places the switch in port channel interface configuration mode.

```
switch(config)#interface port-channel 8
switch(config-if-Po8)#
```

Deleting a Port Channel Interface

The **no interface port-channel** command deletes the configuration mode port channel interface and removes the channel group assignment for each Ethernet channel assigned to the channel associated with the port channel. Removing all Ethernet interfaces from a channel group does not remove the associated port channel interface from ***running-config***.

11.3.3 Configuring Port Channel Subinterfaces

When configuring subinterfaces on a port channel interface (the virtual interface associated with a port channel), the following restrictions apply:

- An L3 interface with subinterfaces configured on it should not be made a member of a port channel.
- An interface that is a member of a port channel should not have subinterfaces configured on it.
- A subinterface cannot be made a made of a port channel.

Port channel subinterfaces are otherwise configured similarly to Ethernet subinterfaces. For additional information, see Subinterfaces.

11.3.4 Configuring LACP

Configuring the LACP Mode

The LACP mode is configured when a channel group is created. A channel group's LACP mode cannot be modified without deleting the entire channel group. A channel group's LACP mode can be altered without deleting the port channel interface associated with the channel group.

Example

- These commands create a channel group and place it in LACP-active mode.

```
switch(config)#interface ethernet 1-2
switch(config-if-Et1-2)#channel-group 10 mode active
switch(config-if-Et1-2)#
```

Configuring the System Priority

Each switch is assigned a globally unique system identifier by concatenating the system priority (16 bits) to the MAC address of one of its physical ports (48 bits). The system identifier is used by peer devices when forming an aggregation to verify that all links are from the same switch. The system identifier is also used when dynamically changing aggregation capabilities in response to LACP information; the system with the numerically lower system identifier is permitted to dynamically change advertised aggregation capabilities.

The **lacp system-priority** command configures the switch's LACP system priority.

Example

- This command assigns the system priority of 8192 to the switch.

```
switch(config)#lacp system-priority 8192
switch(config)#
```

Configuring Port Priority

LACP port priority determines the port that is active in a LAG in fallback mode. Numerically lower values have higher priority. Priority is supported on port channels with LACP-enabled physical interfaces.

The **lacp port-priority** command sets the aggregating port priority for the configuration mode interface.

Example

- This command assigns the port priority of 4096 to Ethernet interface 1.

```
switch(config-if-Et1)#lacp port-priority 4096
switch(config-if-Et1)#
```

Configuring the LACP Packet Transmission Rate

The LACP transmission interval sets the rate for LACP control packets. Supported values include

- ***normal***: 30 seconds on synchronized interfaces; one second on interfaces that are synchronizing.
- ***fast***: one second.

The **lacp rate** command configures the LACP transmission interval on the configuration mode interface.

Example

- This command sets the LACP rate to one second on Ethernet interface 4.

```
switch(config-if-Et4)#lacp rate fast
switch(config-if-Et4)#
```

Configuring LACP Fallback

Fallback mode (static or individual) is configured on a port channel interface with the **port-channel lacp fallback** command. The fallback timeout interval is configured with the **port-channel lacp fallback timeout** command. Fallback timeout settings persist in ***running-config*** without taking effect for interfaces that are not configured into fallback mode. The default fallback timeout period is 90 seconds.

Examples

- These commands enable LACP static fallback mode, then configure an LACP fallback timeout of 100 seconds on port channel interface 13. If LACP negotiation fails, only the member port with the lowest LACP priority will remain active until an LACP PDU is received by one of the member ports.

```
switch(config)#interface port-channel 13
switch(config-if-Po13)#port-channel lacp fallback static
switch(config-if-Po13)#port-channel lacp fallback timeout 100
switch(config-if-Po13)#show active
interface Port-Channel13
   port-channel lacp fallback static
   port-channel lacp fallback timeout 100
switch(config-if-Po13)#
```

- These commands enable LACP individual fallback mode, then configure an LACP fallback timeout of 50 seconds on port channel interface 17. If LACP negotiation fails, all member ports will act as individual switch ports, using port-specific configuration, until a LACP PDU is received by one of the member ports.

```
switch(config)#interface port-channel 17
switch(config-if-Po17)#port-channel lacp fallback individual
switch(config-if-Po17)#port-channel lacp fallback timeout 50
switch(config-if-Po17)#show active
interface Port-Channel17
   port-channel lacp fallback individual
   port-channel lacp fallback timeout 50
switch(config-if-Po17)#
```

Configuring Minimum Links

The **port-channel min-links** command specifies the minimum number of interfaces that the configuration mode LAG requires to be active. If there are fewer ports than specified by this command, the port channel interface does not become active.

Note In static LAGs, the min-links value must be met for the LAG to be active. The LAG will not become active until it has at least the min-links number of functioning links in the channel group. If failed links cause the number to drop below the minimum, the LAG will go down and administrator action will be required to bring it back up.

In dynamic LAGs, the LACP protocol must determine that at least min-links physical ports are aggregable (they are physically compatible and have the same keys both remotely and locally) before it begins negotiating to make any ports active members of the port-channel. Once negotiation begins, however, an error on the partner's side can cause the LAG to become active with fewer than the minimum number of links.

Example

- This command sets four as the minimum number of ports required for port channel 5 to become active.

```
switch(config-if-Po5)#port-channel min-links 4
switch(config-if-Po5)#
```

11.3.5 Displaying Port Channel Information

Port channel information is accessed using some of the **show** commands listed under Interface Display Commands. Note that when using the **show interfaces counters rates** command to display rate information for a port channel, rate values for the individual member ports are less inaccurate than rate values for the port channel.

Both the port channel rate and the individual port rates are calculated approximations; the rate value for a port channel will likely differ from the sum of the rates for the member ports. The discrepancy is likely to be larger for port channels with fewer ports, and will be most obvious in single-port port channels.

11.4 Load Balancing Hash Algorithms

The switch balances packet load across multiple links in a port channel by calculating a hash value based on packet header fields. The hash value determines the active member link through which the packet is transmitted. This method, in addition to balancing the load in the LAG, ensures that all packets in a data stream follow the same network path.

In network topologies that include MLAGs or multiple paths with equal cost (ECMP), programming all switches to perform the same hash calculation increases the risk of hash polarization, which leads to uneven load distribution among LAG and MLAG member links. This uneven distribution is avoided by performing different hash calculations on each switch routing the paths.

The **port-channel load-balance** command specifies the seed for hashing algorithms that balance the load across ports comprising a port channel. Available seed values vary by switch platform.

Example

- This command configures the hash seed of 10 on 7150 Series (FM6000 platform) switches.

```
switch(config)#port-channel load-balance fm6000 10
switch(config)#
```

Hashing algorithm inputs varies by switch platform. These sections describe hashing algorithm inputs for each platform.

- Section 11.4.1: Load Balance Hash Algorithms on 7048 and 7500 Series Switches
- Section 11.4.2: Load Balance Hash Algorithms on 7500E Series Switches
- Section 11.4.3: Load Balance Hash Algorithms on 7050 Series Switches
- Section 11.4.4: Load Balance Hash Algorithms on 7150 Series Switches

11.4.1 Load Balance Hash Algorithms on 7048 and 7500 Series Switches

One command configures the load balance hash algorithm on 7048 and 7500 Series switches:

- **port-channel load-balance petraA fields ip**: controls the hash algorithm for IP packets by specifying the algorithm's use of IP and MAC header fields. Fields that the command can specify include source and destination IP addresses, source and destination port fields (for TCP and UDP packets), and the entire MAC address header.

The hash algorithm for non-IP packets is not configurable and always includes the entire MAC header.

Example

- These commands configure the load balance algorithm for IP packets by using the entire MAC header.

```
switch(config)#port-channel load-balance petraA fields ip mac-header
switch(config)#
```

11.4.2 Load Balance Hash Algorithms on 7500E Series Switches

One command configures the load balance hash algorithm on 7500E Series switches:

- **port-channel load-balance arad fields ip**: controls the hash algorithm for IP packets by specifying the algorithm's use of IP and MAC header fields. Fields that the command can specify include source and destination IP addresses, source and destination port fields (for TCP and UDP packets), and the entire MAC address header.

The hash algorithm for non-IP packets is not configurable and always includes the entire MAC header.

Example

- These commands configure the load balance algorithm for IP packets by using the entire MAC header.

```
switch(config)#port-channel load-balance arad fields ip mac-header
switch(config)#
```

11.4.2.1 Dynamic and Symmetric LAG Hashing

Dynamic LAG hashing enables high link utilization and highly even distribution among LAG members by employing a randomized hashing algorithm. Symmetric LAG hashing allows the two flows of a bidirectional communication link, even when the two flows enter the switch on different ingress ports, to be hashed to the same member of a LAG on egress.

Dynamic and symmetric LAG hashing policies are enabled via named port-channel load-balancing profiles. LAG load-balancing policies can be provisioned on per linecard basis using these profiles. Load-balancing profiles can be used to provision all LAG load-balance attributes, including hash polynomials, hash seeds, and hash fields.

Examples

- These commands configure a load balance profile for symmetric hashing.

```
switch(config)#load-balance policies
switch(config-load-balance-policies)#load-balance arad profile
symmetric-profile-1
switch(config-sand-load-balance-profile-symmetric-profile-1)#hash symmetric
switch(config-sand-load-balance-profile-symmetric-profile-1)#show active
load-balance policies
   load-balance arad profile symmetric-profile-1
      hash symmetric
```

- These commands configure a load balance profile for dynamic hashing.

```
switch(config)#load-balance policies
switch(config-load-balance-policies)#load-balance arad profile
dynamic-hash-profile-1
switch(config-sand-load-balance-profile-dynamic-hash-profile-1)#distribution
clock
switch(config-sand-load-balance-profile-dynamic-hash-profile-1)#show active
load-balance policies
   load-balance arad profile dynamic-hash-profile-1
      distribution clock
```

- This command assigns a named load-balancing profile to a linecard.

```
switch(config)#port-channel load-balance module 3-7 sand profile Linecard5
switch(config)#
```

- This command unassigns a named load-balancing profile to a linecard.

```
switch(config)#no port-channel load-balance module 3-7 sand profile Linecard5
switch(config)#
```

11.4.3 Load Balance Hash Algorithms on 7050 Series Switches

Three commands configure the load balance hash algorithm on 7050 Series switches:

- **port-channel load-balance trident fields ip** controls the hash algorithm for IP packets by specifying the algorithm's use of IP and MAC header fields. Fields that the command can specify include source and destination IP addresses, source and destination port fields (for TCP and UDP packets), and fields specified by the **port-channel load-balance trident fields mac** command.

- **port-channel load-balance trident fields ipv6** controls the hash algorithm for IPv6 packets by specifying the algorithm's use of IP and MAC header fields. Fields that the command can specify include source and destination IP addresses, source and destination port fields (for TCP and UDP packets), and fields specified by the **port-channel load-balance trident fields mac** command.
- **port-channel load-balance trident fields mac** controls the hash algorithm for non-IP packets b specifying the algorithm's use of MAC header fields. Fields that the command can specify include include the MAC source address, MAC destination address, and Ethernet type fields.

Example

- These commands configure the switch's port channel load balance for non IP packets by using the MAC destination and Ethernet type fields in the hashing algorithm.

```
switch(config)#port-channel load-balance trident fields mac dst-mac eth-type
switch(config)#
```

11.4.4 Load Balance Hash Algorithms on 7150 Series Switches

Load balance profiles specify parameters used by hashing algorithms that distribute traffic across ports comprising a port channel or among component ECMP routes. The switch supports 16 load balance profiles, including the default profile. The default load balance profile is configured through **port-channel load-balance fm6000 fields ip** and **port-channel load-balance fm6000 fields mac** commands.

11.4.4.1 Load Balance Profiles

Load balance profiles are managed in load-balance-policies configuration mode. Load-balance-policies mode provides commands that display the contents of all configured profiles and place the switch in load-balance-profile command. Load balance profiles are created by entering load-balance-profile mode and edited while in that mode.

The **load-balance policies** command places the switch in load-balance-policies configuration mode. Load balance profiles specify the inputs used by the hashing algorithms that distribute traffic across ports comprising a port channel or among ECMP routes.

Example

- This command places the switch in load-balance-policies configuration mode.

```
switch(config)#load-balance policies
switch(config-load-balance-policies)#
```

- This command displays the contents of the four load balance profiles configured on the switch.

```
switch(config-load-balance-policies)#show active
load-balance policies
   load-balance fm6000 profile F-01
      port-channel hash-seed 22
      fields ip dscp
      distribution random port-channel
   !
   load-balance fm6000 profile F-02
      fields ip protocol dst-ip
      distribution random port-channel
   !
   load-balance fm6000 profile F-03
      fields ip protocol dst-ip
      fields mac dst-mac eth-type
      distribution random ecmp port-channel
   !
   load-balance fm6000 profile F-04
switch(config-load-balance-policies)#
```

Creating a Load Balance Profile

The **load-balance fm6000 profile** command places the switch in load-balance-profile configuration mode to configure a specified load balance profile. The command specifies the name of the profile that subsequent commands modify. It creates a profile if the profile it references does not exist.

Example

- These commands enter load-balance-profile configuration mode, creates the LB-5 profile, and lists the default settings for the profile.

```
switch(config)#load-balance policies
switch(config-load-balance-policies)#load-balance fm6000 profile LB-5
switch(config-load-balance-profile-LB-5)#show active all
load-balance policies
   load-balance fm6000 profile LB-5
      port-channel hash-seed 0
      fields mac dst-mac src-mac eth-type vlan-priority vlan-id
      fields ip protocol dst-ip dst-port src-ip src-port dscp
      no distribution symmetric-hash
      no distribution random
switch(config-load-balance-profile-LB-5)#
```

Configuring a Load Balance Profile

These commands are available in load-balance-profile configuration mode to specify the parameters that comprise a profile.

- The **fields ip** command specifies the L3/L4 data fields used by the hash algorithm defined by the configuration mode load balance profile.
- The **fields mac** command specifies the L2 data fields used by the hash algorithm defined by the configuration mode load balance profile.
- The **distribution symmetric-hash** command enforces traffic symmetry on data distributed by the hash algorithm defined by the configuration mode load balance profile. Symmetric traffic is the flow of both directions of a data stream across the same physical link.
- The **distribution random** command specifies the random distribution of data packets handled by the hash algorithm defined by the configuration mode load balance profile.

Example

- These commands configure the following components of the hash algorithm defined by the LB-7 load balance profile:
  - L2 header fields: MAC destination address, VLAN priority
  - L3/L4 header fields: Source IP address, protocol field
  - Symmetric hash distribution of IP and non-IP packets.

```
switch(config)#load-balance policies
switch(config-load-balance-policies)#load-balance fm6000 profile LB-7
switch(config-load-balance-profile-LB-7)#fields ip src-ip protocol
switch(config-load-balance-profile-LB-7)#fields mac dst-mac vlan-priority
switch(config-load-balance-profile-LB-7)#distribution symmetric-hash mac-ip
switch(config-load-balance-profile-LB-7)#show active
load-balance policies
   load-balance fm6000 profile LB-7
      fields mac dst-mac vlan-priority
      fields ip protocol src-ip
      distribution symmetric-hash mac-ip
switch(config-load-balance-profile-LB-7)#exit
switch(config-load-balance-policies)#exit
switch(config)#exit
```

Assigning a Load Balance Profile to an Interface

The **ingress load-balance profile** command applies a specified load-balance profile to the configuration mode interface. Load balance profiles specify parameters used by hashing algorithms that distribute traffic across ports comprising a port channel or among ECMP routes. The switch supports 16 load balance profiles, including the default profile.

Example

- This command applies the ***LB-1*** load balance profile to port channel interface 100.

```
switch(config)#interface port-channel 100
switch(config-if-Po100)#ingress load-balance profile LB-1
switch(config-if-Po100)#show active
interface Port-Channel100
   ingress load-balance profile LB-1
switch(config-if-Po100)#
```

11.4.4.2 Default Load Balance Profile

Two commands configure the load balance default profile on 7150 Series switches:

- **port-channel load-balance fm6000 fields ip** controls the hash algorithm for IP packets by specifying the algorithm's use of IP and MAC header fields. Fields that the command can specify include source and destination IP addresses, source and destination port fields (for TCP and UDP packets).
- **port-channel load-balance fm6000 fields mac** controls the hash algorithm for non-IP packets by specifying the algorithm's use of MAC header fields. Fields that the command can specify include include the MAC source address, MAC destination address, and Ethernet type, VLAN-ID, and VLAN-priority fields.

Example

- These commands configure the load balance default profile for IP packets by using source and destination IP address fields, along with source and destination port fields for TCP, and UDP packets.

```
switch(config)#port-channel load-balance fm6000 fields ip ip-tcp-udp-header
switch(config)#
```

- This command applies the default load balance profile to port channel interface 100.

```
switch(config)#interface port-channel 100
switch(config-if-Po100)#no ingress load-balance profile
switch(config-if-Po100)#show active
interface Port-Channel100
switch(config-if-Po100)#
```

11.5 Port Channel and LACP Configuration Commands

Global Port Channel and LACP Configuration Commands

- interface port-channel
- lacp system-priority

Interface Configuration Commands – Ethernet Interface

- channel-group
- lacp port-priority
- lacp rate
- port-channel lacp fallback
- port-channel lacp fallback timeout
- port-channel min-links

Load Balance (Default) Commands

- port-channel load-balance
- port-channel load-balance arad fields ip
- port-channel load-balance fm6000 fields ip
- port-channel load-balance fm6000 fields mac
- port-channel load-balance module
- port-channel load-balance petraA fields ip
- port-channel load-balance trident fields ip
- port-channel load-balance trident fields ipv6
- port-channel load-balance trident fields mac

Load Balance Policies Commands

- distribution random
- distribution symmetric-hash
- fields ip
- fields mac
- ingress load-balance profile
- load-balance fm6000 profile
- load-balance policies
- port-channel hash-seed

EXEC Commands

- show etherchannel
- show lacp aggregates
- show lacp counters
- show lacp interface
- show lacp internal
- show lacp neighbor
- show lacp sys-id
- show load-balance profile
- show port-channel
- show port-channel limits
- show port-channel load-balance fields
- show port-channel summary
- show port-channel traffic

channel-group

The **channel-group** command assigns the configuration mode Ethernet interfaces to a channel group and specifies LACP attributes for the channel. When adding channels to a previously created channel group, the LACP mode for the new channel must match the mode for the existing group.

Channel groups are associated with a port channel interface immediately upon their creation. A command that creates a new channel group also creates a port channel with a matching ID. The port channel is configured in port-channel configuration mode. Configuration changes to a port channel interface propagate to all Ethernet interfaces in the corresponding channel group. The **interface port-channel** command places the switch in interface-port-channel configuration mode.

The **no channel-group** and **default channel group** commands remove the configuration mode interface from the specified channel group.

Command Mode

Interface-Ethernet Configuration

Command Syntax

```
channel-group number LACP_MODE
no channel-group
default channel-group
```

Parameters

- *number* specifies a channel group ID. Values range from 1 through 2000.
- ***LACP_MODE*** specifies the interface LACP mode. Values include:
  - **mode on** Interface is a static port channel, LACP disabled. Port neither verifies nor negotiates port channel membership.
  - **mode active** Interface is an active LACP port that transmits and receives LACP negotiation packets.
  - **mode passive** Interface is a passive LACP port that only responds to LACP negotiation packets.

Guidelines: Port Channels

You can configure a port channel to contain many ports, but only a subset may be active at a time. All active ports in a port channel must be compatible. Compatibility includes many factors and is platform-specific. For example, compatibility may require identical operating parameters such as speed and maximum transmission unit (MTU). Compatibility may only be possible between specific ports because of the internal organization of the switch.

Guidelines: MLAG Configurations

Static LAG is not recommended in MLAG configurations. However, these considerations apply when the channel group mode is ***on*** while configuring static MLAG:

- When configuring multiple interfaces on the same static port channel:
  - all interfaces must physically connect to the same neighboring switch.
  - the neighboring switch must configure all interfaces into the same port channel.

  The switches are misconfigured when these conditions are not met.
- Disable the static port channel membership before moving any cables connected to these interfaces or changing a static port channel membership on the remote switch.

Example

- These commands assign Ethernet interfaces 8 and 9 to channel group 10, and enable LACP in negotiating mode.

```
switch(config)#interface ethernet 8-9
switch(config-if-Et8-9)#channel-group 10 mode active
switch(config-if-Et8-9)#show active
interface Ethernet8
   channel-group 10 mode active
interface Ethernet9
   channel-group 10 mode active
switch(config-if-Et8-9)#
```

distribution random

The **distribution random** command specifies the random distribution of data packets handled by the hash algorithm defined by the configuration mode load balance profile. All data fields and hash seeds that are configured for the profile are used as seeds for the random number generator that defines the distribution of individual packets.

Command options allow for the random distribution of traffic across port channel links and ECMP routes. Random distribution can be enabled for either, both, or neither.

The **no distribution random** and **default distribution random** commands remove random distribution on the configuration mode load balance profile by deleting the corresponding **distribution random** command from the configuration.

Command Mode

Load-balance-profile Configuration

Command Syntax

```
distribution random BALANCE_TYPE
no distribution random
default distribution random
```

Parameters

- ***SCOPE*** Specifies use of random distribution for port channels and ECMP routes. Options include:
  - <no parameter> Random distribution is enabled for ECMP routes and port channel links.
  - **ecmp** Random distribution is enabled for ECMP routes.
  - **port-channel** Random distribution is enabled for port channel links.
  - **ecmp port-channel** Random distribution is enabled for ECMP routes and port channel links.
  - **port-channel ecmp** Random distribution is enabled for ECMP routes and port channel links.

Guidelines

The **distribution random** command takes precedence over the **distribution symmetric-hash** command when both methods are simultaneously enabled.

Related Commands

- **load-balance fm6000 profile** places the switch in load-balance-profile configuration mode.

Example

- These commands configure symmetric hashing on all traffic distributed through the algorithm defined by the LB-1 load balance profile.

```
switch(config)#load-balance policies
switch(config-load-balance-policies)#load-balance fm6000 profile LB-1
switch(config-load-balance-profile-LB-1)#distribution random ecmp port-channel
switch(config-load-balance-profile-LB-1)#show active
load-balance policies
   load-balance fm6000 profile LB-1
      distribution random ecmp port-channel
switch(config-load-balance-profile-LB-1)#
```

distribution symmetric-hash

The **distribution symmetric-hash** command enforces traffic symmetry on data distributed by the hash algorithm defined by the configuration mode load balance profile. Symmetric traffic is the flow of both directions of a data stream across the same physical link.

Two symmetric-hash options specify the traffic upon which symmetry is enforced:

- **distribution symmetric-hash mac** specifies that only non-IP traffic is hashed symmetrically. IP traffic is hashed normally without regard to symmetry.
- **distribution symmetric-hash mac-ip** specifies that all traffic is hashed symmetrically.

The **no distribution symmetric-hash** and **default distribution symmetric-hash** commands remove the specified hashing symmetry restriction on the configuration mode load balance profile by deleting the corresponding **distribution symmetric-hash** command from ***running-config.***

Command Mode

Load-balance-profile Configuration

Command Syntax

```
distribution symmetric-hash FIELD_TYPE
no distribution symmetric-hash
default distribution symmetric-hash
```

Parameters

- ***FIELD_TYPE*** fields the hashing algorithm uses for layer 3 routing. Options include:
  - **mac** non-IP traffic is hashed symmetrically.
  - **mac-ip** all traffic is hashed symmetrically.

Guidelines

The **distribution random** command takes precedence over the **distribution symmetric-hash** command when both methods are simultaneously enabled.

Related Commands

- **load-balance fm6000 profile** places the switch in load-balance-profile configuration mode.

Example

- These commands configure symmetric hashing on all traffic distributed through the algorithm defined by the LB-1 load balance profile.

```
switch(config)#load-balance policies
switch(config-load-balance-policies)#load-balance fm6000 profile LB-1
switch(config-load-balance-profile-LB-1)#distribution symmetric-hash mac-ip
switch(config-load-balance-profile-LB-1)#show active
load-balance policies
   load-balance fm6000 profile LB-1
      distribution symmetric-hash mac-ip
switch(config-load-balance-profile-LB-1)#
```

fields ip

The **fields ip** command specifies the L3/L4 data fields used by the hash algorithm defined by the configuration mode load balance profile. When a load balance profile is assigned to a port channel or Ethernet interface, its associated hash algorithm determines the distribution of packets that ingress the interface. Profile algorithms can load balance packets across port channel links or ECMP routes.

The switch calculates a hash value by using the packet header fields to balance packets across links. The hash value determines the link through which the packet is transmitted. This method also ensures that all packets in a flow follow the same network path. Packet flow is modified by changing the inputs to the port channel hash algorithm.

In network topologies that include MLAGs, programming all switches to perform the same hash calculation increases the risk of hash polarization, which leads to uneven load distribution among LAG and MLAG member links in MLAG switches. This problem is avoided by performing different hash calculations between the MLAG switch, and a non-peer switch connected to it.

The **no fields ip** configures the algorithm not to use L3/L4 data fields. The **default fields ip** command restores the default data L3/L4 fields to the load balancing algorithm defined by the configuration mode profile by removing the corresponding **fields ip** or **no fields ip** command from ***running-config***.

Command Mode

Load-balance-profile Configuration

Command Syntax

```
fields ip IP_FIELD
no fields ip
default fields ip
```

Parameters

- ***IP_FIELD*** specifies the L3/L4 fields the hashing algorithm uses. Options include:
  - **dscp** algorithm uses dscp field.
  - **dst-ip** algorithm uses destination IP address field.
  - **dst-port** algorithm uses destination TCP/UDP port field.
  - **protocol** algorithm uses protocol field.
  - **src-ip** algorithm uses source IP address field.
  - **src-port** algorithm uses source TCP/UDP port field.

  Command may include from one to six fields, in any combination and listed in any order. The default setting is the selection of all fields.

Related Commands

- **load-balance fm6000 profile** places the switch in load-balance-profile configuration mode.

Example

- These commands specify the IP source and protocol fields as components of the hash algorithm defined by the LB-1 load balance profile.

```
switch(config)#load-balance policies
switch(config-load-balance-policies)#load-balance fm6000 profile LB-1
switch(config-load-balance-profile-LB-1)#fields ip src-ip protocol
switch(config-load-balance-profile-LB-1)#show active
load-balance policies
   load-balance fm6000 profile LB-1
      fields ip protocol src-ip
switch(config-load-balance-profile-LB-1)#
```

fields mac

The **fields mac** command specifies the L2 data fields used by the hash algorithm defined by the configuration mode load balance profile. When a load balance profile is assigned to a port channel or Ethernet interface, its associated hash algorithm determines the distribution of packets that ingress the interface. Profile algorithms can load balance packets across port channel links or ECMP routes.

The switch calculates a hash value using the packet header fields to balance packets across links. The hash value determines the link through which the packet is transmitted. This method also ensures that all packets in a flow follow the same network path. Packet flow is modified by changing the inputs to the port channel hash algorithm.

In network topologies that include MLAGs, programming all switches to perform the same hash calculation increases the risk of hash polarization, which leads to uneven load distribution among LAG and MLAG member links in MLAG switches. This problem is avoided by performing different hash calculations between the MLAG switch, and a non-peer switch connected to it.

The **no fields mac** configures the algorithm not to use L2 data fields. The **default fields mac** command restores the default data L2 fields to the load balancing algorithm defined by the configuration mode profile by removing the corresponding **fields mac** or **no fields mac** command from ***running-config***.

Command Mode

Load-balance-profile Configuration

Command Syntax

```
fields mac MAC_FIELD
no fields mac
default fields mac
```

Parameters

- ***MAC_FIELD*** specifies the L2 fields the hashing algorithm uses. Options include:
  - **dst-mac** algorithm uses MAC destination field.
  - **eth-type** algorithm uses MAC destination field.
  - **src-mac** algorithm uses MAC source field.
  - **vlan-id** algorithm uses VLAN ID field.
  - **vlan-priority** algorithm uses VLAN priority field.

Related Commands

- **load-balance fm6000 profile** places the switch in load-balance-profile configuration mode.

Example

- These commands specify the MAC destination and VLAN priority fields as components of the hash algorithm defined by the LB-1 load balance profile.

```
switch(config)#load-balance policies
switch(config-load-balance-policies)#load-balance fm6000 profile LB-1
switch(config-load-balance-profile-LB-1)#fields mac dst-mac vlan-priority
switch(config-load-balance-profile-LB-1)#show active
load-balance policies
   load-balance fm6000 profile LB-1
      fields mac dst-mac vlan-priority
switch(config-load-balance-profile-LB-1)#
```

ingress load-balance profile

The **ingress load-balance profile** command applies the specified load-balance profile to the configuration mode interface. Load balance profiles specify parameters used by hashing algorithms that distribute traffic across ports comprising a port channel or among ECMP routes. The switch supports 16 load balance profiles, including the default profile.

Load balance profiles can be assigned to Ethernet and port channel interfaces. Profiles define the distribution method of traffic that ingresses the interface among the ports comprising a port channel or routes comprising an ECMP.

The default load balance profile is configured through **port-channel load-balance fm6000 fields ip** and **port-channel load-balance fm6000 fields mac** commands.

The **no ingress load-balance profile** and **default ingress load-balance profile** commands restore the default load balance profile for the configuration mode interface by removing the corresponding **ingress load-balance profile** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration
Interface-Port-Channel Configuration

Command Syntax

```
ingress load-balance profile profile_name
no ingress load-balance profile
default ingress load-balance profile
```

Parameters

- *profile_name* name of profile assigned to interface.

Example

- This command applies the ***LB-1*** load balance profile to port channel interface 100.

```
switch(config)#interface port-channel 100
switch(config-if-Po100)#show active
interface Port-Channel100
switch(config-if-Po100)#ingress load-balance profile LB-1
switch(config-if-Po100)#show active
interface Port-Channel100
   ingress load-balance profile LB-1
switch(config-if-Po100)#
```

interface port-channel

The **interface port-channel** command places the switch in port-channel interface configuration mode for modifying parameters of specified link aggregation (LAG) interfaces. When entering configuration mode to modify existing port channel interfaces, the command can specify multiple interfaces.

The command creates a port channel interface if the specified interface does not exist prior to issuing the command. When creating an interface, the command can only specify a single interface.

The **no interface port-channel** and **default interface port-channel** commands delete the specified LAG interfaces from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
interface port-channel p_range
no interface port-channel p_range
default interface port-channel p_range
```

Parameter

- *p_range* port channel interfaces (number, range, or comma-delimited list of numbers and ranges).

  Port channel numbers range from 1 to 2000.

Guidelines

When configuring a port channel, you do not need to issue the **interface port-channel** command before assigning a port to the port channel (see the **channel-group** command). The port channel number is implicitly created when a port is added to the specified port channel with the **channel-group number** command.

To display ports that are members of a port channel, enter **show port-channel**. To view information about hardware limitations for a port channel, enter **show port-channel limits**.

All active ports in a port channel must be compatible. Compatibility comprises many factors and is specific to a given platform. For example, compatibility may require identical operating parameters such as speed and/or maximum transmission unit (MTU). Compatibility may only be possible between specific ports because of internal organization of the switch.

You can configure a port channel with a set of ports such that more than one subset of the member ports are mutually compatible. Port channels in EOS are designed to activate the compatible subset of ports with the largest aggregate capacity. A subset with two 40 Gbps ports (aggregate capacity 80 Gbps) has preference to a subset with five active 10 Gbps ports (aggregate capacity 50 Gbps).

Example

- This example creates port channel interface 3:

```
switch(config)#interface port-channel 3
switch(config-if-Po3)#
```

lacp port-priority

The **lacp port-priority** command sets the aggregating port priority for the configuration mode interface. Priority is supported on port channels with LACP-enabled physical interfaces. LACP port priority determines the port that is active in a LAG in fallback mode. Numerically lower values have higher priority.

Each port in an aggregation is assigned a 32-bit port identifier by prepending the port priority (16 bits) to the port number (16 bits). Port priority determines the ports that are placed in standby mode when hardware limitations prevent a single aggregation of all compatible ports.

Priority numbers range from 0 to 65535. The default is 32768. Interfaces with higher priority numbers are placed in standby mode before interfaces with lower priority numbers.

The **no lacp port-priority** and **default lacp port-priority** commands restore the default port-priority to the configuration mode interface by removing the corresponding **lacp port-priority** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration

Command Syntax

```
lacp port-priority priority_value
no lacp port-priority
default lacp port-priority
```

Parameters

- *priority_level* port priority. Values range from 0 to 65535. Default is 32768

Example

- These commands assign the port priority of 4096 to Ethernet interface 8.

```
switch(config)#interface ethernet 8
switch(config-if-Et8)#lacp port-priority 4096
switch(config-if-Et8)#show active
interface Ethernet8
   lacp port-priority 4096
switch(config-if-Et8)#
```

lacp rate

The **lacp rate** command configures the LACP transmission interval on the configuration mode interface. The LACP timeout specifies the transmission rate of LACP control packets to interfaces supporting LACP. Supported rates include:

- ***normal***: 30 seconds with synchronized interfaces; one second while interfaces are synchronizing.
- ***fast***: one second.

This command is supported on LACP-enabled interfaces. The default value is ***normal***.

The **no lacp rate** and **default lacp rate** commands restore the default value of ***normal*** on the configuration mode interface by deleting the corresponding **lacp rate** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration

Command Syntax

```
lacp rate RATE_LEVEL
no lacp rate
default lacp rate
```

Parameters

- ***RATE_LEVEL*** LACP transmission interval . Options include:
  - **fast** one second.
  - **normal** 30 seconds for synchronized interfaces; one second while interfaces synchronize.

Examples

- This command sets the LACP rate to one second on Ethernet interface 4.

```
Switch(config-if-Et4)#lacp rate fast
Switch(config-if-Et4)#
```

lacp system-priority

The **lacp system-priority** command configures the switch's LACP system priority. Values range between 0 and 65535. Default value is 32768.

Each switch is assigned a globally unique 64-bit system identifier by prepending the system priority (16 bits) to the MAC address of one of its physical ports (48 bits). Peer devices use the system identifier when forming an aggregation to verify that all links are from the same switch. The system identifier is also used when dynamically changing aggregation capabilities resulting from LACP data; the system with the numerically lower system identifier can dynamically change advertised aggregation parameters.

The **no lacp system-priority and default lacp system-priority** commands restore the default system priority by removing the **lacp system-priority** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
lacp system-priority priority_value
no lacp system-priority
default lacp system-priority
```

Parameters

- *priority_value* system priority number. Values range from 0 to 65535. Default is 32768.

Example

- This command assigns the system priority of 8192 to the switch.

```
switch(config)#lacp system-priority 8192
switch(config)#
```

load-balance fm6000 profile

The **load-balance fm6000 profile** command places the switch in load-balance-profile configuration mode to configure a specified load balance profile. The command specifies the name of the profile that subsequent commands modify. It creates a profile if the profile it references does not exist.

Load balance profiles specify parameters used by hashing algorithms that distribute traffic across ports comprising a port channel or among component ECMP routes. The switch supports 16 load balance profiles, including the default profile. The default load balance profile is configured through **port-channel load-balance fm6000 fields ip** and **port-channel load-balance fm6000 fields mac** commands.

The load balance profile name is referenced when it is applied to an interface. The default profile is not associated with a name and is applied to an interface in the absence of a named profile assignment.

The **no load-balance fm6000 profile** and **default load-balance fm6000 profile** commands delete the specified load balance profile from ***running-config***. Profiles that are assigned to an interface cannot be deleted. Attempts to delete an assigned profile generate a ***profile in use*** error messages.

The **load-balance fm6000 profile** command is accessible from load-balance-policies configuration mode. Load-balance-profile configuration mode is not a group change mode; ***running-config*** is changed immediately upon entering commands. Exiting load-balance-policies configuration mode does not affect the configuration. The **exit** command returns the switch to load-balance-policies configuration mode.

Command Mode

Load-balance-policies Configuration

Command Syntax

```
load-balance fm6000 profile profile_name
no load-balance fm6000 profile profile_name
default load-balance fm6000 profile profile_name
```

Parameters

- *profile_name* name of the load-balance profile.

Commands Available in Load-balance-profile Configuration Mode

- **fields ip**
- **fields mac**
- **distribution random**
- **distribution symmetric-hash**
- **port-channel hash-seed**
- **show active** displays the contents of the configuration mode profile.

Related Commands

- **load-balance policies** places the switch in load-balance-policies configuration mode.
- **ingress load-balance profile** applies a load-balance profile to an Ethernet or port channel interface.
- **show load-balance profile** displays the contents of load balance profiles.

Example

- These commands enter load-balance-profile configuration mode, creates the LB-1 profile, and lists the default settings for the profile.

```
switch(config)#load-balance policies
switch(config-load-balance-policies)#load-balance fm6000 profile LB-1
switch(config-load-balance-profile-LB-1)#show active all
load-balance policies
   load-balance fm6000 profile LB-1
      port-channel hash-seed 0
      fields mac dst-mac src-mac eth-type vlan-priority vlan-id
      fields ip protocol dst-ip dst-port src-ip src-port dscp
      no distribution symmetric-hash
      no distribution random
switch(config-load-balance-profile-LB-1)#
```

load-balance policies

The **load-balance policies** command places the switch in load-balance-policies configuration mode. Load-balance-policies configuration mode provides commands for managing load-balance profiles. Load balance profiles specify the inputs used by the hashing algorithms that distribute traffic across ports comprising a port channel or among ECMP routes.

The **no load-balance policies** and **default load-balance policies** commands delete all load balance profiles from ***running-config***. The command generates an error message when at least one profile is assigned to an interface.

Load-balance-policies configuration mode is not a group change mode; ***running-config*** is changed immediately upon entering commands. Exiting load-balance-policies configuration mode does not affect ***running-config***. The **exit** command returns the switch to global configuration mode.

Command Mode

Global Configuration

Command Syntax

```
load-balance policies
no load-balance policies
default load-balance policies
```

Commands Available in Load-balance-policies Configuration Mode

- **load-balance fm6000 profile** places the switch in load-balance-profile configuration mode.
- **show active** displays contents of all load balance profiles.

Related Commands

- **ingress load-balance profile** applies a load-balance profile to an Ethernet or port channel interface.
- **show load-balance profile** displays the contents of load balance profiles.

Example

- This command places the switch in load-balance-policies configuration mode.

```
switch(config)#load-balance policies
switch(config-load-balance-policies)#
```

- This command displays the contents of the three configured load balance profiles.

```
switch(config-load-balance-policies)#show active
load-balance policies
   load-balance fm6000 profile F-01
      port-channel hash-seed 22
      fields ip dscp
      distribution random port-channel
   !
   load-balance fm6000 profile F-02
      fields ip protocol dst-ip
      fields mac dst-mac eth-type
      distribution random ecmp port-channel
   !
   load-balance fm6000 profile F-03
switch(config-load-balance-policies)#
```

port-channel hash-seed

The **port-channel hash-seed** command specifies the seed used by the hash algorithm defined by the configuration mode load balance profile when distributing the load across ports comprising a port channel. When a load balance profile is assigned to a port channel or Ethernet interface, its associated hash algorithm determines the distribution of packets that ingress the interface. Profile algorithms can load balance packets across port channel links or ECMP routes.

The hash seed that the algorithm uses to select port channel links or ECMP routes is configured by the **ip load-sharing** command.

The **no port-channel hash-seed** and **default port-channel hash-seed** commands restore the default hash seed value of 0 to the load balancing algorithm defined by the configuration mode profile by removing the corresponding **port-channel hash-seed** command from ***running-config***.

Command Mode

Load-balance-profile Configuration

Command Syntax

```
port-channel hash-seed number
no port-channel hash-seed
default port-channel hash-seed
```

Parameters

- *number* The hash seed. Value ranges from 0 to 39.

Related Commands

- **load-balance fm6000 profile** places the switch in load-balance-profile configuration mode.

Example

- Thes commands configure the port-channel hash seed of 22 for the hash algorithm defined by the LB-1 load balance profile.

```
switch(config)#load-balance policies
switch(config-load-balance-policies)#load-balance fm6000 profile LB-1
switch(config-load-balance-profile-LB-1)#port-channel hash-seed 22
switch(config-load-balance-profile-LB-1)#show active
load-balance policies
   load-balance fm6000 profile LB-1
      port-channel hash-seed 22
switch(config-load-balance-profile-LB-1)#
```

port-channel lacp fallback

The **port-channel lacp fallback** command enables the LACP fallback mode on the interface.

LACP fallback is unconfigured and disabled by default. An LACP interface without fallback enabled does not form a LAG until it receives PDUs from its peer.

LACP fallback can be configured on an interface in static or individual mode:

- **static mode** the port channel member with the lowest LACP port priority is active and maintains contact with the peer (sending and receiving data) while other port channel members remain in standby mode until a LACP PDU is received. All members continue to send (and can receive) LACP PDUs.
- **individual mode** all port channel members act as individual ports, reverting to their port-specific configuration while the channel is in fallback mode, and continue to send and receive data. All members continue to send LACP PDUs until a LACP PDU is received by one of the member ports.

The **no port-channel lacp fallback** and **default port-channel lacp fallback** commands disable LACP fallback mode on the configuration mode interface by removing the corresponding **port-channel lacp fallback** command from ***running-config***.

Command Mode

Interface-Port-Channel Configuration

Command Syntax

```
port-channel lacp fallback [MODE]
no port-channel lacp fallback
default port-channel lacp fallback
```

Parameters

- ***MODE*** LACP fallback mode. Options include:
  - <no parameter> enables static LACP fallback mode.
  - **static** enables static LACP fallback mode.
  - **individual** enables individual LACP fallback mode.

Related Commands

port-channel lacp fallback timeout configures the fallback timeout period for a port channel interface. The default LACP fallback timeout period is 90 seconds.

lacp port-priority configures the port priority for an individual interface.

Examples

- These commands enable LACP static fallback mode, then configure an LACP fallback timeout of 100 seconds on port channel interface 13. If LACP negotiation fails, only the member port with the lowest LACP priority will remain active until an LACP PDU is received by one of the member ports.

```
switch(config)#interface port-channel 13
switch(config-if-Po13)#port-channel lacp fallback static
switch(config-if-Po13)#port-channel lacp fallback timeout 100
switch(config-if-Po13)#show active
interface Port-Channel13
   port-channel lacp fallback static
   port-channel lacp fallback timeout 100
switch(config-if-Po13)#
```

- These commands enable LACP individual fallback mode, then configure an LACP fallback timeout of 50 seconds on port channel interface 17. If LACP negotiation fails, all member ports will act as individual switch ports, using port-specific configuration, until a LACP PDU is received by one of the member ports.

```
switch(config)#interface port-channel 17
switch(config-if-Po17)#port-channel lacp fallback individual
switch(config-if-Po17)#port-channel lacp fallback timeout 50
switch(config-if-Po17)#show active
interface Port-Channel17
   port-channel lacp fallback individual
   port-channel lacp fallback timeout 50
switch(config-if-Po17)#
```

port-channel lacp fallback timeout

The **port-channel lacp fallback timeout** command specifies the fallback timeout period for the configuration mode interface.

Fallback timeout settings persist in ***running-config*** without taking effect for interfaces that are not configured into fallback mode. The default fallback timeout period is 90 seconds.

The **no port-channel lacp fallback timeout** and **default port-channel lacp fallback timeout** commands restore the default fallback timeout of 90 seconds for the configuration mode interface by removing the corresponding **port-channel lacp fallback timeout** command from ***running-config***.

Command Mode

Interface-Port-Channel Configuration

Command Syntax

```
port-channel lacp fallback timeout period
no port-channel lacp fallback timeout
default port-channel lacp fallback timeout
```

Parameters

- *period* maximum interval between receipt of LACP PDU packets (seconds). Value ranges from 1 to 300 seconds. Default value is 90.

Related Commands

port-channel lacp fallback configures fallback mode for a port channel interface.

Guidelines

The fallback timeout period should not be shorter than the LACP transmision interval (**lacp rate**). The default LACP transmission interval is 30 seconds.

Example

- This command enables LACP fallback mode, then configures an LACP fallback timeout of 100 seconds on port channel interface 13.

```
switch(config)#interface port-channel 13
switch(config-if-Po13)#port-channel lacp fallback
switch(config-if-Po13)#port-channel lacp fallback timeout 100
switch(config-if-Po13)#show active
interface Port-Channel13
   port-channel lacp fallback
   port-channel lacp fallback timeout 100
switch(config-if-Po13)#
```

port-channel load-balance

The **port-channel load-balance** command specifies the seed in the hashing algorithm that balances the load across ports comprising a port channel. Available seed values vary by switch platform.

The **no port-channel load-balance** and **default port-channel load-balance** commands remove the **port-channel load-balance** command from ***running-config***, restoring the default hash seed value of 0.

Command Mode

Global Configuration

Command Syntax

```
port-channel load-balance platform { hash_seed | fields ip fields | hash
hash_function }
no port-channel load-balance platform [hash_seed]
default port-channel load-balance platform [hash_seed]
```

Parameters

Important! Parameter options vary by switch model. Verify available options with the ? command.

- *platform* ASIC switching device. Value depends on the switch model.
- *hash_seed* The numerical seed for the hash function. Value range varies by switch platform:
  - **arad** 0 to 65535.
  - **fm6000** 0 to 39.
  - **petraA** uses field inputs only.
  - **trident** 0 to 47.

    For trident platform switches, algorithms using hash seeds between 0 and 15 typically result in more effective distribution of data streams across the port channels.
- *fields* Which fields will be used as inputs to the port channel hash.
  - **gre** Configure which GRE fields are inputs to the hash.
  - **ip** Configure which fields are inputs to the hash *for IPv4 packets*.
  - **ipv6** Configure which fields are inputs to the hash *for IPv6 packets*.
  - **mac** Configure which MAC fields are inputs to the hash.
  - **mac-in-mac** Configure which MAC-in-MAC fields are inputs to the hash.
  - **mpls** Configure which MPLS fields are inputs to the hash.
  - **destination-ip** Use the layer 3 IP destination address in the hash.
  - **destination-port** Use the layer 4 TCP/UDP destination port in the hash.
  - **dst-ip** Use the destination IP address in the hash.
  - **dst-mac** Use the destination Payload MAC in the hash (or the destination MAC address in the MAC hash).
  - **eth-type** Use the Ethernet type in the MAC hash.
  - **ip-in-ip** Use the outer IP header in the hash for IPv4 over IPv4 GRE tunnel.
  - **ip-in-ipv6** Use the outer IP header in the hash for IPv4 over IPv6 GRE tunnel.
  - **ipv6-in-ip** Use the outer IP header in the hash for IPv6 over IPv4 GRE tunnel.
  - **ipv6-in-ipv6** Use the outer IP header in the hash for IPv6 over IPv6 GRE tunnel.

  - **ip-tcp-udp-header** Use the layer 3 and layer 4 hashes.
  - **isid** Use the MAC-in-MAC ISID in the hash.
  - **label** Use the MPLS label in the hash.
  - **mac-header** Use the MAC hash.
  - **outer-mac** Use the outer MAC of source and destination in the hash.
  - **source-ip** Use the layer 3 IP source address in the hash.
  - **src-ip** Use the source IP address in the hash.
  - **source-port** Use layer 4 TCP/UDP source port in the hash.
  - **src-mac** Use the source payload MAC in the hash (or the source MAC address in the MAC hash).
- *hash_function* Specifies the hash polynomial function. Values range from 0-2.

Example

- This command configures a hash seed of 10 on an FM6000 platform switch.

```
switch(config)#port-channel load-balance fm6000 10
switch(config)#
```

port-channel load-balance arad fields ip

The **port-channel load-balance arad fields ip** command specifies the data fields that the port channel load balance hash algorithm uses for distributing IP packets on Arad platform switches. The hashing algorithm fields used for IP packets differ from the fields used for non-IP packets.

The switch calculates a hash value using the packet header fields to load balance packets across links in a port channel. The hash value determines the link through which the packet is transmitted. This method also ensures that all packets in a flow follow the same network path. Packet flow is modified by changing the inputs to the port channel hash algorithm.

In network topologies that include MLAGs, programming all switches to perform the same hash calculation increases the risk of hash polarization, which leads to uneven load distribution among LAG and MLAG member links in MLAG switches. This problem is avoided by performing different hash calculations between the MLAG switch, and a non-peer switch connected to it.

The **no port-channel load-balance arad fields ip** and **default port-channel load-balance arad fields ip** commands restore the default data fields for the IP packet load balancing algorithm by removing the **port-channel load-balance arad A fields ip** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
port-channel load-balance arad fields ip IP_FIELD_NAME
no port-channel load-balance arad fields ip
default port-channel load-balance arad fields ip
```

Parameters

- ***IP_FIELD_NAME*** fields the hashing algorithm uses for layer 3 routing. Options include:
  - **ip-tcp-udp-header** algorithm uses source and destination IP address fields. Source and destination port fields are included for TCP and UDP packets.
  - **mac-header** algorithm uses entire MAC header.

  A command can only specify one option. The default setting is ***ip-tcp-udp-header***.

Guidelines

The port channel hash algorithm for non-IP packets is not configurable and always includes the entire MAC header.

Related Commands

- **port-channel load-balance** configures the hash seed for the algorithm.

Example

- These commands configure the switch's port channel load balance hash algorithm for IP packets to use source and destination IP address (and port) fields.

```
switch(config)#port-channel load-balance fm6000 fields ip ip-tcp-udp-header
switch(config)#
```

port-channel load-balance fm6000 fields ip

The **port-channel load-balance fm6000 fields ip** command specifies the data fields that the port channel load balance hash algorithm uses for distributing IP packets on FM6000 platform switches. The hashing algorithm fields used for IP packets differ from the fields used for non-IP packets.

The switch calculates a hash value using the packet header fields to load balance packets across links in a port channel. The hash value determines the link through which the packet is transmitted. This method also ensures that all packets in a flow follow the same network path. Packet flow is modified by changing the inputs to the port channel hash algorithm.

In network topologies that include MLAGs, programming all switches to perform the same hash calculation increases the risk of hash polarization, which leads to uneven load distribution among LAG and MLAG member links in MLAG switches. This problem is avoided by performing different hash calculations between the MLAG switch, and a non-peer switch connected to it.

The **no port-channel load-balance fm6000 fields ip** and **default port-channel load-balance fm6000 fields ip** commands restore the default data fields for the IP packet load balancing algorithm by removing the **port-channel load-balance fm6000 fields ip** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
port-channel load-balance fm6000 fields ip IP_FIELD_NAME
no port-channel load-balance fm6000 fields ip
default port-channel load-balance fm6000 fields ip
```

Parameters

- ***IP_FIELD_NAME*** specifies fields the hashing algorithm uses for layer 3 routing. Options include:
  - **ip-tcp-udp-header** algorithm uses source and destination IP address fields. Source and destination port fields are included for TCP and UDP packets.

  A command can only specify one option. The default setting is ***ip-tcp-udp-header***.

Related Commands

- **port-channel load-balance** configures the hash seed for the algorithm.
- **port-channel load-balance fm6000 fields mac** controls the hash algorithm for non-IP packets

Example

- These commands configure the switch's port channel load balance for IP packets by source and destination IP address and port fields.

```
switch(config)#port-channel load-balance fm6000 fields ip ip-tcp-udp-header
switch(config)#
```

port-channel load-balance fm6000 fields mac

The **port-channel load-balance fm6000 fields mac** command specifies data fields that configure the port channel load balance hash algorithm for non-IP packets on FM6000 platform switches. The hashing algorithm fields used for balancing non-IP packets differ from the fields used for IP packets.

The switch calculates a hash value using the packet header fields to load balance packets across links in a port channel. The hash value determines the link through which the packet is transmitted. This method also ensures that all packets in a flow follow the same network path. Packet flow is modified by changing the inputs to the port channel hash algorithm.

In network topologies that include MLAGs, programming all switches to perform the same hash calculation increases the risk of hash polarization, which leads to uneven load distribution among LAG and MLAG member links in MLAG switches. This problem is avoided by performing different hash calculations between the MLAG switch, and a non-peer switch connected to it.

The **no port-channel load-balance fm6000 fields mac** and **default port-channel load-balance fm6000 fields mac** commands restore the default data fields for the non-IP packet load balancing algorithm by removing the **port-channel load-balance fm6000 fields mac** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
port-channel load-balance fm6000 fields mac MAC_FIELD_NAME
no port-channel load-balance fm6000 fields mac
default port-channel load-balance fm6000 fields mac
```

Parameters

- ***MAC_FIELD_NAME*** fields the hashing algorithm uses for layer 2 routing. Options include
  - **dst-mac** MAC destination field
  - **eth-type** EtherType field
  - **src-mac** MAC source field
  - **vlan-id** VLAN ID field
  - **vlan-priority** VLAN priority field

  Command may include from one to five fields, in any combination and listed in any order. The default setting is the selection of all fields.

Related Commands

- **port-channel load-balance** configures the hash seed for the algorithm.
- **port-channel load-balance fm6000 fields ip** controls the hash algorithm for IP packets

Example

- These commands configure the switch's port channel load balance for non-IP packets by using the MAC destination and Ethernet type fields in the hashing algorithm.

```
switch(config)#port-channel load-balance fm6000 fields mac dst-mac eth-type
switch(config)#
```

port-channel load-balance module

The **port-channel load-balance module** command assigns a named load-balancing profile to a linecard.

Note Available on the 7500E platform.

The **no port-channel load-balance module** and **default port-channel load-balance module** commands unassigns the load balancing module, or restores the default data fields for the load balancing module.

Command Mode

Global Configuration

Command Syntax

```
port-channel load-balance module LINECARD_RANGE sand profile PROFILE_NAME
no port-channel load-balance module LINECARD_RANGE sand profile PROFILE_NAME
default port-channel load-balance module LINECARD_RANGE sand profile PROFILE_NAME
```

Parameters

- ***LINECARD_RANGE*** linecard number range includes
  - **<3-10>** linecard number range
- ***PROFILE_NAME*** load-balance profile name

Examples

- This command assigns a named load-balancing profile to a linecard.

```
switch(config)#port-channel load-balance module 3-7 sand profile Linecard5
switch(config)#
```

- This command unassigns a named load-balancing profile to a linecard.

```
switch(config)#no port-channel load-balance module 3-7 sand profile Linecard5
switch(config)#
```

port-channel load-balance petraA fields ip

The **port-channel load-balance petraA fields ip** command specifies the data fields that the port channel load balance hash algorithm uses for distributing IP packets on Petra platform switches. The hashing algorithm fields used for IP packets differ from the fields used for non-IP packets.

The switch calculates a hash value using the packet header fields to load balance packets across links in a port channel. The hash value determines the link through which the packet is transmitted. This method also ensures that all packets in a flow follow the same network path. Packet flow is modified by changing the inputs to the port channel hash algorithm.

In network topologies that include MLAGs, programming all switches to perform the same hash calculation increases the risk of hash polarization, which leads to uneven load distribution among LAG and MLAG member links in MLAG switches. This problem is avoided by performing different hash calculations between the MLAG switch, and a non-peer switch connected to it.

The **no port-channel load-balance petraA fields ip** and **default port-channel load-balance petraA fields ip** commands restore the default data fields for the IP packet load balancing algorithm by removing the **port-channel load-balance petraA fields ip** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
port-channel load-balance petraA fields ip IP_FIELD_NAME
no port-channel load-balance petraA fields ip
default port-channel load-balance petraA fields ip
```

Parameters

- ***IP_FIELD_NAME*** fields the hashing algorithm uses for layer 3 routing. Options include:
  - **ip-tcp-udp-header** algorithm uses source and destination IP address fields. Source and destination port fields are included for TCP and UDP packets.
  - **mac-header** algorithm uses entire MAC header.

  A command can only specify one option. The default setting is ***ip-tcp-udp-header***.

Guidelines

The port channel hash algorithm for non-IP packets is not configurable and always includes the entire MAC header.

Related Commands

- **port-channel load-balance** configures the hash seed for the algorithm.

Example

- These commands configure the switch's port channel load balance hash algorithm for IP packets to use source and destination IP address (and port) fields.

```
switch(config)#port-channel load-balance fm6000 fields ip ip-tcp-udp-header
switch(config)#
```

port-channel load-balance trident fields ip

The **port-channel load-balance trident fields ip** command specifies the data fields that the port channel load balance hash algorithm uses for distributing IP packets on Trident platform switches. The hashing algorithm fields used for IP packets differ from the fields used for non-IP packets.

The switch calculates a hash value using the packet header fields to load balance packets across links in a port channel. The hash value determines the link through which the packet is transmitted. This method also ensures that all packets in a flow follow the same network path. Packet flow is modified by changing the inputs to the port channel hash algorithm.

In network topologies that include MLAGs, programming all switches to perform the same hash calculation increases the risk of hash polarization, which leads to uneven load distribution among LAG and MLAG member links in MLAG switches. This problem is avoided by performing different hash calculations between the MLAG switch, and a non-peer switch connected to it.

The **no port-channel load-balance trident fields ip** and **default port-channel load-balance trident fields ip** commands restore the default data fields for the IP packet load balancing algorithm by removing the **port-channel load-balance trident fields ip** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
port-channel load-balance trident fields ip IP_FIELD_NAME
no port-channel load-balance trident fields ip
default port-channel load-balance trident fields ip
```

Parameters

- ***IP_FIELD_NAME*** specifies fields the hashing algorithm uses for layer 3 routing. Options include:

  Command may include from one to four of the following four options, in any combination and listed in any order.

  - **destination-ip** algorithm uses destination IP address field.
  - **source-ip** algorithm uses source IP address field.
  - **destination-port** algorithm uses destination TCP/UDP port field.
  - **source-port** algorithm uses source TCP/UDP port field.
    - **ip-tcp-udp-header** algorithm uses source and destination IP address fields. Source and destination port fields are included for TCP and UDP packets. ***This option can't be used in combination with any other option***.
  - **mac-header** algorithm uses fields specified by **port-channel load-balance trident fields mac**. *This option can't be used in combination with any other option.*

  Default setting is **ip-tcp-udp-header**

Related Commands

- **port-channel load-balance** configures the hash seed for the algorithm.
- **port-channel load-balance trident fields ipv6** controls the hash algorithm for IPv6 packets
- **port-channel load-balance trident fields mac** controls the hash algorithm for non-IP/IPv6 packets

Example

- These commands configure the switch's port channel load balance for IP packets by using the IPv6 destination field in the hashing algorithm.

```
switch(config)#port-channel load-balance trident fields ip destination-ip
switch(config)#
```

port-channel load-balance trident fields ipv6

The **port-channel load-balance trident fields ipv6** command specifies the data fields that the port channel load balance hash algorithm uses for distributing IPv6 packets on Trident platform switches. The hashing algorithm fields used for IPv6 packets differ from the fields used for non-IPv6 packets.

The switch calculates a hash value using the packet header fields to load balance packets across links in a port channel. The hash value determines the link through which the packet is transmitted. This method also ensures that all packets in a flow follow the same network path. Packet flow is modified by changing the inputs to the port channel hash algorithm.

In network topologies that include MLAGs, programming all switches to perform the same hash calculation increases the risk of hash polarization, which leads to uneven load distribution among LAG and MLAG member links in MLAG switches. This problem is avoided by performing different hash calculations between the MLAG switch, and a non-peer switch connected to it.

The **no port-channel load-balance trident fields ipv6** and **default port-channel load-balance trident fields ipv6** commands restore the default data fields for the IPv6 packet load balancing algorithm by removing the **port-channel load-balance trident fields ipv6** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
port-channel load-balance trident fields ipv6 IP_FIELD_NAME
no port-channel load-balance trident fields ipv6
default port-channel load-balance trident fields ipv6
```

Parameters

- ***IP_FIELD_NAME*** specifies fields the hashing algorithm uses for layer 3 routing. Options include:

  Command may include from one to four of the following four options, in any combination and listed in any order.

  - **destination-ip** algorithm uses destination IPv6 address field.
  - **source-ip** algorithm uses source IPv6 address field.
  - **destination-port** algorithm uses destination TCP/UDP port field.
  - **source-port** algorithm uses source TCP/UDP port field.
    - **ip-tcp-udp-header** algorithm uses source and destination IPv6 address fields. Source and destination port fields are included for TCP and UDP packets. ***This option can't be used in combination with any other option.***
  - **mac-header** algorithm uses fields specified by **port-channel load-balance trident fields mac**. *This option can't be used in combination with any other option.*

  Default setting is **ip-tcp-udp-header**

Related Commands

- **port-channel load-balance** configures the hash seed for the algorithm.
- **port-channel load-balance trident fields ipv6** controls the hash algorithm for non-IP packets
- **port-channel load-balance trident fields mac** controls the hash algorithm for non-IP packets

Example

- These commands configure the switch's port channel load balance for IP packets by using the IPv6 source field in the hashing algorithm.

```
switch(config)#port-channel load-balance trident fields ipv6 source-ip
switch(config)#
```

port-channel load-balance trident fields mac

The **port-channel load-balance trident fields mac** command specifies data fields that the port channel load balance hash algorithm uses for distributing non-IP packets on Trident platform switches. The hashing algorithm fields used for non-IP packets differ from the fields used for IP packets.

The switch calculates a hash value using the packet header fields to load balance packets across links in a port channel. The hash value determines the link through which the packet is transmitted. This method also ensures that all packets in a flow follow the same network path. Packet flow is modified by changing the inputs to the port channel hash algorithm.

In network topologies that include MLAGs, programming all switches to perform the same hash calculation increases the risk of hash polarization, which leads to uneven load distribution among LAG and MLAG member links in MLAG switches. This problem is avoided by performing different hash calculations between the MLAG switch, and a non-peer switch connected to it.

The **no port-channel load-balance trident fields mac** and **default port-channel load-balance trident fields mac** commands restore the default data fields for the non-IP packet load balancing algorithm by removing the **port-channel load-balance trident fields mac** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
port-channel load-balance trident fields mac MAC_FIELD_NAME
no port-channel load-balance trident fields mac
default port-channel load-balance trident fields mac
```

Parameters

- ***MAC_FIELD_NAME*** fields the hashing algorithm uses for layer 2 routing. Options include
  - **dst-mac** MAC destination field
  - **eth-type** EtherType field
  - **src-mac** MAC source field

  Command may include from one to three fields, in any combination and listed in any order. The default setting is the selection of all fields.

Related Commands

- **port-channel load-balance** configures the hash seed for the algorithm.
- **port-channel load-balance trident fields ip** controls the hash algorithm for IP packets
- **port-channel load-balance trident fields ipv6** controls the hash algorithm for IP packets

Example

- These commands configure the switch's port channel load balance for non-IP packets by using the MAC destination and Ethernet type fields in the hashing algorithm.

```
switch(config)#port-channel load-balance trident fields mac dst-mac eth-type
switch(config)#
```

port-channel min-links

The **port-channel min-links** command specifies the minimum number of interfaces that the configuration mode LAG requires to become active. If there are fewer ports than specified by this command, the port channel interface does not become active. The default min-links value is 0.

The **no port-channel min-links** and **default port-channel min-links** commands restore the default min-links setting for the configuration mode LAG by removing the corresponding **port-channel min-links** command from the configuration.

Note In static LAGs, the min-links value must be met for the LAG to be active. The LAG will not become active until it has at least the min-links number of functioning links in the channel group. If failed links cause the number to drop below the minimum, the LAG will go down and administrator action will be required to bring it back up.

In dynamic LAGs, the LACP protocol must determine that at least min-links physical ports are aggregable (they are physically compatible and have the same keys both remotely and locally) before it begins negotiating to make any ports active members of the port-channel. Once negotiation begins, however, an error on the partner's side can cause the LAG to become active with fewer than the minimum number of links.

Command Mode

Interface-Port-Channel Configuration

Command Syntax

```
port-channel min-links quantity
no port-channel min-links
default port-channel min-links
```

Parameters

- *quantity* minimum number of interfaces. Value range varies by platform. Default value is 0.

Example

- These commands set four as the minimum number of ports required for port channel 13 to become active.

```
switch(config)#interface port-channel 13
switch(config-if-Po13)#port-channel min-links 4
switch(config-if-Po13)#show active
interface Port-Channel13
   port-channel min-links 4
switch(config-if-Po13)#
```

show etherchannel

The **show etherchannel** command displays information about members of the specified port channels.

Command Mode

EXEC

Command Syntax

show etherchannel [*MEMBERS*] [*PORT_LIST*] [*INFO_LEVEL*]

Parameters

- ***MEMBERS*** list of port channels for which information is displayed. Options include:
  - <no parameter> all configured port channels.
  - *p_range* ports in specified channel list (number, number range, or list of numbers and ranges).
- ***PORT_LEVEL*** ports displayed, in terms of aggregation status. Options include:
  - <no parameter> Displays information on ports that are active members of the LAG**.**
  - **active-ports** Displays information on ports that are active members of the LAG.
  - **all-ports** Displays information on all ports (active or inactive) configured for LAG.
- ***INFO_LEVEL*** amount of information that is displayed. Options include:
  - <no parameter> Displays information at the brief level.
  - **brief** Displays information at the brief level.
  - **detailed** Displays information at the detail level.

Display Values

- **Port Channel** Type and name of the port channel.
- **Time became active** Time when the port channel came up.
- **Protocol** Protocol operating on the port.
- **Mode** Status of the Ethernet interface on the port. The status value is Active or Inactive.
- **No active ports** Number of active ports on the port channel.
- **Configured but inactive ports** Ports configured but that are not actively up.
- **Reason unconfigured** Reason why the port is not part of the LAG.

Guidelines

The **show etherchannel** and **show port-channel** commands are identical. See **show port-channel** for additional information.

show lacp aggregates

The **show lacp aggregates** command displays aggregate IDs and the list of bundled ports for all specified port channels.

Command Mode

EXEC

Command Syntax

show lacp [*PORT_LIST*] aggregates [*PORT_LEVEL*] [*INFO_LEVEL*]

PORT_LEVEL and ***INFO_LEVEL*** parameters can be placed in any order.

Parameters

- ***PORT_LIST*** port channels for which aggregate information is displayed. Options include:
  - <no parameter> all configured port channels.
  - *c_range* channel list (number, range, or comma-delimited list of numbers and ranges).
- ***PORT_LEVEL*** ports displayed, in terms of aggregation status. Options include:
  - <no parameter> ports bundled by LACP into the port channel.
  - **all-ports** all channel group ports, including channel group members not bundled into the port channel interface.
- ***INFO_LEVEL*** amount of information that is displayed. Options include:
  - <no parameter> aggregate ID and bundled ports for each channel.
  - **brief** aggregate ID and bundled ports for each channel.
  - **detailed** aggregate ID and bundled ports for each channel.

Examples

- This command lists aggregate information for all configured port channels.

```
switch>show lacp aggregates
Port Channel Port-Channel1:
 Aggregate ID:
[(8000,00-1c-73-04-36-d7,0001,0000,0000),(8000,00-1c-73-09-a0-f3,0001,0000,0000
)]
  Bundled Ports: Ethernet43 Ethernet44 Ethernet45 Ethernet46
Port Channel Port-Channel2:
 Aggregate ID:
[(8000,00-1c-73-01-02-1e,0002,0000,0000),(8000,00-1c-73-04-36-d7,0002,0000,0000
)]
  Bundled Ports: Ethernet47 Ethernet48
Port Channel Port-Channel3:
 Aggregate ID:
[(8000,00-1c-73-04-36-d7,0003,0000,0000),(8000,00-1c-73-0c-02-7d,0001,0000,0000
)]
  Bundled Ports: Ethernet3 Ethernet4
Port Channel Port-Channel4:
 Aggregate ID:
[(0001,00-22-b0-57-23-be,0031,0000,0000),(8000,00-1c-73-04-36-d7,0004,0000,0000
)]
  Bundled Ports: Ethernet1 Ethernet2
Port Channel Port-Channel5:
 Aggregate ID:
[(0001,00-22-b0-5a-0c-51,0033,0000,0000),(8000,00-1c-73-04-36-d7,0005,0000,0000
)]
  Bundled Ports: Ethernet41
switch>
```

show lacp counters

The **show lacp counters** command displays LACP traffic statistics.

Command Mode

EXEC

Command Syntax

show lacp [*PORT_LIST*] counters [*PORT_LEVEL*] [*INFO_LEVEL*]

PORT_LEVEL and ***INFO_LEVEL*** parameters can be placed in any order.

Parameters

- ***PORT_LIST*** ports for which port information is displayed. Options include:
  - <no parameter> all configured port channels
  - *c_range* ports in specified channel list (number, number range, or list of numbers and ranges).
  - **interface** ports on all interfaces.
  - **interface ethernet** *e_num* port on Ethernet interface specified by *e_num*.
  - **interface port-channel** *p_num* port on port channel interface specified by *p_num*.
- ***PORT_LEVEL*** ports displayed, in terms of aggregation status. Options include:
  - <no parameter> only ports bundled by LACP into an aggregate.
  - **all-ports** all ports, including LACP candidates that are not bundled.
- ***INFO_LEVEL*** amount of information that is displayed. Options include:
  - <no parameter> displays packet transmission (TX and RX) statistics.
  - **brief** displays packet transmission (TX and RX) statistics.
  - **detailed** displays packet transmission (TX and RX) statistics and actor-partner statistics.

Example

- This command displays transmission statistics for all configured port channels.

```
switch>show lacp counters brief
                      LACPDUs          Markers     Marker Response
Port   Status        RX          TX    RX    TX      RX         TX     Illegal
--------------------------------------------------------------------------------
Port Channel Port-Channel1:
Et43   Bundled   396979     396959    0     0       0          0           0
Et44   Bundled   396979     396959    0     0       0          0           0
Et45   Bundled   396979     396959    0     0       0          0           0
Et46   Bundled   396979     396959    0     0       0          0           0
Port Channel Port-Channel2:
Et47   Bundled   396836     396883    0     0       0          0           0
Et48   Bundled   396838     396883    0     0       0          0           0

switch>
```

show lacp interface

The **show lacp interface** command displays port status for all port channels that include the specified interfaces. Within the displays for each listed port channel, the output displays sys-id, partner port, state, actor port, and port priority for each interface in the channel.

Command Mode

EXEC

Command Syntax

show lacp interface [*INTERFACE_PORT*] [*PORT_LEVEL*] [*INFO_LEVEL*]

INTERFACE_PORT is listed first when present. Other parameters can be listed in any order.

Parameters

- ***INTERFACE_PORT*** interfaces for which information is displayed. Options include:
  - <no parameter> all interfaces in channel groups.
  - **ethernet** *e_num* Ethernet interface specified by *e_num*.
  - **port-channel** *p_num* port channel interface specified by *p_num*.
- ***PORT_LEVEL*** ports displayed, in terms of aggregation status. Options include:
  - <no parameter> command lists data for ports bundled by LACP into the aggregate.
  - **all-ports** command lists data for all ports, including LACP candidates that are not bundled.
- ***INFO_LEVEL*** amount of information that is displayed. Options include:
  - <no parameter> displays same information as **brief** option.
  - **brief** displays LACP configuration data, including sys-id, actor, priorities, and keys.
  - **detailed** includes **brief** option information plus state machine data.

Example

- This command displays LACP configuration information for all ethernet interfaces.

```
switch>show lacp interface
State: A = Active, P = Passive; S=ShortTimeout, L=LongTimeout;
       G = Aggregable, I = Individual; s+=InSync, s-=OutOfSync;
       C = Collecting, X = state machine expired,
       D = Distributing, d = default neighbor state
             |                       Partner                          Actor
Port Status  | Sys-id                 Port# State   OperKey PortPri  Port#
-------------------------------------------------------------------------------
Port Channel Port-Channel1:
Et43 Bundled | 8000,00-1c-73-09-a0-f3    43 ALGs+CD  0x0001   32768     43
Et44 Bundled | 8000,00-1c-73-09-a0-f3    44 ALGs+CD  0x0001   32768     44
Et45 Bundled | 8000,00-1c-73-09-a0-f3    45 ALGs+CD  0x0001   32768     45
Et46 Bundled | 8000,00-1c-73-09-a0-f3    46 ALGs+CD  0x0001   32768     46
Port Channel Port-Channel2:
Et47 Bundled | 8000,00-1c-73-01-02-1e    23 ALGs+CD  0x0002   32768     47
Et48 Bundled | 8000,00-1c-73-01-02-1e    24 ALGs+CD  0x0002   32768     48

             |                Actor
Port Status  |   State        OperKey    PortPriority
-------------------------------------------------------
Port Channel Port-Channel1:
Et43 Bundled |   ALGs+CD       0x0001          32768
Et44 Bundled |   ALGs+CD       0x0001          32768
Et45 Bundled |   ALGs+CD       0x0001          32768
Et46 Bundled |   ALGs+CD       0x0001          32768
Port Channel Port-Channel2:
Et47 Bundled |   ALGs+CD       0x0002          32768
Et48 Bundled |   ALGs+CD       0x0002          32768

switch>
```

show lacp internal

The **show lacp internal** command displays the local LACP state for all specified channels. Local state data includes the state machines and LACP protocol information.

Command Mode

EXEC

Command Syntax

show lacp [*PORT_LIST*] internal [*PORT_LEVEL*] [*INFO_LEVEL*]

Parameters

- ***PORT_LIST*** interface for which port information is displayed. Options include:
  - <no parameter> all configured port channels
  - *c_range* ports in specified channel list (number, number range, or list of numbers and ranges).
  - **interface** ports on all interfaces.
  - **interface ethernet** *e_num* Ethernet interface specified by *e_num*.
  - **interface port-channel** *p_num* port channel interface specified by *p_num*.
- ***PORT_LEVEL*** ports displayed, in terms of aggregation status. Options include:
  - <no parameter> command lists data for ports bundled by LACP into an aggregate.
  - **all-ports** command lists data for all ports, including LACP candidates that are not bundled.
- ***INFO_LEVEL*** amount of information that is displayed. Options include:
  - <no parameter> displays same information as **brief** option.
  - **brief** displays LACP configuration data, including sys-id, actor, priorities, and keys.
  - **detailed** includes **brief** option information plus state machine data.

PORT_LEVEL and ***INFO_LEVEL*** parameters can be placed in any order.

Example

- This command displays internal data for all configured port channels.

```
switch>show lacp internal
LACP System-identifier: 8000,00-1c-73-04-36-d7
State: A = Active, P = Passive; S=ShortTimeout, L=LongTimeout;
       G = Aggregable, I = Individual; s+=InSync, s-=OutOfSync;
       C = Collecting, X = state machine expired,
       D = Distributing, d = default neighbor state
             |Partner                                Actor
Port Status  | Sys-id                Port#  State    OperKey  PortPriority
-------------------------------------------------------------------------------
Port Channel Port-Channel1:
Et43 Bundled | 8000,00-1c-73-09-a0-f3    43  ALGs+CD   0x0001         32768
Et44 Bundled | 8000,00-1c-73-09-a0-f3    44  ALGs+CD   0x0001         32768
Et45 Bundled | 8000,00-1c-73-09-a0-f3    45  ALGs+CD   0x0001         32768
Et46 Bundled | 8000,00-1c-73-09-a0-f3    46  ALGs+CD   0x0001         32768
```

show lacp neighbor

The **show lacp neighbor** command displays the LACP protocol state of the remote neighbor for all specified port channels.

Command Mode

EXEC

Command Syntax

show lacp [*PORT_LIST*] neighbor [*PORT_LEVEL*] [*INFO_LEVEL*]

PORT_LEVEL and ***INFO_LEVEL*** parameters can be placed in any order.

Parameters

- ***PORT_LIST*** interface for which port information is displayed. Options include:
  - <no parameter> displays information for all configured port channels
  - *c_range* ports in specified channel list (number, number range, or list of numbers and ranges).
  - **interface** ports on all interfaces.
  - **interface ethernet** *e_num* Ethernet interface specified by *e_num*.
  - **interface port-channel** *p_num* port channel interface specified by *p_num*.
- ***PORT_LEVEL*** ports displayed, in terms of aggregation status. Options include:
  - <no parameter> command lists data for ports bundled by LACP into an aggregate.
  - **all-ports** command lists data for all ports, including LACP candidates that are not bundled.
- ***INFO_LEVEL*** amount of information that is displayed. Options include:
  - <no parameter> displays same information as **brief** option.
  - **brief** displays LACP configuration data, including sys-id, actor, priorities, and keys.
  - **detailed** includes **brief** option information plus state machine data.

Example

- This command displays the LACP protocol state of the remote neighbor for all port channels.

```
switch>show lacp neighbor
State: A = Active, P = Passive; S=ShortTimeout, L=LongTimeout;
       G = Aggregable, I = Individual; s+=InSync, s-=OutOfSync;
       C = Collecting, X = state machine expired,
       D = Distributing, d = default neighbor state
              |                          Partner
Port   Status  | Sys-id                  Port#   State     OperKey  PortPri
-------------------------------------------------------------------------------
Port Channel Port-Channel1:
Et1    Bundled | 8000,00-1c-73-00-13-19      1   ALGs+CD    0x0001    32768
Et2    Bundled | 8000,00-1c-73-00-13-19      2   ALGs+CD    0x0001    32768
Port Channel Port-Channel2:
Et23   Bundled | 8000,00-1c-73-04-36-d7     47   ALGs+CD    0x0002    32768
Et24   Bundled | 8000,00-1c-73-04-36-d7     48   ALGs+CD    0x0002    32768
Port Channel Port-Channel4*:
Et3    Bundled | 8000,00-1c-73-0b-a8-0e     45   ALGs+CD    0x0001    32768
Et4    Bundled | 8000,00-1c-73-0b-a8-0e     46   ALGs+CD    0x0001    32768
Port Channel Port-Channel5*:
Et19   Bundled | 8000,00-1c-73-0c-30-09     49   ALGs+CD    0x0005    32768
Et20   Bundled | 8000,00-1c-73-0c-30-09     50   ALGs+CD    0x0005    32768
Port Channel Port-Channel6*:
Et6    Bundled | 8000,00-1c-73-01-07-b9     49   ALGs+CD    0x0001    32768
Port Channel Port-Channel7*:
Et5    Bundled | 8000,00-1c-73-0f-6b-22     51   ALGs+CD    0x0001    32768
Port Channel Port-Channel8*:
Et10   Bundled | 8000,00-1c-73-10-40-fa     51   ALGs+CD    0x0001    32768

* - Only local interfaces for MLAGs are displayed. Connect to the peer to
    see the state for peer interfaces.
switch>
```

show lacp sys-id

The **show lacp sys-id** command displays the System Identifier the switch uses when negotiating remote LACP implementations.

Command Mode

EXEC

Command Syntax

show lacp sys-id [*INFO_LEVEL*]

Parameters

- ***INFO_LEVEL*** amount of information that is displayed. Options include:
  - <no parameter> displays system identifier
  - **brief** displays system identifier.
  - **detailed** displays system identifier and system priority, including the MAC address.

Examples

- This command displays the system identifier.

```
switch>show lacp sys-id brief
8000,00-1c-73-04-36-d7
```

- This command displays the system identifier and system priority.

```
switch>show lacp sys-id detailed
System Identifier used by LACP:
System priority: 32768  Switch MAC Address: 00:1c:73:04:36:d7
  802.11.43 representation: 8000,00-1c-73-04-36-d7
```

show load-balance profile

The **show load-balance profile** command displays the contents of the specified load balance profiles. Load balance profiles specify parameters used by hashing algorithms that distribute traffic across ports comprising a port channel or among component ECMP routes.

Command Mode

EXEC

Command Syntax

show load-balance profile [*PROFILES*]

Parameters

- ***PROFILES*** Load balance profiles for which command displays contents. Options include:
  - <no parameter> displays all load balance profiles.
  - *profile_name* displays specified profile.

Related Commands

- **load-balance policies** places the switch in load-balance-policies configuration mode.
- **ingress load-balance profile** applies a load-balance profile to an Ethernet or port channel interface.

Example

- This command displays the contents of the LB-1 load balance profile.

```
switch>show load-balance profile LB-1

---------- LB-1 ----------

Source MAC address hashing               ON
Destination MAC address hashing          ON
Ethernet type hashing                    ON
VLAN ID hashing                          ON
VLAN priority hashing                    ON
IP source address hashing                ON
IP destination address hashing           ON
TCP/UDP source port hashing              ON
TCP/UDP destination port hashing         ON
IP protocol field hashing                ON
DSCP field hashing is                    ON
Symmetric hashing for non-IP packets     OFF
Symmetric hashing for IP packets         OFF
Random distribution for port-channel     ON
Random distribution for ecmp             ON

Profile LB-1 is applied on the following
    Port-Channel100
switch>
```

show port-channel

The **show port-channel** command displays information about members the specified port channels.

Command Mode

EXEC

Command Syntax

show port-channel [*MEMBERS*] [*PORT_LIST*] [*INFO_LEVEL*]

Parameters

- ***MEMBERS*** list of port channels for which information is displayed. Options include:
  - <no parameter> all configured port channels.
  - *p_range* ports in specified channel list (number, number range, or list of numbers and ranges).
- ***PORT_LEVEL*** ports displayed, in terms of aggregation status. Options include:
  - <no parameter> Displays information on ports that are active members of the LAG**.**
  - **active-ports** Displays information on ports that are active members of the LAG.
  - **all-ports** Displays information on all ports (active or inactive) configured for LAG.
- ***INFO_LEVEL*** amount of information that is displayed. Options include:
  - <no parameter> Displays information at the brief level.
  - **brief** Displays information at the brief level.
  - **detailed** Displays information at the detail level.

Display Values

- **Port Channel** Type and name of the port channel.
- **Time became active** Time when the port channel came up.
- **Protocol** Protocol operating on the port channel.
- **Mode** Status of the Ethernet interface on the port. The status value is Active or Inactive.
- **No active ports** Number of active ports on the port channel.
- **Configured but inactive ports** Ports configured but that are not actively up.
- **Reason unconfigured** Reason why the port is not part of the LAG.

Guidelines

The **show etherchannel** and **show port-channel** commands are identical.

You can configure a port channel to contain many ports, but only a subset may be active at a time. All active ports in a port channel must be compatible. Compatibility includes many factors and is platform specific. For example, compatibility may require identical operating parameters such as speed and maximum transmission unit (MTU). Compatibility may only be possible between specific ports because of the internal organization of the switch.

Examples

- This command displays output from the **show port-channel** command:

```
switch>show port-channel 3
Port Channel Port-Channel3:
  Active Ports:
       Port               Time became active       Protocol    Mode
    ------------------------------------------------------------------------
       Ethernet3          15:33:41                 LACP        Active
       PeerEthernet3      15:33:41                 LACP        Active
```

- This command displays output from the **show port-channel active-ports** command:

```
switch>show port-channel active-ports
Port Channel Port-Channel3:
  No Active Ports
Port Channel Port-Channel11:
  No Active Ports
switch>
```

- This command displays output from the **show port-channel all-ports** command:

```
switch>show port-channel all-ports
Port Channel Port-Channel3:
  No Active Ports
  Configured, but inactive ports:
      Port            Time became inactive    Reason unconfigured
    ------------------------------------------------------------------------
      Ethernet3       Always                  not compatible with aggregate

Port Channel Port-Channel11:
  No Active Ports
  Configured, but inactive ports:
      Port            Time became inactive    Reason unconfigured
    ------------------------------------------------------------------------
      Ethernet25      Always                  not compatible with aggregate
      Ethernet26      Always                  not compatible with aggregate

switch>
```

show port-channel limits

The **show port-channel limits** command displays groups of ports that are compatible and may be joined into port channels. Each group of compatible ports is called a LAG group. For each LAG group, the command also displays ***Max interfaces*** and ***Max ports per interface***.

- ***Max interfaces*** defines the maximum number of active port channels that may be formed out of these ports.
- ***Max ports per interface*** defines the maximum number of active ports allowed in a port channel from the compatibility group.

All active ports in a port channel must be compatible. Compatibility comprises many factors and is specific to a given platform. For example, compatibility may require identical operating parameters such as speed and/or maximum transmission unit (MTU). Compatibility may only be possible between specific ports because of internal organization of the switch.

Command Mode

EXEC

Command Syntax

```
show port-channel limits
```

Example

- This command displays **show port-channel list** output:

```
switch>show port-channel limits
LAG Group: focalpoint
------------------------------------------------------------------------
  Max port-channels per group: 24, Max ports per port-channel: 16
  24 compatible ports: Ethernet1  Ethernet2  Ethernet3  Ethernet4
                       Ethernet5  Ethernet6  Ethernet7  Ethernet8
                       Ethernet9  Ethernet10 Ethernet11 Ethernet12
                       Ethernet13 Ethernet14 Ethernet15 Ethernet16
                       Ethernet17 Ethernet18 Ethernet19 Ethernet20
                       Ethernet21 Ethernet22 Ethernet23 Ethernet24
------------------------------------------------------------------------

switch>
```

show port-channel load-balance fields

The **show port-channel load-balance fields** command displays the fields that the hashing algorithm uses to distribute traffic across the interfaces that comprise the port channels.

Command Mode

EXEC

Command Syntax

show port-channel load-balance *HARDWARE* fields

Parameters

- ***HARDWARE*** ASIC switching device. Selection options depend on the switch model and include:
  - **arad**
  - **fm6000**
  - **petraA**
  - **trident**

Examples

- This command displays the hashing fields used for balancing port channel traffic.

```
switch>show port-channel load-balance fm6000 fields
Source MAC address hashing for non-IP packets is ON
Destination MAC address hashing for non-IP packets is ON
Ethernet type hashing for non-IP packets is ON
VLAN ID hashing for non-IP packets is ON
VLAN priority hashing for non-IP packets is ON
Source MAC address hashing for IP packets is ON
Destination MAC address hashing for IP packets is ON
Ethernet type hashing for IP packets is ON
VLAN ID hashing for IP packets is ON
VLAN priority hashing for IP packets is ON
IP source address hashing is ON
IP destination address hashing is ON
IP protocol field hashing is ON
TCP/UDP source port hashing is ON
TCP/UDP destination port hashing is ON
switch>
```

show port-channel summary

The **show port-channel summary** command displays the port-channels on the switch and lists their component interfaces, LACP status, and set flags**.**

Command Mode

EXEC

Command Syntax

show port-channel summary

Examples

- This command displays **show port-channel summary** output:

```
switch>show port-channel summary

    Flags
--------------------------------------------------------------------------------
a - LACP Active            p - LACP Passive
U - In Use                 D - Down
+ - In-Sync                - - Out-of-Sync          i - incompatible with agg
P - bundled in Po          s - suspended            G - Aggregable
I - Individual             S - ShortTimeout         w - wait for agg

Number of channels in use: 2
Number of aggregators:2

   Port-Channel       Protocol    Ports
------------------------------------------------------
   Po1(U)             LACP(a)      Et47(PG+) Et48(PG+)
   Po2(U)             LACP(a)      Et39(PG+) Et40(PG+)
```

show port-channel traffic

The **show port-channel traffic** command displays the traffic distribution between the member ports of the specified port channels. The command displays distribution for unicast, multicast, and broadcast streams.

Command Mode

EXEC

Command Syntax

show port-channel [*MEMBERS*] traffic

Parameters

- ***MEMBERS*** list of port channels for which information is displayed. Options include:
  - <no parameter> all configured port channels.
  - *c_range* ports in specified channel list (number, number range, or list of numbers and ranges).

Examples

- This command displays traffic distribution for all configured port channels.

```
switch>show port-channel traffic
ChanId      Port Rx-Ucst Tx-Ucst Rx-Mcst Tx-Mcst Rx-Bcst Tx-Bcst
------ --------- ------- ------- ------- ------- ------- -------
     8      Et10 100.00% 100.00% 100.00% 100.00%   0.00% 100.00%
------ --------- ------- ------- ------- ------- ------- -------
     1       Et1  13.97%  42.37%  47.71%  30.94%   0.43%  99.84%
     1       Et2  86.03%  57.63%  52.29%  69.06%  99.57%   0.16%
------ --------- ------- ------- ------- ------- ------- -------
     2      Et23  48.27%  50.71%  26.79%  73.22%   0.00% 100.00%
     2      Et24  51.73%  49.29%  73.21%  26.78%   0.00%   0.00%
------ --------- ------- ------- ------- ------- ------- -------
     4       Et3  55.97%  63.29%  51.32%  73.49%   0.00%   0.00%
     4       Et4  44.03%  36.71%  48.68%  26.51%   0.00%   0.00%
------ --------- ------- ------- ------- ------- ------- -------
     5      Et19  39.64%  37.71%  50.00%  90.71%   0.00%   0.00%
     5      Et20  60.36%  62.29%  50.00%   9.29%   0.00% 100.00%
------ --------- ------- ------- ------- ------- ------- -------
     6       Et6 100.00% 100.00% 100.00% 100.00%   0.00% 100.00%
------ --------- ------- ------- ------- ------- ------- -------
     7       Et5 100.00%   0.00% 100.00% 100.00%   0.00%   0.00%
switch>
```

Chapter 12

Multi-Chassis Link Aggregation

Arista switches support Multi-Chassis Link Aggregation (MLAG) to logically aggregate ports across two switches. For example, two 10-gigabit Ethernet ports, one each from two MLAG configured switches, can connect to two 10-gigabit ports on a host, switch, or network device to create a link that appears as a single 20-gigabit port. MLAG-configured ports provide Layer 2 multipathing, increased bandwidth, higher availability, and other improvements on traditional active-passive or Spanning Tree governed infrastructures.

The Multi-Chassis Link Aggregation chapter contains these sections:

- Section 12.1: MLAG Introduction
- Section 12.2: MLAG Conceptual Overview
- Section 12.3: MLAG Maintenance
- Section 12.4: Configuring MLAG
- Section 12.5: MLAG Implementation Example
- Section 12.6: MLAG Commands

12.1 MLAG Introduction

High availability data center topologies typically provide redundancy protection at the expense of oversubscription by connecting top-of-rack (TOR) switches and servers to dual aggregation switches. In these topologies, Spanning Tree Protocol prevents network loops by blocking half of the links to the aggregation switches. This reduces the available bandwidth by 50%.

Deploying MLAG removes oversubscription by configuring an MLAG link between two aggregation switches to create a single logical switching instance that utilizes all connections to the switches. Interfaces on both devices participate in a distributed port channel, enabling all active paths to carry data traffic while maintaining the integrity of the Spanning Tree topology.

MLAG provides these benefits:

- Aggregates multiple Ethernet ports across two switches.
- Provides higher bandwidth links as network traffic increases.
- Utilizes bandwidth more efficiently with fewer links blocked by STP.
- Connects to other switches and servers by static LAG or LACP without other proprietary protocols.
- Supports normal STP operation to prevent loops.
- Supports active-active Layer-2 redundancy.

Note PTP (precision timing protocol) is not supported with MLAG.

12.2 MLAG Conceptual Overview

12.2.1 MLAG Operation Process

A multi-chassis link aggregation group (MLAG) is a pair of links that terminate on two cooperating switches and appear as an ordinary link aggregation group (LAG). The cooperating switches are MLAG peer switches and communicate through an interface called a peer link. While the peer link's primary purpose is exchanging MLAG control information between peer switches, it also carries data traffic from devices that are attached to only one MLAG peer and have no alternative path. An MLAG domain consists of the peer switches and the control links that connect the switches.

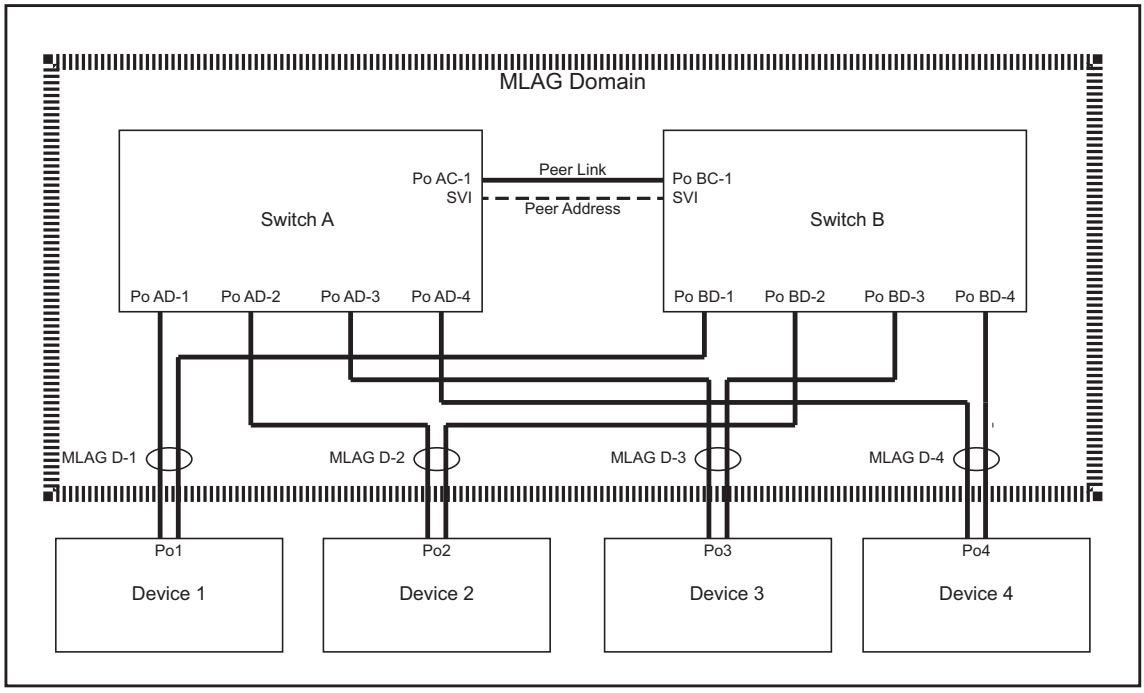
In Figure 12-1, Switch A and Switch B are peer switches in the MLAG domain and connect to each other through the peer link. Each peer switch uses the peer address to form and maintain the peer link.

The MLAG domain ID is a text string configured in each peer switch. MLAG switches use this string to identify their peers. The MLAG System ID (MSI) is the MLAG domain's MAC address. The MSI is automatically derived when the MLAG forms and does not match the bridge MAC address of either peer. Each peer uses the MSI in STP and LACP PDUs.

The topology in Figure 12-1 contains four MLAGs: one MLAG connects each device to the MLAG domain. Each peer switch connects to the four servers through MLAG link interfaces.

In a conventional topology, with dually-attaching devices to multiple switches for redundancy, Spanning Tree Protocol (STP) blocks half of the switch-device links. In the MLAG topology, STP does not block any portion because it views the MLAG Domain as a single switch and each MLAG as a single port. The MLAG protocol facilitates the balancing of device traffic between the peer switches.

Figure 12-1: MLAG Domain Topology



When MLAG is disabled, peer switches revert to their independent state. MLAG is disabled by any of the following:

- MLAG configuration changes.

- The TCP connection breaks.
- The peer-link or local-interface goes down.
- A switch does not receive a response to a keep alive message from its peer within a specified period.

12.2.2 MLAG Interoperability with Other Features

The following sections describe MLAG interaction with other switch features.

12.2.2.1 VLANs

VLAN parameters must be configured identically on each peer for the LAGs comprising the peer link and MLAGs. These parameters include the switchport access VLAN, switchport mode, trunk-allowed VLANs, the trunk native VLAN, and switchport trunk groups. Configuration discrepancies may result in traffic loss in certain failure scenarios. Port-specific bridging configuration originates on the switch where the port is physically located.

12.2.2.2 LACP

Link Aggregation Control Protocol (LACP) should be used on all MLAG interfaces, including the peer-link. LACP control packets reference the MLAG system ID.

12.2.2.3 Static MAC Addresses

A static MAC address configured on an MLAG interface is automatically configured on the peer's corresponding interface. Configuring static MAC addresses on both peers prevents undesired flooding if an MLAG peer relationship fails.

If the MLAG peer relationship is broken or if all local members of an MLAG port channel go down, the peer is no longer automatically configured with the static MAC address.

12.2.2.4 STP

When implementing MLAG in a spanning tree network, spanning tree must be configured globally and on port-channels configured with an MLAG ID. Port specific spanning tree configuration comes from the switch where the port physically resides. This includes spanning-tree PortFast BPDU Guard and BPDU filter.

12.2.2.5 Port Mirroring

A port channel which is a member of an MLAG *must not* be used as the destination port for a port mirroring (port monitoring) session.

12.3 MLAG Maintenance

These sections describe tasks required for MLAG to operate on the switch:

- Section 12.3.1: Ensuring Control Plane ACL Compatibility
- Section 12.3.2: MLAG Availability through a Single Functional Peer
- Section 12.3.3: Upgrading MLAG Peers

12.3.1 Ensuring Control Plane ACL Compatibility

The control plane access control list (ACL) on any interface participating in the MLAG must be configured to allow only the peer link neighbor to generate MLAG control traffic. The required rules are included in the default control plane ACL for Ethernet ports.

Any custom control plane ACL applied to a participating port must include these two rules:

```
permit tcp any any eq mlag ttl eq 255
permit udp any any eq mlag ttl eq 255
```

MLAG peers that function as routers must each have routing enabled.

12.3.2 MLAG Availability through a Single Functional Peer

MLAG high availability advantages are fully realized when all devices that connect to one MLAG switch also connect to the peer switch. A switch can continue supporting MLAG when its peer is offline if the STP agent is restartable. When one peer is offline, data traffic flows from the devices through the MLAG component link that connects to the functioning switch. When a switch is offline, its interfaces and ports do not appear in **show mlag** and ***show spanning tree protocol*** commands of the functioning peer.

To view the restartability status of the STP agent, issue the **show spanning-tree bridge detail** command:

```
switch-1#show spanning-tree bridge detail | grep agent
Stp agent is restartable
```

STP agent restartability requires consistent configuration between the peers of STP, LACP, MLAG, and switchport parameters. Events triggering an STP state machine change may also briefly prevent the STP agent from being restartable.

12.3.2.1 Reload Delay

If an MLAG peer reboots, all ports except those in the peer-link port-channel remain in ***errdisabled*** state for a specified time, called the reload-delay period. This period allows all topology states to stabilize before the switch begins forwarding traffic. Each Arista switch defaults to the recommended reload-delay value, which varies by switch platform:

- **fixed configuration switches**: 300 seconds
- **Trident-2 modular switches**: 900 seconds

  7304
  7308
  7316
  7300X series

- **Sand platform modular switches**: 1800 seconds

  7504
  7508
  7500E series
  7548S

In those cases where network topology requires additional time to stabilize or where a shorter delay can be tolerated, the reload-delay period can be configured using the **reload-delay mlag** command.

Severing the physical connection (cable) that establishes the peer-link between MLAG peers may result in a ***split brain*** state where each peer independently enters spanning tree state to prevent topology loops. Sessions established through one interface of a dual attached device may fail if its path is disrupted by the STP reconvergence, possibly resulting in temporarily lost connectivity. Sessions can be reestablished if permitted by the resulting topology.

12.3.3 Upgrading MLAG Peers

MLAG ISSU (In-Service Software Upgrade) upgrades EOS software on one MLAG peer with minimal traffic disruptions on active MLAG interfaces and without changing the network topology.

12.3.3.1 Verifying Configuration Compatibility

A seamless EOS upgrade on an MLAG peer requires that the following features are configured consistently on each switch:

- VLANs
- Switchport configuration on port channel interfaces that are configured with an MLAG ID
- STP configuration (global)

12.3.3.2 Version Compatibility

A switch running MLAG can be upgraded without disrupting MLAG traffic when the upgrade EOS version is compatible with the version on the peer switch. Refer to the Release Notes for a list of compatible EOS versions.

12.3.3.3 Reload Warning Conditions

Entering an EOS reload command while MLAG is active generates warning messages if conditions that can result in packet loss during the upgrade are present. All warnings should be resolved before confirming the reload request. Table 12-1 displays the reload conditions and a common resolution method for each condition..

Table 12-1 Reload Warning Resolutions

Reload Condition	Resolution Method
Compatibility check	Refer to the Release Notes to verify that the new version is compatible with the currently installed version
Active-partial MLAG warning	Bring up the remote port-channel. If the MLAG is not actively used, then this warning can be ignored.
STP is not restartable	Wait for STP to be restartable: typically 30 seconds, up to 120 seconds for a newly started STP agent. Refer to Section 12.3.2 for information on checking restartability.
Reload delay too low	Configure a reload delay value greater than or equal to the default. Recommend delay is 300 seconds for TOR switches and 900 seconds for modulars.
Peer has error-disabled interfaces	Wait for reload-delay to expire on the peer.

Example

- The following **reload** command generates MLAG warning conditions that should be addressed before confirming the ***proceed with reload*** prompt.

```
switch(config)#reload
If you are performing an upgrade, and the Release Notes for the
new version of EOS indicate that MLAG is not backwards-compatible
with the currently installed version (4.9.2), the upgrade will
result in packet loss.

The following MLAGs are not in Active mode. Traffic to or from these ports will
be lost during the upgrade process.
                                                                    local/remote
  mlag      desc               state      local      remote          status
--------- --------- ------------------- ---------- ---------- ------------
    14               active-partial       Po14       Po14         up/down
    15               active-partial       Po15       Po15         up/down

Stp is not restartable. Topology changes will occur during the upgrade process.

The configured reload delay of 100 seconds is below the default
value of 300 seconds. A longer reload delay allows more time to
rollback an unsuccessful upgrade due to incompatibility.

The other MLAG peer has errdisabled interfaces. Traffic loss will occur during
the upgrade process.

Proceed with reload? [confirm]
```

12.3.3.4 Performing an MLAG ISSU Upgrade

The following procedure performs an MLAG ISSU upgrade:

Step 1 Very configuration consistency on each peer (Section 12.3.3.1).

Step 2 Verify version compatibility between the new and existing images (Section 12.3.3.2).

Step 3 Configure **reload-delay mlag** (Section 12.3.2) if needed. Recommended delay period varies by switch type, and each switch defaults to its recommended delay period.

Step 4 Install the new image onto one of the peers:

a Upload the new image to the switch.

b Set the boot path to the new image.

c Enter the **reload** command.

Step 5 Resolve all reload warnings .

Step 6 Confirm the reload.

Step 7 Wait for MLAG peers to renegotiate to the active state and reload-delay expiry on rebooted peer; until reload-delay period has expired, ports on the rebooted peer (except the peer-link) will be in ***errdisabled*** state with err-disabled reason being ***mlag-issu***.

Avoid configuration changes on both peers until after this step.

Step 8 Repeat the upgrade process for the other peer.

When upgrading modular switches with dual supervisors, upgrade the standby supervisors first, then upgrade the active supervisors.

12.4 Configuring MLAG

These sections describe the basic MLAG configuration steps:

- Section 12.4.1: Configuring the MLAG Peers
- Section 12.4.2: Configuring MLAG Services

12.4.1 Configuring the MLAG Peers

Connecting two switches as MLAG peers requires the establishment of the peer link and an SVI that defines local and peer IP addresses on each switch.

The peer link is composed of a LAG between the switches. When all devices that connect to the MLAG domain are dually connected to the switches through an MLAG, a peer link of two Ethernet interfaces is sufficient to handle MLAG control data and provide N+1 redundancy. When the domain connects to devices through only one MLAG peer, the peer link may require additional Ethernet interfaces to manage data traffic.

Disruptions to peer link connectivity due to forwarding agent restarts may cause an extended MLAG outage. Forwarding agent restart event include some configuration changes, such as port speed change or UFT mode change). The following precautions can reduce the risk of losing peer-link connectivity:

- all switches: constructing peer-links from port-channels in preference to a single Ethernet interface.
- modular systems: peer-link port-channel members should span multiple line cards.
- multi-chip systems: peer-link port-channel member should span multiple chips.

Section 3.6 describes modular systems.

The steps that configure two switches as MLAG peers include:

- Configuring the Port Channels, VLAN Interfaces, and IP addresses
- Configure Peer Parameters
- Configuring MLAG Peer Gateway

12.4.1.1 Configuring the Port Channels, VLAN Interfaces, and IP addresses

The peer link is a normal port channel. The local address is the SVI that maps to the peer link port channel. The port channel and SVI must be configured on each peer switch. The port channel should be an active LACP port. The local and peer addresses must be located on the same IP address subnet. Autostate should be disabled on the SVI configured as the local interface.

Examples

- These commands create an active mode LACP port channel interface from two Ethernet interfaces and configure it as part of a trunk group on each switch.

  The **switchport mode trunk** command permits all VLANs on the interface by default, so all VLANs are permitted on port channel 10 in the following example. The configuration of a trunk group for a VLAN restricts only that specific VLAN to the associated ports: VLAN 4094 is only permitted on port channel 10, and not on any other ports on the switch. It is important to remember that all VLANs must be permitted between the peers on the peer link for correct operation.

Switch 1

```
switch1#config
switch1(config)#vlan 4094
switch1(config-vlan-4094)#trunk group m1peer
switch1#config
switch1(config)#interface ethernet 1-2
switch1(config-if-et1-2)#channel-group 10 mode active
switch1(config-if-et1-2)#interface port-channel 10
switch1(config-if-po10)#switchport mode trunk
switch1(config-if-po10)#switchport trunk group m1peer
switch1(config-if-po10)#exit
switch1(config)#
```

Switch 2

```
switch2#config
switch2(config)#vlan 4094
switch2(config-vlan-4094)#trunk group m1peer
switch2(config-vlan-4094)#exit
switch2(config)#interface ethernet 1-2
switch2(config-if-et1-2)#channel-group 10 mode active
switch2(config-if-et1-2)#interface port-channel 10
switch2(config-if-po10)#switchport mode trunk
switch2(config-if-po10)#switchport trunk group m1peer
switch2(config-if-po10)#exit
switch2(config)#
```

- These commands create an SVI for the local interface and associate it to the trunk group assigned to the peer link port channel.

  The SVI creates a Layer 3 endpoint in the switch and enables MLAG processes to communicate via TCP. The IP address can be any unicast address that does not conflict with other SVIs. STP is disabled for the peer link VLAN 4094 to prevent any potential STP disruption of inter peer communications. Recall that the VLAN has been restricted to port-channel 10 by the earlier trunk group configuration thus preventing potential Layer 2 loop conditions within VLAN 4094.

  Switch 1

  ```
  switch1#config
  switch1(config)#interface vlan 4094
  switch1(config-if-vl4094)#ip address 10.0.0.1/30
  switch1(config-if-vl4094)#no autostate
  switch1(config-if-vl4094)#exit
  switch1(config)#no spanning-tree vlan 4094
  switch1(config)#
  ```

  Switch 2

  ```
  switch2#config
  switch2(config)#interface vlan 4094
  switch2(config-if-vl4094)#ip address 10.0.0.2/30
  switch2(config-if-vl4094)#no autostate
  switch2(config-if-vl4094)#exit
  switch2(config)#no spanning-tree vlan 4094
  switch2(config)#
  ```

12.4.1.2 Configure Peer Parameters

Peer connection parameters configure the connection between the MLAG peer switches. This section describes the following peer configuration parameters.

- MLAG Configuration Mode

- Local VLAN Interface
- Peer Address
- Peer Link
- Domain ID
- Heartbeat Interval and Timeout
- Reload Delay Period

MLAG Configuration Mode

Peer connection parameters are configured in MLAG-configuration mode. The **mlag configuration (global configuration)** command places the switch in MLAG configuration mode.

Example

- This command places the switch in MLAG configuration mode.

```
switch(config)#mlag configuration
switch(config-mlag)#
```

Local VLAN Interface

The local interface specifies the SVI upon which the switch sends MLAG control traffic. The local IP address is specified within the definition of the VLAN associated with the local interface. The Peer Address configures the control traffic destination on the peer switch.

The **local-interface** command specifies a VLAN interface as the peer link SVI.

Example

- This command configures VLAN 4094 as the local interface.

```
switch(config-mlag)#local-interface vlan 4094
switch(config-mlag)#
```

Peer Address

The peer address is the destination address on the peer switch for MLAG control traffic. If the peer IP address is unreachable, MLAG peering fails and both peer switches revert to their independent state.

The **peer-address** command specifies the peer address.

Example

- This command configures a peer address of 10.0.0.2.

```
switch(config-mlag)#peer-address 10.0.0.2
switch(config-mlag)#
```

Peer Link

An MLAG is formed by connecting two switches through an interface called a peer link. The peer link carries MLAG advertisements, keepalive messages, and data traffic between the switches. This information keeps the two switches working together as one. While interfaces comprising the peer links on each switch must be compatible, they need not use the same interface number. Ethernet and Port-channel interfaces can be configured as peer links.

The **peer-link** command specifies the interface the switch uses to communicates MLAG control traffic.

Example

- This command configures port-channel 10 as the peer link.

```
switch(config-mlag)#peer-link port-channel 10
switch(config-mlag)#
```

Domain ID

The MLAG domain ID is a unique identifier for an MLAG domain. The MLAG domain ID must be the identical on each switch to facilitate MLAG communication.

The **domain-id** command configures the MLAG domain ID.

Example

- This command configures ***mlagDomain*** as the domain ID:

```
switch(config-mlag)#domain-id mlagDomain
switch(config-mlag)#
```

Heartbeat Interval and Timeout

The heartbeat interval specifies the period between the transmission of successive keepalive messages. Each MLAG switch transmits keepalive messages and monitors message reception from its peer. The heartbeat timeout is reset when the switch receives a keepalive message. If the heartbeat timeout expires, the switch disables MLAG under the premise that the peer switch is not functioning.

The **heartbeat-interval (MLAG)** command configures the heartbeat interval between 1 and 30 seconds, with a default value of 2 seconds. The heartbeat timeout expiry is 30 seconds.

Important! On 7500 and 7500E Series Switches, Arista recommends setting the heartbeat interval to 10 seconds.

Example

- This command configures the heartbeat interval as 2.5 seconds (2500 ms).

```
switch(config-mlag)#heartbeat-interval 2500
switch(config-mlag)#
```

Reload Delay Period

The reload delay period specifies the interval that non-peer links are disabled after an MLAG peer reboots. This interval allows non-peer links to learn multicast and OSPF states and synchronize ARP caches before the ports start handling traffic. Each Arista switch defaults to the recommended reload-delay value, which varies by switch platform

- fixed configuration switches: 300 seconds (five minutes)
- Trident-2 platform modular switches: 1200 seconds (twenty minutes)
- Sand platform modular switches: 1800 seconds (thirty minutes)

In those cases where network topology requires additional time to stabilize or where a shorter delay can be tolerated, the reload-delay period can be configured using the **reload-delay mlag** command.

Example

- This command configures the reload delay interval as 2.5 minutes (150 seconds).

```
switch(config-mlag)#reload-delay 150
switch(config-mlag)#
```

Shutdown

The **shutdown (MLAG)** command disables MLAG operations without disrupting the MLAG configuration. The **no mlag configuration** command (global configuration mode) disables MLAG and removes the MLAG configuration. The **no shutdown** command resumes MLAG activity.

Examples

- This command disables MLAG activity on the switch.

```
switch(config-mlag)#shutdown
switch(config-mlag)#
```

- This command resumes MLAG activity on the switch.

```
switch(config-mlag)#no shutdown
switch(config-mlag)#
```

12.4.1.3 Configuring MLAG Peer Gateway

In an MLAG setup, routing on a MLAG peer switch is possible using its own bridge system MAC, VARP MAC, or VRRP MAC. On a peer receiving an IP packet with destination MAC set to one of these MACs, a packet gets routed if its hardware has enough information to route the packet. Configuring sending traffic to a cached MAC involves routing the session table and MLAG peer traffic if packets are received with the MAC peer.

Examples

- This command enables the MLAG peer gateway.

```
switch(config)#ip virtual-router mac-address mlag-peer
switch1(config)#
```

- This command disables the MLAG peer gateway.

```
switch(config)#no ip virtual-router mac-address mlag-peer
switch1(config)#
```

12.4.1.4 Configuring Ingress Replication to LAGs

Hardware support for ingress replication to LAGs is enabled by default when the user configures ingress replication. When multicast traffic is sent over the LAG, the hardware uses its built-in algorithm, based on the L2/L3/L4 headers, to load balance traffic over ports in the LAG. When a port goes down in a LAG, the hardware quickly hashes the multicast traffic over the remaining ports in the LAG, resulting in fewer drops than software based LAG support.

Examples

- This command enables ingress replication.

```
switch(config)#platform sand multicast replication default ingress
switch(config)#
```

- This command configures the maximum members (within a range of 1 through 64) for ingress only replication in a multicast group.

```
switch(config)#platform sand multicast replication ingress maximum 32
switch(config)#
```

12.4.2 Configuring MLAG Services

An MLAG is a pair of links that originate on a network attached device and terminate on the two MLAG peer switches. The MLAG switches coordinate traffic to the device through a common **mlag (port-channel interface configuration)** command on the interfaces that connect to the device.

The MLAG ID differs from the MLAG domain ID. The MLAG domain ID is assigned globally per switch in MLAG configuration mode, and the same MLAG domain ID must be on both switches.

It is not recommended that MLAGs are used with static LAGs. Configure the downstream switch or router connected to the MLAG peers to negotiate a LAG with LACP. For Arista Networks switches, this is in respect to a configuration such as **channel-group** *group-number* **mode on**.

Port channels configured as an MLAG must have identical port channel numbers. Although the MLAG ID is a distinct parameter from the port channel number, best practices recommend assigning the MLAG ID to match the port channel number.

The following example does not follow this convention to emphasize the parameters that are distinct. The example in Section 12.5 follows the best practices convention.

Examples

- These Switch1 commands bundle Ethernet interfaces 3 and 4 in port channel 20, then associate that port channel with MLAG 12.

```
switch1(config)#interface ethernet 3-4
switch1(config-if-et3-4)#channel-group 20 mode active
switch1(config-if-et3-4)#interface port-channel 20
switch1(config-if-po20)#mlag 12
switch1(config-if-po20)#exit
switch1(config)#
```

- These Switch2 commands bundle Ethernet interfaces 9 and 10 in port channel 15, then associate that port channel with MLAG 12.

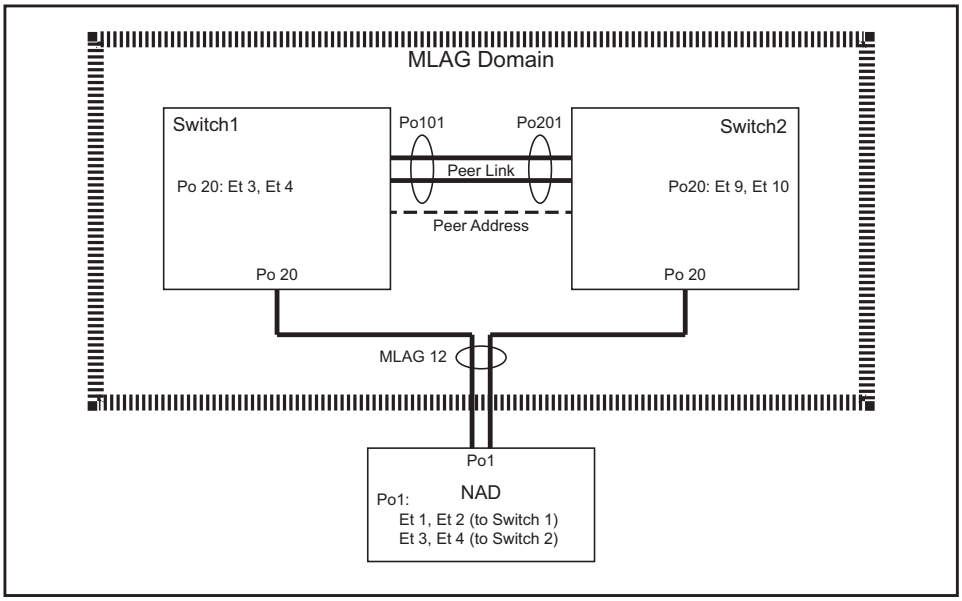
```
switch2(config)#interface ethernet 9-10
switch2(config-if-et9-10)#channel-group 15 mode active
switch2(config-if-et9-10)#interface port-channel 20
switch2(config-if-po20)#mlag 12
switch2(config-if-po20)#exit
switch2(config)#
```

- These commands configure the port channels that attach to the MLAG on network attached device:

```
NAD(config)#interface ethernet 1-4
NAD(config-if-Et1-4)#channel-group 1 mode active
NAD(config-if-Et1-4)#exit
NAD(config)#
```

Figure 12-2 displays the result of the interface MLAG configuration.

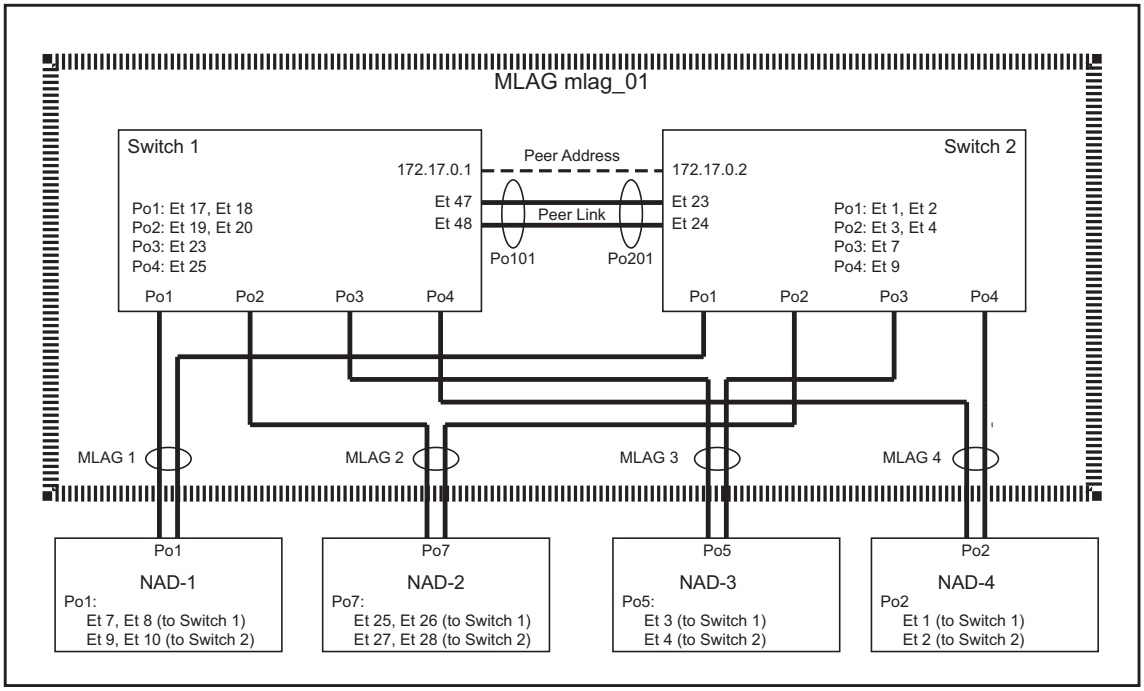
Figure 12-2: MLAG Interface Configuration



12.5 MLAG Implementation Example

This example creates an MLAG Domain, then configures MLAG connections between the peer switches and four Network Attached Devices (NADs). The MLAG switches connect through a LAG and communicate with the NADs through MLAGs. Although the NADs can be any device that supports LACP LAGs, the devices in this example are Arista switches.

Figure 12-3: MLAG Implementation Example



12.5.1 Topology

Figure 12-3 displays the MLAG topology. Switch 1 and Switch 2 are MLAG peers that logically represent a single Layer 2 switch. The peer link between the switches contains the following interfaces:

- Switch 1: Ethernet 47, Ethernet 48
- Switch 2: Ethernet 23, Ethernet 24

The example configures MLAGs from the MLAG Domain to four network attached devices (NAD-1, NAD-2, NAD-3, NAD-4).

12.5.2 Configuring the Peer Switch Connections

To configure the switches in the described topology, perform the tasks in these sections:

- Section 12.5.2.1: Configuring the Peer Switch Port Channels
- Section 12.5.2.2: Configuring the Peer Switch SVIs
- Section 12.5.2.3: Configuring the Peer Links

12.5.2.1 Configuring the Peer Switch Port Channels

These commands create the port channels the switches use to establish the peer link.

These commands create port channels on Switch1

```
switch1(config)#interface ethernet 47-48
switch1(config-if-et47-48)#channel-group 101 mode active
switch1(config-if-et47-48)#interface port-channel 101
switch1(config-if-po101)#switchport mode trunk
switch1(config-if-po101)#switchport trunk group peertrunk
switch1(config-if-po101)#exit
switch1(config)#
```

These commands create port channels on Switch2

```
switch2(config)#interface ethernet 23-24
switch2(config-if-et23-24)#channel-group 201 mode active
switch2(config-if-et23-24)#interface port-channel 201
switch2(config-if-po201)#switchport mode trunk
switch2(config-if-po201)#switchport trunk group trunkpeer
switch2(config-if-po201)#exit
switch2(config)#
```

12.5.2.2 Configuring the Peer Switch SVIs

For each peer switch, these commands create an SVI and associate it to the trunk group assigned to the peer link port channel. STP is disabled on the VLAN.

These commands configure the SVI on Switch1

```
switch1(config)#vlan 4094
switch1(config-vlan-4094)#trunk group peertrunk
switch1(config-vlan-4094)#interface vlan 4094
switch1(config-if-vl4094)#ip address 172.17.0.1/30
switch1(config-if-vl4094)#no autostate
switch1(config-if-vl4094)#exit
switch1(config)#no spanning-tree vlan 4094
switch1(config)#
```

These commands configure the SVI on Switch2

```
switch2(config)#vlan 4094
switch2(config-vlan-4094)#trunk group trunkpeer
switch2(config-vlan-4094)#interface vlan 4094
switch2(config-if-vl4094)#ip address 172.17.0.2/30
switch2(config-if-vl4094)#no autostate
switch2(config-if-vl4094)#exit
switch2(config)#no spanning-tree vlan 4094
switch2(config)#
```

12.5.2.3 Configuring the Peer Links

These commands create the peer links on each MLAG switch.

These commands create peer links on Switch1

```
switch1(config)#mlag configuration
switch1(config-mlag)#local-interface vlan 4094
switch1(config-mlag)#peer-address 172.17.0.2
switch1(config-mlag)#peer-link port-channel 101
switch1(config-mlag)#domain-id mlag_01
switch1(config-mlag)#heartbeat-interval 2500
switch1(config-mlag)#reload-delay 150
switch1(config-mlag)#exit
switch2(config)#
```

These commands create peer links on Switch2

```
switch2(config)#mlag configuration
switch2(config-mlag)#local-interface vlan 4094
switch2(config-mlag)#peer-address 172.17.0.1
switch2(config-mlag)#peer-link port-channel 201
switch2(config-mlag)#domain-id mlag_01
switch2(config-mlag)#heartbeat-interval 2500
switch2(config-mlag)#reload-delay 150
switch2(config-mlag)#exit
switch2(config)#
```

12.5.3 Configuring Peer Switch MLAGs

These commands create the MLAGs that connect the MLAG domain to the network attached devices.

These commands configure MLAG 1 on Switch1

```
switch1(config)#interface ethernet 17-18
switch1(config-if-et17-18)#channel-group 1 mode active
switch1(config-if-et17-18)#interface port-channel 1
switch1(config-if-po1)#mlag 1
switch1(config-if-po1)#exit
switch1(config)#
```

These commands configure MLAG 1 on Switch2

```
switch2(config)#interface ethernet 1-2
switch2(config-if-et1-2)#channel-group 1 mode active
switch2(config-if-et1-2)#interface port-channel 1
switch2(config-if-po1)#mlag 1
switch2(config-if-po1)#exit
switch2(config)#
```

These commands configure MLAG 2 on Switch1

```
switch1(config)#interface ethernet 19-20
switch1(config-if-et19-20)#channel-group 2 mode active
switch1(config-if-et19-20)#interface port-channel 2
switch1(config-if-po2)#mlag 2
switch1(config-if-po2)#exit
switch1(config)#
```

These commands configure MLAG 2 on Switch2

```
switch2(config)#interface ethernet 3-4
switch2(config-if-et3-4)#channel-group 2 mode active
switch2(config-if-et3-4)#interface port-channel 2
switch2(config-if-po2)#mlag 2
switch2(config-if-po2)#exit
switch2(config)#
```

These commands configure MLAG 3 on Switch1

```
switch1(config)#interface ethernet 23
switch1(config-if-et23)#channel-group 3 mode active
switch1(config-if-et23)#interface port-channel 3
switch1(config-if-po3)#mlag 3
switch1(config-if-po3)#exit
switch1(config)#
```

These commands configure MLAG 3 on Switch2

```
switch2(config)#interface ethernet 7
switch2(config-if-et7)#channel-group 3 mode active
switch2(config-if-et7)#interface port-channel 3
switch2(config-if-po3)#mlag 3
switch2(config-if-po3)#exit
switch2(config)#
```

These commands configure MLAG 4 on Switch1

```
switch1(config)#interface ethernet 25
switch1(config-if-et25)#channel-group 4 mode active
switch1(config-if-et25)#interface port-channel 4
switch1(config-if-po4)#mlag 4
switch1(config-if-po4)#exit
switch1(config)#
```

These commands configure MLAG 4 on Switch2

```
switch2(config)#interface ethernet 9
switch2(config-if-et9)#channel-group 4 mode active
switch2(config-if-et9)#interface port-channel 4
switch2(config-if-po4)#mlag 4
switch2(config-if-po4)#exit
switch2(config)#
```

12.5.4 Configuring the Network Attached Devices

These commands create the LAGs on the Network Attached Devices that connect to the MLAG domain.

These commands configure the port channels on NAD-1

```
NAD-1(config)#interface ethernet 7-10
NAD-1(config-if-Et7-10)#channel-group 1 mode active
NAD-1(config-if-Et7-10)#exit
NAD-1(config)#
```

These commands configure the port channels on NAD-2

```
NAD-2(config)#interface ethernet 25-28
NAD-2(config-if-Et25-28)#channel-group 7 mode active
NAD-2(config-if-Et25-28)#exit
NAD-2(config)#
```

These commands configure the port channels on NAD-3

```
NAD-3(config)#interface ethernet 3-4
NAD-3(config-if-Et3-4)#channel-group 5 mode active
NAD-3(config-if-Et3-4)#exit
NAD-3(config)#
```

These commands configure the port channels on NAD-4

```
NAD-4(config)#interface ethernet 1-2
NAD-4(config-if-Et1-2)#channel-group 2 mode active
NAD-4(config-if-Et1-2)#exit
NAD-4(config)#
```

12.5.5 Verification

The following tasks verify the MLAG peer and connection configuration:

- Section 12.5.5.1: Verify the Peer Switch Connection
- Section 12.5.5.2: Verify the MLAGs
- Section 12.5.5.3: Verify Spanning Tree Protocol (STP)
- Section 12.5.5.4: Verify the MLAG Port Channel
- Section 12.5.5.5: Verify the VLAN Membership

12.5.5.1 Verify the Peer Switch Connection

To display the MLAG configuration and the MLAG status on Switch 1, use the **show mlag** command:

```
Switch1#show mlag
MLAG Configuration:
domain-id          :            mlag_01
local-interface    :          Vlan4094
peer-address       :       172.17.0.2
peer-link          :    Port-Channel101

MLAG Status:
state              :             Active
peer-link status   :                 Up
local-int status   :                 Up
system-id          :  02:1c:FF:00:15:38

MLAG Ports:
Disabled           :   0
Configured         :   0
Inactive           :   0
Active-partial     :   0
Active-full        :   4
```

To display the MLAG configuration and the MLAG status on Switch 2, use the **show mlag** command:

```
Switch2#show mlag
MLAG Configuration:
domain-id          :            mlag_01
local-interface    :          Vlan4094
peer-address       :       172.17.0.1
peer-link          :    Port-Channel102

MLAG Status:
state              :             Active
peer-link status   :                 Up
local-int status   :                 Up
system-id          :  02:1c:FF:00:15:41

MLAG Ports:
Disabled           :   0
Configured         :   0
Inactive           :   0
Active-partial     :   0
Active-full        :   4
```

12.5.5.2 Verify the MLAGs

The **show mlag interfaces** command displays MLAG connections between the MLAG switches and the Network Attached Devices.

- This **show mlag interfaces** command displays MLAG connections between the MLAG peer Switch 1 and the network attached devices:

```
Switch1#show mlag interfaces
                                                                  local/remote
  mlag     desc                   state     local     remote         status
-----------------------------------------------------------------------------
     1     sw1.po1          active-full       Po1        Po1          up/up
     2     sw1.po2          active-full       Po2        Po2          up/up
     3     sw1.po3          active-full       Po3        Po3          up/up
     4     sw1.po4          active-full       Po4        Po4          up/up
```

- The following **show mlag interfaces** command, with the ***detail*** option, displays MLAG connections between the MLAG peer Switch 1 and the network attached devices

```
Switch2#show mlag interfaces detail
                                local/remote
mlag       state local remote   oper   config          last change  changes
-----------------------------------------------------------------------------
   1 active-full   Po1    Po1  up/up  ena/ena  6 days, 2:08:28 ago        5
   2 active-full   Po2    Po2  up/up  ena/ena  6 days, 2:08:30 ago        5
   3 active-full   Po3    Po3  up/up  ena/ena  6 days, 2:08:33 ago        5
   4 active-full   Po4    Po4  up/up  ena/ena  6 days, 2:08:41 ago        5
Switch2#
```

12.5.5.3 Verify Spanning Tree Protocol (STP)

STP functions can be displayed from each peer switch. MLAG interfaces are displayed as a single entry. Configured interfaces on each switch that are not included in an MLAG are displayed. Local interfaces have the normal notation; remote interfaces are preceded by ***P*** or ***Peer***.

VLAN Output 1: Assume VLAN 3903 includes MLAG 1

```
Switch1#show spanning-tree vlan 3903
Spanning tree instance for vlan 3903
VL3903
  Spanning tree enabled protocol rapid-pvst
  Root ID    Priority    36671
             Address     001c.730c.3009
             Cost        1999 (Ext) 0 (Int)
             Port        105 (Port-Channel5)
             Hello Time  2.000 sec  Max Age 20 sec  Forward Delay 15 sec

  Bridge ID  Priority    36671  (priority 32768 sys-id-ext 3903)
             Address     021c.7300.1319
             Hello Time  2.000 sec  Max Age 20 sec  Forward Delay 15 sec

Interface        Role       State      Cost      Prio.Nbr Type
---------------- ---------- ---------- --------- -------- --------------------
Po1              root       forwarding 1999      128.105  P2p
Switch1#
```

The output displays MLAG 1 under its local interface name (Po1). A peer interface is not displayed because spanning tree considers the local and remote Port Channels as a single MLAG interface.

VLAN Output 2: Assume VLAN 3908 does not include any MLAGs

```
Switch1#show spanning-tree vlan 3908
Spanning tree instance for vlan 3908
VL3908
  Spanning tree enabled protocol rapid-pvst
  Root ID    Priority    36676
             Address     021c.7300.1319
             This bridge is the root

  Bridge ID  Priority    36676  (priority 32768 sys-id-ext 3908)
             Address     021c.7300.1319
             Hello Time  2.000 sec  Max Age 20 sec  Forward Delay 15 sec

Interface        Role       State      Cost      Prio.Nbr Type
---------------- ---------- ---------- --------- -------- --------------------
Et17             designated forwarding 2000      128.217  P2p
Et18             designated forwarding 2000      128.218  P2p
PEt17            designated forwarding 2000      128.17   P2p
PEt18            designated forwarding 2000      128.18   P2p
```

The output displays all interfaces from both switches. Each interface is explicitly displayed because they are individual units that STP must consider when selecting ports to block.

- Et17 and Et18 are located on the switch where the **show spanning-tree** command is issued.
- PEt17 and PEt18 are located on the remote switch from where the command was issued

An identical command issued on the peer switch displays similar information.

Verify the MLAG does not create topology loops (show spanning-tree blocked)

```
Switch1#show spanning-tree blocked
Name       Blocked Interfaces List
----------
--------------------------------------------------------------------------------

Number of blocked ports (segments) in the system : 0
Switch1#
```

12.5.5.4 Verify the MLAG Port Channel

Issue the command **show port-channel** for channels 1-4 from Switch 1:

```
Switch#show port-channel 1-4
Port Channel Port-Channel1:
  Active Ports: Ethernet17     Ethernet18     PeerEthernet1 PeerEthernet2
Port Channel Port-Channel2:
  Active Ports: Ethernet19     Ethernet20     Ethernet21     Ethernet22
                PeerEthernet3 PeerEthernet4 PeerEthernet5 PeerEthernet6
Port Channel Port-Channel3:
  Active Ports: Ethernet23     Ethernet24     PeerEthernet7 PeerEthernet8
Port Channel Port-Channel4:
  Active Ports: Ethernet25     Ethernet26     PeerEthernet9 PeerEthernet10
```

Issue the command **show port-channel load-balance fields detailed** command for channel 1 from Switch 2:

```
Switch#show port-channel 1 detailed
Port Channel Port-Channel1:
  Active Ports:
       Port                Time became active       Protocol    Mode
     ------------------------------------------------------------------------
       Ethernet17          7/7/11 15:27:36          LACP        Active
       Ethernet18          7/7/11 15:27:36          LACP        Active
       PeerEthernet1       7/7/11 15:27:36          LACP        Active
       PeerEthernet2       7/7/11 15:27:36          LACP        Active
```

12.5.5.5 Verify the VLAN Membership

The **show vlan** command displays VLAN member ports, including MLAG ports and ports on each peer not bundled in an MLAG.

```
Switch1#show vlan 3903, 3908
VLAN  Name                             Status    Ports
----- -------------------------------- --------- -------------------------------
3903  ar.mg.rn.172.17.254.16/29        active    Cpu, Po1
3908  po.ra.ar.mg.172.17.254.64/29     active    Cpu, Et17, Et18, PEt17, PEt18
```

12.6 MLAG Commands

MLAG and Port Channel Commands – Global Configuration Mode

- mlag configuration (global configuration)

Interface Configuration Commands – Interface Configuration Mode

- mlag (port-channel interface configuration)

MLAG Configuration Commands

- domain-id
- heartbeat-interval (MLAG)
- local-interface
- peer-address
- peer-link
- reload-delay mlag
- reload-delay mode
- reload-delay non-mlag
- shutdown (MLAG)

Display Commands

- show mlag
- show mlag interfaces
- show mlag interfaces members
- show mlag interfaces states
- show mlag issu warnings

domain-id

The **domain-id** command specifies a name for the multi-chassis link aggregation (MLAG) domain.

The **no domain-id** and **default domain-id** commands remove the MLAG domain name by deleting the **domain-id** statement from ***running-config***.

Command Mode

MLAG Configuration

Command Syntax

```
domain-id identifier
no domain-id
default domain-id
```

Parameters

- *identifier* alphanumeric string that names the MLAG domain.

Examples

- This command names the MLAG domain ***mlag1***.

```
switch(config)#mlag
switch(config-mlag)#domain-id mlag1
switch(config-mlag)#
```

heartbeat-interval (MLAG)

The **heartbeat-interva**l command configures the interval at which heartbeat messages are issued in a multi-chassis link aggregation (MLAG) configuration.

The **no heartbeat-interval** and **default heartbeat-interval** commands revert the heartbeat interval to the default setting (2 seconds) by removing the **heartbeat-interval** command from ***running-config***.

Command Mode

MLAG Configuration

Command Syntax

```
heartbeat-interval period
no heartbeat-interval
default heartbeat-interval
```

Parameters

- *period* Interval duration (ms). Value ranges from 1000 through 30000. Default interval is 2000 ms.

Guidelines

Heartbeat messages flow independently in both directions between the MLAG peers. If a peer stops receiving heartbeat messages within the expected time frame (30 seconds), the other peer can assume it no longer functions and without intervention or repair, the MLAG becomes disabled. Both switches revert to their independent state.

Important! On 7500 and 7500E Series Switches, Arista recommends setting the heartbeat interval to 10 seconds.

Examples

- This command configures the heartbeat interval to 15000 milliseconds:

```
switch(config)#mlag
switch(config-mlag)#heartbeat-interval 15000
switch(config-mlag)#
```

local-interface

The **local-interface** command assigns a VLAN interface for use in multi-chassis link aggregation (MLAG) configurations. The VLAN interface is used for both directions of communication between the MLAG peers.

The **no local-interface** and **default local-interface** commands delete the VLAN interface assignment by removing the **local-interface** command from ***running-config***.

Command Mode

MLAG Configuration

Command Syntax

```
local-interface vlan vlan_number
no local-interface
default local-interface
```

Parameters

- *vlan_number* VLAN number, in the range from 1 through 4094.

Guidelines

When configuring the local interface, the VLAN interface must exist already. To configure a VLAN interface, issue the command **interface vlan**.

Example

- This command assigns VLAN 4094 as the local interface.

```
switch(config)#mlag
switch(config-mlag)#local-interface vlan 4094
switch(config-mlag)#
```

mlag (port-channel interface configuration)

The **mlag** command assigns an MLAG ID to a port-channel. MLAG peer switches form an MLAG when each switch configures the same MLAG ID to a port-channel interface. Only one MLAG ID can be assigned to an interface. An individual MLAG number cannot be assigned to more than one interface.

The **no mlag** and **default mlag** commands remove the MLAG ID assignment from the configuration mode interface by deleting the corresponding **mlag** command from ***running-config***.

Command Mode

Interface-Port Channel Configuration

Command Syntax

```
mlag number
no mlag
default mlag
```

Parameters

- *number* Number used as MLAG ID. Value ranges from 1 to 2000.

Example

- These commands configures a port channel and assigns it MLAG 4.

```
switch(config)#interface ethernet 5-10
switch(config-if-Et5-10)#channel-group 1 mode active
switch(config-if-Et5-10)#interface port-channel 4
switch(config-if-Po4)#switchport trunk group group4
switch(config-if-Po4)#mlag 4
switch(config-if-Po4)#exit
switch(config)#
```

mlag configuration (global configuration)

The **mlag configuration** command enters MLAG configuration mode to configure multi-chassis link aggregation (MLAG) features. MLAG configuration mode is not a group change mode; ***running-config*** is changed immediately after commands are executed. The **exit** command does not affect the configuration.

The **no mlag configuration** and **default mlag configuration** commands remove all MLAG configuration commands from ***running-config***.

The **exit** command returns the switch to global configuration mode.

Command Mode

Global Configuration

Command Syntax

```
mlag [configuration]
no mlag configuration
default mlag configuration
```

mlag and **mlag configuration** are identical commands.

Guidelines

An MLAG is formed by connecting two switches through an interface called a peer link. The peer link carries control and data traffic between the switches, including advertisements and keepalive messages. This information coordinates the switches. Functioning peers are in the **active** state.

Each peer switch uses IP-level connectivity between their local addresses and the MLAG peer IP address to form and maintain the peer link.

Commands Available in MLAG Configuration Mode

- **domain-id**
- **heartbeat-interval (MLAG)**
- **local-interface**
- **peer-address**
- **peer-link**
- **reload-delay mlag**
- **shutdown (MLAG)**

Example

- These commands enter MLAG configuration mode and configure MLAG parameters:

```
switch(config)#mlag
switch(config-mlag)#local-interface vlan 4094
switch(config-mlag)#peer-address 10.0.0.2
switch(config-mlag)#peer-link port-channel 10
switch(config-mlag)#domain-id mlagDomain
switch(config-mlag)#heartbeat-interval 2500
switch(config-mlag)#reload-delay 2000
switch(config-mlag)#exit
switch(config)#
```

peer-address

The **peer-address** command specifies the peer IPv4 address for a multi-chassis link aggregation (MLAG) domain. MLAG control traffic, including keepalive messages, is sent to the peer IPv4 address. If the peer IPv4 address is unreachable, then MLAG peering fails and both peer switches revert to their independent state.

The **no peer-address** and **default peer-address** commands remove the MLAG peer's IPv4 address assignment by deleting the peer-address command from ***running-config***.

Command Mode

MLAG Configuration

Command Syntax

```
peer-address ipv4_addr
no peer-address
default peer-address
```

Parameters

- *ipv4_addr* MLAG peer IPv4 address.

Example

- These commands configure the MLAG peer address.

```
switch(config)#mlag
switch(config-mlag)#peer-address 10.0.0.2
switch(config-mlag)#
```

peer-link

The **peer-link** command specifies the interface that connects multi-chassis link aggregation (MLAG) peers. To form an MLAG, two switches are connected through an interface called a peer link. The peer link carries control and data traffic between the two switches. Control traffic includes MLAG-related advertisements and keepalive messages. This information keeps the two switches working as one.

The **no peer-link** and **default peer-link** command remove the peer link by deleting the **peer-link** command from ***running-config***.

Command Mode

MLAG Configuration

Command Syntax

```
peer-link INT_NAME
no peer-link
default peer-link
```

Parameters

- ***INT_NAME*** denotes the interface type and number of the interface. Values include:
  - **ethernet** *e_num* Ethernet interface range specified by *e_num*.
  - **port-channel** *p_num* Channel group interface range specified by *p_num*.

Example

- These commands creates a peer link.

```
switch(config)#mlag configuration
switch(config-mlag)#peer-link port-channel 10
switch(config-mlag)
```

reload-delay mlag

The **reload-delay mlag** command configures the reload delay period for MLAG links. The command also specifies the reload delay period for non-MLAG links when the **reload-delay non-mlag** command is not configured.

Each Arista switch defaults to the recommended reload-delay value, which varies by switch platform:

- **fixed configuration switches**: 300 seconds
- **Trident-2 modular switches**: 900 seconds

  7304
  7308
  7316
  7300X series
- **Sand platform modular switches**: 1800 seconds

  7504
  7508
  7500E series
  7548S

The **no reload-delay mlag** and **default reload-delay mlag** commands restore the default value by deleting the **reload-delay mlag** statement from ***running-config***.

Command Mode

MLAG Configuration

Command Syntax

```
reload-delay [mlag] PERIOD
no reload-delay [mlag]
default reload-delay [mlag]
```

Parameters

- ***PERIOD*** Period that non-peer links are disabled after an MLAG peer reboots. Options include:
  - **infinity** link is not enabled after reboot.
  - <***0*** to ***86400***> disabled link interval (seconds). Default varies by switch platform as described above.

Guidelines

The **reload-delay** and **reload-delay mlag** commands are equivalent.

Example

- These commands configure the reload-delay interval to 15 minutes.

```
switch(config)#mlag configuration
switch(config-mlag)#reload-delay mlag 900
switch(config-mlag)#
```

reload-delay mode

The **reload-delay mode** command specifies the state of LACP LAG ports during the MLAG reload delay period. By default, MLAG ports remain in the errdisabled state during reload delay. This command configures MLAG ports to come up to standby mode before the expiration of the reload delay period.

The **no reload-delay mode** and **default reload-delay mode** commands restore the default behavior of MLAG ports by deleting the **reload-delay mode** statement from ***running-config***. The default behavior is for the MLAG ports to remain in the errdisabled state until the expiration of the reload delay period

Command Mode

MLAG Configuration

Command Syntax

```
reload-delay mode lacp standby
no reload-delay mode
default reload-delay mode
```

Related Commands

- **reload-delay mlag** configures the MLAG reload delay period.

Example

- These commands configure the MLAG port to come up to standby state before the end of the reload delay period.

```
switch(config)#mlag configuration
switch(config-mlag)#reload-delay mode lacp standby
switch(config-mlag)#
```

reload-delay non-mlag

The **reload-delay non-mlag** command specifies the period that non-MLAG links are disabled after an MLAG peer reboots. This interval allows non peer links to learn multicast and OSPF states before the ports start handling traffic. The recommended minimum value required to ensure the forwarding hardware is initialized with the topology state depends on the switch platform:

- fixed configuration switches: 300 seconds (five minutes)
- modular switches: 1200 seconds (20 minutes)

When the **reload-delay non-mlag** command is not configured, the **reload-delay mlag** command specifies the reload delay time for non-MLAG and MLAG links.

The **no reload-delay non-mlag** and **default reload-delay non-mlag** command restores the default behavior by deleting the **reload-delay non-mlag** statement from ***running-config***.

Command Mode

MLAG Configuration

Command Syntax

```
reload-delay non-mlag PERIOD
no reload-delay non-mlag
default reload-delay non-mlag
```

Parameters

- ***PERIOD*** Period that non-MLAG links are disabled after an MLAG peer reboots. Options include:
  - **infinity** links are not enabled after reboot.
  - <***0*** to ***86400***> disabled link interval (seconds). Values range from 0 to 86400 (24 hours).

Example

- These commands configure the reload-delay interval of non-MLAG links to 20 minutes.

```
switch(config)#mlag configuration
switch(config-mlag)#reload-delay non-mlag 1200
switch(config-mlag)#
```

show mlag

The **show mlag** command displays information about the multi-chassis link aggregation (MLAG) configuration on bridged Ethernet interfaces.

Command Mode

EXEC

Command Syntax

show mlag [*INFO_LEVEL*]

Parameters

- ***INFO_LEVEL*** specifies information displayed by command. Options include:
  - <no parameter> command displays MLAG configuration, status, and ports.
  - **detail** command displays MLAG configuration, status, ports, and detailed status.

Example

- This command displays output from the **show mlag** command:

```
switch>show mlag
MLAG Configuration:
domain-id          :         ar.mg.mlag
local-interface    :           Vlan3901
peer-address       :       172.17.254.2
peer-link          :      Port-Channel1

MLAG Status:
state              :             Active
peer-link status   :                 Up
local-int status   :                 Up
system-id          :  02:1c:73:00:13:19

MLAG Ports:
Disabled           :   0
Configured         :   0
Inactive           :   0
Active-partial     :   0
Active-full        :   5
switch>
```

show mlag interfaces

The **show mlag interfaces** command displays information about the multi-chassis link aggregation (MLAG) configuration on bridged Ethernet interfaces.

Command Mode

EXEC

Command Syntax

show mlag interfaces [*MLAGS*] [*INFO_LEVEL*]

Parameters

- ***MLAGS*** MLAG channels for which command displays data. Options include:
  - <no parameter> command displays data for all MLAGs.
  - *mlag_id* specifies MLAG for which command displays data. Value ranges from 1 to 2000.
- ***INFO_LEVEL*** specifies information displayed by command. Options include:
  - <no parameter> command displays basic MLAG interface parameters
  - **detail** command displays detailed MLAG interface parameters.

Example

- This command displays output from the **show mlag interfaces detail** command:

```
switch>show mlag interfaces detail
                                 local/remote
mlag       state local remote   oper   config          last change  changes
--------------------------------------------------------------------------------
   4 active-full   Po4    Po4  up/up  ena/ena  6 days, 1:19:26 ago        5
   5 active-full   Po5    Po5  up/up  ena/ena  6 days, 1:19:24 ago        5
   6 active-full   Po6    Po6  up/up  ena/ena  6 days, 1:19:23 ago        5
   7 active-full   Po7    Po7  up/up  ena/ena  6 days, 1:19:23 ago        5
```

show mlag interfaces members

The **show mlag interfaces members command displays information about the** multi-chassis link aggregation **(MLAG) members on bridged Ethernet interfaces.**

Command Mode

EXEC

Command Syntax

```
show mlag interfaces members
```

Example

- This command displays the MLAG interface members.

```
switch#show mlag interface members
Mlag4 is Port-Channel4
  Active Ports: Ethernet3 PeerEthernet3
Mlag5 is Port-Channel5
  Active Ports: Ethernet14
Mlag7 is Port-Channel7
  Active Ports: Ethernet5 PeerEthernet5
Mlag8 is Port-Channel8
  Active Ports: Ethernet10 PeerEthernet10
Mlag9 is Port-Channel9
  Active Ports: Ethernet15 Ethernet21 PeerEthernet19 PeerEthernet20
Mlag10 is Port-Channel10
  Active Ports: Ethernet19 Ethernet20 PeerEthernet21 PeerEthernet22
switch#
```

show mlag interfaces states

The **show mlag interfaces states** command **displays information about the** multi-chassis link aggregation **(MLAG) states on bridged Ethernet interfaces.**

Command Mode

EXEC

Command Syntax

show mlag interfaces [*MLAGS*] states [*STATE_NAMES*] [*INFO_LEVEL*]

Parameters

- ***MLAGS*** MLAG channels for which command displays data. Options include:
  - <no parameter> command displays data for all MLAGs.
  - *mlag_id* specifies MLAG for which command displays data. Value ranges from 1 to 2000.
- ***STATE_NAMES*** MLAG channels for which command displays data. Parameter may specify more than one name, which can be listed in any order. Valid state names include:
  - **active-full** includes active-full interfaces.
  - **active-partial** includes active-partial interfaces.
  - **configured** includes configured interfaces.
  - **disabled** includes disabled interfaces.
  - **inactive** includes inactive interfaces.
- ***INFO_LEVEL*** specifies information displayed by command. Options include:
  - <no parameter> command displays basic MLAG interface parameters
  - **detail** command displays detailed MLAG interface state parameters.

Example

- This command displays the MLAG interface states that are active-full.

```
switch#show mlag interfaces states active-full
                                                                  local/remote
  mlag     desc                       state     local     remote         status
-------- -------------------- --------------- --------- ---------- ------------
     4     b.po1                active-full       Po4        Po4          up/up
     7     ar.mg.au.po1         active-full       Po7        Po7          up/up
     8     co.po1               active-full       Po8        Po8          up/up
     9     k.po5                active-full       Po9        Po9          up/up
    10     ar.mg.pt.ir.po10     active-full      Po10       Po10          up/up
switch#
```

show mlag issu warnings

The **show mlag issu warnings** command displays a warning message regarding the backward-compatibility of this feature before you upgrade.

Command Mode

EXEC

Command Syntax

```
show mlag issu warnings
```

Example

- This command displays the MLAG backward-compatibility warning message. Refer to the latest version of the release notes for additional information before you upgrade.

```
switch##show mlag issu warnings
If you are performing an upgrade, and the Release Notes for the new
version of EOS indicate that MLAG is not backwards-compatible with the
currently installed version, the upgrade will result in packet loss.

Stp is not restartable. Topology changes will occur during the upgrade
process.


switch#
```

shutdown (MLAG)

The **shutdown** command disables MLAG on the switch without modifying the MLAG configuration.

The **no shutdown** and **default shutdown** commands re-enable MLAG by removing the **shutdown** command from ***running-config***.

Command Mode

MLAG Configuration

Command Syntax

```
shutdown
no shutdown
default shutdown
```

Example

- These commands disable MLAG on the switch.

```
switch(config)#mlag configuration
switch(config-mlag)#shutdown
switch(config-mlag)#
```

Chapter 13

802.1x Port Security

This section explains the basic concepts behind 802.1x port security, including switch roles, how the switches communicate, and the procedure used for authenticating clients.

- Section 13.1: 802.1x Port Security Introduction
- Section 13.2: 802.1x Port Security Description
- Section 13.3: Configuring 802.1x Port Security
- Section 13.4: Displaying 802.1x information
- Section 13.5: IEEE 802.1x Configuration Commands

13.1 802.1x Port Security Introduction

Port security control who can send or receive traffic from an individual switch port. An end node is not allowed to send or receive traffic through a port until the node is authenticated by a RADIUS server.

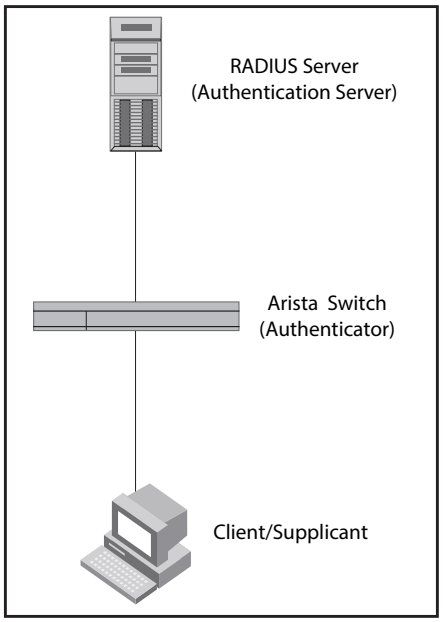
This prevents unauthorized individuals from connecting to a switch port to access your network. Only designated valid users on a RADIUS server will be allowed to use the switch to access the network.

13.2 802.1x Port Security Description

13.2.1 Switch Roles for 802.1x Configurations

The 802.1x standard specifies the roles of **Supplicant (client)**, **Authenticator**, and **Authentication Server** in a network. Figure 11-1 illustrates these roles.

Figure 13-1: Authenticator, Supplicant, and Authentication Server in an 802.1x configuration



Authentication server – The switch that validates the client and specifies whether or not the client may access services on the switch. The switch supports Authentication Servers running RADIUS.

Authenticator – The switch that controls access to the network. In an 802.1x configuration, the switch serves as the Authenticator. As the Authenticator, it moves messages between the client and the Authentication Server. The Authenticator either grants or does not grant network access to the client based on the identity data provided by the client, and the authentication data provided by the Authentication Server.

Supplicant/Client – The client provides a username or password data to the Authenticator. The Authenticator sends this data to the Authentication Server. Based on the supplicant's information, the Authentication Server determines whether the supplicant can use services given by the Authenticator. The Authentication Server gives this data to the Authenticator, which then provides services to the client, based on the authentication result.

13.2.2 Authentication Process

The authentication that occurs between a supplicant, authenticator, and authentication server include the following processes.

- Either the authenticator (a switch port) or the supplicant starts an authentication message exchange. The switch starts an exchange when it detects a change in the status of a port, or if it gets a packet on the port with a source MAC address that is not included in the MAC address table.

- An authenticator starts the negotiation by sending an EAP-Request/Identity packet. A supplicant starts the negotiation with an EAPOL-Start packet, to which the authenticator answers with a EAP-Request/Identity packet.
- The supplicant answers with an EAP-Response/Identity packet to the authentication server via the authenticator.
- The authentication server responds with an EAP-Request packet to the supplicant via the authenticator.
- The supplicant responds with an EAP-Response.
- The authentication server transmits either an EAP-Success packet or EAP-Reject packet to the supplicant.
- If an EAP-Reject is received, the supplicant will receive an EAP-Reject message and their traffic will not be forwarded.

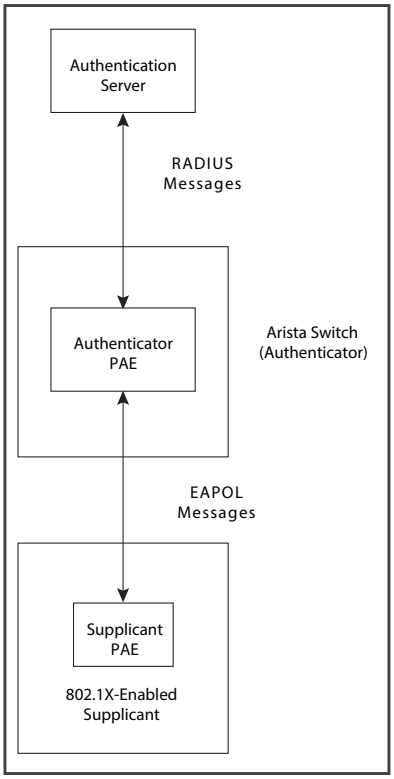
13.2.3 Communication Between the Switches

For communication between the switches, 802.1x port security uses the Extensible Authentication Protocol (EAP), defined in RFC 2284 and the RADIUS authentication protocol.

The 802.1x standard defines a method for encapsulating EAP messages so they can be sent over a LAN. This encapsulated kind of EAP is known as EAP over LAN (EAPOL). The standard also specifies a means of transferring the EAPOL information between the client or Supplicant, Authenticator, and Authentication Server.

EAPOL messages are passed between the Supplicant's and Authenticator's Port Access Entity (PAE). Figure 13-2 shows the relationship between the Authenticator PAE and the Supplicant PAE.

Figure 13-2: Authenticator PAE and Supplicant PAE



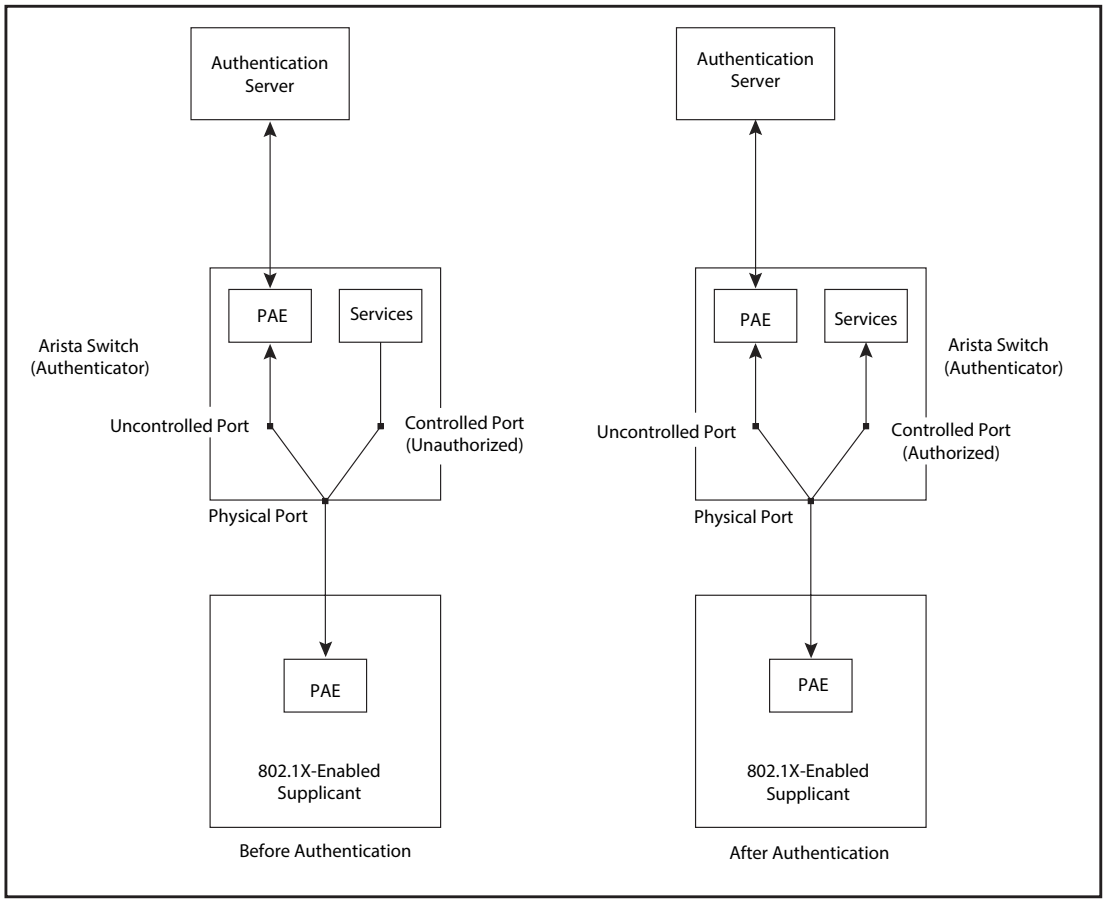
Authenticator PAE: The Authenticator PAE communicates with the Supplicant PAE to receive the Supplicant's identifying information. Behaving as a RADIUS client, the Authenticator PAE passes the Supplicant's information to the Authentication Server, which decides whether to grant the Supplicant access. If the Supplicant passes authentication, the Authenticator PAE allows it access to the port.

Supplicant PAE – The Supplicant PAE provides information about the client to the Authenticator PAE and replies to requests from the Authenticator PAE. The Supplicant PAE may initiate the authentication procedure with the Authenticator PAE, as well as send logoff messages.

13.2.4 Controlled and Uncontrolled Ports

A physical port on the switch used with 802.1x has two virtual access points that include a controlled port and an uncontrolled port. The controlled port grants full access to the network. The uncontrolled port only gives access for EAPOL traffic between the client and the Authentication Server. When a client is authenticated successfully, the controlled port is opened to the client.

Figure 13-3: Ports before and after client authentication



The uncontrolled port on the Authenticator is the only one open before a client is authenticated. The uncontrolled port permits only EAPOL frames to be swapped between the client and the Authentication Server. No traffic is allowed to pass through the controlled port in the unauthorized state.

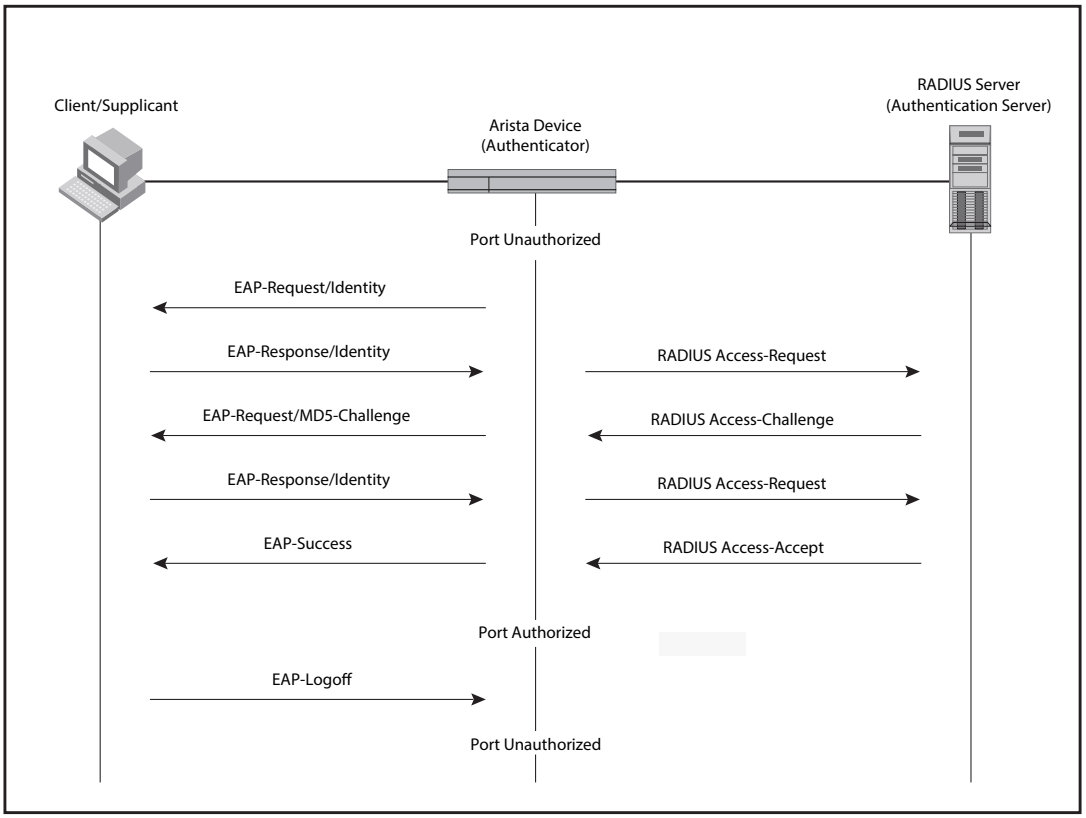
During authentication, EAPOL messages are swapped between the Supplicant PAE and the Authenticator PAE, and RADIUS messages are swapped between the Authenticator PAE and the Authentication Server. If the client is successfully authenticated, the controlled port becomes authorized, and traffic from the client can flow through the port normally.

All controlled ports on the switch are placed in the authorized state, allowing all traffic, by default. When authentication is initiated, the controlled port on the interface is initially set in the unauthorized state. If a client connected to the port is authenticated successfully, the controlled port is set in the authorized state.

13.2.5 Message Exchange During Authentication

Figure 13-4 illustrates an exchange of messages between an 802.1x-enabled client, a switch operating as Authenticator, and a RADIUS server operating as an Authentication Server.

Figure 13-4: Message exchange during authentication



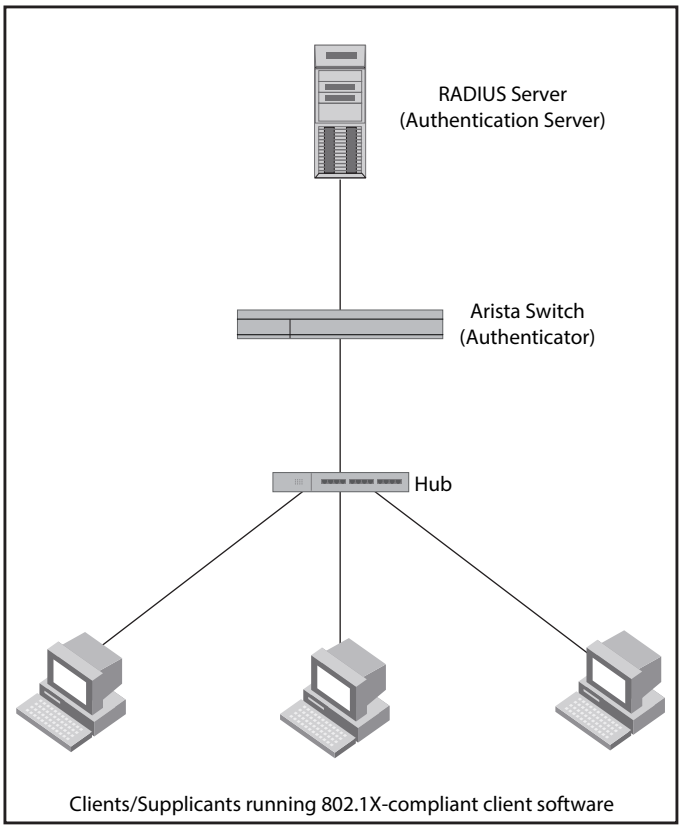
Arista switches support MD5-challenge TLS and any other EAP-encapsulated authentication types in EAP Request or Response messages. In other words, the switches are transparent to the authentication scheme used.

13.2.6 Authenticating Multiple Clients Connected to the Same Port

Arista switches support 802.1x authentication for ports with more than one client connected to them (multi-host mode). Figure 13-5 illustrates a sample configuration where multiple clients are connected to a single 802.1x port.

If there are multiple clients connected to a single 802.1x-enabled port, the switch authenticates each individually. Each client's authentication state is independent of the others, so that if one authenticated client disconnects from the network, it won't impact the authentication status of any of the other authenticated clients.

Figure 13-5: Multiple clients connected to a 802.1x-enabled port



13.3 Configuring 802.1x Port Security

Basic steps to implementing 802.1x Port-based Network Access Control and RADIUS accounting on the switch:

Step 1 A RADIUS server is required on one or more of your network servers or management stations.

802.1x is not supported with the TACACS+ authentication protocol.

Step 2 You must create supplicant accounts on the RADIUS server:

- The account for a supplicant connected to an authenticator port must have a username and password combination when set to the 802.1x authentication mode. The maximum username length is 38 alphanumeric characters and spaces, and the maximum length for a password is 16 alphanumeric characters and spaces.
- An account for the supplicant connected to an authenticator port and placed in the MAC address-based authentication mode needs use the MAC address of the node as both the username and password.
- Connected clients to an 802.1x authenticator port will require 802.1x client software.

Step 3 The RADIUS client must be configured by entering the IP addresses and encryption keys of the authentication servers on your network.

Step 4 The port access control settings must be configured on the switch. This includes the following:

- Specifying the port roles.
- Configuring 802.1x port parameters.
- Enabling 802.1x Port-based Network Access Control.

Guidelines

- Do not set a port that is connected to a RADIUS authentication server to the authenticator role as an authentication server cannot authenticate itself.
- A supplicant connected to an authenticator port set to the 802.1x username and password authentication method must have 802.1x client software.
- To prevent unauthorized individuals from accessing the network through unattended network workstations, end users of 802.1x port-based network access control should always log off when they are finished with a work session.
- The RADIUS client should be configured on the switch before activating port-based access control.

13.3.1 Configuring 802.1x Authentication Methods

IEEE 802.1x port security relies on external client-authentication methods, which must be configured for use. The method currently supported on Arista switches is RADIUS authentication. To configure the switch to use a RADIUS server for client authentication, use the **aaa authentication dot1x** command.

Example

- The **aaa authentication dot1x** command configures the authentication, authorization, and accounting (AAA) methods to be used on interfaces running IEEE 802.1X. The following configures the switch to use RADIUS authentication.

```
switch(config)# aaa authentication dot1x default group radius
switch(config)#
```

13.3.2 Globally Enable IEEE 802.1x

To enable IEEE 802.1X port authentication globally on the switch, use the **dot1x system-auth-control** command.

- This command enables IEEE 802.1X globally on the switch.

```
switch(config)#dot1x system-auth-control
switch(config)
```

13.3.3 Designating Authenticator Ports

For ports to act as authenticator ports to connected supplicants, those ports must be designated using the **dot1x port-control** command.

The **auto** option of the **dot1x port-control** command designates an authenticator port for immediate use, blocking all traffic that is not authenticated by the RADIUS server.

Example

- This command configures Ethernet 1 to immediately begin functioning as an authenticator port.

```
switch(config)#interface ethernet 1
switch(config-if-Et1)#dot1x port-control auto
switch(config-if-Et1)#
```

The **force-authorized** option of the **dot1x port-control** command sets the state of the port to ***authorized*** without authentication, allowing traffic to continue uninterrupted.

Example

- These commands designate Ethernet 1 as an authenticator port that will forward packets without authentication.

```
switch(config)#interface ethernet 1
switch(config-if-Et1)#dot1x port-control force-authorized
switch(config-if-Et1)#
```

To designate a port as an authenticator but prevent it from authorizing any traffic, use the **force-unauthorized** option of the **dot1x port-control** command.

Example

- The **force-unauthorized** option of the **dot1x port-control** command places the specified port in the *unauthorized* state, which will deny any access requests from users of the ports.

```
switch(config)#interface ethernet 1
switch(config-if-Et1)#dot1x port-control force-authorized
switch(config-if-Et1)#
```

13.3.4 Configuring Re-authentication

The **dot1x reauthentication** and **dot1x timeout reauth-period** commands configure authenticator ports to require re-authentication from clients at regular intervals.

Example

- These commands configure the Ethernet interface 1 authenticator to require re-authentication from clients every 6 hours (21600 seconds).

```
switch(config)#interface ethernet 1
switch(config-if-Et1)#dot1x reauthentication
switch(config-if-Et1)#dot1x timeout reauth-period 21600
switch(config-if-Et1)#
```

- These commands deactivate re-authentication on Ethernet interface 1.

```
switch(config)#interface ethernet 1
switch(config-if-Et1)#no dot1x reauthentication
switch(config-if-Et1)#
```

13.3.5 Setting the EAP Request Maximum

The **dot1x max-reauth-req** command configures the number of times the switch retransmits an 802.1x Extensible Authentication Protocol (EAP) request packet before ending the conversation and restarting authentication.

Example

- These commands set the number of times the authenticator sends an EAP request packet to the client before restarting authentication.

```
switch(config)#interface ethernet 1
switch(config-if-Et1)#dot1x max-reauth-req 4
switch(config-if-Et1)#
```

13.3.6 Disabling Authentication on a Port

To disable authentication on an authenticator port, use the **no dot1x port-control** command.

Example

- These commands disable authentication on Ethernet interface 1.

```
switch(config)#interface ethernet 1
switch(config-if-Et1)#no dot1x port-control
switch(config-if-Et1)#
```

13.3.7 Setting the Quiet Period

If the switch fails to immediately authenticate the client, the time the switch waits before trying again is specified by the **dot1x timeout quiet-period** command. This timer also indicates how long a client that failed authentication is blocked.

Example

- These commands set the 802.1x quiet period for Ethernet interface 1 to 30 seconds.

```
switch(config)#interface ethernet 1
switch(config-if-Et1)#dot1x timeout quiet-period 30
```

13.3.8 Setting the Transmission Timeout

Authentication and re-authentication are accomplished by the authenticator sending an Extensible Authentication Protocol (EAP) request to the supplicant and the supplicant sending a reply which the authenticator forwards to an authentication server. If the authenticator doesn't receive a reply to the EAP request, it waits a specified period of time before retransmitting. To configure that wait time, use the **dot1x timeout tx-period** command.

Example

- These commands configure Ethernet interface 1 to wait 30 seconds before retransmitting EAP requests to the supplicant.

```
switch(config)#interface Ethernet 1
switch(config-if-Et1)#dot1x timeout tx-period 30
switch(config-if-Et1)#
```

13.3.9 Clearing 802.1x Statistics

The **clear dot1x statistics** command resets the 802.1x counters.

Example

- This command clears the 802.1x counters on all interfaces.

```
switch#clear dot1x statistics all
switch#
```

- This command clears the 802.1x counters on Ethernet interface 1.

```
switch#clear dot1x statistics interface ethernet 1
switch#
```

13.4 Displaying 802.1x information

You can display information about 802.1x on the switch and on individual ports.

13.4.1 Displaying port security configuration information

The **show dot1x** command shows information about the 802.1x configuration on the specified port or ports.

Example

- This commands displays IEEE 802.1x configuration information for Ethernet interface 5.

```
switch#show dot1x interface ethernet 5
Dot1X Information for Ethernet5
------------------------------------------
PortControl             : auto
QuietPeriod             : 60 seconds
TxPeriod                : 5 seconds
ReauthPeriod            : 3600 seconds
MaxReauthReq            : 2
switch#
```

13.4.2 Displaying 802.1x summary information

Use the **show dot1x all summary** command to display IEEE 802.1x status for all ports.

Example

- The following commands display a summary of IEEE 802.1x status.

```
switch#show dot1x all summary
Interface              Client                 Status
-----------------------------------------------------------
Ethernet5              None                   Unauthorized
switch#
```

13.4.3 Displaying 802.1x statistics

Use the **show dot1x statistics** command to display 802.1x statistics for the specified port or ports.

Example

- This command displays IEEE 802.1x statistics for Ethernet interface 5.

```
switch#show dot1x interface ethernet 5 statistics
Dot1X Authenticator Port Statistics for Ethernet5
-------------------------------------------------
RxStart = 0      RxLogoff = 0    RxRespId = 0
RxResp = 0       RxInvalid = 0   RxTotal = 0
TxReqId = 0      TxReq = 0       TxTotal = 0
RxVersion = 0    LastRxSrcMAC = 0000.0000.0000
switch#
```

13.5 IEEE 802.1x Configuration Commands

Global Configuration Commands

- dot1x system-auth-control

Interface Configuration Commands – Ethernet Interface

- dot1x max-reauth-req
- dot1x pae authenticator
- dot1x port-control
- dot1x reauthentication
- dot1x timeout quiet-period
- dot1x timeout reauth-period
- dot1x timeout tx-period

Privileged EXEC Commands

- clear dot1x statistics
- show dot1x
- show dot1x statistics
- show dot1x all summary

clear dot1x statistics

The **clear dot1x statistics** command resets the 802.1x counters on the specified interface or all interfaces.

Command Mode

Privileged EXEC

Command Syntax

```
clear dot1x statistics INTERFACE_NAME
```

Parameters

- ***INTERFACE_NAME*** Interface type and number. Options include:
  - **all** Display information for all interfaces.
  - **interface ethernet** *e_num* Ethernet interface specified by *e_num*.
  - **interface loopback** *l_num* Loopback interface specified by *l_num*.
  - **interface management** *m_num* Management interface specified by *m_num*.
  - **interface port-channel** *p_num* Port-Channel Interface specified by *p_num*.
  - **interface vlan** *v_num* VLAN interface specified by *v_num*.

Example

- This command resets the 802.1x counters on all interfaces.

```
switch#clear dot1x statistics all
switch#
```

dot1x system-auth-control

The **dot1x system-auth-control** command enables 802.1X authentication on the switch.

The **no dot1x system-auth-control** and **default dot1x system-auth-control** commands disables 802.1X authentication by removing the **dot1x system-auth-control** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
dot1x system-auth-control
no dot1x system-auth-control
default dot1x system-auth-control
```

Example

- This command enables 802.1X authentication on the switch.

```
switch(config)#dot1x system-auth-control
switch(config)#
```

- This command disables 802.1X authentication on the switch.

```
switch(config)#no dot1x system-auth-control
switch(config)#
```

dot1x max-reauth-req

The **dot1x max-reauth-req** command configures how many times the switch retransmits an 802.1x Extensible Authentication Protocol (EAP) request packet before ending the conversation and restarting authentication.

The **no dot1x max-reauth-req** and **default dot1x max-reauth-req** commands restore the default value of 2 by deleting the corresponding **dot1x max-reauth-req** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration
Interface-Management Configuration

Command Syntax

```
dot1x max-reauth-req attempts
no dot1x max-reauth-req
default dot1x max-reauth-req
```

Parameters

- *attempts* maximum number of attempts. Values range from 1 to 10; default value is 2.

Examples

- This command sets the 802.1x EAP-request retransmit limit to 6.

```
switch(config-if-Et1)#dot1x max-reauth-req 6
switch(config-if-Et1)#
```

- This command restores the default request repetition value of 2.

```
switch(config-if-Et1)#no dot1x max-reauth-req
switch(config-if-Et1)#
```

dot1x pae authenticator

The **dot1x pae authenticator** command sets the port access entity (PAE) type of the configuration mode interface to ***authenticator***.

The **no dot1x pae authenticator** and **default dot1x pae authenticator** commands restore the switch default by deleting the corresponding **dot1x pae authenticator** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration
Interface-Management Configuration

Command Syntax

```
dot1x pae authenticator
no dot1x pae authenticator
default dot1x pae authenticator
```

Example

- These commands configure on Ethernet interface 2 as a port access entity (PAE) authenticator, which enables IEEE 802.1x on the port.

```
switch(config-if-Et1)#interface ethernet 2
switch(config-if-Et1)#dot1x pae authenticator
switch(config-if-Et1)#
```

- These commands disable IEEE 802.1x authentication on Ethernet interface 2.

```
switch(config-if-Et1)#interface ethernet 2
switch(config-if-Et1)#no dot1x pae authenticator
switch(config-if-Et1)#
```

dot1x port-control

The **dot1x port-control** command configures the configuration mode interface as an authenticator port and specifies whether it will authenticate traffic.

The **no dot1x port-control** and **default dot1x port-control** commands configure the port to pass traffic without authorization by removing the corresponding **dot1x port-control** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration
Interface-Management Configuration

Command Syntax

```
dot1x port-control STATE
no dot1x port-control
default dot1x port-control
```

Parameters

- ***STATE*** specifies whether the interface will authenticate traffic. The default value is ***force-authorized***. Options include:
  - **auto** configures the port to authenticate traffic using Extensible Authentication Protocol messages.
  - **force-authorized** configures the port to pass traffic without authentication.
  - **force-unauthorized** configures the port to block all traffic regardless of authentication.

Examples

- These commands configure Ethernet interface 1 to pass traffic without authentication. This is the default setting.

```
switch(config)#interface Ethernet 1
switch(config-if-Et1)#dot1x port-control force-authorized
switch(config-if-Et1)#
```

- These commands configure Ethernet interface 1 to block all traffic.

```
switch(config)#interface Ethernet 1
switch(config-if-Et1)#dot1x port-control force-unauthorized
switch(config-if-Et1)#
```

- These commands configure Ethernet interface 1 to authenticate traffic using EAP messages.

```
switch(config)#interface Ethernet 1
switch(config-if-Et1)#dot1x port-control auto
switch(config-if-Et1)#
```

dot1x reauthentication

The **dot1x reauthentication** command configures the configuration mode interface to require re-authentication from clients at regular intervals. The interval is set by the **dot1x timeout reauth-period** command.

The **no dot1x reauthentication** and **default dot1x reauthentication** commands restore the default setting by deleting the corresponding **dot1x reauthentication** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration
Interface-Management Configuration

Command Syntax

```
dot1x reauthentication
no dot1x reauthentication
default dot1x reauthentication
```

Example

- These commands configure the Ethernet interface 1 authenticator to require periodic re-authentication from clients.

```
switch(config)#interface Ethernet 1
switch(config-if-Et1)#dot1x reauthentication
switch(config-if-Et1)#
```

dot1x timeout quiet-period

If the switch fails to immediately authenticate the client, the time the switch waits before trying again is specified by the **dot1x timeout quiet-period** command. This timer also indicates how long a client that failed authentication is blocked.

The **no dot1x timeout quiet-period** and **default dot1x timeout quiet-period** commands restore the default quiet period of 60 seconds by removing the corresponding **dot1x timeout quiet-period** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration
Interface-Management Configuration

Command Syntax

```
dot1x timeout quiet-period quiet_time
no dot1x timeout quiet-period
default dot1x timeout quiet-period
```

Parameters

- *quiet_time* interval in seconds. Values range from 1 to 65535. Default value is 60.

Example

- These commands set the 802.1x quiet period for Ethernet interface 1 to 30 seconds.

```
switch(config)#interface Ethernet 1
switch(config-if-Et1)#dot1x timeout quiet-period 30
switch(config-if-Et1)#
```

dot1x timeout reauth-period

The **dot1x timeout reauth-period** command specifies the time period that the configuration mode interface waits before requiring re-authentication from clients.

The **no dot1x timeout reauth-period** and **default dot1x timeout reauth-period** commands restore the default period of 60 minutes by removing the corresponding **dot1x timeout reauth-period** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration
Interface-Management Configuration

Command Syntax

```
dot1x timeout reauth-period reauth_time
no dot1x timeout reauth-period
default dot1x timeout reauth-period
```

Parameters

- *reauth_time* the number of seconds the interface passes traffic before requiring re-authentication. Values range from 1 to 65535. Default value is 3600.

Example

- These commands configure the Ethernet interface 1 authenticator to require re-authentication from clients every 6 hours (21600 seconds).

```
switch(config)#interface Ethernet 1
switch(config-if-Et1)#dot1x reauthentication
switch(config-if-Et1)#dot1x timeout reauth-period 21600
switch(config-if-Et1)#
```

dot1x timeout tx-period

Authentication and re-authentication are accomplished by the authenticator sending an Extensible Authentication Protocol (EAP) request to the supplicant and the supplicant sending a reply which the authenticator forwards to an authentication server. If the authenticator does not get a reply to the EAP request, it waits a specified period of time before retransmitting. The **dot1x timeout tx-period** command configures that wait time.

The **no dot1x timeout tx-period** and **default dot1x timeout tx-period** commands restore the default wait time by removing the corresponding **dot1x timeout tx-period** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration
Interface-Management Configuration

Command Syntax

```
dot1x timeout tx-period tx_time
no dot1x timeout tx-period
default dot1x timeout tx-period
```

Parameters

- *tx_time* Values range from 1 to 65535. Default value is 5.

Example

- These commands configure Ethernet interface 1 to wait 30 seconds before retransmitting EAP requests to the supplicant.

```
switch(config)#interface Ethernet 1
switch(config-if-Et1)#dot1x timeout tx-period 30
switch(config-if-Et1)#
```

show dot1x

The **show dot1x** command displays 802.1x information for the specified interface.

Command Mode

EXEC

Command Syntax

show dot1x *INTERFACE_NAME INFO*

Parameters

- ***INTERFACE_NAME*** Interface type and number. Options include:
  - **all** Display information for all interfaces.
  - **ethernet** *e_num* Ethernet interface specified by *e_num*.
  - **loopback** *l_num* Loopback interface specified by *l_num*.
  - **management** *m_num* Management interface specified by *m_num*.
  - **port-channel** *p_num* Port-Channel Interface specified by *p_num*.
  - **vlan** *v_num* VLAN interface specified by *v_num*.
- ***INFO*** Type of information the command displays. Values include:
  - <no parameter> displays summary of the specified interface.
  - **detail** displays all 802.1x information for the specified interface.

Example

- This command displays 802.1X summary information for Ethernet interface 5.

```
switch#show dot1x interface ethernet 5
Dot1X Information for Ethernet5
-----------------------------------------
PortControl             : auto
QuietPeriod             : 60 seconds
TxPeriod                : 5 seconds
ReauthPeriod            : 3600 seconds
MaxReauthReq            : 2
switch#
```

- This command displays detailed 802.1X information for Ethernet interface 5.

```
switch#show dot1x interface ethernet 5 detail
Dot1X Information for Ethernet5
-----------------------------------------
PortControl             : auto
QuietPeriod             : 60 seconds
TxPeriod                : 5 seconds
ReauthPeriod            : 3600 seconds
MaxReauthReq            : 2

Dot1X Authenticator Client

Port Status             : Unauthorized
switch#
```

show dot1x statistics

The **show dot1x statistics** command displays 802.1X statistics for the specified port or ports.

Command Mode

EXEC

Command Syntax

show dot1x *INTERFACE_NAME* statistics

Parameters

- ***INTERFACE_NAME*** Interface type and number. Options include:
  - **all** Display information for all interfaces.
  - **ethernet** *e_num* Ethernet interface specified by *e_num*.
  - **loopback** *l_num* Loopback interface specified by *l_num*.
  - **management** *m_num* Management interface specified by *m_num*.
  - **port-channel** *p_num* Port-Channel Interface specified by *p_num*.
  - **vlan** *v_num* VLAN interface specified by *v_num*.

Output Fields

- RxStart Number of EAPOL-Start frames received on the port.
- TxReqId Number of EAP-Request/Identity frames transmitted on the port.
- RxVersion Version number of the last EAPOL frame received on the port.
- RxLogoff Number of EAPOL-Logoff frames received on the port.
- RxInvalid Number of invalid EAPOL frames received on the port.
- TxReq Number of transmitted EAP-Request frames that were not EAP-Request/Identity.
- LastRxSrcMAC The source MAC address in the last EAPOL frame received on the port.
- RxRespId The number of EAP-Response/Identity frames received on the port
- RxTotal The total number of EAPOL frames transmitted on the port.
- TxTotal The total number of EAPOL frames transmitted on the port.

Example

- This command displays the 802.1X statistics for ethernet 5

```
switch#show dot1x interface ethernet 5 statistics
Dot1X Authenticator Port Statistics for Ethernet5
-------------------------------------------------
RxStart = 0      RxLogoff = 0    RxRespId = 0
RxStart= 0       RxInvalid = 0   RxTotal = 0
TxReqId = 0      TxReq = 0       TxTotal = 0
RxVersion = 0    LastRxSrcMAC = 0000.0000.0000
switch#
```

show dot1x all summary

The **show dot1x all summary** command displays the IEEE 802.1X status for all ports.

Command Mode

EXEC

Command Syntax

show dot1x all summary

Example

- This command displays the IEEE 802.1X status.

```
switch#show dot1x all summary
Interface               Client                  Status
-----------------------------------------------------------------
Ethernet5               None                    Unauthorized
switch#
```

Chapter 14

DCBX and Flow Control

This chapter describes Data Center Bridging Capability Exchange (DCBX) configuration tasks. Sections in this chapter include:

- Section 14.1: Introduction
- Section 14.2: DCBX and Priority-Based Flow Control Overview
- Section 14.3: DCBX and PFC Configuration and Verification Procedures
- Section 14.4: DCBX and Flow Control Configuration Commands

14.1 Introduction

EOS implements Link Layer Discovery Protocol (LLDP) and the Data Center Bridging Capability Exchange (DCBX) protocol to help automate the configuration of Data Center Bridging (DCB) parameters, including the Priority-Based Flow Control (PFC) standard, which allows an end-to-end flow-control feature.

This feature enables a switch to recognize when it is connected to an iSCSI device and automatically configure the switch link parameters (such as priority flow control) to provide optimal support for that device. DCBX can be used to prioritize the handling of iSCSI traffic to help ensure that packets are not dropped or delayed. DCBX is off by default.

14.2 DCBX and Priority-Based Flow Control Overview

14.2.1 DCBX Description

DCBX works with LLDP to allow switches to exchange information about their data center bridging (DCB) capabilities and configuration and automatically negotiate common Priority-Based Flow Control (PFC) parameters. Data is exchanged in type-length-value (TLV) format. For DCBX to function on an interface LLDP must be enabled on that interface as well.

14.2.2 Priority-Based Flow Control (PFC) Description

PFC uses a new control packet defined in IEEE 802.1Qbb and is not compatible with 802.3x flow control (FC). An interface that is configured for PFC will be disabled for FC. When PFC is disabled on an interface, the FC configuration for the interface becomes active. Any FC frames received on a PFC configured interface are ignored.

Each priority is configured as either drop or no-drop. If a priority that is designated as no-drop is congested, the priority is paused. Drop priorities do not participate in pause.

When PFC is disabled, the interface defaults to the IEEE 802.3x flow control setting for the interface. PFC is disabled by default.

14.3 DCBX and PFC Configuration and Verification Procedures

14.3.1 Set the Priority Rank to the Traffic Class

The **dcbx application priority** command assigns a priority rank to the specified traffic class in the application priority table. This table is transmitted on each DCBX-enabled interface.

Examples

- These commands tell the DCBX peer that iSCSI frames (TCP ports 860 and 3260) should be assigned the given priority of 5.

```
switch(config)#dcbx application tcp-sctp 860 priority 5
switch(config)#dcbx application tcp-sctp 3260 priority 5
```

- These commands specify a different priority for the two iSCSI traffic ports.

```
switch(config)# dcbx application tcp-sctp 860 priority 3
switch(config)# dcbx application tcp-sctp 3260 priority 4
```

- This command is equivalent to the **dcbx application tcp-sctp** command. The DCBX peer that iSCSI frames are assigned are the given the priority 5.

```
switch(config)#dcbx application iscsi priority 5
switch(config)#
```

- These commands prevent the peers from sending anything about the iSCSI frames.

```
switch(config)#no dcbx application tcp-sctp 860 priority 5
switch(config)#no dcbx application tcp-sctp 3260 priority 5
```

14.3.2 Enable Priority-Flow-Control (PFC)

The **priority-flow-control mode** command enables Priority-Flow-Control (PFC) on an individual port.

Examples

- The **priority-flow-control mode** command in DCBX mode enables PFC on an interface.

```
switch(config)#interface ethernet 2
switch(config-if-Et2)#priority-flow-control mode on
```

14.3.3 Set the Priority Flow Control Priority

The **priority-flow-control priority** command in DCBX mode creates a priority group that pauses priority. Each priority is configured as either drop or no-drop. If a priority that is designated as no-drop is congested, the priority is paused. Drop priorities do not participate in pause.

Examples

- The **priority-flow-control priority** command in DCBX mode creates a priority group that pauses priority 5 on Ethernet 2.

```
switch(config)#interface ethernet 2
switch(config-if-Et2)#priority-flow-control mode on
switch(config-if-Et2)# priority-flow-control priority 5 no-drop
```

- To enable lossy behavior, use the drop option of the **priority-flow-control priority** command.

```
switch(config)#interface ethernet 2
switch(config-if-Et2)#priority-flow-control mode on
switch(config-if-Et2)#priority-flow-control priority 5 drop
```

14.3.4 Disable Priority-Flow-Control (PFC)

To disable priority flow control (PFC) on the configuration mode interface and restore the default packet drop setting on the interface, use the **no priority-flow-control** command.

Example

- To disable PFC, use the **no priority-flow-control** command.

```
switch (config)#interface ethernet 2
switch(config-if-Et2)#no priority-flow-control
```

14.3.5 DCBX Verification

To display the DCBX status and the interfaces on which DCBX is enabled, use the **show dcbx** command.

Examples

- This command displays the DCBX status for Ethernet 50.

```
switch#show dcbx Ethernet 50
Ethernet50:
  IEEE DCBX is enabled and active
  Last LLDPDU received on Thu Feb 14 12:06:01 2013
  No priority flow control configuration TLV received
  No application priority configuration TLV received
switch#
```

14.4 DCBX and Flow Control Configuration Commands

Global Configuration Commands

- dcbx application priority
- platform fm6000 pfc-wm

Interface Configuration Commands – Ethernet Interface

- dcbx mode
- no priority-flow-control
- priority-flow-control mode
- priority-flow-control priority

Privileged EXEC Commands

- show dcbx
- show dcbx application-priority-configuration
- show dcbx priority-flow-control-configuration
- show dcbx status
- show interfaces priority-flow-control
- show platform fm6000 pfc-wm
- show priority-flow-control

dcbx application priority

The **dcbx application priority** command assigns a priority rank to the specified traffic class in the application priority table. This table is transmitted on each DCBX-enabled interface.

The **no dcbx application priority** and **default dcbx application priority** commands remove the specified DCBX traffic class – priority assignment by deleting the corresponding **dcbx application priority** command from ***running-config***. When the command does not specify a traffic class, all DCBX traffic class priority assignments are removed.

Command Mode

Global Configuration

Command Syntax

```
dcbx application APPLICATION_TYPE priority rank
no dcbx application [APPLICATION_TYPE priority]
default dcbx application [APPLICATION_TYPE priority]
```

Parameters

- ***APPLICATION_TYPE*** traffic class receiving the priority assignment. Options include:
  - **ether** *ethertype_number* Ethernet traffic. *Ethertype_number* varies from 1536 to 65535.
  - **iscsci** iSCSCI traffic. Maps to TCP/SCTP ports 860 and 3260.
  - **tcp-sctp** *port_number* TCP/SCTP traffic. Port number varies from 1 to 65535.
  - **tcp-sctp-udp** *port_number* TCP/SCTP/UDP traffic. Port number varies from 1 to 65535.
  - **udp** *port_number* UDP traffic. Port number varies from 1 to 65535.
- *rank* priority assigned to traffic class. Values range from 0 to 7.

Examples

- These commands tell the DCBX peer that iSCSI frames (TCP ports 860 and 3260) should be assigned the given priority of 5.

```
switch(config)#dcbx application tcp-sctp 860 priority 5
switch(config)#dcbx application tcp-sctp 3260 priority 5
```

- These commands specify a different priority for the two iSCSI traffic ports.

```
switch(config)# dcbx application tcp-sctp 860 priority 3
switch(config)# dcbx application tcp-sctp 3260 priority 4
```

- This command is equivalent to the **dcbx application tcp-sctp** command. The DCBX peer that iSCSI frames are assigned to is given priority 5.

```
switch(config)#dcbx application iscsi priority 5
switch(config)#
```

- These commands prevent the peers from sending anything about the iSCSI frames.

```
switch(config)#no dcbx application tcp-sctp 860 priority
switch(config)#no dcbx application tcp-sctp 3260 priority
```

dcbx mode

The **dcbx mode** command enables DCBX mode on the configuration mode interface. The switch supports IEEE P802.1Qaz. When DCBX is enabled, two TLVs are added to outgoing LLDPDUs, which instruct the peer on the interface to configure PFC (priority flow control) and the application priority table in the same way as the switch.

The **no dcbx mode, default dcbx mode,** and **dcbx mode none** commands disable DCBX on the configuration mode interface by removing the corresponding **dcbx mode** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration

Command Syntax

```
dcbx mode MODE_NAME
no dcbx mode
default dcbx mode
```

Parameters

- ***MODE_NAME*** Specifies the DCBX version. Options include:
  - **ieee** IEEE version.
  - **cee** Converged Enhanced Ethernet version.
  - **none** DCBX is disabled.

Examples

- These commands enable interface Ethernet 2 to use IEEE DCBX.

```
switch(config)#interface ethernet 2
switch(config-if-Et2)#dcbx mode ieee
switch(config-if-Et2)#
```

- These commands disable DCBX on interface Ethernet 5.

```
switch(config)#interface ethernet 2
switch(config-if-Et2)#dcbx mode none
switch(config-if-Et2)
```

no priority-flow-control

The **no priority-flow-control** and **default priority-flow-control** commands disable the priority flow control (PFC) on the configuration mode interface and restore the default packet drop setting on the interface, which takes effect when PFC is re-enabled. The commands delete all corresponding **priority-flow-control mode** commands from ***running-config***.

Command Mode

Interface-Ethernet Configuration

Command Syntax

```
no priority-flow-control
default priority-flow-control
```

Examples

- These commands disable priority flow control (PFC) on Ethernet interface 3.

```
switch(config)#interface Ethernet 3
switch(config-if-Et3)#no priority-flow-control
switch(config-if-Et3)#
```

platform fm6000 pfc-wm

The **platform fm6000 pfc-wm** command configures the hardware buffer space allocated to the PFC (Priority Flow Control) RX-Private buffer. The command provides options to configure the buffer size and specify when PFC frames are sent to request that a neighbor stop sending traffic. The default values are as follows:

- RX-Private: 18400 bytes
- on (watermark): 9280 bytes
- off (watermark): 1600 bytes

Values that are entered in the command are rounded up to the closest multiple of 160. The ***RX-Private*** value must be greater than the ***off*** value, which must be larger than the ***on*** value.

The no **platform fm6000 pfc-wm** and default **platform fm6000 pfc-wm** commands restore the default settings by removing the **platform fm6000 pfc-wm** command from ***running-config.***

Command Mode

Global Configuration

Command Syntax

```
platform fm6000 pfc-wm [RX-PRIVATE_SIZE] [PFC-ON_WM] [PFC-OFF_WM]
no platform fm6000 pfc-wm
default platform fm6000 pfc-wm
```

The platform **fm6000 pfc-wm** command must explicitly configure at least one parameter.

Parameters

- ***RX-PRIVATE_SIZE*** Specifies size of rx-private buffer. Options include:
  - <no parameter> rx-private buffer retains previously configured size.
  - **rx-private <*18268* to *102400*>** Size of rx-private buffer (bytes).
- ***PFC-ON_WM*** Buffer capacity that triggers the switch to send PFC frames. Options include:
  - <no parameter> Parameter retains previously configured value.
  - **on <*9134* to *102400*>** Buffer capacity that triggers PFC frames (bytes).
- ***PFC-OFF_WM*** Buffer capacity that triggers the switch to stop PFC frame transmissions. Options include:
  - <no parameter> Parameter retains previously configured value.
  - **off <*1536* to *102400*>** Buffer capacity that turns off PFC frames.

Related Commands

- **show platform fm6000 pfc-wm** displays the PFC RX-Private buffer memory allocations

Example

- This command configures the rx-private hardware buffer.

```
switch(config)#platform fm6000 pfc-wm rx-private 24800 on 16000 off 3200
switch(config)#
```

priority-flow-control mode

The **priority-flow-control mode** command enables priority flow control (PFC) on the configuration mode interface to pause selected traffic classes.

The **no priority-flow-control mode** and **default priority-flow-control mode** commands disable PFC on the configuration mode interface by deleting the corresponding **priority-flow-control mode** command from ***running-config***. The **no priority-flow-control** command also disables PFC on the configuration mode interface.

Command Mode

Interface-Ethernet Configuration

Command Syntax

```
priority-flow-control mode on
no priority-flow-control mode [on]
default priority-flow-control mode [on]
```

Example

- These commands enable PFC on Ethernet interface 3.

```
switch(config)# interface Ethernet 3
switch(config-if-Et3)#priority-flow-control mode on
switch(config-if-Et3)#
```

- These commands disable PFC on Ethernet interface 3.

```
switch(config)# interface Ethernet 3
switch(config-if-Et3)# no priority-flow-control mode
switch(config-if-Et3)#
```

priority-flow-control priority

The **priority-flow-control priority** command configures the packet resolution setting on the configuration mode interface. This setting determines if packets are dropped when priority flow control (PFC) is enabled on the interface. Packets are dropped by default.

The **no priority-flow-control priority** and **default priority-flow-control priority** commands restore the default packet drop setting on the configuration mode interface by deleting the corresponding **priority-flow-control priority** command from ***running-config***. The **no priority-flow-control** command also restores the default setting on the configuration mode interface.

Command Mode

Interface-Ethernet Configuration

Command Syntax

```
priority-flow-control priority pack-drop
no priority-flow-control priority
default priority-flow-control priority
```

Parameters

- *pack-drop* denotes the interfaces. Options include:
  - **drop** Packets are dropped. Default setting.
  - **no drop** Packets are not dropped.

Examples

- These commands in DCBX mode create a priority group that pauses dot1p priority 5 on Ethernet 2.

```
switch(config)#interface ethernet 2
switch(config-if-Et2)#priority-flow-control mode on
switch(config-if-Et2)# priority-flow-control priority 5 no-drop
```

- These commands enable lossy behavior.

```
switch(config)#interface ethernet 2
switch(config-if-Et2)#priority-flow-control mode on
switch(config-if-Et2)#priority-flow-control priority 5 drop
```

- These commands remove the priority group that pauses dot1p priority 5 on Ethernet 2.

```
switch(config)#interface ethernet 2
switch(config-if-Et2)# priority-flow-control mode on
switch(config-if-Et2)# no priority-flow-control priority
```

show dcbx

The **show dcbx** command list DCBX status and the interfaces on which DCBX is enabled.

Command Mode

EXEC

Command Syntax

show dcbx [*INTERFACE*]

Parameters

- ***INTERFACE*** Interface type and number. Options include:
  - <no parameter> all configured DCBX interfaces.
  - **ethernet** *e-num* Ethernet interface specified by *e-num*.

Examples

- This command displays the DCBX status for Ethernet 50.

```
switch#show dcbx Ethernet 50
Ethernet50:
  IEEE DCBX is enabled and active
  Last LLDPDU received on Thu Feb 14 12:06:01 2013
  No priority flow control configuration TLV received
  No application priority configuration TLV received
switch#
```

- This command displays the DCBX status for Ethernet 50 when Priority Flow Control (PFC) is not enabled.

```
switch#show dcbx Ethernet 50
Ethernet50:
  IEEE DCBX is enabled and active
  Last LLDPDU received on Thu Feb 14 12:08:29 2013
  - PFC configuration: willing
    not capable of bypassing MACsec
    supports PFC on up to 4 traffic classes
    PFC enabled on priorities: 5 7
    WARNING: peer PFC configuration does not match the local PFC configuration
  - Application priority configuration:
    2 application priorities configured:
      tcp-sctp 860 priority 5
      tcp-sctp 3260 priority 5
switch#
```

show dcbx application-priority-configuration

The **show dcbx application-priority-configuration** command displays the DCBX peer application priority configuration.

Command Mode

EXEC

Command Syntax

show dcbx [*INTERFACE*] application-priority-configuration

Parameters

- ***INTERFACE*** Interface type and number. Options include:
  - <no parameter> All configured DCBX interfaces.
  - **ethernet** *e-num* Ethernet interface specified by *e-num*.

Guidelines

This command and the **show priority-flow-control** command function identically.

Examples

- This command displays the DCBX peer application priority configuration for all DCBX-enabled interfaces.

```
switch# show dcbx application-priority-configuration
Ethernet1:
  Last LLDPDU received on Thu Feb 14 10:52:20 2013
  No application priority configuration TLV received
Ethernet2:
  Last LLDPDU received on Thu Feb 14 10:52:20 2013
  No application priority configuration TLV received
...
Ethernet50:
  Last LLDPDU received on Thu Feb 14 12:08:29 2013
  - Application priority configuration:
    2 application priorities configured:
      tcp-sctp 860 priority 5
      tcp-sctp 3260 priority 5
switch#
```

show dcbx priority-flow-control-configuration

The **show dcbx priority-flow-control-configuration** command displays the IEEE DCBX peer priority flow control configurations.

Command Mode

EXEC

Command Syntax

show dcbx [*INTERFACE*] priority-flow-control-configuration

Parameters

- ***INTERFACE*** Interface type and number. Options include:
  - <no parameter> all configured DCBX interfaces.
  - **ethernet** *e-num* Ethernet interface specified by *e-num*.

Examples

- This command displays the DCBX peer priority flow control configuration for the DCBX-enabled interfaces on the device.

```
switch#show dcbx priority-flow-control-configuration
Ethernet1:
  Last LLDPDU received on Thu Feb 14 10:52:20 2013
  No priority flow control configuration TLV received
Ethernet2:
  Last LLDPDU received on Thu Feb 14 10:52:20 2013
  No priority flow control configuration TLV received
...
Ethernet50:
  Last LLDPDU received on Thu Feb 14 12:11:29 2013
  - PFC configuration: willing
    not capable of bypassing MACsec
    supports PFC on up to 4 traffic classes
    PFC enabled on priorities: 5 7
    WARNING: peer PFC configuration does not match the local PFC configuration
switch#
```

show dcbx status

The **show dcbx status** command displays the DCBX status on the interfaces on which DCBX is enabled.

Command Mode

EXEC

Command Syntax

show dcbx [*INTERFACE*] status

Parameters

- ***INTERFACE*** Interface type and number. Options include:
  - <no parameter> all configured DCBX interfaces.
  - **ethernet** *e-num* Ethernet interface specified by *e-num*.

Examples

- This command displays the DCBX status for the DCBX-enabled interfaces.

```
switch#show dcbx status
Ethernet1:
  Last LLDPDU received on Thu Feb 14 10:52:20 2013
Ethernet2:
  Last LLDPDU received on Thu Feb 14 10:52:20 2013
Ethernet50:
  IEEE DCBX is enabled and active
  Last LLDPDU received on Thu Feb 14 12:11:54 2013
switch#
```

show interfaces priority-flow-control

The **show interfaces priority-flow-control** command displays the status of PFC on all interfaces.

Command Mode

EXEC

Command Syntax

```
show interfaces [INTERFACE] priority-flow-control [INFO_LEVEL]
```

Parameters

- ***INTERFACE*** Interface type and numbers. Options include:
  - <no parameter> Display information for all interfaces.
  - **ethernet** *e_range* Ethernet interface range specified by *e_range*.
  - **loopback** *l_range* Loopback interface specified by *l_range*.
  - **management** *m_range* Management interface range specified by *m_range*.
  - **port-channel** *p_range* Port-Channel Interface range specified by *p_range*.
  - **vlan** *v_range* VLAN interface range specified by *v_range*.
  - **vxlan** *vx_range* VXLAN interface range specified by *vx_range*.

    Valid range formats include number, number range, or comma-delimited list of numbers and ranges.
- ***INFO_LEVEL*** specifies the type of information displayed. Options include:
  - <no parameter> Displays information about all DCBX neighbor interfaces.
  - **status** Displays the DCBX status.
  - **counters** Displays the DCBX counters.

Guidelines

This command and the **show priority-flow-control** command function identically.

Examples

- This command displays the PFC for all interfaces.

```
switch#show interfaces priority-flow-control
The hardware supports PFC on priorities 0 1 2 3 4 5 6 7

Port    Enabled Priorities Active Note
Et1     No                 No
Et2     No                 No
...
Et50    Yes          5     Yes
...
Port                RxPfc           TxPfc
Et1                     0               0
Et2                     0               0
...
Et50                    0               0
...
switch#
```

show platform fm6000 pfc-wm

The **show platform fm6000 pfc-wm** command displays the buffer space allocated to the RX-Private buffer and buffer levels that trigger PFC frame transmission activities.

Command Mode

Privileged EXEC

Command Syntax

```
show platform fm6000 pfc-wm
```

Related Commands

- **platform fm6000 pfc-wm** specifies the PFC RX-Private buffer memory allocation.

Example

- This command displays the rx-private hardware buffer memory allocation.

```
switch#show platform fm6000 pfc-wm
Pfc_Rx_Private_WM: 24800 Bytes
Pfc_On_WM: 16000 Bytes
Pfc_Off_WM: 3200 Bytes
switch#
```

show priority-flow-control

The **show priority-flow-control** command displays the status of PFC on all interfaces.

Command Mode

EXEC

Command Syntax

```
show priority-flow-control [INT_NAME] [INFO_LEVEL]
```

Parameters

- ***INT_NAME*** Denotes the interfaces to be configured. Options include:
  - <no parameter> Displays information for all interfaces.
  - **interface ethernet** *e_range* Ethernet interface range.
  - **interface loopback** *l_range* Loopback interface range.
  - **interface management** *m_range* Management interface range.
  - **interface port-channel** *c_range* Channel group interface range.
  - **interface vlan** *v_range* VLAN interface range.

    Valid *e_range*, *l_range*, *m_range*, *c_range*, and *v_range* formats include a number, number range, or comma-delimited list of numbers and ranges.
- ***INFO_LEVEL*** Specifies level of information detail provided by the command.
  - <no parameter> Displays information about all DCBX neighbor interfaces.
  - **status** Displays the DCBX status.
  - **counters** Displays the DCBX counters.
  - **counters detail** Displays the DCBX counters for each priority class. This option is available only on Trident switches.

Guidelines

This command and the **show interfaces priority-flow-control** command function identically.

Examples

- This command displays the status of PFC on all interfaces.

```
switch#show priority-flow-control
The hardware supports PFC on priorities 0 1 2 3 4 5 6 7

Port    Enabled Priorities Active Note
Et1     No                 No
Et2     No                 No
...
Et50    Yes          5     Yes
...
Port                RxPfc           TxPfc
Et1                    0               0
Et2                    0               0
...
Et50                   0               0
...
switch#
```

Chapter 15

LLDP

This chapter describes initial configuration and recovery tasks. Refer to the Command Descriptions for information about commands used in this chapter.

This chapter contains these sections:

- Section 15.1: LLDP Introduction
- Section 15.2: LLDP Overview
- Section 15.3: LLDP Configuration Procedures
- Section 15.4: LLDP Configuration Commands

15.1 LLDP Introduction

Link Layer Discovery Protocol (LLDP) lets Ethernet network devices to advertise details about themselves, such as capabilities, identification, and device configurations to directly connected devices on the network that are also using LLDP.

15.2 LLDP Overview

LLDP is a discovery protocol that allows devices to advertise information about themselves to peer devices that are on the same physical LAN and store information about the network. LLDP allows a device to learn higher layer management reachability and connection endpoint information from adjacent devices.

Each switch with an active LLDP agent sends and receives messages on all physical interfaces enabled for LLDP transmission. These messages are sent periodically and are typically configured for short time intervals to ensure that accurate information is always available. These messages are then stored for a configurable period of time, and contained within the received packet. The message information expires and is discarded when the configured value is met. The only other time an advertisement is sent is when a relevant change takes place in the switch. If information changes for any reason, the LLDP agent is notified and will send out and update the new values.

15.2.1 LLDP Data Units

A single LLDP Data Unit (LLDPDU) is transmitted in a single 802.3 Ethernet frame. The basic LLDPDU includes a header and a series of type-length-value elements (TLVs). Each TLV advertises different types of information, such as its device ID, type, or management addresses.

LLDP advertises the following TLVs by default:

- port-description
- system-capabilities
- system-description
- system-name
- management-address
- port-vlan

15.2.2 Transmission and Reception

Every device that uses LLDP has its own LLDP agent. The LLDP agent is responsible for the reception, transmission, and management of LLDP. When LLDP is enabled on a port, transmission and reception of LLDPDUs are both enabled by default, but the agent can be configured to only transmit or only receive.

Transmission

When LLDP transmission is enabled, the LLDP agent advertises information about the switch to neighbors at regular intervals. Each transmitted LLDPDU contains the mandatory TLVs, and any enabled optional TLVs.

Reception

When LLDP reception is enabled, the LLDP agent receives and stores advertised information from neighboring devices.

15.2.3 Storing LLDP Information

Whenever the switch receives a valid and current LLDP advertisement from a neighbor, it stores the information in a Simple Network Management Protocol (SNMP) management information base (MIB).

15.2.4 Guidelines and Limitations

LLDP has the following configuration limitations:

- LLDP must be enabled globally before it can be enabled on an interface.
- LLDP is not supported on virtual interfaces.
- LLDP can discover only one device per port.

15.3 LLDP Configuration Procedures

These sections describe the following configuration processes:

- Section 15.3.1: Enabling LLDP Globally
- Section 15.3.2: Enabling LLDP on an Interface
- Section 15.3.3: Optional LLDP Parameters
- Section 15.3.4: Clearing LLDP Statistics
- Section 15.3.5: Displaying LLDP Information

15.3.1 Enabling LLDP Globally

The **lldp run** command globally enables LLDP on the Arista switch. Once LLDP is enabled, the switch will transmit advertisements from the ports that are configured to send TLVs. The neighbor information table is populated as advertisements from the neighbors arrive on the ports.

Example

- This command enables LLDP globally on the Arista switch.

```
switch(config)# lldp run
switch(config)#
```

15.3.2 Enabling LLDP on an Interface

When enabling LLDP, it is enabled on all interfaces by default. By using the **lldp transmit** and **lldp receive** commands, LLDP can be enabled or disabled on individual interfaces or configured to only send or only receive LLDP packets.

Examples

- These commands enable Ethernet port 3/1 to transmit LLDP packets.

```
switch(config)# interface ethernet 3/1
switch(config-if-Et3/1)# lldp transmit
switch(config-if-Et3/1)#
```

- These commands enable Ethernet port 3/1 to receive LLDP packets.

```
switch(config)# interface ethernet 3/1
switch(config-if-Et3/1)# lldp receive
switch(config-if-Et3/1)#
```

15.3.3 Optional LLDP Parameters

The following sections describe these tasks:

- Section 15.3.3.1: Setting the LLDP Timer
- Section 15.3.3.2: Setting the LLDP Hold Time
- Section 15.3.3.3: Setting the LLDP Re-initialization Timer
- Section 15.3.3.4: Setting the IP Management Address to be used in the TLV
- Section 15.3.3.5: Selecting the LLDP TLV

15.3.3.1 Setting the LLDP Timer

The **lldp timer** command specifies the time in seconds between LLDP updates sent by the switch.

Examples

- This command specifies that the LLDP updates should be sent every 120 seconds.

```
switch(config)# lldp timer 120
switch(config)#
```

- This command reverts the LLDP timer to its default value of 30 seconds.

```
switch(config)# no lldp timer 120
switch(config)#
```

15.3.3.2 Setting the LLDP Hold Time

The **lldp holdtime** command sets the amount of time a receiving device should retain the information sent by the device.

Examples

- This command specifies that the receiving device should retain the information for 180 seconds before discarding it.

```
switch(config)# lldp holdtime 180
switch(config)#
```

- This command reverts the LLDP hold time and to the default value of 120 seconds.

```
switch(config)# no lldp holdtime 180
switch(config)#
```

15.3.3.3 Setting the LLDP Re-initialization Timer

The **lldp reinit** command specifies the amount in time in seconds to delay the re-initialization attempt by the switch.

Example

- This command specifies that the switch waits 10 seconds before attempting to re-initialize.

```
switch(config)# lldp reinit 10
switch(config)#
```

15.3.3.4 Setting the IP Management Address to be used in the TLV

The **lldp management-address** command specifies the IP management address or the IP address of the VRF interface in LLDP type-length-value (TLV) triplets.

Example

- This command specifies the IP management address to be used in the TLV.

```
switch(config)# lldp management-address ethernet 3/1
switch(config)#
```

15.3.3.5 Selecting the LLDP TLV

The **lldp tlv-select** command configures the type, length, and value (TLV) to send and receive in LLDP packets. The **no lldp tlv-select** command removes the TLV configuration.

Example

- This command enables the system descriptions to be included in the TLVs.

```
switch(config)# lldp tlv-select system-description
switch(config)#
```

15.3.4 Clearing LLDP Statistics

- Section 15.3.4.1: Clear LLDP Counters
- Section 15.3.4.2: Clear LLDP Table

15.3.4.1 Clear LLDP Counters

The **clear lldp counters** command resets the LLDP traffic counters to zero.

Example

- This command resets the traffic counters to zero.

```
switch# clear lldp counters
switch#
```

15.3.4.2 Clear LLDP Table

The **clear lldp table** command clears neighbor information from the LLDP table.

Example

- This command clears neighbor information from the LLDP table.

```
switch# clear lldp table
switch#
```

15.3.5 Displaying LLDP Information

- Section 15.3.5.1: Viewing LLDP Global Information
- Section 15.3.5.2: Viewing LLDP Local Information
- Section 15.3.5.3: Viewing LLDP Neighbors
- Section 15.3.5.4: Viewing LLDP Traffic

15.3.5.1 Viewing LLDP Global Information

The **show lldp** command displays LLDP information.

Examples

- This command displays global information about LLDP.

```
switch# show lldp
LLDP transmit interval      : 60 seconds
LLDP transmit holdtime      : 120 seconds
LLDP reinitialization delay : 2 seconds
LLDP Management Address VRF : default

Enabled optional TLVs:
  Port Description
  System Name
  System Description
  System Capabilities
  Management Address (Management0)
  IEEE802.1 Port VLAN ID
  IEEE802.3 Link Aggregation
  IEEE802.3 Maximum Frame Size

Port       Tx Enabled  Rx Enabled
Et3/1      Yes         Yes
             <-------OUTPUT OMITTED FROM EXAMPLE-------->
switch#
```

- This command displays LLDP information.

```
switch# show lldp ethernet interface 3/1
LLDP transmit interval      : 30 seconds
LLDP transmit holdtime      : 120 seconds
LLDP reinitialization delay : 2 seconds
LLDP Management Address VRF : default

Enabled optional TLVs:
  Port Description
  System Name
  System Description
  System Capabilities
switch#
```

15.3.5.2 Viewing LLDP Local Information

The **show lldp local-info** command displays the information contained in the LLDP TLVs to be sent about the local system.

Example

- This command displays information contained in the TLVS about the local systems.

```
switch# show lldp local-info management 1
Local System:
  - Chassis ID type: MAC address (4)
    Chassis ID     : 001c.730f.11a8
  - System Name: "switch.aristanetworks.com"
  - System Description: "Arista Networks EOS version 4.13.2F running on an Arista
Networks DCS-7150S-64-CL"
  - System Capabilities : Bridge, Router
    Enabled Capabilities: Bridge

Interface Management1:
  - Port ID type: Interface name (5)
    Port ID     : "Management1"
  - Port Description: ""
  - Management Address Subtype: IPv4 (1)
    Management Address        : 172.22.30.154
    Interface Number Subtype  : ifIndex (2)
    Interface Number          : 999001
    OID String                :
  - IEEE802.1 Port VLAN ID: 0
  - IEEE802.1/IEEE802.3 Link Aggregation
    Link Aggregation Status: Not Capable (0x00)
    Port ID                : 0
  - IEEE802.3 Maximum Frame Size: 1518 bytes
switch(config)#
```

15.3.5.3 Viewing LLDP Neighbors

The **show lldp neighbors** command displays information about LLDP neighbors.

Example

- This command shows information about LLDP neighbors.

```
switch# show lldp neighbor
Last table change time   : 0:12:33 ago
Number of table inserts  : 33
Number of table deletes  : 0
Number of table drops    : 0
Number of table age-outs : 0

Port      Neighbor Device ID             Neighbor Port ID          TTL
Et3/1     tg104.sjc.aristanetworks.com   Ethernet3/2               120

Ma1/1     dc1-rack11-tor1.sjc            1/1                       120
switch#
```

Example

- This command displays detailed information about the neighbor Ethernet 3/1.

```
switch# show lldp neighbor ethernet 3/1
Last table change time   : 0:16:24 ago
Number of table inserts  : 33
Number of table deletes  : 0
Number of table drops    : 0
Number of table age-outs : 0

Port      Neighbor Device ID              Neighbor Port ID           TTL
Et3/1     tg104.sjc.aristanetworks.com    Ethernet3/2                120
switch#
```

15.3.5.4 Viewing LLDP Traffic

The **show lldp traffic** command displays the LLDP traffic information for the switch.

Example

- This command displays the LLDP counters on the switch.

```
switch# show lldp traffic
Port          Tx Frames Tx Length Exceeded

Et20              69485                 0
Et21              69394                 0
Et22              69203                 0
Et23              57546                 0
Et24                  0                 0
Ma1               69665                 0
Port      Rx Frames     Rx Errors    Rx Discard  TLVs Discard  TLVs Unknown

Et20          69470             0             0             0             0
Et21          69383             0             0             0             0
Et22          69143             0             0             0             0
Et23          55370             0             0             0             0
Et24              0             0             0             0             0
Ma1           69078         69078             0         69078             0
switch#
```

15.4 LLDP Configuration Commands

Global Configuration Commands

- lldp holdtime
- lldp management-address
- lldp management-address vrf
- lldp reinit
- lldp run
- lldp timer
- lldp tlv-select

Interface Configuration Commands – Ethernet Interface

- lldp receive
- lldp transmit

Privileged EXEC Commands

- clear lldp counters
- clear lldp table

EXEC Commands

- show lldp
- show lldp local-info
- show lldp neighbors
- show lldp traffic

clear lldp counters

The **clear lldp counters** command resets the LLDP counters to zero.

Command Mode

Privileged EXEC

Command Syntax

clear lldp counters [*SCOPE*]

Parameters

- ***SCOPE*** Session affected by command. Options include:
  - <no parameter> command affects counters on all CLI sessions.
  - **session** clears LLDP counters for the current CLI session only.

Examples

- This command resets all the LLDP counters to zero.

```
switch(config)# clear lldp counters
switch(config)#
```

- This command resets only the LLDP counters for the current CLI session.

```
switch(config)# clear lldp counters session
switch(config)#
```

clear lldp table

The **clear lldp table** command clears neighbor information from the LLDP table.

Command Mode

Privileged EXEC

Command Syntax

```
clear lldp table
```

Example

- This command clears neighbor information from the LLDP table.

```
switch(config)# clear lldp table
switch(config)#
```

lldp holdtime

The **lldp holdtime** command specifies the amount of time a receiving device should maintain the information sent by the device before discarding it.

Command Mode

Global Configuration

Command Syntax

```
lldp holdtime period
no lldp holdtime
default lldp holdtime
```

Parameters

- *period* The amount of time a receiving device should hold LLDPDU information before discarding it. Value ranges from 10 to 65535 second; default value is 120 seconds.

Examples

- This command sets the amount of time before the receiving device discards LLDPDU information to 180 seconds.

```
switch(config)# lldp holdtime 180
switch(config)#
```

- This command restores the hold time to its default value of 120 seconds.

```
switch(config)# no lldp holdtime 180
switch(config)#
```

lldp management-address

The **lldp management-address** command enables the user to add the IP management address used for LLDP type-length-value (TLV).

Command Mode

Global Configuration

Command Syntax

```
lldp management-address INTERFACE
no lldp management-address [INTERFACE]
default lldp management-address [INTERFACE]
```

Parameters

- ***INTERFACE*** Interface type and number. Options include:
  - **all** all interfaces.
  - **ethernet** *e_num* Ethernet interface specified by *e_num*.
  - **loopback** *l_num* Loopback interface specified by *l_num*.
  - **management** *m_num* Management interface specified by *m_num*.
  - **port-channel** *p_num* Port-Channel Interface specified by *p_num*.
  - **vlan** *v_num* VLAN interface specified by *v_num*.

Examples

- This command specifies the IP management address to be used in the TLV.

```
switch(config)# lldp management-address ethernet 3/1
switch(config)#
```

- This command removes the IP management address used in the TLV.

```
switch(config)# no lldp management-address ethernet 3/1
switch(config)#
```

- This command specifies that VLAN 200 is used in the TLV.

```
switch(config)# lldp management-address vlan 200
switch(config)#
```

- This command removes the VLAN ID used in the TLV.

```
switch(config)# no lldp management-address vlan 200
switch(config)#
```

lldp management-address vrf

The **lldp management-address vrf** command enables the user to add the IP address of the VRF interface used in LLDP type-length-value (TLV).

Command Mode

Global Configuration

Command Syntax

```
lldp management-address vrf VRF_INSTANCE
no lldp management-address vrf VRF_INSTANCE
default lldp management-address vrf VRF_INSTANCE
```

Parameters

- ***VRF_INSTANCE*** specifies the VRF instance.

Examples

- This command specifies the management address VRF to be used in the TLV.

```
switch(config)# lldp management-address vrf test 1
switch(config)#
```

- This command removes the management VRF used in the TLV.

```
switch(config)# no lldp management-address vrf test 1
switch(config)#
```

lldp receive

The **lldp receive** command enables LLDP packets on an interface. The **no lldp receive** command disables the acceptance of LLDP packets.

Command Mode

Interface-Ethernet configuration
Interface-Management configuration

Command Syntax

```
lldp receive
no lldp receive
default lldp receive
```

Examples

- These commands enable the reception of LLDP packets on Ethernet interface 4/1.

```
switch(config)#interface ethernet 4/1
switch(config-if-Et4/1)#lldp receive
switch(config-if-Et4/1)#
```

- These commands disable LLDP the reception of LLDP packets on Ethernet interface 4/1.

```
switch(config)#interface ethernet 4/1
switch(config-if-Et4/1)# no lldp receive
switch(config-if-Et4/1)#
```

lldp reinit

The **lldp reinit** command sets the time delay in seconds for LLDP to initialize.

Command Mode

Global Configuration

Command Syntax

```
lldp reinit delay
no lldp reinit
default lldp reinit
```

Parameters

- *delay* the amount of time the device should wait before re-initialization is attempted. Value ranges from 1 to 20 seconds; default value is 2 seconds.

Examples

- This command specifies that the switch should wait 10 seconds before attempting to re-initialize.

```
switch(config)# lldp reinit 10
switch(config)#
```

- This command restores the default initialization delay of 2 seconds.

```
switch(config)# no lldp reinit 10
switch(config)#
```

lldp run

The **lldp run** command enables LLDP on the Arista switch.

Command Mode

Global Configuration

Command Syntax

```
lldp run
no lldp run
default lldp run
```

Examples

- This command enables LLDP globally on the Arista switch.

```
switch(config)# lldp run
switch(config)#
```

- This command disables LLDP globally on the Arista switch.

```
switch(config)# no lldp run
switch(config)#
```

lldp timer

The **lldp timer** command specifies the amount of time a receiving device should maintain the information sent by the device before discarding it. The **no lldp timer** command removes the configured LLDP timer.

Command Mode

Global Configuration

Command Syntax

lldp timer *transmission_time*
no lldp timer
default lldp timer

Parameters

- *transmission_time* the period of time at which LLDPDUs are transmitted. Values range from 5 to 32768 seconds; the default is 30 seconds.

Examples

- This command configures a period of 80 seconds at which the LLDPDUs are transmitted.

```
switch(config)# lldp timer 180
switch(config)#
```

- This command removes the configured period of time at which the LLDPDUs are transmitted.

```
switch(config)# no lldp timer 180
switch(config)#
```

lldp tlv-select

The **lldp tlv-select** command allows the user to specify the type-length-values (TLVs) to include in LLDP packets.

Command Mode

Global Configuration

Command Syntax

```
lldp tlv-select TLV_NAME
no lldp tlv-select TLV_NAME
default lldp tlv-select TLV_NAME
```

Parameters

- ***TLV_NAME*** Options include:
  - **link-aggregation** specifies the link aggregation TLV.
  - **management-address** specifies the management address TLV.
  - **max-frame-size** specifies the Frame size TLV.
  - **port-description** specifies the port description TLV.
  - **port-vlan** specifies the port VLAN ID TLV.
  - **system-capabilities** specifies the system capabilities TLV.
  - **system-description** specifies the system description TLV.
  - **system-name** specifies the system name TLV.

Example

- This command enables the system description TLV:

```
switch(config)# lldp tlv-select system-description
switch(config)#
```

- This command disables the system description TLV:

```
switch(config)# no lldp tlv-select system-description
switch(config)#
```

- This command enables the max-frame-size TLV:

```
switch(config)# lldp tlv-select max-frame-size
switch(config)#
```

- This command disables the max-frame-size TLV:

```
switch(config)# no lldp tlv-select max-frame-size
switch(config)#
```

lldp transmit

The **lldp transmit** command enables the transit of LLDP packets on an interface.

Command Mode

Interface-Ethernet configuration
Interface-Management configuration

Command Syntax

```
lldp transmit
no lldp transmit
default lldp transmit
```

Examples

- These commands enable the transmission of LLDP packets

```
switch(config)#interface ethernet 4/1
switch(config-if-Et4/1)#lldp transmit
switch(config-if-Et4/1)#
```

- These commands disable the transmission of LLDP packets.

```
switch(config)#interface ethernet 4/1
switch(config-if-Et4/1)#no lldp transmit
switch(config-if-Et4/1)#
```

show lldp

The **show lldp** command displays LLDP information.

Command Mode

EXEC

Command Syntax

show lldp [*INTERFACE*]

Parameters

- ***INTERFACE*** Interface type and numbers. Options include:
  - <no parameter> Display information for all interfaces.
  - **ethernet** *e_range* Ethernet interface range specified by *e_range*.
  - **management** *m_range* Management interface range specified by *m_range*.

    Valid *e_range* and *m_range* formats include number, number range, or comma-delimited list of numbers and ranges.

Examples

- This command displays all LLDP information.

```
switch# show lldp
LLDP transmit interval      : 60 seconds
LLDP transmit holdtime      : 120 seconds
LLDP reinitialization delay : 2 seconds
LLDP Management Address VRF : test

Enabled optional TLVs:
  Port Description
  System Name
  System Description
  System Capabilities
  Management Address (Management0)
  IEEE802.1 Port VLAN ID
  IEEE802.3 Link Aggregation
  IEEE802.3 Maximum Frame Size

Port       Tx Enabled  Rx Enabled
Et3/1      Yes         Yes
             <-------OUTPUT OMITTED FROM EXAMPLE-------->
switch#
```

- This command displays specific information about LLDP for Ethernet interface 3/1.

```
switch# show lldp ethernet 3/1
LLDP transmit interval      : 30 seconds
LLDP transmit holdtime      : 120 seconds
LLDP reinitialization delay : 2 seconds
LLDP Management Address VRF : default

Enabled optional TLVs:
  Port Description
  System Name
  System Description
  System Capabilities
switch#
```

- This command displays specific information about LLDP for management interface 1/1.

```
switch# show lldp management 1/1
LLDP transmit interval      : 60 seconds
LLDP transmit holdtime      : 120 seconds
LLDP reinitialization delay : 2 seconds
LLDP Management Address VRF : default

Enabled optional TLVs:
  Port Description
  System Name
  System Description
  System Capabilities
  Management Address (Management0)
  IEEE802.1 Port VLAN ID
  IEEE802.3 Link Aggregation
  IEEE802.3 Maximum Frame Size

Port       Tx Enabled  Rx Enabled
Ma1/1      Yes         Yes
switch#
```

show lldp local-info

The **show lldp local-info** command displays LLDP errors and overflows.

Command Mode

EXEC

Command Syntax

show lldp local-info [*INTERFACE*]

Parameters

- ***INTERFACE*** Interface type and numbers. Options include:
  - <no parameter> Display information for all interfaces.
  - **ethernet** *e_range* Ethernet interface range specified by *e_range*.
  - **management** *m_range* Management interface range specified by *m_range*.

    Valid *e_range* and *m_range* formats include number, number range, or comma-delimited list of numbers and ranges.

Example

- This command displays the specific LLDP errors and overflows on management interface 1.

```
switch# show lldp local-info management 1
Local System:
  - Chassis ID type: MAC address (4)
    Chassis ID     : 001c.730f.11a8qqq
  - System Name: "switch.aristanetworks.com"
  - System Description: "Arista Networks EOS version 4.13.2F running on an Arista
Networks DCS-7150S-64-CL"
  - System Capabilities : Bridge, Router
    Enabled Capabilities: Bridge

Interface Management1:
  - Port ID type: Interface name (5)
    Port ID     : "Management1"
  - Port Description: ""
  - Management Address Subtype: IPv4 (1)
    Management Address        : 172.22.30.154
    Interface Number Subtype  : ifIndex (2)
    Interface Number          : 999001
    OID String                :
  - IEEE802.1 Port VLAN ID: 0
  - IEEE802.1/IEEE802.3 Link Aggregation
    Link Aggregation Status: Not Capable (0x00)
    Port ID                : 0
  - IEEE802.3 Maximum Frame Size: 1518 bytes
se505.16:01:44#
switch#
```

show lldp neighbors

The **show llpd neighbors** command displays information about the switch's LLDP neighbors.

Command Mode

EXEC

Command Syntax

show lldp neighbors [*INTERFACE*] [*INFO_LEVEL*]

Parameters

- ***INTERFACE*** Interface type and numbers. Options include:
  - <no parameter> displays information for all interfaces.
  - **ethernet** *e_range* Ethernet interface range specified by *e_range*.
  - **management** *m_range* Management interface range specified by *m_range*.

    Valid *e_range* and *m_range* formats include number, number range, or comma-delimited list of numbers and ranges.
- ***INFO_LEVEL*** amount of information that is displayed. Options include:
  - <no parameter> Displays information for all interfaces.
  - **detailed** LLDP information for all the adjacent LLDP devices.

Examples

- This command displays the neighbor's information about LLDP.

```
switch(config)# show lldp neighbor
Last table change time   : 0:12:33 ago
Number of table inserts  : 33
Number of table deletes  : 0
Number of table drops    : 0
Number of table age-outs : 0

Port      Neighbor Device ID             Neighbor Port ID           TTL
Et3/1     tg104.sjc.aristanetworks.com   Ethernet3/2                120
             <-------OUTPUT OMITTED FROM EXAMPLE-------->
Ma1/1     dc1-rack11-tor1.sjc            1/1                        120
switch#
```

- This command displays the neighbor's information about LLDP for Ethernet interface 3/1.

```
switch# show lldp neighbor ethernet 3/1
Last table change time   : 0:16:24 ago
Number of table inserts  : 33
Number of table deletes  : 0
Number of table drops    : 0
Number of table age-outs : 0

Port      Neighbor Device ID             Neighbor Port ID           TTL
Et3/1     tg104.sjc.aristanetworks.com   Ethernet3/2                120
switch#
```

- This command displays detailed neighbor's information about LLDP.

```
switch# show lldp neighbor 3/1 detail

             <-------OUTPUT OMITTED FROM EXAMPLE-------->
Interface Ethernet 3/1 detected 1 LLDP neighbors:

  Neighbor 001c.7300.1506/Ethernet6/25, age 8 seconds
  Discovered 5 days, 3:58:58 ago; Last changed 5 days, 3:56:57 ago
    - Chassis ID type: MAC address (4)
      Chassis ID     : 001c.7300.1506
    - Port ID type: Interface name (5)
      Port ID     : "Ethernet6/25"
    - Time To Live: 120 seconds
    - System Name: "Leaf-Switch1.aristanetworks.com"
    - System Description: "Arista Networks EOS version 4.10.1-SSO running on an
Arista Networks DCS-7504"
    - System Capabilities : Bridge, Router
      Enabled Capabilities: Bridge
    - Management Address Subtype: IPv4 (1)
      Management Address        : 172.22.30.116
      Interface Number Subtype  : ifIndex (2)
      Interface Number          : 999999
      OID String                :
    - IEEE802.1 Port VLAN ID: 1
    - IEEE802.1/IEEE802.3 Link Aggregation
      Link Aggregation Status: Capable, Disabled (0x01)
      Port ID                : 0
    - IEEE802.3 Maximum Frame Size: 9236 bytes
switch#
```

show lldp traffic

The **show lldp traffic** command displays LLDP traffic information for the switch.

Command Mode

EXEC

Command Syntax

show lldp traffic [*INTERFACE*]

Parameters

- ***INTERFACE*** Interface type and numbers. Options include:
  - <no parameter> Display information for all interfaces.
  - **ethernet** *e_range* Ethernet interface range specified by *e_range*.
  - **management** *m_range* Management interface range specified by *m_range*.

    Valid *e_range* and *m_range* formats include number, number range, or comma-delimited list of numbers and ranges.

Example

- This command displays the LLDP counters on the switch.

```
switch# show lldp traffic

Port          Tx Frames Tx Length Exceeded
             <-------OUTPUT OMITTED FROM EXAMPLE-------->
Et20              69485                  0
Et21              69394                  0
Et22              69203                  0
Et23              57546                  0
Et24                  0                  0
Ma1               69665                  0

Port          Rx Frames     Rx Errors    Rx Discard  TLVs Discard  TLVs Unknown
             <-------OUTPUT OMITTED FROM EXAMPLE-------->
Et20              69470             0             0             0             0
Et21              69383             0             0             0             0
Et22              69143             0             0             0             0
Et23              55370             0             0             0             0
Et24                  0             0             0             0             0
Ma1               69078         69078             0         69078             0
switch#
```

Chapter 16

Data Transfer

Arista switches support the transfer of packets (network layer) and frames (data link layer). This chapter describes concepts and processes that are referenced by routing and switching protocols that Arista switches support.

Sections in this chapter include:

- Section 16.1: Data Transfer Introduction
- Section 16.2: Data Transfer Methods
- Section 16.3: MAC Address Table
- Section 16.4: Configuring Ports
- Section 16.5: Monitoring Links
- Section 16.6: Data Transfer Command Descriptions

16.1 Data Transfer Introduction

Arista switches transfer data through switching, routing, and layer 3 switching. This chapter provides an introduction to these transfer methods.

Data structures and processes that support data transfer methods and referenced in specific protocol chapters are also described, including:

- routed ports
- switched ports
- MAC address table
- port mirroring
- storm control
- loopback interfaces
- route redistribution
- null0 interfaces
- MTUs

16.2 Data Transfer Methods

This section describes these data transfer methods:

- Section 16.2.1: Switching and Bridging
- Section 16.2.2: Routing
- Section 16.2.3: Layer 3 Switching

16.2.1 Switching and Bridging

Switching and bridging operations transmit data link layer frames between devices within a single subnet. Each port is assigned a 48 bit Media Access Control (MAC) address. Frames arriving at a hub are bridged, or sent to all other ports on the subnet. Switches can associate ports with their MAC addresses, obviating the need to flood the subnet when sending a frame.

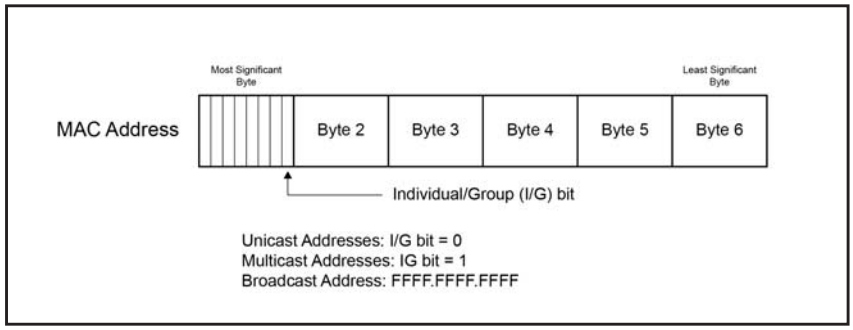
Subnets in the switch are defined by VLANs. A virtual local area network (VLAN) is a group of devices that are configured to communicate as if they are attached to the same network regardless of their physical location. Chapter describes VLANS.

Four MAC address types identify the scope of LAN interfaces that an address represents:

- unicast: represents a single interface.
- broadcast: represents all interfaces.
- multicast: represents a subset of all interfaces.
- reserved: assigned to nodes that have no configured MAC address.

The Individual/Group (I/G) bit distinguishes unicast MAC addresses from multicast addresses. As shown in Figure 16-1, the I/G bit is the least significant bit of the most significant byte in a MAC address.

Figure 16-1: MAC Address Format



- Unicast address: the I/G bit is 0: 1234.1111.1111 is a unicast MAC address (the most significant byte is an even number).
  - Reserved address: all bits set to 0 (0000.0000.0000).
- Multicast address: the I/G bit is 1: 1134.1111.1111 is a multicast MAC address (the most significant byte is an odd number).
- Broadcast address: all bits set to 1 (FFFF.FFFF.FFFF).

Example

- The following are unicast MAC addresses:

```
0200.0000.0000
1400.0000.0000
```

- The following are multicast MAC addresses:

```
0300.0000.0000
2500.0000.0000
```

The following sections describe MAC address functions and data structures.

- Section 10.5.2 describes the process of assigning a MAC address to an interface.
- Section 16.3 describes the MAC Address table.

16.2.2 Routing

Routing transmits network layer packets over connected independent subnets. Each subnet is assigned an IP address range and each device on the subnet is assigned an IP address from that range. Connected subnets have IP address ranges that do not overlap. A router connects multiple subnets. Routers forward inbound packets to the subnet whose address range includes the packets' destination address.

IPv4 and IPv6 are internet layer protocols that facilitate packet-switched networking, including transmissions across multiple networks.

These chapters describe available IP features:

- **IPv4**
- **IPv6**

16.2.2.1 Static Routing

Static routes are entered through the CLI and are typically used when dynamic protocols are unable to establish routes to a specified destination prefix. Static routes are also useful when dynamic routing protocols are not available or appropriate.

Creating a static route associates a destination IP address with a local interface. The routing table refers to these routes as ***connected*** routes that are available for redistribution into routing domains defined by dynamic routing protocols.

These sections describe static route configuration commands:

- Section 24.1.2: IPv4 Address Configuration
- Section 25.3.1.2: Configuring Default and Static IPv6 Routes

16.2.2.2 Dynamic Routing

Dynamic routes are established by dynamic routing protocols. These protocols also maintain the routing table and modify routes to adjust for topology or traffic changes. Routing protocols assist the switch in communicating with other devices to exchange network information, maintaining routing tables, and establishing data paths.

The switch supports these dynamic routing protocols:

- **Open Shortest Path First – Version 2**
- **Open Shortest Path First – Version 3**
- **Border Gateway Protocol (BGP)**
- **Routing Information Protocol**
- **IS-IS**

16.2.3 Layer 3 Switching

Layer 3 switches establish data paths through routing processes (Layer 3) and transfer data as a switch (Layer 2) through speed-optimized hardware. Layer 3 switches use a control plane (routing) and data plane (switching) to manage these processes.

16.2.3.1 Control plane

The control plane builds and maintains the IP routing table, which identifies IP packet routes in terms of destination addresses. The routing table defines a route by its next hop address and the egress interface that accesses the next hop.

The control plane derives routing information from three sources:

- Status of physical and virtual interfaces on the switch.
- Static routes entered through the CLI.
- Routes established through dynamic routing protocols.

Applying an ACL to the Control Plane

Note On hardware purchased after July 3, 2017, control plane ACLs are not available.

The control plane supports routing and management functions, handling packets that are addressed to the switch without regard to any switch interface.

To apply an IP ACL to the control plane, enter **ip access-group (Control Plane mode)** in control-plane mode. The **control-plane** command places the switch in control-plane mode.

Chapter 20: ACLs and Route Maps describes access control lists.

Example

- These commands place the switch in control-plane mode and assigns ***CP-Test1*** to the control plane.

```
switch(config)#control-plane
switch(config-cp)#ip access-group CP-Test1 in
switch(config-cp)#
```

16.2.3.2 Data plane

The data plane routes IP packets based on information derived by the control plane. Each packet's path includes Layer 2 addresses that reach its next hop destination. The data plane also performs other operations required by IP routing, such as recalculating IP header checksums and decrementing the time-to-live (TTL) field.

Arista data planes support these packet forwarding modes:

- Store and forward: the switch accumulates entire packets before forwarding them.
- Cut through: the switch begins forwarding frames before their reception is complete.

Cut through mode reduces switch latency at the risk of decreased reliability. Packet transmissions can begin immediately after the destination address is processed. Corrupted frames may be forwarded because packet transmissions begin before CRC bytes are received.

Packet forwarding mode availability varies by switch platform:

- Arad: store and forward mode only

- FM6000: both modes are available.
- Petra: store and forward mode only
- Trident: both modes are available.
- Trident-II: both modes are available.

The data plane is also referred to as the forwarding plane.

Data Plane Forwarding Mode Configuration

The **switch forwarding-mode** command specifies the forwarding mode of the switch's data plane. This command is available on Trident, Trident-II, and FM6000 platform switches. The forwarding mode is ***store-and-forward*** on Arad and Petra platform switches.

Example

- This command changes the forwarding mode to *store-and-forward*.

```
switch(config)#switch forwarding-mode store-and-forward
switch(config)#
```

The **show switch forwarding-mode** command displays the switch’s forwarding mode.

Example

- This command displays the switch’s forwarding mode.

```
switch(config)#show switch forwarding-mode
Current switching mode:    store and forward
Available switching modes: cut through, store and forward
```

16.3 MAC Address Table

The switch maintains a MAC address table for switching frames efficiently between ports. The MAC address table contains static and dynamic MAC addresses.

- Static MAC addresses are entered into the table through a CLI command.
- Dynamic MAC addresses are entered into the table when the switch receives a frame whose source address is not listed in the MAC address table. The switch builds the table dynamically by referencing the source address of frames it receives.

16.3.1 MAC Address Table Configuration

These sections describe MAC address table configuration tasks.

- Section 16.3.1.1: Static MAC Address Table Entries
- Section 16.3.1.2: Dynamic MAC Address Table Entries

16.3.1.1 Static MAC Address Table Entries

The MAC address table accepts static MAC addresses, including multicast entries. Each table entry references a MAC address, a VLAN, and a list of layer 2 (Ethernet or port channel) ports. The table supports three entry types: unicast drop, unicast, and multicast.

- A drop entry does not include a port.
- A unicast entry includes one port.
- A multicast entry includes at least one port.

  Packets with a MAC address (source or destination) and VLAN specified by a drop entry are dropped. Drop entries are valid for only unicast MAC addresses.

The **mac address-table static** command adds a static entry to the MAC address table.

Example

- This command adds a static entry for unicast MAC address 0012.3694.03ec to the MAC address table.

```
switch(config)#mac address-table static 0012.3694.03ec vlan 3 interface Ethernet
7
switch(config)#show mac address-table static
          Mac Address Table
------------------------------------------------------------------

Vlan    Mac Address       Type        Ports      Moves   Last Move
----    -----------       ----        -----      -----   ---------
   3    0012.3694.03ec    STATIC      Et7
Total Mac Addresses for this criterion: 1

          Multicast Mac Address Table
------------------------------------------------------------------

Vlan    Mac Address       Type        Ports
----    -----------       ----        -----
Total Mac Addresses for this criterion: 0

switch(config)#
```

- This command adds the static entry for the multicast MAC address 0112.3057.8423 to the MAC address table.

```
switch(config)#mac address-table static 0112.3057.8423 vlan 4 interface
port-channel 10 port-channel 12
switch(config)#show mac address-table
          Mac Address Table
------------------------------------------------------------------

Vlan    Mac Address       Type        Ports      Moves   Last Move
----    -----------       ----        -----      -----   ---------
Total Mac Addresses for this criterion: 0

          Multicast Mac Address Table
------------------------------------------------------------------

Vlan    Mac Address       Type        Ports
----    -----------       ----        -----
   4    0112.3057.8423    STATIC      Po10 Po12
Total Mac Addresses for this criterion: 1
switch(config)#
```

16.3.1.2 Dynamic MAC Address Table Entries

Learning Mode

The switch maintains a MAC address table for switching frames efficiently between VLAN ports. When the switch receives a frame, it associates the MAC address of the transmitting interface with the recipient VLAN and port. When MAC address learning is enabled for the recipient port, the entry is added to the MAC address table. When MAC address learning is not enabled, the entry is not added to the table.

The **switchport mac address learning** command enables MAC address learning for the configuration mode interface. MAC address learning is enabled by default on all Ethernet and port channel interfaces.

Example

- These commands disables MAC address learning for Ethernet interface 8, then displays the active configuration for the interface.

```
switch(config)#interface ethernet 8
switch(config-if-Et8)#no switchport mac address learning
switch(config-if-Et8)#show active
interface Ethernet8
no switchport mac address learning
switch(config-if-Et8)#
```

Aging Time

Aging time defines the period an entry is in the table, as measured from the most recent reception of a frame on the entry's VLAN from the specified MAC address. The switch removes entries when their presence in the MAC address table exceeds the aging time.

Aging time ranges from 10 to 1,000,000 seconds with a default of 300 seconds (five minutes).

Example

- This command sets the MAC address table aging time to two minutes (120 seconds).

```
switch(config)#mac address-table aging-time 120
switch(config)#
```

The **mac address-table aging-time** command configures the aging time for MAC address table dynamic entries. Aging time defines the period an entry is in the table, as measured from the most recent reception of a frame on the entry's VLAN from the specified MAC address. The switch removes entries when their presence in the MAC address table exceeds the aging time.

Clearing Dynamic Addresses

The **clear mac address-table dynamic** command removes specified dynamic entries from the MAC address table. Entries are identified by their VLAN and layer 2 (Ethernet or port channel) interface.

Example

- This command clears all dynamic mac address table entries for port channel 5 on VLAN 34.

```
switch(config)#clear mac address-table dynamic vlan 34 interface port-channel 5
switch(config)
```

16.3.2 Displaying the MAC Address Table

The **show mac address-table** command displays the specified MAC address table entries.

Example

- This command displays the MAC address table.

```
switch#show mac address-table
          Mac Address Table
------------------------------------------------------------------

Vlan    Mac Address       Type        Ports      Moves   Last Move
----    -----------       ----        -----      -----   ---------
 101    001c.8224.36d7    DYNAMIC     Po2        1       9 days, 15:57:28 ago
 102    001c.8220.1319    STATIC      Po1
 102    001c.8229.a0f3    DYNAMIC     Po1        1       0:05:05 ago
 661    001c.8220.1319    STATIC      Po1
 661    001c.822f.6b22    DYNAMIC     Po7        1       0:20:10 ago
3000    001c.8220.1319    STATIC      Po1
3000    0050.56a8.0016    DYNAMIC     Po1        1       0:07:38 ago
3909    001c.8220.1319    STATIC      Po1
3909    001c.822f.6a80    DYNAMIC     Po1        1       0:07:08 ago
3911    001c.8220.1319    STATIC      Po1
3911    001c.8220.40fa    DYNAMIC     Po8        1       1:19:58 ago
3912    001c.822b.033e    DYNAMIC     Et11       1       9 days, 15:57:23 ago
3913    001c.8220.1319    STATIC      Po1
3913    001c.822b.033e    DYNAMIC     Po1        1       0:04:35 ago
3984    001c.8220.178f    DYNAMIC     Et8        1       4 days, 15:07:29 ago
3992    001c.8220.1319    STATIC      Po1
3992    001c.8221.07b9    DYNAMIC     Po6        1       4 days, 15:13:15 ago
Total Mac Addresses for this criterion: 24

          Multicast Mac Address Table
------------------------------------------------------------------

Vlan    Mac Address       Type        Ports
----    -----------       ----        -----
Total Mac Addresses for this criterion: 0
switch#
```

16.4 Configuring Ports

This section describes these port properties:

- Section 16.4.1: Port Mirroring
- Section 16.4.2: Storm Control
- Section 16.4.3: Switched and Routed Ports
- Section 16.4.4: Loopback Ports
- Section 16.4.5: MAC Security
- Section 16.4.6: Null0 Interface
- Section 16.4.7: Maximum Transmission Units (MTU)

16.4.1 Port Mirroring

Port mirroring, also known as port monitoring, is the duplication of traffic from a collection of source ports to a destination port. A mirror session correlates a set of source ports to a destination port.

Valid mirror sources are Ethernet or port channel interfaces, including port channels which are part of an MLAG. Mirror destination ports are usually Ethernet interfaces; port channel destination ports are also supported on some platforms.

Note On platforms which support the use of port channels as mirror destinations, a port channel *must not* be used as a mirror destination if it is a member of an MLAG.

Mirror ports cannot be routed ports or Ethernet ports that are port channel members. Mirroring is deactivated on Ethernet mirror ports that are subsequently added to a port channel. Layer 2 control protocols run normally on source ports; PDU traffic is mirrored identically to data traffic. Layer 2 control protocols do not run on destination ports.

An interface cannot be in a more than one mirror session and cannot be simultaneously a source and destination. By default, mirror sessions duplicate ingress and egress traffic but are configurable to mirror traffic from only one direction.

- Ingress Mirroring: Packets received by a source port are duplicated, including all valid data frames and L2 control PDUs. Ports mirror data before forwarding logic is applied. Packets subsequently dropped because of forwarding decisions are mirrored.
- Egress Mirroring: Packets transmitted by a source port are duplicated, with these exceptions:
  - Flooded/Multicast Packets: Packets sent to multiple mirror ports generate one copy, except in multi-chip devices when the mirror source and destination ports are on different chips; in this case, an extra copy is generated.
  - Dropped Packets: Packets dropped by forwarding decisions (such as output STP state checks) on egress sources are not duplicated. Packets dropped because of congestion may be duplicated.

VLAN tags on duplicate packets from an egress source are identical to tags on inbound source packets.

When a packet's path through the switch includes multiple mirror source ports in different mirror sessions, the traffic is duplicated once and sent to the destination of the highest numbered session.

16.4.1.1 Port Mirroring Capacity

Port mirroring capacity varies by platform. This section describes session limits for each platform.

FM6000 Platform Switches

- ***Maximum Number of Sessions:*** 4
- ***Session Sources:*** Ethernet interfaces (any number), Port channel interfaces (any number)
- ***Session Destinations:*** Ethernet interfaces (any number), Port channel interfaces (any number), CPU
- Egress IP ACL on destination port is supported

  Sessions can mirror Rx, Tx, or both ways without impacting the number of available sessions.

  Implementing any of the following reduces the number available sessions by one: ACL Logging, MLAG Peer Link, sFlow, VTEP Learning (VXLAN), LANZ Sampling

Arad Platform Switches

- ***Maximum Number of Sessions:*** 16
- ***Session Sources:*** Ethernet interfaces (any number), Port channel interfaces (any number)
- ***Session Destinations:*** Ethernet interfaces (one)
- Egress IP ACL on Destination Port is supported for Rx sessions

  Sessions can mirror Rx, Tx, or both ways without impacting number of available sessions.

  Although the number of configured source interfaces is unlimited, the number of interfaces that can be effectively mirrored is restricted by the destination port speed.

Petra Platform Switches

- ***Maximum Number of Sessions:*** 16
- ***Session Sources:*** Ethernet interfaces (eight for Rx or Tx sessions; four for both ways)
- ***Session Destinations:*** Ethernet interfaces (eight for Rx or Tx sessions; four for both ways)
- Egress IP ACL on Destination Port is not supported

  Sessions can mirror Rx, Tx, or both ways without impacting number of available sessions.

Trident Platform Switches

- ***Maximum Number of Sessions:*** 4
- ***Session Sources:*** Ethernet interfaces (any number), Port channel interfaces (any number)
- ***Session Destinations:*** Ethernet interfaces (one)
- Egress IP ACL on Destination Port is supported

  Mirroring Rx or Tx requires one session. Mirroring both ways requires two sessions.

Trident-II Platform Switches

- ***Maximum Number of Sessions:*** 4 per chip
- ***Session Sources:*** Ethernet interfaces (any number), Port channel interfaces (any number)
- ***Session Destinations:*** Ethernet interfaces (one)
- Egress IP ACL on Destination Port is supported

  Mirroring Rx or Tx requires one session. Mirroring both ways requires two sessions.

16.4.1.2 Configuring Mirror Ports

Mirror sessions associate a set of source ports to a destination port using the **monitor session source** and **monitor session destination** commands. An interface cannot be used in more than one mirror session and cannot be simultaneously a source and a destination. By default, mirror sessions duplicate ingress and egress traffic but are configurable to mirror traffic from one direction. On Trident and Trident-II platform switches (DCS-7050, DCS-7050X, DCS-7250X, and DCS-7300X series), all frames mirrored on egress are prefixed with an 802.1Q VLAN tag, even when the egress port is configured as an access port. If the capture device cannot process VLAN tags properly, mirroring should be configured exclusively for ingress traffic by specifying **rx** in the **monitor session source** command.

Filtering on TX traffic in a mirror session is not supported.

Example

- These commands configure Ethernet interface 7 as the source port and Ethernet interface 8 as the destination port for the redirect_1 mirroring session. The session mirrors ingress and egress traffic.

```
switch(config)#monitor session redirect_1 source ethernet 7
switch(config)#monitor session redirect_1 destination ethernet 8
```

The **show monitor session** command displays the configuration of the specified port mirroring session.

Example

- This command shows the configuration of the ***redirect_1*** mirroring session.

```
switch(config)#show monitor session

Session redirect_1
------------------------

Source Ports

  Both:        Et7

Destination Port: Et8

switch(config)#
```

The **monitor session ip access-group** command configures an ACL to filter the traffic being mirrored to the destination port.

Example

- These commands create an ACL and apply it to filter the traffic mirrored to the destination port by session “redirect_1.”

```
switch(config)#ip access-list allow-host
switch(config-acl-allow-host)#10 permit ip host 192.168.11.24 host 10.0.215.23
switch(config-acl-allow-host)#20 deny ip any any
switch(config-acl-allow-host)#exit
switch(config)#monitor session redirect_1 ip access-group allow-host
switch(config)#
```

16.4.1.3 Filtered Mirroring to CPU

Filtered mirroring to CPU adds a special destination to port mirroring that allows mirrored traffic to be sent to the switch supervisor. The traffic can then be monitored and analyzed locally without the need of a remote port analyzer. Filtered mirroring to CPU can also be used for debugging and

troubleshooting, configured to mirror RX traffic, TX traffic or both, with up to 14 mirroring profiles used simultaneously. In addition, mirroring to CPU uses control plane protection to limit the rate of the traffic sent to the CPU.

Note On hardware purchased after July 3, 2017, CoPP is not available.

Examples

- These commands configure the source for normal mirroring and the destination to CPU.

```
switch(config)#monitor session mySession source ethernet 3/1 both
switch(config)#monitor session mySession destination cpu
switch(config)#
```

- These commands configure reserved bandwidth and shape rate of mirrored traffic.

```
switch(config)#policy-map type control-plane copp-system-policy
switch(config-pmap-control-plane-copp-system-policy)#class
copp-system-mirroring
switch(config-pmap-c-copp-system-policy-copp-system-mirroring)#bandwidth kbps
2000
switch(config-pmap-c-copp-system-policy-copp-system-mirroring)#shape kbps 4000
switch(config-pmap-c-copp-system-policy-copp-system-mirroring)#
```

- These commands show the current status of mirroring to CPU from the CLI, and display the control plane protection configuration for mirroring to CPU.

```
switch(config)#show monitor session
Session mySession
------------------------
Source Ports:
  Both : Et3/1
Destination Ports:
  Cpu : active (mirror0)
switch(config)#
```

- These commands show the current status of mirroring to CPU from the CLI, and display the control plane protection configuration for mirroring to CPU.

```
switch(config)#show policy-map type control-plane copp-system-policy class
cop-system-mirroring
  Class-map: copp-system-mirroring (match-any)
     shape : None
     bandwidth : None
switch(config)#
```

16.4.2 Storm Control

A traffic storm is a flood of packets entering a network, resulting in excessive traffic and degraded performance. Storm control prevents network disruptions by limiting traffic beyond specified thresholds on individual physical LAN interfaces.

Storm control monitors inbound traffic levels over one-second intervals and compares the traffic level with a specified benchmark.

Storm control has three modes:

- ***Storm control all***: When inbound traffic exceeds the specified threshold within a one-second control interval, all traffic is dropped until the end of the interval.
- ***Storm control broadcast***: When inbound broadcast traffic exceeds the specified threshold within a one-second control interval, broadcast traffic is dropped until the end of the interval.

- ***Storm control multicast***: When inbound multicast traffic exceeds the specified threshold within a one-second control interval, multicast traffic is dropped until the end of the interval.

Broadcast and multicast storm control are independent features and can be enabled simultaneously. The ***storm control all*** threshold overrides broadcast and multicast thresholds.

Storm Control Configuration

The **storm-control** command configures and enables broadcast or multicast storm control on the configuration mode interface. The command provides three mode options:

- **storm-control all** unicast, multicast, and broadcast inbound packet control.
- **storm-control broadcast** broadcast inbound packet control.
- **storm-control multicast** multicast inbound packet control.

An interface configuration can contain three storm-control statements, one with each mode setting. The **storm-control all** threshold overrides broadcast and multicast thresholds.

When storm control is enabled, the switch monitors inbound traffic levels over one second intervals and compares the traffic level with a specified threshold. The threshold is a percentage of the total available port bandwidth and is configurable on each interface for each transmission mode.

Example

- These commands enable multicast storm control on Ethernet interfaces 2 through 4 and set a threshold of 65%. During each one second interval, the interface drops inbound multicast traffic in excess of 65% of capacity.

```
switch(config)#interface ethernet 2 / 3 / 4
switch(config-if-Et4/4/4)#storm-control multicast level 65
switch(config-if-Et4/4/4)#
```

Example

- These commands clear multicast storm control on Ethernet interfaces 2 through 4.

```
switch(config)#interface ethernet 2 / 3 / 4
switch(config-if-Et2/3/4)#no storm-control multicast
switch(config-if-Et2/3/4)#
```

Example

- These commands enable broadcast storm control on Ethernet interfaces 2 through 4 and set broadcast traffic to 50%. During each one second interval, the interface drops inbound multicast traffic in excess of 50% of capacity.

```
switch(config)#interface ethernet 2 / 3 / 4
switch(config-if-Et2/3/4)#storm-control broadcast level 50
switch(config-if-Et2/3/4)#
```

Example

- These commands enable broadcast storm control on Ethernet interfaces 2 through 4 and set a threshold of 5000 packets per second (pps).

```
switch(config)#interface ethernet 2 / 3 / 4
switch(config-if-Et2/3/4)#storm-control broadcast level pps 5000
switch(config-if-Et2/3/4)#
```

Example

- These commands clear broadcast storm control on Ethernet interfaces 2 through 4.

```
switch(config)#interface ethernet 2 / 3 / 4
switch(config-if-Et2/3/4)#no storm-control broadcast
switch(config-if-Et2/3/4)#
```

The **show storm-control** command displays the storm-control level and interface inbound packet capacity for the specified interface.

Example

- This command displays the storm control configuration for Ethernet ports 2 through 4.

```
switch(config-if-Et2/3/4)#show storm-control
Port            Type     Level   Units Rate(Mbps)   Status   Drops Reason
Et2/3/4          all     75.00       %       7500   active       0
           multicast     55.00       %       5500   active       0
           broadcast     50.00       %       5000   active       0
switch(config-if-Et2/3/4)#
```

16.4.3 Switched and Routed Ports

A switched port is an Ethernet or port channel interface that is configured as a layer 2 interface. Switched ports bridge frames and are assigned to at least one VLAN. Switched ports are not associated with any IP addresses. By default, Ethernet and port channel interfaces are in switched port mode.

A routed port is an Ethernet or port channel interface that is configured as a layer 3 interface. Routed ports do not bridge frames and are not members of any VLANs. Routed ports can have IP addresses assigned to them and packets are routed directly to and from the port.

Configuring an interface as a routed port is similar to creating a VLAN with spanning-tree disabled, making the port the only member of that VLAN and configuring the IP address on the switch virtual interface (SVI) associated with the VLAN.

All IP-level interface configuration commands, except **autostate** and **ip virtual-router,** can be used to configure a routed interface. If the interface is reverted to switched port mode, ***running-config*** maintains IP level interface configuration statements. These changes become active again if the interface is configured back to routed port mode.

A LAG that is created with the **channel-group** command inherits the mode of the member port. A LAG created from a routed port becomes a routed LAG. IP-level configuration is not propagated to the LAG from its component members.

The broadcast queue towards the CPU is shared among all interfaces of the forwarding chip. Broadcast storm on a single port adversely impacts other interfaces of the same chip by potentially dropping even low rate broadcast frames. Routed port storm control attempts to mitigate this effect by performing storm control on the broadcast frames for routed ports.

Routed Port Configuration

The switching-routing configuration of Ethernet and port channel interfaces is specified by the **switchport** and **no switchport** commands. These commands only toggle the interface between switched and routed modes. They have no effect on other configuration states.

The **no switchport** command places the configuration mode interface in ***routed port*** mode. Routed ports behave as Layer 3 interfaces. They do not bridge packets and are not VLAN members. An IP address can be assigned to a routed port for the direct routing of packets to and from the interface.

When an interface is configured as a routed port, the switch transparently allocates an internal VLAN whose only member is the routed interface. Internal VLANs are created in the range from 1006 to 4094. VLANs that are allocated internally for a routed interface cannot be directly created or configured. Section 18.3.4: Allocating Internal VLANs describes VLAN allocation configuration procedures.

Example

- This command places Ethernet interface 5 in routed port mode.

```
switch(config)#interface ethernet 5
switch(config-if-Et5)#no switchport
```

Switched Port Configuration

The **switchport** command places the configuration mode interface in ***switched port*** (Layer 2) mode. Switched ports are configurable as members of one or more VLANs through other switchport commands. Switched ports ignore all IP level configuration commands, including IP address assignments. By default, Ethernet and port channel interfaces are switched ports.

Example

- This command places Ethernet interface 5 in switched port mode.

```
switch(config)#interface ethernet 5
switch(config-if-Et5)#switchport
```

The **switchport default mode routed** command places the configuration mode interface for a switch with all ports in ***switched port*** (Layer 3) routed mode, changing the switch with all ports from **switchport default mode access**.

Examples

- This command places a switch with all ports in routed mode.

```
switch(config)#switchport default mode routed
```

- This command places a switch with all ports in access mode.

```
switch(config)#switchport default mode access
```

16.4.4 Loopback Ports

A loopback interface is a virtual network interface implemented in software and does not connect to any hardware. Traffic sent to the loopback interface is immediately received on the sending interface. The switch provides loopback configuration mode for creating loopback interfaces and modifying their operating parameters.

Internet protocols reserve specific addresses for loopback network segments:

- IPv4 designates 127/8 as loopback subnet, which includes 127.0.0.0 through 127.255.255.255.
- IPv6 designates ::1/128 as the loopback address, which includes 0:0:0:0:0:0:0:1 (also written as ::1).

Arista switches support the configuration of 1001 loopback interfaces, numbered from 0 to 1000.

Loopback Interface Configuration

Loopback ports are instantiated by entering loopback interface mode for the desired port number. Loopback interface mode also provides access to loopback configuration commands. Previously instantiated ports are edited by entering loopback interface mode for the specified port.

The **interface loopback** command places the switch in interface-loopback configuration mode for the specified interfaces, creating loopback interfaces for each specified port not previously created.

Example

- These commands instantiate loopback interface 2 and assigns it IP address 10.1.1.42/24.

```
switch(config)#interface loopback 2
switch(config-if-Lo2)#ip address 10.1.1.42
switch(config-if-Lo2)#show active
interface Loopback2
   ip address 10.1.1.42/24
switch(config-if-Lo2)#
```

16.4.5 MAC Security

MAC security restricts input to a switched port by limiting the number of MAC addresses that can access the port. Ports with MAC security enabled restrict traffic to a limited number of hosts, as determined by their MAC addresses. When the limit is exceeded, the port becomes errdisabled.

Port Security Configuration

MAC address security is enabled by **switchport port-security**. The default MAC address limit on an interface where port security is enabled is one; to change that default limit, use the **switchport port-security maximum** command.

Example

- These commands enable MAC security on Ethernet interface 7, set the maximum number of assigned MAC addresses to 2, assign two static MAC addresses to the interface, and clear the dynamic MAC addresses for the interface.

```
switch(config)#interface ethernet 7
switch(config-if-Et7)#switchport port-security
switch(config-if-Et7)#switchport port-security maximum 2
switch(config-if-Et7)#exit
switch(config)#mac address-table static 0034.24c2.8f11 vlan 10 interface ethernet
7
switch(config)#mac address-table static 4464.842d.17ce vlan 10 interface ethernet
7
switch(config)#clear mac address-table dynamic interface ethernet 7
switch(config)#show port-security
Secure Port      MaxSecureAddr  CurrentAddr  SecurityViolation  Security Action
                    (Count)        (Count)      (Count)
------------------------------------------------------------------------------
     Et7              2             2            0            Shutdown
------------------------------------------------------------------------------
Total Addresses in System: 1
switch(config)#show port-security address
          Secure Mac Address Table
------------------------------------------------------------------
Vlan    Mac Address       Type                      Ports   Remaining Age
                                                              (mins)
----    -----------       ----                      -----   -------------
  10    0034.24c2.8f11    SecureConfigured          Et7     N/A
  10    4464.842d.17ce    SecureConfigured          Et7     N/A
------------------------------------------------------------------------
Total Mac Addresses for this criterion: 2
switch(config)#
```

16.4.6 Null0 Interface

The null0 interface is a virtual interface that drops all inbound packets. A null0 route is a network route whose destination is ***null0 interface.*** Inbound packets to a null0 interface are not forwarded to any valid address. Many interface configuration commands provide ***null0*** as an interface option.

16.4.7 Maximum Transmission Units (MTU)

The MTU of a communications protocol refers to the size in bytes of the largest frame (Ethernet) or packet (IP) that can be sent on the network. Different protocols support a variety of MTU sizes. Most IP over Ethernet implementations use the Ethernet V2 frame format, which specifies an MTU of 1500 bytes. Jumbo frames are Ethernet frames containing more than 1500 bytes.

16.4.7.1 Switching interface MTU size

On Arista devices, layer two interfaces (either trunk or access ports) are set with a default ethernet MTU of 9236 bytes. This value cannot be changed and is derived as follows: 9214 + 6 (source MAC) + 6 (dst MAC) + 4 (VLAN tag) + 2 (ether type) + 4 (crc) totals 9236 bytes.

The output of a **show interface** command for a layer two interface displays the following:

Trunk

```
Ethernet1 is up, line protocol is up (connected)
 Hardware is Ethernet, address is 001c.731c.5073 (bia 001c.731c.5073)
 Ethernet MTU 9214 bytes , BW 1000000 kbit
```

Access

```
Ethernet3 is up, line protocol is up (connected)
 Hardware is Ethernet, address is 001c.731c.5075 (bia 001c.731c.5075)
 Ethernet MTU 9214 bytes , BW 1000000 kbit
```

Important! The value 9214 displayed here is NOT the maximum frame size but rather the maximum size of the IP packet that can be encapsulated within a frame leaving the interface.

16.4.7.2 Routing Interface MTU Size

The MTU size on layer 3 interfaces varies between a minimum of 68 to the maximum 9214 bytes. The default size is 1500 bytes. The **show interface** output for a layer three interface displays the following:

VLAN routed interface

```
Vlan100 is up, line protocol is up (connected)
 Hardware is Vlan, address is 001c.731c.5072 (bia 001c.731c.5072)
 Internet address is 10.1.1.2/24
 Broadcast address is 255.255.255.255
 Address determined by manual configuration
 IP MTU 9214 bytes
```

Physical routed interface

```
Ethernet4 is down, line protocol is down (connect)
 Hardware is Ethernet, address is 001c.731c.5072
 Internet address is 10.10.10.10/24
 Broadcast address is 255.255.255.255
 Address determined by manual configuration
 IP MTU 9214 bytes
```

Important! The value 9214 in these outputs is the maximum *IP* ***packet*** size that the interface can transmit or receive.

Important! Some protocols, such as OSPF, may require that neighbor interfaces on physically disparate systems are configured with the same IP MTU.

A routed interface fragments packets that exceed the configured IP MTU on the interface. For example, if a 2000 byte packet is received on routed interface 1 and is forwarded from routed interface 2 then routed interface 2 fragments the packet into a 1500 byte packet plus an additional packet containing the remaining data. This fragmentation should be avoided by configuring a consistent IP MTU across all systems within the operational domain.

The IP MTU set on a routed interface is valid for both IPv4 and IPv6 packets.

MTU Configuration

The **mtu** command configures the IPv4 and IPv6 Maximum Transmission Unit (MTU) size for the configuration mode interface. An interface's MTU value is displayed with the **show interface** command. The command is valid for all routable interfaces.

Example

- This command sets the MTU size of 1492 bytes on VLAN interface 20.

```
switch(config-if-Vl20)#mtu 1492
switch(config-if-Vl20)#
```

- This command displays status for a routed interface.

```
switch(config-if-Et3)#show interface e3
Ethernet3 is up, line protocol is up (connected)
  Hardware is Ethernet, address is 001c.731c.5072
  Internet address is 10.1.1.2/24
  Broadcast address is 255.255.255.255
  Address determined by manual configuration
  IP MTU 1500 bytes , BW 1000000 kbit
  Full-duplex, 1Gb/s, auto negotiation: on, uni-link: unknown
  Up 22 days, 7 hours, 47 minutes, 58 seconds
switch(config)#
```

- Using ping on a Linux host, you can test the maximum transmission through the interface.

```
[user@linux ~]$ ping -M do -s 1472 10.1.1.2
PING 10.1.1.2 (10.1.1.2) 1472(1500) bytes of data.
1480 bytes from 10.1.1.2: icmp_seq=1 ttl=64 time=0.206 ms
1480 bytes from 10.1.1.2: icmp_seq=2 ttl=64 time=0.191 ms
--- 10.1.1.2 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 999ms
rtt min/avg/max/mdev = 0.191/0.198/0.206/0.015 ms
```

The size 1472 has 8 bytes of ICMP information added and 20 bytes of IP headers added, generating a total packet size of 1500 bytes.

- The option ‘-M do’ specifies that fragmentation is prohibited for this test.
- The option ‘-s’ specifies the size of the packet being generated.
- A capture of the frame displays total length of 1514 bytes on the wire which includes the Ethernet headers and type field.

16.5 Monitoring Links

These section describes link monitoring and object tracking processes:

- Section 16.5.1: Object Tracking
- Section 16.5.2: Errdisabled Ports
- Section 16.5.3: Link Flap Monitoring
- Section 16.5.4: Fabric Link Monitoring
- Section 16.5.5: Rapid Automated Indication of Link-Loss

16.5.1 Object Tracking

Object tracking makes it possible for the switch to take action in response to changes in specific switch properties by creating an object to track those properties. When the tracked property changes, the object then changes state, allowing configured agents to react accordingly.

Object Tracking Configuration

The **track** command creates an object that changes state to reflect changes in a specific switch property. Agents configured to track that object are then able to react to the change.

Example

- These commands create an object that tracks the line protocol state on Ethernet interface 8, then configure Ethernet interface 5 to disable VRRP when that tracked object changes state to **down**.

```
switch(config)#track ETH8 interface ethernet 8 line-protocol
switch(config)#interface ethernet 5
switch(config-if-Et5)#vrrp 1 track ETH8 shutdown
switch(config-if-Et5)#
```

These commands use object tracking:

- **link state track**
- **vrrp track**

16.5.2 Errdisabled Ports

The switch places an Ethernet or management interface in ***error-disabled*** state when it detects an error on the interface. ***Error-disabled*** is an operational state that is similar to link-down state. Conditions that error-disable an interface include:

- bpduguard
- link-flap
- no-internal-vlan
- portchannelguard
- portsec
- tapagg
- uplink-failure-detection
- xcvr_unsupported

Most conditions are programmed by the configuration of other features, such as Spanning Tree protocol (bpduguard). Link flap error-disabling is configured through errdisable commands or link flap monitor commands (Section 16.5.3).

Error-disabled interfaces are recovered either through manual or automated methods.

To manually recover an interface, enter its configuration mode and execute **shutdown** and **no shutdown** commands.

Example

- These commands manually recover Ethernet interface 30 from the errdisable state.

```
switch(config)#interface ethernet 30
switch(config-if-Et30)#shutdown
switch(config-if-Et30)#no shutdown
switch(config-if-Et30)#
```

Automated recovery of Ethernet interfaces that are error-disabled by a specified condition is enabled by **errdisable recovery cause**. The **errdisable recovery interval** specifies the period that an interface remains disabled until it is enabled and begins operating normally. When the disabling condition persists, recovered interfaces eventually return to the error-disabled state.

Example

- These commands configure automated recovery for all interfaces that are error-disabled from link flap and bpduguard conditions. Automated recovery begins five minutes after the port is disabled.

```
switch(config)#errdisable recovery cause link-flap
switch(config)#errdisable recovery cause bpduguard
switch(config)#errdisable recovery interval 300
switch(config)#
```

16.5.3 Link Flap Monitoring

Link flap frequency is the quantity of link flaps (connection state changes) over a specified period. Excessive link flaps result in network stability issues, including spanning tree and routing recalculations. Link flaps are often caused by layer 1 issues, such as a bad cable or duplex mismatch. Link flap monitoring specifies link flap thresholds and disables a port when a threshold is exceeded.

Link flap monitoring can be enabled on all interfaces through errdisable link flap commands or on individual interfaces with the link flap monitor.

16.5.3.1 Global Link Flap Monitor

Global link flap detection is configured through two global configuration mode commands:

- **errdisable flap-setting cause link-flap** configures the link-flap frequency that defines link-flap errors on an Ethernet interface.
- **errdisable detect cause link-flap** enables the error-disabling of Ethernet interfaces that exceed the threshold link flap frequency.

Link-flap detection is enabled by default.

Example

- These commands sets the link flap error criteria of 15 connection state changes over a 30 second period, then enables error detection on all interfaces.

```
switch(config)#errdisable flap-setting cause link-flap max-flaps 15 time 30
switch(config)#errdisable detect cause link-flap
switch(config)#
```

16.5.3.2 Interface Link Flap Monitor

An interface is monitored for link flap errors with link flap profiles. A link flap profile specifies conditions that define a link-flap error. Profiles are assigned to Ethernet interfaces. Multiple profiles can be assigned to an interface to monitor a set of error conditions.

The global link flap monitor is used by interfaces that are not individually monitored for link flap errors.

Configuring Link Flap Profiles

Link flap profiles are configuration statements that define a link flap error in terms of these criteria:

- ***flaps*** Threshold number of interface state changes.
- ***period*** Interval when link flaps accumulate to trigger an error condition.
- ***violations*** Number of link flap errors (threshold exceeded over specified period).
- ***intervals*** Quantity of periods.

The **monitor link-flap policy** command places the switch in link-flap configuration mode for configuring link flap profiles and compiling a default-profile set. The **profile max-flaps (Link Flap Configuration)** command configures link flap profiles.

The default-profile set is a list of link-flap profiles that define error-disable criteria for interfaces where link flap monitoring is enabled but link-flap profiles are not assigned. The default-profile set may contain zero, one, or multiple profiles. When the default-profile set is empty, **errdisable flap-setting cause link-flap** specifies default error-disable criteria. When the default-profile set contains multiple profiles, the criteria is satisfied when conditions match any profile.

Example

- These commands enter link flap configuration mode and create four link flap profiles.

```
switch(config)#monitor link-flap policy
switch(config-link-flap)#profile LF01 max-flaps 15 time 60
switch(config-link-flap)#profile LF02 max-flaps 10 time 30 violations 5 intervals
10
switch(config-link-flap)#profile LF03 max-flaps 20 time 75 violations 2 intervals
6
switch(config-link-flap)#profile LF04 max-flaps 30 time 100 violations 4
intervals 7
switch(config-link-flap)#show active
monitor link-flap policy
   profile LF01 max-flaps 15 time 60 violations 1 intervals 1
   profile LF02 max-flaps 10 time 30 violations 5 intervals 10
   profile LF02 max-flaps 20 time 75 violations 2 intervals 6
   profile LF02 max-flaps 30 time 100 violations 4 intervals 7
switch(config-link-flap)#
```

The **default-profiles** command specifies the set of link-flap profiles that define error-disable criteria for interfaces where link flap monitoring is enabled without a link flap profile assignment. Entering a **default-profile** command replaces the current default-profile statement in ***running-config***.

The default-profile set may contain zero, one, or multiple profiles. When the default-profile set is empty, **errdisable flap-setting cause link-flap** specifies default error-disable criteria. When the default-profile set contains multiple profiles, error-disable criteria is satisfied when conditions match any profile. Multiple profiles are assigned to the default-profile set through a single **default-profiles** command.

Example

- This command assigns configures LF01 and LF02 as the default-profile set.

```
switch(config)#monitor link-flap policy
switch(config-link-flap)#default-profiles LF01 LF02
switch(config-link-flap)#show active
monitor link-flap policy
   profile LF01 max-flaps 15 time 60 violations 1 intervals 1
   profile LF02 max-flaps 10 time 30 violations 5 intervals 10
   profile LF02 max-flaps 20 time 75 violations 2 intervals 6
   profile LF02 max-flaps 30 time 100 violations 4 intervals 7
   default-profiles LF01 LF02
switch(config-link-flap)#
```

Interface Link Flap Profile Assignments

Link flap monitoring is enabled on individual Ethernet interfaces and can optionally specify one or more profiles to define link-flap error-disabling criteria. When link flap monitoring is enabled on an interface, the link-flap conditions determine when the interface is error-disabled. Multiple profiles can be assigned to an interface to monitor a set of error conditions; a port is disabled when conditions match any of the profiles assigned to an interface.

The **monitor link-flap profiles** command controls link-flap monitoring on a configuration mode interface. The command provides these link flap detection options:

- **monitor link-flap** (***no profiles listed***): Interface detects link flaps using default-profile set criteria.
- **monitor link-flap** (***at least one profile listed***): Interface detects link flaps using listed profile criteria.
- **default monitor link-flap**: The interface uses global link flap monitor commands (Section 16.5.3.1).
- **no monitor link-flap**: The interface does not detect link flaps.

Examples

- This command assigns LF03 and LF04 link flap profiles to Ethernet interface 33.

```
switch(config)#interface ethernet 33
switch(config-if-Et33)#monitor link-flap profiles LF03 LF04
switch(config-if-Et33)#show active
interface Ethernet33
   monitor link-flap profiles LF04 LF03
switch(config-if-Et33)#
```

- This command disables link-flap monitoring on Ethernet interface 34.

```
switch(config)#interface ethernet 34
switch(config-if-Et34)#no monitor link-flap
switch(config-if-Et34)#show active
interface Ethernet34
   no monitor link-flap
switch(config-if-Et34)#
```

- This command assigns the default-profile set to Ethernet interface 35.

```
switch(config)#interface ethernet 35
switch(config-if-Et35)#monitor link-flap
switch(config-if-Et35)#show active
interface Ethernet35
   monitor link-flap
switch(config-if-Et35)#
```

- This command configures Ethernet interface 36 to use the global link flap monitoring commands

```
switch(config)#interface ethernet 36
switch(config-if-Et36)#default monitor link-flap
switch(config-if-Et36)#show active
interface Ethernet36
switch(config-if-Et36)#
```

16.5.4 Fabric Link Monitoring

Fabric link monitoring enables EOS to monitor low error rate errors on all fabric links for long durations, and automatically isolates fabric links on consistent error detection over an extended time interval. Isolated fabric links are restored when the error rate drops below a configured threshold.

The error rate over each configurable polling interval is derived by comparing the number of cells with CRC errors against the total number of received cells. Links are automatically isolated when the error rate is above the configured threshold for the configured consecutive number of polling intervals.

On an isolated fabric link, control cells (but not data cells) are sent. Once the error rate drops below a set threshold for the configured consecutive number of polling intervals, EOS revives the fabric link to continue sending data traffic.

16.5.4.1 Configuring Fabric Link Monitoring

Configuration mode commands globally enable and disable fabric link monitoring and syslog messages for the settings described below.

The **no platform sand monitor** command disables fabric link monitoring.

Generate Serdes Error Syslog

The **platform sand monitor serdes error log** command generates syslog fabric link monitoring for serdes error logging.

Example

- This command enables the serdes error log for fabric link monitoring.

```
switch(config)#platform sand monitor serdes error log
switch(config)#
```

The following syslog messages are not enabled by default. Fabric link monitoring syslog is enabled by configuring the **platform sand monitor serdes error log** command.

Examples

- The following syslog message is generated when a fabric link for serdes is automatically withdrawn:

```
%SAND-4-SERDES_WITHDRAWN_FROM_FABRIC: Serdes withdrawn from the switch fabric.
```

- Here is another instance where a syslog message is generated when a fabric link is automatically withdrawn:

```
%SAND-4-SERDES_WITHDRAWN_FROM_FABRIC: Serdes Arad10/5-FabricSerdes-11 withdrawn
from the switch fabric.
```

- The following syslog message is generated when a fabric link is restored:

```
%SAND-4-SERDES_RESTORED_TO_FABRIC: Serdes restored to the switch fabric.
```

- Here is another instance where a syslog message is generated when a fabric link is restored:

```
%SAND-4-SERDES_RESTORED_TO_FABRIC: Serdes Arad10/5-FabricSerdes-11 restored to
the switch fabric.
```

Generate Serdes Error Threshold

The **platform sand monitor serdes error threshold** command generates a fabric link monitoring serdes error threshold.

Example

- This command monitors serdes error thresholds over the specified number of received cells, resulting in the isolation of a fabric link between 200 and 30,000 received cells.

```
switch(config)#platform sand monitor serdes error threshold 200 30000
switch(config)#
```

Enable Serdes Poll Period

The **platform sand monitor serdes poll period** command sets the serdes poll period.

Example

- This command changes the serdes polling period for fabric link monitoring to 6 seconds.

```
switch(config)#platform sand monitor serdes poll period 6
switch(config)#
```

Monitor Serdes Poll Threshold Isolation

The **platform sand monitor serdes poll threshold isolation** command sets and enables fabric link monitoring for serdes poll threshold isolation.

Example

- This command changes the number of consecutive polls in which the threshold needs to be detected to isolate a link. In this case the number is 5 consecutive polls.

```
switch(config)#platform sand monitor serdes poll threshold isolation 5
switch(config)#
```

Monitor Serdes Poll Threshold Recovery

The **platform sand monitor serdes poll threshold recovery** command sets and enables fabric link monitoring for serdes poll threshold recovery.

Example

- This command changes the number of consecutive serdes polls used for threshold recovery to 6 seconds.

```
switch(config)#platform sand monitor serdes poll threshold recovery 6
switch(config)#
```

Show Fabric Monitoring Health

The **show fabric monitoring health** command displays the fabric monitoring connected state status with isolated links.

Example

- When fabric links are isolated, their connected state status is shown with isolated links.

```
switch(config)#show platform sand health
Fabric serdes isolated by fabric monitoring: (36 total)

Arad5/0 serdes [0-1, 10-19, 2, 20-29, 3, 30-35, 4-9]

Top fabric serdes list by number of times isolated by monitoring:
Arad5/0 serdes 0: 1 (last occurred: 0:01:04 ago)
Arad5/0 serdes 1: 1 (last occurred: 0:01:04 ago)
Arad5/0 serdes 10: 1 (last occurred: 0:01:04 ago)
Arad5/0 serdes 11: 1 (last occurred: 0:01:04 ago)
Arad5/0 serdes 12: 1 (last occurred: 0:01:04 ago)
Arad5/0 serdes 13: 1 (last occurred: 0:01:04 ago)
Arad5/0 serdes 14: 1 (last occurred: 0:01:04 ago)
Arad5/0 serdes 15: 1 (last occurred: 0:01:04 ago)
Arad5/0 serdes 16: 1 (last occurred: 0:01:04 ago)
Arad5/0 serdes 17: 1 (last occurred: 0:01:04 ago)

switch(config)#
```

16.5.5 Rapid Automated Indication of Link-Loss

Rapid Automated Indication of Link-Loss (RAIL) is a software feature that reduces the wait time of applications on hosts that are blocked due to a failed link. When a link goes down because of link-flapping or the unavailability of a directly connected server, the switch drops all traffic to servers whose next-hop destination was learned on the port connected to the link. Applications that drive the traffic (clients on source hosts) are blocked because of the dropped edge-switch traffic. Connection timeout varies by application and is usually measured in seconds or minutes.

RAIL is functional on a switch if it is routing-enabled and available for servers that set the switch as the default router.

16.5.5.1 RAIL Method

When a link monitored by RAIL goes down, the switch performs these steps for servers that the switch proxies:

Step 1 IP addresses of servers on the failed link are extracted from ARP cache. The interface that accesses the server is determined by searching for the MAC address in the hardware MAC address tables.

Step 2 Upon link shutdown, a dynamic MAC entry is added in the MAC address table for each server that was learned on the failed interface. Each new entry lists its interface as ***CPU***.

Figure 16-2 depicts three switch-server scenarios: link is up, link is down with RAIL disabled, and link is down with RAIL enabled. A failed link with RAIL enabled results in these behaviors:

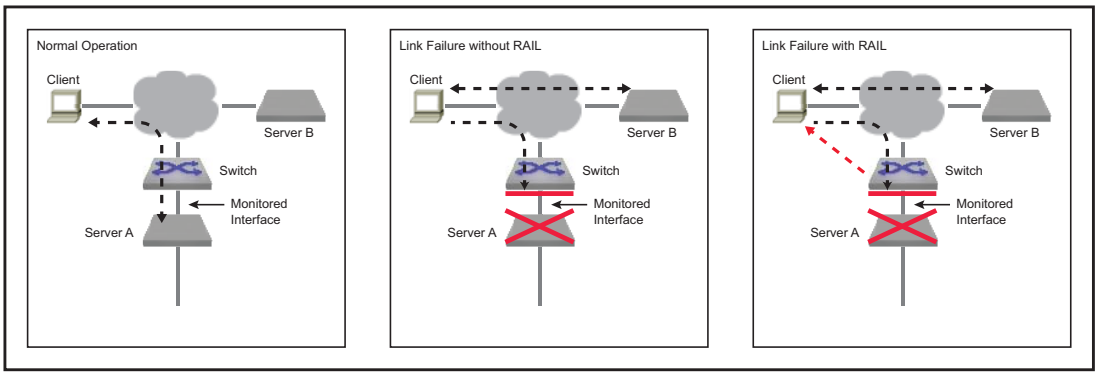
Step 1 All ingress packets whose destination MAC address matches an address added to the MAC address table are sent to the CPU.

Step 2 For packets scheduled to be forwarded to the source address, the switch sends one of the following, based on the type of received segment:

- TCP: TCP RST segment to the source IP address and port.
- UDP: ICMP unreachable segment to the source IP address and port.

Step 3 The client closes the socket associated with the transmitted segment and notifies the application. The application reacts immediately instead of maintaining the block until connection timeout expiry.

Figure 16-2: RAIL Scenarios



16.5.5.2 RAIL Implementation

RAIL defines a state machine that manages the RAIL activity level relative to a specified server. The state machine consists of four states:

- **Up**: Transitions to this state from ***Inactive*** when ARP and MAC entries are added for the server.
- **Proxying**: Transitions to this state from ***Up*** when Link Down is detected and RAIL proxying is enabled. The switch is a proxy for messages to the server.
- **Down**: Transitions to this state from ***Up*** when Link Down is detected and RAIL proxying is not enabled. Messages from the client remain unanswered and the application recovers only after timeout expiry.
- **Inactive**: Transitions to this state upon any of the following conditions:
  - Server's MAC address or ARP entry is deleted (from any state).
  - Proxy timeout expiry (from ***Proxying*** state)
  - Link down timeout expiry (from ***Down*** state).

16.5.5.3 RAIL Configuration

Server-failure configuration mode commands globally enable RAIL and configure RAIL parameters. RAIL is functional on individual interfaces only when it is globally enabled and enabled on the interface. RAIL monitors an interface for link errors when RAIL is globally enabled and enabled on the interface.

Entering Server-failure Configuration Mode

The **monitor server-failure** command places the switch in server-failure configuration mode. The **exit** command returns the switch to global configuration mode. Server-failure mode is not a group change mode; ***running-config*** is changed when commands are entered and not affected by exiting the mode.

The **no monitor server-failure** deletes all server-failure mode commands from ***running-config***.

Examples

- These commands place the switch in server-failure configuration mode.

```
switch(config)#monitor server-failure
switch(config-server-failure)#
```

- This command deletes all server-failure configuration mode commands from ***running-config***.

```
switch(config)#no monitor server-failure
switch(config)#
```

Enabling RAIL on the Switch

RAIL is disabled by default and is enabled by **no shutdown (server-failure configuration mode)**. The **shutdown** command disables RAIL without removing RAIL commands from ***running-config.***

Examples

- These commands enable RAIL globally.

```
switch(config)#monitor server
switch(config-server-failure)#no shutdown
switch(config-server-failure)#show active
monitor server-failure
   no shutdown
switch(config-server-failure)#
```

- This command disables RAIL globally.

```
switch(config-server-failure)#shutdown
switch(config-server-failure)#
```

Enabling Proxy Mode

The **proxy (server-failure configuration mode)** command sets the RAIL proxy setting to ***enabled*** and specifies the interval that RAIL responds to messages sent to servers on failed links. The proxy timeout is measured individually for each server whose link has failed. The switch enters RAIL proxy state only when the proxy setting is enabled.

When RAIL is enabled but the proxy setting is disabled, the switch maintains a list of unavailable servers without responding to messages sent to the servers. The RAIL proxy setting is ***disabled*** by default. When RAIL proxy is enabled, the default period is three minutes.

The **no proxy** and **default proxy** commands return the RAIL proxy setting to ***disabled***. The **no proxy lifetime** and **default proxy lifetime** commands set the proxy timeout to its default of three minutes if the RAIL proxy setting is ***enabled***. The lifetime commands have no effect if RAIL proxy is ***disabled***.

Examples

- These commands enable the RAIL proxy and sets the proxy timeout period of 10 minutes.

```
switch(config)#monitor server
switch(config-server-failure)#proxy lifetime 10
switch(config-server-failure)#show active
monitor server-failure
   proxy lifetime 10
switch(config-server-failure)#
```

- This command sets the proxy timeout period to its default value of 3 minutes.

```
switch(config-server-failure)#no proxy lifetime
switch(config-server-failure)#show active
monitor server-failure
   proxy
switch(config-server-failure)#
```

- This command disables the RAIL proxy.

```
switch(config-server-failure)#no proxy
switch(config-server-failure)#show active
switch(config-server-failure)#
```

Selecting Networks to Monitor

The **network (server-failure configuration mode)** command specifies the IPv4 network space that Rapid Automated Indication of Link-Loss (RAIL) monitors for failed links to connected servers. ***Running-config*** can contain multiple **network** statements, allowing RAIL to monitor multiple disjoint network spaces.

When a server on the specified network is blocked because of a failed Ethernet or port channel link, the switch becomes a proxy for the unavailable server and responds with ***TCP RST*** or ***ICMP Unreachable*** segments to devices sending packets to the unavailable server.

Example

- These commands specify two IPv4 network spaces that RAIL monitors for server failures.

```
switch(config)#monitor server
switch(config-server-failure)#network 10.1.1.0/24
switch(config-server-failure)#network 10.2.1.96/28
switch(config-server-failure)#show active
monitor server-failure
   network 10.2.1.96/28
   network 10.1.1.0/24
switch(config-server-failure)#
```

Enabling RAIL on an Interface

RAIL monitors an interface for link errors only when RAIL is globally enabled and enabled for the interface. The **monitor server-failure link** command enables RAIL on the configuration mode interface. Configuration settings are effective for all Ethernet and port channel interfaces that enable RAIL.

Example

- These commands enable RAIL on port channel interface 100.

```
switch(config)#interface port-channel 100
switch(config-if-Po100)#monitor server-failure link
switch(config-if-Po100)#show active
interface Port-Channel100
   monitor server-failure link
switch(config-if-Po100)#
```

16.5.5.4 Displaying RAIL Status

The switch provides commands to display RAIL configuration and status information:

Displaying RAIL Configuration settings

The **show monitor server-failure** command displays Rapid Automated Indication of Link-Loss (RAIL) configuration settings and the number of servers on each monitored network.

Example

- This command displays RAIL configuration status and lists the number of servers that are on each monitored network.

```
switch>show monitor server-failure
Server-failure monitor is enabled
Proxy service: disabled
Networks being monitored: 3
   10.2.1.96/28      : 0 servers
   10.1.1.0/24       : 0 servers
   10.3.0.0/16       : 3 servers
switch>
```

Displaying RAIL History for All Connected Servers

The **show monitor server-failure history** command displays the time of all link failures detected by Rapid Automated Indication of Link-Loss (RAIL) and includes the interface name for each failure.

Example

- This command displays the link failure history from the time RAIL is instantiated on the switch.

```
switch>show monitor server-failure history
Total server failures: 4

Server IP   Server MAC             Interface          Last Failed
----------- -----------------      -----------        -------------------
10.1.67.92  01:22:ab:cd:ee:ff      Ethernet17         2013-02-02 11:26:22
44.11.11.7  ad:3e:5f:dd:64:cf      Ethernet23         2013-02-10 00:07:56
10.1.1.1    01:22:df:42:78:cd      Port-Channel6      2013-02-09 19:36:09
10.1.8.13   01:33:df:ee:39:91      Port-Channel5      2013-02-10 00:03:39

switch>
```

Displaying Server Configuration and Status

The **show monitor server-failure servers** command displays status and configuration data about each server that RAIL monitors. The display format depends on the parameter specified by the command:

Example

- This command displays RAIL information for the server at IP address 10.11.11.7

```
switch>show monitor server-failure servers 10.11.11.7
Server information:
Server Ip Address        : 10.11.11.7
MAC Address              : ad:3e:5f:dd:64:cf
Current state            : down
Interface                : Ethernet23
Last Discovered          : 2013-01-06 06:47:39
Last Failed              : 2013-02-10 00:07:56
Last Proxied             : 2013-02-10 00:08:33
Last Inactive            : 2013-02-09 23:52:21
Number of times failed   : 3
Number of times proxied  : 1
Number of times inactive : 18

switch>
```

- This command displays RAIL information for the all servers on configured interfaces.

```
switch>show monitor server-failure servers all
Total servers monitored: 5

Server IP   Server MAC         Interface       State Last Failed
----------  -----------------  --------------  ---------  -----------
10.1.67.92  01:22:ab:cd:ee:ff  Ethernet17      inactive   7 days, 12:47:48 ago
44.11.11.7  ad:3e:5f:dd:64:cf  Ethernet23      down       0:06:14 ago
10.1.1.1    01:22:df:42:78:cd  Port-Channel6   up         4:38:01 ago
10.1.8.13   01:33:df:ee:39:91  Port-Channel5   proxying   0:10:31 ago
132.23.23.1 00:11:aa:bb:32:ad  Ethernet1       up         never

switch>
```

16.6 Data Transfer Command Descriptions

Control Plane and Data Plane Commands

- control-plane
- ip access-group (Control Plane mode)
- switch forwarding-mode
- show switch forwarding-mode

Errdisable Commands

- errdisable detect cause link-flap
- errdisable flap-setting cause link-flap
- errdisable recovery cause
- errdisable recovery interval

Fabric Link Monitoring Commands

- platform sand monitor serdes error log
- platform sand monitor serdes error threshold
- platform sand monitor serdes poll period
- platform sand monitor serdes poll threshold isolation
- platform sand monitor serdes poll threshold recovery
- show fabric monitoring health

RAIL Commands

- clear server-failure servers inactive
- monitor server-failure
- monitor server-failure link
- network (server-failure configuration mode)
- proxy (server-failure configuration mode)
- show monitor server-failure
- show monitor server-failure history
- show monitor server-failure servers
- shutdown (server-failure configuration mode)

Link Flap Monitor Commands

- default-profiles
- monitor link-flap policy
- monitor link-flap profiles
- profile max-flaps (Link Flap Configuration)

MAC Address Table Commands

- clear mac address-table dynamic
- mac address-table aging-time
- mac address-table static
- switchport mac address learning
- show mac address-table
- show mac address-table aging time
- show mac address-table count
- show mac address-table mlag-peer
- show mac address-table multicast
- show mac address-table multicast brief

Port Configuration Commands

- clear counters
- description
- interface loopback
- load interval
- switchport
- mtu
- show interfaces
- show interfaces description
- switchport default mode access
- switchport default mode routed

Port Mirroring Commands

- monitor session destination
- monitor session destination cpu
- monitor session ip access-group
- monitor session source
- monitor session source ip access-group
- monitor session truncate
- no monitor session
- show monitor session

Port Security Commands

- switchport port-security
- switchport port-security maximum
- switchport port-security violation protect
- show port-security
- show port-security address
- show port-security interface

Storm Control Commands

- storm-control
- show storm-control

Tracking Commands

- link state group
- link state track
- links minimum
- track
- show link state group
- show track

clear counters

The **clear counters** command resets the counters to zero for the specified interfaces. The command provides the following options:

- No parameter: When no option is selected, the counters are reset on the switch.
- Session parameter: The command resets the counters in software for the current CLI session, establishing a baseline upon which subsequent **show interfaces** or **show interfaces counters** commands are relative. Counters are not affected for other CLI sessions.

Command Mode

Privileged EXEC

Command Syntax

clear counters [*INTERFACE*] [*SCOPE*]

Parameters

- ***INTERFACE*** Interface type and number. Options include:
  - <no parameter> Display information for all interfaces.
  - **ethernet** *e_range* Ethernet interface range specified by *e_range*.
  - **loopback** *l_range* Loopback interface specified by *l_range*.
  - **management** *m_range* Management interface range specified by *m_range*.
  - **port-channel** *p_range* Port-Channel Interface range specified by *p_range*.
  - **vlan** *v_range* VLAN interface range specified by *v_range*.
  - **vxlan** *vx_range* VXLAN interface range specified by *vx_range*.

    Valid *e_range*, *l_range*, *m_range*, *p_range*, *v_range,* and *vx_range* formats include number, number range, or comma-delimited list of numbers and ranges.
- ***SCOPE*** Duration of the reset results. Options include:
  - <no parameter> counters are cleared on the switch.
  - **session** counters are reset only for the current session.

Examples

- These commands display interface counters, clear the counters, then display the counters again.

```
switch#show interfaces ethernet 1
Ethernet1 is up, line protocol is up (connected)
  Hardware is Ethernet, address is 001c.7302.2fff (bia 001c.7302.2fff)
  MTU 9212 bytes, BW 10000000 Kbit
  Full-duplex, 10Gb/s, auto negotiation: off
  Last clearing of "show interface" counters never
  5 minutes input rate 301 bps (0.0% with framing), 0 packets/sec
  5 minutes output rate 0 bps (0.0% with framing), 0 packets/sec
     2285370854005 packets input, 225028582832583 bytes
     Received 29769609741 broadcasts, 3073437605 multicast
     113 runts, 1 giants
     118 input errors, 117 CRC, 0 alignment, 18 symbol
     27511409 PAUSE input
     335031607678 packets output, 27845413138330 bytes
     Sent 14282316688 broadcasts, 54045824072 multicast
     108 output errors, 0 collisions
     0 late collision, 0 deferred
     0 PAUSE output
switch#show interfaces ethernet 1-5 counters
Port                 InOctets     InUcastPkts     InMcastPkts     InBcastPkts
Et1           225028582833321   2252527806659      3073437611     29769609741
Et2            20706544058626    121703943738      7619026884     43349412335
Et3            17473231954010     84335312119     18987530444     25136247381
Et4            21909861242537    119410161405      3792251718     48470646199
Et5                         0              0               0               0

Port                OutOctets    OutUcastPkts    OutMcastPkts    OutBcastPkts
Et1            27845413138330    266703466918     54045824072     14282316688
Et2            39581155181762    384838173282     34879250675     15500233246
Et3            25684397682539    256695349801     25193361878     16244203611
Et4           428040746505736   2285287022532     44408620604     19503612572
Et5                         0              0               0               0
switch#clear counters session
switch#show interfaces ethernet 1
Ethernet1 is up, line protocol is up (connected)
  Hardware is Ethernet, address is 001c.7302.2fff (bia 001c.7302.2fff)
  MTU 9212 bytes, BW 10000000 Kbit
  Full-duplex, 10Gb/s, auto negotiation: off
  Last clearing of "show interface" counters 0:00:10 ago
  5 minutes input rate 322 bps (0.0% with framing), 0 packets/sec
  5 minutes output rate 0 bps (0.0% with framing), 0 packets/sec
     6 packets input, 835 bytes
     Received 0 broadcasts, 6 multicast
     0 runts, 0 giants
     0 input errors, 0 CRC, 0 alignment, 0 symbol
     0 PAUSE input
     0 packets output, 0 bytes
     Sent 0 broadcasts, 0 multicast
     0 output errors, 0 collisions
     0 late collision, 0 deferred
     0 PAUSE output
switch#show interfaces ethernet 1-5 counters
Port                InOctets     InUcastPkts     InMcastPkts     InBcastPkts
Et1                     1204               0               9               0
Et2                     1204               0               9               0
Et3                     1204               0               9               0
```

```
Et4                    1204               0               9               0
Et5                       0               0               0               0

Port               OutOctets    OutUcastPkts    OutMcastPkts    OutBcastPkts
Et1                       0               0               0               0
Et2                       0               0               0               0
Et3                       0               0               0               0
Et4                       0               0               0               0
Et5                       0               0               0               0
switch#
```

clear mac address-table dynamic

The **clear mac address-table dynamic** command removes specified dynamic entries from the MAC address table. Entries are identified by their VLAN and layer 2 (Ethernet or port channel) interface.

- To remove a specific entry, include its VLAN and interface in the command.
- To remove all dynamic entries for a VLAN, do not specify an interface.
- To remove all dynamic entries for an interface, do not specify a VLAN.
- To remove all dynamic entries, do not specify a VLAN or an interface.

Command Mode

Privileged EXEC

Command Syntax

clear mac address-table dynamic [*VLANS*] [*INTERFACE*]

Parameters

- ***VLANS*** Table entries are cleared for specified VLANs. Options include:
  - <no parameter> all VLANs.
  - **vlan** *v_num* VLAN specified by *v_num*.
- ***INTERFACE*** Table entries are cleared for specified interfaces. Options include:
  - <no parameter> all Ethernet and port channel interfaces.
  - **interface ethernet** *e_range* Ethernet interfaces specified by *e_range*.
  - **interface port-channel** *p_range* port channel interfaces specified by *p_range*.
  - **vxlan** *vx_range* VXLAN interfaces specified by *vx_range*.

  Valid *range* formats include number, range, or comma-delimited list of numbers and ranges.

Example

- This command clears all dynamic mac address table entries for port channel 5 on VLAN 34.

```
switch#clear mac address-table dynamic vlan 34 interface port-channel 5
switch#
```

clear server-failure servers inactive

The **clear server-failure servers inactive** command removes all inactive server entries from the server failed history list. The switch maintains this list, even after a server's ARP entry is removed, to maintain a list of servers that are connected to the switch and log the most recent time of the failure of the link that connects the switch to the server.

Command Mode

Privileged EXEC

Command Syntax

```
clear server-failure servers inactive
```

Related Commands

- **show monitor server-failure history**

Example

- This command clears the inactive servers from the server failed history list.

```
switch#clear server-failure servers inactive
switch#
```

control-plane

Note On hardware purchased after July 3, 2017, control plane ACLs are not available.

The **control-plane** command places the switch in control-plane configuration mode. Control-plane mode is used for assigning an ACL (access control list) to the control plane.

Control-plane configuration mode is not a group change mode; ***running-config*** is changed immediately after commands are executed. Exiting control-plane configuration mode does not affect the configuration.

The **exit** command returns the switch to global configuration mode.

Command Mode

Global Configuration

Command Syntax

```
control-plane
```

Commands Available in control-plane Configuration Mode

- **ip access-group (Control Plane mode)**

Examples

- This command places the switch in control plane mode.

```
switch(config)#control-plane
switch(config-cp)
```

- This command assigns the ***control-plane-2*** ACL to the control plane.

```
switch(config-cp)#ip access-group control-plane-2
switch(config-cp)
```

- This command exits control plane mode.

```
switch(config-cp)#exit
switch(config)
```

default-profiles

The **default-profiles** command specifies the set of link-flap profiles that define error-disable criteria for interfaces where link flap monitoring is enabled without a link flap profile assignment. Entering a **default-profile** command replaces the current default-profile statement in ***running-config***.

The default-profile set may contain zero, one, or multiple profiles. When the default-profile set is empty, **errdisable flap-setting cause link-flap** specifies default error-disable criteria. When the default-profile set contains multiple profiles, error-disable criteria is satisfied when conditions match any profile. Multiple profiles are assigned to the default-profile set through a single **default-profiles** command.

The **no default-profiles** and **default default-profiles** commands restore the empty default-profile set by deleting the **default-profiles** command from ***running-config***.

Command Mode

Link-flap Configuration

Command Syntax

```
default-profiles [LF_PROFILES]
no default-profiles
default default-profiles
```

Parameters

- ***LF_PROFILES*** Name of link-flap profiles assigned to default profile set. Parameter may contain zero, one, or multiple link-flap profile names:
  - <no parameter> default-profile set is empty.
  - *profile* name of single link-flap profile.
  - *profile_1 profile_2 ... profile_N* list of link-flap profile names.

Related Commands

- **monitor link-flap policy** places the switch in link-flap-profiles configuration mode.
- **profile max-flaps (Link Flap Configuration)** configures link flap profiles.

Guidelines

The **errdisable flap-setting cause link-flap** statement is also configurable through the **profile max-flaps (Link Flap Configuration)** command.

Example

- This command assigns configures LF01 and LF02 as the default-profile set.

```
switch(config)#monitor link-flap policy
switch(config-link-flap)#default-profiles LF01 LF02
switch(config-link-flap)#show active
monitor link-flap policy
   profile LF01 max-flaps 15 time 60 violations 1 intervals 1
   profile LF02 max-flaps 10 time 30 violations 5 intervals 10
   profile LF03 max-flaps 25 time 100 violations 2 intervals 12
   profile LF04 max-flaps 5 time 15 violations 1 intervals 3
   default-profiles LF01 LF02
switch(config-link-flap)#
```

description

The **description** command adds comment text for the configuration mode interface. The text provides information about the interface and has no effect on interface functions. The **show interfaces description** command displays interface description text.

The **no description** command removes the description text for the configuration mode interface from ***running-config***.

Command Mode

Interface-Ethernet Configuration
Interface-Loopback Configuration
Interface-Management Configuration
Interface-Port-channel Configuration
Interface-VLAN Configuration
Interface-VXLAN Configuration

Command Syntax

```
description label_text
no description
default description
```

Parameters

- *label_text* character string assigned to description attribute.

Examples

- These commands add description text to Ethernet interface 23, then displays the text through a **show interfaces description** command.

```
switch(config)#interface ethernet 23
switch(config-if-Et23)#description external line
switch(config-if-Et23)#show interfaces ethernet 23 description
Interface                      Status         Protocol Description
Et23                           up             up       external line
```

errdisable detect cause link-flap

The **errdisable detect cause link-flap** command enables the error-disabling of Ethernet interfaces when the switch detects a link flap error on the interface. The **errdisable flap-setting cause link-flap** command defines a link flap error in terms of the frequency of connection state changes.

The switch places an interface in ***error-disabled*** state when it detects an error on the interface. ***Error-disabled*** is an operational state that is similar to ***link-down*** state. To re-enable an error-disabled interface, enter **shutdown** and **no shutdown** command in the configuration mode for the interface.

By default, link flap detection is enabled. The **no errdisable detect cause link-flap** command disables the triggering of error-disable actions. The **errdisable detect cause link-flap** and **default errdisable detect cause link-flap** commands enable the triggering of error-disable actions by removing the **no errdisable detect cause link-flap** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
errdisable detect cause link-flap
no errdisable detect cause link-flap
default errdisable detect cause link-flap
```

Examples

- This command disables error detection on the switch.

```
switch(config)#no errdisable detect cause link-flap
switch(config)#
```

- These commands sets the link flap error criteria of 15 connection state changes over a 30 second period, then enables error detection on the switch.

```
switch(config)#errdisable flap-setting cause link-flap max-flaps 15 time 30
switch(config)#errdisable detect cause link-flap
switch(config)#
```

errdisable flap-setting cause link-flap

The **errdisable flap-setting cause link-flap** command configures the link-flap frequency that defines an link-flap error on an Ethernet interface. The **errdisable detect cause link-flap** command uses this criteria to trigger an error-disable action.

The link-flap frequency is defined by the quantity of link flaps (connection state changes) over a specified period. The default settings are five link flaps and ten seconds.

The **no errdisable flap-setting cause link-flap** and **default errdisable flap-setting cause link-flap** commands restore the default link flap cause settings by removing the **errdisable flap-setting cause link-flap** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
errdisable flap-setting cause link-flap max-flaps quantity time period
no errdisable flap-setting cause link-flap
default errdisable flap-setting cause link-flap
```

Parameters

- *quantity* Number of link flaps. Value ranges from 1 to 100. Default value is 5.
- *period* Interval over which link flaps accumulate to trigger an error condition (seconds). Value ranges from 1 to 1800. Default value is 10.

Examples

- This command sets the link flap error criteria of 15 connection state changes over 30 second periods.

```
switch(config)#errdisable flap-setting cause link-flap max-flaps 15 time 30
switch(config)#
```

errdisable recovery cause

The **errdisable recovery cause** command enables the automated recovery of error-disabled Ethernet interfaces. An interface that is disabled as a result of a specified condition attempts normal operation after a specified interval. When the disabling condition persists, recovered interfaces eventually return to the error-disabled state.

When automated recovery is not enabled, interfaces are recovered manually by entering **shutdown** and **no shutdown** from the interface's configuration mode.

Running-config can simultaneously store **errdisable recovery cause** statements for each error-disable condition. By default, error-disable recovery is disabled for all conditions.

The **no errdisable recovery cause** and **default errdisable recovery cause** commands disable automated recovery for interfaces disabled by the specified condition by removing the corresponding **errdisable recovery cause** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
errdisable recovery cause CONDITION
no errdisable recovery cause CONDITION
default errdisable recovery cause CONDITION
```

Parameters

- ***CONDITION*** Disabling condition for which command automates recovery. Options include:
  - **arp-inspection**
  - **bpduguard**
  - **link-flap**
  - **no-internal-vlan**
  - **portchannelguard**
  - **portsec**
  - **tapagg**
  - **uplink-failure-detection**
  - **xcvr_unsupported**

Related Commands

- **errdisable recovery interval** configures the period that an ethernet interface remains disabled before automated recovery begins.

Examples

- This command enables error-disable recovery for interfaces that are disabled by link-flap and bpduguard conditions and sets the errdisable recovery period at 10 minutes.

```
switch(config)#errdisable recovery cause bpduguard
switch(config)#errdisable recovery cause link-flap
switch(config)#errdisable recovery interval 600
switch(config)#show running-config
! Command: show running-config
             <-------OUTPUT OMITTED FROM EXAMPLE-------->
errdisable recovery cause bpduguard
errdisable recovery cause link-flap
errdisable recovery interval 600
!
             <-------OUTPUT OMITTED FROM EXAMPLE-------->
switch(config)#
```

errdisable recovery interval

The **errdisable recovery interval** command specifies the period that an error-disabled Ethernet interface remains disabled before automated errdisable recovery begins. This command affects only interfaces whose automated recovery is enabled for the disabling condition (**errdisable recovery cause**). When automated recovery is not enabled, interfaces are recovered manually by entering **shutdown** and **no shutdown** from the interface's configuration mode.

The **no errdisable recovery interval** and **default errdisable recovery interval** commands restore the default error recovery period of 300 seconds by removing the **errdisable recovery interval** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
errdisable recovery interval period
no errdisable recovery interval
default errdisable recovery interval
```

Parameters

- *period* Error disable recovery period (seconds). Value ranges from 30 to 86400. Default value is 300

Related Commands

- **errdisable recovery cause** enables the automated recovery of error-disabled Ethernet interfaces.

Examples

- This command enables error-disable recovery for interfaces that are disabled by link-flap conditions and sets the errdisable recovery period at 10 minutes.

```
switch(config)#errdisable recovery cause link-flap
switch(config)#errdisable recovery interval 600
switch(config)#show running-config
! Command: show running-config
             <-------OUTPUT OMITTED FROM EXAMPLE-------->
!
errdisable recovery cause link-flap
errdisable recovery interval 600
!
             <-------OUTPUT OMITTED FROM EXAMPLE-------->
!
i
switch(config)#
```

interface loopback

The **interface loopback** command places the switch in loopback-interface configuration mode for the specified interfaces. The command creates loopback interfaces for previously unconfigured interfaces.

The command can specify a single interface or multiple interfaces:

- Single interface: Command creates an interface if it specifies one that was not previously created.
- Multiple interfaces: Command is valid only if all specified interfaces were previously created.

The **no interface loopback** command removes the specified interfaces from ***running-config***, including all interface configuration statements. The **default interface loopback** command removes all configuration statements for the specified loopback interface without deleting the loopback interface from ***running-config***.

The following commands are available in loopback configuration mode:

- description
- exit
- ip address
- ip proxy-arp
- ipv6 address
- ipv6 enable
- load interval
- logging event
- mtu
- shutdown (Interfaces)
- snmp trap

Command Mode

Global Configuration

Command Syntax

```
interface loopback l_range
no interface loopback l_range
default interface loopback l_range
```

Parameters

- *l_range* Loopback interfaces (number, range, or comma-delimited list of numbers and ranges).

  Loopback number ranges from 0 to 1000.

Examples

- This command enters interface configuration mode for loopback interfaces 1 through 5.

```
switch(config)#interface loopback 1-5
switch(config-if-Lo1-5)#
```

- This command creates interface 23 and enters interface configuration mode:

```
switch(config)#interface loopback 23
switch(config-if-Lo23)#
```

- This command removes loopback interfaces 5 through 7 from ***running-config***.

```
switch(config)#no interface loopback 5-7
switch(config)#
```

ip access-group (Control Plane mode)

Note On hardware purchased after July 3, 2017, control plane ACLs are not available.

The **ip access-group** command applies an IPv4 or standard IPv4 access control list (ACL) to the control plane.

The **no ip access-group** and **default ip access-group** commands remove the corresponding **ip access-group** command from ***running-config***.

Command Mode

Control-plane Configuration

Command Syntax

```
ip access-group list_name [VRF_INSTANCE] DIRECTION
no ip access-group [list_name] [VRF_INSTANCE] DIRECTION
default ip access-group [list_name] [VRF_INSTANCE] DIRECTION
```

Parameters

- *list_name* name of ACL assigned to interface.
- ***VRF_INSTANCE*** specifies the VRF instance being modified.
  - <no parameter> changes are made to the default VRF.
  - **vrf** *vrf_name* changes are made to the specified user-defined VRF.
- ***DIRECTION*** transmission direction of packets, relative to interface. Valid options include:
  - **in** inbound packets.

Example

- These commands apply the IPv4 ACL named ***test2*** to the control plane.

```
switch(config)#control-plane
switch(config-cp)#ip access-group test2 in
switch(config-cp)#
```

link state group

The **link state group** command adds the configuration mode interface to a link-state group and specifies whether it is upstream or downstream.

The **no link state group** and **default link state group** commands remove the specified link-state group assignment for the configuration mode interface.

Command Mode

Interface-Ethernet Configuration
Interface-Loopback Configuration
Interface-Management Configuration
Interface-Port-channel Configuration
Interface-VLAN Configuration
Interface-VXLAN Configuration

Command Syntax

```
link state group group_name DIRECTION
no link state group [group_name]
default link state group [group_name]
```

Parameters

- *group_name* link state tracking group name.
- ***DIRECTION*** position of the interface in the link-state group. Valid options include:
  - **upstream**
  - **downstream**

Example

- These commands create link-state group “xyz” and add VLAN interface 100 to the group as an upstream interface.

```
switch(config)#link state track xyz
switch(config-link-state-xyz)#show active
 link state track xyz
switch(config-link-state-xyz)#exit
switch(config)#interface vlan 100
switch(config-if-Vl100)#link state group xyz upstream
switch(config-if-Vl100)#show active
 interface Vlan100
   link state group xyz upstream
switch(config-if-Vl100)#
```

link state track

The **link state track** command creates and enables a link-state group and places the switch in link-state-group configuration mode. A link-state group consists of “upstream” interfaces (connections to servers) and “downstream” interfaces (connections to switches and clients). In the event of a failure of all upstream interfaces in the link-state group, the downstream interfaces are shut down.

The **no link state track** and **default link state track** commands delete the **link state track** from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
link state track group_name
no link state track group_name
default link state track group_name
```

Parameters

- *group_name* link-state group name.

Commands available in link-state Configuration Mode

- **links minimum** configures the minimum number of links that the link-state group requires.

Example

- This command creates and enables link-state group 1.

```
switch(config)#link state track 1
switch(config-link-state-1)#
```

links minimum

The **links minimum** command specifies the minimum number of links the configuration mode link-state group requires.

The **no links minimum** and **default links minimum** commands restore the default minimum value of 1 by deleting the corresponding **links minimum** statement from ***running-config***.

Command Mode

Link-State Configuration

Command Syntax

```
links minimum quantity
no links minimum
default links minimum
```

Parameters

- *quantity* Minimum number of links. Value ranges from 1 to 100000. Default value is 1.

Related Commands

- **link state track** creates and enables a link-state group and places the switch in link-state configuration mode.
- **link state group** adds the configuration mode interface to the specified link-state group.

Examples

- These commands configure link-state tracking group ***link-a*** to have at least 60 links.

```
switch(config)#link state track link-a
switch(config-link-state-1ink-a)links minimum 60
switch(config-link-state-link-a)
```

load interval

The **load-interval** command changes the load interval for the configuration mode interface. Load interval is the time period over which data is used to compute interface rate counters. Interface rates are exponentially weighted moving averages; recent data samples have greater influence than older samples. Statistics calculated with shorter load intervals are usually more sensitive to short traffic bursts.

The **no load-interval** and **default load-interval** commands restore the default value of 300 seconds by removing the corresponding **load-interval** statement from ***running-config***.

Command Mode

Interface-Ethernet Configuration
Interface-Loopback Configuration
Interface-Management Configuration
Interface-Port-channel Configuration
Interface-VLAN Configuration
Interface-VXLAN Configuration

Command Syntax

```
load-interval delay
no load-interval
default load-interval
```

Parameters

- *delay* Load interval delay. Values range from 5 to 600 (seconds). Default value is 300 (five minutes).

Example

- These commands set the load interval for Ethernet interface 7 at 60 seconds.

```
switch(config)#interface ethernet 7
switch(config-if-Et7)#load-interval 60
switch(config-if-Et7)#
```

mac address-table aging-time

The **mac address-table aging-time** command configures the aging time for MAC address table dynamic entries. Aging time defines the period an entry is in the table, as measured from the most recent reception of a frame on the entry's VLAN from the specified MAC address. The switch removes entries when their presence in the MAC address table exceeds the aging time.

Aging time ranges from 10 to 1,000,000 seconds with a default of 300 seconds (five minutes).

The **no mac address-table aging-time** and **default mac address-table aging-time** commands reset the aging time to its default by removing the **mac address-table aging-time** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
mac-address-table aging-time period
no mac-address-table aging-time
default mac-address-table aging-time
```

Parameters

- *period* MAC address table aging time. Default is 300 seconds. Options include:
  - **0** disables deletion of table entries on the basis of aging time.
  - **10** through **1000000** (one million) aging period (seconds).

Example

- This command sets the MAC address table aging time to two minutes (120 seconds).

```
switch(config)#mac address-table aging-time 120
switch(config)#
```

mac address-table static

The **mac address-table static** command adds a static entry to the MAC address table. Each table entry references a MAC address, a VLAN, and a list of layer 2 (Ethernet or port channel) ports. The table supports three entry types: unicast drop, unicast, and multicast.

- A drop entry does not include a port.
- A unicast entry includes one port.
- A multicast entry includes at least one port.

  Packets with a MAC address (source or destination) and VLAN specified by a drop entry are dropped. Drop entries are valid for only unicast MAC addresses.

The command replaces existing dynamic or static table entries with the same VLAN-MAC address. Static entries are not removed by aging (**mac address-table aging-time**). Static MAC entries for mirror destinations or LAG members are typically avoided.

The most important byte of a MAC address distinguishes it as a unicast or multicast address:

- Unicast: most significant byte is an even number. Examples: 0200.0000.0000 1400.0000.0000
- Multicast: most significant byte is an odd number. Examples: 0300.0000.0000 2500.0000.0000

The **no mac address-table static** and **default mac address-table static** commands remove corresponding **mac address-table static** commands from ***running-config*** and MAC address table entries.

Command Mode

Global Configuration

Command Syntax

```
mac address-table static mac_address vlan v_num DESTINATION
no mac address-table static mac_address vlan v_num [DESTINATION]
default mac address-table static mac_address vlan v_num [DESTINATION]
```

Parameters

- *mac_address* Table entry's MAC address (dotted hex notation – H.H.H).
- *v_num* Table entry's VLAN.
- ***DESTINATION*** Table entry's port list.

  For multicast MAC address entries, the command may contain multiple ports, listed in any order. The CLI accepts only one interface for unicast entries.

  - **drop** creates drop entry in table. Valid only for unicast addresses.
  - **interface ethernet** *e_range* Ethernet interfaces specified by *e_range*.
  - **interface port-channel** *p_range* Port channel interfaces specified by *p_range*.
  - <no parameter> Valid for **no** and **default** commands that remove multiple table entries.

  e_range and *p_range* formats include number, range, comma-delimited list of numbers and ranges.

Example

- This command adds a static entry for unicast MAC address 0012.3694.03ec to the MAC address table.

```
switch(config)#mac address-table static 0012.3694.03ec vlan 3 interface
Ethernet 7
switch(config)#show mac address-table static
          Mac Address Table
------------------------------------------------------------------

Vlan    Mac Address       Type        Ports      Moves   Last Move
----    -----------       ----        -----      -----   ---------
   3    0012.3694.03ec    STATIC      Et7
Total Mac Addresses for this criterion: 1

          Multicast Mac Address Table
------------------------------------------------------------------

Vlan    Mac Address       Type        Ports
----    -----------       ----        -----
Total Mac Addresses for this criterion: 0

switch(config)#
```

- These commands adds a static drop entry for MAC address 0012.3694.03ec to the MAC address table, then displays the entry in the MAC address table.

```
switch(config)#mac address-table static 0012.3694.03ec vlan 3 drop
switch(config)#show mac address-table static
          Mac Address Table
------------------------------------------------------------------

Vlan    Mac Address       Type        Ports      Moves   Last Move
----    -----------       ----        -----      -----   ---------
   1    0012.3694.03ec    STATIC
Total Mac Addresses for this criterion: 1

          Multicast Mac Address Table
------------------------------------------------------------------

Vlan    Mac Address       Type        Ports
----    -----------       ----        -----
Total Mac Addresses for this criterion: 0

switch(config)#
```

- This command adds a static entry for the multicast MAC address 0112.3057.8423 to the MAC address table.

```
switch(config)#mac address-table static 0112.3057.8423 vlan 4 interface
port-channel 10 port-channel 12
switch(config)#show mac address-table
          Mac Address Table
------------------------------------------------------------------

Vlan    Mac Address       Type        Ports      Moves   Last Move
----    -----------       ----        -----      -----   ---------
Total Mac Addresses for this criterion: 0

          Multicast Mac Address Table
------------------------------------------------------------------

Vlan    Mac Address       Type        Ports
----    -----------       ----        -----
   4    0112.3057.8423    STATIC      Po10 Po12
Total Mac Addresses for this criterion: 1
switch(config)#
```

monitor link-flap policy

The **monitor link-flap policy** command places the switch in link-flap configuration mode for configuring link flap profiles and compiling a default-profile set. Link-flap configuration mode is not a group change mode; ***running-config*** is changed immediately after commands are executed. The **exit** command does not affect the configuration.

Link flap profiles are assigned to Ethernet interfaces and specify conditions that define a link-flap error. When link flap monitoring is enabled on an interface, the link-flap conditions determine when the interface is error-disabled. Multiple profiles can be assigned to an interface to monitor a set of error conditions.

Command Mode

Global Configuration

Command Syntax

```
monitor link-flap policy
```

Commands Available in link-flap Configuration Mode

- **default-profiles** configures the set of profiles that define the default-profile set.
- **profile max-flaps (Link Flap Configuration)** configures a link-flap profile

Example

- These commands place the switch in link-flap configuration mode.

```
switch(config)#monitor link-flap policy
switch(config-link-flap)#
```

- This command returns the switch to global configuration mode.

```
switch(config-link-flap)#exit
switch(config)#
```

monitor link-flap profiles

The **monitor link-flap profiles** command enables link-flap monitoring on the configuration mode interface and specifies the error-disable criteria for the interface. Entering a **monitor link-flap profiles** command replaces the corresponding statement in ***running-config***.

The command enables the following link flap detection options:

- **monitor link-flap** (***no profiles listed***): The interface detects link flaps using the criteria defined by the default-profile set (**default-profiles**).
- **monitor link-flap profiles** (***at least one profile listed***): The interface detects link flaps using the criteria of the listed profiles. Error-disable criteria require conditions that match at least one profile.
- **default monitor link-flap**: The interface detects link flaps using the **errdisable flap-setting cause link-flap** and **errdisable recovery cause** commands.
- **no monitor link-flap**: The interface does not detect link flaps.

Default monitor link flap is the default setting.

Command Mode

Interface-Ethernet Configuration
Interface-Management Configuration

Command Syntax

```
monitor link-flap [LF_PROFILES]
no monitor link-flap
default monitor link-flap
```

Parameters

- ***LF_PROFILES*** Name of link-flap profiles assigned to interface. Parameter may contain zero, one, or multiple link-flap profile names:
  - <no parameter> Link flap criteria determined by default-profile set.
  - **profiles** *profile_name* Name of single link-flap profile.
  - **profiles** *profile_name_1 profile_name_2 ... profile_name_N* List of link-flap profile names.

Example

- This command applies the LF03 and LF04 link flap profiles to Ethernet interface 33.

```
switch(config)#interface ethernet 33
switch(config-if-Et33)#monitor link-flap profiles LF03 LF04
switch(config-if-Et33)#show active
interface Ethernet33
   monitor link-flap profiles LF04 LF03
switch(config-if-Et33)#
```

- This command disables link-flap monitoring on Ethernet interface 34.

```
switch(config)#interface ethernet 34
switch(config-if-Et34)#no monitor link-flap
switch(config-if-Et34)#show active
interface Ethernet34
   no monitor link-flap
switch(config-if-Et34)#
```

monitor server-failure

The **monitor server-failure** command places the switch in server-failure configuration mode. Rapid Automated Indication of Link-Loss (RAIL) settings are configured in server-failure configuration mode. RAIL is disabled by default and is enabled by the **no shutdown** command in server-failure configuration mode.

The **no monitor server-failure** and **default monitor server-failure** commands disable RAIL and restore all settings to their default state by removing all server-failure configuration mode statements from ***running-config***.

Server-failure configuration mode is not a group change mode; ***running-config*** is changed immediately upon entering commands. Exiting server-failure configuration mode does not affect ***running-config***. The **exit** command returns the switch to global configuration mode.

Command Mode

Global Configuration

Command Syntax

```
monitor server-failure
no monitor server-failure
default monitor server-failure
```

Commands Available in server-failure Configuration Mode

- **network (server-failure configuration mode)**
- **proxy (server-failure configuration mode)**
- **shutdown (server-failure configuration mode)**

Example

- These commands place the switch in server-failure configuration mode and enables RAIL.

```
switch(config)#monitor server-failure
switch(config-server-failure)#show active
switch(config-server-failure)#no shutdown
switch(config-server-failure)#show active
monitor server-failure
   no shutdown
switch(config-server-failure)#
```

- This command deletes all server-failure configuration mode commands from ***running-config***.

```
switch(config)#no monitor server-failure
switch(config)#
```

monitor server-failure link

The **monitor server-failure link** command enables Rapid Automated Indication of Link-Loss (RAIL) on the configuration mode interface. RAIL must be properly configured globally or this command has no effect on switch operation.

When an interface monitored by RAIL goes down, the switch performs these steps for servers that the switch accesses from the interface:

Step 1 IP addresses of the servers are removed from ARP cache.

Step 2 A dynamic MAC entry is added to the MAC address table for each server. The port for each entry is listed as ***CPU***.

The **no monitor server-failure link** and **default monitor server-failure link** commands disable RAIL on the configuration mode interface by deleting the corresponding **monitor server-failure link** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration
Interface-Port-Channel Configuration

Command Syntax

```
monitor server-failure link
no monitor server-failure link
default monitor server-failure link
```

Related Commands

- **monitor server-failure** places the switch in server-failure configuration mode for configuring RAIL.

Example

- These commands enable RAIL on port channel interface 100.

```
switch(config)#interface port-channel 100
switch(config-if-Po100)#monitor server-failure link
switch(config-if-Po100)#show active
interface Port-Channel100
   monitor server-failure link
switch(config-if-Po100)#
```

monitor session destination

The **monitor session destination** command configures an interface as the destination port of a specified port mirroring session. The destination is usually an Ethernet interface, but other options are available on certain platforms (see Restrictions). The **monitor session source** command configures the source port of the mirroring session.

An interface cannot be used in more than one mirror session and cannot be simultaneously a source and destination. By default, mirror sessions duplicate ingress and egress traffic but are configurable to mirror traffic from only one direction.

Note On platforms which support the use of port channels as mirror destinations, a port channel *must not* be used as a mirror destination if it is a member of an MLAG.

The **no monitor session destination** and **default monitor session destination** commands remove the mirroring session destination assignment by deleting the corresponding **monitor session destination** command from ***running-config***. The **no monitor session** removes the entire mirroring session.

Command Mode

Global Configuration

Command Syntax

```
monitor session session_name destination INT_NAME
no monitor session session_name destination INT_NAME
default monitor session session_name destination INT_NAME
```

Parameters

- *session_name* Label assigned to port mirroring session.
- ***INT_NAME*** Destination interface for the mirroring session
  - **ethernet** *e_range* Ethernet interfaces specified by *e_range*.
  - **port-channel** *p_range* Port channel interfaces specified by *p_range*.

Restrictions

Port mirroring capacity varies by platform. Session destination capacity for switches on each platform is listed below:

- ***Arad Platform:*** Ethernet interfaces (one)
- ***FM6000 Platform:*** Ethernet interfaces (any number), Port channel interfaces (any number), CPU
- ***Petra Platform:*** Ethernet interfaces (eight for Rx or Tx sessions; four for both ways)
- ***Trident Platform:*** Ethernet interfaces (one)
- T***rident-II Platform***: Ethernet interfaces (one)

Mirrored frames use Tx properties of the lowest numbered Tx mirror source configured, when there are multiple Tx sources in a monitor session. Packets are modified based on properties.

Example

- For two source interfaces, ethernet8 and ethernet9, the allowed VLANs on ethernet8 is 10, 20, 30 and allowed VLANs on ethernet9 30, 40, and 50. The frames going out of ethernet9 tagged with 10, 20, and 30 will appear at the mirrored destination as tagged frames. The tagged frames with 40 or 50 on ethernet9 will appear at the mirrored destination as untagged frames. Since ethernet8 is the lowest numbered source interface, all Tx frames on ethernet8 will appear as tagged in the mirrored destination.

Example

- This command configures Ethernet interface 8 as the destination port for the ***redirect_1*** mirroring session.

```
switch(config)#monitor session redirect_1 destination ethernet 8
switch(config)#
```

monitor session destination cpu

The **monitor session destination cpu** command configures the CPU as the destination port of a specified port mirroring session. The **monitor session source** command configures the source port of the mirroring session. By default, mirror sessions duplicate ingress and egress traffic but are configurable to mirror traffic from one direction.

The CPU can only be configured as a destination for a mirroring session, not as a source. However, the CPU can serve as the destination for multiple mirroring sessions. Traffic mirrored to the CPU can be viewed using tcpdump.

The **no monitor session destination cpu** and **default monitor session destination cpu** commands remove the mirror session destination assignment by deleting the corresponding **monitor session destination cpu** command from ***running-config***. The **no monitor session** command removes the entire mirror session.

Command Mode

Global Configuration

Command Syntax

```
monitor session session_name destination cpu
no monitor session session_name destination cpu
default monitor session session_name destination cpu
```

Parameters

- *session_name* Label assigned to port mirroring session.

Guidelines

To view the traffic mirrored to the CPU from a source port, use tcpdump from the Bash shell, with the source interface as an argument. This causes tcpdump to capture packets from the kernel interface of the source port.

Examples

- These commands configure Ethernet interface 35 as the source and the CPU as the destination port for the ***redirect_1*** mirroring session, then display the mirror interface.

```
switch(config)#monitor session redirect_1 destination cpu
switch(config)#monitor session redirect_1 source ethernet 35
switch(config)#show monitor session

Session redirect_1
------------------------

Source Ports:

  Both:        Et35

Destination Ports:

    Cpu :  active (mirror0)

switch(config)#
```

- This command uses tcpdump to view the traffic mirrored by the redirect_1 mirroring session. The CPU mirror interface specified in the previous output must be used in the tcpdump expression (in this case, mirror0).

```
switch#bash tcpdump -i mirror0
tcpdump: WARNING: mirror0: no IPv4 address assigned
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on mirror0, link-type EN10MB (Ethernet), capture size 65535 bytes
09:51:12.478363 00:1c:73:27:a6:d3 (oui Arista Networks) > 01:80:c2:00:00:00 (oui
Unknown), 802.3, length 119: LLC, dsap STP (0x42) Individual, ssap STP (0x42)
Command, ctrl 0x03: STP 802.1s, Rapid STP, CIST Flags [Proposal, Learn, Forward,
Agreement], length 102
09:51:14.478235 00:1c:73:27:a6:d3 (oui Arista Networks) > 01:80:c2:00:00:00 (oui
Unknown), 802.3, length 119: LLC, dsap STP (0x42) Individual, ssap STP (0x42)
Command, ctrl 0x03: STP 802.1s, Rapid STP, CIST Flags [Proposal, Learn, Forward,
Agreement], length 102
switch#
```

monitor session ip access-group

The **monitor session ip access-group** command configures an ACL to filter the traffic being mirrored to the destination port.ACLs applied to a source port affect the RX side of the interface, and do not impact the TX side of the interface. TX mirrored packets cannot be filtered, and will continue to be sent to the mirror destination.

The **no monitor session ip access-group** and **default monitor session ip access-group** commands remove the filter from the specified mirror session by deleting the corresponding **monitor session ip access-group** command from ***running-config***. The **no monitor session** command removes the entire mirror session.

Command Mode

Global Configuration

Command Syntax

```
monitor session session_name ip access-group acl_name
no monitor session session_name ip access-group
default monitor session session_name ip access-group
```

Parameters

- *session_name* Label assigned to port mirroring session.
- *acl_name* The ACL to be applied to filter traffic for the specified session.

Examples

- These commands create an ACL and apply it to filter the traffic mirrored to the destination port by session “redirect_1.”

```
switch(config)#ip access-list allow-host
switch(config-acl-allow-host)#10 permit ip host 192.168.11.24 host 10.0.215.23
switch(config-acl-allow-host)#20 deny ip any any
switch(config-acl-allow-host)#exit
switch(config)#
switch(config)#monitor session redirect_1 ip access-group allow-host
switch(config)#
```

- Use the **show monitor session** command to verify the configuration.

```
switch#show monitor session
Session redirect_1
------------------------
Source Ports:
Both:        Et35(Acl:allow-host)
Destination Ports:
Cpu :  active (mirror0)
ip access-group: allow-host
switch#
```

monitor session source

The **monitor session source** command configures the source port of a specified port mirroring session. The **monitor session destination** or **monitor session destination cpu** command configures the destination port of the mirroring session.

An interface cannot be used in more than one mirror session and cannot be simultaneously a source and a destination. An interface which is part of a port channel cannot be used as a source, but a port channel which is a member of an MLAG can be used. By default, mirror sessions duplicate ingress and egress traffic but are configurable to mirror traffic from only one direction.

The **no monitor session source** and **default monitor session source** commands remove the mirroring session source assignment by deleting the corresponding **monitor session source** command from ***running-config***. The **no monitor session** removes entire the mirroring session.

Command Mode

Global Configuration

Command Syntax

```
monitor session session_name source INT_NAME DIRECTION
no monitor session session_name source INT_NAME DIRECTION
default monitor session session_name source INT_NAME DIRECTION
```

Parameters

- *session_name* Label assigned to port mirroring session.
- ***INT_NAME*** Source interface for the mirroring session.
  - **ethernet** *e_range* Ethernet interfaces specified by *e_range*.
  - **port-channel** *p_range* Port channel interfaces specified by *p_range*.
- ***DIRECTION*** transmission direction of traffic to be mirrored.
  - <no parameter> mirrors transmitted and received traffic.
  - **both** mirrors transmitted and received traffic.
  - **rx** mirrors received traffic only.
  - **tx** mirrors transmitted traffic only.

Guidelines

On DCS-7050, DCS-7050X, DCS-7250X, and DCS-7300X series, due to limitations of the switch ASIC, all frames mirrored on egress are prefixed with an 802.1Q VLAN tag, even when the egress port is configured as an access port. If the capture device is unable to process VLAN tags in a desirable manner mirroring should be configured exclusively for ingress traffic by specifying **rx**.

Restrictions

Port mirroring capacity varies by platform. Session source capacity for each platform is listed below:

- ***FM6000 Platform:*** Ethernet interfaces (any number), port channel interfaces (any number)
- ***Arad Platform:*** Ethernet interfaces (any number), port channel interfaces (any number).
- ***Petra Platform:*** Ethernet interfaces (eight for Rx or Tx sessions; four for both ways)
- ***Trident Platform:*** Ethernet interfaces (any number), port channel interfaces (any number)
- ***Trident-II Platform:*** Ethernet interfaces (any number), port channel interfaces (any number)

The number of interfaces that can be effectively mirrored is restricted by the destination port speed.

Example

- This command configures Ethernet interface 7 as the source port for ***redirect_1*** mirroring session.

```
switch(config)#monitor session redirect_1 source ethernet 7
switch(config)#
```

monitor session source ip access-group

The **monitor session source ip access-group** command configures an ACL to filter the traffic being mirrored from a specific source port. This enables the ability to filter traffic using a different ACL on each source port and have the combined matched traffic sent to the destination port.

The **no monitor session source ip access-group** and **default monitor session source ip access-group** commands remove the filter from the specified mirror session by deleting the corresponding **monitor session source ip access-group** command from ***running-config***. The **no monitor session** command removes the entire mirror session.

Command Mode

Global Configuration

Command Syntax

```
monitor session s_name source INT_NAME [DIRECT] ip access-group acl_name
no monitor session s_name source INT_NAME [DIRECT] ip access-group acl_name
default monitor session s_name source INT_NAME [DIRECT] ip access-group acl_name
```

Parameters

- *s_name* Label assigned to port mirroring session.
- ***INT_NAME*** Source interface for the mirroring session.
  - **ethernet** *e_range* Ethernet interfaces specified by *e_range*.
  - **port-channel** *p_range* Port channel interfaces specified by *p_range*.
- ***DIRECT*** transmission direction of traffic to be mirrored. Options include:
  - <no parameter> mirrors received traffic only.
  - **rx** mirrors received traffic only.
- *acl_name* The ACL to be applied to filter traffic for the specified session.

Examples

- These commands create ACLs and apply them to filter the traffic mirrored from two source ports by session “redir_1.”

```
switch(config)#ip access-list allow-host-x
switch(config-acl-allow-host-x)#10 permit ip host 192.168.11.24 host 10.0.215.23
switch(config-acl-allow-host-x)#20 deny ip any any
switch(config-acl-allow-host-x)#exit
switch(config)#ip access-list allow-host-y
switch(config-acl-allow-host-y)#10 permit ip host 172.16.233.80 host 10.0.215.23
switch(config-acl-allow-host-y)#20 deny ip any any
switch(config-acl-allow-host-y)#exit
switch(config)#monitor session redir_1 source ethernet 5,9 rx
switch(config)#monitor session redir_1 source ethernet 5 ip access-group
allow-host-x
switch(config)#monitor session redir_1 source ethernet 9 ip access-group
allow-host-y
switch(config)#
```

monitor session truncate

The **monitor session truncate** command configures a port mirroring session to truncate mirrored packets, retaining only the first 160 bytes. Packet truncation can be used to prevent oversubscription of the session's destination port.

Packet truncation applies to the mirroring session as a whole, and cannot be applied to individual source ports.

The **no monitor session truncate** and **default monitor session truncate** commands restores mirroring of full packets by deleting the corresponding **monitor session truncate** command from ***running-config***. The **no monitor session** removes the entire mirroring session.

Command Mode

Global Configuration

Command Syntax

```
monitor session session_name truncate
no monitor session session_name truncate
default monitor session session_name truncate
```

Parameters

- *session_name* Label assigned to port mirroring session.

Examples

- This command configures mirroring session ***redirect_1*** to truncate mirrored packets.

```
switch(config)#monitor session redirect_1 truncate
switch(config)#
```

mtu

The **mtu** command configures the IPv4 and IPv6 Maximum Transmission Unit (MTU) size for the configuration mode interface. The switch fragments IP packets that are larger than the MTU value for the outbound interface. An interface's MTU value is displayed with the **show interfaces** command.

MTU is independently configurable on all routable interfaces. The switch supports MTU sizes ranging from 68 to 9214 bytes. The default MTU size is 1500 bytes.

The **no mtu** and **default mtu** commands restore the interface's MTU to the default value by removing the corresponding mtu command from ***running-config***.

Command Mode

Interface-Ethernet Configuration
Interface-Loopback Configuration
Interface-Management Configuration
Interface-Port-channel Configuration
Interface-VLAN Configuration

Command Syntax

```
mtu bytes
no mtu
default mtu
```

Parameters

- *bytes* MTU size (bytes). Values range from 68 to 9214.

Examples

- This command sets the MTU size of 1492 bytes on VLAN interface 20.

```
switch(config)#interface vlan 20
switch(config-if-Vl20)#mtu 1492
switch(config-if-Vl20)#
```

network (server-failure configuration mode)

The **network** command specifies the IPv4 network space that Rapid Automated Indication of Link-Loss (RAIL) monitors for failed links to connected servers. RAIL reduces the wait time for applications on directly connected servers that are blocked due to a failed link. ***Running-config*** supports simultaneous network command, allowing RAIL to monitor multiple disjoint network spaces.

When a server on the specified network is blocked because of a failed Ethernet or port channel link, the switch becomes a proxy for the unavailable server and responds with ***TCP RST*** or ***ICMP Unreachable*** segments to devices sending packets to the unavailable server.

The **no network** and **default network** commands terminate the RAIL monitoring of the specified IPv4 address space by deleting the corresponding **network** command from ***running-config***.

Command Mode

Server-failure Configuration

Command Syntax

```
network netv4_address
no network netv4_address
default network netv4_address
```

Parameters

- *netv4_addr* IPv4 subnet address to be monitored (CIDR or address-mask notation).

Related Commands

- **monitor server-failure** places the switch in server-failure configuration mode.

Example

- This command specifies two IPv4 network spaces that RAIL monitors for server failures.

```
switch(config)#monitor server
switch(config-server-failure)#network 10.1.1.0/24
switch(config-server-failure)#network 10.2.1.96/28
switch(config-server-failure)#show active
monitor server-failure
   network 10.2.1.96/28
   network 10.1.1.0/24
switch(config-server-failure)#
```

no monitor session

The **no monitor session** and default monitor session commands remove the specified monitor session from the switch by deleting all corresponding monitor commands from ***running-config***. Commands that remove or alter individual commands within a session configuration are described in the **monitor session destination** and **monitor session source** commands.

Command Mode

Global Configuration

Command Syntax

```
no monitor session session_name
default monitor session session_name
```

Parameters

- *session_name* Label assigned to port mirroring session.

Example

- This command displays the configuration of the ***redirect_1*** mirroring session, deletes the session, then confirms that the session was removed.

```
switch(config)#show monitor session redirect_1

Session redirect_1
------------------------

Source Ports

  Both:        Et7

Destination Port: Et8
switch(config)#no monitor session redirect_1
switch(config)#show monitor session redirect_1
Session not created

switch(config)#
```

platform sand monitor serdes error log

The **platform sand monitor serdes error log** command is used for enabling the serdes error log for fabric link monitoring.

Command Mode

Global Configuration

Command Syntax

```
platform sand monitor serdes error log
```

Example

- This command enables the serdes error log for fabric link monitoring.

```
switch(config)#platform sand monitor serdes error log
switch(config)#
```

platform sand monitor serdes error threshold

The **platform sand monitor serdes error threshold** command is used for generating a fabric link monitoring serdes error threshold.

Command Mode

Global Configuration

Command Syntax

platform sand monitor serdes error threshold

Example

- This command monitors serdes error thresholds over the specified number of received cells, resulting in the isolation of a fabric link between 200 and 30,000 received cells.

```
switch(config)#platform sand monitor serdes error threshold 200 30000
switch(config)#
```

platform sand monitor serdes poll period

The **platform sand monitor serdes poll period** command is used to enable the serdes poll period.

Command Mode

Global Configuration

Command Syntax

```
platform sand monitor serdes poll period
```

Example

- This command changes the serdes polling period for fabric link monitoring to 6 seconds.

```
switch(config)#platform sand monitor serdes poll period 6
switch(config)#
```

platform sand monitor serdes poll threshold isolation

The **platform sand monitor serdes poll threshold isolation** command is used to set and enables fabric link monitoring for serdes poll threshold isolation.

Command Mode

Global Configuration

Command Syntax

```
platform sand monitor serdes poll threshold isolation
```

Example

- This command changes the number of consecutive polls in which the threshold needs to be detected to isolate a link. In this case the number is 5 consecutive polls.

```
switch(config)#platform sand monitor serdes poll threshold isolation 5
switch(config)#
```

platform sand monitor serdes poll threshold recovery

The **platform sand monitor serdes poll threshold recovery** command is used to set and enable fabric link monitoring for serdes poll threshold recovery.

Command Mode

Global Configuration

Command Syntax

```
platform sand monitor serdes poll threshold recovery
```

Example

- This command changes the number of consecutive serdes polls used for threshold recovery to 6 seconds.

```
switch(config)#platform sand monitor serdes poll threshold recovery 6
switch(config)#
```

profile max-flaps (Link Flap Configuration)

The **profile max-flaps** command creates a link flap profile that, when assigned to an Ethernet interface, specifies the conditions that result in an error-disable action. Link flap profile parameters include

- ***flaps*** Threshold number of interface state changes.
- ***period*** Interval when link flaps accumulate to trigger an error condition.
- ***violations*** Number of link flap errors (threshold exceeded over specified period).
- ***intervals*** Quantity of periods.

By default, ***violations*** and ***intervals*** are each set to one, resulting in a profile that triggers a link-flap error when the specified frequency is exceeded once. By configuring violations and intervals, link-flap errors are defined when the frequency is exceeded multiple times over a specified set of intervals.

Default is a reserved profile name that modifies the **errdisable flap-setting cause link-flap** statement in ***running-config***. When configuring the **default** profile, ***violations*** and ***intervals*** are disregarded.

The **no profile max-flaps** command removes the specified profile by deleting the corresponding **profile max-flaps** command from ***running-config***. The **no profile max-flaps default** command restores default **errdisable flap-setting cause link-flap** values by removing that command from ***running-config***.

Command Mode

Link-flap Configuration

Command Syntax

```
profile PROFILE_NAME max-flaps flap_max time period [EXTENTIONS]
no profile LF_PROFILE
```

Parameters

- ***PROFILE_NAME*** Name of link flap profile. Options include:
  - **default** command modifies default values (**errdisable flap-setting cause link-flap**).
  - *profile_name* command modifies specified link-flap profile.
- *flap_max* Threshold number of interface state changes. Value ranges from 1 to 100.
- *period* Interval when flaps accumulate toward threshold (seconds). Value ranges from 1 to 1800.
- ***EXTENSIONS*** Configures multi-flap triggers. Options include:
  - <no parameter> Sets errors and episodes to default values (one).
  - **violations** *errors* **intervals** *episodes* Link flap errors (*errors*) and number of periods (*episodes*).

    Errors range is 1 to 1000. Default value is one.

    Episodes range is 1 to 1000. Default value is one.

Related Commands

- **monitor link-flap policy** places the switch in link-flap configuration mode.

Example

- These commands create two link flap profiles with various trigger settings.

```
switch(config)#monitor link-flap policy
switch(config-link-flap)#profile LF01 max-flaps 15 time 60
switch(config-link-flap)#profile LF02 max-flaps 10 time 30 violations 5 intervals
10
switch(config-link-flap)#show active
monitor link-flap policy
   profile LF01 max-flaps 15 time 60 violations 1 intervals 1
   profile LF02 max-flaps 10 time 30 violations 5 intervals 10
switch(config-link-flap)#
```

proxy (server-failure configuration mode)

The **proxy** command enables the Rapid Automated Indication of Link-Loss (RAIL) proxy setting and specifies the interval that RAIL responds to messages sent to servers on failed links, starting from when the switch detects the failed link. The RAIL state machine is in the proxying state during the timeout interval this command specifies. When RAIL proxy is not enabled, the switch maintains a list of unavailable servers without responding to messages sent the servers. The switch can enter RAIL proxy state only when this command is enabled.

The RAIL proxy setting is ***disabled*** by default. When RAIL proxy is enabled, the default period is three minutes.

The **no proxy** and **default proxy** commands return the RAIL proxy setting to disabled by removing the proxy statement from ***running-config***.

The **no proxy lifetime** and **default proxy lifetime** command sets the proxy time setting to its default value of three minutes if the RAIL proxy setting is ***enabled***. These commands have no effect if the RAIL proxy setting is ***disabled***.

Command Mode

Server-failure Configuration

Command Syntax

```
proxy [lifetime time_span]
no proxy
no proxy [lifetime]
default proxy
default proxy [lifetime]
```

Parameters

- *timespan* proxy timeout period (minutes). Value ranges from 1 to 10080. Default value is 3.

Related Commands

- **monitor server-failure** places the switch in server-failure configuration mode.

Example

- These commands enable the RAIL proxy and sets the proxy timeout period of 10 minutes.

```
switch(config)#monitor server
switch(config-server-failure)#proxy lifetime 10
switch(config-server-failure)#show active
monitor server-failure
   proxy lifetime 10
switch(config-server-failure)#
```

- This command sets the proxy timeout period to its default value of 3 minutes.

```
switch(config-server-failure)#no proxy lifetime
switch(config-server-failure)#show active
monitor server-failure
   proxy
switch(config-server-failure)#
```

- This command disables the RAIL proxy.

```
switch(config-server-failure)#no proxy
switch(config-server-failure)#show active
monitor server-failure
switch(config-server-failure)#
```

show fabric monitoring health

The **platform sand monitor health** command is used to display the fabric monitoring connected state status with isolated links.

Command Mode

Global Configuration

Command Syntax

platform sand monitor health

Example

- This command displays the connected state status with isolated links.

```
switch(config)#show platform sand health
Fabric serdes isolated by fabric monitoring: (36 total)

Arad5/0 serdes [0-1, 10-19, 2, 20-29, 3, 30-35, 4-9]

Top fabric serdes list by number of times isolated by monitoring:
Arad5/0 serdes 0: 1 (last occurred: 0:01:04 ago)
Arad5/0 serdes 1: 1 (last occurred: 0:01:04 ago)
Arad5/0 serdes 10: 1 (last occurred: 0:01:04 ago)
Arad5/0 serdes 11: 1 (last occurred: 0:01:04 ago)
Arad5/0 serdes 12: 1 (last occurred: 0:01:04 ago)
Arad5/0 serdes 13: 1 (last occurred: 0:01:04 ago)
Arad5/0 serdes 14: 1 (last occurred: 0:01:04 ago)
Arad5/0 serdes 15: 1 (last occurred: 0:01:04 ago)
Arad5/0 serdes 16: 1 (last occurred: 0:01:04 ago)
Arad5/0 serdes 17: 1 (last occurred: 0:01:04 ago)

switch(config)#
```

show interfaces

The **show interfaces** command displays operational status and configuration information of specified interfaces. The output includes speed, duplex, flow control information and basic interface statistics.

The input and output bit rates, as displayed, do not include framing bits that are part of the Ethernet standard, the inter-frame gap and preamble that total 20 bytes per packet. The percentage number includes those framing bits to provide a better link utilization estimate.

Command Mode

EXEC

Command Syntax

```
show interfaces [INT_NAME]
```

Parameters

- ***INT_NAME*** Interface type and numbers. Options include:
  - <no parameter> all interfaces.
  - **ethernet** *e_range* Ethernet interface range specified by *e_range*.
  - **loopback** *l_range* Loopback interface specified by *l_range*.
  - **management** *m_range* Management interface range specified by *m_range*.
  - **port-channel** *p_range* Port-Channel Interface range specified by *p_range*.
  - **vlan** *v_range* VLAN interface range specified by *v_range*.
  - **vxlan** *vx_range* VXLAN interface range specified by *vx_range*.

    Valid range formats include number, number range, or comma-delimited list of numbers and ranges.

Example

- This command display configuration and status information for Ethernet interface 1 and 2.

```
switch>show interfaces ethernet 1-2
Ethernet1 is up, line protocol is up (connected)
  Hardware is Ethernet, address is 001c.2481.7647 (bia 001c.2481.7647)
  Description: mkt.1
  MTU 9212 bytes, BW 10000000 Kbit
  Full-duplex, 10Gb/s, auto negotiation: off
  Last clearing of "show interface" counters never
  5 seconds input rate 33.5 Mbps (0.3% with framing), 846 packets/sec
  5 seconds output rate 180 kbps (0.0% with framing), 55 packets/sec
     76437268 packets input, 94280286608 bytes
     Received 2208 broadcasts, 73358 multicast
     0 runts, 0 giants
     0 input errors, 0 CRC, 0 alignment, 0 symbol
     0 PAUSE input
     6184281 packets output, 4071319140 bytes
     Sent 2209 broadcasts, 345754 multicast
     0 output errors, 0 collisions
     0 late collision, 0 deferred
     0 PAUSE output
Ethernet2 is up, line protocol is up (connected)
  Hardware is Ethernet, address is 001c.2481.7648 (bia 001c.2481.7648)
  Description: mkt.2
  MTU 9212 bytes, BW 10000000 Kbit
  Full-duplex, 10Gb/s, auto negotiation: off
  Last clearing of "show interface" counters never
  5 seconds input rate 711 kbps (0.0% with framing), 271 packets/sec
  5 seconds output rate 239 kbps (0.0% with framing), 65 packets/sec
     73746370 packets input, 78455101010 bytes
     Received 11 broadcasts, 83914 multicast
     0 runts, 0 giants
     0 input errors, 0 CRC, 0 alignment, 0 symbol
     0 PAUSE input
     5687714 packets output, 4325064454 bytes
     Sent 15 broadcasts, 107279 multicast
     0 output errors, 0 collisions
     0 late collision, 0 deferred
     0 PAUSE output
switch>
```

show interfaces description

The **show interfaces description** command displays the status and description text of the specified interfaces. The **description** command configures an interface's description parameter.

Command Mode

EXEC

Command Syntax

```
show interfaces [INT_NAME] description
```

Parameters

- ***INT_NAME*** Interface type and labels. Options include:
  - <no parameter> all interfaces.
  - **ethernet** *e_range* Ethernet interface range specified by *e_range*.
  - **loopback** *l_range* Loopback interface specified by *l_range*.
  - **management** *m_range* Management interface range specified by *m_range*.
  - **port-channel** *p_range* Port-Channel Interface range specified by *p_range*.
  - **vlan** *v_range* VLAN interface range specified by *v_range*.
  - **vxlan** *vx_range* VXLAN interface range specified by *vx_range*.

    Range formats include number, number range, or comma-delimited list of numbers and ranges.

Example

- This command displays description text and status of ethernet interfaces 1-10.

```
switch>show interfaces ethernet 1-10 description
Interface                      Status         Protocol Description
Et1                            up             up       ctar_01
Et2                            up             up       ctar_02
Et3                            up             up       ctar_03
Et4                            up             up       fobd_01
Et5                            up             up       fobd_02
Et6                            up             up       yzrq_01
Et7                            up             up       yzrq_02
Et8                            down           down     yzrq_03
Et9                            up             up       yzrq_04
Et10                           up             up       yzrq_05
switch>
```

show link state group

The **show link state group** command displays information about a specified link-state group or about all groups.

Command Mode

EXEC

Command Syntax

show link state group [*DATA_LEVEL*] [*GROUPS*]

Parameters

- ***DATA_LEVEL*** device for which the command provides data. Options include:
  - <no parameter> information about all groups in group list.
  - **detail** detailed information about all groups in group list.
- ***GROUPS***
  - <no parameter> all link-state groups.
  - *group_name* link-state group name.

Example

- This command displays all the link-state group information.

```
switch# show link state group detail
Link State Group: 1 Status: up
Upstream Interfaces : Vlan100
Downstream Interfaces : Vlan200
Number of times disabled : 2
Last disabled 0:10:29 ago

Link State Group: group3 Status: down
Upstream Interfaces : Ethernet24
Downstream Interfaces : Ethernet8
Number of times disabled : 2
Last disabled 0:30:35 ago

Link State Group: 2 Status: up
Upstream Interfaces : Ethernet2 Ethernet5
Downstream Interfaces : Ethernet12
Number of times disabled : 0
Last disabled never
switch#
```

show mac address-table

The **show mac-address-table** command displays the specified MAC address table entries.

Command Mode

EXEC

Command Syntax

show mac address-table [*ENTRY_TYPE*] [*MAC_ADDR*] [*INTF_1 ... INTF_N*] [*VLANS*]

Parameters

- ***ENTRY_TYPE*** command filters display by entry type. Entry types include mlag-peer, dynamic, static, unicast, multicast entries, and configured.
  - <no parameter> all table entries.
  - **configured** static entries; includes unconfigured VLAN entries.
  - **dynamic** entries learned by the switch.
  - **static** entries entered by CLI commands and include a configured VLAN.
  - **unicast** entries with unicast MAC address.
- ***MAC_ADDR*** command uses MAC address to filter displayed entries.
  - <no parameter> all MAC addresses table entries.
  - **address** *mac_address* displays entries with specified address (dotted hex notation – H.H.H).
- ***INTF_X*** command filters display by port list. When parameter lists multiple interfaces, command displays all entries containing at least one listed interface.
  - <no parameter> all Ethernet and port channel interfaces.
  - **ethernet** *e_range* Ethernet interfaces specified by *e_range*.
  - **port-channel** *p_range* Port channel interfaces specified by *p_range*.
- ***VLANS*** command filters display by VLAN.
  - <no parameter> all VLANs.
  - **vlan** *v_num* VLANs specified by *v_num*.

Related Commands

- **show mac address-table mlag-peer**
- **show mac address-table multicast**

Example

- This command displays the MAC address table.

```
switch>show mac address-table
          Mac Address Table
------------------------------------------------------------------

Vlan    Mac Address       Type        Ports      Moves   Last Move
----    -----------       ----        -----      -----   ---------
 101    001c.8224.36d7    DYNAMIC     Po2        1       9 days, 15:57:28 ago
 102    001c.8220.1319    STATIC      Po1
 102    001c.8229.a0f3    DYNAMIC     Po1        1       0:05:05 ago
 661    001c.8220.1319    STATIC      Po1
 661    001c.822f.6b22    DYNAMIC     Po7        1       0:20:10 ago
3000    001c.8220.1319    STATIC      Po1
3000    0050.56a8.0016    DYNAMIC     Po1        1       0:07:38 ago
3902    001c.8220.1319    STATIC      Po1
3902    001c.822b.a80e    DYNAMIC     Po4        2       9 days, 15:57:30 ago
3903    001c.8220.1319    STATIC      Po1
3903    001c.822c.3009    DYNAMIC     Po5        1       4 days, 15:13:03 ago
3908    001c.8220.1319    STATIC      Po1
3908    001c.822c.4e1d    DYNAMIC     Po1        1       0:07:26 ago
3908    001c.822c.55d9    DYNAMIC     Po1        1       0:04:33 ago
3909    001c.8220.1319    STATIC      Po1
3909    001c.822f.6a80    DYNAMIC     Po1        1       0:07:08 ago
3910    001c.730f.6a80    DYNAMIC     Et9        1       4 days, 15:13:07 ago
3911    001c.8220.1319    STATIC      Po1
3911    001c.8220.40fa    DYNAMIC     Po8        1       1:19:58 ago
3912    001c.822b.033e    DYNAMIC     Et11       1       9 days, 15:57:23 ago
3913    001c.8220.1319    STATIC      Po1
3913    001c.822b.033e    DYNAMIC     Po1        1       0:04:35 ago
3984    001c.8220.178f    DYNAMIC     Et8        1       4 days, 15:07:29 ago
3992    001c.8220.1319    STATIC      Po1
3992    001c.8221.07b9    DYNAMIC     Po6        1       4 days, 15:13:15 ago
Total Mac Addresses for this criterion: 25

          Multicast Mac Address Table
------------------------------------------------------------------

Vlan    Mac Address       Type        Ports
----    -----------       ----        -----
Total Mac Addresses for this criterion: 0
switch>
```

show mac address-table aging time

The **show mac-address-table aging time** command displays the aging time for MAC address table dynamic entries. Aging time defines the period an entry is in the table, as measured from the most recent reception of a frame on the entry's VLAN from the specified MAC address. The switch removes entries that exceed the aging time.

Aging time ranges from 10 seconds to 1,000,000 seconds with a default of 300 seconds (five minutes).

Command Mode

EXEC

Command Syntax

show mac address-table aging-time

Examples

- This command shows the MAC address table aging time

```
switch>show mac address-table aging-time
Global Aging Time:  120
switch>
```

show mac address-table count

The **show mac-address-table count** command displays the number of entries in the MAC address table for the specified VLAN or for all VLANs.

Command Mode

EXEC

Command Syntax

show mac address-table count [*VLANS*]

Parameters

- ***VLANS*** The VLANs for which the command displays the entry count.
  - <no parameter> all configured VLANs.
  - **vlan** *v_num* VLAN interface specified by *v_num*.

Examples

- This command displays the number of entries on VLAN 39

```
switch>show mac address-table count vlan 39

Mac Entries for Vlan 39:
---------------------------
Dynamic Address Count            : 1
Unicast Static  Address Count    : 1
Multicast Static  Address Count  : 0
Total Mac Addresses              : 2

switch>
```

show mac address-table mlag-peer

The **show mac-address-table mlag-peer** command displays the specified MAC address table entries learned from the MLAG peer switch.

Command Mode

EXEC

Command Syntax

```
show mac address-table mlag-peer [ENTRY_TYPE][MAC_ADDR][INTF_1 ...
INTF_N][VLANS]
```

Parameters

- ***ENTRY_TYPE*** command filters display by entry type. Entry types include mlag-peer, dynamic, static, unicast, multicast entries, and configured.
  - <no parameter> all MLAG peer entries.
  - **configured** static entries on MLAG peer; includes unconfigured VLAN entries.
  - **dynamic** entries learned on MLAG peer.
  - **static** MLAG entries entered by CLI commands and include a configured VLAN.
  - **unicast** MLAG entries with unicast MAC address.
- ***MAC_ADDR*** command uses MAC address to filter displayed entries.
  - <no parameter> all MAC addresses table entries.
  - **address** *mac_address* displays entries with specified address (dotted hex notation – H.H.H).
- ***INTF_X*** command filters display by port list. When parameter lists multiple interfaces, command displays all entries containing at least one listed interface.
  - <no parameter> all Ethernet and port channel interfaces.
  - **ethernet** *e_range* Ethernet interfaces specified by *e_range*.
  - **port-channel** *p_range* Port channel interfaces specified by *p_range*.
- ***VLANS*** command filters display by VLAN.
  - <no parameter> all VLANs.
  - **vlan** *v_num* VLANs specified by *v_num*.

Related Commands

- **show mac address-table**
- **show mac address-table multicast**

show mac address-table multicast

The **show mac-address-table** command displays the specified multicast MAC address table entries.

Command Mode

EXEC

Command Syntax

show mac address-table multicast [*MAC_ADDR*] [*INTF*] [*VLANS*]

Parameters

- ***MAC_ADDR*** command uses MAC address to filter displayed entries.
  - <no parameter> all MAC addresses table entries.
  - **address** *mac_address* displays entries with specified address (dotted hex notation – H.H.H).
- ***INTF*** command filters display by port list. When parameter lists multiple interfaces, command displays all entries containing at least one listed interface.
  - <no parameter> all Ethernet and port channel interfaces.
  - **ethernet** *e_range* Ethernet interfaces specified by *e_range*.
  - **port-channel** *p_range* Port channel interfaces specified by *p_range*.
- ***VLANS*** command filters display by VLAN.
  - <no parameter> all VLANs.
  - **vlan** *v_num* VLANs specified by *v_num*.

Related Commands

- **show mac address-table**
- **show mac address-table multicast brief**

show mac address-table multicast brief

The **show mac-address-table** command displays a summary of multicast MAC address table entries.

Command Mode

EXEC

Command Syntax

show mac address-table multicast [*VLANS*] brief

Parameters

- ***VLANS*** command filters display by VLAN.
  - <no parameter> all VLANs.
  - **vlan** *v_num* VLANs specified by *v_num*.

Related Commands

- **show mac address-table multicast**

show monitor server-failure

The **show monitor server-failure** command displays Rapid Automated Indication of Link-Loss (RAIL) configuration settings and the number of servers on each monitored network.

Command Mode

EXEC

Command Syntax

show monitor server-failure

Example

- This command displays RAIL configuration status and lists the number of servers that are on each monitored network.

```
switch>show monitor server-failure
Server-failure monitor is enabled
Proxy service: disabled
Networks being monitored: 3
   10.2.1.96/28      : 0 servers
   10.1.1.0/24       : 0 servers
   10.3.0.0/16       : 3 servers
switch>
```

show monitor server-failure history

The **show monitor server-failure history** command displays the time of all link failures detected by Rapid Automated Indication of Link-Loss (RAIL) and includes the interface name for each failure.

The history is cleared by removing RAIL from the switch (**no monitor server-failure**).

Command Mode

EXEC

Command Syntax

```
show monitor server-failure history
```

Related Commands

- **clear server-failure servers inactive**

Example

- This command displays the Fast Server Failure link failure history from the time RAIL is instantiated on the switch.

```
switch>show monitor server-failure history
Total server failures: 4

Server IP   Server MAC             Interface          Last Failed
----------- -----------------      -----------        -------------------
10.1.67.92  01:22:ab:cd:ee:ff      Ethernet17         2013-02-02 11:26:22
44.11.11.7  ad:3e:5f:dd:64:cf      Ethernet23         2013-02-10 00:07:56
10.1.1.1    01:22:df:42:78:cd      Port-Channel6      2013-02-09 19:36:09
10.1.8.13   01:33:df:ee:39:91      Port-Channel5      2013-02-10 00:03:39

switch>
```

show monitor server-failure servers

The **show monitor server-failure servers** command displays status and configuration information about each server that RAIL is monitoring. The display format depends on the parameter specified by the command:

- **single IP address: command displays information about the server at the specified address, including IP address, MAC address, RAIL state, the time of most recent entry of all RAIL states, and the number of failed, proxied, and inactive state entries.**
- no parameter, key specifying a server list: command displays a table. Each row corresponds to a monitored server. Information that the command displays includes **IP address, MAC address, RAIL state, the time of most recent link failure.**

Command Mode

EXEC

Command Syntax

show monitor server-failure servers [*SERVER_LIST*]

Parameters

- ***SERVER_LIST*** Servers for which command displays information. Valid options include:
  - <no parameter> all servers in up, down, and proxying states.
  - *ipv4_addr* individual server; command displays detailed information.
  - **all** all servers on monitored networks.
  - **inactive** all servers in inactive state.
  - **proxying** all servers in proxying state.

Example

- This command displays RAIL information for the server at IP address 10.11.11.7

```
switch>show monitor server-failure servers 10.11.11.7
Server information:
Server Ip Address        : 10.11.11.7
MAC Address              : ad:3e:5f:dd:64:cf
Current state            : down
Interface                : Ethernet23
Last Discovered          : 2013-01-06 06:47:39
Last Failed              : 2013-02-10 00:07:56
Last Proxied             : 2013-02-10 00:08:33
Last Inactive            : 2013-02-09 23:52:21
Number of times failed   : 3
Number of times proxied  : 1
Number of times inactive : 18

switch>
```

- This command displays RAIL data for all servers in monitored networks that are in inactive state.

```
switch>show monitor server-failure servers inactive
Inactive servers: 1

Server IP   Server MAC         Interface    State       Last Failed
----------  -----------------  -----------  --------    -------------
10.1.67.92  01:22:ab:cd:ee:ff  Ethernet17   inactive    7 days, 12:48:06 ago

switch>
```

- This command displays RAIL information for all servers in monitored networks that are in up, down, and proxying states.

```
switch>show monitor server-failure servers
Active servers: 4

Server IP   Server MAC            Interface         State        Last Failed
----------  -----------------     --------------    ---------    -----------
44.11.11.7  ad:3e:5f:dd:64:cf     Ethernet23        down         0:03:21 ago
10.1.1.1    01:22:df:42:78:cd     Port-Channel6     up           4:35:08 ago
10.1.8.13   01:33:df:ee:39:91     Port-Channel5     proxying     0:07:38 ago
132.23.23.1 00:11:aa:bb:32:ad     Ethernet1         up           never

switch>
```

- This command displays RAIL information for all servers on configured interfaces.

```
switch>show monitor server-failure servers all
Total servers monitored: 5

Server IP   Server MAC         Interface       State Last Failed
----------  -----------------  --------------  ---------  -----------
10.1.67.92  01:22:ab:cd:ee:ff  Ethernet17      inactive   7 days, 12:47:48 ago
44.11.11.7  ad:3e:5f:dd:64:cf  Ethernet23      down       0:06:14 ago
10.1.1.1    01:22:df:42:78:cd  Port-Channel6   up         4:38:01 ago
10.1.8.13   01:33:df:ee:39:91  Port-Channel5   proxying   0:10:31 ago
132.23.23.1 00:11:aa:bb:32:ad  Ethernet1       up         never

switch>
```

show monitor session

The **show monitor session** command displays the configuration of the specified port mirroring session. The command displays the configuration of all mirroring sessions on the switch when the session name parameter is omitted.

Command Mode

EXEC

Command Syntax

show monitor session *SESSION_NAME*

Parameters

- ***SESSION_NAME*** Port mirroring session identifier. Options include:
  - <no parameter> displays configuration for all sessions.
  - *label* command displays configuration of the specified session.

Example

- This command displays the mirroring configuration of the specified monitor session.

```
switch>show monitor session redirect_1

Session redirect_1
------------------------

Source Ports

  Both:        Et7

Destination Port: Et8
switch(config)>
```

show port-security

The **show port-security** command displays a summary of MAC address port security configuration and status on each interface where switchport port security is enabled.

Command Mode

EXEC

Command Syntax

```
show port-security
```

Display Values

Each column corresponds to one physical interface. The table displays interfaces with port security enabled.

- Secure Port: Interface with switchport port-security enabled.
- MaxSecureAddr: Maximum quantity of MAC addresses that the specified port can process.
- CurrentAddr: Static MAC addresses assigned to the interface.
- SecurityViolation: Number of frames with unsecured addresses received by port.
- Security Action: Action triggered by a security violation.

Examples

- This command displays switchport port security configuration and status data.

```
switch>show port-security
Secure Port      MaxSecureAddr  CurrentAddr  SecurityViolation  Security Action
                    (Count)        (Count)      (Count)
-------------------------------------------------------------------------------
     Et7              5             3            0            Shutdown
     Et10             1             0            0            Shutdown
-------------------------------------------------------------------------------
Total Addresses in System: 3
switch>
```

show port-security address

The **show port-security address** command display static unicast MAC addresses assigned to interfaces where switchport port security is enabled.

Command Mode

EXEC

Command Syntax

```
show port-security address
```

Example

- This command displays MAC addresses assigned to port-security protected interfaces.

```
switch>show port-security address
          Secure Mac Address Table
------------------------------------------------------------------
Vlan    Mac Address       Type                      Ports   Remaining Age
                                                              (mins)
----    -----------       ----                      -----   -------------
  10    164f.29ae.4e14    SecureConfigured          Et7     N/A
  10    164f.29ae.4f11    SecureConfigured          Et7     N/A
  10    164f.320a.3a11    SecureConfigured          Et7     N/A
------------------------------------------------------------------------
Total Mac Addresses for this criterion: 3
switch>
```

show port-security interface

The **show port-security interface** command displays the switchport port-security status of all specified interfaces.

Command Mode

EXEC

Command Syntax

show port-security interface [*INT_NAME*]

Parameters

- ***INT_NAME*** Interface type and numbers. Options include:
  - <no parameter> Display information for all interfaces.
  - **ethernet** *e_range* Ethernet interface range specified by *e_range*.
  - **loopback** *l_range* Loopback interface specified by *l_range*.
  - **management** *m_range* Management interface range specified by *m_range*.
  - **port-channel** *p_range* Port-Channel Interface range specified by *p_range*.
  - **vlan** *v_range* VLAN interface range specified by *v_range*.
  - **vxlan** *vx_range* VXLAN interface range specified by *vx_range*.

    Valid *range* formats include number, number range, or comma-delimited list of numbers and ranges.

Examples

- This command display port-security configuration and status for the specified interfaces.

```
switch>show port-security interface ethernet 7-8
Interface                  : Ethernet7
Port Security              : Enabled
Port Status                : Secure-down
Violation Mode             : Shutdown
Maximum MAC Addresses      : 5
Aging Time                 : 5 mins
Aging Type                 : Inactivity
SecureStatic Address Aging : Disabled
Total MAC Addresses        : 3
Configured MAC Addresses   : 3
Learn/Move/Age Events      : 5
Last Source Address:Vlan   : 164f.29ae.4e14:10
Last Address Change Time   : 0:39:47 ago
Security Violation Count   : 0

Interface                  : Ethernet8
Port Security              : Disabled
Port Status                : Secure-down
Violation Mode             : Shutdown
Maximum MAC Addresses      : 1
Aging Time                 : 5 mins
Aging Type                 : Inactivity
SecureStatic Address Aging : Disabled
switch>
```

show storm-control

The **show storm-control** command displays the storm-control level and interface inbound packet capacity for the specified interface.

The configured value (**storm-control**) differs from the programmed threshold in that the hardware accounts for Interframe Gaps (IFG) based on the minimum packet size. This command displays the broadcast or multicast rate after this adjustment.

Command Mode

Privileged EXEC

Command Syntax

```
show storm-control [INT_NAME]
```

Parameters

- <no parameter> Command returns data for all interfaces configured for storm control.
- ***INT_NAME*** interface type and port range. Settings include:
  - **ethernet** *e_range* Ethernet interfaces that *e_range* denotes.
  - **port-channel** *p_range* Port channel interfaces that *p_range* denotes.

  When storm control commands exist for a port-channel and an Ethernet port that is a member of the port channel, the command for the port-channel takes precedence.

  Valid *range* formats include number, number range, or comma-delimited list of numbers and ranges.

Example

- This command displays the storm control configuration for Ethernet ports 1 through 5.

```
switch#show storm-control
Port          Type  Level Rate(Mbps)   Status      Drops Reason
Et10/2         all     75       7500   active          0
Et10/3   multicast     55       5500   active          0
Et10/4   broadcast     50       5000   active          0
switch#
```

show switch forwarding-mode

The **show switch forwarding-mode** command displays the switch's current and available forwarding plane hardware modes.

Command Mode

EXEC

Command Syntax

```
show switch forwarding-mode
```

Related Commands

- **switch forwarding-mode** configures the switch's forwarding mode setting.

Example

- This command changes the switch's forward mode to ***store-and-forward***, then displays the forwarding mode.

```
switch(config)#switch forwarding-mode store-and-forward
switch(config)#show switch forwarding-mode
Current switching mode:    store and forward
Available switching modes: cut through, store and forward
```

show track

The **show track** command displays information about tracked objects configured on the switch.

Command Mode

EXEC

Command Syntax

show track [*OBJECT*] [*INFO_LEVEL*]

Parameters

- ***OBJECT*** tracked object for which information is displayed. Options include:
  - <no parameter> displays information for all tracked objects configured on the switch.
  - *object_name* displays information for the specified object.
- ***INFO_LEVEL*** amount of information that is displayed. Options include:
  - <no parameter> displays complete information including object status, number of status changes, time since last change, and client process tracking the object (if any).
  - **brief** displays brief list of all tracked objects and their current status.

Examples

- This command displays all information for tracked object ETH8.

```
switch#show track ETH8
Tracked object ETH8 is up
   Interface Ethernet8 line-protocol
      4 change, last change time was 0:36:12 ago
   Tracked by:
      Ethernet5/1 vrrp instance 50
switch#
```

- This command displays summary information for all tracked objects.

```
switch#show track brief
Tracked object ETH2 is up
Tracked object ETH4 is down
Tracked object ETH6 is up
Tracked object ETH8 is up
switch#
```

shutdown (server-failure configuration mode)

The **shutdown** command disables Rapid Automated Indication of Link-Loss (RAIL). By default, RAIL is disabled.

After entering server-failure configuration mode, a **no shutdown** command is required to enable RAIL.

The **no shutdown** command enables RAIL on the switch. The **shutdown** and **default shutdown** commands disable RAIL by removing the shutdown command from ***running-config***.

Command Mode

Server-failure Configuration

Command Syntax

```
shutdown
no shutdown
default shutdown
```

Examples

- This command enables RAIL on the switch.

```
switch(config)#monitor server
switch(config-server-failure)#no shutdown
switch(config-server-failure)#show active
monitor server-failure
   no shutdown
switch(config-server-failure)#
```

- This command disables RAIL on the switch.

```
switch(config-server-failure)#shutdown
switch(config-server-failure)#show active
monitor server-failure
switch(config-server-failure)#
```

storm-control

The **storm-control** command configures and enables storm control on the configuration mode physical interface. The command provides three mode options:

- **storm-control all** unicast, multicast, and broadcast inbound packet control.
- **storm-control broadcast** broadcast inbound packet control.
- **storm-control multicast** multicast inbound packet control.

An interface configuration can contain three storm-control statements, one with each mode setting. The **storm-control all** threshold overrides broadcast and multicast thresholds.

The threshold is a percentage of the available port bandwidth and is configurable on each interface for each transmission mode.

The **no storm-control** and **default storm-control** commands remove the corresponding **storm-control** statement from ***running-config***, disabling storm control for the specified transmission type on the configuration mode interface.

Command Mode

Interface-Ethernet Configuration
Interface-Port-Channel Configuration

Command Syntax

```
storm-control MODE level threshold
no storm-control mode
default storm-control mode
```

Parameters

- ***MODE*** packet transmission type. Options include:
  - **all**
  - **broadcast**
  - **multicast**
- *threshold* Inbound packet level that triggers storm control, as a percentage of port capacity. Value ranges from 0.01 to 100. Storm control is suppressed by a level of 100.

  The configured value differs from the programmed threshold in that the hardware accounts for Interframe Gaps (IFG) based on the minimum packet size. The **show storm-control** command displays the broadcast or multicast rate after this adjustment.

Restrictions

The **storm-control all** option is not available on Arad platform switches.

Example

- These commands enable multicast and broadcast storm control on Ethernet port 20 and sets thresholds of 65% (multicast) and 50% (broadcast). During each one second interval, the interface drops inbound multicast traffic and broadcast traffic in excess of the specified thresholds.

```
switch(config)#interface ethernet 20
switch(config-if-Et20)#storm-control multicast level 65
switch(config-if-Et20)#storm-control broadcast level 50
switch(config-if-Et20)#show active
interface Ethernet20
   storm-control broadcast level 50
   storm-control multicast level 65
switch(config-if-Et20)#
```

switch forwarding-mode

The **switch forwarding-mode** command specifies the mode of the switch's forwarding plane hardware. The default forwarding mode is ***cut through***.

The **no switch forwarding-mode** and **default switch forwarding-mode** commands restore the default forwarding mode by removing the **switch forwarding-mode** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
switch forwarding-mode MODE_SETTING
no switch forwarding-mode
default switch forwarding-mode
```

Parameters

- ***MODE_SETTING*** Specifies the switch's forwarding plane hardware mode. Options include:
  - **cut-through** the switch begins forwarding frames before their reception is complete.
  - **store-and-forward** the switch accumulates entire packets before forwarding them.

Guidelines

The forwarding plane mode is ***store-and-forward*** on Petra and Arad platform switches.

Related Commands

- **show switch forwarding-mode** displays the current forwarding mode.

Examples

- This command changes the forwarding mode to ***store-and-forward***.

```
switch(config)#switch forwarding-mode store-and-forward
switch(config)#
```

switchport

The **switchport** command places the configuration mode interface in ***switched port*** (Layer 2) mode. Switched ports are configurable as members of one or more VLANs through other switchport commands. Switched ports ignore all IP level configuration commands, including IP address assignments.

The **no switchport** command places the configuration mode interface in ***routed port*** (Layer 3) mode. Routed ports are not members of any VLANs and do not switch or bridge packets. All IP level configuration commands, including IP address assignments, apply directly to the routed port interface.

By default, Ethernet and Port Channel interfaces are in switched port mode. The **default switchport** command also places the configuration mode interface in switched port mode by removing the corresponding **no switchport** command from ***running-config***.

These commands only toggle the interface between switched and routed modes. They have no effect on other configuration states.

Command Mode

Interface-Ethernet Configuration
Interface-Port Channel Configuration

Command Syntax

```
switchport
no switchport
default switchport
```

Guidelines

When an interface is configured as a routed port, the switch transparently allocates an internal VLAN whose only member is the routed interface. Internal VLANs are created in the range from 1006 to 4094. VLANs that are allocated internally for a routed interface cannot be directly created or configured. The **vlan internal allocation policy** command specifies the method that VLANs are allocated.

All IP-level configuration commands, except **autostate** and **ip virtual-router,** can be used to configure a routed interface. Any IP-level configuration changes made to a routed interface are maintained when the interface is toggled to switched port mode.

A LAG that is created with the **channel-group** command inherits the mode of the member port. A LAG created from a routed port becomes a routed LAG. IP-level configuration statements are not propagated to the LAG from its component members.

Examples

- These commands put Ethernet interface 5 in routed port mode.

```
switch(config)#interface ethernet 5
switch(config-if-Et5)#no switchport
switch(config-if-Et5)#
```

- These commands returns Ethernet interface 5 to switched port mode.

```
switch(config)#interface ethernet 5
switch(config-if-Et5)#switchport
switch(config-if-Et5)#
```

switchport default mode access

The **switchport default mode access** command places the configuration mode interface in ***switched port default access*** (Layer 3) mode. Switched ports are configurable as members of one or more VLANs through other switchport commands. Switched ports ignore all IP level configuration commands, including IP address assignments.

Command Mode

Global Configuration

Command Syntax

```
switchport default mode access
```

Related Commands

- **switchport default mode routed** puts a switch with all ports in routed port mode.

Examples

- This command puts a switch with all ports in access port mode.

```
switch(config)#switchport default mode access
```

switchport default mode routed

The **switchport default mode routed** command places the configuration mode interface in ***switched port default routed*** (Layer 3) mode. Switched ports are configurable as members of one or more VLANs through other switchport commands. Switched ports ignore all IP level configuration commands, including IP address assignments.

By default, on a switch with default startup config or no config, all ports come up in access mode. By adding the CLI command **switchport default mode routed** to kickstart config, all ports will come up in routed mode after boot up. On boot up, Zero Touch Provisioning (ZTP) is enabled by default if the startup config (/mnt/flash/startupconfig) is deleted. ZTP can be disabled by setting DISABLE=True in ZTP config (/mnt/flash/zerotouchconfig). Kickstart config (/mnt/flash/kickstart-config) is used when startup config is missing and ZTP is disabled.

Command Mode

Global Configuration

Command Syntax

```
switchport default mode routed
```

Related Commands

- **switchport default mode access** puts a switch with all ports in access port mode.

Examples

- This command puts a switch with all ports in routed port mode.

```
switch(config)#switchport default mode routed
```

switchport mac address learning

The **switchport mac address learning** command enables MAC address learning for the configuration mode interface. MAC address learning is enabled by default on all Ethernet and port channel interfaces.

The switch maintains a MAC address table for switching frames between VLAN ports. When the switch receives a frame, it associates the MAC address of the transmitting interface with the recipient VLAN and port. When MAC address learning is enabled for the recipient port, the entry is added to the MAC address table. When MAC address learning is not enabled, the entry is not added to the table.

The **no switchport mac address learning** command disables MAC address learning for the configuration mode interface. The **switchport mac address learning** and **default switchport mac address learning** commands enable MAC address learning for the configuration mode interface by deleting the corresponding **no switchport mac address learning** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration
Interface-Port Channel Configuration

Command Syntax

```
switchport mac address learning
no switchport mac address learning
default switchport mac address learning
```

Example

- These commands disables MAC address learning for Ethernet interface 8, then displays the active configuration for the interface.

```
switch(config)#interface ethernet 8
switch(config-if-Et8)#no switchport mac address learning
switch(config-if-Et8)#show active
interface Ethernet8
   no switchport mac address learning
switch(config-if-Et8)#
```

switchport port-security

The **switchport port-security** command enables MAC address port security on the configuration mode interface. Ports with port security enabled restrict traffic to a limited number of hosts, as determined by their MAC addresses. The **switchport port-security maximum** command specifies the maximum number of MAC addresses. The **switchport port-security violation protect** command enables port security in protect mode.

The **no switchport port-security** and **default switchport port-security** commands disable port security on the configuration mode interface by removing the corresponding **switchport port-security** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration
Interface-Port Channel Configuration

Command Syntax

```
switchport port-security
no switchport port-security
default switchport port-security
```

Examples

- These commands enable port security on ethernet interface 7.

```
switch(config)#interface ethernet 7
switch(config-if-Et7)#switchport port-security
switch(config-if-Et7)#
```

switchport port-security maximum

The **switchport port-security maximum** command specifies the maximum MAC address limit for the configuration mode interface when configured as a secure port. When port security is enabled, the port accepts traffic and adds source addresses to the ARP table until the maximum is reached. Once the maximum is reached, if any traffic arrives from a source not already in the ARP table for the secure port, the port becomes errdisabled. The **switchport port-security** command configures an interface as a secure port.

The **no switchport port-security maximum** and **default switchport port-security maximum** commands restore the maximum MAC address limit of one on the configuration mode interface by removing the corresponding **switchport port-security maximum** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration
Interface-Port Channel Configuration

Command Syntax

```
switchport port-security maximum max_addr
no switchport port-security maximum
default switchport port-security maximum
```

Parameters

- *max_addr* maximum number of MAC addresses. Value ranges from 1 to 1000. Default value is 1.

Examples

- These commands configure a maximum of five incoming addresses for secure port channel interface 14.

```
switch(config)#interface port-channel 14
switch(config-if-Po14)#switchport port-security maximum 5
switch(config-if-Po14)#
```

switchport port-security violation protect

The **switchport port-security violation protect** command enables port security in protect mode (with the option of enabling logging). When port security is enabled, the port accepts traffic and adds source addresses to the ARP table until the maximum is reached. Once the maximum is reached, if any traffic arrives from a source not already in the ARP table for the secure port, the port becomes errdisabled. The **switchport port-security** command configures an interface as a secure port.

The **no switchport port-security** and **no switchport port-security violation protect log** commands disable port security protect mode and port security protect mode logging on the configuration mode interface.

Command Mode

Interface-Ethernet Configuration
Interface-Port Channel Configuration

Command Syntax

```
switchport port-security violation protect
switchport port-security violation protect log
no switchport port-security
no switchport port-security violation protect log
```

Examples

- These commands configure port security violation protect mode for secure port channel interface 14.

```
switch(config)#interface port-channel 14
switch(config-if-Po14)#switchport port-security violation protect
switch(config-if-Po14)#
```

- These commands configure port security violation protect logging mode for secure port channel interface 14.

```
switch(config)#interface port-channel 14
switch(config-if-Po14)#switchport port-security violation protect log
switch(config-if-Po14)#
```

track

The **track** command creates an object whose state changes to provide information to a client process. The client process must be separately configured for object tracking to have an effect on the switch.

The **no track** and **default track** commands remove the specified tracked object by removing the corresponding **track** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
track object_name interface INTERFACE_NAME PROPERTY
no track object_name
default track object_name
```

Parameters

- *object_name* User-created name for the tracked object.
- ***INTERFACE_NAME*** Interface associated with the tracked object. Options include:
  - **ethernet** *e_num* Ethernet interface specified by *e_num*.
  - **loopback** *l_num* Loopback interface specified by *l_num*.
  - **management** *m_num* Management interface specified by *m_num*.
  - **port-channel** *p_num* Port-channel interface specified by *p_num*.
  - **vlan** *v_num* VLAN interface specified by *v_num*.
  - **vxlan** *vx_num* VXLAN interface specified by *vx_num*.
- ***PROPERTY*** Tracked property. Options include:
  - **line-protocol** Object changes when the state of the associated interface changes.

Example

- This command creates a tracked object which tracks the state of the line protocol on Ethernet interface 8.

```
switch(config)#track ETH8 interface ethernet 8 line-protocol
switch(config)#
```

Chapter 17

Tap Aggregation

This chapter describes tap aggregation and the data structures that it requires. Sections in this chapter include:

- Section 17.1: Tap Aggregation Introduction
- Section 17.2: Tap Aggregation Description
- Section 17.3: Tap Aggregation Configuration
- Section 17.4: Tap Aggregation Traffic Steering
- Section 17.5: Tap Aggregation GUI
- Section 17.6: Keyframe and Timestamp Configuration
- Section 17.7: Tap Aggregation Command Descriptions

Port mirroring is described in Port Mirroring.

17.1 Tap Aggregation Introduction

Ethernet based switches are commonly deployed in dedicated networks to support tap and mirror port traffic towards one or more analysis applications. Ports configured to mirror data can simultaneously switch traffic to its primary destination while directing a copy of that traffic to analysis or test devices. Tap ports are typically part of a dedicated environment that allows for the aggregation of data streams from multiple sources that can be directed to multiple destinations.

Arista switches support port mirroring and tap aggregation and the data structures required by these functions.

17.2 Tap Aggregation Description

These sections describe tap aggregation, timestamps, and keyframes:

- Section 17.2.1: Tap Aggregation
- Section 17.2.2: Timestamps and Keyframes

17.2.1 Tap Aggregation

Tap aggregation is the accumulation of data streams and subsequent dispersal of these streams to devices and applications that analyze, test, verify, parse, detect, or store data. Tap aggregation requires an environment free from switching operations. Arista switches operate in one of two device modes:

- Switching mode: The switch performs normal switching and routing operations. Data mirroring is supported in switching mode. Tap aggregation is not available in switching mode.
- Tap aggregation mode: The switch is a data monitoring device and does not provide normal switching and routing services. Data mirroring is not available in tap aggregation mode.

  Access control lists, port channels, LAGs, QoS, and VLANs function normally in both modes.

Ethernet and port channel interfaces are configured as ***tap*** and ***tool*** ports to support tap aggregation.

- Tap ports: A tap port is an interface that receives a data stream that two network ports exchange.

  Tap ports prohibit egress traffic. MAC learning is disabled. All control plane interaction is prevented. Traps for inbound traffic are disabled. Tap ports are in STP forwarding mode.
- Tool ports: A tool port is an interface that replicates data streams received by one or more tap ports. Tool ports connect to devices that process the monitored data streams.

  Tool ports prohibit ingress traffic. MAC learning is disabled. All control plane interaction is prevented. Tool ports are in STP forwarding mode.

Tap and tool ports are configured with the **switchport mode** command. These ports are active when the switch is in tap aggregation mode and error-disabled when the switch is in switching mode.

Tap aggregation groups are data structures that map a set of tap ports to a set of tool ports. Both tap and tool ports may belong to multiple tap aggregation groups, and a tap aggregation group may contain multiple tap and tool ports.

Tap and tool ports are designated through switchport mode commands and act similar to trunk ports, in that they can allow access to VLANs specified through allowed-VLAN lists. Tap ports also specify a native VLAN for handling untagged frames.

- Access, trunk, and dot1q-tunnel mode ports are active when the switch is in switching mode and error-disabled when the switch is in tap aggregation mode.
- Tap and tool mode ports are active when the switch is in tap aggregation mode and error-disabled when the switch is in switching mode.

17.2.2 Timestamps and Keyframes

FM6000 platform switches support packet timestamping of packets sent from any port at line rate. Timestamps are used to correlate network events and in performance analysis. Keyframes provide information to assist in the interpretation of timestamps.

The switch contains two 64-bit counters to maintain ASIC time and UTC time. ASIC time is based on an internal 350 MHz counter. UTC is absolute time that is maintained by a precision oscillator and synchronized through PTP.

Timestamps are derived from the least significant 31 bits of ASIC time. Based on the 350 MHz counter period and 31-bit resolution, timestamp values repeat every 6.135 seconds.

Keyframes are periodically inserted into the data stream to provide context for interpreting timestamps. Keyframes contain the 64-bit value of the ASIC time counter, the corresponding 64-bit value of the UTC time counter, and the elapsed time since the last PTP synchronization of the UTC counter. Inserting one keyframe every second into the data stream assures that the timestamp value in each egress packet can be associated with values of the complete 64-bit ASIC time counter and the corresponding UTC counter.

17.2.2.1 Timestamps

Timestamps are based on a frame's ingress time and applied to frames sent on egress ports, ensuring that timestamps on monitored traffic reflect ingress timing of the original frames. Timestamping is configured on the egress port where the timestamp is applied to the frame.

A timestamp consists of the least significant 31 bits of the ASIC time counter. The most significant bit of the least significant byte is a 0 pad, resulting in a 32 bit timestamp with 31 bits of data. The keyframe mechanism provides recovery of the most significant 33 bits of the ASIC counters and a map to UTC time. Applications use this mechanism to determine the absolute time of the frame timestamp.

The switch supports three timestamp modes, which are configurable on individual Ethernet ports. The modes differ in the management of the egress frame's 32-bit frame check sequence (FCS):

- Disabled: Timestamping is disabled.
- FCS Replacement Mode: The original FCS is discarded, the ingress timestamp is appended to frame data, followed by a new FCS that is based on the appended timestamp. The result is a valid Ethernet frame, but the headers of all nested protocols are not updated to reflect the timestamp.
- FCS Appending Mode: The original FCS is discarded and replaced by the ingress timestamp. The size of the original frame is maintained without any latency impact, but the FCS is not valid.

17.2.2.2 Keyframes

Keyframes contain routable IP packets that provide information to relate timestamps with the complete ASIC counter and absolute UTC time. Keyframes have valid L2 and L3 headers. Keyframes contain these header fields:

- MAC fields (12 bytes):
  - Source MAC address is the address of the egress interface transmitting the keyframe.
  - Destination MAC address is configured through a CLI command.
- IP Header (20 bytes):
  - Source IP address is configured through CLI; default is management interface IP address.
  - Destination IP address is configured through a CLI command.
  - TTL is set to 64.
  - TOS is set to 0.
  - Protocol field is set to 253.
  - IP header's ID field is set to 0.

Keyframes contain these payload fields:

- ASIC time: (64 bits) ASIC time counter. (2.857 ns resolution).
- UTC time:(64 bits) Unix time that corresponds to ASIC time (ns).
- Last sync time: (64 bits) ASIC time of most recent PTP synchronization.

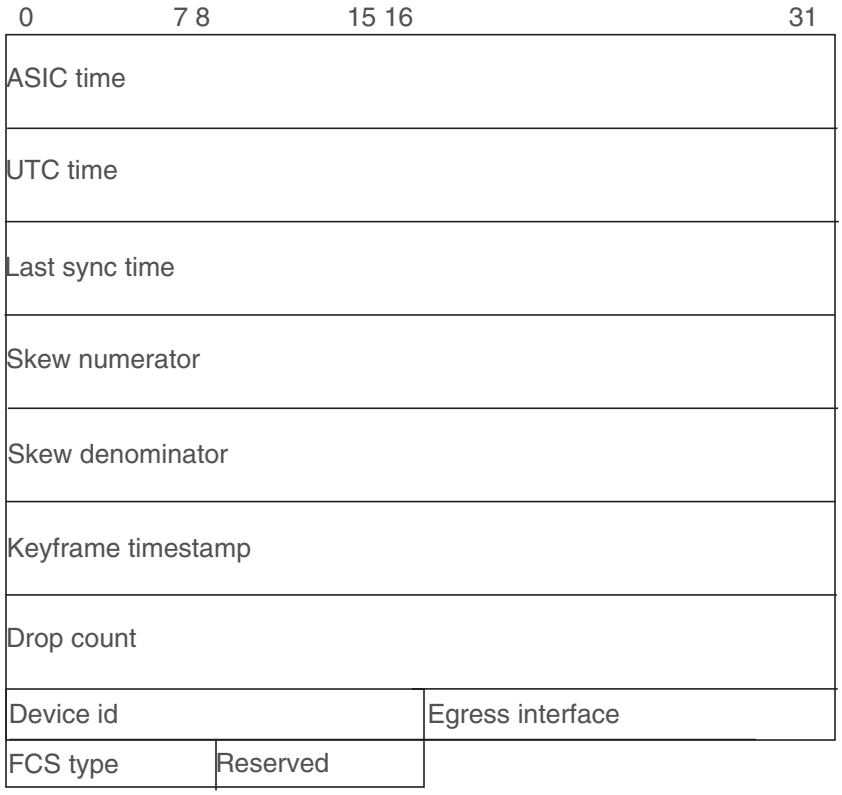
- Keyframe time: (64 bits) ASIC time of the keyframe’s egress (ns).
- Egress interface drops: (64 bits) Number of dropped frames on keyframe’s egress interface.
- Device ID: (16 bits) device ID (user defined).
- Egress interface: (16 bits) Keyframe's egress switchport.
- FCS type (8 bits): Timestamping mode configured on keyframe’s egress port.
  - 0: timestamping disabled.
  - 1: timestamp is appended to payload; new FCS is added to the frame.
  - 2: timestamp overwrites the existing FCS.
- Reserved (8 bits): Reserved for future use.
- Skew numerator/skew denominator: Form a ratio indicating the ASIC clock skew. If the ratio is greater than 1, the clock is skewed fast; if the ratio is less than 1, the clock is skewed slow.

  Last sync time equals 0 when there was no previous synchronization or the time since the last synchronization is greater than 8 hours.

  The 31-bit frame timestamp provides high-resolution timing, rolling over about every 6.135 seconds (31 bits at 2.857ns per tick). To obtain the full ASIC time and to correlate the timestamp to an absolute UTC time, the switch sends keyframes. Each keyframe contains the current ASIC time and UTC time; hence an application can compute the high order bits of the ASIC time (for precise, relative timing) from the ASIC to UTC time mapping, and then determine absolute time.

  ASIC to UTC time conversion is not quite immediate, so the UTC time in the frame will not be the ‘current’ time. A keyframe timestamp is provided for this purpose. The frame also includes the timestamping mode (FCS type) so applications can dynamically determine the timestamp’s byte offset. Each field is described in the following table.

Figure 17-1: Key frame payload



17.3 Tap Aggregation Configuration

These sections describe tap aggregation configuration tasks:

- Section 17.3.1: Enabling Tap Aggregation Mode
- Section 17.3.2: Tap Port Configuration
- Section 17.3.3: Tool Port Configuration
- Section 17.3.4: Identity VLAN Tagging
- Section 17.3.5: Tap Aggregation Group Configuration

17.3.1 Enabling Tap Aggregation Mode

The switch supports switching mode and tap aggregation mode. In switching mode, normal switching and routing functions are supported while tap aggregation functions are disabled. In tap aggregation mode, tap aggregation functions are enabled while normal switching and routing functions are disabled. By default, the switch is in switching mode.

A port's switchport status depends on its switchport mode and the switch's tap aggregation mode.

- Tap aggregation mode enabled: Tap and tool ports are enabled. Switching ports are errdisabled.
- Tap aggregation mode disabled: Tap and tool ports are errdisabled. Switching ports are enabled.

To enable the switch to carry out **tap aggregation**, first enter tap aggregation configuration mode and then set the mode.

Example

- These commands enter tap-agg configuration mode, then place the switch in tap aggregation mode.

```
switch(config)#tap aggregation
switch(config-tap-agg)#mode exclusive
switch(config-tap-agg)#show active
tap aggregation
   mode exclusive
switch(config-tap-agg)#
```

To return the switch to switching mode, remove the mode command from ***running-config***.

Examples

- These commands enter tap-agg configuration mode, then place the switch in switching mode.

```
switch(config)#tap aggregation
switch(config-tap-agg)#no mode
switch(config-tap-agg)#show active
switch(config-tap-agg)#
```

- These commands enter switching mode and remove all tap-agg configuration mode statements.

```
switch(config)#no tap aggregation
switch(config)#
```

17.3.2 Tap Port Configuration

Tap ports function when the switch is in tap aggregation mode. Tap ports receive traffic for replication to one or more tool ports. In tap aggregation mode, tap ports are in STP forwarding state and prohibit egress traffic. MAC learning, control plane interaction and traps for inbound traffic are disabled.

Tap mode ports are configured through switchport mode commands. Tap mode command settings persist in ***running-config*** without taking effect when the switch is not in tap aggregation mode or the interface is not in tap aggregation mode.

This section describes the following tap port configuration steps.

- Configuring an interface as a Tap Mode Port
- Tap Port Allowed VLAN List Configuration
- Tap Port Native VLAN
- Tap Port Packet Truncation

Configuring an interface as a Tap Mode Port

Ethernet and port channel interfaces are configured as tap ports with the **switchport mode** command.

Example

- These commands configure ethernet interfaces 41 through 43 as tap mode ports.

```
switch(config)#interface ethernet 41-43
switch(config-if-Et41-43)#switchport mode tap
switch(config-if-Et41-43)#show interface ethernet 41-43 tap
Port      Configured     Status         Native   Id   Truncation Default
          Mode                          Vlan     Vlan            Group
-------------------------------------------------------------------------
Et41      tap            tap            1        1    0          ---
Et42      tap            tap            1        1    0          ---
Et43      tap            tap            1        1    0          ---
switch(config-if-Et41-43)#
```

Tap Port Allowed VLAN List Configuration

By default, tap mode interfaces handle tagged traffic for all VLANs. The **switchport tap allowed vlan** command creates or modifies the set of VLANs for which a tap port handles tagged traffic.

Example

- These commands create tap mode allowed VLAN lists for Ethernet interface 41 through 43.

```
switch(config)#interface ethernet 41
switch(config-if-Et41)#switchport tap allowed vlan 401-410
switch(config-if-Et41)#interface ethernet 42
switch(config-if-Et42)#switchport tap allowed vlan 411-420
switch(config-if-Et41)#interface ethernet 41-42
switch(config-if-Et41-42)#show active
interface Ethernet41
   switchport mode tap
   switchport tap allowed vlan 401-410
interface Ethernet42
   switchport mode tap
   switchport tap allowed vlan 411-420
switch(config-if-Et41-42)#
```

Tap Port Native VLAN

Tap mode Interfaces associate untagged frames with the tap mode native VLAN. The **switchport tap native vlan** command specifies the tap mode native VLAN for the configuration mode interface. The default tap mode native VLAN for all interfaces is VLAN 1.

Example

- These commands assign VLAN 400 as the tap mode native VLAN for Ethernet interface 41.

```
switch(config)#interface ethernet 41
switch(config-if-Et41)#switchport tap native vlan 400
switch(config-if-Et41)#show interface ethernet 41-43 tap
Port      Configured     Status         Native   Id   Truncation Default
          Mode                          Vlan     Vlan            Group
-------------------------------------------------------------------------
Et41      tap            tap            400      1    0          ---
Et42      tap            tap            1        1    0          ---
Et43      tap            tap            1        1    0          ---
switch(config-if-Et41)#
```

Tap Port Packet Truncation

Tap ports can be configured to truncate inbound packets. The **switchport tap truncation** command configures the configuration mode interface, as a tap port, to truncate inbound packets to the specified packet size. By default, tap ports do not truncate packets.

Example

- These commands configure ethernet interface 41 to truncate packets to 150 bytes.

```
switch(config)#interface ethernet 41
switch(config-if-Et41)#switchport tap truncation 150
switch(config-if-Et41)#show interface ethernet 41-43 tap
Port      Configured     Status         Native   Id   Truncation Default
          Mode                          Vlan     Vlan            Group
-------------------------------------------------------------------------
Et41      tap            tap            400      1    150        ---
Et42      tap            tap            1        1    0          ---
Et43      tap            tap            1        1    0          ---
switch(config-if-Et41)#
```

- These commands configure ethernet interface 41 to send complete packets for replication.

```
switch(config-if-Et41)#no switchport tap truncation
switch(config-if-Et41)#show interface ethernet 41 tap
Port      Configured     Status         Native   Id   Truncation Default
          Mode                          Vlan     Vlan            Group
-------------------------------------------------------------------------
Et41      tap            tap            400      1    0          ---
switch(config-if-Et41)#
```

17.3.3 Tool Port Configuration

Tool ports replicate traffic received by tap ports. Tool ports are mapped to the tap ports through tap aggregation groups. A tool port may belong to multiple aggregation groups and an aggregation group may contain multiple tool ports.

Tool ports function when the switch is in tap aggregation mode. In this switch mode, tool ports are in STP forwarding state and ingress traffic is prohibited. MAC learning, control plane interaction and traps for inbound traffic are disabled. All control plane interaction is prevented and L2 agents do not send PDUs to tool mode interfaces. When the switch is in switching mode, tool ports are error-disabled.

Tool mode ports are configured through switchport commands. Tool mode command settings persist in ***running-config*** without taking effect when the switch is not in tap aggregation mode or the interface is not in tap aggregation mode.

This section describes the following tool port configuration steps.

- Configuring an interface as a Tool Mode Port
- Tool Port Allowed VLAN List Configuration
- Tool Port Packet Truncation

Configuring an interface as a Tool Mode Port

Ethernet and port channel interfaces are configured as tool ports with the **switchport mode** command.

Example

- These commands configure port channel interfaces 101 through 103 as tool mode ports.

```
switch(config)#interface port-channel 101-103
switch(config-if-Po101-103)#switchport mode tool
switch(config-if-Po101-103)#show interface port-channel 101-103 tool
Port      Configured     Status         Allowed             Id   Timestamp
          Mode                          Vlans               Tag  Mode
-------------------------------------------------------------------------
Po101     tool           tool           All                 Off  ---
Po102     tool           tool           All                 Off  ---
Po103     tool           tool           All                 Off  ---
switch(config-if-Po101-103)#
```

Tool Port Allowed VLAN List Configuration

By default, tool mode interfaces handle tagged traffic for all VLANs. The **switchport tool allowed vlan** command creates or modifies the set of VLANs for which a tool port handles tagged traffic.

Example

- These commands create tool mode allowed VLAN lists for port channel interfaces 101 through 103.

```
switch(config)#interface port-channel 101-103
switch(config-if-Po101-103)#switchport tool allowed vlan 1010-1020
switch(config-if-Po101-103)#interface port-channel 101
switch(config-if-Po101)#switchport tool allowed vlan add 1001-1009
switch(config-if-Po103)#interface port-channel 102
switch(config-if-Po102)#switchport tool allowed vlan remove 1016-1020
switch(config-if-Po102)#interface port-channel 103
switch(config-if-Po103)#switchport tool allowed vlan add 1021-1030
switch(config-if-Po103)#show interface port-channel 101-103 tool
Port      Configured     Status         Allowed             Id   Timestamp
          Mode                          Vlans               Tag  Mode
-------------------------------------------------------------------------
Po101     tool           tool           1001-1020           Off  ---
Po102     tool           tool           1010-1015           Off  ---
Po103     tool           tool           1010-1030           Off  ---
switch(config-if-Po103)#
```

Tool Port Packet Truncation

Tool ports can be configured to truncate outbound packets. The **switchport tool truncation** command configures the configuration mode interface, as a tool port, to truncate outbound packets to 160 bytes. By default, tool ports do not truncate packets.

Tool port packet truncation is supported only on the 7150 series platform.

Examples

- These commands configure ethernet interface 41, as a tool port, to truncate packets on egress to 160 bytes.

```
switch(config)#interface ethernet 41
switch(config-if-Et41)#switchport mode tool
switch(config-if-Et41)#switchport tool truncation 160
switch(config-if-Et41)#
```

- These commands configure ethernet interface 41 to send complete packets.

```
switch(config-if-Et41)#no switchport tool truncation
switch(config-if-Et41)#
```

17.3.4 Identity VLAN Tagging

By default, tool port output packets are identical to the replicated packets they receive from the tap ports to which they are associated. Identity tagging modifies packets sent by tool ports by adding a dot1q VLAN tag that identifies the originating tap port. Each tap port is associated with an identity number. Tool ports that are configured to add an identity tag append the originating tap port's identity number in the outer layer (or s-VLAN) tag.

These procedures describe commands that support identity VLAN tagging

- Tap Port Identity Value Configuration
- Tool Port Identity Tag Configuration

Tap Port Identity Value Configuration

The **switchport tap identity** command configures the tap port identity value for the configuration mode interface. The default identity value for all tap ports is 1.

Example

- These commands configure 1042 as the identity value for Ethernet interface 42.

```
switch(config)#interface ethernet 42
switch(config-if-Et42)#switchport tap identity 1042
switch(config-if-Et42)#show interface ethernet 41-43 tap
Port      Configured     Status         Native   Id   Truncation Default
          Mode                          Vlan     Vlan            Group
-------------------------------------------------------------------------
Et41      tap            tap            400      1    0          ---
Et42      tap            tap            1        1042 0          ---
Et43      tap            tap            1        1    0          ---
switch(config-if-Et42)#
```

Tool Port Identity Tag Configuration

The **switchport tool identity** command configures the configuration mode interface to include a tier 1 VLAN tag (dot1q) to packets it transmits. The VLAN number on the dot1q tag is specified by identity value configured for the tap port that supplies the packets. By default, tool ports do not encapsulate packets with the tier 1 VLAN tag.

Example

- These commands configure port channel 102 to include the identity tag in packets it transmits.

```
switch(config)#interface port-channel 102
switch(config-if-Po102)#switchport tool identity dot1q
switch(config-if-Po102)#show interface port-channel 101-103 tool
Port      Configured     Status         Allowed             Id   Timestamp
          Mode                          Vlans               Tag  Mode
-------------------------------------------------------------------------
Po101     tool           tool           1001-1020           Off  ---
Po102     tool           tool           1010-1015           On   ---
Po103     tool           tool           1010-1030           Off  ---
switch(config-if-Po102)#
```

17.3.5 Tap Aggregation Group Configuration

Tap aggregation groups associate a set of tap ports with a set of tool ports. Tool port replicates packets it receives from tap ports that are in the aggregation groups to which it belongs. A tap port can be assigned to a maximum of one tap aggregation group. Tool ports may belong to multiple tap aggregation groups. Tap aggregation groups may contain multiple tap ports and multiple tool ports.

These procedures describe commands that configure tap aggregation groups:

- Assigning a Tool Port to Tap Aggregation Groups
- Assigning Tap Ports to a Tap Aggregation Group
- Viewing Tap Aggregation Groups Assignments
- Replication to LAGs in Tool Groups with Multiple Members

Assigning a Tool Port to Tap Aggregation Groups

Tool ports are assigned to tap aggregation group through the **switchport tool group** command. Each command either creates a list or alters the existing list of groups to which a tool port belongs.

Example

- These commands create the list of tap aggregation groups for port channel interface 101.

```
switch(config)#interface port-channel 101
switch(config-if-Po101)#switchport tool group set analyze1 analyze2 analyze3
switch(config-if-Po101)#show active
interface Port-Channel101
   switchport mode tool
   switchport tap identity 2101
   switchport tool allowed vlan 1001-1020
   switchport tap default group tag-9
   switchport tool group set analyze3 analyze1 analyze2
switch(config-if-Po101)#
```

- These commands remove analyze-1 from port channel 101's tap aggregation group list.

```
switch(config-if-Po101)#switchport tool group remove analyze1
switch(config-if-Po101)#show active
interface Port-Channel101
   switchport mode tool
   switchport tap identity 2101
   switchport tool allowed vlan 1001-1020
   switchport tap default group tag-9
   switchport tool group set analyze3 analyze2
switch(config-if-Po101)#
```

Assigning Tap Ports to a Tap Aggregation Group

Tap ports are assigned to a tap aggregation group through the **switchport tap default group** command. Multiple ports are added to a group by entering interface configuration mode for all interfaces.

Example

- These commands assign Ethernet interface 41 through 43 to tap aggregation groups analyze2 (41 and 42) and analyze3 (43).

```
switch(config)#interface ethernet 41-42
switch(config-if-Et41-42)#switchport tap default group analyze2
switch(config-if-Et41-42)#interface ethernet 43
switch(config-if-Et43)#switchport tap default group analyze3
switch(config-if-Et43)#show interface ethernet 41-43 tap
Port      Configured     Status         Native   Id   Truncation Default
          Mode                          Vlan     Vlan            Group
-------------------------------------------------------------------------
Et41      tap            tap            400      1    0          analyze2
Et42      tap            tap            1        1042 0          analyze2
Et43      tap            tap            1        1    0          analyze3
switch(config-if-Et43)#
```

Viewing Tap Aggregation Groups Assignments

Tap aggregation group membership is displayed by **show tap aggregation groups** . Options facilitate the display of individual groups or all configured groups. The command displays active tool and tap ports by default and provides an option to display configured ports that are not active.

Example

- This command displays the contents of all configured tap aggregation groups.

```
switch>show tap aggregation groups
Group Name                          Tool Members
------------------------------------------------------------
analyze2                            Po101, Po102
analyze3                            Po101, Po103

Group Name                          Tap Members
------------------------------------------------------------
analyze2                            Et41, Et42
analyze3                            Et43
switch>
```

Replication to LAGs in Tool Groups with Multiple Members

Replication to LAGs in tool groups with multiple members results in traffic load balanced out of all LAG members, regardless whether the LAG is the only member of a tool group.

- This commands changes the system-wide default multicast replication mode.

```
switch(config)#platform sand multicast replication default ingress
switch(config)#
```

17.4 Tap Aggregation Traffic Steering

Traffic steering is a tap aggregation process that uses class maps and policy maps to direct data streams at tool ports that are not otherwise associated to the ingress tap port. A policy map is a data structure that filters data streams upon which identity VLAN tagging or tap aggregation group assignment is implemented.

Tapagg class maps and policy maps are similar to QoS and control plane maps. However, policy maps and their components are not interchangeable among function types.

17.4.1 Tapagg Policies

A policy map filters data packets by using classes and match rules. Each class contains an eponymous class map and a traffic resolution command. Each match rule contains packet content descriptors and a traffic resolution parameter.

- A class map uses ACLs that identify packets that comprise a specified data stream
- Packet content descriptors specify packet field values that are compared to inbound packets.
- A traffic resolution command or parameter specifies data handling methods for filtered traffic.

Each data packet entering an entity to which a policy map is assigned is managed as defined by the traffic resolution command of the highest priority class or rule that matches the packet.

Class maps are user created and can be edited or deleted. They filter traffic with IPv4 ACLs and are listed in ***running-config***. Tapagg traffic resolution commands perform one of the following:

- specify a tap aggregation group to direct the packet
- specify a VLAN number for identity tagging the packet.

Tap Aggregation policy maps do not define an implicit deny statement. Packets that do not match a policy map class or rule are replicated and sent out tool ports specified by the default aggregation group assigned to the ingress tap port. If no default group is selected, these packets are dropped.

17.4.2 Configuring Tapagg Traffic Policies

Tapagg traffic policies are implemented by creating class maps and policy maps, then applying the policy maps to Ethernet and port channel interfaces.

Creating Class Maps

A class map is an ordered list of IPv4 access control lists (ACLs). Each ACL is assigned a sequence number that specifies its priority in the class map. Tapagg class maps utilize ACL permit rules to pass packets and deny rules to drop packets.

Class maps are created and modified in class-map configuration mode, which is entered through **class-map type tapagg**. The **match (class-map (tapagg))** command inserts a specified ACL into the class map, assigning it a sequence number that denotes its placement.

Class-map configuration mode is a group-change mode. Changes made in a group-change mode are saved by exiting the mode. The **show active** command displays the saved version of class map. The **exit** command returns the switch to global configuration mode and saves pending class map changes. The **abort** command returns the switch to global configuration mode and discards pending changes.

Examples

- This command creates a tapagg class map named ***t-class_1*** and places the switch in class-map configuration mode.

```
switch(config)#class-map type tapagg match-any t-class_1
switch(config-cmap-t-class_1)#
```

- These commands add two IPv4 ACLs (***tacl-1***, ***tacl-2***) to the ***t-class_1*** class map. The commands use the default method of assigning sequence numbers to the ACLs.

```
switch(config-cmap-t-class_1)#match ip access-group tacl-1
switch(config-cmap-t-class_1)#match ip access-group tacl-2
switch(config-cmap-t-class_1)#
```

- These commands exit class-map configuration mode, stores pending changes to ***running-config***, then displays the class map.

```
switch(config-cmap-t-class_1)#exit
switch(config)#class-map type tapagg match-any t-class_1
switch(config-cmap-t-class_1)#show active
 class-map type tapagg match-any t-class_1
      10 match ip access-group tacl-1
      20 match ip access-group tacl-2
switch(config-cmap-t-class_1)#
```

Creating Policy Maps

Policy maps are created and modified in policy-map configuration mode. A policy map is an ordered list of classes and match rules. Policy maps are edited by adding or removing map elements. Data packets are managed by commands of the highest priority class or rule that matches the packet.

Classes: Each class contains a class map, a set command, and a sequence number:

- The class map identifies a data stream by using an ordered list of ACLs. Class maps are configured in class-map (tapagg) configuration mode.
- The **Set** command specifies the replication method for filtered data packets, either through an associated aggregation group or identity VLAN tagging.
- **Sequence number** specifies the class's priority within the policy map. Lower sequence numbers denote higher priority.

Matching rules: Each rule contains a filter list, an action, and a sequence number:

- The filter list identifies a data stream by using a set of packet field values.
- The action, (***SET_VALUE*** parameter) specifies the replication method of filtered data packets, either through an associated aggregation group or identity VLAN tagging.
- **Sequence number** specifies the rule's priority within the policy map. Lower sequence numbers denote higher priority.

Policy-map and policy-map-class configuration modes are group-change modes. Changes are saved with the **exit** command or discarded with the **abort** command. The **show active** and **show pending** commands display the saved and modified policy map versions, respectively.

The **policy-map type tapagg** command enters policy-map configuration mode.

Example

- This command creates the tapagg policy map named ***t-policy_1*** and places the switch in policy-map configuration mode.

```
switch(config)#policy-map type tapagg t-policy_1
switch(config-pmap-t-policy_1)#
```

The **class (policy-map (tapagg))** command adds a class to the configuration mode policy map and places the switch in policy-map-class configuration mode for adding a traffic resolution command to the class. The **set (policy-map-class (tapagg))** command specifies the data replication method for traffic filtered by the associated class map in the configuration mode policy map. The **set** command specifies one of these replication actions for filtered data packets.

- specifies an aggregation group
- specifies a VLAN identity tag for replicated packets.
- specifies an aggregation group and a VLAN identity tag.

Example

- These commands add the ***t-class_1*** class map to the ***t-policy_1*** policy map, associate a set statement with class, then saves the policy map by exiting the modes. Packets filtered by the class map are identity tagged with VLAN 444 and replicated as specified by the ***t-grp*** aggregation group.

```
switch(config-pmap-t-policy_1)#class t-class_1
switch(config-pmap-c-t-policy_1-t-class_1)#set aggregation-group t-grp id-tag
444
switch(config-pmap-c-t-policy_1-t-class_1)#exit
switch(config-pmap-t-policy_1)#exit
switch(config)#policy-map type tapagg t-policy_1
switch(config-pmap-t-policy_1)#show active
 policy-map type tapagg t-policy_1
   10 class t-class_1
      set aggregation-group t-group id-tag 444
switch(config-pmap-t-policy_1)#
```

The **match (policy-map (tapagg))** command adds a match rule to the configuration mode tapagg policy map.

Example

- This command enters policy-map configuration mode for ***t-policy_1,*** then creates a match rule for the policy map that filters OSPF packets and replicates them as specified by ***t-grp*** tap aggregation group.

```
switch(config-pmap-t-policy_1)#match ip ospf any any set aggregation-group t-grp
switch(config-pmap-t-policy_1)#
```

Applying Policy Maps to an Interface

The **service-policy type tapagg (Interface mode)** command applies a specified policy map to the configuration mode interface.

Example

- These commands applies the ***t-policy_1*** policy map to Ethernet interface 17.

```
switch(config)#interface ethernet 17
switch(config-if-Et17)#service-policy type tapagg input t-policy_1
switch(config-if-Et17)#
```

17.5 Tap Aggregation GUI

The switch provides a graphical user interface (GUI) for creating and viewing a tap aggregation configuration and displaying LANZ traffic statistics.

All commands available on the GUI are accessible through the CLI. The tap aggregation configuration created through either the CLI or the GUI can be viewed and modified through either medium.

This section provides a brief description of the tap aggregation GUI.

17.5.1 Accessing the GUI

The URL for the tap-agg GUI is

https://<hostname>/apps/TapAgg/index.html

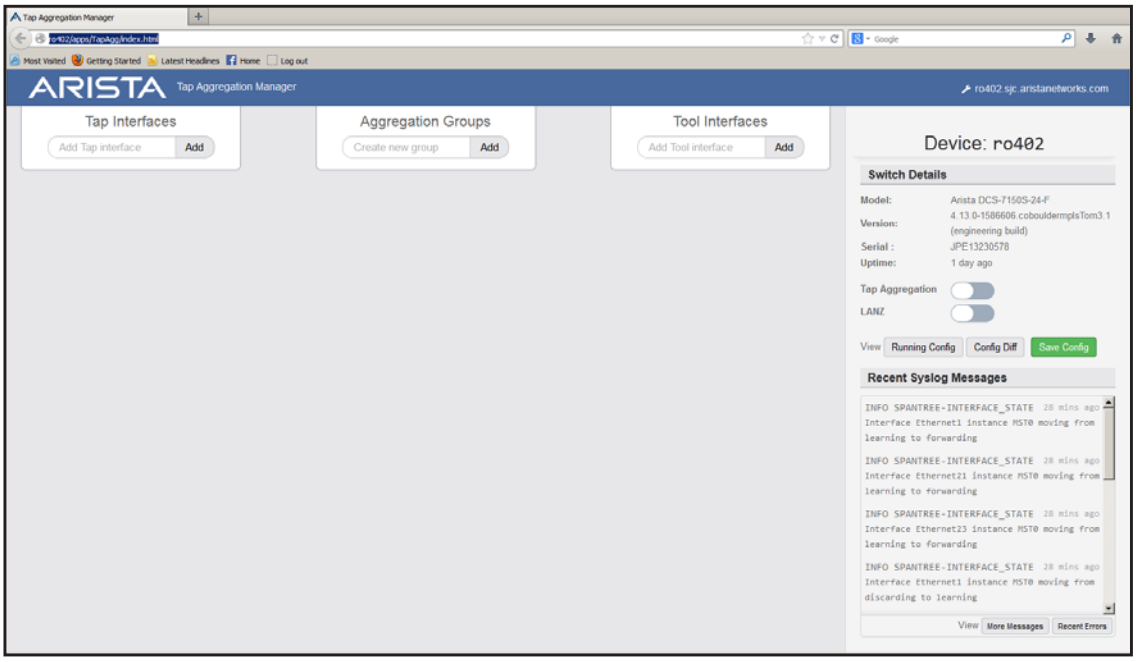
where <hostname> is the switch's configured hostname. Figure 17-2 displays the initial tap-agg panel for the switch with the hostname ***ro402***.

The tap-agg panel contains two sections:

- The configuration section displays the tap-agg configuration, including the tap interfaces, tool interfaces, and aggregation groups. Links are displayed to indicate interface group membership.
- The component section displays information and control buttons for the active configuration entity. When an entity is not selected, the section displays information for the switch (device).

The configuration section displays tap-agg components only when the switch is in tap aggregation mode. To enter tap aggregation mode, click the Tap Aggregation icon in the component section for the device. The icon is a toggle mechanism; clicking it again disables tag aggregation mode.

Figure 17-2: Tag-Agg GUI Initial Panel



17.5.2 Viewing Tap Aggregation Component Details

Figure 17-3 displays the tap-agg panel when the switch is in tap aggregation mode. The configuration section indicates that the tap aggregation configuration consists of three tool interfaces, one tap interface, and four aggregation groups. Ethernet 10 is the active component; configuration control and traffic information for this interface is available in the component section.

The active component is changed by clicking on the desired component in the configuration section. To display device (switch) information, click on any configuration section outside of any component.

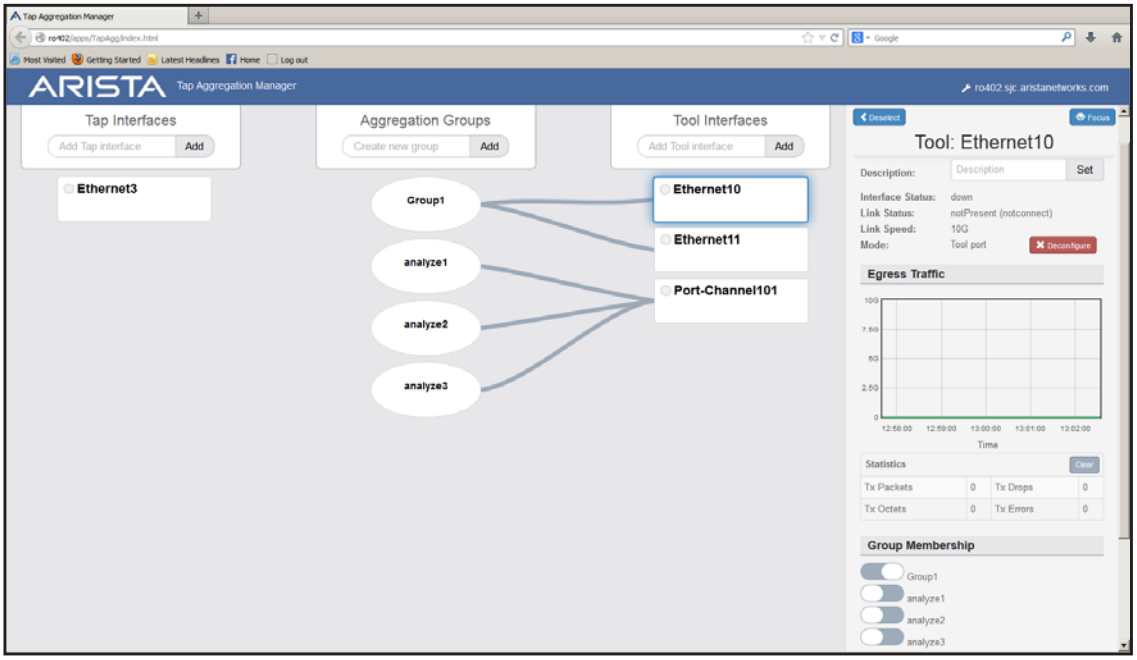
17.5.3 Modifying a Tap Aggregation Configuration

The tap aggregation configuration can be modified only when the switch is in tap-agg mode, (Section 17.5.1). The following is a partial list of configuration tasks that are available from the GUI:

- To add a tap or tool interface: Begin typing the interface name in the desired add interface data entry area to access a drop-down list of available interfaces. Select the desired interface and press the Add button.
- Removing an interface from the configuration: Select the desired interface in the configuration section and click the deconfigure button in that interface's component section.
- To add an aggregation group: Type the desired name of the new group in the data entry area and press the Add button.
- To add an interface to an aggregation group: Select the desired interface in the configuration section, then press the icon of the group in the group membership area of the interface's component section.

  Group icons are toggle buttons; clicking the icon of a group to which the interface belongs removes that interface from the group.

Figure 17-3: Tag-Agg GUI Panel – Tap aggregation mode enabled



17.6 Keyframe and Timestamp Configuration

17.6.1 Keyframe Generation

Keyframes contain routable IP packets that provide information to relate timestamps with the complete ASIC counter and absolute UTC time. The switch supports a maximum of ten keyframes, which are distinguished by their name label. Each keyframe can egress from every ethernet port.

Keyframe generation is enabled by the **platform fm6000 keyframe** command. Command options specify ports that transmit keyframes along with the destination MAC address and IP address in the keyframe's header. Other keyframe commands specify the transmission rate and the frame's source:

- The **platform fm6000 keyframe rate** command configures the keyframe's transmission rate.
- The **platform fm6000 keyframe source** command configures the source IP address that is placed in each keyframe's header. The management interface IP address is the default source address.

  The source MAC address is the MAC address of the egress interface transmitting the keyframe.
- The **platform fm6000 keyframe device** command configures the 16-bit number that keyframes list as the device ID in their payload.
- The **platform fm6000 keyframe fields skew** command enables the inclusion of clock skew fields in the keyframe.
- The **show platform fm6000 keyframe** command displays keyframe configuration information.

Examples

- This command enables the generation of a keyframe named key-1. This keyframe egresses from Ethernet interfaces 11 through 15, specifies a source IP address of 10.21.1.4 and a MAC address of 10.4E21.9F11.

  ```
  switch(config)#platform fm6000 keyframe key-1 interface ethernet 11-15 10.21.1.4
  10.4E21.9F11
  switch(config)#
  ```
- This command configures the generation rate for the keyframe of 10 frames per second on each of the five interfaces that it is configured to egress.

  ```
  switch(config)#platform fm6000 keyframe key-1 rate 10
  switch(config)#
  ```
- These commands enable the generation of a keyframe named key-1, then configures 100 as the value that is placed in the keyframe's device ID field.

  ```
  switch(config)#platform fm6000 keyframe key-1 device 100
  switch(config)#
  ```
- This command enables the inclusion of clock skew fields in the keyframe named "key-1".

  ```
  switch(config)#platform fm6000 keyframe key-1 fields skew
  switch(config)#
  ```

- This command displays key-1 configuration information.

```
switch(config)#show platform fm6000 keyframe

Keyframe key-1
------------------------
Egress Interface(s): Ethernet11, Ethernet12, Ethernet13, Ethernet14, Ethernet15
Source IP: 172.22.30.142
Destination IP: 10.21.1.4
Destination MAC: 00:10:4e:21:9f:11
Device ID: 100
Rate: 10 packet(s) per second

switch(config)#
```

17.6.2 Enabling Timestamp Insertion on an Interface

Timestamps are based on a frame's ingress time and applied to frames sent on egress ports, ensuring that timestamps on monitored traffic reflect ingress timing of the original frames. Time-stamping is configured on the egress port where the timestamp is applied to the frame.

When timestamping is enabled on an egress interface, packets leave the interface with timestamps that were applied in hardware upon arriving at the switch. This is facilitated by applying a hardware timestamp to all frames arriving on all interfaces when timestamping is enabled on any interface, then removing timestamps on packets egressing interfaces where timestamping is not enabled.

The **mac timestamp** command enables time-stamping on the configuration mode interface. The switch supports two timestamp modes, which differ in managing the egress frame's 32-bit frame check sequence (FCS):

- before-fcs: the switch discards the original FCS, appends the ingress timestamp at the end of the frame data, recalculates a new FCS based on the appended timestamp, then appends the new FCS to the end of the frame. This creates a valid Ethernet frame but does not update headers of any nested protocols.
- replace-fcs: the switch replaces the original FCS with the timestamp. This mode maintains the size of the original frame without any latency impact, but the FCS is not valid.

Examples

- These commands enable timestamping in before-fcs mode on Ethernet interface 44.

```
switch(config)#interface ethernet 44
switch(config-if-Et44)#mac timestamp before-fcs
switch(config-if-Et44)#show active
interface Ethernet44
   mac timestamp before-fcs
switch(config-if-Et44)#
```

- These commands disable timestamping on Ethernet interface 44.

```
switch(config-if-Et44)#no mac timestamp
switch(config-if-Et44)#show active
interface Ethernet44
switch(config-if-Et44)#
```

17.7 Tap Aggregation Command Descriptions

Global Configuration Commands

- platform fm6000 keyframe
- platform fm6000 keyframe device
- platform fm6000 keyframe fields skew
- platform fm6000 keyframe rate
- platform fm6000 keyframe source
- tap aggregation

Interface Configuration Commands

- mac timestamp
- switchport tap allowed vlan
- switchport tap default group
- switchport tap identity
- switchport tap native vlan
- switchport tap truncation
- switchport tool allowed vlan
- switchport tool group
- switchport tool identity

Tap Aggregation Configuration Mode

- mode (tap-agg configuration mode)
- mode exclusive no-errdisable (tap-agg configuration mode)

Tap Aggregation Traffic Steering

- class (policy-map (tapagg))
- class-map type tapagg
- match (class-map (tapagg))
- match (policy-map (tapagg))
- policy-map type tapagg
- resequence (class-map (tapagg))
- resequence (policy-map (tapagg))
- service-policy type tapagg (Interface mode)
- set (policy-map-class (tapagg))

Display Commands – EXEC Mode

- show interfaces tap
- show interfaces tool
- show platform fm6000 keyframe
- show tap aggregation groups

class (policy-map (tapagg))

The **class** command places the switch in policy-map-class (tapagg) configuration mode, which is a group change mode that defines a tapagg class by associating the class's eponymous class-map to a **set** statement. Upon exiting the policy-map-class mode, the class is placed in the policy-map as specified by an assigned sequence number.

A policy map is an ordered list of classes and match rules. Each class contains a class map, a set command, and a sequence number:

- The class map identifies a data stream by using an ordered list of ACLs. Class maps are configured in class-map (tapagg) configuration mode. Data packets are managed by commands of the highest priority class or rule that matches the packet.
- **Set** commands specify the replication method of filtered data packets, either through an associated aggregation group or identity VLAN tagging.
- **Sequence number** specifies the class's priority within the policy map. Lower sequence numbers denote higher priority.

The **exit** command returns the switch to policy-map configuration mode. However, saving policy-map-class changes also require an exit from policy-map mode. This saves all pending policy map and policy-map-class changes to ***running-config*** and returns the switch to global configuration mode. The **abort** command discards pending changes, and returns the switch to global configuration mode.

The **no class** and **default class** commands remove the class assignment from the configuration mode policy map by deleting the corresponding **class** configuration from ***running-config***.

Command Mode

Policy-Map (tapagg) Configuration
accessed through **policy-map type tapagg**

Command Syntax

```
[SEQ_NUM] class class_name
no [SEQ_NUM] class class_name
default [SEQ_NUM] class class_name
```

Parameters

- ***SEQ_NUM*** Priority of the class within the policy map. Lower numbers denote higher priority.
  - <no parameter> Number is derived by adding 10 to number of the map's last class or rule.
  - <***1*** to ***4294967295***> Number assigned to class.
- *class_name* name of the class.

Guidelines

When a class is not associated with a **set (policy-map-class (tapagg))** command, the filtered traffic is managed as specified by the tap port's default aggregation group.

Commands Available in Policy-map-class (tapagg) Configuration Mode

- **set (policy-map-class (tapagg))** assigns VLAN identity tag or tap aggregation group to class.
- **exit** returns the switch to parent policy map configuration mode.
- **abort** discards pending class map changes, then returns the switch to global configuration mode.

Related Commands

- **policy-map type tapagg** places the switch in policy-map (tapagg) configuration mode.

- **match (policy-map (tapagg))** assigns a match rule to a tapagg policy map.

Example

- These commands place the switch in policy-map-class and add the ***t-class_1*** class map to the ***t-policy_1*** policy map. Packets filtered by the class map are identity tagged with VLAN 444.

```
switch(config)#policy-map type tapagg t-policy_1
switch(config-pmap-t-policy_1)#class t-class_1
switch(config-pmap-c-t-policy_1-t-class_1)#set id-tag 444
switch(config-pmap-c-t-policy_1-t-class_1)#exit
switch(config-pmap-t-policy_1)#exit
switch(config)#policy-map type tapagg t-policy_1
switch(config-pmap-t-policy_1)#show active
 policy-map type tapagg t-policy_1
   10 class t-class_1
      set id-tag 444
switch(config-pmap-t-policy_1)#
```

class-map type tapagg

The **class-map type tapagg** command places the switch in class-map (tapagg) configuration mode, which is a group change mode that modifies a tapagg class map. A tapagg class map is a data structure that uses access control lists (ACLs) to define a data stream by specifying characteristics of data packets that comprise the stream. Tapagg policy maps use class maps to specify traffic that is managed by policy map criteria.

The **exit** command saves pending class map changes to ***running-config***, then returns the switch to global configuration mode. Class map changes are also saved by entering a different configuration mode. The **abort** command discards pending changes and returns the switch to global configuration mode.

The **no class-map type tapagg** and **default class-map type tapagg** commands delete the specified class map by removing the corresponding **class-map type qos** command and its associated configuration.

Command Mode

Global Configuration

Command Syntax

```
class-map type tapagg match-any class_name
no class-map type tapagg match-any class_name
default class-map type tapagg match-any class_name
```

Parameters

- *class_name* Name of class map.

Commands Available in Class-Map (tapagg) configuration mode

- **match (class-map (tapagg))**
- **resequence (class-map (tapagg))**

Related Commands

- **policy-map type tapagg**
- **class (policy-map (tapagg))**

Example

- This command creates a tapagg class map named ***t-class_1*** and places the switch in class-map configuration mode.

```
switch(config)#class-map type tapagg match-any t-class_1
switch(config-cmap-t-class_1)#
```

mac timestamp

The **mac timestamp** command enables timestamping on the configuration mode interface.

When timestamping is enabled on an egress interface, packets leave the interface with timestamps that were applied in hardware upon arriving at the switch. This is facilitated by applying a hardware timestamp to all frames arriving on all interfaces when timestamping is enabled on any interface, then removing timestamps on packets egressing interfaces where timestamping is not enabled.

The switch supports two timestamp modes, which differ in managing the egress frame's 32-bit frame check sequence (FCS):

- before-fcs: the switch discards the original FCS, appends the ingress timestamp at the end of the frame data, recalculates a new FCS based on the appended timestamp, then appends the new FCS to the end of the frame. This creates a valid Ethernet frame but does not update headers of any nested protocols.
- replace-fcs: the switch replaces the original FCS with the timestamp. This mode maintains the size of the original frame without any latency impact, but the FCS is not valid.

The **no mac timestamp** and **default mac timestamp** commands restore the default behavior of disabling timestamping on the configuration mode interface by removing the corresponding **mac timestamp** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration

Command Syntax

```
mac timestamp TS_PROPERTY
no mac timestamp
default mac timestamp
```

Parameters

- ***TS_PROPERTY*** Specifies the timestamp insertion mode. Options include:
  - **before-fcs** The ingress timestamp is appended to the frame and the FCS is recalculated.
  - **replace-fcs** The ingress timestamp replaces the original FCS.

Example

- These commands enable timestamping in before-fcs mode on Ethernet interface 44.

```
switch(config)#interface ethernet 44
switch(config-if-Et44)#mac timestamp before-fcs
switch(config-if-Et44)#show active
interface Ethernet44
   mac timestamp before-fcs
switch(config-if-Et44)#
```

- These commands disable timestamping on Ethernet interface 44.

```
switch(config-if-Et44)#no mac timestamp
switch(config-if-Et44)#show active
interface Ethernet44
switch(config-if-Et44)#
```

match (class-map (tapagg))

The **match** command adds an ACL to the configuration mode class map and associates a sequence number to the ACL. A class map is an ordered list of ACLs that define a data stream; the sequence number specifies an ACL's priority within the list. A class map is used by policy maps to filter data packets. Tapagg class maps utilize ACL permit rules to pass packets and deny rules to drop packets.

Class map (tapagg) configuration mode is a group change mode. **Match** statements are not saved to ***running-config*** until the edit session is completed by exiting the mode.

The **no match** and **default match** commands remove the specified **match** statement from the configuration mode class map by deleting the corresponding command from ***running-config***.

Command Mode

Class-map (tagagg) configuration
accessed through **class-map type tapagg** command

Command Syntax

```
[SEQ_NUM] match ip access-group list_name
no [SEQ_NUM] match ip access-group list_name
default [SEQ_NUM] match ip access-group list_name
```

Parameters

- ***SEQ_NUM*** Sequence number assigned to the ACL. Options include:
  - <no parameter> Number is derived by adding 10 to the number of the map's last ACL.
  - <***1*** to ***4294967295***> Number assigned to ACL.
- *list_name* name of ACL assigned to class map.

Guidelines

Match statements accept IPv4 ACLs.

Related Commands

- **class-map type tapagg** places the switch in Class-Map configuration mode.
- **exit** saves pending class map changes, then returns the switch to global configuration mode.
- **abort** discards pending class map changes, then returns the switch to global configuration mode.
- **class (policy-map (tapagg))** assigns a class map to a policy map.

Example

- These commands add two IPv4 ACLs (***tacl-1***, ***tacl-2***) to the ***t-class_1*** class map, saves the command by exiting class-map mode, then re-enters the mode to display the added ACLs.

```
switch(config)#class-map type tapagg match-any t-class_1
switch(config-cmap-t-class_1)#match ip access-group tacl-1
switch(config-cmap-t-class_1)#match ip access-group tacl-2
switch(config-cmap-t-class_1)#exit
switch(config)#class-map type tapagg match-any t-class_1
switch(config-cmap-t-class_1)#show active
 class-map type tapagg match-any t-class_1
      10 match ip access-group tacl-1
      20 match ip access-group tacl-2
switch(config-cmap-t-class_1)#
```

match (policy-map (tapagg))

The **match** command adds a rule to the configuration mode tapagg policy map. A policy map is an ordered list of classes and rules. Each rule contains a filter list, an action, and a sequence number:

- The filter list identifies a data stream through a set of packet field values.
- The action, (***SET_VALUE*** parameter) specifies the replication method of filtered data packets, either through an associated aggregation group or identity VLAN tagging.
- **Sequence number** specifies the rule's priority with the policy map.

The **no match** and **default match** commands remove the **match** rule from the configuration mode policy by deleting the corresponding statement from ***running-config***.

Command Mode

Policy-Map (tapagg) Configuration
accessed through **policy-map type tapagg**

Command Syntax

[*SEQ_NUM*] match [*VLAN_TAG*] ip *SOURCE_ADDR* [*SOURCE_PORT*] *DEST_ADDR* [*DEST_PORT*] [*PROTOCOL*][*FLAGS*][*MESSAGE*][fragments][tracked][*DSCP_FILTER*][*TTL_FILTER*][log] *SET_VALUE*

no match [*VLAN_TAG*] ip *SOURCE_ADDR* [*SOURCE_PORT*] *DEST_ADDR* [*DEST_PORT*] [*PROTOCOL*][*FLAGS*][*MESSAGE*][fragments][tracked][*DSCP_FILTER*][*TTL_FILTER*][log] *SET_VALUE*

no *SEQ_NUM*

default match [*VLAN_TAG*] ip *SOURCE_ADDR* [*SOURCE_PORT*] *DEST_ADDR* [*DEST_PORT*] [*PROTOCOL*][*FLAGS*][*MESSAGE*][fragments][tracked][*DSCP_FILTER*][*TTL_FILTER*][log] *SET_VALUE*

default *SEQ_NUM*

Commands use a subset of the listed fields. Available parameters depend on specified protocol. Use CLI syntax assistance to view options for specific protocols when creating a permit rule.

Parameters

- ***SEQ_NUM*** Priority of the rule within the policy map. Lower numbers denote higher priority.
  - <no parameter> Number derived by adding 10 to number of the map's last class or rule.
  - <***1*** to ***4294967295***> Number assigned to class.
- ***VLAN_TAG*** VLAN field filter. Options include:
  - <no parameter> packets are not filtered by VLAN field.
  - **vlan** <***1*** to ***4094***> <***0*** to ***4095***> VLAN ID and mask.
  - **vlan inner** <***1*** to ***4094***> <***0*** to ***4095***> VLAN ID and mask.
  - **vlan** <***1*** to ***4094***> <***0*** to ***4095***> **inner** <***1*** to ***4094***> <***0*** to ***4095***> VLAN ID and mask.
- ***PROTOCOL*** protocol field filter. Values include:
  - <no parameter> packets are not filtered by host name.
  - **ahp** authentication header protocol (51).
  - **icmp** internet control message protocol (1).
  - **igmp** internet group management protocol (2).

- **ip** internet protocol – IPv4 (4).
- **ospf** open shortest path first (89).
- **pim** protocol independent multicast (103).
- **tcp** transmission control protocol (6).
- **udp** user datagram protocol (17).
- **vrrp** virtual router redundancy protocol (112).
- *protocol_num* integer corresponding to an IP protocol. Values range from 0 to 255.

- ***SOURCE_ADDR*** and ***DEST_ADDR*** source and destination address filters. Options include:
  - *network_addr* subnet address (CIDR or address-mask).
  - **any** Packets from all addresses are filtered.
  - **host** *ip_addr* IP address (dotted decimal notation).

  Source and destination subnet addresses support discontiguous masks.
- ***SOURCE_PORT*** and ***DEST_PORT*** source and destination port filters. Options include:
  - **any** all ports
  - **eq** *port-1 port-2* ... *port-n* A list of ports. Maximum list size is 10 ports.
  - **neq** *port-1 port-2* ... *port-n* The set of all ports not listed. Maximum list size is 10 ports.
  - **gt** *port* The set of ports with larger numbers than the listed port.
  - **lt** *port* The set of ports with smaller numbers than the listed port.
  - **range** *port_1 port_2* The set of ports whose numbers are between the range.
- **fragments** filters packets with FO bit set (indicates a non-initial fragment packet).
- ***FLAGS*** flag bit filters (TCP packets). Use CLI syntax assistance (?) to display options.
- ***MESSAGE*** message type filters (ICMP packets). Use CLI syntax assistance (?) to display options.
- **tracked** rule filters packets in existing ICMP, UDP, or TCP connections.
  - Valid in ACLs applied to the control plane.
  - Validity in ACLs applied to data plane varies by switch platform.
- ***DSCP_FILTER*** rule filters packet by its DSCP value. Values include:
  - <no parameter> Rule does not use DSCP to filter packets.
  - **dscp** *dscp_value* Packets match if DSCP field in packet is equal to *dscp_value*.
- ***TTL_FILTER*** rule filters packet by its TTL (time-to-live) value. Values include:
  - <no parameter> Rule does not use TTL field to filter packets.
  - **ttl eq** *ttl_value* Packets match if ***ttl*** in packet is equal to *ttl_value*.
  - **ttl gt** *ttl_value* Packets match if ***ttl*** in packet is greater than *ttl_value*.
  - **ttl lt** *ttl_value* Packets match if ***ttl*** in packet is less than *ttl_value*.
  - **ttl neq** *ttl_value* Packets match if ***ttl*** in packet is not equal to *ttl_value*.
- **log** triggers an informational log message to the console about the matching packet.
  - Valid in ACLs applied to the control plane.
  - Validity in ACLs applied to data plane varies by switch platform.
- ***SET_VALUE*** specifies the replication method for filtered packets.
  - **set aggregation group** *agg_group* Replication specified by aggregation group.

- **set id-tag <*1* to *4094*>** Packet is identity tagged with specified VLAN number.
- **set aggregation group** *agg_group* **id-tag <*1* to *4094*>** Assigns agg group and identity tag.

Related Commands

- **policy-map type tapagg** places the switch in policy-map (tapagg) configuration mode.
- **class (policy-map (tapagg))** assigns a class to the configuration mode policy-map.

Example

- This command creates a match rule for the ***t-policy_1*** policy map that filters OSPF packets and replicates them as specified by the ***t-group*** tap aggregation group.

```
switch(config)#policy-map type tapagg t-policy_1
switch(config-pmap-t-policy_1)#match ip ospf any any set aggregation-group
t-group
switch(config-pmap-t-policy_1)#exit
switch(config)#policy-map type tapagg t-policy_1
switch(config-pmap-t-policy_1)#show active
 policy-map type tapagg t-policy_1
   10 match ip ospf any any set aggregation-group t-group
switch(config-pmap-t-policy_1)#
```

mode (tap-agg configuration mode)

The **mode** command configures the switch's tap aggregation mode. The switch supports these **mode** command options:

- ***exclusive***: Tap aggregation mode is enabled. Switching mode is disabled.

  Tap and tool ports are enabled. Switching ports are errdisabled.

- <not configured>: Tap aggregation mode is disabled. Switching mode is enabled.

  Tap and tool ports are errdisabled. Switching mode ports are enabled.

The default setting enables switching mode and disables tap aggregation mode.

The **no mode** and **default mode** commands disables tap aggregation mode and enables switching mode by removing the mode command from ***running-config***.

Command Mode

Tap-agg Configuration

Command Syntax

```
mode TAP_MODE
no mode TAP_MODE
default mode TAP_MODE
```

Parameters

- ***TAP_MODE*** specifies the switch's switch's tap aggregation mode.
  - **exclusive** tap aggregation is enabled.

Related Commands

- **tap aggregation** places the switch in tap-aggregation configuration mode.

Example

- These commands place the switch in tap-agg configuration mode and enable tap aggregation mode.

```
switch(config)#tap aggregation
switch(config-tap-agg)#mode exclusive
switch(config-tap-agg)#show active
tap aggregation
   mode exclusive
switch(config-tap-agg)#
```

- These commands disable tap aggregation mode by removing the mode command from ***running-config***.

```
switch(config-tap-agg)#no mode
switch(config-tap-agg)#show active
switch(config-tap-agg)#
```

mode exclusive no-errdisable (tap-agg configuration mode)

The **mode exclusive no-errdisable** command configures the specified interface to remain enabled regardless of its switchport mode, when tap aggregation is enabled. This command is used primarily to configure a port to support PTP functions while the switch operates as a tap aggregator.

Each command configures one Ethernet or port channel interface. Subsequent mode exclusive no-errdisable commands add to the list of ports that remain enabled when tap aggregation is enabled.

The **no mode exclusive no-errdisable** and **default mode exclusive no-errdisable** commands configure the specified interface to be errdisabled when programmed in access, trunk, or dot1q-tunnel switching mode when tap aggregation is enabled by removing the corresponding **mode exclusive no-errdisable** command from ***running-config***.

Command Mode

Tap-agg Configuration

Command Syntax

```
mode exclusive no-errdisable INT_NAME
no mode exclusive no-errdisable INT_NAME
default mode exclusive no-errdisable INT_NAME
```

Parameters

- ***INT_NAME*** Interface type and numbers. Options include:
  - **ethernet** *e_num* Ethernet interface specified by *e_num*.
  - **port-channel** *p_num* Port-Channel Interface specified by *p_num*.

Related Commands

- **tap aggregation** places the switch in tap-aggregation configuration mode.
- **mode (tap-agg configuration mode)** configures the switch's tap-aggregation mode.

Example

- These commands places the switch in tap-agg configuration mode and places Ethernet interface 21/3 in no-errdisable mode.

```
switch(config)#tap aggregation
switch(config-tap-agg)#mode exclusive
switch(config-tap-agg)#mode exclusive no-errdisable ethernet 21/4
switch(config-tap-agg)#
```

platform fm6000 keyframe

The **platform fm6000 keyframe** command enables keyframe generation for data streams transmitted from specified ethernet interfaces. Keyframes are routable IP packets that the switch inserts into a data stream to provide contextual information that correlate timestamps inserted into data packets with absolute UTC time and the switch's complete ASIC time counter.

The switch supports a maximum of ten keyframes. The keyframe name is the label that distinguishes different keyframes. Each keyframe can egress from every ethernet port. Command options specify the destination MAC address and IP address in the keyframe's header. Other keyframe commands specify the transmission rate and the frame's source.

The **no platform fm6000 keyframe and default platform fm6000 keyframe** commands disable generation of the specified keyframe by deleting the corresponding platform fm6000 keyframe command from ***running-config***. These command also remove all supporting **platform fm6000 keyf**rame commands for the specified keyframe.

Command Mode

Global Configuration

Command Syntax

```
platform fm6000 keyframe kf_name interface ethernet e_range ipv4_addr mac_addr
no platform fm6000 keyframe kf_name
default platform fm6000 keyframe kf_name
```

Parameters

- *kf_name* The keyframe's name.
- *e_range* Ethernet interface range over which the keyframe egresses. Valid formats include number, range, or comma-delimited list of numbers and ranges.
- *ipv4_addr* Destination IPv4 address inserted into keyframes. (Dotted decimal notation)
- *mac_addr* Destination MAC address inserted into keyframes. (48-bit dotted hex notation).

Guidelines

Subsequent issuance of this command for a specified keyframe replaces the existing command in ***running-config***. Ethernet interfaces are inserted into an existing keyframe only by issuing the complete command that identifies all interfaces through which the keyframe is transmitted.

Examples

- This command enables the generation of a keyframe named key-1. This keyframe egresses from Ethernet interfaces 11 through 15, specifies a source IP address of 10.21.1.4 and a MAC address of 10.4E21.9F11.

```
switch(config)#platform fm6000 keyframe key-1 interface ethernet 11-15 10.21.1.4
10.4E21.9F11
switch(config)#
```

platform fm6000 keyframe device

The **platform fm6000 keyframe device** command configures the 16-bit number that the specified keyframe lists as the device ID in its payload. By default, the default device value placed in the specified keyframes is 0.

The **no platform fm6000 keyframe device** and **default platform fm6000 keyframe device** commands restore the default device ID insertion value of 0 for the specified keyframe by removing the corresponding **platform fm6000 keyframe device** command from running-config. The **no platform fm6000 keyframe and default platform fm6000 keyframe** commands also remove the corresponding **platform fm6000 keyframe device** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
platform fm6000 keyframe kf_name device device_id
no platform fm6000 keyframe kf_name device
default platform fm6000 keyframe kf_name device
```

Parameters

- *kf_name* Keyframe name.
- *device_id* Value inserted in keyframe's device ID field. Value ranges from 0 to 65535. Default is 0.

Examples

- These commands enable the generation of a keyframe named key-1, then configures 100 as the value that is placed in the keyframe's device ID field.

```
switch(config)#platform fm6000 keyframe key-1 interface ethernet 11-15 10.21.1.4
10.4E21.9F11
switch(config)#platform fm6000 keyframe key-1 device 100
switch(config)#
```

platform fm6000 keyframe fields skew

Keyframes may optionally include skew numerator and skew denominator fields. These skew fields form a ratio indicating the ASIC clock skew. If the ratio is greater than 1, the clock is skewed fast; if the ratio is less than 1, the clock is skewed slow. Clock skew fields are omitted by default.

The **platform fm6000 keyframe fields skew** command enables the inclusion of clock skew fields in the keyframe.

The **no platform fm6000 keyframe fields skew** and **default platform fm6000 keyframe fields skew** remove the clock skew fields from the keyframe.

Command Mode

Global Configuration

Command Syntax

```
platform fm6000 keyframe kf_name fields skew
no platform fm6000 keyframe kf_name fields skew
default platform fm6000 keyframe kf_name fields skew
```

Parameters

- *kf_name* Keyframe name.

Examples

- This command enables the inclusion of clock skew fields in the keyframe named “key-1”.

```
switch(config)#platform fm6000 keyframe key-1 fields skew
switch(config)#
```

platform fm6000 keyframe rate

The **platform fm6000 keyframe rate** command specifies the transmission rate for the specified keyframe from each interface from which it is configured to egress. By default, one keyframe is sent per second.

The **no platform fm6000 keyframe rate** and **default platform fm6000 keyframe rate** commands restore the default transmission rate for the specified keyframe of one per second by removing the corresponding **platform fm6000 keyframe rate** command from running-config. The **no platform fm6000 keyframe and default platform fm6000 keyframe** commands also remove the corresponding **platform fm6000 keyframe rate** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
platform fm6000 keyframe kf_name rate tx_rate
no platform fm6000 keyframe kf_name rate
default platform fm6000 keyframe kf_name rate
```

Parameters

- *kf_name* The keyframe's name.
- *tx_rate* Keyframe transmission rate (frames per second). Value ranges from 1 to 100. Default value is 1.

Examples

- These commands enable the generation of a keyframe named key-1, then configures the generation rate for the keyframe of 10 frames per second on each of the five interfaces that it is configured to egress.

```
switch(config)#platform fm6000 keyframe key-1 interface ethernet 11-15 10.21.1.4
10.4E21.9F11
switch(config)#platform fm6000 keyframe key-1 rate 10
switch(config)#
```

platform fm6000 keyframe source

The **platform fm6000 keyframe source** command configures the source IP address that the specified keyframe lists in its IP header. By default, keyframes use the IP address of the management interface as their source address.

The **no platform fm6000 keyframe source** and **default platform fm6000 keyframe source** commands restore the management interface IP address as the specified keyframe's source IP address by removing the corresponding **platform fm6000 keyframe source** command from running-config. The **no platform fm6000 keyframe and default platform fm6000 keyframe** commands also remove the corresponding **platform fm6000 keyframe source** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
platform fm6000 keyframe kf_name source ip ipv4_addr
no platform fm6000 keyframe kf_name source ip
default platform fm6000 keyframe kf_name source ip
```

Parameters

- *kf_name* Keyframe's name.
- *ipv4_addr* Keyframe's source IPv4 address. (Dotted decimal notation – A.B.C.D)

Examples

- These commands enable the generation of a keyframe named key-1, then configures the IP address of keyframes as 10.1.1.101.

```
switch(config)#platform fm6000 keyframe key-1 interface ethernet 11-15 10.21.1.4
10.4E21.9F11
switch(config)#platform fm6000 keyframe key-1 source 10.1.1.101
switch(config)#
```

policy-map type tapagg

The **policy-map type tapagg** command places the switch in policy-map (tapagg) configuration mode, which is a group change mode that modifies a tapagg policy map. A tapagg policy map is a data structure that consists of class maps and match statements that filter a specific data stream. Packets in that data stream are either managed as specified by a tap aggregation group or modified to add a VLAN identity tag. Policy maps manage traffic when applied to an Ethernet or port channel interface.

The **exit** command saves pending policy map changes to ***running-config*** and returns the switch to global configuration mode. Policy map changes are also saved by entering a different configuration mode. The **abort** command discards pending changes, returning the switch to global configuration mode.

The **no policy-map type tapagg** and **default policy-map type tapagg** commands delete the specified policy map by removing the corresponding **policy-map type tapagg** command and the associated policy map statements.

Command Mode

Global Configuration

Command Syntax

```
policy-map type tapagg map_name
no policy-map type tapagg map_name
default policy-map type tapagg map_name
```

Parameters

- *map_name* Name of policy map.

Commands Available in Policy-Map configuration mode

- **class (policy-map (tapagg))**
- **match (policy-map (tapagg))**

Related Commands

- **class-map type tapaggs**
- **service-policy type tapagg (Interface mode)**

Example

- This command creates the tapagg policy map named ***t-policy_1*** and places the switch in policy-map configuration mode.

```
switch(config)#policy-map type tapagg t-policy_1
switch(config-pmap-t-policy_1)#
```

resequence (class-map (tapagg))

The **resequence** command assigns sequence numbers to access control lists (ACLs) in the configuration mode tapagg class map. Sequence numbers denote an ACL's priority within the class map. Command parameters specify the number of the first ACL and the numeric interval between consecutive ACLs.

Maximum rule sequence number is 4294967295.

Command Mode

Class-map (tagagg) configuration
accessed through **class-map type tapagg** command

Command Syntax

resequence [*start_num* [*inc_num*]]

Parameters

- *start_num* sequence number assigned to the first rule. Default is 10.
- *inc_num* numeric interval between consecutive rules. Default is 10.

Example

- These commands display a policy map whose entities were entered with default sequence numbers, then renumber the contents.

```
switch(config-pmap-t-policy_1)#show active
 policy-map type tapagg t-policy_1
   10 match ip ospf any any set aggregation-group t-group
   20 class fred
      set aggregation-group t-group id-tag 444
   30 class t-class_2
      set id-tag 500
   40 class t-class_3
      set id-tag 600
   50 class t-class_4
      set id-tag 700
switch(config-pmap-t-policy_1)#resequence 100 20
switch(config-pmap-t-policy_1)#exit
switch(config)#policy-map type tapagg t-policy_1
switch(config-pmap-t-policy_1)#show active
 policy-map type tapagg t-policy_1
   100 match ip ospf any any set aggregation-group t-group
   120 class fred
      set aggregation-group t-group id-tag 444
   140 class t-class_2
      set id-tag 500
   160 class t-class_3
      set id-tag 600
   180 class t-class_4
      set id-tag 700
switch(config-pmap-t-policy_1)#
```

resequence (policy-map (tapagg))

The **resequence** command assigns sequence numbers to classes and rules in the configuration mode tapagg policy map. Sequence numbers denote a class or rule's priority within the policy map. Command parameters specify the number of the first policy map entity and the numeric interval between consecutive entities.

Maximum rule sequence number is 4294967295.

Command Mode

Policy-Map (tapagg) Configuration
accessed through **policy-map type tapagg** command

Command Syntax

resequence [*start_num* [*inc_num*]]

Parameters

- *start_num* sequence number assigned to the first rule. Default is 10.
- *inc_num* numeric interval between consecutive rules. Default is 10.

Example

- These commands display a policy map whose entities were entered with default sequence numbers, then use the **resequence** command to renumber the contents.

```
switch(config-pmap-t-policy_1)#show active
 policy-map type tapagg t-policy_1
   10 match ip ospf any any set aggregation-group t-group
   20 class fred
      set aggregation-group t-group id-tag 444
   30 class t-class_2
      set id-tag 500
   40 class t-class_3
      set id-tag 600
   50 class t-class_4
      set id-tag 700
switch(config-pmap-t-policy_1)#resequence 100 20
switch(config-pmap-t-policy_1)#exit
switch(config)#policy-map type tapagg t-policy_1
switch(config-pmap-t-policy_1)#show active
 policy-map type tapagg t-policy_1
   100 match ip ospf any any set aggregation-group t-group
   120 class fred
      set aggregation-group t-group id-tag 444
   140 class t-class_2
      set id-tag 500
   160 class t-class_3
      set id-tag 600
   180 class t-class_4
      set id-tag 700
switch(config-pmap-t-policy_1)#
```

service-policy type tapagg (Interface mode)

The **service-policy type tapagg** command applies a specified tapagg policy map to the configuration mode interface. A policy map is a data structure that identifies data traffic through class maps and match rules, then specifies the method of replicating the traffic. This command is active only when tap aggregation mode is enabled on the switch.

The **no service-policy type tapagg** and **service-policy type tapagg** commands remove the policy map assignment from the configuration mode interface by deleting the corresponding **service-policy tapagg** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration
Interface-Port-Channel Configuration

Command Syntax

```
service-policy type tapagg TRAFFIC_DIRECTION policymap_name
no service-policy tapagg TRAFFIC_DIRECTION
default service-policy tapagg TRAFFIC_DIRECTION
```

Parameters

- ***TRAFFIC_DIRECTION*** IP address or peer group name. Values include:
  - **input** Policy map applies to inbound packet streams.
- *map_name* Name of policy map.

Guidelines

A policy map that is attached to a port channel interface takes precedence for member interfaces of the port channel over their individual Ethernet interface configuration. Members that are removed from a port channel revert to the policy map implementation specified by its Ethernet interface configuration.

Related Commands

- **policy-map type tapagg** places the switch in policy-map configuration mode to create a policy map.

Example

- These commands apply the ***t-policy_1*** policy map to Ethernet interface 17.

```
switch(config)#interface ethernet 17
switch(config-if-Et17)#service-policy type tapagg input t-policy_1
switch(config-if-Et17)#
```

set (policy-map-class (tapagg))

The **set** command specifies the data replication method for traffic filtered by the associated class map in the configuration mode policy map. The **set** command specifies one of these replication actions for filtered data packets.

- specifies an aggregation group
- specifies a VLAN identity tag for replicated packets.
- specifies an aggregation group and a VLAN identity tag.

The **no set** and **default set** commands remove the specified set command data action from the configuration mode class by deleting the associated **set** command from ***running-config***.

Command Mode

Policy-map-class (tapagg) configuration
accessed through **class (policy-map (tapagg))** command

Command Syntax

```
set SET_VALUE
no set SET_VALUE
default set SET_VALUE
```

Parameters

- ***SET_VALUE*** specifies the replication method for filtered packets.
  - **aggregation group** *agg_group* Replication specified by aggregation group.
  - **id-tag** <***1*** to ***4094***> Packet is identity tagged with specified VLAN number.
  - **aggregation group** *agg_group* **id-tag** <***1*** to ***4094***> Assigns agg group and identity tag.

Related Commands

- **policy-map type tapagg** places the switch in policy-map (tapagg) configuration mode.
- **class (policy-map (tapagg))** assigns a class to the configuration mode policy-map.
- **match (policy-map (tapagg))** assigns a rule to the configuration mode policy-map.

Guidelines

When a class is not associated with a **set** command, the filtered traffic is managed as specified by the tap port's default aggregation group.

Example

- These commands place the switch in policy-map-class to add the ***t-class_1*** class map to the ***t-policy_1*** policy map. Packets filtered by the class map are identity tagged with VLAN 444 and replicated as specified through the ***t-group*** aggregation group.

```
switch(config)#policy-map type tapagg t-policy_1
switch(config-pmap-t-policy_1)#class t-class_1
switch(config-pmap-c-t-policy_1-t-class_1)#set aggregation-group t-group id-tag
444
switch(config-pmap-c-t-policy_1-t-class_1)#exit
switch(config-pmap-t-policy_1)#exit
switch(config)#policy-map type tapagg t-policy_1
switch(config-pmap-t-policy_1)#show active
 policy-map type tapagg t-policy_1
   10 class t-class_1
      set aggregation-group t-group id-tag 444
switch(config-pmap-t-policy_1)#
```

show interfaces tap

The **show interfaces tap** command displays tap port configuration information for the specified interfaces.

Command Mode

EXEC

Command Syntax

show interfaces [*INTERFACE*] tap [*INFO_LEVEL*]

Parameters

- ***INTERFACE*** Interface type and numbers. Options include:
  - <no parameter> all interfaces.
  - **ethernet** *e_range* Ethernet interface range specified by *e_range*.
  - **management** *m_range* Management interface range specified by *m_range*.
  - **port-channel** *p_range* Port-Channel Interface range specified by *p_range*.

    Valid *e_range*, *m_range*, and *p_range* formats include number, number range, or comma-delimited list of numbers and ranges.
- ***INFO_LEVEL*** amount of information that is displayed. Options include:
  - <no parameter> command displays table that summarizes tap data.
  - **detail** command tap data summary table and a list of ACLS applied to tap ports.

Example

- This command displays tap port configuration information for ethernet interfaces 36 through 40.

```
switch>show interface ethernet 31-35 tap
Port      Configured     Status         Native   Id   Truncation Default
          Mode                          Vlan     Vlan            Group
-------------------------------------------------------------------------
Et31      tap            tap            301      31   0          tag_1
Et32      tap            tap            1        132  0          tag_1
Et33      tap            tap            303      233  0          tag_1
Et34      tap            tap            1        334  0          tag_3
Et35      tap            tap            1        345  0          tag_3
switch>
```

- This command displays detailed tap port configuration information for ethernet interface 31.

```
switch>show interface ethernet 31 tap detail
Port      Configured     Status         Native   Id   Truncation Default
          Mode                          Vlan     Vlan            Group
-------------------------------------------------------------------------
Et31      tap            tap            301      31   0          tag_1



Port      ACLs Applied
-------------------------------------------------------------------
switch>
```

show interfaces tool

The **show interfaces tool** command displays tool port configuration information for the specified interfaces.

Command Mode

EXEC

Command Syntax

show interfaces [*INTERFACE*] tool

Parameters

- ***INTERFACE*** Interface type and numbers. Options include:
  - <no parameter> all interfaces.
  - **ethernet** *e_range* Ethernet interface range specified by *e_range*.
  - **management** *m_range* Management interface range specified by *m_range*.
  - **port-channel** *p_range* Port-Channel Interface range specified by *p_range*.

    Valid *e_range*, *m_range*, and *p_range* formats include number, number range, or comma-delimited list of numbers and ranges.

Example

- This command displays tool port configuration information for ethernet interfaces 36 through 40.

```
switch>show interface ethernet 36-40 tool
Port      Configured     Status         Allowed             Id   Timestamp
          Mode                          Vlans               Tag  Mode
--------------------------------------------------------------------------
Et36      tool           tool           201-205             Off  None
Et37      tool           tool           201-205             Off  None
Et38      tool           tool           201-205             Off  None
Et39      access         errdisabled    All                 Off  None
Et40      tool           tool           All                 On   None

switch>
```

show platform fm6000 keyframe

The **show platform fm6000 keyframe** command displays configured information for the specified keyframes. Keyframes are routable IP packets that the switch inserts into a data stream to provide contextual information that correlate timestamps inserted into data packets with the absolute UTC time and the switch's complete ASIC time counter.

Command Mode

Privileged EXEC

Command Syntax

show platform fm6000 keyframe [*KEYFRAME_ID*]

Parameters

- ***KEYFRAME_ID*** Specifies keyframes that the command displays. Options include:
  - <no parameter> Command displays all configured keyframes.
  - *kf_name* Name of single keyframe that the command displays.

Examples

- This command displays information concerning the three keyframes that the switch sends.

```
switch#show platform fm6000 keyframe

Keyframe key-2
------------------------
Egress Interface(s): Ethernet17, Ethernet18, Ethernet19, Ethernet20, Ethernet21
Source IP: 10.22.30.144
Destination IP: 10.21.1.14
Destination MAC: 00:09:00:09:00:09
Device ID: 0
Rate: 5 packet(s) per second


Keyframe key-1
------------------------
Egress Interface(s): Ethernet11, Ethernet12, Ethernet13, Ethernet14, Ethernet15
Source IP: 10.22.30.146
Destination IP: 10.21.1.4
Destination MAC: 00:10:4e:21:9f:11
Device ID: 0
Rate: 2 packet(s) per second

switch#
```

show tap aggregation groups

The **show tap aggregation groups** command displays the tap and tool port members of the specified tap aggregation groups.

Command Mode

EXEC

Command Syntax

show tap aggregation groups [*INFO_LEVEL*] [*GROUP_NAMES*]

Parameters

- ***INFO_LEVEL*** Port information that is displayed.
  - <no parameter> command displays active tap and tool ports.
  - **detail** command displays all configured tap and tool ports, including inactive ports.
- ***GROUP_NAMES*** Tap aggregation groups. Options include:
  - <no parameter> All tap aggregation groups
  - *group_list* Tap aggregation group list.

    Valid *group_list* format is a space-delimited list of one or more tap aggregation group names.

Example

- This command displays the contents of all configured tap aggregation groups.

```
switch>show tap aggregation groups
Group Name                          Tool Members
------------------------------------------------------
analyze2                            Po101, Po102
analyze3                            Po101, Po103

Group Name                          Tap Members
------------------------------------------------------
analyze2                            Et41, Et42
analyze3                            Et43
switch>
```

switchport tap allowed vlan

The **switchport tap allowed vlan** command creates or modifies the list of VLANs for which the configuration mode interface, in tap mode, handles tagged traffic. By default, interfaces handle tagged traffic for all VLANs. Command settings persist in ***running-config*** without taking effect when the switch is not in tap aggregation mode or the interface is not in tap aggregation mode.

The **no switchport tap allowed vlan** and **default switchport tap allowed vlan** commands restore the tap mode default allowed VLAN setting of ***all*** by removing the corresponding **switchport tap allowed vlan** statement from ***running-config***.

Command Mode

Interface-Ethernet Configuration
Interface-Port Channel Configuration

Command Syntax

```
switchport tap allowed vlan EDIT_ACTION
no switchport tap allowed vlan
default switchport tap allowed vlan
```

Parameters

- ***EDIT_ACTION*** modifications to the VLAN list.
  - *v_range* Creates VLAN list from *v_range*.
  - **add** *v_range* Adds specified VLANs to current list.
  - **all** VLAN list contains all VLANs.
  - **except** *v_range* VLAN list contains all VLANs except those specified.
  - **none** VLAN list is empty (no VLANs).
  - **remove** *v_range* Removes specified VLANs from current list.

    Valid *v_range* formats include number (1 to 4094), range, or comma-delimited list of numbers and ranges.

Example

- These commands create the tap mode allowed VLAN list of 26-30 for Ethernet interface 20.

```
switch(config)#interface ethernet 20
switch(config-if-Et20)#switchport tap allowed vlan 26-30
switch(config-if-Et20)#show active
interface Ethernet20
   switchport mode tap
   switchport tap allowed vlan 26-30
switch(config-if-Et20)#
```

switchport tap default group

The **switchport tap default group** command assigns the configuration mode interface as a tap port member to the specified tool group. Tap aggregation groups associate a set of tap ports with a set of tool ports. Both tap ports and tool ports may belong to multiple tap aggregation groups.

The **no switchport tap default group** and **default switchport tap default group** commands remove the configuration mode interface from the tap aggregation group to which it is assigned by deleting the corresponding statement from ***running-config***.

Command Mode

Interface-Ethernet Configuration
Interface-Port Channel Configuration

Command Syntax

```
switchport tap default group group_name
no switchport tap default group
default switchport tap default group
```

Parameters

- *group_name* tool group name.

Restriction

This command is only available on FM6000 platform switches.

Example

- These commands assign port channel 101 to tap aggregation group ***tag-1***.

```
switch(config)#interface port-channel 101
switch(config-if-Po101)#switchport tap default group tag-1
switch(config-if-Po101)#show interfaces port-channel 101 tap
Port      Configured     Status         Native   Id   Truncation Default
          Mode                          Vlan     Vlan            Group
-------------------------------------------------------------------------
Po101     access         notconnect     1        1    0          tag-1
switch(config)#
```

switchport tap identity

The **switchport tap identity** command associates a VLAN number to the configuration mode tap interface. Tool ports that are configured to encapsulate packets with an dot1q-style tag enter the number specified by this command as the s-VLAN (tier 1) for packets received from this tap port. The default identity value is 1.

The **no switchport tap identity** and **default switchport tap identity** commands restore VLAN 1 as the configuration mode port's identity vlan by removing the corresponding switchport tap identity command from ***running-config***.

Command Mode

Interface-Ethernet Configuration
Interface-Port Channel Configuration

Command Syntax

```
switchport tap identity port_id
no switchport tap identity
default switchport tap identity
```

Parameters

- *port_id* port's identity VLAN. Value ranges from 1 to 4094. Default is 1.

Related Commands

- **switchport tool identity** configures a tool port to encapsulate packets received from tap ports.

Restriction

This command is available on FM6000 platform switches.

Example

- These commands 171 as the identity value for ethernet interface 17.

```
switch(config)#interface ethernet 17
switch(config-if-Et17)#switchport tap identity 171
switch(config-if-Et17)#show active
interface Ethernet17
   switchport tap identity 171
Switch(config-if-Et17)#show interfaces ethernet 17 tap
Port      Configured     Status         Native   Id   Truncation Default
          Mode                          Vlan     Vlan            Group
-------------------------------------------------------------------------
Et17      access         connected      1        171  0          ---
switch(config-if-Et17)#
```

switchport tap native vlan

The **switchport tap native vlan** command specifies the tap mode native VLAN for the configuration mode interface. Interfaces in tap mode associate untagged frames with the native VLAN. The default native VLAN for all interfaces is VLAN 1. Command settings persist in ***running-config*** without taking effect when the switch is not in tap aggregation mode or the interface is not in tap mode.

The **no switchport tap native vlan** and **default switchport tap native vlan** commands restore VLAN 1 as the tap mode native VLAN to the configuration mode interface by removing the corresponding switchport tap native vlan command from ***running-config***.

Command Mode

Interface-Ethernet Configuration
Interface-Port Channel Configuration

Command Syntax

```
switchport tap native vlan v_num
no switchport tap native vlan
default switchport tap native vlan
```

Parameters

- *v_num* tap mode native VLAN ID. Value ranges from 1 to 4094. Default is 1.

Restriction

This command is available on FM6000 platform switches.

Example

- These commands assign VLAN 25 as the tap mode native VLAN for Ethernet interface 7.

```
switch(config)#interface ethernet 7
switch(config-if-Et7)#switchport tap native vlan 25
switch(config-if-Et7)#show interface ethernet 7 tap
Port      Configured     Status         Native   Id   Truncation Default
          Mode                          Vlan     Vlan            Group
-------------------------------------------------------------------------
Et7       tool           connected      25       1    0          ---
switch(config-if-Et7)#
```

switchport tap truncation

The **switchport tap truncation** command configures the configuration mode interface, as a tap port, to truncate inbound packets to the specified packet size. This command is in effect when the port is in tap mode and the switch is in tap aggregation mode. Command settings persist in ***running-config*** without taking effect when the switch is not in tap aggregation mode or the interface is not in tap mode. By default, tap ports do not truncate inbound packets.

The **no switchport tap truncation** and **default switchport tap truncation** commands restore the default behavior of not truncating packets received by the configuration mode interface by removing the corresponding **switchport tap truncation** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration
Interface-Port Channel Configuration

Command Syntax

```
switchport tap truncation packet_size
no switchport tap truncation
default switchport tap truncation
```

Parameters

- *packet_size* Size of truncated packets (bytes). Value ranges from 100 to 9236. Default value of 0 corresponds to not truncating packets.

Restriction

This command is available on FM6000 platform switches.

Examples

- These commands configure ethernet interface 38 to truncate packets to 150 bytes.

```
switch(config)#interface ethernet 38
switch(config-if-Et38)#switchport tap truncation 150
switch(config-if-Et38)#show interface ethernet 38 tap
Port      Configured     Status         Native   Id   Truncation Default
          Mode                          Vlan     Vlan            Group
-------------------------------------------------------------------------
Et38      access         notconnect     1        1    150        ---
switch(config-if-Et38)#
```

- These commands configure ethernet interface 38 to send complete packets to tool ports in its tap aggregation group.

```
switch(config-if-Et38)#no switchport tap truncation
switch(config-if-Et38)#show interface ethernet 38 tap
Port      Configured     Status         Native   Id   Truncation Default
          Mode                          Vlan     Vlan            Group
-------------------------------------------------------------------------
Et38      access         notconnect     1        1    0          ---
switch(config-if-Et38)#
```

switchport tool allowed vlan

The **switchport tool allowed vlan** command creates or modifies the list of VLANs for which the configuration mode interface, in tool mode, handles tagged traffic. By default, interfaces handle tagged traffic for all VLANs. Command settings persist in ***running-config*** without taking effect when the switch is not in tap aggregation mode or the interface is not in tap aggregation mode.

The **no switchport tool allowed vlan** and **default switchport tool allowed vlan** commands restore the tool mode default allowed VLAN setting of ***all*** by removing the corresponding **switchport tool allowed vlan** statement from ***running-config***.

Command Mode

Interface-Ethernet Configuration
Interface-Port Channel Configuration

Command Syntax

```
switchport tool allowed vlan EDIT_ACTION
no switchport tool allowed vlan
default switchport tool allowed vlan
```

Parameters

- ***EDIT_ACTION*** modifications to the VLAN list.
  - *v_range* Creates VLAN list from *v_range*.
  - **add** *v_range* Adds specified VLANs to current list.
  - **all** VLAN list contains all VLANs.
  - **except** *v_range* VLAN list contains all VLANs except those specified.
  - **none** VLAN list is empty (no VLANs).
  - **remove** *v_range* Removes specified VLANs from current list.

  Valid *v_range* formats include number, range, or comma-delimited list of numbers and ranges.

Example

- These commands create the tool mode allowed VLAN list of 16-20 for Ethernet interface 38.

```
switch(config)#interface ethernet 38
switch(config-if-Et38)#switchport tool allowed vlan 16-20
switch(config-if-Et38)#show interfaces ethernet 38 tool
Port      Configured     Status         Allowed             Id   Timestamp
          Mode                          Vlans               Tag  Mode
-------------------------------------------------------------------------
Et38      access         notconnect     16-20               Off  None
switch(config-if-Et38)#
```

switchport tool group

The **switchport tool group** command modifies the configuration mode interface's tool port membership in the specified tap aggregation groups. Tool ports may belong to multiple tap aggregation groups. Command options for configuring a port's tap aggregation group membership include:

- specify the groups to which the port belongs (supersedes the port's previous group memberships).
- add to the list of groups to which the port is a member.
- delete from the list of groups to which the port is a member.

Tap aggregation groups associate a set of tap ports with a set of tool ports. A tap port can belong to a maximum of one default tap aggregation group.

The **no switchport tool default group** and **default switchport tool default group** commands remove the configuration mode interface from all tap aggregation groups to which it is assigned as a tool port by modifying the corresponding statements in ***running-config***.

Command Mode

Interface-Ethernet Configuration
Interface-Port Channel Configuration

Command Syntax

```
switchport tool group EDIT_ACTION
no switchport tool group
default switchport tool group
```

Parameters

- ***EDIT_ACTION*** specifies changes to the list of groups to which interface is a member.
  - **add** *group_list* Specifies additional groups to which port belongs.
  - **remove** *group_list* Removes interface as a tool port member from specified groups.
  - **set** *group_list* Specifies groups to which interface belongs as a tool port.

    Valid *group_list* format is a space-delimited list of one or more tap aggregation group names.

Restriction

This command is available on FM6000 platform switches.

Example

- These commands associate interface ethernet 40 with three tap aggregation groups.

```
switch(config)#interface ethernet 40
switch(config-if-Et40)#switchport tool group set tag-1 tag-2 tag-3
switch(config-if-Et40)#show active
interface Ethernet40
   switchport tool group set tag-3 tag-2 tag-1
switch(config-if-Et40)#
```

- These commands add tag-7 to the tap aggregation groups of which ethernet interface 40 belongs.

```
switch(config-if-Et40)#switchport tool group add tag-7
switch(config-if-Et40)#show active
interface Ethernet40
   switchport tool group set tag-3 tag-7 tag-2 tag-1
switch(config-if-Et40)#
```

- These commands specify tag-9 as the only group of which ethernet interface 40 is a member.

```
switch(config-if-Et40)#switchport tool group set tag-9
switch(config-if-Et40)#show active
interface Ethernet40
   switchport tool group set tag-9
switch(config-if-Et40)#
```

switchport tool identity

The **switchport tool identity** command configures the configuration mode interface to include a tier 1 VLAN tag (dot1q) to packets it receives from tap ports. The VLAN number on the dot1q tag is specified by the **switchport tap identity** command configured for the tap port that supplies the packets. By default, tool ports do not encapsulate packets with the tier 1 VLAN tag.

The **no switchport tool identity** and **default switchport tool identity** commands restore the default VLAN handling method for the configuration mode interface by removing the corresponding **switchport tool identity** statement from ***running-config***.

Command Mode

Interface-Ethernet Configuration
Interface-Port Channel Configuration

Command Syntax

```
switchport tool identity dot1q
no switchport tool identity
default switchport tool identity
```

Restriction

This command is available on FM6000 platform switches.

Example

- These commands configure ethernet interface 40 to include a dot1q tag on egress packets.

```
switch(config)#interface ethernet 40
switch(config-if-Et40)#switchport tool identity dot1q
switch(config-if-Et40)#show active
interface Ethernet40
   switchport mode tool
   switchport tool identity dot1q
   switchport tool group set tag-9
switch(config-if-Et40)#
```

switchport tool truncation

The **switchport tool truncation** command configures the configuration mode interface, as a tool port, to truncate outbound packets to 160 bytes. This command is in effect when the port is in tool mode and the switch is in tap aggregation mode. Command settings persist in ***running-config*** without taking effect when the switch is not in tap aggregation mode or the interface is not in tool mode. By default, tool ports do not truncate outbound packets.

The **no switchport tool truncation** and **default switchport tool truncation** commands restore the default behavior (not truncating packets that exit the configuration mode interface) by removing the corresponding **switchport tool truncation** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration
Interface-Port Channel Configuration

Command Syntax

```
switchport tool truncation packet_size
no switchport tool truncation
default switchport tool truncation
```

Parameters

- *packet_size* Size of truncated packets in bytes. The only permitted value is 160.

Examples

- These commands configure ethernet interface 38, as a tool port, to truncate packets on egress to 160 bytes.

```
switch(config)#interface ethernet 38
switch(config-if-Et38)#switchport mode tool
switch(config-if-Et38)#switchport tool truncation 160
switch(config-if-Et38)#
```

- These commands configure ethernet interface 38 to send complete packets.

```
switch(config-if-Et38)#no switchport tool truncation
switch(config-if-Et38)#
```

tap aggregation

The **tap aggregation** command places the switch in tap-agg configuration mode. The switch's tap aggregation mode is enabled or disabled by the **mode** command in tap-agg configuration mode.

When tap aggregation mode is enabled, normal switching and routing operations are disabled. A port's switchport status depends on the switch's tap aggregation mode and the port's switchport mode:

- tap aggregation mode enabled: tap and tool ports are enabled. Switching ports are errdisabled.
- tap aggregation mode disabled: tap and tool ports are errdisabled. Switching ports are enabled.

The **no tap aggregation** and **default tap aggregation** commands disable tap aggregation mode on the switch by removing all tap-agg configuration mode commands from ***running-config***.

Tap-agg configuration mode is not a group change mode; ***running-config*** is changed immediately upon entering commands. Exiting tap-agg configuration mode does not affect ***running-config***. The **exit** command returns the switch to global configuration mode.

Command Mode

Global Configuration

Command Syntax

```
tap aggregation
no tap aggregation
default tap aggregation
```

Commands Available in Tap Aggregation Configuration Mode

- **mode (tap-agg configuration mode)**
- **switchport mode**

Related Commands

- **switchport mode**

Example

- These commands place the switch in tap-agg configuration mode and enables tap aggregation mode.

```
switch(config)#tap aggregation
switch(config-tap-agg)#mode exclusive
switch(config-tap-agg)#show active
tap aggregation
   mode exclusive
switch(config-tap-agg)#
```

- These commands disables tap aggregation mode by removing all tap-agg configuration mode commands from ***running-config***.

```
switch(config)#no tap aggregation
switch(config)#
```

Chapter 18

VLANs

This chapter describes Arista's VLAN implementation and MAC address tables.

Sections in this chapter include:

- Section 18.1: VLAN Introduction
- Section 18.2: VLAN Conceptual Overview
- Section 18.3: VLAN Configuration Procedures
- Section 18.4: VLAN Configuration Commands

18.1 VLAN Introduction

Arista switches support industry standard 802.1q VLANs. Arista EOS provides tools to manage and extend VLANs throughout the data center network.

18.2 VLAN Conceptual Overview

18.2.1 VLAN Definition

A virtual local area network (VLAN) allows a group of devices to communicate as if they were in the same network regardless of their physical location. VLANs are layer 2 structures based on the 802.1Q standard.

These parameters are associated with a VLAN:

- VLAN number (1-4094): VLAN numbers uniquely identify the VLAN within a network. VLAN 1 exists by default; all other VLANs only exist after they are configured.
- VLAN name (optional): The VLAN name is a text string that describes the VLAN.
- VLAN state (***active*** or ***suspended***): The state specifies the VLAN transmission status within the switch. In the ***suspended*** state, VLAN traffic is blocked on all switch ports. The default state is ***active***.

VLANs define layer 2 broadcast domains in a layer 2 network, in which each device can receive broadcast frames sent by any other within the domain. Switches accommodating multiple broadcast domains serve as multi-port bridges where each broadcast domain is a distinct virtual bridge. Traffic does not pass directly between different VLANs within a switch or between two switches.

18.2.2 VLAN Switching

Ethernet and port channel interfaces are configured as switched ports by default. Switched ports are configurable as members of one or more VLANs. Switched ports ignore all IP-level configuration commands, including IP address assignments.

18.2.2.1 VLAN Trunking and Trunk Groups

Trunking extends multiple VLANs beyond the switch through a common interface or port channel.

A trunk group is the set of physical interfaces that comprise the trunk and the collection of VLANs whose traffic is carried on the trunk. The traffic of a VLAN that belongs to one or more trunk groups is carried only on ports that are members of trunk groups to which the VLAN belongs, i.e., VLANs configured in a trunk group are 'pruned' off all ports that are not associated with the trunk group. See the Trunk Ports example section for further details.

Important! Be cautious when using allowed VLAN lists or trunk groups to ensure that the VLAN topology is consistent with any Layer-2 control protocol topology, or unpredictable results can occur.

VLAN traffic is carried through Ethernet or LAG ports. A port's switchport mode defines the number of VLANs for which the port can carry traffic.

- Access ports carry traffic for one VLAN – the access VLAN. Access ports associate untagged frames with the access VLAN. Access ports drop tagged frames that are not tagged with the access VLAN.
- Trunk ports carry traffic for multiple VLANs. Tag frames specify the VLAN for which trunk ports process packets.

18.2.2.2 Q-in-Q Trunking

A Q-in-Q network is a multi-tier layer 2 VLAN network. A typical Q-in-Q network is composed of a service provider network (tier 1) where each node connects to a customer network (tier 2).

802.1ad is a networking standard that supports Q-in-Q networks by allowing multiple 802.1Q tags in an Ethernet frame.

Each interface in a customer network is assigned to a customer-VLAN (c-VLAN). Packets in c-VLANs contain 802.1q tags that switch traffic within the network. c-VLANs access the service provider VLAN (s-VLAN) through a provider switch. Customer switch ports connect to an s-VLAN through provider switch edge ports, which are configured as dot1q ports and operate as follows:

- Inbound traffic (from customer switches): adds an s-VLAN tag, then forwards packets to the provider network.
- Outbound traffic (to customer switches): removes the s-VLAN tag, then forwards packets to the customer network.

18.2.2.3 TPID (Configurable Ethertypes)

By default, VLAN-tagged packets carry a tag protocol identifier (TPID) of 0x8100. On some Arista platforms, however, the TPID of a switchport can be modified in accordance with IEEE 802.1ad to allow for the use of 802.1q TPIDs other than 0x8100. Well known and standard tags include:

- **0x8100** customer VLAN
- **0x88a8** service VLAN tag used in provider bridging
- **0x9100** service VLAN tag used in provider bridging (common, but not standardized)

Other non-standard TPID values may also be configured for interoperability with legacy equipment or non-standard systems. Values range from 0x600 (1536) through 0xFFFF (65535).

Non-default TPID values are most commonly used for provider bridging on a network-to-network interface.

18.2.3 VLAN Routing

Each VLAN can be associated with a switch virtual interface (SVI), also called a VLAN interface. The VLAN interface functions in a routed network (layer 3) with an assigned IP subnet address. Connecting different VLANs requires layer 3 networking.

18.2.3.1 VLAN Interfaces

A switched virtual interface (SVI) connects to the VLAN segment on the switch to provide layer 3 processing for packets from the VLAN. An SVI can be activated only after it is connected to a VLAN. SVIs are typically configured for a VLAN to a default gateway for a subnet to facilitate traffic routing with other subnets.

In a layer 3 network, each VLAN SVI is associated with an IP subnet, with all stations in the subnet members of the VLAN. Traffic between different VLANs is routed when IP routing is enabled.

18.2.3.2 Internal VLANs

A routed port is an Ethernet or port channel interface that functions as a layer 3 interface. Routed ports do not bridge frames nor switch VLAN traffic. Routed ports have IP addresses assigned to them and packets are routed directly to and from the port.

The switch allocates an internal VLAN for an interface when it is configured as a routed port. The internal VLAN is assigned a previously unused VLAN ID. The switch prohibits the subsequent configuration of VLANs and VLAN interfaces with IDs corresponding to allocated internal VLANs.

18.2.3.3 VLAN Translation

VLAN translation allows you to map packets from one VLAN to another.

18.3 VLAN Configuration Procedures

These sections describe basic VLAN configuration tasks.

- Section 18.3.1: Creating and Configuring VLANs
- Section 18.3.2: Configuring VLAN Switching
- Section 18.3.3: Creating and Configuring VLAN Interfaces
- Section 18.3.4: Allocating Internal VLANs
- Section 18.3.5: VLAN Translation

18.3.1 Creating and Configuring VLANs

The CLI provides two methods of creating VLANs.

- Explicitly through the **vlan** command.
- Implicitly through the **switchport access vlan** command.

  The **switchport access vlan** command generates a warning message when it creates a VLAN.

To create a VLAN, use the **vlan** command in global configuration mode. Valid VLAN numbers range between 1 and 4094. To create multiple VLANs, specify a range of VLAN numbers.

To edit an existing VLAN, enter the **vlan** command with the number of the existing VLAN.

Example

- This command creates VLAN 45 and enters VLAN configuration mode for the new VLAN.

```
switch(config)#vlan 45
switch(config-vlan-45)#
```

Use the **name (VLAN configuration mode)** command to assign a name to a VLAN.

Example

- These commands assign the name Marketing to VLAN 45.

```
switch(config)#vlan 45
switch(config-vlan-45)#name Marketing
switch(config-vlan-45)#show vlan 45
VLAN Name                            Status    Ports
---- ------------------------------- --------- -----------------------
45   Marketing                       active    Et1

switch(config-vlan-45)#
```

To change a VLAN's state, use the **state** command in VLAN configuration mode.

Examples

- These commands suspend VLAN 45. VLAN traffic is blocked on all switch ports.

```
switch(config)#vlan 45
switch(config-vlan-45)#state suspend
switch(config-vlan-45)#show vlan 45
VLAN Name                            Status    Ports
---- ------------------------------- --------- -----------------------
45   Marketing                       suspended

switch(config-vlan-45)#
```

- These commands activate VLAN 45.

```
switch(config)#vlan 45
switch(config-vlan-45)#state active
switch(config-vlan-45)#show vlan 45
VLAN Name                            Status    Ports
---- ------------------------------- --------- -----------------------
45   Marketing                       active    Et1

switch(config-vlan-45)#
```

18.3.2 Configuring VLAN Switching

The following describe the configuration of VLAN ports.

18.3.2.1 Access Ports

Access ports carry traffic for one VLAN, as designated by a **switchport access vlan** command. Access ports associate untagged frames with the access VLAN. Tagged frames received by the interface are dropped unless they are tagged with the access VLAN.

To configure an interface group as an access port, use the **switchport mode** command.

Example

- These commands configure Ethernet interface 1 as an access port.

```
switch(config)#interface ethernet 1
switch(config-if-Et1)#switchport mode access
switch(config-if-Et1)#
```

To specify the port's access VLAN, use the **switchport access vlan** command.

Examples

- These commands configure VLAN 15 as the access VLAN for Ethernet interface 5.

```
switch(config)#interface ethernet 5
switch(config-if-Et5)#switchport access vlan 15
switch(config-if-Et5)#
```

- These commands configure Ethernet interface 1 through 3 as access ports that process untagged frames as VLAN 5 traffic.

```
switch(config)#interface Ethernet 1-3
switch(config-if-Et1-3)#switchport mode access
switch(config-if-Et1-3)#switchport access vlan 5
switch(config-if-Et1-3)#show interfaces ethernet 1-3 vlans
Port       Untagged Tagged
Et1        None     23,25
Et2        18       -
Et3        None     14
switch(config-if-Et1-3)#
```

18.3.2.2 Trunk Ports

Trunk ports carry traffic for multiple VLANs. Messages use tagged frames to specify the VLAN for which trunk ports process traffic.

- The **vlan trunk list** specifies the VLANs for which the port handles tagged frames. The port drops any packets tagged for VLANs not in the VLAN list.
- The **native vlan** is the VLAN where the port switches untagged frames.

To configure an interface group as a trunk port, use the **switchport mode** command.

Example

- These commands configure Ethernet interface 8 as a trunk port.

```
switch(config)#interface ethernet 8
switch(config-if-Et8)#switchport mode trunk
switch(config-if-Et8)#
```

By default all VLANs are permitted on a port configured with 'switchport mode trunk'. To limit the port's VLAN trunk list, use the **switchport trunk allowed vlan** command. Only VLANs in the allowed list will be permitted.

Examples

- These commands configure VLAN 15, 20, 21, 22, 40, and 75 as the explicitly permitted VLAN trunk list for Ethernet interface 12-16.

```
switch(config)#interface ethernet 12-16
switch(config-if-Et12-16)#switchport trunk allowed vlan 15,20-22,40,75
switch(config-if-Et12-16)#
```

- These commands explicitly permit VLAN 100 through 120 to the VLAN trunk list for Ethernet interface 14.

```
switch(config)#interface ethernet 14
switch(config-if-Et14)#switchport trunk allowed vlan add 100-120
switch(config-if-Et14)#
```

To specify the port's native VLAN, use the **switchport trunk native vlan** command.

Example

- These commands configure VLAN 12 as the native VLAN trunk for Ethernet interface 10.

```
switch(config)#interface ethernet 10
switch(config-if-Et10)#switchport trunk native vlan 12
switch(config-if-Et10)#
```

By default, ports send native VLAN traffic with untagged frames. The **switchport trunk native vlan** command can also configure the port to send native VLAN traffic with tag frames.

Examples

- These commands configure Ethernet interface 10 to send native VLAN traffic as tagged.

```
switch(config)#interface ethernet 10
switch(config-if-Et10)#switchport trunk native vlan tag
switch(config-if-Et10)#
```

- These commands configure Ethernet interface 12 as a trunk with VLAN 15 as the native VLAN. The port's trunk list includes all VLANs except 201-300.

```
switch(config)#interface ethernet 12
switch(config-if-Et12)#switchport mode trunk
switch(config-if-Et12)#switchport trunk native vlan 15
switch(config-if-Et12)#switchport trunk allowed vlan except 201-300
switch(config-if-Et12)#
```

Example

- Assume that all ports on the switch are configured with switchport mode trunk similar to ethernet 1 and 2 shown below:

```
!
interface ethernet 1
    switchport mode trunk
!
interface ethernet 2
    switchport mode trunk
!
```

Further assume that VLAN 30 is **not** configured as part of a trunk group

```
switch#show vlan
VLAN  Name                             Status    Ports
----- -------------------------------- --------- -------------------------------
1     default                          active    Et1, Et2
30    vlan30                           active    Et1, Et2
```

Now configure VLAN 30 as part of trunk group 30:

```
switch(config)#vlan 30
switch(config-vlan-30)#trunk group 30
```

This updates the VLAN membership for VLAN 30.

```
switch#show vlan
VLAN  Name                             Status    Ports
----- -------------------------------- --------- -------------------------------
1     default                          active    Et1, Et2
30    vlan30                           active
```

Note: Vlan 30 is no longer on Et1, Et2 i.e. it has been ‘pruned’ due to the trunk group command in the vlan configuration.

To permit VLAN 30 on Et1 you need to associate the interface with the trunk group as follows:

```
switch(config-if-Et1)#switchport trunk group 30

Now we see Et1 included in the vlan 30 list

switch#show vlan
VLAN  Name                             Status    Ports
----- -------------------------------- --------- -------------------------------
1     default                          active    Et1, Et2
30    vlan30                           active    Et1
```

The trunk group command is not additive to the allowed vlan command

```
interface ethernet 1
        switchport mode trunk
        switchport trunk allowed vlan 10
        switchport trunk group trunk30

Vlan 30 will not be permitted on the interface as it is not listed in the allowed
vlan list.
```

18.3.2.3 Dot1q Tunnel Ports

Dot1q (802.1Q) is a tunneling protocol that encapsulates traffic from multiple customer (c-tag) VLANs in an additional single outer service provider (s-tag) VLAN for transit across a larger network structure that includes traffic from all customers. Tunneling eliminates the service provider requirement that every VLAN be configured from multiple customers, avoiding overlapping address space issues.

Tunneling preserves the inner VLANs through the tunneled network; these inner VLANs are ignored by intermediate devices that make forwarding decisions based only on the outermost VLAN tag (S-Tag)

A dot1q-tunnel port sits at the edge of the tunneled network. Unlike regular access ports, a dot1q-tunnel port does not drop traffic that arrives with 802.1Q tags in place; it ignores existing 802.1Q information and associates arriving traffic (with or without 802.1Q headers) with a new tunnel VLAN ID.

Packets arriving at a tunnel port are encapsulated with an additional 802.1Q tag that can be trunked between multiple devices like any traditional VLAN. When exiting a dot1-tunnel port, the S-Tag is removed to revert the customer traffic to its original tagged or untagged state.

To configure an interface group as a dot1q tunnel port, use the **switchport mode** command.

Example

- These commands configure Ethernet interface 12 as a dot1q tunnel port.

```
switch(config)#interface ethernet 12
switch(config-if-Et12)#switchport mode dot1q-tunnel
switch(config-if-Et12)#
```

To specify the dot1q-tunnel port's access VLAN, use the **switchport access vlan** command. The port then handles all inbound traffic as untagged VLAN traffic.

Example

- These commands configure VLAN 60 as the access VLAN for Ethernet interface 12.

```
switch(config)#interface ethernet 12
switch(config-if-Et12)#switchport access vlan 60
switch(config-if-Et12)#
```

18.3.2.4 TPID Configuration

The default tag protocol identifier (TPID, also called dot1q ethertype) on all switch ports is 0x8100. To configure a different TPID on a port, use the **switchport dot1q ethertype** command. This feature is available only on 7280E and 7500E platforms.

Important! If dot1q tunneling is enabled on the interface, a TPID configured on the interface becomes irrelevant.

Example

- In this provider bridging example, Ethernet interface 1 is the user network interface and Ethernet interface 2 is the network-to-network interface. These commands configure dot1q tunneling on Ethernet interface 1 and set the TPID of Ethernet interface 2 to 0x9100.

```
switch(config)#interface ethernet 1
switch(config-if-Et1)#switchport mode dot1q-tunnel
switch(config-if-Et1)#interface ethernet 2
switch(config-if-Et2)#switchport mode trunk
switch(config-if-Et2)#switchport dot1q ethertype 0x9100
switch(config-if-Et2)#
```

In the above configuration, packets from Et1 to Et2 will undergo dot1q-tunneling (stacking of an additional dot1q tag), with an outer TPID of 0x9100 at egress, while packets with outer TPID 0x9100 going from Et2 to Et1 will have the outer tag removed at egress.

18.3.2.5 Layer 2 802.1Q Encapsulation

Layer 2 traffic encapsulation is enabled on the configuration mode interface for a specified VLAN through **l2-protocol encapsulation dot1q vlan**.

Example

- These commands enable traffic encapsulation for VLAN 200 traffic passing through Ethernet interface 2/5.

```
switch(config)#interface ethernet 5/2
switch(config-if-Et5/2)#l2-protocol encapsulation dot1q vlan 200
```

18.3.3 Creating and Configuring VLAN Interfaces

The **interface vlan** command places the switch in VLAN-interface configuration mode for modifying an SVI. An SVI provides a management address point and Layer 3 processing for packets from all VLAN ports.

Example

- This command enters VLAN-interface configuration mode for VLAN 12. The command also creates VLAN 12 interface if it was not previously created.

```
switch#config t
switch(config)#interface vlan 12
switch(config-if-Vl12)#
```

18.3.4 Allocating Internal VLANs

The **vlan internal allocation policy** command specifies the VLANs that the switch allocates as internal VLANs when configuring routed ports and the order of their allocation. By default, the switch allocates VLANs in ascending order. The default allocation range is between VLAN 1006 and VLAN 4094.

The **no switchport** command converts an Ethernet or port channel interface into a routed port, disabling layer 2 switching for the interface.

Examples

- This command configures the switch to allocate internal VLANs in ascending order starting with 1006.

```
switch(config)#vlan internal allocation policy ascending
switch(config)#
```

- This command configures the switch to allocate internal VLANs in descending order starting with 4094.

```
switch(config)#vlan internal allocation policy descending
switch(config)#
```

- This command configures the switch to allocate internal VLANs in descending order from 4094 through 4000.

```
switch(config)#vlan internal allocation policy descending range 4000 4094
switch(config)#
```

18.3.5 VLAN Translation

VLAN translation allows you to map packets from one VLAN to another. This can be carried out only on packets having a dot1q header (tagged frames). The translation rewrites the VID field (VLAN ID) in dot1q headers on packets passing through a switched port without changing any other fields.

VLAN translation also supports the ability to translate packets with a dot1q header to the internal VLAN for a routed port. The VLAN in the incoming packets is mapped to the internal VLAN of the routed port and packets egressing the routed port are encapsulated with a dot1q header for the specified VLAN. For egress packets, no priority information is added to the dot1q header and the priority from the incoming encapsulation will be retained.

When configuring the VLAN translation mode, consider the following:

- VLAN translation is only supported for tagged packets.
- BPDUs from STP, LLDP and other protocols are not affected by this mapping.
- VLAN translation is not applicable for access ports.
- Untagged packets entering the switch on the trunk native VLAN are not mapped.
- TPID and VLAN priority does not get re-written during the translation.

Per-port VLAN Translation on Switched Ports

The **switchport vlan mapping** command allows translation of the VLAN tag of traffic entering or exiting a switched port.

To use VLAN translation on a switched port, the port must be configured as a trunk port using the **switchport mode** command.

Example

- This command configures Ethernet interface 5 as a trunk port.

```
switch(config)#interface ethernet 5
switch(config-if-Et5)#switchport mode trunk
switch(config-if-Et5)#
```

By default, the translation is bidirectional: packets ingressing an interface through VLAN A are internally mapped to VLAN B; VLAN B packets egressing the same interface are mapped to VLAN A.

Examples

- These commands map Ethernet interface 5 traffic with dot1q tag 50 to bridging VLAN 60.

```
switch(config)#interface ethernet 5
switch(config-if-Et5)# switchport vlan mapping 50 60
switch(config-if-Et5)#
```

- These commands provides multiple 1:1 VLAN mappings under an interface.

```
switch(config)#interface ethernet 5
switch(config-if-Et5)# switchport vlan mapping 50 60
switch(config-if-Et5)# switchport vlan mapping 61 71
switch(config-if-Et5)# switchport vlan mapping 62 72
switch(config-if-Et5)#
```

- These commands translate only incoming packets.

```
switch(config)#interface ethernet 5
switch(config-if-Et5)# switchport vlan mapping in 50 60
switch(config-if-Et5)#
```

- These commands translate only egress packets.

```
switch(config)#interface ethernet 5
switch(config-if-Et5)#switchport vlan mapping out 60 50
switch(config-if-Et5)#
```

Per-port VLAN Translation on Routed Ports

On routed ports, the **encapsulation dot1q vlan** command (permitted only on routed ports) configures the VLAN on the interface to act as the native VLAN. This command will map packets ingressing with the specified VLAN ID to the internal VLAN ID of the routed port. All traffic egressing out of the routed port will be tagged with the VLAN ID specified in the command.

Examples

- These commands translate between VLAN 50 and the internal VLAN for Ethernet interface 5 (a routed port).

```
switch(config)#interface ethernet 5
switch(config-if-Et5)# no switchport
switch(config-if-Et5)# encapsulation dot1q vlan 50
switch(config-if-Et5)#
```

18.4 VLAN Configuration Commands

Global VLAN Configuration Commands

- interface vlan
- vlan
- vlan internal allocation policy

VLAN Configuration Mode Commands

- name (VLAN configuration mode)
- state
- trunk group

Layer 2 Interface (Ethernet and Port Channel) Configuration Commands

- switchport access vlan
- switchport mode
- switchport trunk allowed vlan
- switchport trunk group
- switchport trunk native vlan
- switchport vlan mapping

VLAN Interface Configuration Mode Commands

- autostate
- encapsulation dot1q vlan
- l2-protocol encapsulation dot1q vlan

Show Commands

- show dot1q-tunnel
- show interfaces switchport
- show interfaces switchport backup
- show interfaces trunk
- show interfaces vlans
- show vlan
- show vlan dynamic
- show vlan internal allocation policy
- show vlan internal usage
- show vlan summary
- show vlan trunk group

autostate

When autostate is ***enabled***, the VLAN interface will be up when:

- the corresponding VLAN exists and is in the active state.
- one or more layer 2 ports in the VLAN are up and in spanning-tree forwarding state.
- the VLAN interface exists and is not in a **shutdown** state.

Autostate is ***enabled*** by default. When autostate is ***disabled***, the VLAN interface is forced to be active.

- The **no autostate** command disables autostate on the configuration mode interface. The **no autostate** command is stored to ***running-config***.
- The **autostate** command enables the autostate function on the configuration mode VLAN SVI by removing the corresponding **no autostate** statement from ***running-config***.
- The **default autostate** command restores the autostate default state of ***enabled*** by removing the corresponding **no autostate** statement from ***running-config***.

Command Mode

Interface-VLAN Configuration

Command Syntax

```
autostate
no autostate
default autostate
```

Guidelines

Autostate should be disabled on SVIs configured as an MLAG local interface.

Examples

- These commands disable autostate on VLAN 100.

```
switch(config)#interface vlan 100
switch(config-if-Vl100)#no autostate
switch(config-if-Vl100)#
```

- These commands enable autostate on VLAN 100.

```
switch(config)#interface vlan 100
switch(config-if-Vl100)#autostate
switch(config-if-Vl100)#
```

encapsulation dot1q vlan

Routed Port VLAN Translation

In the configuration mode for an Ethernet or port channel interface, the **encapsulation dot1q vlan** translates packets with a dot1q header to the internal VLAN for a routed port. The VLAN in the incoming packets is mapped to the internal VLAN of the routed port, and packets egressing the routed port are encapsulated with a dot1q header for the specified VLAN. For egress packets, no priority information is added to the dot1q header and the priority from the incoming encapsulation will be retained.

Subinterface VLAN Assignment

When used in the configuration mode for an Ethernet or port channel subinterface, however, the **encapsulation dot1q vlan** command assigns a dot1q tag to the subinterface. Traffic ingressing on the parent interface with that dot1q tag will then be sent to the configured subinterface. See **Subinterfaces** and **Subinterface Configuration** for details.

The **no encapsulation dot1q vlan** and **default encapsulation dot1q vlan** commands restore the default VLAN to the configuration mode interface by removing the corresponding **encapsulation dot1q vlan** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration
Interface-port-channel Configuration
Subinterface-Ethernet Configuration
Subinterface-port-channel Configuration

Command Syntax

```
encapsulation dot1q vlan vlan_id
no encapsulation dot1q vlan
default encapsulation dot1q vlan
```

Parameters

- *vlan_id* For VLAN translation, the ID of the external VLAN to be translated; for subinterface configuration, the VLAN of the subinterface. Values range from 1 to 4094.

Example

- These commands translate between VLAN 50 and the internal VLAN for Ethernet interface 5 (a routed port).

```
switch(config)#interface ethernet 5
switch(config-if-Et5)# no switchport
switch(config-if-Et5)# encapsulation dot1q vlan 50
switch(config-if-Et5)#
```

- These commands assign packets ingressing on Ethernet interface 1/1 with VLAN ID 100 to Ethernet subinterface 1/1.1.

```
switch(config)#interface ethernet1/1.1
switch(config-if-Et1/1.1)# no switchport
switch(config-if-Et1/1.1)#encapsulation dot1q vlan 100
switch(config-if-Et1/1.1)#
```

interface vlan

The **interface vlan** command places the switch in VLAN-interface configuration mode for modifying parameters of the switch virtual interface (SVI). An SVI provides Layer 3 processing for packets from all ports associated with the VLAN. There is no physical interface for the VLAN.

When entering configuration mode to modify existing SVIs, the command can specify multiple interfaces. The command creates an SVI if the specified interface does not exist prior to issuing the command. When creating an SVI, the command can only specify a single interface.

The **no interface vlan** command deletes the specified SVI interfaces from ***running-config***. The **default interface vlan** commands remove all configuration statements for the specified SVI interfaces from ***running-config*** without deleting the interfaces.

Command Mode

Global Configuration

Command Syntax

```
interface vlan v_range
no interface vlan v_range
default interface vlan v_range
```

Parameter

- *v_range* VLAN interfaces (number, range, or comma-delimited list of numbers and ranges).

  VLAN number ranges from 1 to 4094.

Restrictions

Internal VLANs: A VLAN interface cannot be created or configured for internal VLAN IDs. The switch rejects any **interface vlan** command that specifies an internal VLAN ID.

Example

- This example creates an SVI for VLAN 12:

```
switch#config
switch(config)#interface vlan 12
switch(config-if-Vl12)#
```

l2-protocol encapsulation dot1q vlan

The **l2-protocol encapsulation dot1q vlan** command enables Layer 2 802.1Q traffic encapsulation on the configuration mode interface for a specified VLAN. The default VLAN for all interfaces is VLAN 1.

The **no l2-protocol encapsulation dot1q vlan** and **default l2-protocol encapsulation dot1q vlan** commands disable the specified encapsulation on the configuration mode interface by removing the corresponding **l2-protocol encapsulation dot1q vlan** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration
Interface-Port-channel Configuration

Command Syntax

```
l2-protocol encapsulation dot1q vlan vlan_id
no l2-protocol encapsulation dot1q vlan
default l2-protocol encapsulation dot1q vlan
```

Parameters

- *vlan_id* the ID of the native VLAN. Values range from 1 to 4094.

Example

- These commands enable 802.1Q encapsulation of traffic on VLAN 200.

```
switch(config)#interface ethernet 5/2
switch(config-if-Et5/2)#l2-protocol encapsulation dot1q vlan 200
switch(s1)(config-if-Et5/2)#show active
interface Ethernet5/2
   l2-protocol encapsulation dot1q vlan 200
switch(config-if-Et5/2)#
```

name (VLAN configuration mode)

The **name** command configures the VLAN name. The name can have up to 32 characters. The default name for VLAN 1 is ***default***. The default name for all other VLANs is VLANxxxx, where xxxx is the VLAN number. The default name for VLAN 55 is VLAN0055. The **show vlan** command displays the VLAN name.

The **name** command accepts all characters except the space.

The **no name** and **default name** commands restore the default name by removing the **name** command from ***running-config***.

Command Mode

VLAN Configuration

Command Syntax

name *label_text*
no name
default name

Parameters

- *label_text* character string assigned to name attribute. Maximum length is 32 characters. The space character is not permitted in the name string.

Examples

- These commands assign corporate_100 as the name for VLAN 25, then displays the VLAN name.

```
switch(config)#vlan 25
switch(config-vlan-25)#name corporate_100
switch(config-vlan-25)#show vlan 25
VLAN  Name                             Status    Ports
----- -------------------------------- --------- --------------------------------
25    corporate_100                    active

switch(config-vlan-25)#
```

show dot1q-tunnel

The **show dot1q-tunnel** command displays the ports that are configured in dot1q-tunnel switching mode. The **switchport mode** command configures the switching mode for the configuration mode interface.

Command Mode

EXEC

Command Syntax

show dot1q-tunnel [*INTERFACE*]

Parameters

- ***INTERFACE*** Interface type and numbers. Options include:
  - <no parameter> Display information for all interfaces.
  - **ethernet** *e_range* Ethernet interface range specified by *e_range*.
  - **loopback** *l_range* Loopback interface specified by *l_range*.
  - **management** *m_range* Management interface range specified by *m_range*.
  - **port-channel** *p_range* Port-Channel Interface range specified by *p_range*.
  - **vlan** *v_range* VLAN interface range specified by *v_range*.
  - **vxlan** *vx_range* VXLAN interface range specified by *vx_range*.

    Valid *range* formats include number, number range, or comma-delimited list of numbers and ranges.

Example

- This command displays the ports that are configured in dot1q-tunnel switching mode.

```
switch>show dot1q-tunnel
dot1q-tunnel mode LAN Port (s)
------------------------------
Po4
Po21
Po22
switch>
```

show interfaces switchport

The **show interfaces switchport** command displays the switching configuration and operational status of the specified ports.

Command Mode

EXEC

Command Syntax

```
show interfaces [INTERFACE] switchport
```

Parameters

- ***INTERFACE*** Interface type and numbers. Options include:
  - <no parameter> Display the switching status for all interfaces.
  - **ethernet** *e_range* Ethernet interface range specified by *e_range*.
  - **loopback** *l_range* Loopback interface specified by *l_range*.
  - **management** *m_range* Management interface range specified by *m_range*.
  - **port-channel** *p_range* Port-Channel Interface range specified by *p_range*.
  - **vlan** *v_range* VLAN interface range specified by *v_range*.

    Valid *e_range*, *l_range*, *m_range*, *p_range*, and *v_range* formats include number, number range, or comma-delimited list of numbers and ranges.

Example

- This command displays the switching status for all interfaces.

```
switch(config)#show interface switchport
Default switchport mode: access

Name: Et5/1
Switchport: Enabled
Administrative Mode: static access
Operational Mode: static access
MAC Address Learning: enabled
Access Mode VLAN: 1 (default)
Trunking Native Mode VLAN: 1 (default)
Administrative Native VLAN tagging: disabled
Trunking VLANs Enabled: ALL
Static Trunk Groups:
Dynamic Trunk Groups:

Name: Et5/2
Switchport: Enabled
Administrative Mode: static access
Operational Mode: static access
MAC Address Learning: enabled
Access Mode VLAN: 1 (default)
Trunking Native Mode VLAN: 1 (default)
Administrative Native VLAN tagging: disabled
Trunking VLANs Enabled: ALL
Static Trunk Groups:
Dynamic Trunk Groups:

[...]

switch(config)#
```

- This command displays the switching status of port channel interfaces 21 and 22.

```
switch>show interface port-channel 21-22 switchport
Name: Po21
Switchport: Enabled
Administrative Mode: tunnel
Operational Mode: tunnel
Access Mode VLAN: 1 (inactive)
Trunking Native Mode VLAN: 100 (VLAN0100)
Administrative Native VLAN tagging: disabled
Trunking VLANs Enabled: ALL
Trunk Groups: foo

Name: Po22
Switchport: Enabled
Administrative Mode: tunnel
Operational Mode: tunnel
Access Mode VLAN: 1 (inactive)
Trunking Native Mode VLAN: 1 (inactive)
Administrative Native VLAN tagging: disabled
Trunking VLANs Enabled: ALL
Trunk Groups:

switch>
```

show interfaces switchport backup

The **show interfaces switchport backup** command displays interfaces that are configured as switchport backup pairs and the operational status of each interface. For each pair, the command displays the names, roles, status, and VLAN traffic of each interface.

Command Mode

EXEC

Command Syntax

show interfaces [*INTERFACE*] switchport backup

Parameters

- ***INTERFACE*** Interface type and numbers. Options include:
  - <no parameter> Display information for all interfaces.
  - **ethernet** *e_range* Ethernet interface range specified by *e_range*.
  - **loopback** *l_range* Loopback interface specified by *l_range*.
  - **management** *m_range* Management interface range specified by *m_range*.
  - **port-channel** *p_range* Port-Channel Interface range specified by *p_range*.
  - **vlan** *v_range* VLAN interface range specified by *v_range*.

    Valid *e_range*, *l_range*, *m_range*, *p_range*, and *v_range* formats include number, number range, or comma-delimited list of numbers and ranges.

Display Values

- **State** Operational status of the interface. Values include:
  - ***Up*** Spanning tree mode is *backup*, interface status is ***up***.
  - ***Down*** Spanning tree mode is ***backup***, interface status is ***down***.
  - ***Inactive Configuration*** The spanning tree mode is not ***backup***.
- **Forwarding vlans** VLANs forwarded by the interface. Depends on interface operation status and prefer option specified by the **switchport backup** command.

Example

- This command displays the configured switchport primary-backup pairs.

```
switch>show interfaces switchport backup
Switch backup interface pair: Ethernet17, Ethernet18
Primary Interface: Ethernet17       State:  Up
Backup Interface:  Ethernet18       State:  Up
Ethernet17 forwarding vlans:  1-20
Ethernet18 forwarding vlans:
```

show interfaces trunk

The **show interfaces trunk** command displays configuration and status information for interfaces configured in switchport trunk mode.

Command Mode

EXEC

Command Syntax

show interfaces [*INTERFACE*] trunk

Parameters

- ***INTERFACE*** Interface type and numbers. Options include:
  - <no parameter> Display information for all interfaces.
  - **ethernet** *e_range* Ethernet interface range specified by *e_range*.
  - **management** *m_range* Management interface range specified by *m_range*.
  - **port-channel** *p_range* Port-Channel Interface range specified by *p_range*.

    Valid *e_range*, *m_range*, and *p_range* formats include number, number range, or comma-delimited list of numbers and ranges.

Example

- This command displays the trunk status for all interfaces configured in switchport trunk mode.

```
switch>show interfaces trunk
Port            Mode            Status          Native vlan
Po1             trunk           trunking        1
Po2             trunk           trunking        1

Port            Vlans allowed
Po1             1-15
Po2             16-30

Port            Vlans allowed and active in management domain
Po1             1-10
Po2             21-30


Port            Vlans in spanning tree forwarding state
Po1             1-10
Po2             21-30

switch>
```

show interfaces vlans

The **show interfaces vlans** command displays a table that lists the VLANs that are carried by the specified interfaces. Interfaces that do not carry VLANs are not listed in the table. The table lists the untagged (native or access) and tagged VLANs for each interface.

Command Mode

EXEC

Command Syntax

```
show interfaces [INT_NAME] vlans
```

Parameters

- ***INT_NAME*** Interface type and number. Values include
  - **ethernet** *e_num* Ethernet interface specified by *e_num*.
  - **management** *m_num* Management interface specified by *m_num*.
  - **port-channel** *p_num* Port-Channel Interface specified by *p_num*.

Example

- This command displays the VLANs carried by all L2 ports.

```
switch>show interfaces vlans
Port       Untagged Tagged
Et9        3910     -
Et11       3912     -
Et16       500      -
Et17       3908     -
Et18       3908     -
Po1        1        101-102,500,721,3000,
Po2        101      -
Po4        3902     -
Po5        3903     -
Po6        3992     -
Po7        661      -
Po8        3911     -
```

show vlan

The **show vlan** command displays the VLAN ID, name, status, and member ports of all configured VLANs. The command only displays active ports by default; by specifying ***configured-ports***, the command displays all ports that are members of a configured VLAN regardless of their activity status, including Ethernet ports that are members of a port channel.

Command Mode

EXEC

Command Syntax

show vlan [*VLAN_LIST*] [*PORT_ACTIVITY*]

Parameters

- ***VLAN_LIST*** List of VLANs displayed by command. Options include:
  - <no parameter> all VLANs.
  - *v_range* VLANs specified by *v_range*.
  - **id** *v_range* VLANs specified by *v_range*.
  - **name** *v_name* VLANs specified by the VLAN name *v_name*.

  v_range formats include number, number range, or comma-delimited list of numbers and ranges.
- ***PORT_ACTIVITY*** Ports listed in table. Options include:
  - <no parameter> table displays only active ports (same as **active-configuration** option).
  - **active-configuration** table displays only active ports.
  - **configured-ports** table displays all configured ports.

Display Values

- **VLAN** The VLAN ID.
- **Name** The name of the VLAN.
- **Status** The status of the VLAN.
- **Ports** The ports that are members of the VLAN.

Example

- This command displays status and ports of VLANs 1-1000.

```
switch>show vlan 1-1000
VLAN  Name                             Status    Ports
----- -------------------------------- --------- -------------------------------
1     default                          active    Po1
184   fet.arka                         active    Cpu, Po1, Po2
262   mgq.net                          active    PPo2, Po1
512   sant.test                        active    Cpu, Et16, Po1
821   ipv6.net                         active    Cpu, Po1, Po7

switch>
```

show vlan dynamic

The **show vlan dynamic** command displays the source and quantity of dynamic VLANs on the switch. Dynamic VLANs support VM Tracer monitoring sessions.

Command Mode

EXEC

Command Syntax

show vlan dynamic

Example

- This command displays the source and quantity of dynamic VLANs on the switch.

```
switch>show vlan dynamic
Dynamic VLAN source       VLANS
vmtracer-poc              88
switch>
```

show vlan internal allocation policy

The **show vlan internal allocation policy** command displays the method the switch uses to allocate VLANs to routed ports. The **vlan internal allocation policy** command configures the allocation method.

The allocation method consists of two configurable components:

- range: the list of VLANs that are allocated to routed ports.
- direction: the direction by which VLANs are allocated (ascending or descending).

Command Mode

EXEC

Command Syntax

show vlan internal allocation policy

Example

- This command displays the internal allocation policy.

```
switch>show vlan internal allocation policy
Internal VLAN Allocation Policy: ascending
Internal VLAN Allocation Range: 1006-4094
switch>
```

show vlan internal usage

The **show vlan internal usage** command shows the VLANs that are allocated as internal VLANs for routed ports.

A routed port is an Ethernet or port channel interface that is configured as a layer 3 interface. Routed ports do not bridge frames and are not members of any VLANs. Routed ports can have IP addresses assigned to them and packets are routed directly to and from the port.

When an interface is configured as a routed port, the switch allocates an SVI with a previously unused VLAN ID. The switch prohibits the configuration of VLANs with numbers corresponding to internal VLAN interfaces allocated to a routed port. VLAN interfaces corresponding to SVIs allocated to a routed port cannot be configured by VLAN interface configuration mode commands.

Command Mode

EXEC

Command Syntax

```
show vlan internal usage
```

Example

- This command displays the VLANs that are allocated to routed ports.

```
switch>show vlan internal usage
1006  Ethernet3
1007  Ethernet4
switch>
```

show vlan summary

The **show vlan summary** command displays the number of VLANs that are configured on the switch.

Command Mode

EXEC

Command Syntax

show vlan summary

Example

- This command displays the number of VLANs on the switch.

```
switch>show vlan summary
Number of existing VLANs           : 18

switch>
```

show vlan trunk group

The **show vlan trunk group** command displays the trunk group membership of the specified VLANs.

Command Mode

EXEC

Command Syntax

show vlan [*VLAN_LIST*] trunk group

Parameters

- ***VLAN_LIST*** VLAN list. Options include:
  - <no parameter> all VLANs.
  - *v_range* VLANs specified by *v_range*.
  - **id** *v_range* VLANs specified by *v_range*.
  - **name** *v_name* VLANs specified by the VLAN name *v_name*.

Display Values

- **VLAN** VLAN ID.
- **Trunk Group**s Trunk groups associated with the listed VLANs.

Example

- This command displays the trunk group membership of all configured VLANs.

```
switch>show vlan trunk group
VLAN     Trunk Groups
----     --------------------------------------------------------------------
5
10       first_group
12
40       second_group
100      third_group
101      middle_group
102
200

switch>
```

state

The **state** command configures the VLAN transmission state of the configuration mode VLAN.

- **Active** state: Ports forward VLAN traffic.
- **Suspend** state: Ports block VLAN traffic.

The default transmission status is ***active***.

The **no state** command restores the default VLAN transmission state to the configuration mode VLAN by removing the corresponding **state** command from ***running-config***.

Command Mode

VLAN Configuration

Command Syntax

```
state OPERATION_STATE
no state
default state
```

Parameters

- ***OPERATION_STATE*** VLAN transmission state. Options include:
  - **active** VLAN traffic is forwarded
  - **suspend** LAN traffic is blocked.

Example

- These commands suspend VLAN traffic on VLANs 100-102.

```
switch(config)#vlan 100-102
switch(config-vlan-100-102)#state suspend
switch(config-vlan-100-102)#
```

switchport dot1q ethertype

The **switchport dot1q ethertype** command configures the tag protocol identifier (TPID, also known as a dot1q ethertype), of the configuration mode interface. By default, all switch ports use the standard TPID of 0x8100.

The **no switchport dot1q ethertype** and **default switchport dot1q ethertype** commands restore the TPID to 0x8100 by removing the corresponding **switchport dot1q ethertype** statement from ***running-config***.

Command Mode

Interface-Ethernet Configuration

Command Syntax

```
switchport dot1q ethertype ethertype
no switchport dot1q ethertype
default switchport dot1q ethertype
```

Parameters

- *ethertype* ethertype number (TPID). Value ranges from 0x600 (1536) through 0xFFFF (65535), and can be entered in decimal or hexadecimal notation. Value is stored and displayed in hexadecimal form; the default value is 0x8100.

Example

- These commands configure 0x9100 as the TPID of Ethernet interface 5.

```
switch(config)#interface ethernet 5
switch(config-if-Et5)#switchport dot1q ethertype 0x9100
switch(config-if-Et5)#
```

switchport access vlan

The **switchport access vlan** command specifies the access VLAN of the configuration mode interface. Ethernet or port channel interfaces that are in access mode are members of only the access VLAN. Untagged frames that the interface receives are associated with the access VLAN. Frames tagged with the access VLAN are also associated with the access VLAN. The interface drops all other tagged frames that it receives. By default, VLAN 1 is the access VLAN of all Ethernet and port channel interfaces.

An interface's access mode is effective only when the interface is in access mode or dot1q-tunnel mode, as specified by the switchport mode command. Interfaces in dot1q-tunnel mode handle inbound traffic as untagged traffic and associate all traffic with the access VLAN. Interfaces configured to switchport trunk mode maintain and ignore existing switchport access commands.

The **no switchport access vlan** and **default switchport access vlan** commands restore VLAN 1 as the access VLAN of the configuration mode interface by removing the corresponding **switchport access vlan** statement from ***running-config***.

Command Mode

Interface-Ethernet Configuration
Interface-Port-channel Configuration

Command Syntax

```
switchport access vlan v_num
no switchport access vlan
default switchport access vlan
```

Parameters

- *v_num* number of access VLAN. Value ranges from 1 to 4094. Default is 1.

Example

- These commands assign VLAN 100 as the access VLAN to Ethernet interface 5.

```
switch(config)#interface ethernet 5
switch(config-if-Et5)#switchport access vlan 100
switch(config-if-Et5)#
```

switchport mode

The **switchport mode** command specifies the switching mode of the configuration mode interface. The switch supports five switching modes: access, trunk, dot1q-tunnel, tap, and tool.

- **Access switching mode**: The interface is a member of one VLAN, called the access VLAN, as specified by the **switchport access vlan** command. Tagged frames received on the interface are dropped unless they are tagged with the access VLAN. Frames transmitted from the interface are always untagged.
- **Trunk switching mode**: The interface may be a member of multiple VLANs, as configured by the **switchport trunk allowed vlan** command. Untagged traffic is associated with the interface's native VLAN, as configured with the **switchport trunk native vlan** command.
- **Dot1q-tunnel switching mode**: The interface treats all inbound packets as untagged traffic and handles them as traffic of its access VLAN, as specified by the **switchport access vlan** command.
- **Tap mode**: The interface operates as a tap port. Tap ports receive traffic for replication on one or more tool ports.The interface may be a member of multiple VLANs, as configured by the **switchport tap allowed vlan** command. Untagged traffic is associated with the interface's native VLAN, as configured with the **switchport tap native vlan** command.

  Tap ports are in STP forwarding state and prohibit egress traffic. MAC learning, control plane interaction and traps for inbound traffic are disabled.
- **Tool mode**: The interface operates as a tool port. Tool ports replicate traffic received by tap ports. The interface may be a member of multiple VLANs, as configured by the **switchport tool allowed vlan** command. MAC learning, control plane interaction and traps for inbound traffic are disabled.

  Tool ports are in STP forwarding state and prohibit ingress traffic that uses port settings.

The status of switchport configured ports depends on the switch's tap aggregation mode (which can be viewed by using the **mode (tap-agg configuration mode)** command):

- tap aggregation mode enabled: tap and tool ports are enabled. Switching ports are errdisabled.
- tap aggregation mode disabled: tap and tool ports are errdisabled. Switching ports are enabled.

The **no switchport mode** and **default switchport mode** commands return the configuration mode interface to its default setting as an access port by deleting the corresponding **switchport mode** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration
Interface-Port-channel Configuration

Command Syntax

```
switchport mode MODE_TYPE
no switchport mode
default switchport mode
```

Parameters

- ***MODE_TYPE*** switching mode of the configuration mode interfaces. Options include:
  - **access** access switching mode.
  - **dot1q-tunnel** dot1q-tunnel switching mode.
  - **tap** tap switching mode.
  - **tool** tool switching mode.
  - **trunk** trunk switching mode.

Restrictions

Dot1q-tunnel switching mode is not available on Petra platform switches.

Tap aggregation (tap and tool modes) is available on FM6000 and Arad platform switches.

Example

- These commands configure Ethernet 4 interface as a trunk port.

```
switch(config)#interface ethernet 4
switch(config-if-Et4)#switchport mode trunk
switch(config-if-Et4)#
```

switchport trunk allowed vlan

The **switchport trunk allowed vlan** command creates or modifies the list of VLANs for which the configuration mode interface, in trunk mode, handles tagged traffic. By default, interfaces handle tagged traffic for all VLANs. Command settings persist in ***running-config*** without taking effect when the switch is in tap aggregation mode or the interface is not in trunk mode.

The **no switchport trunk allowed vlan** and **default switchport trunk allowed vlan** commands restore the trunk mode default allowed VLAN setting of ***all*** by removing the corresponding **switchport trunk allowed vlan** statement from ***running-config***.

Command Mode

Interface-Ethernet Configuration
Interface-Port-channel Configuration

Command Syntax

```
switchport trunk allowed vlan EDIT_ACTION
no switchport trunk allowed vlan
default switchport trunk allowed vlan
```

Parameters

- ***EDIT_ACTION*** modifications to the VLAN list.
  - *v_range* Creates VLAN list from *v_range*.
  - **add** *v_range* Adds specified VLANs to current list.
  - **all** VLAN list contains all VLANs.
  - **except** *v_range* VLAN list contains all VLANs except those specified.
  - **none** VLAN list is empty (no VLANs).
  - **remove** *v_range* Removes specified VLANs from current list.

  Valid *v_range* formats include number, range, or comma-delimited list of numbers and ranges.

Example

- These commands create the trunk mode allowed VLAN list of 6-10 for Ethernet interface 14, then verifies the VLAN list.

```
switch(config)#interface ethernet 14
switch(config-if-Et14)#switchport trunk allowed vlan 6-10
switch(config-if-Et14)#show interfaces ethernet 14 switchport
Name: Et14
Switchport: Enabled
Administrative Mode: trunk
Operational Mode: trunk
Access Mode VLAN: 1 (inactive)
Trunking Native Mode VLAN: 1 (inactive)
Administrative Native VLAN tagging: disabled
Trunking VLANs Enabled: 6-10
Trunk Groups:

switch(config-if-Et14)#
```

switchport trunk group

The **switchport trunk group** command assigns the configuration mode interface to the specified trunk group. Trunk group ports handle traffic of the VLANs assigned to the group.

The **no switchport trunk group** and **default switchport trunk group** commands remove the configuration mode interface from the specified trunk group by deleting the corresponding statement from ***running-config***. If the command does not specify a trunk group, the interface is removed from all trunk groups to which it is assigned.

Note On platforms which support the use of port channels as mirror destinations, a port channel which is being used as a mirror destination *must not* be assigned to an MLAG.

Command Mode

Interface-Ethernet Configuration
Interface-Port-channel Configuration

Command Syntax

```
switchport trunk group group_name
no switchport trunk group [group_name]
default switchport trunk group [group_name]
```

Parameters

- *group_name* trunk group name.

Example

- These commands assign port channel 4 to trunk group ***fe-1***.

```
switch(config)#interface port-channel 4
switch(config-if-Po4)#switchport trunk group fe-1
switch(config-if-Po4)#
```

switchport trunk native vlan

The **switchport trunk native vlan** command specifies the trunk mode native VLAN for the configuration mode interface. Interfaces in trunk mode associate untagged frames with the native VLAN. Trunk mode interfaces can also be configured to drop untagged frames. The default native VLAN for all interfaces is VLAN 1.

The **no switchport trunk native vlan** and **default switchport trunk native vlan** commands restore VLAN 1 as the trunk mode native VLAN to the configuration mode interface by removing the corresponding **switchport trunk native vlan** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration
Interface-Port-channel Configuration

Command Syntax

```
switchport trunk native vlan VLAN_ID
no switchport trunk native vlan
default switchport trunk native vlan
```

Parameters

- ***VLAN_ID*** the ID of the native VLAN. Options include
  - v_num VLAN number. Value ranges from 1 to 4094
  - **tag** interface drops all untagged frames.

Example

- These commands configure VLAN 100 as the native VLAN for port channel 21.

```
switch(config)#interface port-channel 21
switch(config-if-Po21)#switchport trunk native vlan 100
switch(config-if-Po21)#
```

switchport vlan mapping

The **switchport vlan mapping** command allows you to map packets from one VLAN to another using VLAN translation. This can be carried out only on packets having a dot1q header (tagged frames). The translation rewrites the VID field (VLAND ID) in dot1q headers on packets passing through a switched port without changing any other fields.

By default, the translation is bidirectional: packets ingressing an interface through VLAN A are internally mapped to VLAN B; VLAN B packets egressing the same interface are mapped to VLAN A.

To use VLAN translation on a switched port, the port must be configured as a trunk port using the **switchport mode** command.

VLAN translation on routed ports is accomplished through the **encapsulation dot1q vlan** command.

The **no switchport vlan mapping** and **default switchport vlan mapping** commands remove VLAN mapping by removing the switchport vlan mapping command from ***running-config***.

Command Mode

Interface-Ethernet Configuration
Interface-Port-channel Configuration

Command Syntax

```
switchport vlan mapping [DIRECTION] incoming_vlanid new_vlanid
no switchport vlan mapping incoming_vlanid new_vlanid
no switchport vlan mapping DIRECTION incoming_vlanid
default switchport vlan mapping incoming_vlanid new_vlanid
default switchport vlan mapping DIRECTION incoming_vlanid
```

Parameters

- ***DIRECTION*** transmission direction of traffic to be translated.
  - <no parameter> translates the specified VLAN IDs for transmitted and received traffic.
  - **in** translates the specified VLAN IDs for received traffic only.
  - **out** translates the specified VLAN IDs for transmitted traffic only.
- *incoming_vlanid* The VLAN ID to be translated. Value ranges from 1 to 4094.
- *new_vlanid* The new VLAN ID or bridging VLAN ID which will be used internally. Value ranges from 1 to 4094.

Example

- These commands translate only incoming packets, changing the VID to 2008 in the dot1q header of packets ingressing on VLAN 201.

```
switch(config)# interface ethernet 5
switch(config-if-Et5)# switchport vlan mapping in 201 2008
switch(config-if-Et5)#
```

- These commands translate multiple VLAN mappings under an interface.

```
switch(config)#interface ethernet 5
switch(config-if-Et5)# switchport vlan mapping 50 60
switch(config-if-Et5)# switchport vlan mapping 61 71
switch(config-if-Et5)# switchport vlan mapping 62 72
switch(config-if-Et5)#
```

trunk group

The **trunk group** command assigns the configuration mode VLAN to a specified trunk group.

A trunk group is the set of physical interfaces that comprise the trunk and the collection of VLANs whose traffic is carried on the trunk. The traffic of a VLAN that belongs to one or more trunk groups is carried only on ports that are members of trunk groups to which the VLAN belongs. Switchport commands specify the physical interfaces that carry trunk group traffic.

The **no trunk group** and **default trunk group** commands remove the configuration mode VLAN from the specified trunk group by removing the corresponding **trunk group** statement from ***running-config***. If a trunk group is not specified, the commands remove the configuration mode VLAN from all trunk groups.

Command Mode

VLAN Configuration

Command Syntax

```
trunk group name
no trunk group [name]
default trunk group [name]
```

Parameters

- *name* a name representing the trunk group.

Example

- These commands assigns VLAN 49 to the trunk group ***mlagpeer***:

```
switch(config)#vlan 49
switch(config-vlan-49)#trunk group mlagpeer
switch(config-vlan-49)#
```

vlan

The **vlan** command places the switch in VLAN configuration mode to configure a set of virtual LANs. The command creates the specified VLANs if they do not exist prior to issuing the command. A VLAN that is in use as an internal VLAN may not be created or configured. The switch rejects any **vlan** command that specifies an internal VLAN ID.

The **default vlan** and **no vlan** commands removes the VLAN statements from ***running-config*** for the specified VLANs.

The **exit** command returns the switch to global configuration mode.

Command Mode

Global Configuration

Command Syntax

```
vlan vlan_range
no vlan vlan_range
default vlan vlan_range
```

Parameters

- *vlan_range* VLAN list.

  Formats include a name, number, number range, or comma-delimited list of numbers and ranges.

Commands Available in VLAN configuration mode

- **name (VLAN configuration mode)**
- **state**
- **trunk group**

Guidelines

In MLAG configurations, VLANs operate as follows:

- The VLAN must be configured identically on both MLAG peer switches.
- The port-specific bridging configuration originates on the switch where the port is physically located. This configuration includes the switchport access VLAN, switchport mode (trunk or access), trunk-allowed VLANs, the trunk native VLAN, and the switchport trunk groups.

Example

- This command creates VLAN 49 and enters VLAN configuration mode for the new VLAN:

```
switch(config)#vlan 49
switch(config-vlan-49)#
```

vlan internal allocation policy

The **vlan internal allocation policy** command specifies the range that the switch can allocate as internal VLANs when configuring routed ports and the order of their allocation. By default, the switch allocates VLANs in ascending order from VLAN 1006 to VLAN 4094.

The **no vlan internal allocation policy** and **default vlan internal allocation policy** commands revert the policy to its default.

Command Mode

Global Configuration

Command Syntax

```
vlan internal allocation policy DIRECTION [RANGE_VLAN]
no vlan internal allocation policy
default vlan internal allocation policy
```

Parameters

- ***DIRECTION*** VLAN allocation number direction. Options include:
  - *ascending* allocates internal VLANs from lower VLAN bound to upper VLAN bound.
  - *descending* allocates internal VLAN from upper VLAN bound to lower VLAN bound.
- ***RANGE_VLAN*** allocation range. Options include:
  - <no parameter> 1006 (lower bound) to 4094 (upper bound).
  - *range lower upper* specifies lower bound (*lower*) and upper bound (*upper*).

Examples

- This command configures the switch to allocate internal VLANS from 3000 through 3999.

```
switch(config)#vlan internal allocation policy ascending range 3000 3999
switch(config)#
```

- This command configures the switch to allocate internal VLANS from 4094 through 1006.

```
switch(config)#vlan internal allocation policy descending
switch(config)#
```

- This command configures the switch to allocate internal VLANS from 4094 down through 4000.

```
switch(config)#vlan internal allocation policy descending range 4000 4094
switch(config)#
```

- This command reverts the allocation policy to its default (***ascending***, between ***1006*** and ***4094***).

```
switch(config)#no vlan internal allocation policy
switch(config)#
```

Chapter 19

VXLAN

This chapter describes Arista's VXLAN implementation. Sections in this chapter include:

- Section 19.1: VXLAN Introduction
- Section 19.2: VXLAN Description
- Section 19.3: VXLAN Configuration
- Section 19.4: VXLAN Command Descriptions

19.1 VXLAN Introduction

Virtual Extensible LAN (VXLAN) is a networking technology that encapsulates MAC-based Layer 2 Ethernet frames within Layer 3 UDP packets to aggregate and tunnel multiple layer 2 networks across a Layer 3 infrastructure. VXLAN scales up to 16 million logical networks and supports layer 2 adjacency across IP networks. Multicast transmission architecture is used for broadcast, multicast, and unknown unicast traffic.

For a list of VXLAN feature support in a specific EOS release, consult the appropriate release notes here: https://www.arista.com/en/support/software-download.

For a list of VXLAN feature support by platform in the latest EOS release, see http://www.arista.com/support/supported-features

Note VxLAN and NAT cannot co-exist.

19.2 VXLAN Description

These sections describe VXLAN architecture, the data objects that comprise a VXLAN network, and process of bridging packets through a VXLAN network.

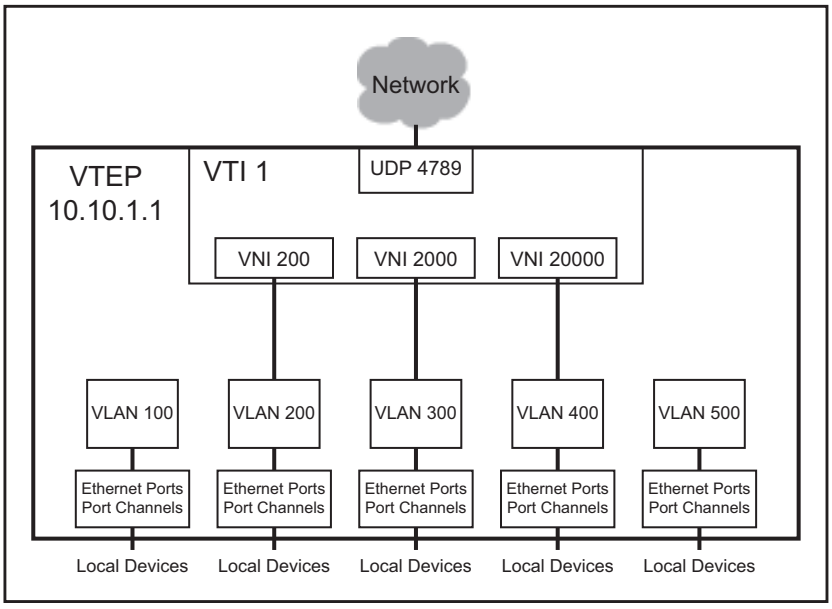
- Section 19.2.1: VXLAN Architecture
- Section 19.2.2: VXLAN Processes
- Section 19.2.3: Multicast and Broadcast over VXLAN
- Section 19.2.4: VXLAN Gateway
- Section 19.2.5: VXLAN and MLAG
- Section 19.2.6: Data Structures

19.2.1 VXLAN Architecture

The VXLAN architecture extends an L2 network by connecting VLANs from multiple hosts through UDP tunnels called VXLAN segments. VXLAN segments are identified by a 24-bit virtual network identifier (VNI). Within a host, each VLAN whose network is extended to other hosts is associated with a VNI. An extended L2 network comprises the devices attached to VLANs from all hosts that are on VLANs that are associated with the same VNI.

Figure 19-1 displays the data objects that comprise a VXLAN implementation on a local host.

Figure 19-1: VXLAN Architecture



- **VXLAN Tunnel End Point (VTEP):** a host with at least one VXLAN Tunnel Interface (VTI).
- **VXLAN Tunnel Interface (VTI):** a switchport linked to a UDP socket that is shared with VLANs on various hosts. Packets bridged from a VLAN to the VTI are sent out the UDP socket with a VXLAN header. Packets arriving on the VTI through the UDP socket are demuxed to VLANs for bridging.
- **Virtual Network Identifier (VNI):** a 24-bit number that distinguishes between the VLANs carried on a VTI. It facilitates the multiplexing of several VLANs over a single VTI.

  VNIs can be expressed in digital or dotted decimal formats. VNI values range from 1 to 16777215 or from 0.0.1 to 255.255.255.

The network in Figure 19-1 has the following assignments:

- VTEP IP address of 10.10.1.1
- UDP port of 4789
- One VTI that supports three VXLAN segments (UDP tunnels): VNI 200, VNI 2000, and VNI 20000
- Five VLANs, of which three VLANs can communicate with remote devices over Layer 2.

19.2.2 VXLAN Processes

When a packet enters a VLAN from a member (ingress) port, the VLAN learns the source address by adding an entry to the MAC address table that associates the source to the ingress-port. The VLAN then searches the table for destination address. If the MAC address table lists the address, the packet is sent out the corresponding port. If the MAC address table does not lists the address, the packet is flooded to all ports except the ingress port.

VXLANs extend VLANs through the addition of a VXLAN address table that correlates remote MAC addresses to their port and resident host IP address. Packets that are destined to a remote device are sent to the VXLAN tunnel interface (VTI), which is the switchport that is linked to the UDP socket. The packet is encapsulated with a VXLAN header which includes the VNI associated with the VLAN and the IP mapping of the destination host. The packet is sent through a UDP socket to the destination VTEP IP. The VTI on the remote host extracts the original packet and bridges it to the VLAN associated with the VNI on the remote host.

UDP port 4789 is recognized as the VXLAN socket and listed as the destination port on the UDP packets. The UDP source port field is filled with a hash of the inner header to facilitate load balancing.

Figure 19-2: VXLAN Implementation

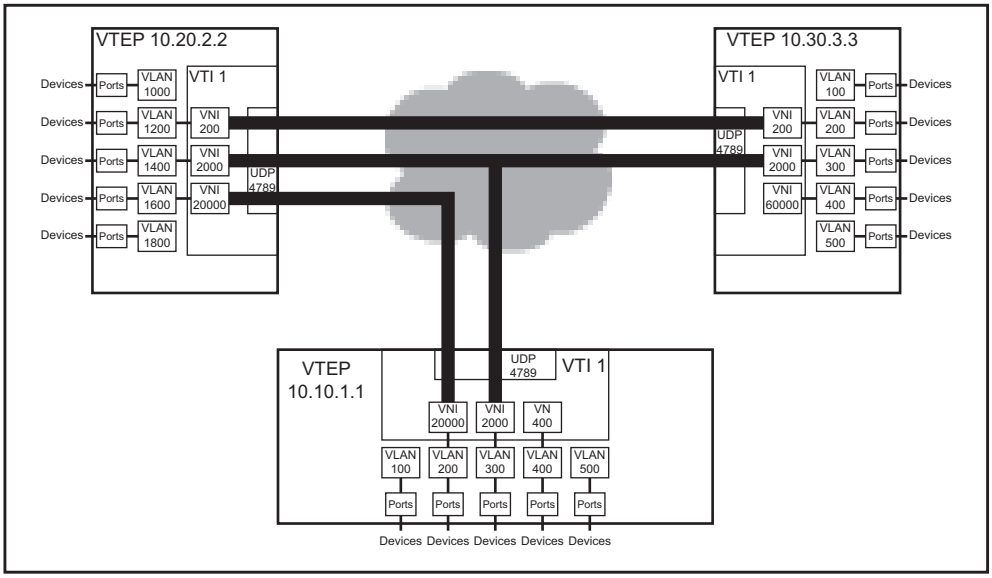


Figure 19-2 displays a configuration that includes three VTEPs. The VXLAN defines three inter-host L2 networks. The VLANs that comprise the networks include:

- VNI 200: VTEP 10.20.2.2: VLAN 1200 and VTEP 10.30.3.3: VLAN 200
- VNI 2000: VTEP 10.10.1.1: VLAN 300, VTEP 10.20.2.2: VLAN 1400, and VTEP 10.30.3.3: VLAN 300
- VNI 20000: VTEP 10.10.1.1: VLAN 200, and VTEP 10.20.2.2: VLAN 1600

VXLAN Routing

VXLAN routing is enabled by creating a VLAN interface on the VXLAN-enabled VLAN and assigning an IP address to the VLAN interface. The IP address serves as VXLAN gateway for devices that are accessible from the VXLAN-enabled VLAN.

19.2.3 Multicast and Broadcast over VXLAN

These sections describe multicast and broadcast over VXLANs. Multicast packet flooding describes broadcast and multicast transmission by associating a multicast group to a VTI through a configuration command. Head-end Replication (HER) optimizes flooding of inter VTEP broadcast, unknown unicast and broadcast (BUM) traffic by using hardware and flood lists to perform replication on the supported platform.

19.2.3.1 Multicast Packet Flooding

Multicast packet flooding is supported with VXLAN bridging without MLAG. A VTI is associated with a multicast group through a configuration command.

VXLAN and Broadcast

When a VLAN receives or sends a broadcast packet the VTI is treated as a bridging domain L2 interface. The packet is sent from this interface on the multicast group associated with the VTI. The VTIs on remote VTEPs that receive this packet extract the original packet, which is then handled by the VLAN associated with the packet's VNI. The VLAN floods the packet, excluding the VTI. When the broadcast results in a response, the resulting packet can be unicast back to the originating VTEP because the VXLAN address table obtained the host MAC to VTEP association from the broadcast packet.

VXLAN and Multicast

A VTI is treated as an L2 interface in the VLAN for handling multicast traffic, which is mapped from the VLAN to the multicast group associated with the VTI. All VTEPs join the configured multicast group for inter-VTEP communication within a VXLAN segment; this multicast group is independent of any other multicast groups that the hosts in the VLAN join.

The IP address space for the inter-host VXLAN communication may be sourced from a different VRF than the address space of the hosts in the VLAN. The multicast group for inter-VTEP transmissions must not be used for other purposes by any device in the VXLAN segment space.

19.2.3.2 Head-end Replication

Head-end replication uses a flood list to support broadcast, unknown unicast, and multicast (BUM) traffic over VXLAN. The flood list specifies a list of remote VTEPs. The switch replicates BUM data locally for bridging across the remote VTEPs specified by the flood list. This data flooding facilitates remote MAC address learning by forwarding data with unknown MAC addresses.

Head-end replication is required for VXLAN routing and to support VXLANs over MLAG.

19.2.4 VXLAN Gateway

A VXLAN gateway is a service that exchanges VXLAN data and packets with devices connected to different network segments. VXLAN traffic must pass through a VXLAN gateway to access services on physical devices in a distant network.

A VXLAN gateway requires the following information:

- An IP address that is designated as the VXLAN interface source.

- VLAN to VNI mapping.
- VTEP list for each VNI.
- A method for handling broadcast, unknown unicast, and multicast (BUM) packets.

Arista switches manually perform VXLAN gateway services. The switch connects to VXLAN gateways that serve other network segments. MAC address learning is performed in hardware from inbound VXLAN packets. BUM packets are supported though one of the methods specified in Section 19.2.3.

19.2.5 VXLAN and MLAG

The switch platforms which use this feature are:

- DCS-7050X

VXLAN over MLAG provides redundancy in hardware VTEPs. VTI configuration must be identical on each MLAG peer for them to act as a single VTEP. This also prevents the remote MAC from flapping between the remote VTEPs by ensuring that the rest of the network sees a host that is connected to the MLAG interface as residing behind a single VTEP.

Differences between VXLAN bridging and routing implementations over MLAG are applicable for the DCS-7050X series platform.

- VXLAN routing recirculates a packet twice, with the first iteration performing the routing action involving an L2 header rewrite, and the second recirculation performing VXLAN encap and decap operations. Recirculation is achieved by MAC loopback on dedicated loopback interfaces.
- The configuration for VXLAN routing on an MLAG VTEP includes separate Recirc-Channel configuration on both peers. The virtual IP, virtual MAC, and virtual VARP VTEP IP addresses are identical on both peers.

The following VTI elements must be configured identically on both MLAG peers:

- VLAN-VNI mappings
- VTEP IP address of the source loopback interface
- Flood VTEP list used for head-end replication

If OSPF is also in use, configure the OSPF router ID manually to prevent the switch from using the common VTEP IP address as the router ID.

The following rules are observed by MLAG switches so that they behave as a single VXLAN VTEP:

- Only the MLAG peer that receives a packet performs VXLAN encapsulation on it.
- Packets are not VXLAN encapsulated if they are received from the peer link.
- If a packet is decapsulated and sent over the peer link, it should not be flooded to active MLAG interfaces.
- If a packet is sent over the peer link to the CPU, it is not head-end replicated to other remote VTEPs.
- If a packet's destination is the VTEP IP address, it is terminated by the MLAG peer that receives it.

The following example is applicable to the DCS-7050X series platform.

Example

- These commands complete the configuration required for a VXLAN routing deployment.

```
switch(config)#interface Vxlan1
switch(config-if-Vx1)#vxlan source-interface Loopback0
switch(config-if-Vx1)#vxlan udp-port 4789
switch(config-if-Vx1)#vxlan vlan 2417 vni 8358534
switch(config-if-Vx1)#vxlan flood vtep 1.0.1.1 1.0.2.1
switch(config-if-Vx1)#interface Vlan2417
switch(config-if-Vl2417)#ip address 1.0.4.1/24
switch(config-if-Vl2417)#interface Loopback0
switch(config-if-Lo0)#ip address 1.0.1.1/32
switch(config-if-Lo0)#ip routing
switch(config)#interface Recirc-Channel627
switch(config-if-Re627)#switchport recirculation features vxlan
switch(config-if-Re627)#interface Ethernet 1
switch(config-if-Et1)#traffic-loopback source system device mac
switch(config-if-Et1)#channel-group recirculation 627
switch(config-if-Et1)#exit
switch(config)#interface Ethernet 2
switch(config-if-Et2)#traffic-loopback source system device mac
switch(config-if-Et2)#channel-group recirculation 627
switch(config-if-Et2)#
```

Configuring UnconnectedEthernet interfaces for recirculation

On systems where bandwidth is not fully used by the front panel ports, unused bandwidth is used for recirculation.

The following example is applicable to the DCS-7050X series platform.

Example

- These commands expose UnconnectedEthernet interfaces which are used for recirculation, in order to use them to replace or use along with front panel Ethernet interfaces.

```
switch(config)#service interface unconnected expose
switch(config)#interface UnconnectedEthernet 2
switch(config-if-Ue2)#traffic-loopback source system device mac
switch(config-if-Ue2)#channel-group recirculation 627
```

19.2.6 Data Structures

VXLAN implementation requires two VXLAN tables and a MAC address table accommodation.

19.2.6.1 MAC Address Table VXLAN Support

MAC address table entries correlate MAC addresses with the port upon which packets arrive. In addition to Ethernet and port channels, the port column may specify a VTI for packets that arrive on a VLAN from a remote port through the VXLAN segment.

19.2.6.2 VTEP-MAC Address Table

VTEP-MAC address table entries correlate MAC address with the IP address of the VTEP from where packets bearing the MAC address arrive. The VTI uses this table to determine the destination address for packets that are sent to remote hosts.

19.2.6.3 VNI-VLAN Map

The VNI-VLAN map displays the one-to-one correspondence between the VNIs assigned on the switch and the VLANs to which they are assigned. Each VNI can be assigned to only one VLAN; each VLAN can be assigned a maximum of one VNI. Each VNI-VLAN assignment constitutes a VXLAN segment.

19.3 VXLAN Configuration

These sections describe VXLAN configuration tasks:

- Section 19.3.1: Configuring the VTI
- Section 19.3.2: Head End Replication Configuration
- Section 19.3.3: VXLAN Routing Configuration
- Section 19.3.4: Configuring VXLAN Routing with Overlay VRFs
- Section 19.3.5: Configuring VXLAN over MLAG
- Section 19.3.6: Configuring VXLAN Control Service
- Section 19.3.7: Configuring VXLAN Multicast Decapsulation
- Section 19.3.8: VXLAN Rules Support for Mirror ACLs Configuration
- Section 19.3.9: Displaying VXLAN Configuration

19.3.1 Configuring the VTI

Configuring the VTI enables VXLAN bridging and is a requirement for VXLAN Routing. The following sections describe the steps required to enabling VXLAN bridging by bringing up the VXLAN line protocol. Section 19.3.3 describes the additional steps required to enable VXLAN routing.

Instantiating the VTI and VXLAN Configuration Mode

The **interface vxlan** command places the switch in VXLAN-interface configuration mode for modifying the specified VXLAN tunnel interface (VTI). The command also instantiates the interface if it was not previously created.

VXLAN interface configuration mode is not a group change mode; ***running-config*** is changed immediately after commands are executed. The **exit** command does not affect the configuration.

Example

- These commands create VXLAN tunnel interface 1, place the switch in VXLAN-interface configuration mode, and display parameters of the new VTI.

```
switch(config)#interface vxlan 1
switch(config-if-Vx1)#show active
interface Vxlan1
   vxlan udp-port 4789
switch(config-if-Vx1)#
```

Assigning an IP address to the VTEP

The **vxlan source-interface** command specifies the loopback interface from which the VTEP derives the source address (IP) that it uses when exchanging VXLAN frames. This address is used by UDP headers to specify source and destination addresses of hosts that send or receive VXLAN encapsulated packets.

There is no default source interface assignment. A valid VXLAN configuration requires the assignment of a loopback interface to the VTEP and the assignment of a valid IP address to the specified interface.

Example

- These commands configure VTI 1 to use IP address 10.25.25.3 (loopback interface 15) as the source interface in the encapsulation fields of outbound VXLAN frames.

```
switch(config)#interface loopback 15
switch(config-if-Lo15)#ip address 10.25.25.3/24
switch(config-if-Lo15)#exit
switch(config)#interface vxlan 1
switch(config-if-Vx1)#vxlan source-interface loopback 15
switch(config-if-Vx1)#show active
interface Vxlan1
   vxlan source-interface Loopback15
   vxlan udp-port 4789
switch(config-if-Vx1)#
```

Assigning a UDP Port to the VTEP

Packets bridged to the VTI from a VLAN are encapsulated with a VXLAN header, then sent through a pre-configured UDP port. Packets that arrive through this port are assumed to be VXLAN encapsulated and sent to the bridging domain of the recipient VLAN as determined by the VNI in the VXLAN header and the VNI-VLAN map.

The **vxlan udp-port** command associates a UDP port with the configuration mode VXLAN interface (VTI). By default, UDP port 4789 is associated with the VTI.

Important! UDP port 4789 is reserved by convention for VXLAN usage. Under most typical applications, this parameter should be set to the default value.

Example

- This command associates UDP port 5500 with VXLAN interface 1.

```
switch(config)#interface vxlan 1
switch(config-if-Vx1)#vxlan udp-port 5500
switch(config-if-Vx1)#show active
interface Vxlan1
   vxlan udp-port 5500
switch(config-if-Vx1)#
```

- This command resets the VXLAN interface 1 UDP port association of 4789.

```
switch(config-if-Vx1)#no vxlan udp-port
switch(config-if-Vx1)#show active
interface Vxlan1
   vxlan udp-port 4789
switch(config-if-Vx1)#
```

Assigning a VNI to a VLAN

When a VLAN bridges a packet to the VTI, the packet is encapsulated with a VXLAN header that includes the VNI associated with the VLAN. Packets that arrive on the VTI's UDP socket are bridged to the VLAN that is associated with the VNI specified by the VXLAN header that encapsulates the packet.

The VTI requires a one-to-one correspondence between specified VLANs and VNI values. Commands that assign a new VNI to a previously configured VLAN replace existing VLAN assignment statements in ***running-config***. Commands that attempt to assign a VNI value to a second VLAN generate a CLI error.

The **vxlan vlan vni** command associates a VLAN ID with a virtual network identifier (VNI).

Example

- These commands associate VLAN 100 to VNI 100 and VLAN 200 to VNI 10.10.200.

```
             switch(config)#interface vxlan 1
switch(config-if-Vx1)#vxlan vlan 100 vni 100
switch(config-if-Vx1)#vxlan vlan 200 vni 10.10.200
switch(config-if-Vx1)#show active
interface Vxlan1
   vxlan udp-port 4789
   vxlan vlan 200 vni 658120
   vxlan vlan 100 vni 100
switch(config-if-Vx1)#vxlan vni notation dotted
switch(config-if-Vx1)#show active
interface Vxlan1
   vxlan udp-port 4789
   vxlan vlan 100 vni 0.0.100
   vxlan vlan 200 vni 10.10.200
switch(config-if-Vx1)#
```

Assigning a Multicast Group to the VTI

The VTI maps multicast traffic from its associated VLANS to a specified multicast group. Inter-VTEP multicast communications include all VTEPs that are associated with the specified multicast group, which is independent of any other multicast groups that VLAN hosts may join.

The **vxlan multicast-group** command associates a specified multicast group with the configuration mode VXLAN interface (VTI), which handles multicast and broadcast traffic as a layer 2 interface in a bridging domain.

Example

- This command associates the multicast address of 227.10.1.1 with VTI 1.

```
switch(config)#interface vxlan 1
switch(config-if-Vx1)#vxlan multicast-group 227.10.1.1
switch(config-if-Vx1)#show active
interface Vxlan1
   vxlan multicast-group 227.10.1.1
   vxlan udp-port 4789
switch(config-if-Vx1)#
```

Verifying the VXLAN Configuration

The **show interface vxlan 1** displays the configuration and connection status of the VXLAN

Example

- This command indicates that the VXLAN line protocol status is ***up.***

```
switch(config-if-Vx1)#show interface vxlan 1
Vxlan1 is up, line protocol is up (connected)
  Hardware is Vxlan
  Source interface is Loopback15 and is active with 10.25.25.3
  Static vlan to vni mapping is
    [100, 0.0.100]    [200, 10.10.200]
  Multicast group address is 227.1.1.1
switch(config-if-Vx1)#
```

19.3.2 Head End Replication Configuration

Head-end replication is a data distribution method that supports broadcast, unknown unicast, and multicast (BUM) traffic over VXLANs by replicating BUM data locally for transmission to the set of remote VTEPs specified by a flood list. This data flooding facilitates remote MAC address learning through the forwarding of data with unknown MACs.

Each **vxlan flood vtep** statement in ***running-config*** associates a set of VTEP addresses to an access VNI. A default flood list is also configurable that applies to all VNIs for which a flood list is not configured.

The VTEP flood list is created and modified through the **vxlan flood vtep** command. When configuring VXLAN bridging, the flood list can replace **vxlan multicast-group**.

Example

- These commands create a default VXLAN head-end replication flood list.

```
switch(config)#interface vxlan 1
switch(config-if-Vx1)#vxlan flood vtep 10.1.1.1 10.1.1.2
switch(config-if-Vx1)#show active
 interface Vxlan1
   vxlan flood vtep 10.1.1.1 10.1.1.2
   vxlan udp-port 4789
switch(config-if-Vx1)#
```

- These commands create VXLAN head-end replication flood lists for the VNIs accessed through VLANs 101 and 102.

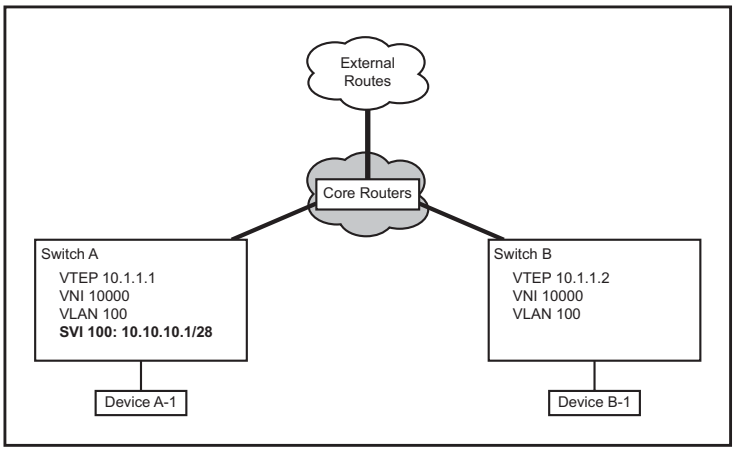
```
switch(config-if-Vx1)#vxlan vlan 101-102 flood vtep 11.1.1.1 11.1.1.2 11.1.1.3
switch(config-if-Vx1)#show active
 interface Vxlan1
   vxlan flood vtep 10.1.1.1 10.1.1.2
   vxlan vlan 101 flood vtep 11.1.1.1 11.1.1.2 11.1.1.3
   vxlan vlan 102 flood vtep 11.1.1.1 11.1.1.2 11.1.1.3
   vxlan udp-port 4789
switch(config-if-Vx1)#
```

19.3.3 VXLAN Routing Configuration

19.3.3.1 Implementing VXLAN Routing

VXLAN routing is enabled by creating a VLAN interface (SVI) on a VLAN that is associated to a VNI. In Figure 19-3, VXLAN routing is enabled on Switch A by configuring a VLAN interface with an IP address of 10.10.10.1. Packets from Devices A-1 and B-2 that have destinations other than 10.10.10.0/28 are VXLAN-bridged to the default gateway (10.10.10.1), then routed from Switch A.

Figure 19-3: Implementing VXLAN Routing



Example

- These commands configure Switch A to perform VXLAN routing. The example includes OSPF routing that is used for underlay routing.

```
switch-A(config)#route-map vxlanvlan permit 10
switch-A(config-route-map-vxlanvlan)#match interface loopb5
switch-A(config-route-map-vxlanvlan)#exit
switch-A(config)#route-map vxlanvlan permit 20
switch-A(config-route-map-vxlanvlan)#match interface vlan 100
switch-A(config-route-map-vxlanvlan)#exit
switch-A(config)#router ospf 1
switch-A(config-router-ospf)#redistribute connected route-map vxlanvlan
switch-A(config-router-ospf)#exit
switch-A(config)#interface loopback 5
switch-A(config-if-Lo5)#ip address 10.25.25.3/24
switch-A(config-if-Lo5)#exit
switch-A(config)#interface vxlan 1
switch-A(config-if-Vx1)#vxlan source-interface loopback 5
switch-A(config-if-Vx1)#vxlan vlan 100 vni 10000
switch-A(config)#interface vlan 100
switch-A(config-if-Vl100)#ip address 10.10.10.1/28
switch-A(config-if-Vl100)#exit
```

19.3.3.2 Configuring Direct VXLAN Routing

In Figure 19-3, VXLAN routing is enabled on Switch A only; Switch B supports VXLAN bridging. Traffic from Switch B devices to the external routes must go through the core route twice: once as they are bridged to is VXLAN gateway and once when routed to its next hop device.

Direct VXLAN routing with VXLAN enabled addresses this issue by configuring each VTEP with all VLANs. This allows packets to be VXLAN-bridged to a local VTEP and routed to remote VTEPs. Indirect routing scales well but is complex to engineer efficiently, and naked routing provides the same scalability to indirect routing. Direct routing leads to the most efficient traffic flows, with the number of virtual subnets or virtual machines increasing at scale, and is thereby optimal from a data plane viewpoint.

The following sections describe conventions required to implement Direct VXLAN Routing, then presents a direct VXLAN routing implementation.

Configuring VARP addresses

For direct routing, an anycast IP address is used as the gateway address on the SVI for a VLAN on all hardware VTEPs associated with that VLAN.

Examples

- These commands configure an IP virtual-router and virtual MAC address.

```
switch(config)#interface Vlan2417
switch(config-if-Vl2417)#ip address 1.0.4.50/24
switch(config-if-Vl2417)#ip virtual-router address 1.0.4.1
switch(config-if-Vl2417)#ip virtual-router mac-address 00:00:11:11:22:22
switch(config)#
```

- These commands configure an IP virtual address (instead of IP virtual-router address) for the VLAN SVI, and a secondary address on the loopback interface for the virtual VTEP IP. The virtual VTEP IP is the logical VTEP hosting the virtual MAC address.

```
switch(config)#interface Vlan2417
switch(config-if-Vl2417)#ip address virtual 1.0.4.1/24
switch(config-if-Vl2417)#exit
switch(config)#interface Loopback0
switch(config-if-Lo0)#ip address 1.0.1.1/32
switch(config-if-Lo0)#ip address 1.0.1.2/32 secondary
switch(config-if-Lo0)#ip virtual-router mac-address 00:00:11:11:22:22
switch(config)#
```

Virtual IP and MAC Addresses

Virtual-router IP addresses can be configured on VLAN interfaces in addition to a primary address. All VTEPs in a direct VXLAN network can be configured with the same virtual router address. This allows devices to use a common IP address as their VXLAN gateway.

The **ip address virtual** command configures a specified address as the primary IPv4 address and as a virtual IP address for the configuration mode VLAN interface. This results in the virtual MAC address (**ip virtual-router mac-address**) assignment to the VLAN interface. In large VXLAN networks, using distinct primary IP addresses for each VTEP limits the number addresses on its subnet for connected hosts. Defining a common virtual IP address for all VTEPs and using that their primary addresses conserves subnet addresses

Example

- These commands specify a virtual router address of 00:00:00:00:00:48 for the switch and, for VLAN 100, a primary address of 10.10.10.10/28 and a virtual IP address of 10.10.10.10.

```
switch(config)#ip virtual-router mac-address 00:00:00:00:00:48
switch(config)#interface vlan 100
switch(config-if-Vl100)#ip address virtual 10.10.10.10/28
switch(config-if-Vl100)#show active
 interface Vlan100
   ip address virtual 10.10.10.10/28
switch(config-if-Vl100)#
```

Virtual VTEP Configuration

A virtual VTEP address is specified by configuring a secondary address on the loopback interface designated as the VXLAN's source interface. All VTEPs in the direct routing topology share the same virtual VTEP address.

You must also configure the secondary VTEP IP on the flood-list of the downstream VXLAN VTEPS as shown below.

Example

- These commands specify a primary (10.1.1.1) and virtual VTEP address (10.2.2.2).

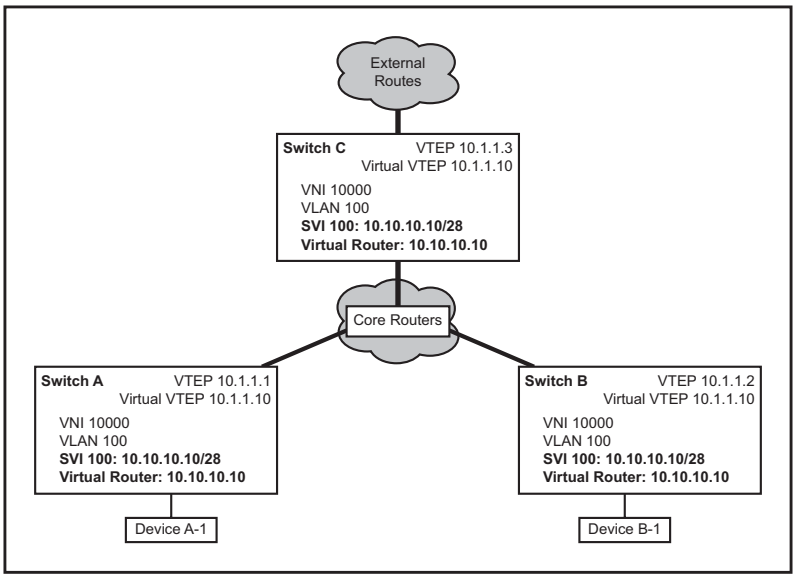
```
Switch1
switch(config)#interface loopback 5
switch(config-if-Lo5)#ip address 10.1.1.1/24
switch(config-if-Lo5)#ip address 10.2.2.2/24 secondary
switch(config-if-Lo5)#show active
 interface Loopback5
   ip address 10.1.1.1/24
   ip address 10.2.2.2/24 secondary
switch(config-if-Lo5)#exit
switch(config)#interface vxlan 1
switch(config-if-Vx1)#vxlan source-interface loopback 5
switch(config-if-Vx1)#show active
 interface Vxlan1
   vxlan source-interface Loopback5
   vxlan udp-port 4789
   vxlan vlan 100 vni 10000
switch(config-if-Vx1)#

Switch2
switch(config)#interface vxlan1
switch(config-if-Vx1)#vxlan flood vtep 10.1.1.1
switch(config-if-Vx1)#vxlan flood vtep 10.2.2.2
```

Direct VXLAN Topology

Figure 19-4 displays a direct VXLAN topology, where each VTEP is configured with the same set of VNIs, VLAN interfaces, and virtual VTEP address.

Figure 19-4: Direct VXLAN Routing



Example

- These commands configure VXLAN parameters for Switch-A in Figure 19-4.

```
switch-A(config)#route-map vxlanvlan permit 10
switch-A(config-route-map-vxlanvlan)#match interface loopb5
switch-A(config-route-map-vxlanvlan)#exit
switch-A(config)#route-map vxlanvlan permit 20
switch-A(config-route-map-vxlanvlan)#match interface vlan 100
switch-A(config-route-map-vxlanvlan)#exit
switch-A(config)#router ospf 1
switch-A(config-router-ospf)#redistribute connected route-map vxlanvlan
switch-A(config-router-ospf)#exit
switch-A(config)#ip virtual-router mac-address 00:00:00:00:00:48
switch-A(config)#interface loopback 5
switch-A(config-if-Lo5)#ip address 10.1.1.3/24
switch-A(config-if-Lo5)#ip address 10.1.1.10/24 secondary
switch-A(config-if-Lo5)#exit
switch-A(config)#interface vxlan 1
switch-A(config-if-Vx1)#vxlan source-interface loopback 5
switch-A(config-if-Vx1)#vxlan vlan 100 vni 10000
switch-A(config)#interface vlan 100
switch-A(config-if-Vl100)#ip address virtual 10.10.10.10/28
switch-A(config-if-Vl100)#exit
```

19.3.3.3 Configuring VXLAN VTEP counters

The switch platforms which use this feature are:

- DCS-7050X
- DCS-7250X
- DCS-7300X

The VXLAN VTEP counters feature enables a device to count VXLAN packets received and sent by the device on a per VTEP basis. Specifically, it enables the device to count bytes and packets that are getting encapsulated and decapsulated as they are passing through.

The counters are logically split up in the two VXLAN directions. Encapsulated on the device and directed to the core, “encap” counters count packets coming from the edge. Decapsulated on the device and heading towards the edge, “decap” counters count packets coming from the core.

To be able to count VXLAN packets the device has to support VXLAN and have a VXLAN interface correctly configured.

Examples

- This command configures the enabling of VXLAN VTEP counters for encap.

```
switch(config)#hardware counter feature vtep encap
switch(config)#
```

- This command configures the disabling of VXLAN VTEP counters for encap.

```
switch(config)#no hardware counter feature vtep encap
switch(config)#
```

- This commands configures the enabling of VXLAN VTEP counters for decap.

```
switch(config)#hardware counter feature vtep decap
switch(config)#
```

- This commands configures the disabling of VXLAN VTEP counters for decap.

```
switch(config)#no hardware counter feature vtep decap
switch(config)#
```

19.3.4 Configuring VXLAN Routing with Overlay VRFs

The switch platforms which use this feature are:

- DCS-7050X
- DCS-7250X
- DCS-7300X

VXLAN SVIs configured in non-default VRFs are supported with VXLAN routing using overlay VRFs. Overlay SVIs are configured in non-default VRFs but underlay SVIs, which provide IP connectivity between VTEPs, must remain in the default VRF. VXLAN routing is deployable by allowing users to configure separate overlay routing domains using VRFs per tenant, thereby allowing support for overlapping IP addresses in the overlay. This provides separation between overlay and underlay traffic, including simpler and cleaner protocol configuration, without using complicated route-maps to control distribution of prefixes to peers in the overlay VRFs and underlay SVIs. IPv4 based VXLAN routing is currently supported.

19.3.5 Configuring VXLAN over MLAG

VTI configuration must be identical on each MLAG peer for them to act as a single VTEP.

The following VTI elements must be configured identically on both MLAG peers:

VLAN-VNI Mappings

Configure identical VLAN to VNI mappings on both MLAG peers using the **vxlan vlan vni** command.

Example

- These commands associate VLAN 100 to VNI 100 and VLAN 200 to VNI 10.10.200.

```
switch(config)#interface vxlan 1
switch(config-if-Vx1)#vxlan vlan 100 vni 100
switch(config-if-Vx1)#vxlan vlan 200 vni 10.10.200
switch(config-if-Vx1)#
```

VTEP IP Address of the Source Loopback Interface

Configure the same VTEP IP address for the source loopback interface on both MLAG peers using the **vxlan source-interface** command.

Example

- These commands configure a primary VTEP address.

```
switch(config)#interface loopback 5
switch(config-if-Lo5)#ip address 10.1.1.1/24
switch(config-if-Lo5)#exit
switch(config)#interface vxlan 1
switch(config-if-Vx1)#vxlan source-interface loopback 5
switch(config-if-Vx1)#
```

Flood VTEP List

Configure the same VTEP flood list on both MLAG peers using the **vxlan flood vtep** command.

Example

- These commands create a default VXLAN head-end replication flood list.

```
switch(config)#interface vxlan 1
switch(config-if-Vx1)#vxlan flood vtep 10.1.1.1 10.1.1.2
switch(config-if-Vx1)#
```

OSPF Configuration

If OSPF is in use, configure the OSPF router ID using the **router-id (OSPFv2)** command to prevent the switch from using the common VTEP IP address as the router ID.

Example

- These commands assign 10.0.0.1 as the OSPFv2 router ID.

```
switch(config)#router ospf 100
switch(config-router-ospf)#router-id 10.0.0.1
switch(config-router-ospf)#
```

19.3.6 Configuring VXLAN Control Service

The VXLAN Control Service (VCS) provides a mechanism by which hardware VTEPs share states between each other in order to establish VXLAN tunnels, without the need for a multicast control plane. This feature enables the use of a VCS client.

Examples

- These commands connect a switch to the VCS running on CVX. The server host IP address is the management IP address of the CVX controller or the IP address that CVX is listening on for client connections.

```
switch(config)#management cvx
switch(config-mgmt-cvx)#server host 172.27.6.248
switch(config-mgmt-cvx)#no shutdown
switch(config-mgmt-cvx)#
```

- These commands configure the VXLAN interface, except for the multicast group configuration, in order to learn from the controller.

```
switch(config)#interface vxlan 1
switch(config-if-Vx1)#vxlan controller-client
switch(config-if-Vx1)#
```

19.3.7 Configuring VXLAN Multicast Decapsulation

The switch platforms which use this feature are:

- DCS-7050X
- DCS-7250X
- DCS-7300X

VXLAN multicast decapsulation enables VTEPs that support Head End Replication (HER). Multicast encapsulated Broadcast/Unknown/Multicast (BUM) packets terminate VTEPs from remote VTEPs that do not support HER.

Examples

- These commands enable VXLAN multicast decapsulation.

```
switch(config)#interface vxlan 1
switch(config-config-if-Vx1)#vxlan multicast-group decap 230.1.1.1
switch(config-config-if-Vx1)#
```

- These commands disable VXLAN multicast decapsulation.

```
switch(config)#interface vxlan 1
switch(config-config-if-Vx1)#no vxlan multicast-group decap 230.1.1.1
switch(config-config-if-Vx1)#
```

19.3.8 VXLAN Rules Support for Mirror ACLs Configuration

The switch platforms which use this feature are:

- DCS-7150S

VXLAN rules support for mirror ACLs configuration permit VXLAN deep inspection rules to be specified in the mirroring ACLs when the switch is operating in normal mode.

Examples

The following are examples of VXLAN rules specified in mirroring ACLs.

- These commands permit all VXLAN traffic (udp protocol and destination port 4789).

```
switch(config)#ip access-list miracl
switch(config-acl-miracl)#permit vxlan any any
switch(config-acl-miracl)#
```

- These commands permit VXLAN traffic with vni 1001 only.

```
switch(config)#ip access-list miracl
switch(config-acl-miracl)#permit vxlan any any vni 1001 0x000000
switch(config-acl-miracl)#
```

- These commands deny VXLAN traffic with vni 0x1000 through 0x100f.

```
switch(config)#ip access-list miracl
switch(config-acl-miracl)#permit vxlan any any vni 0x1000 0x100f
switch(config-acl-miracl)#
```

19.3.9 Displaying VXLAN Configuration

The following section describes the commands that control the display format of VNIs and the commands that list VXLAN configuration and transmission information.

Configuring VNI Display Format

The **vxlan vni notation dotted** command configures the switch to display VNIs in dotted decimal notation. VNI values range from 1 to 16777215 in decimal notation and from 0.0.1 to 255.255.255 in dotted decimal notation.

The command affects the VNI number display in all **show** commands, including **show running-config**. Commands that include VNI as a parameter may use decimal or dotted decimal notion regardless of the setting of this command. By default, show commands display VNI number in decimal notation.

Examples

- These commands configure the switch to display vni numbers in dotted decimal notation, then displays a configuration that includes a VNI setting.

```
switch(config)#vxlan vni notation dotted
switch(config)#interface vxlan 1
switch(config-if-Vx1)#show active
interface Vxlan1
   vxlan udp-port 4789
   vxlan vlan 333 vni 3.4.5
switch(config-if-Vx1)#
```

- These commands configure the switch to display vni numbers in decimal notation, then displays a configuration that includes a VNI setting.

```
switch(config)#no vxlan vni notation dotted
switch(config)#interface vxlan 1
switch(config-if-Vx1)#show active
interface Vxlan1
   vxlan udp-port 4789
   vxlan vlan 333 vni 197637
switch(config-if-Vx1)#
```

MAC Address Table

The MAC address table indicates a MAC address from a device on a remote host by indicating Vx interface as the port that corresponds to the address.

Example

- This command displays a MAC address table that includes entries of devices from remote hosts by specifying Vx1 as the corresponding port.

```
switch>show mac address-table
          Mac Address Table
------------------------------------------------------------------

Vlan    Mac Address       Type        Ports      Moves   Last Move
----    -----------       ----        -----      -----   ---------
   1    0050.5682.6725    DYNAMIC     Et16       1       0:02:01 ago
   1    0050.568e.58e9    DYNAMIC     Et23       2       0:08:53 ago
   1    0050.56a0.474a    DYNAMIC     Et16       1       0:18:04 ago
  51    0000.0051.0004    DYNAMIC     Et5        1       12 days, 1:02:44 ago
  51    0000.0051.0005    DYNAMIC     Et5        1       12 days, 1:02:44 ago
  51    0000.0051.0101    DYNAMIC     Vx1        1       12 days, 0:17:30 ago
  51    0000.0051.0102    DYNAMIC     Vx1        1       12 days, 0:17:30 ago
  61    0000.0061.0005    DYNAMIC     Et5        1       12 days, 1:02:44 ago
Total Mac Addresses for this criterion: 8

          Multicast Mac Address Table
------------------------------------------------------------------

Vlan    Mac Address       Type        Ports
----    -----------       ----        -----
Total Mac Addresses for this criterion: 0
switch>
```

VXLAN MAC Address Table

VXLAN MAC address table entries correlate MAC addresses accessible through remote VTEPs with the local VLAN and the IP address of the VTEP through which the addressed device is accessed. The VTI uses this table when constructing the VXLAN encapsulation to specify the destination IP address of the recipient VTEP and the VNI segment through which the device's remote VLAN is accessed.

The **show vxlan address-table** command displays the VXLAN MAC address table.

Example

- This command displays the VXLAN address table.

```
switch>show vxlan address-table
          Vxlan Mac Address Table
----------------------------------------------------------------------

Vlan  Mac Address     Type     Prt  Vtep            Moves   Last Move
----  -----------     ----     ---  ----            -----   ---------
  51  0000.0051.0101  DYNAMIC  Vx1  10.25.2.12      1       4 days, 0:37:14 ago
  51  0000.0051.0102  DYNAMIC  Vx1  10.25.2.12      1       4 days, 0:37:14 ago
  51  0000.0051.0103  DYNAMIC  Vx1  10.25.2.12      1       4 days, 0:37:14 ago
  51  0000.0051.0104  DYNAMIC  Vx1  10.25.2.12      1       4 days, 0:37:14 ago
  51  0000.0051.0105  DYNAMIC  Vx1  10.25.2.12      1       4 days, 0:37:14 ago
  61  0000.0061.0103  DYNAMIC  Vx1  10.25.2.12      1       4 days, 0:37:14 ago
  61  0000.0061.0104  DYNAMIC  Vx1  10.25.2.12      1       4 days, 0:37:14 ago
  61  0000.0061.0105  DYNAMIC  Vx1  10.25.2.12      1       4 days, 0:37:14 ago
switch>
```

VXLAN MAC Address Table

The **show vxlan vtep** command displays information about remote VTEPs that the configured VTI has discovered and with whom it has exchanged packets.

Example

- These commands display the VTEPs that have exchanged data with the configured VTI.

```
switch>show vxlan vtep
Remote vteps for Vxlan1:
10.52.2.12
Total number of remote vteps:  1
switch>
```

VXLAN Counters

The **clear vxlan counters** command resets the VXLAN counters. The **show vxlan counters** command displays the VXLAN counters.

Example

- This command displays the VXLAN counters

```
switch>show vxlan counters software
encap_bytes:3452284
encap_pkts:27841
encap_read_err:1
encap_discard_runt:0
encap_discard_vlan_range:0
encap_discard_vlan_map:0
encap_send_err:0
encap_timeout:1427
decap_bytes_total:382412426
decap_pkts_total:2259858
decap_bytes:0
decap_pkts:0
decap_runt:0
decap_pkt_filter:45128
decap_bytes_filter:5908326
decap_discard_vxhdr:0
decap_discard_vlan_map:2214730
decap_timeout:0
decap_sock_err:1
switch>
```

19.4 VXLAN Command Descriptions

VXLAN Global Configuration Commands

- interface vxlan
- ip address virtual
- vxlan vni notation dotted

VXLAN Interface Configuration Commands

- vxlan flood vtep
- vxlan multicast-group
- vxlan multicast-group decap
- vxlan source-interface
- vxlan udp-port
- vxlan vlan vni

VXLAN Display and Clear Commands

- clear vxlan counters
- show service vxlan
- show vxlan address-table
- show vxlan counters
- show vxlan flood vtep
- show vxlan vtep

clear vxlan counters

The **clear vxlan counters** command resets the VXLAN counters.

Command Mode

Privileged EXEC

Command Syntax

```
clear vxlan counters ROUTE_TYPE
```

Parameters

- ***ROUTE_TYPE*** Specifies the type of VXLAN counter reset by the command.
  - **software** Command resets software counters.
  - **varp** Command resets virtual-ARP counters.

Related Commands

- **show vxlan counters** displays the VXLAN counters.

Example

- This command resets the VXLAN counters

```
switch#clear vxlan counters software
switch#show vxlan counters software
encap_bytes:0
encap_pkts:0
encap_read_err:0
encap_discard_runt:0
encap_discard_vlan_range:0
encap_discard_vlan_map:0
encap_send_err:0
encap_timeout:0
decap_bytes_total:0
decap_pkts_total:0
decap_bytes:0
decap_pkts:0
decap_runt:0
decap_pkt_filter:0
decap_bytes_filter:0
decap_discard_vxhdr:0
decap_discard_vlan_map:0
decap_timeout:0
decap_sock_err:0
switch#
```

interface vxlan

The **interface vxlan** command places the switch in VXLAN-interface configuration mode for modifying the specified VXLAN tunnel interface (VTI). The command also instantiates the interface if it was not previously created.

VXLAN interface configuration mode is not a group change mode; ***running-config*** is changed immediately after commands are executed. The **exit** command does not affect the configuration.

The **no interface vxlan** deletes the specified VTI interface, including its configuration statements, from ***running-config***. The **default interface vxlan** command removes all configuration statements for the specified VTI from ***running-config*** without deleting the interfaces.

Command Mode

Global Configuration

Command Syntax

```
interface vxlan vx_range
no interface vxlan vx_range
default interface vxlan vx_range
```

Parameter

- *vx_range* VXLAN interface number. The only permitted value is 1.

Commands Available in link-flap Configuration Mode

- **vxlan multicast-group**
- **vxlan source-interface**
- **vxlan udp-port**
- **vxlan vlan vni**

Example

- These commands create VXLAN tunnel interface 1, place the switch in VXLAN-interface configuration mode, then display parameters of the new VTI.

```
switch(config)#interface vxlan 1
switch(config-if-Vx1)#show active
interface Vxlan1
   vxlan udp-port 4789
switch(config-if-Vx1)#
```

- This command exits VXLAN-interface configuration mode, placing the switch in global configuration mode.

```
switch(config-if-Vx1)#exit
switch(config)#
```

ip address virtual

The **ip address virtual** command configures a specified address as the primary IPv4 address and as a virtual IP address for the configuration mode VLAN interface. The address resolves to the virtual MAC address configured through the **ip virtual-router mac-address** command. The command includes a subnet designation that is required in primary IP address assignments.

This command is typically used in VXLAN routing configurations as an alternative to assigning a unique IP address to each VTEP. All existing IPv4 addresses must be removed from the interface before executing this command.

The **no ip address virtual** and **default ip address virtual** commands remove the IPv4 address and virtual IP assignment from the configuration mode interface by deleting the **ip address virtual** command from ***running-config***.

Removing the IPv4 address assignments from an interface disables IPv4 processing on that port.

Command Mode

Interface-VLAN Configuration

Command Syntax

```
ip address virtual ipv4_subnet
no ip address virtual
default ip address virtual
```

Parameters

- *ipv4_subnet* IPv4 and subnet address (CIDR or address-mask notation).

Related Commands

- **ip address**
- **ip address virtual**
- **ip virtual-router mac-address**

Example

- This command configures 10.10.10.1 as the IPv4 address and virtual address for VLAN 100.

```
switch(config-if-Vl100)#show active
 interface Vlan100
   ip address virtual 10.10.10.1/28
switch(config-if-Vl100)#
```

show service vxlan

The **show service vxlan** command displays the status of the Vxlan Control Service (VCS) and the received (from all connected VTEPs) and advertised (to all connected VTEPs) MAC address reachability information.

Command Mode

EXEC

Command Syntax

show service vxlan [status | switch [*SWITCH_TYPE*] | vni [*VNI_INFO*]]

Parameters

- ***SWITCH_TYPE*** displayed by switch type. Options include:
  - ***word*** hostname, IP address, or ID of the switch.
  - **all** all switches.
- ***VNI_INFO*** displayed with VNI information. Options include:
  - **advertised** advertised MAC addresses.
  - **received** received MAC addresses.

Example

- This command displays the status of the VCS.

```
switch(config)#show service vxlan status
Vxlan Controller Service is   : stopped
Mac learning                  : Control plane
Resync period                 : 300 seconds
Resync in progress            : No
Capability                    : VXLAN v4 overlay routing
                                VXLAN v4 overlay indirect routing
fm319(config-if-Vx1)#show service vxlan status
Vxlan Controller Service is   : stopped
Mac learning                  : Control plane
Resync period                 : 300 seconds
Resync in progress            : No
Capability                    : VXLAN v4 overlay routing
                                VXLAN v4 overlay indirect routing

switch(config)#
```

show vxlan address-table

The **show vxlan address-table** command displays the VXLAN address table. Entries are created by extracting information from packets received from remote VTEPs.

The VXLAN address table correlates MAC addresses that are accessible through remote VTEPs with the local VLAN and the IP address of the VTP through which the addressed device is accessible. The VTI uses this table when constructing the VXLAN encapsulation fields to specify the destination IP address of the recipient VTEP and the VNI segment through which the device's remote VLAN is accessed.

Command Mode

EXEC

Command Syntax

show vxlan address-table [*ENTRY_TYPE*][*MAC_ADDR*][*VLANS*][*REMOTE_VTEP*]

Parameters

- ***ENTRY_TYPE*** command filters display by entry type. Options include:
  - <no parameter> all table entries.
  - **configured** static entries; includes unconfigured VLAN entries.
  - **dynamic** entries learned though packet receipts.
  - **static** entries entered by CLI commands.
  - **unicast** entries with unicast MAC address.
- ***MAC_ADDR*** command uses MAC address to filter displayed entries.
  - <no parameter> all MAC addresses table entries.
  - **address** *mac_address* displays entries with specified address (dotted hex notation – H.H.H).
- ***VLANS*** command filters display by VLAN.
  - <no parameter> all VLANs.
  - **vlan** *v_num* VLAN specified by *v_num*.
- ***REMOTE_VTEP*** Filters entries by IP address of the remote VTEPs. Options include:
  - <no parameter> all items.
  - **vtep** *ipaddr_1* [*ipaddr_2*...*ipaddr_n*] Identifies VTEPs by their IP address.

Example

- This command displays the VXLAN address table.

```
switch>show vxlan address-table
          Vxlan Mac Address Table
----------------------------------------------------------------------

Vlan  Mac Address     Type      Prt  Vtep            Moves   Last Move
----  -----------     ----      ---  ----            -----   ---------
  51  0000.0051.0101  DYNAMIC   Vx1  10.25.2.12      1       4 days, 0:37:14 ago
  51  0000.0051.0102  DYNAMIC   Vx1  10.25.2.12      1       4 days, 0:37:14 ago
  51  0000.0051.0103  DYNAMIC   Vx1  10.25.2.12      1       4 days, 0:37:14 ago
  51  0000.0051.0104  DYNAMIC   Vx1  10.25.2.12      1       4 days, 0:37:14 ago
  51  0000.0051.0105  DYNAMIC   Vx1  10.25.2.12      1       4 days, 0:37:14 ago
  61  0000.0061.0102  DYNAMIC   Vx1  10.25.2.12      1       4 days, 0:37:14 ago
  61  0000.0061.0103  DYNAMIC   Vx1  10.25.2.12      1       4 days, 0:37:14 ago
  61  0000.0061.0104  DYNAMIC   Vx1  10.25.2.12      1       4 days, 0:37:14 ago
  61  0000.0061.0105  DYNAMIC   Vx1  10.25.2.12      1       4 days, 0:37:14 ago
switch>
```

show vxlan counters

The **show vxlan counters** command displays the VXLAN counters.

Command Mode

EXEC

Command Syntax

```
show vxlan counters ROUTE_TYPE
```

Parameters

- ***ROUTE_TYPE*** Specifies the type of VXLAN counter displayed by the command.
  - **software** Command displays software routers.
  - **varp** Command displays virtual-ARP counters.
  - **vtep** Command displays counters for VTEPs which are identified by their IP address. An optional keyword allows the user to view a single direction of the counters:
    - **encap** "encap" counters count packets coming from the edge, encapsulated on the device and directed to the core.
    - **decap** "decap" counters count packets coming from the core, decapsulated on the device and heading towards the edge.

Related Commands

- **clear vxlan counters** resets the VXLAN counters.

Example

- This command displays the VXLAN counters for software routers.

```
switch>show vxlan counters software
encap_bytes:3452284
encap_pkts:27841
encap_read_err:1
encap_discard_runt:0
encap_discard_vlan_range:0
encap_discard_vlan_map:0
encap_send_err:0
encap_timeout:1427
decap_bytes_total:382412426
decap_pkts_total:2259858
decap_bytes:0
decap_pkts:0
decap_runt:0
decap_pkt_filter:45128
decap_bytes_filter:5908326
decap_discard_vxhdr:0
decap_discard_vlan_map:2214730
decap_timeout:0
decap_sock_err:1
switch>
```

- This command displays the VXLAN counters for VTEPs.

```
switch>show vxlan counters vtep
                                                                    Decap Drop or
                                  Decap Known           Decap BUM      Exception
VTEP         Decap Bytes      Unicast Packets           Packets          Packets
-------- --------------- -------------------- -------------------- -------------
1.0.14.1     62526968000            312632701            312636979             2
1.0.16.1             800                    2                    6     312279633
1.0.23.1             800                    2                    6             2
unlearnt               0                    0                    0             0
                                                Encap Drop or
                                                    Exception
VTEP         Encap Bytes        Encap Packets       Packets
-------- --------------- -------------------- -------------
1.0.14.1     30579308814            268239551             2
1.0.16.1            1140                   10             2
1.0.23.1               0                    0             0

switch>
```

show vxlan flood vtep

The **show vxlan flood vtep** command displays the flood list that the switch is using to perform head-end replication. Head-end replication is a data distribution method that supports broadcast, unknown unicast, and multicast (BUM) traffic over VXLANs by replicating BUM data locally for transmission to the set of remote VTEPs that a flood list specifies. The command displays the VLAN ID that references the configured VNIs (**vxlan vlan vni**).

The flood list is determined by the **vxlan flood vtep** command.

Command Mode

EXEC

Command Syntax

show vxlan flood vtep [***VLANS***]

Parameters

- ***VLANS*** command filters display by the reference VLAN.
  - <no parameter> all VLANs.
  - **vlan** *v_range* VLANs specified by *v_range*.

  Valid *v_range* formats include number, range, or comma-delimited list of numbers and ranges.

Guidelines

The command displays flood list contents only when the VLAN line protocol status is ***up***.

Related Commands

- **vxlan flood vtep** configures the flood list.

Example

- These commands display the VTEPs that have exchanged data with the configured VTI.

```
switch>show vxlan flood vtep vlan 100-102
          Vxlan Flood Vtep Table
-------------------------------------------------------

Vlan   Ip Address
----   ------------------------------------------------
100    3.3.3.3
101    11.1.1.1         11.1.1.2         11.1.1.3
102    11.1.1.1         11.1.1.2         11.1.1.3
       12.1.1.1
switch>
```

show vxlan vtep

The **show vxlan vtep** command displays information about remote VTEPs that the configured VTI has discovered and with whom it has exchanged packets.

Command Mode

EXEC

Command Syntax

show vxlan vtep

Example

- These commands display the VTEPs that have exchanged data with the configured VTI.

```
switch>show vxlan vtep
Remote vteps for Vxlan1:
10.52.2.12
Total number of remote vteps:  1
switch>
```

vxlan flood vtep

The **vxlan flood vtep** command supports VXLAN head-end replication by creating or modifying a list that specifies remote VTEPs to which the switch bridges replicated traffic. Head-end replication is a data distribution method that supports broadcast, unknown unicast, and multicast (BUM) traffic over VXLANs by replicating BUM data locally for transmission to the set of remote VTEPs that a flood list specifies. This data flooding facilitates remote MAC address learning through the forwarding of data with unknown MACs.

Each **vxlan flood vtep** statement in ***running-config*** associates a set of VTEP addresses to an access VNI. A default flood list is also configurable that applies to all VNIs for which a flood list is not configured. The **vxlan flood vtep** command is available in the following formats to create or modify corresponding ***running-config*** statements:

- **vxlan flood vtep** creates a statement for a specified VNI and replaces existing statements for that VNI.
- **vxlan flood vtep add** modifies an existing flood statement by adding the specified VTEPs. This statement creates a list if it references a VNI that has no flood statement.
- **vxlan flood vtep remove** modifies an existing flood statement by deleting the specified VTEPs. This statement has no effect if it references a VNI that has no flood statement.

The **vxlan flood vtep** command specifies a VNI by referencing its associated VLAN ID (**vxlan vlan vni**). The command provides these options for specifying the reference VLANs:

- ***a single VLAN***: creates or modifies a single statement referenced by the command.
- ***a range of VLANs***: creates or modifies all statements referenced by the VLAN range.
- ***no VLAN***: creates or modifies the default list

The **no vxlan flood vtep** and **default vxlan flood vtep** commands remove the specified flood list by deleting the corresponding **vxlan flood vtep** statements from ***running-config***. Commands that specify a VLAN range remove all corresponding statements.

Command Mode

Interface-VXLAN Configuration

Command Syntax

```
vxlan [ACCESS_VNI] flood vtep [MODIFY] VTEP_1 [VTEP_2] ... [VTEP_N]
no vxlan [ACCESS_VNI] flood vtep
default vxlan [ACCESS_VNI] flood vtep
```

Parameters

- ***ACCESS_VNI*** VLAN ID associated to the flood list's target VNI. Value ranges from 1 to 4094.
  - <no parameter > default list.
  - **vlan** *vlan_range* List of VLANs. (Number, range, comma-delimited list of numbers and ranges). Numbers range from 1 to 4094.
- ***MODIFY*** Statement modification method. Options include:
  - <no parameter > creates new list for specified VLANs. Current list is overwritten.
  - **add** specified VTEPs are added to existing list.
  - **remove** specified VTEPs are deleted from existing list.
- ***VTEP_X*** IPv4 address of VTEPs that are added or removed from the list.

Example

- These commands create a default VXLAN head-end replication flood list.

```
switch(config)#interface vxlan 1
switch(config-if-Vx1)#vxlan flood vtep 10.1.1.1 10.1.1.2
switch(config-if-Vx1)#show active
 interface Vxlan1
   vxlan flood vtep 10.1.1.1 10.1.1.2
   vxlan udp-port 4789
switch(config-if-Vx1)#
```

- These commands create VXLAN head-end replication flood lists for the VNIs accessed through VLANs 101 and 102.

```
switch(config-if-Vx1)#vxlan vlan 101-102 flood vtep 11.1.1.1 11.1.1.2 11.1.1.3
switch(config-if-Vx1)#show active
 interface Vxlan1
   vxlan flood vtep 10.1.1.1 10.1.1.2
   vxlan vlan 101 flood vtep 11.1.1.1 11.1.1.2 11.1.1.3
   vxlan vlan 102 flood vtep 11.1.1.1 11.1.1.2 11.1.1.3
   vxlan udp-port 4789
switch(config-if-Vx1)#
```

- These commands add two VTEPs for the VNI access through VLAN 102.

```
switch(config-if-Vx1)#vxlan vlan 102 flood vtep add 12.1.1.1
switch(config-if-Vx1)#show active
 interface Vxlan1
   vxlan flood vtep 10.1.1.1 10.1.1.2
   vxlan vlan 101 flood vtep 11.1.1.1 11.1.1.2 11.1.1.3
   vxlan vlan 102 flood vtep 11.1.1.1 11.1.1.2 11.1.1.3 12.1.1.1
   vxlan udp-port 4789
switch(config-if-Vx1)#
```

vxlan multicast-group

The **vxlan multicast-group** command associates a specified multicast group with the configuration mode VXLAN interface (VTI), which handles multicast and broadcast traffic as a layer 2 interface in a bridging domain.

The VTI maps multicast traffic from its associated VLANs to the specified multicast group. Inter-VTEP multicast communications include all VTEPs that are associated with the specified multicast group, which is independent of any other multicast groups that VLAN hosts may join.

A VTI can be associated with one multicast group. By default, a VTI is not associated with any multicast group.

The **no vxlan multicast-group** and **default vxlan multicast-group** commands removes the multicast group – VTI association by removing the **vxlan multicast-group** command from ***running-config***.

Command Mode

Interface-VXLAN Configuration

Command Syntax

```
vxlan multicast-group group_addr
no vxlan multicast-group
default vxlan multicast-group
```

Parameters

- *group_addr* IPv4 address of multicast group. Dotted decimal notation of a valid multicast address.

Related Commands

- **interface vxlan** places the switch in VXLAN interface configuration mode.

Examples

- This command associates the multicast address of 227.10.1.1 with VTI 1.

```
switch(config)#interface vxlan 1
switch(config-if-Vx1)#vxlan multicast-group 227.10.1.1
switch(config-if-Vx1)#show active
interface Vxlan1
   vxlan multicast-group 227.10.1.1
   vxlan udp-port 4789
switch(config-if-Vx1)#
```

- This command changes VTI 1's multicast group association.

```
switch(config-if-Vx1)#vxlan multicast-group 227.10.5.5
switch(config-if-Vx1)#show active
interface Vxlan1
   vxlan multicast-group 227.10.5.5
   vxlan udp-port 4789
switch(config-if-Vx1)#
```

- This command removes the multicast group association from VTI 1.

```
switch(config-if-Vx1)#no vxlan multicast-group
switch(config-if-Vx1)#show active
interface Vxlan1
   vxlan udp-port 4789
switch(config-if-Vx1)#
```

vxlan multicast-group decap

The **vxlan multicast-group decap** command enables VXLAN multicast decapsulation.

VTEPs are enabled by VXLAN multicast decapsulation, supporting Head End Replication (HER). Multicast encapsulated Broadcast/Unknown/Multicast (BUM) packets terminate VTEPs from remote VTEPs that do not support HER.

The **no vxlan multicast-group decap** and **default vxlan multicast-group decap** commands disable VXLAN multicast decapsulation.

Command Mode

Interface-VXLAN Configuration

Command Syntax

```
vxlan multicast-group decap group_addr
no vxlan multicast-group decap
default vxlan multicast-group decap
```

Parameters

- *group_addr* IPv4 address of multicast group. Dotted decimal notation of a valid multicast address.

Examples

- This command enables VXLAN multicast decapsulation.

```
switch(config)#interface vxlan 1
switch(config-config-if-Vx1)#vxlan multicast-group decap 230.1.1.1
switch(config-config-if-Vx1)#
```

- This command disables VXLAN multicast decapsulation.

```
switch(config)#interface vxlan 1
switch(config-config-if-Vx1)#no vxlan multicast-group decap 230.1.1.1
switch(config-config-if-Vx1)#
```

vxlan source-interface

The **vxlan source-interface** command specifies the interface from which the configuration mode VXLAN interface (VTI) derives the source address (IP) that it uses when exchanging VXLAN frames. There is no default source interface assignment.

The **no vxlan source-interface** and **default vxlan source-interface** commands remove the source interface assignment from the configuration mode VXLAN interface by deleting the corresponding **ip vxlan source-interface** command from ***running-config***.

Command Mode

Interface-VXLAN Configuration

Command Syntax

```
vxlan source-interface INT_NAME
no vxlan source-interface
default vxlan source-interface
```

Parameters

- ***INT_NAME*** Interface type and number. Options include:
  - **loopback** *l_num* Loopback interface specified by *l_num*.

Guidelines

A VXLAN interface is inoperable without the source-interface assignment.

Related Commands

- **interface vxlan** places the switch in VXLAN interface configuration mode.

Example

- These commands configure VTI 1 to use the IP address 10.25.25.3 as the source address of outbound VXLAN frames.

```
switch(config)#interface loopback 15
switch(config-if-Lo15)#ip address 10.25.25.3/24
switch(config-if-Lo15)#exit
switch(config)#interface vxlan 1
switch(config-if-Vx1)#vxlan source-interface loopback 15
switch(config-if-Vx1)#show active
interface Vxlan1
   vxlan source-interface Loopback15
   vxlan udp-port 4789
switch(config-if-Vx1)#
```

vxlan udp-port

The **vxlan udp-port** command associates a UDP port with the configuration mode VXLAN interface (VTI). By default, UDP port 4789 is associated with the VTI.

Packets bridged to the VTI from a VLAN are encapsulated with a VXLAN header that includes the VNI associated with the VLAN and the IP address of the VTEP that connects to the recipient, then sent through the UDP port. Packets that arrive through the UDP port are sent to the bridging domain of the recipient VLAN as determined by the VNI number in the VXLAN header and the interface's VNI-VLAN map.

The **no vxlan udp-port** and **default vxlan udp-port** command restores the default UDP port association (4789) on the configuration mode interface by deleting the corresponding **vxlan udp-port** command from ***running-config***.

Command Mode

Interface-VXLAN Configuration

Command Syntax

```
vxlan udp-port port_id
no vxlan udp-port
default vxlan udp-port
```

Parameters

- *port_id* UDP port number. Value ranges from 1024 to 65535.

Guidelines

UDP port 4789 is reserved by convention for VXLAN usage. Under most typical applications, this parameter should be set to the default value.

Related Commands

- **interface vxlan** places the switch in VXLAN interface configuration mode.

Example

- This command associates UDP port 5500 with VXLAN interface 1.

```
switch(config)#interface vxlan 1
switch(config-if-Vx1)#vxlan udp-port 5500
switch(config-if-Vx1)#show active
interface Vxlan1
   vxlan udp-port 5500
switch(config-if-Vx1)#
```

- This command resets the VXLAN interface 1 UDP port association of 4789.

```
switch(config-if-Vx1)#no vxlan udp-port
switch(config-if-Vx1)#show active
interface Vxlan1
   vxlan udp-port 4789
switch(config-if-Vx1)#
```

vxlan vlan vni

The **vxlan vlan vni** command associates a VLAN ID with a virtual network identifier (VNI). A VNI is a 24-bit number that is assigned to a VLAN to distinguish it from other VLANs that are on a VXLAN tunnel interface (VTI). VNI values range from 1 to 16777215 in decimal notation and from 0.0.1 to 255.255.255 in dotted decimal notation.

When a VLAN bridges a packet to the VTI, the packet is encapsulated with a VXLAN header that includes the VNI that is associated with the VLAN. Packets that arrive on the VTI's UDP socket are bridged to the VLAN that is associated with the VNI specified by the VXLAN header that encapsulates the packet.

The VTI requires a one-to-one correspondence between specified VLANs and VNI values. Commands that assign a new VNI to a previously configured VLAN replace the existing VLAN assignment statement in ***running-config***. Commands that attempt to assign a VNI value to a second VLAN generate a CLI error.

The **no vxlan vlan vni** and **default vxlan vlan vni** commands remove the specified VLAN-VNI association from the configuration mode interface by deleting the corresponding **vxlan vlan** command from ***running-config***.

Command Mode

Interface-VXLAN Configuration

Command Syntax

```
vxlan vlan vlan_id vni vni_id
no vxlan vlan vlan_id vni [vni_id]
default vxlan vlan vlan_id vni [vni_id]
```

Parameters

- *vlan_id* number of access VLAN. Value ranges from 1 to 4094.
- *vni_id* VNI number. Valid formats: decimal <***1*** to ***16777215***> or dotted decimal <***0.0.1*** to ***255.255.255***>.

Example

- These commands associate VLAN 100 to VNI 100 and VLAN 200 to VNI 10.10.200.

```
switch(config)#interface vxlan 1
switch(config-if-Vx1)#vxlan vlan 100 vni 100
switch(config-if-Vx1)#vxlan vlan 200 vni 10.10.200
switch(config-if-Vx1)#show active
interface Vxlan1
   vxlan udp-port 4789
   vxlan vlan 200 vni 658120
   vxlan vlan 100 vni 100
switch(config-if-Vx1)#vxlan vni notation dotted
switch(config-if-Vx1)#show active
interface Vxlan1
   vxlan udp-port 4789
   vxlan vlan 200 vni 10.10.200
   vxlan vlan 100 vni 0.0.100
switch(config-if-Vx1)#
```

vxlan vni notation dotted

The **vxlan vni notation dotted** command configures the switch to display VNIs in dotted decimal notation. A virtual network identifier (VNI) is a 24-bit number that is assigned to a VLAN to distinguish it from other VLANs that are on a VXLAN tunnel interface. VNI values range from 1 to 16777215 in decimal notation and from 0.0.1 to 255.255.255 in dotted decimal notation.

The command affects the VNI number display in all **show** commands, including **show running-config**. Commands that include VNI as a parameter may use decimal or dotted decimal notion regardless of the setting of this command. By default, show commands display VNI number in decimal notation.

The **no vxlan vni notation dotted** and **default vxlan vni notation dotted** commands restore the default setting of displaying vni numbers in decimal notation by deleting the **vxlan vni notation dotted** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
vxlan vni notation dotted
no vxlan vni notation dotted
default vxlan vni notation dotted
```

Examples

- These commands configure the switch to display vni numbers in dotted decimal notation, then displays a configuration that includes a vni setting.

```
switch(config)#vxlan vni notation dotted
switch(config)#interface vxlan 1
switch(config-if-Vx1)#show active
interface Vxlan1
   vxlan udp-port 4789
   vxlan vlan 333 vni 3.4.5
switch(config-if-Vx1)#
```

- These commands configure the switch to display vni numbers in decimal notation, then displays a configuration that includes a vni setting.

```
switch(config)#no vxlan vni notation dotted
switch(config)#interface vxlan 1
switch(config-if-Vx1)#show active
interface Vxlan1
   vxlan udp-port 4789
   vxlan vlan 333 vni 197637
switch(config-if-Vx1)#
```

Chapter 20

ACLs and Route Maps

The switch uses rule-based lists to control packet access to ports and to select routes for redistribution to routing domains defined by dynamic routing protocols. This chapter describes the construction of access control lists (ACLs), prefix lists, and route maps.

This chapter includes the following sections:

- Section 20.1: ACL, Service ACL, Route Map, and Prefix List Introduction
- Section 20.2: Access Control Lists
- Section 20.3: Service ACLs
- Section 20.4: Route Maps
- Section 20.5: Prefix Lists
- Section 20.6: ACL, Route Map, and Prefix List Commands

20.1 ACL, Service ACL, Route Map, and Prefix List Introduction

Note On hardware purchased after July 3, 2017, security ACLs are only available on AlgoMatch-enabled platforms.

Note On hardware purchased after July 3, 2017, control plane ACLs are not available.

Access control lists (ACLs), Service ACLs, route maps, and prefix lists are all processed in order, beginning with the first rule and proceeding until a match is encountered.

An access control list (ACL) is a list of rules that control the inbound flow of packets into Ethernet interfaces, port channel interfaces or the switch control plane. The switch supports the implementation of a wide variety of filtering criteria including IP and MAC addresses, TCP/UDP ports with include/exclude options without compromising its performance or feature set. Filtering syntax is industry standard.

A Service ACL is an ACL applied by a control-plane process to control connections to, or packets processed by, the agent process.

Note Service ACLs are only available on EOS versions 4.19.0F and above.

A route map is a list of rules that control the redistribution of IP routes into a protocol domain on the basis of such criteria as route metrics, access control lists, next hop addresses, and route tags. Route maps can also alter parameters of routes as they are redistributed.

A prefix list is a list of rules that defines route redistribution access for a specified IP address space. Route maps often use prefix lists to filter routes.

20.2 Access Control Lists

These sections describe Access Control Lists:

- Section 20.2.1: ACL Description
- Section 20.2.2: ACL Configuration
- Section 20.2.3: Applying ACLs

20.2.1 ACL Description

This section describes ACL composition and function. The switch supports the following ACL types:

- IPv4
- IPv6
- Standard IPv4
- Standard IPv6
- MAC

20.2.1.1 ACL Structure

An ACL is an ordered list of rules that defines access restrictions for the entities (the control plane, or an interface) to which it is applied. ACLs are also used by route maps to select routes for redistribution into specified routing domains.

ACL rules specify the data to which packet contents are compared when filtering data.

- The interface forwards packets that match all conditions in a permit rule.
- The interface drops packets that match all conditions in a deny rule.
- The interface drops packets that do not match at least one rule.

Upon its arrival at an interface, a packet's fields are compared to the first rule of the ACL applied to the interface. Packets that match the rule are forwarded (permit rule) or dropped (deny rule). Packets that do not match the rule are compared to the next rule in the list. This process continues until the packet either matches a rule or the rule list is exhausted. The interface drops packets not matching a rule.

The sequence number designates the rule's placement in the ACL.

20.2.1.2 ACL Rules

ACL rules consist of a condition list that is compared to inbound packet fields. When all of a rule's criteria match a packet's contents, the interface performs the action specified by the rule.

The set of available conditions depend on the ACL type and the specified protocol within rule. The following is a list of conditions available for supported ACL types

IPv4 ACL Rule Parameters

All rules in IPv4 ACLs include the following criteria:

- **Protocol:** The packet's IP protocol. Valid rule inputs include:
  - Protocol name for a limited set of common protocols.
  - Assigned protocol number for all IP protocols.
- **Source Address:** The packet's source IPv4 address. Valid rule inputs include:
  - a subnet address (CIDR or address-mask). Discontiguous masks are supported.

- a host IP address (dotted decimal notation).
- ***any*** to denote that the rule matches all source addresses.

- **Destination Address:** The packet's destination IP address. Valid rule inputs include:
  - a subnet address (CIDR or address-mask). Discontiguous masks are supported.
  - a host IP address (dotted decimal notation).
  - ***any*** to denote that the rule matches all destination addresses.

All rules in IPv4 ACLs ***may*** include the following criteria:

- **Fragment:** Rules filter on the fragment bit.
- **Time-to-live:** Compares the TTL (time-to-live) value in the packet to a specified value. Valid in ACLs applied to the control plane. Validity in ACLs applied to the data plane varies by switch platform. Comparison options include:
  - ***Equal:*** Packets match if packet value equals statement value.
  - ***Greater than:*** Packets match if packet value is greater than statement value.
  - ***Less than:*** Packets match if packet value is less than statement value.
  - ***Not equal:*** Packets match if packet value does not equal statement value.

The availability of the following optional criteria depends on the specified protocol:

- **Source Ports / Destination Ports:** A rule filters on ports when the specified protocol supports IP address-port combinations. Rules provide one of these port filtering values:
  - ***any*** denotes that the rule matches all ports.
  - A list of ports that matches the packet port. Maximum list size is 10 ports.
  - Negative port list. The rule matches any port not in the list. Maximum list size is 10 ports.
  - Integer (lower bound): The rule matches any port with a number larger than the integer.
  - Integer (upper bound): The rule matches any port with a number smaller than the integer.
  - Range integers: The rule matches any port whose number is between the integers.
- **Flag bits**: Rules filter TCP packets on flag bits.
- **Message type**: Rules filter ICMP type or code.
- **Tracked:** Matches packets in existing ICMP, UDP, or TCP connections. Valid in ACLs applied to the control plane. Validity in ACLs applied to the data plane varies by switch platform.

IPv6 ACL Rule Parameters

Note When calculating the size of ACLs, be aware that Arista switches install four rules in every IPv6 ACL so that ICMPv6 neighbor discovery packets bypass the default drop rule.

All rules in IPv6 ACLs include the following criteria:

- **Protocol:** All rules filter on the packet's IP protocol field. Rule input options include:
  - Protocol name for a limited set of common protocols.
  - Assigned protocol number for all IP protocols.
- **Source Address:** The packet's source IPv6 address. Valid rule inputs include:
  - an IPv6 prefix (CIDR). Discontiguous masks are supported.
  - a host IP address (dotted decimal notation).
  - ***any*** to denote that the rule matches all addresses.

- **Destination Address:** The packet's destination IP address. Valid rule inputs include:
  - a subnet address (CIDR or address-mask). Discontiguous masks are supported.
  - a host IP address (dotted decimal notation).
  - ***any*** to denote that the rule matches all addresses.

All rules in IPv6 ACLs ***may*** include the following criteria:

- **Fragment:** Rules filter on the fragment bit.
- ***HOP*** Compares the packet's hop-limit value to a specified value. Comparison options include:
  - ***Equal:*** Packets match if packet value equals statement value.

The availability of the following optional criteria depends on the specified protocol:

- **Source Ports / Destination Ports:** A rule filters on ports when the specified protocol supports IP address-port combinations. Rules provide one of these port filtering values:
  - ***any*** denotes that the rule matches all ports.
  - A list of ports that matches the packet port. Maximum list size is 10 ports.
  - Negative port list. The rule matches any port not in the list. Maximum list size is 10 ports.
  - Integer (lower bound): The rule matches any port with a number larger than the integer.
  - Integer (upper bound): The rule matches any port with a number smaller than the integer.
  - Range integers: The rule matches any port whose number is between the integers.
- **Flag bits**: Rules filter TCP packets on flag bits.
- **Message type**: Rules filter ICMP type or code.
- **Tracked:** Matches packets in existing ICMP, UDP, or TCP connections. Valid in ACLs applied to the control plane. Validity in ACLs applied to the data plane varies by switch platform.

Standard IPv4 and IPv6 ACL Rule Parameters

Note When calculating the size of ACLs, be aware that Arista switches install four rules in every IPv6 ACL so that ICMPv6 neighbor discovery packets bypass the default drop rule.

Standard ACLs filter only on the source address.

MAC ACL Rule Parameters

MAC ACLs filter traffic on a packet's layer 2 header. Criteria that MAC ACLs use to filter packets include:

- **Source Address** and **Mask:** The packet's source MAC address. Valid rule inputs include:
  - MAC address range (address-mask in 3x4 dotted hexadecimal notation).
  - ***any*** to denote that the rule matches all source addresses.
- **Destination Address** and **Mask:** The packet's destination MAC address. Valid rule inputs include:
  - MAC address range (address-mask in 3x4 dotted hexadecimal notation).
  - ***any*** to denote that the rule matches all destination addresses.
- **Protocol:** The packet's protocol as specified by its EtherType field contents. Valid inputs include:
  - Protocol name for a limited set of common protocols.
  - Assigned protocol number for all protocols.

20.2.1.3 Creating and Modifying Lists

The switch provides configuration modes for creating and modifying ACLs. The command that enters an ACL configuration mode specifies the name of the list that the mode modifies. The switch saves the list to the running configuration when the configuration mode is exited.

- ACLs are created and modified in ACL configuration mode.
- Standard ACLs are created and modified in Standard-ACL-configuration mode.
- MAC ACLs are created and modified in MAC-ACL-configuration mode.

Lists that are created in one mode cannot be modified in any other mode.

A sequence number designates the rule's placement in a list. New rules are inserted into a list according to their sequence numbers. A rule's sequence number can be referenced when deleting it from a list.

Section 20.2.2 describes procedures for configuring ACLs.

20.2.1.4 Implementing Access Control Lists

Note On hardware purchased after July 3, 2017, security ACLs are only available on AlgoMatch-enabled platforms.

Note On hardware purchased after July 3, 2017, control plane ACLs are not available.

An access control list is implemented by assigning the list to an Ethernet or Port Channel interface, or to the Control Plane. The switch assigns a default ACL to the Control Plane unless the configuration contains a valid Control-Plane ACL assignment statement. Ethernet and Port Channel interfaces are not assigned an ACL by default. Standard ACLs are applied to interfaces in the same manner as other ACLs.

IPv4 and MAC ACLs are separately applied for inbound and outbound packets. An interface can be assigned multiple ACLs, with a limit of one ACL per packet direction per ACL type. Egress ACLs are supported on a subset of all available switches. The control-plane does not support egress ACLs.

Section 20.2.3 describes procedures for applying ACLs to interfaces or the control plane.

20.2.1.5 ACL Rule Tracking

ACL rule tracking determines the impact of ACL rules on the traffic accessing interfaces upon which they are applied. ACLs provide two tracking mechanisms:

- ACL logging: A syslog entry is logged when a packet matches specified ACL rules.
- ACL counters: ACL counters increment when a packet matches a rule in specified ACLs.

ACL Logging

ACL rules provide a **log** option that produces a log message when a packet matches the rule. ACL logging creates a syslog entry when a packet matches an ACL rule where logging is enabled. Packets that match a logging-enabled ACL rule are copied to the CPU by the hardware. These packets trigger the creation of a syslog entry. The information provided in the entry depends on the ACL type or the protocol specified by the ACL. Hardware rate limiting is applied to packets written to the CPU, avoiding potential DoS attacks. The rate of logging is also software limited to avoid the creation of syslog lists that are too large for practical use by human operators.

Section 20.2.2.3 describes procedures for configuring and enabling ACL logging.

ACL Counters

An ACL counter is assigned to each ACL rule. The activity of the ACL counters for rules within a list depend on the list's counter state. When the list is in counting state, the ACL counter of a rule increments when the rule matches a packet. When the list is in non-counting state, the counter does not increment. A list's counter state applies to all rules in the ACL. The initial state for new ACLs is non-counting.

When an ACL changes from counting state to non-counting state, or when the ACL is no longer applied to any interfaces that increment counters, counters for all rules in the list maintain their values and do not reset. When the ACL returns to counting mode or is applied to an interface that increments counters, the counter operation resumes from its most recent value.

Counters never decrement and are reset only through CLI commands.

Section 20.2.2.3 describes procedures for configuring and enabling ACL counters.

20.2.2 ACL Configuration

Access Control Lists are created and modified in an ACL-configuration mode. A list can be edited only in the mode where it was created. The switch provides five configuration modes for creating and modifying Access Control Lists:

- **ACL configuration mode** for IPv4 Access Control Lists.
- **IPv6-ACL configuration mode** for IPv6 Access Control Lists.
- **Std-ACL configuration mode** for Standard IPv4 Access Control Lists.
- **Std-IPv6-ACL configuration mode** for Standard IPv6 Access Control Lists.
- **MAC-ACL configuration mode** for MAC Access Control Lists.

These sections describe the creation and modification of ACLs:

- Section 20.2.2.1: Managing ACLs
- Section 20.2.2.2: Modifying an ACL
- Section 20.2.2.3: ACL Rule Tracking Configuration
- Section 20.2.2.4: Displaying ACLs

20.2.2.1 Managing ACLs

Creating and Opening a List

To create an ACL, enter one of the following commands, followed by the name of the list:

- **ip access-list** for IPv4 ACLs.
- **ipv6 access-list** for IPv6 ACLs.
- **ip access-list standard** for standard IPv4 ACLs.
- **ipv6 access-list standard** for standard IPv6 ACLs.
- **mac access-list** for MAC ACLs.

The switch enters the appropriate ACL configuration mode for the list. If the command is followed by the name of an existing ACL, subsequent commands edit that list (see Modifying an ACL for additional information).

Examples

- This command places the switch in ACL configuration mode to create an ACL named ***test1***.

```
switch(config)#ip access-list test1
switch(config-acl-test1)#
```

- This command places the switch in Standard-ACL-configuration mode to create a Standard ACL named ***stest1***.

```
switch(config)#ip access-list standard stest1
switch(config-std-acl-stest1)#
```

- This command places the switch in MAC-ACL configuration mode to create an MAC ACL named ***mtest1***.

```
switch(config)#mac access-list mtest1
switch(config-mac-acl-mtest1)#
```

Saving List Modifications

ACL configuration modes are group-change modes. Changes made in a group-change mode are saved by exiting the mode.

Important! After exiting ACL mode, the running-config file must be saved to the startup configuration file to preserve an ACL after a system restart.

Example

- The second example in Adding a Rule (page 991) results in this edited ACL:

```
switch(config-acl-test1)#show
IP Access List test1
        10 permit ip 10.10.10.0/24 any
        20 permit ip 10.30.10.0/24 host 10.20.10.1
        30 deny ip host 10.10.10.1 host 10.20.10.1
        40 permit ip any any
```

Because the changes were not yet saved, the ACL remains empty, as shown by **show ip access-lists**.

```
switch(config-acl-test1)#show ip access-lists test1
switch(config-acl-test1)#
```

To save all current changes to the ACL and exit ACL configuration mode, type **exit**.

```
switch(config-acl-test1)#exit
switch(config)#show ip access-lists test1
IP Access List test1
        10 permit ip 10.10.10.0/24 any
        20 permit ip 10.30.10.0/24 host 10.20.10.1
        30 deny ip host 10.10.10.1 host 10.20.10.1
        40 permit ip any any
```

Discarding List Changes

The **abort** command exits ACL configuration mode without saving pending changes.

Example

- Example 2 in Adding a Rule (page 991) results in this edited ACL:

```
switch(config-acl-test1)#show
IP Access List test1
        10 permit ip 10.10.10.0/24 any
        20 permit ip 10.30.10.0/24 host 10.20.10.1
        30 deny ip host 10.10.10.1 host 10.20.10.1
        40 permit ip any any
```

To discard the changes, enter **abort**. If the ACL existed before entering ACL-configuration mode, **abort** restores the version that existed before entering ACL-configuration mode. Otherwise, **show ip access-lists** shows the ACL was not created.

```
switch(config-acl-test1)#abort
switch(config)#
```

20.2.2.2 Modifying an ACL

An existing ACL, including those currently applied to interfaces, can be modified by entering the appropriate configuration mode for the ACL as described in Creating and Opening a List. By default, while an ACL is being modified all traffic is blocked on any interface to which the ACL has been applied.

Permit All Traffic During ACL Update

Because blocking ports during ACL modifications can result in packet loss and can interfere with features such as routing and dynamic NAT, 7150 series switches can be configured instead to permit all traffic on Ethernet and VLAN interfaces while ACLs applied to those interfaces are being modified. This is done with the **hardware access-list update default-result permit** command.

These commands add deny rules to the appropriate ACL:

- **deny (IPv4 ACL)** adds a deny rule to an IPv4 ACL.
- **deny (IPv6 ACL)** adds a deny rule to an IPv6 ACL.
- **deny (Standard IPv4 ACL)** adds a deny rule to an IPv4 standard ACL.
- **deny (Standard IPv6 ACL)** adds a deny rule to an IPv6 standard ACL.
- **deny (MAC ACL)** adds a deny rule to a MAC ACL.

These commands add permit rules to the appropriate ACL:

- **permit (IPv4 ACL)** adds a permit rule to an IPv4 ACL.
- **permit (IPv6 ACL)** adds a permit rule to an IPv6 ACL.
- **permit (Standard IPv4 ACL)** adds a permit rule to an IPv4 standard ACL.
- **permit (Standard IPv6 ACL)** adds a permit rule to an IPv6 standard ACL.
- **permit (MAC ACL)** adds a permit rule to a MAC ACL.

Adding a Rule

To append a rule to the end of a list, enter the rule without a sequence number while in ACL configuration mode for the list. The new rule's sequence number is derived by adding 10 to the last rule's sequence number.

Examples

- This command configures the switch to permit all traffic during ACL modifications on interfaces to which the ACL has been applied. The rules in modified ACLs are applied after exiting ACL configuration mode, and after the ACL rules have been populated in hardware.

```
switch(config)#hardware access-list update default-result permit
```

- These commands enter the first three rules into a new ACL.

```
switch(config-acl-test1)#permit ip 10.10.10.0/24 any
switch(config-acl-test1)#permit ip any host 10.20.10.1
switch(config-acl-test1)#deny ip host 10.10.10.1 host 10.20.10.1
```

To view the edited list, type **show**.

```
switch(config-acl-test1)#show
IP Access List test1
        10 permit ip 10.10.10.0/24 any
        20 permit ip any host 10.20.10.1
        30 deny ip host 10.10.10.1 host 10.20.10.1
```

- This command appends a rule to the ACL. The new rule's sequence number is 40.

```
switch(config-acl-test1)#permit ip any any
switch(config-acl-test1)#show
IP Access List test1
        10 permit ip 10.10.10.0/24 any
        20 permit ip any host 10.20.10.1
        30 deny ip host 10.10.10.1 host 10.20.10.1
        40 permit ip any any
```

Inserting a Rule

To insert a rule into a ACL, enter the rule with a sequence number between the existing rules' numbers.

Example

- This command inserts a rule between the first two rules by assigning it the sequence number 15.

```
Switch(config-acl-test1)#15 permit ip 10.30.10.0/24 host 10.20.10.1
Switch(config-acl-test1)#show
IP Access List test1
        10 permit ip 10.10.10.0/24 any
        15 permit ip 10.30.10.0/24 host 10.20.10.1
        20 permit ip any host 10.20.10.1
        30 deny ip host 10.10.10.1 host 10.20.10.1
        40 permit ip any any
```

Deleting a Rule

To remove a rule from the current ACL, perform one of these commands:

- Enter **no**, followed by the sequence number of the rule to be deleted.
- Enter **no**, followed by the rule be deleted.
- Enter **default**, followed by the rule to be deleted.

Example

- These equivalent commands remove rule 20 from the list.

```
switch(config-acl-test1)#no 20

switch(config-acl-test1)#no permit ip any host 10.20.10.1

switch(config-acl-test1)#default permit ip any host 10.20.10.1
```

This ACL results from entering one of the preceding commands.

```
switch(config-acl-test1)#show
ip access list test1
        10 permit ip 10.10.10.0/24 any
        15 permit ip 10.30.10.0/24 host 10.20.10.1
        30 deny ip host 10.10.10.1 host 10.20.10.1
        40 permit ip any any
```

Resequencing Rule Numbers

Sequence numbers determine the order of the rules in an Access Control List. After a list editing session where existing rules are deleted and new rules are inserted between existing rules, the sequence number distribution may not be uniform. Resequencing rule numbers changes adjusts the sequence number of rules to provide a constant difference between adjacent rules. The **resequence (ACLs)** command adjusts the sequence numbers of ACL rules.

Example

- The **resequence** command renumbers rules in the test1 ACL. The sequence number of the first rule is 100; subsequent rules numbers are incremented by 20.

```
switch(config-acl-test1)#show
IP Access List test1
        10 permit ip 10.10.10.0/24 any
        25 permit ip any host 10.20.10.1
        30 deny ip host 10.10.10.1 host 10.20.10.1
        50 permit ip any any
        90 remark end of list
switch(config-acl-test1)#resequence 100 20
switch(config-acl-test1)#show
IP Access List test1
        100 permit ip 10.10.10.0/24 any
        120 permit ip any host 10.20.10.1
        140 deny ip host 10.10.10.1 host 10.20.10.1
        160 permit ip any any
        180 remark end of list
```

20.2.2.3 ACL Rule Tracking Configuration

ACL rules provide a **log** option that produces a syslog message about the packets matching packet. ACL logging creates a syslog entry when a packet matches an ACL rule with logging enabled.

This feature is currently available on Arad switches and on 7100 series switches. On 7100 series switches, matches are logged only on ingress, not on egress.

Example

- This command creates an ACL rule with logging enabled.

```
switch(config-acl-test1)#15 permit ip 10.30.10.0/24 host 10.20.10.1 log
switch(config-acl-test1)#
```

The format of the generated syslog message depends on the ACL type and the specified protocol:

- Messages generated by a TCP or UDP packet matching an IP ACL use this format:

  IPACCESS: list *acl intf filter protocol src-ip*(*src_port*) **->** *dst-ip*(*dst_port*)

- Messages generated by ICMP packets matching an IP ACL use this format:

  IPACCESS: list *acl intf filter* **icmp** *src-ip*(*src-port*) **->** *dst-ip*(*dst-port*) **type=***n* **code=***m*

- Messages generated by all other IP packets matching an IP ACL use this format:

  IPACCESS: list *acl intf filter protocol src-ip* **->** *dst-ip*

- Messages generated by packets matching a MAC ACL use this format:

  MACACCESS: list *acl intf filter vlan ether src_mac* **->** *dst_mac*

- Messages generated by a TCP or UDP packet matching a MAC ACL use this format:

  MACACCESS: list *acl intf filter vlan ether ip-prt src-mac src-ip* **:** *src-prt* **->** *dst-mac dst-ip* **:** *dst-prt*

- Messages generated by any other IP packet matching a MAC ACL use this format:

  MACACCESS: list *acl intf filter vlan ether src_mac src_ip* **->** *dst_mac dst_ip*

Variables in the syslog messages display the following values:

- *acl* Name of ACL.
- *intf* Name of interface that received the packet.
- *filter* Action triggered by ACL (**denied** or **permitted**).
- *protocol* IP protocol specified by packet.
- *vlan* Number of VLAN receiving packet.
- *ether* EtherType protocol specified by packet.
- *src-ip* and *dst-ip* source and destination IP addresses.
- *src-prt* and *dst-prt* source and destination ports.
- *src-mac* and *dst-mac* source and destination MAC addresses.

ACLs provide a command that configures its counter state (counting or non-counting). The counter state applies to all rules in the ACL. The initial state for new ACLs is non-counting.

The **statistics per-entry (ACL configuration modes)** command places the ACL in counting mode.

- This command places the configuration mode ACL in counting mode.

```
switch(config-acl-test1)#statistics per-entry
switch(config-acl-test1)#exit
switch(config-acl-test1)#show ip access-list test1
IP Access List test1
        statistics per-entry
        10 permit ip 10.10.10.0/24 any
        20 permit ip any host 10.20.10.1
        30 deny ip host 10.10.10.1 host 10.20.10.1
        40 permit ip any any
        50 remark end of list
```

The **clear ip access-lists counters** and **clear ipv6 access-lists counters** commands set the IP access list counters to zero for the specified IP access list.

- This command clears the ACL counter for the test1 ACL.

```
switch(config)#clear ip access-lists counters test1
switch(config)#
```

20.2.2.4 Displaying ACLs

ACLs can be displayed by a **show running-config** command. The **show ip access-lists** also displays ACL rosters and contents, as specified by command parameters.

When editing an ACL, the **show (ACL configuration modes)** command displays the current or pending list, as specified by command parameters.

Displaying a List of ACLs

To display the roster of ACLs on the switch, enter **show ip access-lists** with the **summary** option.

Example

- This command lists the available Access Control Lists.

```
switch(config)#show ip access-list summary
IPV4 ACL default-control-plane-acl
        Total rules configured: 12
        Configured on: control-plane
        Active on    : control-plane

IPV4 ACL list2
        Total rules configured: 3

IPV4 ACL test1
        Total rules configured: 6

IPV4 ACL test_1
        Total rules configured: 1

IPV4 ACL test_3
        Total rules configured: 0

switch(config)#
```

Displaying Contents of an ACL

These commands display ACL contents.

- **show ip access-lists**
- **show ipv6 access-lists**
- **show mac access-lists**

Each command can display the contents of one ACL or of all ACLs of the type specified by the command:

- To display the contents of one ACL, enter **show ip access-lists** followed by the name of the ACL.
- To display the contents of all ACLs on the switch, enter the command without any options.

ACLs that are in counting mode display the number of inbound packets each rule in the list matched and the elapsed time since the last match.

Example

- This command displays the rules in the ***default-control-plane-acl*** ACL.

```
switch#show ip access-lists default-control-plane-acl
IP Access List default-control-plane-acl [readonly]
        statistics per-entry
        10 permit icmp any any
        20 permit ip any any tracked [match 1725, 0:00:00 ago]
        30 permit ospf any any
        40 permit tcp any any eq ssh telnet www snmp bgp https
        50 permit udp any any eq bootps bootpc snmp [match 993, 0:00:29 ago]
        60 permit tcp any any eq mlag ttl eq 255
        70 permit udp any any eq mlag ttl eq 255
        80 permit vrrp any any
        90 permit ahp any any
        100 permit pim any any
        110 permit igmp any any [match 1316, 0:00:23 ago]
        120 permit tcp any any range 5900 5910
```

- This command displays the rules in all ACLs on the switch.

```
switch#show ip access-lists
IP Access List default-control-plane-acl [readonly]
        statistics per-entry
        10 permit icmp any any
        20 permit ip any any tracked [match 1371, 0:00:00 ago]
        30 permit ospf any any
        40 permit tcp any any eq ssh telnet www snmp bgp https
        50 permit udp any any eq bootps bootpc snmp
        60 permit tcp any any eq mlag ttl eq 255
        70 permit udp any any eq mlag ttl eq 255
        80 permit vrrp any any
        90 permit ahp any any
        100 permit pim any any
        110 permit igmp any any [match 1316, 0:00:23 ago]
        120 permit tcp any any range 5900 5910

IP Access List list2
        10 permit ip 10.10.10.0/24 any
        20 permit ip 10.30.10.0/24 host 10.20.10.1
        30 permit ip any host 10.20.10.1
        40 deny ip host 10.10.10.1 host 10.20.10.1
        50 permit ip any any

IP Access List test1
            <-------OUTPUT OMITTED FROM EXAMPLE-------->

Switch(config)#
```

Displaying ACL Modifications

While editing an ACL in ACL-configuration mode, the **show (ACL configuration modes)** command provides options for displaying ACL contents.

- To display the list, as modified in ACL configuration mode, enter **show** or **show pending**.
- To display the list, as stored in ***running-config***, enter **show active**.
- To display differences between the pending list and the stored list, enter **show diff**.

Examples

The examples in this section assume these ACL commands were previously entered.

These commands are stored in the configuration:

```
10 permit ip 10.10.10.0/24 any
20 permit ip any host 10.21.10.1
30 deny ip host 10.10.10.1 host 10.20.10.1
40 permit ip any any
50 remark end of list
```

The current edit session removed this command. This change is not yet stored to running-config:

```
20 permit ip any host 10.21.10.1
```

The current edit session added these commands ACL. They are not yet stored to running-config:

```
20 permit ip 10.10.0.0/16 any
25 permit tcp 10.10.20.0/24 any
45 deny pim 239.24.124.0/24 10.5.8.4/30
```

- This command displays the pending ACL, as modified in ACL configuration mode.

```
switch(config-acl-test_1)#show pending
IP Access List test_1
        10 permit ip 10.10.10.0/24 any
        20 permit ip 10.10.0.0/16 any
        25 permit tcp 10.10.20.0/24 any
        30 deny ip host 10.10.10.1 host 10.20.10.1
        40 permit ip any any
        45 deny pim 239.24.124.0/24 10.5.8.4/30
        50 remark end of list
```

- This command displays the ACL, as stored in the configuration.

```
switch(config-acl-test_1)#show active
IP Access List test_1
        10 permit ip 10.10.10.0/24 any
        20 permit ip any host 10.21.10.1
        30 deny ip host 10.10.10.1 host 10.20.10.1
        40 permit ip any any
        50 remark end of list
```

- This command displays the difference between the saved and modified ACLs.
  - Rules added to the pending list are denoted with a plus sign (+).
  - Rules removed from the saved list are denoted with a minus sign (-).

```
switch(config-acl-test_1)#show diff
---
+++
@@ -1,7 +1,9 @@
 IP Access List test_1
         10 permit ip 10.10.10.0/24 any
-        20 permit ip any host 10.21.10.1
+        20 permit ip 10.10.0.0/16 any
+        25 permit tcp 10.10.20.0/24 any
         30 deny ip host 10.10.10.1 host 10.20.10.1
         40 permit ip any any
+        45 deny pim 239.24.124.0/24 10.5.8.4/30
```

20.2.3 Applying ACLs

Access Control Lists become active when they are assigned to an interface or the Control Plane. This section describes the process of adding and removing ACL interface assignments.

Note On hardware purchased after July 3, 2017, security ACLs are only available on AlgoMatch-enabled platforms.

Note On hardware purchased after July 3, 2017, control plane ACLs are not available.

Applying an ACL to an Interface

The switch must be in interface configuration mode to assign an ACL to an interface.

- The **ip access-group** command applies the specified IP or standard IP ACL to the configuration mode interface.
- The **mac access-group** command applies the specified MAC ACL to the configuration mode interface.

IPv4, IPv6, and MAC ACLs are separately applied for inbound and outbound packets. An interface can be assigned multiple ACLs, with a limit of one ACL per packet direction per ACL type. Egress ACLs are supported on a subset of all available switches. IPv6 egress ACLs have limited availability, and IPv6 egress ACLs applied to routed interfaces across the same chip on the DCS-7500E and the DCS-7280E series can be shared. In addition to that, the DSCP value can match on IPv6 egress ACLs. This result in a more efficient utilization of system resources, and is particularly useful for environments with few, potentially large, IPv6 egress ACLs applied across multiple routed interfaces.

Example

- These commands assign ***test1*** ACL to Ethernet interface 3, then verify the assignment.

```
switch(config)#interface ethernet 3
switch(config-if-Et3)#ip access-group test1 in
switch(config-if-Et3)#show running-config interfaces ethernet 3
interface Ethernet3
   ip access-group test1 in
switch(config-if-Et3)#
```

- This command enables shared ACLs.

```
switch(config)#hardware access-list resource sharing vlan ipv6 out
switch(config)#
```

- This command disables shared ACLs.

```
switch(config)#no hardware access-list resource sharing vlan ipv6 out
switch(config)#
```

Removing an ACL from an Interface

The **no ip access-group** command removes an IP ACL assignment statement from ***running-config*** for the configuration mode interface. After an ACL is removed, the interface is not associated with an IP ACL.

The **no mac ip access-group** command removes a MAC ACL assignment statement from ***running-config*** for the configuration mode interface. After a MAC ACL is removed, the interface is not associated with an MAC ACL.

To remove an ACL from the control plane, enter the **no ip access-group** command in control plane configuration mode. Removing the control plane ACL command from ***running-config*** reinstates ***default-control-plane-acl*** as the control plane ACL.

Examples

- These commands remove the assigned IPv4 ACL from Ethernet interface 3.

```
switch(config)#interface ethernet 3
switch(config-if-Et3)#no ip access-group test in
switch(config-if-Et3)#
```

- These commands place the switch in control plane configuration mode and remove the ACL assignment from ***running-config***, restoring ***default-control-plane-acl*** as the Control Place ACL.

```
switch(config)#control-plane
switch(config-cp)#no ip access-group test_cp in
switch(config-cp)#
```

20.3 Service ACLs

Note Service ACLs are only available on EOS versions 4.19.0F and above.

These sections describe Service ACLs:

- Section 20.3.1: Service ACL Description
- Section 20.3.2: Configuring Service ACLs and Displaying Status and Counters

20.3.1 Service ACL Description

Service ACL enforcement is a feature added to a control plane service (the SSH server, the SNMP server, routing protocols, etc) that allows the switch administrator to restrict the processing of packets and connections by the control plane processes that implement that service. The control plane program run by the control plane process checks already received packets and connections against a user configurable Access Control List (ACL), a Service ACL. The Service ACL contains permit and deny rules matching any of the source address, destination address, and TCP or UDP ports of received packets or connections. After receiving a packet or connection, the control plane process evaluates the packet or connection against the rules of the Service ACL configured for the control plane process, and if the received packet or connection matches a deny rule the control plane process drops or closes it without further processing.

Control Plane Process Enforced Access Control enables the system administrator to restrict which systems on the network can access the services provided by the switch. Each service has its own access control list, giving the system administrator fine grained control over access to the switch's control plane services. The CLI for this uses the familiar pattern of access control lists assigned for a specific purpose, in this case for each control plane service.

20.3.2 Configuring Service ACLs and Displaying Status and Counters

20.3.2.1 SSH Server

To apply the SSH server Service ACLs for IPv4 and IPv6 traffic, use the **ip access-group (Service ACLs)** and **ipv6 access-group (Service ACLs)** commands in mgt-ssh configuration mode as shown below.

```
(config)# management ssh
(config-mgmt-ssh)# ip access-group <acl_name> [vrf <vrf_name>] in
(config-mgmt-ssh)# ipv6 access-group <acl_name> [vrf <vrf_name>] in
```

In EOS 4.19.0, all VRFs are required to use the same SSH server Service ACL. The Service ACL assigned without the **vrf** keyword is applied to all VRFs where the SSH server is enabled.

To display the status and counters of the SSH server Service ACLs, use the following commands.

```
(switch)>show management ssh ip access-list
(switch)>show management ssh ipv6 access-list
```

20.3.2.2 SNMP Server

To apply the SNMP server Service ACLs to restrict which hosts can access SNMP services on the switch, use the **snmp-server community** command as shown below.

```
(config)# snmp-server community <community-name> [view <viewname>] [ro | rw]
<acl_name>
(config)# snmp-server community <community-name> [view <viewname>] [ro | rw] ipv6
<ipv6_acl_name>
```

20.3.2.3 EAPI

To apply Service ACLs to the EOS application programming interface (EAPI) server, use the **ip access-group (Service ACLs)** and **ipv6 access-group (Service ACLs)** commands as shown below.

```
(config)#management api http-commands
(config-mgmt-api-http-cmds)#vrf <vrf_name>
(config-mgmt-api-http-cmds-vrf-<vrf>)#ip access-group <acl_name>
(config-mgmt-api-http-cmds-vrf-<vrf>)#ipv6 access-group <ipv6_acl_name>
```

Note To configure a Service ACL for the EAPI server in the default VRF, use the **vrf default** command to enter the per-VRF configuration mode for the default VRF before using the **ip access-group (Service ACLs)** or **ipv6 access-group (Service ACLs)** command.

To display the status and counters of the EAPI server Service ACLs, use the following commands.

```
(switch)> show management api http-commands ip access-list
(switch)> show management api http-commands ipv6 access-list
```

20.3.2.4 BGP

To apply Service ACLs for controlling connections to the BGP routing protocol agent, use the **ip access-group (Service ACLs)** and **ipv6 access-group (Service ACLs)** commands as shown below.

```
(config)# router bgp <asn>
(config-router-bgp)# ip access-group <acl_name>
(config-router-bgp)# ipv6 access-group <ipv6_acl_name>
(config-router-bgp)# vrf <vrf_name>
(config-router-bgp-vrf-<vrf>)# ip access-group <acl_name>
(config-router-bgp-vrf-<vrf>)# ipv6 access-group <ipv6_acl_name>
```

To display the status and counters of the BGP routing protocol Service ACLs, use the following commands.

```
(switch)> show bgp ipv4 access-list
(switch)> show bgp ipv6 access-list
```

20.3.2.5 OSPF

To apply Service ACLs for controlling packets processed by the OSPF routing protocol agent, use the **ip access-group (Service ACLs)** and **ipv6 access-group (Service ACLs)** commands as shown below.

```
(config)# router ospf <id>
(config-router-ospf)# ip access-group <acl_name>
(config-router-ospf)# ipv6 access-group <ipv6_acl_name>
```

When using VRFs, each per-VRF OSPF instance must be assigned its Service ACL explicitly.

To display the status and counters of the OSPF routing protocol Service ACLs, use the following commands.

```
(switch)> show ospf ipv4 access-list
(switch)> show ospf ipv6 access-list
```

20.3.2.6 PIM

To apply Service ACLs for controlling packets processed by the PIM routing protocol agent, use the **ip access-group (Service ACLs)** and **ipv6 access-group (Service ACLs)** commands as shown below.

```
(config)# router pim
(config-router-pim)# ip pim access-group <acl_name>
(config-router-pim)# vrf <vrf_name>
(config-router-pim-vrf-<vrf>)# ip pim access-group <acl_name>
```

To display the status and counters of the PIM routing protocol Service ACLs, use the following commands.

```
(switch)> show ip pim access-list
```

20.3.2.7 IGMP

To apply Service ACLs for controlling packets processed by the IGMP management protocol agent, use the **ip igmp access-group** command as shown below.

```
(config)# router igmp
(config-router-igmp)# ip igmp access-group <acl_name>
(config-router-igmp)# vrf <vrf_name>
(config-router-igmp-vrf-<vrf>)# ip igmp access-group <acl_name>
```

To display the status and counters of the IGMP management protocol Service ACLs, use the following commands.

```
(switch)> show ip igmp access-list
```

20.3.2.8 DHCP Relay

To apply Service ACLs for controlling packets processed by the DHCP relay agent, use the **ip dhcp relay access-group** and **ipv6 dhcp relay access-group** commands as shown below.

```
(config)# ip dhcp relay access-group <acl_name> [vrf <vrf_name>]
(config)# ipv6 dhcp relay access-group <acl_name> [vrf <vrf_name>]
```

To display the status and counters of the DHCP relay agent Service ACLs, use the following commands.

```
(switch)> show ip dhcp relay access-list
(switch)> show ipv6 dhcp relay access-list
```

20.3.2.9 LDP

To apply Service ACLs for controlling packets and connections processed by the LDP MPLS label distribution protocol, use the **ip access-group (Service ACLs)** command as shown below.

```
(config)# mpls ldp
(config-mpls-ldp)# ip access-group <acl_name>
```

To display the status and counters of the LDP Service ACLs, use the following command.

```
(switch)> show mpls ldp access-list
```

20.3.2.10 LANZ

To apply Service ACLs for controlling connections accepted by the LANZ agent, use the **ip access-group (Service ACLs)** and **ipv6 access-group (Service ACLs)** commands as shown below.

```
(config)# queue-monitor streaming
(config-qm-streaming)# ip access-group <acl_name>
(config-qm-streaming)# ipv6 access-group <ipv6_acl_name>
```

To display the status and counters of the LDP Service ACLs, use the following command.

```
(switch)> show queue-monitor streaming access-lists
```

20.3.2.11 MPLS Ping and Traceroute

To apply Service ACLs for controlling connections accepted by the MPLS Ping agent, use the **ip access-group (Service ACLs)** and **ipv6 access-group (Service ACLs)** commands as shown below.

```
(config)# mpls ping
(config-mpls-ping)# ip access-group <acl_name> [vrf <vrf_name>]
(config-mpls-ping)# ipv6 access-group <ipv6_acl_name> [vrf <vrf_name>]
```

20.3.2.12 Telnet Server

To apply Service ACLs to the Telnet server, use the **ip access-group (Service ACLs)** and **ipv6 access-group (Service ACLs)** commands as shown below.

```
(config)# management telnet
(config-mgmt-telnet)# ip access-group <acl_name> [vrf <vrf_name>] in
(config-mgmt-telnet)# ipv6 access-group <ipv6_acl_name> [vrf <vrf_name>] in
```

In EOS 4.19.0, all VRFs are required to use the same Telnet server Service ACL. The Service ACL assigned without the **vrf** keyword is applied to all VRFs where the Telnet server is enabled.

To display the status and counters of the LDP Service ACLs, use the following commands.

```
(switch)>show management telnet ip access-list
(switch)>show management telnet ipv6 access-list
```

20.4 Route Maps

A route map is an ordered set of rules that control the redistribution of IP routes into a protocol domain on the basis of such criteria as route metrics, access control lists, next hop addresses, and route tags. Route maps can also alter parameters of routes as they are redistributed.

These sections describe the route map implementation:

- Section 20.4.1 describes route maps.
- Section 20.4.2 describes the route map configuration process.
- Section 20.4.3 describes the usage of route maps.

20.4.1 Route Map Description

Route maps are composed of route map clauses, each of which consists of a list of match and set statements.

Route Map Clauses

Route map clauses are categorized by the resolution of routes that the clause filters.

- Permit clauses facilitate the redistribution of matched routes.
- Deny clauses prevent the redistribution of matched routes.

Route map clause elements include name, sequence number, filter type, match statements, set statements, and continue statements.

- **name** identifies the route map to which the clause belongs.
- **sequence number** designates the clause's placement within the route map.
- **filter type** specifies the route resolution. Valid types are ***permit*** and ***deny***.
- **match statements** specify criteria that select routes that the clause is evaluating for redistribution.
- **set statements** modify route parameters for redistributed routes.
- **continue statements** prolong the route map evaluation of routes that match a clause.

Clauses filter routes for redistribution. Routes that clauses pass are redistributed (permit clauses) or rejected (deny clauses). Routes that clauses fail are filtered by the next clause in the route map.

- When a clause does not contain a **match** statement, the clause passes all routes.
- When a clause contains a single **match** statement that lists a single object, the clause passes routes whose parameters match the object.
- When a clause contains a single **match** statement that lists multiple objects, the clauses passes routes whose parameters match at least one object.
- When a clause contains multiple **match** statements, the clause passes routes whose parameters match all match statements.

Set statements modify parameters for redistributed routes. Set statements are valid in permit clauses.

Example

- The following route map clause is named MAP_1 with sequence number 10. The clause matches all routes from BGP Autonomous system 10 and redistributes them with a local preference set to 100. Routes that do not match the clause are evaluated against the next clause in the route map.

```
route-map MAP_1 permit 10
   match as 10
   set local-preference 100
```

Route Maps with Multiple Clauses

A route map consists of clauses with the same name and different sequence numbers. Clauses filter routes in ascending order of their sequence numbers. When a clause passes a route, the redistribution action is performed as specified by the filter type and all subsequent clauses are ignored. When the clause fails the route, the clause with the next lowest sequence number filters the route.

All route maps have an implied final clause that contains a single deny clause with no match statement. This denies redistribution to routes that are not passed by any clause.

Example

- The following route map is named MAP_1 with two permit clauses. Routes that do not match either clause are denied redistribution into the target protocol domain.

```
route-map MAP_1 permit 10
   match as 10
   set local-preference 100
!
route-map MAP_1 permit 20
   match metric-type type-1
   match as 100
```

Section 20.4.2 describes route map configuration procedures.

Route Maps with Multiple Clauses and Continue Statements

Route map clauses that contain a **continue (route-map)** command support additional route map evaluation of routes whose parameters meet the clauses's matching statements. Routes that match a clause containing a **continue** statement are evaluated against the clause specified by the **continue** statement.

When a route matches multiple route-map clauses, the filter action (deny or permit) is determined by the last clause that the route matches. The **set** statements in all clauses matching the route are applied to the route after the route map evaluation is complete. Multiple set statements are applied in the same order by which the route was evaluated against the clauses containing them.

Example

- The following route map is named MAP_1 with a permit clause and a deny clause. The permit clause contains a continue statement. Routes that map clause 10 are evaluated against clause 20.

```
route-map MAP_2 permit 10
   match as 10
   continue 20
   set local-preference 100
!
route-map MAP_2 deny 20
   match metric-type type-1
   match as 100
```

  The route is redistributed if it passes clause 10 and is rejected by clause 20. The route is denied redistribution in all other instances. The **continue** statement guarantees the evaluation of all routes against both clauses.

20.4.2 Route Map Configuration

Route maps are created and modified in route map configuration mode. These sections describe the configuration mode and its commands.

- Section 20.4.2.1: Route Map Creation and Editing

- Section 20.4.2.2: Modifying Route Map Components

20.4.2.1 Route Map Creation and Editing

Creating a Route Map Clause

To create a route map, enter **route-map** followed by the map name and filter type (**deny** or **permit**). The default sequence number is assigned to the clause if the command does not include a number.

Example

- This command places the switch in route map configuration mode to create a route map clause named ***map1*** with a sequence number of 50.

```
switch(config)#route-map map1 permit 50
switch(config-route-map-map1)#
```

Editing a Route Map Clause

To edit an existing route map clause, enter **route-map** with the map's name and clause's number. The switch enters route-map configuration mode for the clause. Subsequent **match (route-map)** and **set (route-map)** commands add the corresponding statements to the clause.

The **show** command displays contents of the existing route map.

Example

- This command places the switch in route map configuration mode to edit an existing route map clause. The **show** command displays contents of all clauses in the route map.

```
switch(config)#route-map MAP2
switch(config-route-map-MAP2)#show
route-map MAP2 deny 10
  Match clauses:
    match as 10
    match tag 333
  Set clauses:
    set local-preference 100
switch(config-route-map-MAP2)#
```

Saving Route Map Modifications

Route map configuration mode is a group-change mode. Changes are saved by exiting the mode.

Example

- The first command creates the ***map1*** clause with sequence number of 10. The second command is not yet saved to the route map, as displayed by the **show** command.

```
switch(config)#route-map map1 permit
switch(config-route-map-map1)#match as 100
switch(config-route-map-map1)#show


switch(config-route-map-map1)#
```

The **exit** command saves the **match** command.

```
switch(config-route-map-map1)#exit
switch(config)#show route-map map1
route-map map1 permit 10
  Match clauses:
    match as 100
  Set clauses:
switch(config)#
```

Discarding Route Map Modifications

The **abort** command discards all pending changes and exits route map configuration mode.

Example

- The **abort** command discards the pending **match** command and restores the original route map.

```
switch(config)#route-map map1 permit
switch(config-route-map-map1)#match as 100
switch(config-route-map-map1)#abort
switch(config)#show route-map map1
switch(config)#
```

20.4.2.2 Modifying Route Map Components

These commands add rules to the configuration mode route-map:

- **match (route-map)** adds a match rule to a route map.
- **set (route-map)** adds a set rule to a route map.

Editing a Clause

To append a rule to a list, enter the rule without a sequence number in route map configuration mode for the list. The new rule's sequence number is derived by adding 10 to the last rule's sequence number.

Example

- These commands enter route map configuration mode for an existing route map clause, then adds a set and match statement to the clause.

```
switch(config)#route-map Map1 permit 20
switch(config-route-map-Map1)#set ip next-hop 10.2.4.5
switch(config-route-map-Map1)#match tag 500
switch(config-route-map-Map1)#
```

This command displays the contents of the clause before saving the statements.

```
switch(config-route-map-Map1)#show
route-map Map1 deny 10
  Match clauses:
    match as 10
    match tag 333
  Set clauses:
    set local-preference 100
route-map Map1 permit 20
  Match clauses:
    match metric-type type-1
    match as-path List1
  Set clauses:
```

This command exits route map configuration mode, saves the new statements, and displays the contents of the clause after the statements are saved.

```
switch(config-route-map-Map1)#exit
switch(config)#show route-map Map1
route-map Map1 deny 10
  Match clauses:
    match as 10
    match tag 333
  Set clauses:
    set local-preference 100
route-map Map1 permit 20
  Match clauses:
    match metric-type type-1
    match as-path List1
    match tag 500
  Set clauses:
    set ip next-hop 10.2.4.5
switch(config)#
```

Inserting a Clause

To insert a new clause into an existing route map, create a new clause with a sequence number that differs from any existing clause in the map.

Example

- This command adds clause 50 to the ***Map1*** route map, then displays the new route map.

```
switch(config)#route-map Map1 permit 50
switch(config-route-map-Map1)#match as 150
switch(config-route-map-Map1)#exit
switch(config)#show route-map Map1
route-map Map1 deny 10
  Match clauses:
    match as 10
    match tag 333
  Set clauses:
    set local-preference 100
route-map Map1 permit 50
  Match clauses:
    match as 150
  Set clauses:
switch(config)#
```

Deleting Route Map Components

To remove a component from a route map, perform one of the following:

- To remove a statement from a clause, enter **no**, followed by the statement to be removed.
- To remove a clause, enter **no** followed by the sequence number of the clause to be removed.
- To remove a route map, enter **no** followed by the route map without a sequence number.

20.4.3 Using Route Maps

Protocol redistribution commands include a route map parameter that determines the routes to be redistributed into the specified protocol domain.

Example

- This command uses ***Map1*** route map to select OSPFv2 routes for redistribution into BGP AS1.

```
switch(config)#router bgp 1
switch(config-router-bgp)#redistribute ospf route-map Map1
switch(config-router-bgp)#exit
switch(config)#
```

20.5 Prefix Lists

A prefix list is an ordered set of rules that defines route redistribution access for a specified IP address space. A prefix list rules consists of a filter action (deny or permit), an address space identifier (IPv4 subnet address or IPv6 prefix), and a sequence number.

Prefix lists are referenced by route-map match commands when filtering routes for redistribution.

- Section 20.5.1 describes the prefix list configuration process.
- Section 20.5.2 describes the use of prefix lists.

20.5.1 Prefix List Configuration

A prefix list is an ordered set of rules that defines route redistribution access for a specified IP address space. A prefix list rule consists of a filter action (deny or permit), a network address (IPv4 subnet or IPv6 prefix), and a sequence number. A rule may also include a alternate mask size.

The switch supports IPv4 and IPv6 prefix lists. IPv4 lists are constructed in global configuration mode, whereas the switch is placed in a Prefix-list configuration mode to create and edit IPv6 prefix lists.

20.5.1.1 IPv4 Prefix Lists

IPv4 prefix lists are created or modified by adding an IPv4 prefix list rule in global configuration mode. Each rule includes the name of a prefix list, in addition to the sequence number, network address, and filter action. A list consists of all rules that have the same prefix list name.

The **ip prefix-list** command creates a prefix list or adds a rule to an existing list. Route map match statements use prefix lists to filter routes for redistribution into OSPF, RIP, or BGP domains.

Example

- These commands creates two IPv4 prefix lists: a two-rule list named route-one and a three-rule list named route-two.

```
switch(config)#ip prefix-list route-one seq 10 deny 10.1.1.1/24
switch(config)#ip prefix-list route-one seq 20 deny 10.1.2.1/16
switch(config)#ip prefix-list route-two seq 10 deny 10.1.1.0/24 ge 26 le 30
switch(config)#ip prefix-list route-two seq 20 deny 10.1.0.0/16
switch(config)#ip prefix-list route-two seq 30 permit 12.15.4.9/32
switch(config)#ip prefix-list route-two seq 40 deny 1.1.1.0/24
switch(config)#show running-config
             <-------OUTPUT OMITTED FROM EXAMPLE-------->
!
ip prefix-list route-one seq 10 deny 10.1.1.0/24
ip prefix-list route-one seq 20 deny 10.1.0.0/16
ip prefix-list route-two seq 10 deny 10.1.1.0/24 ge 26 le 30
ip prefix-list route-two seq 20 deny 10.1.0.0/16
ip prefix-list route-two seq 30 permit 12.15.4.9/32
ip prefix-list route-two seq 40 deny 1.1.1.0/24
!
             <-------OUTPUT OMITTED FROM EXAMPLE-------->
switch(config)#
```

IPv4 prefix lists are referenced in route-map **match (route-map)** commands.

20.5.1.2 IPv6 Prefix Lists

Creating an IPv6 Prefix List

The switch provides IPv6 prefix-list configuration mode for creating and modifying IPv6 prefix lists. A list can be edited only in the mode where it was created.

To create an IP ACL, enter the **ipv6 prefix-list** command, followed by the name of the list. The switch enters IPv6 prefix-list configuration mode for the list. If the command is followed by the name of an existing ACL, subsequent commands edit that list.

Example

- This command places the switch in IPv6 prefix list configuration mode to create an IPv6 prefix list named ***map1***.

```
switch(config)#ipv6 prefix-list map1
switch(config-ipv6-pfx)#
```

Adding a Rule

To append a rule to the end of a list, enter the rule without a sequence number while in Prefix-List configuration mode for the list. The new rule's sequence number is derived by adding 10 to the last rule's sequence number.

Example

- These commands enter the first two rules into a new prefix list.

```
switch(config-ipv6-pfx)#permit 3:4e96:8ca1:33cf::/64
switch(config-ipv6-pfx)#permit 3:11b1:8fe4:1aac::/64
```

To view the list, save the rules by exiting the prefix-list command mode, then re-enter the configuration mode and type **show active**.

```
switch(config-ipv6-pfx)#exit
switch(config)#ipv6 prefix-list map1
switch(config-ipv6-pfx)#show active
ipv6 prefix-list map1
   seq 10 permit 3:4e96:8ca1:33cf::/64
   seq 20 permit 3:11b1:8fe4:1aac::/64
switch(config-ipv6-pfx)#
```

This command appends a rule to the end of the prefix list. The new rule's sequence number is 30.

```
switch(config-ipv6-pfx)#permit 3:1bca:1141:ab34::/64
switch(config-ipv6-pfx)#exit
switch(config)#ipv6 prefix-list map1
switch(config-ipv6-pfx)#show active
ipv6 prefix-list map1
   seq 10 permit 3:4e96:8ca1:33cf::/64
   seq 20 permit 3:11b1:8fe4:1aac::/64
   seq 30 permit 3:1bca:1141:ab34::/64
switch(config-ipv6-pfx)#
```

Inserting a Rule

To insert a rule into a prefix list, use the **seq (IPv6 Prefix Lists)** command to enter a rule with a sequence number that is between numbers of two existing rules.

Example

- This command inserts a rule between the first two rules by assigning it the sequence number 15.

```
switch(config-ipv6-pfx)#seq 15 deny 3:4400::/64
switch(config-ipv6-pfx)#exit
switch(config)#show ipv6 prefix-list map1
ipv6 prefix-list map1
seq 10 permit 3:4e96:8ca1:33cf::/64
seq 15 deny 3:4400::/64
seq 20 permit 3:11b1:8fe4:1aac::/64
seq 30 permit 3:1bca:3ff2:634a::/64
switch(config)#
```

Deleting a Rule

To remove a rule from the configuration mode prefix list, enter **no seq** (see **seq (IPv6 Prefix Lists)**), followed by the sequence number of the rule to be removed.

Example

- These commands remove rule 20 from the prefix list, then displays the resultant prefix list.

```
switch(config-ipv6-pfx)#no seq 20
switch(config-ipv6-pfx)#exit
switch(config)#show ipv6 prefix-list map1
ipv6 prefix-list map1
seq 10 permit 3:4e96:8ca1:33cf::/64
seq 15 deny 3:4400::/64
seq 30 permit 3:1bca:3ff2:634a::/64
switch(config)#
```

20.5.2 Using Prefix Lists

Route map match statements include an option that matches on a specified prefix list.

Example

- The MAP_1 route map uses a match statement that references the PL_1 prefix list.

```
switch(config)#route-map MAP_1 permit
switch(config-route-map-MAP_1)#match ip address prefix-list PL_1
switch(config-route-map-MAP_1)#set community 500
switch(config-route-map-MAP_1)#exit
```

20.6 ACL, Route Map, and Prefix List Commands

This section describes CLI commands that this chapter references.

ACL Creation and Access Commands

- ip access-list
- ip access-list standard
- ipv6 access-list
- ipv6 access-list standard
- mac access-list

ACL Implementation Commands

- ip access-group
- ipv6 access-group
- mac access-group

Service ACL Implementation Commands

- ip access-group (Service ACLs)
- ipv6 access-group (Service ACLs)

ACL Edit Commands

- hardware access-list update default-result permit
- no <sequence number> (ACLs)
- resequence (ACLs)
- show (ACL configuration modes)
- statistics per-entry (ACL configuration modes)

ACL Rule Commands

- deny (IPv4 ACL)
- deny (IPv6 ACL)
- deny (MAC ACL)
- deny (Standard IPv4 ACL)
- deny (Standard IPv6 ACL)
- permit (IPv4 ACL)
- permit (IPv6 ACL)
- permit (MAC ACL)
- permit (Standard IPv4 ACL)
- permit (Standard IPv6 ACL)
- remark

ACL List Counter Commands

- clear ip access-lists counters
- clear ipv6 access-lists counters
- hardware counter feature acl out

ACL Display Commands

- show ip access-lists
- show ipv6 access-lists
- show mac access-lists

Prefix List Creation and Access Commands

- ip prefix-list

- ipv6 prefix-list

Prefix List Edit Commands

- deny (IPv6 Prefix List)
- permit (IPv6 Prefix List)
- seq (IPv6 Prefix Lists)

Prefix List Display Commands

- show ip prefix-list
- show ipv6 prefix-list

Route Map Creation and Access Command

- route-map

Route Map Edit Commands

- continue (route-map)
- description (route-map)
- match (route-map)
- set (route-map)
- set community (route-map)
- set extcommunity (route-map)

Route Map Display Commands

- show route-map

clear ip access-lists counters

The **clear ip access-lists counters** command sets ACL counters to zero for the specified IPv4 access control list (ACL). The ***session*** parameter limits ACL counter clearing to the current CLI session.

Command Mode

Privileged EXEC

Command Syntax

clear ip access-lists counters [*ACL_NAME*] [*SCOPE*]

Parameters

- ***ACL_NAME*** Name of ACL. Options include:
  - <no parameter> all ACLs.
  - *access_list* name of ACL.
- ***SCOPE*** Session affected by command. Options include:
  - <no parameter> command affects counters on all CLI sessions.
  - **session** affects only current CLI session.

Example

- This command resets all IPv4 ACL counters.

```
switch(config)#clear ip access-lists counters
switch(config)#
```

clear ipv6 access-lists counters

The **clear ipv6 access-lists counters** command sets ACL counters to zero for the specified IPv6 access control list (ACL). The ***session*** parameter limits ACL counter clearing to the current CLI session.

Command Mode

Privileged EXEC

Command Syntax

clear ipv6 access-lists counters [*ACL_NAME*] [*SCOPE*]

Parameters

- ***ACL_NAME*** name of ACL. Options include:
  - <no parameter> all IPv6 ACLs.
  - *access_list* name of IPv6 ACL.
- ***SCOPE*** Session affected by command. Options include:
  - <no parameter> command affects counters on all CLI sessions.
  - **session** affects only current CLI session.

Example

- This command resets all IPv6 ACL counters.

```
switch(config)#clear ipv6 access-lists counters
switch(config)#
```

continue (route-map)

The **continue** command creates a route map clause entry that enables additional route map evaluation of routes whose parameters meet the clause's matching criteria.

A clause typically contains a **match (route-map)** and a **set (route-map)** statement. The evaluation of routes whose settings are the same as **match** statement parameters normally end and the clause's **set** statement are applied to the route. Routes that match a clause containing a **continue** statement are evaluated against the clause specified by the continue statement.

When a route matches multiple route-map clauses, the filter action (deny or permit) is determined by the last clause that the route matches. The **set** statements in all clauses matching the route are applied to the route after the route map evaluation is complete. Multiple set statements are applied in the same order by which the route was evaluated against the clauses containing them.

The **no continue** and **default continue** commands remove the corresponding **continue** statement from the configuration mode route map clause by deleting the corresponding command from ***running-config***.

Command Mode

Route-Map Configuration

Command Syntax

```
continue NEXT_SEQ
no continue NEXT_SEQ
default continue NEXT_SEQ
```

Parameters

- ***NEXT_SEQ*** specifies next clause for evaluating matching routes. Options include:
  - <no parameter> Next clause in the route map, as determined by sequence number.
  - *seq_number* Specifies the number of the next clause. Values range from 1 to 16777215.

Restrictions

A continue statement cannot specify a sequence number smaller than the sequence number of its route-map clause.

Related Commands

- **route-map** enters route-map configuration mode.

Example

- This command creates route map map1, clause 40 with a match statement, a set statement, and a continue statement. Routes that match the clause are subsequently evaluated against clause 100. The **set local-preference** statement is applied to matching routes regardless of subsequent matching operations.

```
switch(config)#route-map map1 deny 40
switch(config-route-map-map1)#match as 15
switch(config-route-map-map1)#continue 100
switch(config-route-map-map1)#set local-preference 50
switch(config-route-map-map1)#
```

deny (IPv4 ACL)

The **deny** command adds a deny rule to the configuration mode IPv4 access control list (ACL). Packets filtered by a **deny** rule are dropped by interfaces to which the ACL is applied. Sequence numbers determine rule placement in the ACL. Sequence numbers for commands without numbers are derived by adding 10 to the number of the ACL's last rule.

The **no deny** and **default deny** commands remove the specified rule from the configuration mode ACL. The **no <sequence number> (ACLs)** command also removes the specified rule from the ACL.

Command Mode

ACL Configuration

Command Syntax

```
[SEQ_NUM] deny PROTOCOL SOURCE_ADDR [SOURCE_PORT] DEST_ADDR [DEST_PORT]
[FLAGS][MESSAGE][fragments][tracked][DSCP_FILTER][TTL_FILTER][log]

no deny PROTOCOL SOURCE_ADDR [SOURCE_PORT] DEST_ADDR [DEST_PORT]
[FLAGS][MESSAGE][fragments][tracked][DSCP_FILTER][TTL_FILTER][log]

default deny PROTOCOL SOURCE_ADDR [SOURCE_PORT] DEST_ADDR [DEST_PORT]
[FLAGS][MESSAGE][fragments][tracked][DSCP_FILTER][TTL_FILTER][log]
```

Commands use a subset of the listed fields. Available parameters depend on specified protocol. Use CLI syntax assistance to view options for specific protocols when creating a deny rule.

Parameters

- ***SEQ_NUM*** Sequence number assigned to the rule. Options include:
  - <no parameter> Number is derived by adding 10 to the number of the ACL's last rule.
  - <***1 – 4294967295***> Number assigned to entry.
- ***PROTOCOL*** protocol field filter. Values include:
  - **ahp** Authentication Header Protocol (51).
  - **icmp** Internet Control Message Protocol (1).
  - **igmp** Internet Group Management Protocol (2).
  - **ip** Internet Protocol v4 (4).
  - **ospf** Open Shortest Path First (89).
  - **pim** Protocol Independent Multicast (103).
  - **tcp** Transmission Control Protocol (6).
  - **udp** user datagram protocol (17).
  - **vrrp** Virtual Router Reduncancy Protocol (112).
  - *protocol_num* integer corresponding to an IP protocol. Values range from 0 to 255.
- ***SOURCE_ADDR*** and ***DEST_ADDR*** source and destination address filters. Options include:
  - *network_addr* subnet address (CIDR or address-mask).
  - **any** Packets from all addresses are filtered.
  - **host** *ip_addr* IP address (dotted decimal notation).

  Subnet addresses support discontiguous masks.
- ***SOURCE_PORT*** and ***DEST_PORT*** source and destination port filters. Options include:

- **any** all ports
- **eq** *port-1 port-2* ... *port-n* A list of ports. Maximum list size is 10 ports.
- **neq** *port-1 port-2* ... *port-n* The set of all ports not listed. Maximum list size is 10 ports.
- **gt** *port* The set of ports with larger numbers than the listed port.
- **lt** *port* The set of ports with smaller numbers than the listed port.
- **range** *port_1 port_2* The set of ports whose numbers are between the range.

- **fragments** filters packets with FO bit set (indicates a non-initial fragment packet).
- ***FLAGS*** flag bit filters (TCP packets). Use CLI syntax assistance (?) to display options.
- ***MESSAGE*** message type filters (ICMP packets). Use CLI syntax assistance (?) to display options.
- **tracked** rule filters packets in existing ICMP, UDP, or TCP connections.
  - Valid in ACLs applied to the control plane.
  - Validity in ACLs applied to data plane varies by switch platform.
- ***DSCP_FILTER*** rule filters packet by its DSCP value. Values include:
  - <no parameter> Rule does not use DSCP to filter packets.
  - **dscp** *dscp_value* Packets match if DSCP field in packet is equal to *dscp_value*.
- ***TTL_FILTER*** rule filters packet by its TTL (time-to-live) value. Values include:
  - **ttl eq** *ttl_value* Packets match if ***ttl*** in packet is equal to *ttl_value*.
  - **ttl gt** *ttl_value* Packets match if ***ttl*** in packet is greater than *ttl_value*.
  - **ttl lt** *ttl_value* Packets match if ***ttl*** in packet is less than *ttl_value*.
  - **ttl neq** *ttl_value* Packets match if ***ttl*** in packet is not equal to *ttl_value*.
    - Valid in ACLs applied to the control plane.
    - Validity in ACLs applied to data plane varies by switch platform.
- **log** triggers an informational log message to the console about the matching packet.
  - Valid in ACLs applied to the control plane.
  - Validity in ACLs applied to data plane varies by switch platform.

Examples

- This command appends a **deny** statement at the end of the ACL. The **deny** statement drops OSPF packets from 10.10.1.1/24 to any host.

```
switch(config)#ip access-list text1
switch(config-acl-text1)#deny ospf 10.1.1.0/24 any
switch(config-acl-text1)#
```

- This command inserts a **deny** statement with the sequence number 65. The **deny** statement drops all PIM packets.

```
switch(config-acl-text1)#65 deny pim any any
switch(config-acl-text1)#
```

deny (IPv6 ACL)

The **deny** command adds a deny rule to the configuration mode IPv6 access control list (ACL). Packets filtered by a **deny** rule are dropped by interfaces to which the ACL is applied. Sequence numbers determine rule placement in the ACL. Sequence numbers for commands without numbers are derived by adding 10 to the number of the ACL's last rule.

The **no deny** and **default deny** commands remove the specified rule from the configuration mode ACL. The **no <sequence number> (ACLs)** command also removes the specified rule from the ACL.

Command Mode

IPv6-ACL Configuration

Command Syntax

[*SEQ_NUM*] deny *PROT SRC_ADDR* [*SRC_PT*] *DEST_ADDR* [*DEST_PT*][*FLAG*][*MSG*][*HOP*] [tracked][*DSCP_FILTER*][log]

no deny *PROT SRC_ADDR* [*SRC_PT*] *DEST_ADDR* [*DEST_PT*][*FLAG*][*MSG*][*HOP*][tracked] [*DSCP_FILTER*][log]

default deny *PROT SRC_ADDR* [*SRC_PT*] *DEST_ADDR* [*DEST_PT*][*FLAG*][*MSG*][*HOP*][tracked] [*DSCP_FILTER*][log]

Commands use a subset of the listed fields. Available parameters depend on specified protocol. Use CLI syntax assistance to view options for specific protocols when creating a deny rule.

Parameters

- ***SEQ_NUM*** Sequence number assigned to the rule. Options include:
  - <no parameter> Number is derived by adding 10 to the number of the ACL's last rule.
  - <***1 – 4294967295***> Number assigned to entry.
- ***PROT*** protocol field filter. Values include:
  - **icmpv6** Internet Control Message Protocol for version 6 (58).
  - **ipv6** Internet Protocol – IPv6 (41).
  - **ospf** Open Shortest Path First (89).
  - **tcp** Transmission Control Protocol (6).
  - **udp** User Datagram Protocol (17).
  - *protocol_num* integer corresponding to an IP protocol. Values range from 0 to 255.
- ***SRC_ADDR*** and ***DEST_ADDR*** source and destination address filters. Options include:
  - *ipv6_prefix* IPv6 address with prefix length (CIDR notation).
  - **any** Packets from all addresses are filtered.
  - **host** *ipv6_addr* IPv6 host address.
- ***SRC_PT*** and ***DEST_PT*** source and destination port filters. Options include:
  - **any** all ports.
  - **eq** *port-1 port-2 ... port-n* A list of ports. Maximum list size is 10 ports.
  - **neq** *port-1 port-2 ... port-n* The set of all ports not listed. Maximum list size is 10 ports.
  - **gt** *port* The set of ports with larger numbers than the listed port.
  - **lt** *port* The set of ports with smaller numbers than the listed port.
  - **range** *port_1 port_2* The set of ports whose numbers are between the range.

- ***HOP*** filters by packet's hop-limit value. Options include:
  - <no parameter> Rule does not use hop limit to filter packets.
  - **hop-limit eq** *hop_value* Packets match if ***hop-limit*** value in packet equals *hop_value*.
  - **hop-limit gt** *hop_value* Packets match if ***hop-limit*** in packet is greater than *hop_value*.
  - **hop-limit lt** *hop_value* Packets match if ***hop-limit*** in packet is less than *hop_value*.
  - **hop-limit neq** *hop_value* Packets match if ***hop-limit*** in packet is not equal to *hop_value*.
- ***FLAG*** flag bit filters (TCP packets). Use CLI syntax assistance (?) to display options.
- ***MSG*** message type filters (ICMPv6 packets). Use CLI syntax assistance (?) to display options.
- **tracked** rule filters packets in existing ICMP, UDP, or TCP connections.
  - Valid in ACLs applied to the control plane.
  - Validity in ACLs applied to data plane varies by switch platform.
- ***DSCP_FILTER*** rule filters packet by its DSCP value. Values include:
  - <no parameter> Rule does not use DSCP to filter packets.
  - **dscp** *dscp_value* Packets match if DSCP field in packet is equal to *dscp_value*.
- **log** triggers an informational log message to the console about the matching packet.
  - Valid in ACLs applied to the control plane.
  - Validity in ACLs applied to data plane varies by switch platform.

Example

- This command appends a **deny** statement at the end of the ACL. The **deny** statement drops IPv6 packets from 3710:249a:c643:ef11::/64 to any host.

```
switch(config)#ipv6 access-list text1
switch(config-acl-text1)#deny ipv6 3710:249a:c643:ef11::/64 any
switch(config-acl-text1)#
```

deny (IPv6 Prefix List)

The **deny** command adds a rule to the configuration mode IPv6 prefix list. Route map match statements use prefix lists to filter routes for redistribution into OSPF, RIP, or BGP domains. Routes are denied access when they match the prefix that a **deny** statement specifies.

The **no deny** and **default deny** commands remove the specified rule from the configuration mode prefix list. The **no seq (IPv6 Prefix Lists)** command also removes the specified rule from the prefix list.

Command Mode

IPv6-pfx Configuration

Command Syntax

[*SEQUENCE*] deny *ipv6_prefix* **[*MASK*]**

Parameters

- ***SEQUENCE*** Sequence number assigned to the rule. Options include:
  - <no parameter> Number is derived by adding 10 to the number of the list's last rule.
  - **seq** *seq_num* Number is specified by *seq_num*. Value ranges from 0 to 65535.
- *ipv6_prefix* IPv6 prefix upon which command filters routes (CIDR notation).
- ***MASK*** range of the prefix to be matched.
  - <no parameter> exact match with the subnet mask is required.
  - **eq** *mask_e* prefix length is equal to *mask_e*.
  - **ge** *mask_g* range is from *mask_g* to 128.
  - **le** *mask_l* range is from *subnet* mask length to *mask_l*.
  - **ge** *mask_l* **le** *mask_g* range is from *mask_g* to *mask_l*.

    mask_e, mask_l and *mask_g* range from 1 to 128.

    when **le** and **ge** are specified, *subnet* mask > *mask_g*>*mask_l*

Example

- This command appends a **deny** statement at the end of the text1 prefix list. The **deny** statement denies redistribution of routes with the specified prefix.

```
switch(config)#ipv6 prefix-list route-five
switch(config-ipv6-pfx)#deny 3100::/64
switch(config-ipv6-pfx)#
```

deny (MAC ACL)

The **deny** command adds a deny rule to the configuration mode MAC access control list (ACL). Packets filtered by a deny rule are dropped by interfaces to which the ACL is applied. Sequence numbers determine rule placement in the ACL. Sequence numbers for commands without numbers are derived by adding 10 to the number of the ACL's last rule.

The **no deny** and **default deny** commands remove the specified rule from the configuration mode ACL. The **no <sequence number> (ACLs)** command also removes the specified rule from the ACL.

Command Mode

MAC-ACL Configuration

Command Syntax

```
[SEQ_NUM] deny SOURCE_ADDR DEST_ADDR [PROTOCOL] [log]
no deny SOURCE_ADDR DEST_ADDR [PROTOCOL] [log]
default deny SOURCE_ADDR DEST_ADDR [PROTOCOL] [log]
```

Parameters

- ***SEQ_NUM*** Sequence number assigned to the rule. Options include:
  - <no parameter> Number is derived by adding 10 to the number of the ACL's last rule.
  - <***1 – 4294967295***> Number assigned to entry.
- ***SOURCE_ADDR*** and ***DEST_ADDR*** source and destination address filters. Options include:
  - *mac_address mac_mask* MAC address and mask
  - **any** Packets from all addresses are filtered.

    mac_address specifies a MAC address in 3x4 dotted hexadecimal notation (hhhh.hhhh.hhhh)
    mac_mask specifies a MAC address mask in 3x4 dotted hexadecimal notation (hhhh.hhhh.hhhh)
  - *0* bits require an exact match to filter
  - *1* bits filter on any value
- ***PROTOCOL*** protocol field filter. Values include:
  - **aarp** Appletalk Address Resolution Protocol (0x80f3)
  - **appletalk** Appletalk (0x809b)
  - **arp** Address Resolution Protocol (0x806)
  - **ip** Internet Protocol Version 4 (0x800)
  - **ipx** Internet Packet Exchange (0x8137)
  - **lldp** LLDP (0x88cc)
  - **novell** Novell (0x8138)
  - **rarp** Reverse Address Resolution Protocol (0x8035)
  - *protocol_num* integer corresponding to a MAC protocol. Values range from 0 to 65535
- **log** triggers an informational log message to the console about the matching packet.

Examples

- This command appends a permit statement at the end of the ACL. The deny statement drops all aarp packets from 10.1000.0000 through 10.1000.FFFF to any host.

```
switch(config)#mac access-list text1
switch(config-mac-acl-text1)#deny 10.1000.0000 0.0.FFFF any aarp
```

- This command inserts a permit statement with the sequence number 25. The deny statement drops all packets through the interface.

```
switch(config-mac-acl-text1)#25 deny any any
```

deny (Standard IPv4 ACL)

The **deny** command adds a deny rule to the configuration mode standard IPv4 access control list (ACL). Standard ACL rules filter on the source field.

Packets filtered by a **deny** rule are dropped by interfaces to which the ACL is applied. Sequence numbers determine rule placement in the ACL. Sequence numbers for commands without numbers are derived by adding 10 to the number of the ACL's last rule.

The **no deny** and **default deny** commands remove the specified rule from the configuration mode ACL. The **no <sequence number> (ACLs)** command also removes the specified rule from the ACL.

Command Mode

Std-ACL Configuration

Command Syntax

```
[SEQ_NUM] deny SOURCE_ADDR [log]
no deny SOURCE_ADDR [log]
default deny SOURCE_ADDR [log]
```

Parameters

- ***SEQ_NUM*** Sequence number assigned to the rule. Options include:
  - <no parameter> Number is derived by adding 10 to the number of the ACL's last rule.
  - <***1 – 4294967295***> Number assigned to entry.
- ***SOURCE_ADDR*** source address filter. Options include:
  - *network_addr* subnet address (CIDR or address-mask).
  - **any** packets from all addresses are filtered.
  - **host** *ip_addr* IP address (dotted decimal notation).

    Subnet addresses support discontiguous masks.
- **log** triggers an informational log message to the console about the matching packet.
  - Valid in ACLs applied to the control plane.
  - Validity in ACLs applied to data plane varies by switch platform.

Example

- This command appends a **deny** statement at the end of the ACL. The **deny** statement drops packets from 10.10.1.1/24.

```
switch(config)#ip access-list standard text1
switch(config-std-acl-text1)#deny 10.1.1.1/24
switch(config-std-acl-text1)#
```

deny (Standard IPv6 ACL)

The **deny** command adds a deny rule to the configuration mode standard IPv6 access control list (ACL). Standard ACL rules filter on the source field.

Packets filtered by a **deny** rule are dropped by interfaces to which the ACL is applied. Sequence numbers determine rule placement in the ACL. Sequence numbers for commands without numbers are derived by adding 10 to the number of the ACL's last rule.

The **no deny** and **default deny** commands remove the specified rule from the configuration mode ACL. The **no <sequence number> (ACLs)** command also removes the specified rule from the ACL.

Command Mode

Std-IPv6-ACL Configuration

Command Syntax

```
[SEQ_NUM] deny SOURCE_ADDR
no deny SOURCE_ADDR
default deny SOURCE_ADDR
```

Parameters

- ***SEQ_NUM*** Sequence number assigned to the rule. Options include:
  - <no parameter> Number is derived by adding 10 to the number of the ACL's last rule.
  - <***1 – 4294967295***> Number assigned to entry.
- ***SOURCE_ADDR*** source address filter. Options include:
  - *ipv6_prefix* IPv6 address with prefix length (CIDR notation).
  - **any** Packets from all addresses are filtered.
  - **host** *ipv6_addr* IPv6 host address.

Example

- This command appends a **deny** statement at the end of the ACL. The **deny** statement drops packets from 2103::/64.

```
switch(config)#ipv6 access-list standard text1
switch(config-std-acl-ipv6-text1)#deny 2103::/64
switch(config-std-acl-ipv6-text1)#
```

description (route-map)

The **description** command adds a text string to the configuration mode route map. The string has no functional impact on the route map.

The **no description** and **default description** commands remove the text string from the configuration mode route map by deleting the corresponding **description** command from ***running-config***.

Command Mode

Route-Map Configuration

Command Syntax

```
description label_text
no description
default description
```

Parameters

- *label_text* character string assigned to the route map configuration.

Related Commands

- **route-map** enters route-map configuration mode.

Examples

- These commands add description text to the XYZ-1 route map.

```
switch(config)#route-map XYZ-1
switch(config-route-map-XYZ-1)#description This is the first map.
switch(config-route-map-XYZ-1)#exit
switch(config)#show route-map XYZ-1
route-map XYZ-1 permit 10
  Description:
    description This is the first map.
  Match clauses:
  Set clauses:
switch(config)#
```

hardware access-list update default-result permit

The **hardware access-list update default-result permit** command configures the switch to permit all traffic on Ethernet and VLAN interfaces with ACLs applied to them while those ACLs are being modified. Traffic is permitted beginning when the ACL is opened for modification using one of the **ip access-list** commands, and ending when the ACL configuration mode is exited and rules are populated in hardware. This feature is disabled by default.

The **no hardware access-list update default-result permit** and **default hardware access-list update default-result permit** commands restore the switch to its default state (blocking traffic during ACL modifications) by removing the corresponding **hardware access-list update default-result permit** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
hardware access-list update default-result permit
no hardware access-list update default-result permit
default hardware access-list update default-result permit
```

Restrictions

This feature available only on 7150 series switches.

This feature does not work well with egress ACLs.

While this feature is enabled, modification of an ACL may cause packet loss even on interfaces where an unrelated ACL is applied.

While this feature is enabled, static NAT and ACL-based mirroring will also be affected during ACL updates.

Example

- This command configures a 7150 series switch to permit all traffic on Ethernet and VLAN interfaces with ACLs applied to them while those ACLs are being modified.

```
switch(config)#hardware access-list update default-result permit
   switch(config)#
```

hardware counter feature acl out

The **hardware counter feature acl out** command enables egress ACL hardware counters for IPv4 or IPv6, which count the number of packets hitting rules associated with egress ACLs applied to various interfaces on a switch.

The **no hardware counter feature acl out** and **default hardware counter feature acl out** commands disable or return the egress ACL hardware counters to the default state.

Command Mode

Global Configuration

Command Syntax

```
hardware counter feature acl out [OPTIONS]
no hardware counter feature acl out [OPTIONS]
default hardware counter feature acl out [OPTIONS]
```

Parameters

- *OPTIONS* ACL hardware counter options include:
  - ipv4 address family IPv4.
  - ipv6 address family IPv6.

Example

- This command enables IPv4 egress ACL hardware counters.

```
switch(config)#hardware counter feature acl out ipv4
switch(config)#
```

- This command disables IPv4 egress ACL hardware counters.

```
switch(config)#no hardware counter feature acl out ipv4
switch(config)#
```

ip access-group

Note On hardware purchased after July 3, 2017, security ACLs are only available on AlgoMatch-enabled platforms.

The **ip access-group** command applies an IPv4 or standard IPv4 access control list (ACL) to the configuration mode interface.

The **no ip access-group** and **default ip access-group** commands remove the corresponding **ip access-group** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration
Interface-Port-Channel Configuration
Interface-VLAN Configuration

Command Syntax

```
ip access-group list_name DIRECTION
no ip access-group list_name DIRECTION
default ip access-group list_name DIRECTION
```

Parameters

- *list_name* name of ACL assigned to interface.
- ***DIRECTION*** transmission direction of packets, relative to interface. Valid options include:
  - **in** inbound packets.
  - **out** outbound packets.

Restrictions

Filtering of outbound packets by ACLs is not supported on Petra platform switches.

Filtering of outbound packets by ACLs on FM6000 switches is supported only on physical interfaces (Ethernet and port channels).

Example

- These commands apply the IPv4 ACL named ***test2*** to Ethernet interface 3.

```
switch(config)#interface ethernet 3
switch(config-if-Et3)#ip access-group test2 in
switch(config-if-Et3)#
```

ip access-group (Service ACLs)

The **ip access-group (Service ACLs)** command configures a Service ACL to be applied by a control-plane service. The service is specified by the command mode in which the Service ACL is applied.

The **no ip access-group (Service ACLs)** and **default ip access-group (Service ACLs)** commands remove the corresponding **ip access-group (Service ACLs)** command from ***running-config***.

Note Service ACLs are only available on EOS versions 4.19.0F and above.

Command Mode

Mgmt-SSH Configuration
Mgmt-API Configuration
Router-BGP Configuration
Router-OSPF Configuration
Router-IGMP Configuration
MPLS-LDP Configuration
Queue-Monitor-Streaming Configuration
MPLS-Ping Configuration
Mgmt-Telnet Configuration

Command Syntax

```
ip access-group acl_name [vrf vrf_name] [in]
no ip access-group [acl_name] [vrf vrf_name] [in]
default ip access-group acl_name [vrf vrf_name] [in]
```

Parameters

Parameters vary by process.

- *acl_name* name of the Service ACL assigned to control-plane service.
- **vrf** *vrf_name* specifies the VRF in which the Service ACL is to be applied.
- **in** specifies inbound connections or packets only (keyword required for SSH and Telnet services).

Example

- These commands apply the Service ACL ***bgpacl*** to the BGP routing protocol in VRF ***purple***.

```
(config)# router bgp 5
(config-router-bgp)# vrf purple
(config-router-bgp-vrf-purple)# ip access-group bgpacl
```

For additional configuration examples, see Section 20.3.2: Configuring Service ACLs and Displaying Status and Counters.

ip access-list

The **ip access-list** command places the switch in ACL configuration mode, which is a group change mode that modifies an IPv4 access control list. The command specifies the name of the IPv4 ACL that subsequent commands modify and creates an ACL if it references a nonexistent list. All changes in a group change mode edit session are pending until the end of the session.

The **exit** command saves pending ACL changes to ***running-config***, then returns the switch to global configuration mode. ACL changes are also saved by entering a different configuration mode.

The **abort** command discards pending ACL changes, returning the switch to global configuration mode.

The **no ip access-list** and **default ip access-list** commands delete the specified IPv4 ACL.

Command Mode

Global Configuration

Command Syntax

```
ip access-list list_name
no ip access-list list_name
default ip access-list list_name
```

Parameters

- *list_name* Name of ACL.
  Must begin with an alphabetic character. Cannot contain spaces or quotation marks.

Commands Available in ACL configuration mode:

- **deny (IPv4 ACL)**
- **no <sequence number> (ACLs)**
- **permit (IPv4 ACL)**
- **remark**
- **resequence (ACLs)**
- **show (ACL configuration modes)**

Related Commands

- **ip access-list standard** enters std-acl configuration mode for editing standard IP ACLs.
- **show ip access-lists** displays IP and standard ACLs.

Examples

- This command places the switch in ACL configuration mode to modify the ***filter1*** IPv4 ACL.

```
switch(config)#ip access-list filter1
switch(config-acl-filter1)#
```

- This command saves changes to ***filter1*** ACL, then returns the switch to global configuration mode.

```
switch(config-acl-filter1)#exit
switch(config)#
```

- This command discards changes to ***filter1***, then returns the switch to global configuration mode.

```
switch(config-acl-filter1)#abort
switch(config)#
```

ip access-list standard

The **ip access-list standard** command places the switch in std-ACL configuration mode, which is a group change mode that modifies a standard IPv4 access control list. The command specifies the name of the standard IPv4 ACL that subsequent commands modify, and creates an ACL if it references a nonexistent list. All group change mode edit session changes are pending until the session ends.

The **exit** command saves pending ACL changes to ***running-config***, then returns the switch to global configuration mode. Pending changes are also saved by entering a different configuration mode.

The **abort** command discards pending ACL changes, returning the switch to global configuration mode.

The **no ip access-list standard** and **default ip access-list standard** commands delete the specified ACL.

Command Mode

Global Configuration

Command Syntax

```
ip access-list standard list_name
no ip access-list standard list_name
default ip access-list standard list_name
```

Parameters

- *list_name* Name of standard ACL.
  Must begin with an alphabetic character. Cannot contain spaces or quotation marks.

Commands Available in std-ACL configuration mode:

- **deny (Standard IPv4 ACL)**
- **no <sequence number> (ACLs)**
- **permit (Standard IPv4 ACL)**
- **remark**
- **resequence (ACLs)**
- **show (ACL configuration modes)**

Related Commands

- **ip access-list** enters ACL configuration mode for editing IPv4 ACLs.
- **show ip access-lists** displays IPv4 and standard IPv4 ACLs.

Examples

- This command places the switch in std-ACL configuration mode to modify the ***filter2*** IPv4 ACL.

```
switch(config)#ip access-list standard filter2
switch(config-std-acl-filter2)#
```

- This command saves changes to ***filter2*** ACL, then returns the switch to global configuration mode.

```
switch(config-std-acl-filter2)#exit
switch(config)#
```

- This command discards changes to ***filter2***, then returns the switch to global configuration mode.

```
switch(config-std-acl-filter2)#abort
switch(config)#
```

ip prefix-list

The **ip prefix-list** command creates a prefix list or adds an entry to an existing list. Route map match statements use prefix lists to filter routes for redistribution into OSPF, RIP, or BGP domains.

A prefix list comprises all prefix list entries with the same label. The sequence numbers of the rules in a prefix list specify the order that the rules are applied to a route that the match statement is evaluating.

The **no ip prefix-list** and **default ip prefix-list** commands delete the specified prefix list entry by removing the corresponding **ip prefix-list** statement from ***running-config***. If the **no** or **default ip prefix-list** command does not list a sequence number, the command deletes all entries of the prefix list.

Command Mode

Global Configuration

Command Syntax

```
ip prefix-list list_name [SEQUENCE] FILTER_TYPE network_addr [MASK]
no ip prefix-list list_name [SEQUENCE]
default ip prefix-list list_name [SEQUENCE]
```

Parameters

- *list_name* The label that identifies the prefix list.
- ***SEQUENCE*** Sequence number of the prefix list entry. Options include
  - <no parameter> entry's number is ten plus highest sequence number in current list.
  - **seq** *seq_num* number assigned to entry. Value ranges from 0 to 65535.
- ***FILTER_TYPE*** specifies route access when it matches IP prefix list. Options include:
  - **permit** routes are permitted access when they match the specified subnet.
  - **deny** routes are denied access when they match the specified subnet.
- *network_addr* Subnet upon which command filters routes. Format is CIDR or address-mask.
- ***MASK*** range of the prefix to be matched.
  - <no parameter> exact match with the subnet mask is required.
  - **eq** *mask_e* prefix length is equal to *mask_e*.
  - **ge** *mask_g* range is from *mask_g* to 32.
  - **le** *mask_l* range is from *subnet* mask length to *mask_l*.
  - **ge** *mask_l* **le** *mask_g* range is from *mask_g* to *mask_l*.

    mask_e, mask_l and *mask_g* range from 1 to 32.

    when **le** and **ge** are specified, *subnet* mask < *mask_g*<*mask_l*

Example

- These commands create a two-entry prefix list named route-one.

```
switch(config)#ip prefix-list route-one seq 10 deny 10.1.1.1/24 ge 26 le 30
switch(config)#ip prefix-list route-one seq 20 deny 10.1.2.1/16
switch(config)#
```

ipv6 access-group

Note On hardware purchased after July 3, 2017, security ACLs are only available on AlgoMatch-enabled platforms.

The **ipv6 access-group** command applies an IPv6 or standard IPv6 access control list (ACL) to the configuration mode interface.

The **no ipv6 access-group** and **default ipv6 access-group** commands remove the corresponding **ipv6 access-group** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration
Interface-Port-Channel Configuration
Interface-VLAN Configuration

Command Syntax

```
ipv6 access-group list_name DIRECTION
no ipv6 access-group list_name DIRECTION
default ipv6 access-group list_name DIRECTION
```

Parameters

- *list_name* name of ACL assigned to interface.
- ***DIRECTION*** transmission direction of packets, relative to interface. Valid options include:
  - **in** inbound packets.
  - **out** outbound packets.

Examples

- These commands assign the IPv6 ACL named ***test2*** to the Ethernet 3 interface.

```
switch(config)#interface ethernet 3
switch(config-if-Et3)#ipv6 access-group test2 in
switch(config-if-Et3)#
```

ipv6 access-group (Service ACLs)

The **ipv6 access-group (Service ACLs)** command configures an IPv6 or standard IPv6 Service ACL to be applied by a control-plane service. The service is specified by the command mode in which the Service ACL is applied.

The **no ipv6 access-group (Service ACLs)** and **default ipv6 access-group (Service ACLs)** commands remove the corresponding **ipv6 access-group (Service ACLs)** command from ***running-config***.

Note Service ACLs are only available on EOS versions 4.19.0F and above.

Command Mode

Mgmt-SSH Configuration
Mgmt-API Configuration
Router-BGP Configuration
Router-OSPF Configuration
MPLS-LDP Configuration
Queue-Monitor-Streaming Configuration
MPLS-Ping Configuration
Mgmt-Telnet Configuration

Command Syntax

```
ipv6 access-group ipv6_acl_name [vrf vrf_name] [in]
no ipv6 access-group [ipv6_acl_name] [vrf vrf_name] [in]
default ip access-group ipv6_acl_name [vrf vrf_name] [in]
```

Parameters

Parameters vary by process.

- *ipv6_acl_name* name of the IPv6 Service ACL assigned to control-plane service.
- **vrf** *vrf_name* specifies the VRF in which the Service ACL is to be applied.
- **in** specifies inbound connections or packets only (keyword required for SSH and Telnet services).

Example

- These commands apply the IPv6 Service ACL ***bgpacl*** to the BGP routing protocol in VRF ***purple***.

```
(config)# router bgp 5
(config-router-bgp)# vrf purple
(config-router-bgp-vrf-purple)# ipv6 access-group bgpacl
```

For additional configuration examples, see Section 20.3.2: Configuring Service ACLs and Displaying Status and Counters.

ipv6 access-list

The **ipv6 access-list** command places the switch in IPv6-ACL configuration mode, which is a group change mode that modifies an IPv6 access control list. The command specifies the name of the IPv6 ACL that subsequent commands modify and creates an ACL if it references a nonexistent list. All changes in a group change mode edit session are pending until the end of the session.

The **exit** command saves pending ACL changes to ***running-config***, then returns the switch to global configuration mode. ACL changes are also saved by entering a different configuration mode.

The **abort** command discards pending ACL changes, returning the switch to global configuration mode.

The **no ipv6 access-list** and **default ipv6 access-list** commands delete the specified IPv6 ACL.

Command Mode

Global Configuration

Command Syntax

```
ipv6 access-list list_name
no ipv6 access-list list_name
default ipv6 access-list list_name
```

Parameters

- *list_name* Name of ACL.
  Must begin with an alphabetic character. Cannot contain spaces or quotation marks.

Commands Available in IPv6-ACL configuration mode:

- **deny (IPv6 ACL)**
- **no <sequence number> (ACLs)**
- **permit (IPv6 ACL)**
- **remark**
- **resequence (ACLs)**
- **show (ACL configuration modes)**

Related Commands

- **ipv6 access-list standard** enters std-ipv6-acl configuration mode for editing standard IPv6 ACLs.
- **show ipv6 access-lists** displays IPv6 and standard IPv6 ACLs.

Examples

- This command places the switch in IPv6-ACL configuration mode to modify the ***filter1*** IPv6 ACL.

```
switch(config)#ipv6 access-list filter1
switch(config-ipv6-acl-filter1)#
```

- This command saves changes to ***filter1*** ACL, then returns the switch to global configuration mode.

```
switch(config-ipv6-acl-filter1)#exit
switch(config)#
```

- This command discards changes to ***filter1***, then returns the switch to global configuration mode.

```
switch(config-ipv6-acl-filter1)#abort
switch(config)#
```

ipv6 access-list standard

The **ipv6 access-list standard** command places the switch in std-IPv6-ACL-configuration mode, which is a group change mode that modifies a standard IPv6 access control list. The command specifies the name of the standard IPv6 ACL that subsequent commands modify and creates an ACL if it references a nonexistent list. All group change mode edit session changes are pending until the session ends.

The **exit** command saves pending ACL changes to ***running-config***, then returns the switch to global configuration mode. Pending changes are also saved by entering a different configuration mode.

The **abort** command discards pending ACL changes, returning the switch to global configuration mode.

The **no ipv6 access-list standard** and **default ipv6 access-list standard** commands delete the specified ACL.

Command Mode

Global Configuration

Command Syntax

```
ipv6 access-list standard list_name
no ipv6 access-list standard list_name
default ipv6 access-list standard list_name
```

Parameters

- *list_name* Name of ACL.
  Must begin with an alphabetic character. Cannot contain spaces or quotation marks.

Commands Available in std-IPv6-ACL configuration mode:

- **deny (Standard IPv6 ACL)**
- **no <sequence number> (ACLs)**
- **permit (Standard IPv6 ACL)**
- **remark**
- **resequence (ACLs)**
- **show (ACL configuration modes)**

Related Commands

- **ipv6 access-list** enters IPv6-ACL configuration mode for editing IPv6 ACLs.
- **show ipv6 access-lists** displays IPv6 and standard IPv6 ACLs.

Examples

- This command places the switch in Std-IPv6 ACL configuration mode to modify the ***filter2*** ACL.

```
switch(config)#ipv6 access-list standard filter2
switch(config-std-ipv6-acl-filter2)#
```

- This command saves changes to ***filter2*** ACL, then returns the switch to global configuration mode.

```
switch(config-std-ipv6-acl-filter2)#exit
switch(config)#
```

- This command discards changes to ***filter2***, then returns the switch to global configuration mode.

```
switch(config-std-ipv6-acl-filter2)#abort
switch(config)#
```

ipv6 prefix-list

The **ip prefix-list** command places the switch in IPv6 prefix-list configuration mode, which is a group change mode that modifies an IPv6 prefix list. The command specifies the name of the IPv6 prefix list that subsequent commands modify and creates a prefix list if it references a nonexistent list. All changes in a group change mode edit session are pending until the end of the session.

The **exit** command saves pending prefix list changes to ***running-config***, then returns the switch to global configuration mode. ACL changes are also saved by entering a different configuration mode.

The **abort** command discards pending changes, returning the switch to global configuration mode.

The **no ipv6 prefix-list** and **default ipv6 prefix-list** commands delete the specified IPv6 prefix list.

Command Mode

Global Configuration

Command Syntax

```
ipv6 prefix-list list_name
no ipv6 prefix-list list_name
default ipv6 prefix-list list_name
```

Parameters

- *list_name* Name of prefix list.
  Must begin with an alphabetic character. Cannot contain spaces or quotation marks.

Commands Available in IPv6-pfx configuration mode:

- **deny (IPv6 Prefix List)**
- **permit (IPv6 Prefix List)**
- **seq (IPv6 Prefix Lists)**

Examples

- This command places the switch in IPv6 prefix-list configuration mode to modify the ***route-five*** prefix list.

```
switch(config)#ipv6 prefix-list route-five
switch(config-ipv6-pfx)#
```

- This command saves changes to the prefix list, then returns the switch to global configuration mode.

```
switch(config-ipv6-pfx)#exit
switch(config)#
```

- This command saves changes to the prefix list, then places the switch in interface-Ethernet mode.

```
switch(config-ipv6-pfx)#interface ethernet 3
switch(config-if-Et3)#
```

- This command discards changes to the prefix list, then returns the switch to global configuration mode.

```
switch(config-ipv6-pfx)#abort
switch(config)#
```

mac access-group

The **mac access-group** command applies a MAC-ACL (access control list) to the configuration mode interface.

The **no mac access-group** and **default mac access-group** commands remove the specified **mac access-group** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration
Interface-Port-Channel Configuration

Command Syntax

```
mac access-group list_name DIRECTION
no mac access-group list_name DIRECTION
default mac access-group list_name DIRECTION
```

Parameters

- *list_name* name of MAC ACL.
- ***DIRECTION*** transmission direction of packets, relative to interface. Valid options include:
  - **in** inbound packets.
  - **out** outbound packets.

Restrictions

Filtering of outbound packets by MAC ACLs is supported only on Helix, Trident, and Trident-II platform switches.

Example

- These commands assign the MAC ACL named ***mtest2*** to the Ethernet 3 interface to filter inbound packets.

```
switch(config)#interface ethernet 3
switch(config-if-Et3)#mac access-group mtest2 in
switch(config-if-Et3)#
```

mac access-list

The **mac access-list** command places the switch in MAC-ACL configuration mode, which is a group change mode that modifies a MAC access control list. The command specifies the name of the MAC ACL that subsequent commands modify and creates an ACL if it references a nonexistent list. All changes in a group change mode edit session are pending until the end of the session.

The **exit** command saves pending ACL changes to ***running-config***, then returns the switch to global configuration mode. ACL changes are also saved by entering a different configuration mode.

The **abort** command discards pending ACL changes, returning the switch to global configuration mode.

The **no mac access-list** and **default mac access-list** commands delete the specified list.

Command Mode

Global Configuration

Command Syntax

```
mac access-list list_name
no mac access-list list_name
default mac access-list list_name
```

Parameters

- *list_name* Name of MAC ACL.
  Names must begin with an alphabetic character and cannot contain a space or quotation mark.

Commands Available in MAC-ACL configuration mode:

- **deny (MAC ACL)**
- **no <sequence number> (ACLs)**
- **permit (MAC ACL)**
- **remark**
- **resequence (ACLs)**
- **show (ACL configuration modes)**

Examples

- This command places the switch in MAC-ACL configuration mode to modify the ***mfilter1*** MAC ACL.

```
switch(config)#mac access-list mfilter1
switch(config-mac-acl-mfilter1)#
```

- This command saves changes to ***mfilter1*** ACL, then returns the switch to global configuration mode.

```
switch(config-mac-acl-mfilter1)#exit
switch(config)#
```

- This command saves changes to ***mfilter1*** ACL, then places the switch in interface-Ethernet mode.

```
switch(config-mac-acl-mfilter1)#interface ethernet 3
switch(config-if-Et3)#
```

- This command discards changes to ***mfilter1***, then returns the switch to global configuration mode.

```
switch(config-mac-acl-mfilter1)#abort
switch(config)#
```

match (route-map)

The **match** command creates a route map clause entry that specifies one route filtering condition. When a clause contains multiple match commands, the permit or deny filter applies to a route only if its properties are equal to corresponding parameters in each **match** statement. When a route's properties do not equal the statement parameters, the route is evaluated against the next clause in the route map, as determined by sequence number. If all clauses fail to permit or deny the route, the route is denied.

The **no match** and **default match** commands remove the **match** statement from the configuration mode route map clause by deleting the corresponding command from ***running-config***.

Command Mode

Route-Map Configuration

Command Syntax

```
match CONDITION
no match CONDITION
default match CONDITION
```

Parameters

- ***CONDITION*** specifies criteria for evaluating a route. Options include:
  - **as** <***1*** to ***4294967295***> BGP autonomous system number.
  - **as-path** *path_name* BGP autonomous system path access list.
  - **community** ***NAME*** BGP community. Options for ***NAME*** include:
    - *listname* BGP community.
    - *listname* **exact-match** BGP community; list must match set that is present.
  - **extcommunity** *listname* BGP extended community. Options for ***NAME*** include:
    - *listname* BGP community.
    - *listname* **exact-match** BGP community; list must match set that is present.
  - **interface** ***INTF_NAME*** Specifies an interface. Options for ***INTF_NAME*** include:
    - **ethernet** *e_num* Ethernet interface.
    - **loopback** *l_num* Loopback interface.
    - **port-channel** *p_num* Port channel interface.
    - **vlan** *v_num* VLAN interface.
  - **ip address** ***LIST*** IPv4 address filtered by an ACL or prefix list. ***LIST*** options include:
    - **access-list** *acl_name* IPv4 address filtered by Access Control List (ACL).
    - **prefix-list** *plv4_name* IPv4 address filtered by IP prefix list.
  - **ip next-hop prefix-list** *plv4_name* IPv4 next-hop filtered by IP prefix list.
  - **ip resolved-next-hop prefix-list** *plv4_name* IPv4 resolved nexthop filtered by IP prefix list.
  - **ipv6 address prefix-list** *plv6_name* IPv6 address filtered by IPv6 prefix list.
  - **ipv6 next-hop prefix-list** *plv6_name* IPv6 next-hop filtered by IPv6 prefix list.
  - **ipv6 resolved-next-hop prefix-list** *plv6_name* IPv6 resolved nexthop filtered by IPv6 prefix list.
  - **local-preference** <***1*** to ***4294967295***> BGP local preference metric.
  - **metric** <***1*** to ***4294967295***> route metric.

- **metric-type** ***OSPF_TYPE*** OSPF metric type. Options include:
  - **type-1** OSPF type 1 metric.
  - **type-2** OSPF type 2 metric.
- **source-protocol** *protocol_type* Routing protocol of route's source. Options include:
  - **bgp**
  - **connected**
  - **ospf**
  - **rip**
  - **static**
- **tag** <***1*** to ***4294967295***> route tag.

Related Commands

- **route-map** enters route-map configuration mode.

Example

- This command creates a route-map match rule that filters routes from BGP AS 15.

```
switch(config)#route-map map1
switch(config-route-map-map1)#match as 15
switch(config-route-map-map1)#
```

no <sequence number> (ACLs)

The **no <sequence number>** command removes the rule with the specified sequence number from the ACL. The **default <sequence number>** command also removes the specified rule.

Command Mode

ACL Configuration
IPv6-ACL Configuration
Std-ACL Configuration
Std-IPv6-ACL Configuration
MAC-ACL Configuration

Command Syntax

no *line_num*
default *line_num*

Parameters

- *line_num* sequence number of rule to be deleted. Values range from ***1 to 4294967295***.

Example

- This command removes statement 30 from the list

```
switch(config-acl-test1)#show
IP Access List test1
        10 permit ip 10.10.10.0/24 any
        20 permit ip any host 10.20.10.1
        30 deny ip host 10.10.10.1 host 10.20.10.1
        40 permit ip any any
        50 remark end of list
switch(config-acl-test1)#no 30
switch(config-acl-test1)#show
IP Access List test1
        10 permit ip 10.10.10.0/24 any
        20 permit ip any host 10.20.10.1
        40 permit ip any any
        50 remark end of list
```

permit (IPv4 ACL)

The **permit** command adds a permit rule to the configuration mode IPv4 access control list (ACL). Packets filtered by a permit rule are accepted by interfaces to which the ACL is applied. Sequence numbers determine rule placement in the ACL. Sequence numbers for commands without numbers are derived by adding 10 to the number of the ACL's last rule.

The **no permit** and **default permit** commands remove the specified rule from the configuration mode ACL. The **no <sequence number> (ACLs)** command also removes a specified rule from the ACL.

Command Mode

ACL Configuration

Command Syntax

```
[SEQ_NUM] permit PROTOCOL SOURCE_ADDR [SOURCE_PORT] DEST_ADDR [DEST_PORT]
[FLAGS][MESSAGE][fragments][tracked][DSCP_FILTER][TTL_FILTER][log]

no permit PROTOCOL SOURCE_ADDR [SOURCE_PORT] DEST_ADDR [DEST_PORT]
[FLAGS][MESSAGE][fragments][tracked][DSCP_FILTER][TTL_FILTER][log]

default permit PROTOCOL SOURCE_ADDR [SOURCE_PORT] DEST_ADDR [DEST_PORT]
[FLAGS][MESSAGE][fragments][tracked][DSCP_FILTER][TTL_FILTER][log]
```

Commands use a subset of the listed fields. Available parameters depend on specified protocol. Use CLI syntax assistance to view options for specific protocols when creating a permit rule.

Parameters

- ***SEQ_NUM*** Sequence number assigned to the rule. Options include:
  - <no parameter> Number is derived by adding 10 to the number of the ACL's last rule.
  - <***1 – 4294967295***> Number assigned to entry.
- ***PROTOCOL*** protocol field filter. Values include:
  - **ahp** Authentication Header Protocol (51).
  - **icmp** Internet Control Message Protocol (1).
  - **igmp** Internet Group Management Protocol (2).
  - **ip** Internet Protocol v4 (4).
  - **ospf** Open Shortest Path First (89).
  - **pim** Protocol Independent Multicast (103).
  - **tcp** Transmission Control Protocol (6).
  - **udp** user datagram protocol (17).
  - **vrrp** Virtual Router Reduncancy Protocol (112).
  - *protocol_num* integer corresponding to an IP protocol. Values range from 0 to 255.
- ***SOURCE_ADDR*** and ***DEST_ADDR*** source and destination address filters. Options include:
  - *network_addr* subnet address (CIDR or address-mask).
  - **any** Packets from all addresses are filtered.
  - **host** *ip_addr* IP address (dotted decimal notation).

  Source and destination subnet addresses support discontiguous masks.
- ***SOURCE_PORT*** and ***DEST_PORT*** source and destination port filters. Options include:

- **any** all ports
- **eq** *port-1 port-2* ... *port-n* A list of ports. Maximum list size is 10 ports.
- **neq** *port-1 port-2* ... *port-n* The set of all ports not listed. Maximum list size is 10 ports.
- **gt** *port* The set of ports with larger numbers than the listed port.
- **lt** *port* The set of ports with smaller numbers than the listed port.
- **range** *port_1 port_2* The set of ports whose numbers are between the range.

- **fragments** filters packets with FO bit set (indicates a non-initial fragment packet).
- ***FLAGS*** flag bit filters (TCP packets). Use CLI syntax assistance (?) to display options.
- ***MESSAGE*** message type filters (ICMP packets). Use CLI syntax assistance (?) to display options.
- **tracked** rule filters packets in existing ICMP, UDP, or TCP connections.
  - Valid in ACLs applied to the control plane.
  - Validity in ACLs applied to data plane varies by switch platform.
- ***DSCP_FILTER*** rule filters packet by its DSCP value. Values include:
  - <no parameter> Rule does not use DSCP to filter packets.
  - **dscp** *dscp_value* Packets match if DSCP field in packet is equal to *dscp_value*.
- ***TTL_FILTER*** rule filters packet by its TTL (time-to-live) value. Values include:
  - **ttl eq** *ttl_value* Packets match if ***ttl*** in packet is equal to *ttl_value*.
  - **ttl gt** *ttl_value* Packets match if ***ttl*** in packet is greater than *ttl_value*.
  - **ttl lt** *ttl_value* Packets match if ***ttl*** in packet is less than *ttl_value*.
  - **ttl neq** *ttl_value* Packets match if ***ttl*** in packet is not equal to *ttl_value*.
    - Valid in ACLs applied to the control plane.
    - Validity in ACLs applied to data plane varies by switch platform.
- **log** triggers an informational log message to the console about the matching packet.
  - Valid in ACLs applied to the control plane.
  - Validity in ACLs applied to data plane varies by switch platform.

Examples

- This command appends a **permit** statement at the end of the ACL. The **permit** statement passes all OSPF packets from 10.10.1.1/24 to any host.

```
switch(config)#ip access-list text1
switch(config-acl-text1)#permit ospf 10.1.1.0/24 any
switch(config-acl-text1)#
```

- This command inserts a **permit** statement with the sequence number 25. The **permit** statement passes all PIM packets through the interface.

```
switch(config-acl-text1)#25 permit pim any any
switch(config-acl-text1)#
```

permit (IPv6 ACL)

The **permit** command adds a permit rule to the configuration mode IPv6 access control list (ACL). Packets filtered by a permit rule are accepted by interfaces to which the ACL is applied. Sequence numbers determine rule placement in the ACL. Sequence numbers for commands without numbers are derived by adding 10 to the number of the ACL's last rule.

The **no permit** and **default permit** commands remove the specified rule from the configuration mode ACL. The **no <sequence number> (ACLs)** command also removes a specified rule from the ACL.

Command Mode

IPv6-ACL Configuration

Command Syntax

```
[SEQ_NUM] permit PROT SRC_ADDR [SRC_PT] DEST_ADDR [DEST_PT][FLAG][MSG][HOP]
[tracked][DSCP_FILTER][log]

no permit PROT SRC_ADDR [SRC_PT] DEST_ADDR [DEST_PT][FLAG][MSG][HOP][tracked]
[DSCP_FILTER][log]

default permit PROT SRC_ADDR [SRC_PT] DEST_ADDR [DEST_PT][FLAG][MSG][HOP]
[tracked][DSCP_FILTER][log]
```

Commands use a subset of the listed fields. Available parameters depend on specified protocol. Use CLI syntax assistance to view options for specific protocols when creating a permit rule.

Parameters

- ***SEQ_NUM*** Sequence number assigned to the rule. Options include:
  - <no parameter> Number is derived by adding 10 to the number of the ACL's last rule.
  - <***1 – 4294967295***> Number assigned to entry.
- ***PROT*** Protocol field filter. Values include:
  - **icmpv6** Internet Control Message Protocol for v6 (58).
  - **ipv6** Internet Protocol – IPv6 (41).
  - **ospf** Open Shortest Path First (89).
  - **tcp** Transmission Control Protocol (6).
  - **udp** User Datagram Protocol (17).
  - *protocol_num* integer corresponding to an IP protocol. Values range from 0 to 255.
- ***SRC_ADDR*** and ***DEST_ADDR*** Source and destination address filters. Options include:
  - *ipv6_prefix* IPv6 address with prefix length (CIDR notation).
  - **any** Packets from all addresses are filtered.
  - **host** *ipv6_addr* IPv6 host address.
- ***SRC_PT*** and ***DEST_PT*** Source and destination port filters. Options include:
  - **any** all ports.
  - **eq** *port-1 port-2 ... port-n* A list of ports. Maximum list size is 10 ports.
  - **neq** *port-1 port-2 ... port-n* The set of all ports not listed. Maximum list size is 10 ports.
  - **gt** *port* The set of ports with larger numbers than the listed port.
  - **lt** *port* The set of ports with smaller numbers than the listed port.
  - **range** *port_1 port_2* The set of ports whose numbers are between the range.

- ***HOP*** filters by packet's hop-limit value. Options include:
  - <no parameter> Rule does not use hop limit to filter packets.
  - **hop-limit eq** *hop_value* Packets match if ***hop-limit*** value in packet equals *hop_value*.
  - **hop-limit gt** *hop_value* Packets match if ***hop-limit*** in packet is greater than *hop_value*.
  - **hop-limit lt** *hop_value* Packets match if ***hop-limit*** in packet is less than *hop_value*.
  - **hop-limit neq** *hop_value* Packets match if ***hop-limit*** in packet is not equal to *hop_value*.
- ***FLAG*** flag bit filters (TCP packets). Use CLI syntax assistance (?) to display options.
- ***MSG*** message type filters (ICMPv6 packets). Use CLI syntax assistance (?) to display options.
- **tracked** rule filters packets in existing ICMP, UDP, or TCP connections.
  - Valid in ACLs applied to the control plane.
  - Validity in ACLs applied to data plane varies by switch platform.
- ***DSCP_FILTER*** rule filters packet by its DSCP value. Values include:
  - <no parameter> Rule does not use DSCP to filter packets.
  - **dscp** *dscp_value* Packets match if DSCP field in packet is equal to *dscp_value*.
- **log** triggers an informational log message to the console about the matching packet.
  - Valid in ACLs applied to the control plane.
  - Validity in ACLs applied to data plane varies by switch platform.

Example

- This command appends a **permit** statement at the end of the ACL. The **permit** statement passes all IPv6 packets with the source address 3710:249a:c643:ef11::/64 and with any destination address.

```
switch(config)#ipv6 access-list text1
switch(config-acl-text1)#permit ipv6 3710:249a:c643:ef11::/64 any
switch(config-acl-text1)#
```

permit (IPv6 Prefix List)

The **permit** command adds a rule to the configuration mode IPv6 prefix list. Route map match statements use prefix lists to filter routes for redistribution into OSPF, RIP, or BGP domains. Routes are redistributed into the specified domain when they match the prefix that a **permit** statement specifies.

The **no permit** and **default permit** commands remove the specified rule from the configuration mode prefix list. The **no seq (IPv6 Prefix Lists)** command also removes the specified rule from the prefix list.

Command Mode

IPv6-pfx Configuration

Command Syntax

[*SEQUENCE*] deny *ipv6_prefix* **[*MASK*]**

Parameters

- ***SEQUENCE*** Sequence number assigned to the rule. Options include:
  - <no parameter> Number is derived by adding 10 to the number of the list's last rule.
  - **seq** *seq_num* Number is specified by *seq_num*. Value ranges from 0 to 65535.
- *ipv6_prefix* IPv6 prefix upon which command filters routes (CIDR notation).
- ***MASK*** Range of the prefix to be matched.
  - <no parameter> exact match with the subnet mask is required.
  - **eq** *mask_e* prefix length is equal to *mask_e*.
  - **ge** *mask_g* range is from *mask_g* to 128.
  - **le** *mask_l* range is from *subnet* mask length to *mask_l*.
  - **ge** *mask_l* **le** *mask_g* range is from *mask_g* to *mask_l*.

    mask_e, mask_l and *mask_g* range from 1 to 128.

    When **le** and **ge** are specified, the prefix list size > *mask_g*>*mask_l*

Example

- This command appends a **permit** statement at the end of the text1 prefix list. The **permit** statement allows redistribution of routes with the specified prefix.

```
switch(config)#ipv6 prefix-list route-five
switch(config-ipv6-pfx)#permit 3100::/64
switch(config-ipv6-pfx)#
```

permit (MAC ACL)

The **permit** command adds a permit rule to the configuration mode MAC access control list packets through the interface to which the list is applied. Rule filters include protocol, source, and destination.

The **no permit** and **default permit** commands remove the specified rule from the configuration mode ACL. The **no <sequence number> (ACLs)** command also removes the specified rule from the ACL.

Command Mode

MAC-ACL Configuration

Command Syntax

```
[SEQ_NUM] permit SOURCE_ADDR DEST_ADDR [PROTOCOL] [log]
no permit SOURCE_ADDR DEST_ADDR [PROTOCOL] [log]
default permit SOURCE_ADDR DEST_ADDR [PROTOCOL] [log]
```

Parameters

- ***SEQ_NUM*** Sequence number assigned to the rule. Options include:
  - <no parameter> Number is derived by adding 10 to the number of the ACL's last rule.
  - <***1 – 4294967295***> Number assigned to entry.
- ***SOURCE_ADDR*** and ***DEST_ADDR*** source and destination address filters. Options include:
  - *mac_address mac_mask* MAC address and mask
  - **any** Packets from all addresses are filtered.

    mac_address specifies a MAC address in 3x4 dotted hexadecimal notation (hhhh.hhhh.hhhh)
    mac_mask specifies a MAC address mask in 3x4 dotted hexadecimal notation (hhhh.hhhh.hhhh)
  - *0* bits require an exact match to filter
  - *1* bits filter on any value
- ***PROTOCOL*** protocol field filter. Values include:
  - **aarp** Appletalk Address Resolution Protocol (0x80f3)
  - **appletalk** Appletalk (0x809b)
  - **arp** Address Resolution Protocol (0x806)
  - **ip** Internet Protocol Version 4 (0x800)
  - **ipx** Internet Packet Exchange (0x8137)
  - **lldp** LLDP (0x88cc)
  - **novell** Novell (0x8138)
  - **rarp** Reverse Address Resolution Protocol (0x8035)
  - *protocol_num* integer corresponding to a MAC protocol. Values range from 0 to 65535
- **log** triggers an informational log message to the console about the matching packet.

Examples

- This command appends a **permit** statement at the end of the ACL. The **permit** statement passes all aarp packets from 10.1000.0000 through 10.1000.FFFF to any host.

```
switch(config)#mac access-list text1
switch(config-mac-acl-text1)#permit 10.1000.0000 0.0.FFFF any aarp
switch(config-mac-acl-text1)#
```

- This command inserts a **permit** statement with the sequence number 25. The **permit** statement passes all packets through the interface.

```
switch(config-mac-acl-text1)#25 permit any any
switch(config-mac-acl-text1)#
```

permit (Standard IPv4 ACL)

The **permit** command adds a permit rule to the configuration mode standard IPv4 access control list (ACL). Standard ACL rules filter on the source field.

Packets filtered by a permit rule are accepted by interfaces to which the ACL is applied. Sequence numbers determine rule placement in the ACL. Sequence numbers for commands without numbers are derived by adding 10 to the number of the ACL's last rule.

The **no permit** and **default permit** commands remove the specified rule from the configuration mode ACL. The **no <sequence number> (ACLs)** command also removes the specified rule from the ACL.

Command Mode

Std-ACL Configuration

Command Syntax

```
[SEQ_NUM] permit SOURCE_ADDR [log]
no permit SOURCE_ADDR [log]
default permit SOURCE_ADDR [log]
```

Parameters

- ***SEQ_NUM*** Sequence number assigned to the rule. Options include:
  - <no parameter> Number is derived by adding 10 to the number of the ACL's last rule.
  - <***1 – 4294967295***> Number assigned to entry.
- ***SOURCE_ADDR*** source address filter. Options include:
  - *network_addr* subnet address (CIDR or address-mask).
  - **any** Packets from all addresses are filtered.
  - **host** *ip_addr* IP address (dotted decimal notation).

    Subnet addresses support discontiguous masks.
- **log** triggers an informational log message to the console about the matching packet.
  - Valid in ACLs applied to the control plane.
  - Validity in ACLs applied to data plane varies by switch platform.

Example

- This command appends a **permit** statement at the end of the ACL. The **permit** statement passes all packets with a source address of 10.10.1.1/24.

```
switch(config)#ip access-list standard text1
switch(config-std-acl-text1)#permit 10.1.1.1/24
switch(config-std-acl-text1)#
```

permit (Standard IPv6 ACL)

The **permit** command adds a permit rule to the configuration mode standard IPv6 access control list. Standard ACL rules filter on the source field.

Packets filtered by a permit rule are accepted by interfaces to which the ACL is applied. Sequence numbers determine rule placement in the ACL. Sequence numbers for commands without numbers are derived by adding 10 to the number of the ACL's last rule.

The **no permit** and **default permit** commands remove the specified rule from the configuration mode ACL. The **no <sequence number> (ACLs)** command also removes the specified rule from the ACL.

Command Mode

Std-IPv6-ACL Configuration

Command Syntax

```
[SEQ_NUM] permit SOURCE_ADDR
no permit SOURCE_ADDR
default permit SOURCE_ADDR
```

Parameters

- ***SEQ_NUM*** Sequence number assigned to the rule. Options include:
  - <no parameter> Number is derived by adding 10 to the number of the ACL's last rule.
  - <***1 – 4294967295***> Number assigned to entry.
- ***SOURCE_ADDR*** source address filter. Options include:
  - *ipv6_prefix* IPv6 address with prefix length (CIDR notation).
  - **any** Packets from all addresses are filtered.
  - **host** *ipv6_addr* IPv6 host address.

Example

- This command appends a **permit** statement at the end of the ACL. The **permit** statement drops packets with a source address of 2103::/64.

```
switch(config)#ipv6 access-list standard text1
switch(config-std-acl-ipv6-text1)#permit 2103::/64
switch(config-std-acl-ipv6-text1)#
```

remark

The **remark** command adds a non-executable comment statement into the pending ACL. Remarks entered without a sequence number are appended to the end of the list. Remarks entered with a sequence number are inserted into the list as specified by the sequence number.

The **default remark** command removes the comment statement from the ACL.

The **no remark** command removes the comment statement from the ACL. The command can specify the remark by content or by sequence number.

Command Mode

ACL Configuration
IPv6-ACL Configuration
Std-ACL Configuration
Std-IPv6-ACL Configuration
MAC-ACL Configuration

Command Syntax

```
remark text
line_num remark [text]
no remark text
default remark text
```

Parameters

- *text* the comment text.
- *line_num* sequence number assigned to the remark statement. Value ranges from ***1*** to ***4294967295***

Example

- This command appends a comment to the list.

```
switch(config-acl-test1)#remark end of list
switch(config-acl-test1)#show
IP Access List test1
        10 permit ip 10.10.10.0/24 any
        20 permit ip any host 10.20.10.1
        30 deny ip host 10.10.10.1 host 10.20.10.1
        40 permit ip any any
        50 remark end of list
```

resequence (ACLs)

The **resequence** command assigns sequence numbers to rules in the configuration mode ACL. Command parameters specify the number of the first rule and the numeric interval between consecutive rules.

Maximum rule sequence number is 4294967295.

Command Mode

ACL Configuration
IPv6-ACL Configuration
Std-ACL Configuration
Std-IPv6-ACL Configuration
MAC-ACL Configuration

Command Syntax

resequence [*start_num* [*inc_num*]]

Parameters

- *start_num* sequence number assigned to the first rule. Default is 10.
- *inc_num* numeric interval between consecutive rules. Default is 10.

Example

- The **resequence** command renumbers the list, starting the first command at number 100 and incrementing subsequent lines by 20.

```
switch(config-acl-test1)#show
IP Access List test1
        10 permit ip 10.10.10.0/24 any
        20 permit ip any host 10.20.10.1
        30 deny ip host 10.10.10.1 host 10.20.10.1
        40 permit ip any any
        50 remark end of list
switch(config-acl-test1)#resequence 100 20
switch(config-acl-test1)#show
IP Access List test1
        100 permit ip 10.10.10.0/24 any
        120 permit ip any host 10.20.10.1
        140 deny ip host 10.10.10.1 host 10.20.10.1
        160 permit ip any any
        180 remark end of list
```

route-map

The **route-map** command places the switch in route-map configuration mode, which is a group change mode that modifies a route map clause. The command specifies the name and number of the route map clause that subsequent commands modify and creates a route map clause if it references a nonexistent clause. All changes in a group change mode edit session are pending until the end of the session.

Route maps define conditions for redistributing routes between routing protocols. A route map clause is identified by a name, filter type (permit or deny) and sequence number. Clauses with the same name are components of a single route map; the sequence number determines the order in which the clauses are compared to a route.

The **exit** command saves pending route map clause changes to ***running-config***, then returns the switch to global configuration mode. ACL changes are also saved by entering a different configuration mode.

The **abort** command discards pending changes, returning the switch to global configuration mode.

The **no route-map** and **default route-map** commands delete the specified route map clause from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
route-map map_name [FILTER_TYPE] [sequence_number]
no route-map map_name [FILTER_TYPE] [sequence_number]
default route-map map_name [FILTER_TYPE] [sequence_number]
```

Parameters

- *map_name* label assigned to route map. Protocols reference this label to access the route map.
- ***FILTER_TYPE*** disposition of routes matching conditions specified by route map clause.
  - **permit** routes are redistributed when they match route map clause.
  - **deny** routes are not redistributed when they match route map clause.
  - <No parameter> assigns **permit** as the ***FILTER_TYPE***.

  When a route does not match the route map criteria, the next clause within the route map is evaluated to determine the redistribution action for the route.
- *sequence_number* the route map position relative to other clauses with the same name.
  - <no parameter> sequence number of 10 (default) is assigned to the route map.
  - <1-16777215> specifies sequence number assigned to route map.

Commands Available in route-map configuration mode:

- **continue (route-map)**
- **match (route-map)**
- **set (route-map)**

Examples

- This command creates the route map named ***map1*** and places the switch in route-map configuration mode. The route map is configured as a permit map.

```
switch(config)#route-map map1 permit 20
switch(config-route-map-map1)#
```

- This command saves changes to ***map1*** route map, then returns the switch to global configuration mode.

```
switch(config-route-map-map1)#exit
switch(config)#
```

- This command saves changes to ***map1*** route map, then places the switch in interface-Ethernet mode.

```
switch(config-route-map-map1)#interface ethernet 3
switch(config-if-Et3)#
```

- This command discards changes to ***map1*** route map, then returns the switch to global configuration mode.

```
switch(config-route-map-map1)#abort
switch(config)#
```

seq (IPv6 Prefix Lists)

The **no seq** command removes the rule with the specified sequence number from the ACL. The **default seq** command also removes the specified rule.

The **seq** keyword is a command option used at the beginning of **deny (IPv6 Prefix List)** and **permit (IPv6 Prefix List)** commands that places a new rule between two existing rules.

Command Mode

IPv6-pfx Configuration

Command Syntax

no seq *line_num*
default seq *line_num*

Parameters

- *line_num* sequence number of rule to be deleted. Valid rule numbers range from 0 to 65535.

Example

- These commands remove rule 20 from the ***map1*** prefix list, then displays the resultant list.

```
switch(config)#ipv6 prefix-list map1
switch(config-ipv6-pfx)#no seq 20
switch(config-ipv6-pfx)#exit
switch(config)#show ipv6 prefix-list map1
ipv6 prefix-list map1
seq 10 permit 3:4e96:8ca1:33cf::/64
seq 15 deny 3:4400::/64
seq 30 permit 3:1bca:3ff2:634a::/64
seq 40 permit 3:1bca:1141:ab34::/64
switch(config)#
```

set (route-map)

The **set** command specifies modifications to routes that are selected for redistribution by the configuration mode route map.

The **no set** and **default set** commands remove the specified **set** statement from the configuration mode route map clause by deleting the corresponding **set** statement from ***running-config***.

Command Mode

Route-Map Configuration

Command Syntax

```
set CONDITION
no set CONDITION
default set CONDITION
```

Parameters

- ***CONDITION*** specifies the route modification parameter and value. Options include:
  - **as-path prepend** BGP AS number that is prepended to as-path.
    - <***1*** to ***4294967295***> BGP AS number to prepend.
    - **auto** use peer AS number for inbound and local AS for outbound to prepend.
  - **distance** <***1*** to 2***55***> Protocol independent administrative distance.
  - **ip next-hop** *ipv4_address* next hop IPv4 address.
  - **ip next-hop peer-address** Use BGP peering address as next hop IPv4 address.
  - **ipv6 next-hop** *ipv6_address* next hop IPv6 address.
  - **ipv6 next-hop peer-address** Use BGP peering address as next hop IPv6 address.
  - **local-preference** <***1*** to ***4294967295***> BGP local preference metric.
  - **metric** <***1*** to ***4294967295***> route metric.
  - **metric +**<***1*** to ***4294967295***> add specified value to current route metric.
  - **metric –** <***1*** to ***4294967295***> subtract specified value to current route metric.
  - **metric-type *OSPF_TYPE*** OSPF metric type. Options include:
    - **type-1** OSPF type 1 metric.
    - **type-2** OSPF type 2 metric.
  - **origin *O_TYPE*** BGP origin attribute. Options for ***O_TYPE*** include
    - **egp** exterior BGP route.
    - **igp** interior BGP route.
    - **incomplete** BGP route of unknown origin.
  - **tag** <***1*** to ***4294967295***> route tag.
  - **weight** <***1*** to ***65535***> BGP weight parameter.

Related Commands

- **route-map** enters route-map configuration mode.
- **set (route-map)** specifies community modifications for the redistributed routes.
- **set community (route-map)** specifies ext community modifications for the redistributed routes.

Example

- This command creates a route-map entry that sets the local preference metric to 100 on redistributed routes.

```
switch(config)#route-map map1
switch(config-route-map-map1)#set local-preference 100
switch(config-route-map-map1)#
```

set community (route-map)

The **set community** command specifies community attribute modifications to routes that are selected for redistribution by the configuration mode route map. The **set community none** command removes community attributes from the route.

The **no set community** and **default set community** commands remove the specified **set community** statement from the configuration mode route map clause by deleting the corresponding statement from ***running-config***.

Command Mode

Route-Map Configuration

Command Syntax

```
set community COND_1 [COND_2] [COND_N][MOD_TYPE]
set community community-list clist_name [MOD_TYPE]
set community none
no set community COND_1 [COND_2] [COND_N][MOD_TYPE]
no set community community-list clist_name [MOD_TYPE]
no set community none
default set community COND_1 [COND_2] [COND_N][MOD_TYPE]
default set community community-list clist_name [MOD_TYPE]
default set community none
```

Parameters

- ***COND_X*** Specifies community modification. Command may contain multiple attributes. Options include:
  - <***0*** to ***65535***>:<***0*** to ***65535***> community number, expressed as AS:NN.
  - <***1*** to ***4294967040***> community number. ***Running-config*** stores in AS:NN format.
  - **internet** Advertise to Internet community (community 0:0).
  - **local-as** Do not send outside local AS.
  - **no-advertise** Do not advertise to any peer.
  - **no-export** Do not export to next AS.
- *clist_name* Name of community list.
- ***MOD_TYPE*** Specifies community modification method. Options include:
  - <no parameter> command replaces existing community with specified parameters.
  - **additive** command adds specified parameters to existing community.
  - **delete** command removes specified parameters to existing community.

Related Commands

- **route-map** enters route-map configuration mode.
- **set (route-map)** specifies attribute modifications for the redistributed routes.
- **set community (route-map)** specifies extended community attributes modifications for redistributed routes.

Example

- This command creates a route-map entry that sets the community attribute to 0:0.

```
switch(config)#route-map map1
switch(config-route-map-map1)#set community internet
switch(config-route-map-map1)#
```

set extcommunity (route-map)

The **set extcommunity** command specifies extended community attribute modifications to routes that are selected for redistribution by the configuration mode route map. The **set extcommunity none** command removes extended community attributes from the route.

The **no set extcommunity** and **default set extcommunity** commands remove the specified **set extcommunity** statement from the configuration mode route map clause by deleting the corresponding statement from ***running-config***.

Command Mode

Route-Map Configuration

Command Syntax

```
set extcommunity COND_1  [COND_2]  [COND_N][MOD_TYPE]
set extcommunity none
no set extcommunity COND_1  [COND_2]  [COND_N][MOD_TYPE]
no set extcommunity none
default set extcommunity COND_1  [COND_2]  [COND_N][MOD_TYPE]
default set extcommunity none
```

Parameters

- ***COND_X*** Specifies extended community route map modification. Command may contain mulitple attributes. Options include:
  - **rt** *ASN:nn* Route target attribute (AS:network number).
  - **rt** *IP-address:nn* Route target attribute (IP address: network number).
  - **soo** *ASN:nn* Site of origin attribute (AS:network number).
  - **soo** *IP-address:nn* Site of origin attribute (IP address: network number).
- ***MOD_TYPE*** Specifies route map modification method. Options include:
  - <no parameter> command replaces existing route map with specified parameters.
  - **additive** command adds specified parameters to existing route map.
  - **delete** command removes specified parameters from existing route map.

Related Commands

- **route-map** enters route-map configuration mode.
- **set (route-map)** specifies attribute modifications for the redistributed routes
- **set (route-map)** specifies community modifications for the redistributed routes.

Example

- This command creates a route-map entry in map1 that sets the route target extended community attribute.

```
switch(config)#route-map map1
switch(config-route-map-map1)#set extcommunity rt 10.13.2.4:100
switch(config-route-map-map1)#
```

show (ACL configuration modes)

The **show** command displays the ACL (Access Control List) contents:

- **show** or **show pending** – displays the list as modified in ACL configuration mode.
- **show active** – displays the list as stored in ***running-config***.
- **show comment** – displays the comment stored with the list.
- **show diff** – displays the modified and stored lists, with flags denoting the modified rules.

Exiting the ACL configuration mode stores all pending ACL changes to ***running-config***.

Command Mode

ACL Configuration
IPv6-ACL Configuration
Std-ACL Configuration
Std-IPv6-ACL Configuration
MAC-ACL Configuration

Command Syntax

```
show
show active
show comment
show diff
show pending
```

Examples

The examples in this section assume these ACL commands are entered as specified.

These commands are stored in running-config:

```
10 permit ip 10.10.10.0/24 any
20 permit ip any host 10.21.10.1
30 deny ip host 10.10.10.1 host 10.20.10.1
40 permit ip any any
50 remark end of list
```

The current edit session removed this command. This change is not yet stored to running-config:

```
20 permit ip any host 10.21.10.1
```

The current edit session added these commands ACL. They are not yet stored to running-config:

```
20 permit ip 10.10.0.0/16 any
25 permit tcp 10.10.20.0/24 any
45 deny pim 239.24.124.0/24 10.5.8.4/30
```

- This command displays the ACL, as stored in the configuration.

```
switch(config-acl-test_1)#show active
IP Access List test_1
        10 permit ip 10.10.10.0/24 any
        20 permit ip any host 10.21.10.1
        30 deny ip host 10.10.10.1 host 10.20.10.1
        40 permit ip any any
        50 remark end of list
```

- This command displays the pending ACL, as modified in ACL configuration mode.

```
switch(config-acl-test_1)#show pending
IP Access List test_1
        10 permit ip 10.10.10.0/24 any
        20 permit ip 10.10.0.0/16 any
        25 permit tcp 10.10.20.0/24 any
        30 deny ip host 10.10.10.1 host 10.20.10.1
        40 permit ip any any
        45 deny pim 239.24.124.0/24 10.5.8.4/30
        50 remark end of list
```

- This command displays the difference between the saved and modified ACLs.
  - Rules added to the pending list are denoted with a plus sign (+).
  - Rules removed from the saved list are denoted with a minus sign (-)

```
switch(config-acl-test_1)#show diff
---
+++
@@ -1,7 +1,9 @@
 IP Access List test_1
         10 permit ip 10.10.10.0/24 any
-        20 permit ip any host 10.21.10.1
+        20 permit ip 10.10.0.0/16 any
+        25 permit tcp 10.10.20.0/24 any
         30 deny ip host 10.10.10.1 host 10.20.10.1
         40 permit ip any any
+        45 deny pim 239.24.124.0/24 10.5.8.4/30
```

show ip access-lists

The **show ip access-list** command displays the contents of IPv4 and standard IPv4 access control lists (ACLs) on the switch. Use the **summary** option to display only the name of the lists and the number of lines in each list.

Command Mode

Privileged EXEC

Command Syntax

show ip access-list [*LIST*] [*SCOPE*]

Parameters

- ***LIST*** name of lists to be displayed. Selection options include:
  - <no parameter> all IPv4 ACLs are displayed.
  - *list_name* specified IPv4 ACL is displayed.
- ***SCOPE*** information displayed. Selection options include:
  - <no parameter> all rules in the specified lists are displayed.
  - **summary** the number of rules in the specified lists are displayed.

Examples

- This command displays all rules in ***test1*** IPv4 ACL.

```
switch#show ip access-list list2
IP Access List list2
        10 permit ip 10.10.10.0/24 any
        20 permit ip any host 10.20.10.1
        30 deny ip host 10.10.10.1 host 10.20.10.1
switch#
```

- This command displays the name of, and number of rules in, each list on the switch.

```
switch#show ip access-list summary
IPV4 ACL default-control-plane-acl
        Total rules configured: 12
        Configured on: control-plane
        Active on    : control-plane

IPV4 ACL list2
        Total rules configured: 3

IPV4 ACL test1
        Total rules configured: 6

Standard IPV4 ACL test_1
        Total rules configured: 1

IPV4 ACL test_3
        Total rules configured: 0

switch#
```

show ip prefix-list

The **show ip prefix-list** command displays all rules for the specified IPv4 prefix list. The command displays all IPv4 prefix list rules if a prefix list name is not specified.

Command Mode

EXEC

Command Syntax

show ip prefix-list [*DISPLAY_ITEMS*]

Parameters

- ***DISPLAY_ITEMS*** specifies the name of prefix lists for which rules are displayed. Options include:
  - <no parameter> all IPv4 prefix list rules are displayed.
  - *list_name* specifies the IPv4 prefix list for which rules are displayed.

Examples

- This command displays all rules in the route-one IPv4 prefix list:

```
switch>show ip prefix-list route-one
ip prefix-list route-one seq 10 deny 10.1.1.0/24
ip prefix-list route-one seq 20 deny 10.1.0.0/16
switch>
```

- This command displays all prefix list rules:

```
switch>show ip prefix-list
ip prefix-list route-one seq 10 deny 10.1.1.0/24
ip prefix-list route-one seq 20 deny 10.1.0.0/16
ip prefix-list route-two seq 10 deny 10.1.1.0/24 ge 26 le 30
ip prefix-list route-two seq 20 deny 10.1.0.0/16
ip prefix-list route-two seq 30 deny 3.3.3.3/32
ip prefix-list route-two seq 500 deny 1.1.1.0/24 ge 28 le 30
switch>
```

show ipv6 access-lists

The **show ipv6 access-list** command displays the contents of all IPv6 access control lists (ACLs) on the switch. Use the **summary** option to display only the name of the lists and the number of lines in each list.

Command Mode

Privileged EXEC

Command Syntax

show ipv6 access-list [*LIST*] [*SCOPE*]

Parameters

- ***LIST*** name of lists to be displayed. Selection options include:
  - <no parameter> all IPv6 ACLs are displayed.
  - *list_name* specified IPv6 ACL is displayed.
- ***SCOPE*** information displayed. Selection options include:
  - <no parameter> all rules in the specified lists are displayed.
  - **summary** the number of rules in the specified lists are displayed.

Examples

- This command displays all rules in test1 IPv6 ACL.

```
switch#show ipv6 access-list list2
IP Access List list2
        10 permit ipv6 3891:3c58:6300::/64 any
        20 permit ipv6 any host 2fe1:b468:024a::
        30 deny ipv6 host 3411:91c1:: host 4210:cc23:d2de:::
switch#
```

- This command displays the name of, and number of rules in, each list on the switch.

```
switch#show ipv6 access-list summary
IPV6 ACL list2
        Total rules configured: 3

IPV6 ACL test1
        Total rules configured: 6

IPV6 ACL test_1
        Total rules configured: 1

Standard IPV6 ACL test_3
        Total rules configured: 0

switch#
```

show ipv6 prefix-list

The **show ipv6 prefix-list** command displays all rules for the specified IPv6 prefix list. The command displays all IPv6 prefix lists if a prefix list name is not specified.

Command Mode

EXEC

Command Syntax

show ipv6 prefix-list [*DISPLAY_ITEMS*]

Parameters

- ***DISPLAY_ITEMS*** specifies the name of prefix lists for which rules are displayed. Options include:
  - <no parameter> all IPv6 prefix lists are displayed.
  - *list_name* specifies the IPv6 prefix list for which rules are displayed.

Examples

- This command displays all rules in the map1 IPv6 prefix list:

```
switch>show ipv6 prefix-list map1
ipv6 prefix-list map1
seq 10 permit 3:4e96:8ca1:33cf::/64
seq 15 deny 3:4400::/64
seq 20 permit 3:11b1:8fe4:1aac::/64
seq 30 permit 3:1bca:3ff2:634a::/64
seq 40 permit 3:1bca:1141:ab34::/64
switch>
```

- This command displays all prefix lists:

```
switch>show ipv6 prefix-list
ipv6 prefix-list map1
seq 10 permit 3:4e96:8ca1:33cf::/64
seq 15 deny 3:4400::/64
seq 20 permit 3:11b1:8fe4:1aac::/64
seq 30 permit 3:1bca:3ff2:634a::/64
seq 40 permit 3:1bca:1141:ab34::/64
ipv6 prefix-list FREDD
ipv6 prefix-list route-five
ipv6 prefix-list map2
seq 10 deny 10:1:1:1::/64 ge 72 le 80
seq 20 deny 10:1::/32
switch>
```

show mac access-lists

The **show mac access-list** command displays the contents of all MAC access control lists (ACLs) on the switch. Use the summary to display only the name of the lists and the number of lines in each list.

Command Mode

Privileged EXEC

Command Syntax

show mac access-lists [*LIST*] [*SCOPE*]

Parameters

- ***LIST*** name of lists to be displayed. Selection options include:
  - <no parameter> command displays all ACLs.
  - *list_name* command displays ACL specified by parameter.
- ***SCOPE*** information displayed. Selection options include:
  - <no parameter> command displays all rules in specified lists.
  - **summary** command displays the number of rules in specified lists.

Examples

- This command displays all rules in ***mtest2*** MAC ACL.

```
switch#show mac access-list mlist2
IP Access List mlist2
        10 permit 1024.4510.F125 0.0.0 any aarp
        20 permit any 4100.4500.0000 0.FF.FFFF novell
        30 deny any any
switch#
```

- This command displays the number of rules in each MAC ACL on the switch.

```
switch#show mac access-list summary
MAC ACL mlist1
        Total rules configured: 6

MAC ACL mlist2
        Total rules configured: 3

MAC ACL mlist3
        Total rules configured: 1

MAC ACL mlist4
        Total rules configured: 0

switch#
```

show route-map

The **show route-map** command displays the contents of the specified route maps. The command displays all route maps if an individual map is not specified.

Command Mode

EXEC

Command Syntax

show route-map [*MAP*]

Parameters

- ***MAP*** name of maps to be displayed. Selection options include:
  - <no parameter> command displays all ACLs.
  - *map_name* route map that the command displays.

Example

- This command displays the ***map1*** route map.

```
switch>show route-map map1
route-map map1 permit 5
  Match clauses:
    match as 456
Set clauses:
route-map map1 permit 10
  Match clauses:
match ip next-hop 2.3.4.5
    match as-path path_2
  Set clauses:
    set local-preference 100
```

statistics per-entry (ACL configuration modes)

The **statistics per-entry** command places the ACL in counting mode. An ACL in counting mode displays the number of instances each rule in the list matches an inbound packet and the elapsed time since the last match. The show access list commands display the statistics next to each rule in the ACL.

On the FM6000 platform, this command has no effect when used in an ACL that is part of a PBR class map.

The **no statistics per-entry** and **default statistics per-entry** command places the ACL in non-counting mode.

Command Mode

ACL Configuration
IPv6-ACL Configuration
Std-ACL Configuration
Std-IPv6-ACL Configuration
MAC-ACL Configuration

Command Syntax

```
statistics per-entry
no statistics per-entry
default statistics per-entry
```

Examples

- This command places the test1 ACL in counting mode.

```
switch(config)#ip access-list test1
switch(config-acl-test1)#statistics per-entry
switch(config-acl-test1)#
```

- This command displays the ACL, with counter information, for an ACL in counting mode.

```
switch#show ip access-lists
IP Access List default-control-plane-acl [readonly]
     statistics per-entry
     10 permit icmp any any
     20 permit ip any any tracked [match 12041, 0:00:00 ago]
     30 permit ospf any any
     40 permit tcp any any eq ssh telnet www snmp bgp https [match 11, 1:41:07 ago]
     50 permit udp any any eq bootps bootpc snmp rip [match 78, 0:00:27 ago]
     60 permit tcp any any eq mlag ttl eq 255
     70 permit udp any any eq mlag ttl eq 255
     80 permit vrrp any any
     90 permit ahp any any
     100 permit pim any any
     110 permit igmp any any [match 14, 0:23:27 ago]
     120 permit tcp any any range 5900 5910
     130 permit tcp any any range 50000 50100
     140 permit udp any any range 51000 51100
```

Chapter 21

VRRP and VARP

A virtual IP (VIP) address is an IP address that does not directly connect to a specific interface. Inbound packets sent to a Virtual IP address are redirected to a physical network interface. VIPs supports connection redundancy by assigning the address to multiple switches. If one device becomes unavailable, packets sent to the address are still serviced by the functioning device.

Arista switches support virtual IP addresses through Virtual Router Redundancy Protocol, version 2 (VRRPv2), Virtual Router Redundancy Protocol, version 3 (VRRPv3), and Virtual-ARP (VARP). This chapter describes the Arista switch support of virtual IP addresses and contains these sections:

- Section 21.1: VRRP and VARP Conceptual Overview
- Section 21.2: VRRP and VARP Implementation Procedures
- Section 21.3: VRRP and VARP Implementation Examples
- Section 21.4: VRRP and VARP Configuration Commands

21.1 VRRP and VARP Conceptual Overview

21.1.1 VRRPv2

A virtual router, also known as a *virtual router group*, is defined by a virtual router identifier (VRID) and a virtual IP address. A virtual router's mapping of VRID and IP address must be consistent among all switches implementing the virtual router group. A virtual router's scope is restricted to a single LAN.

A LAN may contain multiple virtual routers for distributing traffic. Each virtual router on a LAN is assigned a unique VRID. A switch may be configured with virtual routers among multiple LANs.

VRRP uses priority ratings to assign Master or Backup roles for each VRRP router configured for a virtual router group. The Master router sends periodic VRRP Advertisement messages along the LAN and forwards packets received by the virtual router to their destination. Backup routers are inactive but are available to assume Master router duties when the current Master fails.

A VRRP can be configured to allow VRRP routers with higher priority to take over Master router duties. Alternatively, the group can be configured to prevent a router from preemptively assuming the Master role. A VRRP router is always assigned the Master of any virtual router configured with the address owned by the VRRP router, regardless of the preemption prevention setting.

21.1.2 VRRPv3

RFC 5798 defines version 3 of the Virtual Router Redundancy Protocol (VRRP) for both IPv4 and IPv6. It is based on version 2 of VRRP, as defined in RFC 3768 .

21.1.3 VARP

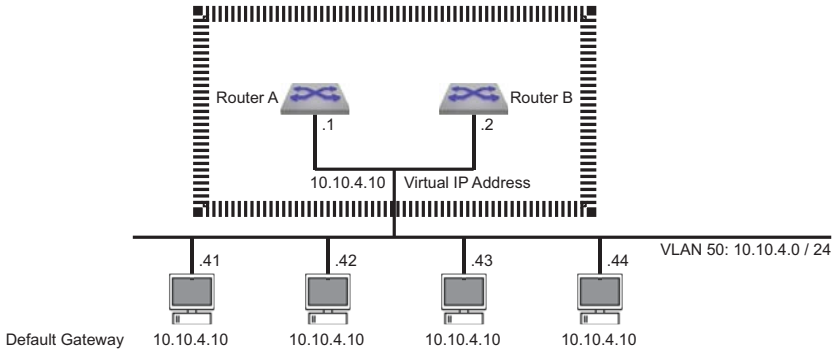
Virtual-ARP (VARP) allows multiple switches to simultaneously route packets from a common IP address in an active-active router configuration. Each switch is configured with the same set of virtual IP addresses on corresponding VLAN interfaces and a common virtual MAC address. In MLAG configurations, VARP is preferred over VRRP because VARP does not require traffic to traverse the peer-link to the master router as VRRP would.

A maximum of 500 virtual IP addresses can be assigned to a VLAN interface. All virtual addresses on all VLAN interfaces resolve to the same virtual MAC address.

VARP functions by having each switch respond to ARP and GARP requests for the configured router IP address with the virtual MAC address. The virtual MAC address is only for inbound packets and never used in the source field of outbound packets.

When ***ip routing*** is enabled, packets to the virtual MAC address are routed to the next hop destination.

Figure 21-1: VARP Configuration



21.2 VRRP and VARP Implementation Procedures

This section contains the following configuration instructions:

- Section 21.2.1: VRRP Configuration for IPv4
- Section 21.2.2: VRRP Configuration for IPv6
- Section 21.2.3: VARP Configuration

21.2.1 VRRP Configuration for IPv4

To implement a virtual router, it must be configured and enabled. A virtual router is typically configured before it is enabled; this ensures that the VRRP router operates as required before its priority settings immediately make it the master virtual router. Because assigning a primary address to a virtual router enables it, address assignment is normally performed after all other configuration tasks.

The **no vrrp** command removes all vrrp commands for the specified virtual router from ***running-config***.

21.2.1.1 Virtual Router Configuration

Most configuration tasks are optional because all mandatory parameters have a default value. The following virtual router parameters are configurable:

- VRRP version (default = version 2)
- Router priority (default = 100)
- Preemption option (default is enabled)
- Advertisement timer (default = one second)
- Description (optional parameter)
- Authentication (optional parameter)
- Secondary IP addresses (optional parameter)

VRRP version

The **vrrp ip version** command sets the version of VRRP used on an interface. The version selected in a VRRP group must be the same for all group members. By default, Arista switches use VRRP version 2, which supports only IPv4 environments. VRRP version 3 supports both IPv4 and IPv6 environments.

Example

- This command causes VLAN 20 to use VRRP version 3.

```
switch(config)#interface vlan 20
switch(config-if-vl20)#vrrp 1 ip version 3
switch(config-if-vl20)#
```

Master and Backup Router

The VRRP routers within a virtual router group determine the Master router through priority settings. Priority values range from 254 (highest priority) to 1 (lowest priority). Priority is either set by a CLI command or is assigned the default value of 100. A switch specifies priority settings for each of its virtual routers. Once set, VRRP priority can also be changed by a tracked object. The **vrrp track** command configures the VRRP client process to track an object created by the **vrrp track** command and react if its status changes to ***down***.

Preemption mode determines when a VRRP router with a higher priority rating becomes the Master router. If preemption is enabled, the VRRP router with the highest priority immediately becomes the Master router. If preemption is disabled, a VRRP router with a higher priority value does not become the Master router unless the current Master becomes unavailable; this is applicable when a new VRRP router becomes available on the LAN or VRRP router's priority value changes for the virtual router.

The **vrrp priority** command configures the switch's priority setting for the specified virtual router.

Example

- This command sets the priority value of 250 for the virtual router with VRID 15 on VLAN 20.

```
switch(config-if-vl20)#vrrp 15 priority 250
switch(config-if-vl20)#
```

The **vrrp preempt** command controls the preempt mode setting of the specified virtual router. By default, preempt mode is enabled.

Examples

- This command disables preempt mode for the virtual router 15 on VLAN 20.

```
switch(config-if-vl20)#no vrrp 15 preempt
switch(config-if-vl20)#
```

- This command enables preempt mode for the virtual router 30 on VLAN 20.

```
switch(config-if-vl20)#vrrp 30 preempt
switch(config-if-vl20)#
```

The **vrrp preempt delay** command configures a period between an event that elevates a switch to master vrrp router status and the switch's assumption of master vrrp router role. Command options configure delays during normal operation and after a switch reboot.

Advertisement Timer

The Master router sends periodic VRRP Advertisement messages to other VRRP routers. The **vrrp timers advertise** command specifies the interval between successive advertisement message transmissions.

The advertisement interval also defines the timeout that determines when the switch assumes the Master router role. This timeout interval is three times the advertisement interval.

Example

- This command sets the advertisement interval of 10 seconds for virtual router 35 on VLAN 100.

```
switch(config-if-vl100)#vrrp 35 timers advertise 10
switch(config-if-vl100)#
```

Description

The **vrrp description** command associates a text string to the specified virtual router. The maximum string length is 80 characters. The string has no functional impact on the virtual router.

Example

- This command associates the text string ***Laboratory Router*** to virtual router 15 on VLAN 20.

```
switch(config-if-vl20)#vrrp 15 description Laboratory Router
switch(config-if-vl20)#
```

Authentication

VRRP authentication validates VRRP advertisement packets that the switch receives from other VRRP routers in a specified virtual router group. When a virtual router uses authentication, all VRRP routers in the group must use the same authentication parameters.

The **vrrp authentication** command configures virtual router authentication parameters for the specified virtual router.

Example

- This command implements plain-text authentication, using 12345 as the key, for virtual router 40 on VLAN 100.

```
switch(config-if-vl100)#vrrp 40 authentication text 12345
switch(config-if-vl100)#
```

Secondary Addresses

The **vrrp ip secondary** command assigns a secondary IP address to a virtual router. Secondary addresses are optional; a virtual router's configuration may include more than one secondary address command. The primary and secondary address list must be identical for all switches in a virtual router group.

A primary IP address is assigned to a virtual router with the **vrrp ip** command (Section 21.2.1.2).

Example

- This command assigns the IP address of 10.2.4.5 as the secondary IP address for the virtual router 15 on VLAN 20

```
switch(config-if-vl20)#vrrp 15 ip 10.2.4.5 secondary
switch(config-if-vl20)#
```

21.2.1.2 Virtual Router Enabling and the Primary IP address

The **vrrp ip** command configures the primary IP address of the specified virtual router and enables the virtual router if the primary address is contained within the configuration mode interface's IP address subnet. A virtual router's configuration may contain only one primary IP address assignment command; subsequent **vrrp ip** commands reassign the virtual router's primary IP address.

Example

- This command enables virtual router group 15 (VRID) on VLAN 20 and assigns 10.1.1.5 as the virtual router's primary address.

```
switch(config-if-vl20)#vrrp 15 ip 10.1.1.5
switch(config-if-vl20)#
```

21.2.1.3 VRRP Disabling and Shutdown

The **vrrp shutdown** command places the switch in stopped state for the specified virtual router. While in stopped state, the switch cannot act as a Master or backup router for the virtual router group. The ***no vrrp shutdown*** command changes the switch's virtual router state to ***backup*** or ***master*** if the virtual router is properly configured.

VRRP can also be shut down when the status of a tracked object configured by the **vrrp track** command changes to **down**.

Examples

- This command places the switch in stopped mode for virtual router 24 on VLAN 20.

```
switch(config-if-vl20)#vrrp 24 shutdown
switch(config-if-vl20)#
```

- This command moves the switch out of stopped mode for virtual router 24 on VLAN 20.

```
switch(config-if-vl20)#no vrrp 24 shutdown
switch(config-if-vl20)#
```

- This command configures the switch to enter stopped mode for virtual router 24 on VLAN 20 if the status of tracked object interfaceE6/48 changes to **down**.

```
switch(config-if-vl20)#vrrp 24 track interfaceE6/48 shutdown
switch(config-if-vl20)#
```

The **no vrrp** and **no vrrp ip** commands delete the specified virtual IP address from the interface. Additionally, the **no vrrp** command removes all residual VRRP commands for the virtual router.

Examples

- This command removes all vrrp configuration commands for virtual router 10 on VLAN 15.

```
switch(config-if-vl15)#no vrrp 10
switch(config-if-vl15)#
```

- This command disables virtual router 25 on VLAN 20 and removes the primary IP address from its configuration.

```
switch(config-if-vl20)#no vrrp 25 ip 10.1.1.5
switch(config-if-vl20)#
```

21.2.2 VRRP Configuration for IPv6

To implement a virtual router, it must be configured and enabled. A virtual router is typically configured before it is enabled; this ensures that the VRRP router operates as required before its priority settings immediately make it the master virtual router. Because assigning a primary address to a virtual router enables it, address assignment is normally performed after all other configuration tasks.

The **no vrrp** command removes all VRRP commands for the specified virtual router from ***running-config.***

21.2.2.1 Configuring VRRP for IPv6

Specify the VRRP version

The **vrrp ip version** command sets the version of VRRP used on an interface. The version selected in a VRRP group must be the same for all group members. By default, Arista switches use VRRP version 2, which is not compatible with IPv6.

Example

- This command causes VLAN 20 to use VRRP version 3.

```
switch(config)#interface vlan 20
switch(config-if-vl20)#vrrp 1 ip version 3
switch(config-if-vl20)#
```

Create a VRRP Group and Configure a Virtual IPv6 Address

The **vrrp ipv6** command assigns an IPv6 address to the interface being configured and creates a VRRP group.

Example

- These commands create VRRP group 3 and configure a virtual IPv6 address for the VRRP group on the VLAN 20 interface.

```
switch(config)#interface vlan 20
switch(config-if-vl20)#vrrp 3 ipv6 2001:db8:0:1::1
switch(config-if-vl20)#
```

ConfigureTracking

The **vrrp track** command configures the VRRP client process to track an object created by the **vrrp track** command and react if its status changes to ***down***.

The **vrrp track** command configures VRRP to track a specified track entry.

Example

- This command causes interface VLAN 20 to disable VRRP when tracked object ETH8 changes state.

```
switch(config-if-vl20)#vrrp 1 track ETH8 shutdown
switch(config-if-vl20)#
```

Configure the Priority

The **vrrp priority** command configures the switch's priority setting for the specified virtual router.

Example

- This command sets the priority value of 250 for the virtual router with VRID 15 on VLAN 20.

```
switch(config-if-vl20)#vrrp 15 priority 250
switch(config-if-vl20)#
```

Configure the Preemption Mode

Preemption mode determines when a VRRP router with a higher priority rating becomes the Master router. If preemption is enabled, the VRRP router with the highest priority immediately becomes the Master router. If preemption is disabled, a VRRP router with a higher priority value does not become the Master router unless the current Master becomes unavailable; this is applicable when a new VRRP router becomes available on the LAN or VRRP router's priority value changes for the virtual router.

The **vrrp preempt** command controls the preempt mode setting of the specified virtual router. By default, preempt mode is enabled.

Example

- This command enables preempt mode for the virtual router 30 on VLAN 20.

```
switch(config-if-vl20)#vrrp 30 preempt
```

Configure the VRRP Advertisement Interval

The **ip virtual-router mac-address advertisement-interval** command specifies the interval between advertisement packets sent by the master router to the VRRP group members.

Examples

- This command configures a MAC address advertisement interval of one minute (60 seconds).

```
switch(config)#interface vlan 20
switch(config-if-vl20)#ip virtual-router mac-address advertisement-interval 60
switch(config-if-vl20)#
```

21.2.2.2 Verify VRRP IPv6 Configurations

Use the following commands to display the VRRP configurations and status.

Show VRRP Group

The **show vrrp** command displays information about the Virtual Router Redundancy Protocol (VRRP) groups configured on a specified interface.

Examples

- This command displays a table of information for VRRP groups on the switch.

```
switch>show vrrp interface vlan 3060 brief
Interface Id  Ver Pri Time  State   VrIps
Vlan3060  1   3   100 3609  Master  2001::2
                                    2001::3
Vlan3060  2   3   100 3609  Master  2002::2
                                    2002::3
switch>
```

Show VRRP Internal

The **show vrrp internal** command displays the internal Pluggable Authentication Modules(PAM) packet counters on the switch.

Examples

- This command displays the internal PAM packet counters on the switch.

```
switch>show vrrp internal
VRRP PAM Counters
----------------
ARP Responder:
         numSent : 0
         numRcvd : 0
      numBadRcvd : 0
ND Responder:
         numSent : 0
         numRcvd : 0
      numBadRcvd : 0
IPv4 VRRP Packet Manager:
         numSent : 0
         numRcvd : 0
      numBadRcvd : 0
IPv6 VRRP Packet Manager:
         numSent : 0
         numRcvd : 0
      numBadRcvd : 0
switch>
```

21.2.3 VARP Configuration

Implementing VARP consists of assigning virtual IP addresses to VLAN interfaces and configuring a virtual MAC address.

Virtual IP Addresses

The **ip virtual-router address** command assigns a virtual IP address to the VLAN interface being configured. Unlike VRRP, the virtual IP address does not have to be in the same subnet as the physical interface.

A virtual IPv4 address may optionally be configured with a subnet, but doing so will modify the behavior of ARP requests sent from the router. When the router sends an ARP request for an IPv4 address in a virtual subnet, the ARP request will use the virtual IPv4 address as the source IP address and the virtual MAC address as the source MAC address inside the ARP header. For virtual IP addresses configured without the subnet option, no modifications are made to outgoing ARP requests.

Examples

- These commands configure a Switch Virtual Interface (SVI) and a virtual IP address for VLAN 10.

```
switch(config)#interface vlan 10
switch(config-if-Vl10)#ip address 10.0.0.2/24
switch(config-if-Vl10)#ip virtual-router address 10.0.0.6
switch(config-if-Vl10)#ip address 2001::1/64
switch(config-if-Vl10)#ip6 virtual-router address 2001::2
switch(config-if-Vl10)#exit
switch(config)#
```

- These commands configure a Switch Virtual Interface (SVI) and a virtual IPv4 address with a subnet for VLAN 10. A static route is added to indicate that the virtual subnet is reachable through VLAN 10.

```
switch(config)#ip route 192.0.0.0/24 vlan 10
switch(config)#interface vlan 10
switch(config-if-Vl10)#ip address 10.0.0.2/24
switch(config-if-Vl10)#ip virtual-router address 192.0.0.6/24
switch(config-if-Vl10)#exit
switch(config)#
```

Virtual MAC Address

The **ip virtual-router mac-address** command assigns a virtual MAC address to the switch. The switch maps all virtual router IP addresses to this MAC address. The address is receive-only; the switch never sends packets with this address as the source.

When the destination MAC of a packet destined to a remote network matches the virtual MAC address, the MLAG peer forwards the traffic to the next hop destination. Each MLAG peer must have the same routes available, either though static configuration or learned through a dynamic routing protocol.

Example

- This command configures a virtual MAC address.

```
switch(config)#ip virtual-router mac-address 001c.7300.0099
switch(config)#
```

Show Virtual MAC Address

To display the virtual router MAC and IP addresses, enter the **show ip virtual-router** command.

Example

- This command displays the virtual router addresses assigned on the switch.

```
switch>show ip virtual-router
IP virtual router is configured with MAC address: 24cd.5a29.cc31
Interface  IP Address        Virtual IP Address   Status            Protocol
Vlan15     10.1.1.3/24       10.1.1.15            up                up
Vlan15     10.1.1.3/24       10.1.1.16            up                up
Vlan15     10.1.1.3/24       10.1.1.17            up                up
Vlan20     10.12.1.6/24      10.1.1.51            up                up
Vlan20     10.12.1.6/24      10.1.1.53            up                up
Vlan20     10.12.1.6/24      10.1.1.55            up                up
switch>
```

Show IPv6 Virtual-Router

The **show ipv6 virtual-router** command displays the virtual MAC address assigned to the switch and all virtual IPv6 addresses assigned to each VLAN interface.

Examples

- This command displays a table of information for IPv6 VRRP groups on the switch.

```
switch>show ipv6 virtual-router
IP virtual router is configured with MAC address: 001c.7300.0099
MAC address advertisement interval: 30 seconds
Interface Vlan4094
  State is up
  Protocol is up
  IPv6 address
    2001:b8:2001::1011/64
  Virtual IPv6 address
    2001:db8:ac10:fe01::
switch>
```

21.3 VRRP and VARP Implementation Examples

This section contains the following example set:

- Section 21.3.1: VRRP Examples
- Section 21.3.2: VARP Example

21.3.1 VRRP Examples

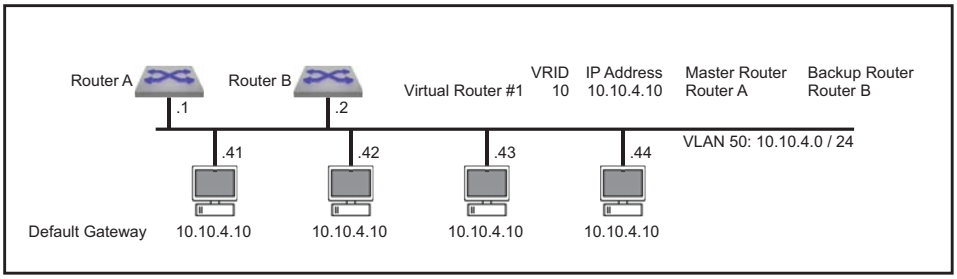
This section provides code that implements three VRRP configurations:

- Example 1 configures two switches in a single virtual router group. This implementation protects the LAN against the failure of one router.
- Example 2 configures two switches into two virtual routers within a single LAN. This implementation protects the LAN against the failure of one router and balances traffic between the routers.
- Example 3 configures three switches to implement virtual routers on two LANs. Each LAN contains two virtual routers. One switch is configured into four virtual routers – two on each LAN.

21.3.1.1 VRRP Example 1: One Virtual Router on One LAN

Figure 21-2 displays the Example 1 network. Two switches are configured as VRRP routers to form one virtual router.

Figure 21-2: VRRP Example 1 Network Diagram



The following code configures the first switch (Router A) as the master router and the second switch (Router B) as a backup router for virtual router 10 on VLAN 50. Router A becomes the Master virtual router by setting its priority at 200; Router B maintains the default priority of 100. The advertisement interval is three seconds on both switches. Priority preemption is enabled by default.

Switch code that implements Router A on the first switch

```
switch-A(config)#interface vlan 50
switch-A(config-if-vl50)#ip address 10.10.4.1/24
switch-A(config-if-vl50)#no vrrp 10
switch-A(config-if-vl50)#vrrp 10 priority 200
switch-A(config-if-vl50)#vrrp 10 timers advertise 3
switch-A(config-if-vl50)#vrrp 10 ip 10.10.4.10
switch-A(config-if-vl50)#exit
```

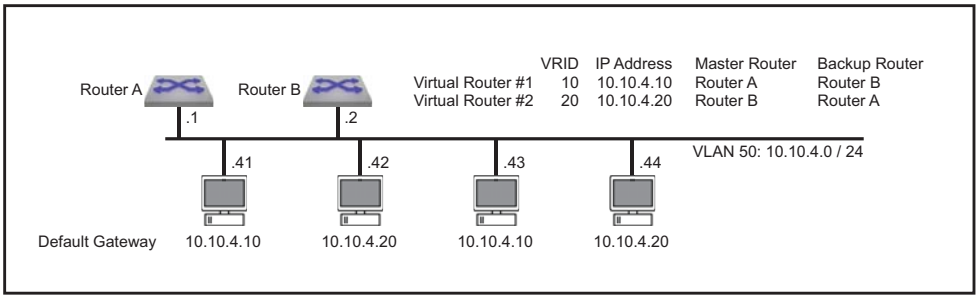
Switch code that implements Router B on the second switch

```
switch-B(config)#interface vlan 50
switch-B(config-if-vl50)#ip address 10.10.4.2/24
switch-B(config-if-vl50)#no vrrp 10
switch-B(config-if-vl50)#vrrp 10 timers advertise 3
switch-B(config-if-vl50)#vrrp 10 ip 10.10.4.10
switch-B(config-if-vl50)#exit
```

21.3.1.2 VRRP Example 2: Two Virtual Routers on One LAN

Figure 21-3 displays Example 2. Two switches are configured as VRRP routers to form two virtual routers on one LAN. Using two virtual routers distributes the LAN traffic between the switches.

Figure 21-3: VRRP Example 2 Network Diagram



The following code configures two switches as a master and a backup router for two virtual routers on VLAN 50.

- Router A is the master for virtual router 10 and backup for virtual router 20.
- Router B is the master for virtual router 20 and backup for virtual router 10.
- VRRP advertisement interval is 3 seconds on virtual router 10 and 5 seconds on virtual router 20.
- Priority preemption is enabled by default for both virtual routers.

Switch code that implements Router A on the first switch

```
switch-A(config)#interface vlan 50
switch-A(config-if-vl50)#ip address 10.10.4.1/24
switch-A(config-if-vl50)#no vrrp 10
switch-A(config-if-vl50)#vrrp 10 priority 200
switch-A(config-if-vl50)#vrrp 10 timers advertise 3
switch-A(config-if-vl50)#vrrp 10 ip 10.10.4.10
switch-A(config-if-vl50)#no vrrp 20
switch-A(config-if-vl50)#vrrp 20 timers advertise 5
switch-A(config-if-vl50)#vrrp 20 ip 10.10.4.20
switch-A(config-if-vl50)#exit
```

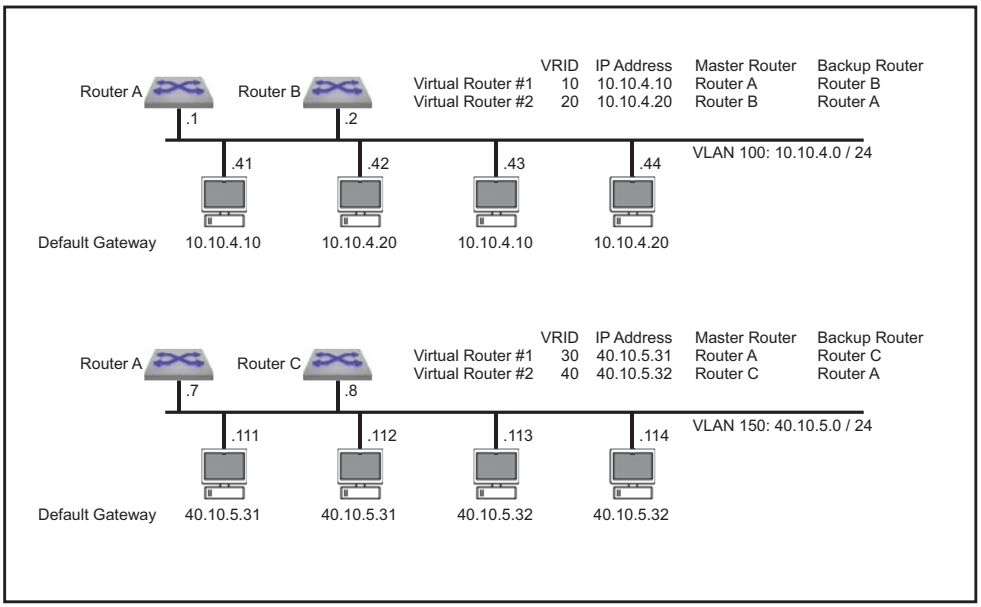
Switch code that implements Router B on the second switch

```
switch-B(config)#interface vlan 50
switch-B(config-if-vl50)#ip address 10.10.4.2/24
switch-B(config-if-vl50)#no vrrp 10
switch-B(config-if-vl50)#vrrp 10 timers advertise 3
switch-B(config-if-vl50)#vrrp 10 ip 10.10.4.10
switch-B(config-if-vl50)#no vrrp 20
switch-B(config-if-vl50)#vrrp 20 priority 200
switch-B(config-if-vl50)#vrrp 20 timers advertise 5
switch-B(config-if-vl50)#vrrp 20 ip 10.10.4.20
switch-B(config-if-vl50)#exit
```

21.3.1.3 VRRP Example 3: Two Virtual Routers on Two LANs

Figure 21-4 displays Example 3. Three switches are configured as VRRP routers to form four virtual router groups – two groups on each of two LANs.

Figure 21-4: VRRP Example 3 Network Diagram



The following code configures the three switches as follows:

- Router A is the master for virtual router 10 and backup for virtual router 20 on VLAN 100.
- Router A is the master for virtual router 30 and backup for virtual router 40 on VLAN 150.
- Router B is the master for virtual router 20 and backup for virtual router 10 on VLAN 100.
- Router C is the master for virtual router 40 and backup for virtual router 30 on VLAN 150.
- VRRP advertisement interval is set to one second on all virtual routers.
- Priority preemption is disabled on all virtual routers.

Switch code that implements Router A on the first switch

```
switch-A(config)#interface vlan 100
switch-A(config-if-vl100)#ip address 10.10.4.1/24
switch-A(config-if-vl100)#no vrrp 10
switch-A(config-if-vl100)#vrrp 10 priority 200
switch-A(config-if-vl100)#no vrrp 10 preempt
switch-A(config-if-vl100)#vrrp 10 ip 10.10.4.10
switch-A(config-if-vl100)#no vrrp 20
switch-A(config-if-vl100)#no vrrp 20 preempt
switch-A(config-if-vl100)#vrrp 20 ip 10.10.4.20
switch-A(config-if-vl100)#interface vlan 150
switch-A(config-if-vl150)#ip address 40.10.5.7/24
switch-A(config-if-vl150)#no vrrp 30
switch-A(config-if-vl150)#vrrp 30 priority 200
switch-A(config-if-vl150)#no vrrp 30 preempt
switch-A(config-if-vl150)#vrrp 30 ip 40.10.5.31
switch-A(config-if-vl150)#no vrrp 40
switch-A(config-if-vl150)#no vrrp 40 preempt
switch-A(config-if-vl150)#vrrp 40 ip 40.10.5.32
switch-A(config-if-vl150)#exit
```

Switch code that implements Router B on the second switch

```
switch-B(config)#interface vlan 100
switch-B(config-if-vl100)#ip address 10.10.4.2/24
switch-B(config-if-vl100)#no vrrp 10
switch-B(config-if-vl100)#no vrrp 10 preempt
switch-B(config-if-vl100)#vrrp 10 ip 10.10.4.10
switch-B(config-if-vl100)#no vrrp 20
switch-B(config-if-vl100)#vrrp 20 priority 200
switch-B(config-if-vl100)#no vrrp 20 preempt
switch-B(config-if-vl100)#vrrp 20 ip 10.10.4.20
switch-B(config-if-vl100)#exit
```

Switch code that implements Router C on the third switch

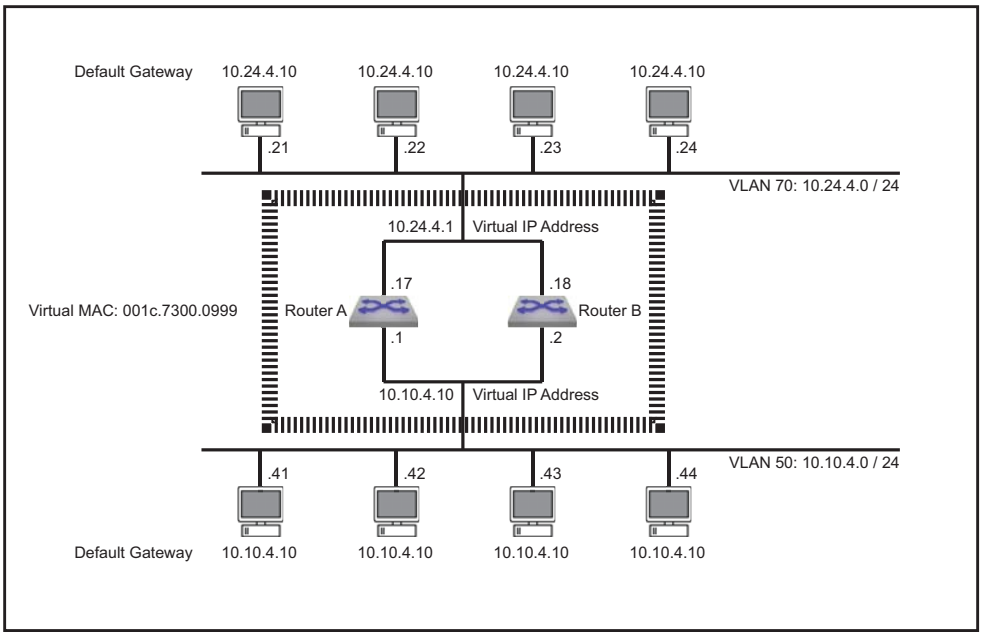
```
switch-C(config)#interface vlan 150
switch-C(config-if-vl150)#ip address 40.10.5.8/24
switch-C(config-if-vl150)#no vrrp 30
switch-C(config-if-vl150)#no vrrp 30 preempt
switch-C(config-if-vl150)#vrrp 30 ip 40.10.5.31
switch-C(config-if-vl150)#no vrrp 40
switch-C(config-if-vl150)#vrrp 40 priority 200
switch-C(config-if-vl150)#no vrrp 40 preempt
switch-C(config-if-vl150)#vrrp 40 ip 40.10.5.32
switch-C(config-if-vl150)#exit
```

21.3.2 VARP Example

This section provides code that implements a VARP configuration. Figure 21-5 displays the Example 1 network. Two switches in an MLAG domain are configured as VARP routers.

The following code configures 10.10.4.10 as the virtual IP address for VLAN 50, 10.24.4.1 as the virtual IP address for VLAN 70, and 001c.7300.0999 as the virtual MAC address on both switches.

Figure 21-5: VARP Example Network Diagram



Switch code that implements VARP on the first switch

```
switch-A(config)#ip virtual-router mac-address 001c.7300.0999
switch-A(config)#interface vlan 50
switch-A(config-if-vl50)#ip address 10.10.4.1/24
switch-A(config-if-vl50)#ip virtual-router address 10.10.4.10
switch-A(config-if-vl50)#interface vlan 70
switch-A(config-if-vl70)#ip address 10.24.4.17/24
switch-A(config-if-vl70)#ip virtual-router address 10.24.4.1
switch-A(config-if-vl70)#exit
```

Switch code that implements VARP on the second switch

```
switch-B(config)#ip virtual-router mac-address 001c.7300.0999
switch-B(config)#interface vlan 50
switch-B(config-if-vl50)#ip address 10.10.4.2/24
switch-B(config-if-vl50)#ip virtual-router address 10.10.4.10
switch-B(config-if-vl50)#interface vlan 70
switch-B(config-if-vl70)#ip address 10.24.4.18/24
switch-B(config-if-vl70)#ip virtual-router address 10.24.4.1
switch-B(config-if-vl70)#exit
```

21.4 VRRP and VARP Configuration Commands

This section contains descriptions of CLI commands that support VRRP and VARP.

Global Configuration Commands

- ip fhrp accept-mode
- ip virtual-router mac-address
- ip virtual-router mac-address advertisement-interval

Interface Configuration Commands – Ethernet, Port Channel, and VLAN Interfaces

- ip virtual-router address
- ipv6 virtual-router address
- no vrrp
- vrrp authentication
- vrrp delay reload
- vrrp description
- vrrp ip
- vrrp ip secondary
- vrrp ip version
- vrrp ipv6
- vrrp mac-address advertisement-interval
- vrrp preempt
- vrrp preempt delay
- vrrp priority
- vrrp shutdown
- vrrp timers advertise
- vrrp track

Privileged EXEC Commands

- show ip virtual-router
- show ipv6 virtual-router
- show vrrp
- show vrrp internal

ip fhrp accept-mode

The **ip fhrp accept-mode** command configures the switch to permit SSH access to the VRRP Master and VARP Master router. All routers within a VRRP or VARP group should be configured consistently. By default, SSH access to the VRRP and VARP Master routers is not permitted.

The **no ip fhrp accept-mode** and **default ip fhrp accept-mode** commands restores the default SSH access availability by removing the **ip fhrp accept-mode** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
ip fhrp accept-mode
no ip fhrp accept-mode
default ip fhrp accept-mode
```

Example

- This command configures the switch to permit SSH access to the VRRP and VARP Master routers.

```
switch(config)#ip fhrp accept-mode
switch(config)#show running-config

!
ip fhrp accept-mode
!

switch(config)#
```

ip virtual-router address

The **ip virtual-router address** command assigns a virtual IPv4 address to the VLAN interface being configured. (To assign a virtual IPv6 address to a VLAN interface, use the **ipv6 virtual-router address** command.) Unlike VRRP, the virtual IP address does not have to be in the same subnet as the physical interface.

A virtual IP address may optionally be configured with a subnet, but doing so will modify the behavior of ARP requests sent from the router. When the router sends an ARP request for an IP address in a virtual subnet, the ARP request will use the virtual IP address as the source IP address and the virtual MAC address as the source MAC address inside the ARP header. For virtual IP addresses configured without the subnet option, no modifications are made to outgoing ARP requests.

A maximum of 500 virtual IP addresses can be assigned to a VLAN interface. All virtual addresses on all VLAN interfaces resolve to the same virtual MAC address configured through the **ip virtual-router mac-address** command.

This command is typically used in MLAG configurations to create identical virtual routers on switches connected to the MLAG domain through an MLAG.

The **no ip virtual-router address** and **default ip virtual-router address** commands remove the specified virtual IP address from the configuration mode interface by deleting the corresponding **ip virtual-router address** command from ***running-config***. If the command does not specify an address, all virtual IPv4 addresses are removed from the interface.

Command Mode

Interface-VLAN Configuration

Command Syntax

```
ip virtual-router address ipv4_addr
no ip virtual-router address [ipv4_addr]
default ip virtual-router address [ipv4_addr]
```

Parameters

- *ipv4_addr* IP address of router. Dotted decimal notation.

Examples

- These commands configure a Switch Virtual Interface (SVI) and a virtual IP address for VLAN 10.

```
switch(config)#interface vlan 10
switch(config-if-Vl10)#ip address 10.0.0.2/24
switch(config-if-Vl10)#ip virtual-router address 10.0.0.6
switch(config-if-Vl10)#exit
switch(config)#
```

- These commands configure a Switch Virtual Interface (SVI) and a virtual IP address with a subnet for VLAN 10. A static route is added to indicate that the virtual subnet is reachable through VLAN 10.

```
switch(config)#ip route 192.0.0.0/24 vlan 10
switch(config)#interface vlan 10
switch(config-if-Vl10)#ip address 10.0.0.2/24
switch(config-if-Vl10)#ip virtual-router address 192.0.0.6/24
switch(config-if-Vl10)#exit
switch(config)#
```

ip virtual-router mac-address

The **ip virtual-router mac-address** command assigns a virtual MAC address to the switch. The switch maps all virtual router IP addresses to this MAC address. The address is receive-only; the switch never sends packets with this address as the source. The virtual router is not configured on the switch until this virtual mac-address is assigned.

This command is typically used in MLAG configurations to create identical virtual routers on switches connected to the MLAG domain through an MLAG. When the destination MAC of a packet destined to a remote network matches the virtual MAC address, the MLAG peer forwards the traffic to the next hop destination. Each MLAG peer must have the same routes available, either though static configuration or learned through a dynamic routing protocol.

The **no ip virtual-router mac-address** command removes a virtual MAC address from the interface by deleting the corresponding **ip virtual-router mac-address** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
ip virtual-router mac-address mac_addr
no ip virtual-router mac address [mac_addr]
```

Parameters

- *mac_addr* MAC IP address (dotted hex notation). Select an address that will not otherwise appear on the switch.

Examples

- This command configures a virtual MAC address.

```
switch(config)#ip virtual-router mac-address 001c.7300.0099
switch(config)#
```

ip virtual-router mac-address advertisement-interval

The **ip virtual-router mac-address advertisement interval** command specifies the period between the transmission of consecutive gratuitous ARP requests that contain the virtual router mac address for each virtual-router IP address configured on the switch. The default period is 30 seconds.

The **no ip virtual-router mac-address advertisement-interval** command restores the default period of 30 seconds by removing the **ip virtual-router mac-address advertisement-interval** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
ip virtual-router mac-address advertisement-interval period
no ip virtual-router mac-address advertisement-interval
default ip virtual-router mac-address advertisement-interval
```

Parameters

- *period* advertisement interval (seconds). Values range from 0 to 86400. Default is 30.

Examples

- This command configures a MAC address advertisement interval of one minute (60 seconds).

```
switch(config)#ip virtual-router mac-address advertisement-interval 60
switch(config)#
```

ipv6 virtual-router address

The **ipv6 virtual-router address** command assigns a virtual IPv6 address to the VLAN interface being configured. (To assign a virtual IPv4 address to a VLAN interface, use the **ip virtual-router address** command.) Unlike VRRP, the virtual IP address does not have to be in the same subnet as the physical interface.

A maximum of 500 virtual IP addresses can be assigned to a VLAN interface. All virtual addresses on all VLAN interfaces resolve to the same virtual MAC address configured through the **ip virtual-router mac-address** command.

This command is typically used in MLAG configurations to create identical virtual routers on switches connected to the MLAG domain through an MLAG.

The **no ipv6 virtual-router address** and **default ipv6 virtual-router address** commands remove the specified virtual IPv6 address from the configuration mode interface by deleting the corresponding **ipv6 virtual-router address** command from ***running-config***. If the command does not specify an address, all virtual IPv6 addresses are removed from the interface.

Command Mode

Interface-VLAN Configuration

Command Syntax

```
ipv6 virtual-router address net_addr
no ipv6 virtual-router address [net_addr]
default ipv6 virtual-router address [net_addr]
```

Parameters

- *net_addr* network IPv6 address.

Examples

- These commands configure a Switch Virtual Interface (SVI) and a virtual IPv6 address for VLAN 10.

```
switch(config)#interface vlan 10
switch(config-if-Vl10)#ipv6 address 2001:0DB8:0:1::1/64
switch(config-if-Vl10)#ipv6 virtual-router address 2001:0DB8:0:1::2
switch(config-if-Vl10)#exit
switch(config)#
```

no vrrp

The **no vrrp** command removes all vrrp configuration commands for the specified virtual router on the configuration mode interface. The default vrrp command also reverts vrrp configuration parameters to default settings by removing the corresponding **vrrp** commands.

Commands removed by the **no vrrp** command include:

- **vrrp authentication**
- **vrrp description**
- **vrrp ip**
- **vrrp ip secondary**
- **vrrp preempt**
- **vrrp preempt delay**
- **vrrp priority**
- **vrrp shutdown**
- **vrrp timers advertise**

Command Mode

Interface-Ethernet Configuration
Interface-Port-Channel Configuration
Interface-VLAN Configuration

Command Syntax

```
no vrrp group
default vrrp group
```

Parameters

- *group* virtual router identifier (VRID). Values range from 1 to 255.

Examples

- This command removes all vrrp configuration commands for virtual router group 10 on VLAN 15.

```
switch(config)#interface vlan 15
switch(config-if-vl15)#no vrrp 10
switch(config-if-vl15)#
```

show ip virtual-router

The **show ip virtual-router** command displays the virtual MAC address assigned to the switch and all virtual IP addresses assigned to each VLAN interface.

Command Mode

EXEC

Command Syntax

```
show ip virtual-router
```

Messages

- **IP virtual router is not configured** a virtual MAC address is not assigned to the switch.
- **No interface with virtual IP address** no virtual IP addresses are assigned to any VLAN interfaces.

Examples

- This command displays a table of information for VRRP groups on the switch.

```
switch>show ip virtual-router
IP virtual router is configured with MAC address: 24cd.5a29.cc31
Interface  IP Address        Virtual IP Address   Status            Protocol
Vlan15     10.1.1.3/24       10.1.1.15            up                up
Vlan15     10.1.1.3/24       10.1.1.16            up                up
Vlan15     10.1.1.3/24       10.1.1.17            up                up
Vlan20     10.12.1.6/24      10.1.1.51            up                up
Vlan20     10.12.1.6/24      10.1.1.53            up                up
Vlan20     10.12.1.6/24      10.1.1.55            up                up
switch>
```

- This command generates a response that indicates a virtual MAC address is not assigned to the switch.

```
switch>show ip virtual-router
IP virtual router is not configured
switch>
```

show ipv6 virtual-router

The **show ipv6 virtual-router** command displays the virtual MAC address assigned to the switch and all virtual IPv6 addresses assigned to each VLAN interface.

Command Mode

EXEC

Command Syntax

show ipv6 virtual-router

Messages

- **IPv6 virtual router is not configured** a virtual MAC address is not assigned to the switch.
- **No interface with virtual IPv6 address** no virtual IPv6 addresses are assigned to any VLAN interfaces.

Examples

- This command displays a table of information for IPv6 VRRP groups on the switch.

```
switch>show ipv6 virtual-router
IP virtual router is configured with MAC address: 001c.7300.0099
MAC address advertisement interval: 30 seconds
Interface Vlan4094
  State is up
  Protocol is up
  IPv6 address
    2001:b8:2001::1011/64
  Virtual IPv6 address
    2001:db8:ac10:fe01::
switch>
```

show vrrp

The **show vrrp** command displays information about the Virtual Router Redundancy Protocol (VRRP) groups configured on a specified interface. Parameter options control the amount and formatting of the displayed information.

Command Mode

Privileged EXEC

Command Syntax

```
show vrrp [INFO_LEVEL] [STATES]
show vrrp INTF [GROUP_NUM] [INFO_LEVEL] [STATES]
show vrrp GROUP_NUM INTF_GROUP [INFO_LEVEL] [STATES]
```

Parameters

- ***INTF*** specifies the VRRP groups for which the command displays status. When the parameter is omitted or specifies only an interface, the group list is filtered by the ***STATES*** parameter.
  - <no parameter> specified groups on all interfaces.
  - **interface ethernet** *e_num* specified groups on Ethernet interface.
  - **interface loopback** *l_num* specified groups on loopback interface.
  - **interface management** *m_num* specified groups on management interface.
  - **interface port-channel** *p_num* specified groups on port channel interface.
  - **interface vlan** *v_num* specified groups on VLAN interface.
  - **interface vxlan** *vx_num* specified groups on VXLAN interface.
- ***GROUP_NUM*** the VRRP ID number of the group for which the command displays status.
  - <no parameter> all groups on specified interface.
  - *vrid_num* virtual router identifier (VRID). Value ranges from 1 to 255.
- ***INFO_LEVEL*** Specifies format and amount of displayed information. Options include:
  - <no parameter> displays a block of data for each VRRP group.
  - **brief** displays a single table that lists information for all VRRP groups.
- ***STATES*** Specifies the groups, by VRRP router state, that are displayed. Options include:
  - <no parameter> displays data for groups in the ***master*** or ***backup*** states.
  - **all** displays all groups, including groups in the ***stopped*** and ***interface down*** states.

Examples

- This command displays a table of information for VRRP groups on the switch.

```
switch>show vrrp brief
Interface Id  Ver Pri Time  State   VrIps
Vlan1006  3   2   100 3609  Master  127.38.10.2
Vlan1006  4   3   100 3609  Master  127.38.10.10
Vlan1010  1   2   100 3609  Master  128.44.5.3
Vlan1014  2   2   100 3609  Master  127.16.14.2
switch>
```

- This command displays data blocks for all VRRP groups on VLAN 46, regardless of the VRRP state.

```
switch>show vrrp interface vlan 1006 all
Vlan1010 - Group 1
  VRRP Version 2
  State is Stopped
  Virtual IPv4 address is 128.44.5.3
  Virtual MAC address is 0000.5e00.0101
  Mac Address Advertisement interval is 30s
  VRRP Advertisement interval is 1s
  Preemption is enabled
  Preemption delay is 0s
  Preemption reload delay is 0s
  Priority is 100
  Master Router is 0.0.0.0
  Master Advertisement interval is 1s
  Skew time is 0.609s
  Master Down interval is 3.609s
switch>
```

- This command displays data for all VRRP group 2 on VLAN 1014.

```
switch>show vrrp interface vlan 1014 group 2
Vlan1006 - Group 2
  VRRP Version 2
  State is Master
  Virtual IPv4 address is 127.38.10.2
  Virtual MAC address is 0000.5e00.0103
  Mac Address Advertisement interval is 30s
  VRRP Advertisement interval is 1s
  Preemption is enabled
  Preemption delay is 0s
  Preemption reload delay is 0s
  Priority is 100
  Master Router is 127.38.10.1 (local), priority is 100
  Master Advertisement interval is 1s
  Skew time is 0.609s
  Master Down interval is 3.609s
switch>
```

show vrrp internal

The **show vrrp internal** command displays the internal PAM packet counters on the switch.

Command Mode

EXEC

Command Syntax

show vrrp internal

Examples

- This command displays the internal Packet Access Method(PAM) packet counters on the switch.

```
switch>show vrrp internal
VRRP PAM Counters
----------------
ARP Responder:
         numSent : 0
         numRcvd : 0
      numBadRcvd : 0
ND Responder:
         numSent : 0
         numRcvd : 0
      numBadRcvd : 0
IPv4 VRRP Packet Manager:
         numSent : 0
         numRcvd : 0
      numBadRcvd : 0
IPv6 VRRP Packet Manager:
         numSent : 0
         numRcvd : 0
      numBadRcvd : 0
switch>
```

vrrp authentication

The **vrrp authentication** command configures parameters the switch uses to authenticate virtual router packets it receives from other VRRP routers in the group. This feature is only supported for VRRP IPv4.

The **no vrrp authentication** and **default vrrp authentication** commands disable VRRP authentication of packets from the specified virtual router by removing the corresponding **vrrp authentication** command from ***running-config***. The **no vrrp** command also removes the **vrrp authentication** command for the specified virtual router.

Command Mode

Interface-Ethernet Configuration
Interface-Port-Channel Configuration
Interface-VLAN Configuration

Command Syntax

```
vrrp group authentication AUTH_PARAMETER
no vrrp group authentication
default vrrp group authentication
```

Parameters

- *group* virtual router identifier (VRID). Values range from 1 to 255.
- ***AUTH_PARAMETER*** encryption level and authentication key used by router. Options include:
  - **text** *text_key* plain-text authentication, *text_key* is text.
  - *text_key* plain-text authentication, *text_key* is text.
  - **ietf-md5 key-string 0** *text_key* IP authentication of MD5 key hash, *text_key* is text.
  - **ietf-md5 key-string** *text_key* IP authentication of MD5 key hash, *text_key* is text.
  - **ietf-md5 key-string 7** *coded_key* IP authentication of MD5 key hash, *coded_key* is MD5 hash.

Examples

- This command implements plain-text authentication, using 12345 as the key, for virtual router 40 on VLAN 100.

```
switch(config)#interface vlan 100
switch(config-if-vl100)#vrrp 40 authentication text 12345
switch(config-if-vl100)#
```

- This command implements ietf-md5 authentication, using 12345 as the key.

```
switch(config-if-vl100)#vrrp 40 authentication ietf-md5 key-string 0 12345
switch(config-if-vl100)#
```

- This command implements ietf-md5 authentication, using 12345 as the key. The key is entered as the MD5 hash equivalent of the text string.

```
switch(config-if-vl100)#vrrp 40 authentication ietf-md5 key-string 7
EA3TUPxdddFCLYT8mb+kxw==
switch(config-if-vl100)#
```

vrrp delay reload

The **vrrp delay reload** command delays the time for VRRP initialization after a system reboot.

The **no vrrp delay reload** and **default vrrp delay reload** commands restore the default value of 0 by deleting the **vrrp delay reload** statement from ***running-config***.

Command Mode

Interface-Ethernet Configuration
Interface-Port-Channel Configuration
Interface-VLAN Configuration

Command Syntax

```
vrrp group delay reload [INTERVAL]
no vrrp group delay reload
default vrrp group delay reload
```

Parameters

- ***INTERVAL*** The number of seconds for the delay (seconds). Options include:
  - <no parameter> Default value of 0 seconds.
  - <***0*** to ***3600***> Ranges between 0 and 60 minutes.

Example

- These commands configure the VRRP reload delay interval to 15 minutes.

```
switch(config)#interface vlan 100
switch(config-if-Vl100)#vrrp 2 delay reload 900
switch(config-if-Vl100)#
```

- These commands removes the VRRP reload delay interval .

```
switch(config)#interface vlan 100
switch(config-if-Vl100)#no vrrp 2 delay reload
switch(config-if-Vl100)#
```

vrrp description

The **vrrp description** command associates a text string to a VRRP virtual router on the configuration mode interface. The string has no functional impact on the virtual router. The maximum length of the string is 80 characters.

The **no vrrp description** and **default vrrp description** commands remove the text string association from the VRRP virtual router by deleting the corresponding **vrrp description** command from ***running-config***. The **no vrrp** command also removes the **vrrp description** command for the specified virtual router.

Command Mode

Interface-Ethernet Configuration
Interface-Port-Channel Configuration
Interface-VLAN Configuration

Command Syntax

```
vrrp group description label_text
no vrrp group description
default vrrp group description
```

Parameters

- *group* virtual router identifier (VRID). Values range from 1 to 255.
- *label_text* text that describes the virtual router. Maximum string length is 80 characters.

Example

- This command associates the text string ***Laboratory Router*** to virtual router 15 on VLAN 20.

```
switch(config)#interface vlan 20
switch(config-if-vl20)#vrrp 15 description Laboratory Router
switch(config-if-vl20)#
```

vrrp ip

The **vrrp ip** command configures the primary IP address for the specified VRRP virtual router. The command also activates the virtual router if the primary address is contained in the interface's subnet. A VRRP virtual router's configuration may contain only one primary IP address assignment command; subsequent **vrrp ip** commands replace the existing primary address assignment.

The **vrrp ip secondary** command assigns a secondary IP address to the VRRP virtual router.

The **no vrrp ip** and **default vrrp ip** commands disable the VRRP virtual router and deletes the primary IP address by removing the corresponding **vrrp ip** statement from ***running-config.*** The **no vrrp** command also removes the **vrrp ip** command for the specified virtual router.

Command Mode

Interface-Ethernet Configuration
Interface-Port-Channel Configuration
Interface-VLAN Configuration

Command Syntax

```
vrrp group ip ipv4_address
no vrrp group ip ipv4_address
default vrrp group ip ipv4_address
```

Parameters

- *group* virtual router identifier (VRID). Values range from 1 to 255.
- *ipv4_address* IPv4 address of the virtual router.

Related Commands

- **vrrp ip secondary**

Example

- This command enables virtual router 15 on VLAN 20 and designates 10.1.1.5 as the virtual router's primary address.

```
switch(config)#interface vlan 20
switch(config-if-vl20)#vrrp 15 ip 10.1.1.5
switch(config-if-vl20)#
```

vrrp ip secondary

The **vrrp ip secondary** command assigns a secondary IP address to the specified virtual router. Secondary IP addresses are an optional virtual router parameter. A virtual router may contain multiple secondary address commands. The IP address list must be identical for all VRRP routers in a virtual router group.

The virtual router is assigned a primary IP address with the **vrrp ip** command.

The **no vrrp ip secondary** and **default vrrp ip secondary** commands remove the secondary IP address for the specified VRRP virtual router by deleting the corresponding **vrrp ip secondary** statement from ***running-config.*** The **no vrrp** command also removes all **vrrp secondary** commands for the specified virtual router.

Command Mode

Interface-Ethernet Configuration
Interface-Port-Channel Configuration
Interface-VLAN Configuration

Command Syntax

```
vrrp group ip ipv4_addr secondary
no vrrp group ip ipv4_addr secondary
default vrrp group ip ipv4_addr secondary
```

Parameters

- *group* virtual router identifier (VRID). Values range from 1 to 255.
- *ipv4_addr* secondary IPv4 address of the virtual router.

Related Commands

- **vrrp ip**

Example

- This command assigns the IP address of 10.2.4.5 as the secondary IP address for the virtual router with VRID of 15 on VLAN 20

```
switch(config)#interface vlan 20
switch(config-if-vl20)#vrrp 15 ip 10.2.4.5 secondary
switch(config-if-vl20)#
```

vrrp ip version

The **vrrp ip version** command enables VRRP on the configuration mode interface and configures the VRRP version for the specified VRRP virtual router.

The **no vrrp ip version** and **default vrrp ip version** commands restore the default VRRP version to VRRPv2 by removing the corresponding **vrrp ip version** statement from ***running-config.***

Command Mode

Interface-Ethernet Configuration
Interface-Port-Channel Configuration
Interface-VLAN Configuration

Command Syntax

```
vrrp group ip version VERSION_NUMBER
no vrrp group ip version
default vrrp group ip version
```

Parameters

- *group* virtual router identifier (VRID). Values range from 1 to 255.
- ***VERSION_NUMBER*** Specifies VRRP version that the switch uses. Default value is 2 (VRRPv2) Options include:
  - **2** VRRP v2 supports IPv4 environment.
  - **3** VRRP v3 supports IPv4 and IPv6 environment.

Example

- This command enables VRRPv3 for IPv6 on interface Ethernet 3.

```
switch#(config)#interface ethernet 3
switch#(config-if-Et3)# vrrp 1 ip version 3
switch#
```

- This command removes VRRPv3 from interface Ethernet 3 and reverts to the default VRRPv2.

```
switch#(config)#interface ethernet 3
switch#(config-if-Et3)# no vrrp 1 ip version
switch#(config-if-Et3)#
```

vrrp ipv6

The **vrrp ipv6** command configures the IPv6 address for the specified VRRP virtual router. The command also activates the virtual router if the primary address is contained in the interface's subnet.

The **no vrrp ipv6** and **default vrrp ipv6** commands disable the VRRP virtual router and deletes the IPv6 address by removing the corresponding **vrrp ipv6** statement from ***running-config.*** The **no vrrp** command also removes the **vrrp ipv6** command for the specified virtual router.

Command Mode

Interface-Ethernet Configuration
Interface-Port-Channel Configuration
Interface-VLAN Configuration

Command Syntax

```
vrrp group ip ipv6_address
no vrrp group ip ipv6_address
default vrrp group ip ipv6_address
```

Parameters

- *group* virtual router identifier (VRID). Values range from 1 to 255.
- *ipv6_address* IPv6 address of the virtual router.

Example

- This command enables addresss 2001:db8:0:1::1 for IPv6 VRRP on Vlan 20.

```
switch(config)#interface vlan 20
switch(config-if-vl20)#vrrp 3 ipv6 2001:db8:0:1::1
switch(config-if-vl20)#
```

- This command disables VRRPv3 on Vlan 20 from virtual router 3.

```
switch(config)#interface vlan 20
switch(config-if-vl20)#no vrrp 3 ipv6 2001:db8:0:1::1
switch(config-if-vl20)#
```

vrrp mac-address advertisement-interval

The **vrrp mac-address advertisement-interval** command specifies the interval between advertisement packets sent by the master router to the VRRP group members.

The **vrrp mac-address advertisement-interval 0, no vrrp mac-address advertisement-interval** and **default vrrp mac-address advertisement-interval** commands disable the feature by removing the **vrrp mac-address advertisement-interval** command from ***running-config.***

Command Mode

Interface-Ethernet Configuration
Interface-Port-Channel Configuration
Interface-VLAN Configuration

Command Syntax

vrrp *group* **mac-address advertisement-interval** *period*
no vrrp *group* **mac-address**
default vrrp *group* **mac-address**

Parameters

- *group* virtual router identifier (VRID). Values range from 1 to 255.
- *period* interval in which the master router sends advertisement packets (seconds). Value ranges from 0 to 3600. Selecting 0 as the interval disables this feature.

Example

- This command specifies the interval between advertisement packets sent to the members of VRRP group 3 on VLAN 20.

```
switch(config)#interface vlan 20
switch(config-if-vl20)#vrrp 3 mac-address advertisement-interval 60
switch(config-if-vl20)#
```

- This command disables the feature on VLAN 20.

```
switch(config)#interface vlan 20
switch(config-if-vl20)#no vrrp 3 mac-address advertisement-interval
switch(config-if-vl20)#
```

vrrp preempt

The **vrrp preempt** command controls a virtual router's preempt mode setting. When preempt mode is enabled, if the switch has a higher priority it will preempt the current master virtual router. When preempt mode is disabled, the switch can become the master virtual router only when a master virtual router is not present on the subnet, regardless of VRRP priority settings. By default, preempt mode is enabled.

The **no vrrp preempt** and **default vrrp preempt** commands disable preempt mode for the specified virtual router; the **default vrrp prempt** command stores a corresponding **no vrrp preempt** statement in ***running-config***. The **vrrp preempt** command enables preempt mode by removing the corresponding **no vrrp preempt** statement from ***running-config***.

The **no vrrp** command also enables preempt mode by removing the **no vrrp preempt** command for the specified virtual router.

Command Mode

Interface-Ethernet Configuration
Interface-Port-Channel Configuration
Interface-VLAN Configuration

Command Syntax

```
vrrp group preempt
no vrrp group preempt
default vrrp group preempt
```

Parameters

- *group* virtual router identifier (VRID). Values range from 1 to 255.

Related Commands

- **vrrp preempt delay**

Examples

- This command disables preempt mode for virtual router 20 on VLAN 40.

```
switch(config)#interface vlan 40
switch(config-if-vl40)#no vrrp 20 preempt
switch(config-if-vl40)#
```

- This command enables preempt mode for virtual router 20 on VLAN 40.

```
switch(config-if-vl40)#vrrp 20 preempt
switch(config-if-vl40)#
```

vrrp preempt delay

The **vrrp preempt delay** command specifies the interval between a VRRP preemption event and the point when the switch becomes the master vrrp router. A preemption event is any event that results in the switch having the highest virtual router priority setting while preemption is enabled. The **vrrp preempt** command enables preemption for a specified virtual router.

The command configures two delay periods:

- **minimum** time delays master vrrp takeover when VRRP is fully implemented.
- **reload** time delays master vrrp takeover after VRRP is initialized following a switch reload (boot). The switch bypasses the reload time to become the VRRP master immediately if it senses there are no other active switches in the virtual router group.

Running-config maintains separate delay statements for **minimum** and **reload** parameters. Commands may list both parameters. Commands that list one parameter do not affect the omitted parameter. Values range from 0 to 3600 seconds (one hour). The default delay is zero seconds for both parameters.

The **no vrrp preempt delay** and **default vrrp preempt delay** commands reset the specified delay to the default of zero seconds. Commands that do no list either parameter resets both periods to zero. The **no vrrp** command also removes all **vrrp preempt delay** commands for the specified virtual router.

Command Mode

Interface-Ethernet Configuration
Interface-Port-Channel Configuration
Interface-VLAN Configuration

Command Syntax

```
vrrp group preempt delay [MINIMUM_INTERVAL] [RELOAD_INTERVAL]
no vrrp group preempt delay [DELAY_TYPE]
default vrrp group preempt delay [DELAY_TYPE]
```

Parameters

- *group* virtual router identifier (VRID). Values range from 1 to 255.
- ***MINIMUM_INTERVAL*** period between preempt event and takeover of master vrrp router role.
  - <no parameter> minimum delay is not altered by command.
  - **minimum** *min_time* delay during normal operation (seconds). Values range from 0 to 3600.
- ***RELOAD_INTERVAL*** period after reboot-VRRP initialization and takeover of master vrrp router role.
  - <no parameter> reload delay is not altered by command.
  - **reload** *reload_time* delay after reboot (seconds). Values range from 0 to 3600.
- ***DELAY_TYPE*** delay type that is reset to default value.
  - <no parameter> reload and minimum delays are reset to default.
  - **minimum** minimum delay is reset to default.
  - **reload** reload delay is reset to default.

  (***DELAY_TYPE*** parameter is only used in **no vrrp preempt delay** and **default vrrp preempt delay** commands).

Related Commands

- **vrrp preempt**

Examples

- This command sets the minimum preempt time of 90 seconds for virtual router 20 on VLAN 40.

```
switch(config)#interface vlan 40
switch(config-if-vl40)#vrrp 20 preempt delay minimum 90
switch(config-if-vl40)#
```

- This command sets the minimum and reload preempt time to zero for virtual router 20 on VLAN 40.

```
switch(config-if-vl40)#no vrrp 20 preempt delay
switch(config-if-vl40)#
```

vrrp priority

The **vrrp priority** command configures the switch's priority setting for a VRRP virtual router. Priority values range from 1 to 254. The default value is 100.

The router with the highest vrrp priority setting for a group becomes the master virtual router for that group. The master virtual router controls the IP address and is responsible for forwarding traffic sent. The **vrrp preempt** command controls the time when a switch can become the master virtual router.

The **no vrrp priority** and **default vrrp priority** commands restore the default priority of 100 to the virtual router on the configuration mode interface by removing the corresponding **vrrp priority** command from ***running-config***. The **no vrrp** command also removes the **vrrp priority** command for the specified virtual router.

Command Mode

Interface-Ethernet Configuration
Interface-Port-Channel Configuration
Interface-VLAN Configuration

Command Syntax

```
vrrp group priority level
no vrrp group priority
default vrrp group priority
```

Parameters

- *group* virtual router identifier (VRID). Values range from 1 to 255.
- *level* priority setting for the specified virtual router. Values range from 1 to 254.

Examples

- This command sets the virtual router priority value of 250 for virtual router group 45 on VLAN 20.

```
switch(config)#interface vlan 20
switch(config-if-vl20)#vrrp 45 priority 250
switch(config-if-vl20)#
```

vrrp shutdown

The **vrrp shutdown** command places the switch in stopped state for the specified virtual router. While in stopped state, the switch cannot act as a Master or backup router for the virtual router group.

The **no vrrp shutdown** and **default vrrp shutdown** commands remove the corresponding **vrrp shutdown** command from ***running-config***. This changes the switch's virtual router state to ***backup*** or ***master*** if the virtual router is properly configured.

Command Mode

Interface-Ethernet Configuration
Interface-Port-Channel Configuration
Interface-VLAN Configuration

Command Syntax

```
vrrp group shutdown
no vrrp group shutdown
default vrrp group shutdown
```

Parameters

- *group* virtual router identifier (VRID). Values range from 1 to 255.

Example

- These commands place the switch in stopped mode for virtual router 24 on VLAN 20.

```
switch(config)#interface vlan 20
switch(config-if-vl20)#vrrp 24 shutdown
switch(config-if-vl20)#
```

- This command moves the switch out of stopped mode for virtual router 24 on VLAN 20.

```
switch(config-if-vl20)#no vrrp 24 shutdown
switch(config-if-vl20)#
```

vrrp timers advertise

The **vrrp timers advertise** command configures the interval between successive advertisement messages that the switch sends to VRRP routers in the specified virtual router group. The switch must be the group's Master virtual router to send advertisement messages. The advertisement interval must be configured identically on all physical routers in the virtual router group.

The advertisement interval also influences the timeout interval that defines when the virtual router becomes the master virtual router. When preemption is enabled, the virtual router becomes the master when three times the advertisement interval elapses after the switch detects master router priority conditions.

The **no vrrp timers advertise** and **default vrrp timers advertise** commands restore the default advertisement interval of one second for the specified virtual router by removing the corresponding **vrrp timers advertise** command from ***running-config***. The **no vrrp** command also removes the **vrrp timers advertise** command for the specified virtual router.

Command Mode

```
Interface-Ethernet Configuration
Interface-Port-Channel Configuration
Interface-VLAN Configuration
```

Command Syntax

```
vrrp group timers advertise adv_time
no vrrp group timers advertise
default vrrp group timers advertise
```

Parameters

- *group* virtual router identifier (VRID). Values range from 1 to 255.
- *adv_time* advertisement interval (seconds). Values range from 1 to 255. Default value is 1.

Example

- This command sets the advertisement interval of five seconds for the virtual router 35 on VLAN 100.

```
switch(config)#interface vlan 100
switch(config-if-vl100)#vrrp 35 timers advertise 5
switch(config-if-vl100)#
```

vrrp track

The **vrrp track** command configures the VRRP client process on the configuration mode interface to track the specified tracked object and react when its status changes to **down**. The tracked object is created by the **vrrp track** command.

The **no vrrp track** and **default vrrp track** commands cause the VRRP client process to stop tracking the specified tracked object by removing the corresponding **vrrp track** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration
Interface-Port-Channel Configuration
Interface-VLAN Configuration

Command Syntax

```
vrrp group track object_name ACTION amount
no vrrp group track object_name ACTION
default vrrp group track object_name ACTION
```

Parameters

- *group* virtual router identifier (VRID). Values range from 1 to 255.
- *object_name* name of tracked object.
- *amount* amount to decrement VRRP priority. Values range from 1 to 254.
- ***ACTION*** The action that VRRP is to take when the tracked object's status changes to **down**. Options include:
  - **decrement** decrease VRRP priority by *amount*.
  - **shutdown** shut down VRRP on the configuration mode interface.

  If both **decrement** and **shutdown** are configured on the same interface for the same VRRP group, then VRRP will be shut down on the interface if the tracked object is down.

Related Commands

- **track**

Example

- This command causes Ethernet interface 5 to disable VRRP when tracked object ETH8 changes state.

```
switch(config-if-Et5)#vrrp 1 track ETH8 shutdown
switch(config-if-Et5)#
```

Chapter 22

Spanning Tree Protocol

Spanning Tree Protocols prevent bridging loops in Layer 2 Ethernet networks. Arista switches support Rapid Spanning Tree, Multiple Spanning Tree, and Rapid-Per VLAN Spanning Tree protocols.

These sections describe the Arista Spanning Tree Protocol implementation.

- Section 22.1: Introduction to Spanning Tree Protocols
- Section 22.2: Spanning Tree Overview
- Section 22.3: Configuring a Spanning Tree
- Section 22.4: STP Commands

22.1 Introduction to Spanning Tree Protocols

Arista Switches support the leading spanning tree protocols: RSTP, MST and Rapid-PVST. This variety of options simplifies integration into existing networks without compromising network reliability, scalability or performance.

22.2 Spanning Tree Overview

An Ethernet network functions properly when only one active path exists between any two stations. A spanning tree is a loop-free subset of a network topology. STP is a L2 network protocol that ensures a loop-free topology for any bridged Ethernet LAN. STP allows a network to include spare links as automatic backup paths that are available when an active link fails without creating loops or requiring manual intervention. The original STP is standardized as IEEE 802.1D.

Several variations to the original STP improve performance and add capacity. Arista switches support these STP versions:

- Rapid Spanning Tree (RSTP)
- Multiple Spanning Tree (MSTP)
- Rapid Per-VLAN Spanning Tree (Rapid-PVST)

The Overview contains the following sections:

- Section 22.2.1: Spanning Tree Protocol Versions
- Section 22.2.2: Structure of a Spanning Tree Instance
- Section 22.2.3: BPDUs

22.2.1 Spanning Tree Protocol Versions

STP versions supported by Arista switches address two limitations of the original Spanning Tree protocol that was standardized as IEEE 802.1D:

- Slow convergence to the new spanning tree topology after a network change
- The entire network is covered by one spanning tree instance.

The following sections describe the supported STP versions, compatibility issues in networks containing switches running different STP versions, and supported alternatives to spanning tree.

22.2.1.1 Rapid Spanning Tree Protocol (RSTP)

RSTP is specified in 802.1w and supersedes STP. RSTP provides rapid convergence after network topology changes. RSTP provides a single spanning tree instance for the entire network, similar to STP. Standard 802.1D-2004 incorporates RSTP and obsoletes STP.

The RSTP instance is the base unit of MST and Rapid-PVST spanning trees.

22.2.1.2 Rapid Per-VLAN Spanning Tree Protocol (Rapid-PVST)

Rapid Per-VLAN Spanning Tree (PVST) extends the original STP to support a spanning tree instance on each VLAN in the network. The quantity of PVST instances in a network equals the number of configured VLANs, up to a maximum of 4094 instances. PVST can load balance layer-2 traffic without creating a loop because it handles each VLAN as a separate network. However, PVST does not address slow network convergence after a network topology change.

Arista switches support Rapid-PVST, which is a variation of PVST based on RSTP instances. Rapid-PVST provides rapid connectivity recovery after the failure of a bridge, port, or LAN. Rapid-PVST can be enabled or disabled on individual VLANs.

22.2.1.3 Multiple Spanning Tree Protocol (MSTP)

MST extends rapid spanning tree protocol (RSTP) to support multiple spanning tree instances on a network, but is still compatible with RSTP. By default, Arista switches use MSTP.

MST supports multiple spanning tree instances, similar to Rapid PVST. However, MST associates an instance with multiple VLANs. This architecture supports load balancing by providing multiple forwarding paths for data traffic. Network fault tolerance is improved because failures in one instance do not affect other instances.

MST Regions

An *MST region* is a group of connected switches with identical MST configuration. Each region can support a maximum of 65 spanning-tree instances. MST regions are identified by a version number, name, and VLAN-to-instance map; these parameters must be configured identically on all switches in the region. Only MST region members participate with the MST instances defined in the region. A VLAN can only be assigned to one spanning-tree instance at a time. MST does not specify the maximum number of regions that a network can contain.

MST Instances

Each MST instance is identified by an instance number that ranges from 0 to 4094 and is associated with a set of VLANs. An MST region contains two types of spanning tree instances: an internal spanning tree instance (IST) and multiple spanning tree instances (MSTI).

- The *Internal Spanning Tree Instance* (IST) is the default spanning tree instance in an MST region and is always instance 0. It gives the root switch for the region and contains all VLANs configured on the switch that are not assigned to a MST instance.

- *Multiple Spanning Tree instances* (MSTIs) consist of VLANs that are assigned through MST configuration statements. VLANs assigned to an MSTI are removed from the IST instance. VLANs in an MSTI operate as a part of a single Spanning Tree topology. Because each VLAN can belong to only one instance, MST instances (and the IST) are topologically independent.

22.2.1.4 Version Interoperability

A network can contain switches running different spanning tree versions. The common spanning tree (CST) is a single forwarding path the switch calculates for STP, RSTP, MSTP, and Rapid-PVST topologies in networks containing multiple spanning tree variations.

In multi-instance topologies, the following instances correspond to the CST:

- **Rapid-PVST**: VLAN 1
- **MST**: IST (instance 0)

RSTP and MSTP are compatible with other spanning tree versions:

- An RSTP bridge sends 802.1D (original STP) BPDUs on ports connected to an STP bridge.
- RSTP bridges operating in 802.1D mode remain in 802.1D mode even after all STP bridges are removed from their links.
- An MST bridge detects a port is at a region boundary when it receives an STP BPDU or an MST BPDU from a different region.
- MST ports assume they are boundary ports when the bridges to which they connect join the same region.

The **clear spanning-tree detected-protocols** command forces MST ports to renegotiate with their neighbors.

22.2.1.5 Switchport Interface Pairs

Switchport interface pairs associate two interfaces in a primary-backup configuration. When the primary interface is functioning, the backup interface remains dormant in standby mode. When the primary interface stops functioning, the backup interface handles the traffic.

An alternative implementation balances traffic between the primary and backup interfaces. If either interface shuts down, the other handles traffic addressed to the pair.

The following guidelines apply to switchport interface pairs.

- Ethernet and Port Channels can be primary interfaces.
- Ethernet, Port Channel, Management, Loopback, and VLAN interfaces can be backup interfaces.
- The primary and backup interfaces can be different interface types.
- Interface pairs should be similarly configured to ensure consistent behavior.
- An interface can be associated with a maximum of one backup interface.
- An interface can back up a maximum of one interface.
- Any Ethernet interface configured in an interface pair cannot be a port channel member.
- STP is disabled on ports configured as primary or backup interfaces.
- Static MAC addresses should be configured after primary-backup pairs are established.

22.2.1.6 Disabling Spanning Tree

When spanning tree is disabled and switchport interface pairs are not configured, all interfaces forward packets as specified by their configuration. STP packets are not generated and inbound STP packets are forwarded on the VLAN where they are received as normal multicast data packets.

Important! Disabling all Spanning Tree Protocols on the switch is strongly discouraged.

22.2.2 Structure of a Spanning Tree Instance

A layer 2 network consists of bridges and network segments. A loop exists when multiple active paths connect two components. Spanning tree protocols allow only one active path between any two network components. Loops are removed by blocking selected ports that connect bridges to network segments.

Ports are assigned cost values that reflect their transmission speed and any other criteria selected by the administrator. Ports with faster transmission speeds and other desirable characteristics are assigned lower costs. High cost ports are blocked in deference to lower cost ports.

A network topology defines multiple possible spanning trees. Network bridges collectively compute and implement one spanning tree to maintain connectivity between all network components while blocking ports that could result in loops. Administrators improve network performance by adjusting parameter settings to select the most efficient spanning tree.

Spanning tree bridges continuously transmit topology information to notify all other bridges on the network when topology changes are required, such as when a link fails. Bridge Protocol Data Units (BPDUs) are STP information packets that bridges exchange.

The following sections describe spanning tree configuration parameters.

22.2.2.1 Root and Designated Bridges

The **root bridge** is the center of the STP topology. A spanning tree instance has one root bridge. Spanning tree bases path calculations on each network component's distance from the root bridge.

All other network bridges calculate paths to the root bridge when selecting spanning tree links. STP calculates the distance to the root bridge to build a loop-free topology that features the shortest distance between devices among all possible paths.

Each switch is assigned a unique bridge ID number for each instance. All network switches collectively elect the root bridge by comparing bridge IDs. The root bridge is the switch with the lowest bridge ID.

The bridge ID contains the following eight bytes, in order of decreasing significance:

- Port priority (four bits)
- Instance number (12 bits): VLAN number (Rapid-PVST); instance number (MST); 0 (RST)
- MAC address of switch (six bytes)

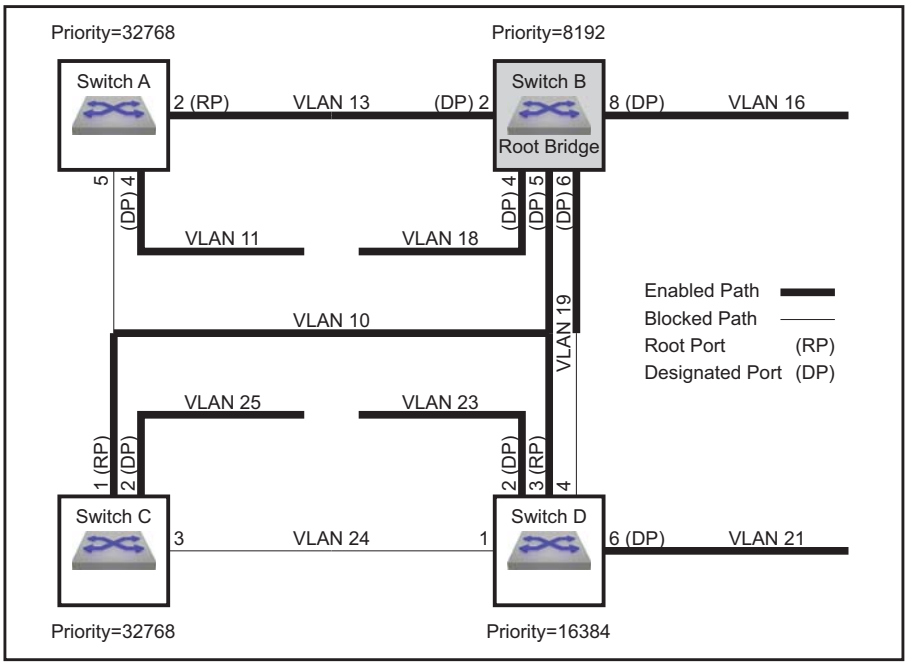
A **designated bridge** is defined for each network segment as the switch that provides the segment's shortest path to the root bridge. A designated bridge is selected for each segment after a root bridge is selected; a switch can be a designated bridge for multiple segments.

The following network calculations in Figure 22-1 assume that each path has the same cost:

- Switch B is the root bridge – its bridge ID is lowest because it has the smallest port priority.
- Switch A is the designated bridge for VLAN 11.
- Switch B is the designated bridge for VLAN 10, VLAN 13, VLAN 16, VLAN 18, VLAN 19.
- Switch C is the designated bridge for VLAN 25.

- Switch D is the designated bridge for VLAN 21, VLAN 23.

Figure 22-1: Spanning Tree Network Example



22.2.2.2 Port Roles

Messages from connected devices to the root bridge traverse a least-cost path, which has the smallest cost among all possible paths to the root bridge. The cost of a path is the sum of the costs of all path segments, as defined through port cost settings.

Active ports in a least cost-path fulfill one of two possible roles: root port and designated port. STP blocks all other network ports. STP also defines alternate and backup ports to handle traffic when an active port is inaccessible.

- **Root port (RP) accesses the bridge's least-cost path to the root bridge.** Each bridge selects its root port after calculating the cost of each possible path to the root bridge.

  The following ports in Figure 22-1 are root ports:

  - **Switch A:** port 2
  - **Switch C**: port 1
  - **Switch D**: port 3
- **Designated port (DP) accesses a network segment's designated bridge.** Each segment defines one DP. Switches can provide DPs for multiple segments. All ports on the root bridge are DPs.

  The following ports in Figure 22-1 are designated ports:

  - **Switch A:** port 4 (VLAN 11)
  - **Switch B:** port 2 (VLAN 13), port 4 (VLAN 18), port 5 (VLAN 10), port 6 (VLAN 19), port 8 (VLAN 16)
  - **Switch C:** port 2 (VLAN 25)
  - **Switch D:** port 2 (VLAN 23), port 6 (VLAN 21)

- **Alternate ports** provide backup paths from their bridges to the root bridge. An alternate port is blocked until a network change transforms it into a root port.
- **Backup ports** provide alternative paths from VLANs to their designated bridges. A backup port is blocked until a network change transforms it into a designated port.

22.2.2.3 Port Activity States

A port's activity state defines its current STP activity level. STP monitors BPDUs for network changes that require an activity state transition.

STP defines three port activity states:

- **Forwarding:** The port receives and sends data. Root ports and designated ports are either in, or transitioning to, this state.
- **Discarding:** The port does not receive or send data. Blocked ports receive BPDU packets. All ports except RPs and DPs are blocked, including alternate and backup ports.
- **Learning:** The transitional post-discarding state where the port prepares to forward frames by adding source addresses from inbound data packets to the switching database.

22.2.2.4 Port Types

Port type is a configurable parameter that reflects the type of network segment that is connected to the port. Proper port type configuration results in rapid convergence after network topology changes. RSTP port types include normal, network, and edge ports. ***Normal*** is the default port type.

- **Normal** ports have an unspecified topology.
- **Network** ports connect only to switches or bridges.

  RSTP immediately transitions network ports to the discarding state.
- **Edge** ports connect directly to end stations.

  Edge ports transition directly to forwarding state because they do not create loops. An edge port becomes a normal port when it receives a BPDU.

22.2.2.5 Link Types

Link type is a configurable parameter that determines candidates for RSTP fast state transition.

- the default link type for full-duplex ports is ***point-to-point***.
- the default link type for half-duplex ports is ***shared***.

Fast state transitions are allowed on point-to-point links that connect bridges. Fast state transitions are not allowed on shared ports regardless of the duplex setting.

22.2.3 BPDUs

Spanning tree rules specify a root bridge, select designated bridges, and assign roles to ports. STP rule implementation requires that network topology information is available to each switch. Switches exchange topology information through bridge protocol data units (BPDUs). Information provided by BPDU packets include bridge IDs and root path costs.

22.2.3.1 BPDU Types

STP defines three BPDU types:

- Configuration BPDU (CBPDU), used for computing the spanning tree.

- Topology Change Notification (TCN) BPDU, announces network topology changes.
- Topology Change Notification Acknowledgment (TCA), acknowledges topology changes.

Bridges enter the following addresses in outbound BPDU frames:

- source address: outbound port's MAC address.
- destination address: STP multicast address 01:80:C2:00:00:00.

Bridges regularly exchange BPDUs to track network changes that trigger STP recomputations and port activity state transitions. The *hello timer* specifies the period between consecutive BPDU messages; the default is two seconds.

22.2.3.2 Bridge Timers

Bridge timers specify parameter values that the switch includes in BPDU packets that it sends as a root bridge. Bridge timers include:

- **hello-time**: transmission interval between consecutive BPDU packets.
- **forward-time**: the period that ports remain in learning state.
- **max-age**: the period that BPDU data remains valid after it is received.
- **max-hop**: the number of bridges in an MST region that a BPDU can traverse before it is discarded.

The switch recomputes the spanning tree topology if it does not receive another BPDU before the max-age timer expires. When ***edge*** ports and ***point-to-point*** links are properly configured, RSTP network convergence does not require forward-delay and max-age timers.

22.2.3.3 MSTP BPDUs

MSTP BPDUs are targeted at a single instance and provide STP information for the entire region. MSTP encodes a standard BPDU for the IST, then adds region information and MST instance messages for all configured instances, where each message conveys spanning tree data for an instance. Frames assigned to VLANs operate in the instance to which the VLAN is assigned. Bridges enter an MD5 digest of the VLAN-to-instance map table in BPDUs to avoid including the entire table in each BPDU. Recipients use this digest and other administratively configured values to identify bridges in the same MST region.

MSTP BPDUs are compatible with RSTP. RSTP bridges view an MST region as a single-hop RSTP bridge regardless of the number of bridges inside the region because:

- RSTP bridges interpret MSTP BPDUs as RSTP BPDUs.
- RSTP bridges increment the ***message age timer*** only once while data flows through an MST region; MSTP measures ***time to live*** with a **remaining hops** variable, instead of the **message age timer**.

Ports at the edge of an MST region connecting to a bridge (RSTP or STP) or to an endpoint are *boundary ports*.

22.3 Configuring a Spanning Tree

These sections describe the following configuration processes:

- Section 22.3.1: Version Configuration and Instance Creation
- Section 22.3.2: Spanning Tree Instance Configuration
- Section 22.3.3: Port Roles and Rapid Convergence
- Section 22.3.4: Configuring BPDU Transmissions

22.3.1 Version Configuration and Instance Creation

The switch supports three STP versions and switchport backup interface pairs. Disabling spanning tree is also supported but not recommended.

The **spanning-tree mode** global configuration command specifies the spanning tree version the switch runs. This section describes command options that enable and configure STP versions.

22.3.1.1 Multiple Spanning Tree (MST)

Multiple Spanning Tree is enabled by the **spanning-tree mode** command with the ***mstp*** option. MSTP is the default STP version.

Example

- This command enables Multiple Spanning Tree.

```
switch(config)#spanning-tree mode mstp
switch(config)#
```

Configuring MST Regions

All switches in an MST region must have the same name, revision, and VLAN-to-instance map. MST configuration mode commands sets the region parameters. MST configuration mode is a group-change mode where changes are saved by exiting the mode.

Example

- The **spanning-tree mst configuration** command places the switch in MST configuration mode.

```
switch(config)#spanning-tree mst configuration
switch(config-mst)#
```

The **instance** command assigns VLANs to MST instances. The **name (mst-configuration mode)** and **revision (mst-configuration mode)** commands configure the MST region name and revision.

Examples

- These commands assign VLANs 4-7 and 9 to instance 8 and remove VLAN 6 from instance 10.

```
switch(config-mst)#instance 8 vlans 4-7,9
switch(config-mst)#no instance 10 vlans 6
switch(config-mst)#
```

- These commands assign the **name** (***corporate_1***) and **revision** (***3***) to the switch.

```
switch(config-mst)#name corporate_1
switch(config-mst)#revision 3
switch(config-mst)#
```

The **exit (mst-configuration mode)** command transitions the switch out of MST configuration mode and saves all pending changes. The **abort (mst-configuration mode)** command exits MST configuration mode without saving the pending changes.

Example

- This command exits MST configuration mode and saves all pending changes.

```
switch(config-mst)#exit
switch(config)#
```

Configuring MST Instances

These STP commands provide an optional MST instance parameter. These commands apply to instance 0 when the optional parameter is not included.

- **spanning-tree priority**
- **spanning-tree root**
- **spanning-tree port-priority**

Examples

- This command configures priority for MST instance 4.

```
switch(config)#spanning-tree mst 4 priority 4096
switch(config)#
```

- Each of these commands configure priority for MST instance 0.

```
switch(config)#spanning-tree mst 0 priority 4096
```

or

```
switch(config)#spanning-tree priority 4096
```

22.3.1.2 Rapid Spanning Tree (RST)

Rapid spanning tree is enabled through the **spanning-tree mode** command with the ***rstp*** option.

Example

- This command enables Rapid Spanning Tree.

```
switch(config)#spanning-tree mode rstp
switch(config)#
```

These STP commands, when they do not include an optional MST or VLAN parameter, apply to RSTP. Commands that configure MSTP instance 0 also apply to the RSTP instance.

- **spanning-tree priority**
- **spanning-tree root**
- **spanning-tree port-priority**

Examples

- These commands apply to the RST instance.

```
switch(config)#spanning-tree priority 4096
```

and

```
switch(config)#spanning-tree mst 0 priority 4096
```

- These commands do not apply to the RST instance.

```
switch(config)#spanning-tree mst 4 priority 4096
```

and

```
switch(config)#spanning-tree VLAN 3 priority 4096
```

Show commands (such as **show spanning-tree**) displays the RSTP instance as MST0 (MST instance 0).

Example

- This command, while the switch is in RST mode, displays RST instance information.

```
switch(config)#show spanning-tree
MST0
  Spanning tree enabled protocol rstp              <---RSTP mode indicator
  Root ID    Priority    32768
             Address     001c.730c.1867
             This bridge is the root

  Bridge ID  Priority    32768  (priority 32768 sys-id-ext 0)
             Address     001c.730c.1867
             Hello Time  2.000 sec  Max Age 20 sec  Forward Delay 15 sec

Interface        Role       State      Cost      Prio.Nbr Type
---------------- ---------- ---------- --------- -------- --------------------
Et51             designated forwarding 2000      128.51   P2p

switch(config)#
```

22.3.1.3 Rapid Per-VLAN Spanning Tree (Rapid-PVST)

Rapid-PVST mode is enabled by the **spanning-tree mode** command with the ***rapid-pvst*** option.

Example

- This command enables Rapid Per-VLAN Spanning Tree.

```
switch(config)#spanning-tree mode rapid-pvst
switch(config)#
```

These commands provide an optional VLAN parameter for configuring Rapid-PVST instances.

- **spanning-tree priority**
- **spanning-tree root**
- **spanning-tree port-priority**

Example

- This command configures bridge priority for VLAN 4.

```
switch(config)#spanning-tree VLAN 4 priority 4096
switch(config)#
```

22.3.1.4 Switchport Backup Mode

Switchport backup interface pairs are enabled through the **spanning-tree mode** command with the ***backup*** option. Enabling switchport backup disables all spanning-tree modes. For loop avoidance under switchport backup mode, use the Loop Protection feature.

Example

- This command enables switchport backup.

```
switch(config)#spanning-tree mode backup
switch(config)#
```

The **switchport backup interface** command establishes an interface pair between the command mode interface (primary) and the interface specified by the command (backup).

Examples

- These commands establish Ethernet interface 7 as the backup port for Ethernet interface 1.

```
switch(config)#interface ethernet 1
switch(config-if-Et1)#switchport backup interface ethernet 7
switch(config-if-Et1)#
```

The ***prefer*** option of the **switchport backup interface** command establishes a peer relationship between the primary and backup interfaces and specifies VLAN traffic that the backup interface normally carries. If either interface goes down, the other interface carries traffic normally handled by both interfaces.

Examples

These steps perform the following:

- configures Ethernet interface 1 as a trunk port that handles VLANs 4 through 9 traffic.
- configures Ethernet interface 2 as the backup interface.
- assigns Ethernet 2 as the preferred interface for VLANs 7 through 9.

Step 1 Enter configuration mode for the primary interface

```
switch(config)#interface ethernet 1
```

Step 2 Configure the primary interface as a trunk port that services VLANs 4-9

```
switch(config-if-Et1)#switchport mode trunk
switch(config-if-Et1)#switchport trunk allowed vlan 4-9
```

Step 3 Configure the backup interface and specify the VLANs that it normally services.

```
switch(config-if-Et1)#switchport backup Ethernet 2 prefer vlan 7-9
switch(config-if-Et1)#
```

22.3.1.5 Loop Protection

Loop protection is a loop detection and prevention method which is independent of STP and is not disabled when the switch is in switchport backup mode. When loop protection is active on an interface, that interface periodically sends out loop-detection frames; if one is received that originated on the switch, the receiving port is errdisabled until a timeout period has passed or it is manually reset.

Loop protection is configured and enabled per VLAN, but individual ports in a VLAN can be configured to disable loop protection.

Note Loop protection cannot be enabled on an MLAG peer link.

This feature is disabled by default. To enable it, use the **monitor loop-protection** command to enter loop-protection configuration mode, then use the **no shutdown (Loop-protection)** command to enable the feature. To enable loop protection on a VLAN, use the **protect vlan** command. To exclude a port from loop protection, use the **no loop-protection** command.

The feature is configured with the following additional commands:

- **transmit-interval**
- **disabled-time**
- **rate-limit**

Examples

- This command enters loop protection configuration mode.

```
switch(config)#monitor loop-protection
switch(config-monitor-loop-protect)#
```

- These commands enable loop protection on VLANs 1025-2000.

```
switch(config)#monitor loop-protection
switch(config-monitor-loop-protect)#no shutdown
switch(config-monitor-loop-protect)#protect vlan 1025-2000
switch(config-monitor-loop-protect)#
```

- These commands exclude Ethernet interface 38 from loop protection.

```
switch(config)#interface ethernet 38
switch(config-if-Et38)#no loop-protection
switch(config-if-Et38)#
```

- These commands configure loop protection with a transmission interval of 10 seconds, a disabled time of two days, and a maximum rate of 500 loop detection frames per second.

```
switch(config-monitor-loop-protect)#transmit-interval 10
switch(config-monitor-loop-protect)#disabled-time 172800
switch(config-monitor-loop-protect)#rate-limit 500
switch(config-monitor-loop-protect)#
```

22.3.1.6 Disabling Spanning Tree

Spanning tree is disabled by the **spanning-tree mode** command with the ***none*** option. The switch does not generate STP packets. Switchport interfaces forward packets when connected to other ports. The switch forwards inbound STP packets as multicast data packets on the VLAN where they are received.

Examples

- This command disables all STP functions.

```
switch(config)#spanning-tree mode none
switch(config)#
```

22.3.2 Spanning Tree Instance Configuration

A network performs these steps to set up an STP instance:

Step 1 The bridge with the lowest ID is elected root bridge.

Step 2 Root ports (RP) are selected on all other bridges.

Step 3 Designated bridges are selected for each network segment.

Step 4 Designated ports (DP) are selected on each designated bridge.

Step 5 Networks begin forwarding data through RPs and DPs. All other ports are blocked.

22.3.2.1 Root Bridge Parameters

STPs use bridge IDs for electing the root bridge. Switches denote a bridge ID for each configured Spanning Tree instance. The bridge ID composition is

- Priority (four bits)

  Priority is expressed as a multiple of 4096 because it is stored as the four most significant bits of a two-byte number.
- Protocol Dependent (twelve bits)
  - Rapid-PVST: VLAN number
  - MST: Instance number
  - RST: 0
- MAC address of switch (six bytes)

Example

- The switch defines bridge IDs for three MST instances:
  - MST 0: 32768 (Priority (32768)+Instance number(0)) and 001c.7301.23de (MAC address)
  - MST101: 32869 (Priority (32768)+Instance number(101)) and 001c.7301.23de (MAC address)
  - MST102: 32870 (Priority (32768)+Instance number(102)) and 001c.7301.23de (MAC address)

  This command displays a table of root bridge information.

```
switch>show spanning-tree root
                  Root ID          Root    Hello  Max  Fwd
Instance     Priority    MAC addr   Cost    Time   Age  Dly  Root Port
----------   ------------------- --------- -----  ---  ---  ------------
MST0         32768 001c.7301.23de         0    2    20  15   Po937
MST101       32869 001c.7301.23de      3998    0     0   0   Po909
MST102       32870 001c.7301.23de      3998    0     0   0   Po911
```

The switch provides two commands that configure the switch priority: **spanning-tree priority** and **spanning-tree root**. The commands differ in the available parameter options:

- **spanning-tree priority** options are integer multiples of 4096 between 0 and 61440.
- **spanning-tree root** options are ***primary*** and ***secondary***.
- ***primary*** assigns a priority of 8192.
- ***secondary*** assigns a priority of 16384.

The default priority value is 32768.

The following examples configure bridge IDs with both commands.

Examples

- These commands configure MST instance bridge priorities with the **root** command:

```
switch(config)#spanning-tree mst 0 root primary
switch(config)#spanning-tree mst 1 root secondary
switch>show spanning-tree root
                   Root ID          Root    Hello  Max  Fwd
Instance     Priority    MAC addr   Cost    Time   Age  Dly  Root Port
----------   ------------------- --------- -----  ---  ---  ------------
MST0          8192 001c.7301.6017         0    2    20  15   None
MST1         16385 001c.7301.6017         0    0     0   0   None
MST2         32770 001c.7301.6017         0    0     0   0   None
```

  - Instance 0 root priority is 8192: primary priority plus the instance number of 0.
  - Instance 1 root priority is 16385: secondary priority plus the instance number of 1.
  - Instance 2 root priority is 32770: default priority plus the instance number of 2.

  These priority settings normally program the switch to be the primary root bridge for instance 0, the secondary root bridge for instance 1, and a normal bridge for instance 2. Primary and secondary root bridge elections also depend on the configuration of other network bridges.

- These priority commands configure Rapid-PVST VLAN bridge priorities:

```
switch(config)#spanning-tree vlan 1 priority 8192
switch(config)#spanning-tree vlan 2 priority 16384
switch(config)#spanning-tree vlan 3 priority 8192
switch(config)#no spanning-tree vlan 4 priority
switch(config)#show spanning-tree root
                   Root ID          Root    Hello  Max  Fwd
Instance     Priority    MAC addr   Cost    Time   Age  Dly  Root Port
----------   ------------------- --------- -----  ---  ---  ------------
VL1           8193 001c.7301.6017         0    2    20  15   None
VL2          16386 001c.7301.6017         0    2    20  15   None
VL3           8195 001c.7301.6017         0    2    20  15   None
VL4          32788 001c.7301.6017         0    2    20  15   None
```

  - VLAN 1 root priority is 8193: configured priority plus the VLAN number of 1.
  - VLAN 2 root priority is 16386: configured priority plus the VLAN number of 2.
  - VLAN 3 root priority is 8195: configured priority plus the VLAN number of 3.
  - VLAN 4 root priority is 32788: default priority plus the VLAN number of 4.

  These priority settings normally program the switch to be the primary root bridge for VLANs 1 and 3, the secondary root bridge for VLAN2, and a normal bridge for VLAN 4. Primary and secondary root bridge elections also depend on the configuration of other network bridges.

22.3.2.2 Path Cost

Spanning tree calculates the costs of all possible paths from each component to the root bridge. The path cost is equal to the sum of the cost assigned to each port in the path. Ports are assigned a cost by default or through CLI commands. Cost values range from 1 to 200000000 (200 million).

The default cost is a function of the interface speed:

- 1 gigabit interfaces have a default cost of 20000.
- 10 gigabit interfaces have a default cost of 2000.

The **spanning-tree cost** command configures the path cost of the configuration mode interface. Costs can be specified for Ethernet and port channel interfaces. The command provides a mode parameter for assigning multiple costs to a port for MST instances or Rapid-PVST VLANs.

Examples

- These commands configure a port cost of 25000 to Ethernet interface 5. This cost is valid for RSTP or MSTP instance 0.

```
switch(config)#interface ethernet 5
switch(config-if-Et5)#spanning-tree cost 25000
switch(config-if-Et5)#
```

- This command configures a path cost of 300000 to Ethernet interface 5 in MST instance 200.

```
switch(config)#interface ethernet 5
switch(config-if-Et5)#spanning-tree mst 200 cost 300000
switch(config-if-Et5)#
```

- This command configures a path cost of 10000 to Ethernet interface 5 in Rapid-PVST VLAN 200-220.

```
switch(config)#interface ethernet 5
switch(config-if-Et5)#spanning-tree vlan 200-220 cost 10000
switch(config-if-Et5)#
```

22.3.2.3 Port Priority

STP uses the port priority interface parameter to select ports when resolving loops. The port with the lower port priority numerical value is placed in forwarding mode. When multiple ports are assigned equal port priority numbers, the port with the lower interface number is placed in forwarding mode. Valid port-priority numbers are multiples of 16 between 0 and 240; the default is 128.

The **spanning-tree port-priority** command configures the port-priority number for the configuration mode interface. The command provides a mode option for assigning different priority numbers to a port for multiple MST instances or Rapid-PVST VLANs. Port-priority can be specified for Ethernet and port channel interfaces.

Examples

- This command sets the access port priority of 144 for Ethernet 5 interface.

```
switch(config)#interface ethernet 5
switch(config-if-Et5)#spanning-tree port-priority 144
switch(config-if-Et5)#
```

- This command sets the access port priority of 144 for Ethernet 5 interface in MST instance 10.

```
switch(config)#interface ethernet 5
switch(config-if-Et5)#spanning-tree mst 10 port-priority 144
switch(config-if-Et5)#
```

22.3.3 Port Roles and Rapid Convergence

Spanning Tree provides the following options for controlling port configuration and operation:

- **PortFast**: Allows ports to skip learning state before entering the forwarding state.
- **Port type** and **link type:** Designates ports for rapid transitions to the forwarding state.
- **Root guard**: Ensures that a port will not become the root port.
- **Loop guard**: Prevents loops resulting from unidirectional failure of links.
- **Bridge assurance**: Prevents loops caused by unidirectional links or a malfunctioning switch.

22.3.3.1 PortFast

PortFast allows devices to gain immediate network access before convergence of the spanning tree. Enabling PortFast on ports connected to another switch can create loops.

A ***portfast*** port that receives a BPDU sets its operating state to ***non-portfast*** while remaining in ***portfast*** configured state. In this state, the port is subject to topology changes and can enter the discarding state.

The **spanning-tree portfast** command programs access ports to immediately enter the forwarding state. PortFast connects devices attached to an access port, such as a single workstation, to the network immediately without waiting for STP convergence. PortFast can also be enabled on trunk ports.

Example

- This command unconditionally enables portfast on Ethernet 5 interface.

```
switch(config)#interface ethernet 5
switch(config-if-Et5)#spanning-tree portfast
switch(config-if-Et5)#
```

22.3.3.2 Port Type and Link Type Configuration

RSTP only achieves rapid transition to forwarding state on edge ports and point-to-point links.

Port Type

Edge ports are directly connected to end stations. Because edge ports do not create loops, they transition directly to forwarding state when a link is established.

The **spanning-tree portfast <port type>** command sets the configuration mode interface's port type. Spanning tree ports can be configured as ***edge*** ports, ***network*** ports, or ***normal*** ports. The default port type is ***normal***.

- **Edge ports** connect to a host (end station). Configuring a port that connects to a bridge as an edge port may create a loop. Edge ports that receive a BPDU become a normal spanning tree port.
- **Network ports** connect only to a Layer 2 switch or bridge. Configuring a port connected to a host as a network port transitions the port to the discarding state.
- N**ormal ports** have an unspecified topology.

Example

- This command configures Ethernet 5 interface as a network port.

```
switch(config)#interface ethernet 5
switch(config-if-Et5)#spanning-tree portfast network
switch(config-if-Et5)#
```

Auto-edge detection converts ports into edge ports when they do not receive a new BPDU before the current BPDU expires, as measured by the max-age timer. The **spanning-tree portfast auto** command enables auto-edge detection on the configuration mode interface, superseding the **spanning-tree portfast** command. Auto-edge detection is enabled by default.

Example

- This command enables auto-edge detection on Ethernet interface 5.

```
switch(config)#interface ethernet 5
switch(config-if-Et5)#spanning-tree portfast auto
switch(config-if-Et5)#
```

Link Type

The switch derives a port's default link type from its duplex mode:

- full-duplex ports are ***point-to-point***.
- half-duplex ports are ***shared***.

The **spanning-tree link-type** command specifies the configuration mode interface's link-type. RSTP fast transition is not allowed on ***shared link*** ports, regardless of their duplex setting. Because the ports are full-duplex by default, the default link-type setting is ***point-to-point***.

Example

- This command configures Ethernet 5 interface as a shared port.

```
switch(config)#interface ethernet 5
switch(config-if-Et5)#spanning-tree link-type shared
switch(config-if-Et5)#
```

22.3.3.3 Root Guard and Loop Guard

Root guard stops a port from becoming a root port, which stops connected switches from becoming root bridges. When a switch detects a new root bridge, its root-guard-enabled ports enter blocked (root-inconsistent) state. When the switch no longer detects a new root, these ports enter learning state.

Root guard is enabled on a per-port basis. The setting applies to all STP instances. Disabling root guard places the port in learning state.

The **spanning-tree guard** command, with the root option, enables root guard on the configuration mode interface.

Example

- This command enables root guard on Ethernet 5 interface.

```
switch(config)#interface ethernet 5
switch(config-if-Et5)#spanning-tree guard root
switch(config-if-Et5)#
```

Loop guard prevents loops resulting from unidirectional failure of point-to-point links by verifying that non-designated ports (root, blocked, and alternate) are receiving BPDUs from their designated ports. A loop-guard-enabled root or blocked port that stops receiving BPDUs transitions to the discarding (loop-inconsistent) state. The port recovers from this state when it receives a BPDU.

Loop guard, when enabled globally, applies to all point-to-point ports. Loop guard is configurable on individual ports and applies to all STP instances of an enabled port. Loop-inconsistent ports transition to learning state when loop guard is disabled.

If loop guard is enabled on a root switch, it takes effect only if the switch becomes a nonroot switch.

When using loop guard:

- Do not enable loop guard on portfast-enabled ports.
- Loop guard is not functional on ports not connected to point-to-point links.
- Loop guard has no effect on disabled spanning tree instances.

Loop guard aspects on port channels include:

- BPDUs are sent over the channel's first operational port. Loop guard blocks the channel if that link becomes unidirectional even when other channel links function properly.
- Creating a new channel destroys state information for its component ports; new channels with loop-guard-enabled ports can enter forwarding state as a DP.
- Dissembling a channel destroys its state information; component ports from a blocked channel can enter the forwarding state as DPs, even if the channel contained unidirectional links.
- If a link on any port of the channel becomes unidirectional, the channel is blocked. Transmission resumes if the port is removed from the channel or the bidirectional communication is restored.

Loop guard configuration commands include:

- **spanning-tree loopguard default** command enables loop guard as a default on all switch ports.
- **spanning-tree guard** control the loop guard setting on the configuration mode interface. This command overrides the default command for the specified interface.

Examples

- This command enables loop guard as the default on all switch ports.

```
switch(config)#spanning-tree loopguard default
switch(config)#
```

- This command enables loop guard on Ethernet 6 interface.

```
switch(config)#interface ethernet 6
switch(config-if-Et6)#spanning-tree guard loop
switch(config-if-Et6)#
```

22.3.3.4 Bridge Assurance

Bridge assurance protects against unidirectional link failures, other software failures, and devices that continue forwarding data traffic after they quit running spanning tree.

Bridge assurance programs the switch to send BPDUs at each hello time period through all bridge assurance-enabled ports (i.e., network ports). Bridge assurance operates only on ***network*** ports with ***point-to-point*** links, ideally with bridge assurance enabled on each side of the link. Bridge assurance-enabled ports will not necessarily be blocked when they link to a port where bridge assurance is not enabled.

Ports not receiving a BPDU packet within a hello time period enter inconsistent (blocking) state. In this case, the **show spanning-tree bridge assurance** command will show a bridge assurance status of "inconsistent" for the port. If the other side of the link has bridge assurance enabled, or if the other switch is the root bridge, it will send periodic BPDUs, preventing an "inconsistent" blocking state.

Bridge assurance is globally enabled by default, but must also be enabled on a per-port basis by designating the port as a network port with the **spanning-tree portfast <port type>** command. The **no spanning-tree bridge assurance** command disables bridge assurance globally.

Example

- These commands enable bridge assurance on the switch, then enable bridge assurance on Ethernet port 5 by designating it a network port.

```
switch(config)#spanning-tree bridge assurance
switch(config)#interface ethernet 5
switch(config-if-Et5)#spanning-tree portfast network
switch(config-if-Et5)#
```

22.3.4 Configuring BPDU Transmissions

The following sections describe instructions that configure BPDU packet contents and transmissions.

22.3.4.1 Bridge Timers

Bridge timers configure parameter values that the switch includes in BPDU packets that it sends as a root bridge. Bridge timers include:

- **hello-time**: the transmission interval between consecutive outbound BPDU packets.
- **forward-time**: the period that ports are in learning state prior to forwarding packets.
- **max-age**: the period that BPDU data remains valid after it is received. The switch recomputes the spanning tree topology if it does not receive another BPDU packet before the timer expires.
- **max-hop**: the number of bridges in an MST region that a BPDU can traverse before it is discarded.

In standard STP, ports passively wait for **forward_delay** and **max_age** periods before entering the forwarding state. RSTP achieves faster convergence by relying on edge port and link type definitions to start forwarding traffic. When edge ports and link types are properly configured, bridge timers are used in RSTP as backup or when interacting with networks running standard STP.

The **spanning-tree hello-time** command configures the hello time.

Example

- This command configures a hello-time of 1 second (1000 ms).

```
switch(config)#spanning-tree hello-time 1000
switch(config)#
```

The **spanning-tree max-hops** command specifies the max hop setting that the switch inserts into BPDUs that it sends out as the root bridge.

Example

- This command sets the max hop value to 40.

```
switch(config)#spanning-tree max-hops 40
switch(config)#
```

The **spanning-tree forward-time** command configures the forward delay setting that the switch inserts into BPDUs that it sends out as the root bridge.

Example

- This command sets the forward delay timer value to 25 seconds.

```
switch(config)#spanning-tree forward-time 25
switch(config)#
```

The **spanning-tree max-age** command configures the max age setting that the switch inserts into BPDUs that it sends out as the root bridge.

Example

- This command sets the max age timer value to 25 seconds.

```
switch(config)#spanning-tree max-age 25
switch(config)#
```

22.3.4.2 BPDU Transmit Hold-Count

The **spanning-tree transmit hold-count** command specifies the maximum number of BPDUs per second that the switch can send from an interface. Valid settings range from 1 to 10 BPDUs with a default of 6 BPDUs.

Higher hold-count settings can significantly impact CPU utilization, especially in Rapid-PVST mode. Smaller values can slow convergence in some configurations.

Example

- This command configures a transmit hold-count of 8 BPDUs.

```
switch(config)#spanning-tree transmit hold-count 8
switch(config)#
```

22.3.4.3 BPDU Guard

PortFast interfaces do not receive BPDUs in a valid configuration. BPDU Guard provides a secure response to invalid configurations by disabling ports when they receive a BPDU. Disabled ports differ from blocked ports in that they are re-enabled only through manual intervention.

- When configured globally, BPDU Guard is enabled on ports in the operational portfast state.
- When configured on an individual interface, BPDU Guard disables the port when it receives a BPDU, regardless of the port's portfast state.

The **spanning-tree portfast bpduguard default** global configuration command enables BPDU guard by default on all portfast ports. BPDU guard is disabled on all ports by default.

The **spanning-tree bpduguard** interface configuration command controls BPDU guard on the configuration mode interface. This command takes precedence over the default setting configured by **spanning-tree portfast bpduguard default**.

- **spanning-tree bpduguard** enables BPDU guard on the configuration mode interface.
- **spanning-tree bpduguard disable** disables BPDU guard on the configuration mode interface.
- **no spanning-tree bpduguard** reverts the configuration mode interface to the default BPDU guard setting.

Example

- These commands enable BPDU guard by default on all portfast ports, then disable BPDU guard on Ethernet 5.

```
switch(config)#spanning-tree portfast bpduguard default
switch(config)#interface ethernet 5
switch(config-if-Et5)#spanning-tree bpduguard disable
switch(config-if-Et5)
```

22.3.4.4 BPDU Filter

BPDU filtering prevents the switch from sending or receiving BPDUs on specified ports. BPDU filtering is configurable on Ethernet and port channel interfaces.

Ports with BPDU filtering enabled do not send BPDUs and drops inbound BPDUs. Enabling BPDU filtering on a port not connected to a host can result in loops as the port continues forwarding data while ignoring inbound BPDU packets.

The **spanning-tree bpdufilter** command controls BPDU filtering on the configuration mode interface. BPDU filtering is disabled by default.

Example

- These commands enable BPDU filtering on Ethernet 5.

```
switch(config)#interface ethernet 5
switch(config-if-Et5)#spanning-tree bpdufilter enable
switch(config-if-Et5)#
```

22.3.4.5 BPDU Rate Limit

BPDU input rate limiting restricts the number of BPDUs that a port with BPDU guard and BPDU filter disabled can process during a specified interval. The port discards all BPDUs that it receives in excess of the specified limit. Configuring the rate limiter requires two steps:

- Establishing the rate limit threshold.
- Enabling rate limiting.

Establishing the Rate Limit Threshold

The **spanning-tree bpduguard rate-limit count (interface)** commands specify BPDU reception rate (quantity per interval) that trigger the discarding of BPDUs. Commands are available in global and interface configuration modes.

- The **spanning-tree bpduguard rate-limit count** global command specifies the maximum reception rate for ports not covered by interface rate limit count commands. The default quantity is 10 times the number of VLANs. The default interval is the hello time (**spanning-tree hello-time**).
- The **spanning-tree bpduguard rate-limit count** interface command defines the maximum BPDU reception rate for the configuration mode interface. The global command specifies the default limit.

Examples

- This command configures the global limit of 5000 BPDUs over a four second interval.

```
switch(config)#spanning-tree bpduguard rate-limit count 5000 interval 4
switch(config)#
```

- These commands configures a limit of 7500 BPDUs over an 8 second interval on Ethernet interface 2.

```
switch(config)#interface ethernet 2
switch(config-if-Et2)#spanning-tree bpduguard rate-limit count 7500 interval 8
switch(config-if-Et2)#
```

Enabling Rate Limiting

BPDU rate limiting is enabled globally or on individual ports:

- **spanning-tree bpduguard rate-limit default** enables rate limiting on all ports with no interface rate limiting command. The default setting is ***enabled***.
- **spanning-tree bpduguard rate-limit enable / disable** interface command enables or disables BPDU rate limiting on the configuration mode interface. This command has precedence over the global command.

Examples

- This command enables rate limiting on ports not covered by interface rate limit commands.

```
switch(config)#spanning-tree bpduguard rate-limit default
switch(config)#
```

- These commands enables rate limiting on Ethernet 15.

```
switch(config)#interface ethernet 15
switch(config-if-Et15)#spanning-tree bpduguard rate-limit enable
switch(config-if-Et15)#
```

22.4 STP Commands

Spanning Tree Commands: Global Configuration

- spanning-tree bpduguard rate-limit default
- spanning-tree bpduguard rate-limit count (global)
- spanning-tree bridge assurance
- spanning-tree forward-time
- spanning-tree hello-time
- spanning-tree loopguard default
- spanning-tree max-age
- spanning-tree max-hops
- spanning-tree mode
- spanning-tree mst configuration
- spanning-tree portchannel guard misconfig
- spanning-tree portfast bpdufilter default
- spanning-tree portfast bpduguard default
- spanning-tree priority
- spanning-tree root
- spanning-tree transmit hold-count
- spanning-tree vlan

Loop Protection Commands

- disabled-time
- loop-protection
- monitor loop-protection
- protect vlan
- rate-limit
- shutdown (Loop-protection)
- transmit-interval

Spanning Tree Commands: Interface Configuration Mode

- spanning-tree bpdufilter
- spanning-tree bpduguard
- spanning-tree bpduguard rate-limit count (interface)
- spanning-tree bpduguard rate-limit enable / disable
- spanning-tree cost
- spanning-tree guard
- spanning-tree link-type
- spanning-tree port-priority
- spanning-tree portfast
- spanning-tree portfast auto
- spanning-tree portfast <port type>
- switchport backup interface

MST Configuration Commands

- abort (mst-configuration mode)
- exit (mst-configuration mode)
- instance
- name (mst-configuration mode)
- revision (mst-configuration mode)
- show (mst-configuration mode)

Display Commands

- show spanning-tree
- show spanning-tree blockedports
- show spanning-tree bridge
- show spanning-tree counters
- show spanning-tree interface
- show spanning-tree mst
- show spanning-tree mst configuration
- show spanning-tree mst interface
- show spanning-tree mst test information
- show spanning-tree root
- show spanning-tree topology status

Clear Commands

- clear spanning-tree counters
- clear spanning-tree counters session
- clear spanning-tree detected-protocols

abort (mst-configuration mode)

The **abort** command, in MST-configuration mode, discards pending changes to the MST region configuration, then returns the switch to global configuration mode.

The **exit (mst-configuration mode)** command saves MST region changes to ***running-config*** before returning the switch to global configuration mode.

Command Mode

MST-configuration

Command Syntax

abort

Examples

- This command discards changes to the MST region, then returns the switch to global configuration mode.

```
switch(config-mst)#abort
switch(config)#
```

clear spanning-tree counters

The **clear spanning-tree counters** command resets the BPDU counters for the specified interfaces to zero in all CLI sessions.

Command Mode

Privileged EXEC

Command Syntax

clear spanning-tree counters [*INT_NAME*]

Parameters

- ***INT_NAME*** Interface type and number. Options include:
  - <no parameter> resets counters for all interfaces.
  - **interface ethernet** *e_num* Ethernet interface specified by *e_num*.
  - **interface loopback** *l_num* Loopback interface specified by *l_num*.
  - **interface management** *m_num* Management interface specified by *m_num*.
  - **interface port-channel** *p_num* Port-Channel Interface specified by *p_num*.
  - **interface vlan** *v_num* VLAN interface specified by *v_num*.

Examples

- This command resets the BPDU counters on Ethernet 15 interface.

```
switch#show spanning-tree counters
            Port       Sent      Received       Tagged Error    Other Error
-------------------------------------------------------------------------------
      Ethernet15      32721             0                  0              0
  Port-Channel10       8487             0                  0              0

switch#clear spanning-tree counters interface ethernet 15    <---Clear command
switch#show spanning-tree counters
            Port      Sent       Received       Tagged Error    Other Error
-------------------------------------------------------------------------------
      Ethernet15        11              0                  0              0
  Port-Channel10         8494           2                  6              0

switch#
```

clear spanning-tree counters session

The **clear spanning-tree counter session** command resets the BPDU counters to zero on all interfaces in the current CLI session. Counters in other CLI sessions are not affected.

Command Mode

Privileged EXEC

Command Syntax

clear spanning-tree counters session

Examples

- This command resets the BPDU counters in the current CLI session.

```
switch#show spanning-tree counters
            Port       Sent      Received       Tagged Error    Other Error
-------------------------------------------------------------------------------
      Ethernet15      32721             0                  0              0
  Port-Channel10       8487             0                  0              0

switch#clear spanning-tree counters session
switch#show spanning-tree counters
            Port      Sent       Received       Tagged Error    Other Error
-------------------------------------------------------------------------------
      Ethernet15        11             0                  0              0
  Port-Channel10         7             2                  6              0

switch#
```

clear spanning-tree detected-protocols

The **clear spanning-tree detected-protocols** command restarts the spanning tree protocol (STP) migration state machine on the specified interfaces. The switch is reset to running rapid spanning tree protocol on an interface where it previously detected a bridge running an old version of the protocol.

Command Mode

Privileged EXEC

Command Syntax

clear spanning-tree detected-protocols [*INT_NAME*]

Parameters

- ***INT_NAME*** Interface type and number. Values include:
  - <no parameter> all interfaces.
  - **ethernet** *e_num* Ethernet interface specified by *e_num*.
  - **loopback** *l_num* Loopback interface specified by *l_num*.
  - **management** *m_num* Management interface specified by *m_num*.
  - **port-channel** *p_num* Port-Channel Interface specified by *p_num*.
  - **vlan** *v_num* VLAN interface specified by *v_num*.

Examples

- This command restarts the STP migration machine on all switch interfaces.

```
switch#clear spanning-tree detected-protocols
switch#
```

disabled-time

The **disabled-time** command sets the time for which the port remains disabled after a loop is detected by loop protection. The **no disabled-time** and **default disabled-time** commands reset the disabled time to the default of 604800 seconds (seven days).

Note If this value is changed, interfaces that are already disabled by loop protection will remain disabled for the previously configured period.

Command Mode

Loop-protection Configuration

Command Syntax

disabled-time *period*
no disabled-time [*period*]
default disabled-time [*period*]

Parameters

- *period* Time in seconds for which the port remains disabled. Values range from 0 to 604800 (seven days). Default is 604800. A value of 0 disables the interface until it is manually reset, even if the disabled time is later set to a non-zero value. To restore the port manually, shut it down and then re-enable it.

Example

- This command configures loop protection to disable a port on which a loop is detected for a period of two days (172800 seconds).

```
switch(config-monitor-loop-protect)#disabled-time 172800
switch(config-monitor-loop-protect)#
```

exit (mst-configuration mode)

The **exit** command, in MST-configuration mode, saves changes to the MST region configuration, then returns the switch to global configuration mode. MST region configuration changes are also saved by entering a different configuration mode.

Command Mode

MST-configuration

Command Syntax

```
exit
```

Examples

- This command saves changes to the MST region, then returns the switch to global configuration mode.

```
switch(config-mst)#exit
switch(config)#
```

- This command saves changes to the MST region, then places the switch in Interface-Ethernet mode.

```
switch(config-mst)#interface ethernet 3
switch(config-if-Et3)#
```

instance

The **instance** command inserts an entry into the VLAN-to-instance map that associates a set of VLANs to an MST instance. In addition to defining the MST topology, the VLAN-to-instance map is one of three parameters, along with the MST name and revision number, that identifies the switch's MST region.

The **no instance** command removes specified entries from the VLAN-to-instance map. If the command does not provide a VLAN list, all entries are removed for the specified instance. The **no instance** and **default instance** commands function identically.

Command Mode

MST-Configuration

Command Syntax

```
instance mst_inst vlans v_range
no instance mst_inst [vlans v_range]
no default instance mst_inst [vlans v_range]
```

Parameters

- *mst_inst* MST instance number. Value of *mst_inst* ranges from 0 to 4094.
- *v_range* VLAN list. Formats include a number, number range, or comma-delimited list of numbers and ranges.

Examples

- This command maps VLANs 20-39 to MST instance 2

```
switch(config)#spanning-tree mst configuration
switch(config-mst)#instance 2 vlans 20-39
switch(config-mst)#
```

- This command removes all VLAN mappings to MST instance 10.

```
switch(config-mst)#no instance 10
switch(config-mst)#
```

loop-protection

The **loop-protection** command enables loop protection on the configuration mode interface. All interfaces in a VLAN under loop protection have loop protection enabled by default. The **no loop-protection** and **default loop-protection** commands disable loop protection on the interface.

When loop protection is disabled (at the VLAN or interface level), the computed state of the interface is forgotten and packets queued to be sent are dropped. If an interface is err-disabled by loop protection, disabling loop protection removes the err-disable.

Command Mode

Interface-Ethernet Configuration

Command Syntax

```
loop-protection
no loop-protection
default loop-protection
```

Example

- These commands disable loop protection on Ethernet interface 2/4. If the interface is currently err-disabled by loop protection, the err-disable will be removed.

```
switch(config)#interface ethernet 2/4
switch(config-if-Et2/4)#no loop-protection
switch(config-if-Et2/4)#
```

monitor loop-protection

The **monitor loop-protection** command places the switch in loop-protection configuration mode.

Command Mode

Global Configuration

Command Syntax

```
monitor loop-protection
```

Commands available in loop-protection configuration mode:

- **shutdown (Loop-protection)**
- **protect vlan**
- **transmit-interval**
- **disabled-time**
- **rate-limit**

Example

- This command places the switch in loop-protection configuration mode.

```
switch(config)#monitor loop-protection
switch(config-monitor-loop-protect)#
```

name (mst-configuration mode)

The **name** command configures the MST region name. The name is one of three parameters, along with the MST revision number and VLAN-to-instance map, that identifies the switch's MST region.

The name has up to 32 characters. The default name is an empty string. The name string accepts all characters except the space.

The **no name** and **default name** commands restore the default name by removing the **name** command from ***running-config***.

Command Mode

MST-Configuration

Command Syntax

name *label_text*
no name
default name

Parameters

- *label_text* character string assigned to name attribute. Maximum 32 characters. The space character is not permitted in the name string.

Example

- This command assigns corporate_100 as the MST region name.

```
switch(config)#spanning-tree mst configuration
switch(config-mst)#name corporate_100
switch(config-mst)#show pending
Active MST configuration
Name      [corporate_100]
Revision  0    Instances configured 1

Instance  Vlans mapped
--------
------------------------------------------------------------------------
0         1-4094
------------------------------------------------------------------------------
-
```

protect vlan

The **protect vlan** command specifies which VLANs will participate in loop protection. The **no protect vlan** and **default protect vlan** commands remove loop protection from the specified VLANs.

Command Mode

Loop-protectionConfiguration

Command Syntax

protect vlan *vlan-range*

Parameters

- *vlan-range* List of VLANs (number, range, comma-delimited list of numbers and ranges). Numbers range from 1 to 4094.

Example

- This command enables loop protection on VLANs 1025-2000.

```
switch(config-monitor-loop-protect)#protect vlan 1025-2000
switch(config-monitor-loop-protect)#
```

rate-limit

The **rate-limit** command sets the maximum number of loop detection frames which can be sent by the switch per second. The **no rate-limit** and **default rate-limit** commands return the rate limit to the default value of 1000.

Command Mode

Loop-protection Configuration

Command Syntax

```
rate-limit frames
no rate-limit [frames]
default rate-limit [frames]
```

Parameters

- *frames* Maximum number of frames sent per second. Values range from 0-1000, default is 1000. A value of zero disables throttling.

Example

- This command sets the maximum number of loop detection frames to 500 per second.

```
switch(config-monitor-loop-protect)#rate-limit 500
   switch(config-monitor-loop-protect)#
```

revision (mst-configuration mode)

The **revision** command configures the MST revision number. The revision number is one of three parameters, along with the MST name and VLAN-to-instance map, that identifies the switch's MST region. Revision numbers range from 0 to 65535. The default revision number is 0.

The **no revision** and **default revision** commands restore the revision number to its default value by removing the revision command from ***running-config***.

Command Mode

MST-Configuration

Command Syntax

```
revision rev_number
no revision
default revision
```

Parameters

- *rev_number* revision number. Possible ranges from 0 to 65535 with a default of 0.

Examples

- This command sets the revision number to 15.

```
switch(config)#spanning-tree mst configuration
switch(config-mst)#revision 15
switch(config-mst)#show pending
Active MST configuration
Name      []
Revision 15   Instances configured 1

Instance  Vlans mapped
-------- ---------------------------------------------------------------
0         1-4094
---------------------------------------------------------------------------
```

show (mst-configuration mode)

The **show** command displays the current and pending MST configuration:

Exiting MST configuration mode stores all pending configuration changes to ***running-config***.

Command Mode

MST-Configuration

Command Syntax

`show [EDIT_VERSION]`

Parameters

- ***EDIT_VERSION*** specifies configuration version that the command displays. Options include:
  - <no parameter> command displays pending MST configuration.
  - **active** command displays MST configuration stored in ***running-config***.
  - **current** command displays MST configuration stored in ***running-config***.
  - **pending** command displays pending MST configuration.

Example

- These commands contrast the difference between the active and pending configuration by adding MST configuration commands, then showing the configurations.

```
switch(config-mst)#show pending
Active MST configuration
Name      []
Revision  0    Instances configured 1

Instance  Vlans mapped
--------
------------------------------------------------------------------------
0         1-4094
------------------------------------------------------------------------------
-
switch(config-mst)#instance 2 vlan 20-29,102
switch(config-mst)#revision 2
switch(config-mst)#name baseline
switch(config-mst)#show pending
Pending MST configuration
Name      [baseline]
Revision  2    Instances configured 2

Instance  Vlans mapped
--------
------------------------------------------------------------------------
0         1-19,30-101,103-4094
2         20-29,102
------------------------------------------------------------------------------
-
switch(config-mst)#show active

Active MST configuration
Name      []
Revision  0    Instances configured 1

Instance  Vlans mapped
--------
------------------------------------------------------------------------
0         1-4094
------------------------------------------------------------------------------
-
```

show loop-protection

The **show loop-protection** command displays loop protection status.

Command Syntax

show loop-protection [*detail*]

Examples

- This command displays basic loop protection information.

```
switch>show loop-protection
Loop protection is enabled
Transmit interval: 5
Disable Time: 604800(or Permanent)
Packets Transmitted rate: 12/second(or Unthrottled)
Total: 3 Vlans enabled.
switch>
```

- This command displays detailed information about loop protection.

```
switch>show loop-protection detail
Loop protection is enabled

Transmit interval: 5
Disable Time: 604800
Packets Transmitted rate: 12/second
Total: 3 Vlans enabled.
Destination address: ffff.ffff.ffff
Ethernet type: 0x88b7
Receive action: Interface Disable

Vlan        Loop     Disabled Intfs  Total Latest
            Detected                 Intfs Disabled Time
----------- --------- -------------- ----- -------------
1           Yes       Et1-2          20    18:01
2           No        -              20    -
3           No        -              20    -
switch>
```

- This command displays loop protection information for the interfaces in VLANS 3-4.

```
switch>show loop-protection vlan 3-4
Vlan  Intf  LP Enabled State     LP       Disabled Bring
                                Disabled at       up at
----- ----- ---------- --------- -------  -------- ------
3     Et1   Yes        shutdown   Yes      17:21    18:21
3     Et2   Yes        shutdown   No       -        -
3     Et3   Yes        enabled    No       -        -
3     Et4   No         -          -        -        -
4     -     No         -          -        -        -
switch>
```

- This command displays the number of loop detection packets sent and received.

```
switch>show loop-protection counters
VLAN        Tx      Rx    Rx-Other
-------     -----   ----  --------
2           200     0     100
3           200     1     0

Intfs       Tx      Rx    Rx-Other
-------     -----   ----  --------
Et1         200     0     100
Et2         200     1     0
switch>
```

show spanning-tree

The **show spanning-tree command** displays spanning tree protocol (STP) data, organized by instance.

Command Mode

EXEC

Command Syntax

show spanning-tree [*VLAN_ID*] [*INFO_LEVEL*]

Parameters

- ***VLAN_ID*** specifies the VLANs for which the command displays information. Formats include:
  - <no parameter> displays information for all VLANs.
  - **vlan** displays data for instances containing the first VLAN listed in ***running-config***.
  - **vlan** *v_range* displays data for instances containing a VLAN in the specified range.
- ***INFO_LEVEL*** specifies level of information detail provided by the command.
  - <no parameter> displays table for each instance listing status, configuration, and history.
  - **detail** displays data blocks for each instance and all ports on each instance.

Display Values

- **Root ID** Displays information on the ROOT ID (elected spanning tree root bridge ID):
  - ***Priority***: Priority of the bridge. Default value is 32768.
  - ***Address***: MAC address of the bridge.
- **Bridge ID** bridge status and configuration information for the locally configured bridge:
  - ***Priority*** Priority of the bridge. The default priority is 32768.
  - ***Address*** MAC address of the bridge.
  - ***Hello Time*** Interval (seconds) between bridge protocol data units (BPDUs) transmissions.
  - ***Max Age*** Maximum time that a BPDU is saved.
  - ***Forward Delay*** Time (in seconds) that is spent in the learning state.
- **Interface** STP configuration participants. Link-down interfaces are not shown.
- **Role** Role of the port as one of the following:
  - ***Root*** The best port for a bridge to a root bridge used for forwarding.
  - ***Designated*** A forwarding port for a LAN segment.
  - ***Alternate*** A port acting as an alternate path to the root bridge.
  - ***Backup*** A port acting as a redundant path to another bridge port.
- **State** Displays the interface STP state as one of the following:
  - Learning
  - Discarding
  - Forwarding
- **Cost** STP port path cost value.
- **Prio. Nbr.** STP port priority. Values range from 0 to 240. Default is 128.
- **Type** The link type of the interface (automatically derived from the duplex mode of an interface):
  - ***P2p Peer (STP)*** Point to point full duplex port running standard STP.

- ***shr Peer (STP)*** Shared half duplex port running standard STP.

Examples

- This command displays STP data, including a table of port parameters.

```
switch>show spanning-tree vlan 1000
MST0
  Spanning tree enabled protocol rstp
  Root ID    Priority    32768
             Address     001c.7301.07b9
             Cost        1999 (Ext) 0 (Int)
             Port        101 (Port-Channel2)
             Hello Time  2.000 sec  Max Age 20 sec  Forward Delay 15 sec

  Bridge ID  Priority    32768  (priority 32768 sys-id-ext 0)
             Address     001c.7304.195b
             Hello Time  2.000 sec  Max Age 20 sec  Forward Delay 15 sec

Interface        Role       State      Cost      Prio.Nbr Type
---------------- ---------- ---------- --------- -------- --------------------
Et4              designated forwarding 20000     128.4    P2p
Et5              designated forwarding 20000     128.5    P2p
Et6              designated forwarding 20000     128.6    P2p
Et23             designated forwarding 20000     128.23   P2p
Et26             designated forwarding 20000     128.26   P2p
Et32             designated forwarding 2000      128.32   P2p

switch>
This command displays output from the show spanning-tree command:
Switch#show spanning-tree
MST0
  Spanning tree enabled protocol mstp
  Root ID    Priority    32768
             Address     0011.2201.0301
             This bridge is the root

  Bridge ID  Priority    32768  (priority 32768 sys-id-ext 0)
             Address     0011.2201.0301
             Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec

Interface        Role       State      Cost      Prio.Nbr Type
---------------- ---------- ---------- --------- -------- --------------------
Et4              designated forwarding 2000      128.4    P2p
Et5              designated forwarding 2000      128.5    P2p
...
PEt4             designated forwarding 2000      128.31   P2p
PEt5             designated forwarding 2000      128.44   P2p
...
Po3              designated forwarding 1999      128.1003 P2p
```

- This command displays STP data, including an information block for each interface running STP.

```
switch>show spanning-tree vlan 1000 detail
 MST0 is executing the rstp Spanning Tree protocol
  Bridge Identifier has priority 32768, sysid 0, address 001c.7304.195b
  Configured hello time 2.000, max age 20, forward delay 15, transmit hold-count 6
  Current root has priority 32768, address 001c.7301.07b9
  Root port is 101 (Port-Channel2), cost of root path is  1999 (Ext) 0 (Int)
  Number of topology changes 4109 last change occurred 1292651 seconds ago
          from Ethernet13

 Port 4 (Ethernet4) of MST0 is designated forwarding
   Port path cost 20000, Port priority 128, Port Identifier 128.4.
   Designated root has priority 32768, address 001c.7301.07b9
   Designated bridge has priority 32768, address 001c.7304.195b
   Designated port id is 128.4, designated path cost 1999 (Ext) 0 (Int)
   Timers: message age 1, forward delay 15, hold 20
   Number of transitions to forwarding state: 1
   Link type is point-to-point by default, Internal
   BPDU: sent 452252, received 0, taggedErr 0, otherErr 0, rateLimiterCount 0
   Rate-Limiter: enabled, Window: 10 sec, Max-BPDU: 400

Port 5 (Ethernet5) of MST0 is designated forwarding
   Port path cost 20000, Port priority 128, Port Identifier 128.5.
   Designated root has priority 32768, address 001c.7301.07b9
   Designated bridge has priority 32768, address 001c.7304.195b
   Designated port id is 128.5, designated path cost 1999 (Ext) 0 (Int)
   Timers: message age 1, forward delay 15, hold 20
   Number of transitions to forwarding state: 1
   Link type is point-to-point by default, Internal
   BPDU: sent 1006266, received 0, taggedErr 0, otherErr 0, rateLimiterCount 0
   Rate-Limiter: enabled, Window: 10 sec, Max-BPDU: 400

            <-------OUTPUT OMITTED FROM EXAMPLE-------->

switch>
```

show spanning-tree blockedports

The **show spanning-tree blockedports** command displays the list of blocked (discarding) ports.

Command Mode

EXEC

Command Syntax

show spanning-tree blockedports

Example

- This command shows the ports that are in discarding state.

```
switch>show spanning-tree blockedports
Name       Blocked Interfaces List
----------
------------------------------------------------------------------------
MST0      Po903, Po905, Po907, Po909, Po911, Po913, Po915, Po917, Po919, Po921,
Po923

          Po925, Po927, Po929, Po931, Po933, Po935, Po939, Po941, Po943, Po945,
Po947


Number of blocked ports (segments) in the system : 22
switch>
```

show spanning-tree bridge

The **show spanning-tree bridge** command displays spanning tree protocol bridge configuration settings for each instance on the switch. The display includes Bridge ID, Hello Time, Max Age, and Forward Delay times.

The command also displays the restartability of the STP agent when the **detail** option is selected. A switch can continue support of MLAG operation when its peer is offline and the STP agent is unavailable.

Command Mode

EXEC

Command Syntax

show spanning-tree bridge [*INFO_LEVEL*]

Parameters

- ***INFO_LEVEL*** specifies level of information detail provided by the command.
  - <no parameter> command displays information in a data table.
  - **detail** command displays bridge information in data blocks for each instance.

Examples

- This command displays a bridge data table.

```
switch>show spanning-tree bridge
                            Bridge ID                     Hello  Max  Fwd
Instance             Priority             MAC addr        Time   Age  Dly
----------   ---------------------------------------     -----  ---  ---
MST0         32768(32768, sys-id 0   ) 001c.7302.2f98      2000    20   15
MST101       32869(32768, sys-id 101 ) 001c.7302.2f98      2000    20   15
MST102       32870(32768, sys-id 102 ) 001c.7302.2f98      2000    20   15

switch>
```

- This command displays bridge data blocks.

```
switch>show spanning-tree bridge detail
Stp agent is restartable
MST0
  Bridge ID  Priority    32768  (priority 32768 sys-id-ext 0)
             Address     001c.7302.2f98
             Hello Time  2.000 sec  Max Age 20 sec  Forward Delay 15 sec
MST101
  Bridge ID  Priority    32869  (priority 32768 sys-id-ext 101)
             Address     001c.7302.2f98
             Hello Time  2.000 sec  Max Age 20 sec  Forward Delay 15 sec
MST102
  Bridge ID  Priority    32870  (priority 32768 sys-id-ext 102)
             Address     001c.7302.2f98
             Hello Time  2.000 sec  Max Age 20 sec  Forward Delay 15 sec
switch>
```

show spanning-tree bridge assurance

The **show spanning-tree bridge assurance** command displays spanning tree protocol bridge assurance information for network ports or for all ports. Bridge assurance-enabled ports will not necessarily be blocked when they link to a port where bridge assurance is not enabled, but if they do not receive periodic BPDUs from the other side of the link the **show spanning-tree bridge assurance** command will show a bridge assurance status of “inconsistent” (blocking) for that port.

Command Mode

EXEC

Command Syntax

show spanning-tree bridge assurance *INFO_LEVEL*

Parameters

- ***INFO_LEVEL*** specifies level of information detail provided by the command.
  - <no parameter> command displays bridge assurance information for network ports.
  - **detail** command displays bridge assurance information for all ports.

Examples

- This command displays the bridge assurance status of network ports.

```
switch>show spanning-tree bridge assurance
Name                 Bridge Assurance Status
-------------------------------------------
VL1        Et5/1       consistent

Number of bridge assurance inconsistent ports in the system : 0
switch>
```

show spanning-tree counters

The **show spanning-tree counters** command displays the number of BPDU transactions on each interface running spanning tree.

Command Mode

EXEC

Command Syntax

show spanning-tree counters

Example

- This command displays the BPDU counter status on each interface running spanning tree.

```
switch>show spanning-tree counters
         Port      Sent   Received   Tagged Error   Other Error  sinceTimer
-------------------------------------------------------------------------------
    Ethernet2   1008399          0              0             0           0
    Ethernet3   1008554          0              0             0           0
    Ethernet4    454542          0              0             0           0
    Ethernet5   1008556          0              0             0           0
    Ethernet6    827133          0              0             0           0
    Ethernet8   1008566          0              0             0           0
   Ethernet10    390732          0              0             0           0
   Ethernet11   1008559          0              0             0           0
   Ethernet15    391379          0              0             0           0
   Ethernet17    621253          0              0             0           0
   Ethernet19    330855          0              0             0           0
   Ethernet23    245243          0              0             0           0
   Ethernet25    591695          0              0             0           0
   Ethernet26   1007903          0              0             0           0
   Ethernet32   1010429          8              0             0           0
   Ethernet33    510227          0              0             0           0
   Ethernet34    827136          0              0             0           0
   Ethernet38   1008397          0              0             0           0
   Ethernet39   1008564          0              0             0           0
   Ethernet40   1008185          0              0             0           0
   Ethernet41   1007467          0              0             0           0
   Ethernet42     82925          0              0             0           0
Port-Channel1   1008551          0              0             0           0
Port-Channel2    334854     678589              0             0           3
Port-Channel3   1010420          4              0             0           0

switch>
```

show spanning-tree interface

The **show spanning-tree interface** command displays spanning tree protocol information for the specified interface.

Command Mode

EXEC

Command Syntax

show spanning-tree interface *INT_NAME* [*INFO_LEVEL*]

Parameters

- ***INT_NAME*** Interface type and number. Values include:
  - **ethernet** *e_num* Ethernet interface specified by *e_num*.
  - **peerethernet** *e_num* Ethernet interface specified by *e_num*.
  - **port-channel** *p_num* Port-Channel Interface specified by *p_num*.
  - **peerport-channel** *p_num* Port-Channel Interface specified by *p_num*.
- ***INFO_LEVEL*** specifies level of detail provided by the output. Options include:
  - <no parameter> command displays a table of STP data for the specified interface.
  - **detail** command displays a data block for the specified interface.

Examples

- This command displays an STP table for Ethernet interface 5.

```
switch>show spanning-tree interface ethernet 5
Instance         Role       State      Cost      Prio.Nbr Type
---------------- ---------- ---------- --------- -------- --------------------
MST0             designated forwarding 20000     128.5    P2p
switch>
```

- This command displays a data block for Ethernet interface 5.

```
switch>show spanning-tree interface ethernet 5 detail
 Port 5 (Ethernet5) of MST0 is designated forwarding
   Port path cost 20000, Port priority 128, Port Identifier 128.5.
   Designated root has priority 32768, address 001c.7301.07b9
   Designated bridge has priority 32768, address 001c.7304.195b
   Designated port id is 128.5, designated path cost 1999 (Ext) 0 (Int)
   Timers: message age 1, forward delay 15, hold 20
   Number of transitions to forwarding state: 1
   Link type is point-to-point by default, Internal
   BPDU: sent 1008766, received 0, taggedErr 0, otherErr 0, rateLimiterCount 0
   Rate-Limiter: enabled, Window: 10 sec, Max-BPDU: 400

switch>
```

show spanning-tree mst

The **show spanning-tree mst** command displays configuration and state information for Multiple Spanning Tree Protocol (MST) instances.

Command Mode

EXEC

Command Syntax

show spanning-tree mst [*INSTANCE*] [*INFO_LEVEL*]

Parameters

- ***INSTANCE*** – MST instance for which the command displays information. Options include:
  - <no parameter> all MST instances.
  - *mst_inst* MST instance number. Value of *mst_inst* ranges from 0 to 4094.
- ***INFO_LEVEL*** – type and amount of information in the output. Options include:
  - <no parameter> output is interface data in tabular format.
  - **detail** output is a data block for each interface.

Examples

- This command displays interface data blocks for MST instance 3.

```
switch>show spanning-tree mst 3 detail
##### MST3    vlans mapped:    3
Bridge        address 0011.2233.4402  priority      32771 (32768 sysid 3)
Root          address 0011.2233.4401  priority      32771 (32768 sysid 3)

Ethernet1 of MST3 is root forwarding
Port info             port id          128.1  priority    128  cost       2000
Designated root       address 0011.2233.4401  priority  32768  cost          0
Designated bridge     address 0011.2233.4401  priority  32768  port id   128.1

Ethernet2 of MST3 is alternate discarding
Port info             port id          128.2 priority    128  cost       2000
Designated root       address 0011.2233.4401 priority  32768  cost          0
Designated bridge     address 0011.2233.4401 priority  32768  port id   128.2

Ethernet3 of MST3 is designated forwarding
Port info             port id          128.3 priority    128  cost       2000
Designated root       address 0011.2233.4401 priority  32768  cost       2000
Designated bridge     address 0011.2233.4402 priority  32768  port id   128.3
```

- This command displays interface tables for all MST instances.

```
switch>show spanning-tree mst
##### MST0    vlans mapped:    1,4-4094
Bridge        address 0011.2233.4402  priority      32768 (32768 sysid 0)
Root          address 0011.2233.4401  priority      32768 (32768 sysid 0)
Regional Root address 0011.2233.4401  priority      32768 (32768 sysid 0)

Interface        Role       State      Cost      Prio.Nbr Type
---------------- ---------- ---------- --------- -------- --------------------
Et1              root       forwarding 2000      128.1    P2p
Et2              alternate  discarding 2000      128.2    P2p
Et3              designated forwarding 2000      128.3    P2p
Et4              designated forwarding 2000      128.4    P2p

##### MST2 vlans mapped: 2
Bridge        address 0011.2233.4402  priority       8194 (8192 sysid 2)
Root          this switch for MST2

Interface        Role       State      Cost      Prio.Nbr Type
---------------- ---------- ---------- --------- -------- --------------------
Et1              designated forwarding 2000      128.1    P2p
Et2              designated forwarding 2000      128.2    P2p
Et3              designated forwarding 2000      128.3    P2p
Et4              designated forwarding 2000      128.4    P2p

##### MST3 vlans mapped: 3
Bridge        address 0011.2233.4402  priority      32771 (32768 sysid 3)
Root          address 0011.2233.4401  priority      32771 (32768 sysid 3)

Interface        Role       State      Cost      Prio.Nbr Type
---------------- ---------- ---------- --------- -------- --------------------
Et1              root       forwarding 2000      128.1    P2p
Et2              alternate  discarding 2000      128.2    P2p
Et3              designated forwarding 2000      128.3    P2p
Et4              designated forwarding 2000      128.4    P2p
```

show spanning-tree mst configuration

The **show spanning-tree mst configuration** command displays information about the MST region's VLAN-to-instance mapping. The command provides two display options:

- **default** displays a table that lists the instance to VLAN map.
- **digest** displays the configuration digest.

The configuration digest is a 16-byte hex string calculated from the md5 encoding of the VLAN-to-instance mapping table. Switches with identical mappings have identical digests.

Command Mode

EXEC

Command Syntax

show spanning-tree mst configuration [*INFO_LEVEL*]

Parameters

- ***INFO_LEVEL*** specifies data provided by the output. Options include:
  - <no parameter> command displays VLAN-to-instance map.
  - **digest** command displays the MST configuration digest.

Examples

- This command displays the MST region's VLAN-to-instance map.

```
switch>show spanning-tree mst configuration
Name      []
Revision  0    Instances configured 3

Instance  Vlans mapped
--------
---------------------------------------------------------------------
0         1,4-4094
2         2
3         3
---------------------------------------------------------------------------
-
switch>
```

- This command displays the MST region's configuration digest.

```
switch>show spanning-tree mst configuration digest
Name      []
Revision  0    Instances configured 1
Digest         0xAC36177F50283CD4B83821D8AB26DE62
switch>
```

show spanning-tree mst interface

The **show spanning-tree mst interface** command displays Multiple Spanning Tree Protocol (MSTP) information for a specified interface on the specified MST instances.

Command Mode

EXEC

Command Syntax

show spanning-tree mst [*INSTANCE*] interface *INT_NAME* [*INFO_LEVEL*]

Parameters

- ***INSTANCE*** MST instance for which the command displays information. Options include:
  - <no parameter> all MST instances.
  - *mst_inst* denotes a single MST instance. Value of *mst_inst* ranges from 0 to 4094.
- ***INT_NAME*** Interface type and number. Values include:
  - **ethernet** *e_num* Ethernet interface specified by *e_num*.
  - **peerethernet***e_num* Ethernet interface specified by *e_num*.
  - **port-channel** *p_num* Port-channel interface specified by *p_num*.
  - **peerport-channel***p_num* Port-channel interface specified by *p_num*.
- ***INFO_LEVEL*** specifies level of detail provided by the output. Options include:
  - <no parameter> command displays a table of STP instance data for the specified interface
  - **detail** command displays a data block for all specified instance-interface combinations.

Examples

- This command displays an table of STP instance data for Ethernet 1 interface:

```
switch>show spanning-tree mst interface ethernet 1
Ethernet1 of MST0 is root forwarding
Edge port: no                              bpdu guard: disabled
Link type: point-to-point
Boundary : Internal
Bpdus sent 2120, received 2164, taggedErr 0, otherErr 0

Instance Role Sts Cost      Prio.Nbr Vlans mapped
-------- ---- --- --------- -------- ------------------------------
0        Root FWD 2000      128.1    1,4-4094
2        Desg FWD 2000      128.1    2
3        Root FWD 2000      128.1    3
```

- This command displays blocks of STP instance information for Ethernet 1 interface.

```
switch>show spanning-tree mst 3 interface ethernet 1 detail
Edge port: no                              bpdu guard: disabled
Link type: point-to-point
Boundary : Internal
Bpdus sent 2321, received 2365, taggedErr 0, otherErr 0

Ethernet1 of MST3 is root forwarding
Vlans mapped to MST3 3
Port info             port id          128.1  priority    128 cost        2000
Designated root       address 0011.2233.4401  priority  32768 cost           0
Designated bridge     address 0011.2233.4401  priority  32768 port id    128.1
```

show spanning-tree mst test information

The **show spanning-tree mst test information** displays diagnostic spanning tree protocol information.

Command Mode

EXEC

Command Syntax

show spanning-tree mst test information

Examples

- This command displays diagnostic STP information.

```
switch>show spanning-tree mst test information
bi = MstInfo.BridgeInfo( "dut" )
bi.stpVersion = "rstp"
bi.mstpRegionId = ""
bi.bridgeAddr = "00:1c:73:01:60:17"
si = MstInfo.BridgeStpiInfo( "Mst" )
bi.stpiInfoIs( "Mst", si )
si.cistRoot = Tac.Value( "Stp::BridgeId", priority=32768, systemId=0,
address='00:1c:73:01:60:17' )
si.cistPathCost = 0
bmi = MstInfo.BridgeMstiInfo( "Mst0" )
bmi.bridgeId = Tac.Value( "Stp::BridgeId", priority=32768, systemId=0,
address='00:1c:73:01:60:17' )
bmi.designatedRoot = Tac.Value( "Stp::BridgeId", priority=32768, systemId=0,
address='00:1c:73:01:60:17' )
si.mstiInfoIs( "Mst0", bmi )
bmii = MstInfo.BridgeMstiIntfInfo( "Mst0", "Ethernet15" )
bmii.portId = Tac.Value( "Stp::PortId",
                    portPriority=128, portNumber=15 )
bmii.role = "designated"
bmii.operIntPathCost = 2000
bmii.fdbFlush = 1
bmi.mstiIntfInfoIs( "Ethernet15", bmii )
bii = MstInfo.BridgeIntfInfo( "Ethernet15" )
bii.operExtPathCost = 2000
si.intfInfoIs( "Ethernet15", bii )
bmii = MstInfo.BridgeMstiIntfInfo( "Mst0", "Port-Channel10" )
bmii.portId = Tac.Value( "Stp::PortId",
                    portPriority=128, portNumber=101 )
bmii.role = "designated"
bmii.operIntPathCost = 1999
bmii.fdbFlush = 1
bmi.mstiIntfInfoIs( "Port-Channel10", bmii )
bii = MstInfo.BridgeIntfInfo( "Port-Channel10" )
bii.operExtPathCost = 1999
si.intfInfoIs( "Port-Channel10", bii )
switch>
```

show spanning-tree root

The **show spanning-tree root** command displays the Bridge-ID, cost to the root bridge, root port, and the root bridge timer settings for all instances.

Command Mode

EXEC

Command Syntax

show spanning-tree root [*INFO_LEVEL*]

Parameters

- ***INFO_LEVEL*** specifies output format. Options include:
  - <no parameter> output displays data in tabular format.
  - **detail** output displays a data block for each instance.

Examples

- This command displays a table of root bridge information.

```
switch>show spanning-tree root
                   Root ID          Root    Hello  Max  Fwd
Instance     Priority    MAC addr   Cost    Time   Age  Dly  Root Port
----------   ------------------- --------- -----  ---  ---  ------------
MST0         32768 001c.7301.23de         0    2    20  15   Po937
MST101       32869 001c.7301.23de      3998    0     0   0   Po909
MST102       32870 001c.7301.23de      3998    0     0   0   Po911
switch>
```

- This command displays root bridge data blocks for each MSTP instance.

```
switch>show spanning-tree root detail
MST0
MST0
  Root ID    Priority    32768
             Address     001c.7301.23de
             Cost        0 (Ext) 3998 (Int)
             Port        100 (Port-Channel937)
             Hello Time  2.000 sec  Max Age 20 sec  Forward Delay 15 sec
MST101
  Root ID    Priority    32869
             Address     001c.7301.23de
             Cost        3998
             Port        107 (Port-Channel909)
             Hello Time  0.000 sec  Max Age  0 sec  Forward Delay  0 sec
MST102
  Root ID    Priority    32870
             Address     001c.7301.23de
             Cost        3998
             Port        104 (Port-Channel911)
             Hello Time  0.000 sec  Max Age  0 sec  Forward Delay  0 sec
switch>
```

show spanning-tree topology status

The **show spanning-tree topology status** command displays the forwarding state of ports on the specified VLANs.

Command Mode

EXEC

Command Syntax

show spanning-tree topology [*VLAN_NAME*] status [*INFO_LEVEL*]

Parameters

- ***VLAN_NAME*** specifies the VLANs that the output displays. Options include:
  - <no parameter> output includes all VLANs.
  - **vlan** output includes all VLANs.
  - **vlan** *v_num* command includes specified VLAN; *v_num* ranges from 1 to 4094.
- ***INFO_LEVEL*** specifies information provided by output. Options include:
  - <no parameter> output lists forwarding state of interfaces.
  - **detail** output lists forwarding state and change history of interfaces.

Examples

- This command displays forwarding state for ports mapped to all VLANs.

```
switch>show spanning-tree topology status
Topology: Cist
  Mapped Vlans: 1-4,666,1000-1001,1004-1005
  Cpu:               forwarding
  Ethernet2:         forwarding
  Ethernet3:         forwarding
  Ethernet4:         forwarding
  Ethernet5:         forwarding
  Ethernet6:         forwarding
  Ethernet8:         forwarding
  Ethernet10:        forwarding
  Port-Channel1:     forwarding
  Port-Channel2:     forwarding
  Port-Channel3:     forwarding

switch>
```

- This command displays forwarding state and history for ports mapped to VLAN 1000.

```
switch>show spanning-tree topology vlan 1000 status detail
Topology: Cist
  Mapped Vlans: 1000
  Cpu:               forwarding (1 changes, last 23 days, 22:54:43 ago)
  Ethernet2:         forwarding (3 changes, last 23 days, 22:48:59 ago)
  Ethernet4:         forwarding (3 changes, last 10 days, 19:54:17 ago)
  Ethernet5:         forwarding (3 changes, last 23 days, 22:54:38 ago)
  Ethernet6:         forwarding (3 changes, last 19 days, 15:49:10 ago)
  Ethernet10:        forwarding (3 changes, last 9 days, 7:37:05 ago)
  Port-Channel1:     forwarding (3 changes, last 23 days, 22:54:34 ago)
  Port-Channel3:     forwarding (5 changes, last 21 days, 4:56:41 ago)

switch>
```

shutdown (Loop-protection)

The **shutdown (Loop-protection)** command disables loop protection globally. The feature is disabled by default, and is enabled by using the **no shutdown (Loop-protection)** command.

Note To function, loop protection must also be enabled on a per-VLAN basis using the **protect vlan** command.

Command Mode

Loop-protection Configuration

Command Syntax

```
shutdown
no shutdown
```

Example

- This command enables loop protection globally on the switch.

```
switch(config-monitor-loop-protect)#no shutdown
switch(config-monitor-loop-protect)#
```

spanning-tree bpdufilter

The **spanning-tree bpdufilter** command controls bridge protocol data unit (BPDU) filtering on the configuration mode interface. BPDU filtering is disabled by default.

- **spanning-tree bpdufilter enabled** enables BPDU filtering.
- **spanning-tree bpdufilter disabled** disables BPDU filtering by removing the **spanning-tree bpdufilter** command from ***running-config***.

The BPDU filter default setting for portfast ports is configured by the **spanning-tree portfast bpdufilter default** command; BPDU filter is disabled by default on all non-portfast ports.

The **no spanning-tree bpdufilter** and **default spanning-tree bpdufilter** commands restore the global BPDU filter setting on the configuration mode interface by removing the corresponding **spanning-tree bpdufilter** command from ***running-config.***

Command Mode

Interface-Ethernet Configuration
Interface-Port-Channel Configuration

Command Syntax

```
spanning-tree bpdufilter FILTER_STATUS
no spanning-tree bpdufilter
default spanning-tree bpdufilter
```

Parameters

- ***FILTER_STATUS*** BPDU filtering status. Options include:
  - **enabled** BPDU filter is enabled on the interface.
  - **disabled** BPDU filter is disabled on the interface.

Examples

- This command enables BPDU filtering on Ethernet 5 interface.

```
switch(config)#interface ethernet 5
switch(config-if-Et5)#spanning-tree bpdufilter enabled
switch(config-if-Et5)#
```

spanning-tree bpduguard

The **spanning-tree bpduguard** command controls BPDU guard on the configuration mode interface. A BPDU guard-enabled port is disabled when it receives a BPDU packet.

The BPDU guard default setting for portfast ports is configured by the **spanning-tree portfast bpduguard default** command; BPDU guard is disabled by default on all non-portfast ports.

The **no spanning-tree bpduguard** and **default spanning-tree bpduguard** commands restore the global BPDU guard setting on the configuration mode interface by removing the corresponding **spanning-tree bpduguard** command from ***running-config.***

Command Mode

Interface-Ethernet Configuration
Interface-Port-Channel Configuration

Command Syntax

```
spanning-tree bpduguard GUARD_ACTION
no spanning-tree bpduguard
default spanning-tree bpduguard
```

Parameters

- ***GUARD_ACTION*** BPDU guard setting. Options include:
  - **disable** Disable bpduguard
  - **enable** Enable bpduguard
  - **rate-limit** BPDU Input Rate Limiter options

Examples

- These commands enable BPDU guard on Ethernet interface 5.

```
switch(config)#interface ethernet 5
switch(config-if-Et5)#spanning-tree bpduguard enabled
switch(config-if-Et5)
```

spanning-tree bpduguard rate-limit count (global)

The **spanning-tree bpduguard rate-limit count** command sets the maximum BPDU reception rate (quantity per interval) for ports not covered by a **spanning-tree bpduguard rate-limit count (interface)** command.

- The default quantity is 10 times the number of VLANs.
- The default interval is the hello time (**spanning-tree hello-time**).

BPDU rate limiting restricts the number of BPDUs that ports with BPDU guard or BPDU filter disabled can process during a specified interval. Ports discard BPDUs it receives in excess of the specified limit. BPDU rate limiting is enabled or disabled by **spanning-tree bpduguard rate-limit enable / disable** commands.

The **no spanning-tree bpduguard rate-limit count** and **default spanning-tree bpduguard rate-limit count** commands restore the global setting to its default value by removing the **spanning-tree bpduguard rate-limit count** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
spanning-tree bpduguard rate-limit count max_bpdu [TIMER]
no spanning-tree bpduguard rate-limit count
default spanning-tree bpduguard rate-limit count
```

Parameters

- *max_bpdu* BPDU quantity. Value ranges from 1 to 20000.
- ***TIMER*** BPDU reception interval (seconds). Options include:
  - <no parameter> reception interval defaults to ***hello-time***.
  - **interval** *period* Value of *period* ranges from 1 to 15.

Example

- This command configures the global rate limit as 5000 BPDUs per four second period.

```
switch(config)#spanning-tree bpduguard rate-limit count 5000 interval 4
switch(config)#
```

spanning-tree bpduguard rate-limit count (interface)

The **spanning-tree bpduguard rate-limit count** command configures the maximum BPDU reception rate for the configuration mode interface. The default rate limit is specified by the **spanning-tree bpduguard rate-limit count (global)** command.

BPDU rate limiting restricts the number of BPDUs that ports with BPDU guard or BPDU filter disabled can process during a specified interval. Ports discard BPDUs it receives in excess of the specified limit. BPDU rate limiting is enabled or disabled by **spanning-tree bpduguard rate-limit enable / disable** commands.

The **no spanning-tree bpduguard rate-limit count** and **default spanning-tree bpduguard rate-limit count** commands restore the interface value to the global setting by removing the corresponding **spanning-tree bpduguard rate-limit count** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration
Interface-Port-Channel Configuration

Command Syntax

```
spanning-tree bpduguard rate-limit count max_bpdu [TIMER]
no spanning-tree bpduguard rate-limit count
default spanning-tree bpduguard rate-limit count
```

Parameters

- *max_bpdu* BPDU quantity. Value ranges from 1 to 20,000.
- ***TIMER*** BPDU reception interval (seconds). Options include:
  - <no parameter> reception interval defaults to ***hello-time***.
  - **interval** *period* Value of *period* ranges from 1 to 15.

Example

- These commands configure rate limit as 7500 BPDUs per 8 second period on Ethernet 2.

```
switch(config)#interface ethernet 2
switch(config-if-Et2)#spanning-tree bpduguard rate-limit count 7500 interval 8
switch(config-if-Et2)#
```

spanning-tree bpduguard rate-limit default

The **spanning-tree bpduguard rate-limit default** command configures the global BPDU rate limit setting. The global BPDU rate limit setting provides the default for individual ports whose configuration does not include a **spanning-tree bpduguard rate-limit enable / disable** command. The default global setting is ***enabled***.

BPDU rate limiting restricts the number of BPDUs that ports with BPDU guard or BPDU filter disabled can process during a specified interval. Ports discard BPDUs it receives in excess of the specified limit. BPDU rate limits are established by **spanning-tree bpduguard rate-limit count (global)** commands.

The **no spanning-tree bpduguard rate-limit default** sets the global BPDU rate limit setting to ***disabled***. The **spanning-tree bpduguard rate-limit default** and **default spanning-tree bpduguard rate-limit default** commands restore the default global rate limit setting to ***enabled*** by removing the **no spanning-tree bpduguard rate-limit default** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
spanning-tree bpduguard rate-limit default
no spanning-tree bpduguard rate-limit default
default spanning-tree bpduguard rate-limit default
```

Example

- This command enables rate limiting on all ports not covered by an interface rate limit command.

```
switch(config)#spanning-tree bpduguard rate-limit default
switch(config)#
```

spanning-tree bpduguard rate-limit enable / disable

These commands enable and disable BPDU rate limiting on the configuration mode interface:

- **spanning-tree bpduguard rate-limit enable** enables BPDU rate limiting.
- **spanning-tree bpduguard rate-limit disable** disables BPDU rate limiting.

The **spanning-tree bpduguard rate-limit default** command enables BPDU rate limiting on all ports not configured with a **spanning-tree bpduguard rate-limit** command.

BPDU rate limiting restricts the number of BPDUs that ports with BPDU guard or BPDU filter disabled can process during a specified interval. Ports discard BPDUs it receives in excess of the specified limit. BPDU rate limits are established by **spanning-tree bpduguard rate-limit count (interface)** commands.

The **no spanning-tree bpduguard rate-limit** and **default spanning-tree bpduguard rate-limit** commands restore the global rate limit setting on the configuration mode interface by removing the corresponding **spanning-tree bpduguard rate-limit** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration
Interface-Port-Channel Configuration

Command Syntax

```
spanning-tree bpduguard rate-limit enable
spanning-tree bpduguard rate-limit disable
no spanning-tree bpduguard rate-limit
default spanning-tree bpduguard rate-limit
```

Example

- These commands enable rate limiting on Ethernet 15.

```
switch(config)#interface ethernet 15
switch(config-if-Et15)#spanning-tree bpduguard rate-limit enable
switch(config-if-Et15)#
```

spanning-tree bridge assurance

The **spanning-tree bridge assurance** command enables bridge assurance globally, which enables bridge assurance on all ports with a port type of ***network***. Bridge assurance protects against unidirectional link failure, other software failure, and devices that quit running a spanning tree algorithm.

Bridge assurance is available only on point-to-point links on spanning tree ***network*** ports. Both ends of the link should ideally have bridge assurance enabled. Bridge assurance-enabled ports will not necessarily be blocked when they link to a port where bridge assurance is not enabled, but if they do not receive periodic BPDUs from the other side of the link the **show spanning-tree bridge assurance** command will show a bridge assurance status of “inconsistent” (blocking) for that port.

The **no spanning-tree bridge assurance** command disables bridge assurance.

The **spanning-tree bridge assurance** and **default spanning-tree bridge assurance** commands restore the default behavior by removing the **no spanning-tree bridge assurance** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
spanning-tree bridge assurance
no spanning-tree bridge assurance
default spanning-tree bridge assurance
```

Example

- This command enables bridge assurance on the switch.

```
switch(config)#spanning-tree bridge assurance
switch(config)#
```

spanning-tree cost

The **spanning-tree cost** command configures the path cost of the configuration mode interface. Cost values range from 1 to 200000000 (200 million). The default cost depends on the interface speed:

- 1 gigabit interface: cost = 20000
- 10 gigabit interface: cost = 2000

The **spanning-tree cost** command provides a mode option:

- RST instance cost is configured by not including a mode.
- MST instance 0 cost is configured by not including a mode or with the **mst** mode option.
- MST instance cost is configured with the **mst** mode option.
- Rapid-PVST VLAN cost is configured with the **vlan** mode option.

The **no spanning-tree cost** and **default spanning-tree cost** commands restore the default cost on the configuration mode interface by removing the corresponding **spanning-tree cost** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration
Interface-Port-Channel Configuration

Command Syntax

```
spanning-tree MODE cost value
no spanning-tree MODE cost
default spanning-tree MODE cost
```

Parameters

- ***MODE*** specifies the spanning tree instances for which the cost is configured. Values include:
  - <no parameter> RST instance, MST instance 0, or all Rapid-PVST instances permitted on the interface.
  - **mst** *m_range* specified MST instances. *m_range* formats include a number, number range, or comma-delimited list of numbers and ranges. Instance numbers range from 0 to 4094.
  - **vlan** *v_range* specified Rapid-PVST instances. *v_range* formats include a number, number range, or comma-delimited list of numbers and ranges. VLAN numbers range from 1 to 4094.
- *value* path cost assigned to interface. Values range from 1 to 200000000 (200 million). Default values are 20000 (1 G interfaces) or 2000 (10 G interfaces).

Examples

- These commands configure a port cost of 25000 for Ethernet interface 5 when configured as an RST port, as a port in MST instance 0, or all unconfigured Rapid-PVST instances that are not explicitly configured.

```
switch(config)#interface ethernet 5
switch(config-if-Et5)#spanning tree cost 25000
```

- This command configures a port cost of 30000 for Ethernet interface 5 when configured as a port in MST instance 200.

```
switch(config-if-Et5)#spanning tree mst 200 cost 30000
```

- This command configures a port cost of 100000 for Ethernet interface 5 when configured as a port in VLANs 200-220.

```
switch(config-if-Et5)#spanning tree vlan 200-220 cost 100000
switch(config-if-Et5)#
```

spanning-tree forward-time

The **spanning-tree forward-time** command configures the forward delay timer. Forward delay is the time that a port is in learning state before it begins forwarding data packets.

The switch inserts the forward delay timer value in BPDU packets it sends as the root bridge. The forward delay value ranges from 4 to 30 seconds with a default of 15 seconds.

The **no spanning-tree forward-time** and **default spanning-tree forward-time** commands restore the forward delay timer default of 15 seconds by removing the **spanning-tree forward-time** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
spanning-tree forward-time period
no spanning-tree forward-time
default spanning-tree forward-time
```

Parameters

- *period* forward delay timer (seconds). Value ranges from 4 to 30. Default is 15.

Examples

- This command sets the forward delay timer value to 25 seconds.

```
switch(config)#spanning-tree forward-time 25
switch(config)#
```

spanning-tree guard

The **spanning-tree guard** command enables root guard or loop guard on the configuration mode interface. The **spanning-tree loopguard default** command configures the global loop guard setting.

- Root guard prevents a port from becoming a root or blocked port. A root guard port that receives a superior BPDU transitions to the root-inconsistent (blocked) state.
- Loop guard protects against loops resulting from unidirectional link failures on point-to-point links by preventing non-designated ports from becoming designated ports. When loop guard is enabled, a root or blocked port transitions to loop-inconsistent (blocked) state if it stops receiving BPDUs from its designated port. The port returns to its prior state when it receives a BPDU.

The **no spanning-tree guard** and **default spanning-tree guard** commands sets the configuration mode interface to the global loop guard mode by removing the **spanning-tree guard** statement from ***running-config***. The **spanning-tree guard none** command disables loop guard and root guard on the interface, overriding the global setting.

Command Mode

Interface-Ethernet Configuration
Interface-Port-Channel Configuration

Command Syntax

```
spanning-tree guard PORT_MODE
no spanning-tree guard
default spanning-tree guard
```

Parameters

- ***PORT_MODE*** the port mode. Options include:
  - **loop** enables loop guard on the interface.
  - **root** enables root guard on the interface.
  - **none** disables root guard and loop guard.

Examples

- This command enables root guard on Ethernet 5 interface.

```
switch(config)#interface ethernet 5
switch(config-if-Et5)#spanning-tree guard root
switch(config-if-Et5)#
```

spanning-tree hello-time

The **spanning-tree hello-time** command configures the hello time, which specifies the transmission interval between consecutive bridge protocol data units (BPDU) that the switch sends as a root bridge. The hello time is also inserted in outbound BPDUs.

This hello time ranges from 0.2 seconds to 10 seconds with a default of 2 seconds.

The **no spanning-tree hello-time** and **default spanning-tree hello-time** commands restore the hello time default of 2 seconds by removing the **spanning-tree hello-time** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
spanning-tree hello-time period
no spanning-tree hello-time
default spanning-tree hello-time
```

Parameters

- *period* hello-time (milliseconds). Value ranges from 200 to 10000. Default is 2000.

Examples

- This command configures a hello-time of one second.

```
switch(config)#spanning-tree hello-time 1000
switch(config)#
```

spanning-tree link-type

The **spanning-tree link-type** command specifies the configuration mode interface's link type, which is normally derived from the port's duplex setting. The default setting depends on a port's duplex mode:

- full-duplex ports are ***point-to-point***.
- half-duplex ports are ***shared***.

The **no spanning-tree link-type** and **default spanning-tree link-type** commands restore the default link type on the configuration mode interface by removing the corresponding **spanning-tree link-type** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration
Interface-Port-Channel Configuration

Command Syntax

```
spanning-tree link-type TYPE
no spanning-tree link-type
default spanning-tree link-type
```

Parameters

- ***TYPE*** link type of the configuration mode interface. Options include:
  - **point-to-point**
  - **shared**

Examples

- This command configures Ethernet 5 interface as a shared port.

```
switch(config)#interface ethernet 5
switch(config-if-Et5)#spanning-tree link-type shared
switch(config-if-Et5)#
```

spanning-tree loopguard default

The **spanning-tree loopguard default** command configures the global loop guard setting as ***enabled***. Ports not covered by a **spanning-tree guard** command use the global loop guard setting. Loop guard prevents blocked or root ports from becoming a designated port due to failures resulting in a unidirectional link. The **spanning-tree guard** interface configuration statement overrides the global setting for a specified interface. The default global loop guard setting is ***disabled***.

The **no spanning-tree loopguard default** and **default spanning-tree loopguard default** commands restore the global loop guard setting of ***disabled*** by removing the **spanning-tree loopguard default** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
spanning-tree loopguard default
no spanning-tree loopguard default
default spanning-tree loopguard default
```

Examples

- This command enables loop guard as the default on all switch ports.

```
switch(config)#spanning-tree loopguard default
switch(config)#
```

spanning-tree max-age

The **spanning-tree max-age** command configures the switch's max age timer, which specifies the max age value that the switch inserts in outbound BPDU packets it sends as a root bridge. The max-age time value ranges from 6 to 40 seconds with a default of 20 seconds.

Max age is the interval, specified in the BPDU, that BPDU data remains valid after its reception. The bridge recomputes the spanning tree topology if it does not receive a new BPDU before max age expiry.

The **no spanning-tree max-age** and **default spanning-tree max-age** commands restore the max-age default of 20 seconds by removing the **spanning-tree max-age** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
spanning-tree max-age period
no spanning-tree max-age
default spanning-tree max-age
```

Parameters

- *period* max age period (seconds). Value ranges from 6 to 40. Default is 20.

Examples

- This command sets the max age timer value to 25 seconds.

```
switch(config)#spanning-tree max-age 25
switch(config)#
```

spanning-tree max-hops

The **spanning-tree max-hops** command specifies the max hop setting that the switch inserts into BPDUs that it sends out as the root bridge. The max hop setting determines the number of bridges in an MST region that a BPDU can traverse before it is discarded. The max-hop value ranges from 1 to 40 with a default of 20.

The **no spanning-tree max-hops** and **default spanning-tree max-hops** commands restore the max-hops setting to its default value of 20 by removing the **spanning-tree max-hops** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
spanning-tree max-hops ports
no spanning-tree max-hops
default spanning-tree max-hops
```

Parameters

- *ports* max hops (bridges). Value ranges from 1 to 40. Default is 20.

Example

- This command sets the max hop value to 40.

```
switch(config)#spanning-tree max-hop 40
switch(config)#
```

spanning-tree mode

The **spanning-tree mode** command specifies the spanning tree protocol version that the switch runs. The default mode is Multiple Spanning Tree (mstp).

The **no spanning-tree mode** and **default spanning-tree mode** commands restore the default spanning tree protocol version.

Caution The **spanning-tree mode** command may disrupt user traffic. When the switch starts a different STP version, all spanning-tree instances are stopped, then restarted in the new mode.

Command Mode

Global Configuration

Command Syntax

```
spanning-tree mode VERSION
no spanning-tree mode
default spanning-tree mode
```

Parameters

- ***VERSION*** spanning tree version that the switch runs. Options include:
  - **mstp** multiple spanning tree protocol described in the IEEE 802.1Q-2005 specification and originally specified in the IEEE 802.1s specification.
  - **rstp** rapid spanning tree protocol described in the IEEE 802.1D-2004 specification and originally specified in the IEEE 802.1w specification.
  - **rapid-pvst** rapid per-VLAN spanning tree protocol described in the IEEE 802.1D-2004 specification and originally specified in the IEEE 802.1w specification.
  - **backup** disables STP and enables switchport interface pairs configured with the **switchport backup interface** command.
  - **none** disables STP. The switch does not generate STP packets. Each switchport interface forwards data packets to all connected ports and forwards STP packets as multicast data packets on the VLAN where they are received.

Guidelines

Backup mode is not available on Trident platform switches.

Example

- This command configures the switch to run multiple spanning tree protocol.

```
switch(config)#spanning-tree mode mstp
switch(config)#
```

spanning-tree mst configuration

The **spanning-tree mst configuration** command places the switch in MST-configuration mode, which is the group change mode where MST region parameters are configured.

Changes made in a group change mode are saved by leaving the mode through the **exit** command or by entering another configuration mode. To discard changes from the current edit session, leave the mode with the **abort** command.

These commands are available in MST-configuration mode:

- **abort (mst-configuration mode)**
- **exit (mst-configuration mode)**
- **instance**
- **name (mst-configuration mode)**
- **revision (mst-configuration mode)**
- **show (mst-configuration mode)**

The **no spanning-tree mst configuration** and **default spanning-tree mst configuration** commands restore the MST default configuration.

Command Mode

Global Configuration

Command Syntax

```
spanning-tree mst configuration
no spanning-tree mst configuration
default spanning-tree mst configuration
```

Examples

- This command enters MST configuration mode.

```
switch(config)#spanning-tree mst configuration
switch(config-mst)#
```

- This command exits MST configuration mode, saving MST region configuration changes to ***running-config***.

```
switch(config-mst)#exit
switch(config)#
```

- This command exits MST configuration mode without saving MST region configuration changes to ***running-config***.

```
switch(config-mst)#abort
switch(config)#
```

spanning-tree portchannel guard misconfig

The **spanning-tree portchannel guard misconfig** command enables the switch to detect misconfigured port channels that may cause network loops by monitoring inbound BPDUs. When a port channel receives 75 inconsistent BPDUs within 30 seconds, the switch error disables the port. When a port channel receives 5 BPDUs with the same source BPDU during the 30 second measurement interval, the error counter is reset and the port continues normal port channel operation. Misconfigured port channel detection is disabled by default.

The **no spanning-tree portchannel guard misconfig** and **default spanning-tree portchannel guard misconfig** commands disables the detection of misconfigured port channels by removing the **spanning-tree portchannel guard misconfig** statement from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
spanning-tree portchannel guard misconfig
no spanning-tree portchannel guard misconfig
default spanning-tree portchannel guard misconfig

spanning-tree etherchannel guard misconfig
no spanning-tree etherchannel guard misconfig
default spanning-tree etherchannel guard misconfig
```

Guidelines

The **spanning-tree portchannel guard misconfig** and **spanning-tree etherchannel guard misconfig** commands are equivalent.

Examples

- This command enables port channel misconfiguration detection on the switch.

```
switch(config)#spanning-tree portchannel guard misconfig
switch(config)#show running-config

!
spanning-tree mode mstp
spanning-tree portchannel guard misconfig
!
              <-------OUTPUT OMITTED FROM EXAMPLE-------->
!
end
switch(config)#
```

- This command disables port channel misconfiguration detection on the switch.

```
switch(config)#no spanning-tree portchannel guard misconfig
switch(config)#show running-config
              <-------OUTPUT OMITTED FROM EXAMPLE-------->
!
spanning-tree mode mstp
!
              <-------OUTPUT OMITTED FROM EXAMPLE-------->
!
end
switch(config)#
```

spanning-tree portfast

The **spanning-tree portfast** command programs configuration mode ports to immediately enter forwarding state when they establish a link. PortFast ports are included in spanning tree topology calculations and can enter discarding state. This command overrides the **spanning-tree portfast auto** command.

The **no spanning-tree portfast** and **default spanning-tree portfast** commands remove the corresponding **spanning-tree portfast** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration
Interface-Port-Channel Configuration

Command Syntax

```
spanning-tree portfast
no spanning-tree portfast
default spanning-tree portfast
```

Example

- This command unconditionally enables portfast on Ethernet 5.

```
switch(config)#interface ethernet 5
switch(config-if-Et5)#spanning-tree portfast
switch(config-if-Et5)#
```

spanning-tree portfast auto

The **spanning-tree portfast auto** command enables auto-edge detection on the configuration mode interface. When auto-edge detection is enabled, the port is configured as an edge port if it does not receive a new BPDU before the current BPDU expires. Auto-edge detection is enabled by default. The **spanning-tree portfast** command, when configured, has priority over this command.

The **no spanning-tree portfast auto** command disables auto-edge port detection. This command is removed from ***running-config*** with the **spanning-tree portfast auto** and **default spanning-tree portfast auto** commands.

Command Mode

Interface-Ethernet Configuration
Interface-Port-Channel Configuration

Command Syntax

```
spanning-tree portfast auto
no spanning-tree portfast auto
default spanning-tree portfast auto
```

Example

- This command enables auto-edge detection on Ethernet interface 5.

```
switch(config)#interface ethernet 5
switch(config-if-Et5)#spanning-tree portfast auto
switch(config-if-Et5)#
```

spanning-tree portfast bpdufilter default

The **spanning-tree portfast bpdufilter default** command configures the global BPDU filter setting as ***enabled***. Ports not covered by a **spanning-tree bpdufilter** command use the global BPDU filter setting.

Command Mode

Global Configuration

Command Syntax

```
spanning-tree portfast bpdufilter default
no spanning-tree portfast bpdufilter default
default spanning-tree portfast bpdufilter default
```

Example

- This command configures the BPDU filter global setting to ***enabled***.

```
switch(config)#spanning-tree portfast bpdufilter default
switch(config)#
```

spanning-tree portfast bpduguard default

The **spanning-tree portfast bpduguard default** command sets the global BPDU guard setting as ***enabled***. Ports not covered by a **spanning-tree bpduguard** command use the global BPDU guard setting.

Command Mode

Global Configuration

Command Syntax

```
spanning-tree portfast bpduguard default
no spanning-tree portfast bpduguard default
default spanning-tree portfast bpduguard default
```

Example

- This command configures the global BPDU guard setting to ***enabled***.

```
switch(config)#spanning-tree portfast bpduguard default
switch(config)#
```

spanning-tree portfast <port type>

The **spanning-tree portfast <port-type>** command specifies the STP port mode for the configuration mode interface. Default port mode is ***normal***.

Port modes include:

- ***Edge***: Edge ports connect to hosts and transition to the forwarding state when the link is established. An edge port that receives a BPDU becomes a normal port.
- ***Network***: Network ports connect only to switches or bridges and support bridge assurance. Network ports that connect to hosts or other edge devices transition to the discarding state.
- ***Normal***: Normal ports function as normal STP ports and can connect to any type of device.

The **no spanning-tree portfast <port-type>** and **default spanning-tree portfast <port-type>** commands restore the default port mode of normal by removing the corresponding **spanning-tree portfast <port-type>** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration
Interface-Port-Channel Configuration

Command Syntax

```
spanning-tree portfast PORT_MODE
no spanning-tree portfast PORT_MODE
default spanning-tree portfast PORT_MODE
```

Parameters

- ***PORT_MODE*** STP port mode. Options include:
  - **edge**
  - **network**
  - **normal**

    The **normal** option is not available for the **no** and **default** commands.

Related Commands

The **spanning-tree portfast <port-type> command** also affects the **spanning-tree portfast auto** and **spanning-tree portfast** configuration for the configuration mode interface:

- **spanning-tree portfast normal**: **spanning-tree portfast auto** is enabled.
- **spanning-tree portfast edge**: **spanning-tree portfast** is enabled.
- **spanning-tree portfast network**: **spanning-tree portfast auto** is disabled.

Example

- This command configures Ethernet 5 interface as a network port.

```
switch(config)#interface ethernet 5
switch(config-if-Et5)#spanning-tree portfast network
switch(config-if-Et5)#
```

spanning-tree port-priority

The **spanning-tree port-priority** command specifies the configuration mode interface's port-priority number. The switch uses this number to determine which interface it places into forwarding mode when resolving a loop. Valid settings are all multiples of 16 between 0 and 240. Default value is 128. Ports with lower numerical priority values are selected over other ports.

The **no spanning-tree port-priority** and default spanning-tree port-priority commands restore the default of 128 for the configuration mode interface by removing the **spanning-tree port-priority** command from ***running-config***.

The **spanning-tree port-priority** command provides a mode option:

- RST instance port-priority is configured by not including a mode.
- MST instance 0 port-priority is configured by not including a mode or with the **mst** mode option.
- MST instance port-priority is configured with the **mst** mode option.
- Rapid-PVST VLAN port-priority is configured with the **vlan** mode option.

Command Mode

Interface-Ethernet Configuration
Interface-Port-Channel Configuration

Command Syntax

```
spanning-tree [MODE] port-priority value
no spanning-tree [MODE] port-priority
default spanning-tree [MODE] port-priority
```

Parameters

- ***MODE*** specifies the spanning tree instances for which the cost is configured. Values include:
  - <no parameter> RST instance or MST instance 0.
  - **mst** *m_range* specified MST instances. *m_range* formats include a number, number range, or comma-delimited list of numbers and ranges. Instance numbers range from 0 to 4094.
  - **vlan** *v_range* specified Rapid-PVST instances. *v_range* formats include a number, number range, or comma-delimited list of numbers and ranges. VLAN numbers range from 1 to 4094.
- *value* bridge priority number. Values range from 0 to 240 and must be a multiple of 16.

Example

- This command sets the port priority of Ethernet 5 interface to 144.

```
switch(config)#interface ethernet 5
switch(config-if-Et5)#spanning-tree port-priority 144
switch(config-if-Et5)#
```

spanning-tree priority

The **spanning-tree priority** command configures the bridge priority number. The bridge priority is the four most significant digits of the bridge ID, which is used by spanning tree algorithms to select the root bridge and choose among redundant links. Bridge ID numbers range from 0 to 65535 (16 bits); bridges with smaller bridge IDs are elected over other bridges.

Because bridge priority sets the four most significant bits of the bridge ID, valid settings include all multiples of 4096 between 0 and 61440. Default value is 32768.

The **spanning-tree priority** command provides a mode option:

- RST instance priority is configured by not including a mode.
- MST instance 0 priority is configured by not including a mode or with the **mst** mode option.
- MST instance priority is configured with the **mst** mode option.
- Rapid-PVST VLAN priority is configured with the **vlan** mode option.

The **no spanning-tree priority** and **default spanning-tree priority** commands restore the bridge priority default of 32768 for the specified mode by removing the corresponding **spanning-tree priority** command from ***running-config***.

Another method of adding **spanning-tree priority** commands to the configuration is through the **spanning-tree root** command. Similarly, the **no spanning-tree root** command removes the corresponding **spanning-tree priority** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
spanning-tree [MODE] priority level
no spanning-tree [MODE] priority
default spanning-tree [MODE] priority
```

Parameters

- ***MODE*** spanning tree instances for which the command configures priority. Options include:
  - <no parameter> RST instance or MST instance 0.
  - **mst** *m_range* specified MST instances. *m_range* formats include a number, number range, or comma-delimited list of numbers and ranges. Instance numbers range from 0 to 4094.
  - **vlan** *v_range* specified Rapid-PVST instances. *v_range* formats include a number, number range, or comma-delimited list of numbers and ranges. VLAN numbers range from 1 to 4094.
- *level* priority number. Values include multiples of 4096 between 0 and 61440. Default is 32768.

Examples

- This command configures a bridge priority value of 20480 for Rapid-PVST VLANs 20, 24, 28, and 32.

```
switch(config)#spanning-tree vlan 20,24,28,32 priority 20480
switch(config)#
```

- This command configures a bridge priority value of 36864 for the RST instance. When MST is enabled, this command configures a priority of 36864 for MST instance 0.

```
switch(config)#spanning-tree priority 36864
switch(config)#
```

spanning-tree root

The **spanning-tree root** command configures the bridge priority number by adding a **spanning-tree priority** command to the configuration. Parameter settings set the following priority values:

- ***primary*** sets the bridge priority to 8192.
- ***secondary*** sets the bridge priority to 16384.

The bridge priority is the four most significant digits of the bridge ID, which is used by spanning tree algorithms to select the root bridge and choose among redundant links. Bridge ID numbers range from 0 to 65535 (16 bits); bridges with smaller bridge IDs are elected over other bridges.

When no other switch in the network is similarly configured, assigning the primary value to the switch facilitates its selection as the root switch. Assigning the secondary value to the switch facilitates its selection as the backup root in a network that contains one switch with a smaller priority number.

The **spanning-tree root** command provides a mode option:

- RST instance priority is configured by not including a mode.
- MST instance 0 priority is configured by not including a mode or with the **mst** mode option.
- MST instance priority is configured with the **mst** mode option.
- Rapid-PVST VLAN priority is configured with the **vlan** mode option.

The **no spanning-tree root** and **default spanning-tree root** commands restore the bridge priority default of 32768 by removing the corresponding **spanning-tree priority** command from ***running-config***. The **no spanning-tree root**, **no spanning-tree priority, default spanning-tree root** and **default spanning-tree priority** commands perform the same function.

Command Mode

Global Configuration

Command Syntax

```
spanning-tree [MODE] root TYPE
no spanning-tree [MODE] root
default spanning-tree [MODE] root
```

Parameters

- ***MODE*** specifies the spanning tree instances for which priority is configured. Values include:
  - <no parameter> RST instance or MST instance 0.
  - **mst** *m_range* specified MST instances. *m_range* formats include a number, number range, or comma-delimited list of numbers and ranges. Instance numbers range from 0 to 4094.
  - **vlan** *v_range* specified Rapid-PVST instances. *v_range* formats include a number, number range, or comma-delimited list of numbers and ranges. VLAN numbers range from 1 to 4094.
- ***TYPE*** sets the bridge priority number. Values include:
  - **primary** sets the bridge priority to 8192.
  - **secondary** sets the bridge priority to 16384.

Examples

- This command configures a bridge priority value of 8192 for Rapid-PVST VLANs 20-36.

```
switch(config)#spanning-tree vlan 20-36 root primary
```

- This command configures a bridge priority value of 16384 for the RSTP instance and MST instance 0.

```
switch(config)#spanning-tree root secondary
```

spanning-tree transmit hold-count

The **spanning-tree transmit hold-count** command specifies the maximum number of BPDUs per second that the switch can send from an interface. Valid settings range from 1 to 10 BPDUs with a default of 6 BPDUs.

The **no spanning-tree transmit hold-count** and **default spanning-tree transmit hold-count** commands restore the transmit hold count default of 6 BPDUs by removing the **spanning-tree transmit hold-count** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
spanning-tree transmit hold-count max_bpdu
no spanning-tree transmit hold-count
default spanning-tree transmit hold-count
```

Parameters

- *max_bpdu* BPDU packets. Value ranges from 1 to 10. Default is 6.

Examples

- This command configures a transmit hold-count of 8 BPDUs.

```
switch(config)#spanning-tree transmit hold-count 8
switch(config)#
```

spanning-tree vlan

The **spanning-tree vlan** command enables Spanning Tree Protocol (STP) on specified VLANs by removing the corresponding **no spanning-tree vlan** statements from ***running-config***. Spanning-tree is enabled on all VLANs by default.

The **no spanning-tree vlan** and **default spanning-tree** commands disable STP on the specified interfaces.

Warning Disabling STP is not recommended, even in topologies free of physical loops; STP guards against configuration mistakes and cabling errors. When disabling STP, ensure that there are no physical loops in the VLAN.

Important! When disabling STP on a VLAN, ensure that all switches and bridges in the network disable STP for the same VLAN. Disabling STP on a subset of switches and bridges in a VLAN may have unexpected results because switches and bridges running STP will have incomplete information regarding the network's physical topology.

The following STP global configuration commands provide a ***vlan*** option for configuring Rapid-PVST VLAN instances:

- **spanning-tree priority**
- **spanning-tree root**

Command Mode

Global Configuration

Command Syntax

```
spanning-tree vlan v_range
no spanning-tree vlan v_range
default spanning-tree vlan v_range
```

Parameters

- *v_range* VLAN list. Formats include a number, number range, or comma-delimited list of numbers and ranges. VLAN numbers range from 1 to 4094.

Examples

- This command disables STP on VLANs 200-205

```
switch(config)#no spanning-tree vlan 200-205
switch(config)#
```

- This command enables STP on VLAN 203

```
switch(config)#spanning-tree vlan 203
switch(config)#
```

switchport backup interface

The **switchport backup interface** command establishes a primary-backup configuration for forwarding VLAN traffic between the command mode interface and a specified interface. The **show interfaces switchport backup** command displays the state of backup interface pairs on the switch:

- the primary interface is the command mode interface.
- the backup interface is the interface specified in the command.

The following guidelines apply to primary and backup interfaces.

- Ethernet and Port Channels can be primary interfaces.
- Ethernet, Port Channel, Management, Loopback, and VLANs can be backup interfaces.
- The primary and backup interfaces can be different interface types.
- Interface pairs should be similarly configured to ensure consistent behavior.
- An interface can be associated with a maximum of one backup interface.
- An interface can back up a maximum of one interface.
- Any Ethernet interface configured in an interface pair cannot be a port channel member.
- The STP mode is backup.
- Static MAC addresses should be configured after primary-backup pairs are established.

When load balancing is not enabled, the primary and backup interfaces cannot simultaneously forward VLAN traffic. When the primary interface is forwarding VLAN traffic, the backup interface drops all traffic. If the primary interface fails, the backup interface forwards VLAN traffic until the primary interface is functional.

The ***prefer vlan*** option balances the load across the primary and backup interfaces. When the command includes the ***prefer vlan*** option, each interface is the primary for a subset of the vlans carried by the pair. When both interfaces are up, prefer option vlans are forwarded on the backup interface and all other configured vlans are carried by the primary interface.

The **no switchport backup interface** and **default switchport backup interface** commands remove the primary-backup configuration for the configuration mode interface.

Command Mode

Interface-Ethernet Configuration
Interface-Port Channel Configuration

Command Syntax

```
switchport backup interface INT_NAME [BALANCE]
no switchport backup interface
default switchport backup interface
```

Parameters

- ***INT_NAME*** the backup interface. Options include:
  - **ethernet** *e_num* Ethernet interface specified by *e_num*.
  - **loopback** *l_num* Loopback interface specified by *l_num*.
  - **management** *m_num* Management interface specified by *m_num*.
  - **port-channel** *p_num* Channel group interface specified by *p_num*.
  - **vlan** *v_num* VLAN interface specified by *v_num*.
  - **vxlan** *vx_num* VXLAN interface specified by *vx_num*.
- ***BALANCE*** VLANs whose traffic is normally handled on the backup interfaces. Values include:

- <no parameter> backup interface handles no traffic if the primary interface is operating.
- **prefer vlan** *v_range* list of VLANs whose traffic is handled by backup interface.

Examples

- These commands establish Ethernet interface 7 as the backup port for Ethernet interface 1.

```
switch(config)#interface ethernet 1
switch(config-if-Et1)#switchport backup interface ethernet 7
switch(config-if-Et1)#
```

- These commands configure the following:
  - Ethernet interface 1 as a trunk port that handles VLAN 4 through 9 traffic.
  - Ethernet interface 2 as its backup interface.
  - Ethernet 2 as the preferred interface for VLANs 7 through 9.

```
switch(config-if-Et1)#switchport mode trunk
switch(config-if-Et1)#switchport trunk allowed vlan 4-9
switch(config-if-Et1)#switchport backup Ethernet 2 prefer vlan 7-9
switch(config-if-Et1)#
```

transmit-interval

The **transmit-interval** command sets the interval at which loop detection packets are transmitted. The **no transmit-interval** and **default transmit-interval** commands restore the transmission interval to the default of 5 seconds.

Command Mode

Loop-protection Configuration

Command Syntax

transmit-interval *interval*

Parameters

- *interval* Interval in seconds at which loop-detection packets are transmitted. Values range from 1 to 10; default is 5.

Example

- This command sets the loop detection packet transmission interval to 10 seconds.

```
switch(config-monitor-loop-protect)#transmit-interval 10
switch(config-monitor-loop-protect)#
```

Chapter 23

Quality of Service

This chapter describes Arista's Quality of Service (QoS) implementation, including configuration instructions and command descriptions. Topics covered by this chapter include:

- Section 23.1: Quality of Service Conceptual Overview
- Section 23.2: QoS Configuration: Arad Platform Switches
- Section 23.3: QoS Configuration: Jericho Platform Switches
- Section 23.4: QoS Configuration: FM6000 Platform Switches
- Section 23.5: QoS Configuration: Petra Platform Switches
- Section 23.6: QoS Configuration: Trident Platform Switches
- Section 23.7: QoS Configuration: Trident-II and Helix Platform Switches
- Section 23.8: Quality of Service Configuration Commands

23.1 Quality of Service Conceptual Overview

QoS processes apply to traffic that flows through Ethernet ports and control planes. These processes can modify data fields (CoS or DSCP) or assign data streams to traffic classes for prioritized handling. Transmission queues are configurable for individual Ethernet ports to shape traffic based on its traffic class. Many switches also support traffic policies that apply to data that is filtered by access control lists.

The following sections describe QoS features:

- Section 23.1.1: QoS Data Fields and Traffic Classes
- Section 23.1.2: Transmit Queues and Port Shaping
- Section 23.1.3: Explicit Congestion Notification (ECN)
- Section 23.1.4: ACL Policing

23.1.1 QoS Data Fields and Traffic Classes

Quality of Service defines a method of differentiating data streams to provide varying levels of service to the different streams. Criteria determining a packet's priority level include packet field contents and the port where data packets are received. QoS settings are translated into traffic classes, which are then used by switches to manage all traffic flows. Traffic flow management varies with each switch platform.

23.1.1.1 QoS Data Fields

Quality of service decisions are based on the contents of the following packet fields:

- CoS (three bits): Class of service (CoS) is a 3-bit field in Ethernet frame headers using VLAN tagging. The field specifies a priority value between zero and seven. Class of service operates at layer 2.
- DSCP (six bits): Differentiated Service Code Point (DSCP) is a 6-bit field in the Type Of Service (TOS) field of IP packet headers.

23.1.1.2 Port Settings – Trust Mode and Traffic Class

Ethernet and port channel interfaces support three QoS trust modes:

- CoS Trust: Ports use inbound packet CoS field contents to derive the traffic class.
- DSCP Trust: Ports use inbound packets DSCP field contents to derive the traffic class.
- Untrusted: Ports use their default values to derive the traffic class, ignoring packet contents.

The default mode setting is ***CoS trust*** for switched ports and ***DSCP trust*** for routed ports.

Ports are associated with default CoS, DSCP, and traffic class settings; defaults vary by platform.

These sections describe procedures for configuring port settings:

- Section 23.2.1: CoS and DSCP Port Settings – Arad Platform Switches
- Section 23.4.1: CoS and DSCP Port Settings – FM6000 Platform Switches
- Section 23.5.1: CoS and DSCP Port Settings – Petra Platform Switches
- Section 23.6.1: CoS and DSCP Port Settings – Trident Platform Switches
- Section 23.7.1: CoS and DSCP Port Settings – Trident-II and Helix Platform Switches

23.1.1.3 Rewriting CoS and DSCP

CoS Rewrite

Switches can rewrite the CoS field for outbound tagged packets. The new CoS value is configurable, and is derived from a data stream's traffic class as specified by the traffic class-to-CoS rewrite map. CoS rewrite is disabled on all the traffic received on CoS trusted ports.

On Arad, Jericho, FM6000, Trident, Trident-II, and Helix platform switches, CoS rewrite can be enabled or disabled on DSCP trusted ports and untrusted ports.

- CoS rewrite is globally enabled by default for packets received on untrusted ports and DSCP trusted ports if at least one port is explicitly configured in ***DSCP trust*** or ***untrusted*** mode.
- CoS rewrite is globally disabled by default for packets received on untrusted ports and DSCP trusted ports if there are no ports explicitly configured in ***DSCP trust*** or ***untrusted*** mode.

On Petra platform switches, CoS rewrite is always enabled on DSCP trusted ports and untrusted ports.

DSCP Rewrite

Switches can rewrite the DSCP field for outbound IP packets. On FM6000, Trident, Trident-II, and Helix platform switches, DSCP rewrite is disabled by default on all ports and always disabled for traffic received on DSCP trusted ports. On Petra, Arad, and Jericho platform switches, DSCP rewrite is always disabled.

FM6000, Trident, Trident-II, and Helix platform switches provide a command that enables or disables DSCP rewrite for packets received on CoS trusted ports and untrusted ports. The new DSCP value is configurable, based on the data stream's traffic class, as specified by the traffic class-to-DSCP rewrite map.

These sections describe procedures for rewriting CoS and DSCP fields:

- Section 23.2.3: CoS Rewrite – Arad Platform Switches
- Section 23.4.3: CoS and DSCP Rewrite – FM6000 Platform Switches
- Section 23.5.3: CoS Rewrite – Petra Platform Switches
- Section 23.6.3: CoS and DSCP Rewrite – Trident Platform Switches
- Section 23.7.3: CoS and DSCP Rewrite – Trident-II and Helix Platform Switches

23.1.1.4 Traffic Classes

Data stream distribution is based on their traffic classes. Data stream management varies by switch platform. Traffic classes are derived from these data stream, inbound port, and switch attributes:

- CoS field contents
- DSCP field contents
- Inbound port trust setting
- CoS default setting (Arad, Jericho, FM6000, Trident, Trident-II, and Helix platform switches)
- DSCP default setting (Arad, Jericho, FM6000, Trident, and Trident-II platform switches)
- Traffic class default setting (Petra platform switches)

When a port is configured to derive a data stream's traffic class from the CoS or DSCP value associated with the stream, the traffic class is determined from a conversion map.

- A CoS-to-traffic class map derives a traffic class from a CoS value.
- A DSCP-to-traffic class map derives a traffic class from a DSCP value.

Map entries are configurable through CLI commands. Default maps determine the traffic class value when CLI map entry commands are not configured. Default maps vary by switch platform.

These sections describe traffic class configuration procedures:

- Section 23.2.2: Traffic Class Derivations – Arad Platform Switches
- Section 23.3.2: Traffic Class Derivations – Jericho Platform Switches
- Section 23.4.2: Traffic Class Derivations – FM6000 Platform Switches
- Section 23.5.2: Traffic Class Derivations – Petra Platform Switches
- Section 23.6.2: Traffic Class Derivations – Trident Platform Switches
- Section 23.7.2: Traffic Class Derivations – Trident-II and Helix Platform Switches

23.1.2 Transmit Queues and Port Shaping

Transmit queues are logical partitions of an Ethernet port's egress bandwidth. Data streams are assigned to queues based on their traffic class, then sent as scheduled by port and transmit settings. Support varies by switch platform. A queue's label determines its priority: queues with the suffix "0" have the lowest priority.

Parameters that determine transmission schedules include:

- **Traffic class-to-transmit queue mapping** determines the transmit queue for transmitting data streams based on traffic class. The set of available transmit maps vary by switch platforms:
  - Arad, Jericho, FM6000, Trident-II, and Helix platforms: one map for all unicast and multicast traffic.
  - Trident platform: one map for unicast traffic and one map for multicast traffic.
  - Petra platform: one map for unicast traffic. Queue shaping is not available for multicast traffic.
- **Port shaping** specifies a port's maximum egress bandwidth.

- **Queue shaping** specifies a transmit queue's maximum egress bandwidth, and implementation varies by platform.
  - Trident platform: queue shaping is configurable separately for unicast and multicast queues.
  - Trident-II platform: queue shaping is configurable for transmit queues. Port shaping and queue shaping are supported only in store-and-forward switching mode.
  - Petra platform: queue shaping is not available for multicast traffic.
  - Helix platform: queue shaping is configurable for transmit queues.
  - FM6000 platform: switches do not support simultaneous port shaping and queue shaping. Enabling port shaping on an FM6000 switch disables queue shaping, regardless of the previous configuration.
- **Guaranteed bandwidth** guarantees the allocation of a specified bandwidth for a transmit queue. Guaranteed bandwidth is supported only on Trident-II platforms.
- **Queue priority** specifies the priority at which a transmit queue is serviced. The switch defines two queue priority types:
  - *Strict priority* queues are serviced in the order of their priority rank - subject to each queue's configured maximum bandwidth. Data is not handled for a queue until all queues with higher priority are emptied or their transmission limit is reached. These queues typically carry low latency real time traffic and require highest available priority.
  - *Round robin* queues are serviced simultaneously subject to assigned bandwidth percentage and configured maximum bandwidth. All round robin queues have lower priority than strict priority queues. Round robin queues can be starved by strict priority queues.
- Queue scheduling determines how packets from different transmit queues are serviced to be sent out on the port.
- **Queue bandwidth allocation** specifies the time slice (percentage) assigned to a round robin queue, relative to all other round robin queues.

These sections describe transmit queue and port shaping configuration procedures:

- Section 23.2.4: Transmit Queues and Port Shaping – Arad Platform Switches
- Section 23.3.4: Transmit Queues and Port Shaping – Jericho Platform Switches
- Section 23.4.4: Transmit Queues and Port Shaping – FM6000 Platform Switches
- Section 23.5.4: Transmit Queues and Port Shaping – Petra Platform Switches
- Section 23.6.4: Transmit Queues and Port Shaping – Trident Platform Switches
- Section 23.7.4: Transmit Queues and Port Shaping – Trident-II and Helix Platform Switches

23.1.3 Explicit Congestion Notification (ECN)

Explicit Congestion Notification (ECN) is an IP and TCP extension that facilitates end-to-end network congestion notification without dropping packets. ECN recognizes early congestion and sets flags that signal affected hosts. Trident, Trident II, and Helix platform switches extend ECN support to non-TCP packets.

ECN usage requires that it is supported and enabled by both endpoints. Although only unicast flows are modified by ECN markers, the multicast, broadcast, and unmarked unicast flows can affect network congestion and influence the indication of unicast packet congestion.

23.1.3.1 ECN Conceptual Overview

The ECN field in the IP header (bits 6 and 7 in the IPv4 TOS or IPv6 traffic class octet) advertises ECN capabilities:

- 00: Router does not support ECN.
- 10: Router supports ECN.
- 01: Router supports ECN.
- 11: Congestion encountered.

Networks typically signal congestion by dropping packets. After an ECN-capable host negotiates ECN, it signals impending congestion by marking the IP header of packets encountering the congestion instead of dropping the packets. The recipient echoes the congestion indication back to the sender, which reduces its transmission rate as if it had detected a dropped packet.

Switches support ECN for unicast queues through Weighted Random Early Detection (WRED), an active queue management (AQM) algorithm that extends Random Early Detection (RED) to define multiple thresholds for an individual queue. WRED determines congestion by comparing average queue size with queue thresholds. Average queue size depends on the previous average and current queue size:

average queue size = (old_avg * (1-2^(-weight))) + (current_queue_size * 2^(-weight))

where weight is the exponential weight factor used for averaging the queue size.

Packets are marked based on WRED as follows:

- If average queue size is below the minimum threshold, packets are queued as in normal operation without ECN.
- If average queue size is greater than the maximum threshold, packets are marked for congestion.
- If average queue size is between minimum and maximum queue threshold, packets are either queued or marked. The proportion of packets that are marked increases linearly from 0% at the minimum threshold to 100% at the maximum threshold.

Treatment of packets marked as not ECN capable varies by platform.

These sections describe ECN configuration procedures:

- Section 23.2.5: ECN Configuration – Arad Platform Switches
- Section 23.6.5: ECN Configuration – Trident Platform Switches

23.1.4 ACL Policing

Access Control List (ACL) policing for ingress ACL support provides the ability to monitor the data rates for a particular class of traffic, and perform actions when that traffic exceeds user-configured values. This allows the user to control ingress bandwidth based on packet classification. Ingress policing is done by policing meters, which mark incoming traffic and perform actions based on the results.

ACL policing on Arista switches uses a single-rate, two-color marker system. The flow mode for single-rate two-color marker is as follows:

- One meter – one token bucket.
- Packet is marked green if there are enough tokens in the bucket to allow the packet, otherwise it is marked red.

ACL policing is supported on Trident, Trident-II, Trident+, FM6000, Arad, and Jericho platforms. Policing on LAG interfaces is not supported.

These sections describe ACL policing configuration procedures:

- Section 23.2.6: ACL Policing – Arad Platform Switches
- Section 23.3.5: ACL Policing – Jericho Platform Switches

23.2 QoS Configuration: Arad Platform Switches

Implementing QoS on an Arad platform switch consists of configuring port trust settings, default port settings, default traffic classes, conversion maps, and transmit queues.

- Section 23.2.1: CoS and DSCP Port Settings – Arad Platform Switches
- Section 23.2.2: Traffic Class Derivations – Arad Platform Switches
- Section 23.2.3: CoS Rewrite – Arad Platform Switches
- Section 23.2.4: Transmit Queues and Port Shaping – Arad Platform Switches
- Section 23.2.5: ECN Configuration – Arad Platform Switches
- Section 23.2.6: ACL Policing – Arad Platform Switches

Note QoS traffic policy is supported on Trident, Trident-II, FM6000, Arad, and Jericho.

23.2.1 CoS and DSCP Port Settings – Arad Platform Switches

Section 23.1.1.2 describes port trust and default port CoS and DSCP values.

Configuring Port Trust Settings

The **qos trust** command configures the QoS port trust mode for the configuration mode interface. Trust enabled ports use packet CoS or DSCP values to classify traffic. The port-trust default for switched ports is ***CoS***. The port-trust default for routed ports is ***DSCP***.

- **qos trust cos** specifies ***CoS*** as the port's port-trust mode.
- **qos trust dscp** specifies ***DSCP*** as the port's port-trust mode.
- **no qos trust** specifies ***untrusted*** as the port's port-trust mode.

The **show qos interfaces trust** command displays the trust mode of specified interfaces.

Example

- These commands configure and display the following trust modes:
  - Ethernet 3/5/1: dscp
  - Ethernet 3/5/2: untrusted
  - Ethernet 3/5/3: cos
  - Ethernet 3/5/4: default as a switched port

- Ethernet 3/6/1: default as a routed port

```
switch(config)#interface ethernet 3/5/1
switch(config-if-Et3/5/1)#qos trust dscp
switch(config-if-Et3/5/1)#interface ethernet 3/5/2
switch(config-if-Et3/5/2)#no qos trust
switch(config-if-Et3/5/2)#interface ethernet 3/5/3
switch(config-if-Et3/5/3)#qos trust cos
switch(config-if-Et3/5/3)#interface ethernet 3/5/4
switch(config-if-Et3/5/4)#switchport
switch(config-if-Et3/5/4)#default qos trust
switch(config-if-Et3/5/4)#interface ethernet 3/6/1
switch(config-if-Et3/6/1)#no switchport
switch(config-if-Et3/6/1)#default qos trust
switch(config-if-Et3/6/1)#show qos interface ethernet 3/5/1 - 3/6/1 trust
Port                                    Trust Mode
                              Operational           Configured
---------------------------------------------------------------
Ethernet3/5/1                 DSCP                  DSCP
Ethernet3/5/2                 UNTRUSTED             UNTRUSTED
Ethernet3/5/3                 COS                   COS
Ethernet3/5/4                 COS                   DEFAULT
Ethernet3/6/1                 DSCP                  DEFAULT

switch(config-if-Et3/6/1)#
```

Configuring Default Port Settings

Default CoS and DSCP values are assigned to each Ethernet and port channel interface. These commands specify the configuration mode interface commands specify the port's default CoS and DSCP values.

- **qos cos** configures a port's default CoS value.
- **qos dscp** configures a port's default DSCP value.

Example

```
These commands configure default CoS (4) and DSCP (44) values on Ethernet
interface 3/6/2.
switch(config)#interface ethernet 3/6/2
switch(config-if-Et3/6/2)#qos cos 4
switch(config-if-Et3/6/2)#qos dscp 44
switch(config-if-Et3/6/2)#show active
interface Ethernet3/6/2
   qos cos 4
   qos dscp 44
switch(config-if-Et3/6/2)#show qos interfaces ethernet 3/6/2
Ethernet3/6/2:
   Trust Mode: COS
   Default COS: 4
   Default DSCP: 44
              <-------OUTPUT OMITTED FROM EXAMPLE-------->
switch(config-if-Et3/6/2)#
```

23.2.2 Traffic Class Derivations – Arad Platform Switches

Section 23.1.1.4 describes traffic classes.

Traffic Class Derivation Source

Table 23-1 displays the source for deriving a data stream's traffic class.

Table 23-1 Traffic Class Derivation Source: Arad Platform Switches

	Untrusted	CoS Trusted	DSCP Trusted
Untagged Non-IP	Default CoS (port)	Default CoS (port)	Default DSCP (port)
Untagged IP	Default CoS (port)	Default CoS (port)	DSCP (packet)
Tagged Non-IP	Default CoS (port)	CoS (packet)	Default DSCP (port)
Tagged IP	Default CoS (port)	CoS (packet)	DSCP (packet)

Section 23.2.1 describes the default CoS and DSCP settings for each port.

Mapping CoS to Traffic Class

The **qos map cos** command assigns a traffic class to a list of CoS values. Multiple commands create a complete CoS to traffic class map. The switch uses this map to assign a traffic class to data packets on the basis of the packet's CoS field or the chip upon which it is received.

Example

- This command assigns the traffic class of 5 to the classes of service 1, 3, 5, and 7.

```
switch(config)#qos map cos 1 3 5 7 to traffic-class 5
switch(config)#show qos maps
   Number of Traffic Classes supported: 8
             <-------OUTPUT OMITTED FROM EXAMPLE-------->

   Cos-tc map:
     cos:  0  1  2  3  4  5  6  7
     ----------------------------
     tc:   1  5  2  5  4  5  6  5

             <-------OUTPUT OMITTED FROM EXAMPLE-------->
switch(config)#
```

Table 23-2 displays the default CoS to Traffic Class map on Arad platform switches.

Table 23-2 Default CoS to Traffic Class Map: Arad Platform Switches

Inbound CoS	Untagged	0	1	2	3	4	5	6	7
Traffic Class	Derived: use default CoS as inbound	1	0	2	3	4	5	6	7

Mapping DSCP to Traffic Class

The **qos map dscp** command assigns a traffic class to a set of DSCP values. Multiple commands create a complete DSCP to traffic class map. The switch uses this map to assign a traffic class to data packets on the basis of the packet's DSCP field or the chip upon which it is received.

Example

- This command assigns the traffic class of 0 to DSCP values of 12, 24, 41, and 44-47.

```
switch(config)#qos map dscp 12 24 41 44 45 46 47 to traffic-class 0
switch(config)#show qos maps
   Number of Traffic Classes supported: 8
             <-------OUTPUT OMITTED FROM EXAMPLE-------->

   Dscp-tc map:
     d1 :  d2 0  1  2  3  4  5  6  7  8  9
     ----------------------------------
      0 :     1  1  1  1  1  1  1  1  0  0
      1 :     0  0  0  0  0  0  2  2  2  2
      2 :     2  2  2  2  0  3  3  3  3  3
      3 :     3  3  4  4  4  4  4  4  4  4
      4 :     5  0  5  5  0  0  0  0  6  6
      5 :     6  6  6  6  6  6  7  7  7  7
      6 :     7  7  7  7

             <-------OUTPUT OMITTED FROM EXAMPLE-------->
switch(config)#
```

Table 23-3 displays the default DSCP to traffic class map on Arad platform switches.

Table 23-3 Default DSCP to Traffic Class Map: Arad Platform Switches

Inbound DSCP	0-7	8-15	16-23	24-31	32-39	40-47	48-55	56-63
Traffic Class	1	0	2	3	4	5	6	7

23.2.3 CoS Rewrite – Arad Platform Switches

Section 23.1.1.3 describes the CoS rewrite functions.

Traffic Class to CoS Rewrite Map

The CoS rewrite value is configurable and based on a data stream's traffic class, as specified by the traffic class-CoS rewrite map. The **qos map traffic-class to cos** command assigns a CoS rewrite value to a list of traffic classes. Multiple commands create the complete traffic class–CoS rewrite map.

Example

- This command assigns the CoS of two to traffic classes 1, 3, and 5.

```
switch(config)#qos map traffic-class 1 3 5 to cos 2
switch(config)#show qos map
   Number of Traffic Classes supported: 8
             <-------OUTPUT OMITTED FROM EXAMPLE-------->

   Tc-cos map:
     tc:   0  1  2  3  4  5  6  7
     ----------------------------
     cos:  1  2  2  2  4  2  6  7

             <-------OUTPUT OMITTED FROM EXAMPLE-------->
switch(config)#
```

Table 23-4 displays the default Traffic Class to CoS rewrite value map on Arad platform switches.

Table 23-4 Default Traffic Class to CoS Rewrite Value Map: Arad Platform Switches

Traffic Class	0	1	2	3	4	5	6	7
CoS Rewrite Value	1	0	2	3	4	5	6	7

Traffic Class to DSCP Rewrite Map

DSCP rewrite is always disabled on Arad platform switches.

23.2.4 Transmit Queues and Port Shaping – Arad Platform Switches

Section 23.1.2 describes transmit queues and port shaping.

Arad platform switches provide 16 physical queues for each egress port: eight unicast and eight multicast queues. Data is scheduled to the physical queues based on transmit queue assignments.

Multicast queue capacity that remains after multicast traffic is serviced is available for unicast traffic of a corresponding priority. Similarly, unicast queue capacity that remains after unicast traffic is serviced is available for overflow multicast traffic. Under conditions of unicast and multicast congestion, egress traffic is evenly split between unicast and multicast traffic.

A data stream's traffic class determines the transmit queue it uses. The switch defines a single traffic class–transmit queue map for unicast and multicast traffic on all Ethernet and port channel interfaces. The **show qos maps** command displays the traffic class–transmit queue map.

Table 23-5 displays the default traffic class to transmit queue map on Arad platform switches.

Table 23-5 Default Traffic Class to Transmit Queue Map: Arad Platform Switches

Traffic Class	0	1	2	3	4	5	6	7
Transmit Queue	0	1	2	3	4	5	6	7

Transmit queue parameters are configured in tx-queue configuration command mode, which is entered from interface-ethernet configuration mode.

Mapping Traffic Classes to a Transmit Queue

The **qos map traffic-class to tx-queue** command assigns traffic classes to a transmit queue. Multiple commands complete the traffic class-transmit queue map. Traffic class 7 and transmit queue 7 are always mapped to each other. This association is not editable.

Example

- These commands assign traffic classes of 1, 3, and 5 to transmit queue 1, traffic classes 2, 4, and 6 to transmit queue 2, and traffic class 0 to transmit queue 0, then display the resultant map.

```
switch(config)#qos map traffic-class 1 3 5 to tx-queue 1
switch(config)#qos map traffic-class 2 4 6 to tx-queue 2
switch(config)#qos map traffic-class 0 to tx-queue 0
switch(config)#show qos maps
   Number of Traffic Classes supported: 8
   Number of Transmit Queues supported: 8
             <-------OUTPUT OMITTED FROM EXAMPLE-------->

   Tc - tx-queue map:
     tc:        0  1  2  3  4  5  6  7
     --------------------------------
     tx-queue:  0  1  2  1  2  1  2  7

switch(config)#
```

Entering Tx-Queue Configuration Mode

The **tx-queue (Arad/Jericho)** command places the switch in tx-queue configuration mode to configure a transmit queue on the configuration mode interface. Tx-queue 7 is not configurable. The **show qos interfaces** displays the transmit queue configuration for a specified port.

Example

- This command enters Tx-queue configuration mode for transmit queue 4 of Ethernet interface 3/3/3.

```
switch(config)#interface ethernet 3/3/3
switch(config-if-Et3/3/3)#tx-queue 4
switch(config-if-Et3/3/3-txq-4)#
```

Configuring the Shape Rate – Port and Transmit Queues

A port's shape rate specifies its maximum outbound traffic bandwidth. A transmit queue's shape rate specifies the queue's maximum outbound bandwidth. Shape rate commands specify data rates in kbps.

- To configure a port's shape rate, enter **shape rate (Interface – Arad/Jericho)** from the port's interface configuration mode.
- To configure a transmit queue's shape rate, enter **shape rate (Tx-queue – Arad/Jericho)** from the queue's tx-queue configuration mode.

Examples

- This command configures a port shape rate of 5 Gbps on Ethernet interface 3/5/1.

```
switch(config)#interface ethernet 3/5/1
switch(config-if-Et3/5/1)#shape rate 5000000
switch(config-if-Et3/5/1)#show qos interfaces ethernet 3/5/1
Ethernet3/5/1:
             <-------OUTPUT OMITTED FROM EXAMPLE-------->
   Port shaping rate: 5000012 / 5000000 kbps

  Tx    Bandwidth       Shape Rate        Priority  ECN
 Queue  (percent)        (units)
   ------------------------------------------------------
   7      - / -        - / -    (  -  )    SP / SP    D
             <-------OUTPUT OMITTED FROM EXAMPLE-------->
switch(config-if-Et3/5/1)#
```

- These commands configure a shape rate of 1 Gbps on transmit queues 3 and 4 of Ethernet interface 3/4/1.

```
switch(config)#interface ethernet 3/4/1
switch(config-if-Et3/4/1)#tx-queue 4
switch(config-if-Et3/4/1-txq-4)#shape rate 1000000 kbps
switch(config-if-Et3/4/1-txq-4)#tx-queue 3
switch(config-if-Et3/4/1-txq-3)#shape rate 1000000 kbps
switch(config-if-Et3/4/1-txq-3)#show qos interface ethernet 3/4/1
Ethernet3/4/1:
             <-------OUTPUT OMITTED FROM EXAMPLE-------->
                Port shaping rate: disabled

  Tx    Bandwidth       Shape Rate        Priority  ECN
 Queue  (percent)        (units)
   ------------------------------------------------------
   7      - / -        - / -    (  -  )    SP / SP    D
   6      - / -        - / -    (  -  )    SP / SP    D
   5      - / -        - / -    (  -  )    SP / SP    D
   4      - / -      999 / 1000 ( Mbps )   SP / SP    D
   3      - / -      999 / 1000 ( Mbps )   SP / SP    D
   2      - / -        - / -    (  -  )    SP / SP    D
   1      - / -        - / -    (  -  )    SP / SP    D
   0      - / -        - / -    (  -  )    SP / SP    D
             <-------OUTPUT OMITTED FROM EXAMPLE-------->
switch(config-if-Et3/4/1-txq-3)#
```

Configuring Queue Priority

The **priority (Arad/Jericho)** command configures a transmit queue's priority type:

- The **priority strict** command configures the queue as a strict priority queue.
- The **no priority** command configures the queue as a round robin queue.

  A queue's configuration as ***round robin*** also applies to all lower priority queues regardless of other configuration statements.

The **bandwidth percent (Arad/Jericho)** command configures a round robin queue's bandwidth share. The cumulative operational bandwidth of all round robin queues is always less than or equal to 100%. If the cumulative configured bandwidth is greater than 100%, each port's operational bandwidth is its configured bandwidth divided by the cumulative configured bandwidth.

Example

- These commands configure queues 0 through 3 (Ethernet interface 3/5/1) as round robin, then allocate bandwidth for three queues at 30% and one queue at 10%.

  The **no priority** statement for queue 3 also configures queues 0, 1, and 2 as round robin queues. Removing this statement reverts the other queues to ***strict priority*** type unless ***running-config*** contains a **no priority** statement for one of these queues.

```
switch(config)#interface ethernet 3/5/1
switch(config-if-Et3/5/1)#tx-queue 3
switch(config-if-Et3/5/1-txq-3)#no priority
switch(config-if-Et3/5/1-txq-3)#bandwidth percent 10
switch(config-if-Et3/5/1-txq-3)#tx-queue 2
switch(config-if-Et3/5/1-txq-2)#bandwidth percent 30
switch(config-if-Et3/5/1-txq-2)#tx-queue 1
switch(config-if-Et3/5/1-txq-1)#bandwidth percent 30
switch(config-if-Et3/5/1-txq-1)#tx-queue 0
switch(config-if-Et3/5/1-txq-0)#bandwidth percent 30
switch(config-if-Et3/5/1-txq-0)#show qos interfaces ethernet 3/5/1
Ethernet3/5/1:

  Tx    Bandwidth       Shape Rate        Priority  ECN
 Queue  (percent)        (units)
   ------------------------------------------------------
   7      - / -        - / -    (  -  )    SP / SP    D
   6      - / -        - / -    (  -  )    SP / SP    D
   5      - / -        - / -    (  -  )    SP / SP    D
   4      - / -        - / -    (  -  )    SP / SP    D
   3     10 / 10       - / -    (  -  )    RR / RR    D
   2     30 / 30       - / -    (  -  )    RR / SP    D
   1     30 / 30       - / -    (  -  )    RR / SP    D
   0     30 / 30       - / -    (  -  )    RR / SP    D
            <-------OUTPUT OMITTED FROM EXAMPLE-------->
switch(config-if-Et3/5/1-txq-0)#
```

Changing the bandwidth percentage for queue 3 to 30 changes the operational bandwidth of each queue to its configured bandwidth divided by 120% (10%+20%+30%+60%).

```
switch(config-if-Et3/5/1-txq-0)#tx-queue 3
switch(config-if-Et3/5/1-txq-3)#bandwidth percent 30
switch(config-if-Et3/5/1-txq-3)#show qos interfaces ethernet 3/5/1
Ethernet3/5/1:
             <-------OUTPUT OMITTED FROM EXAMPLE-------->
   Port shaping rate: disabled

  Tx    Bandwidth       Shape Rate        Priority  ECN
 Queue  (percent)        (units)
   ------------------------------------------------------
   7      - / -        - / -    (  -  )    SP / SP    D
   6      - / -        - / -    (  -  )    SP / SP    D
   5      - / -        - / -    (  -  )    SP / SP    D
   4      - / -        - / -    (  -  )    SP / SP    D
   3     24 / 30       - / -    (  -  )    RR / RR    D
   2     24 / 30       - / -    (  -  )    RR / SP    D
   1     24 / 30       - / -    (  -  )    RR / SP    D
   0     24 / 30       - / -    (  -  )    RR / SP    D

Note: Values are displayed as Operational/Configured
             <-------OUTPUT OMITTED FROM EXAMPLE-------->
switch(config-if-Et3/5/1-txq-3)#
```

23.2.5 ECN Configuration – Arad Platform Switches

Section 23.1.3 describes Explicit Congestion Notification (ECN).

ECN is independently configurable on all egress queues of each Ethernet interface. ECN settings for Port-Channels are applied on each of the channel's member Ethernet interfaces. Average queue length is tracked for transmit queues. When it reaches maximum threshold, all subsequent packets are marked.

Although the switch does not limit the number of queues that can be configured for ECN, hardware table limitations restrict the number of queues that can simultaneously implement ECN.

The **random-detect ecn (Arad/Jericho)** command enables ECN marking for the configuration mode unicast transmit queue and specifies threshold queue sizes.

Example

- These commands enable ECN marking of unicast packets from unicast transmit queue 4 of Ethernet interface 3/5/1, setting thresholds at 128 kbytes and 1280 kbytes.

```
switch(config)#interface ethernet 3/5/1
switch(config-if-Et3/5/1)#tx-queue 4
switch(config-if-Et3/5/1-txq-4)#random-detect ecn minimum-threshold 128 kbytes
maximum-threshold 1280 kbyte
switch(config-if-Et3/5/1-txq-4)#show active
interface Ethernet3/5/1
   tx-queue 4
     random-detect ecn minimum-threshold 128 kbytes maximum-threshold 1280 kbytes
switch(config-if-Et3/5/1-txq-4)#
```

23.2.6 ACL Policing – Arad Platform Switches

Section 23.1.4 describes ACL policing.

Implementing ACL policing consists of configuring the following:

- policy-map settings
- class-name
- committed information rate (CIR) the data speed committed to any given circuit regardless of the number of users
- burst size the maximum burst size in bytes the network commits to moving under normal conditions

The default unit for the metering rate CIR is bits per second; the default unit for the burst size is bytes.

The policer is applied to the class inside the policy map. Policy maps can contain one or more policy map classes, each with different match criteria and policer.

Default behavior and available policing actions are as follows:

- Policy map can be applied on multiple interfaces. Interfaces on the same chip will share the policer. (Applicable for Arad only.)
- If there is no policer configured within a class, all traffic is transmitted without any policing.
- If there are any actions configured, the configured actions are applied:
  - Conform-action (green): transmit (default)
  - Violate-action (red): drop (default)

Example

These commands configure ACL policing in single-rate, two-color mode.

```
switch(config)#class-map type qos match-any class1
switch(config-cmap-class1)#match ip access-group acl1
switch(config-cmap-class1)#exit
switch(config)#policy-map type qos policy1
switch(config-policy1)#class class1
switch(config-policy1-class1)#police cir 512000 bc 96000
switch(config-policy1-class1)#exit
switch(config-policy1)#exit
switch(config)#
```

Displaying ACL Policing Information

Examples

- This command shows the contents of all policy maps on the switch.

```
switch(config)#show policy-map
Service-policy policy1

Class-map: class1 (match-any)
Match: ip access-group name acl1
Police cir 512000 bps bc 96000 bytes

Class-map: class-default (match-any)
switch(config)#
```

- This command shows the interface-specific police counters for Ethernet interface 1.

```
switch(config)#show policy-map interface Ethernet 1 input counters
Service-policy input: policy1
Hardware programming status: Successful

Class-map: class1 (match-any) Match: ip access-group name acl1
Police cir 512000 bps bc 96000 bytes Conformed 4351 packets, 1857386 bytes
Conformed 2536 packets, 3384260 bytes

Class-map: class-default (match-any) matched packets: 0

switch(config)#
```

- This command shows the counters associated with the policy map called “p1.”

```
switch(config)#show policy-map type qos p1 input counters
Service-policy input: p1 Class-map: c1 (match-any)
Match: ip access-group name a1
Police cir 512000 bps bc 96000 bytes Interface: Ethernet1
Conformed 4351 packets, 1857386 bytes
Exceeded 2536 packets, 3384260 bytes
Interface: Ethernet2
Conformed 2351 packets, 957386 bytes
Exceeded 1536 packets, 1384260 bytes
Class-map: class-default (match-any)
Matched packets: 3229

switch(config)#
```

- This command shows the QoS policy map for Ethernet interface 1.

```
switch(config)#show policy-map interface Ethernet 1 input type qos
Interface: Ethernet 1 Service-policy input: policy1
Hardware programming status: Successful Class-map: class1 (match-any)
Match: ip access-group name acl1
Police cir 512000 bps bc 9000 bytes
Class-map: class2 (match-any)
Match: ip access-group name acl2 set dscp 2
Class-map: class3 (match-any) Match: ip access-group name acl3
Police cir 1280000 bps bc 9000 bytes
Class-map: class-default (match-any)

switch(config)#
```

23.3 QoS Configuration: Jericho Platform Switches

Implementing QoS on an Jericho platform switch consists of configuring port trust settings, default port settings, default traffic classes, conversion maps, and transmit queues.

- Section 23.3.1: CoS and DSCP Port Settings – Jericho Platform Switches
- Section 23.3.2: Traffic Class Derivations – Jericho Platform Switches
- Section 23.3.3: CoS Rewrite – Jericho Platform Switches
- Section 23.3.4: Transmit Queues and Port Shaping – Jericho Platform Switches
- Section 23.3.5: ACL Policing – Jericho Platform Switches

Note QoS traffic policy is supported on Trident, Trident-II, FM6000, Arad. and Jericho.

23.3.1 CoS and DSCP Port Settings – Jericho Platform Switches

Section 23.1.1.2 describes port trust and default port CoS and DSCP values.

Configuring Port Trust Settings

The **qos trust** command configures the QoS port trust mode for the configuration mode interface. Trust enabled ports use packet CoS or DSCP values to classify traffic. The port-trust default for switched ports is ***CoS***. The port-trust default for routed ports is ***DSCP***.

- **qos trust cos** specifies ***CoS*** as the port's port-trust mode.
- **qos trust dscp** specifies ***DSCP*** as the port's port-trust mode.
- **no qos trust** specifies ***untrusted*** as the port's port-trust mode.

The **show qos interfaces trust** command displays the trust mode of specified interfaces.

Example

- These commands configure and display the following trust modes:
  - Ethernet 3/5/1: dscp
  - Ethernet 3/5/2: untrusted
  - Ethernet 3/5/3: cos
  - Ethernet 3/5/4: default as a switched port

- Ethernet 3/6/1: default as a routed port

```
switch(config)#interface ethernet 3/5/1
switch(config-if-Et3/5/1)#qos trust dscp
switch(config-if-Et3/5/1)#interface ethernet 3/5/2
switch(config-if-Et3/5/2)#no qos trust
switch(config-if-Et3/5/2)#interface ethernet 3/5/3
switch(config-if-Et3/5/3)#qos trust cos
switch(config-if-Et3/5/3)#interface ethernet 3/5/4
switch(config-if-Et3/5/4)#switchport
switch(config-if-Et3/5/4)#default qos trust
switch(config-if-Et3/5/4)#interface ethernet 3/6/1
switch(config-if-Et3/6/1)#no switchport
switch(config-if-Et3/6/1)#default qos trust
switch(config-if-Et3/6/1)#show qos interface ethernet 3/5/1 - 3/6/1 trust
Port                                    Trust Mode
                              Operational           Configured
---------------------------------------------------------------
Ethernet3/5/1                 DSCP                  DSCP
Ethernet3/5/2                 UNTRUSTED             UNTRUSTED
Ethernet3/5/3                 COS                   COS
Ethernet3/5/4                 COS                   DEFAULT
Ethernet3/6/1                 DSCP                  DEFAULT

switch(config-if-Et3/6/1)#
```

Configuring Default Port Settings

Default CoS and DSCP values are assigned to each Ethernet and port channel interface. These commands specify the configuration mode interface commands specify the port's default CoS and DSCP values.

- **qos cos** configures a port's default CoS value.
- **qos dscp** configures a port's default DSCP value.

Example

```
These commands configure default CoS (4) and DSCP (44) values on Ethernet
interface 3/6/2.
switch(config)#interface ethernet 3/6/2
switch(config-if-Et3/6/2)#qos cos 4
switch(config-if-Et3/6/2)#qos dscp 44
switch(config-if-Et3/6/2)#show active
interface Ethernet3/6/2
   qos cos 4
   qos dscp 44
switch(config-if-Et3/6/2)#show qos interfaces ethernet 3/6/2
Ethernet3/6/2:
   Trust Mode: COS
   Default COS: 4
   Default DSCP: 44
              <-------OUTPUT OMITTED FROM EXAMPLE-------->
switch(config-if-Et3/6/2)#
```

23.3.2 Traffic Class Derivations – Jericho Platform Switches

Section 23.1.1.4 describes traffic classes.

Traffic Class Derivation Source

Table 23-1 displays the source for deriving a data stream's traffic class.

Table 23-6 Traffic Class Derivation Source: Jericho Platform Switches

	Untrusted	CoS Trusted	DSCP Trusted
Untagged Non-IP	Default CoS (port)	Default CoS (port)	Default DSCP (port)
Untagged IP	Default CoS (port)	Default CoS (port)	DSCP (packet)
Tagged Non-IP	Default CoS (port)	CoS (packet)	Default DSCP (port)
Tagged IP	Default CoS (port)	CoS (packet)	DSCP (packet)

Section 23.2.1 describes the default CoS and DSCP settings for each port.

Mapping CoS to Traffic Class

The **qos map cos** command assigns a traffic class to a list of CoS values. Multiple commands create a complete CoS to traffic class map. The switch uses this map to assign a traffic class to data packets on the basis of the packet's CoS field or the chip upon which it is received.

Example

- This command assigns the traffic class of 5 to the classes of service 1, 3, 5, and 7.

```
switch(config)#qos map cos 1 3 5 7 to traffic-class 5
switch(config)#show qos maps
   Number of Traffic Classes supported: 8
             <-------OUTPUT OMITTED FROM EXAMPLE-------->

   Cos-tc map:
     cos:  0  1  2  3  4  5  6  7
     ----------------------------
     tc:   1  5  2  5  4  5  6  5

             <-------OUTPUT OMITTED FROM EXAMPLE-------->
switch(config)#
```

Table 23-2 displays the default CoS to Traffic Class map on Jericho platform switches.

Table 23-7 Default CoS to Traffic Class Map: Jericho Platform Switches

Inbound CoS	Untagged	0	1	2	3	4	5	6	7
Traffic Class	Derived: use default CoS as inbound	1	0	2	3	4	5	6	7

Mapping DSCP to Traffic Class

The **qos map dscp** command assigns a traffic class to a set of DSCP values. Multiple commands create a complete DSCP to traffic class map. The switch uses this map to assign a traffic class to data packets on the basis of the packet's DSCP field or the chip upon which it is received.

Example

- This command assigns the traffic class of 0 to DSCP values of 12, 24, 41, and 44-47.

```
switch(config)#qos map dscp 12 24 41 44 45 46 47 to traffic-class 0
switch(config)#show qos maps
   Number of Traffic Classes supported: 8
             <-------OUTPUT OMITTED FROM EXAMPLE-------->

   Dscp-tc map:
     d1 :  d2 0  1  2  3  4  5  6  7  8  9
     ----------------------------------------
      0 :     1  1  1  1  1  1  1  1  0  0
      1 :     0  0  0  0  0  0  2  2  2  2
      2 :     2  2  2  2  0  3  3  3  3  3
      3 :     3  3  4  4  4  4  4  4  4  4
      4 :     5  0  5  5  0  0  0  0  6  6
      5 :     6  6  6  6  6  6  7  7  7  7
      6 :     7  7  7  7

             <-------OUTPUT OMITTED FROM EXAMPLE-------->
switch(config)#
```

Table 23-3 displays the default DSCP to traffic class map on Jericho platform switches.

Table 23-8 Default DSCP to Traffic Class Map: Jericho Platform Switches

Inbound DSCP	0-7	8-15	16-23	24-31	32-39	40-47	48-55	56-63
Traffic Class	1	0	2	3	4	5	6	7

23.3.3 CoS Rewrite – Jericho Platform Switches

Section 23.1.1.3 describes the CoS rewrite functions.

Traffic Class to CoS Rewrite Map

The CoS rewrite value is configurable and based on a data stream's traffic class, as specified by the traffic class-CoS rewrite map. The **qos map traffic-class to cos** command assigns a CoS rewrite value to a list of traffic classes. Multiple commands create the complete traffic class–CoS rewrite map.

Example

- This command assigns the CoS of two to traffic classes 1, 3, and 5.

```
switch(config)#qos map traffic-class 1 3 5 to cos 2
switch(config)#show qos map
   Number of Traffic Classes supported: 8
             <-------OUTPUT OMITTED FROM EXAMPLE-------->

   Tc-cos map:
     tc:   0  1  2  3  4  5  6  7
     ----------------------------
     cos:  1  2  2  2  4  2  6  7

             <-------OUTPUT OMITTED FROM EXAMPLE-------->
switch(config)#
```

Table 23-4 displays the default Traffic Class to CoS rewrite value map on Jericho platform switches.

Table 23-9 Default Traffic Class to CoS Rewrite Value Map: Jericho Platform Switches

Traffic Class	0	1	2	3	4	5	6	7
CoS Rewrite Value	1	0	2	3	4	5	6	7

Traffic Class to DSCP Rewrite Map

DSCP rewrite is always disabled on Jericho platform switches.

23.3.4 Transmit Queues and Port Shaping – Jericho Platform Switches

Section 23.1.2 describes transmit queues and port shaping.

Jericho platform switches provide 16 physical queues for each egress port: eight unicast and eight multicast queues. Data is scheduled to the physical queues based on transmit queue assignments.

Multicast queue capacity that remains after multicast traffic is serviced is available for unicast traffic of a corresponding priority. Similarly, unicast queue capacity that remains after unicast traffic is serviced is available for overflow multicast traffic. Under conditions of unicast and multicast congestion, egress traffic is evenly split between unicast and multicast traffic.

A data stream's traffic class determines the transmit queue it uses. The switch defines a single traffic class–transmit queue map for unicast and multicast traffic on all Ethernet and port channel interfaces. The **show qos maps** command displays the traffic class–transmit queue map.

Table 23-5 displays the default traffic class to transmit queue map on Jericho platform switches.

Table 23-10 Default Traffic Class to Transmit Queue Map: Jericho Platform Switches

Traffic Class	0	1	2	3	4	5	6	7
Transmit Queue	0	1	2	3	4	5	6	7

Transmit queue parameters are configured in tx-queue configuration command mode, which is entered from interface-ethernet configuration mode.

Mapping Traffic Classes to a Transmit Queue

The **qos map traffic-class to tx-queue** command assigns traffic classes to a transmit queue. Multiple commands complete the traffic class-transmit queue map. Traffic class 7 and transmit queue 7 are always mapped to each other. This association is not editable.

Example

- These commands assign traffic classes of 1, 3, and 5 to transmit queue 1, traffic classes 2, 4, and 6 to transmit queue 2, and traffic class 0 to transmit queue 0, then display the resultant map.

```
switch(config)#qos map traffic-class 1 3 5 to tx-queue 1
switch(config)#qos map traffic-class 2 4 6 to tx-queue 2
switch(config)#qos map traffic-class 0 to tx-queue 0
switch(config)#show qos maps
   Number of Traffic Classes supported: 8
   Number of Transmit Queues supported: 8
             <-------OUTPUT OMITTED FROM EXAMPLE-------->

   Tc - tx-queue map:
     tc:        0  1  2  3  4  5  6  7
     --------------------------------
     tx-queue:  0  1  2  1  2  1  2  7

switch(config)#
```

Entering Tx-Queue Configuration Mode

The **tx-queue (Arad/Jericho)** command places the switch in tx-queue configuration mode to configure a transmit queue on the configuration mode interface. Tx-queue 7 is not configurable. The **show qos interfaces** displays the transmit queue configuration for a specified port.

Example

- This command enters Tx-queue configuration mode for transmit queue 4 of Ethernet interface 3/3/3.

```
switch(config)#interface ethernet 3/3/3
switch(config-if-Et3/3/3)#tx-queue 4
switch(config-if-Et3/3/3-txq-4)#
```

Configuring the Shape Rate – Port and Transmit Queues

A port's shape rate specifies its maximum outbound traffic bandwidth. A transmit queue's shape rate specifies the queue's maximum outbound bandwidth. Shape rate commands specify data rates in kbps.

- To configure a port's shape rate, enter **shape rate (Interface – Arad/Jericho)** from the port's interface configuration mode.
- To configure a transmit queue's shape rate, enter **shape rate (Tx-queue – Arad/Jericho)** from the queue's tx-queue configuration mode.

Examples

- This command configures a port shape rate of 5 Gbps on Ethernet interface 3/5/1.

```
switch(config)#interface ethernet 3/5/1
switch(config-if-Et3/5/1)#shape rate 5000000
switch(config-if-Et3/5/1)#show qos interfaces ethernet 3/5/1
Ethernet3/5/1:
             <-------OUTPUT OMITTED FROM EXAMPLE-------->
   Port shaping rate: 5000012 / 5000000 kbps

  Tx    Bandwidth       Shape Rate        Priority  ECN
 Queue  (percent)        (units)
   ------------------------------------------------------
   7      - / -        - / -    (  -  )    SP / SP    D
             <-------OUTPUT OMITTED FROM EXAMPLE-------->
switch(config-if-Et3/5/1)#
```

- These commands configure a shape rate of 1 Gbps on transmit queues 3 and 4 of Ethernet interface 3/4/1.

```
switch(config)#interface ethernet 3/4/1
switch(config-if-Et3/4/1)#tx-queue 4
switch(config-if-Et3/4/1-txq-4)#shape rate 1000000 kbps
switch(config-if-Et3/4/1-txq-4)#tx-queue 3
switch(config-if-Et3/4/1-txq-3)#shape rate 1000000 kbps
switch(config-if-Et3/4/1-txq-3)#show qos interface ethernet 3/4/1
Ethernet3/4/1:
             <-------OUTPUT OMITTED FROM EXAMPLE-------->
                Port shaping rate: disabled

  Tx    Bandwidth       Shape Rate        Priority  ECN
 Queue  (percent)        (units)
   ------------------------------------------------------
   7      - / -        - / -    (  -  )    SP / SP    D
   6      - / -        - / -    (  -  )    SP / SP    D
   5      - / -        - / -    (  -  )    SP / SP    D
   4      - / -      999 / 1000 ( Mbps )   SP / SP    D
   3      - / -      999 / 1000 ( Mbps )   SP / SP    D
   2      - / -        - / -    (  -  )    SP / SP    D
   1      - / -        - / -    (  -  )    SP / SP    D
   0      - / -        - / -    (  -  )    SP / SP    D
             <-------OUTPUT OMITTED FROM EXAMPLE-------->
switch(config-if-Et3/4/1-txq-3)#
```

Configuring Queue Priority

The **priority (Arad/Jericho)** command configures a transmit queue's priority type:

- The **priority strict** command configures the queue as a strict priority queue.
- The **no priority** command configures the queue as a round robin queue.

  A queue's configuration as ***round robin*** also applies to all lower priority queues regardless of other configuration statements.

The **bandwidth percent (Arad/Jericho)** command configures a round robin queue's bandwidth share. The cumulative operational bandwidth of all round robin queues is always less than or equal to 100%. If the cumulative configured bandwidth is greater than 100%, each port's operational bandwidth is its configured bandwidth divided by the cumulative configured bandwidth.

Example

- These commands configure queues 0 through 3 (Ethernet interface 3/5/1) as round robin, then allocate bandwidth for three queues at 30% and one queue at 10%.

  The **no priority** statement for queue 3 also configures queues 0, 1, and 2 as round robin queues. Removing this statement reverts the other queues to ***strict priority*** type unless ***running-config*** contains a **no priority** statement for one of these queues.

```
switch(config)#interface ethernet 3/5/1
switch(config-if-Et3/5/1)#tx-queue 3
switch(config-if-Et3/5/1-txq-3)#no priority
switch(config-if-Et3/5/1-txq-3)#bandwidth percent 10
switch(config-if-Et3/5/1-txq-3)#tx-queue 2
switch(config-if-Et3/5/1-txq-2)#bandwidth percent 30
switch(config-if-Et3/5/1-txq-2)#tx-queue 1
switch(config-if-Et3/5/1-txq-1)#bandwidth percent 30
switch(config-if-Et3/5/1-txq-1)#tx-queue 0
switch(config-if-Et3/5/1-txq-0)#bandwidth percent 30
switch(config-if-Et3/5/1-txq-0)#show qos interfaces ethernet 3/5/1
Ethernet3/5/1:

  Tx    Bandwidth       Shape Rate        Priority  ECN
 Queue  (percent)        (units)
   ------------------------------------------------------
   7      - / -        - / -    (  -  )    SP / SP    D
   6      - / -        - / -    (  -  )    SP / SP    D
   5      - / -        - / -    (  -  )    SP / SP    D
   4      - / -        - / -    (  -  )    SP / SP    D
   3     10 / 10       - / -    (  -  )    RR / RR    D
   2     30 / 30       - / -    (  -  )    RR / SP    D
   1     30 / 30       - / -    (  -  )    RR / SP    D
   0     30 / 30       - / -    (  -  )    RR / SP    D
            <-------OUTPUT OMITTED FROM EXAMPLE-------->
switch(config-if-Et3/5/1-txq-0)#
```

Changing the bandwidth percentage for queue 3 to 30 changes the operational bandwidth of each queue to its configured bandwidth divided by 120% (10%+20%+30%+60%).

```
switch(config-if-Et3/5/1-txq-0)#tx-queue 3
switch(config-if-Et3/5/1-txq-3)#bandwidth percent 30
switch(config-if-Et3/5/1-txq-3)#show qos interfaces ethernet 3/5/1
Ethernet3/5/1:
             <-------OUTPUT OMITTED FROM EXAMPLE-------->
   Port shaping rate: disabled

  Tx    Bandwidth       Shape Rate        Priority  ECN
 Queue  (percent)        (units)
   ------------------------------------------------------
   7      - / -        - / -    (  -  )    SP / SP    D
   6      - / -        - / -    (  -  )    SP / SP    D
   5      - / -        - / -    (  -  )    SP / SP    D
   4      - / -        - / -    (  -  )    SP / SP    D
   3     24 / 30       - / -    (  -  )    RR / RR    D
   2     24 / 30       - / -    (  -  )    RR / SP    D
   1     24 / 30       - / -    (  -  )    RR / SP    D
   0     24 / 30       - / -    (  -  )    RR / SP    D

Note: Values are displayed as Operational/Configured
             <-------OUTPUT OMITTED FROM EXAMPLE-------->
switch(config-if-Et3/5/1-txq-3)#
```

23.3.5 ACL Policing – Jericho Platform Switches

Section 23.1.4 describes ACL policing.

Implementing ACL policing consists of configuring the following:

- policy-map settings
- class-name
- committed information rate (CIR) the data speed committed to any given circuit regardless of the number of users
- burst size the maximum burst size in bytes the network commits to moving under normal conditions

The default unit for the metering rate CIR is bits per second; the default unit for the burst size is bytes.

The policer is applied to the class inside the policy map. Policy maps can contain one or more policy map classes, each with different match criteria and policer.

Default behavior and available policing actions are as follows:

- Policy map can be applied on multiple interfaces. Interfaces on the same chip will share the policer. (Applicable for Arad and Jericho only.)
- If there is no policer configured within a class, all traffic is transmitted without any policing.
- If there are any actions configured, the configured actions are applied:
  - Conform-action (green): transmit (default)
  - Violate-action (red): drop (default)

Example

These commands configure ACL policing in single-rate, two-color mode.

```
switch(config)#class-map type qos match-any class1
switch(config-cmap-class1)#match ip access-group acl1
switch(config-cmap-class1)#exit
switch(config)#policy-map type qos policy1
switch(config-policy1)#class class1
switch(config-policy1-class1)#police cir 512000 bc 96000
switch(config-policy1-class1)#exit
switch(config-policy1)#exit
switch(config)#
```

Displaying ACL Policing Information

Examples

- This command shows the contents of all policy maps on the switch.

```
switch(config)#show policy-map
Service-policy policy1

Class-map: class1 (match-any)
Match: ip access-group name acl1
Police cir 512000 bps bc 96000 bytes

Class-map: class-default (match-any)
switch(config)#
```

- This command shows the interface-specific police counters for Ethernet interface 1.

```
switch(config)#show policy-map interface Ethernet 1 input counters
Service-policy input: policy1
Hardware programming status: Successful

Class-map: class1 (match-any) Match: ip access-group name acl1
Police cir 512000 bps bc 96000 bytes Conformed 4351 packets, 1857386 bytes
Conformed 2536 packets, 3384260 bytes

Class-map: class-default (match-any) matched packets: 0

switch(config)#
```

- This command shows the counters associated with the policy map called “p1.”

```
switch(config)#show policy-map type qos p1 input counters
Service-policy input: p1 Class-map: c1 (match-any)
Match: ip access-group name a1
Police cir 512000 bps bc 96000 bytes Interface: Ethernet1
Conformed 4351 packets, 1857386 bytes
Exceeded 2536 packets, 3384260 bytes
Interface: Ethernet2
Conformed 2351 packets, 957386 bytes
Exceeded 1536 packets, 1384260 bytes
Class-map: class-default (match-any)
Matched packets: 3229

switch(config)#
```

- This command shows the QoS policy map for Ethernet interface 1.

```
switch(config)#show policy-map interface Ethernet 1 input type qos
Interface: Ethernet 1 Service-policy input: policy1
Hardware programming status: Successful Class-map: class1 (match-any)
Match: ip access-group name acl1
Police cir 512000 bps bc 9000 bytes
Class-map: class2 (match-any)
Match: ip access-group name acl2 set dscp 2
Class-map: class3 (match-any) Match: ip access-group name acl3
Police cir 1280000 bps bc 9000 bytes
Class-map: class-default (match-any)

switch(config)#
```

23.4 QoS Configuration: FM6000 Platform Switches

Implementing QoS on an FM6000 platform switch consists of configuring port trust settings, default port settings, default traffic classes, conversion maps, and transmit queues.

- Section 23.4.1: CoS and DSCP Port Settings – FM6000 Platform Switches
- Section 23.4.2: Traffic Class Derivations – FM6000 Platform Switches
- Section 23.4.3: CoS and DSCP Rewrite – FM6000 Platform Switches
- Section 23.4.4: Transmit Queues and Port Shaping – FM6000 Platform Switches

23.4.1 CoS and DSCP Port Settings – FM6000 Platform Switches

Section 23.1.1.2 describes port trust and default port CoS and DSCP values.

Configuring Port Trust Settings

The **qos trust** command configures the QoS port trust mode for the configuration mode interface. Trust enabled ports use packet CoS or DSCP values to classify traffic. The port-trust default for switched ports is ***cos***. The port-trust default for routed ports is ***dscp***.

- **qos trust cos** specifies ***cos*** as the port's port-trust mode.
- **qos trust dscp** specifies ***dscp*** as the port's port-trust mode.
- **no qos trust** specifies ***untrusted*** as the port's port-trust mode.

The **show qos interfaces trust** command displays the trust mode of specified interfaces.

Example

- These commands configure and display the following trust modes:
  - Ethernet 15: dscp
  - Ethernet 16: untrusted
  - Ethernet 17: cos
  - Ethernet 18: default as a switched port

- Ethernet 19: default as a routed port

```
switch(config)#interface ethernet 15
switch(config-if-Et15)#qos trust dscp
switch(config-if-Et15)#interface ethernet 16
switch(config-if-Et16)#no qos trust
switch(config-if-Et16)#interface ethernet 17
switch(config-if-Et17)#qos trust cos
switch(config-if-Et17)#interface ethernet 18
switch(config-if-Et18)#switchport
switch(config-if-Et18)#default qos trust
switch(config-if-Et19)#interface ethernet 19
switch(config-if-Et19)#no switchport
switch(config-if-Et19)#default qos trust
switch(config-if-Et19)#show qos interface ethernet 15 - 19 trust
Port                                        Trust Mode
                                Operational           Configured
---------------------------------------------------------------
Ethernet15                      DSCP                  DSCP
Ethernet16                      UNTRUSTED             UNTRUSTED
Ethernet17                      COS                   COS
Ethernet18                      COS                   DEFAULT
Ethernet19                      DSCP                  DEFAULT

switch(config-if-Et19)#
```

Configuring Default Port Settings

Default CoS and DSCP settings are assigned to individual port channel and Ethernet interfaces. These configuration mode interface commands specify the port's default CoS and DSCP values.

- **qos cos** configures a port's default CoS value.
- **qos dscp** configures a port's default DSCP value.

Example

- These commands configure default CoS (4) and DSCP (44) settings on Ethernet interface 19.

```
switch(config)#interface ethernet 19
switch(config-if-Et19)#qos cos 4
switch(config-if-Et19)#qos dscp 44
switch(config-if-Et19)#show active
interface Ethernet19
   qos cos 4
   qos dscp 44
switch(config-if-Et19)#show qos interfaces ethernet 19
Ethernet19:
   Trust Mode: COS
   Default COS: 4
   Default DSCP: 44
             <-------OUTPUT OMITTED FROM EXAMPLE-------->
switch(config-if-Et19)#
```

23.4.2 Traffic Class Derivations – FM6000 Platform Switches

Section 23.1.1.4 describes traffic classes.

Traffic Class Derivation Source

Table 23-11 displays the source for deriving a data stream's traffic class.

Table 23-11 Traffic Class Derivation Source: FM6000 Platform Switches

	Untrusted	CoS Trusted	DSCP Trusted
Untagged Non-IP	Default CoS (port)	Default CoS (port)	Default DSCP (port)
Untagged IP	Default CoS (port)	Default CoS (port)	DSCP (packet)
Tagged Non-IP	Default CoS (port)	CoS (packet)	Default DSCP (port)
Tagged IP	Default CoS (port)	CoS (packet)	DSCP (packet)

Section 23.4.1 describes the default CoS and DSCP settings for each port.

Mapping CoS to Traffic Class

The **qos map cos** command assigns a traffic class to a list of CoS settings. Multiple commands create a complete CoS to traffic class map. The switch uses this map to assign a traffic class to data packets on the basis of the packet's CoS field or the port upon which it is received.

Example

- This command assigns the traffic class of 5 to the classes of service 1, 3, 5, and 7.

```
switch(config)#qos map cos 1 3 5 7 to traffic-class 5
switch(config)#show qos maps
   Number of Traffic Classes supported: 8
   Number of Transmit Queues supported: 8
            <-------OUTPUT OMITTED FROM EXAMPLE-------->

   Cos-tc map:
     cos:  0  1  2  3  4  5  6  7
     ----------------------------
     tc:   1  5  2  5  4  5  6  5

            <-------OUTPUT OMITTED FROM EXAMPLE-------->
switch(config)#
```

Table 23-12 displays the default CoS to Traffic Class map on FM6000 platform switches.

Table 23-12 Default CoS to Traffic Class Map: FM6000 Platform Switches

Inbound CoS	Untagged	0	1	2	3	4	5	6	7
Traffic Class	Derived: use default CoS as inbound CoS	1	0	2	3	4	5	6	7

Mapping DSCP to Traffic Class

The **qos map dscp** command assigns a traffic class to a set of DSCP values. Multiple commands create a complete DSCP to traffic class map. The switch uses this map to assign a traffic class to data packets on the basis of the packet's DSCP field or the chip upon which it is received.

Example

- This command assigns the traffic class of three to the DSCP values of 12, 13, 25, and 37.

```
switch(config)#qos map dscp 12 13 25 37 to traffic-class 3
switch(config)#show qos map
   Number of Traffic Classes supported: 8
             <-------OUTPUT OMITTED FROM EXAMPLE-------->

   Dscp-tc map:
     d1 :  d2 0  1  2  3  4  5  6  7  8  9
     ----------------------------------------
      0 :     1  1  1  1  1  1  1  1  0  0
      1 :     0  0  3  3  0  0  2  2  2  2
      2 :     2  2  2  2  3  3  3  3  3  3
      3 :     3  3  4  4  4  4  4  3  4  4
      4 :     5  5  5  5  5  5  5  5  6  6
      5 :     6  6  6  6  6  6  7  7  7  7
      6 :     7  7  7  7

             <-------OUTPUT OMITTED FROM EXAMPLE-------->
switch(config)#
```

Table 23-13 displays the default DSCP to Traffic Class map on FM6000 platform switches.

Table 23-13 Default DSCP to Traffic Class Map: FM6000 Platform Switches

Inbound DSCP	0-7	8-15	16-23	24-31	32-39	40-47	48-55	56-63
Traffic Class	1	0	2	3	4	5	6	7

23.4.3 CoS and DSCP Rewrite – FM6000 Platform Switches

Section 23.1.1.3 describes the CoS and DSCP rewrite functions.

Traffic Class to CoS Rewrite Map

The CoS rewrite value is configurable and based on a data stream's traffic class, as specified by the traffic class-CoS rewrite map. The **qos map traffic-class to cos** command assigns a CoS rewrite value to a list of traffic classes. Multiple commands create the complete traffic class–CoS rewrite map.

Example

- This command assigns the CoS rewrite value of two to traffic classes 1, 3, and 5.

```
switch(config)#qos map traffic-class 1 3 5 to cos 2
switch(config)#show qos map
   Number of Traffic Classes supported: 8
             <-------OUTPUT OMITTED FROM EXAMPLE-------->

   Tc - tx-queue map:
     tc:        0  1  2  3  4  5  6  7
     ---------------------------------
     tx-queue:  0  1  2  3  4  5  6  7

             <-------OUTPUT OMITTED FROM EXAMPLE-------->
switch(config)#
```

Table 23-14 displays the default traffic class–CoS rewrite map on FM6000 platform switches.

Table 23-14 Default Traffic Class to CoS Rewrite Map: FM6000 Platform Switches

Traffic Class	0	1	2	3	4	5	6	7
CoS Rewrite Value	1	0	2	3	4	5	6	7

Traffic Class to DSCP Rewrite Map

The DSCP rewrite value is configurable and based on a data stream's traffic class, as specified by the traffic class-DSCP rewrite map. The **qos map traffic-class to dscp** command assigns a DSCP rewrite value to a list of traffic classes. Multiple commands create the complete traffic class-DSCP rewrite map.

Example

- This command assigns the DSCP rewrite value of 37 to traffic classes 2, 4, and 6.

```
switch(config)#qos map traffic-class 2 4 6 to dscp 37
switch(config)#show qos map
   Number of Traffic Classes supported: 8
             <-------OUTPUT OMITTED FROM EXAMPLE-------->

   Tc-dscp map:
     tc:    0  1  2  3  4  5  6  7
     ----------------------------
     dscp:  8  0 37 24 37 40 37 56

             <-------OUTPUT OMITTED FROM EXAMPLE-------->
switch(config)#
```

Table 23-15 displays the default traffic class–DSCP rewrite map on on FM6000 platform switches.

Table 23-15 Default Traffic Class to DSCP Rewrite Map: FM6000 Platform Switches

Traffic Class	0	1	2	3	4	5	6	7
DSCP Rewrite Value	8	0	16	24	32	40	48	56

23.4.4 Transmit Queues and Port Shaping – FM6000 Platform Switches

Section 23.1.2 describes transmit queues and port shaping.

A data stream's traffic class determines the transmit queue it uses. The switch defines a single traffic class-transmit queue map for all Ethernet and port channel interfaces and is used for unicast and multicast traffic. The **show qos maps** command displays the traffic class to transmit queue map.

Table 23-16 displays the default traffic class to transmit queue map on FM6000 platform switches.

Table 23-16 Default Traffic Class to Transmit Queue Map: FM6000 Platform Switches

Traffic Class	0	1	2	3	4	5	6	7
Transmit Queue	0	1	2	3	4	5	6	7

Mapping Traffic Classes to a Transmit Queue

The **qos map traffic-class to tx-queue** command assigns traffic classes to a transmit queue. Multiple commands create the complete map.

Example

- These commands assign traffic classes of 1, 3, and 5 to transmit queue 1, traffic classes 2, 4, and 6 to transmit queue 2, and traffic class 0 to transmit queue 0, then display the resultant map.

```
switch(config)#qos map traffic-class 1 3 5 to tx-queue 1
switch(config)#qos map traffic-class 2 4 6 to tx-queue 2
switch(config)#qos map traffic-class 0 to tx-queue 0
switch(config)#show qos maps
   Number of Traffic Classes supported: 8
   Number of Transmit Queues supported: 8
             <-------OUTPUT OMITTED FROM EXAMPLE-------->

   Tc - tx-queue map:
     tc:        0  1  2  3  4  5  6  7
     --------------------------------
     tx-queue:  0  1  2  1  2  1  2  7

switch(config)#
```

Entering TX-Queue Configuration Mode

Transmit queues are configurable on Ethernet ports and port channels. Queue parameters are configured in tx-queue configuration command mode, which is entered from interface ethernet configuration mode. The **tx-queue (FM6000)** command places the switch in tx-queue configuration mode. The **show qos interfaces** displays the transmit queue configuration for a specified port.

Example

- This command enters tx-queue configuration mode for transmit queue 3 of Ethernet interface 5.

```
switch(config)#interface ethernet 5
switch(config-if-Et5)#tx-queue 3
switch(config-if-Et5-txq-3)#
```

Configuring the Shape Rate – Port and Transmit Queues

A port's shape rate specifies its maximum outbound traffic bandwidth. A transmit queue's shape rate specifies the queue's maximum outbound bandwidth. Shape rate commands specify data rates in kbps.

Important! Enabling port shaping on an FM6000 interface disables queue shaping internally. Disabling port shaping restores queue shaping as specified in ***running-config***.

- To configure a port's shape rate, enter **shape rate (Interface – FM6000)** from the port's interface configuration mode.
- To configure a transmit queue's shape rate, enter **shape rate (Tx-queue – FM6000)** from the queue's tx-queue configuration mode.

Example

- These commands configure a shape rate of 5 Gbs on Ethernet port 3, then configure the shape rate for the following transmit queues:
  - transmit queues 0, 1, and 2: 500 Mbps

- transmit queues 3, 4, and 5: 400 Mbps

```
switch(config)#interface ethernet 3
switch(config-if-Et3)#shape rate 5000000
switch(config-if-Et3)#tx-queue 0
switch(config-if-Et3-txq-0)#shape rate 500000
switch(config-if-Et3-txq-0)#tx-queue 1
switch(config-if-Et3-txq-1)#shape rate 500000
switch(config-if-Et3-txq-1)#tx-queue 3
switch(config-if-Et3-txq-3)#shape rate 400000
switch(config-if-Et3-txq-3)#tx-queue 4
switch(config-if-Et3-txq-4)#shape rate 400000
switch(config-if-Et3-txq-4)#tx-queue 5
switch(config-if-Et3-txq-5)#shape rate 400000
switch(config-if-Et3-txq-5)#exit
switch(config-if-Et3)#show qos interface ethernet 3
Ethernet3:
             <-------OUTPUT OMITTED FROM EXAMPLE-------->

   Port shaping rate: 5000000Kbps

   Tx-Queue   Bandwidth    Shape Rate     Priority
              (percent)       (Kbps)
   ---------------------------------------------
          7         N/A     disabled        strict
          6         N/A     disabled        strict
          5         N/A       400000        strict
          4         N/A       400000        strict
          3         N/A       400000        strict
          2         N/A     disabled        strict
          1         N/A       500000        strict
          0         N/A       500000        strict

switch(config-if-Et3)#
```

Configuring Queue Priority

Queue priority rank is denoted by the queue number; transmit queues with higher numbers have higher priority. The **priority (FM6000)** command configures a transmit queue's priority type:

- **priority strict** configures the queue as a strict priority queue.
- **no priority** configures the queue as a round robin queue.

  A queue's configuration as ***round robin*** also applies to all lower priority queues regardless of other configuration statements.

The **bandwidth percent (FM6000)** command configures a round robin queue's bandwidth share. The cumulative operational bandwidth of all round robin queues is always less than or equal to 100%. If the cumulative configured bandwidth is greater than 100%, each port's operational bandwidth is its configured bandwidth divided by the cumulative configured bandwidth.

Example

- These commands configure transmit queue 3 (on Ethernet interface 19) as a round robin queue, then allocates 10%, 20%, 30%, and 40% bandwidth to queues 0 through 3.

The **no priority** statement for queue 3 also configures queues 0, 1, and 2 as round robin queues. Removing this statement reverts the other queues to ***strict priority*** type unless ***running-config*** contains a **no priority** statement for one of these queues.

```
switch(config)#interface ethernet 19
switch(config-if-Et19)#tx-queue 3
switch(config-if-Et19-txq-3)#no priority
switch(config-if-Et19-txq-3)#bandwidth percent 40
switch(config-if-Et19-txq-3)#tx-queue 2
switch(config-if-Et19-txq-2)#bandwidth percent 30
switch(config-if-Et19-txq-2)#tx-queue 1
switch(config-if-Et19-txq-1)#bandwidth percent 20
switch(config-if-Et19-txq-1)#tx-queue 0
switch(config-if-Et19-txq-0)#bandwidth percent 10
switch(config-if-Et19-txq-0)#show qos interface ethernet 19
Ethernet19:
            <-------OUTPUT OMITTED FROM EXAMPLE-------->
   Port shaping rate: disabled

   Tx-Queue   Bandwidth    Shape Rate     Priority
              (percent)       (Kbps)
   ---------------------------------------------
          7         N/A     disabled        strict
          6         N/A     disabled        strict
          5         N/A     disabled        strict
          4         N/A     disabled        strict
          3          40     disabled   round-robin
          2          30     disabled   round-robin
          1          20     disabled   round-robin
          0          10     disabled   round-robin

switch(config-if-Et19-txq-0)#
```

Changing the bandwidth percentage for queue 3 to 60 changes the operational bandwidth of each queue to its configured bandwidth divided by 120% (10%+20%+30%+60%).

```
switch(config-if-Et19-txq-0)#tx-queue 3
switch(config-if-Et19-txq-3)#bandwidth percent 60
switch(config-if-Et19-txq-3)#show qos interface ethernet 19
Ethernet19:
            <-------OUTPUT OMITTED FROM EXAMPLE-------->
   Port shaping rate: disabled

   Tx-Queue   Bandwidth    Shape Rate     Priority
              (percent)       (Kbps)
   ---------------------------------------------
          7         N/A     disabled        strict
          6         N/A     disabled        strict
          5         N/A     disabled        strict
          4         N/A     disabled        strict
          3          49     disabled   round-robin
          2          24     disabled   round-robin
          1          16     disabled   round-robin
          0           8     disabled   round-robin

switch(config-if-Et19-txq-3)#
```

23.5 QoS Configuration: Petra Platform Switches

Implementing QoS on a Petra platform switch consists of configuring port trust settings, default port settings, default traffic classes, conversion maps, and transmit queues.

- Section 23.5.1: CoS and DSCP Port Settings – Petra Platform Switches
- Section 23.5.2: Traffic Class Derivations – Petra Platform Switches
- Section 23.5.3: CoS Rewrite – Petra Platform Switches
- Section 23.5.4: Transmit Queues and Port Shaping – Petra Platform Switches

23.5.1 CoS and DSCP Port Settings – Petra Platform Switches

Section 23.1.1.2 describes port trust and default port CoS and DSCP values.

Configuring Port Trust Settings

The **qos trust** command configures the QoS port trust mode for the configuration mode interface. Trust enabled ports use packet CoS or DSCP values to classify traffic. The port-trust default for switched ports is ***cos***. The port-trust default for routed ports is ***dscp***.

- **qos trust cos** specifies ***cos*** as the port's port-trust mode.
- **qos trust dscp** specifies ***dscp*** as the port's port-trust mode.
- **no qos trust** specifies ***untrusted*** as the port's port-trust mode.

The **show qos interfaces trust** command displays the trust mode of specified interfaces.

Example

- These commands configure and display the following trust modes:
  - Ethernet 3/25: dscp
  - Ethernet 3/26: untrusted
  - Ethernet 3/27: cos
  - Ethernet 3/28: default as a switched port

- Ethernet 3/29: default as a routed port

```
switch(config)#interface ethernet 3/25
switch(config-if-Et3/25)#qos trust dscp
switch(config-if-Et3/25)#interface ethernet 3/26
switch(config-if-Et3/26)#no qos trust
switch(config-if-Et3/26)#interface ethernet 3/27
switch(config-if-Et3/27)#qos trust cos
switch(config-if-Et3/27)#interface ethernet 3/28
switch(config-if-Et3/28)#switchport
switch(config-if-Et3/28)#default qos trust
switch(config-if-Et3/28)#interface ethernet 3/29
switch(config-if-Et3/29)#no switchport
switch(config-if-Et3/29)#default qos trust
switch(config-if-Et3/29)#show qos interface ethernet 3/25 - 3/29 trust
Port                                    Trust Mode
                              Operational           Configured
-------------------------------------------------------------
Ethernet3/25                  DSCP                  DSCP
Ethernet3/26                  UNTRUSTED             UNTRUSTED
Ethernet3/27                  COS                   COS
Ethernet3/28                  COS                   DEFAULT
Ethernet3/29                  DSCP                  DEFAULT

switch(config-if-Et3/29)#
```

Configuring Default Port Settings

Port channel and Ethernet interfaces are not assigned default CoS or DSCP settings.

23.5.2 Traffic Class Derivations – Petra Platform Switches

Section 23.1.1.4 describes traffic classes.

Traffic Class Derivation Source

Table 23-17 displays the source for deriving a data stream's default traffic class.

Table 23-17 Traffic Class Derivation Source: Petra Platform Switches

	Untrusted	CoS Trusted	DSCP Trusted
Untagged Non-IP	Default TC (chip)	Default TC (chip)	Default TC (chip)
Untagged IP	Default TC (chip)	Default TC (chip)	DSCP (packet)
Tagged Non-IP	Default TC (chip)	CoS (packet)	Default TC (chip)
Tagged IP	Default TC (chip)	CoS (packet)	DSCP (packet)

Configuring Default Traffic Class

Petra platform switches assign a default traffic class to the set of Ethernet interfaces controlled by individual PetraA chips. Default traffic class values are configurable for each PetraA chip, not individual interfaces.

The **platform petraA traffic-class** command specifies the default traffic class used by all ports controlled by a specified chip. The **show platform petraA traffic-class** command displays traffic class assignments.

Example

- This command configures the default traffic class to five for the ports 32-39 on linecard 3 (7500 Series).

```
switch(config)#platform petraA petra3/4 traffic-class 5
switch(config)#show platform petraA module 3 traffic-class
Petra3/0 traffic-class: 1
Petra3/1 traffic-class: 1
Petra3/2 traffic-class: 1
Petra3/3 traffic-class: 1
Petra3/4 traffic-class: 5
Petra3/5 traffic-class: 1
switch(config)#
```

- This command configures the default traffic class to three for all ports on linecard 6 (7500 Series).

```
switch(config)#platform petraA module 6 traffic-class 6
switch(config)#show platform petraA module 6 traffic-class
Petra6/0 traffic-class: 6
Petra6/1 traffic-class: 6
Petra6/2 traffic-class: 6
Petra6/3 traffic-class: 6
Petra6/4 traffic-class: 6
Petra6/5 traffic-class: 6
switch(config)#
```

Mapping CoS to Traffic Class

The **qos map cos** command assigns a traffic class to a list of CoS settings. Multiple commands create a complete CoS–traffic class map. The switch uses this map to assign a traffic class to data packets on the basis of the packet's CoS field or the port upon which it is received.

Example

- This command assigns the traffic class of 4 to the classes of service 1, 3, 5, and 7.

```
switch(config)#qos map cos 1 3 5 7 to traffic-class 4
switch(config)#show qos maps
   Number of Traffic Classes supported: 8
             <-------OUTPUT OMITTED FROM EXAMPLE-------->

   Cos-tc map:
     cos:  0  1  2  3  4  5  6  7
     ---------------------------
     tc:   1  4  2  4  4  4  6  4

             <-------OUTPUT OMITTED FROM EXAMPLE-------->
switch(config)#
```

Table 23-18 displays the default CoS to traffic class map on Petra platform switches.

Table 23-18 Default CoS to Traffic Class Map: Petra Platform Switches

Inbound CoS	untagged	0	1	2	3	4	5	6	7
Traffic Class	Derived: use default CoS as inbound CoS	1	0	2	3	4	5	6	7

Mapping DSCP to Traffic Class

The **qos map dscp** command assigns a traffic class to a set of DSCP values. Multiple commands create a complete DSCP to traffic class map. The switch uses this map to assign a traffic class to data packets on the basis of the packet's DSCP field or the chip upon which it is received.

Example

- This command assigns the traffic class of three to the DSCP values of 12, 13, 25, and 37.

```
switch(config)#qos map dscp 12 13 14 25 48 to traffic-class 3
switch(config)#show qos maps
   Number of Traffic Classes supported: 8
             <-------OUTPUT OMITTED FROM EXAMPLE-------->

   Dscp-tc map:
     d1 :  d2 0  1  2  3  4  5  6  7  8  9
     ----------------------------------
      0 :     1  1  1  1  1  1  1  1  0  0
      1 :     0  0  3  3  3  0  2  2  2  2
      2 :     2  2  2  2  3  3  3  3  3  3
      3 :     3  3  4  4  4  4  4  4  4  4
      4 :     5  5  5  5  5  5  5  5  3  6
      5 :     6  6  6  6  6  6  7  7  7  7
      6 :     7  7  7  7

             <-------OUTPUT OMITTED FROM EXAMPLE-------->
switch(config)#
```

Table 23-19 displays the default DSCP to Traffic Class map on Petra platform switches.

Table 23-19 Default DSCP to Traffic Class Map: Petra Platform Switches

Inbound DSCP	0-7	8-15	16-23	24-31	32-39	40-47	48-55	56-63
Traffic Class	1	0	2	3	4	5	6	7

23.5.3 CoS Rewrite – Petra Platform Switches

Section 23.1.1.3 describes the CoS rewrite function.

Traffic Class to CoS Rewrite Map

The CoS rewrite value is configurable and based on a data stream's traffic class, as specified by the traffic class-CoS rewrite map. The **qos map traffic-class to cos** command assigns a CoS rewrite value to a list of traffic classes. Multiple commands create the complete traffic class-CoS rewrite map.

Example

- This command assigns the CoS of two to traffic classes 1, 3, and 5.

```
switch(config)#qos map traffic-class 1 3 5 to cos 2
switch(config)#show qos map
   Number of Traffic Classes supported: 8
             <-------OUTPUT OMITTED FROM EXAMPLE-------->

   Tc-cos map:
     tc:   0  1  2  3  4  5  6  7
     ---------------------------
     cos:  1  2  2  2  4  2  6  7

             <-------OUTPUT OMITTED FROM EXAMPLE-------->
switch(config)#
```

Table 23-20 displays the default Traffic Class to CoS rewrite value map on Petra platform switches.

Table 23-20 Default Traffic Class to CoS Rewrite Value Map: Petra Platform Switches

Traffic Class	0	1	2	3	4	5	6	7
CoS Rewrite Value	1	0	2	3	4	5	6	7

Traffic Class to DSCP Rewrite Map

DSCP rewrite is always disabled on Petra platform switches.

23.5.4 Transmit Queues and Port Shaping – Petra Platform Switches

Section 23.1.2 describes transmit queues and port shaping.

Petra platform switches provide four physical queues for each egress port: Unicast High, Unicast Low, Multicast High, and Multicast Low. Data is scheduled for the high or low queue based on its priority as defined by its transmit queue assignment (unicast traffic) or traffic class (multicast traffic), as shown in Table 23-21. A Petra transmit queue is a data structure that defines scheduling of unicast traffic among phyical egress queues.

Table 23-21 Traffic Distribution to Egress Port Queues

	High Priority Queue	Low Priority Queue
Unicast Traffic	Transmit Queues 5 – 7	Transmit Queues 0 – 4
Multicast Traffic	Traffic Classes 5 – 7	Traffic Classes 0 – 4

Multicast queue capacity that is available after multicast traffic is serviced is used for unicast traffic of a corresponding priority. Similarly, unicast queue capacity that is available after unicast traffic is serviced is used for overflow multicast traffic. Under conditions of unicast and multicast congestion, egress traffic is evenly split between unicast and multicast traffic.

Section 23.5.4.1 describes unicast transmit queues and shaping. Section 23.5.4.2 describes multicast priority and traffic classes.

23.5.4.1 Unicast Transmit Queues and Port Shaping

A data stream's traffic class determines the transmit queue it uses. The switch defines a single traffic class–transmit queue map for unicast traffic on all Ethernet interfaces. The **show qos maps** command displays the traffic class–transmit queue map. Table 23-22 displays the default traffic class to transmit queue map on Petra platform switches.

Table 23-22 Default Traffic Class to Transmit Queue Map: Petra Platform Switches

Traffic Class	0	1	2	3	4	5	6	7
Transmit Queue	0	1	2	3	4	5	6	7

Transmit queue parameters are configured in tx-queue configuration command mode.

Mapping Traffic Classes to a Transmit Queue

The **qos map traffic-class to tx-queue** command assigns traffic classes to a transmit queue. Multiple commands complete the traffic class-transmit queue map. Traffic class 7 and transmit queue 7 are always mapped to each other. This association is not editable.

Example

- These commands assign traffic classes of 1, 3, and 5 to transmit queue 1, traffic classes 2, 4, and 6 to transmit queue 2, and traffic class 0 to transmit queue 0, then display the resultant map.

```
switch(config)#qos map traffic-class 1 3 5 to tx-queue 1
switch(config)#qos map traffic-class 2 4 6 to tx-queue 2
switch(config)#qos map traffic-class 0 to tx-queue 0
switch(config)#show qos maps
   Number of Traffic Classes supported: 8
   Number of Transmit Queues supported: 8
             <-------OUTPUT OMITTED FROM EXAMPLE-------->
   Tc - tx-queue map:
     tc:        0  1  2  3  4  5  6  7
     --------------------------------
     tx-queue:  0  1  2  1  2  1  2  7

switch(config)#
```

Entering Tx-Queue Configuration Mode

The **tx-queue (Petra)** command places the switch in tx-queue configuration mode to configure a transmit queue on the configuration mode interface. Tx-queue 7 is not configurable. The **show qos interfaces** displays the transmit queue configuration for a specified port.

Example

- This command enters tx-queue configuration mode for transmit queue 3 of Ethernet interface 3/28

```
switch(config)#interface ethernet 3/28
switch(config-if-Et3/28)#tx-queue 3
switch(config-if-Et3/28-txq-3)#
```

Configuring the Shape Rate – Port and Transmit Queues

A port's shape rate specifies its maximum outbound traffic bandwidth. A transmit queue's shape rate specifies the queue's maximum outbound bandwidth. Shape rate commands specify data rates in kbps.

- To configure a port's shape rate, enter **shape rate (Interface – Petra)** from the port's interface configuration mode.
- To configure a transmit queue's shape rate, enter **shape rate (Tx-queue – Petra)** from the queue's tx-queue configuration mode.

Example

- These commands configure a shape rate of 5 Gbs on Ethernet port 3, then configure the shape rate for the following transmit queues:
  - transmit queues 0, 1, and 2: 500 Mbps

- transmit queues 3, 4, and 5: 400 Mbps

```
switch(config)#interface ethernet 3/28
switch(config-if-Et3/28)#shape rate 5000000
switch(config-if-Et3/28)#tx-queue 0
switch(config-if-Et3/28-txq-0)#shape rate 500000
switch(config-if-Et3/28-txq-0)#tx-queue 1
switch(config-if-Et3/28-txq-1)#shape rate 500000
switch(config-if-Et3/28-txq-1)#tx-queue 2
switch(config-if-Et3/28-txq-2)#shape rate 500000
switch(config-if-Et3/28-txq-5)#tx-queue 3
switch(config-if-Et3/28-txq-3)#shape rate 400000
switch(config-if-Et3/28-txq-3)#tx-queue 4
switch(config-if-Et3/28-txq-4)#shape rate 400000
switch(config-if-Et3/28-txq-4)#tx-queue 5
switch(config-if-Et3/28-txq-5)#shape rate 400000
switch(config-if-Et3/28-txq-5)#show qos interface ethernet 3/28
Ethernet3/28:
             <-------OUTPUT OMITTED FROM EXAMPLE-------->

   Port shaping rate: 5000000Kbps

   Tx-Queue   Bandwidth    Shape Rate     Priority
              (percent)       (Kbps)
   ---------------------------------------------
          7         N/A     disabled        strict
          6         N/A     disabled        strict
          5         N/A       400000        strict
          4         N/A       400000        strict
          3         N/A       400000        strict
          2         N/A       500000        strict
          1         N/A       500000        strict
          0         N/A       500000        strict

switch(config-if-Et3/28-txq-5)#
```

Configuring Queue Priority

The **priority (Petra)** command configures a transmit queue's priority type:

- The **priority strict** command configures the queue as a strict priority queue.
- The **no priority** command configures the queue as a round robin queue.

  A queue's configuration as ***round robin*** also applies to all lower priority queues regardless of other configuration statements.

The **bandwidth percent (Petra)** command configures a round robin queue's bandwidth share. The cumulative operational bandwidth of all round robin queues is always less than or equal to 100%. If the cumulative configured bandwidth is greater than 100%, each port's operational bandwidth is its configured bandwidth divided by the cumulative configured bandwidth.

Example

- These commands configure transmit queue 3 (on Ethernet interface 3/28) as a round robin queue, then allocates 10%, 20%, 30%, and 40% bandwidth to queues 0 through 3.

The **no priority** statement for queue 3 also configures queues 0, 1, and 2 as round robin queues. Removing this statement reverts the other queues to ***strict priority*** type unless ***running-config*** contains a **no priority** statement for one of these queues.

```
switch(config-if-Et3/28)#tx-queue 3
switch(config-if-Et3/28-txq-3)#no priority
switch(config-if-Et3/28-txq-3)#bandwidth percent 40
switch(config-if-Et3/28-txq-3)#tx-queue 2
switch(config-if-Et3/28-txq-2)#bandwidth percent 30
switch(config-if-Et3/28-txq-2)#tx-queue 1
switch(config-if-Et3/28-txq-1)#bandwidth percent 20
switch(config-if-Et3/28-txq-1)#tx-queue 0
switch(config-if-Et3/28-txq-0)#bandwidth percent 10
switch(config-if-Et3/28-txq-0)#show qos interface ethernet 3/28
Ethernet3/28:
            <-------OUTPUT OMITTED FROM EXAMPLE-------->
   Port shaping rate: 5000000Kbps

   Tx-Queue   Bandwidth    Shape Rate     Priority
              (percent)       (Kbps)
   ------------------------------------------------
          7         N/A     disabled        strict
          6         N/A     disabled        strict
          5         N/A       400000        strict
          4         N/A       400000        strict
          3          40       400000   round-robin
          2          30       500000   round-robin
          1          20       500000   round-robin
          0          10       500000   round-robin

switch(config-if-Et3/28-txq-0)#
```

Changing the bandwidth percentage for queue 3 to 60 changes the operational bandwidth of each queue to its configured bandwidth divided by 120% (10%+20%+30%+60%).

```
switch(config-if-Et3/28-txq-0)#tx-queue 3
switch(config-if-Et3/28-txq-3)#bandwidth percent 60
switch(config-if-Et3/28-txq-3)#show qos interface ethernet 3/28
Ethernet3/28:
            <-------OUTPUT OMITTED FROM EXAMPLE-------->
   Port shaping rate: 5000000Kbps

   Tx-Queue   Bandwidth    Shape Rate     Priority
              (percent)       (Kbps)
   ------------------------------------------------
          7         N/A     disabled        strict
          6         N/A     disabled        strict
          5         N/A       400000        strict
          4         N/A       400000        strict
          3          49       400000   round-robin
          2          24       500000   round-robin
          1          16       500000   round-robin
          0           8       500000   round-robin
switch(config-if-Et3/28-txq-3)#
```

23.5.4.2 Multicast Egress Scheduling

Multiclass traffic is not affected by traffic class assignment or port shaping statements. Multicast traffic is assigned to port egress queues based on traffic class and uses strict priority to schedule egress between the high and low queues.

23.6 QoS Configuration: Trident Platform Switches

Implementing QoS on a Trident platform switch consists of configuring port trust settings, default port settings, default traffic classes, conversion maps, and transmit queues.

- Section 23.6.1: CoS and DSCP Port Settings – Trident Platform Switches
- Section 23.6.2: Traffic Class Derivations – Trident Platform Switches
- Section 23.6.3: CoS and DSCP Rewrite – Trident Platform Switches
- Section 23.6.4: Transmit Queues and Port Shaping – Trident Platform Switches
- Section 23.6.5: ECN Configuration – Trident Platform Switches

23.6.1 CoS and DSCP Port Settings – Trident Platform Switches

Configuring Port Trust Settings

The **qos trust** command configures the QoS port trust mode for the configuration mode interface. Trust-enabled ports use packet CoS or DSCP values to classify traffic. The port-trust default for switched ports is ***CoS***. The port-trust default for routed ports is ***DSCP***.

- **qos trust cos** specifies ***CoS*** as the port's trust mode.
- **qos trust dscp** specifies ***DSCP*** as the port's trust mode.
- **no qos trust** specifies ***untrusted*** as the port's trust mode.

The **show qos interfaces trust** command displays the trust mode of specified interfaces.

Example

- These commands configure and display the following trust modes:
  - Ethernet 15: dscp
  - Ethernet 16: untrusted
  - Ethernet 17: cos
  - Ethernet 18: default as a switched port

- Ethernet 19: default as a routed port

```
switch(config)#interface ethernet 15
switch(config-if-Et15)#qos trust dscp
switch(config-if-Et15)#interface ethernet 16
switch(config-if-Et16)#no qos trust
switch(config-if-Et16)#interface ethernet 17
switch(config-if-Et17)#qos trust cos
switch(config-if-Et17)#interface ethernet 18
switch(config-if-Et18)#switchport
switch(config-if-Et18)#default qos trust
switch(config-if-Et18)#interface ethernet 19
switch(config-if-Et19)#no switchport
switch(config-if-Et19)#default qos trust
switch(config-if-Et19)#show qos interface ethernet 15 - 19 trust
Port                                        Trust Mode
                                Operational           Configured
---------------------------------------------------------------
Ethernet15                      DSCP                  DSCP
Ethernet16                      UNTRUSTED             UNTRUSTED
Ethernet17                      COS                   COS
Ethernet18                      COS                   DEFAULT
Ethernet19                      DSCP                  DEFAULT

switch(config-if-Et19)#
```

Configuring Default Port Settings

Default CoS and DSCP settings are assigned to individual port channel and Ethernet interfaces. These configuration mode interface commands specify the port's default CoS and DSCP values.

- **qos cos** configures a port's default CoS value.
- **qos dscp** configures a port's default DSCP value.

Example

- These commands configure default CoS (4) and DSCP (44) values on Ethernet interface 7.

```
switch(config)#interface ethernet 7
switch(config-if-Et7)#qos cos 4
switch(config-if-Et7)#qos dscp 44
switch(config-if-Et7)#show active
interface Ethernet7
   qos cos 4
   qos dscp 44
switch(config-if-Et7)#show qos interfaces ethernet 7
Ethernet7:
   Trust Mode: COS
   Default COS: 4
   Default DSCP: 44
             <-------OUTPUT OMITTED FROM EXAMPLE-------->
switch(config-if-Et7)#
```

23.6.2 Traffic Class Derivations – Trident Platform Switches

Section 23.1.1.4 describes traffic classes.

Traffic Class Derivation Source

Table 23-23 displays the source for deriving a data stream's traffic class.

Table 23-23 Traffic Class Derivation Source: Trident Platform Switches

	Untrusted	CoS Trusted	DSCP Trusted
Untagged Non-IP	Default CoS (port)	Default CoS (port)	Default DSCP (port)
Untagged IP	Default CoS (port)	Default CoS (port)	DSCP (packet)
Tagged Non-IP	Default CoS (port)	CoS (packet)	Default DSCP (port)
Tagged IP	Default CoS (port)	CoS (packet)	DSCP (packet)

Section 23.6.1 describes the default CoS and DSCP settings for each port.

Mapping CoS to Traffic Class

The **qos map cos** command assigns a traffic class to a list of CoS settings. Multiple commands create a complete CoS to traffic class map. The switch uses this map to assign a traffic class to data packets on the basis of the packet's CoS field or the port upon which it is received.

Example

- This command assigns the traffic class of 5 to the classes of service 1, 3, 5, and 7.

```
switch(config)#qos map cos 1 3 5 7 to traffic-class 5
switch(config)#show qos maps
   Number of Traffic Classes supported: 8
             <-------OUTPUT OMITTED FROM EXAMPLE-------->

   Cos-tc map:
     cos:  0  1  2  3  4  5  6  7
     ----------------------------
     tc:   1  5  2  5  4  5  6  5

             <-------OUTPUT OMITTED FROM EXAMPLE-------->
switch(config)#
```

Table 23-24 displays the default CoS–traffic class map on Trident platform switches.

Table 23-24 Default CoS to Traffic Class Map: Trident Platform Switches

Inbound CoS	0	1	2	3	4	5	6	7
Traffic Class	1	0	2	3	4	5	6	7

Mapping DSCP to Traffic Class

The **qos map dscp** command assigns a traffic class to a set of DSCP values. Multiple commands create a complete DSCP to traffic class map. The switch uses this map to assign a traffic class to data packets on the basis of the packet's DSCP field or the chip upon which it is received.

Example

- This command assigns the traffic class of 0 to DSCP values of 12, 24, 41, and 44-47.

```
switch(config)#qos map dscp 12 24 41 44 45 46 47 to traffic-class 0
switch(config)#show qos maps
   Number of Traffic Classes supported: 8
             <-------OUTPUT OMITTED FROM EXAMPLE-------->

   Dscp-tc map:
     d1 :  d2 0  1  2  3  4  5  6  7  8  9
     ----------------------------------------
      0 :     1  1  1  1  1  1  1  1  0  0
      1 :     0  0  0  0  0  0  2  2  2  2
      2 :     2  2  2  2  0  3  3  3  3  3
      3 :     3  3  4  4  4  4  4  4  4  4
      4 :     5  0  5  5  0  0  0  0  6  6
      5 :     6  6  6  6  6  6  7  7  7  7
      6 :     7  7  7  7

             <-------OUTPUT OMITTED FROM EXAMPLE-------->
switch(config)#
```

Table 23-25 displays the default DSCP–traffic class map on Trident platform switches.

Table 23-25 Default DSCP to Traffic Class Map: Trident Platform Switches

Inbound DSCP	0-7	8-15	16-23	24-31	32-39	40-47	48-55	56-63
Traffic Class	1	0	2	3	4	5	6	7

23.6.3 CoS and DSCP Rewrite – Trident Platform Switches

Section 23.1.1.3 describes the CoS and DSCP rewrite functions.

Traffic Class to CoS Rewrite Map

The CoS rewrite value is configurable and based on a data stream's traffic class, as specified by the traffic class-CoS rewrite map. The **qos map traffic-class to cos** command assigns a CoS rewrite value to a list of traffic classes. Multiple commands create the complete traffic class–CoS rewrite map.

Example

- This command assigns the CoS of two to traffic classes 1, 3, and 5.

```
switch(config)#qos map traffic-class 1 3 5 to cos 2
switch(config)#show qos map
   Number of Traffic Classes supported: 8


   Tc-cos map:
     tc:   0  1  2  3  4  5  6  7
     ----------------------------
     cos:  1  2  2  2  4  2  6  7

             <-------OUTPUT OMITTED FROM EXAMPLE-------->
switch(config)#
```

Table 23-26 displays the default Traffic Class to CoS rewrite value map on Trident platform switches.

Table 23-26 Default Traffic Class to CoS Rewrite Value Map: Trident Platform Switches

Traffic Class	0	1	2	3	4	5	6	7
CoS Rewrite Value	1	0	2	3	4	5	6	7

Traffic Class to DSCP Rewrite Map

The DSCP rewrite value is configurable and based on a data stream's traffic class, as specified by the traffic class-DSCP rewrite map. The **qos map traffic-class to dscp** command assigns a DSCP rewrite value to a list of traffic classes. Multiple commands create the complete traffic class-DSCP rewrite map.

Example

- This command assigns the DSCP value of 29 to traffic classes 2, 4, and 6.

```
switch(config)#qos map traffic-class 2 4 6 to dscp 29
switch(config)#show qos map
   Number of Traffic Classes supported: 8
             <-------OUTPUT OMITTED FROM EXAMPLE-------->

   Tc-dscp map:
     tc:    0  1  2  3  4  5  6  7
     ----------------------------
     dscp:  8  0 29 24 29 40 29 56

             <-------OUTPUT OMITTED FROM EXAMPLE-------->
switch(config)#
```

Table 23-27 displays the default traffic class to DSCP rewrite map on Trident platform switches.

Table 23-27 Traffic Class to DSCP Rewrite Value Map: Trident Platform Switches

Traffic Class	0	1	2	3	4	5	6	7
DSCP	8	0	16	24	32	40	48	56

23.6.4 Transmit Queues and Port Shaping – Trident Platform Switches

Section 23.1.2 describes transmit queues and port shaping.

Trident platform switches define 12 transmit queues: eight unicast (UC0 – UC7) and four multicast (MC0 – MC03). The traffic class–transmit queue maps are configured globally and apply to all Ethernet interfaces. The **show qos maps** command displays the traffic class–transmit queue maps.

Table 23-28 displays the default traffic class–transmit queue maps.

Table 23-28 Default Traffic Class to Transmit Queue Map: Trident Platform Switches

Traffic Class	0	1	2	3	4	5	6	7
Unicast Transmit Queue	0	1	2	3	4	5	6	7
Multicast Transmit Queue	0	0	1	1	2	2	3	3

Mapping Traffic Classes to a Transmit Queue

These commands assign traffic classes to a transmit queue:

- **qos map traffic-class to uc-tx-queue** associates a unicast queue to a traffic class set.
- **qos map traffic-class to mc-tx-queue** associates a multicast queue to a traffic class set.

Multiple commands create the complete maps.

Example

- These commands assign the following on Ethernet interface 7:
  - traffic classes 1, 3, and 5 to unicast queue 1
  - traffic classes 2, 4, and 6 to unicast queue 5
  - traffic classes 1, 2, and 3 to multicast queue 1
  - traffic classes 4, 5, and 6 to multicast queue 3
  - traffic class 0 to unicast queue 0 and multicast queue 0

```
switch(config)#default interface ethernet 7
switch(config)#qos map traffic-class 1 3 5 to uc-tx-queue 1
switch(config)#qos map traffic-class 2 4 6 to uc-tx-queue 5
switch(config)#qos map traffic-class 1 2 3 to mc-tx-queue 1
switch(config)#qos map traffic-class 4 5 6 to mc-tx-queue 3
switch(config)#qos map traffic-class 0 to uc-tx-queue 0
switch(config)#qos map traffic-class 0 to mc-tx-queue 0
switch(config)#show qos maps
   Number of Traffic Classes supported: 8
   Number of Transmit Queues supported: 12
            <-------OUTPUT OMITTED FROM EXAMPLE-------->

   Tc - uc-tx-queue map:
     tc:           0  1  2  3  4  5  6  7
     ----------------------------------
     uc-tx-queue:  0  1  5  1  5  1  5  7

   Tc - mc-tx-queue map:
     tc:           0  1  2  3  4  5  6  7
     ----------------------------------
     mc-tx-queue:  0  1  1  1  3  3  3  3

switch(config)#
```

Entering a Transmit Queue Configuration Mode

Transmit queues are configurable on individual Ethernet ports. Parameters for individual transmit queues are configured in one of two transmit queue configuration modes. Transmit queue modes are accessed from an interface-ethernet configuration mode.

- **uc-tx-queue** places the switch in uc-tx-queue mode to configure a unicast transmit queue.
- **mc-tx-queue** places the switch in mc-tx-queue mode to configure a multicast transmit queue.

The **show qos interfaces** displays the transmit queue configuration for a specified port.Examples

Example

- This command enters the mode that configures unicast transmit queue 3 of Ethernet interface 5.

```
switch(config)#interface ethernet 5
switch(config-if-Et5)#uc-tx-queue 3
switch(config-if-Et5-uc-txq-3)#
```

- This command enters the mode to configure multicast transmit queue 3 of Ethernet interface 5.

```
switch(config-if-Et5)#mc-tx-queue 2
switch(config-if-Et5-mc-txq-2)#
```

Configuring the Shape Rate – Port and Transmit Queues

A port's shape rate specifies the port's maximum outbound traffic bandwidth. A shape rate can also be configured for all transmit queues on each port. All shape rate commands use kbps to specify data rates.

- To configure a port's shape rate, enter **shape rate (Interface – Trident)** from the port's interface configuration mode.
- To configure a transmit queue's shape rate, enter **shape rate (Tx-queues – Trident)** from the queue's tx-queue configuration mode.

Example

- These commands configure a shape rate of 5 Gbs on Ethernet port 7, then configure the shape rate for the following transmit queues:
  - unicast transmit queues 0 and 1: 500 Mbps
  - unicast transmit queues 3 and 4: 400 Mbps
  - multicast transmit queues 0 and 2: 300 Mbps

```
switch(config)#interface ethernet 7
switch(config-if-Et7)#shape rate 5000000
switch(config-if-Et7)#uc-tx-queue 0
switch(config-if-Et7-uc-txq-0)#shape rate 500000
switch(config-if-Et7-uc-txq-0)#uc-tx-queue 1
switch(config-if-Et7-uc-txq-1)#shape rate 500000
switch(config-if-Et7-uc-txq-1)#uc-tx-queue 3
switch(config-if-Et7-uc-txq-3)#shape rate 400000
switch(config-if-Et7-uc-txq-3)#uc-tx-queue 5
switch(config-if-Et7-uc-txq-5)#shape rate 400000
switch(config-if-Et7-uc-txq-5)#mc-tx-queue 0
switch(config-if-Et7-mc-txq-0)#shape rate 300000
switch(config-if-Et7-mc-txq-0)#mc-tx-queue 2
switch(config-if-Et7-mc-txq-2)#shape rate 300000
switch(config-if-Et7-mc-txq-2)#exit
switch(config-if-Et7)#show qos interface ethernet 7
Ethernet7:
             <-------OUTPUT OMITTED FROM EXAMPLE-------->

                Port shaping rate: 5000000Kbps

   Tx-Queue    Bandwidth    Shape Rate     Priority   Priority Group
               (percent)       (Kbps)
   ------------------------------------------------------------------
        UC7         N/A     disabled        strict                1
        UC6         N/A     disabled        strict                1
        MC3         N/A     disabled        strict                1
        UC5         N/A       400000        strict                0
        UC4         N/A     disabled        strict                0
        MC2         N/A       300000        strict                0
        UC3         N/A       400000        strict                0
        UC2         N/A     disabled        strict                0
        MC1         N/A     disabled        strict                0
        UC1         N/A       500000        strict                0
        UC0         N/A       500000        strict                0
        MC0         N/A       300000        strict                0

switch(config-if-Et7)#
```

Configuring Queue Priority

Trident platform switch queues are categorized into two priority groups. Priority group 1 queues have priority over priority 0 queues. The following lists display the priority group queues in order from higher priority to lower priority.

- Priority Group 1: UC7, UC6, MC3
- Priority Group 0: UC5, UC4, MC2, UC3, UC2, MC1, UC1, UC0, MC0

The **priority (Trident)** command configures a transmit queue's priority type:

- The **priority strict** command configures the queue as a strict priority queue.
- The **no priority** command configures the queue as a round robin queue.

A queue's configuration as ***round robin*** also applies to all lower priority queues regardless of other configuration statements.

The **bandwidth percent (Trident)** command configures a round robin queue's bandwidth share. The cumulative operational bandwidth of all round robin queues is always 100%. If the cumulative configured bandwidth is greater than 100%, each port's operational bandwidth is its configured bandwidth divided by the cumulative configured bandwidth.

Priority Group 1 queues (UC7, UC6, MC3) are not configurable as round robin queues. The **bandwidth percent** command is not available for these queues.

Example

- These commands configure unicast transmit queue 3 as a round robin queue, then allocates 5%, 15%, 25%, 35%, 8%, and 12% bandwidth to unicast transmit queues 0 through 3 and multicast transmit queues 0 and 1, respectively.

The **no priority** statement for queue 3 also configures priority for all lower priority queues. Removing the statement reverts the other queues to ***strict priority*** type unless ***running-config*** contains a **no priority** statement for one of these queues.

```
switch(config)#interface ethernet 7
switch(config-if-Et7)#uc-tx-queue 3
switch(config-if-Et7-uc-txq-3)#no priority
switch(config-if-Et7-uc-txq-3)#bandwidth percent 5
switch(config-if-Et7-uc-txq-3)#uc-tx-queue 2
switch(config-if-Et7-uc-txq-2)#bandwidth percent 15
switch(config-if-Et7-uc-txq-2)#uc-tx-queue 1
switch(config-if-Et7-uc-txq-1)#bandwidth percent 25
switch(config-if-Et7-uc-txq-1)#uc-tx-queue 0
switch(config-if-Et7-uc-txq-0)#bandwidth percent 35
switch(config-if-Et7-uc-txq-0)#mc-tx-queue 1
switch(config-if-Et7-mc-txq-1)#bandwidth percent 12
switch(config-if-Et7-mc-txq-1)#mc-tx-queue 0
switch(config-if-Et7-mc-txq-0)#bandwidth percent 8
switch(config-if-Et7-mc-txq-0)#show qos interface ethernet 7
Ethernet7:
             <-------OUTPUT OMITTED FROM EXAMPLE-------->

   Port shaping rate: disabled

   Tx-Queue   Bandwidth    Shape Rate     Priority   Priority Group
              (percent)       (Kbps)
   ------------------------------------------------------------------
        UC7         N/A     disabled        strict                1
        UC6         N/A     disabled        strict                1
        MC3         N/A     disabled        strict                1
        UC5         N/A     disabled        strict                0
        UC4         N/A     disabled        strict                0
        MC2         N/A     disabled        strict                0
        UC3           5     disabled   round-robin                0
        UC2          15     disabled   round-robin                0
        MC1          12     disabled   round-robin                0
        UC1          25     disabled   round-robin                0
        UC0          35     disabled   round-robin                0
        MC0           8     disabled   round-robin                0

switch(config-if-Et7-mc-txq-0)#
```

Changing the bandwidth percentage for unicast queue 3 to 30 changes the operational bandwidth of each queue to its configured bandwidth divided by 125% (8%+12%+30%+15%+25%+35%).

```
switch(config-if-Et7-uc-txq-0)#uc-tx-queue 3
switch(config-if-Et7-uc-txq-3)#bandwidth percent 30
switch(config-if-Et7-uc-txq-3)#show qos interface ethernet 7
Ethernet7:
             <-------OUTPUT OMITTED FROM EXAMPLE-------->

   Port shaping rate: disabled

   Tx-Queue    Bandwidth    Shape Rate     Priority   Priority Group
               (percent)       (Kbps)
   ------------------------------------------------------------------
        UC7          N/A     disabled        strict                1
        UC6          N/A     disabled        strict                1
        MC3          N/A     disabled        strict                1
        UC5          N/A     disabled        strict                0
        UC4          N/A     disabled        strict                0
        MC2          N/A     disabled        strict                0
        UC3           24     disabled   round-robin                0
        UC2           12     disabled   round-robin                0
        MC1            9     disabled   round-robin                0
        UC1           20     disabled   round-robin                0
        UC0           28     disabled   round-robin                0
        MC0            6     disabled   round-robin                0

switch(config-if-Et7-uc-txq-3)#
```

23.6.5 ECN Configuration – Trident Platform Switches

Section 23.1.3 describes Explicit Congestion Notification (ECN).

ECN is independently configurable on all egress queues of each Ethernet interface. ECN settings for Port-Channels are applied on each of the channel's member Ethernet interfaces. ECN is also globally configurable to mark packets from the shared pool used for dynamically allocating memory to the queues. Multicast packets contribute to the globally shared pool and can contribute to global level congestion that result in ECN marking of unicast packets queued after the multicast packets.

Average queue length is tracked for transmit queues and the global pool independently. When either entity reaches its maximum threshold, all subsequent packets are marked.

Although the switch does not limit the number of queues that can be configured for ECN, hardware table limitations restrict the number of queues (including the global shared pool) that can simultaneously implement ECN.

The **qos random-detect ecn global-buffer (Trident)** command enables ECN marking for globally shared packet memory and specifies minimum and maximum queue threshold sizes.

Example

- This command enables ECN marking of unicast packets from the global data pool and sets the minimum and maximum thresholds at 20 and 500 segments.

```
switch(config)#qos random-detect ecn global-buffer minimum-threshold 20 segments
maximum-threshold 500 segments
switch(config)#
```

- This command disables ECN marking of unicast packets from the global data pool

```
switch(config)#no qos random-detect ecn global-buffer
switch(config)#
```

The **random-detect ecn (Trident)** command enables ECN marking for the configuration mode unicast transmit queue and specifies threshold queue sizes.

Example

- These commands enable ECN marking of unicast packets from transmit queue 4 of Ethernet interface 15, setting thresholds at 10 and 100 segments.

```
switch(config)#interface ethernet 15
switch(config-if-Et15)#uc-tx-queue 4
switch(config-if-Et15-uc-txq-4)#random-detect ecn minimum-threshold 10 segments
maximum-threshold 100 segments
switch(config-if-Et15-uc-txq-4)#show active
interface Ethernet15
   uc-tx-queue 4
      random-detect ecn minimum-threshold 10 segments maximum-threshold 100
segments
switch(config-if-Et15-uc-txq-4)#exit
switch(config-if-Et15)#
```

- This command disables ECN marking of unicast packets from transmit queue 4 of Ethernet interface 15.

```
switch(config-if-Et15-uc-txq-4)#no random-detect ecn
switch(config-if-Et15-uc-txq-4)#show active
interface Ethernet15
switch(config-if-Et15-uc-txq-4)#exit
switch(config-if-Et15)#
```

23.7 QoS Configuration: Trident-II and Helix Platform Switches

Implementing QoS on a Trident platform switch consists of configuring port trust settings, default port settings, default traffic classes, conversion maps, and transmit queues.

- Section 23.7.1: CoS and DSCP Port Settings – Trident-II and Helix Platform Switches
- Section 23.7.2: Traffic Class Derivations – Trident-II and Helix Platform Switches
- Section 23.7.3: CoS and DSCP Rewrite – Trident-II and Helix Platform Switches
- Section 23.7.4: Transmit Queues and Port Shaping – Trident-II and Helix Platform Switches

23.7.1 CoS and DSCP Port Settings – Trident-II and Helix Platform Switches

Configuring Port Trust Settings

The **qos trust** command configures the QoS port trust mode for the configuration mode interface. Trust enabled ports use packet CoS or DSCP values to classify traffic. The port-trust default for switched ports is ***cos***. The port-trust default for routed ports is ***dscp***.

- **qos trust cos** specifies ***cos*** as the port's port-trust mode.
- **qos trust dscp** specifies ***dscp*** as the port's port-trust mode.
- **no qos trust** specifies ***untrusted*** as the port's port-trust mode.

The **show qos interfaces trust** command displays the trust mode of specified interfaces.

Example

- These commands configure and display the following trust modes:
  - Ethernet 7/1: dscp
  - Ethernet 7/2: untrusted
  - Ethernet 7/3: cos
  - Ethernet 7/4: default as a switched port
  - Ethernet 8/1: default as a routed port

```
switch(config)#interface ethernet 7/
switch(config-if-Et7/1)#qos trust dscp
switch(config-if-Et7/1)#interface ethernet 7/2
switch(config-if-Et7/2)#no qos trust
switch(config-if-Et7/2)#interface ethernet 7/3
switch(config-if-Et7/3)#qos trust cos
switch(config-if-Et7/3)#interface ethernet 7/4
switch(config-if-Et7/4)#switchport
switch(config-if-Et7/4)#default qos trust
switch(config-if-Et7/4)#interface ethernet 8/1
switch(config-if-Et8/1)#no switchport
switch(config-if-Et8/1)#default qos trust
switch(config-if-Et8/1)#show qos interface ethernet 7/1 - 8/1 trust
Port                                    Trust Mode
                              Operational           Configured
---------------------------------------------------------------
Ethernet7/1                   DSCP                  DSCP
Ethernet7/2                   UNTRUSTED             UNTRUSTED
Ethernet7/3                   COS                   COS
Ethernet7/4                   COS                   DEFAULT
Ethernet8/1                   DSCP                  DEFAULT
switch(config-if-Et8/1)#
```

Configuring Default Port Settings

Default CoS and DSCP settings are assigned to individual port channel and Ethernet interfaces. These configuration mode interface commands specify the port's default CoS and DSCP values.

- **qos cos** configures a port's default CoS value.
- **qos dscp** configures a port's default DSCP value.

Example

- These commands configure default CoS (4) and DSCP (44) values on Ethernet interface 7/3.

```
switch(config)#interface ethernet 7/3
switch(config-if-Et7/3)#qos cos 4
switch(config-if-Et7/3)#qos dscp 44
switch(config-if-Et7/3)#show active
interface Ethernet7/3
   qos cos 4
   qos dscp 44
switch(config-if-Et7/3)#show qos interfaces ethernet 7/3
Ethernet7/3:
   Trust Mode: COS
   Default COS: 4
   Default DSCP: 44
              <-------OUTPUT OMITTED FROM EXAMPLE-------->
switch(config-if-Et7/3)#
```

23.7.2 Traffic Class Derivations – Trident-II and Helix Platform Switches

Section 23.1.1.4 describes traffic classes.

Note Qos traffic policy is supported on Trident-II platform switches.

Traffic Class Derivation Source

Table 23-29 displays the source for deriving a data stream's traffic class.

Table 23-29 Traffic Class Derivation Source: Trident-II Platform Switches

	Untrusted	CoS Trusted	DSCP Trusted
Untagged Non-IP	Default CoS (port)	Default CoS (port)	Default DSCP (port)
Untagged IP	Default CoS (port)	Default CoS (port)	DSCP (packet)
Tagged Non-IP	Default CoS (port)	CoS (packet)	Default DSCP (port)
Tagged IP	Default CoS (port)	CoS (packet)	DSCP (packet)

Section 23.7.1 describes the default CoS and DSCP settings for each port.

Mapping CoS to Traffic Class

The **qos map cos** command assigns a traffic class to a list of CoS settings. Multiple commands create a complete CoS to traffic class map. The switch uses this map to assign a traffic class to data packets on the basis of the packet's CoS field or the port upon which it is received.

Example

- This command assigns the traffic class of 5 to the classes of service 1, 3, 5, and 7.

```
switch(config)#qos map cos 1 3 5 7 to traffic-class 5
switch(config)#show qos maps
   Number of Traffic Classes supported: 8
             <-------OUTPUT OMITTED FROM EXAMPLE-------->

   Cos-tc map:
     cos:  0  1  2  3  4  5  6  7
     ---------------------------
     tc:   1  5  2  5  4  5  6  5

             <-------OUTPUT OMITTED FROM EXAMPLE-------->
switch(config)#
```

Table 23-30 displays the default CoS–traffic class map on Trident-II platform switches.

Table 23-30 Default CoS to Traffic Class Map: Trident-II Platform Switches

Inbound CoS	0	1	2	3	4	5	6	7
Traffic Class	1	0	2	3	4	5	6	7

Mapping DSCP to Traffic Class

The **qos map dscp** command assigns a traffic class to a set of DSCP values. Multiple commands create a complete DSCP to traffic class map. The switch uses this map to assign a traffic class to data packets on the basis of the packet's DSCP field or the chip upon which it is received.

Example

- This command assigns the traffic class of 0 to DSCP values of 12, 24, 41, and 44-47.

```
switch(config)#qos map dscp 12 24 41 44 45 46 47 to traffic-class 0
switch(config)#show qos maps
   Number of Traffic Classes supported: 8
             <-------OUTPUT OMITTED FROM EXAMPLE-------->

   Dscp-tc map:
     d1 :  d2 0  1  2  3  4  5  6  7  8  9
     ----------------------------------------
      0 :     1  1  1  1  1  1  1  0  0
      1 :     0  0  0  0  0  2  2  2  2
      2 :     2  2  2  2  0  3  3  3  3  3
      3 :     3  3  4  4  4  4  4  4  4  4
      4 :     5  0  5  5  0  0  0  0  6  6
      5 :     6  6  6  6  6  6  7  7  7  7
      6 :     7  7  7  7

             <-------OUTPUT OMITTED FROM EXAMPLE-------->
switch(config)#
```

Table 23-31 displays the default DSCP–traffic class map on Trident-II platform switches.

Table 23-31 Default DSCP to Traffic Class Map: Trident-II Platform Switches

Inbound DSCP	0-7	8-15	16-23	24-31	32-39	40-47	48-55	56-63
Traffic Class	1	0	2	3	4	5	6	7

23.7.3 CoS and DSCP Rewrite – Trident-II and Helix Platform Switches

Section 23.1.1.3 describes the CoS and DSCP rewrite functions.

Traffic Class to CoS Rewrite Map

The CoS rewrite value is configurable and based on a data stream's traffic class, as specified by the traffic class-CoS rewrite map. The **qos map traffic-class to cos** command assigns a CoS rewrite value to a list of traffic classes. Multiple commands create the complete traffic class–CoS rewrite map.

Example

- This command assigns the CoS of two to traffic classes 1, 3, and 5.

```
switch(config)#qos map traffic-class 1 3 5 to cos 2
switch(config)#show qos map
   Number of Traffic Classes supported: 8
             <-------OUTPUT OMITTED FROM EXAMPLE-------->

   Tc-cos map:
     tc:   0  1  2  3  4  5  6  7
     ----------------------------
     cos:  1  2  2  2  4  2  6  7

             <-------OUTPUT OMITTED FROM EXAMPLE-------->
switch(config)#
```

Table 23-32 displays the default Traffic Class to CoS rewrite value map on Trident-II platform switches.

Table 23-32 Default Traffic Class to CoS Rewrite Value Map: Trident-II Platform Switches

Traffic Class	0	1	2	3	4	5	6	7
CoS Rewrite Value	1	0	2	3	4	5	6	7

Traffic Class to DSCP Rewrite Map

The DSCP rewrite value is configurable and based on a data stream's traffic class, as specified by the traffic class-DSCP rewrite map. The **qos map traffic-class to dscp** command assigns a DSCP rewrite value to a list of traffic classes. Multiple commands create the complete traffic class-DSCP rewrite map.

Example

- This command assigns the DSCP value of 29 to traffic classes 2, 4, and 6.

```
switch(config)#qos map traffic-class 2 4 6 to dscp 29
switch(config)#show qos map
   Number of Traffic Classes supported: 8
             <-------OUTPUT OMITTED FROM EXAMPLE-------->

   Tc-dscp map:
     tc:    0  1  2  3  4  5  6  7
     ------------------------------
     dscp:  8  0 29 24 29 40 29 56

             <-------OUTPUT OMITTED FROM EXAMPLE-------->
switch(config)#
```

Table 23-33 displays the default traffic class to DSCP rewrite map on Trident-II platform switches.

Table 23-33 Traffic Class to DSCP Rewrite Value Map: Trident-II Platform Switches

Traffic Class	0	1	2	3	4	5	6	7
DSCP	8	0	16	24	32	40	48	56

23.7.4 Transmit Queues and Port Shaping – Trident-II and Helix Platform Switches

Section 23.1.2 describes transmit queues and port shaping.

A data stream's traffic class determines the transmit queue it uses. The switch defines a single traffic class-transmit queue map for all Ethernet interfaces and is used for unicast and multicast traffic. The traffic class to transmit queue maps are configured globally and apply to all Ethernet and port channel interfaces. The **show qos maps** command displays the traffic class to transmit queue map.

Trident-II platform switches have eight unicast (UC0 – UC7) and eight multicast (MC0 – MC7) queues. Each UCx-MCx queue set is combined into a single queue group (L1.x), which is exposed to the CLI through this command.

Table 23-34 displays the default traffic class to transmit queue maps.

Table 23-34 Default Traffic Class to Transmit Queue Map: Trident-II Platform Switches

Traffic Class	0	1	2	3	4	5	6	7
Transmit Queue Group	0	1	2	3	4	5	6	7

Mapping Traffic Classes to a Transmit Queue

The **qos map traffic-class to tx-queue** command assigns traffic classes to a transmit queue. Multiple commands create the complete map.

Example

- These commands assign traffic classes of 1, 3, and 5 to transmit queue 1, traffic classes 2, 4, and 6 to transmit queue 2, and traffic class 0 to transmit queue 0, then display the resultant map.

```
switch(config)#qos map traffic-class 1 3 5 to tx-queue 1
switch(config)#qos map traffic-class 2 4 6 to tx-queue 2
switch(config)#qos map traffic-class 0 to tx-queue 0
switch(config)#show qos maps
   Number of Traffic Classes supported: 8
   Number of Transmit Queues supported: 8
             <-------OUTPUT OMITTED FROM EXAMPLE-------->

   Tc - tx-queue map:
     tc:        0  1  2  3  4  5  6  7
     --------------------------------
     tx-queue:  0  1  2  1  2  1  2  7

switch(config)#
```

Entering a Transmit Queue Configuration Mode

Transmit queues are configurable on Ethernet ports and port channels. Queue parameters are configured in tx-queue configuration command mode, which is entered from the appropriate interface configuration mode. The **tx-queue (Trident-II)** command places the switch in tx-queue configuration mode. The **show qos interfaces** displays the transmit queue configuration for a specified port.

Example

- This command enters tx-queue configuration mode for transmit queue 3 of Ethernet interface 5.

```
switch(config)#interface ethernet 5
switch(config-if-Et5)#tx-queue 3
switch(config-if-Et5-txq-3)#
```

Configuring the Shape Rate – Port and Transmit Queues

A port's shape rate specifies the port's maximum outbound traffic bandwidth. A shape rate can also be configured for all transmit queues on each port. All shape rate commands use kbps to specify data rates.

- To configure a port's shape rate, enter **shape rate (Interface – Trident-II)** from the port's interface configuration mode.
- To configure a transmit queue's shape rate, enter **shape rate (Tx-queue – Trident-II)** from the queue's tx-queue configuration mode.

Example

- These commands configure a shape rate of 5 Gbs on Ethernet port 3, then configure the shape rate for the following transmit queues:
  - transmit queues 0, 1, and 2: 500 Mbps
  - transmit queues 3, 4, and 5: 400 Mbps

```
switch(config)#interface ethernet 17/3
switch(config-if-Et17/3)#shape rate 5000000
switch(config-if-Et17/3)#tx-queue 0
switch(config-if-Et17/3-txq-0)#shape rate 500000
switch(config-if-Et17/3-txq-0)#tx-queue 1
switch(config-if-Et17/3-txq-1)#shape rate 500000
switch(config-if-Et17/3-txq-1)#tx-queue 3
switch(config-if-Et17/3-txq-3)#shape rate 400000
switch(config-if-Et17/3-txq-3)#tx-queue 4
switch(config-if-Et17/3-txq-4)#shape rate 400000
switch(config-if-Et17/3-txq-4)#tx-queue 5
switch(config-if-Et17/3-txq-5)#shape rate 400000
switch(config-if-Et17/3-txq-5)#exit
switch(config-if-Et17/3)#show qos interface ethernet 17/3
Ethernet17/3:
             <-------OUTPUT OMITTED FROM EXAMPLE-------->
  Tx       Bandwidth                 Shape Rate        Priority
 Queue     Guaranteed (units)         (units)
   ------------------------------------------------------------
   7        - / -    (  -  )       - / -    (  -  )    SP / SP
   6        - / -    (  -  )       - / -    (  -  )    SP / SP
   5        - / -    (  -  )     400 / 400  ( Mbps )   SP / SP
   4        - / -    (  -  )     400 / 400  ( Mbps )   SP / SP
   3        - / -    (  -  )     400 / 400  ( Mbps )   SP / SP
   2        - / -    (  -  )       - / -    (  -  )    SP / SP
   1        - / -    (  -  )     500 / 500  ( Mbps )   SP / SP
   0        - / -    (  -  )     500 / 500  ( Mbps )   SP / SP
             <-------OUTPUT OMITTED FROM EXAMPLE-------->
switch(config-if-Et17/3)#
```

Configuring Queue Priority

Queue priority rank is denoted by the queue number; transmit queues with higher numbers have higher priority. Trident-II supports strict priority queues; round robin queues are not supported.

The **bandwidth guaranteed (Trident-II)** command configures specifies the minimum bandwidth for outbound traffic on the transmit queue.

Example

- These commands configure a minimum egress bandwidth of 1 Mbps for transmit queue 4 of Ethernet interface 17/3.

```
switch(config-if-Et17/3)#tx-queue 4
switch(config-if-Et17/3-txq-4)#show qos interface ethernet 17/3

  Tx       Bandwidth                 Shape Rate        Priority
 Queue     Guaranteed (units)         (units)
   ------------------------------------------------------------
   7        - / -    (  -  )       - / -    (  -  )    SP / SP
   6        - / -    (  -  )       - / -    (  -  )    SP / SP
   5        - / -    (  -  )     400 / 400  ( Mbps )   SP / SP
   4        1 / 1    ( Mbps )    400 / 400  ( Mbps )   SP / SP
   3        - / -    (  -  )     400 / 400  ( Mbps )   SP / SP
   2        - / -    (  -  )       - / -    (  -  )    SP / SP
   1        - / -    (  -  )     500 / 500  ( Mbps )   SP / SP
   0        - / -    (  -  )     500 / 500  ( Mbps )   SP / SP
             <-------OUTPUT OMITTED FROM EXAMPLE-------->
switch(config-if-Et17/3-txq-4)#
```

23.7.5 Ingress Policing on LAG

Ingress policing on a port-channel polices the matched traffic from all member interfaces combined, i.e. it provides aggregate policing and statistics (DCS-7050X, DCS-7010T, DCS-7250X, and DCS-7300X series). When a per-interface policer is attached to a port-channel, one set of TCAM entries is created for all member interfaces. The associated interface bitmap is updated, and aggregate policing is performed on all member interfaces.

Example

- These commands configure a service-policy (with policer action) on LAG by creating the service-policy and applying the service-policy on a port-channel.

```
switch(config)#policy-map policy-1
switch(config-pmap-qos-policy-1)#class class-1
switch(config-pmap-c-qos-policy-1-class-1)#police cir 512000 bps bc 96000
switch(config-pmap-c-qos-policy-1-class-1)#exit
switch(config-pmap-qos-policy-1)#exit
switch(config)#interface Et 4 / 5 / 4
switch(config-if-Et4/5/4)#channel-group 2 mode active
switch(config-if-Et4/5/4)#exit
switch(config)#interface po2
switch(config-if-Po2)#service-policy type qos input policy-1
switch(config-if-Po2)#exit
switch(config)#
```

These commands configure ACL policing in single-rate, two-color mode.

```
switch(config)#class-map type qos match-any class1
switch(config-cmap-qos-class1)#match ip access-group acl1
switch(config-cmap-qos-class1)#exit
switch(config)#policy-map type qos policy1
switch(config-pmap-qos-policy1)#class class1
switch(config-pmap-c-qos-policy1-class1)#police cir 512000 bc 96000
switch(config-pmap-c-qos-policy1-class1)#exit
switch(config-pmap-qos-policy1)#exit
switch(config)#show policy-map
Service-policy policy1

  Class-map: class1 (match-any)
    Match: ip access-group name acl1
       police rate 512000 bps burst-size 96000 bytes

  Class-map: class-default (match-any)

switch(config)#
```

23.8 Quality of Service Configuration Commands

QoS Data Field and Traffic Class Configuration Commands

- qos cos
- qos dscp
- qos trust
- qos map cos
- qos map dscp
- qos map traffic-class to cos
- qos map traffic-class to dscp
- qos map traffic-class to tx-queue
- qos map traffic-class to uc-tx-queue
- qos map traffic-class to mc-tx-queue
- platform petraA traffic-class
- qos rewrite cos
- qos rewrite dscp

QoS Data Field and Traffic Class Display Commands

- show qos interfaces
- show qos maps
- show qos interfaces trust
- show platform petraA traffic-class

ECN Configuration Commands

- qos random-detect ecn global-buffer (Helix)
- qos random-detect ecn global-buffer (Trident)
- random-detect ecn (Arad/Jericho)
- random-detect ecn (Helix)
- random-detect ecn (Trident)
- show qos interfaces random-detect ecn
- show qos random-detect ecn

Transmit Queue and Port Shaping Commands – Arad and Jericho Platforms

- tx-queue (Arad/Jericho)
- bandwidth percent (Arad/Jericho)
- priority (Arad/Jericho)
- shape rate (Tx-queue – Arad/Jericho)
- shape rate (Interface – Arad/Jericho)

Transmit Queue and Port Shaping Commands – FM6000 Platform

- tx-queue (FM6000)
- bandwidth percent (FM6000)
- priority (FM6000)
- shape rate (Tx-queue – FM6000)
- shape rate (Interface – FM6000)

Transmit Queue and Port Shaping Commands – Helix Platform

- tx-queue (Helix)
- bandwidth guaranteed (Helix)
- shape rate (Tx-queue – Helix)
- shape rate (Interface – Helix)

Transmit Queue and Port Shaping Commands – Petra Platform

- tx-queue (Petra)
- bandwidth percent (Petra)
- priority (Petra)
- shape rate (Tx-queue – Petra)
- shape rate (Interface – Petra)

Transmit Queue and Port Shaping Commands – Trident Platform

- uc-tx-queue
- mc-tx-queue
- bandwidth percent (Trident)
- priority (Trident)
- shape rate (Tx-queues – Trident)
- shape rate (Interface – Trident)

Transmit Queue and Port Shaping Commands – Trident-II Platform

- tx-queue (Trident-II)
- bandwidth guaranteed (Trident-II)
- shape rate (Tx-queue – Trident-II)
- shape rate (Interface – Trident-II)

bandwidth guaranteed (Helix)

The **bandwidth guaranteed** command specifies the minimum bandwidth for outbound traffic on the transmit queue. By default, no bandwidth is guaranteed to any transmit queue.

The **no bandwidth guaranteed** and **default bandwidth guaranteed** commands remove the minimum bandwidth guarantee on the transmit queue by deleting the corresponding **bandwidth guaranteed** command from ***running-config***.

Command Mode

Tx-Queue Configuration

Command Syntax

```
bandwidth guaranteed rate DATA_MIN
no bandwidth guaranteed
default bandwidth guaranteed
```

Parameters

- ***DATA_MIN*** minimum bandwidth. Value range varies with data unit:
  - <***8*** to ***40000000***> 8 to 40,000,000 kbytes per second.
  - <***8*** to ***40000000***>**kbps** 8 to 40,000,000 kbytes per second.
  - <***8*** to ***60000000***>**pps** 1 to 60,000,000 packets per second.

Related Commands

- **tx-queue (Helix)** places the switch in tx-queue configuration mode.

Example

- These commands configure a minimum egress bandwidth of 1 Mbps for transmit queue 4 of Ethernet interface 17/3.

```
switch(config)#interface ethernet 17
switch(config-if-Et17)#tx-queue 4
switch(config-if-Et17-txq-4)#bandwidth guaranteed 1000 kbps
switch(config-if-Et17-txq-4)#show qos interfaces ethernet 17
Ethernet17/3:
   Trust Mode: COS
   Default COS: 0
   Default DSCP: 0

   Port shaping rate: disabled

  Tx       Bandwidth                Shape Rate        Priority
 Queue     Guaranteed (units)        (units)
   ---------------------------------------------------------------
   7        - / -    (  -  )       - / -    (  -  )    SP / SP
   6        - / -    (  -  )       - / -    (  -  )    SP / SP
   5        - / -    (  -  )       - / -    (  -  )    SP / SP
   4        1 / 1    ( Mbps )      - / -    (  -  )    SP / SP
   3        - / -    (  -  )       - / -    (  -  )    SP / SP
   2        - / -    (  -  )       - / -    (  -  )    SP / SP
   1        - / -    (  -  )       - / -    (  -  )    SP / SP
   0        - / -    (  -  )       - / -    (  -  )    SP / SP

Note: Values are displayed as Operational/Configured
             <-------OUTPUT OMITTED FROM EXAMPLE-------->
switch(config-if-Et17-txq-4)#
```

bandwidth guaranteed (Trident-II)

The **bandwidth guaranteed** command specifies the minimum bandwidth for outbound traffic on the transmit queue. By default, no bandwidth is guaranteed to any transmit queue.

The **no bandwidth guaranteed** and **default bandwidth guaranteed** commands remove the minimum bandwidth guarantee on the transmit queue by deleting the corresponding **bandwidth guaranteed** command from ***running-config***.

Command Mode

Tx-Queue Configuration

Command Syntax

```
bandwidth guaranteed rate DATA_MIN
no bandwidth guaranteed
default bandwidth guaranteed
```

Parameters

- ***DATA_MIN*** minimum bandwidth. Value range varies with data unit:
  - <***8*** to ***40000000***> 8 to 40,000,000 kbytes per second.
  - <***8*** to ***40000000***>**kbps** 8 to 40,000,000 kbytes per second.
  - <***8*** to ***60000000***>**pps** 1 to 60,000,000 packets per second.

Related Commands

- **tx-queue (Trident-II)** places the switch in tx-queue configuration mode.

Example

- These commands configure a minimum egress bandwidth of 1 Mbps for transmit queue 4 of Ethernet interface 17/3.

```
switch(config)#interface ethernet 17/3
switch(config-if-Et17/3)#tx-queue 4
switch(config-if-Et17/3-txq-4)#bandwidth guaranteed 1000 kbps
switch(config-if-Et17/3-txq-4)#show qos interfaces ethernet 17/3
Ethernet17/3:
   Trust Mode: COS
   Default COS: 0
   Default DSCP: 0

   Port shaping rate: disabled

  Tx       Bandwidth                Shape Rate        Priority
 Queue     Guaranteed (units)        (units)
   ---------------------------------------------------------------
   7        - / -    (  -  )       - / -    (  -  )    SP / SP
   6        - / -    (  -  )       - / -    (  -  )    SP / SP
   5        - / -    (  -  )       - / -    (  -  )    SP / SP
   4        1 / 1    ( Mbps )      - / -    (  -  )    SP / SP
   3        - / -    (  -  )       - / -    (  -  )    SP / SP
   2        - / -    (  -  )       - / -    (  -  )    SP / SP
   1        - / -    (  -  )       - / -    (  -  )    SP / SP
   0        - / -    (  -  )       - / -    (  -  )    SP / SP

Note: Values are displayed as Operational/Configured
             <-------OUTPUT OMITTED FROM EXAMPLE-------->
switch(config-if-Et17/3-txq-4)#
```

bandwidth percent (Arad/Jericho)

The **bandwidth percent** command configures the bandwidth share of the transmit queue when configured for round robin priority. Bandwidth is allocated to all queues based on the cumulative configured bandwidth of all the port's round robin queues.

The cumulative operational bandwidth of all round robin queues is always less than or equal to 100%. If the cumulative configured bandwidth is greater than 100%, each port's operational bandwidth is its configured bandwidth divided by the cumulative configured bandwidth.

The **no bandwidth percent** and **default bandwidth percent** commands restore the default bandwidth share of the transmit queue by removing the corresponding **bandwidth percent** command from ***running-config***.

Command Mode

Tx-Queue Configuration

Command Syntax

```
bandwidth percent proportion
no bandwidth percent
default bandwidth percent
```

Parameters

- *proportion* Bandwidth percentage assigned to queues. Values range from 1 to 100.

Related Commands

- **tx-queue (Arad/Jericho)** places the switch in tx-queue configuration mode.

Example

- These commands configure queues 0 through 3 (Ethernet interface 3/5/1) as round robin, then allocate bandwidth for three queues at 30% and one queue at 10%.

```
switch(config)#interface ethernet 3/5/1
switch(config-if-Et3/5/1)#tx-queue 3
switch(config-if-Et3/5/1-txq-3)#no priority
switch(config-if-Et3/5/1-txq-3)#bandwidth percent 10
switch(config-if-Et3/5/1-txq-3)#tx-queue 2
switch(config-if-Et3/5/1-txq-2)#bandwidth percent 30
switch(config-if-Et3/5/1-txq-2)#tx-queue 1
switch(config-if-Et3/5/1-txq-1)#bandwidth percent 30
switch(config-if-Et3/5/1-txq-1)#tx-queue 0
switch(config-if-Et3/5/1-txq-0)#bandwidth percent 30
switch(config-if-Et3/5/1-txq-0)#show qos interfaces ethernet 3/5/1
Ethernet3/5/1:
             <-------OUTPUT OMITTED FROM EXAMPLE-------->
  Tx    Bandwidth       Shape Rate        Priority  ECN
 Queue  (percent)        (units)
   ------------------------------------------------------
   7      - / -        - / -    (  -  )    SP / SP    D
   6      - / -        - / -    (  -  )    SP / SP    D
   5      - / -        - / -    (  -  )    SP / SP    D
   4      - / -        - / -    (  -  )    SP / SP    D
   3     10 / 10       - / -    (  -  )    RR / RR    D
   2     30 / 30       - / -    (  -  )    RR / SP    D
   1     30 / 30       - / -    (  -  )    RR / SP    D
   0     30 / 30       - / -    (  -  )    RR / SP    D
             <-------OUTPUT OMITTED FROM EXAMPLE-------->
switch(config-if-Et3/5/1-txq-0)#
```

- These commands re-configure the bandwidth share of the fourth queue at 30%.

```
switch(config-if-Et3/5/1-txq-0)#tx-queue 3
switch(config-if-Et3/5/1-txq-3)#bandwidth percent 30
switch(config-if-Et3/5/1-txq-3)#show qos interfaces ethernet 3/5/1
Ethernet3/5/1:
             <-------OUTPUT OMITTED FROM EXAMPLE-------->
   Port shaping rate: disabled

  Tx    Bandwidth       Shape Rate        Priority  ECN
 Queue  (percent)        (units)
   -------------------------------------------------------
   7      - / -        - / -    (  -  )    SP / SP    D
   6      - / -        - / -    (  -  )    SP / SP    D
   5      - / -        - / -    (  -  )    SP / SP    D
   4      - / -        - / -    (  -  )    SP / SP    D
   3     24 / 30       - / -    (  -  )    RR / RR    D
   2     24 / 30       - / -    (  -  )    RR / SP    D
   1     24 / 30       - / -    (  -  )    RR / SP    D
   0     24 / 30       - / -    (  -  )    RR / SP    D

Note: Values are displayed as Operational/Configured
             <-------OUTPUT OMITTED FROM EXAMPLE-------->
switch(config-if-Et3/5/1-txq-3)#
```

- These commands configure the bandwidth share of the fourth queue at 2%.

```
switch(config-if-Et3/5/1-txq-3)#bandwidth percent 2
switch(config-if-Et3/5/1-txq-3)#show qos interfaces ethernet 3/5/1
Ethernet3/5/1:
             <-------OUTPUT OMITTED FROM EXAMPLE-------->
   Port shaping rate: disabled

  Tx    Bandwidth       Shape Rate        Priority  ECN
 Queue  (percent)        (units)
   -------------------------------------------------------
   7      - / -        - / -    (  -  )    SP / SP    D
   6      - / -        - / -    (  -  )    SP / SP    D
   5      - / -        - / -    (  -  )    SP / SP    D
   4      - / -        - / -    (  -  )    SP / SP    D
   3      2 / 2        - / -    (  -  )    RR / RR    D
   2     30 / 30       - / -    (  -  )    RR / SP    D
   1     30 / 30       - / -    (  -  )    RR / SP    D
   0     30 / 30       - / -    (  -  )    RR / SP    D

Note: Values are displayed as Operational/Configured
             <-------OUTPUT OMITTED FROM EXAMPLE-------->
switch(config-if-Et3/5/1-txq-3)#
```

bandwidth percent (FM6000)

The **bandwidth percent** command configures the bandwidth share of the transmit queue when configured for round robin priority. Bandwidth is allocated to all queues based on the cumulative configured bandwidth of all the port's round robin queues.

The cumulative operational bandwidth of all round robin queues is always less than or equal to 100%. If the cumulative configured bandwidth is greater than 100%, each port's operational bandwidth is its configured bandwidth divided by the cumulative configured bandwidth.

The **no bandwidth percent** and **default bandwidth percent** commands restore the default bandwidth share of the transmit queue by removing the corresponding **bandwidth percent** command ***running-config***.

Command Mode

Tx-Queue Configuration

Command Syntax

```
bandwidth percent proportion
no bandwidth percent
default bandwidth percent
```

Parameters

- *proportion* Configured bandwidth percentage. Value ranges from 1 to 100. Default value is 0.

Related Commands

- **tx-queue (FM6000)** places the switch in tx-queue configuration mode.

Example

- These commands configure queues 0 through 3 (Ethernet interface 19) as round robin, then allocate bandwidth for three queues at 30% and one queue at 10%.

```
switch(config)#interface ethernet 19
switch(config-if-Et19)#tx-queue 3
switch(config-if-Et19-txq-3)#no priority
switch(config-if-Et19-txq-3)#bandwidth percent 10
switch(config-if-Et19-txq-3)#tx-queue 2
switch(config-if-Et19-txq-2)#bandwidth percent 30
switch(config-if-Et19-txq-2)#tx-queue 1
switch(config-if-Et19-txq-1)#bandwidth percent 30
switch(config-if-Et19-txq-1)#tx-queue 0
switch(config-if-Et19-txq-0)#bandwidth percent 30
switch(config-if-Et19-txq-0)#show qos interface ethernet 19
Ethernet19:
   Trust Mode: COS
             <-------OUTPUT OMITTED FROM EXAMPLE-------->
   Tx-Queue   Bandwidth    Shape Rate     Priority
              (percent)       (Kbps)
   ------------------------------------------------
          6         N/A     disabled        strict
          5         N/A     disabled        strict
          4         N/A     disabled        strict
          3          10     disabled   round-robin
          2          30     disabled   round-robin
          1          30     disabled   round-robin
          0          30     disabled   round-robin

switch(config-if-Et19-txq-0)#
```

- These commands re-configure the bandwidth share of transmit queue 3 at 30%.

```
switch(config-if-Et19-txq-0)#tx-queue 3
switch(config-if-Et19-txq-3)#bandwidth percent 30
switch(config-if-Et19-txq-3)#show qos interface ethernet 19
Ethernet19:
   Trust Mode: COS
              <-------OUTPUT OMITTED FROM EXAMPLE-------->
   Tx-Queue   Bandwidth    Shape Rate     Priority
              (percent)       (Kbps)
   ----------------------------------------------
          6         N/A     disabled        strict
          5         N/A     disabled        strict
          4         N/A     disabled        strict
          3          24     disabled   round-robin
          2          24     disabled   round-robin
          1          24     disabled   round-robin
          0          24     disabled   round-robin

switch(config-if-Et19-txq-3)#
```

- These commands re-configure the bandwidth share of transmit queue 3 at 2%.

```
switch(config-if-Et19-txq-3)#bandwidth percent 2
switch(config-if-Et19-txq-3)#show qos interface ethernet 19
Ethernet19:
   Trust Mode: COS
              <-------OUTPUT OMITTED FROM EXAMPLE-------->
   Tx-Queue   Bandwidth    Shape Rate     Priority
              (percent)       (Kbps)
   ----------------------------------------------
          6         N/A     disabled        strict
          5         N/A     disabled        strict
          4         N/A     disabled        strict
          3           2     disabled   round-robin
          2          30     disabled   round-robin
          1          30     disabled   round-robin
          0          30     disabled   round-robin

switch(config-if-Et19-txq-3)#
```

bandwidth percent (Petra)

The **bandwidth percent** command configures the bandwidth share of the transmit queue when configured for round robin priority. Bandwidth is allocated to all queues based on the cumulative configured bandwidth of all the port's round robin queues.

The cumulative operational bandwidth of all round robin queues is always less than or equal to 100%. If the cumulative configured bandwidth is greater than 100%, each port's operational bandwidth is its configured bandwidth divided by the cumulative configured bandwidth.

The **no bandwidth percent** and **default bandwidth percent** commands restore the default bandwidth share of the transmit queue by removing the corresponding **bandwidth percent** command ***running-config***.

Command Mode

Tx-Queue Configuration

Command Syntax

```
bandwidth percent proportion
no bandwidth percent
default bandwidth percent
```

Parameters

- *proportion* Bandwidth percentage assigned to queues. Values range from 1 to 100.

Related Commands

- **tx-queue (Petra)** places the switch in tx-queue configuration mode.

Example

- These commands configure queues 0 through 3 (Ethernet interface 3/28) as round robin, then allocate bandwidth for three queues at 30% and one queue at 10%.

```
switch(config)#interface ethernet 3/28
switch(config-if-Et3/28)#tx-queue 3
switch(config-if-Et3/28-txq-3)#no priority
switch(config-if-Et3/28-txq-3)#bandwidth percent 10
switch(config-if-Et3/28-txq-3)#tx-queue 2
switch(config-if-Et3/28-txq-2)#bandwidth percent 30
switch(config-if-Et3/28-txq-2)#tx-queue 1
switch(config-if-Et3/28-txq-1)#bandwidth percent 30
switch(config-if-Et3/28-txq-1)#tx-queue 0
switch(config-if-Et3/28-txq-0)#bandwidth percent 30
switch(config-if-Et3/28-txq-0)#show qos interface ethernet 3/28
Ethernet3/28:
             <-------OUTPUT OMITTED FROM EXAMPLE-------->
   Tx-Queue   Bandwidth    Shape Rate     Priority
              (percent)       (Kbps)
   ------------------------------------------------
          7         N/A     disabled        strict
          6         N/A     disabled        strict
          5         N/A     disabled        strict
          4         N/A     disabled        strict
          3          10     disabled   round-robin
          2          30     disabled   round-robin
          1          30     disabled   round-robin
          0          30     disabled   round-robin

switch(config-if-Et3/28-txq-0)#
```

- These commands re-configure the bandwidth share of the fourth queue at 30%.

```
switch(config-if-Et3/28-txq-0)#tx-queue 3
switch(config-if-Et3/28-txq-3)#bandwidth percent 30
switch(config-if-Et3/28-txq-3)#show qos interface ethernet 3/28
Ethernet3/28:
   Trust Mode: COS
              <-------OUTPUT OMITTED FROM EXAMPLE-------->
   Tx-Queue   Bandwidth    Shape Rate     Priority
              (percent)       (Kbps)
------------------------------------------------
          7         N/A     disabled        strict
          6         N/A     disabled        strict
          5         N/A     disabled        strict
          4         N/A     disabled        strict
          3          24     disabled   round-robin
          2          24     disabled   round-robin
          1          24     disabled   round-robin
          0          24     disabled   round-robin

switch(config-if-Et3/28-txq-3)#
```

- These commands configure the bandwidth share of the fourth queue at 2%.

```
switch(config-if-Et3/28)#tx-queue 3
switch(config-if-Et3/28-txq-3)#bandwidth percent 2
switch(config-if-Et3/28-txq-3)#show qos interface ethernet 3/28
Ethernet3/28:
   Trust Mode: COS
              <-------OUTPUT OMITTED FROM EXAMPLE-------->

   Tx-Queue   Bandwidth    Shape Rate     Priority
              (percent)       (Kbps)
   ------------------------------------------------
          7         N/A     disabled        strict
          6         N/A     disabled        strict
          5         N/A     disabled        strict
          4         N/A     disabled        strict
          3           2     disabled   round-robin
          2          30     disabled   round-robin
          1          30     disabled   round-robin
          0          30     disabled   round-robin

switch(config-if-Et3/28-txq-3)#
```

bandwidth percent (Trident)

The **bandwidth percent** command configures the bandwidth share of the transmit queue when configured for round robin priority. Bandwidth is allocated to all queues based on the cumulative configured bandwidth of all the port's round robin queues.

The cumulative operational bandwidth of all round robin queues is always less than or equal to 100%. If the cumulative configured bandwidth is greater than 100%, each port's operational bandwidth is its configured bandwidth divided by the cumulative configured bandwidth.

The **no bandwidth percent** and **default bandwidth percent** commands restore the default bandwidth share of the transmit queue by removing the corresponding **bandwidth percent** command ***running-config***.

Command Mode

Mc-Tx-Queue configuration
Uc-Tx-Queue configuration

Command Syntax

```
bandwidth percent proportion
no bandwidth percent
default bandwidth percent
```

Parameters

- *proportion* Bandwidth percentage assigned to queues. Values range from 1 to 100.

Related Commands

- **mc-tx-queue** places the switch in mc-tx-queue configuration mode.
- **uc-tx-queue** places the switch in uc-tx-queue configuration mode.

Example

- These commands configure unicast transmit queue 3 (and all other queues of lower priority) as round robin, then allocate bandwidth for unicast transmit queues 1, 2, and 3 at 30% and multicast transmit queue 1 at 10%.

```
switch(config)#interface ethernet 7
switch(config-if-Et7)#uc-tx-queue 3
switch(config-if-Et7-uc-txq-3)#no priority
switch(config-if-Et7-uc-txq-3)#bandwidth percent 30
switch(config-if-Et7-uc-txq-3)#uc-tx-queue 2
switch(config-if-Et7-uc-txq-2)#bandwidth percent 30
switch(config-if-Et7-uc-txq-2)#uc-tx-queue 1
switch(config-if-Et7-uc-txq-1)#bandwidth percent 30
switch(config-if-Et7-uc-txq-1)#mc-tx-queue 1
switch(config-if-Et7-mc-txq-1)#bandwidth percent 10
switch(config-if-Et7-mc-txq-1)#show qos interfaces ethernet 7
Ethernet7:
   Trust Mode: COS
   Default COS: 0
   Default DSCP: 0

   Port shaping rate: disabled

   Tx-Queue   Bandwidth    Shape Rate     Priority   Priority Group
              (percent)       (Kbps)
   ------------------------------------------------------------------
        UC7         N/A     disabled        strict                1
        UC6         N/A     disabled        strict                1
        MC3         N/A     disabled        strict                1
        UC5         N/A     disabled        strict                0
        UC4         N/A     disabled        strict                0
        MC2         N/A     disabled        strict                0
        UC3          30     disabled   round-robin                0
        UC2          30     disabled   round-robin                0
        MC1          10     disabled   round-robin                0
        UC1          30     disabled   round-robin                0
        UC0           0     disabled   round-robin                0
        MC0           0     disabled   round-robin                0

switch(config-if-Et7-mc-txq-1)#
```

- These commands re-configure the bandwidth share of unicast queue 3 at 55%.

```
switch(config-if-Et7-mc-txq-1)#uc-tx-queue 3
switch(config-if-Et7-uc-txq-3)#bandwidth percent 55
switch(config-if-Et7-uc-txq-3)#show qos interface ethernet 7
Ethernet7:
   Trust Mode: COS
   Default COS: 0
   Default DSCP: 0

   Port shaping rate: disabled

   Tx-Queue   Bandwidth    Shape Rate     Priority   Priority Group
              (percent)       (Kbps)
   ------------------------------------------------------------------
        UC7         N/A     disabled        strict                1
        UC6         N/A     disabled        strict                1
        MC3         N/A     disabled        strict                1
        UC5         N/A     disabled        strict                0
        UC4         N/A     disabled        strict                0
        MC2         N/A     disabled        strict                0
        UC3          44     disabled   round-robin                0
        UC2          24     disabled   round-robin                0
        MC1           8     disabled   round-robin                0
        UC1          24     disabled   round-robin                0
        UC0           0     disabled   round-robin                0
        MC0           0     disabled   round-robin                0

switch(config-if-Et7-uc-txq-3)#
```

mc-tx-queue

The **mc-tx-queue** command places the switch in mc-tx-queue configuration mode to configure a multicast transmit queue on the configuration mode interface. Mc-tx-queue configuration mode is not a group change mode; ***running-config*** is changed immediately after commands are executed. The **exit** command does not affect the configuration.

Trident switches have four multicast queues (MC0 – MC03) and eight unicast queues (UC0 – UC7), categorized into two priority groups. All queues are exposed through the CLI and are user configurable.

- Priority Group 1: UC7, UC6, MC3
- Priority Group 0: UC5, UC4, MC2, UC3, UC2, MC1, UC1, UC0, MC0

The **exit** command returns the switch to the configuration mode for the base Ethernet or port channel interface.

The **no mc-tx-queue** and **default mc-tx-queue** commands remove the configuration for the specified transmit queue by deleting the all corresponding **mc-tx-queue** mode commands from ***running-config***.

Command Mode

Interface-Ethernet Configuration
Interface-Port-Channel Configuration

Command Syntax

```
mc-tx-queue queue_level
```

Parameters

- *queue_level* The multicast transmit queue number. Values range from ***0*** to ***3***.

Commands Available in tx-queue Configuration Mode

- **bandwidth percent (Trident)**
- **priority (Trident)**
- **shape rate (Tx-queues – Trident)**

Related Commands

- **uc-tx-queue**: Configures unicast transmit queues on Trident platform switches.

Example

- This command enters mc-tx-queue configuration mode for multicast transmit queue 3 of Ethernet interface 5.

```
switch(config)#interface ethernet 5
switch(config-if-Et5)#mc-tx-queue 3
switch(config-if-Et5-mc-txq-3)#
```

platform petraA traffic-class

The **platform petraA traffic-class** command configures the default traffic class used by all ports on a specified chip. The default traffic class is implemented by Petra platform switches to replace **qos cos** and **qos dscp** commands. Traffic class values range from 0 to 7. The default traffic class is one.

When **platform ?** returns ***Petra***:

- CoS trusted ports: inbound untagged packets are assigned to the default traffic class. Tagged packets are assigned to the traffic class that corresponds to the contents of its CoS field.
- DSCP trusted ports: inbound non-IP packets are assigned to the default traffic class. IP packets are assigned to the traffic class that corresponds to the contents of its DSCP field.
- Untrusted ports: all inbound packets are assigned to the default traffic class.

The **no platform petraA traffic-class** and **default platform petraA traffic-class** commands restore the default traffic class of one for all ports on the specified chips by deleting the corresponding **platform petraA traffic-class** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
platform petraA [CHIP_NAME] traffic-class tc_value
no platform petraA [CHIP_NAME] traffic-class
default platform petraA [CHIP_NAME] traffic-class
```

Parameters

- ***CHIP_NAME*** trust mode assigned to the specified ports. Port designation options include:
  - <no parameter> all ports on the switch.
  - **module** *cardX* all ports on specified linecard (7500 Series).
  - **petra***cardX*/*chipY* all ports on PetraA chip *chipY* on linecard *cardX* (7500 Series).
  - **petra**-*chipZ* all ports on PetraA chip chipZ (7048 Series)

  7500 Series

  Switches can contain up to eight linecards. *cardX* varies from 3 to 10.

  Each linecard contains six PetraA chips. Each chip controls eight ports. *chipY* varies from 0 to 5:

  - 0 controls ports 1 through 8
  - 1 controls ports 9 through 16
  - 2 controls ports 17 through 24
  - 3 controls ports 25 through 32
  - 4 controls ports 33 through 40
  - 5 controls ports 41 through 48

  7048 Series

  Each switch contains two PetraA chips. *chipZ* varies from 0 to 1:

  - 0 controls ports 1 through 32
  - 1 controls ports 33 through 52
- *tc_value* Traffic class value. Values range from 0 to 7. Default value is 1.

Related Commands

- **show platform petraA traffic-class** displays the traffic class assignment on all specified Petra chips.

Example

- This command configures the default traffic class to six for ports 25-32 on linecard 5.

```
switch(config)#platform petraA petra5/3 traffic-class 6
switch(config)#
```

priority (Arad/Jericho)

The **priority** command specifies the priority of the transmit queue. The switch supports two queue priorities:

- strict priority: contents are removed from the queue - subject to maximum bandwidth limits, before data from lower priority queues. The default setting on all queues is strict priority.
- round robin priority: contents are removed proportionately from all round robin queues - subject to maximum bandwidth limits assigned to the strict priority queues.

Tx-queue 7 is set to strict priority and is not configurable.

When a queue is configured as a round robin queue, all lower priority queues also function as round robin queues. A queue's numerical label denotes its priority: higher labels denote higher priority. Tx-queue 6 has higher priority than Tx-queue 5, and Tx-queue 0 has the lowest priority.

The **priority strict** and **default priority** commands configure a transmit queue to function as a strict priority queue unless a higher priority queue is configured as a round robin queue.

The **no priority** command configures a transmit queue as a round robin queue. All lower priority queues also function as round robin queues regardless of their configuration.

Command Mode

Tx-Queue Configuration

Command Syntax

```
priority strict
no priority
default priority
```

Related Commands

- **tx-queue (Arad/Jericho)** places the switch in tx-queue configuration mode.

Example

- These commands perform the following on Ethernet interface 3/4/1:
  - Displays the default state of all transmit queues.
  - Configures transmit queue 3 as a round robin queue.

- Displays the effect of the **no priority** command on all transmit queues on the interface.

```
switch(config)#interface ethernet 3/4/1
switch(config-if-Et3/4/1)#show qos interfaces ethernet 3/4/1
Ethernet3/4/1:
             <-------OUTPUT OMITTED FROM EXAMPLE-------->
  Tx    Bandwidth       Shape Rate        Priority  ECN
 Queue  (percent)        (units)
   -----------------------------------------------------
   7      - / -        - / -    (  -  )    SP / SP    D
   6      - / -        - / -    (  -  )    SP / SP    D
   5      - / -        - / -    (  -  )    SP / SP    D
   4      - / -      999 / 1000 ( Mbps )   SP / SP    D
   3      - / -      999 / 1000 ( Mbps )   SP / SP    D
   2      - / -        - / -    (  -  )    SP / SP    D
   1      - / -        - / -    (  -  )    SP / SP    D
   0      - / -        - / -    (  -  )    SP / SP    D

Note: Values are displayed as Operational/Configured
             <-------OUTPUT OMITTED FROM EXAMPLE-------->
switch(config-if-Et3/4/1)#tx-queue 3
switch(config-if-Et3/4/1-txq-3)#no priority
switch(config-if-Et3/4/1-txq-3)#show qos interfaces ethernet 3/4/1
Ethernet3/4/1:
             <-------OUTPUT OMITTED FROM EXAMPLE-------->
  Tx    Bandwidth       Shape Rate        Priority  ECN
 Queue  (percent)        (units)
   -----------------------------------------------------
   7      - / -        - / -    (  -  )    SP / SP    D
   6      - / -        - / -    (  -  )    SP / SP    D
   5      - / -        - / -    (  -  )    SP / SP    D
   4      - / -      999 / 1000 ( Mbps )   SP / SP    D
   3     25 / -      999 / 1000 ( Mbps )   RR / RR    D
   2     25 / -        - / -    (  -  )    RR / SP    D
   1     25 / -        - / -    (  -  )    RR / SP    D
   0     25 / -        - / -    (  -  )    RR / SP    D

Note: Values are displayed as Operational/Configured
             <-------OUTPUT OMITTED FROM EXAMPLE-------->
switch(config-if-Et3/4/1-txq-3)#
```

priority (FM6000)

The **priority** command specifies the priority of the transmit queue. The switch supports two queue priorities:

- strict priority: contents are removed from the queue - subject to maximum bandwidth limits, before data from lower priority queues. The default setting on all queues is strict priority.
- round robin priority: contents are removed proportionately from all round robin queues - subject to maximum bandwidth limits assigned to the strict priority queues.

When a queue is configured as a round robin queue, all lower priority queues also function as round robin queues. A queue's numerical label denotes its priority: higher labels denote higher priority. Tx-queue 6 has higher priority than Tx-queue 5, and Tx-queue 0 has the lowest priority.

The **priority strict** and **default priority** commands configure a transmit queue to function as a strict priority queue unless a higher priority queue is configured as a round robin queue.

The **no priority** command configures a transmit queue as a round robin queue. All lower priority queues also function as round robin queues regardless of their configuration.

Command Mode

Tx-Queue Configuration

Command Syntax

```
priority strict
no priority
default priority
```

Related Commands

- **tx-queue (FM6000)** places the switch in tx-queue configuration mode.

Example

- These commands perform the following on Ethernet interface 2:
  - Displays the default state of all transmit queues.
  - Configures transmit queue 3 as a round robin queue.
  - Displays the effect of the **no priority** command on all transmit queues on the interface.

```
switch(config)#interface ethernet 19
switch(config-if-Et19)#show qos interface ethernet 19
Ethernet19:
   Trust Mode: COS
              <-------OUTPUT OMITTED FROM EXAMPLE-------->
   Tx-Queue   Bandwidth    Shape Rate     Priority
              (percent)       (Kbps)
   ---------------------------------------------
          6         N/A     disabled        strict
          5         N/A     disabled        strict
          4         N/A     disabled        strict
          3         N/A     disabled        strict
          2         N/A     disabled        strict
          1         N/A     disabled        strict
          0         N/A     disabled        strict

switch(config-if-Et19)#tx-queue 3
switch(config-if-Et19-txq-3)#no priority
switch(config-if-Et19-txq-3)#show qos interface ethernet 19
Ethernet19:
   Trust Mode: COS
              <-------OUTPUT OMITTED FROM EXAMPLE-------->
   Tx-Queue   Bandwidth    Shape Rate     Priority
              (percent)       (Kbps)
   ---------------------------------------------
          6         N/A     disabled        strict
          5         N/A     disabled        strict
          4         N/A     disabled        strict
          3          25     disabled   round-robin
          2          25     disabled   round-robin
          1          25     disabled   round-robin
          0          25     disabled   round-robin

switch(config-if-Et19-txq-3)#
```

priority (Petra)

The **priority** command specifies the priority of the transmit queue. The switch supports two queue priorities:

- strict priority: contents are removed from the queue - subject to maximum bandwidth limits, before data from lower priority queues. The default setting on all queues is strict priority.
- round robin priority: contents are removed proportionately from all round robin queues - subject to maximum bandwidth limits assigned to the strict priority queues.

Tx-queue 7 is set to strict priority and is not configurable.

When a queue is configured as a round robin queue, all lower priority queues also function as round robin queues. A queue's numerical label denotes its priority: higher labels denote higher priority. Tx-queue 6 has higher priority than Tx-queue 5, and Tx-queue 0 has the lowest priority.

The **priority strict** and **default priority** commands configure a transmit queue to function as a strict priority queue unless a higher priority queue is configured as a round robin queue.

The **no priority** command configures a transmit queue as a round robin queue. All lower priority queues also function as round robin queues regardless of their configuration.

Command Mode

Tx-Queue Configuration

Command Syntax

```
priority strict
no priority
default priority
```

Related Commands

- **tx-queue (Petra)** places the switch in tx-queue configuration mode.

Example

- These commands perform the following on Ethernet interface 3/28:
  - Displays the default state of all transmit queues.
  - Configures transmit queue 3 as a round robin queue.

- Displays the effect of the **no priority** command on all transmit queues on the interface.

```
switch(config)#interface ethernet 3/28
switch(config-if-Et3/28)#show qos interface ethernet 3/28
Ethernet3/28:
   Trust Mode: COS
              <-------OUTPUT OMITTED FROM EXAMPLE-------->
   Tx-Queue   Bandwidth    Shape Rate     Priority
              (percent)       (Kbps)
   ------------------------------------------------
          7         N/A     disabled        strict
          6         N/A     disabled        strict
          5         N/A     disabled        strict
          4         N/A     disabled        strict
          3         N/A     disabled        strict
          2         N/A     disabled        strict
          1         N/A     disabled        strict
          0         N/A     disabled        strict

switch(config-if-Et3/28)#tx-queue 3
switch(config-if-Et3/28-txq-3)#no priority
switch(config-if-Et3/28-txq-3)#show qos interface ethernet 3/28
Ethernet3/28:
   Trust Mode: COS
              <-------OUTPUT OMITTED FROM EXAMPLE-------->
   Tx-Queue   Bandwidth    Shape Rate     Priority
              (percent)       (Kbps)
   ------------------------------------------------
          7         N/A     disabled        strict
          6         N/A     disabled        strict
          5         N/A     disabled        strict
          4         N/A     disabled        strict
          3          25     disabled   round-robin
          2          25     disabled   round-robin
          1          25     disabled   round-robin
          0          25     disabled   round-robin

switch(config-if-Et3/28-txq-3)#
```

priority (Trident)

The **priority** command specifies the priority of the transmit queue. The switch supports two queue priorities:

- strict priority: contents are removed from the queue - subject to maximum bandwidth limits, before data from lower priority queues. The default setting on all other queues is strict priority.
- round robin priority: contents are removed proportionately from all round robin queues - subject to maximum bandwidth limits assigned to the strict priority queues.

Trident switches have eight unicast queues (UC0 – UC7) and four multicast queues (MC0 – MC03), categorized into two priority groups. Priority group 1 queues have priority over priority 0 queues. The following lists display the priority group queues in order from higher priority to lower priority.

- Priority Group 1: UC7, UC6, MC3
- Priority Group 0: UC5, UC4, MC2, UC3, UC2, MC1, UC1, UC0, MC0

Priority group 1 queues are strict priority queues and are not configurable as round robin. Priority 0 queues are strict priority by default and are configurable as round robin. When a queue is configured as a round robin queue, all lower priority queues automatically function as round robin queues.

The **priority strict** and **default priority** commands configure a transmit queue to function as a strict priority queue unless a higher priority queue is configured as a round robin queue.

The **no priority** command configures a transmit queue as a round robin queue. All lower priority queues also function as round robin queues regardless of their configuration.

Command Mode

Mc-Tx-Queue configuration
Uc-Tx-Queue configuration

Command Syntax

```
priority strict
no priority
default priority
```

Related Commands

- **mc-tx-queue** places the switch in mc-tx-queue configuration mode.
- **uc-tx-queue**: places the switch in uc-tx-queue configuration mode.

Example

- These commands perform the following on Ethernet interface 7:
  - Displays the default state of all transmit queues.
  - Configures transmit queue 3 as a round robin queue.
  - Displays the effect of the **no priority** command on all transmit queues on the interface.

```
switch(config)#interface ethernet 7
switch(config-if-Et7)#show qos interface ethernet 7
Ethernet7:
   Trust Mode: COS
             <-------OUTPUT OMITTED FROM EXAMPLE-------->
   Tx-Queue   Bandwidth    Shape Rate     Priority   Priority Group
              (percent)       (Kbps)
   ------------------------------------------------------------------
        UC7         N/A     disabled        strict                1
        UC6         N/A     disabled        strict                1
        MC3         N/A     disabled        strict                1
        UC5         N/A     disabled        strict                0
        UC4         N/A     disabled        strict                0
        MC2         N/A     disabled        strict                0
        UC3         N/A     disabled        strict                0
        UC2         N/A     disabled        strict                0
        MC1         N/A     disabled        strict                0
        UC1         N/A     disabled        strict                0
        UC0         N/A     disabled        strict                0
        MC0         N/A     disabled        strict                0

switch(config-if-Et7)#uc-tx-queue 3
switch(config-if-Et7-uc-txq-3)#no priority
switch(config-if-Et7-uc-txq-3)#show qos interface ethernet 7
Ethernet7:
   Trust Mode: COS
             <-------OUTPUT OMITTED FROM EXAMPLE-------->
   Tx-Queue   Bandwidth    Shape Rate     Priority   Priority Group
              (percent)       (Kbps)
   ------------------------------------------------------------------
        UC7         N/A     disabled        strict                1
        UC6         N/A     disabled        strict                1
        MC3         N/A     disabled        strict                1
        UC5         N/A     disabled        strict                0
        UC4         N/A     disabled        strict                0
        MC2         N/A     disabled        strict                0
        UC3          20     disabled   round-robin                0
        UC2          16     disabled   round-robin                0
        MC1          16     disabled   round-robin                0
        UC1          16     disabled   round-robin                0
        UC0          16     disabled   round-robin                0
        MC0          16     disabled   round-robin                0

switch(config-if-Et7-uc-txq-3)#
```

qos cos

The **qos cos** command specifies the default class of service (CoS) value of the configuration mode interface. CoS values range from 0 to 7. Default value is 0.

Arad, Jericho, fm6000, trident, and ***Trident-II*** platform switches:

- CoS trusted ports: the default CoS value determines the traffic class for inbound untagged packets. Tagged packets are assigned to the traffic class that corresponds to the contents of its CoS field.
- Untrusted ports: the default CoS value determines the traffic class for all inbound packets.

Petra platform switches:

- CoS trusted ports: inbound untagged packets are assigned to the default traffic class, as configured by **platform petraA traffic-class**. Tagged packets are assigned to the traffic class that corresponds to the contents of its CoS field.
- Untrusted ports: all inbound packets are assigned to the default traffic class.

The **no qos cos** and **default qos cos** commands restore the port's default CoS value to zero by deleting the corresponding **qos cos** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration
Interface-Port-Channel Configuration

Command Syntax

```
qos cos cos_value
no qos cos
default qos cos
```

Parameters

- *cos_value* CoS value assigned to port. Value ranges from 0 to 7. Default value is 0.

Example

- This command configures the default CoS of four on Ethernet interface 8.

```
switch(config-if-Et8)#qos cos 4
switch(config-if-Et8)#
```

qos dscp

The **qos dscp** command specifies the default Differentiated Services Code Point (DSCP) value of the configuration mode interface. The default DSCP determines the traffic class for non-IP packets that are inbound on DSCP trusted ports. DSCP trusted ports determine the traffic class for inbound packets as follows:

- ***Arad, Jericho, fm6000, trident***, and ***Trident-II*** platform switches:
  - non-IP packets: default DSCP value specified by **qos dscp** determines the traffic class.
  - IP packets: assigned to the traffic class corresponding to its DSCP field contents.
- ***Petra*** platform switches:
  - non-IP packets: assigned to default traffic class configured by **platform petraA traffic-class**.
  - IP packets: assigned to the traffic class corresponding to its DSCP field contents.

The **no qos dscp** and **default qos dscp** commands restore the port's default DSCP value to zero by deleting the corresponding **qos dscp** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration
Interface-Port-Channel Configuration

Command Syntax

```
qos dscp dscp_value
no qos dscp
default qos dscp
```

Parameters

- *dscp_value* DSCP value assigned to the port. Value ranges from 0 to 63. Default value is 0.

Example

- This command sets the default DSCP of 44 on Ethernet 7 interface.

```
switch(config)#interface ethernet 7
switch(config-if-Et7)#qos dscp 44
switch(config-if-Et7)
```

qos trust

The **qos trust** command configures the quality of service port trust mode for the configuration mode interface. Trust-enabled ports classify traffic by examining the traffic's CoS or DSCP value. Port trust mode default setting is ***cos*** for switched interfaces and ***dscp*** for routed interfaces.

The **default qos trust** command restores the default trust mode on the configuration mode interface by removing the corresponding **qos trust** or **no qos trust** statement from ***running-config***.

The **no qos trust** command performs the following:

- **no qos trust** places the port in ***untrusted*** mode.
- **no qos trust cos** removes the corresponding **qos trust cos** statement.
- **no qos trust dscp** removes the corresponding **qos trust dscp** statement.

Command Mode

Interface-Ethernet Configuration
Interface-Port-Channel Configuration

Command Syntax

```
qos trust MODE
no qos trust [MODE]
default qos trust
```

Parameters

- ***MODE*** trust mode assigned to the port. Options include:
  - **cos** enables cos trust mode.
  - **dscp** enables dscp trust mode.

  no qos trust enables untrusted mode on the port.

Examples

- This command configures trust mode of dscp for Ethernet interface 5.

```
switch(config)#interface Ethernet 7
switch(config-if-Et7)#qos trust dscp
switch(config-if-Et7)#show active
interface Ethernet7
   qos trust dscp
switch(config-if-Et7)#
```

- This command configures trust mode of untrusted for Port Channel interface 23.

```
switch(config)#interface port-channel 23
switch(config-if-Po23)#no qos trust
switch(config-if-Po23)#show active
interface Port-Channel23
   no qos trust
switch(config-if-Po23)#
```

qos map cos

The **qos map cos** command associates a traffic class to a list of class of service (CoS) settings. Multiple commands create a complete CoS to traffic class map. The switch uses this map to assign a traffic class to data packets on the basis of the packet's CoS field or the port upon which it is received.

The **no qos map cos** and **default qos map cos** commands restore the specified CoS values to their default traffic class setting by deleting the corresponding **qos map cos** statements from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
qos map cos cos_value_1 [cos_value_2 ... cos_value_n] to traffic-class tc_value
no qos map cos cos_value_1 [cos_value_2 ... cos_value_n]
default qos map cos cos_value_1 [cos_value_2 ... cos_value_n]
```

Parameters

- *cos_value_x* Class of service (CoS) value. Value ranges from 0 to 7.
- *tc_value* Traffic class value. Value range varies by platform.

  Default CoS to traffic class map varies by platform (Table 23-35).

Default Inbound CoS to Traffic Class Map

Table 23-35 displays the default CoS to traffic class map for each platform.

Table 23-35 Default CoS to Traffic Class Map

Inbound CoS	untagged	0	1	2	3	4	5	6	7
Traffic Class (Arad/Jericho)	Derived: use default CoS as inbound CoS	1	0	2	3	4	5	6	7
Traffic Class (FM6000)	Derived: use default CoS as inbound CoS	1	0	2	3	4	5	6	7
Traffic Class (Helix)	Derived: use default CoS as inbound CoS	1	0	2	3	4	5	6	7
Traffic Class (Petra)	Assigned default traffic class	1	0	2	3	4	5	6	7
Traffic Class (Trident)	Derived: use default CoS as inbound CoS	1	0	2	3	4	5	6	7
Traffic Class (Trident-II)	Derived: use default CoS as inbound CoS	1	0	2	3	4	5	6	7

Related Commands

- **qos cos** specifies the default CoS
- **platform petraA traffic-class** specifies the default traffic class

Example

- This command assigns the traffic class of 5 to the classes of service 1, 3, 5, and 7.

```
switch(config)#qos map cos 1 3 5 7 to traffic-class 5
switch(config)#
```

qos map dscp

The **qos map dscp** command associates a traffic class to a set of Differentiated Services Code Point (DSCP) values. Multiple commands create a complete DSCP to traffic class map. The switch uses this map to assign a traffic class to data packets on the basis of the packet's DSCP field or the chip upon which it is received.

The **no qos map dscp** and **default qos map dscp** commands restore the specified DSCP values to their default traffic class settings by deleting corresponding **qos map dscp** statements from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
qos map dscp dscpv_1 [dscpv_2 ... dscpv_n] to traffic-class tc_value
no qos map dscp dscpv_1 [dscpv_2 ... dscpv_n]
default qos map dscp dscpv_1 [dscpv_2 ... dscpv_n]
```

Parameters

- *dscpv_x* Differentiated services code point (DSCP) value. Value ranges from 0 to 63.
- *tc_value* Traffic class value. Value range varies by platform.

  Default map varies by platform (Table 23-36).

Default Inbound DSCP to Traffic Class Map

Table 23-36 displays the default DSCP to traffic class map for each platform.

Table 23-36 Default DSCP to Traffic Class Map

Inbound DSCP	0-7	8-15	16-23	24-31	32-39	40-47	48-55	56-63
Traffic Class (Arad/Jericho)	1	0	2	3	4	5	6	7
Traffic Class (FM6000)	1	0	2	3	4	5	6	7
Traffic Class (Helix)	1	0	2	3	4	5	6	7
Traffic Class (Petra)	1	0	2	3	4	5	6	7
Traffic Class (Trident)	1	0	2	3	4	5	6	7
Traffic Class (Trident-II)	1	0	2	3	4	5	6	7

Example

- This command assigns the traffic class of three to the DSCP values of 12, 13, 25, and 37.

```
switch(config)#qos map dscp 12 13 25 37 to traffic-class 3
switch(config)#
```

qos map traffic-class to cos

The **qos map traffic-class to cos** command associates a class of service (CoS) to a list of traffic classes. Multiple commands create a complete traffic class to CoS map. The switch uses this map in CoS rewrite operations to fill the CoS field in outbound packets. This map is applicable to DSCP trusted ports and untrusted ports. CoS rewrite is disabled on CoS trusted ports. The **show qos maps** command displays the CoS to traffic class map.

The **no qos traffic-class to cos** and **default qos traffic-class to cos** commands restore the specified traffic class values to their default CoS settings by removing the corresponding **qos map traffic-class to cos** command from ***running-config***.

Command Mode

```
Global Configuration
```

Command Syntax

```
qos map traffic-class tc_num_1 [tc_num_2 ... tc_num_n] to cos cos_value
no qos map traffic-class tc_num_1 [tc_num_2 ... tc_num_n] to cos
default qos map traffic-class tc_num_1 [tc_num_2 ... tc_num_n] to cos
```

Parameters

- *tc_num_x* Traffic class value. Value range varies by switch platform.
- *cos_value* Cass of service (CoS) value. Value ranges from 0 to 7.

Default Inbound Traffic Class to CoS Map

Table 23-37 displays the default traffic class to CoS map for each platform.

Table 23-37 Default Traffic Class to CoS Rewrite Value Map

Traffic Class	0	1	2	3	4	5	6	7
CoS Rewrite Value (Arad/Jericho)	1	0	2	3	4	5	6	7
CoS Rewrite Value (FM6000)	1	0	2	3	4	5	6	7
CoS Rewrite Value (Helix)	1	0	2	3	4	5	6	7
CoS Rewrite Value (Petra)	1	0	2	3	4	5	6	7
CoS Rewrite Value (Trident)	1	0	2	3	4	5	6	7
CoS Rewrite Value (Trident-II)	1	0	2	3	4	5	6	7

Example

- This command assigns the CoS of two to traffic classes 1, 3, and 5.

```
switch(config)#qos map traffic-class 1 3 5 to cos 2
switch(config)#
```

qos map traffic-class to dscp

The **qos map traffic-class to dscp** command associates a Differentiated Services Code Point (DSCP) value to a list of traffic classes. Multiple commands create a complete traffic class to DSCP map. The switch uses this map in DSCP rewrite operations to fill the DSCP field in outbound packets. This map is applicable to CoS trusted ports and untrusted ports but disabled by default on these ports. DSCP rewrite is disabled on DSCP trusted ports. The **show qos maps** command displays the traffic class to DSCP map.

The **no qos traffic-class to dscp** and **default qos traffic-class to dscp** commands restore the specified traffic class values to their default DSCP settings by removing the corresponding **qos map traffic-class to dscp** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
qos map traffic-class tc_num_1 [tc_num_2 ... tc_num_n] to dscp dscp_value
no qos map traffic-class tc_num_1 [tc_num_2 ... tc_num_n] to dscp
default qos map traffic-class tc_num_1 [tc_num_2 ... tc_num_n] to dscp
```

Parameters

- *tc_num_x* Traffic class value. Value range varies by switch platform.
- *dscp_value* Differentiated services code point (DSCP) value. Value ranges from 0 to 63.

Default Inbound Traffic Class to DSCP Map

Table 23-38 displays the default traffic class to DSCP map for each platform.

Table 23-38 Default Traffic Class to DSCP Rewrite Value Map

Traffic Class	0	1	2	3	4	5	6	7
DSCP Rewrite Value (FM6000)	8	0	16	24	32	40	48	56
DSCP Rewrite Value (Helix)	8	0	16	24	32	40	48	56
DSCP Rewrite Value (Trident)	8	0	16	24	32	40	48	56
DSCP Rewrite Value (Trident-II)	8	0	16	24	32	40	48	56

Example

- This command assigns the DSCP value of 17 to traffic classes 1, 2, and 4.

```
switch(config)#qos map traffic-class 1 2 4 to dscp 17
switch(config)#
```

qos map traffic-class to mc-tx-queue

The **qos map traffic-class to mc-tx-queue** command associates a multicast transmit queue to a list of traffic classes. Multiple commands create a complete traffic class to mc-tx-queue map. The switch uses this map to route outbound packets to transmit queues, which in turn schedules their transmission from the switch. The **show qos maps** command displays the traffic class to multicast transmit queue map.

The **no qos traffic-class to mc-tx-queue** and **default qos traffic-class to mc-tx-queue** commands restore the default traffic class to multicast transmit queue map for the specified traffic class values by removing the corresponding **qos map traffic-class to mc-tx-queue** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
qos map traffic-class tc_num_1 [tc_num_2 ... tc_num_n] to mc-tx-queue mtq_value
no qos map traffic-class tc_num_1 [tc_num_2 ... tc_num_n] to mc-tx-queue
default qos map traffic-class tc_num_1 [tc_num_2 ... tc_num_n] to mc-tx-queue
```

Parameters

- *tc_num_x* Traffic class value. Value ranges from 0 to 7.
- *mtq_value* Multicast transmit queue number. Value ranges from ***0*** to ***3***.

Default Inbound Traffic Class to Multicast Transmit Queue Map

Table 23-39 displays the default traffic class to multicast transmit queue map for Trident platform switches

Table 23-39 Default Traffic Class to Multicast Transmit Queue Map

Traffic Class	0	1	2	3	4	5	6	7
Multicast Transmit Queue (Trident)	0	0	1	1	2	2	3	3

Related Commands

- **qos map traffic-class to uc-tx-queue** (Trident) associates traffic classes to a multicast transmit queue.
- **qos map traffic-class to tx-queue** (all other platforms) associates traffic classes to a transmit queue.

Example

- This command maps traffic classes 0, 4, and 5 to mc-tx-queue 2.

```
switch(config)#qos map traffic-class 0 4 5 to mc-tx-queue 2
switch(config)#
```

qos map traffic-class to tx-queue

The **qos map traffic-class to tx-queue** command associates a transmit queue (tx-queue) to a list of traffic classes. Multiple commands create a complete traffic to tx-queue map. The switch uses this map to route outbound packets to transmit queues, which in turn schedules their transmission from the switch. The **show qos maps** command displays the transmit queue to traffic class map.

The **no qos traffic-class to tx-queue** and **default qos traffic-class to tx-queue** commands restore the specified traffic class values to their default transmit queue settings by removing the corresponding **qos map traffic-class to tx-queue** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
qos map traffic-class tc_num_1 [tc_num_2 ... tc_num_n] to tx-queue txq_value
no qos map traffic-class tc_num_1 [tc_num_2 ... tc_num_n] to tx-queue
default qos map traffic-class tc_num_1 [tc_num_2 ... tc_num_n] to tx-queue
```

Parameters

- *tc_num_x* Traffic class value. Value range varies by platform.
- *txq_value* Transmit queue value. Value range varies by platform.

Restrictions

FM6000: When priority flow control (PFC) is enabled, traffic classes are mapped to their corresponding transmit queues, regardless of existing **qos map traffic-class to tx-queue** statements.

Arad, Jericho, and Petra: Traffic class 7 always maps to transmit queue 7. This association is not editable.

Default Inbound Traffic Class to Transmit Queue Map

Table 23-40 displays the transmit queue to traffic class map.

Table 23-40 Default Traffic Class to Transmit Queue Map

Traffic Class	0	1	2	3	4	5	6	7
Transmit Queue (Arad/Jericho)	0	1	2	3	4	5	6	7
Transmit Queue (FM6000)	0	1	2	3	4	5	6	7
Transmit Queue (Helix)	0	1	2	3	4	5	6	7
Transmit Queue (Petra)	0	1	2	3	4	5	6	7
Transmit Queue (Trident-II)	0	1	2	3	4	5	6	7

Related Commands

- **qos map traffic-class to mc-tx-queue** (Trident) associates traffic classes to a unicast transmit queue.
- **qos map traffic-class to uc-tx-queue** (Trident) associates traffic classes to a multicast transmit queue.

Example

- This command maps traffic classes 0, 4, and 5 to tx-queue 4.

```
switch(config)#qos map traffic-class 0 4 5 to tx-queue 4
switch(config)#
```

qos map traffic-class to uc-tx-queue

The **qos map traffic-class to uc-tx-queue** command associates a unicast transmit queue to a list of traffic classes. Multiple commands create a complete traffic class to unicast transmit queue map. The switch uses this map to route outbound packets to transmit queues, which in turn schedules their transmission from the switch. The **show qos maps** command displays the traffic class to unicast transmit queue map.

The **no qos traffic-class to uc-tx-queue** and **default qos traffic-class to uc-tx-queue** commands restore the default traffic class to unicast transmit queue map for the specified traffic class values by removing the corresponding **qos map traffic-class to uc-tx-queue** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
qos map traffic-class tc_num_1 [tc_num_2 ... tc_num_n] to uc-tx-queue utq_value
no qos map traffic-class tc_num_1 [tc_num_2 ... tc_num_n] to uc-tx-queue
default qos map traffic-class tc_num_1 [tc_num_2 ... tc_num_n] to uc-tx-queue
```

Parameters

- *tc_num_x* Traffic class value. Value ranges from 0 to 7.
- *utq_value* Unicast transmit queue number. Value ranges from 0 to 7.

Default Inbound Traffic Class to Unicast Transmit Queue Map

Table 23-41 displays the default traffic class to Unicast transmit queue map for Trident platform switches.

Table 23-41 Default Traffic Class to Unicast Transmit Queue Map

Traffic Class	0	1	2	3	4	5	6	7
Unicast Transmit Queue (Trident)	0	1	2	3	4	5	6	7

Related Commands

- **qos map traffic-class to mc-tx-queue** (Trident) associates traffic classes to a unicast transmit queue.
- **qos map traffic-class to tx-queue** (all other platforms) associates traffic classes to a transmit queue.

Example

- This command maps traffic classes 0, 4, and 5 to unicast transmit queue 4.

```
switch(config)#qos map traffic-class 0 4 5 to uc-tx-queue 4
switch(config)#
```

qos random-detect ecn global-buffer (Helix)

The **qos random-detect ecn global-buffer** command enables ECN marking for globally shared packet memory and specifies minimum and maximum queue threshold sizes. Hosts can advertise their ECN capabilities in the ToS DiffServ field's two least significant bits:

- 00 Non ECN Capable transport.
- 10 ECN Capable transport.
- 01 ECN Capable transport.
- 11 Congestion encountered.

Congestion is determined by comparing average queue size with queue thresholds. Average queue size is calculated through a formula based on the previous average and current queue size. Packets are marked based on this average size and the specified thresholds:

- Average queue size below minimum threshold: Packets are queued normally.
- Average queue size above maximum threshold: Packets are marked ***congestion encountered***.
- Average queue size between minimum and maximum thresholds. Packets are queued or marked ***congestion encountered***. The proportion of marked packets varies linearly with average queue size:
  - 0% are marked when average queue size is less than or equal to minimum threshold.
  - 100% are marked when average queue size is greater than or equal to maximum threshold.

When transmitted packets are marked ***Non ECN Capable***, congestion packets are dropped, not marked.

The **no qos random-detect ecn global-buffer** and **default qos random-detect ecn global-buffer** commands disables ECN marking for the shared buffer by removing the **qos random-detect ecn global-buffer** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
qos random-detect ecn global-buffer minimum-threshold MIN maximum-threshold MAX
no qos random-detect ecn global-buffer
default qos random-detect ecn global-buffer
```

Guidelines

Packet memory is divided into 46080 208-byte cells, whose allocation is managed by the memory management unit (MMU). The MMU tracks the cells that each entity uses and determines the number of cells that can be allocated to an entity.

Related Commands

- **random-detect ecn (Helix)** enables ECN marking for a unicast transmit queue.

Parameters

MIN and ***MAX*** parameters must use the same data unit.

- ***MIN*** Minimum threshold. Options include:
  - <***1*** to ***19456***> **segments** 208-byte segments units
  - <***1*** to ***4***> **mbytes** Megabyte units
  - <***1*** to ***4046***> **kbytes** Kilobyte units

- <***1*** to ***4046848***> **bytes** Byte units

- ***MAX*** Maximum threshold. Options include:
  - <***1*** to ***46080***> **segments** 208-byte segments units
  - <***1*** to ***4***> **mbytes** Megabyte units
  - <***1*** to ***4046***> **kbytes** Kilobyte units
  - <***1*** to ***4046848***> **bytes** Byte units

Examples

- This command enables ECN marking of unicast packets from the global data pool and sets the minimum and maximum thresholds at 20 and 500 segments.

```
switch(config)#qos random-detect ecn global-buffer minimum-threshold 20 segments
maximum-threshold 500 segments
switch(config)#
```

- This command disables ECN marking of unicast packets from the global data pool

```
switch(config)#no qos random-detect ecn global-buffer
switch(config)#
```

qos random-detect ecn global-buffer (Trident)

The **qos random-detect ecn global-buffer** command enables ECN marking for globally shared packet memory and specifies minimum and maximum queue threshold sizes. Hosts can advertise their ECN capabilities in the ToS DiffServ field's two least significant bits:

- 00 Non ECN Capable transport.
- 10 ECN Capable transport.
- 01 ECN Capable transport.
- 11 Congestion encountered.

Congestion is determined by comparing average queue size with queue thresholds. Average queue size is calculated through a formula based on the previous average and current queue size. Packets are marked based on this average size and the specified thresholds:

- Average queue size below minimum threshold: Packets are queued normally.
- Average queue size above maximum threshold: Packets are marked ***congestion encountered***.
- Average queue size between minimum and maximum thresholds. Packets are queued or marked ***congestion encountered***. The proportion of marked packets varies linearly with average queue size:
  - 0% are marked when average queue size is less than or equal to minimum threshold.
  - 100% are marked when average queue size is greater than or equal to maximum threshold.

When transmitted packets are marked ***Non ECN Capable***, congestion packets are dropped, not marked.

The **no qos random-detect ecn global-buffer** and **default qos random-detect ecn global-buffer** commands disables ECN marking for the shared buffer by removing the **qos random-detect ecn global-buffer** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
qos random-detect ecn global-buffer minimum-threshold MIN maximum-threshold MAX
no qos random-detect ecn global-buffer
default qos random-detect ecn global-buffer
```

Guidelines

Packet memory is divided into 46080 208-byte cells, whose allocation is managed by the memory management unit (MMU). The MMU tracks the cells that each entity uses and determines the number of cells that can be allocated to an entity.

Related Commands

- **random-detect ecn (Trident)** enables ECN marking for a unicast transmit queue.

Parameters

MIN and ***MAX*** parameters must use the same data unit.

- ***MIN*** Minimum threshold. Options include:
  - <***1*** to ***46080***> **segments** 208-byte segments units
  - <***1*** to ***9***> **mbytes** Megabyte units
  - <***1*** to ***9584***> **kbytes** Kilobyte units

  - <*1* to *9584640*> **bytes** Byte units
- ***MAX*** Maximum threshold. Options include:
  - <*1* to *46080*> **segments** 208-byte segments units
  - <*1* to *9*> **mbytes** Megabyte units
  - <*1* to *9584*> **kbytes** Kilobyte units
  - <*1* to *9584640*> **bytes** Byte units

Examples

- This command enables ECN marking of unicast packets from the global data pool and sets the minimum and maximum thresholds at 20 and 500 segments.

```
switch(config)#qos random-detect ecn global-buffer minimum-threshold 20 segments
maximum-threshold 500 segments
switch(config)#
```

- This command disables ECN marking of unicast packets from the global data pool

```
switch(config)#no qos random-detect ecn global-buffer
switch(config)#
```

qos rewrite cos

The **qos rewrite cos** command enables the rewriting of the CoS field for outbound tagged packets that were received on DSCP trusted ports and untrusted ports. CoS rewrite is always disabled on CoS trusted ports. The CoS value that is written into the packet is based on the data stream's traffic class. CoS rewriting is active by default.

The **no qos rewrite cos** command disables CoS rewriting on the switch. The default **qos rewrite cos** command restores the default setting of enabling CoS rewriting by removing the **no qos rewrite cos** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
qos rewrite cos
no qos rewrite cos
default qos rewrite cos
```

Related Commands

- **qos map traffic-class to cos** configures the traffic class to CoS rewrite map.

Example

- This command enables CoS rewrite.

```
switch(config)#qos rewrite cos
switch(config)#
```

qos rewrite dscp

The **qos rewrite dscp** command enables the rewriting of the DSCP field for outbound tagged packets that were received on CoS trusted ports and untrusted ports. DSCP rewrite is always disabled on DSCP trusted ports. The DSCP value that is written into the packet is based on the data stream's traffic class. DSCP rewriting is disabled by default.

The **no qos rewrite dscp** and **default qos rewrite dscp** commands disable DSCP rewriting on the switch by removing the **no qos rewrite dscp** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
qos rewrite dscp
no qos rewrite dscp
default qos rewrite dscp
```

Related Commands

- **qos map traffic-class to dscp** configures the traffic class to DSCP rewrite map.

Example

- This command enables DSCP rewrite.

```
switch(config)#qos rewrite dscp
switch(config)#
```

random-detect ecn (Arad/Jericho)

The **random-detect ecn** command enables ECN marking for the configuration mode unicast transmit queue and specifies threshold queue sizes. Hosts can advertise their ECN capabilities in the ToS DiffServ field's two least significant bits:

- 00 Non ECN Capable transport.
- 10 ECN Capable transport.
- 01 ECN Capable transport.
- 11 Congestion encountered.

Congestion is determined by comparing average queue size with queue thresholds. Average queue size is calculated through a formula based on the previous average and current queue size. Packets are marked based on this average size and the specified thresholds:

- Average queue size below minimum threshold: Packets are queued normally.
- Average queue size above maximum threshold: Packets are marked ***congestion encountered***.
- Average queue size between minimum and maximum thresholds. Packets are queued or marked ***congestion encountered***. The proportion of marked packets varies linearly with average queue size:
  - 0% are marked when average queue size is less than or equal to minimum threshold.
  - 100% are marked when average queue size is greater than or equal to maximum threshold.

When transmitted packets are marked ***Non ECN Capable***, congestion packets are dropped, not marked.

The **no random-detect ecn** and **default qos random-detect ecn** commands disables ECN marking for the shared buffer by removing the **qos random-detect ecn** command from ***running-config***.

Command Mode

Tx-Queue configuration

Command Syntax

```
random-detect ecn minimum-threshold MIN maximum-threshold MAX
no random-detect ecn
default random-detect ecn
```

Parameters

MIN and ***MAX*** parameters must use the same data unit.

- ***MIN*** Minimum threshold. Options include:
  - <***1*** to ***256***> **mbytes** Megabyte units
  - <***1*** to ***256000***> **kbytes** Kilobyte units
  - <***1*** to ***256000000***> **bytes** Byte units
- ***MAX*** Maximum threshold. Options include:
  - <***1*** to ***256***> **mbytes** Megabyte units
  - <***1*** to ***256000***> **kbytes** Kilobyte units
  - <***1*** to ***256000000***> **bytes** Byte units

Related Commands

- **tx-queue (Arad/Jericho)** places the switch in tx-queue configuration mode.

Examples

- These commands enable ECN marking of unicast packets from unicast transmit queue 4 of Ethernet interface 3/5/1, setting thresholds at 128 kbytes and 1280 kbytes.

```
switch(config)#interface ethernet 3/5/1
switch(config-if-Et3/5/1)#tx-queue 4
switch(config-if-Et3/5/1-txq-4)#random-detect ecn minimum-threshold 128 kbytes
maximum-threshold 1280 kbyte
switch(config-if-Et3/5/1-txq-4)#show active
interface Ethernet3/5/1
   tx-queue 4
     random-detect ecn minimum-threshold 128 kbytes maximum-threshold 1280 kbytes
switch(config-if-Et3/5/1-txq-4)#
```

random-detect ecn (Helix)

The **random-detect ecn** command enables ECN marking for the configuration mode unicast transmit queue and specifies threshold queue sizes. Hosts can advertise their ECN capabilities in the ToS DiffServ field's two least significant bits:

- 00 Non ECN Capable transport.
- 10 ECN Capable transport.
- 01 ECN Capable transport.
- 11 Congestion encountered.

Congestion is determined by comparing average queue size with queue thresholds. Average queue size is calculated through a formula based on the previous average and current queue size. Packets are marked based on this average size and the specified thresholds:

- Average queue size below minimum threshold: Packets are queued normally.
- Average queue size above maximum threshold: Packets are marked ***congestion encountered***.
- Average queue size between minimum and maximum thresholds. Packets are queued or marked ***congestion encountered***. The proportion of marked packets varies linearly with average queue size:
  - 0% are marked when average queue size is less than or equal to minimum threshold.
  - 100% are marked when average queue size is greater than or equal to maximum threshold.

When transmitted packets are marked ***Non ECN Capable***, congestion packets are dropped, not marked.

Average queue length is tracked for transmit queues and the global pool independently. When either entity reaches its maximum threshold, all subsequent packets are marked.

The **no random-detect ecn** and **default random-detect ecn** commands disable ECN marking on the configuration mode queue, deleting the corresponding **random-detect ecn** command from ***running-config***.

Command Mode

Tx-Queue configuration

Command Syntax

```
random-detect ecn minimum-threshold MIN maximum-threshold MAX
no random-detect ecn
default random-detect ecn
```

Related Commands

- **tx-queue (Helix)** places the switch in tx-queue configuration mode.
- **qos random-detect ecn global-buffer (Helix)** enables ECN marking for globally shared packet memory.

Parameters

MIN and ***MAX*** parameters must use the same data unit.

- ***MIN*** Minimum threshold. Options include:
  - <***1*** to ***46080***> **segments** 208-byte segments units
  - <***1*** to ***9***> **mbytes** Megabyte units
  - <***1*** to ***9584***> **kbytes** Kilobyte units

  - <*1* to *9584640*> **bytes** Byte units
- ***MAX*** Maximum threshold. Options include:
  - <*1* to *46080*> **segments** 208-byte segments units
  - <*1* to *9*> **mbytes** Megabyte units
  - <*1* to *9584*> **kbytes** Kilobyte units
  - <*1* to *9584640*> **bytes** Byte units

Examples

- These commands enable ECN marking of unicast packets from transmit queue 4 of Ethernet interface 15, setting thresholds at 10 and 100 segments.

```
switch(config)#interface ethernet 15
switch(config-if-Et15)#uc-tx-queue 4
switch(config-if-Et15-txq-4)#random-detect ecn minimum-threshold 10 segments
maximum-threshold 100 segments
switch(config-if-Et15-txq-4)#show active
interface Ethernet15
   tx-queue 4
      random-detect ecn minimum-threshold 10 segments maximum-threshold 100
segments
switch(config-if-Et15-txq-4)#exit
switch(config-if-Et15)
```

- This command disables ECN marking of unicast packets from transmit queue 4 of Ethernet interface 15.

```
switch(config-if-Et15-txq-4)#no random-detect ecn
switch(config-if-Et15-txq-4)#show active
interface Ethernet15
switch(config-if-Et15-txq-4)#exit
switch(config-if-Et15)#
```

random-detect ecn (Trident)

The **random-detect ecn** command enables ECN marking for the configuration mode unicast transmit queue and specifies threshold queue sizes. Hosts can advertise their ECN capabilities in the ToS DiffServ field's two least significant bits:

- 00 Non ECN Capable transport.
- 10 ECN Capable transport.
- 01 ECN Capable transport.
- 11 Congestion encountered.

Congestion is determined by comparing average queue size with queue thresholds. Average queue size is calculated through a formula based on the previous average and current queue size. Packets are marked based on this average size and the specified thresholds:

- Average queue size below minimum threshold: Packets are queued normally.
- Average queue size above maximum threshold: Packets are marked ***congestion encountered***.
- Average queue size between minimum and maximum thresholds. Packets are queued or marked ***congestion encountered***. The proportion of marked packets varies linearly with average queue size:
  - 0% are marked when average queue size is less than or equal to minimum threshold.
  - 100% are marked when average queue size is greater than or equal to maximum threshold.

When transmitted packets are marked ***Non ECN Capable***, congestion packets are dropped, not marked.

Average queue length is tracked for transmit queues and the global pool independently. When either entity reaches its maximum threshold, all subsequent packets are marked.

The **no random-detect ecn** and **default random-detect ecn** commands disable ECN marking on the configuration mode queue, deleting the corresponding **random-detect ecn** command from ***running-config***.

Command Mode

Uc-Tx-Queue configuration

Command Syntax

```
random-detect ecn minimum-threshold MIN maximum-threshold MAX
no random-detect ecn
default random-detect ecn
```

Related Commands

- **uc-tx-queue** places the switch in uc-tx-queue configuration mode.
- **qos random-detect ecn global-buffer (Trident)** enables ECN marking for globally shared packet memory.

Parameters

MIN and ***MAX*** parameters must use the same data unit.

- ***MIN*** Minimum threshold. Options include:
  - <***1*** to ***46080***> **segments** 208-byte segments units
  - <***1*** to ***9***> **mbytes** Megabyte units
  - <***1*** to ***9584***> **kbytes** Kilobyte units

  - <*1* to *9584640*> **bytes** Byte units
- ***MAX*** Maximum threshold. Options include:
  - <*1* to *46080*> **segments** 208-byte segments units
  - <*1* to *9*> **mbytes** Megabyte units
  - <*1* to *9584*> **kbytes** Kilobyte units
  - <*1* to *9584640*> **bytes** Byte units

Examples

- These commands enable ECN marking of unicast packets from unicast transmit queue 4 of Ethernet interface 15, setting thresholds at 10 and 100 segments.

```
   switch(config)#interface ethernet 15
switch(config-if-Et15)#uc-tx-queue 4
switch(config-if-Et15-uc-txq-4)#random-detect ecn minimum-threshold 10 segments
maximum-threshold 100 segments
switch(config-if-Et15-uc-txq-4)#show active
interface Ethernet15
   uc-tx-queue 4
      random-detect ecn minimum-threshold 10 segments maximum-threshold 100
segments
switch(config-if-Et15-uc-txq-4)#exit
switch(config-if-Et15)#
```

- This command disables ECN marking of unicast packets from unicast transmit queue 4 of Ethernet interface 15.

```
switch(config-if-Et15-uc-txq-4)#no random-detect ecn
switch(config-if-Et15-uc-txq-4)#show active
interface Ethernet15
switch(config-if-Et15-uc-txq-4)#exit
switch(config-if-Et15)#
```

shape rate (Interface – Arad/Jericho)

The **shape rate** command specifies the maximum bandwidth for outbound traffic on the configuration mode interface, also known as queue shaping. The shape rate for individual transmit queues is configured by the **shape rate (Tx-queue – Arad/Jericho)** command. By default, outbound transmission rate is not bounded by a shape rate.

The **no shape rate** and **default shape rate** commands remove the shape rate bandwidth limit on the configuration mode interface by deleting the corresponding **shape rate** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration
Interface-Port-Channel Configuration

Command Syntax

```
shape rate byte_limit [kbps]
no shape rate
default shape rate
```

Parameters

- *byte_limit* shape rate applied to interface (Kbps). Value ranges from 162 to 100000000.

Example

- This command configures a port shape rate of 5 Gbps on Ethernet interface 3/5/1.

```
switch(config)#interface ethernet 3/5/1
switch(config-if-Et3/5/1)#shape rate 5000000
switch(config-if-Et3/5/1)#show qos interfaces ethernet 3/5/1
Ethernet3/5/1:
             <-------OUTPUT OMITTED FROM EXAMPLE-------->
   Port shaping rate: 5000012 / 5000000 kbps

  Tx    Bandwidth       Shape Rate        Priority  ECN
 Queue  (percent)        (units)
   ------------------------------------------------------
   7      - / -        - / -    (  -  )    SP / SP    D
   6      - / -        - / -    (  -  )    SP / SP    D
   5      - / -        - / -    (  -  )    SP / SP    D
   4      - / -        - / -    (  -  )    SP / SP    D
   3      - / -        - / -    (  -  )    SP / SP    D
   2      - / -        - / -    (  -  )    SP / SP    D
   1      - / -        - / -    (  -  )    SP / SP    D
   0      - / -        - / -    (  -  )    SP / SP    D

             <-------OUTPUT OMITTED FROM EXAMPLE-------->
switch(config-if-Et3/5/1)#
```

shape rate (Interface – FM6000)

The **shape rate** command specifies the maximum bandwidth for outbound traffic on the configuration mode interface, also known as queue shaping. The shape rate for individual transmit queues is configured by the **shape rate (Tx-queue – FM6000)** command. By default, outbound transmission rate is not bounded by a shape rate.

The **no shape rate** and **default shape rate** commands remove the shape rate bandwidth limit on the configuration mode interface by deleting the corresponding **shape rate** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration
Interface-Port-Channel Configuration

Command Syntax

```
shape rate byte_limit [kbps]
no shape rate
default shape rate
```

Parameters

- *byte_limit* shape rate applied to interface (Kbps). Value ranges from 7000 to 10000000.

Guidelines

Enabling port shaping on an FM6000 interface disables queue shaping internally. Disabling port shaping restores queue shaping as specified in ***running-config***.

Example

- This command configures a port shape rate of 5 Gbps on Ethernet interface 5.

```
switch(config)#interface ethernet 5
switch(config-if-Et5)#shape rate 5000000
switch(config-if-Et5)#
```

shape rate (Interface – Helix)

The **shape rate** command specifies the maximum bandwidth for outbound traffic on the configuration mode interface, also known as queue shaping. The shape rate for individual transmit queues is configured by the **shape rate (Tx-queue – Helix)** command. By default, outbound transmission rate is not bounded by a shape rate.

The **no shape rate** and **default shape rate** commands remove the shape rate bandwidth limit on the configuration mode interface by deleting the corresponding **shape rate** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration
Interface-Port-Channel Configuration

Command Syntax

```
shape rate DATA_LIMIT
no shape rate
default shape rate
```

Parameters

- ***DATA_LIMIT*** shape rate applied to interface. Value range varies with data unit:
  - <***8*** to ***40000000***> 8 to 40,000,000 kbytes per second.
  - <***8*** to ***40000000***>**kbps** 8 to 40,000,000 kbytes per second.
  - <***8*** to ***60000000***>**pps** 8 to 60,000,000 packets per second.

Guidelines

Shaping rates of at least 8 kbps are supported. At shaping rates smaller than 1 Mbps, granularity and rounding errors may skew the actual shaping rate by 20% from the specified rate.

Example

- This command configures a port shape rate of 5 Gbps on Ethernet interface 17.

```
switch(config)#interface ethernet 17
switch(config-if-Et17)#shape rate 5000000 kbps
switch(config-if-Et17)#show qos interface ethernet 17/3
Ethernet17:
   Trust Mode: COS
   Default COS: 0
   Default DSCP: 0

   Port shaping rate: 5000000 / 5000000 kbps

  Tx       Bandwidth                 Shape Rate        Priority
 Queue     Guaranteed (units)         (units)
   ------------------------------------------------------------
   7        - / -    (  -  )       - / -    (  -  )    SP / SP
   6        - / -    (  -  )       - / -    (  -  )    SP / SP
            <-------OUTPUT OMITTED FROM EXAMPLE-------->
switch(config-if-Et17)#
```

shape rate (Interface – Petra)

The **shape rate** command specifies the maximum bandwidth for outbound traffic on the configuration mode interface, also known as queue shaping. The shape rate for individual transmit queues is configured by the **shape rate (Tx-queue – Petra)** command. By default, outbound transmission rate is not bounded by a shape rate.

The **no shape rate** and **default shape rate** commands remove the shape rate bandwidth limit on the configuration mode interface by deleting the corresponding **shape rate** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration
Interface-Port-Channel Configuration

Command Syntax

```
shape rate data_limit [kbps]
no shape rate
default shape rate
```

Parameters

- *data_limit* shape rate applied to interface (Kbps). Value ranges from 100 to 10000000.

Guidelines

The following port shaping rates are supported:

- 1G ports: above 100 kbps.
- 10G ports: above 7900 kbps.

Commands that specify a smaller shape rate disable port shaping on the interface.

Example

- This command configures a port shape rate of 5 Gbps on Ethernet interface 3/3.

```
switch(config)#interface ethernet 3/3
switch(config-if-Et3/3)#shape rate 5000000
switch(config-if-Et3/3)#show active
interface Ethernet3/3
   shape rate 5000000
switch(config-if-Et3/3)#
```

shape rate (Interface – Trident)

The **shape rate** command specifies the maximum bandwidth for outbound traffic on the configuration mode interface, also known as queue shaping. The shape rate for individual transmit queues is configured by the **shape rate (Tx-queues – Trident)** command. By default, outbound transmission rate is not bounded by a shape rate.

The **no shape rate** and **default shape rate** commands remove the shape rate bandwidth limit on the configuration mode interface by deleting the corresponding **shape rate** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration
Interface-Port-Channel Configuration

Command Syntax

```
shape rate DATA_LIMIT
no shape rate
default shape rate
```

Parameters

- ***DATA_LIMIT*** shape rate applied to interface. Value range varies with data unit:
  - <***8*** to ***40000000***> 8 to 40,000,000 kbytes per second.
  - <***8*** to ***40000000***>**kbps** 8 to 40,000,000 kbytes per second.
  - <***8*** to ***60000000***>**pps** 8 to 60,000,000 packets per second.

Guidelines

Shaping rates of at least 8 kbps are supported. At shaping rates smaller than 1 Mbps, granularity and rounding errors may skew the actual shaping rate by 20% from the specified rate.

Example

- This command configures a port shape rate of 5 Gbps on Ethernet interface 5.

```
switch(config)#interface ethernet 5
switch(config-if-Et5)#shape rate 5000000
switch(config-if-Et5)#
```

shape rate (Interface – Trident-II)

The **shape rate** command specifies the maximum bandwidth for outbound traffic on the configuration mode interface, also known as queue shaping. The shape rate for individual transmit queues is configured by the **shape rate (Tx-queue – Trident-II)** command. By default, outbound transmission rate is not bounded by a shape rate.

The **no shape rate** and **default shape rate** commands remove the shape rate bandwidth limit on the configuration mode interface by deleting the corresponding **shape rate** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration
Interface-Port-Channel Configuration

Command Syntax

```
shape rate DATA_LIMIT
no shape rate
default shape rate
```

Parameters

- ***DATA_LIMIT*** shape rate applied to interface. Value range varies with data unit:
  - <***8*** to ***40000000***> 8 to 40,000,000 kbytes per second.
  - <***8*** to ***40000000***>**kbps** 8 to 40,000,000 kbytes per second.
  - <***8*** to ***60000000***>**pps** 8 to 60,000,000 packets per second.

Guidelines

Shaping rates of at least 8 kbps are supported. At shaping rates smaller than 1 Mbps, granularity and rounding errors may skew the actual shaping rate by 20% from the specified rate.

Example

- This command configures a port shape rate of 5 Gbps on Ethernet interface 17/3.

```
switch(config)#interface ethernet 17/3
switch(config-if-Et17/3)#shape rate 5000000 kbps
switch(config-if-Et17/3)#show qos interface ethernet 17/3
Ethernet17/3:
   Trust Mode: COS
   Default COS: 0
   Default DSCP: 0

   Port shaping rate: 5000000 / 5000000 kbps

  Tx       Bandwidth                 Shape Rate        Priority
 Queue     Guaranteed (units)         (units)
   ------------------------------------------------------------
   7        - / -    (  -  )       - / -    (  -  )    SP / SP
   6        - / -    (  -  )       - / -    (  -  )    SP / SP
             <-------OUTPUT OMITTED FROM EXAMPLE-------->
switch(config-if-Et17/3)#
```

shape rate (Tx-queue – Arad/Jericho)

The **shape rate** command specifies the maximum bandwidth for outbound traffic on the transmit queue, also known as queue shaping. The shape rate for interfaces is configured by the **shape rate (Interface – Arad/Jericho)** command. By default, the configured outbound transmission rate is not bounded by a transmit queue shape rate.

Shaping rates greater than 50000 kbps are supported. At lower shaping rates (less than 10 Mbps), granularity and rounding errors may skew the actual shaping rate by 20% from the specified rate.

The **no shape rate** and **default shape rate** commands remove the shape rate bandwidth limit on the configuration mode queue by deleting the corresponding **shape rate** command from ***running-config***.

Command Mode

Tx-Queue Configuration

Command Syntax

```
shape rate byte_limit [kbps]
no shape rate
default shape rate
```

Parameters

- *byte_limit* shape rate applied to interface (Kbps). Value ranges from 50000 to 100000000.

Related Commands

- **tx-queue (Arad/Jericho)** places the switch in tx-queue configuration mode.

Example

- These commands configure a shape rate of 1 Gbps on transmit queues 3 and 4 of Ethernet interface 3/4/1.

```
switch(config)#interface ethernet 3/4/1
switch(config-if-Et3/4/1)#tx-queue 4
switch(config-if-Et3/4/1-txq-4)#shape rate 1000000 kbps
switch(config-if-Et3/4/1-txq-4)#tx-queue 3
switch(config-if-Et3/4/1-txq-3)#shape rate 1000000 kbps
switch(config-if-Et3/4/1-txq-3)#show qos interface ethernet 3/4/1
Ethernet3/4/1:
             <-------OUTPUT OMITTED FROM EXAMPLE-------->
   Port shaping rate: disabled

  Tx    Bandwidth       Shape Rate        Priority  ECN
 Queue  (percent)        (units)
   ------------------------------------------------------
   7      - / -        - / -    (  -  )    SP / SP    D
   6      - / -        - / -    (  -  )    SP / SP    D
   5      - / -        - / -    (  -  )    SP / SP    D
   4      - / -      999 / 1000 ( Mbps )   SP / SP    D
   3      - / -      999 / 1000 ( Mbps )   SP / SP    D
   2      - / -        - / -    (  -  )    SP / SP    D
   1      - / -        - / -    (  -  )    SP / SP    D
   0      - / -        - / -    (  -  )    SP / SP    D
             <-------OUTPUT OMITTED FROM EXAMPLE-------->
switch(config-if-Et3/4/1-txq-3)#
```

shape rate (Tx-queue – FM6000)

The **shape rate** command specifies the maximum bandwidth for outbound traffic on the transmit queue, also known as queue shaping. The shape rate for interfaces is configured by the **shape rate (Interface – FM6000)** command. By default, the configured outbound transmission rate is not bounded by a transmit queue shape rate.

Queue shaping on an FM6000 port is supported only when port shaping is not enabled on the interface. Enabling port shaping on a port disables queue shaping internally. Disabling port shaping restores queue shaping as specified by ***running-config***.

Shaping rates greater than 460 kbps are supported. At lower shaping rates (less than 10 Mbps), granularity and rounding errors may skew the actual shaping rate by 20% from the specified rate.

The **no shape rate** and **default shape rate** commands remove the shape rate bandwidth limit on the transmit queue by deleting the corresponding **shape rate** command from ***running-config***.

Command Mode

Tx-Queue Configuration

Command Syntax

```
shape rate byte_limit [kbps]
no shape rate
default shape rate
```

Parameters

- *byte_limit* shape rate applied to interface (Kbps). Value ranges from 464 to 10000000.

Related Commands

- **tx-queue (FM6000)** places the switch in tx-queue configuration mode
- **shape rate (Interface – FM6000)** configures the shape rate for a configuration mode interface.

Example

- These commands configure a shape rate of 1 Gbps (1,000,000 Kbps) on transmit queues 3 and 4 of Ethernet interface 19.

```
switch(config)#interface ethernet 19
switch(config-if-Et19)#tx-queue 4
switch(config-if-Et19-txq-4)#shape rate 1000000
switch(config-if-Et19-txq-4)#tx-queue 3
switch(config-if-Et19-txq-3)#shape rate 1000000
switch(config-if-Et19-txq-3)#show qos interface ethernet 19
Ethernet19:
   Trust Mode: COS
             <-------OUTPUT OMITTED FROM EXAMPLE-------->
   Tx-Queue   Bandwidth    Shape Rate     Priority
              (percent)       (Kbps)
   ------------------------------------------------
          6         N/A     disabled        strict
          5         N/A     disabled        strict
          4         N/A      1000000        strict
          3          25      1000000   round-robin
          2          25     disabled   round-robin
          1          25     disabled   round-robin
          0          25     disabled   round-robin

switch(config-if-Et19-txq-3)#
```

shape rate (Tx-queue – Helix)

The **shape rate** command specifies the maximum bandwidth for outbound traffic on the transmit queue, also known as queue shaping. The shape rate for interfaces is configured by the **shape rate (Interface – Helix)** command. By default, the configured outbound transmission rate is not bounded by a transmit queue shape rate.

The **no shape rate** and **default shape rate** commands remove the shape rate bandwidth limit on the configuration mode transmit queue by deleting the corresponding **shape rate** command from ***running-config***.

Command Mode

Tx-Queue Configuration

Command Syntax

```
shape rate byte_limit [kbps]
no shape rate
default shape rate
```

Parameters

- ***DATA_LIMIT*** shape rate applied to the queue. Value range varies with data unit:
  - <***8*** to ***40000000***> 8 to 40,000,000 kbytes per second.
  - <***8*** to ***40000000***>**kbps** 8 to 40,000,000 kbytes per second.
  - <***8*** to ***60000000***>**pps** 8 to 60,000,000 packets per second.

Restrictions

Queue shaping is not supported in cut-through mode.

Related Commands

- **tx-queue (Helix)** places the switch in tx-queue configuration mode.
- **shape rate (Interface – Helix)** configures the shape rate for a configuration mode interface.

Example

- These commands configure a shape rate of 1 Gbps (1,000,000 Kbps) on transmit queues 3 and 4 of Ethernet interface 17/3.

```
switch(config)#interface ethernet 17/3
switch(config-if-Et17/3)#tx-queue 4
switch(config-if-Et17/3-txq-4)#shape rate 1000000 kbps
switch(config-if-Et17/3-txq-4)#tx-queue 3
switch(config-if-Et17/3-txq-3)#shape rate 1000000 kbps
switch(config-if-Et17/3-txq-3)#show qos interface ethernet 17/3
Ethernet17/3:
            <-------OUTPUT OMITTED FROM EXAMPLE-------->
  Tx       Bandwidth                 Shape Rate        Priority
 Queue     Guaranteed (units)         (units)
   ------------------------------------------------------------
   7        - / -    (  -  )       - / -    (  -  )    SP / SP
   6        - / -    (  -  )       - / -    (  -  )    SP / SP
   5        - / -    (  -  )       - / -    (  -  )    SP / SP
   4        - / -    (  -  )       1 / 1    ( Gbps )   SP / SP
   3        - / -    (  -  )       1 / 1    ( Gbps )   SP / SP
   2        - / -    (  -  )       - / -    (  -  )    SP / SP
   1        - / -    (  -  )       - / -    (  -  )    SP / SP
   0        - / -    (  -  )       - / -    (  -  )    SP / SP
            <-------OUTPUT OMITTED FROM EXAMPLE-------->
switch(config-if-Et17/3-txq-3)#
```

shape rate (Tx-queue – Petra)

The **shape rate** command specifies the maximum bandwidth for outbound traffic on the configuration mode transmit queue, also known as queue shaping. The shape rate for interfaces is configured by the **shape rate (Interface – Petra)** command. By default, the configured outbound transmission rate is not bounded by a transmit queue shape rate.

Queue shaping applies only to unicast traffic. Shaping rates of at least 162 Kbps are supported.

The **no shape rate** and **default shape rate** commands remove the shape rate bandwidth limit on the configuration mode queue by deleting the corresponding **shape rate** command from ***running-config***.

Command Mode

Tx-Queue Configuration

Command Syntax

```
shape rate DATA_LIMIT
no shape rate
default shape rate
```

Parameters

- ***DATA_LIMIT*** shape rate applied to the queue. Value range varies with data unit:
  - <***8*** to ***40000000***> 8 to 40,000,000 kbytes per second.
  - <***8*** to ***40000000***>**kbps** 8 to 40,000,000 kbytes per second.
  - <***8*** to ***60000000***>**pps** 8 to 60,000,000 packets per second.

Shaping rates greater than 460 kbps are supported. At lower shaping rates (less than 10 Mbps), granularity and rounding errors may skew the actual shaping rate by 20% from the specified rate.

Related Commands

- **tx-queue (Petra)** places the switch in tx-queue configuration mode
- **shape rate (Interface – Petra)** configures the shape rate for a configuration mode interface.

Example

- These commands configure a shape rate of 1 Gbps (1,000,000 Kbps) on transmit queues 3 and 4 of Ethernet interface 3/28.

```
switch(config)#interface ethernet 3/28
switch(config-if-Et3/28)#tx-queue 4
switch(config-if-Et3/28-txq-4)#shape rate 1000000
switch(config-if-Et3/28-txq-4)#tx-queue 3
switch(config-if-Et3/28-txq-3)#shape rate 1000000
switch(config-if-Et3/28-txq-3)#show qos interface ethernet 3/28
Ethernet3/28:
            <-------OUTPUT OMITTED FROM EXAMPLE-------->
   Tx-Queue   Bandwidth    Shape Rate     Priority
              (percent)       (Kbps)
   ---------------------------------------------
          7         N/A     disabled        strict
          6         N/A     disabled        strict
          5         N/A     disabled        strict
          4         N/A      1000000        strict
          3          25      1000000   round-robin
          2          25     disabled   round-robin
          1          25     disabled   round-robin
          0          25     disabled   round-robin

switch(config-if-Et3/28-txq-3)#
```

shape rate (Tx-queues – Trident)

The **shape rate** command specifies the maximum bandwidth for outbound traffic on the configuration mode transmit queue, also known as queue shaping. The shape rate for interfaces is configured by the **shape rate (Interface – Trident)** command. By default, the configured outbound transmission rate is not bounded by a transmit queue shape rate.

The **no shape rate** and **default shape rate** commands remove the shape rate limit from the configuration mode transmit queue by deleting the corresponding **shape rate** command from ***running-config***.

Command Mode

Mc-Tx-Queue configuration
Uc-Tx-Queue configuration

Command Syntax

```
shape rate DATA_LIMIT
no shape rate
default shape rate
```

Parameters

- ***DATA_LIMIT*** shape rate applied to the queue. Value range varies with data unit:
  - <***8*** to ***40000000***> 8 to 40,000,000 kbytes per second.
  - <***8*** to ***40000000***>**kbps** 8 to 40,000,000 kbytes per second.
  - <***8*** to ***60000000***>**pps** 8 to 60,000,000 packets per second.

Related Commands

- **mc-tx-queue** places the switch in mc-tx-queue configuration mode.
- **uc-tx-queue** places the switch in uc-tx-queue configuration mode.
- **shape rate (Interface – Trident)** configures the shape rate for a configuration mode interface.

Guidelines

Shaping rates of at least 8 kbps are supported. At shaping rates smaller than 1 Mbps, granularity and rounding errors may skew the actual shaping rate by 20% from the specified rate.

When two queues source traffic from the same traffic class and the higher priority queue is shaped, that queue consumes all internal buffers, starving the lower priority queue even if bandwidth is available.

Example

- These commands configure a shape rate of 1 Gbps (1,000,000 Kbps) on unicast transmit queues 3 and multicast transmit 4 of Ethernet interface 7.

```
switch(config)#interface ethernet 7
switch(config-if-Et7)#uc-tx-queue 3
switch(config-if-Et7-uc-txq-3)#shape rate 1000000
switch(config-if-Et7-uc-txq-3)#mc-tx-queue 2
switch(config-if-Et7-mc-txq-2)#shape rate 1000000
switch(config-if-Et7-mc-txq-2)#show qos interface ethernet 7
Ethernet7:
            <-------OUTPUT OMITTED FROM EXAMPLE-------->
   Tx-Queue   Bandwidth    Shape Rate     Priority   Priority Group
              (percent)       (Kbps)
   ------------------------------------------------------------------
        UC7         N/A     disabled        strict                1
        UC6         N/A     disabled        strict                1
        MC3         N/A     disabled        strict                1
        UC5         N/A     disabled        strict                0
        UC4         N/A     disabled        strict                0
        MC2         N/A      1000000        strict                0
        UC3          20      1000000   round-robin                0
        UC2          16     disabled   round-robin                0
        MC1          16     disabled   round-robin                0
        UC1          16     disabled   round-robin                0
        UC0          16     disabled   round-robin                0
        MC0          16     disabled   round-robin                0

switch(config-if-Et7-mc-txq-2)#
```

shape rate (Tx-queue – Trident-II)

The **shape rate** command specifies the maximum bandwidth for outbound traffic on the configuration mode transmit queue, also known as queue shaping. The shape rate for interfaces is configured by the **shape rate (Interface – Trident-II)** command. By default, the configured outbound transmission rate is not bounded by a transmit queue shape rate.

The **no shape rate** and **default shape rate** commands remove the shape rate bandwidth limit on the configuration mode transmit queue by deleting the corresponding **shape rate** command from ***running-config***.

Command Mode

Tx-Queue Configuration

Command Syntax

```
shape rate byte_limit [kbps]
no shape rate
default shape rate
```

Parameters

- ***DATA_LIMIT*** shape rate applied to the queue. Value range varies with data unit:
  - <***8*** to ***40000000***> 8 to 40,000,000 kbytes per second.
  - <***8*** to ***40000000***>**kbps** 8 to 40,000,000 kbytes per second.
  - <***8*** to ***60000000***>**pps** 8 to 60,000,000 packets per second.

Restrictions

Queue shaping is not supported in cut-through mode

Related Commands

- **tx-queue (Trident-II)** places the switch in tx-queue configuration mode.
- **shape rate (Interface – Trident-II)** configures the shape rate for a configuration mode interface.

Example

- These commands configure a shape rate of 1 Gbps (1,000,000 Kbps) on transmit queues 3 and 4 of Ethernet interface 17/3.

```
switch(config)#interface ethernet 17/3
switch(config-if-Et17/3)#tx-queue 4
switch(config-if-Et17/3-txq-4)#shape rate 1000000 kbps
switch(config-if-Et17/3-txq-4)#tx-queue 3
switch(config-if-Et17/3-txq-3)#shape rate 1000000 kbps
switch(config-if-Et17/3-txq-3)#show qos interface ethernet 17/3
Ethernet17/3:
             <-------OUTPUT OMITTED FROM EXAMPLE-------->
  Tx       Bandwidth                 Shape Rate        Priority
 Queue     Guaranteed (units)         (units)
   ------------------------------------------------------------
   7        - / -    (  -  )       - / -    (  -  )    SP / SP
   6        - / -    (  -  )       - / -    (  -  )    SP / SP
   5        - / -    (  -  )       - / -    (  -  )    SP / SP
   4        - / -    (  -  )       1 / 1    ( Gbps )   SP / SP
   3        - / -    (  -  )       1 / 1    ( Gbps )   SP / SP
   2        - / -    (  -  )       - / -    (  -  )    SP / SP
   1        - / -    (  -  )       - / -    (  -  )    SP / SP
   0        - / -    (  -  )       - / -    (  -  )    SP / SP
             <-------OUTPUT OMITTED FROM EXAMPLE-------->
switch(config-if-Et17/3-txq-3)#
```

show platform petraA traffic-class

The **show platform petraA traffic-class** command displays the traffic class assignment on all specified Petra chips. Each chip controls eight Ethernet interfaces. The default traffic class of an interface is specified by the traffic class assigned to the chip that controls the interface.

Traffic class assignments are configured with the **platform petraA traffic-class** command.

Valid command options include:

- **show platform petraA traffic-class** traffic class of all chips on all linecard.
- **show platform petraA *CHIP_NAME* traffic-class** traffic class of specified chip.
- **show platform petraA *MODULE_NAME* traffic-class** traffic class of all chips on specified linecard.

Command Mode

EXEC

Command Syntax

```
show platform petraA traffic-class
show platform petraA CHIP_NAME traffic-class
show platform petraA MODULE_NAME traffic-class
```

Parameters

- ***CHIP_NAME*** Name of Petra chip on linecard that control Ethernet ports. Options include:
  - **petra*cardX*/*chipY*** all ports on PetraA chip *chipY* on linecard *cardX* (7500 Series).
  - **petra-*chipZ*** all ports on PetraA chip *chipZ* (7048 Series)

  7500 Series

  Switches can contain up to eight linecards. *cardX* varies from 3 to 10.

  Each linecard contains six PetraA chips. Each chip controls eight ports. *chipY* varies from 0 to 5:

  - 0 controls ports 1 through 8
  - 1 controls ports 9 through 16
  - 2 controls ports 17 through 24
  - 3 controls ports 25 through 32
  - 4 controls ports 33 through 40
  - 5 controls ports 41 through 48

  7048 Series

  Each switch contains two PetraA chips. *chipZ* varies from 0 to 1:

  - 0 controls ports 1 through 32
  - 1 controls ports 33 through 52
- ***MODULE_NAME*** Name and number of linecard (7500 Series). Options include:
  - **module linecard** *mod_num* . Linecard number (3 to 10).
  - **module** *mod_num* Linecard number (3 to 10).

Related Commands

- **platform petraA traffic-class** configures the default traffic class used by all ports on a specified chip

Example

- This command displays the traffic class of all chips on linecard 3.

```
switch#show platform petraA module linecard 3 traffic-class
Petra3/0 traffic-class: 1
Petra3/1 traffic-class: 1
Petra3/2 traffic-class: 1
Petra3/3 traffic-class: 1
Petra3/4 traffic-class: 5
Petra3/5 traffic-class: 1
switch#
```

show qos interfaces

The **show qos interfaces** command displays the QoS, DSCP, and transmit queue configuration on a specified interface. Information provided by this command includes the ports trust setting, the default CoS value, and the DSCP value.

Command Mode

EXEC

Command Syntax

show qos interfaces *INTERFACE_NAME*

Parameters

- ***INTERFACE_NAME*** Interface For which command returns data. Options include:
  - <no parameter> returns data for all interfaces.
  - **ethernet** *e_num* Ethernet interface specified by *e_num*.
  - **port-channel** *p_num* Port-Channel Interface specified by *p_num*.

Examples

- This command lists the QoS configuration for Ethernet interface 4.

```
switch>show qos interfaces ethernet 4
Ethernet4:
   Trust Mode: COS
   Default COS: 0
   Default DSCP: 0

   Port shaping rate: 5000000Kbps

   Tx-Queue   Bandwidth    ShapeRate      Priority
              (percent)       (Kbps)
   ------------------------------------------------
          0          50     disabled   round-robin
          1          50     disabled   round-robin
          2         N/A     disabled        strict
          3         N/A      1000000        strict
          4         N/A      1000000        strict
          5         N/A      1500000        strict
          6         N/A      2000000        strict

switch>
```

show qos interfaces random-detect ecn

The **show qos interfaces random-detect ecn** command displays the Explicit Congestion Notification (ECN) configuration for each transmit queue on the specified interfaces.

Command Mode

EXEC

Command Syntax

show qos interfaces [*INTERFACE_NAME*] random-detect ecn

Parameters

- ***INTERFACE_NAME*** Interface for which command returns data. Options include:
  - <no parameter> returns data for all interfaces.
  - **ethernet** *e_range* Ethernet interfaces specified by *e_range*.
  - **port-channel** *p_range* Port-Channel Interfaces specified by *p_range*.

Examples

- This command configures ECN parameters for transmit queues 0 through 3 on Ethernet interface 3/5/1, then displays that configuration.

```
switch(config)#interface ethernet 3/5/1
switch(config-if-Et3/5/1)#tx-queue 0
switch(config-if-Et3/5/1-txq-0)#random-detect ecn minimum-threshold 2560 kbytes
maximum-threshold 256000 kbytes
switch(config-if-Et3/5/1-txq-0)#tx-queue 1
switch(config-if-Et3/5/1-txq-1)#random-detect ecn minimum-threshold 25600 kbytes
maximum-threshold 128000 kbytes
switch(config-if-Et3/5/1-txq-1)#tx-queue 2
switch(config-if-Et3/5/1-txq-2)#random-detect ecn minimum-threshold 25600 bytes
maximum-threshold 128000 bytes
switch(config-if-Et3/5/1-txq-2)#tx-queue 3
switch(config-if-Et3/5/1-txq-3)#random-detect ecn minimum-threshold 25 mbytes
maximum-threshold 128 mbytes
switch(config-if-Et3/5/1-txq-3)#show qos interfaces ethernet 3/5/1 random-detect
ecn
Ethernet3/5/1:

   Tx-Queue   Mininimum Threshold    Maximum Threshold         Threshold Unit
   ----------------------------------------------------------------------------
          7                     -                    -                      -
          6                     -                    -                      -
          5                     -                    -                      -
          4                     -                    -                      -
          3                    25                  128                 mbytes
          2                 25600               128000                  bytes
          1                 25600               128000                kbytes
          0                  2560               256000                kbytes

switch(config-if-Et3/5/1-txq-3)#
```

show qos maps

The **show qos maps** command lists the number of traffic classes that the switch supports and displays the CoS-Traffic Class, DSCP-Traffic Class, Traffic Class-CoS, and Traffic Class-Transmit Queue maps.

Command Mode

EXEC

Command Syntax

show qos maps

Examples

- This command displays the QoS maps that are configured on the switch.

```
switch>show qos maps
Number of Traffic Classes supported: 8
   Number of Transmit Queues supported: 8
   Cos Rewrite:  Disabled
   Dscp Rewrite:  Disabled

   Cos-tc map:
     cos:  0  1  2  3  4  5  6  7
     ----------------------------
     tc:   1  0  2  3  4  5  6  7

   Dscp-tc map:
     d1 :  d2 0  1  2  3  4  5  6  7  8  9
     ---------------------------------------
      0 :     1  1  1  1  1  1  1  1  0  0
      1 :     0  0  0  0  0  0  2  2  2  2
      2 :     2  2  2  2  3  3  3  3  3  3
      3 :     3  3  4  4  4  4  4  4  4  4
      4 :     5  5  5  5  5  5  5  5  6  6
      5 :     6  6  6  6  6  6  7  7  7  7
      6 :     7  7  7  7

   Tc-cos map:
     tc:   0  1  2  3  4  5  6  7
     ----------------------------
     cos:  1  0  2  3  4  5  6  7

   Tc-dscp map:
     tc:    0  1  2  3  4  5  6  7
     ------------------------------
     dscp:  8  0 16 24 32 40 48 56

   Tc - tx-queue map:
     tc:        0  1  2  3  4  5  6  7
     ---------------------------------
     tx-queue:  0  1  2  3  4  5  6  7

switch>
```

show qos random-detect ecn

The **show qos random-detetct ecn** command displays the global Explicit Congestion Notification (ECN) configuration.

Command Mode

EXEC

Command Syntax

show qos random-detect ecn

Examples

- These commands configure global ECN parameters, then displays that configuration.

```
switch(config)#qos random-detect ecn global-buffer minimum-threshold 2 mbytes
maximum-threshold 5 mbytes
switch(config)#show qos random-detect ecn
   Minimum Threshold:  2
   Maximum Threshold:  5
   Threshold Unit:  mbytes

switch(config)#
```

show qos interfaces trust

The **show qos interfaces trust** command displays the configured and operational QoS trust mode of all specified interfaces.

Command Mode

EXEC

Command Syntax

show qos interfaces [*INTERFACE_NAME*] trust

Parameters

- ***INTERFACE_NAME*** Interface for which command returns data. Options include:
  - <no parameter> returns data for all interfaces.
  - **ethernet** *e_range* Ethernet interfaces specified by *e_range*.
  - **port-channel** *p_range* Port-Channel Interfaces specified by *p_range*.

Examples

- These commands configure a variety of QoS trust settings on a set of interfaces, then displays the QoS trust mode on these interfaces.

```
switch(config)#interface ethernet 1/1
switch(config-if-Et1/1)#qos trust cos
switch(config-if-Et1/1)#interface ethernet 1/2
switch(config-if-Et1/2)#qos trust dscp
switch(config-if-Et1/2)#interface ethernet 1/3
switch(config-if-Et1/3)#no qos trust
switch(config-if-Et1/3)#interface ethernet 1/4
switch(config-if-Et1/4)#default qos trust
switch(config-if-Et1/4)#interface ethernet 2/1
switch(config-if-Et2/1)#no switchport
switch(config-if-Et2/1)#default qos trust
switch(config-if-Et2/1)#show qos interface ethernet 1/1 - 2/4 trust
Port                                    Trust Mode
                              Operational           Configured
-------------------------------------------------------------
Ethernet1/1                   COS                   COS
Ethernet1/2                   DSCP                  DSCP
Ethernet1/3                   UNTRUSTED             UNTRUSTED
Ethernet1/4                   COS                   DEFAULT
Ethernet2/1                   DSCP                  DEFAULT
Ethernet2/2                   COS                   DEFAULT
Ethernet2/3                   COS                   DEFAULT
Ethernet2/4                   COS                   DEFAULT

switch(config-if-Et2/1)#
```

tx-queue (Arad/Jericho)

The **tx-queue** command places the switch in Tx-queue configuration mode to configure a transmit queue on the configuration mode interface. Tx-queue configuration mode is not a group change mode; ***running-config*** is changed immediately after commands are executed. The **exit** command does not affect the configuration.

Arad and Jericho platform switches have eight queues, ***0*** through ***7***, and all queues are exposed through the CLI. However, queue 7 is not user-configurable. Queue 7 is always mapped to traffic class 7, which is reserved for control traffic.

The **exit** command returns the switch to the configuration mode for the base Ethernet or port channel interface.

The **no tx-queue** and **default tx-queue** commands remove the configuration for the specified transmit queue by deleting all corresponding **tx-queue** mode statements from ***running-config***.

Command Mode

Interface-Ethernet Configuration
Interface-Port-Channel Configuration

Command Syntax

```
tx-queue queue_level
```

Parameters

- *queue_level* the transmit queue. Values range from ***0*** to ***6***.

Commands Available in tx-queue Configuration Mode

- **bandwidth percent (Arad/Jericho)**
- **priority (Arad/Jericho)**
- **shape rate (Tx-queue – Arad/Jericho)**

Guidelines

Arad and Jericho platform switch queues handle unicast traffic. Queues for multicast traffic are not supported.

Example

- This command enters Tx-queue configuration mode for transmit queue 4 of Ethernet interface 3/3/3.

```
switch(config)#interface ethernet 3/3/3
switch(config-if-Et3/3/3)#tx-queue 4
switch(config-if-Et3/3/3-txq-4)#
```

tx-queue (FM6000)

The **tx-queue** command places the switch in Tx-queue configuration mode to configure a transmit queue on the configuration mode interface. Tx-queue configuration mode is not a group change mode; ***running-config*** is changed immediately after commands are executed. The **exit** command does not affect the configuration.

FM6000 platform switches have eight queues, ***0*** through ***7***. All queues are exposed through the CLI and are user configurable.

The **exit** command returns the switch to the configuration mode for the base Ethernet or port channel interface.

The **no tx-queue** and **default tx-queue** commands remove the configuration for the specified transmit queue by deleting the all corresponding **tx-queue** mode commands from ***running-config***.

Command Mode

Interface-Ethernet Configuration
Interface-Port-Channel Configuration

Command Syntax

```
tx-queue queue_level
```

Parameters

- *queue_level* the transmit queue. Values range from ***0*** to ***7***.

Commands Available in tx-queue Configuration Mode

- **bandwidth percent (FM6000)**
- **priority (FM6000)**
- **shape rate (Tx-queue – FM6000)**

Guidelines

FM6000 platform switch queues handle unicast and multicast traffic.

Example

- This command enters Tx-queue configuration mode for transmit queue 3 of Ethernet interface 5.

```
switch(config)#interface ethernet 5
switch(config-if-Et5)#tx-queue 3
switch(config-if-Et5-txq-3)#
```

tx-queue (Helix)

The **tx-queue** command places the switch in Tx-queue configuration mode to configure a transmit queue on the configuration mode interface. Tx-queue configuration mode is not a group change mode; ***running-config*** is changed immediately after commands are executed. The **exit** command does not affect the configuration.

Helix platform switches have eight unicast (UC0 – UC7) and eight multicast (MC0 – MC7) queues. Each UCx-MCx queue set is combined into a single queue group (L1.x), which is exposed to the CLI through this command.

The **exit** command returns the switch to the configuration mode for the base Ethernet or port channel interface.

The **no tx-queue** and **default tx-queue** commands remove the configuration for the specified transmit queue by deleting the all corresponding **tx-queue** mode commands from ***running-config***.

Command Mode

Interface-Ethernet Configuration
Interface-Port-Channel Configuration

Command Syntax

```
tx-queue queue_level
```

Parameters

- *queue_level* transmit queue group number. Values range from ***0*** to ***7***.

Commands Available in tx-queue Configuration Mode

- **bandwidth guaranteed (Helix)**
- **shape rate (Tx-queue – Helix)**

Guidelines

Helix platform switch queues handle unicast and multicast traffic.

Example

- This command enters Tx-queue configuration mode for transmit queue 4 of Ethernet interface 17/3.

```
switch(config)#interface ethernet 17/3
switch(config-if-Et17/3)#tx-queue 4
switch(config-if-Et17/3-txq-4)#
```

tx-queue (Petra)

The **tx-queue** command places the switch in Tx-queue configuration mode to configure a transmit queue on the configuration mode interface. Tx-queue configuration mode is not a group change mode; ***running-config*** is changed immediately after commands are executed. The **exit** command does not affect the configuration.

Petra platform switches have eight queues, ***0*** through ***7***, and all queues are exposed through the CLI. However, queue 7 is not user-configurable. Queue 7 is always mapped to traffic class 7, which is reserved for control traffic.

The **exit** command returns the switch to the configuration mode for the base Ethernet or port channel interface.

The **no tx-queue** and **default tx-queue** commands remove the configuration for the specified transmit queue by deleting the all corresponding **tx-queue** mode commands from ***running-config***.

Command Mode

Interface-Ethernet Configuration
Interface-Port-Channel Configuration

Command Syntax

```
tx-queue queue_level
```

Parameters

- *queue_level* the transmit queue. Values range from ***0*** to ***6***.

Commands Available in tx-queue Configuration Mode

- **bandwidth percent (Petra)**
- **priority (Petra)**
- **shape rate (Tx-queue – Petra)**

Guidelines

Petra platform switch queues handle unicast traffic. Queues for multicast traffic are not supported.

Example

- This command enters Tx-queue configuration mode for transmit queue 3 of Ethernet interface 3/3.

```
switch(config)#interface ethernet 3/3
switch(config-if-Et3/3)#tx-queue 3
switch(config-if-Et3/3-txq-3)#
```

tx-queue (Trident-II)

The **tx-queue** command places the switch in Tx-queue configuration mode to configure a transmit queue on the configuration mode interface. Tx-queue configuration mode is not a group change mode; ***running-config*** is changed immediately after commands are executed. The **exit** command does not affect the configuration.

Trident-II platform switches have eight unicast (UC0 – UC7) and eight multicast (MC0 – MC7) queues. Each UCx-MCx queue set is combined into a single queue group (L1.x), which is exposed to the CLI through this command.

The **exit** command returns the switch to the configuration mode for the base Ethernet or port channel interface.

The **no tx-queue** and **default tx-queue** commands remove the configuration for the specified transmit queue by deleting the all corresponding **tx-queue** mode commands from ***running-config***.

Command Mode

Interface-Ethernet Configuration
Interface-Port-Channel Configuration

Command Syntax

```
tx-queue queue_level
```

Parameters

- *queue_level* transmit queue group number. Values range from ***0*** to ***7***.

Commands Available in tx-queue Configuration Mode

- **bandwidth guaranteed (Trident-II)**
- **shape rate (Tx-queue – Trident-II)**

Guidelines

Trident-II platform switch queues handle unicast and multicast traffic.

Example

- This command enters Tx-queue configuration mode for transmit queue 4 of Ethernet interface 17/3.

```
switch(config)#interface ethernet 17/3
switch(config-if-Et17/3)#tx-queue 4
switch(config-if-Et17/3-txq-4)#
```

uc-tx-queue

The **uc-tx-queue** command places the switch in uc-tx-queue configuration mode to configure a unicast transmit queue on the configuration mode interface. Uc-tx-queue configuration mode is not a group change mode; ***running-config*** is changed immediately after commands are executed. The **exit** command does not affect the configuration.

Trident switches have eight unicast queues (UC0 – UC7) and four multicast queues (MC0 – MC03), categorized into two priority groups. All queues are exposed through the CLI and are user-configurable.

- Priority Group 1: UC7, UC6, MC3
- Priority Group 0: UC5, UC4, MC2, UC3, UC2, MC1, UC1, UC0, MC0

The **exit** command returns the switch to the configuration mode for the base Ethernet or port channel interface.

The **no uc-tx-queue** and **default uc-tx-queue** commands remove the configuration for the specified transmit queue by deleting the all corresponding **uc-tx-queue** mode commands from ***running-config***.

Command Mode

Interface-Ethernet Configuration
Interface-Port-Channel Configuration

Command Syntax

```
uc-tx-queue queue_level
```

Parameters

- *queue_level* The multicast transmit queue number. Values range from ***0*** to ***7***.

Commands Available in uc-tx-queue Configuration Mode

- **bandwidth percent (Trident)**
- **priority (Trident)**
- **shape rate (Tx-queues – Trident)**

Related Commands

- **mc-tx-queue**: Configures multicast transmit queues on Trident platform switches.

Example

- This command enters mc-tx-queue configuration mode for multicast transmit queue 3 of Ethernet interface 5.

```
switch(config)#interface ethernet 5
switch(config-if-Et5)#uc-tx-queue 4
switch(config-if-Et5-mc-txq-4)#
```

Chapter 24

IPv4

Arista switches support Internet Protocol version 4 (IPv4) and Internet Protocol version 6 (IPv6) for routing packets across network boundaries. This chapter describes Arista's implementation of IPv4 and includes these sections:

- Section 24.1: IPv4 Addressing
- Section 24.2: IPv4 Routing
- Section 24.3: IPv4 Multicast Counters
- Section 24.4: Route Management
- Section 24.5: IPv4 Route Scale
- Section 24.6: IP Source Guard
- Section 24.7: DHCP Relay Across VRF
- Section 24.8: IP NAT
- Section 24.9: IPv4 Command Descriptions

24.1 IPv4 Addressing

Each IPv4 network device is assigned a 32-bit IP address that identifies its network location. These sections describe IPv4 address formats, data structures, configuration tasks, and display options:

- Section 24.1.1: IPv4 Address Formats
- Section 24.1.2: IPv4 Address Configuration
- Section 24.1.3: Address Resolution Protocol (ARP)
- Section 24.2.4: Viewing IPv4 Routes and Network Components

24.1.1 IPv4 Address Formats

IPv4 addresses are composed of 32 bits, expressed in dotted decimal notation by four decimal numbers, each ranging from 0 to 255. A subnet is identified by an IP address and an address space defined by a routing prefix. The switch supports the following subnet formats:

- **IP address and subnet mask:** The subnet mask is a 32-bit number (dotted decimal notation) that specifies the subnet address space. The subnet address space is calculated by performing an AND operation between the IP address and subnet mask.
- **IP address and wildcard mask:** The wildcard mask is a 32-bit number (dotted decimal notation) that specifies the subnet address space. Wildcard masks differ from subnet masks in that the bits are inverted. Some commands use wildcard masks instead of subnet masks.

- **CIDR notation:** CIDR notation specifies the scope of the subnet space by using a decimal number to identify the number of leading ones in the routing prefix. When referring to wildcard notation, CIDR notation specifies the number of leading zeros in the routing prefix.

Example

- These subnets (subnet mask and CIDR notation) are calculated identically:

```
10.24.154.13 255.255.255.0
10.24.154.13/24
```

  The defined space includes all addresses between ***10.24.154.0*** and ***10.24.154.255***.

- These subnets (wildcard mask and CIDR notation) are calculated identically:

```
124.17.3.142 0.0.0.15
124.17.3.142/28
```

  The defined space includes all addresses between ***124.17.3.128*** and ***124.17.3.143***.

24.1.2 IPv4 Address Configuration

Assigning an IPv4 Address to an Interface

The **ip address** command specifies the IPv4 address of an interface and the mask for the subnet to which the interface is connected.

Example

- These commands configure an IPv4 address with subnet mask for VLAN 200:

```
switch(config)#interface vlan 200
switch(config-if-Vl200)#ip address 10.0.0.1/24
switch(config-if-Vl200)#
```

24.1.3 Address Resolution Protocol (ARP)

Address Resolution Protocol (ARP) is a protocol that maps IP addresses to MAC addresses that local network devices recognize. The ARP cache is a table that stores the correlated addresses of the devices for which the router facilitates data transmissions.

After receiving a packet, routers use ARP to find the MAC address of the device assigned to the packet's destination IP address. If the ARP cache contains both addresses, the router sends the packet to the specified port. If the ARP cache does not contain the addresses, ARP broadcasts a request packet to all devices in the subnet. The device at the requested IP address responds and provides its MAC address. ARP updates the ARP cache with a dynamic entry and forwards the packet to the responding device. Static ARP entries can also be added to the cache through the CLI.

Proxy ARP is an ARP variant. A network device (proxy) responds to ARP requests for network addresses on a different network with its MAC address. Traffic to the destination is directed to the proxy device which then routes the traffic toward the ultimate destination.

Configuring ARP

The switch uses ARP cache entries to correlate 32-bit IP addresses to 48-bit hardware addresses. The **arp timeout** command specifies the duration of dynamic address entries in the Address Resolution Protocol (ARP) cache for addresses learned through the layer 3 interface. The default duration is 14400 seconds (four hours).

Static ARP entries never time out and must be removed from the table manually.

Example

- This command specifies an ARP cache duration of 7200 seconds (two hours) for dynamic addresses added to the ARP cache that were learned through VLAN 200.

```
switch(config)#interface vlan 200
switch(config-if-Vl200)#arp timeout 7200
switch(config-if-Vl200)#show active
interface Vlan200
   arp timeout 7200
switch(config-if-Vl200)#
```

The **arp** command adds a static entry to an Address Resolution Protocol (ARP) cache.

Example

- This command adds a static entry to the ARP cache in the default VRF.

```
switch(config)#arp 172.22.30.52 0025.900e.c63c arpa
switch(config)#
```

Displaying ARP Entries

The **show ip arp** command displays ARP cache entries that map an IP address to a corresponding MAC address. The table displays addresses by their host names when the command includes the ***resolve*** argument.

Example

- This command displays ARP cache entries that map MAC addresses to IPv4 addresses.

```
switch>show ip arp
Address         Age (min)  Hardware Addr   Interface
172.25.0.2              0  004c.6211.021e  Vlan101, Port-Channel2
172.22.0.1              0  004c.6214.3699  Vlan1000, Port-Channel1
172.22.0.2              0  004c.6219.a0f3  Vlan1000, Port-Channel1
172.22.0.3              0  0045.4942.a32c  Vlan1000, Ethernet33
172.22.0.5              0  f012.3118.c09d  Vlan1000, Port-Channel1
172.22.0.6              0  00e1.d11a.a1eb  Vlan1000, Ethernet5
172.22.0.7              0  004f.e320.cd23  Vlan1000, Ethernet6
172.22.0.8              0  0032.48da.f9d9  Vlan1000, Ethernet37
172.22.0.9              0  0018.910a.1fc5  Vlan1000, Ethernet29
172.22.0.11             0  0056.cbe9.8510  Vlan1000, Ethernet26
switch>
```

- This command displays ARP cache entries that map MAC addresses to IPv4 addresses. Host names assigned to IP addresses are displayed in place of the address.

```
switch>show ip arp resolve
Address         Age (min)  Hardware Addr   Interface
green-vl101.new         0  004c.6211.021e  Vlan101, Port-Channel2
172.22.0.1              0  004c.6214.3699  Vlan1000, Port-Channel1
orange-vl1000.n         0  004c.6219.a0f3  Vlan1000, Port-Channel1
172.22.0.3              0  0045.4942.a32c  Vlan1000, Ethernet33
purple.newcompa         0  f012.3118.c09d  Vlan1000, Port-Channel1
pink.newcompany         0  00e1.d11a.a1eb  Vlan1000, Ethernet5
yellow.newcompa         0  004f.e320.cd23  Vlan1000, Ethernet6
172.22.0.8              0  0032.48da.f9d9  Vlan1000, Ethernet37
royalblue.newco         0  0018.910a.1fc5  Vlan1000, Ethernet29
172.22.0.11             0  0056.cbe9.8510  Vlan1000, Ethernet26
switch>
```

24.1.3.1 ARP Inspection

Address Resolution Protocol (ARP) inspection command **ip arp inspection vlan** activates a security feature that protects the network from ARP spoofing. ARP requests and responses on untrusted interfaces are intercepted on specified VLANs, and intercepted packets are verified to have valid IP-MAC address bindings. All invalid ARP packets are dropped. On trusted interfaces, all incoming ARP packets are processed and forwarded without verification.

Enabling and Disabling ARP Inspection

By default, ARP inspection is disabled on all VLANs.

Examples

- This command enables ARP inspection on VLANs 1 through 150.

```
switch(config)#ip arp inspection vlan 1 - 150
switch(config)#
```

- This command disables ARP inspection on VLANs 1 through 150.

```
switch(config)#no ip arp inspection vlan 1 - 150
switch(config)#
```

- This command sets the ARP inspection default to VLANs 1 through 150.

```
switch(config)#default ip arp inspection vlan 1 - 150
switch(config)#
```

- These commands enable ARP inspection on multiple VLANs 1 through 150 and 200 through 250.

```
switch(config)#ip arp inspection vlan 1-150,200-250
switch(config)#
```

Syslog for Invalid ARP Packets Dropped

When an invalid ARP packet is dropped, the following syslog message appears. The log severity level can be set higher if required.

```
%SECURITY-4-ARP_PACKET_DROPPED: Dropped ARP packet on interface Ethernet28/1 Vlan
2121 because invalid mac and ip binding. Received: 00:0a:00:bc:00:de/1.1.1.1.
```

Displaying ARP Inspection States

The command **show ip arp inspection vlan** displays the configuration and operation state of ARP inspection. For a VLAN range specified by **show ip arp inspection vlan** only VLANs with ARP inspection enabled will be displayed. If no VLAN is specified, all VLANs with ARP inspection enabled are displayed. The operation state turns to `Active` when hardware is ready to trap ARP packets for inspection.

Example

- This command displays the configuration and operation state of ARP inspection for VLANs 1 through 150.

```
switch(config)#show ip arp inspection vlan 1 - 150
VLAN 1
----------
Configuration
: Enabled
Operation State : Active
VLAN 2
----------
Configuration
: Enabled
Operation State : Active
{...}
VLAN 150
----------
Configuration
: Enabled
Operation State : Active

switch(config)#
```

Displaying ARP Inspection Statistics

The command **show ip arp inspection statistics** displays the statistics of inspected ARP packets. For a VLAN specified by **show ip arp inspection vlan** only VLANs with ARP inspection enabled will be displayed. If no VLAN is specified, all VLANs with ARP inspection enabled are displayed.

The command **clear ip arp inspection statistics** clears ARP inspection.

Examples

- This command displays ARP inspection statistics for VLAN 1.

```
switch(config)#show ip arp inspection statistics vlan 2
Vlan : 2
------------
ARP Req Forwarded = 20
ARP Res Forwarded = 20
ARP Req Dropped = 1
ARP Res Dropped = 1

Last invalid ARP:
Time: 10:20:30 ( 5 minutes ago )
Reason: Bad IP/Mac match
Received on: Ethernet 3/1
Packet:
  Source MAC: 00:01:00:01:00:01
  Dest MAC: 00:02:00:02:00:02
  ARP Type: Request
  ARP Sender MAC: 00:01:00:01:00:01
  ARP Sender IP: 1.1.1

switch(config)#
```

- This command displays ARP inspection statistics for Ethernet interface 3/1.

```
switch(config)#show ip arp inspection statistics ethernet interface 3/1
Interface : 3/1
--------
ARP Req Forwarded = 10
ARP Res Forwarded = 10
ARP Req Dropped = 1
ARP Res Dropped = 1

Last invalid ARP:
Time: 10:20:30 ( 5 minutes ago )
Reason: Bad IP/Mac match
Received on: VLAN 10
Packet:
  Source MAC: 00:01:00:01:00:01
  Dest MAC: 00:02:00:02:00:02
  ARP Type: Request
  ARP Sender MAC: 00:01:00:01:00:01
  ARP Sender IP: 1.1.1

switch(config)#
```

- This command clears ARP inspection statistics.

```
switch(config)#clear ip arp inspection statistics
switch(config)#
```

Configure Trust Interface

By default, all interfaces are untrusted. The command **ip arp inspection trust** configures the trust state of an interface.

Examples

- This command configures the trust state of an interface.

```
switch(config)#ip arp inspection trust
switch(config)#
```

- This command configures the trust state of an interface to untrusted.

```
switch(config)#no ip arp inspection trust
switch(config)#
```

- This command configures the trust state of an interface to its default (untrusted).

```
switch(config)#default ip arp inspection trust
switch(config)#
```

Configure Rate Limit

When ARP inspection is enabled, ARP packets are trapped to the CPU. Two actions can be taken when the incoming ARP rate exceeds expectation. For notification purpose, the command **ip arp inspection logging** will enable logging of the incoming ARP packets. To prevent a denial-of-service attack, the command **ip arp inspection limit** will error-disable interfaces.

Examples

- This command enables logging of incoming ARP packets when its rate exceeds the configured value, and sets the rate to 2048 (which is the upper limit for the number of invalid ARP packets allowed per second), and sets the burst consecutive interval over which the interface is monitored for a high ARP rate to 15 seconds.

```
switch(config)#ip arp inspection logging rate 2048 burst interval 15
switch(config)#
```

- This command configures the rate limit of incoming ARP packets to errdisable the interface when the incoming ARP rate exceeds the configured value, sets the rate to 512 (which is the upper limit for the number of invalid ARP packets allowed per second), and sets the burst consecutive interval over which the interface is monitored for a high ARP rate to 11 seconds.

```
switch(config)#ip arp inspection limit rate 512 burst interval 11
switch(config)#
```

- This command displays verification of the interface specific configuration.

```
switch(config)#interface Ethernet 3 / 1
switch(config)#ip arp inspection limit rate 20 burst interval 5
switch(config)#interface Ethernet 3 / 3
switch(config)#ip arp inspection trust
switch(config)#show ip arp inspection interfaces
 Interface      Trust State  Rate (pps) Burst Interval
 -------------  -----------  ---------- --------------
 Et3/1          Untrusted    20         5
 Et3/3          Trusted      None       N/A

switch(config)#
```

Configure Errdisable Caused by ARP Inspection

If the incoming ARP packet rate on an interface exceeds the configured rate limit in burst interval, the interface will be errdisabled (by default). If errdisabled, the interface will stay in this state until you intervene with the command **errdisable detect cause arp-inspection** (e.g., after you perform a **shutdown** or **no shutdown** of the interface) or it automatically recovers after a certain time period. The command **errdisable recovery cause arp-inspection** will enable auto recovery. The command **errdisable recovery interval** will enable sharing the auto recovery interval among all errdisable interfaces. (See the chapter “Data Transfer” for information on all **errdisable** commands.

Examples

- This command enables errdisable caused by an ARP inspection violation.

```
switch(config)#errdisable detect cause arp-inspection
switch(config)#
```

- This command disables errdisable caused by an ARP inspection violation.

```
switch(config)#no errdisable detect cause arp-inspection
switch(config)#
```

- This command enables auto recovery.

```
switch(config)#errdisable recovery cause arp-inspection
switch(config)#
```

- This command disables auto recovery.

```
switch(config)#no errdisable recovery cause arp-inspection
switch(config)#
```

- This command enables sharing the auto recovery interval of 10 seconds among all errdisable interfaces.

```
switch(config)#errdisable recovery interval 10
switch(config)#
```

- This command disables sharing the auto recovery interval of 10 seconds among all errdisable interfaces.

```
switch(config)#no errdisable recovery interval 10
switch(config)#
```

- This command displays the reason for a port entering the errdisable state.

```
switch(config)#show interfaces status errdisabled
 Port                Name             Status             Reason
-------------------- ---------------- ------------------ ------------------
 Et3/2                                errdisabled        arp-inspection

switch(config)#
```

Configure Static IP MAC Binding

The ARP inspection command **ip source binding** allows users to add static IP-MAC binding. If enabled, ARP inspection verifies incoming ARP packets based on the configured IP-MAC bindings. The static IP-MAC binding entry can only be configured on Layer 2 ports. By default, there is no binding entry on the system.

Examples

- This command configures static IP-MAC binding for IP address 127.0.0.1, MAC address 0001.0001.0001, VLAN 1, and Ethernet interface slot 4 and port 1.

```
switch(config)#ip source binding 127.0.0.1 0001.0001.0001 vlan 1 interface
ethernet 4/1
switch(config)#
```

- This command configures static IP-MAC binding for IP address 127.0.0.1, MAC address 0001.0001.0001, VLAN 1, and port-channel interface 20.

```
switch(config)#ip source binding 127.0.0.1 0001.0001.0001 vlan 1 interface
port-channel 20
switch(config)#
```

- This command displays the configured IP-MAC binding entries. Note that the Lease column is mainly used for displaying dynamic DHCP snooping binding entries. For static binding entries, lease time is shown as infinite.

```
switch(config)#show ip source binding 127.0.0.1 0001.0001.0001 static vlan 1
interface port-channel 20
MacAddress       IpAddress   Lease(sec)     Type   VLAN           Interface
---------------- ----------- ------------ -------- ------- --------------------
0001.0001.0001   127.0.0.1     infinite   static   1       Port-Channel20
switch(config)#
```

24.2 IPv4 Routing

Internet Protocol version 4 (IPv4) is a communications protocol used for relaying network packets across a set of connected networks using the Internet Protocol suite. Routing transmits network layer data packets over connected independent subnets. Each subnet is assigned an IP address range and each device on the subnet is assigned an IP address from that range. The connected subnets have IP address ranges that do not overlap.

A router is a network device that connects multiple subnets. Routers forward inbound packets to the subnet whose address range includes the packets' destination address. IPv4 and IPv6 are internet layer protocols that define packet-switched internetworking, including source-to-destination datagram transmission across multiple networks.

These sections describe IPv4 routing and route creation options:

- Section 24.2.1: Enabling IPv4 Routing
- Section 24.2.2: Static and Default IPv4 Routes
- Section 24.2.3: Dynamic IPv4 Routes
- Section 24.2.4: Viewing IPv4 Routes and Network Components

24.2.1 Enabling IPv4 Routing

When IPv4 routing is enabled, the switch attempts to deliver inbound packets to destination IPv4 addresses by forwarding them to interfaces or next hop addresses specified by the forwarding table.

The **ip routing** command enables IPv4 routing.

Example

- This command enables IP routing:

```
switch(config)#ip routing
switch(config)#
```

24.2.2 Static and Default IPv4 Routes

Static routes are entered through the CLI and are typically used when dynamic protocols are unable to establish routes to a specified destination prefix. Static routes are also useful when dynamic routing protocols are not available or appropriate. Creating a static route associates a destination IP address with a local interface. The routing table refers to these routes as ***connected*** routes that are available for redistribution into routing domains defined by dynamic routing protocols.

The **ip route** command creates a static route. The destination is a network segment; the nexthop is either an IP address or a routable interface port. When multiple routes exist to a destination prefix, the route with the lowest administrative distance takes precedence.

By default, the administrative distance assigned to static routes is 1. Assigning a higher administrative distance to a static route configures it to be overridden by dynamic routing data. For example, a static route with a distance value of 200 is overridden by OSPF intra-area routes, which have a default distance of 110.

A route tag is a 32-bit number that is attached to a route. Route maps use tags to filter routes. Static routes have a default tag value of 0.

Example

- This command creates a static route:

```
switch(config)#ip route 172.17.252.0/24 vlan 500
switch(config)#
```

Creating Default IPv4 Routes

The default route denotes the packet forwarding rule that takes effect when no other route is configured for a specified IPv4 address. All packets with destinations that are not established in the routing table are sent to the destination specified by the default route.

The IPv4 destination prefix is 0.0.0.0/0 and the next-hop is the default gateway.

Example

- This command creates a default route and establishes 192.14.0.4 as the default gateway address:

```
switch(config)#ip route 0.0.0.0/0 192.14.0.4
switch(config)#
```

24.2.3 Dynamic IPv4 Routes

Dynamic routes are established by dynamic routing protocols. These protocols also maintain the routing table and modify routes to adjust for topology or traffic changes. Routing protocols assist the switch in communicating with other devices to exchange network information, maintaining routing tables, and establishing data paths.

The switch supports these dynamic IPv4 routing protocols:

- **Open Shortest Path First – Version 2**
- **Border Gateway Protocol (BGP)**
- **Routing Information Protocol**
- **IS-IS**

24.2.4 Viewing IPv4 Routes and Network Components

Displaying the FIB and Routing Table

The **show ip route** command displays routing table entries that are in the forwarding information base (FIB), including static routes, routes to directly connected networks, and dynamically learned routes. Multiple equal-cost paths to the same prefix are displayed contiguously as a block, with the destination prefix displayed only on the first line.

The **show running-config** command displays configured commands not in the FIB. The **show ip route summary** command displays the number of routes, categorized by source, in the routing table.

Examples

- This command displays IP routes learned through BGP.

```
switch>show ip route bgp
Codes: C - connected, S - static, K - kernel,
       O - OSPF, IA - OSPF inter area, E1 - OSPF external type 1,
       E2 - OSPF external type 2, N1 - OSPF NSSA external type 1,
       N2 - OSPF NSSA external type2, B I - iBGP, B E - eBGP,
       R - RIP, A - Aggregate

 B E    170.44.48.0/23 [20/0] via 170.44.254.78
 B E    170.44.50.0/23 [20/0] via 170.44.254.78
 B E    170.44.52.0/23 [20/0] via 170.44.254.78
 B E    170.44.54.0/23 [20/0] via 170.44.254.78
 B E    170.44.254.112/30 [20/0] via 170.44.254.78
 B E    170.53.0.34/32 [1/0] via 170.44.254.78
 B I    170.53.0.35/32 [1/0] via 170.44.254.2
                             via 170.44.254.13
                             via 170.44.254.20
                             via 170.44.254.67
                             via 170.44.254.35
                             via 170.44.254.98

switch>
```

- This command displays a summary of routing table contents.

```
switch>show ip route summary
Route Source         Number Of Routes
-------------------------------------
connected                   15
static                       0
ospf                        74
  Intra-area: 32 Inter-area:33 External-1:0 External-2:9
  NSSA External-1:0 NSSA External-2:0
bgp                          7
  External: 6 Internal: 1
internal                    45
attached                    18
aggregate                    0
switch>
```

Displaying the IP Route Age

The **show ip route age** command displays the current state of the routing table and specifies the last time the route was updated.

Example

- This command displays the amount of time since the last update to ip route 172.17.0.0/20.

```
switch>show ip route 172.17.0.0/20 age
Codes: C - connected, S - static, K - kernel,
       O - OSPF, IA - OSPF inter area, E1 - OSPF external type 1,
       E2 - OSPF external type 2, N1 - OSPF NSSA external type 1,
       N2 - OSPF NSSA external type2, B I - iBGP, B E - eBGP,
       R - RIP, I - ISIS, A - Aggregate

 B E    172.17.0.0/20 via 172.25.0.1, age 3d01h
switch>
```

Displaying Gateways

A gateway is a router that provides access to another network. The gateway of last resort, also known as the default route, is the route that a packet uses when the route to its destination address is unknown. The IPv4 default route in is 0.0.0.0/0.

The **show ip route gateway** command displays IP addresses of all gateways (next hops) used by active routes.

Example

- This command displays next hops used by active routes.

```
switch>show ip route gateway
The following gateways are in use:
   172.25.0.1 Vlan101
   172.17.253.2 Vlan2000
   172.17.254.2 Vlan2201
   172.17.254.11 Vlan2302
   172.17.254.13 Vlan2302
   172.17.254.17 Vlan2303
   172.17.254.20 Vlan2303
   172.17.254.66 Vlan2418
   172.17.254.67 Vlan2418
   172.17.254.68 Vlan2768
   172.17.254.29 Vlan3020
switch>
```

Displaying Host Routes

The **show ip route host** command displays all host routes in the host forwarding table. Host routes are those whose destination prefix is the entire address (mask = 255.255.255.255 or prefix = /32). Each displayed host route is labeled with its purpose:

- F static routes from the FIB.
- R routes defined because the IP address is an interface address.
- B broadcast address.
- A routes to any neighboring host for which the switch has an ARP entry.

Example

- This command displays all host routes in the host forwarding table.

```
switch#show ip route host
R - receive B - broadcast F - FIB, A - attached

F   127.0.0.1 to cpu
B   172.17.252.0 to cpu
A   172.17.253.2 on Vlan2000
R   172.17.253.3 to cpu
A   172.17.253.10 on Vlan2000
R   172.17.254.1 to cpu
A   172.17.254.2 on Vlan2901
B   172.17.254.3 to cpu
B   172.17.254.8 to cpu
A   172.17.254.11 on Vlan2902
R   172.17.254.12 to cpu

F   172.26.0.28 via 172.17.254.20 on Vlan3003
                via 172.17.254.67 on Vlan3008
                via 172.17.254.98 on Vlan3492
via 172.17.254.86 on Vlan3884
                via 172.17.253.2 on Vlan3000
F   172.26.0.29 via 172.25.0.1 on Vlan101
F   172.26.0.30 via 172.17.254.29 on Vlan3910
F   172.26.0.31 via 172.17.254.33 on Vlan3911
F   172.26.0.32 via 172.17.254.105 on Vlan3912
switch#
```

24.3 IPv4 Multicast Counters

IPv4 multicast counters allow association of IPv4 multicast routes with a packet or byte counter.

This chapter contains the following sections.

- Section 24.3.1: Multicast Counters Hardware Overview
- Section 24.3.2: Multicast Counters iBGP and eBGP Configuration
- Section 24.3.3: Multicast Counters CLI

24.3.1 Multicast Counters Hardware Overview

This section describes a hardware overview for multicast counters, and contains the following sections.

- Section 24.3.1.1: Platform Independent Requirements for Counters
- Section 24.3.1.2: Policer Counter Overview
- Section 24.3.1.3: BGP Functions Supported for Arista Switches
- Section 24.3.1.4: Additional Requirements

24.3.1.1 Platform Independent Requirements for Counters

The following platform independent requirements include:

- Enable/Disable counters
- Clear counters
- Show counters
- Configure counter mode for byte (default) or frame mode

24.3.1.2 Policer Counter Overview

The switch hardware has two policer banks, each with 4k entries and each entry has one 32 bit entry1, and one 32 bit entry2, which can be used as either packet counter or byte counter.

In the pipeline, each bank can have one policer index coming from upstream blocks, which means different features cannot update multiple policer entries in the same bank simultaneously. Therefore, different features cannot share entries in the same bank.

In switch hardware routing, each FFU/BST entry points to a corresponding RAM. A policer index is saved in the action ram, so when installing a multicast route into hardware, platform code will get a policer index and saved in the action field. If a policer index is unavailable, a counter is not added to the action field.

Switch hardware can have multiple features competing for the policer banks. It is desirable to have a platform command to reserve policer banks dedicated for a certain feature.

The following command reserves one or two policer banks to be used only by the named feature:

```
[no] platform fm6000 [nat | acl | qos | multicast] policer banks <1|2>
```

Available bank(s) are reserved for the feature. Otherwise the command takes effect at the next reboot or FocalPointV2 agent restart. This reservation guarantees the configured number of bank(s) for this feature. However, the feature can still possibly obtain the other policer bank if it needs more, and the other bank is available.

If a feature has a pending reservation request which is not fulfilled because of availability, and some other feature frees a bank, the bank will be allocated to the pending feature.

24.3.1.3 BGP Functions Supported for Arista Switches

Arista switches support these BGP functions:

- A single BGP instance
- Simultaneous internal (IBGP) and external (EBGP) peering
- Multiprotocol BGP
- BGP Confederations

24.3.1.4 Additional Requirements

On switch hardware, the following additional requirements include:

- Reservation of policer banks
- Notification of policer bank availability when a policer entry is freed by other features

24.3.2 Multicast Counters iBGP and eBGP Configuration

This section describes the commands required to configure an iBGP and an eBGP topology, and contains the following sections.

- Section 24.3.2.1: Policer Usage

24.3.2.1 Policer Usage

There are two types of counters – those created by wildcard creation and by specific creation. When a specific counter is required and the hardware runs out of policer entries, a wildcard counter is forced to give up its policer entry.

If the user configures a specific counter and the starter group (SG) already has a wildcard-created counter for it, then this counter is upgraded to a specific one, with no change in hardware policer index. If the user configures both a wildcard counter and specific counter for this SG, and subsequently deletes the specific counter, the counter for this SG is downgraded to a wildcard, with no change in hardware policer index. However, if another specific counter is pending for a hardware policer index, then this policer entry will be assigned to that counter due to its higher precedence.

Even if a counter is configured by the user, in order to conserve the use of hardware resources, we should not allocate a policer entry until a real route (G, S) is programmed into the frame filtering and forwarding unit (FFU).

24.3.3 Multicast Counters CLI

IPv4 multicast counters are configured by enabling wildcard counters and changing byes and packets mode, through the process of “specific creation” and “wildcard creation.”

Command Mode

EXEC

Command Syntax

`ip multicast count [bytes | packets]`

- Enables wildcard counters. Also used to change bytes / packets mode.

`ip multicast count` *<G> <S>*

- This is only takes affect when **`ip multicast count`** is enabled. If *<G, S>* is specified, a counter will be created for this route only. We call creation of counters in this way “specific creation.”

`clear ip multicast count` *<G> <S>*

- This is only takes affect when **ip multicast count** is enabled. If <*G, S*> is specified, a counter will be cleared for this route only.

show ip mfib <*G*> *count*

- Total ***count*** is the sum of counts from all sources in that group. The count value can be **N/A** if an mroute does not have an associated counter. If the count value for any source in a **<*G*>** is N/A, then the total counts for **<*G*>** will be shown as N/A. However, the count values for other sources are still shown.

no ip multicast count

- No IP multicast count. Deletes all multicast counters, including explicit G S routes.

no ip multicast count <*G*> <*S*>

- Removes the config. Does not delete the counter because the wildcard is still active. If no <*G, S*> is specified, all multicast routes will have counters unless hardware runs out of resource. We call creation of counters in this way “wildcard creation.”

Parameters

- ***bytes* | *packets***: specifies the bytes / packets mode.

24.3.3.1 IPv4 Multicast Counter CLI Steps

IPv4 multicast counters are configured by performing the following platform independent CLI steps:

Step 1 Execute the global configuration command:

- ***no* | *default* ip multicast count *bytes* | *packets***

  Enables wildcard counters. Also used to change bytes / packets mode. When hardware runs of resources, specific creation has priority to preempt counters from wildcard creation. The ***bytes* | *packets*** optional keyword enables the counter to be in either bytes mode or packets mode. This mode applies to all counters. When the counter mode changes, all counter values will be reset to zero.

- ***no* | *default* ip multicast count <*G*> <*S*>**

  This is only takes affect when **ip multicast count** is enabled. Either **<*G, S*>** or ***bytes* | *packets*** optional keyword is used. They can not be used concurrently.

  No | default Commands: (default is same as no)

- no ip multicast count
  – Deletes all multicast counters, including explicit <*G*> <*S*> routes
- no ip multicast count <*G*> <*S*>
  – Removes the config. Does not delete the counter because the wildcard is still active.
  - If no <*G, S*> is specified, all multicast routes will have counters unless the hardware runs out of resources. The creation of counters is referred to as “wildcard creation.”
  - If <*G, S*> is specified, only <*G, S*> will get a counter (and no other route). The creation of counters is referred to as “specific creation.” By default, all mcast routes will have counters allocated. This <*G, S*> configuration is applicable when the hardware runs out of resources. Specific <*G, S*> creation has priority to preempt counters from wildcard creation.

  The *byte | frame* optional keyword enables the counter to be in either byte mode or frame mode. This mode applies to all counters. When the counter mode changes, all counter values will be reset to zero.

Either <*G, S*> or *byte | frame* optional keywords are used and cannot be used together. All counters are *byte | frame*. The *byte | frame* mode is global, and not applicable on a <*G, S*> basis.

Step 2 Execute clear command:

- **clear ip multicast count <*G*> <*S*>**

Step 3 Execute show command:

- **show ip mfib <*G*> *count***

  This command currently exists but does not show anything.

  This show command is intended to display the following (example):

```
waltartr15(config)#sho ip mfib c
Activity poll time: 60 seconds
225.1.1.1 100.0.0.2
Byte: 123
Vlan100 (iif)
Vlan200
Activity 0:00:47 ago
```

Total counts is the sum of counts from all sources in that group.

The count value can be N/A if a mroute does not have an associated counter.

If the count value for any source in a <*G*> is N/A, then the total counts for <*G*> will be shown as N/A. However, the count values for other sources are still shown.

24.4 Route Management

When routing is enabled, the switch discovers the best route to a packet's destination address by exchanging routing information with other devices. IP routing is disabled by default.

The following sections describes routing features that the switch supports

- Section 24.4.1: Route Redistribution
- Section 24.4.2: Equal Cost Multipath Routing (ECMP) and Load Sharing
- Section 24.4.3: Unicast Reverse Path Forwarding (uRPF)
- Section 24.4.4: Routing Tables / Virtual Routing and Forwarding (VRF)

24.4.1 Route Redistribution

Route redistribution is the advertisement, into a dynamic routing protocol's routing domain, of connected (static) routes or routes established by other routing protocols. By default, the switch advertises only routes in a routing domain that are established by the protocol that defined the domain.

Route redistribution commands specify the scope of the redistribution action. By default, all routes from a specified protocol (or all static routes) are advertised into the routing domain. Commands can also filter routes by applying a route map, which defines the subset of routes to be advertised.

24.4.2 Equal Cost Multipath Routing (ECMP) and Load Sharing

Equal cost multi-path (ECMP) is a routing strategy where traffic is forwarded over multiple paths that have equal routing metric values.

Configuring ECMP (IPv4)

All ECMP paths are assigned the same tag value; commands that change the tag value of a path also change the tag value of all paths in the ECMP route.

In a network topology using ECMP routing, hash polarization may result when all switches perform identical hash calculations. Hash polarization leads to uneven load distribution among the data paths. Hash polarization is avoided when switches use different hash seeds to perform hash calculations.

The **ip load-sharing** command provides the hash seed to an algorithm that the switch uses to distribute data streams among multiple equal-cost routes to a specified subnet.

Example

- This command sets the IPv4 load sharing hash seed to 20:

```
switch(config)#ip load-sharing fm6000 20
switch(config)#
```

Multicast Traffic Over ECMP

The switch attempts to spread outbound unicast and multicast traffic to all ECMP route paths equally. To disable the sending of multicast traffic over ECMP, use the **ip multicast multipath none** command.

Resilient ECMP (IPv4)

The default method of adding or removing next hop entries, as required by the active hashing algorithm, leads to inefficient management of the ECMP table, which can result in rerouting of packets to different next hops that breaks TCP packet flows. Resilient ECMP configures a fixed number of next hop entries in the hardware ECMP table for a specified IP address prefix.

The **ip hardware fib ecmp resilience** command specifies the maximum number of next hop addresses that the hardware ECMP table can contain for a specified IP prefix and a redundancy factor that facilitates the duplication of nexthop addresses in the table. The fixed table space for the address is the maximum number of next hops multiplied by the redundancy factor. When the table contains the maximum number of nexthop addresses, the redundancy factor specifies the number of times each address is listed in the table. When the table contains fewer than the maximum number of nexthop addresses, the table space entries are filled by additional duplication of the nexthop addresses.

Resilient ECMP is also available for IPv6 IP addresses.

Example

- This command configures a hardware ECMP table space of 24 entries for the IP address 10.14.2.2/24. A maximum of six nexthop addresses can be specified for the IP address. When the table contains six nexthop addresses, each appears in the table four times. When the table contains fewer than six nexthop addresses, each is duplicated until the 24 table entries are filled.

```
switch(config)#ip hardware fib ecmp resilience 10.14.2.2/24 capacity 6 redundancy
4
switch(config)#
```

24.4.3 Unicast Reverse Path Forwarding (uRPF)

Unicast Reverse Path Forwarding (uRPF) verifies the accessibility of source IP addresses in packets that the switch forwards. The switch drops a packet when uRPF determines that the routing table does not contain an entry with a valid path to that packet's source IP address.

IPv4 and IPv6 uRPF operate independently. uRPF is VRF aware; commands that do not specify a VRF utilize the default instance.Multicast routing is not affected by uRPF.

uRPF defines two operational modes: strict mode and loose mode.

- Strict mode: uRPF also verifies that a packet is received on the interface that its routing table entry specifies for its return packet.
- Loose mode: uRPF validation does not consider the inbound packet's ingress interface.

24.4.3.1 uRPF Operation

uRPF is configurable on interfaces. For packets arriving on a uRPF-enabled interfaces, the source IP address is verified by examining the source and destination addresses of unicast routing table entries.

uRPF requires a reconfigured routing table to support IP address verification. When uRPF is enabled for the first time, unicast routing is briefly disabled to facilitate the routing table reconfiguration. Multicast routing is not affected by the initial uRPF enabling.

A packet fails uRPF verification if the table does not contain an entry whose source or destination address matches the packet's source IP address. In strict mode, the uRPF also fails when the matching entry's outbound interface does not match the packet's ingress interface.

uRPF verification is not available for the following packets:

- DHCP (Source is 0.0.0.0 – Destination is 255.255.255.255)
- IPv6 link local (FE80::/10)
- Multicast packets

ECMP uRPF

When verifying ECMP routes, strict mode checks all possible paths to determine that a packet is received on the correct interface. Strict mode is supported for ECMP groups with a maximum of eight routing table entries. The switch reverts to loose mode for ECMP groups that exceed eight entries.

Default Routes

uRPF strict mode provides an ***allow-default*** option that accepts default routes. On interfaces that enable allow-default and a default route is defined, uRPF strict mode validates a packet even when the routing table does not contain an entry that matches the packet's source IP address. When allow-default is not enabled, uRPF does not consider the default route when verifying an inbound packet.

Null Routes

NULL0 routes drop traffic destined to a specified prefix. When uRPF is enabled, traffic originating from a null route prefixes is dropped in strict and loose modes.

24.4.3.2 uRPF Configuration

Unicast Reverse Path Forwarding (uRPF) is enabled for IPv4 packets ingressing the configuration mode interface through the **ip verify** command.

Note uRPF cannot be enabled on interfaces with ECMP member FECs.

Example

- This command enables uRPF loose mode on VLAN interface 17.

```
switch(config)#interface vlan 17
switch(config-if-Vl17)#ip verify unicast source reachable-via any
switch(config-if-Vl17)#show active
 interface Vlan17
   ip verify unicast source reachable-via any
switch(config-if-Vl17)#
```

- This command enables uRPF strict mode on VLAN interface 18.

```
switch(config)#interface vlan 18
switch(config-if-Vl18)#ip verify unicast source reachable-via rx
switch(config-if-Vl18)#show active
 interface Vlan18
   ip verify unicast source reachable-via rx
switch(config-if-Vl18)#
```

24.4.4 Routing Tables / Virtual Routing and Forwarding (VRF)

An IP routing table is a data table that lists the routes to network destinations and metrics (distances) associated with those routes. A routing table is also known as a routing information base (RIB).

Virtual Routing and Forwarding (VRF) allows traffic separation by maintaining multiple routing tables. Arista switches support multiple VRF instances: one global or default VRF called "default" and multiple user-defined VRFs; the number of user-defined VRFs supported varies by platform. VRFs can be used as management or data plane VRFs.

- Management VRFs have routing disabled. They are typically used for management-related traffic.
- Dataplane VRFs have routing enabled. They support routing protocols and packet forwarding (hardware and software).

Dataplane VRFs are supported by Trident, FM6000, and Arad platform switches.

VRFs support unicast IPv4 traffic; multicast and IPv6 traffic is not supported on L3 interfaces configured into a VRF. Loopback, SVI, and routed ports may be added to VRFs. Management ports may be added without any hardware forwarding.

To allow overlap in the sets of IP addresses used by different VRF instances, a route distinguisher (RD) is prepended to each address. RDs are defined in RFC 4364.

24.4.4.1 Default VRF

The default VRF on Arista switches is called “default.” It is created automatically and cannot be renamed or configured. Some configuration options accept “default” as a VRF input.

24.4.4.2 User-Defined VRFs

A user-defined VRF is created with the **vrf definition** command. After its creation, a VRF is activated by assigning it a route distinguisher with the **rd (VRF configuration mode)** command.

Example

- These commands create a VRF named “purple,” place the switch in VRF configuration mode for that VRF, and specify a route distinguisher for the VRF identifying the administrator as AS 530 and assigning 12 as its local number.

```
switch(config)#vrf definition purple
switch(config-vrf-purple)#rd 530:12
switch(config-vrf-purple)#
```

To add interfaces to a user-defined VRF, enter configuration mode for the interface and use the **vrf forwarding** command. Loopback, SVI, and routed ports can be added to a VRF.

Example

- These commands add VLAN 20 to the VRF named “purple.”

```
switch(config)#interface VLAN 20
switch(config-if-Vl20)#vrf forwarding purple
switch(config-if-Vl20)#
```

The **show vrf** command shows information about user-defined VRFs on the switch.

Example

- This command displays information for the VRF named “purple”.

```
switch>show vrf purple
   Vrf         RD             Protocols      State        Interfaces
---------- -------------- -------------- -------------- ------------
   purple      64496:237      ipv4           no routing   Vlan42, Vlan43

switch>
```

24.4.4.3 Context-Active VRF

The context-active VRF specifies the default VRF that VRF-context aware commands use when displaying or refreshing routing table data.

VRF-context aware commands include:

- **clear arp-cache**
- **copy**

- **install source**
- **ping**
- **show ip**
- **show ip arp**
- **show ip route**
- **show ip route gateway**
- **show ip route host**
- **tcpdump**
- **telnet**
- **traceroute**

The **routing-context vrf** command specifies the context-active VRF.

Example

- This command specifies ***magenta*** as the context-active VRF.

```
switch#routing-context vrf magenta
switch#show routing-context vrf
Current VRF routing-context is magenta
```

The **show routing-context vrf** command displays the context-active VRF.

Example

- This command displays the context-active VRF.

```
switch>show routing-context vrf
Current VRF routing-context is magenta
switch>
```

24.5 IPv4 Route Scale

IPv4 routes are optimized to achieve route scale when route distribution has a large number of routes of one or two parameters, with each parameter consisting of prefix lengths 12, 16, 20, 24, 28, and 32. If two separate prefix lengths are configured (in any order), one of them must be the prefix length of 32.

The following sections describes IPv4 route scale configuration, show commands, and syslog messages:

- Section 24.5.1: Configuring IPv4 Route Scale
- Section 24.5.2: Show Commands
- Section 24.5.3: Syslog

24.5.1 Configuring IPv4 Route Scale

IPv4 route scale is enabled by the **ip hardware fib optimize** command for the configuration mode interface. The platform layer 3 agent is restarted to ensure IPv4 routes are optimized with the **agent SandL3Unicast terminate** command for the configuration mode interface.

Example

- This configuration command allows configuring prefix lengths 12 and 32.

```
switch(config)#ip hardware fib optimize exact-match prefix-length 12 32
! Please restart layer 3 forwarding agent to ensure IPv4 routes are optimized
```

One of the two prefixes in this command is a prefix-length of 32, which is required in the instance where there are two prefixes. For this command to take effect, the platform layer 3 agent must be restarted.

This configuration command restarts the platform layer 3 agent to ensure IPv4 routes are optimized.

```
switch(config)#agent SandL3Unicast terminate
SandL3Unicast was terminated
```

Restarting the platform layer 3 agent results in deletion of all IPv4 routes, which are re-added to the hardware.

Example

- This configuration command allows configuring prefix lengths 32 and 16.

```
switch(config)#ip hardware fib optimize exact-match prefix-length 32 16
! Please restart layer 3 forwarding agent to ensure IPv4 routes are optimized
```

One of the two prefixes in this command is a prefix-length of 32, which is required in the instance where there are two prefixes. For this command to take effect, the platform layer 3 agent must be restarted.

This configuration command restarts the platform layer 3 agent to ensure IPv4 routes are optimized.

```
switch(config)#agent SandL3Unicast terminate
SandL3Unicast was terminated
```

Restarting the platform layer 3 agent results in deletion of all IPv4 routes, which are re-added to the hardware.

Example

- This configuration command allows configuring prefix length 24.

```
switch(config)#ip hardware fib optimize exact-match prefix-length 24
! Please restart layer 3 forwarding agent to ensure IPv4 routes are optimized
```

In this instance, there is only one prefix-length, so a prefix-length of 32 is not required. For this command to take effect, the platform layer 3 agent must be restarted.

This configuration command restarts the platform layer 3 agent to ensure IPv4 routes are optimized.

```
switch(config)#agent SandL3Unicast terminate
SandL3Unicast was terminated
```

Restarting the platform layer 3 agent results in deletion of all IPv4 routes, which are re-added to the hardware.

Example

- This configuration command allows configuring prefix length 32.

```
switch(config)#ip hardware fib optimize exact-match prefix-length 32
! Please restart layer 3 forwarding agent to ensure IPv4 routes are optimized
```

For this command to take effect, the platform layer 3 agent must be restarted.

This configuration command restarts the platform layer 3 agent to ensure IPv4 routes are optimized.

```
switch(config)#agent SandL3Unicast terminate
SandL3Unicast was terminated
```

Restarting the platform layer 3 agent results in deletion of all IPv4 routes, which are re-added to the hardware.

Example

- This configuration command disables configuring prefix lengths 12 and 32.

```
switch(config)#no ip hardware fib optimize exact-match prefix-length 12 32
! Please restart layer 3 forwarding agent to ensure IPv4 routes are not optimized
```

One of the two prefixes in this command is a prefix-length of 32, which is required in the instance where there are two prefixes. For this command to take effect, the platform layer 3 agent must be restarted.

This configuration command restarts the platform layer 3 agent to ensure IPv4 routes are not optimized.

```
switch(config)#agent SandL3Unicast terminate
SandL3Unicast was terminated
```

Restarting the platform layer 3 agent results in deletion of all IPv4 routes, which are re-added to the hardware.

Example

- This configuration command attempts to configure prefix length 20 and 28 which triggers an error exception. One of the two prefixes in this command must be a prefix-length of 32, which is required in the instance where there are two prefixes.

```
switch(config)#ip hardware fib optimize exact-match prefix-length 20 28
% One of the prefix lengths must be 32
```

24.5.2 Show Commands

The IPv4 route scale summary is displayed by the **show platform arad ip route summary** command for the configuration mode interface. Resources for all IPv4 route scale routes are displayed by the **show platform arad ip route** command for the configuration mode interface.

Example

This command shows hardware resource usage of IPv4 routes.

```
switch(config)#show platform arad ip route summary
Total number of VRFs: 1
Total number of routes: 25
Total number of route-paths: 21
Total number of lem-routes: 4
```

Example

This command shows resources for all IPv4 routes in hardware. Routes that use the additional hardware resources will appear with an asterisk.

```
switch(config)#show platform arad ip route
Tunnel Type: M(mpls), G(gre)
* - Routes in LEM

-------------------------------------------------------------------------------
-----------------
|                        Routing Table                                |          |
|-----------------------------------------------------------------------------
-----------------
|VRF|  Destination   |     |                  |    |Acl  |                |ECMP
| FEC | Tunnel
|ID |    Subnet      | Cmd |    Destination    |VID |Label| MAC / CPU Code
|Index|Index|T Value

-------------------------------------------------------------------------------
-----------------
|0  |0.0.0.0/8      |TRAP |CoppSystemL3DstMiss|0   | -   |ArpTrap         |  -
|1030 |   -
|0  |100.1.0.0/32   |TRAP |CoppSystemIpBcast  |0   | -   |BcastReceive    |  -
|1032 |   -
|0  |100.1.0.0/32   |TRAP |CoppSystemIpUcast  |0   | -   |Receive         |  -
|32766|   -
|0  |100.1.255.255/32|TRAP |CoppSystemIpBcast  |0   | -   |BcastReceive    |  -
|1032 |   -
|0  |200.1.255.255/32|TRAP |CoppSystemIpBcast  |0   | -   |BcastReceive    |  -
|1032 |   -
|0  |200.1.0.0/16   |TRAP |CoppSystemL3DstMiss|1007| -   |ArpTrap         |  -
|1029 |   -
|0  |0.0.0.0/0      |TRAP |CoppSystemL3LpmOver|0   | -   |SlowReceive     |  -
|1024 |   -
|0  |4.4.4.0/24*    |ROUTE|Et10               |1007| -   |00:01:00:02:00:03|  -
|1033 |   -
|0  |10.20.30.0/24* |ROUTE|Et9                |1006| -   |00:01:00:02:00:03|  -
|1027 |   -
```

24.5.3 Syslog

When the number of routes exceed additional hardware resources, the `ROUTING_LEM_RESOURCE_FULL` syslog message is displayed.

24.6 IP Source Guard

IP Source Guard (IPSG) prevents IP spoofing attacks. It filters inbound IP packets based on their source MAC and IP addresses. IPSG is supported in hardware. IPSG enabled on a Layer 2 port verifies IP packets received on this port. Packets are permitted if each packet source MAC and IP addresses match any of the user-configured IP-MAC binding entries on the receiving VLAN and port. Packets with no match are dropped immediately.

24.6.1 Configuring IPSG

IPSG is applicable only to Layer 2 ports, and is enabled by the **ip verify source** command for the configuration mode interface. When configured on Layer 3 ports, IPSG does not take effect until this interface is converted to Layer 2.

IPSG is supported on Layer 2 Port-Channels, not member ports. The IPSG configuration on port channels supersedes the configuration on the physical member ports. Hence, source IP MAC binding entries should be configured on port channels using the **ip source binding** command. When configured on a port channel member port, IPSG does not take effect until this port is deleted from the port channel configuration.

Example

- These configuration commands exclude VLAN IDs 1 through 3 from IPSG filtering. When enabled on a trunk port, IPSG filters the inbound IP packets on all allowed VLANs. IP packets received on VLANs 4 through 10 on Ethernet 36 will be filtered by IPSG, while those received on VLANs 1 through 3 are permitted.

```
switch(config)#no ip verify source vlan 1-3
switch(config)#interface ethernet 36
switch(config-if-Et36)#switchport mode trunk
switch(config-if-Et36)#switchport trunk allowed vlan 1-10
switch(config-if-Et36)#ip verify source
switch(config-if-Et36)#
```

  This configuration command configures source IP-MAC binding entries to IP address 10.1.1.1, MAC address 0000.aaaa.1111, VLAN ID 4094, and Ethernet interface 36.

```
switch(config)#ip source binding 10.1.1.1 0000.aaaa.1111 vlan 4094 interface
ethernet 36
switch(config)#
```

24.6.2 Show Commands

The IPSG configuration and operational states and IP-MAC binding entries are displayed by the **show ip verify source** command for the configuration mode interface.

Example

This command verifies the IPSG configuration and operational states.

```
switch(config)#show ip verify source
Interface       Operational State
--------------- ------------------------
Ethernet1       IP source guard enabled
Ethernet2       IP source guard disabled
```

Example

This command displays all VLANs configured in **no ip verify source vlan**. Hardware programming errors, e.g.,VLAN classification failed, are indicated in the operational state. If an error occurs, this VLAN will be considered as enabled for IPSG. Traffic on this VLAN will still be filtered by IPSG.

```
switch(config)#show ip verify source vlan
IPSG disabled on VLANS: 1-2
VLAN            Operational State
--------------- ------------------------
1               IP source guard disabled
2               Error: vlan classification failed
```

Example

This command displays all source IP-MAC binding entries configured for IPSG. A source binding entry is considered active if it is programmed in hardware. IP traffic matching any active binding entry will be permitted. If a source binding entry is configured on an interface or a VLAN whose operational state is IPSG disabled, this entry will not be installed in the hardware, in which case an “IP source guard disabled” state will be shown. If a port channel has no member port configured, binding entries configured for this port channel will not be installed in hardware, and a “Port-Channel down” state will be shown.

```
switch(config)#show ip verify source detail
Interface       IP Address    MAC Address      VLAN   State
--------------- ------------ ---------------- ------ ------------------------
Ethernet1       10.1.1.1      0000.aaaa.1111   5      active
Ethernet1       10.1.1.5      0000.aaaa.5555   1      IP source guard disabled
Port-Channel1   20.1.1.1      0000.bbbb.1111   4      Port-Channel down
```

24.7 DHCP Relay Across VRF

The EOS DHCP relay agent supports forwarding of DHCP requests to DHCP servers located in a different VRF to the DHCP client interface VRF. In order to enable VRF support for the DHCP relay agent, Option 82 (DHCP Relay Agent Information Option) must first be enabled. The DHCP relay agent uses Option 82 to pass client specific information to the DHCP server.

These sections describe DHCP Relay across VRF features:

- Section 24.7.1: Global Configuration
- Section 24.7.2: Show Commands

The DHCP relay agent inserts Option 82 information into the DHCP forwarded request, which requires the DHCP server belongs to a network on an interface, and that interface belongs to a different VRF than the DHCP client interface. Option 82 information includes the following:

- **VPN identifier**: The VRF name for the ingress interface of the DHCP request, inserted as sub-option 151.

SubOpt	Len	ASCII VRF Identifier						
151	7	V	R	F	N	A	M	E

Figure 24-1: VPN Identifier

- **Link selection**: The subnet address of the interface that receives the DHCP request, inserted as sub-option 5. When the DHCP smart relay is enabled, the link selection is filled with the subnet of the active address. The relay agent will set the Gateway IP address (gIPaddr) to its own IP address so that DHCP messages can be routed over the network to the DHCP server.

SubOpt	Len	Subnet IP Address			
5	4	A1	A2	A3	A4

Figure 24-2: Link Selection

- **Server identifier override**: The primary IP address of the interface that receives the DHCP request, inserted as sub-option 11. When the DHCP smart relay is enabled, the server identifier is filled with the active address (one of the primary or secondary addresses chosen by smart relay mechanism).

SubOpt	Len	Overriding Server Identifier Address			
11	4	B1	B2	B3	B4

Figure 24-3: Link Selection

- **VSS control suboption as suboption 152**: The DHCP server will strip out this suboption when sending the response to the relay, indicating that the DHCP server used VPN information to allocate IP address.

Note The DHCP server must be capable of handling VPN identifier information in option 82.

Direct communication between DHCP client and server may not be possible as they are in separate VRFs. The Server identifier override and Link Selection sub-options set the relay agent to act as the DHCP server, and enable all DHCP communication to flow through the relay agent.

The relay agent adds all the appropriate sub-options, and forwards all (including renew and release) request packets to the DHCP server. When the DHCP server response messages are received by the relay, Option 82 information is removed and the response is forwarded to the DHCP client in the client VRF.

24.7.1 Global Configuration

The DHCP relay agent information option is inserted in DHCP messages relayed to the DHCP server. The existing command below will turn on the attachment of VRF related tags in the relay agent information option. Determination of whether to add the tags is made if the client or server is on a default VRF.

If both the DHCP Client interface and Server interface are on the same VRF (default or non-default), then no VRF related DHCP Relay Agent information option is inserted.

Example

This command configures the DHCP relay to add option 82 information.

```
switch(config)#ip dhcp relay information option
```

Example

This command configures VRFs.

```
switch(config)#vrf definition mtxxg-vrf
switch(config)#rd 5546:5546
switch(config)#vrf definition qchyh-vrf
switch(config)#rd 218:218
```

Example

This command configures the interface connected to the DHCP client.

```
switch(config)#interface Ethernet9
switch(config)#switchport access vlan 2
switch(config)#no switchport
```

Example

This command configures the DHCP client interface in VRF mtxxg-vrf.

```
switch(config)#vrf forwarding mtxxg-vrf
switch(config)#ip address 10.10.0.1/16
```

Example

This command configures the DHCP server in the default VRF configuration.

```
switch(config)#ip helper-address 10.40.2.3 vrf default
```

If there is a DHCP server in the default VRF configured with keyword *default*, IP addresses in a different VRF will have overlapping addresses.

Example

This command configures the DHCP server in non-default VRF configuration that is configured with the VRF name:

```
switch(config)#ip helper-address 10.40.2.3 vrf qchyh-vrf
```

24.7.2 Show Commands

Example

This command displays the VRF specifier for the server:

```
rtr1#show ip dhcp relay
DHCP Relay is active
DHCP Relay Option 82 is enabled
DHCP Smart Relay is disabled
Interface: Ethernet9
Option 82 Circuit ID: Ethernet9
DHCP Smart Relay is disabled
DHCP servers: 10.40.2.3
10.40.2.3:vrf=qchyh-vrf
```

This command displays the VRF specifier for the server.

```
rtr1#show ip helper-address
DHCP Relay is active
DHCP Relay Option 82 is enabled
DHCP Smart Relay is disabled
Interface: Ethernet9
Option 82 Circuit ID: Ethernet9
DHCP Smart Relay is disabled
DHCP servers: 10.40.2.3
10.40.2.3:vrf=qchyh-vrf
```

24.8 IP NAT

Network address translation (NAT) is a router process that modifies address information of IP packets in transit. NAT is typically used to correlate address spaces between a local network and a remote, often public, network. Static NAT defines a one-to-one map between local and remote IP addresses. Static maps are configured manually through CLI commands. An interface can support multiple NAT commands, but each command must specify a unique local IP address-port location.

NAT is configured on routers that have interfaces connecting to the local networks and interfaces connecting to a remote network.

NAT is available only on FM6000 platform switches (the 7150 series).

Inside and Outside Addresses

In NAT configurations, IP addresses are placed into one of two categories: inside or outside. Inside refers to IP addresses used within the organizational network. Outside refers to addresses on an external network outside the organizational network.

24.8.1 Static IP NAT

Static NAT configurations create a one-to-one mapping and translate a particular address to another address. This type of configuration creates a permanent entry in the NAT table as long as the configuration is present, and it enables both inside and outside hosts to initiate a connection.

Static NAT options include source NAT, destination NAT, and twice NAT.

- Source NAT modifies the source address in the IP header of a packet exiting the interface, and can optionally change the source port referenced in the TCP/UDP headers.
- Destination NAT modifies the destination address in the IP header of a packet entering the interface, and can optionally change the destination port referenced in the TCP/UDP headers.
- Twice NAT modifies both the source and destination address of packets entering and exiting the interface, and can optionally change the L4 port information in the TCP/UDP headers. Twice NAT is generally used when inside network addresses overlap or otherwise conflict with outside network addresses. When a packet exits the interface, local source and destination addresses are translated to global source and destination addresses. When a packet enters the interface, global source and destination addresses are translated to local source and destination addresses.

24.8.1.1 Configuring Static NAT

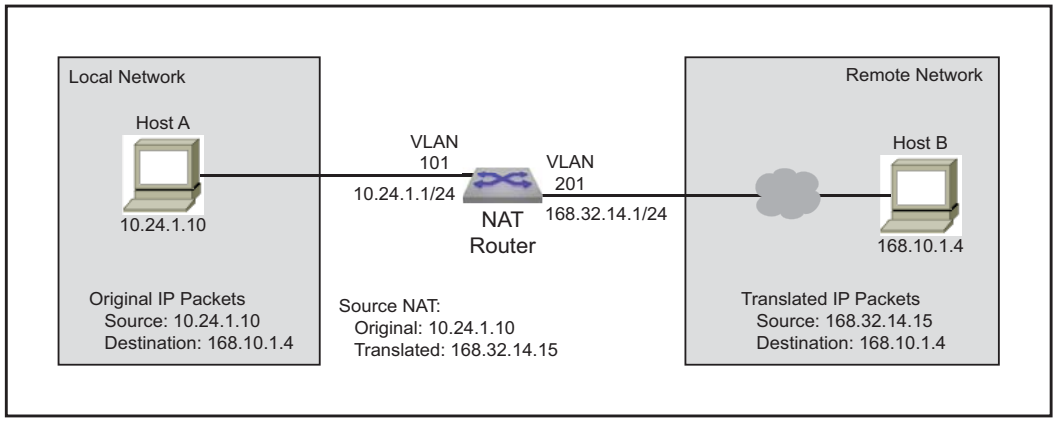
Configuring Source NAT

Network address translation of a source address (source NAT) is enabled by the **ip nat source static** command for the configuration mode interface. Applying source NAT to interfaces that connect to local hosts shields the IP address of the host when sending IP packets to remote destinations.

This command installs hardware translation entries for forward and reverse unicast traffic. When the rule specifies a multicast group, the command does not install the reverse path in hardware. The command may include an access control list to filter packets for translation.

Note The switch uses a common NAT table for the entire switch, not a per interface one. For example, if a customer has the same inside local address translated to different inside global addresses depending on which interface it exits. It might be translated to exit interface B's inside global address even though it exits through interface A. A way to avoid this is to use an access list that differentiates based on the destination IP address.

Figure 24-4: Source NAT Example



Example

- These commands configure VLAN 201 to translate source address 10.24.1.10 to 168.32.14.15.

```
switch(config)#interface vlan 201
switch(config-if-Vl201)#ip nat source static 10.24.1.10 168.32.14.15
switch(config-if-Vl201)#
```

The **ip nat source static** command may include an ACL to limit packet translation. Only packets whose source IP address matches the ACL are cleared. ACLs configured for source NAT must specify a source IP address of **any**. Source port or protocol matching is not permitted. The destination may be an IP subnet. Commands referencing nonexistent ACLs are accepted by the CLI but not installed in hardware until the ACL is created. Modifying a referenced ACL causes the corresponding hardware entries to be replaced by entries that match the new command.

Example

- These commands configure VLAN 101 to translate the source address 10.24.1.10 to 168.32.14.15 for all packets with IP destination addresses in the 168.10.1.1/32 subnet.

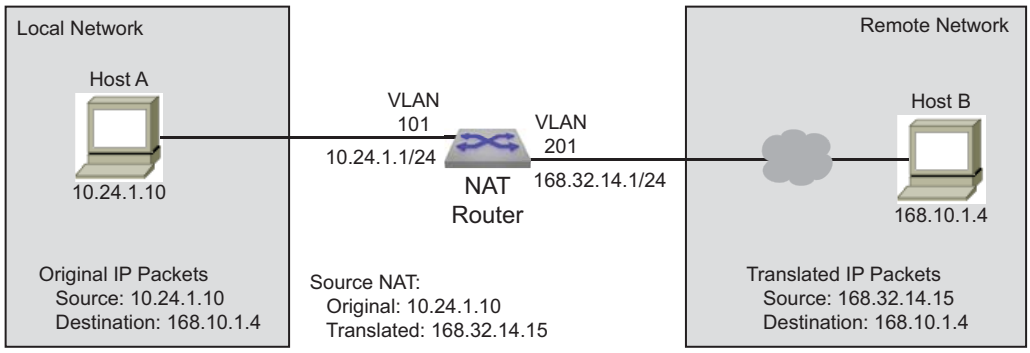
```
switch(config)#ip access-list ACL1
switch(config-acl-ACL1)#permit ip 168.10.1.0/24 any
switch(config-acl-ACL1)#exit
switch(config)#interface vlan 101
switch(config-if-Vl101)#ip nat source static 168.32.14.15 access-list ACL1
10.24.1.10
switch(config-if-Vl101)#
```

Configuring Destination NAT

Network address translation of a destination address (destination NAT) is enabled by the **ip nat destination static** command for the configuration mode interface. Applying destination NAT to interfaces that connect to remote hosts shields the IP address of the recipient host when receiving IP packets from remote destinations.

This command installs hardware translation entries for forward and reverse unicast traffic. When the rule specifies a multicast group, the command does not install the reverse path in hardware. The command may include an access control list to filter packets for translation.

Figure 24-5: Destination NAT Example



Example

- These commands configure VLAN 201 to translate destination address 168.32.14.15 to 10.24.1.10.

```
switch(config)#interface vlan 201
switch(config-if-Vl201)#ip nat destination static 168.32.14.15 10.24.1.10
switch(config-if-Vl201)#
```

The **ip nat destination static** command may include an ACL to limit packet translation. Only packets whose source IP address matches the ACL are cleared. ACLs configured for destination NAT must specify a destination IP address of **any**. Destination port or protocol matching is not permitted. The source may be an IP subnet. Commands referencing nonexistent ACLs are accepted by the CLI but not installed in hardware until the ACL is created. Modifying a referenced ACL causes the corresponding hardware entries to be replaced by entries that match the new command.

Example

- These commands configure VLAN 201 to translate the source address 10.24.1.10 to 168.32.14.15 for all packets with IP destination addresses in the 168.10.1.1/32 subnet.

```
switch(config)#ip access-list ACL2
switch(config-acl-ACL2)#permit ip 168.10.1.1/32 any
switch(config-acl-ACL2)#exit
switch(config)#interface vlan 201
switch(config-if-Vl201)#ip nat destination static 10.24.1.10 access-list ACL2
168.32.14.15
switch(config-if-Vl201)#
```

Configuring Twice NAT

Network address translation of both source and destination addresses on the same interface (twice NAT) is enabled by creating one source NAT rule and one destination NAT rule on the same interface and associating them through a NAT group using the **ip nat source static** and **ip nat destination static** commands.

The **ip nat source static** command translates the actual local source address to a source address which can be used outside the local network to reference the source. The **ip nat destination static** command translates an internally used destination address to the actual IP address that is the destination of the packet.

The source and destination NAT rules must reference the same NAT group, and both should either specify only IP addresses or specify both IP addresses and L4 port information. If L4 port information is configured in one rule but not in the other, an error message will be displayed.

Each NAT rule installs hardware translation entries for forward and reverse unicast traffic. When the rule specifies a multicast group, the command does not install the reverse path in hardware. Twice NAT does not support the use of access control lists to filter packets for translation.

Example

- These commands configure Ethernet interface 2 to translate the local source address 10.24.1.10 to the global source address 168.32.14.15, and to translate the local destination address 10.68.104.3 to the global destination address 168.25.10.7 for all packets moving through the interface. The use of NAT group 3 is arbitrary, but must be the same in both rules.

```
switch(config)#interface ethernet 2
switch(config-if-Et2)#ip nat source static 10.24.1.10 168.32.14.15 group 3
switch(config-if-Et2)#ip nat destination static 10.68.104.3 168.25.10.7 group 3
```

24.8.1.2 Static NAT Configuration Considerations

Egress VLAN filter for static NAT

When a static source NAT is configured on an interface, the source IP translation happens only for those packets that is going 'out' of this interface. If a packet is egressing on an interface which does not have NAT configured, then the source IP is not translated.

When there are two interfaces on which static SNAT is configured then the translation specified for one interface can be applied to a packet going out on the other interface.

Example

- In this example, the packets with source IP 20.1.1.1 going out of E1 will still have the source IP translated to 172.1.1.1 even though the rule is configured in E2 and not on E1.

```
switch(config)#interface ethernet 1
switch(config-if-Et1)# ip nat source static 10.1.1.1 171.1.1.1
switch(config)#interface ethernet 2
switch(config-if-Et2)#ip nat source static 20.1.1.1 172.1.1.1
```

  To prevent this, use an ACL to filter the traffic that needs NAT on the interfaces.

```
switch(config)#ip access-list acl1
switch(config-acl-acl1)#permit ip any 171.1.1.0/24
switch(config)#ip access-list acl2
switch(config-acl-acl2)#permit ip any 172.1.1.0/24
switch(config)#interface ethernet 1
switch(config-if-Et1)# ip nat source static 10.1.1.1 access-list acl1 171.1.1.1
switch(config)#interface ethernet 2
switch(config-if-Et2)#ip nat source static 20.1.1.1 access-list acl2 172.1.1.1
```

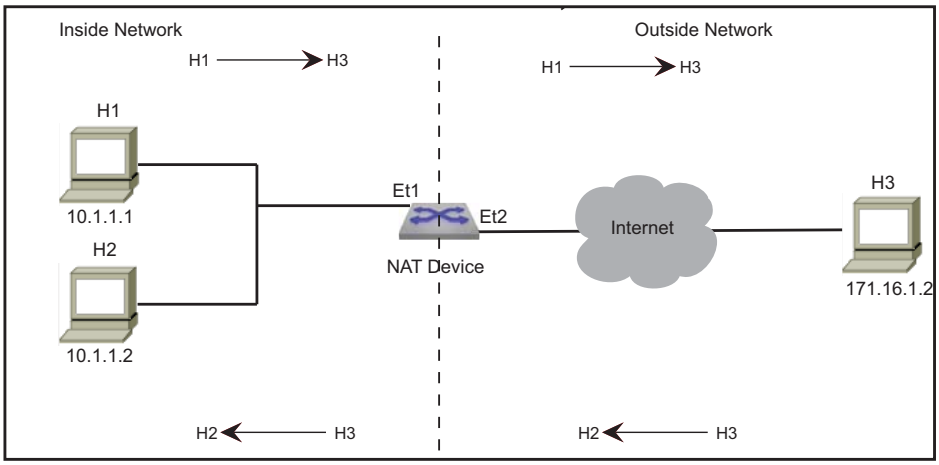
ACL filtering is not supported when using twice NAT.

24.8.2 Dynamic NAT

Dynamic NAT can be used when fewer addresses are accessible than the number of hosts to be translated. A NAT table entry is created when the host starts a connection and establishes a one-to-one mapping between addresses. The mapping can vary and is dependent upon the registered addresses in the pool at the time of the communication. Dynamic NAT sessions are only allowed to be initiated only from inside networks. NAT should be configured on a Layer 3 interface, either a routed port or

Switch Virtual Interface (SVI). If the host doesn't communicate for a specific period, dynamic NAT entries are removed from the translation table. The address will then returned to the pool for use by another host

Figure 24-6: Dynamic NAT Scenario



Dynamic NAT options:

- Many-to-Many NAT

  Maps local addresses to a global address that is selected from a pool of global addresses. After pool is configured, the first available address from the pool is picked dynamically on receiving the first packet.

- Many-to-One NAT (PAT)

  PAT is a form of dynamic NAT where multiple local addresses are mapped to a single global address (many-to-one) using different source ports. This method is also called NAT Overloading, NAPT (Network and Port address translation), and Masquerade. The global address can be the IP address configured on the outside interface.

  Hardware entries that translate packets are created when the CLI command is processed. Entries for forward and reverse traffic are created for unicast traffic. The hardware entry for reverse traffic is not created for multicast traffic.

  Commands may include ACLs to filter packets that are cleared. Source NAT use ACLs to filter packets based on destination IP address. Destination NAT use ACLs to filter packets based on source IP address. Upon using NAT, inside usually refers to a private network while outside usually refers to a public network.

  A switch with NAT configured translates forwarded traffic between inside and outside interfaces, and the flow that matches the criteria specified for translation.

  The same IP address can't be used for the NAT static configuration and in the pool for dynamic NAT configurations. Public IP addresses must be unique. The global addresses used in static translations aren't excluded with dynamic pools containing the same global addresses.

Hardware entries that translate packets are created when the CLI command is processed. Entries for forward and reverse traffic are created for unicast traffic. The hardware entry for reverse traffic is not created for multicast traffic.

Commands may include ACLs to filter packets that are cleared. Source NAT use ACLs to filter packets based on destination IP address. Destination NAT use ACLs to filter packets based on source IP address. When using NAT, inside usually refers to a private network while outside usually refers to a public network.

A switch with NAT configured translates forwarded traffic between inside and outside interfaces, and the flow that matches the criteria specified for translation.

Important! The same IP address can't be used for the NAT static configuration and in the pool for dynamic NAT configurations. Public IP addresses must be unique. The global addresses used in static translations aren't excluded with dynamic pools containing the same global addresses.

24.8.2.1 Configuring Dynamic NAT

Prerequisites

- Configure an ACL to specify IP addresses allowed to be translated.
- Determine if you should use an IP address as the translated source address.
- Decide on a public IP address pool for address translation.

Configure the Address Pool

The addresses used for translation are configured by issuing the **ip nat pool** command in global configuration mode.

Example

- This command configures the pool of addresses using start address, and end address.

```
switch(config)#ip nat pool p1 10.15.15.15 10.15.15.25
switch(config)#
```

Set the IP Address

The **ip address** command configures VLAN 201 with an IP address.

Example

- This command configures an IPv4 address for VLAN 201.

```
switch(config)#interface vlan 201
switch(config-if-Vl201)#ip address 10.0.0.1/24
switch(config-if-Vl201)#
```

- This command configures the dynamic NAT source address and sets the NAT overload for pool P2.

```
switch(config-if-Vl201)#ip nat source dynamic access-list ACL2 pool p2
switch(config-if-Vl201)#
```

Define the NAT Source Address for Translation

The **ip nat source dynamic** command specifies a dynamic translation from the source IP address to the pool and to overload the pool address (or addresses).

Example

- This command configures the dynamic NAT source address and sets the pool P2 NAT overload.

```
switch(config)#interface ethernet 3/1
switch(config-if-Et3/1)#ip nat source dynamic access-list ACL2 pool p2 overload
switch(config-if-Et3/1)#
```

Specify the Timeout Values

The **ip nat translation tcp-timeout** or **ip nat translation udp-timeout** commands alter the translation timeout period for NAT translation table entries.

Example

- This command globally sets the timeout for TCP to 600 seconds.

```
switch(config)# ip nat translation tcp-timeout 600
switch(config)#
```

- This command globally sets the timeout for UDP to 800 seconds.

```
switch(config)# ip nat translation udp-timeout 800
switch(config)#
```

24.8.2.2 Verify the NAT Configuration

Display the Address Pools

The **show ip nat pool** command displays the configuration of the address pool.

Example

- This command displays all the address pools configured on the switch.

```
switch#show ip nat pool
Pool                StartIp              EndIp                Prefix
p1                  10.15.15.15          10.15.15.25          24
p2                  10.10.15.15          10.10.15.25          22
p3                  10.12.15.15          10.12.15.25          12
switch#
```

24.8.2.3 Clearing IP NAT Translations

Use the **clear ip nat translation** command to remove all or the specified NAT table entries.

Example

- This command clears all dynamic entries from the NAT translation table

```
switch#clear ip nat translation
switch#
```

24.8.2.4 Dynamic NAT Configuration Considerations

Configuring Dynamic NAT Using Pools in a L2 Adjacent Network

When many-to-one dynamic NAT is configured using a NAT pool, and the next hop router for the NAT device is on the same network (L2 adjacent), then you must configure the IP addresses in the NAT pool as a secondary address on the interface.

Example

- The IP addresses in the NAT pool are configured as the secondary address on the interface.

```
switch(config)#ip nat pool p1 10.1.1.1 10.1.1.4 prefix-length 24
switch(config)#interface ethernet 1
switch(config-if-Et1)#ip nat source dynamic access-list a1 pool p1
switch(config-if-Et1)#ip address 10.1.1.1/24 secondary
switch(config-if-Et1)#ip address 10.1.1.2/24 secondary
switch(config-if-Et1)#ip address 10.1.1.3/24 secondary
switch(config-if-Et1)#ip address 10.1.1.4/24 secondary
```

Configuring Dynamic NAT Using Pool in a L3 Network

If the next hop of the NAT device is on a different subnet, then you should configure a static Null route for the IP addresses in the NAT pool. Redistribute the static route using BGP/OSPF.

Example

- Outside Interface

```
switch(config)#interface port-channel 319
switch(config-if-Po319)#ip nat source dynamic access-list dynamic-nat-m2m pool
natpl-dynamic-nat-m2m
switch(config)#ip access-list dynamic-nat-m2m
switch(config-acl-dynamic-nat-m2m)#10 permit ip 192.168.93.0/24 any
switch(config)#ip nat pool natpl-dynamic-nat-m2m prefix-length 24
switch(config-natpool-p1)#range 11.3.3.2 11.3.3.10
```

- Static Null Route for Virtual IP

```
switch(config)#ip route 11.0.0.0/8 Null0
switch(config)#router ospf 1
switch(config-router-ospf)#redistribute static
```

Configuring Dynamic NAT Using Overload with ECMP Routes

Dynamic many-to-one NAT using overload (PAT) should not be configured on interfaces that form an ECMP group. When one interface in the group goes down, the return packet for connections that are already established will continue to go to the IP address of the interface that went down and will not be forwarded to the inside host. For this type of scenario, use Dynamic Nat with pool configurations.

24.9 IPv4 Command Descriptions

IP Routing and Address Commands

- agent SandL3Unicast terminate
- clear ip arp inspection statistics
- ip address
- ip arp inspection limit
- ip arp inspection logging
- ip arp inspection trust
- ip arp inspection vlan
- ip hardware fib ecmp resilience
- ip hardware fib optimize
- ip icmp redirect
- ip load-sharing
- ip route
- ip routing
- ip source binding
- ip verify
- ip verify source
- show ip
- show ip arp inspection vlan
- show ip arp inspection statistics
- show ip interface
- show ip interface brief
- show ip route
- show ip route age
- show ip route gateway
- show ip route host
- show ip route summary
- show ip route tag
- show ip verify source
- show platform arad ip route
- show platform arad ip route summary

IPv4 DHCP Relay

- clear ip dhcp relay counters
- ip dhcp relay always-on
- ip dhcp relay information option (Global)
- ip dhcp relay information option circuit-id
- ip dhcp smart-relay
- ip dhcp smart-relay global
- ip helper-address
- show ip dhcp relay
- show ip dhcp relay counters
- show ip helper-address

IPv4 DHCP Snooping

- clear ip dhcp snooping counters
- ip dhcp snooping
- ip dhcp snooping information option

- ip dhcp snooping vlan
- show ip dhcp snooping
- show ip dhcp snooping counters
- show ip dhcp snooping hardware

IPv4 Multicast Counters

-

IPv4 NAT

- clear ip nat translation
- ip nat destination static
- ip nat pool
- ip nat source dynamic
- ip nat source static
- ip nat translation low-mark
- ip nat translation max-entries
- ip nat translation tcp-timeout
- ip nat translation udp-timeout
- show ip nat access-list interface
- show ip nat pool
- show ip nat translations

ARP Table

- arp
- arp cache persistent
- arp timeout
- clear arp-cache
- clear ip arp
- ip local-proxy-arp
- ip proxy-arp
- show arp
- show ip arp

VRF Commands

- description (VRF)
- rd (VRF configuration mode)
- routing-context vrf
- show routing-context vrf
- show vrf
- vrf definition
- vrf forwarding

Trident Forwarding Table Commands

- platform trident forwarding-table partition
- platform trident routing-table partition
- show platform trident forwarding-table partition

agent SandL3Unicast terminate

The **agent SandL3Unicast terminate** command restarts the platform layer 3 agent to ensure IPv4 routes are optimized.

Command Mode

Global Configuration

Command Syntax

```
agent SandL3Unicast terminate
```

Related Commands

- **ip hardware fib optimize** enables IPv4 route scale.
- **show platform arad ip route** shows resources for all IPv4 routes in hardware. Routes that use the additional hardware resources will appear with an asterisk.
- **show platform arad ip route summary** shows hardware resource usage of IPv4 routes.

Example

- This configuration command restarts the platform layer 3 agent to ensure IPv4 routes are optimized.

  ```
  switch(config)#agent SandL3Unicast terminate
  SandL3Unicast was terminated
  ```

  Restarting the platform layer 3 agent results in deletion of all IPv4 routes, which are re-added to the hardware.

arp

The **arp** command adds a static entry to an Address Resolution Protocol (ARP) cache. The switch uses ARP cache entries to correlate 32-bit IP addresses to 48-bit hardware addresses.

The **no arp** and **default arp** commands remove the ARP cache entry with the specified IP address. When multiple VRFs contain ARP cache entries for identical IP addresses, each entry can only be removed individually.

Command Mode

Global Configuration

Command Syntax

```
arp [VRF_INSTANCE] ipv4_addr mac_addr arpa
no arp [VRF_INSTANCE] ipv4_addr
default arp [VRF_INSTANCE] ipv4_addr
```

Parameters

- ***VRF_INSTANCE*** specifies the VRF instance being modified.
  - <no parameter> changes are made to the default VRF.
  - **vrf** *vrf_name* changes are made to the specified user-defined VRF.
- *ipv4_addr* IPv4 address of ARP entry.
- *mac_addr* local data-link (hardware) address (48-bit dotted hex notation – H.H.H).

Examples

- This command adds a static entry to the ARP cache in the default VRF.

```
switch(config)#arp 172.22.30.52 0025.900e.c63c arpa
switch(config)#
```

- This command adds the same static entry to the ARP cache in the VRF named “purple.”

```
switch(config)#arp vrf purple 172.22.30.52 0025.900e.c63c arpa
switch(config)#
```

arp cache persistent

The **arp cache persistent** command restores the dynamic entries in the Address Resolution Protocol (ARP) cache after reboot.

The **no arp cache persistent** and **default arp cache persistent** commands remove the ARP cache persistent configuration from the *running-config*.

Command Mode

Global Configuration

Command Syntax

```
arp cache persistent
no arp cache persistent
default arp cache persistent
```

Example

- This command restores the ARP cache after reboot.

```
switch(config)#arp cache persistent
switch(config)#
```

arp timeout

The **arp timeout** command specifies the duration of dynamic address entries in the Address Resolution Protocol (ARP) cache for addresses learned through the configuration mode interface. The default duration is 14400 seconds (four hours).

The **arp timeout** and **default arp timeout** commands restores the default ARP timeout for addresses learned on the configuration mode interface by deleting the corresponding **arp timeout** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration
Interface-Loopback Configuration
Interface-Management Configuration
Interface-Port-channel Configuration
Interface-VLAN Configuration

Command Syntax

```
arp timeout arp_time
no arp timeout
default arp timeout
```

Parameters

- *arp_time* ARP timeout period (seconds). Values range from 60 to 65535. Default value is 14400.

Examples

- This command specifies an ARP cache duration of 7200 seconds (two hours) for dynamic addresses added to the ARP cache that were learned through VLAN 200.

```
switch(config)#interface vlan 200
switch(config-if-Vl200)#arp timeout 7200
switch(config-if-Vl200)#show active
interface Vlan200
   arp timeout 7200
switch(config-if-Vl200)#
```

clear arp-cache

The **clear arp-cache** command refreshes dynamic entries in the Address Resolution Protocol (ARP) cache. Refreshing the ARP cache updates current ARP table entries and removes expired ARP entries not yet deleted by an internal, timer-driven process.

The command, without arguments, refreshes ARP cache entries for all enabled interfaces. With arguments, the command refreshes cache entries for the specified interface. Executing **clear arp-cache** for all interfaces can result in extremely high CPU usage while the tables are resolving.

Command Mode

Privileged EXEC

Command Syntax

clear arp-cache [*VRF_INSTANCE*][*INTERFACE_NAME*]

Parameters

- ***VRF_INSTANCE*** specifies the VRF instance for which arp data is refreshed.
  - <no parameter> specifies the context-active VRF.
  - **vrf** *vrf_name* specifies name of VRF instance. System default VRF is specified by **default**.
- ***INTERFACE_NAME*** interface upon which ARP cache entries are refreshed. Options include:
  - <no parameter> All ARP cache entries.
  - **interface ethernet** *e_num* ARP cache entries of specified Ethernet interface.
  - **interface loopback** *l_num* ARP cache entries of specified loopback interface.
  - **interface management** *m_num* ARP cache entries of specified management interface.
  - **interface port-channel** *p_num* ARP cache entries of specified port-channel Interface.
  - **interface vlan** *v_num* ARP cache entries of specified VLAN interface.
  - **interface vxlan** *vx_num* VXLAN interface specified by *vx_num*.

Related Commands

- **routing-context vrf** specifies the context-active VRF.

Example

- These commands display the ARP cache before and after ARP cache entries are refreshed.

```
switch#show arp
Address         Age (min)  Hardware Addr   Interface
172.22.30.1             0  001c.730b.1d15  Management1
172.22.30.118           0  001c.7301.6015  Management1

switch#clear arp-cache

switch#show arp
Address         Age (min)  Hardware Addr   Interface
172.22.30.1             0  001c.730b.1d15  Management1
switch#
```

clear ip arp

The **clear ip arp** command removes the specified dynamic ARP entry for the specified IP address from the Address Resolution Protocol (ARP) table.

Command Mode

Privileged EXEC

Command Syntax

clear ip arp [*VRF_INSTANCE*] *ipv4_addr*

Parameters

- ***VRF_INSTANCE*** specifies the VRF instance for which arp data is removed.
  - <no parameter> specifies the context-active VRF.
  - **vrf** *vrf_name* specifies name of VRF instance. System default VRF is specified by **default**.
- *ipv4_addr* IPv4 address of dynamic ARP entry.

Related Commands

- **routing-context vrf** specifies the context-active VRF.

Example

- These commands display the ARP table before and after the removal of dynamic ARP entry for IP address 172.22.30.52.

```
switch#show arp
Address         Age (min)  Hardware Addr   Interface
172.22.30.1             0  001c.730b.1d15  Management1
172.22.30.52            0  0025.900e.c468  Management1
172.22.30.53            0  0025.900e.c63c  Management1
172.22.30.133           0  001c.7304.3906  Management1
switch#clear ip arp 172.22.30.52
switch#show arp
Address         Age (min)  Hardware Addr   Interface
172.22.30.1             0  001c.730b.1d15  Management1
172.22.30.53            0  0025.900e.c63c  Management1
172.22.30.133           0  001c.7304.3906  Management1
switch#
```

clear ip arp inspection statistics

The **clear ip arp inspection statistics** command clears ARP inspection statistics.

Command Mode

EXEC

Command Syntax

```
clear ip arp inspection statistics
```

Related Commands

- **ip arp inspection limit**
- **ip arp inspection logging**
- **ip arp inspection trust**
- **ip arp inspection vlan**
- **show ip arp inspection vlan**
- **show ip arp inspection statistics**

Examples

- This command clears ARP inspection statistics.

```
switch(config)#clear ip arp inspection statistics
switch(config)#
```

clear ip dhcp relay counters

The **clear ip dhcp relay counters** command resets the DHCP relay counters. The configuration mode determines which counters are reset:

- Interface configuration: command clears the counter for the configuration mode interface.

Command Mode

Privileged EXEC

Command Syntax

clear ip dhcp relay counters [***INTERFACE_NAME***]

Parameters

- ***INTERFACE_NAME*** entity for which counters are cleared. Options include:
  - <no parameter> clears counters for the switch and for all interfaces.
  - **interface ethernet** *e_num* clears counters for the specified Ethernet interface.
  - **interface loopback** *l_num* clears counters for the specified loopback interface.
  - **interface port-channel** *p_num* clears counters for the specified port-channel Interface.
  - **interface vlan** *v_num* clears counters for the specified VLAN interface.

Examples

- These commands clear the DHCP relay counters for VLAN 1045 and shows the counters before and after the **clear** command.

```
switch#show ip dhcp relay counters

          |  Dhcp Packets  |
Interface | Rcvd Fwdd Drop |         Last Cleared
----------|----- ---- -----|----------------------
  All Req |  376  376    0 | 4 days, 19:55:12 ago
 All Resp |  277  277    0 |
          |                |
 Vlan1001 |  207  148    0 | 4 days, 19:54:24 ago
 Vlan1045 |  376  277    0 | 4 days, 19:54:24 ago

switch#clear ip dhcp relay counters interface vlan 1045

          |  Dhcp Packets  |
Interface | Rcvd Fwdd Drop |         Last Cleared
----------|----- ---- -----|----------------------
  All Req |  380  380    0 | 4 days, 21:19:17 ago
 All Resp |  281  281    0 |
          |                |
 Vlan1000 |  207  148    0 | 4 days, 21:18:30 ago
 Vlan1045 |    0    0    0 |          0:00:07 ago
```

- These commands clear all DHCP relay counters on the switch.

```
switch(config-if-Vl1045)#exit
switch(config)#clear ip dhcp relay counters
switch(config)#show ip dhcp relay counters

          |  Dhcp Packets  |
Interface | Rcvd Fwdd Drop | Last Cleared
----------|----- ---- -----|-------------
  All Req |    0    0    0 |  0:00:03 ago
 All Resp |    0    0    0 |
          |                |
 Vlan1000 |    0    0    0 |  0:00:03 ago
 Vlan1045 |    0    0    0 |  0:00:03 ago
```

clear ip dhcp snooping counters

The **clear ip dhcp snooping counters** command resets the DHCP snooping packet counters.

Command Mode

Privileged EXEC

Command Syntax

clear ip dhcp snooping counters [*COUNTER_TYPE*]

Parameters

- ***COUNTER_TYPE*** The type of counter that the command resets. Options include:
  - <no parameter> counters for each VLAN.
  - **debug** aggregate counters and drop cause counters.

Example

- This command clears the DHCP snooping counters for each VLAN.

```
switch#clear ip dhcp snooping counters
switch#show ip dhcp snooping counters

     | Dhcp Request Pkts | Dhcp Reply Pkts |
Vlan |  Rcvd  Fwdd  Drop | Rcvd Fwdd  Drop | Last Cleared
-----|------ ----- ------|----- ---- ------|-------------
 100 |     0     0     0 |    0    0     0 |  0:00:10 ago

switch#
```

- This command clears the aggregate DHCP snooping counters.

```
switch#clear ip dhcp snooping counters debug
switch#show ip dhcp snooping counters debug

Counter                       Snooping to Relay Relay to Snooping
----------------------------- ----------------- -----------------
Received                                      0                 0
Forwarded                                     0                 0
Dropped - Invalid VlanId                      0                 0
Dropped - Parse error                         0                 0
Dropped - Invalid Dhcp Optype                 0                 0
Dropped - Invalid Info Option                 0                 0
Dropped - Snooping disabled                   0                 0

Last Cleared:  0:00:08 ago
switch#
```

clear ip nat translation

The **clear ip nat translation** command clears all or the specified NAT table entries.

Command Mode

Privileged EXEC

Command Syntax

clear ip nat translation [*HOST_ADDR* [*DEST_ADDR*]] [*INTF*] [*PROT_TYPE*]

Parameters

DEST_ADDR immediately follows ***HOST_ADDR***. All other parameters, including ***HOST_ADDR,*** may be placed in any order.

- ***HOST_ADDR*** Host address to be modified. Options include:
  - <no parameter> All packets with specified destination address are cleared.
  - **address** *local_ipv4* IPv4 address.
  - **address** *local_ipv4* *local_port* IPv4 address and port (port value ranges from 1 to 65535).
- ***DEST_ADDR*** Destination address of translated packet. Destination address can be entered only when the ***HOST_ADDR*** is specified. Options include:
  - <no parameter> All packets with specified destination address are cleared.
  - *global_ipv4* IPv4 address.
  - *global_ipv4* *global_port* IPv4 address and port (port value ranges from 1 to 65535).
- ***INTF*** Route source. Options include:
  - <no parameter> All packets with specified destination address are cleared.
  - **interface ethernet** *e_num* Ethernet interface specified by *e_num*.
  - **interface loopback** *l_num* Loopback interface specified by *l_num*.
  - **interface management** *m_num* Management interface specified by *m_num*.
  - **interface port-channel** *p_num* Port-channel interface specified by *p_num*.
  - **interface vlan** *v_num* VLAN interface specified by *v_num*.
- ***PROT_TYPE*** Filters packets based on protocol type. Options include:
  - <no parameter> All packets with specified destination address are cleared.
  - **tcp** TCP packets with specified destination address are cleared.
  - **udp** UDP packets with specified destination address are cleared.

Example

- This command clears all dynamic entries from the NAT translation table

```
switch#clear ip nat translation
switch#
```

- This command clears a specific NAT IP address 172.22.30.52.

```
switch#clear ip nat translation address 172.22.30.52
switch#
```

- This command clears the inside entry that maps the private address 10.10.10.3 to Internet address 172.22.30.52.

```
switch#clear ip nat translation address 172.22.30.52 10.10.10.3
switch#
```

description (VRF)

The **description** command adds a text string to the configuration mode VRF. The string has no functional impact on the VRF.

The **no description** and **default description** commands remove the text string from the configuration mode VRF by deleting the corresponding **description** command from ***running-config***.

Command Mode

VRF Configuration

Command Syntax

```
description label_text
no description
default description
```

Parameters

- *label_text* character string assigned to the VRF configuration.

Related Commands

- **vrf definition** places the switch in VRF configuration mode.

Examples

- These commands add description text to the magenta VRF.

```
switch(config)#vrf definition magenta
switch(config-vrf-magenta)#description This is the first vrf
switch(config-vrf-magenta)#show active
 vrf definition magenta
   description This is the first vrf
switch(config-vrf-magenta)#
```

ip address

The **ip address** command configures the IPv4 address and connected subnet on the configuration mode interface. Each interface can have one primary address and multiple secondary addresses.

The **no ip address** and **default ip address** commands remove the IPv4 address assignment from the configuration mode interface. Entering the command without specifying an address removes the primary and all secondary addresses from the interface. The primary address cannot be deleted until all secondary addresses are removed from the interface.

Removing all IPv4 address assignments from an interface disables IPv4 processing on that port.

Command Mode

Interface-Ethernet Configuration
Interface-Loopback Configuration
Interface-Management Configuration
Interface-Port-channel Configuration
Interface-VLAN Configuration

Command Syntax

```
ip address ipv4_subnet [PRIORITY]
no ip address [ipv4_subnet] [PRIORITY]
default ip address [ipv4_subnet] [PRIORITY]
```

Parameters

- *ipv4_subnet* IPv4 and subnet address (CIDR or address-mask notation). ***Running-config*** stores value in CIDR notation.
- ***PRIORITY*** interface priority. Options include:
  - <no parameter> the address is the primary IPv4 address for the interface.
  - **secondary** the address is the secondary IPv4 address for the interface.

Guidelines

The **ip address** command is supported on routable interfaces.

Example

- This command configures an IPv4 address for VLAN 200.

```
switch(config)#interface vlan 200
switch(config-if-Vl200)#ip address 10.0.0.1/24
switch(config-if-Vl200)#
```

ip arp inspection limit

The **ip arp inspection limit** command err-disables the interface if the incoming ARP rate exceeds the configured value rate limit the incoming ARP packets on an interface.

Command Mode

EXEC

Command Syntax

[no | default] ip arp inspection limit [*RATE <pps>*] [*BURST_INTERVAL <sec>* | none]

Parameters

- ***RATE*** specifies the ARP inspection limit rate in packets per second.
  - <pps> ARP inspection limit rate packets per second.
- ***BURST_INTERVAL*** specifies the ARP inspection limit burst interval.
  - <sec> burst interval second.

Related Commands

- **ip arp inspection limit**
- **ip arp inspection trust**
- **ip arp inspection vlan**
- **show ip arp inspection vlan**

Examples

- This command configures the rate limit of incoming ARP packets to errdisable the interface when the incoming ARP rate exceeds the configured value, sets the rate to 512 (which is the upper limit for the number of invalid ARP packets allowed per second), and sets the burst consecutive interval over which the interface is monitored for a high ARP rate to 11 seconds.

```
switch(config)#ip arp inspection limit rate 512 burst interval 11
switch(config)#
```

- This command displays verification of the interface specific configuration.

```
switch(config)#interface Ethernet 3 / 1
switch(config)#ip arp inspection limit rate 20 burst interval 5
switch(config)#interface Ethernet 3 / 3
switch(config)#ip arp inspection trust
switch(config)#show ip arp inspection interfaces
 Interface      Trust State  Rate (pps) Burst Interval
 -------------  -----------  ---------- --------------
 Et3/1          Untrusted    20         5
 Et3/3          Trusted      None       N/A

switch(config)#
```

ip arp inspection logging

The **ip arp inspection logging** command enables logging of incoming ARP packets on the interface if the rate exceeds the configured value.

Command Mode

EXEC

Command Syntax

[no | default] ip arp inspection logging [*RATE <pps>*] [*BURST_INTERVAL <sec>* | none]

Parameters

- ***RATE*** specifies the ARP inspection limit rate in packets per second.
  - <pps> ARP inspection limit rate packets per second.
- ***BURST_INTERVAL*** specifies the ARP inspection limit burst interval.
  - <sec> burst interval second.

Related Commands

- **ip arp inspection limit**
- **ip arp inspection trust**
- **ip arp inspection vlan**
- **show ip arp inspection vlan**

Example

- This command enables logging of incoming ARP packets when the incoming ARP rate exceeds the configured value on the interface, sets the rate to 2048 (which is the upper limit for the number of invalid ARP packets allowed per second), and sets the burst consecutive interval over which the interface is monitored for a high ARP rate to 15 seconds.

```
switch(config)#ip arp inspection logging rate 2048 burst interval 15
switch(config)#
```

ip arp inspection trust

The **ip arp inspection trust** command configures the trust state of an interface. By default, all interfaces are untrusted.

Command Mode

EXEC

Command Syntax

```
[no | default] ip arp inspection trust
```

Related Commands

- **ip arp inspection limit**
- **ip arp inspection logging**
- **ip arp inspection vlan**
- **show ip arp inspection vlan**

Examples

- This command configures the trust state of an interface.

```
switch(config)#ip arp inspection trust
switch(config)#
```

- This command configures the trust state of an interface to untrusted.

```
switch(config)#no ip arp inspection trust
switch(config)#
```

- This command configures the trust state of an interface to its default (untrusted).

```
switch(config)#default ip arp inspection trust
switch(config)#
```

ip arp inspection vlan

The **ip arp inspection vlan** command enables ARP inspection. ARP requests and responses on untrusted interfaces are intercepted on specified VLANs, and intercepted packets are verified to have valid IP-MAC address bindings. All invalid ARP packets are dropped. On trusted interfaces, all incoming ARP packets are processed and forwarded without verification. By default, ARP inspection is disabled on all VLANs.

Command Mode

EXEC

Command Syntax

```
ip arp inspection vlan [LIST]
```

Parameters

- ***LIST*** specifies the VLAN interface number.

Related Commands

- **ip arp inspection limit**
- **ip arp inspection trust**
- **show ip arp inspection vlan**

Examples

- This command enables ARP inspection on VLANs 1 through 150.

```
switch(config)#ip arp inspection vlan 1 - 150
switch(config)#
```

- This command disables ARP inspection on VLANs 1 through 150.

```
switch(config)#no ip arp inspection vlan 1 - 150
switch(config)#
```

- This command sets the ARP inspection default to VLANs 1 through 150.

```
switch(config)#default ip arp inspection vlan 1 - 150
switch(config)#
```

- These commands enable ARP inspection on multiple VLANs 1 through 150 and 200 through 250.

```
switch(config)#ip arp inspection vlan 1-150,200-250
switch(config)#
```

ip dhcp relay always-on

The **ip dhcp relay always-on** command enables the switch DHCP relay agent on the switch regardless of the DHCP relay agent status on any interface. By default, the DHCP relay agent is enabled only if at least one routable interface is configured with an **ip helper-address** statement.

The **no ip dhcp relay always-on** and **default ip dhcp relay always-on** commands remove the **ip dhcp relay always-on** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
ip dhcp relay always-on
no ip dhcp relay always-on
default ip dhcp relay always-on
```

Related Commands

These commands implement DHCP relay agent.

- **ip helper-address**
- **ip dhcp relay information option (Global)**
- **ip dhcp relay information option circuit-id**

Example

- This command enables the DHCP relay agent.

```
switch(config)#ip dhcp relay always-on
switch(config)#
```

ip dhcp relay information option (Global)

The **ip dhcp relay information option** command configures the switch to attach tags to DHCP requests before forwarding them to the DHCP servers designated by **ip helper-address** commands. The **ip dhcp relay information option circuit-id** command specifies the tag contents for packets forwarded by the interface that it configures.

The **no ip dhcp relay information option** and **default ip dhcp relay information option** commands restore the switch's default setting of not attaching tags to DHCP requests by removing the **ip dhcp relay information option** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
ip dhcp relay information option
no ip dhcp relay information option
default ip dhcp relay information option
```

Related Commands

These commands implement DHCP relay agent.

- **ip helper-address**
- **ip dhcp relay always-on**
- **ip dhcp relay information option circuit-id**

Example

- This command enables the attachment of tags to DHCP requests that are forwarded to DHCP server addresses.

```
switch(config)#ip dhcp relay information option
switch(config)#
```

ip dhcp relay information option circuit-id

The **ip dhcp relay information option circuit-id** command specifies the content of tags that the switch attaches to DHCP requests before they are forwarded from the configuration mode interface to DHCP server addresses specified by **ip helper-address** commands. Tags are attached to outbound DHCP requests only if the information option is enabled on the switch (**ip dhcp relay information option circuit-id**. The default value for each interface is the name and number of the interface.

The **no ip dhcp relay information option circuit-id** and **default ip dhcp relay information option circuit-id** commands restore the default content setting for the configuration mode interface by removing the corresponding command from ***running-config***.

Command Mode

Interface-Ethernet Configuration
Interface-Loopback Configuration
Interface-Management Configuration
Interface-Port-channel Configuration
Interface-VLAN Configuration

Command Syntax

```
ip dhcp relay information option circuit-id id_label
no ip dhcp relay information option circuit-id
default ip dhcp relay information option circuit-id
```

Parameters

- *id_label* Tag content. Format is alphanumeric characters (maximum 15 characters).

Related Commands

- **ip helper-address**
- **ip dhcp relay always-on**
- **ip dhcp relay information option (Global)**

Example

- This command configures ***x-1234*** as the tag content for packets send from VLAN 200.

```
switch(config)#interface vlan 200
switch(config-if-Vl200)#ip dhcp relay information option circuit-id x-1234
switch(config-if-Vl200)#
```

ip dhcp smart-relay

The **ip dhcp smart-relay** command configures the DHCP smart relay status on the configuration mode interface. DHCP smart relay supports forwarding DHCP requests with a client's secondary IP addresses in the gateway address field. Enabling DHCP smart relay on an interface requires that DHCP relay is also enabled on that interface.

By default, an interface assumes the global DHCP smart relay setting as configured by the **ip dhcp smart-relay global** command. The **ip dhcp smart-relay** command, when configured, takes precedence over the global smart relay setting.

The **no ip dhcp smart-relay** command disables DHCP smart relay on the configuration mode interface. The **default ip dhcp smart-relay** command restores the interface's to the default DHCP smart relay setting, as configured by the **ip dhcp smart-relay global** command, by removing the corresponding **ip dhcp smart-relay** or **no ip dhcp smart-relay** statement from ***running-config***.

Command Mode

Interface-Ethernet Configuration
Interface-Port-channel Configuration
Interface-VLAN Configuration

Command Syntax

```
ip dhcp smart-relay
no ip dhcp smart-relay
default ip dhcp smart-relay
```

Examples

- This command enables DHCP smart relay on VLAN interface 100.

```
switch(config)#interface vlan 100
switch(config-if-Vl100)#ip helper-address 10.4.4.4
switch(config-if-Vl100)#ip dhcp smart-relay
switch(config-if-Vl100)#show ip dhcp relay
DHCP Relay is active
DHCP Relay Option 82 is disabled
DHCP Smart Relay is enabled
Interface: Vlan100
  DHCP Smart Relay is enabled
  DHCP servers: 10.4.4.4
switch(config-if-Vl100)#
```

- This command disables DHCP smart relay on VLAN interface 100.

```
switch(config-if-Vl100)#no ip dhcp smart-relay
switch(config-if-Vl100)#show active
 interface Vlan100
   no ip dhcp smart-relay
   ip helper-address 10.4.4.4
switch(config-if-Vl100)#show ip dhcp relay
DHCP Relay is active
DHCP Relay Option 82 is disabled
DHCP Smart Relay is enabled
Interface: Vlan100
  DHCP Smart Relay is disabled
  DHCP servers: 10.4.4.4
switch(config-if-Vl100)#
```

- This command enables DHCP smart relay globally, configures VLAN interface 100 to use the global setting, then displays the DHCP relay status

```
switch(config)#ip dhcp smart-relay global
switch(config)#interface vlan 100
switch(config-if-Vl100)#ip helper-address 10.4.4.4
switch(config-if-Vl100)#default ip dhcp relay
switch(config-if-Vl100)#show ip dhcp relay
DHCP Relay is active
DHCP Relay Option 82 is disabled
DHCP Smart Relay is enabled
Interface: Vlan100
  Option 82 Circuit ID: 333
  DHCP Smart Relay is enabled
  DHCP servers: 10.4.4.4
switch(config-if-Vl100)#
```

ip dhcp smart-relay global

The **ip dhcp smart-relay global** command configures the global DHCP smart relay setting. DHCP smart relay supports forwarding DHCP requests with a client's secondary IP addresses in the gateway address field. The default global DHCP smart relay setting is disabled.

The global DHCP smart relay setting is applied to all interfaces for which an **ip dhcp smart-relay** statement is not configured. Enabling DHCP smart relay on an interface requires that DHCP relay is also enabled on that interface.

The **no ip dhcp smart-relay global** and **default ip dhcp smart-relay global** commands restore the global DHCP smart relay default setting of disabled by removing the **ip dhcp smart-relay global** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
ip dhcp smart-relay global
no ip dhcp smart-relay global
default ip dhcp smart-relay global
```

Related Commands

- **ip helper-address** enables the DHCP relay agent on a configuration mode interface.
- **ip dhcp smart-relay** enables the DHCP smart relay agent on a configuration mode interface.

Example

- This command configures the global DHCP smart relay setting to ***enabled***.

```
switch(config)#ip dhcp smart-relay global
switch(config)#
```

ip dhcp snooping

The **ip dhcp snooping** command enables DHCP snooping globally on the switch. DHCP snooping is a set of layer 2 processes that can be configured on LAN switches and used with DHCP servers to control network access to clients with specific IP/MAC addresses. The switch supports Option-82 insertion, which is a DHCP snooping process that allows relay agents to provide remote-ID and circuit-ID information to DHCP reply and request packets. DHCP servers use this information to determine the originating port of DHCP requests and associate a corresponding IP address to that port. DHCP servers use port information to track host location and IP address usage by authorized physical ports.

DHCP snooping uses the information option (Option-82) to include the switch MAC address (router-ID) along with the physical interface name and VLAN number (circuit-ID) in DHCP packets. After adding the information to the packet, the DHCP relay agent forwards the packet to the DHCP server as specified by the DHCP protocol.

DHCP snooping on a specified VLAN requires all of these conditions to be met:

- DHCP snooping is globally enabled.
- Insertion of option-82 information in DHCP packets is enabled.
- DHCP snooping is enabled on the specified VLAN.
- DHCP relay is enabled on the corresponding VLAN interface.

The **no ip dhcp snooping** and **default ip dhcp snooping** commands disables global DHCP snooping by removing the **ip dhcp snooping** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
ip dhcp snooping
no ip dhcp snooping
default ip dhcp snooping
```

Related Commands

- **ip dhcp snooping information option** enables insertion of option-82 snooping data.
- **ip helper-address** enables the DHCP relay agent on a configuration mode interface.

Example

- This command globally enables snooping on the switch, displaying DHCP snooping status prior and after invoking the command.

```
switch(config)#show ip dhcp snooping
DHCP Snooping is disabled
switch(config)#ip dhcp snooping
switch(config)#show ip dhcp snooping
DHCP Snooping is enabled
DHCP Snooping is not operational
DHCP Snooping is configured on following VLANs:
  None
DHCP Snooping is operational on following VLANs:
  None
Insertion of Option-82 is disabled
switch(config)#
```

ip dhcp snooping information option

The **ip dhcp snooping information option** command enables the insertion of option-82 DHCP snooping information in DHCP packets on VLANs where DHCP snooping is enabled. DHCP snooping is a layer 2 switch process that allows relay agents to provide remote-ID and circuit-ID information to DHCP reply and request packets. DHCP servers use this information to determine the originating port of DHCP requests and associate a corresponding IP address to that port.

DHCP snooping uses information option (Option-82) to include the switch MAC address (router-ID) along with the physical interface name and VLAN number (circuit-ID) in DHCP packets. After adding the information to the packet, the DHCP relay agent forwards the packet to the DHCP server through DHCP protocol processes.

DHCP snooping on a specified VLAN requires all of these conditions to be met:

- DHCP snooping is globally enabled.
- Insertion of option-82 information in DHCP packets is enabled.
- DHCP snooping is enabled on the specified VLAN.
- DHCP relay is enabled on the corresponding VLAN interface.

When global DHCP snooping is not enabled, the **ip dhcp snooping information option** command persists in ***running-config*** without any operational effect.

The **no ip dhcp snooping information option** and **default ip dhcp snooping information option** commands disable the insertion of option-82 DHCP snooping information in DHCP packets by removing the **ip dhcp snooping information option** statement from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
ip dhcp snooping information option
no ip dhcp snooping information option
default ip dhcp snooping information option
```

Related Commands

- **ip dhcp snooping** globally enables DHCP snooping.
- **ip helper-address** enables the DHCP relay agent on a configuration mode interface.

Example

- These commands enable DHCP snooping on DHCP packets from ports on snooping-enabled VLANs. DHCP snooping was previously enabled on the switch.

```
switch(config)#ip dhcp snooping information option
switch(config)#show ip dhcp snooping
DHCP Snooping is enabled
DHCP Snooping is operational
DHCP Snooping is configured on following VLANs:
  100
DHCP Snooping is operational on following VLANs:
  100
Insertion of Option-82 is enabled
  Circuit-id format: Interface name:Vlan ID
  Remote-id: 00:1c:73:1f:b4:38 (Switch MAC)
switch(config)#
```

ip dhcp snooping vlan

The **ip dhcp snooping vlan** command enables DHCP snooping on specified VLANs. DHCP snooping is a layer 2 process that allows relay agents to provide remote-ID and circuit-ID information in DHCP packets. DHCP servers use this data to determine the originating port of DHCP requests and associate a corresponding IP address to that port. DHCP snooping is configured on a global and VLAN basis.

VLAN snooping on a specified VLAN requires each of these conditions:

- DHCP snooping is globally enabled.
- Insertion of option-82 information in DHCP packets is enabled.
- DHCP snooping is enabled on the specified VLAN.
- DHCP relay is enabled on the corresponding VLAN interface.

When global DHCP snooping is not enabled, the **ip dhcp snooping vlan** command persists in ***running-config*** without any operational affect.

The **no ip dhcp snooping information option** and **default ip dhcp snooping information option** commands disable DHCP snooping operability by removing the **ip dhcp snooping information option** statement from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
ip dhcp snooping vlan v_range
no ip dhcp snooping vlan v_range
default ip dhcp snooping vlan v_range
```

Parameters

- *v_range* VLANs upon which snooping is enabled. Formats include a number, a number range, or a comma-delimited list of numbers and ranges. Numbers range from 1 to 4094.

Related Commands

- **ip dhcp snooping** globally enables DHCP snooping.
- **ip dhcp snooping information option** enables insertion of option-82 snooping data.
- **ip helper-address** enables the DHCP relay agent on a configuration mode interface.

Example

- These commands enable DHCP snooping globally, DHCP on VLAN interface100, and DHCP snooping on VLAN 100.

```
switch(config)#ip dhcp snooping
switch(config)#ip dhcp snooping information option
switch(config)#ip dhcp snooping vlan 100
switch(config)#interface vlan 100
switch(config-if-Vl100)#ip helper-address 10.4.4.4
switch(config-if-Vl100)#show ip dhcp snooping
DHCP Snooping is enabled
DHCP Snooping is operational
DHCP Snooping is configured on following VLANs:
  100
DHCP Snooping is operational on following VLANs:
  100
Insertion of Option-82 is enabled
  Circuit-id format: Interface name:Vlan ID
  Remote-id: 00:1c:73:1f:b4:38 (Switch MAC)
switch(config)#
```

ip hardware fib ecmp resilience

The **ip hardware fib ecmp resilience** command configures a fixed number of next hop entries in the hardware ECMP table for the specified IP address prefix. In addition to specifying the maximum number of next hop addresses that the table can contain for the prefix, the command includes a redundancy factor that allows duplication of each next hop address. The fixed table space for the address is the maximum number of next hops multiplied by the redundancy factor.

The default method of adding or removing next hop entries when required by the active hashing algorithm leads to inefficient management of the ECMP table, which can result in the rerouting of packets to different next hops that breaks TCP packet flows. Implementing fixed table entries for a specified IP address allows data flows that are hashed to a valid next hop number to remain intact. Additionally, traffic is evenly distributed over a new set of next hops.

The **no ip hardware fib ecmp resilience** and **default ip hardware fib ecmp resilience** commands restore the default hardware ECMP table management by removing the **ip hardware fib ecmp resilience** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
ip hardware fib ecmp resilience net_addr capacity nhop_max redundancy duplicates
no ip hardware fib ecmp resilience net_addr
default ip hardware fib ecmp resilience net_addr
```

Parameters

- *net_addr* IP address prefix managed by command. (CIDR or address-mask).
- *nhop_max* Maximum number of nexthop addresses for specified IP address prefix. Value range varies by platform:
  - Helix: <***2*** to ***64***>
  - Trident: <***2*** to ***32***>
  - Trident II: <***2*** to ***64***>
- *duplicates* Specifies the redundancy factor. Value ranges from 1 to 128.

Example

- This command configures a hardware ECMP table space of 24 entries for the IP address 10.14.2.2/24. A maximum of six nexthop addresses can be specified for the IP address. When the table contains six nexthop addresses, each appears in the table four times. When the table contains fewer than six nexthop addresses, each is duplicated until the 24 table entries are filled.

```
switch(config)#ip hardware fib ecmp resilience 10.14.2.2/24 capacity 6 redundancy
4
switch(config)#
```

ip hardware fib optimize

The **ip hardware fib optimize** command enables IPv4 route scale. The platform layer 3 agent is restarted to ensure IPv4 routes are optimized with the **agent SandL3Unicast terminate** command for the configuration mode interface.

Command Mode

Global Configuration

Command Syntax

```
ip hardware fib optimize exact-match prefix-length <prefix-length>
<optional: prefix-length>
no ip hardware fib optimize exact-match prefix-length <prefix-length>
<optional: prefix-length>
```

Parameters

- *prefix-length* The length of the prefix equal to 12, 16, 20, 24, 28, or 32. One additional prefix-length limited to the prefix-length of 32 is optional.

Related Commands

- **agent SandL3Unicast terminate** enables restarting the layer 3 agent to ensure IPv4 routes are optimized.
- **show platform arad ip route** shows resources for all IPv4 routes in hardware. Routes that use the additional hardware resources will appear with an asterisk.
- **show platform arad ip route summary** shows hardware resource usage of IPv4 routes.

Examples

- This configuration command allows configuring prefix lengths 12 and 32.

  ```
  switch(config)#ip hardware fib optimize exact-match prefix-length 12 32
  ! Please restart layer 3 forwarding agent to ensure IPv4 routes are optimized
  ```

  One of the two prefixes in this command is a prefix-length of 32, which is required in the instance where there are two prefixes. For this command to take effect, the platform layer 3 agent must be restarted.

  This configuration command restarts the platform layer 3 agent to ensure IPv4 routes are optimized.

  ```
  switch(config)#agent SandL3Unicast terminate
  SandL3Unicast was terminated
  ```

  Restarting the platform layer 3 agent results in deletion of all IPv4 routes, which are re-added to the hardware.

- This configuration command allows configuring prefix lengths 32 and 16.

  ```
  switch(config)#ip hardware fib optimize exact-match prefix-length 32 16
  ! Please restart layer 3 forwarding agent to ensure IPv4 routes are optimized
  ```

  One of the two prefixes in this command is a prefix-length of 32, which is required in the instance where there are two prefixes. For this command to take effect, the platform layer 3 agent must be restarted.

  This configuration command restarts the platform layer 3 agent to ensure IPv4 routes are optimized.

  ```
  switch(config)#agent SandL3Unicast terminate
  SandL3Unicast was terminated
  ```

Restarting the platform layer 3 agent results in deletion of all IPv4 routes, which are re-added to the hardware.

- This configuration command allows configuring prefix length 24.

```
switch(config)#ip hardware fib optimize exact-match prefix-length 24
! Please restart layer 3 forwarding agent to ensure IPv4 routes are optimized
```

In this instance, there is only one prefix-length, so a prefix-length of 32 is not required. For this command to take effect, the platform layer 3 agent must be restarted.

This configuration command restarts the platform layer 3 agent to ensure IPv4 routes are optimized.

```
switch(config)#agent SandL3Unicast terminate
SandL3Unicast was terminated
```

Restarting the platform layer 3 agent results in deletion of all IPv4 routes, which are re-added to the hardware.

- This configuration command allows configuring prefix length 32.

```
switch(config)#ip hardware fib optimize exact-match prefix-length 32
! Please restart layer 3 forwarding agent to ensure IPv4 routes are optimized
```

For this command to take effect, the platform layer 3 agent must be restarted.

This configuration command restarts the platform layer 3 agent to ensure IPv4 routes are optimized.

```
switch(config)#agent SandL3Unicast terminate
SandL3Unicast was terminated
```

Restarting the platform layer 3 agent results in deletion of all IPv4 routes, which are re-added to the hardware.

Example

- This configuration command disables configuring prefix lengths 12 and 32.

```
switch(config)#no ip hardware fib optimize exact-match prefix-length 12 32
! Please restart layer 3 forwarding agent to ensure IPv4 routes are not optimized
```

One of the two prefixes in this command is a prefix-length of 32, which is required in the instance where there are two prefixes. For this command to take effect, the platform layer 3 agent must be restarted.

ip helper-address

The **ip helper-address** command enables the DHCP relay agent on the configuration mode interface and specifies a forwarding address for DHCP requests. An interface that is configured with multiple helper-addresses forwards DHCP requests to all specified addresses.

The **no ip helper-address** and **default ip helper-address** commands remove the corresponding **ip helper-address** command from ***running-config***. Commands that do not specify an IP helper-address removes all helper-addresses from the interface.

Command Mode

Interface-Ethernet Configuration
Interface-Port-channel Configuration
Interface-VLAN Configuration

Command Syntax

```
ip helper-address ipv4_addr
no ip helper-address [ipv4_addr]
default ip helper-address [ipv4_addr]
```

Parameters

- *ipv4_addr* DHCP server address accessed by interface.

Related Commands

- **ip dhcp relay always-on**
- **ip dhcp relay information option (Global)**
- **ip dhcp relay information option circuit-id**

Example

- This command enables DHCP relay on VLAN interface 200 and configure the switch to forward DHCP requests received on this interface to the server at 10.10.41.15.

```
switch(config)#interface vlan 200
switch(config-if-Vl200)#ip helper-address 10.10.41.15
switch(config-if-Vl200)#show active
interface Vlan200
   ip helper-address 10.10.41.15
switch(config-if-Vl200)#
```

ip icmp redirect

The **ip icmp redirect** command enables the transmission of ICMP redirect messages. Routers send ICMP redirect messages to notify data link hosts of the availability of a better route for a specific destination.

The **no ip icmp redirect** disables the switch from sending ICMP redirect messages.

Command Mode

Global Configuration

Command Syntax

```
ip icmp redirect
no ip icmp redirect
default ip icmp redirect
```

Example

- This command disables the redirect messages.

```
switch(config)#no ip icmp redirect
switch(config)#show running-config
              <-------OUTPUT OMITTED FROM EXAMPLE-------->
!
no ip icmp redirect
ip routing
!
              <-------OUTPUT OMITTED FROM EXAMPLE-------->
switch(config)#
```

ip load-sharing

The **ip load-sharing** command provides the hash seed to an algorithm that the switch uses to distribute data streams among multiple equal-cost routes to an individual IPv4 subnet.

In a network topology using Equal-Cost Multipath routing, all switches performing identical hash calculations may result in hash polarization, leading to uneven load distribution among the data paths. Hash polarization is avoided when switches use different hash seeds to perform different hash calculations.

The **no ip load-sharing** and **default ip load-sharing** commands return the hash seed to the default value of zero by removing the **ip load-sharing** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
ip load-sharing HARDWARE seed
no ip load-sharing HARDWARE
default ip load-sharing HARDWARE
```

Parameters

- ***HARDWARE*** The ASIC switching device. The available option depend on the switch platform. Verify available options with the CLI ? command.
  - **arad**
  - **fm6000**
  - **petraA**
  - **trident**
- *seed* The hash seed. Value range varies by switch platform. The default value on all platforms is 0.:
  - when *HARDWARE*=**arad** *seed* ranges from 0 to 2.
  - when ***HARDWARE***=**fm6000** *seed* ranges from 0 to 39.
  - when *HARDWARE*=**petraA** *seed* ranges from 0 to 2.
  - when *HARDWARE*=**trident** *seed* ranges from 0 to 5.

Example

- This command sets the IPv4 load sharing hash seed to one on FM6000 platform switches.

```
switch(config)#ip load-sharing fm6000 1
switch(config)#
```

ip local-proxy-arp

The **ip local-proxy-arp** command enables local proxy ARP (Address Resolution Protocol) on the configuration mode interface. When local proxy ARP is enabled, ARP requests received on the configuration mode interface will return an IP address even when the request comes from within the same subnet.

The **no ip local-proxy-arp** and **default ip local-proxy-arp** commands disable local proxy ARP on the configuration mode interface by removing the corresponding **ip local-proxy-arp** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration
Interface-Loopback Configuration
Interface-Management Configuration
Interface-Port-channel Configuration
Interface-VLAN Configuration

Command Syntax

```
ip local-proxy-arp
no ip local-proxy-arp
default ip local-proxy-arp
```

Example

- These commands enable local proxy ARP on VLAN interface 140.

```
switch(config)#interface vlan 140
switch(config-if-Vl140)#ip local-proxy-arp
switch(config-if-Vl140)#show active
interface Vlan140
   ip local-proxy-arp
switch(config-if-Vl140)#
```

ip nat destination static

The **ip nat destination static** command enables NAT of a specified destination address for the configuration mode interface. This command installs hardware translation entries for forward and reverse unicast traffic. When the rule specifies a multicast group, the command does not install the reverse path in hardware. The command may include an access control list to filter packets for translation.

When configuring twice NAT, an arbitrary NAT group number is used to associate the source NAT and destination NAT rules. This number must be the same in both rules.

The **no ip nat destination static** and **default ip nat destination static** commands disables NAT translation of the specified destination address by removing the corresponding **ip nat destination static** command from ***running_config***.

Command Mode

Interface-Ethernet Configuration
Interface-Port-channel Configuration
Interface-VLAN Configuration

Command Syntax

```
ip nat destination static ORIGINAL [FILTER] TRANSLATED [PROT_TYPE] [group group_number]
no ip nat destination static ORIGINAL [FILTER] TRANSLATED [PROT_TYPE] [group group_number]
default ip nat destination static ORIGINAL [FILTER] TRANSLATED [PROT_TYPE] [group group_number]
```

Parameters

- ***ORIGINAL*** Destination address to be modified. Options include:
  - *local_ipv4* IPv4 address.
  - *local_ipv4 local_port* IPv4 address and port (port value ranges from 1 to 65535).
- ***FILTER*** Access control list that filters packets. Options include:
  - <no parameter> All packets with specified destination address are cleared.
  - **access-list** *list_name* List that specifies the packets that are cleared. Not supported when configuring twice NAT.
- ***TRANSLATED*** Destination address of translated packet. Options include:
  - *global_ipv4* IPv4 address.
  - *global_ipv4 global_port* IPv4 address and port (port value ranges from 1 to 65535). When configuring twice NAT, source and destination NAT rules must either both specify a port translation or both not specify a port translation.
- ***PROT_TYPE*** Filters packets based on protocol type. Options include:
  - <no parameter> All packets with specified destination address are cleared.
  - **protocol tcp** TCP packets with specified destination address are cleared.
  - **protocol udp** UDP packets with specified destination address are cleared.
- **group** *group_number* Used only when configuring twice NAT, the NAT group number associates a source NAT rule with a destination NAT rule on the same interface. The group number (values range from 1 to 255) is arbitrary, but must be the same in both rules.

Example

- These commands configure VLAN 201 to translate destination address 168.32.14.15 to 10.24.1.10.

```
switch(config)#interface vlan 201
switch(config-if-Vl201)#ip nat destination static 10.24.1.10 168.32.14.15
switch(config-if-Vl201)#
```

- These commands configure VLAN 201 to translate the source address 10.24.1.10 to 168.32.14.15 for all packets with IP destination addresses in the 168.10.1.1/32 subnet.

```
switch(config)#ip access-list ACL2
switch(config-acl-ACL2)#permit ip 168.10.1.1/32 any
switch(config-acl-ACL2)#exit
switch(config)#interface vlan 201
switch(config-if-Vl201)#ip nat destination static 10.24.1.10 access-list ACL2
168.32.14.15
switch(config-if-Vl201)#
```

- These commands configure Ethernet interface 2 to translate the local source address 10.24.1.10 to the global source address 168.32.14.15, and to translate the local destination address 10.68.104.3 to the global destination address 168.25.10.7 for all packets moving through the interface. The use of NAT group 3 is arbitrary, but must be the same in both rules.

```
switch(config)#interface ethernet 2
switch(config-if-Et2)#ip nat source static 10.24.1.10 168.32.14.15 group 3
switch(config-if-Et2)#ip nat destination static 10.68.104.3 168.25.10.7 group 3
```

ip nat pool

The **ip nat pool** command identifies a pool of addresses using start address, end address, and either netmask or prefix length. If its starting IP address and ending IP address are the same, there is only one address in the address pool.

The **no ip nat pool** removes the **ip nat pool** command from ***running_config***.

Command Mode

Global Configuration

Command Syntax

```
ip nat pool pool_name [ADDRESS_SPAN] SUBNET_SIZE
no ip nat pool pool_name
default ip nat pool pool_name
```

Parameters

- *pool_name* name of the IP address pool.
- ***ADDRESS_SPAN*** Options include:
  - *start_addr* The first IP address in the address pool (IPv4 addresses in dotted decimal notation).
  - *end_addr* The last IP address in the address pool. (IPv4 addresses in dotted decimal notation).
- ***SUBNET_SIZE*** this functions as a sanity check to ensure it is not a network or broadcast network. Options include:
  - **netmask** *ipv4_addr* The netmask of the address pool's network (dotted decimal notation).
  - **prefix-length** <***0*** to ***32***> The number of bits of the netmask (of the address pool's network) that are ones (how many bits of the address indicate network).

Examples

- This command configures the pool of addresses using start address, end address, and prefix length of 24.

```
switch(config)#ip nat pool poo1 10.15.15.15 10.15.15.25 prefix-length 24
switch(config)
```

- This command removes the pool of addresses.

```
switch(config)# no ip nat pool poo1 10.15.15.15 10.15.15.25 prefix-length 24
switch(config)
```

ip nat source dynamic

The **ip nat source dynamic** command enables NAT of a specified source address for packets sent and received on the configuration mode interface. This command installs hardware translation entries for forward and reverse traffic. When the rule specifies a multicast group, the command does not install the reverse path in hardware. The command may include an access control list to filter packets for translation.

The **no ip nat source dynamic** and **default ip nat source dynamic** commands disables NAT translation of the specified destination address by removing the corresponding **ip nat source dynamic** command from ***running_config***.

Note Ethernet and Port-channel interfaces should be configured as routed ports.

Command Mode

Interface-Ethernet Configuration
Interface-Port-channel Configuration
Interface-VLAN Configuration

Command Syntax

```
ip nat source dynamic access-list acl_name POOL_TYPE
no ip nat source dynamic access-list acl_name
default ip nat source dynamic access-list acl_name
```

Parameters

- *acl_name* Access control list that controls the internal network addresses eligible for NAT.
- ***POOL_TYPE*** Options include:
  - **overload** Translates multiple local addresses to a single global address. When overloading is enabled, conversations using the same IP address are distinguished by their TCP or UDP port number.
  - **pool** *pool_name* The name of the IP address pool. The pool is defined using the **ip nat pool** command.

    The pool option is required even if the pool has just one address. NAT uses that one address for all of the translations.

    - **pool_fullcone** Enables full cone NAT where all requests from the same internal IP address and port are mapped to the same external IP address and port.

Example

- This command configures the dynamic NAT source address and sets the NAT overload for pool P2.

```
switch(config)#interface ethernet 3/1
switch(config-if-Et3/1)#ip nat source dynamic access-list ACL2 pool p2
switch#
```

- This command disables the NAT source translation on interface Ethernet 3/1.

```
switch(config)#interface ethernet 3/1
switch(config-if-Et3/1)# no ip nat source dynamic access-list ACL2
switch(config-if-Et3/1)#
```

ip nat source static

The **ip nat source static** command enables NAT of a specified source address for the configuration mode interface. This command installs hardware translation entries for forward and reverse unicast traffic. When the rule specifies a multicast group, the command does not install the reverse path in hardware. The command may include an access control list to filter packets for translation.

When configuring twice NAT, an arbitrary NAT group number is used to associate the source NAT and destination NAT rules. This number must be the same in both rules.

The **no ip nat source static** and **default ip nat source static** commands disables NAT translation of the specified source address by removing the corresponding **ip nat source** command from ***running_config***.

Command Mode

Interface-Ethernet Configuration
Interface-Port-channel Configuration
Interface-VLAN Configuration

Command Syntax

```
ip nat source static ORIGINAL [FILTER] TRANSLATED [PROT_TYPE] [group
group_number]
no ip nat source static ORIGINAL [FILTER] TRANSLATED [PROT_TYPE] [group
group_number]
default ip nat source static ORIGINAL [FILTER] TRANSLATED [PROT_TYPE] [group
group_number]
```

Parameters

- ***ORIGINAL*** Source address to be modified. Options include:
  - *original_ipv4* IPv4 address.
  - *original_ipv4* *original_port* IPv4 address and port (port value ranges from 1 to 65535).
- ***FILTER*** Access control list that filters packets. Options include:
  - <no parameter> All packets with specified source address are cleared.
  - **access-list** *list_name* List that specifies the packets that are cleared. Not supported when configuring twice NAT.
- ***TRANSLATED*** Source address of translated packet. Options include:
  - *translated_ipv4* IPv4 address.
  - *translated_ipv4* *translated_port* IPv4 address and port (port value ranges from 1 to 65535). When configuring twice NAT, source and destination NAT rules must either both specify a port translation or both not specify a port translation.
- ***PROT_TYPE*** Filters packets based on protocol type. Options include:
  - <no parameter> All packets with specified source address are cleared.
  - **protocol tcp** TCP packets with specified source address are cleared.
  - **protocol udp** UDP packets with specified source address are cleared.
- **group** *group_number* Used only when configuring twice NAT, the NAT group number associates a source NAT rule with a destination NAT rule on the same interface. The group number (values range from 1 to 255) is arbitrary, but must be the same in both rules.

Restrictions

- If ***ORIGINAL*** includes a port, TRANSLATED must also include a port.
- If ***ORIGINAL*** does not include a port, TRANSLATED cannot include a port.

Example

- These commands configure VLAN 101 to translate source address 10.24.1.10 to 168.32.14.15.

```
switch(config)#interface vlan 101
switch(config-if-Vl101)#ip nat source static 10.24.1.10 168.32.14.15
switch(config-if-Vl101)#
```

- These commands configure VLAN 100 to translate the source address 10.24.1.10 to 168.32.14.15 for all packets with IP destination addresses in the 168.10.1.1/32 subnet.

```
switch(config)#ip access-list ACL1
switch(config-acl-ACL1)#permit ip any 168.10.1.1/32
switch(config-acl-ACL1)#exit
switch(config)#interface vlan 101
switch(config-if-Vl101)#ip nat source static 10.24.1.10 access-list ACL1
168.32.141.15
switch(config-if-Vl101)#
```

- These commands configure Ethernet interface 2 to translate the local source address 10.24.1.10 to the global source address 168.32.14.15, and to translate the local destination address 10.68.104.3 to the global destination address 168.25.10.7 for all packets moving through the interface. The use of NAT group 3 is arbitrary, but must be the same in both rules.

```
switch(config)#interface ethernet 2
switch(config-if-Et2)#ip nat source static 10.24.1.10 168.32.14.15 group 3
switch(config-if-Et2)#ip nat destination static 10.68.104.3 168.25.10.7 group 3
```

ip nat translation low-mark

The **ip nat translation low-mark** command configures the minimum threshold that triggers the resumption of programming new NAT translation connections.

The **ip nat translation max-entries** command specifies the maximum number of NAT translation connections that can be stored. When this limit is reached, new connections are dropped instead of being programmed in hardware or software. At this point no new connections will be programmed until the number of stored entries drop below the configured low-mark, expressed as a percentage of the max-entries value. The default low mark value is 90%.

The **no ip nat translation low-mark** and **default ip nat translation low-mark** commands restores the default low-mark value by removing the **ip nat translation low-mark** command from ***running_config***.

Command Mode

Global Configuration

Command Syntax

```
ip nat translation low-mark threshold
no ip nat translation low-mark
default ip nat translation low-mark
```

Parameters

- *threshold* Percentage of maximum connection entries. Value ranges from ***1*** to ***99***. Default is 90.

Examples

- This command globally sets the translation low mark of 93%.

```
switch(config)#ip nat translation low-mark 93
switch(config)#
```

ip nat translation max-entries

The **ip nat translation max-entries** command specifies maximum number of NAT translation connections. After this threshold is reached, new connections are dropped until the number of programmed connections is reduced below the level specified by the **ip nat translation low-mark** command.

The **no ip nat translation max-entries** and **default ip nat translation max-entries** commands removes the maximum connection limit and resets the parameter value to zero by removing the **ip nat translation max-entries** command from ***running_config***.

Command Mode

Global Configuration

Command Syntax

```
ip nat translation max-entries connections
no ip nat translation max-entries
default ip nat translation max-entries
```

Parameters

- *connections* The maximum number of NAT translation connections. Value ranges from ***0*** to ***4294967295***. Default value is 0, which removes the connection limit.

Examples

- This command limits the number of NAT translation connections the switch can store to 3000.

```
switch(config)#ip nat translation max-entries 3000
switch(config)#
```

ip nat translation tcp-timeout

The **ip nat translation tcp-timeout** command specifies the translation timeout period for translation table entries. The timeout period specifies the interval during which the switch will attempt to reuse an existing TCP translation for devices specified by table entries.

The **no ip nat translation tcp-timeout** and **default ip nat translation tcp-timeout** commands reset the timeout to its default by removing the corresponding **ip nat translation tcp-timeout** command from ***running_config***.

Command Mode

Global Configuration

Command Syntax

```
ip nat translation tcp-timeout period
no ip nat translation tcp-timeout
default ip nat translation tcp-timeout
```

Parameters

- *period* Time-out period in seconds for port translations. Value ranges from ***0*** to ***4294967295***. Default value is 86400 (24 hours).

Examples

- This command sets the TCP timeout for translations to 600 seconds.

```
switch(config)# ip nat translation tcp-timeout 600
switch(config)#
```

- This command removes the TCP translation timeout.

```
switch(config)# no ip nat translation tcp-timeout
switch(config)#
```

ip nat translation udp-timeout

The **ip nat translation udp-timeout** command specifies the translation timeout period for translation table entries. The timeout period specifies the interval the switch attempts to establish a UDP connection with devices specified by table entries.

The **no ip nat translation udp-timeout** and **default ip nat translation udp-timeout** commands disables NAT translation of the specified destination address by removing the corresponding **ip nat translation udp-timeout** command from ***running_config***.

Command Mode

Global Configuration

Command Syntax

```
ip nat translation udp-timeout period
no ip nat translation udp-timeout
default ip nat translation udp-timeout
```

Parameters

- *period* Value ranges from 0 to 4294967295. Default value is 300 (5 minutes).

Examples

- This command globally sets the timeout for UDP to 800 seconds.

```
switch(config)# ip nat translation udp-timeout 800
switch(config)#
```

- This command removes the timeout for UDP.

```
switch(config)# no ip nat translation udp-timeout
switch(config)#
```

ip proxy-arp

The **ip proxy-arp** command enables proxy ARP on the configuration mode interface. Proxy ARP is disabled by default.

The **no ip proxy-arp** and **default ip proxy-arp** commands disable proxy ARP on the configuration mode interface by removing the corresponding **ip proxy-arp** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration
Interface-Loopback Configuration
Interface-Management Configuration
Interface-Port-channel Configuration
Interface-VLAN Configuration

Command Syntax

```
ip proxy-arp
no ip proxy-arp
default ip proxy-arp
```

Examples

- This command enables proxy ARP on Ethernet interface 4.

```
switch(config)#interface ethernet 4
switch(config-if-Et4)#ip proxy-arp
switch(config-if-Et4)#
```

ip route

The **ip route** command creates a static route. The destination is a network segment; the nexthop address is either an IPv4 address or a routable port. When multiple routes exist to a destination prefix, the route with the lowest administrative distance takes precedence.

By default, the administrative distance assigned to static routes is 1. Assigning a higher administrative distance to a static route configures it to be overridden by dynamic routing data. For example, a static route with an administrative distance value of 200 is overridden by OSPF intra-area routes, which have a default administrative distance of 110.

Tags are used by route maps to filter routes. The default tag value on static routes is 0.

Multiple routes with the same destination and the same administrative distance comprise an Equal Cost Multi-Path (ECMP) route. The switch attempts to spread outbound traffic equally through all ECMP route paths. All paths comprising an ECMP are assigned identical tag values; commands that change the tag value of a path change the tag value of all paths in the ECMP.

The **no ip route** and **default ip route** commands delete the specified static route by removing the corresponding **ip route** command from ***running-config***. Commands that do not list a nexthop address remove all **ip route** statements with the specified destination from ***running-config***. If an **ip route** statement exists for the same IP address in multiple VRFs, each must be removed separately. All static routes in a user-defined VRF are deleted when the VRF is deleted.

Command Mode

Global Configuration

Command Syntax

```
ip route [VRF_INSTANCE] dest_net NEXTHOP [DISTANCE] [TAG_OPTION] [RT_NAME]
no ip route [VRF_INSTANCE] dest_net [NEXTHOP] [DISTANCE]
default ip route [VRF_INSTANCE] dest_net [NEXTHOP] [DISTANCE]
```

Parameters

- ***VRF_INSTANCE*** Specifies the VRF instance being modified.
  - <no parameter> Changes are made to the default VRF.
  - **vrf** *vrf_name* Changes are made to the specified VRF.
- *dest_net* Destination IPv4 subnet (CIDR or address-mask notation).
- ***NEXTHOP*** Location or access method of next hop device. Options include:
  - *ipv4_addr* An IPv4 address.
  - **null0** Null0 interface.
  - **ethernet** *e_num* Ethernet interface specified by *e_num*.
  - **loopback** *l_num* Loopback interface specified by *l_num*.
  - **management** *m_num* Management interface specified by *m_num*.
  - **port-channel** *p_num* Port-channel interface specified by *p_num*.
  - **vlan** *v_num* VLAN interface specified by *v_num*.
  - **vxlan** *vx_num* VXLAN interface specified by *vx_num*.
- ***DISTANCE*** Administrative distance assigned to route. Options include:
  - <no parameter> Route assigned default administrative distance of one.
  - <1-255> The administrative distance assigned to route.
- ***TAG_OPTION*** static route tag. Options include:

  - <no parameter> Assigns default static route tag of 0.
  - **tag** *t_value* Static route tag value. *t_value* ranges from ***0*** to ***4294967295***.
- ***RT_NAME*** Associates descriptive text to the route. Options include:
  - <no parameter> No text is associated with the route.
  - **name** *descriptive_text* The specified text is assigned to the route.

Related Commands

- **ip route nexthop-group** command creates a static route that specifies a Nexthop Group to determine the Nexthop address.

Example

- This command creates a static route in the default VRF.

```
switch(config)#ip route 172.17.252.0/24 vlan 2000
switch(config)#
```

ip routing

The **ip routing** command enables IPv4 routing. When IPv4 routing is enabled, the switch attempts to deliver inbound packets to destination IPv4 addresses by forwarding them to interfaces or next hop addresses specified by the forwarding table.

The **no ip routing** and **default ip routing** commands disable IPv4 routing by removing the **ip routing** command from ***running-config***. When IPv4 routing is disabled, the switch attempts to deliver inbound packets to their destination MAC addresses. When this address matches the switch's MAC address, the packet is delivered to the CPU. IP packets with IPv4 destinations that differ from the switch's address are typically discarded. The **delete-static-routes** option removes static entries from the routing table.

IPv4 routing is disabled by default.

Command Mode

Global Configuration

Command Syntax

```
ip routing [VRF_INSTANCE]
no ip routing [DELETE_ROUTES] [VRF_INSTANCE]
default ip routing [DELETE_ROUTES] [VRF_INSTANCE]
```

Parameters

- ***DELETE_ROUTES*** Resolves routing table static entries when routing is disabled.
  - <no parameter> Routing table retains static entries.
  - **delete-static-routes** Static entries are removed from the routing table.
- ***VRF_INSTANCE*** specifies the VRF instance being modified.
  - <no parameter> changes are made to the default VRF.
  - **vrf** *vrf_name* changes are made to the specified user-defined VRF.

Example

- This command enables IPv4 routing.

```
switch(config)#ip routing
switch(config)#
```

ip source binding

IP source guard (IPSG) is supported on Layer 2 Port-Channels, not member ports. The IPSG configuration on port channels supersedes the configuration on the physical member ports. Hence, source IP MAC binding entries should be configured on port channels. When configured on a port channel member port, IPSG does not take effect until this port is deleted from the port channel configuration.

Note IP source bindings are also used by static ARP inspection.

The **no ip source binding** and **default ip source binding** commands exclude parameters from IPSG filtering, and set the default for **ip source binding**.

Command Mode

Interface-Ethernet Configuration

Command Syntax

ip source binding [*IP_ADDRESS***] [***MAC_ADDRESS***] vlan [***VLAN_RANGE***] interface [***INTERFACE***]**
no ip source binding [*IP_ADDRESS***] [***MAC_ADDRESS***] vlan [***VLAN_RANGE***] interface [***INTERFACE***]**
default ip source binding [*IP_ADDRESS***] [***MAC_ADDRESS***] vlan [***VLAN_RANGE***] interface [***INTERFACE***]**

Parameters

- ***IP_ADDRESS*** Specifies the IP ADDRESS.
- ***MAC_ADDRESS*** Specifies the MAC ADDRESS.
- ***VLAN_RANGE*** Specifies the VLAN ID range.
- ***INTERFACE*** Specifies the Ethernet interface.

Related Commands

- **ip verify source**
- **show ip verify source**

Example

- This command configures source IP-MAC binding entries to IP address 10.1.1.1, MAC address 0000.aaaa.1111, VLAN ID 4094, and Ethernet interface 36.

```
switch(config)#ip source binding 10.1.1.1 0000.aaaa.1111 vlan 4094 interface
ethernet 36
switch(config)#
```

ip verify

The **ip verify** command configures Unicast Reverse Path Forwarding (uRPF) for inbound IPv4 packets on the configuration mode interface. uRPF verifies the accessibility of source IP addresses in packets that the switch forwards.

uRPF defines two operational modes: strict mode and loose mode.

- Strict mode: uRPF verifies that a packet is received on the interface that its routing table entry specifies for its return packet.
- Loose mode: uRPF validation does not consider the inbound packet's ingress interface only that there is a valid return path.

The **no ip verify** and **default ip verify** commands disable uRPF on the configuration mode interface by deleting the corresponding **ip verify** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration
Interface-Loopback Configuration
Interface-Management Configuration
Interface-Port-Channel Configuration
Interface-VLAN Configuration

Command Syntax

```
ip verify unicast source reachable-via RPF_MODE
no ip verify unicast
default ip verify unicast
```

Parameters

- ***RPF_MODE*** Specifies the uRPF mode. Options include:
  - **any** Loose mode.
  - **rx** Strict mode.
  - **rx allow-default** Strict mode. All inbound packets are forwarded if a default route is defined.

Guidelines

The first IPv4 uRPF implementation briefly disrupts IPv4 unicast routing. Subsequent **ip verify** commands on any interface do not disrupt IPv4 routing.

Example

- This command enables uRPF loose mode on VLAN interface 17.

```
switch(config)#interface vlan 17
switch(config-if-Vl17)#ip verify unicast source reachable-via any
switch(config-if-Vl17)#show active
 interface Vlan17
   ip verify unicast source reachable-via any
switch(config-if-Vl17)#
```

- This command enables uRPF strict mode on VLAN interface 18.

```
switch(config)#interface vlan 18
switch(config-if-Vl18)#ip verify unicast source reachable-via rx
switch(config-if-Vl18)#show active
 interface Vlan18
   ip verify unicast source reachable-via rx
switch(config-if-Vl18)#
```

ip verify source

The **ip verify source** command configures IP source guard (IPSG) applicable only to Layer 2 ports. When configured on Layer 3 ports, IPSG does not take effect until this interface is converted to Layer 2.

IPSG is supported on Layer 2 Port-Channels, not member ports. The IPSG configuration on port channels supersedes the configuration on the physical member ports. Hence, source IP MAC binding entries should be configured on port channels. When configured on a port channel member port, IPSG does not take effect until this port is deleted from the port channel configuration.

The **no ip verify source** and **default ip verify source** commands exclude VLAN IDs from IPSG filtering, and set the default for **ip verify source**.

Command Mode

Interface-Ethernet Configuration

Command Syntax

```
ip verify source vlan [VLAN_RANGE]
no ip verify source [VLAN_RANGE]
default ip verify source
```

Parameters

- ***VLAN_RANGE*** Specifies the VLAN ID range.

Related Commands

- **ip source binding**
- **show ip verify source**

Example

- This command excludes VLAN IDs 1 through 3 from IPSG filtering. When enabled on a trunk port, IPSG filters the inbound IP packets on all allowed VLANs. IP packets received on VLANs 4 through 10 on Ethernet 36 will be filtered by IPSG, while those received on VLANs 1 through 3 are permitted.

```
switch(config)#no ip verify source vlan 1-3
switch(config)#interface ethernet 36
switch(config-if-Et36)#switchport mode trunk
switch(config-if-Et36)#switchport trunk allowed vlan 1-10
switch(config-if-Et36)#ip verify source
switch(config-if-Et36)#
```

platform trident forwarding-table partition

The **platform trident forwarding-table partition** command provides a shared table memory for L2, L3 and algorithmic LPM entries that can be partitioned in different ways.

Instead of having fixed-size tables for L2 MAC entry tables, L3 IP forwarding tables, and Longest Prefix Match (LPM) routes, the tables can be unified into a single shareable forwarding table.

Important! Changing the Unified Forwarding Table mode causes the forwarding agent to restart, briefly disrupting traffic forwarding on all ports.

The **no platform trident forwarding-table partition** and **default platform trident forwarding-table partition** commands remove the **platform trident forwarding-table partition** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
platform trident forwarding-table partition SIZE
no platform trident forwarding-table partition
default platform trident forwarding-table partition
```

Parameters

- ***SIZE*** Size of partition. Options include:
  - **0** 288k l2 entries, 16k host entries, 16k lpm entries
  - **1** 224k l2 entries, 80k host entries, 16k lpm entries
  - **2** 160k l2 entries, 144k host entries, 16k lpm entries
  - **3** 96k l2 entries, 208k host entries, 16k lpm entries

  Default value is ***2*** (160k l2 entries, 144k host entries, 16k lpm entries).

Example

- This command sets the single shareable forwarding table to option 2 that supports 160k L2 entries, 144k host entries, and 16k LPM entries.

```
switch(config)#platform trident forwarding-table partition 2
switch(config)
```

- This command sets the single shareable forwarding table to option 3 that supports 96k L2 entries, 208k host entries, and 16k LPM entries. Since the switch was previously configured to option 2, you'll see a warning notice before the changes are implemented.

```
#switch(config)# platform trident forwarding-table partition 3
Warning: StrataAgent will restart immediately
```

platform trident routing-table partition

The **platform trident routing-table partition** command manages the partition sizes for the hardware LPM table that stores IPv6 routes of varying sizes.

An IPv6 route of length /64 (or shorter) requires half the hardware resources of an IPv6 route that is longer than /64. The switch installs routes of varying lengths in different table partitions. This command specifies the size of these partitions to optimize table usage.

Important! Changing the routing table partition mode causes the forwarding agent to restart, briefly disrupting traffic forwarding on all ports

The **no platform trident routing-table partition** and **default platform trident routing-table partition** commands restore the default partitions sizes by removing the **platform trident routing-table partition** command from ***running-config*.**

Command Mode

Global Configuration

Command Syntax

```
platform trident routing-table partition SIZE
no platform trident routing-table partition
default platform trident routing-table partition
```

Parameters

- ***SIZE*** Size of partition. Options include:
  - **1** 16k IPv4 entries, 6k IPv6 (/64 and smaller) entries, 1k IPv6 (any prefix length)
  - **2** 16k IPv4 entries, 4k IPv6 (/64 and smaller) entries, 2k IPv6 (any prefix length)
  - **3** 16k IPv4 entries, 2k IPv6 (/64 and smaller) entries, 3k IPv6 (any prefix length)

  Default value is ***2*** (16k IPv4 entries, 4k IPv6 (/64 and smaller) entries, 2k IPv6 (any prefix length).

Restrictions

Partition allocation cannot be changed from the default setting when uRPF is enabled for IPv6 traffic.

Example

- This command sets the shareable routing table to option 1 that supports 6K prefixes equal to or shorter than /64 and 1K prefixes longer than /64.

```
switch(config)#platform trident routing-table partition 1
switch(config)
```

rd (VRF configuration mode)

The **rd** command adds a route distinguisher (RD) to the configuration mode VRF. RDs internally identify routes belonging to a VRF to distinguish overlapping or duplicate IP address ranges. This allows the creation of distinct routes to the same IP address for different VPNs. The RD is a 64-bit number made up of an AS number or IPv4 address followed by a user-selected ID number.

Once an RD has been assigned to a VRF it cannot be changed. If the RD must be changed, remove the VRF using the **no default** form of the **vrf definition** command, then create it again.

Command Mode

VRF Configuration

Command Syntax

rd *admin_ID*:*local_assignment*

Parameters

- *admin_ID* An AS number or globally assigned IPv4 address identifying the entity assigning the RD. This should be an IANA-assigned identifying number.
- *local_assignment* A locally assigned number distinguishing the VRF. Values range from 0-65535 if the *admin_ID* is an IPv4 address, or from 0-4,294,967,295 if the *admin_ID* is an AS number. If the *admin_ID* is an AS number, the *local_assignment* can also be entered in the form of an IPv4 address.

Example

- These commands identify the administrator of the VRF called “purple” as AS 530 and assign 12 as its local number.

```
switch(config)#vrf definition purple
switch(config-vrf-purple)#rd 530:12
switch(config-vrf-purple)#
```

routing-context vrf

The **routing-context vrf** command specifies the context-active VRF. The context-active VRF determines the default VRF that VRF-context aware commands use when displaying routing table data.

Command Mode

Privileged EXEC

Command Syntax

```
routing-context vrf [VRF_ID]
```

Parameters

- ***VRF_ID*** Name of VRF assigned as the current VRF scope. Options include:
  - *vrf_name* Name of user-defined VRF.
  - **default** System-default VRF.

Guidelines

VRF-context aware commands include:

- **clear arp-cache**
- **show ip**
- **show ip arp**
- **show ip route**
- **show ip route gateway**
- **show ip route host**
- **copy**
- **install source**
- **ping**
- **tcpdump**
- **telnet**
- **traceroute**

Related Commands

- **show routing-context vrf** displays the context-active VRF.

Example

- These commands specify ***magenta*** as the context-active VRF, then display the context-active VRF.

```
switch#routing-context vrf magenta
switch#show routing-context vrf
Current VRF routing-context is magenta
switch#
```

show arp

The **show arp** command displays all ARP tables. This command differs from the **show ip arp** command in that it shows MAC bindings for all protocols, whereas **show ip arp** only displays MAC address – IP address bindings. Addresses are displayed as their host name by including the ***resolve*** argument.

Command Mode

EXEC

Command Syntax

show arp [*VRF_INST*][*FORMAT*][*HOST_ADD*][*HOST_NAME*][*INTF*][*MAC_ADDR*][*DATA*]

Parameters

The ***VRF_INST*** and ***FORMAT*** parameters are always listed first and second. The ***DATA*** parameter is always listed last. All other parameters can be placed in any order.

- ***VRF_INST*** specifies the VRF instance for which data is displayed.
  - <no parameter> context-active VRF.
  - **vrf** *vrf_name* specifies name of VRF instance. System default VRF is specified by **default**.
- ***FORMAT*** Display format of host address. Options include:
  - <no parameter> entries associate hardware address with an IPv4 address.
  - **resolve** entry associate hardware address with a host name (if it exists).
- ***HOST_ADD*** IPv4 address by which routing table entries are filtered. Options include:
  - <no parameter> routing table entries are not filtered by host address.
  - *ipv4_addr* table entries matching specified IPv4 address.
- ***HOST_NAME*** Host name by which routing table entries are filtered. Options include:
  - <no parameter> routing table entries are not filtered by host name.
  - **host** *hostname* entries matching *hostname* (text).
- ***INTF*** interfaces for which command displays status.
  - <no parameter> Routing table entries are not filtered by interface.
  - **interface ethernet** *e_num* Routed Ethernet interface specified by *e_num*.
  - **interface loopback** *l_num* Routed loopback interface specified by *l_num*.
  - **interface management** *m_num* Routed management interface specified by *m_num*.
  - **interface port-channel** *p_num* Routed port channel Interface specified by *p_num*.
  - **interface vlan** *v_num* VLAN interface specified by *v_num*.
  - **interface vxlan** *vx_num* VXLAN interface specified by *vx_num*.
- ***MAC_ADDR*** MAC address by which routing table entries are filtered. Options include:
  - <no parameter> Routing table entries are not filtered by interface MAC address.
  - **mac_address** *mac_address* entries matching *mac_address* (dotted hex notation – H.H.H).
- ***DATA*** Detail of information provided by command. Options include:
  - <no parameter> Routing table entries.
  - **summary** Summary of ARP table entries.
  - **summary total** Number of ARP table entries.

Related Commands

- **routing-context vrf** specifies the context-active VRF.

Example

- This command displays the ARP table.

```
switch>show arp
Address         Age (min)  Hardware Addr   Interface
172.22.30.1             0  001c.730b.1d15  Management1
172.22.30.133           0  001c.7304.3906  Management1
switch>
```

show ip

The **show ip** command displays IPv4 routing, IPv6 routing, IPv4 multicast routing, and VRRP status on the switch.

Command Mode

EXEC

Command Syntax

show ip

Example

- This command displays IPv4 routing status.

```
switch>show ip

IP Routing : Enabled
IP Multicast Routing : Disabled
VRRP: Configured on 0 interfaces

IPv6 Unicast Routing : Enabled
IPv6 ECMP Route support : False
IPv6 ECMP Route nexthop index: 5
IPv6 ECMP Route num prefix bits for nexthop index: 10

switch>
```

show ip arp

The **show ip arp** command displays ARP cache entries that map an IPv4 address to a corresponding MAC address. The table displays addresses by their host names when the command includes the ***resolve*** argument.

Command Mode

EXEC

Command Syntax

show ip arp [*VRF_INST*][*FORMAT*][*HOST_ADD*][*HOST_NAME*][*INTF*][*MAC_ADDR*][*DATA*]

Parameters

The ***VRF_INST*** and ***FORMAT*** parameters are always listed first and second. The ***DATA*** parameter is always listed last. All other parameters can be placed in any order.

- ***VRF_INST*** specifies the VRF instance for which data is displayed.
  - <no parameter> context-active VRF.
  - **vrf** *vrf_name* specifies name of VRF instance. System default VRF is specified by **default**.
- ***FORMAT*** Display format of host address. Options include:
  - <no parameter> entries associate hardware address with an IPv4 address.
  - **resolve** entry associate hardware address with a host name (if it exists).
- ***HOST_ADDR*** IPv4 address by which routing table entries are filtered. Options include:
  - <no parameter> routing table entries are not filtered by host address.
  - *ipv4_addr* table entries matching specified IPv4 address.
- ***HOST_NAME*** Host name by which routing table entries are filtered. Options include:
  - <no parameter> routing table entries are not filtered by host name.
  - **host** *hostname* entries matching *hostname* (text).
- ***INTERFACE_NAME*** interfaces for which command displays status.
  - <no parameter> Routing table entries are not filtered by interface.
  - **interface ethernet** *e_num* Routed Ethernet interface specified by *e_num*.
  - **interface loopback** *l_num* Routed loopback interface specified by *l_num*.
  - **interface management** *m_num* Routed management interface specified by *m_num*.
  - **interface port-channel** *p_num* Routed port channel Interface specified by *p_num*.
  - **interface vlan** *v_num* VLAN interface specified by *v_num*.
  - **interface vxlan** *vx_num* VXLAN interface specified by *vx_num*.
- ***MAC_ADDR*** MAC address by which routing table entries are filtered. Options include:
  - <no parameter> Routing table entries are not filtered by interface MAC address.
  - **mac_address** *mac_address* entries matching *mac_address* (dotted hex notation – H.H.H).
- ***DATA*** Detail of information provided by command. Options include:
  - <no parameter> Routing table entries.
  - **summary** Summary of ARP table entries.
  - **summary total** Number of ARP table entries.

Related Commands

- **routing-context vrf** specifies the context-active VRF.

Examples

- This command displays ARP cache entries that map MAC addresses to IPv4 addresses.

```
switch>show ip arp
Address         Age (min)  Hardware Addr   Interface
172.25.0.2              0  004c.6211.021e  Vlan101, Port-Channel2
172.22.0.1              0  004c.6214.3699  Vlan1000, Port-Channel1
172.22.0.2              0  004c.6219.a0f3  Vlan1000, Port-Channel1
172.22.0.3              0  0045.4942.a32c  Vlan1000, Ethernet33
172.22.0.5              0  f012.3118.c09d  Vlan1000, Port-Channel1
172.22.0.6              0  00e1.d11a.a1eb  Vlan1000, Ethernet5
172.22.0.7              0  004f.e320.cd23  Vlan1000, Ethernet6
172.22.0.8              0  0032.48da.f9d9  Vlan1000, Ethernet37
172.22.0.9              0  0018.910a.1fc5  Vlan1000, Ethernet29
172.22.0.11             0  0056.cbe9.8510  Vlan1000, Ethernet26
switch>
```

- This command displays ARP cache entries that map MAC addresses to IPv4 addresses. Host names assigned to IP addresses are displayed in place of the address.

```
switch>show ip arp resolve
Address         Age (min)  Hardware Addr   Interface
green-vl101.new         0  004c.6211.021e  Vlan101, Port-Channel2
172.22.0.1              0  004c.6214.3699  Vlan1000, Port-Channel1
orange-vl1000.n         0  004c.6219.a0f3  Vlan1000, Port-Channel1
172.22.0.3              0  0045.4942.a32c  Vlan1000, Ethernet33
purple.newcompa         0  f012.3118.c09d  Vlan1000, Port-Channel1
pink.newcompany         0  00e1.d11a.a1eb  Vlan1000, Ethernet5
yellow.newcompa         0  004f.e320.cd23  Vlan1000, Ethernet6
172.22.0.8              0  0032.48da.f9d9  Vlan1000, Ethernet37
royalblue.newco         0  0018.910a.1fc5  Vlan1000, Ethernet29
172.22.0.11             0  0056.cbe9.8510  Vlan1000, Ethernet26
switch>
```

show ip arp inspection vlan

The **show ip arp inspection vlan** command displays the configuration and operation state of ARP inspection. For a VLAN range specified, only VLANs with ARP inspection enabled will be displayed. If no VLAN is specified, all VLANs with ARP inspection enabled are displayed. The operation state turns to `Active` when hardware is ready to trap ARP packets for inspection.

Command Mode

EXEC

Command Syntax

`show ip arp inspection vlan [LIST]`

Parameters

- ***LIST*** specifies the VLAN interface number.

Related Commands

- **ip arp inspection limit**
- **ip arp inspection trust**
- **show ip arp inspection statistics**

Example

- This command displays the configuration and operation state of ARP inspection for VLANs 1 through 150.

```
switch(config)#show ip arp inspection vlan 1 - 150
VLAN 1
----------
Configuration
: Enabled
Operation State : Active
VLAN 2
----------
Configuration
: Enabled
Operation State : Active
{...}
VLAN 150
----------
Configuration
: Enabled
Operation State : Active

switch(config)#
```

show ip arp inspection statistics

The **show ip arp inspection statistics** command displays the statistics of inspected ARP packets. For a VLAN specified, only VLANs with ARP inspection enabled will be displayed. If no VLAN is specified, all VLANs with ARP inspection enabled are displayed.

Command Mode

EXEC

Command Syntax

show ip arp inspection statistics [vlan [*VID*] | [INTERFACE] interface <*intf_slot/intf_port*>]

Parameters

- ***VID*** specifies the VLAN interface ID.
- ***INTERFACE*** specifies the interface (e.g., Ethernet).
  - <intf_slot> interface slot.
  - <intf_port> interface port.
- ***INTF*** specifies the VLAN interface slot and port.

Related Commands

- **ip arp inspection limit**
- **ip arp inspection trust**
- **show ip arp inspection vlan**

Examples

- This command display statistics of inspected ARP packets for VLAN 10.

```
switch(config)#show ip arp inspection statistics vlan 10
Vlan : 10
--------------
ARP
Req Forwarded = 20
ARP Res Forwarded = 20
ARP Req Dropped = 1
ARP Res Dropped = 1
Last invalid ARP:
Time: 10:20:30 ( 5 minutes ago )
Reason: Bad IP/Mac match
Received on: Ethernet 3/1
Packet:
  Source MAC: 00:01:00:01:00:01
  Dest MAC: 00:02:00:02:00:02
  ARP Type: Request
  ARP Sender MAC: 00:01:00:01:00:01
  ARP Sender IP: 1.1.1

switch(config)#
```

- This command displays ARP inspection statistics for Ethernet interface 3/1.

```
switch(config)#show ip arp inspection statistics ethernet interface 3/1
Interface : 3/1
--------
ARP Req Forwarded = 10
ARP Res Forwarded = 10
ARP Req Dropped = 1
ARP Res Dropped = 1

Last invalid ARP:
Time: 10:20:30 ( 5 minutes ago )
Reason: Bad IP/Mac match
Received on: VLAN 10
Packet:
  Source MAC: 00:01:00:01:00:01
  Dest MAC: 00:02:00:02:00:02
  ARP Type: Request
  ARP Sender MAC: 00:01:00:01:00:01
  ARP Sender IP: 1.1.1

switch(config)#
```

show ip dhcp relay

The **show ip dhcp relay** command displays the DHCP relay agent configuration status on the switch.

Command Mode

EXEC

Command Syntax

show ip dhcp relay

Example

- This command displays the DHCP relay agent configuration status.

```
switch>show ip dhcp relay
DHCP Relay is active
DHCP Relay Option 82 is disabled
DHCP Smart Relay is enabled
Interface: Vlan100
  DHCP Smart Relay is disabled
  DHCP servers: 10.4.4.4
switch>
```

show ip dhcp relay counters

The **show ip dhcp relay counters** command displays the number of DHCP packets received, forwarded, or dropped on the switch and on all interfaces enabled as DHCP relay agents.

Command Mode

EXEC

Command Syntax

show ip dhcp relay counters

Example

- This command displays the IP DHCP relay counter table.

```
switch>show ip dhcp relay counters

           |  Dhcp Packets  |
Interface | Rcvd Fwdd Drop |         Last Cleared
----------|----- ---- -----|---------------------
  All Req |  376  376    0 | 4 days, 19:55:12 ago
 All Resp |  277  277    0 |
          |                |
 Vlan1000 |    0    0    0 | 4 days, 19:54:24 ago
 Vlan1036 |  376  277    0 | 4 days, 19:54:24 ago

switch>
```

show ip dhcp snooping

The **show ip dhcp snooping** command displays the DHCP snooping configuration.

Command Mode

EXEC

Command Syntax

```
show ip dhcp snooping
```

Related Commands

- **ip dhcp snooping** globally enables DHCP snooping.
- **ip dhcp snooping vlan** enables DHCP snooping on specified VLANs.
- **ip dhcp snooping information option** enables insertion of option-82 snooping data.
- **ip helper-address** enables the DHCP relay agent on a configuration mode interface.

Example

- This command displays the switch's DHCP snooping configuration.

```
switch>show ip dhcp snooping
DHCP Snooping is enabled
DHCP Snooping is operational
DHCP Snooping is configured on following VLANs:
  100
DHCP Snooping is operational on following VLANs:
  100
Insertion of Option-82 is enabled
  Circuit-id format: Interface name:Vlan ID
  Remote-id: 00:1c:73:1f:b4:38 (Switch MAC)
switch>
```

show ip dhcp snooping counters

The **show ip dhcp snooping counters** command displays counters that track the quantity of DHCP request and reply packets that the switch receives. Data is either presented for each VLAN or aggregated for all VLANs with counters for packets dropped.

Command Mode

EXEC

Command Syntax

show ip dhcp snooping counters [***COUNTER_TYPE***]

Parameters

- ***COUNTER_TYPE*** The type of counter that the command resets. Formats include:
  - <no parameter> command displays counters for each VLAN.
  - **debug** command displays aggregate counters and drop cause counters.

Example

- This command displays the number of DHCP packets sent and received on each VLAN.

```
switch>show ip dhcp snooping counters

     | Dhcp Request Pkts | Dhcp Reply Pkts |
Vlan |  Rcvd  Fwdd  Drop | Rcvd Fwdd  Drop | Last Cleared
-----|------ ----- ------|----- ---- ------|-------------
 100 |     0     0     0 |    0    0     0 |  0:35:39 ago

switch>
```

- This command displays the number of DHCP packets sent on the switch.

```
switch>show ip dhcp snooping counters debug

Counter                       Snooping to Relay Relay to Snooping
----------------------------- ----------------- -----------------
Received                                      0                 0
Forwarded                                     0                 0
Dropped - Invalid VlanId                      0                 0
Dropped - Parse error                         0                 0
Dropped - Invalid Dhcp Optype                 0                 0
Dropped - Invalid Info Option                 0                 0
Dropped - Snooping disabled                   0                 0

Last Cleared:  3:37:18 ago
switch>
```

show ip dhcp snooping hardware

The **show ip dhcp snooping hardware** command displays internal hardware DHCP snooping status on the switch.

Command Mode

EXEC

Command Syntax

show ip dhcp snooping hardware

Example

- This command DHCP snooping hardware status.

```
switch>show ip dhcp snooping hardware
DHCP Snooping is enabled
DHCP Snooping is enabled on following VLANs:
    None
    Vlans enabled per Slice
        Slice:  FixedSystem
        None
switch>
```

show ip helper-address

The **show ip helper-address** command displays the status of DHCP relay agent parameters on the switch and each interface where at least one feature parameter is listed. The command provides status on the following parameters:

- Global: DHCP relay agent Always-on mode, DHCP relay agent Information option
- Interface: DHCP server (list of addresses), Circuit ID contents

Command Mode

EXEC

Command Syntax

show ip helper-address

Example

- This command displays the DHCP Agent Relay parameter status.

```
switch>show ip helper-address
DHCP Relay Agent Information Option Enabled
DHCP Relay Agent Always-On Mode Enabled
Interface: Vlan200
   Circuit ID: V-200
   DHCP servers: 10.3.31.14
switch>
```

show ip interface

The **show ip interface** command displays the status of specified interfaces that are configured as routed ports. The command provides the following information:

- Interface description
- Internet address
- Broadcast address
- Address configuration method
- Proxy-ARP status
- MTU size

Command Mode

EXEC

Command Syntax

```
show ip interface [INTERFACE_NAME][VRF_INST]
```

Parameters

- ***INTERFACE_NAME*** interfaces for which command displays status.
  - <no parameter> all routed interfaces.
  - *ipv4_addr* Neighbor IPv4 address.
  - **ethernet** *e_range* Routed Ethernet interfaces specified by *e_range*.
  - **loopback** *l_range* Routed loopback interfaces specified by *l_range*.
  - **management** *m_range* Routed management interfaces specified by *m_range*.
  - **port-channel** *p_range* Routed port channel Interfaces specified by *p_range*.
  - **vlan** *v_range* VLAN interfaces specified by *v_range*.
  - **vxlan** *vx_range* VXLAN interfaces specified by *vx_range*.
- ***VRF_INST*** specifies the VRF instance for which data is displayed.
  - <no parameter> context-active VRF.
  - **vrf** *vrf_name* specifies name of VRF instance. System default VRF is specified by **default**.

Example

- This command displays IP status of configured VLAN interfaces numbered between 900 and 910.

```
switch>show ip interface vlan 900-910
! Some interfaces do not exist
Vlan901 is up, line protocol is up (connected)
  Description: ar.pqt.mlag.peer
  Internet address is 170.23.254.1/30
  Broadcast address is 255.255.255.255
  Address determined by manual configuration
  Proxy-ARP is disabled
  MTU 9212 bytes
Vlan903 is up, line protocol is up (connected)
  Description: ar.pqt.rn.170.23.254.16/29
  Internet address is 170.23.254.19/29
  Broadcast address is 255.255.255.255
  Address determined by manual configuration
  Proxy-ARP is disabled
  MTU 9212 bytes
```

show ip interface brief

Use the **show ip interface brief** command output to display the status summary of the specified interfaces that are configured as routed ports. The command provides the following information for each specified interface:

- IP address
- Operational status
- Line protocol status
- MTU size

Command Mode

EXEC

Command Syntax

```
show ip interface [INTERFACE_NAME][VRF_INST] brief
```

Parameters

- ***INTERFACE_NAME*** interfaces for which command displays status.
  - <no parameter> all routed interfaces.
  - *ipv4_addr* Neighbor IPv4 address.
  - **ethernet** *e_range* Routed Ethernet interfaces specified by *e_range*.
  - **loopback** *l_range* Routed loopback interfaces specified by *l_range*.
  - **management** *m_range* Routed management interfaces specified by *m_range*.
  - **port-channel** *p_range* Routed port channel Interfaces specified by *p_range*.
  - **vlan** *v_range* VLAN interfaces specified by *v_range*.
  - **vxlan** *vx_range* VXLAN interface range specified by *vx_range*.
- ***VRF_INST*** specifies the VRF instance for which data is displayed.
  - <no parameter> context-active VRF.
  - **vrf** *vrf_name* specifies name of VRF instance. System default VRF is specified by **default**.

Example

- This command displays the summary status of VLAN interfaces 900-910

```
switch>show ip interface vlan 900-910 brief
! Some interfaces do not exist
Interface              IP Address         Status     Protocol         MTU
Vlan901                170.33.254.1/30    up         up              9212
Vlan902                170.33.254.14/29   up         up              9212
Vlan905                170.33.254.17/29   up         up              1500
Vlan907                170.33.254.67/29   up         up              9212
Vlan910                170.33.254.30/30   up         up              9212
```

show ip nat access-list interface

The **show ip nat acl interface** command displays the access control lists (ACLs) that are configured as source NAT or destination NAT filters. The display indicates ACL rules that do not comply with these NAT requirements:

- Source IP address is ***any***.
- Destination IP address may use any mask size.
- Source port matching is not allowed.
- Protocol matching is not allowed.

Command Mode

EXEC

Command Syntax

show ip nat access-list [*INTF*] [*LISTS*]

Parameters

- ***INTF*** Filters NAT statements by interface. Options include:
  - <no parameter> includes all statements on all interfaces.
  - **interface ethernet** *e_num* Statements on specified Ethernet interface.
  - **interface loopback** *l_num* Statements on specified Loopback interface.
  - **interface management** *m_num* Statements on specified Management interface.
  - **interface port-channel** *p_num* Statements on specified Port-Channel Interface.
  - **interface vlan** *v_num* Statements on specified VLAN interface.
  - **interface vxlan** *vx_num* Statements on specified VXLAN interface.
- ***LISTS*** ACLs displayed by command. Options include:
  - <no parameter> all ACLs.
  - *acl_name* Specifies individual ACL.

Example

- These commands display the NAT command usage of the ACL1 and ACL2 access control lists.

```
switch>show ip nat acl ACL1
acl ACL1
        (0.0.0.0/0, 168.10.1.1/32)
Interfaces using this ACL for Nat:
        Vlan100

switch>show ip nat acl ACL2
acl ACL2
        (168.10.1.1/32, 0.0.0.0/0)
Interfaces using this ACL for Nat:
        Vlan201
switch>
```

show ip nat pool

The **show ip nat pool** command displays the configuration of the address pool.

Command Mode

EXEC

Command Syntax

show ip nat pool *POOL_SET*

Parameters

- *pool_name* The name of the pool.
- ***POOL_SET*** Options include:
  - <no parameter> all configured port channels.
  - *pool_name* The name of the pool.

Example

- This command displays all the address pools configured on the switch.

```
switch#show ip nat pool
Pool                 StartIp              EndIp                Prefix
p1                   10.15.15.15          10.15.15.25          24
p2                   10.10.15.15          10.10.15.25          22
p3                   10.12.15.15          10.12.15.25          12
switch#
```

- These commands display specific information for the address pools configured on the switch.

```
switch#show ip nat pool p1
Pool                 StartIp              EndIp                Prefix
p1                   4.1.1.1              4.1.1.2              24
                     1.1.1.1              1.1.1.2              24
                     3.1.1.1              3.1.1.2              24
switch#show ip nat pool p2
Pool                 StartIp              EndIp                Prefix
p2                   10.1.1.1             10.1.1.2             16
switch#
```

show ip nat translations

The **show ip nat translations** command displays configured NAT statements in the switch hardware.

Command Mode

EXEC

Command Syntax

show ip nat translations [*INTF*][*ADDR*][*TYPE*][*DIR*][*H_STATE*][*K_STATE*][*V_STATE*]

Command position of ***INTF, ADDR***, ***TYPE***, and ***DIR*** parameters are interchangeable.

Parameters

- ***INTF*** Filters NAT statements by interface. Options include:
  - <no parameter> includes all statement on all interfaces.
  - **interface ethernet** *e_num* Statements on specified Ethernet interface.
  - **interface loopback** *l_num* Statements on specified Loopback interface.
  - **interface management** *m_num* Statements on specified Management interface.
  - **interface port-channel** *p_num* Statements on specified Port-Channel Interface.
  - **interface vlan** *v_num* Statements on specified VLAN interface.
- ***ADDR*** Filters NAT statements by status. Options include:
  - <no parameter> includes all NAT statements, including those not installed in hardware.
  - **address** *ipv4_addr* includes only NAT statements installed in hardware.
- ***TYPE*** Filters NAT statements by status. Options include:
  - <no parameter> includes all NAT statements, including those not installed in hardware.
  - **static** includes only NAT statements installed in hardware.
  - **dynamic** includes only NAT statements installed in hardware.
- ***DIR*** Filters NAT statements by status. Options include:
  - <no parameter> includes all NAT statements, including those not installed in hardware.
  - **source** includes only NAT statements installed in hardware.
  - **destination** includes only NAT statements installed in hardware.
- ***H_STATE*** Filters NAT statements by status. Options include:
  - <no parameter> includes all NAT statements, including those not installed in hardware.
  - **hardware** includes only NAT statements installed in hardware.
- ***K_STATE*** Filters NAT statements by status. Options include:
  - <no parameter> includes all NAT statements, including those not installed in hardware.
  - **kernel** includes only NAT statements installed in hardware.
- ***V_STATE*** Specifies information that the command returns. Options include:
  - <no parameter> displays table of NAT translations.
  - **detail** displays table of NAT translations.

Example

- This command displays all configured NAT translations.

```
switch#show ip nat translations
Source IP            Destination IP        Translated IP         TGT Type Intf
------------------------------------------------------------------------------
-
192.168.1.10:62822   172.22.22.40:53       172.17.254.161:62822  SRC DYN Vl3925
192.152.1.10:20342   172.22.22.40:80      172.17.254.161:22222  SRC STAT  Vl3945
switch#
```

show ip route

The **show ip route** command displays routing table entries that are in the Forwarding Information Base (FIB), including static routes, routes to directly connected networks, and dynamically learned routes. Multiple equal-cost paths to the same prefix are displayed contiguously as a block, with the destination prefix displayed only on the first line.

The **show running-config** command displays configured commands not in the FIB.

Command Mode

EXEC

Command Syntax

```
show ip route [VRF_INSTANCE][ADDRESS][ROUTE_TYPE][INFO_LEVEL][PREFIX]
```

Parameters

The ***VRF_INSTANCE*** and ***ADDRESS*** parameters are always listed first and second, respectively. All other parameters can be placed in any order.

- ***VRF_INSTANCE*** specifies the VRF instance for which data is displayed.
  - <no parameter> context-active VRF.
  - **vrf** *vrf_name* specifies name of VRF instance. System default VRF is specified by **default**.
- ***ADDRESS*** Filters routes by IPv4 address or subnet.
  - <no parameter> all routing table entries.
  - *ipv4_addr* routing table entries matching specified address.
  - *ipv4_subnet* routing table entries matching specified subnet (CIDR or address-mask).
- ***ROUTE_TYPE*** Filters routes by specified protocol or origin. Options include:
  - <no parameter> all routing table entries.
  - **aggregate** entries for BGP aggregate routes.
  - **bgp** entries added through BGP protocol.
  - **connected** entries for routes to networks directly connected to the switch.
  - **isis** entries added through ISIS protocol.
  - **kernel** entries appearing in Linux kernel but not added by EOS software.
  - **ospf** entries added through OSPF protocol.
  - **rip** entries added through RIP protocol.
  - **static** entries added through CLI commands.
- ***INFO_LEVEL*** Filters entries by next hop connection. Options include:
  - <no parameter> filters routes whose next hops are directly connected.
  - **detail** displays all routes.
- ***PREFIX*** filters routes by prefix.
  - <no parameter> specific route entry that matches the ADDRESS parameter.
  - **longer-prefixes** all subnet route entries in range specified by ADDRESS parameter.

Related Commands

- **routing-context vrf** specifies the context-active VRF.

Example

- This command displays IPv4 routes learned through BGP.

```
switch>show ip route bgp
Codes: C - connected, S - static, K - kernel,
       O - OSPF, IA - OSPF inter area, E1 - OSPF external type 1,
       E2 - OSPF external type 2, N1 - OSPF NSSA external type 1,
       N2 - OSPF NSSA external type2, B I - iBGP, B E - eBGP,
       R - RIP, A - Aggregate

 B E    170.44.48.0/23 [20/0] via 170.44.254.78
 B E    170.44.50.0/23 [20/0] via 170.44.254.78
 B E    170.44.52.0/23 [20/0] via 170.44.254.78
 B E    170.44.54.0/23 [20/0] via 170.44.254.78
 B E    170.44.254.112/30 [20/0] via 170.44.254.78
 B E    170.53.0.34/32 [1/0] via 170.44.254.78
 B I    170.53.0.35/32 [1/0] via 170.44.254.2
                             via 170.44.254.13
                             via 170.44.254.20
                             via 170.44.254.67
                             via 170.44.254.35
                             via 170.44.254.98
```

show ip route age

The **show ip route age** command displays the current state of the routing table and specifies the last time the route was updated.

Command Mode

EXEC

Command Syntax

show ip route *ADDRESS* age

Parameters

- ***ADDRESS*** Filters routes by IPv4 address or subnet.
  - *ipv4_addr* routing table entries matching specified address.
  - *ipv4_subnet* routing table entries matching specified subnet (CIDR or address-mask).

Example

- This command shows the amount of time since the last update to ip route 172.17.0.0/20.

```
switch>show ip route 172.17.0.0/20 age
Codes: C - connected, S - static, K - kernel,
       O - OSPF, IA - OSPF inter area, E1 - OSPF external type 1,
       E2 - OSPF external type 2, N1 - OSPF NSSA external type 1,
       N2 - OSPF NSSA external type2, B I - iBGP, B E - eBGP,
       R - RIP, I - ISIS, A - Aggregate

 B E    172.17.0.0/20 via 172.25.0.1, age 3d01h
switch>
```

show ip route gateway

The **show ip route gateway** command displays IP addresses of all gateways (next hops) used by active routes.

Command Mode

EXEC

Command Syntax

show ip route [*VRF_INSTANCE*] gateway

Parameters

- ***VRF_INSTANCE*** specifies the VRF instance for which data is displayed.
  - <no parameter> context-active VRF.
  - **vrf** *vrf_name* specifies name of VRF instance. System default VRF is specified by **default**.

Related Commands

- **routing-context vrf** specifies the context-active VRF.

Example

- This command displays next hops used by active routes.

```
switch>show ip route gateway
The following gateways are in use:
   172.25.0.1 Vlan101
   172.17.253.2 Vlan3000
   172.17.254.2 Vlan3901
   172.17.254.11 Vlan3902
   172.17.254.13 Vlan3902
   172.17.254.17 Vlan3903
   172.17.254.20 Vlan3903
   172.17.254.66 Vlan3908
   172.17.254.67 Vlan3908
   172.17.254.68 Vlan3908
   172.17.254.29 Vlan3910
   172.17.254.33 Vlan3911
   172.17.254.35 Vlan3911
   172.17.254.105 Vlan3912
   172.17.254.86 Vlan3984
   172.17.254.98 Vlan3992
   172.17.254.99 Vlan3992
switch>
```

show ip route host

The **show ip route host** command displays all host routes in the host forwarding table. Host routes are those whose destination prefix is the entire address (mask = 255.255.255.255 or prefix = /32). Each entry includes a code of the route's purpose:

- F static routes from the FIB.
- R routes defined because the IP address is an interface address.
- B broadcast address.
- A routes to any neighboring host for which the switch has an ARP entry.

Command Mode

EXEC

Command Syntax

show ip route [*VRF_INSTANCE*] host

Parameters

- ***VRF_INSTANCE*** specifies the VRF instance for which data is displayed.
  - <no parameter> context-active VRF.
  - **vrf** *vrf_name* specifies name of VRF instance. System default VRF is specified by **default**.

Related Commands

- **routing-context vrf** specifies the context-active VRF.

Example

- This command displays all host routes in the host forwarding table.

```
switch>show ip route host
R - receive B - broadcast F - FIB, A - attached

F   127.0.0.1 to cpu
B   172.17.252.0 to cpu
A   172.17.253.2 on Vlan2000
R   172.17.253.3 to cpu
A   172.17.253.10 on Vlan2000
B   172.17.253.255 to cpu
B   172.17.254.0 to cpu
R   172.17.254.1 to cpu
B   172.17.254.3 to cpu
B   172.17.254.8 to cpu
A   172.17.254.11 on Vlan2902
R   172.17.254.12 to cpu

F   172.26.0.28 via 172.17.254.20 on Vlan3003
                via 172.17.254.67 on Vlan3008
                via 172.17.254.98 on Vlan3492
                via 172.17.254.2 on Vlan3601
                via 172.17.254.13 on Vlan3602
via 172.17.253.2 on Vlan3000
F   172.26.0.29 via 172.25.0.1 on Vlan101
F   172.26.0.30 via 172.17.254.29 on Vlan3910
F   172.26.0.32 via 172.17.254.105 on Vlan3912
switch>
```

show ip route summary

The show ip route summary command displays the number of routes, categorized by destination prefix, in the routing table.

Command Mode

EXEC

Command Syntax

show ip route [*VRF_INSTANCE*] summary

Parameters

- ***VRF_INSTANCE*** specifies the VRF instance for which data is displayed.
  - <no parameter> context-active VRF.
  - **vrf** *vrf_name* specifies name of VRF instance. System default VRF is specified by **default**.

Example

- This command displays a summary of the routing table contents.

```
switch>show ip route summary
Route Source         Number Of Routes
-------------------------------------
connected                    15
static                        0
ospf                         74
  Intra-area: 32 Inter-area:33 External-1:0 External-2:9
  NSSA External-1:0 NSSA External-2:0
bgp                           7
  External: 6 Internal: 1
internal                     45
attached                     18
aggregate                     0
switch>
```

show ip route tag

The **show ip route tag** command displays the route tag assigned to the specified IPv4 address or subnet. Route tags are added to static routes for use by route maps.

Command Mode

EXEC

Command Syntax

show ip route [*VRF_INSTANCE*] *ADDRESS* tag

Parameters

- ***VRF_INSTANCE*** specifies the VRF instance for which data is displayed.
  - <no parameter> context-active VRF.
  - **vrf** *vrf_name* specifies name of VRF instance. System default VRF is specified by **default**.
- ***ADDRESS*** displays routes of specified IPv4 address or subnet.
  - *ipv4_addr* routing table entries matching specified IPv4 address.
  - *ipv4_subnet* routing table entries matching specified IPv4 subnet (CIDR or address-mask).

Example

- This command displays the route tag for the specified subnet.

```
switch>show ip route 172.17.50.0/23 tag
Codes: C - connected, S - static, K - kernel,
       O - OSPF, IA - OSPF inter area, E1 - OSPF external type 1,
       E2 - OSPF external type 2, N1 - OSPF NSSA external type 1,
       N2 - OSPF NSSA external type2, B I - iBGP, B E - eBGP,
       R - RIP, A - Aggregate

 O E2   172.17.50.0/23 tag 0

switch>
```

show ip verify source

The **show ip verify source** command displays the IP source guard (IPSG) configuration, operational states, and IP-MAC binding entries for the configuration mode interface.

Command Mode

EXEC

Command Syntax

show ip verify source [*VLAN* | *DETAIL*]

Parameters

- ***VLAN*** displays all VLANs configured in **no ip verify source vlan**.
- ***DETAIL*** displays all source IP-MAC binding entries configured for IPSG.

Related Commands

- **ip source binding**
- **ip verify source**

Example

This command verifies the IPSG configuration and operational states.

```
switch(config)#show ip verify source
Interface       Operational State
--------------- ------------------------
Ethernet1       IP source guard enabled
Ethernet2       IP source guard disabled
```

Example

This command displays all VLANs configured in **no ip verify source vlan**. Hardware programming errors, e.g.,VLAN classification failed, are indicated in the operational state. If an error occurs, this VLAN will be considered as enabled for IPSG. Traffic on this VLAN will still be filtered by IPSG.

```
switch(config)#show ip verify source vlan
IPSG disabled on VLANS: 1-2
VLAN            Operational State
--------------- ------------------------
1               IP source guard disabled
2               Error: vlan classification failed
```

Example

This command displays all source IP-MAC binding entries configured for IPSG. A source binding entry is considered active if it is programmed in hardware. IP traffic matching any active binding entry will be permitted. If a source binding entry is configured on an interface or a VLAN whose operational state is IPSG disabled, this entry will not be installed in the hardware, in which case an “IP source guard disabled” state will be shown. If a port channel has no member port configured, binding entries configured for this port channel will not be installed in hardware, and a “Port-Channel down” state will be shown.

```
switch(config)#show ip verify source detail
Interface       IP Address    MAC Address      VLAN   State
--------------- ------------ ---------------- ------ ------------------------
Ethernet1       10.1.1.1      0000.aaaa.1111   5      active
Ethernet1       10.1.1.5      0000.aaaa.5555   1      IP source guard disabled
Port-Channel1   20.1.1.1      0000.bbbb.1111   4      Port-Channel down
```

show platform arad ip route

The **show platform arad ip route** command shows resources for all IPv4 routes in hardware. Routes that use the additional hardware resources will appear with an asterisk.

Command Mode

EXEC

Command Syntax

```
show platform arad ip route
```

Related Commands

- **agent SandL3Unicast terminate** enables restarting the layer 3 agent to ensure IPv4 routes are optimized.
- **ip hardware fib optimize** enables IPv4 route scale.
- **show platform arad ip route summary** shows hardware resource usage of IPv4 routes.

Examples

- This command shows resources for all IPv4 routes in hardware. Routes that use the additional hardware resources will appear with an asterisk.

```
switch(config)#show platform arad ip route
Tunnel Type: M(mpls), G(gre)
* - Routes in LEM

-------------------------------------------------------------------------------
----------------
|                        Routing Table                                |          |
|-------------------------------------------------------------------------------
------------------
|VRF|  Destination   |     |                  |    |Acl  |                |ECMP
| FEC | Tunnel
|ID |    Subnet      | Cmd |    Destination    |VID |Label| MAC / CPU Code
|Index|Index|T Value

-------------------------------------------------------------------------------
----------------
|0  |0.0.0.0/8      |TRAP |CoppSystemL3DstMiss|0   | -   |ArpTrap         |  -
|1030 |   -
|0  |100.1.0.0/32   |TRAP |CoppSystemIpBcast  |0   | -   |BcastReceive    |  -
|1032 |   -
|0  |100.1.0.0/32   |TRAP |CoppSystemIpUcast  |0   | -   |Receive         |  -
|32766|   -
|0  |100.1.255.255/32|TRAP |CoppSystemIpBcast  |0   | -   |BcastReceive    |  -
|1032 |   -
|0  |200.1.255.255/32|TRAP |CoppSystemIpBcast  |0   | -   |BcastReceive    |  -
|1032 |   -
|0  |200.1.0.0/16   |TRAP |CoppSystemL3DstMiss|1007| -   |ArpTrap         |  -
|1029 |   -
|0  |0.0.0.0/0      |TRAP |CoppSystemL3LpmOver|0   | -   |SlowReceive     |  -
|1024 |   -
|0  |4.4.4.0/24*    |ROUTE|Et10               |1007| -   |00:01:00:02:00:03|  -
|1033 |   -
|0  |10.20.30.0/24* |ROUTE|Et9                |1006| -   |00:01:00:02:00:03|  -
|1027 |   -

switch(config)#
```

show platform arad ip route summary

The **show platform arad ip route summary** command shows hardware resource usage of IPv4 routes.

Command Mode

EXEC

Command Syntax

```
show platform arad ip route summary
```

Related Commands

- **agent SandL3Unicast terminate** enables restarting the layer 3 agent to ensure IPv4 routes are optimized.
- **ip hardware fib optimize** enables IPv4 route scale.
- **show platform arad ip route** shows resources for all IPv4 routes in hardware. Routes that use the additional hardware resources will appear with an asterisk.

Example

- This command shows hardware resource usage of IPv4 routes.

```
switch(config)#show platform arad ip route summary
Total number of VRFs: 1
Total number of routes: 25
Total number of route-paths: 21
Total number of lem-routes: 4

switch(config)#
```

show platform trident forwarding-table partition

The **show platform trident forwarding-table partition** command displays the size of the L2 MAC entry tables, L3 IP forwarding tables, and Longest Prefix Match (LPM) routes.

Command Mode

Privileged EXEC

Command Syntax

```
show platform trident forwarding-table partition
```

Example

- This command shows the Trident forwarding table information.

```
switch(config)#show platform trident forwarding-table partition
L2 Table Size: 96k
L3 Host Table Size: 208k
LPM Table Size: 16k
switch(config)#
```

show routing-context vrf

The **show routing-context vrf** command displays the context-active VRF. The context-active VRF determines the default VRF that VRF-context aware commands use when displaying routing table data from a specified VRF.

Command Mode

EXEC

Command Syntax

```
show routing-context vrf
```

Related Commands

- **routing-context vrf** specifies the context-active VRF.

Example

- This command displays the context-active VRF.

```
switch>show routing-context vrf
Current VRF routing-context is PURPLE
switch>
```

show vrf

The **show vrf** command displays the VRF name, RD, supported protocols, state and included interfaces for the specified VRF or for all VRFs on the switch.

Command Mode

EXEC

Command Syntax

show vrf [*VRF_INSTANCE*]

Parameters

- ***VRF_INSTANCE*** specifies the VRF instance to display.
  - <no parameter> information is displayed for all VRFs.
  - **vrf** *vrf_name* information is displayed for the specified user-defined VRF.

Example

- This command displays information for the VRF named “purple.”

```
switch>show vrf purple
   Vrf          RD              Protocols       State         Interfaces
------------ --------------- --------------- ---------------- --------------
   purple       64496:237       ipv4            no routing    Vlan42, Vlan43

switch>
```

vrf definition

The **vrf definition** command places the switch in VRF configuration mode for the specified VRF. If the named VRF does not exist, this command creates it. The number of user-defined VRFs supported varies by platform.

To add an interface to the VRF once it is created, use the **vrf forwarding** command.

The **no vrf definition** and **default vrf definition** commands delete the specified VRF instance by removing the corresponding **vrf definition** command from ***running-config***. This also removes all IP addresses associated with interfaces that belong to the deleted VRF.

The **exit** command returns the switch to global configuration mode.

Command Mode

Global Configuration

Command Syntax

```
vrf definition vrf_name
no vrf definition vrf_name
default vrf definition vrf_name
```

Parameters

- *vrf_name* Name of VRF being created, deleted or configured. The names “main” and “default” are reserved.

Commands Available in VRF Configuration Mode

- **rd (VRF configuration mode)**

Example

- This command creates a VRF named “purple” and places the switch in VRF configuration mode for that VRF.

```
switch(config)#vrf definition purple
switch(config-vrf-purple)#
```

vrf forwarding

The **vrf forwarding** command adds the configuration mode interface to the specified VRF. You must create the VRF first, using the **vrf definition** command.

The **no vrf forwarding** and **default vrf forwarding** commands remove the configuration mode interface from the specified VRF by deleting the corresponding **vrf forwarding** command from ***running-config***.

All forms of the **vrf forwarding** command remove all IP addresses associated with the configuration mode interface.

Command Mode

Interface-Ethernet Configuration
Interface-Loopback Configuration
Interface-Management Configuration
Interface-Port-channel Configuration
Interface-VLAN Configuration

Command Syntax

```
vrf forwarding vrf_name
no vrf forwarding [vrf_name]
default vrf forwarding [vrf_name]
```

Parameters

- *vrf_name* name of configured VRF.

Examples

- These commands add the configuration mode interface (VLAN 20) to the VRF named “purple”.

```
switch(config)#interface vlan 20
switch(config-if-Vl20)#vrf forwarding purple
switch(config-if-Vl20)#
```

- These commands remove the configuration mode interface from VRF “purple”.

```
switch(config)#interface vlan 20
switch(config-if-Vl20)#no vrf forwarding purple
switch(config-if-Vl20)#
```

Chapter 25

IPv6

Arista switches support Internet Protocol version 4 (IPv4) and Internet Protocol version 6 (IPv6) for routing packets across network boundaries. This chapter describes Arista's implementation of IPv6 and includes these sections:

- Section 25.1: Introduction
- Section 25.2: IPv6 Description
- Section 25.3: Configuring IPv6
- Section 25.4: IPv6 Command Descriptions

25.1 Introduction

Routing transmits network layer data packets over connected independent subnets. Each subnet is assigned an IP address range and each device on the subnet is assigned an IP address from that range.

Connected subnets have IP address ranges that do not overlap. A router is a network device connecting multiple subnets. Routers forward inbound packets to the subnet whose address range includes the packets' destination address.

IPv4 and IPv6 are internet layer protocols that define packet-switched inter-networking, including source-to-destination datagram transmission across multiple networks. The switch supports IP Version 4 (IPv4) and IP Version 6 (IPv6).

IPv6 is described by RFC 2460: Internet Protocol, Version 6 (IPv6) Specification. RFC 2463 describes ICMPv6 for IPv6. ICMPv6 is a core protocol of the Internet Protocol suite.

25.2 IPv6 Description

Internet Protocol Version 6 is a communications protocol used for relaying network packets across a set of connected networks using the Internet Protocol suite. Each network device is assigned a 128 bit IP address that identifies its network location.

IPv6 specifies a packet format that minimizes router processing of packet headers. Since the IPv4 and IPv6 packet headers differ significantly, the protocols are not interoperable. Many transport and application-layer protocols require little or no change to operate over IPv6.

25.2.1 IPv6 Address Format

IPv6 addresses have 128 bits, represented by eight 16-bit hexadecimal numbers separated by colons. IPv6 addresses are abbreviated as follows:

- Leading zeroes in each 16-bit number may be omitted.
- One set of consecutive 16-bit numbers that equal zero may be replaced by a double colon.

Example

- The following three IPv6 hexadecimal number representations refer to the same address:

```
d28e:0000:0000:0000:0234:812f:61ed:4419
d28e:0:0:0:234:812f:61ed:4419
d28e::234:812f:61ed:4419
```

IPv6 addresses typically denote a 64-bit network prefix and a 64-bit host address.

Unicast and Anycast Addressing

Unicast addressing defines a one-to-one association between the destination address and a network endpoint. Each destination address uniquely identifies a single receiver endpoint. Anycast addressing defines a one-to-one-of-many association: packets to a single member of a group of potential receivers identified by the same destination address.

Unicast and anycast addresses are typically composed as follows:

- a 64-bit network prefix that identifies the network segment.
- a 64-bit interface identifier that is based on interface MAC address.

The format of a network address identifies the scope of the address

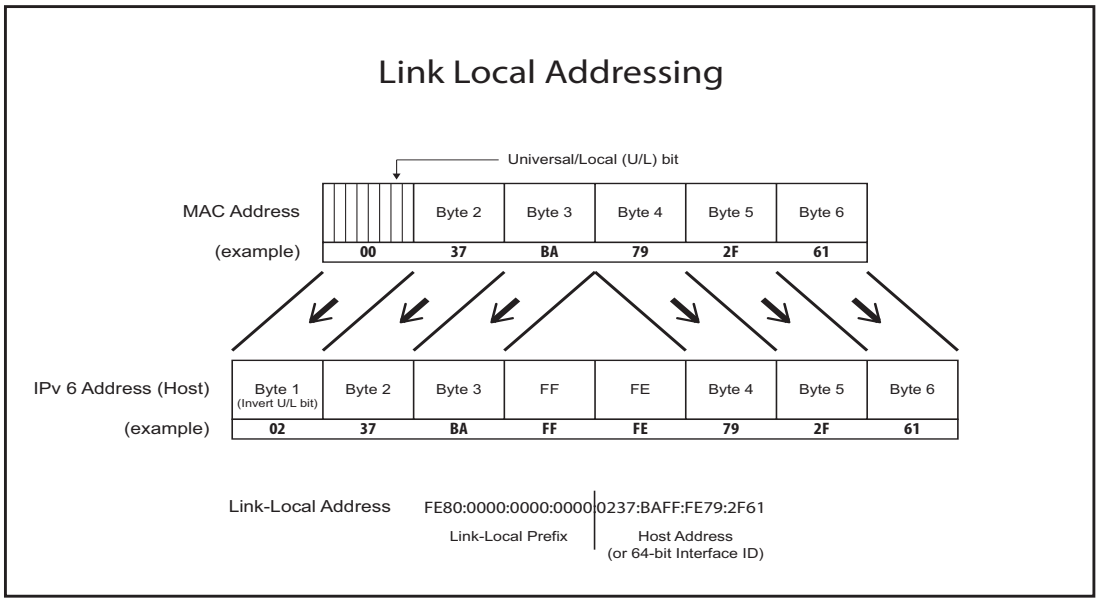
- Global address: valid in all networks and connect with other addresses with global scope anywhere or to addresses with link-local scope on the directly attached network.
- Link-local address: scope extends only to the link to which the interface is directly connected. Link-local addresses are not routable off the link.

  Link-local addresses are created by the switch and are not configurable. Figure 25-1 depicts the switch's link local address derivation method.

Multicast Addressing

Multicast addressing defines a one-to-many association: packets are simultaneously routed from a single sender to multiple endpoints in a single transmission. The network replicates packets as required by network links that contain a recipient endpoint. One multicast address is assigned to an interface for each multicast group to which the interface belongs.

Figure 25-1: Link Local Address Derivation



A solicited-node multicast address is an IPv6 multicast address whose scope extends only to the link to which the interface is directly connected. All IPv6 hosts have at least one such address per interface. Solicited-node multicast addresses are used by the Neighbor Discovery Protocol to obtain layer 2 link-layer addresses of other nodes.

25.2.2 Neighbor Discovery Protocol

The Neighbor Discovery Protocol (RFC 4861) operates with IPv6 to facilitate the following tasks for nodes within a specified prefix space:

- autoconfiguring a node's IPv6 address
- sensing other nodes on the link
- discovering the link-local addresses of other nodes on the link
- detecting duplicate addresses
- discovering available routers
- discovering DNS servers
- discovering the link's address prefix
- maintaining path reachability data to other active neighbor nodes

The Neighbor Discovery Protocol protocol defines five different ICMPv6 packet types:

- Router Solicitation
- Router Advertisement
- Neighbor Solicitation
- Neighbor Advertisement
- Redirect

25.3 Configuring IPv6

These sections describe IPv6 configuration tasks:

- Section 25.3.1: Configuring IPv6 on the Switch
- Section 25.3.2: Configuring IPv6 on an Interface
- Section 25.3.3: Viewing IPv6 Network Components
- Section 25.3.4: DHCP Relay Agent for IPv6

25.3.1 Configuring IPv6 on the Switch

25.3.1.1 Enabling IPv6 Unicast Routing on the Switch

The **ipv6 unicast-routing** command enables the forwarding of IPv6 unicast packets. When routing is enabled, the switch attempts to deliver inbound packets to destination addresses by forwarding them to interfaces or next hop addresses specified by the IPv6 routing table.

Example

- This command enables IPv6 unicast-routing.

```
switch(config)#ipv6 unicast-routing
switch(config)#
```

25.3.1.2 Configuring Default and Static IPv6 Routes

The **ipv6 route** command creates an IPv6 static route. The destination is a IPv6 prefix; the source is an IPv6 address or a routable interface port. When multiple routes exist to a destination prefix, the route with the lowest administrative distance takes precedence.

By default, the administrative distance assigned to static routes is 1. Assigning a higher administrative distance to a static route configures it to be overridden by dynamic routing data. For example, a static route with a distance value of 200 is overridden by OSPF intra-area routes, which have a default distance of 110.

Example

- This command creates an IPv6 static route.

```
switch(config)#ipv6 route 10:23:31:00:01:32:93/24 vlan 300
switch(config)#
```

The default route denotes the packet forwarding rule that takes effect when no other route is configured for a specified IPv6 address. All packets with destinations that are not established in the routing table are sent to the destination specified by the default route.

The IPv6 default route source is ::/0. The default route destination is referred to as the default gateway.

Example

- This command creates a default route and establishes fd7a:629f:52a4:fe61::2 as the default gateway address.

```
switch(config)#ipv6 route ::/0 fd7a:629f:52a4:fe61::2
switch(config)#
```

25.3.1.3 IPv6 ECMP

Multiple routes that are configured to the same destination with the same administrative distance comprise an Equal Cost Multi-Path (ECMP) route. The switch attempts to spread outbound traffic across all ECMP route paths equally. All ECMP paths are assigned the same tag value; commands that change the tag value of any ECMP path change the tag value of all paths in the ECMP.

Resilient ECMP is available for IPv6 routes. Section 24.4.2 describes resilient ECMP. The **ipv6 hardware fib ecmp resilience** command implements IPv6 resilient ECMP.

Example

- This command implements IPv6 resilient ECMP by configuring a hardware ECMP table space of 15 entries for IPv6 address 2001:db8:0::/64. A maximum of five nexthop addresses can be specified for the address. When the table contains five addresses, each appears in the table three times. When the table contains fewer than five addresses, each is duplicated until the 15 table entries are filled.

```
switch(config)#ipv6 hardware fib ecmp resilience 2001:db8:0::/64 capacity 5
redundancy 3
switch(config)#
```

25.3.2 Configuring IPv6 on an Interface

25.3.2.1 Enabling IPv6 on an Interface

The **ipv6 enable** command enables IPv6 on the configuration mode interface if it does not have a configured IPv6 address. It also configures the interface with an IPv6 address.

The **no ipv6 enable** command disables IPv6 on a configuration mode interface not configured with an IPv6 address. Interfaces configured with an IPv6 address are not disabled by this command.

Example

- This command enables IPv6 on VLAN interface 200.

```
switch(config)#interface vlan 200
switch(config-vl200)#ipv6 enable
switch(config-vl200)#
```

25.3.2.2 Assigning an IPv6 Address to an Interface

The **ipv6 address** command enables IPv6 on the configuration mode interface, assigns a global IPv6 address to the interface, and defines the prefix length. This command is supported on routable interfaces. Multiple global IPv6 addresses can be assigned to an interface.

Example

- These commands configure an IPv6 address with subnet mask for VLAN 200:

```
switch(config)#interface vlan 200
switch(config-if-Vl200)#ipv6 address 10:23:31:00:01:32:93/24
switch(config-if-Vl200)#
```

25.3.2.3 IPv6 Neighbor Discovery

The IPv6 Neighbor Discovery protocol defines a method for nodes to perform the following network maintenance tasks:

- determine layer 2 addresses for neighbors known to reside on attached links

- detect changed layer 2 addresses
- purge invalid values from the neighbor cache table
- (hosts) find neighboring routers to forward packets
- track neighbor reachability status

IPv6 Neighbor Discovery is defined by RFC 2461. IPv6 Stateless Address Autoconfiguration is described by RFC 2462.

The following sections describe Neighbor Discovery configuration tasks.

Reachable Time

The **ipv6 nd reachable-time** command specifies the time period that the switch includes in the reachable time field of Router Advertisements (RAs) sent from the configuration mode interface. The reachable time defines the period that a remote IPv6 node is considered reachable after a reachability confirmation event.

Example

- These commands configure the entry of 25000 (2**5** seconds) in the reachable time field of RAs sent from VLAN 200.

```
switch(config)#interface vlan 200
switch(config-if-Vl200)#ipv6 nd reachable-time 25000
switch(config-if-Vl200)#show active
interface Vlan200
   ipv6 address fd7a:4321::1/64
   ipv6 nd reachable-time 25000
switch(config-if-Vl200)#
```

Router Advertisement Interval

The **ipv6 nd ra interval** command configures the interval between IPv6 RA transmissions from the configuration mode interface.

Example

- These commands configure a RA transmission interval of 60 seconds on VLAN interface 200, then displays the interface status.

```
switch(config)#interface vlan 200
switch(config-if-Vl200)#ipv6 nd ra interval 60
switch(config-if-Vl200)#show active
interface Vlan200
   ipv6 nd ra interval 60
switch(config-if-Vl200)#
```

Router Lifetime

The **ipv6 nd ra lifetime** command specifies the value that the switch places in the ***router lifetime*** field of IPv6 RAs sent from the configuration mode interface.

If the value is set to 0, IPv6 peers connected to the specified interface will remove the switch from their lists of default routers. Values greater than 0 indicate the time in seconds that peers should keep the router on their default router lists without receiving further RAs from the switch. Unless the value is 0, the router lifetime value should be equal to or greater than the interval between unsolicited RAs sent on the interface.

Example

- This command configures the switch to enter 2700 in the router lifetime field of RAs tranmitted from VLAN 200.

```
switch(config)#interface vlan 200
switch(config-if-Vl200)#ipv6 nd ra lifetime 2700
switch(config-if-Vl200)#show active
interface Vlan200
   ipv6 nd ra lifetime 2700
switch(config-if-Vl200)#
```

Router Advertisement Prefix

The **ipv6 nd prefix** command configures neighbor discovery router advertisement prefix inclusion for RAs sent from the configuration mode interface.

By default, all prefixes configured as IPv6 addresses are advertised in the interface's RAs. The **ipv6 nd prefix** command with the ***no-advertise*** option prevents advertising of the specified prefix without affecting the advertising of other prefixes specified as IPv6 addresses. When an interface configuration includes at least one **ipv6 nd prefix** command that enables prefix advertising, RAs advertise only prefixes specified through **ipv6 nd prefix** commands.

Commands enabling prefix advertising also specify the advertised valid and preferred lifetime periods. Default periods are 2,592,000 (valid) and 604,800 (preferred) seconds.

Example

- These commands enable neighbor discovery advertising for IPv6 address 3012:D678::/64, specifying a valid lifetime of 1,296,000 seconds and the default preferred lifetime.

```
switch(config)#interface vlan 200
switch(config-if-Vl200)#ipv6 nd prefix 3012:D678::/64 1296000
switch(config-if-Vl200)#
```

Router Advertisement Suppression

The **ipv6 nd ra suppress** command suppress IPv6 RA transmissions on the configuration mode interface. By default, only unsolicited RAs that are transmitted periodically are suppressed. The **all** option configures the switch to suppress all RAs, including those responding to a router solicitation.

Example

- This command suppresses all RAs on VLAN interface 200.

```
switch(config)#interface vlan 200
switch(config-vl200)#ipv6 nd ra suppress all
switch(config-vl200)#
```

Router Advertisement MTU Suppression

The **ipv6 nd ra mtu suppress** command suppresses the router advertisement MTU option on the configuration mode interface. The MTU option causes an identical MTU value to be advertised by all nodes on a link. By default, the router advertisement MTU option is not suppressed.

Example

- This command suppresses the MTU option on VLAN interface 200.

```
switch(config)#interface vlan 200
switch(config-vl200)#ipv6 nd ra mtu suppress
switch(config-vl200)#
```

Router Advertisment Flag Configuration

The following commands sets the specified configuration flag in IPv6 RAs transmitted from the configuration mode interface:

- The **ipv6 nd managed-config-flag** command command sets the ***managed address configuration*** flag. This bit instructs hosts to use stateful address autoconfiguration.
- The **ipv6 nd other-config-flag** command sets the ***other stateful configuration*** flag. This bit indicates availability of autoconfiguration information, other than addresses. Hosts should use stateful autoconfiguration when available. The setting of this flag has no effect if the ***managed address configuration*** flag is set.
- These commands configure the switch to set the ***managed address configuration*** flag in advertisements sent from VLAN interface 200.

```
switch(config)#interface vlan 200
switch(config-if-Vl200)#ipv6 nd managed-config-flag
switch(config-if-Vl200)#
```

- These commands configure the switch to set the ***other stateful configuration*** flag in advertisements sent from VLAN interface 200.

```
switch(config)#interface vlan 200
switch(config-if-Vl200)#ipv6 nd other-config-flag
switch(config-if-Vl200)#
```

25.3.2.4 IPv6 Router Preference

The IPv6 Router Preference protocol supports an extension to RA messages for communicating default router preferences and more specific routes from routers to hosts. This provides assistance to hosts when selecting a router. RFC 4191 describes the IPv6 Router Preference Protocol.

The **ipv6 nd router-preference** command specifies the value that the switch enters in the Default Router Preference (DRP) field of RAs that it sends from the configuration mode interface. The default field entry value is ***medium***.

Example

- This command configures the switch as a medium preference router on RAs sent from VLAN 200.

```
switch(config)#interface vlan 200
switch(config-if-Vl200)#ipv6 nd router-preference medium
switch(config-if-Vl200)#
```

25.3.2.5 uRPF Configuration

Unicast Reverse Path Forwarding (uRPF) verifies the accessibility of source IP addresses in packets that the switch forwards. Section 24.4.3 describe uRPF. uRPF is enabled for IPv6 packets ingressing the configuration mode interface through the **ipv6 verify** command.

uRPF defines two operational modes: strict mode and loose mode.

- Strict mode: uRPF verifies that a packet is received on the interface that its routing table entry specifies for its return packet.
- Loose mode: uRPF validation does not consider the inbound packet's ingress interface only that there is a valid return path.

Example

- This command enables uRPF strict mode on VLAN interface 100. If a default route is configured on the interface, all inbound packets will pass the uRPF check as valid.

```
switch(config)#interface vlan 100
switch(config-if-Vl100)#ipv6 verify unicast source reachable-via rx
allow-default
switch(config-if-Vl100)#show active
 interface Vlan100
   ipv6 verify unicast source reachable-via rx allow-default
switch(config-if-Vl100)#
```

25.3.3 Viewing IPv6 Network Components

Displaying the FIB and Routing Table

The **show ipv6 route** command displays routing table entries that are in the Forwarding Information Base (FIB), including static routes, routes to directly connected networks, and dynamically learned routes. Multiple equal cost paths to the same prefix are displayed contiguously as a block, with the destination prefix displayed only on the first line.

Example

- This command displays a route table entry for a specific IPv6 route.

```
switch>show ipv6 route fd7a:3418:52a4:fe18::/64
IPv6 Routing Table - 77 entries
Codes: C - connected, S - static, K - kernel, O - OSPF, B - BGP, R - RIP, A -
Aggregate

O   fd7a:3418:52a4:fe18::/64 [10/20]
     via f180::21c:73ff:fe00:1319, Vlan3601
     via f180::21c:73ff:fe00:1319, Vlan3602
     via f180::21c:73ff:fe00:1319, Vlan3608
     via f180::21c:73ff:fe0f:6a80, Vlan3610
     via f180::21c:73ff:fe00:1319, Vlan3611

switch>
```

Displaying the Route Age

The **show ipv6 route age** command displays the IPv6 route age to the specified IPv6 address or prefix.

Example

- This command displays the route age for the specified prefix.

```
switch>show ipv6 route 2001::3:0/11 age
IPv6 Routing Table - 74 entries
Codes: C - connected, S - static, K - kernel, O - OSPF, B - BGP, R - RIP, A -
Aggregate

C 2001::3:0/11 age 00:02:34
switch>
```

Displaying Host Routes

The **show ipv6 route host** command displays all host routes in the IPv6 host forwarding table. Host routes are those whose destination prefix is the entire address (prefix = /128). Each displayed host route is labeled with its purpose:

- F static routes from the FIB.
- R routes defined because the IP address is an interface address.
- A routes to any neighboring host for which the switch has an ARP entry.

Example

- This command displays all IPv6 host routes in the host forwarding table.

```
switch#show ipv6 route host
R - receive F - FIB, A - attached

 F  ::1 to cpu
 A  fee7:48a2:0c11:1900:400::1 on Vlan102
 R  fee7:48a2:0c11:1900:400::2 to cpu
 F  fee7:48a2:0c11:1a00::b via fe80::21c:73ff:fe0b:a80e on Vlan3902
 R  fee7:48a2:0c11:1a00::17 to cpu
 F  fee7:48a2:0c11:1a00::20 via fe80::21c:73ff:fe0b:33e on Vlan3913
 F  fee7:48a2:0c11:1a00::22 via fe80::21c:73ff:fe01:5fe1 on Vlan3908
                            via fe80::21c:73ff:fe01:5fe1 on Vlan3902

switch#
```

Displaying Route Summaries

The **show ipv6 route summary** command displays the current number of routes of the IPv6 routing table in summary format.

Example

- This command displays the route source and the corresponding number of routes in the IPv6 routing table.

```
switch>show ipv6 route summary
   Route Source    Number Of Routes
------------------ ---------------
   connected                    2
   static                       0
   ospf                         5
   bgp                          7
   isis                         0
   internal                     1
   attached                     0
   aggregate                    2

   Total Routes                 17
switch>
```

25.3.4 DHCP Relay Agent for IPv6

25.3.4.1 Configuring IPv6 DHCP Relay

Configuring the IPv6 DHCP Relay Agent (Global)

The **ipv6 dhcp relay always-on** command enables the switch DHCP relay agent globally regardless of the DHCP relay agent status on any interface. The DHCP relay agent is enabled by default if at least one routable interface is configured with an **ipv6 dhcp relay destination** statement.

Example

- This command enables the DHCP relay agent.

```
switch(config)#ipv6 dhcp relay always-on
switch(config)#
```

Configuring DHCP for IPv6 relay agent

The **ipv6 dhcp relay destination** command enables the DHCPv6 relay agent function and specifies the client message destination address on an interface.

Example

- This command enables the DHCPv6 relay agent function and sets the client message destination address to **2001:0db8:0:1::1** on Ethernet interface 4.

```
switch(config)interface ethernet 4
switch(config-if-Et4)#ipv6 dhcp relay destination 2001:0db8:0:1::1
```

Clearing IPv6 DHCP Relay Counters

The **clear ipv6 dhcp relay counters** command resets the DHCP relay counters. The configuration mode determines which counters are reset:

- Global configuration: command clears the counters for the switch and for all interfaces.
- Interface configuration: command clears the counter for the configuration mode interface.

Example

- These commands clear all DHCP relay counters on the switch.

```
switch(config-if-Et4)#exit
switch(config)#clear ipv6 dhcp relay counters
switch(config)#
```

- These commands clear the DHCP relay counters for Ethernet interface 4.

```
switch(config)#interface ethernet 4
switch(config-if-Et4)#clear ipv6 dhcp relay counters
switch(config)#
```

25.3.4.2 Viewing IPv6 DHCP Relay Information

IPv6 DHCP Status

The **show ipv6 helper-address** command displays the status of DHCP relay agent parameters on the switch and each interface where at least one feature parameter is listed. The command displays the status for both global and interface configurations.

Example

- This command displays the DHCP Agent Relay parameter status.

```
switch>show ipv6 helper-address
DHCP Relay Agent Information Option Enabled
DHCP Relay Agent Always-On Mode Enabled
Interface: Ethernet4
   Circuit ID: V-200
   DHCP servers: 2001:db8:0:1::1
switch>
```

IPv6 DHCP Relay Counters

The **show ipv6 dhcp relay counters** command displays the number of DHCP packets received, forwarded, or dropped on the switch and on all interfaces enabled as DHCP relay agents.

Example

- This command displays the IP DHCP relay counter table.

```
switch>show ipv6 dhcp relay counters

          |  Dhcp Packets  |
Interface | Rcvd Fwdd Drop |         Last Cleared
----------|----- ---- -----|---------------------
  All Req |  376  376    0 | 4 days, 19:55:12 ago
 All Resp |  277  277    0 |
          |                |
Ethernet4 |  207  148    0 | 4 days, 19:54:24 ago

switch>
```

25.4 IPv6 Command Descriptions

Global Configuration Commands

- ipv6 dhcp relay always-on
- ipv6 hardware fib aggregate-address
- ipv6 hardware fib ecmp resilience
- ipv6 hardware fib nexthop-index
- ipv6 neighbor
- ipv6 neighbor cache persistent
- ipv6 route
- ipv6 unicast-routing

Interface Configuration Commands

- ipv6 address
- ipv6 dhcp relay destination
- ipv6 enable
- ipv6 helper-address
- ipv6 nd managed-config-flag
- ipv6 nd ns-interval
- ipv6 nd other-config-flag
- ipv6 nd prefix
- ipv6 nd ra dns-server
- ipv6 nd ra dns-servers lifetime
- ipv6 nd ra dns-suffix
- ipv6 nd ra dns-suffixes lifetime
- ipv6 nd ra hop-limit
- ipv6 nd ra interval
- ipv6 nd ra lifetime
- ipv6 nd ra mtu suppress
- ipv6 nd ra suppress
- ipv6 nd reachable-time
- ipv6 nd router-preference
- ipv6 verify

Privileged EXEC Commands

- clear ipv6 dhcp relay counters
- clear ipv6 neighbors

EXEC Commands

- show ipv6 dhcp relay counters
- show ipv6 hardware fib aggregate-address
- show ipv6 helper-address
- show ipv6 interface
- show ipv6 nd ra internal state
- show ipv6 neighbors
- show ipv6 route
- show ipv6 route age
- show ipv6 route host
- show ipv6 route interface
- show ipv6 route summary
- show ipv6 route tag

clear ipv6 dhcp relay counters

The **clear ipv6 dhcp relay counters** command resets the DHCP relay counters. When no port is specified, the command clears the counters for the switch and for all interfaces. Otherwise, the command clears the counter for the specified interface.

Command Mode

Privileged EXEC

Command Syntax

```
clear ipv6 dhcp relay counters [PORT]
```

Parameters

- ***PORT*** Interface through which neighbor is accessed. Options include:
  - <no parameter> all dynamic entries are removed.
  - **interface ethernet** *e_num* Ethernet interface specified by *e_num*.
  - **interface loopback** *l_num* Loopback interface specified by *l_num*.
  - **interface port-channel** *p_num* Port-channel interface specified by *p_num*.
  - **interface vlan** *v_num* VLAN interface specified by *v_num*.

Examples

- These commands clear the DHCP relay counters for Ethernet interface 4 and shows the counters before and after the **clear** command.

```
switch(config)#show ipv6 dhcp relay counters

          |  Dhcp Packets  |
Interface | Rcvd Fwdd Drop |         Last Cleared
----------|----- ---- -----|---------------------
  All Req |  376  376    0 | 4 days, 19:55:12 ago
 All Resp |  277  277    0 |
          |                |
Ethernet4 |  207  148    0 | 4 days, 19:54:24 ago


switch(config)#interface ethernet 4
switch(config-if-Et4)#clear ipv6 dhcp relay counters

          |  Dhcp Packets  |
Interface | Rcvd Fwdd Drop |         Last Cleared
----------|----- ---- -----|---------------------
  All Req |  380  380    0 | 4 days, 21:19:17 ago
 All Resp |  281  281    0 |
          |                |
Ethernet4 |    0    0    0 |4 days, 21:18:30 ago
These commands clear all DHCP relay counters on the switch.
switch(config-if-Et4)#exit
switch(config)#clear ipv6 dhcp relay counters
switch(config)#show ipv6 dhcp relay counters

          |  Dhcp Packets  |
Interface | Rcvd Fwdd Drop | Last Cleared
----------|----- ---- -----|-------------
  All Req |    0    0    0 |  0:00:03 ago
 All Resp |    0    0    0 |
          |                |
Ethernet4 |    0    0    0 | 0:00:03 ago
switch(config)#
```

clear ipv6 neighbors

The **clear ipv6 neighbors** command removes the specified dynamic IPv6 neighbor discovery cache entries. Commands that do not specify an IPv6 address remove all dynamic entries for the listed interface. Commands that do not specify an interface remove all dynamic entries.

Command Mode

Privileged EXEC

Command Syntax

clear ipv6 neighbors [*PORT*] [*DYNAMIC_IPV6*]

Parameters

- ***PORT*** Interface through which neighbor is accessed. Options include:
  - <no parameter> all dynamic entries are removed.
  - **ethernet** *e_num* Ethernet interface specified by *e_num*.
  - **loopback** *l_num* Loopback interface specified by *l_num*.
  - **management** *m_num* Management interface specified by *m_num*.
  - **port-channel** *p_num* Port-channel interface specified by *p_num*.
  - **vlan** *v_num* VLAN interface specified by *v_num*.
  - **vxlan** *vx_num* VXLAN interface specified by *vx_num*.
- **DYNAMIC_IPV6** Address of entry removed by the command. Options include:
  - <no parameter> all dynamic entries for specified interface are removed.
  - *ipv6_addr* IPv6 address of entry.

Example

- This command removes all dynamic neighbor entries for VLAN interface 200.

```
switch#clear ipv6 neighbors vlan 200
switch#
```

ipv6 address

The **ipv6 address** command assigns a global IPv6 address to the IPv6 interface, and defines the prefix length. This command is supported on routable interfaces. Multiple global IPv6 addresses can be assigned to an interface.

The **no ipv6 address** and **default ipv6 address** commands remove the IPv6 address assignment from the configuration mode interface by deleting the corresponding **ipv6 address** command from ***running-config***. If the command does not include an address, all address assignments are removed from the interface. IPv6 remains enabled on the interface after the removal of all IPv6 addresses only if an **ipv6 enable** command is configured on the interface.

Command Mode

Interface-Ethernet Configuration
Interface-Loopback Configuration
Interface-Management Configuration
Interface-Port-channel Configuration
Interface-VLAN Configuration

Command Syntax

```
ipv6 address ipv6_prefix
no ipv6 address [ipv6_prefix]
default ipv6 address [ipv6_prefix]
```

Parameters

- *ipv6_prefix* address assigned to the interface (CIDR notation).

Guidelines

This command is supported on routable interfaces.

Example

- These commands configure an IPv6 address and prefix length for VLAN 200:

```
switch(config)#interface vlan 200
switch(config-if-Vl200)#ipv6 address 10:23:31:00:01:32:93/64
switch(config-if-Vl200)#
```

ipv6 dhcp relay always-on

The **iv6p dhcp relay always-on** command enables the switch DHCP relay agent on the switch regardless of the DHCP relay agent status on any interface. By default, the DHCP relay agent is enabled only if at least one routable interface is configured with an **ipv6 dhcp relay destination** statement.

The **no ipv6 dhcp relay always-on** and **default ipv6 dhcp relay always-on** commands remove the **ipv6 dhcp relay always-on** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
ipv6 dhcp relay always-on
no ipv6 dhcp relay always-on
default ipv6 dhcp relay always-on
```

Example

- This command enables the DHCP relay agent.

```
switch(config)#ipv6 dhcp relay always-on
switch(config)#
```

ipv6 dhcp relay destination

The **ipv6 relay destination** command enables the DHCPv6 relay agent and sets the destination address on the configuration mode interface.

The **no ipv6 relay destination** and **default ipv6 relay destination** commands remove the corresponding **ipv6 relay destination** command from ***running-config***. When the commands do not list an IPv6 address, all **ipv6 relay destination** commands are removed from ***running-config***.

Command Mode

Interface-Ethernet Configuration
Interface-Port-channel Configuration
Interface-VLAN Configuration

Command Syntax

```
ipv6 dhcp relay destination ipv6_addr
no ipv6 dhcp relay destination [ipv6_addr]
default ipv6 dhcp relay destination [ipv6_addr]
```

Parameters

- *ipv6_addr* DCHP Server's IPv6 address.

Example

- This command enables the DHCPv6 relay agent and sets the destination address to 2001:0db8:0:1::1 on Ethernet interface 4.

```
switch(config)#interface ethernet 4
switch(config-if-Et4)#ipv6 dhcp relay destination 2001:0db8:0:1::1
switch(config-if-Et4)#show active
interface Ethernet4
   ipv6 dhcp relay destination 2001:db8:0:1::1
switch(config-if-Et4)#
```

ipv6 enable

The **ipv6 enable** command enables IPv6 on the configuration mode interface. Assigning an IPv6 address to an interface also enables IPv6 on the interface.

The **no ipv6 enable** and **default ipv6 enable** command remove the corresponding **ipv6 enable** command from ***running-config***. This action disables IPv6 on interfaces that are not configured with an IPv6 address.

Command Mode

Interface-Ethernet Configuration
Interface-Loopback Configuration
Interface-Management Configuration
Interface-Port-channel Configuration
Interface-VLAN Configuration

Command Syntax

```
ipv6 enable
no ipv6 enable
default ipv6 enable
```

Example

- This command enables IPv6 on VLAN interface 200.

```
switch(config)#interface vlan 200
switch(config-vl200)#ipv6 enable
switch(config-vl200)#
```

ipv6 hardware fib aggregate-address

The **ipv6 hardware fib aggregate-address** command specifies the routing table repository of specified IPv6 route.

By default, routes that are created statically through the CLI or dynamically through routing protocols are initially stored in software routing tables, then entered in the hardware routing table by the routing agent. This command prevents the entry of the specified route into the hardware routing table. Specified routes that are in the hardware routing table are removed by this command. Specific routes that are encompassed within the specified route prefix are affected by this command.

The **no ipv6 hardware fib aggregate-address** and **default ipv6 hardware fib aggregate-address** commands remove the restriction from the hardware routing table for the specified routes by removing the corresponding **ipv6 hardware fib aggregate-address** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

ipv6 hardware fib aggregate-address *ipv6_prefix* **summary-only software-forward**
no ipv6 hardware fib aggregate-address *ipv6_prefix*
default ipv6 hardware fib aggregate-address *ipv6_prefix*

Parameters

- *ipv6_prefix* IPv6 prefix that is restricted from the hardware routing table (CIDR notation).

Example

- These commands configure a hardware routing restriction for an IPv6 prefix, then displays that restriction.

```
switch(config)#ipv6 hardware fib aggregate-address fd77:4890:5313:ffed::/64
summary-only software-forward
switch(config)#show ipv6 hardware fib aggregate-address
Codes: S - Software Forwarded
S  fd77:4890:5313:ffed::/64

switch(config)#
```

ipv6 hardware fib ecmp resilience

The **ip hardware fib ecmp resilience** command configures a fixed number of next hop entries in the hardware ECMP table for the specified IPv6 address prefix. In addition to specifying the maximum number of next hop addresses that the table can contain for the prefix, the command introduces a redundancy factor that allows duplication of each next hop address. The fixed table space for the address is the maximum number of next hops multiplied by the redundancy factor.

The default method of adding or removing next hop entries when required by the active hashing algorithm leads to inefficient management of the ECMP table, which can result in the rerouting of packets to different next hops that breaks TCP packet flows. Implementing fixed table entries for a specified IP address allows data flows that are hashed to a valid next hop number to remain intact. Additionally, traffic is evenly distributed over a new set of next hops.

The **no ip hardware fib ecmp resilience** and **default ip hardware fib ecmp resilience** commands restore the default hardware ECMP table management by removing the **ip hardware fib ecmp resilience** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
ipv6 hardware fib ecmp resilience net_prfx capacity nhop_max redundancy
duplicates
no ipv6 hardware fib ecmp resilience net_addr
default ipv6 hardware fib ecmp resilience net_addr
```

Parameters

- *net_prfx* IPv6 address prefix managed by command.
- *nhop_max* Specifies maximum number of nexthop entries for specified IP address prefix. Value range varies by platform:
  - Helix: <***2*** to ***64***>
  - Trident: <***2*** to ***32***>
  - Trident II: <***2*** to ***64***>
- *duplicates* Specifies the redundancy factor. Value ranges from 1 to 128.

Example

- This command configures a hardware ECMP table space of 15 entries for the IPv6 address 2001:db8:0::/64. A maximum of five nexthop addresses can be specified for the address. When the table contains five nexthop addresses, each appears in the table three times. When the table contains fewer than five nexthop addresses, each is duplicated until the 15 table entries are filled.

  ```
  switch(config)#ipv6 hardware fib ecmp resilience 2001:db8:0::/64 capacity 5
  redundancy 3
  ```

ipv6 hardware fib nexthop-index

The **ipv6 hardware fib nexthop-index** command deterministically selects the next hop used for ECMP routes. By default, routes that are created statically through the CLI or dynamically through routing protocols are initially stored in software routing tables, then entered in the hardware routing table by the routing agent. This command specifies the method of creating an index-offset number that points to the next hop from the list of the route's ECMP next hops.

The index-offset is calculated by adding the next hop index to a prefix offset.

- Next hop index: specified in the command.
- Prefix offset: the least significant bits of the route's prefix.

The command specifies the number of bits that comprise the prefix offset. The prefix offset is set to the prefix when the command specifies a prefix size larger than the prefix. If the command specifies an prefix size of zero, the prefix-offset is also zero and the index-offset is set to the next hop index.

When the index-offset is greater than the number of next hops in the table, the position of the next hop is the remainder of the division of the index-offset by the number of next hop entries.

The **no ipv6 hardware fib nexthop-index** and **default ipv6 hardware fib nexthop-index** commands remove the specified nexthop used for ECMP routes by removing the **ipv6 hardware fib nexthop-index** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
ipv6 hardware fib nexthop nxthop_index [PREFIX]
no ipv6 hardware fib nexthop
default ipv6 hardware fib nexthop
```

Parameters

- *nxthop_index* specifies the next hop index. Value ranges from 0 to 32.
- ***PREFIX*** Number of bits of the route's prefix to use as the prefix-offset. Value ranges from 0 to 64.
  - <no parameter> The prefix offset is set to zero.
  - **prefix-bits** <***0*** to ***64***> Specifies the number bits to use as the prefix-offset.

Example

- This command specifies the next hop from the list of ECMP next hops for the route.

```
switch(config)#ipv6 hardware fib nexthop-index 5 prefix-bits 10
switch>show ip
IP Routing : Enabled
IP Multicast Routing : Disabled
VRRP: Configured on 0 interfaces

IPv6 Unicast Routing : Enabled
IPv6 ECMP Route support : False
IPv6 ECMP Route nexthop index: 5
IPv6 ECMP Route num prefix bits for nexthop index: 10
switch>
```

ipv6 helper-address

The **ipv6 helper-address** command enables the DHCP relay agent on the configuration mode interface and specifies a forwarding address for DHCP requests. An interface that is configured with multiple helper-addresses forwards DHCP requests to all specified addresses.

The **no ipv6 helper-address** and **default ipv6 helper-address** commands remove the corresponding **ipv6 helper-address** command from ***running-config***. Commands that do not specify an IP helper-address removes all helper-addresses from the interface.

Command Mode

Interface-Ethernet Configuration
Interface-Management Configuration
Interface-Port-channel Configuration

Command Syntax

```
ipv6 helper-address ipv6_addr
no ipv6 helper-address [ipv6_addr]
default ipv6 helper-address [ipv6_addr]
```

Parameters

- *ipv6_addr* DHCP server address accessed by interface.

Example

- This command enables the DHCP relay agent on VLAN interface 200 and configures the switch to forward DHCP requests received on this interface to the server at 2001:0db8:0:1::1.

```
switch(config)#interface vlan 200
switch(config-if-Vl200)#ipv6 helper-address 2001:0db8:0:1::1
switch(config-if-Vl200)#show active
interface Vlan200
   ipv6 helper-address 2001:0db8:0:1::1
switch(config-if-Vl200)#
```

ipv6 nd managed-config-flag

The **ipv6 nd managed-config-flag** command causes the ***managed address configuration*** flag to be set in IPv6 RA packets transmitted from the configuration mode interface.

The **no ipv6 nd managed-config-flag** and **default ipv6 nd managed-config-flag** commands restore the default setting where the ***managed address configuration*** flag is not set in IPv6 RA packets transmitted by the interface by removing the corresponding **ipv6 nd managed-config-flag** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration
Interface-Loopback Configuration
Interface-Management Configuration
Interface-Port-channel Configuration
Interface-VLAN Configuration

Command Syntax

```
ipv6 nd managed-config-flag
no ipv6 nd managed-config-flag
default ipv6 nd managed-config-flag
```

Example

- These commands cause the *managed address configuration* flag to be set in IPv6 RA packets sent from VLAN interface 200.

```
switch(config)#interface vlan 200
switch(config-if-Vl200)#ipv6 nd managed-config-flag
switch(config-if-Vl200)#
```

ipv6 nd ns-interval

The **ipv6 nd ns-interval** command configures the interval between IPv6 neighbor solicitation (NS) transmissions from the configuration mode interface.

The **no ipv6 nd ns-interval** and **default ipv6 nd ns-interval** commands return the IPv6 NS transmission interval for the configuration mode interface to the default value of 1000 milliseconds by removing the corresponding **ipv6 nd ns-interval** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration
Interface-Loopback Configuration
Interface-Management Configuration
Interface-Port-channel Configuration
Interface-VLAN Configuration

Command Syntax

```
ipv6 nd ns-interval period
no ipv6 nd ns-interval
default ipv6 nd ns-interval
```

Parameters

- *period* interval in milliseconds between successive IPv6 neighbor solicitation transmissions. Values range from 1000 to 4294967295. The default period is 1000 milliseconds.

Example

- This command configures a neighbor solicitation transmission interval of 30 seconds on VLAN interface 200.

```
switch(config)#interface vlan 200
switch(config-if-Vl200)#ipv6 nd ns-interval 30000
switch(config-if-Vl200)#
```

ipv6 nd other-config-flag

The **ipv6 nd other-config-flag** command configures the configuration mode interface to send IPv6 RAs with the ***other stateful configuration*** flag set.

The **no ipv6 nd other-config-flag** and **default ipv6 nd other-config-flag** commands restore the default setting by removing the corresponding **ipv6 nd other-config-flag** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration
Interface-Loopback Configuration
Interface-Management Configuration
Interface-Port-channel Configuration
Interface-VLAN Configuration

Command Syntax

```
ipv6 nd other-config-flag
no ipv6 nd other-config-flag
default ipv6 nd other-config-flag
```

Example

- These commands configure the switch to set the other stateful configuration flag in advertisements sent from VLAN interface 200.

```
switch(config)#interface vlan 200
switch(config-if-Vl200)#ipv6 nd other-config-flag
switch(config-if-Vl200)#
```

ipv6 nd prefix

The **ipv6 nd prefix** command configures neighbor discovery Router Advertisements (RAs) prefix inclusion for RAs sent from the configuration mode interface.

By default, all prefixes configured as IPv6 addresses (**ipv6 address**) are advertised in the interface's RAs. The **ipv6 nd prefix** command with the ***no-advertise*** option prevents advertising of the specified prefix without affecting the advertising of other prefixes specified as IPv6 addresses. When an interface configuration includes at least one **ipv6 nd prefix** command that enables prefix advertising, RAs advertise only prefixes specified through **ipv6 nd prefix** commands.

Commands enabling prefix advertising also specify the advertised valid and preferred lifetime periods. Default periods are 2,592,000 (valid) and 604,800 (preferred) seconds.

The **no ipv6 nd prefix** and **default ipv6 nd prefix** commands remove the corresponding **ipv6 nd prefix** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration
Interface-Loopback Configuration
Interface-Management Configuration
Interface-Port-channel Configuration
Interface-VLAN Configuration

Command Syntax

```
ipv6 nd prefix ipv6_prefix LIFETIME [FLAGS]
ipv6 nd prefix ipv6_prefix no-advertise
no ipv6 nd prefix ipv6_prefix
default ipv6 nd prefix ipv6_prefix
```

Parameters

- *ipv6_prefix* IPv6 prefix (CIDR notation).
- **no-advertise** Prevents advertising of the specified prefix.
- ***LIFETIME*** Period that the specified IPv6 prefix is advertised (seconds). Options include
  - *valid preferred* Two values that set the ***valid*** and ***preferred*** lifetime periods.
  - *valid* One value that sets the ***valid*** lifetime. The ***preferred*** lifetime is set to the default value.
  - <no parameter> The ***valid*** and ***preferred*** lifetime periods are set to their default values.

    Options for *valid*: <***0*** to ***4294967295***> and **infinite**. Default value is 2592000
    Options for *preferred*: <***0*** to ***4294967295***> and **infinite**. Default value is 604800
    The maximum value (***4294967295***) and **infinite** are equivalent settings.
- ***FLAGS*** ***on-link*** and ***autonomous address-configuration*** flag values in RAs.
  - <no parameter> both flags are set.
  - **no-autoconfig** ***autonomous address-configuration*** flag is reset.
  - **no-onlink** ***on-link*** flag is reset.
  - **no-autoconfig no-onlink** both flags are reset.
  - **no-onlink no-autoconfig** both flags are reset.

Example

- These commands enable neighbor discovery advertising for IPv6 address 3012:D678::/64, on VLAN interface 200, specifying a valid lifetime of 1,296,000 seconds and the default preferred lifetime.

```
switch(config)#interface vlan 200
switch(config-if-Vl200)#ipv6 nd prefix 3012:D678::/64 1296000
```

ipv6 nd ra dns-server

The **ipv6 nd ra dns-server** command configures the IPv6 address of a preferred Recursive DNS Server (RDNSS) for the command mode interface to include in its neighbor-discovery Router Advertisements (RAs). Including RDNSS information in RAs provides DNS server configuration for connected IPv6 hosts without requiring DHCPv6.

Multiple servers can be configured on the interface by using the command repeatedly. A lifetime value for the RDNSS can optionally be specified with this command, and overrides any default value configured for the interface using the **ipv6 nd ra dns-servers lifetime** command.

The **no ipv6 nd ra dns-server** and **default ipv6 nd ra dns-server** commands remove the corresponding **ipv6 nd ra dns-server** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration
Interface-Loopback Configuration
Interface-Management Configuration
Interface-Port-channel Configuration
Interface-VLAN Configuration

Command Syntax

```
ipv6 nd ra dns-server ipv6_addr SERVER_LIFE
no ipv6 nd ra dns-server ipv6_addr
default ipv6 nd ra dns-server ipv6_addr
```

Parameters

- *ipv6_addr* RDNSS address to be included in RAs from the command mode interface.
- ***SERVER_LIFE*** maximum lifetime value for the specified RDNSS entry. This value overrides any default lifetime value. Value should be between the RA interval configured on the interface and two times that interval. Options include:
  - <no parameter> lifetime period is the default lifetime period configured on the interface. If no lifetime period is configured on the interface, the default value is 1.5 times the maximum RA interval set by the **ipv6 nd ra interval** command.
  - **lifetime *0*** the configured RDNSS is not to be used.
  - **lifetime <*1* to *4294967295*>** specifies the lifetime period for this RDNSS in seconds.

Example

- This command configures the RDNSS at 2001:0db8:0:1::1 as a preferred RDNSS for VLAN interface 200 to include in its neighbor-discovery route advertisements, and sets its lifetime value to 300 seconds.

```
switch(config)#interface vlan 200
switch(config-if-Vl200)#ipv6 nd ra dns-server 2001:0db8:0:1::1 lifetime 300
switch(config-if-Vl200)#
```

ipv6 nd ra dns-servers lifetime

The **ipv6 nd ra dns-servers lifetime** command sets the default value that the configuration mode interface uses for the lifetime of any Recursive DNS Server (RDNSS) configured on the interface. A lifetime value set for an individual RDNSS overrides this value. The lifetime value is the maximum amount of time after a route advertisement packet is sent that the RDNSS referenced in the packet may be used for name resolution.

The **no ipv6 nd ra dns-servers lifetime** and **default ipv6 nd ra dns-servers lifetime** commands remove the default lifetime value from the interface by removing the corresponding **ipv6 nd ra dns-servers lifetime** command from ***running-config***. When there is no default RDNSS lifetime value configured on the interface, an RDNSS without a custom lifetime value will default to 1.5 times the RA interval configured on the interface. A lifetime of zero seconds means that the RDNSS must not be used for name resolution.

Command Mode

Interface-Ethernet Configuration
Interface-Loopback Configuration
Interface-Management Configuration
Interface-Port-channel Configuration
Interface-VLAN Configuration

Command Syntax

```
ipv6 nd ra dns-servers lifetime period
no ipv6 nd ra dns-servers lifetime
default ipv6 nd ra dns-servers lifetime
```

Parameters

- *period* the RDNSS lifetime value for the configuration mode interface. Options include:
  - <***0***> any RDNSS configured on the command mode interface without a custom lifetime value must not be used.
  - ***<1*** to ***4294967295>*** maximum RDNSS lifetime value for the configuration mode interface. This value is overridden by any lifetime value set with the **ipv6 nd ra dns-server** command. Should be between the router advertisement interval configured on the interface and two times that interval.

Example

- This command sets the default RDNSS maximum lifetime value for VLAN 200 to 350 seconds.

```
switch(config)#interface vlan 200
switch(config-if-Vl200)#ipv6 nd ra dns-servers lifetime 350
switch(config-if-Vl200)#
```

ipv6 nd ra dns-suffix

The **ipv6 nd ra dns-suffix** command creates a DNS Search List (DNSSL) for the command mode interface to include in its neighbor-discovery Router Advertisements as defined in RFC 6106 . The DNSSL contains the domain names of DNS suffixes for IPv6 hosts to append to short, unqualified domain names for DNS queries.

Multiple DNS domain names can be added to the DNSSL by using the command repeatedly. A lifetime value for the DNSSL can optionally be specified with this command, and overrides any default value configured for the interface using the **ipv6 nd ra dns-suffixes lifetime** command.

The **no ipv6 nd ra dns-suffix** and **default ipv6 nd ra dns-suffix** commands remove the corresponding **ipv6 nd ra dns-suffix** command from running-config.

Command Mode

Interface-Ethernet Configuration
Interface-Loopback Configuration
Interface-Management Configuration
Interface-Port-channel Configuration
Interface-VLAN Configuration

Command Syntax

```
ipv6 nd ra dns-suffix domain SUFFIX_LIFE
no ipv6 nd ra dns-suffix ipv6_addr
default ipv6 nd ra dns-suffix ipv6_addr
```

Parameters

- *domain* domain suffix for IPv6 hosts to append to short, unqualified domain names for DNS queries. Suffix must contain only alphanumeric characters, “.” and “-” and must begin and end with an alphanumeric character.
- ***SUFFIX_LIFE*** maximum lifetime value for the specified domain suffix. This value overrides any default lifetime value. Value should be between the RA interval configured on the interface and two times that interval. Options include:
  - <no parameter> lifetime period is the default lifetime period configured on the interface. If no lifetime period is configured on the interface, the default value is 1.5 times the maximum RA interval set by the ipv6 nd ra interval command.
  - **lifetime *0*** the configured domain suffix is not to be used.
  - **lifetime** <***1*** to ***4294967295***> specifies the lifetime period for this domain suffix in seconds.

Example

- These commands create a DNSSL for VLAN interface 200 to include in its neighbor-discovery route advertisements, and set its lifetime value to 300 seconds.

```
switch(config)#interface vlan 200
switch(config-if-Vl200)#ipv6 nd ra dns-suffix test.com lifetime 300
switch(config-if-Vl200)#
```

ipv6 nd ra dns-suffixes lifetime

The **ipv6 nd ra dns-suffixes lifetime** command sets the default value that the configuration mode interface uses for the lifetime of any DNS Search List (DNSSL) configured on the interface. A lifetime value set for an individual DNSSL overrides this value. The lifetime value is the maximum amount of time after a route advertisement packet is sent that the DNSSL included in the packet may be used for name resolution.

The **no ipv6 nd ra dns-suffixes lifetime** and **default ipv6 nd ra dns-suffixes lifetime** commands remove the default lifetime value from the interface by removing the corresponding **ipv6 nd ra dns-suffixes lifetime** command from ***running-config***. When there is no default DNSSL lifetime value configured on the interface, a DNSSL without a custom lifetime value will default to 1.5 times the RA interval configured on the interface. A lifetime of zero seconds means that the DNSSL must not be used for name resolution.

Command Mode

Interface-Ethernet Configuration
Interface-Loopback Configuration
Interface-Management Configuration
Interface-Port-channel Configuration
Interface-VLAN Configuration

Command Syntax

```
ipv6 nd ra dns-suffixes lifetime period
no ipv6 nd ra dns-suffixes lifetime
default ipv6 nd ra dns-suffixes lifetime
```

Parameters

- *period* the DNSSL lifetime value for the configuration mode interface. Options include:
  - <***0***> any DNSSL configured on the command mode interface without a custom lifetime value must not be used.
  - ***<1*** to ***4294967295>*** maximum DNSSL lifetime value for the configuration mode interface. This value is overridden by any lifetime value set with the **ipv6 nd ra dns-suffix** command. Should be between the RA interval configured on the interface and two times that interval.

Example

- This command sets the default DNSSL maximum lifetime value for VLAN 200 to 350 seconds.

```
switch(config)#interface vlan 200
switch(config-if-Vl200)#ipv6 nd ra dns-suffixes lifetime 350
switch(config-if-Vl200)#
```

ipv6 nd ra hop-limit

The **ipv6 nd ra hop-limit** command sets a suggested hop-limit value to be included in Router Advertisement (RA) packets. The hop-limit value is to be used by attached hosts in outgoing packets.

The **no ipv6 nd ra hop-limit** and **default ipv6 nd ra hop-limit** commands remove the corresponding **ipv6 nd ra hop-limit** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration
Interface-Loopback Configuration
Interface-Management Configuration
Interface-Port-channel Configuration
Interface-VLAN Configuration

Command Syntax

```
ipv6 nd ra hop-limit quantity
no ipv6 nd ra hop-limit lifetime
default ipv6 nd ra hop-limit lifetime
```

Parameters

- *quantity* the hop-limit value to be included in RA packets sent by the configuration mode interface. Options include:
  - <***0***> indicates that outgoing packets from attached hosts are to be immediately discarded.
  - <***1*** to ***255***> number of hops. The default value is 64.

Example

- These commands include a hop-limit value of 100 in RA packets sent by VLAN 200.

```
switch(config)#interface vlan 200
switch(config-if-Vl200)#ipv6 nd ra hop-limit
switch(config-if-Vl200)#
```

ipv6 nd ra interval

The **ipv6 nd ra interval** command configures the interval between IPv6 Router Advertisement transmissions from the configuration mode interface.

The **no ipv6 nd ra interval** and **default ipv6 nd ra interval** commands return the IPv6 RA transmission interval for the configuration mode interface to the default value of 200 seconds by removing the corresponding **ipv6 nd ra interval** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration
Interface-Loopback Configuration
Interface-Management Configuration
Interface-Port-channel Configuration
Interface-VLAN Configuration

Command Syntax

```
ipv6 nd ra interval [SCALE] ra_period [minimum_period]
no ipv6 nd ra interval
default ipv6 nd ra interval
```

Parameters

- ***SCALE*** timescale in which command parameter values are expressed.
  - <no parameter> seconds
  - **msec** milliseconds
- *ra_period* maximum interval between successive IPv6 RA transmissions. The default period is 200 seconds.
  - ***<4 - 1800>*** valid range when **scale** is set to default value (seconds).
  - ***<500 - 1800000>*** valid range when **scale** is set to ***msec***.
- *minimum_period* minimum interval between successive IPv6 RA transmissions. Must be smaller than *ra_period*. By default, a minimum period is not defined.
  - <no parameter> Command does not specify a minimum period.
  - ***<3 - 1799>*** valid range when **scale** is set to default value (seconds).
  - ***<375 - 1799999>*** valid range when **scale** is set to ***msec***.

Example

- These commands configure a RA transmission interval of 60 seconds on VLAN interface 200, then displays the interface status.

```
switch(config)#interface vlan 200
switch(config-if-Vl200)#ipv6 nd ra interval 60
switch(config-if-Vl200)#show active
interface Vlan200
   ipv6 nd ra interval 60
switch(config-if-Vl200)#
```

ipv6 nd ra lifetime

The **ipv6 nd ra lifetime** command specifies the value that the switch places in the ***router lifetime*** field of IPv6 Router Advertisements sent from the configuration mode interface.

If the value is set to 0, IPv6 peers connected to the specified interface will remove the switch from their lists of default routers. Values greater than 0 indicate the time in seconds that peers should keep the router on their default router lists without receiving further RAs from the switch. Unless the value is 0, the router lifetime value should be equal to or greater than the interval between unsolicited RAs sent on the interface.

The **no ipv6 nd ra lifetime** and **default ipv6 nd ra lifetime** commands return the IPv6 RA lifetime data entry filed for the configuration mode interface to the default value of 1800 seconds by removing the corresponding **ipv6 nd ra lifetime** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration
Interface-Loopback Configuration
Interface-Management Configuration
Interface-Port-channel Configuration
Interface-VLAN Configuration

Command Syntax

```
ipv6 nd ra lifetime ra_lifetime
no ipv6 nd ra lifetime
default ipv6 nd ra lifetime
```

Parameters

- *ra_lifetime* router lifetime period (seconds). Default value is 1800. Options include
  - *<0>* Router should not be considered as a default router
  - ***<1 - 65535>*** Lifetime period advertised in RAs. Should be greater than or equal to the interval between IPv6 RA transmissions from the configuration mode interface as set by the **ipv6 nd ra interval** command.

Example

- This command configures the switch to enter 2700 in the router lifetime field of RAs transmitted from VLAN 200.

```
switch(config)#interface vlan 200
switch(config-if-Vl200)#ipv6 nd ra lifetime 2700
switch(config-if-Vl200)#show active
interface Vlan20
   ipv6 nd ra lifetime 2700
switch(config-if-Vl200)#
```

ipv6 nd ra mtu suppress

The **ipv6 nd ra mtu suppress** command suppresses the Router Advertisement (RA) MTU option on the configuration mode interface. The MTU option causes an identical MTU value to be advertised by all nodes on a link. By default, the RA MTU option is not suppressed.

The **no ipv6 nd ra mtu suppress** and **default ipv6 nd ra mtu suppress** commands restores the MTU option setting to enabled by for the configuration mode interface by removing the corresponding **ipv6 nd ra mtu suppress** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration
Interface-Loopback Configuration
Interface-Management Configuration
Interface-Port-channel Configuration
Interface-VLAN Configuration

Command Syntax

```
ipv6 nd ra mtu suppress
no ipv6 nd ra mtu suppress
default ipv6 nd ra mtu suppress
```

Example

- This command suppresses the MTU option on VLAN interface 200.

```
switch(config)#interface vlan 200
switch(config-vl200)#ipv6 nd ra mtu suppress
switch(config-vl200)#
```

ipv6 nd ra suppress

The **ipv6 nd ra suppress** command suppress IPv6 Router Advertisement (RA) transmissions on the configuration mode interface. By default, only unsolicited RAs that are transmitted periodically are suppressed. The **all** option configures the switch to suppress all RAs, including those responding to a router solicitation.

The **no ipv6 nd ra suppress** and **default ipv6 nd ra suppress** commands restore the transmission of RAs on the configuration mode interface by deleting the corresponding **ipv6 nd ra suppress** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration
Interface-Loopback Configuration
Interface-Management Configuration
Interface-Port-channel Configuration
Interface-VLAN Configuration

Command Syntax

```
ipv6 nd ra suppress [SCOPE]
no ipv6 nd ra suppress
default ipv6 nd ra suppress
```

Parameters

- ***SCOPE*** specifies the RAs that are suppressed.
  - <no parameter> Periodic unsolicited RAs are suppressed.
  - **all** All RAs are suppressed.

Example

- This command suppresses all RAs on VLAN interface 200.

```
switch(config)#interface vlan 200
switch(config-vl200)#ipv6 nd ra suppress all
switch(config-vl200)#
```

ipv6 nd reachable-time

The **ipv6 nd reachable-time** command specifies the time period that the switch includes in the reachable time field of RAs sent from the configuration mode interface. The reachable time defines the period that a remote IPv6 node is considered reachable after a reachability confirmation event.

RAs that advertise zero seconds indicate that the router does not specify a reachable time. The default advertisement value is 0 seconds. The switch reachability default period is 30 seconds.

The **no ipv6 nd reachable-time** and **default ipv6 nd reachable-time** commands restore the entry of the default value (0) in RAs sent from the configuration mode interface by deleting the corresponding **ipv6 nd reachable-time** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration
Interface-Loopback Configuration
Interface-Management Configuration
Interface-Port-channel Configuration
Interface-VLAN Configuration

Command Syntax

```
ipv6 nd reachable-time period
no ipv6 nd reachable-time
default ipv6 nd reachable-time
```

Parameters

- *period* Reachable time value (milliseconds). Value ranges from ***0*** to ***4294967295***. Default is **0**.

Example

- These commands configure the entry of 25000 (25 seconds) in the reachable time field of RAs sent from VLAN 200.

```
switch(config)#interface vlan 200
switch(config-if-Vl200)#ipv6 nd reachable-time 25000
interface Vlan200
   ipv6 address fd7a:4321::1/64
   ipv6 nd reachable-time 25000
switch(config-if-Vl200)#
```

ipv6 nd router-preference

The **ipv6 nd router-preference** command specifies the value that the switch enters in the Default Router Preference (DRP) field of Router Advertisements (RAs) that it sends from the configuration mode interface. The default field entry value is ***medium***.

The **no ipv6 nd router-preference** and **default ipv6 nd router-preference** commands restore the switch to enter the default DRP field value of ***medium*** in RAs sent from the configuration mode interface by deleting the corresponding **ipv6 nd router-preference** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration
Interface-Loopback Configuration
Interface-Management Configuration
Interface-Port-channel Configuration
Interface-VLAN Configuration

Command Syntax

```
ipv6 nd router-preference RANK
no ipv6 nd router-preference
default ipv6 nd router-preference
```

Parameters

- ***RANK*** Router preference value. Options include:
  - **high**
  - **low**
  - **medium**

Example

- This command configures the switch as a medium preference router on RAs sent from VLAN 200.

```
switch(config)#interface vlan 200
switch(config-if-Vl200)#ipv6 nd router-preference medium
switch(config-if-Vl200)#
```

ipv6 neighbor

The **ipv6 neighbor** command creates an IPv6 neighbor discovery cache static entry. The command converts pre-existing dynamic cache entries for the specified address to static entries.

The **no ipv6 neighbor** and **default ipv6 neighbor** commands remove the specified static entry from the IPV6 neighbor discovery cache and delete the corresponding **ipv6 neighbor** command from ***running-config***. These commands do not affect any dynamic entries in the cache.

Command Mode

Global Configuration

Command Syntax

```
ipv6 neighbor ipv6_addr PORT mac_addr
no ipv6 neighbor ipv6_address PORT
default ipv6 neighbor ipv6_addr PORT
```

Parameters

- *ipv6_addr* Neighbor's IPv6 address.
- ***PORT*** Interface through which the neighbor is accessed. Options include:
  - **ethernet** *e_num* Ethernet interface specified by *e_num*.
  - **loopback** *l_num* Loopback interface specified by *l_num*.
  - **management** *m_num* Management interface specified by *m_num*.
  - **port-channel** *p_num* Port-channel interface specified by *p_num*.
  - **vlan** *v_num* VLAN interface specified by *v_num*.
  - **vxlan** *vx_num* VXLAN interface specified by *vx_num*.
- *mac_addr* Neighbor's data-link (hardware) address. (48-bit dotted hex notation – H.H.H).

Example

- This command will add a static entry to the neighbor discovery cache for the neighbor located at 3100:4219::3EF2 with hardware address 0100.4EA1.B100 and accessible through VLAN 200.

```
switch(config)#ipv6 neighbor 3100:4219::3EF2 vlan 200 0100.4EA1.B100
switch(config)#
```

ipv6 neighbor cache persistent

The **ipv6 neighbor cache persistent** command restores the IPv6 neighbor cache after reboot.

The **no ipv6 neighbor cache persistent** and **default ipv6 neighbor cache persistent** commands remove the ARP cache persistant configuration from the *running-config*.

Command Mode

Global Configuration

Command Syntax

```
ipv6 neighbor cache persistent
no ipv6 neighbor cache persistent
default ipv6 neighbor cache persistent
```

Example

- This command restores the ipv6 neighbor cache after reboot.

```
switch(config)# ipv6 neighbor cache persistent
switch(config)#
```

ipv6 route

The **ipv6 route** command creates an IPv6 static route. The destination is a IPv6 prefix; the source is an IPv6 address or a routable interface port. When multiple routes exist to a destination prefix, the route with the lowest administrative distance takes precedence.

By default, the administrative distance assigned to static routes is 1. Assigning a higher administrative distance to a static route configures it to be overridden by dynamic routing data. For example, a static route with a distance value of 200 is overridden by OSPF intra-area routes, which have a default distance of 110.

The command provides these methods of designating the nexthop location:

- null0: Traffic to the specified destination is dropped.
- IPv6 gateway: Switch identifies egress interface by recursively resolving the next-hop.
- Egress interface: Switch assumes destination subnet is directly connected to interface; when routing to any subnet address, the switch sends an ARP request to find the MAC address for the first packet.
- Combination Egress interface and IPv6 gateway: Switch does not assume subnet is directly connected to interface; the only ARP traffic is for the nexthop address for the first packet on the subnet. Combination routes are not recursively resolved.

Multiple routes that are configured to the same destination with the same administrative distance comprise an Equal Cost Multi-Path (ECMP) route. The switch attempts to spread outbound traffic across all ECMP route paths equally. All ECMP paths are assigned the same tag value; commands that change the tag value of any ECMP path change the tag value of all paths in the ECMP.

The **no ipv6 route** and **default ipv6 route** commands delete static routes by removing the corresponding **ipv6 route** statements from ***running-config***. Commands not including a source delete all statements to the destination. Only statements with parameters that match specified command arguments are deleted. Parameters that are not in the command line are not evaluated.

Command Mode

Global Configuration

Command Syntax

```
ipv6 route dest_prefix NEXTHOP [DISTANCE] [TAG_OPT] [RT_NAME]
no ipv6 route dest_prefix [nexthop_addr] [DISTANCE]
default ipv6 route dest_prefix [nexthop_addr] [DISTANCE]
```

Parameters

- *dest_prefix* destination IPv6 prefix (CIDR notation).
- ***NEXTHOP*** Access method of next hop device. Options include:
  - **null0** Null0 interface – route is dropped.
  - *nexthop_addr* IPv6 address of nexthop device.
  - **ethernet** *e_num* Ethernet interface specified by *e_num*.
  - **loopback** *l_num* Loopback interface specified by *l_num*.
  - **management** *m_num* Management interface specified by *m_num*.
  - **port-channel** *p_num* Port-channel interface specified by *p_num*.
  - **vlan** *v_num* VLAN interface specified by *v_num*.
  - **vxlan** *vx_num* VXLAN interface specified by *vx_num*.
  - **ethernet** *e_num* *nexthop_addr* Combination route (Ethernet interface and gateway).

- **loopback** *l_num nexthop_addr* Combination route (loopback interface and gateway).
- **management** *m_num nexthop_addr* Combination route (management interface and gateway).
- **port-channel** *p_num nexthop_addr* Combination route (port channel interface and gateway).
- **vlan** *v_num nexthop_addr* Combination route (VLAN interface and gateway).
- **vxlan** *vx_num nexthop_addr* Combination route (VXLAN interface and gateway)

- ***DISTANCE*** administrative distance assigned to route. Options include:
  - <no parameter> route assigned default administrative distance of one.
  - <***1*** to ***255***> The administrative distance assigned to route.
- ***TAG_OPT*** static route tag. Options include:
  - <no parameter> assigns default static route tag of 0.
  - **tag** <***0*** to ***4294967295***> Static route tag value.
- ***RT_NAME*** Associates descriptive text to the route. Options include:
  - <no parameter> No text is associated with the route.
  - **name** *descriptive_text* The specified text is assigned to the route.

Example

- This command creates an IPv6 static route.

```
switch(config)#ipv6 route 10:23:31:00:01:32:93/24 vlan 300
```

ipv6 unicast-routing

The **ipv6 unicast-routing** command enables the forwarding of IPv6 unicast packets. When routing is enabled, the switch attempts to deliver inbound packets to destination addresses by forwarding them to interfaces or next hop addresses specified by the IPv6 routing table.

The **no ipv6 unicast-routing** and default **ip ipv6 unicast-routing** commands disable IPv6 unicast routing by removing the **ipv6 unicast-routing** command from ***running-config***. Dynamic routes added by routing protocols are removed from the routing table. Static routes are preserved by default; the **delete-static-routes** option removes static entries from the routing table.

IPv6 unicast routing is disabled by default.

Command Mode

Global Configuration

Command Syntax

```
ipv6 unicast-routing
no ipv6 unicast-routing [DELETE_ROUTES]
default ipv6 unicast-routing [DELETE_ROUTES]
```

Parameters

- ***DELETE_ROUTES*** Resolves routing table static entries when routing is disabled.
  - <no parameter> Routing table retains static entries.
  - **delete-static-routes** Static entries are removed from the routing table.

Example

- This command enables IPv6 unicast-routing.

```
switch(config)#ipv6 unicast-routing
switch(config)#
```

ipv6 verify

The **ipv6 verify** command configures Unicast Reverse Path Forwarding (uRPF) for inbound IPv6 packets on the configuration mode interface. uRPF verifies the accessibility of source IP addresses in packets that the switch forwards.

uRPF defines two operational modes: strict mode and loose mode.

- Strict mode: uRPF also verifies that a packet is received on the interface that its routing table entry specifies for its return packet.
- Loose mode: uRPF validation does not consider the inbound packet's ingress interface.

The **no ipv6 verify** and **default ipv6 verify** commands disable uRPF on the configuration mode interface by deleting the corresponding **ipv6 verify** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration
Interface-Loopback Configuration
Interface-Management Configuration
Interface-Port-Channel Configuration
Interface-VLAN Configuration

Command Syntax

```
ipv6 verify unicast source reachable-via RPF_MODE
no ipv6 verify unicast
default ipv6 verify unicast
```

Parameters

- ***RPF_MODE*** Specifes the uRPF mode. Options include:
  - **any** Loose mode.
  - **rx** Strict mode
  - **rx allow-default** Strict mode. All inbound packets are forwarded if a default route is defined.

Guidelines

The first IPv6 uRPF implementation briefly disables IPv6 unicast routing. Subsequent **ip verify** commands on any interface do not disable IPv6 routing.

Example

- This command enables uRPF strict mode on VLAN interface 100. When a default route is configured on the interface, all inbound packets are checked as valid.

```
switch(config)#interface vlan 100
switch(config-if-Vl100)#ipv6 verify unicast source reachable-via rx
allow-default
switch(config-if-Vl100)#show active
 interface Vlan100
   ipv6 verify unicast source reachable-via rx allow-default
switch(config-if-Vl100)#
```

show ipv6 dhcp relay counters

The **show ipv6 dhcp relay counters** command displays the number of DHCP packets received, forwarded, or dropped on the switch and on all interfaces enabled as DHCP relay agents.

Command Mode

EXEC

Command Syntax

show ipv6 dhcp relay counters

Example

- This command displays the IP DHCP relay counter table.

```
switch>show ipv6 dhcp relay counters

           |  Dhcp Packets  |
Interface | Rcvd Fwdd Drop |         Last Cleared
----------|----- ---- -----|---------------------
  All Req |  376  376    0 | 4 days, 19:55:12 ago
 All Resp |  277  277    0 |
           |                |
Ethernet4 |  207  148    0 | 4 days, 19:54:24 ago

switch>
```

show ipv6 hardware fib aggregate-address

The **show ipv6 hardware fib aggregate-address** command displays the IPv6 prefixes that are restricted from entry into the hardware routing table. The **ipv6 hardware fib aggregate-address** command configures IPv6 prefix restictions.

Command Mode

EXEC

Command Syntax

show ipv6 address fib aggregate-address [*ADDRESS*][*RESTRICTION*]

Parameters

- ***ROUTE_FILTER*** filters by IPv6 address. Options include:
  - <no parameter> Displays all routes.
  - *ipv6_addr* Command displays only specified address.
  - *ipv6_prefix* Command displays addresses filtered by specified prefix (CIDR notation).
- ***RESTRICTION*** filters by route restriction.
  - <no parameter> displays routes restricted from the hardware routing table.
  - **software-forward** displays routes restricted from the hardware routing table.

Example

- This command displays the routes that are restricted from the hardware routing table.

```
switch>show ipv6 hardware fib aggregate-address
Codes: S - Software Forwarded
S  fd77:4890:5313:aaed::/64
S  fd77:4890:5313:ffed::/64

switch>
```

show ipv6 helper-address

The **show ipv6 helper-address** command displays the status of DHCP relay agent parameters on the switch and each interface where at least one feature parameter is listed. The command provides status on the following parameters:

- Global: DHCP relay agent Always-on mode, DHCP relay agent Information option
- Interface: DHCP server (list of addresses), Circuit ID contents.

Command Mode

EXEC

Command Syntax

show ipv6 helper-address

Example

- This command displays the DHCP Agent Relay parameter status.

```
switch>show ipv6 helper-address
DHCP Relay Agent Information Option Enabled
DHCP Relay Agent Always-On Mode Enabled
Interface: Ethernet4
   Circuit ID: V-200
   DHCP servers: 2001:db8:0:1::1
switch>
```

show ipv6 interface

The **ipv6 interface** command displays the status of specified routed interfaces that are configured for IPv6.

Command Mode

EXEC

Command Syntax

show ipv6 interface [*INTERFACE_NAME*] [*INFO_LEVEL*]

Parameters

- ***INTERFACE_NAME*** interfaces for which command displays status.
  - <no parameter> all routed interfaces.
  - **ethernet** *e_num* Ethernet interface specified by *e_num*.
  - **loopback** *l_num* Loopback interface specified by *l_num*.
  - **management** *m_num* Management interface specified by *m_num*.
  - **port-channel** *p_num* Port-Channel Interface specified by *p_num*.
  - **vlan** *v_num* VLAN interface specified by *v_num*.
  - **vxlan** *vx_num* VXLAN interface specified by *vx_num*.
- ***INFO_LEVEL*** amount of information that is displayed. Options include:
  - <no parameter> command displays data block for each specified interface.
  - **brief** command displays table that summarizes IPv6 interface data.

Example

- This command displays the status of VLAN 903.

```
switch>show ipv6 interface vlan 903
Vlan903 is up, line protocol is up (connected)
  IPv6 is enabled, link-local is fe80::21c:73ff:fe01:21e/64
  Global unicast address(es):
    fd7a:629f:52a4:fe10::3, subnet is fd7a:629f:52a4:fe10::/64
  Joined group address(es):
    ff02::1
    ff02::1:ff01:21e
    ff02::1:ff00:3
    ff01::2
switch>
```

show ipv6 nd ra internal state

The **ipv6 nd ra internal state** command displays the state of the IPv6 Router Advertisement (RA) daemon for the specified routable interface.

Command Mode

EXEC

Command Syntax

show ipv6 nd ra internal state [*INTERFACE_NAME*]

Parameters

- ***INTERFACE_NAME*** interfaces for which command displays status.
  - <no parameter> all routed interfaces.
  - **ethernet** *e_num* Ethernet interface specified by *e_num*.
  - **loopback** *l_num* Loopback interface specified by *l_num*.
  - **management** *m_num* Management interface specified by *m_num*.
  - **port-channel** *p_num* Port-Channel Interface specified by *p_num*.
  - **vlan** *v_num* VLAN interface specified by *v_num*.
  - **vxlan** *vx_num* VXLAN interface specified by *vx_num*.

Example

- This command displays the IPv6 RA daemon for VLAN interface 1243.

```
switch>show ipv6 nd ra internal state vlan 1243
INTERFACE: Vlan3908
                                  ifindex : 0x00000021
                                      mtu : 9212
                               numIpv6Addr : 2
                        numPrefixToAdvertise : 0
                         numPrefixToSuppress : 0
                                  RaSuppress : 0
                               RsRspSuppress : 0
                           raIntervalMaxMsec : 200000
                           raIntervalMinMsec : 0
                           managedConfigFlag : 0
                             otherConfigFlag : 0
                               raMtuSuppress : 0
                                  raLifetime : 1800
                              reacheableTime : 0
                             routerPreference : 0
                                   lastRaTime : 2012-05-01 09:22:57.020634
                           lastRsRspSentTime :
                                 nextTimeout : 171.474535 (sec)
                        raNotSentIntfNotReady : 0
                                    numRaSent : 219
                                    numRsRcvd : 0
                                numRsSuppressed : 0
                                   numRsRspSent : 0
                numRsDroppedInvalidHopLimit : 0
                numPktDroppedUnexpectedType : 0
                                initialized : 1
switch>
```

show ipv6 neighbors

The **show ipv6 neighbors** command displays the IPv6 neighbor discovery cache. The command provides filters to restrict the list to a specified IPv6 address or routable interface.

Command Mode

EXEC

Command Syntax

show ipv6 neighbors [*PORT*] [*SOURCE*] [*INFO_LEVEL*]

Parameters

- ***PORT*** Filters by interface through which neighbor is accessed. Options include:
  - <no parameter> all routed interfaces.
  - **ethernet** *e_num* Ethernet interface specified by *e_num*.
  - **loopback** *l_num* Loopback interface specified by *l_num*.
  - **management** *m_num* Management interface specified by *m_num*.
  - **port-channel** *p_num* Port-channel interface specified by *p_num*.
  - **vlan** *v_num* VLAN interface specified by *v_num*.
  - **vxlan** *vx_num* VXLAN interface specified by *vx_num*.
- ***SOURCE*** Filters by neighbor IPv6 address. Options include:
  - <no parameter> all IPv6 neighbors.
  - *ipv6_addr* IPv6 address of individual neighbor.
- ***INFO_LEVEL*** amount of information that is displayed. Options include:
  - <no parameter> command displays the discovery cache for the specified interfaces.
  - **summary** command displays summary information only.

Example

- This command displays the IPv6 neighbor discovery cache for IPv6 address ***fe80::21c:73ff:fe01:5fe1***.

```
switch>show ipv6 neighbors fe80::21c:73ff:fe01:5fe1
IPv6 Address                              Age Hardware Addr    State Interface
fe80::21c:73ff:fe01:5fe1                    0 001c.d147.8214   REACH Et12
fe80::21c:73ff:fe01:5fe1                    0 001c.d147.8214   REACH Po999
fe80::21c:73ff:fe01:5fe1                    0 001c.d147.8214   REACH Vl102
fe80::21c:73ff:fe01:5fe1                    0 001c.d147.8214   REACH Vl103
fe80::21c:73ff:fe01:5fe1                    0 001c.d147.8214   REACH Vl205
fe80::21c:73ff:fe01:5fe1                    0 001c.d147.8214   REACH Vl207
fe80::21c:73ff:fe01:5fe1                    0 001c.d147.8214   REACH Vl3901
fe80::21c:73ff:fe01:5fe1                    0 001c.d147.8214   REACH Vl3902
fe80::21c:73ff:fe01:5fe1                    0 001c.d147.8214   REACH Vl3903
fe80::21c:73ff:fe01:5fe1                    0 001c.d147.8214   REACH Vl3904
fe80::21c:73ff:fe01:5fe1                    0 001c.d147.8214   REACH Vl3905
fe80::21c:73ff:fe01:5fe1                    0 001c.d147.8214   REACH Vl3996
```

show ipv6 route

The **show ipv6 route** command displays IPv6 routing table entries that are in the Forwarding Information Base (FIB), including static routes, routes to directly connected networks, and dynamically learned routes. Multiple equal cost paths to the same prefix are displayed contiguously as a block, with the destination prefix displayed only on the first line.

The **show running-config** command displays all configured routes.

Command Mode

EXEC

Command Syntax

```
show ipv6 route [ADDRESS] [ROUTE_TYPE] [INFO_LEVEL]
```

Parameters

Address, when present, is always listed first. All other parameters can be placed in any order.

- ***ADDRESS*** filters routes by IPv6 address or prefix.
  - <no parameter> all routing table entries.
  - *ipv6_address* routing table entries matching specified IPv6 address.
  - *ipv6_prefix* routing table entries matching specified IPv6 prefix (CIDR notation).
- ***ROUTE_TYPE*** filters routes by specified protocol or origin.
  - <no parameter> all routing table entries.
  - **aggregate** entries for BGP aggregate routes.
  - **bgp** entries added through BGP protocol.
  - **connected** entries for routes to networks directly connected to the switch.
  - **kernel** entries appearing in Linux kernel but not added by EOS software.
  - **isis** entries added through IS-IS protocol.
  - **ospf** entries added through OSPF protocol.
  - **static** entries added through CLI commands.
- ***INFO_LEVEL*** Filters entries by next hop connection.
  - <no parameter> filters routes whose next hops are directly connected.
  - **detail** displays all routes.

Example

- This command displays a route table entry for a specific IPv6 route.

```
switch>show ipv6 route fd7a:3418:52a4:fe18::/64
IPv6 Routing Table - 77 entries
Codes: C - connected, S - static, K - kernel, O - OSPF, B - BGP, R - RIP, A -
Aggregate

O   fd7a:3418:52a4:fe18::/64 [10/20]
     via fe80::21c:73ff:fe00:1319, Vlan3601
     via fe80::21c:73ff:fe00:1319, Vlan3602
     via fe80::21c:73ff:fe00:1319, Vlan3608
     via fe80::21c:73ff:fe0f:6a80, Vlan3610
     via fe80::21c:73ff:fe00:1319, Vlan3611

switch>
```

show ipv6 route age

The **show ipv6 route age** command displays the IPv6 route age to the specified IPv6 address or prefix.

Command Mode

EXEC

Command Syntax

show ipv6 route *ADDRESS* age

Parameters

- ***ADDRESS*** filters routes by IPv6 address or prefix.
  - *ipv6_address* routing table entries matching specified address (A:B:C:D:E:F:G:H).
  - *ipv6_prefix* routing table entries matching specified IPv6 prefix (A:B:C:D:E:F:G:H/PL).

Example

- This command displays the route age for the specified prefix.

```
switch>show ipv6 route 2001::3:0/11 age
IPv6 Routing Table - 74 entries
Codes: C - connected, S - static, K - kernel, O - OSPF, B - BGP, R - RIP, A -
Aggregate

C 2001::3:0/11 age 00:02:34
switch>
```

show ipv6 route host

The **show ipv6 route host** command displays all host routes in the IPv6 host forwarding table. Host routes are those whose destination prefix is the entire address (prefix = /128). Each displayed host route is labeled with its purpose:

- F static routes from the FIB.
- R routes defined because the IP address is an interface address.
- A routes to any neighboring host for which the switch has an ARP entry.

Command Mode

EXEC

Command Syntax

```
show ipv6 route host
```

Example

- This command displays all IPv6 host routes in the host forwarding table.

```
switch>show ipv6 route host
R - receive F - FIB, A - attached

 F  ::1 to cpu
 A  fee7:48a2:0c11:1900:400::1 on Vlan102
 R  fee7:48a2:0c11:1900:400::2 to cpu
 F  fee7:48a2:0c11:1a00::b via fe80::21c:73ff:fe0b:a80e on Vlan3902
 R  fee7:48a2:0c11:1a00::17 to cpu
 F  fee7:48a2:0c11:1a00::20 via fe80::21c:73ff:fe0b:33e on Vlan3913
 F  fee7:48a2:0c11:1a00::22 via fe80::21c:73ff:fe01:5fe1 on Vlan3908
                            via fe80::21c:73ff:fe01:5fe1 on Vlan3902

switch>
```

show ipv6 route interface

The **show ipv6 route interface** command displays routing table entries on a specified routed port.

Command Mode

EXEC

Command Syntax

show ipv6 route [*ADDRESS*] interface *PORT_NAME* [*INFO_LEVEL*]

Parameters

ADDRESS, when present, is always listed first. All other parameters can be placed in any order.

- ***ADDRESS*** filters routes by IPv6 address or prefix.
  - <no parameter> all routing table entries.
  - *ipv6_address* routing table entries matching specified IPv6 address.
  - *ipv6_prefix* routing table entries matching specified IPv6 prefix (CIDR notation).
- ***PORT_NAME*** interfaces for which command displays status.
  - **ethernet** *e_num* Ethernet interface specified by *e_num*.
  - **loopback** *l_num* Loopback interface specified by *l_num*.
  - **management** *m_num* Management interface specified by *m_num*.
  - **port-channel** *p_num* Port-Channel Interface specified by *p_num*.
  - **vlan** *v_num* VLAN interface specified by *v_num*.
  - **vxlan** *vx_num* VXLAN interface specified by *vx_num*.
- ***INFO_LEVEL*** Filters entries by next hop connection.
  - <no parameter> filters routes whose next hops are directly connected.
  - **detail** displays all routes.

Example

- This command displays the IPv6 routes in VLAN interface 661.

```
switch>show ipv6 route interface ethernet 8
IPv6 Routing Table - 77 entries
Codes: C - connected, S - static, K - kernel, O - OSPF, B - BGP, R - RIP, A -
Aggregate

O   fd7a:629f:63af:1232::/64 [150/11]
     via fe80::823c:73ff:fe00:3640, Ethernet8
O   fd7a:629f:63af:4118::/64 [150/11]
     via fe80::823c:73ff:fe00:3640, Ethernet8
O   fd7a:629f:63af:4119::/64 [150/11]
     via fe80::823c:73ff:fe00:3640, Ethernet8
O   fd7a:629f:63af:411a::/64 [150/11]
     via fe80::823c:73ff:fe00:3640, Ethernet8
O   fd7a:629f:63af:fe78::/64 [150/11]
     via fe80::823c:73ff:fe00:3640, Ethernet8
C   fd7a:629f:63af:fe88::/64 [0/1]
     via ::, Ethernet12
O   fd7a:629f:63af:fe8c::/64 [10/20]
     via fe80::21c:73ff:fe00:3640, Ethernet8
C   fe80:0:40::/64 [0/1]
     via ::, Ethernet8
```

show ipv6 route summary

The **show ipv6 route summary** command displays the information about the IPv6 routing table.

Command Mode

EXEC

Command Syntax

show ipv6 route summary

Example

- This command displays the route source and the corresponding number of routes in the IPv6 routing table.

```
switch>show ipv6 route summary
   Route Source    Number Of Routes
------------------ ----------------
   connected                    2
   static                       0
   ospf                         5
   bgp                          7
   isis                         0
   internal                     1
   attached                     0
   aggregate                    2

   Total Routes                 17
switch>
```

show ipv6 route tag

The **show ipv6 route tag** command displays the route tag assigned to the specified IPv6 address or prefix. Route tags are added to static routes for use by route maps.

Command Mode

EXEC

Command Syntax

show ipv6 route *ADDRESS* tag

Parameters

- ***ADDRESS*** filters routes by IPv6 address or prefix.
  - *ipv6_address* routing table entries matching specified address (A:B:C:D:E:F:G:H)
  - *ipv6_prefix* routing table entries matching specified IPv6 prefix (A:B:C:D:E:F:G:H/PL).

Example

- This command displays the route tag for the specified prefix.

```
switch>show ipv6 route fd7b:789f:5314:fe08::/64 tag
IPv6 Routing Table - 74 entries
Codes: C - connected, S - static, K - kernel, O - OSPF, B - BGP, R - RIP, A -
Aggregate

C   fd7b:789f:5314:fe08::/64 tag 0

switch>
```

Chapter 26

Traffic Management

This chapter describes Arista's Traffic Management, including configuration instructions and command descriptions. Topics covered by this chapter include:

- Section 26.1: Traffic Management Conceptual Overview
- Section 26.2: Traffic Management Configuration – Arad Platform Switches
- Section 26.3: Traffic Management Configuration – FM6000 Platform Switches
- Section 26.4: Traffic Management Configuration – Petra Platform Switches
- Section 26.5: Traffic Management Configuration – Trident Platform Switches
- Section 26.6: Traffic Management Configuration – Trident-II Platform Switches
- Section 26.7: Traffic Management Configuration Commands

26.1 Traffic Management Conceptual Overview

Traffic is managed through policy maps that apply data shaping methods to specific data streams. A policy map is a data structure that identifies specific data streams and then defines shaping parameters that modify packets within the streams. The switch defines three types of policies:

- Section 26.1.1: Control Plane Policies: Control plane policy maps are applied to the control plane.
- Section 26.1.2: QoS Policies: QoS policy maps are applied to Ethernet and port channel interfaces.
- Section 26.1.3: PBR Policies: PBR policy maps are applied to Ethernet interfaces, port channel interfaces and switch virtual interfaces (SVIs).
- Section 26.1.4: PDP Policies: PDP policy maps are assigned to Ethernet interfaces.

Note PDP is available only in EOS versions 4.19.0F and above.

A policy map consists of classes. Each class contains an eponymous class map and traffic resolution commands.

- A class map is a data structure that defines a data stream by specifying characteristics of data packets that comprise that stream. Each class map is typed as either QoS, control plane, PBR, or PDP and is available only to identically typed policy maps.
- Traffic resolution commands specify data handling methods for traffic that matches a class map. Traffic resolution options vary by policy map type.

Data packets that enter an entity to which a policy map is assigned are managed with traffic resolution commands of the first class that matches the packets.

26.1.1 Control Plane Policies

Note On hardware purchased after July 3, 2017, CoPP is not available.

The switch defines one control plane policy map named **copp-system-policy**. The **copp-system-policy** policy map is always applied to the control plane and cannot be removed from the switch. Other control plane policy maps cannot be added. C**opp-system-policy** consists of preconfigured classes, each containing a static class map and traffic resolution commands. Preconfigured classes cannot be removed from **copp-system-policy**.

Static class maps are provided by the switch and cannot be modified or deleted. The naming convention of static class maps is **copp-system-*name***, where ***name*** differentiates the class maps. Static class maps have pre-defined internal conditions, are not based on ACLs, and are only listed in ***running-config*** as components of **copp-system-policy**. The sequence of static class maps in the policy map is not significant. Traffic resolution commands define minimum (bandwidth) and maximum (shape) transmission rates for data streams matching the corresponding class map.

Copp-system-policy can be modified through the following steps:

- Add classes consisting of an eponymous dynamic class map and traffic resolution commands.

  Dynamic class maps are user created, can be edited or deleted, filter traffic with a single IPv4 ACL, and are listed in ***running-config***.
- Change traffic resolution commands for a preconfigured class.

These sections describe describe control plane traffic policy configuration procedures:

- Section 26.2.1: Configuring Control Plane Traffic Policies – Arad Platform Switches
- Section 26.3.1: Configuring Control Plane Traffic Policies – FM6000 Platform Switches
- Section 26.4.1: Configuring Control Plane Traffic Policies – Petra Platform Switches
- Section 26.5.1: Configuring Control Plane Traffic Policies – Trident Platform Switches

26.1.2 QoS Policies

QoS policy maps are user defined. The switch does not provide preconfigured QoS policy maps and in the default configuration, policy maps are not applied to any Ethernet or port channel interface. Policy maps and class maps are created and applied to interfaces through configuration commands.

A QoS policy map is composed of one or more classes. Each class contains an eponymous dynamic class map and traffic resolution commands. Dynamic class maps are user created, can be edited or deleted, filter traffic with a single IPv4 ACL, and are listed in ***running-config***.

QoS traffic resolution commands perform one of the following:

- Set the layer 2 CoS field
- Set the DSCP value in the ToS byte
- Specify a traffic class queue

The last class in all QoS policy maps is ***class-default***, which is composed as follows:

- The ***class-default*** class map matches all traffic except IPv4 or IPv6 traffic and is not editable.
- By default, ***class-default*** class contains no traffic resolution commands. Traffic resolution commands can be added through configuration commands.

Data packets that enter an interface to which a policy map is assigned are managed with traffic resolution commands that correspond to the first class that matches the packet.

These sections describe describe QoS traffic policy configuration procedures:

- Section 26.2.2: Configuring QoS Traffic Policies – Arad Platform Switches
- Section 26.3.2: Configuring QoS Traffic Policies – FM6000 Platform Switches
- Section 26.4.2: Configuring QoS Traffic Policies – Petra Platform Switches
- Section 26.5.2: Configuring QoS Traffic Policies – Trident Platform Switches

26.1.3 PBR Policies

Policy-Based Routing (PBR) allows the operator to specify the next hop for selected incoming packets on an L3 interface, overriding the routing table. Incoming packets are filtered through a policy map referencing one or more ACLs, and matching packets are routed to the next hop specified.

A PBR policy map is composed of one or more classes and can include next-hop information for each class. It can also include single-line raw match statements, which have the appearance and function of a single line from an ACL. Each class contains an eponymous class map. Class maps are user-created, can be edited or deleted, filter traffic using IPv4 ACLs, and are listed in ***running-config***.

These sections describe PBR policy configuration procedures:

- Section 26.2.3: Configuring PBR Policies – Arad Platform Switches
- Section 26.3.3: Configuring PBR Policies – FM6000 Platform Switches
- Section 26.4.3: Configuring PBR Policies – Petra Platform Switches
- Section 26.5.3: Configuring PBR Policies – Trident Platform Switches

26.1.4 PDP Policies

Note PDP is available only in EOS versions 4.19.0F and above.

Per-port Denial-of-service Protection (PDP) is a new, specialized, port-level packet classification and policing service, similar to the existing port-level QoS policy service, targeted specifically at identifying and rate limiting potential denial of service traffic.

Default policies and shared-policy support vary by platform.

Arad and Jericho platforms only shared policies are supported.

Alta and XP only unshared policies are supported

Trident II, Tomahawk, and Helix4 both shared and unshared policies are supported.

Creating a PDP policy and assigning it to an interface uses procedures similar to those for QoS, but with the following commands.

- To create an unshared PDP policy, use the **policy-map type pdp** command.
- To create a shared PDP policy, use the **policy-map type pdp** command with the **shared** keyword.
- To assign a policy to an input interface, use the **service-policy type pdp (Interface mode)** command in the configuration mode for the interface.

26.2 Traffic Management Configuration – Arad Platform Switches

Traffic policies are implemented by policy maps, which are applied to the control plane, or to L3 interfaces for Policy-Based Routing (PBR). Policy maps contain classes, which are composed of class maps and traffic resolution commands.

Section 26.1 describes traffic policies.

26.2.1 Configuring Control Plane Traffic Policies – Arad Platform Switches

Note On hardware purchased after July 3, 2017, CoPP is not available.

Default control plane traffic policies are implemented automatically without user intervention. These policies are modified by associating traffic resolution commands with static classes that comprise the control plane policy map.

Static Class Maps

Control plane traffic policies utilize static class maps, which are provided by the switch, are not editable, and cannot be deleted.

Editing the Policy Map

The only control plane policy map is **copp-system-policy**, which cannot be deleted. In its default form, **copp-system-policy** consists of the classes listed in Table 26-1. Although the underlying class map of each class cannot be edited, the traffic resolution commands can be adjusted. The default classes cannot be removed from the policy map and their sequence within the policy map is not editable.

Table 26-1 copp-system-policy default classes: Arad Platform Switches

Class Name	shape (pps)	bandwidth (pps)
copp-system-bgp	2500	250
copp-system-bpdu	2500	1250
copp-system-default	2500	250
copp-system-ipbroadcast	2500	250
copp-system-ipmc	2500	250
copp-system-ipmcmiss	2500	250
copp-system-ipunicast	NO LIMIT	250
copp-system-l2broadcast	2500	250
copp-system-l2unicast	NO LIMIT	250
copp-system-l3destmiss	2500	250
copp-system-l3lpmoverflow	2500	250
copp-system-l3slowpath	2500	250
copp-system-l3ttl1	2500	250
copp-system-lacp	2500	1250
copp-system-linklocal	2500	250
copp-system-lldp	2500	250
copp-system-mlag	2500	250

Table 26-1 copp-system-policy default classes: Arad Platform Switches

Class Name	shape (pps)	bandwidth (pps)
copp-system-multicastsnoop	2500	250
copp-system-OspfIsis	2500	250
copp-system-sflow	2500	250

Policy maps are modified in policy-map configuration mode. The **policy-map type control-plane** command enters policy-map configuration mode.

Example

- This command enters policy-map configuration mode for editing ***copp-system-policy***.

```
switch(config)#policy-map type control-plane copp-system-policy
switch(config-pmap-copp-system-policy)#
```

The **class (policy-map (control-plane) – Arad)** command enters policy-map-class configuration mode, where traffic resolution commands are modified for the configuration mode class.

Example

- This command enters policy-map-class configuration mode for the copp-system-lacp static class.

```
switch(config-pmap-copp-system-policy)#class copp-system-lacp
switch(config-pmap-c-copp-system-policy-copp-system-lacp)#
```

Two traffic resolution commands determine bandwidth parameters for class traffic:

- **bandwidth (policy-map-class (control-plane) – Arad)** specifies the minimum bandwidth.
- **shape (policy-map-class (control-plane) – Arad)** specifies the maximum bandwidth.

Example

- These commands configure a bandwidth range of 2000 to 4000 packets per seconds (pps) for traffic filtered by the copp-system-lacp class map:

```
switch(config-pmap-c-copp-system-policy-copp-system-lacp)#bandwidth kbps 2000
switch(config-pmap-c-copp-system-policy-copp-system-lacp)#shape kbps 4000
switch(config-pmap-c-copp-system-policy-copp-system-lacp)#
```

Policy-map and policy-map-class configuration modes are group-change modes. Changes are saved with the **exit** command or discarded with the **abort** command. The **show active** command displays the saved version of policy map. The **show pending** command displays the modified policy map.

Example

- These commands exit policy-map-class configuration mode, display the pending policy-map, then exit policy-map configuration mode, which saves the altered policy map to ***running-config***.

```
switch(config-pmap-c-copp-system-policy-copp-system-lacp)#exit
switch(config-pmap-copp-system-policy)#show pending
policy-map type control-plane copp-system-policy
  class copp-system-bpdu

  class copp-system-lldp

  class copp-system-lacp
    shape kbps 4000
    bandwidth kbps 2000

  class copp-system-l3ttl1

  class copp-system-l3slowpath
            <-------OUTPUT OMITTED FROM EXAMPLE-------->

switch(config-pmap-copp-system-policy)#exit
switch(config)#
```

Applying Policy Maps to the Control Plane

The **copp-system-policy** policy map is always applied to the control plane. No commands are available to add or remove this assignment.

Displaying Policy Maps

The **show policy-map interface type qos** command displays the configured values of the policy map's classes and the number of packets filtered and dropped as a result of the class maps.

Example

- These commands exit policy-map-class configuration mode, display the pending policy-map, then exit policy-map configuration mode, which saves the altered policy map to ***running-config***.

```
switch(config)#show policy-map interface control-plane copp-system-policy
Service-policy input: copp-system-policy
  Hardware programming status: InProgress

  Class-map: copp-system-mlag (match-any)
       shape : 10000001 kbps
       bandwidth : 10000001 kbps
      Out Packets : 0
      Drop Packets : 0

  Class-map: copp-system-bpdu (match-any)
       shape : 2604 kbps
       bandwidth : 1302 kbps
      Out Packets : 0
      Drop Packets : 0

  Class-map: copp-system-lacp (match-any)
       shape : 4230 kbps
       bandwidth : 2115 kbps
      Out Packets : 0
      Drop Packets : 0

              <-------OUTPUT OMITTED FROM EXAMPLE-------->

switch(config)#


switch(config-pmap-c-copp-system-policy-copp-system-lacp)#exit
```

26.2.2 Configuring QoS Traffic Policies – Arad Platform Switches

QoS traffic policies are not supported on Arad platform switches.

26.2.3 Configuring PBR Policies – Arad Platform Switches

Policy Based Routing (PBR) is implemented by creating class maps and policy maps, then applying the policy maps to Ethernet interfaces, port channel interfaces or switch virtual interfaces (SVIs).

Creating PBR Class Maps

PBR policies utilize class maps that are created and modified in class-map configuration mode. The **class-map type pbr** command enters class-map configuration mode.

Example

- This command enters class-map configuration mode to create a PBR class map named CMAP1.

```
switch(config)#class-map type pbr match-any CMAP1
switch(config-cmap-PBR-CMAP1)#
```

A class map contains one or more access control lists (ACLs). The **match (policy-map (pbr))** command assigns an ACL to the class map. Subsequent **match** commands add additional ACLs to the class map. Class maps filter traffic only on ACL permit rules. Deny ACL rules are disregarded; if a class map includes ACLs with deny rules, the configuration reverts to its previous state.

Example

- This command adds the ACL named ACL1 to the class map.

```
switch(config-cmap-PBR-CMAP1)#match ip access-group ACL1
switch(config-cmap-PBR-CMAP1)#
```

Class-map configuration mode is a group-change mode. Changes made in a group-change mode are saved by exiting the mode. The **show active** command displays the saved version of class map.

- The **show active** command indicates that the configuration mode class map is not stored in ***running-config***.

```
switch(config-cmap-PBR-CMAP1)#show active
switch(config-cmap-PBR-CMAP1)#
```

The **exit** command returns the switch to global configuration mode and saves pending class map changes. The **abort** command returns the switch to global configuration mode and discards pending changes.

Example

- This command exits class-map configuration mode and stores pending changes to ***running-config***.

```
switch(config-cmap-PBR-CMAP1)#exit
switch(config)#show class-map type pbr CMAP1
class-map type pbr match-any CMAP1
   10 match ip access-group ACL1
switch(config)#
```

Creating PBR Policy Maps

Policy maps are created and modified in policy-map configuration mode. The **policy-map type pbr** command enters policy-map configuration mode.

Example

- This command enters policy-map configuration mode for creating a PBR policy map named PMAP1.

```
switch(config)#policy-map type pbr PMAP1
switch(config-pmap-PMAP1)#
```

Policy map are edited by adding or removing classes. A class automatically contains its eponymous class map; next-hop commands are added or edited in policy-map-class configuration mode. The **class (policy-map (pbr))** command adds a class to the configuration mode policy map and places the switch in policy-map-class configuration mode, where next-hop commands are added to the class.

Example

- This command adds the CMAP1 class to the policy map and places the switch in policy-map-class configuration mode.

```
switch(config-pmap-PMAP1)#class CMAP1
switch(config-pmap-c-PMAP1-CMAP1)#
```

The **set nexthop (policy-map-class – pbr)** command configures the next hop for data that passes the class map.

- This command configures the policy map to set the next hop to 10.12.0.5 on packets filtered by the class map.

```
switch(config-pmap-c-PMAP1-CMAP1)#set nexthop 10.12.0.5
switch(config-pmap-c-PMAP1-CMAP1)#
```

The **set nexthop-group (policy-map-class(pbr) – Arad)** command configures a nexthop group as the next hop for data that passes the class map.

- These commands configure the policy map PMAP1 to set the next hop to a nexthop group named GROUP1 for traffic defined by class map CMAP1.

```
switch(config)#policy-map type pbr PMAP1
switch(config-pmap-PMAP1)#class CMAP1
switch(config-pmap-c-PMAP1-CMAP1)#set nexthop-group GROUP1
switch(config-pmap-c-PMAP1-CMAP1)#
```

Policy-map and policy-map-class configuration modes are group-change modes. Changes are saved with the **exit** command or discarded with the **abort** command. The **show active** command displays the currently saved map version.

Example

- These commands exit policy-map-class configuration mode, then exit policy-map configuration mode to save the altered policy map to ***running-config***.

```
switch(config-pmap-c-PMAP1-CMAP1)#exit
switch(config-pmap-PMAP1)#exit
switch(config)#
```

Applying a PBR Policy Map to an Interface

The **service-policy type pbr (Interface mode)** command applies the specified PBR policy map to the configuration mode interface. Only one PBR service policy is supported per interface.

- These commands apply the PMAP1 PBR policy map to Ethernet interface 8.

```
switch(config)#interface ethernet 8
switch(config-if-Et8)#service-policy type pbr input PMAP1
switch(config-if-Et8)#
```

Hardware Decapsulation

When hardware decapsulation takes place, PBR policy maps on Arad platform switches match on outer packet headers (i.e., they match based on the attributes of the packet before it is decapsulated).

26.3 Traffic Management Configuration – FM6000 Platform Switches

Traffic policies are implemented by policy maps, which are applied to the control plane or an interface. Policy maps contain classes, which are composed of class maps and traffic resolution commands. Section 26.1 describes traffic policies.

FM6000 platform switches support the following traffic policies

- Control plane policies manage control plane traffic.
- QoS traffic policies manage traffic on Ethernet and port channel interfaces.

These sections describe the construction and application of policy maps on FM6000 platform switches:

- Section 26.3.1: Configuring Control Plane Traffic Policies – FM6000 Platform Switches
- Section 26.3.2: Configuring QoS Traffic Policies – FM6000 Platform Switches
- Section 26.3.3: Configuring PBR Policies – FM6000 Platform Switches

26.3.1 Configuring Control Plane Traffic Policies – FM6000 Platform Switches

Note On hardware purchased after July 3, 2017, CoPP is not available.

Default control plane traffic policies are implemented automatically without user intervention. These policies are modified by associating traffic resolution commands with static classes that comprise the control plane policy map.

Static Class Maps

Control plane traffic policies utilize static class maps, which are provided by the switch, are not editable, and cannot be deleted.

Editing the Policy Map

The only control plane policy map is **copp-system-policy**, which cannot be deleted. In its default form, **copp-system-policy** consists of the classes listed in Table 26-2. Although the underlying class map of each class cannot be edited, the traffic resolution commands can be adjusted. The default classes cannot be removed from the policy map and their sequence within the policy map is not editable.

Table 26-2 copp-system-policy default classes: FM6000 Platform Switches

Class Name	shape (pps)	bandwidth (pps)
copp-system-arp	10000	1000
copp-system-default	8000	1000
copp-system-ipmcrsvd	10000	1000
copp-system-ipmcmiss	10000	1000
copp-system-igmp	10000	1000
copp-system-l2rsvd	10000	10000
copp-system-l3slowpath	10000	1000
copp-system-pim-ptp	10000	1000
copp-system-ospf-isis	10000	1000
copp-system-selfip	5000	5000

Table 26-2 copp-system-policy default classes: FM6000 Platform Switches

Class Name	shape (pps)	bandwidth (pps)
copp-system-selfip-tc6to7	5000	5000
copp-system-sflow	25000	1000

Policy maps are modified in policy-map configuration mode. The **policy-map type control-plane** command enters policy-map configuration mode.

Example

- This command enters policy-map configuration mode for editing ***copp-system-policy***.

```
switch(config)#policy-map type control-plane copp-system-policy
switch(config-pmap-copp-system-policy)#
```

The **class (policy-map (control-plane) – FM6000)** command enters policy-map-class configuration mode, where traffic resolution commands are modified for the configuration mode class.

Example

- This command enters policy-map-class configuration mode for the copp-system-arp static class.

```
switch(config-pmap-copp-system-policy)#class copp-system-arp
switch(config-pmap-c-copp-system-policy-copp-system-arp)#
```

Two traffic resolution commands determine bandwidth parameters for class traffic:

- **bandwidth (policy-map-class (control-plane) – FM6000)** specifies the minimum bandwidth.
- **shape (policy-map-class (control-plane) – FM6000)** specifies the maximum bandwidth.

Example

- These commands configure a bandwidth range of 2000 to 4000 packets per seconds (pps) for traffic filtered by the copp-system-arp class map:

```
switch(config-pmap-c-copp-system-policy-copp-system-arp)#bandwidth pps 2000
switch(config-pmap-c-copp-system-policy-copp-system-arp)#shape pps 4000
switch(config-pmap-c-copp-system-policy-copp-system-arp)#
```

Policy-map and policy-map-class configuration modes are group-change modes. Changes are saved with the **exit** command or discarded with the **abort** command. The **show active** command displays the saved version of policy map. The **show pending** command displays the modified policy map.

Example

- These commands exit policy-map-class configuration mode, display the pending policy-map, then exit policy-map configuration mode, which saves the altered policy map to ***running-config***.

```
switch(config-pmap-c-copp-system-policy-CP-CMAP_1)#exit
switch(config-pmap-copp-system-policy)#show pending
policy-map type control-plane copp-system-policy
  class CP-CMAP_1
    shape pps 4000
    bandwidth pps 2000

  class copp-system-bpdu

  class copp-system-lldp

  class copp-system-lacp

  class copp-system-arp

            <-------OUTPUT OMITTED FROM EXAMPLE-------->

  class copp-system-arpresolver

  class copp-system-default

switch(config-pmap-copp-system-policy)#exit
switch(config)#
```

Applying Policy Maps to the Control Plane

The **copp-system-policy** policy map is always applied to the control plane. No commands are available to add or remove this assignment.

26.3.2 Configuring QoS Traffic Policies – FM6000 Platform Switches

QoS traffic policies are implemented by creating class maps and policy maps, then applying the policy maps to Ethernet and port channel interfaces.

Creating Class Maps

QoS traffic policies utilize dynamic class maps that are created and modified in class-map configuration mode. The **class-map type qos** command enters class-map configuration mode.

Example

- This command enters class-map configuration mode to create QoS class map named Q-CMap_1.

```
switch(config)#class-map type qos match-any Q-CMap_1
switch(config-cmap-Q-CMap_1)#
```

A class map contains one IPv4 access control list (ACL). The **match (class-map (qos) – FM6000)** command assigns an ACL to the class map. Subsequent **match** commands replace the existing **match** command. Class maps filter traffic only on ACL permit rules. Deny ACL rules are disregarded.

Example

- This command adds the IPv4 ACL named ACL_1 to the class map.

```
switch(config-cmap-Q-CMap_1)#match ip access-group ACL_1
switch(config-cmap-Q-CMap_1)#
```

Class-map configuration mode is a group-change mode. Changes made in a group-change mode are saved by exiting the mode. The **show active** command displays the saved version of class map. The **show pending** command displays the unsaved class map.

Example

- The **show active** command indicates that the configuration mode class map is not stored in ***running-config***. The **show pending** command displays the class map to be stored upon exiting class-map configuration mode.

```
switch(config-cmap-Q-CMap_1)#show active
switch(config-cmap-Q-CMap_1)#show pending
class-map type qos match-any Q-CMap_1
   match ip access-group ACL_1

switch(config-cmap-Q-CMap_1)#
```

The **exit** command returns the switch to global configuration mode and saves pending class map changes. The **abort** command returns the switch to global configuration mode and discards pending changes.

Example

- This command exits class-map configuration mode and stores pending changes to ***running-config***.

```
switch(config-cmap-CP-CMAP_1)#exit
switch(config)#show class-map type control-plane CP-CMAP_1
  Class-map: CP-CMAP_1 (match-any)
    Match: ip access-group name ACLv4_1
switch(config)#
```

Creating Policy Maps

Policy maps are created and modified in policy-map configuration mode. The **policy-map type qos** command enters policy-map configuration mode.

Example

- This command places the switch in policy-map configuration mode and creates a QoS policy map named Q-PMAP_1.

```
switch(config)#policy-map type qos Q-PMAP_1
switch(config-pmap-Q-PMAP_1)#
```

Policy map are edited by adding or removing classes. A class automatically contains its eponymous class map; traffic resolution commands are added or edited in policy-map-class configuration mode. The **class (policy-map (qos) – FM6000)** command adds a class to the configuration mode policy map and places the switch in policy-map-class configuration mode, where traffic resolution commands are added to the class.

Example

- This command adds the Q-CMap_1 class to the Q-PMAP_1 policy map and places the switch in policy-map-class configuration mode.

```
switch(config-pmap-Q-PMAP_1)#class Q-CMap_1
switch(config-pmap-c-Q-PMAP_1-Q-CMap_1)#
```

set (policy-map-class (qos) – FM6000) commands configure traffic resolution methods for data that passes the class map:

- **set cos** sets the layer 2 CoS field.

- **set dscp** sets the DSCP value in the ToS byte.
- **set traffic class** specifies a traffic class queue.

Example

- These commands configure the policy map to set the CoS field to 7 on packets filtered by the class map, then assigns those packets to traffic class 4.

```
switch(config-pmap-c-Q-PMAP_1-Q-CMap_1)#set cos 7
switch(config-pmap-c-Q-PMAP_1-Q-CMap_1)#set traffic-class 4
switch(config-pmap-c-Q-PMAP_1-Q-CMap_1)#
```

Policy-map and policy-map-class configuration modes are group-change modes. Changes are saved with the **exit** command or discarded with the **abort** command. The **show active** and **show pending** commands display the saved and modified policy map versions, respectively.

Example

- These commands exit policy-map-class configuration mode, display the pending policy-map, then exit policy-map configuration mode to save the altered policy map to ***running-config***.

```
switch(config-pmap-c-Q-PMAP_1-Q-CMap_1)#exit
switch(config-pmap-Q-PMAP_1)#show pending
policy-map type qos Q-PMAP_1
  class Q-CMap_1
    set cos 7
    set traffic-class 4

  class class-default

switch(config-pmap-Q-PMAP_1)#exit
switch(config)#
```

The last class in all QoS policy maps is ***class-default***. The ***class-default*** class map matches all traffic except IPv4 or IPv6 traffic and provides no traffic resolution commands. The ***class-default*** class map is not editable; traffic resolution commands can be added to the ***class-default*** class.

To modify traffic resolution commands for the ***class-default*** class, enter policy-map-class configuration mode for the class, then enter the desired **set** commands.

Example

- These commands enter policy-map-class configuration mode for ***class-default***, configures the stream to enter traffic class 2, and saves the altered policy map to ***running-config***.

```
switch(config)#policy-map type qos Q-PMap_1
switch(config-pmap-Q-PMap_1)#class class-default
switch(config-pmap-c-Q-PMap_1-class-default)#set traffic-class 2
switch(config-pmap-c-Q-PMap_1-class-default)#exit
switch(config-pmap-Q-PMap_1)#exit
switch(config)#show policy-map type qos Q-PMap_1
Service-policy Q-PMap_1

  Class-map: Q-CMap_1 (match-any)
    Match: ipv6 access-group name ACLv6_1
       set cos 7
       set traffic-class 4

  Class-map: class-default (match-any)
       set traffic-class 2

switch(config)#
```

Applying Policy Maps to an Interface

The **service-policy type qos (Interface mode)** command applies a specified policy map to the configuration mode interface.

- These commands apply PMAP-1 policy map to Ethernet interface 8.

```
switch(config)#interface ethernet 8
switch(config-if-Et8)#show active
switch(config-if-Et8)#service-policy input PMAP-1
switch(config-if-Et8)#show active
interface Ethernet8
   service-policy type qos input PMAP-1
switch(config-if-Et8)#
```

26.3.3 Configuring PBR Policies – FM6000 Platform Switches

Policy Based Routing (PBR) is implemented by creating class maps and policy maps, then applying the policy maps to Ethernet interfaces, port channel interfaces or switch virtual interfaces (SVIs).

Creating PBR Class Maps

PBR policies utilize class maps that are created and modified in class-map configuration mode. The **class-map type pbr** command enters class-map configuration mode.

Example

- This command enters class-map configuration mode to create a PBR class map named CMAP1.

```
switch(config)#class-map type pbr match-any CMAP1
switch(config-cmap-PBR-CMAP1)#
```

A class map contains one or more IPv4 access control lists (ACLs). The **match (policy-map (pbr))** command assigns an ACL to the class map. Subsequent **match** commands add additional ACLs to the class map. Class maps filter traffic only on ACL permit rules. Deny ACL rules are disregarded; if a class map includes ACLs with deny rules, the configuration reverts to its previous state.

On FM6000 platform switches, counters are not supported, so a **statistics per-entry (ACL configuration modes)** command in an ACL is ignored.

Example

- This command adds the IPv4 ACL named ACL1 to the class map.

```
switch(config-cmap-PBR-CMAP1)#match ip access-group ACL1
switch(config-cmap-PBR-CMAP1)#
```

Class-map configuration mode is a group-change mode. Changes made in a group-change mode are saved by exiting the mode. The **show active** command displays the saved version of class map.

- The **show active** command indicates that the configuration mode class map is not stored in ***running-config***.

```
switch(config-cmap-PBR-CMAP1)#show active
switch(config-cmap-PBR-CMAP1)#
```

The **exit** command returns the switch to global configuration mode and saves pending class map changes. The **abort** command returns the switch to global configuration mode and discards pending changes.

Example

- This command exits class-map configuration mode and stores pending changes to ***running-config***.

```
switch(config-cmap-PBR-CMAP1)#exit
switch(config)#show class-map type pbr CMAP1
class-map type pbr match-any CMAP1
   10 match ip access-group ACL1
switch(config)#
```

Creating PBR Policy Maps

Policy maps are created and modified in policy-map configuration mode. The **policy-map type pbr** command enters policy-map configuration mode.

Example

- This command enters policy-map configuration mode for creating a PBR policy map named PMAP1.

```
switch(config)#policy-map type pbr PMAP1
switch(config-pmap-PMAP1)#
```

Policy map are edited by adding or removing classes. A class automatically contains its eponymous class map; next-hop commands are added or edited in policy-map-class configuration mode. The **class (policy-map (pbr))** command adds a class to the configuration mode policy map and places the switch in policy-map-class configuration mode, where next-hop commands are added to the class.

Example

- This command adds the CMAP1 class to the policy map and places the switch in policy-map-class configuration mode.

```
switch(config-pmap-PMAP1)#class CMAP1
switch(config-pmap-c-PMAP1-CMAP1)#
```

The **set nexthop (policy-map-class – pbr)** command configures the next hop for data that passes the class map.

- This command configures the policy map to set the next hop to 10.12.0.5 on packets filtered by the class map.

```
switch(config-pmap-c-PMAP1-CMAP1)#set nexthop 10.12.0.5
switch(config-pmap-c-PMAP1-CMAP1)#
```

Policy-map and policy-map-class configuration modes are group-change modes. Changes are saved with the **exit** command or discarded with the **abort** command. The **show active** command displays the currently saved map version.

Example

- These commands exit policy-map-class configuration mode, then exit policy-map configuration mode to save the altered policy map to ***running-config***.

```
switch(config-pmap-c-PMAP1-CMAP1)#exit
switch(config-pmap-PMAP1)#exit
switch(config)#
```

Applying a PBR Policy Map to an Interface

The **service-policy type pbr (Interface mode)** command applies the specified PBR policy map to the configuration mode interface. Only one PBR service policy is supported per interface.

- These commands apply the PMAP1 PBR policy map to Ethernet interface 8.

```
switch(config)#interface ethernet 8
switch(config-if-Et8)#service-policy type pbr input PMAP1
switch(config-if-Et8)#
```

Hardware Decapsulation

When hardware decapsulation takes place, PBR policy maps on FM6000 platform switches match on outer packet headers (i.e., they match based on the attributes of the packet before it is decapsulated).

26.4 Traffic Management Configuration – Petra Platform Switches

Traffic policies are implemented by policy maps, which are applied to the control plane. Policy maps contain classes, which are composed of class maps and traffic resolution commands. QoS traffic policies are not supported on 7500 Series switches.

Section 26.1 describes traffic policies.

26.4.1 Configuring Control Plane Traffic Policies – Petra Platform Switches

Note On hardware purchased after July 3, 2017, CoPP is not available.

Default control plane traffic policies are implemented automatically without user intervention. These policies are modified by associating traffic resolution commands with static classes that comprise the control plane policy map.

Static Class Maps

Control plane traffic policies utilize static class maps, which are provided by the switch, are not editable, and cannot be deleted.

Editing the Policy Map

The only control plane policy map is **copp-system-policy**, which cannot be deleted. In its default form, **copp-system-policy** consists of the classes listed in Table 26-3. Although the underlying class map of each class cannot be edited, the traffic resolution commands can be adjusted. The default classes cannot be removed from the policy map and their sequence within the policy map is not editable.

Table 26-3 copp-system-policy default classes: Petra Platform Switches

Class Name	shape (pps)	bandwidth (pps)
copp-system-bpdu	2500	1250
copp-system-default	2500	250
copp-system-igmp	2500	250
copp-system-ipbroadcast	2500	250
copp-system-ipmc	2500	250
copp-system-ipmcmiss	2500	250
copp-system-ipmcrsvd	2500	250
copp-system-ipunicast	NO LIMIT	250
copp-system-l3destmiss	2500	250
copp-system-l3slowpath	2500	250
copp-system-l3ttl0	2500	250
copp-system-l3ttl1	2500	250
copp-system-lacp	2500	1250
copp-system-lldp	2500	250
copp-system-unicast-arp	2500	250

Policy maps are modified in policy-map configuration mode. The **policy-map type control-plane** command enters policy-map configuration mode.

Example

- This command enters policy-map configuration mode for editing ***copp-system-policy***.

```
switch(config)#policy-map type control-plane copp-system-policy
switch(config-pmap-copp-system-policy)#
```

The **class (policy-map (control-plane) – Petra)** command enters policy-map-class configuration mode, where traffic resolution commands are modified for the configuration mode class.

Example

- This command enters policy-map-class configuration mode for the copp-system-lldp static class.

```
switch(config-pmap-copp-system-policy)#class copp-system-lldp
switch(config-pmap-c-copp-system-policy-copp-system-lldp)#
```

Two traffic resolution commands determine bandwidth parameters for class traffic:

- **bandwidth (policy-map-class (control-plane) – Petra)** specifies the minimum bandwidth.
- **shape (policy-map-class (control-plane) – Petra)** specifies the maximum bandwidth.

Example

- These commands configure a bandwidth range of 2000 to 4000 packets per seconds (pps) for traffic filtered by the copp-system-arp class map:

```
switch(config-pmap-c-copp-system-policy-copp-system-lldp)#bandwidth kbps 2000
switch(config-pmap-c-copp-system-policy-copp-system-lldp)#shape kbps 4000
switch(config-pmap-c-copp-system-policy-copp-system-lldp)#
```

Policy-map and policy-map-class configuration modes are group-change modes. Changes are saved with the **exit** command or discarded with the **abort** command. The **show active** command displays the saved version of policy map. The **show pending** command displays the configured policy map.

Petra platform switches do not support all discrete rate values. When a **bandwidth** or **shape** command specifies a value that is not supported, the switch converts the rate to the next highest discrete value that it supports. The **show policy-map interface type qos** command displays the converted rate and not the user configured rate.

Example

- These commands exit policy-map-class configuration mode, display the pending policy-map, then exit policy-map configuration mode, which saves the altered policy map to ***running-config***.

```
switch(config-pmap-c-copp-system-policy-copp-system-lacp)#exit
switch(config-pmap-copp-system-policy)#show pending
policy-map type control-plane copp-system-policy
  class copp-system-bpdu

  class copp-system-lldp
    shape kbps 4000
    bandwidth kbps 2000

  class copp-system-lacp



switch(config-pmap-copp-system-policy)#exit
switch(config)#
```

Changes are saved with the **exit** command or discarded with the **abort** command. The **show active** command displays the saved version of policy map. The **show pending** command displays the modified policy map.

Displaying Policy Maps

The **show policy-map interface type qos** command displays the traffic resolution rates of the policy map's classes and the number of packets filtered and dropped as a result of the class maps. The shape and bandwidth rates may differ from configured values, because the switch does not support all discrete rate values.

Example

- These commands exit policy-map-class configuration mode, display the pending policy-map, then exit policy-map configuration mode, which saves the altered policy map to ***running-config***.

```
switch(config)#show policy-map interface control-plane copp-system-policy
Service-policy input: copp-system-policy
  Hardware programming status: InProgress

  Class-map: copp-system-mlag (match-any)
       shape : 10000001 kbps
       bandwidth : 10000001 kbps
      Out Packets : 0
      Drop Packets : 0

             <-------OUTPUT OMITTED FROM EXAMPLE-------->

  Class-map: copp-system-lacp (match-any)
       shape : 2604 kbps
       bandwidth : 1302 kbps
      Out Packets : 0
      Drop Packets : 0

             <-------OUTPUT OMITTED FROM EXAMPLE-------->

switch(config)#
```

Applying Policy Maps to the Control Plane

The **copp-system-policy** policy map is always applied to the control plane. No commands are available to add or remove this assignment.

26.4.2 Configuring QoS Traffic Policies – Petra Platform Switches

QoS traffic policies are not supported on Petra platform switches.

26.4.3 Configuring PBR Policies – Petra Platform Switches

PBR policies are not supported on Petra platform switches.

26.5 Traffic Management Configuration – Trident Platform Switches

Traffic policies are implemented by policy maps, which are applied to the control plane or an interface. Policy maps contain classes, which are composed of class maps and traffic resolution commands. Section 26.1 describes traffic policies.

Trident platform switches support the following traffic policies

- Control plane policies manage control plane traffic.
- QoS traffic policies manage traffic on Ethernet and port channel interfaces.

These sections describe the construction and application of policy maps:

- Section 26.5.1: Configuring Control Plane Traffic Policies – Trident Platform Switches
- Section 26.5.2: Configuring QoS Traffic Policies – Trident Platform Switches
- Section 26.5.3: Configuring PBR Policies – Trident Platform Switches

26.5.1 Configuring Control Plane Traffic Policies – Trident Platform Switches

Note On hardware purchased after July 3, 2017, CoPP is not available.

Default control plane traffic policies are implemented automatically without user intervention. These policies are modified by creating class maps and editing the policy map to include the new class maps.

Creating Class Maps

Control plane traffic policies utilize static and dynamic class maps. Static class maps are provided by the switch, are not editable, and cannot be deleted. Dynamic class maps are created and modified in class-map configuration mode. The **class-map type control-plane** command enters class-map configuration mode.

Example

- This command enters class-map configuration mode for creating or editing a control plane dynamic class map named CP-CMAP_1.

```
switch(config)#class-map type control-plane match-any CP-CMAP_1
switch(config-cmap-CP-CMAP_1)#
```

Class maps contain one IPv4 or IPv6 access control list (ACL). The **match (class-map (control-plane) – Trident)** command assigns an ACL to the class map. Subsequent **match** commands replace the existing **match** command. Class maps filter traffic only on ACL permit rules. Deny ACL rules are disregarded.

Example

- This command assigns the IPv4 ACL named ACLv4_1 to the class map.

```
switch(config-cmap-CP-CMAP_1)#match ip access-group ACLv4_1
switch(config-cmap-CP-CMAP_1)#
```

Class-map configuration mode is a group-change mode. Changes are saved by exiting the mode. The **show active** command displays the saved version of class map. The **show pending** command displays the unsaved class map.

Example

- The **show active** command indicates that the configuration mode class map is not stored in ***running-config***. The **show pending** command displays the class map to be stored upon exiting class-map configuration mode.

```
switch(config-cmap-CP-CMAP_1)#show active
switch(config-cmap-CP-CMAP_1)#show pending
class-map type control-plane match-any CP-CMAP_1
   match ip access-group ACLv4_1

switch(config-cmap-CP-CMAP_1)#
```

The **exit** command returns the switch to global configuration mode and saves pending class map changes. The **abort** command returns the switch to global configuration mode and discards pending class map changes.

Example

- This command exits class-map configuration mode and stores pending changes to ***running-config***.

```
switch(config-cmap-CP-CMAP_1)#exit
switch(config)#show class-map type control-plane CP-CMAP_1
  Class-map: CP-CMAP_1 (match-any)
    Match: ip access-group name ACLv4_1
switch(config)#
```

Editing the Policy Map

The only control plane policy map is **copp-system-policy**, which cannot be deleted. In its default form, **copp-system-policy** consists of the classes listed in Table 26-4. Although the underlying class map of each class cannot be edited, the traffic resolution commands can be adjusted. The default classes cannot be removed from the policy map and their sequence within the policy map is not editable.

Table 26-4 copp-system-policy default classes: Trident Platform Switches

Class Name	shape (pps)	bandwidth (pps)
copp-system-bpdu	5000	5000
copp-system-lacp	5000	5000
copp-system-selfip-tc6to7	5000	5000
copp-system-selfip	5000	5000
copp-system-tc6to7	10000	1000
copp-system-lldp	10000	1000
copp-system-ipmcrsvd	10000	1000
copp-system-igmp	10000	1000
copp-system-ipmcmiss	10000	1000
copp-system-glean	10000	1000
copp-system-tc3to5	10000	1000
copp-system-arp	10000	1000
copp-system-arpresolver	10000	1000
copp-system-l3destmiss	10000	1000
copp-system-l3slowpath	10000	1000
copp-system-l3ttl1	10000	1000

Table 26-4 copp-system-policy default classes: Trident Platform Switches

Class Name	shape (pps)	bandwidth (pps)
copp-system-default	8000	1000
copp-system-acllog	10000	1000
copp-system-sflow	25000	0

Policy maps are modified in policy-map configuration mode. The **policy-map type control-plane** command enters policy-map configuration mode.

Example

- This command enters policy-map configuration mode for editing ***copp-system-policy***.

```
switch(config)#policy-map type control-plane copp-system-policy
switch(config-pmap-copp-system-policy)#
```

Dynamic classes are inserted in front of the static classes. Classes automatically contain their eponymous class map; traffic resolution commands are created or edited in policy-map-class configuration mode. The **class (policy-map (control-plane) – Trident)** command adds a class to the policy map and places the switch in policy-map-class configuration mode, where traffic resolution commands are added to the class.

Example

- This command adds the CP-CMAP_1 class to the copp-system-policy policy map and places the switch in policy-map-class configuration mode.

```
switch(config-pmap-copp-system-policy)#class CP-CMAP_1
switch(config-pmap-c-copp-system-policy-CP-CMAP_1)#
```

Two traffic resolution commands determine bandwidth parameters for class traffic:

- **bandwidth (policy-map-class (control-plane) – Trident)** specifies the minimum bandwidth.
- **shape (policy-map-class (control-plane) – Trident)** specifies the maximum bandwidth.

Example

- These commands configure a bandwidth range of 2000 to 4000 packets per seconds (pps) for traffic filtered by the CP-CMAP_1 class map:

```
switch(config-pmap-c-copp-system-policy-CP-CMAP_1)#bandwidth pps 2000
switch(config-pmap-c-copp-system-policy-CP-CMAP_1)#shape pps 4000
switch(config-pmap-c-copp-system-policy-CP-CMAP_1)#
```

Policy-map and policy-map-class configuration modes are group-change modes. Changes are saved with the **exit** command or discarded with the **abort** command. The **show active** command displays the saved version of policy map. The **show pending** command displays the modified policy map.

Example

- These commands exit policy-map-class configuration mode, display the pending policy-map, then exit policy-map configuration mode, which saves the altered policy map to ***running-config***.

```
switch(config-pmap-c-copp-system-policy-CP-CMAP_1)#exit
switch(config-pmap-copp-system-policy)#show pending
policy-map type control-plane copp-system-policy
  class CP-CMAP_1
    shape pps 4000
    bandwidth pps 2000

  class copp-system-bpdu

  class copp-system-lldp

  class copp-system-lacp

  class copp-system-arp



  class copp-system-arpresolver

  class copp-system-default

switch(config-pmap-copp-system-policy)#exit
switch(config)#
```

To modify traffic resolution commands for a static class, enter policy-map-class configuration mode for the class, then enter the desired **bandwidth** and **shape** commands.

Example

- These commands enter policy-map-class configuration mode for copp-system-bpdu class, change the bandwidth range for the class, then save the altered policy map to ***running-config***.

```
switch(config)#policy-map type control-plane copp-system-policy
switch(config-pmap-copp-system-policy)#class copp-system-bpdu
switch(config-pmap-c-copp-system-policy-copp-system-bpdu)#shape pps 200
switch(config-pmap-c-copp-system-policy-copp-system-bpdu)#bandwidth pps 100
switch(config-pmap-c-copp-system-policy-copp-system-bpdu)#exit
switch(config-pmap-copp-system-policy)#show pending
policy-map type control-plane copp-system-policy
  class CP-CMAP_1
    shape pps 4000
    bandwidth pps 2000

  class copp-system-bpdu
    shape pps 200
    bandwidth pps 100

  class copp-system-lldp

              <-------OUTPUT OMITTED FROM EXAMPLE-------->
switch(config-pmap-copp-system-policy)#exit
switch(config)#
```

Applying Policy Maps to the Control Plane

The **copp-system-policy** policy map is always applied to the control plane. No commands are available to add or remove this assignment.

26.5.2 Configuring QoS Traffic Policies – Trident Platform Switches

QoS traffic policies are implemented by creating class maps and policy maps, then applying the policy maps to Ethernet and port channel interfaces.

Creating Class Maps

QoS traffic policies utilize dynamic class maps that are created and modified in class-map configuration mode. The **class-map type qos** command enters class-map configuration mode.

Example

- This command enters class-map configuration mode to create QoS class map named Q-CMap_1.

```
switch(config)#class-map type qos match-any Q-CMap_1
switch(config-cmap-Q-CMap_1)#
```

A class map contains one IPv4 or IPv6 access control list (ACL). The **match (class-map (qos) – Trident)** command assigns an ACL to the class map. Subsequent **match** commands replace the existing **match** command. Class maps filter traffic only on ACL permit rules. Deny ACL rules are disregarded.

Example

- This command adds the IPv6 ACL named ACLv6_1 to the class map.

```
switch(config-cmap-Q-CMap_1)#match ipv6 access-group ACLv6_1
switch(config-cmap-Q-CMap_1)#
```

Class-map configuration mode is a group-change mode. Changes made in a group-change mode are saved by exiting the mode. The **show active** command displays the saved version of class map. The **show pending** command displays the unsaved class map.

Example

- The **show active** command indicates that the configuration mode class map is not stored in ***running-config***. The **show pending** command displays the class map to be stored upon exiting class-map configuration mode.

```
switch(config-cmap-Q-CMap_1)#show active
switch(config-cmap-Q-CMap_1)#show pending
class-map type qos match-any Q-CMap_1
   match ipv6 access-group ACLv6_1

switch(config-cmap-Q-CMap_1)#
```

The **exit** command returns the switch to global configuration mode and saves pending class map changes. The **abort** command returns the switch to global configuration mode and discards pending class map changes.

Example

- This command exits class-map configuration mode and stores pending changes to ***running-config***.

```
switch(config-cmap-CP-CMAP_1)#exit
switch(config)#show class-map type control-plane CP-CMAP_1
  Class-map: CP-CMAP_1 (match-any)
    Match: ip access-group name ACLv4_1
switch(config)#
```

Creating Policy Maps

Policy maps are created and modified in policy-map configuration mode. The **policy-map type qos** command enters policy-map configuration mode.

Example

- This command enters policy-map configuration mode for creating a QoS policy map named Q-PMAP_1.

```
switch(config)#policy-map type qos Q-PMAP_1
switch(config-pmap-Q-PMAP_1)#
```

Policy maps are edited by adding or removing classes. A class automatically contains its eponymous class map; traffic resolution commands are added or edited in policy-map-class configuration mode. The **class (policy-map (qos) – Trident)** command adds a class to the configuration mode policy map and places the switch in policy-map-class configuration mode, where traffic resolution commands are added to the class.

Example

- This command adds the Q-CMap_1 class to the Q-PMAP_1 policy map and places the switch in policy-map-class configuration mode.

```
switch(config-pmap-Q-PMAP_1)#class Q-CMap_1
switch(config-pmap-c-Q-PMAP_1-Q-CMap_1)#
```

The set (policy-map-class (qos) – Trident) command configures traffic resolution methods for data that passes the class map:

- **set cos** sets the layer 2 CoS field.
- **set dscp** sets the DSCP value in the ToS byte.
- **set traffic class** specifies a traffic class queue.

Example

- These commands configure the policy map to set the CoS field to 7 on packets filtered by the class map, then assigns those packets to traffic class 4.

```
switch(config-pmap-c-Q-PMAP_1-Q-CMap_1)#set cos 7
switch(config-pmap-c-Q-PMAP_1-Q-CMap_1)#set traffic-class 4
switch(config-pmap-c-Q-PMAP_1-Q-CMap_1)#
```

Policy-map and policy-map-class configuration modes are group-change modes. Changes are saved with the **exit** command or discarded with the **abort** command. The **show active** and **show pending** commands display the saved and modified policy map versions, respectively.

Example

- These commands exit policy-map-class configuration mode, display the pending policy-map, then exit policy-map configuration mode to ***running-config***.

```
switch(config-pmap-c-Q-PMAP_1-Q-CMap_1)#exit
switch(config-pmap-Q-PMAP_1)#show pending
policy-map type qos Q-PMAP_1
  class Q-CMap_1
    set cos 7
    set traffic-class 4

  class class-default

switch(config-pmap-Q-PMAP_1)#exit
switch(config)#
```

The last class in all QoS policy maps is ***class-default***. The ***class-default*** class map matches all traffic except IPv4 or IPv6 traffic and provides no traffic resolution commands. The ***class-default*** class map is not editable; traffic resolution commands can be added to the ***class-default*** class.

To modify traffic resolution commands for the ***class-default*** class, enter policy-map-class configuration mode for the class, then enter the desired **set** commands.

Example

- These commands enter policy-map-class configuration mode for ***class-default***, configures the stream to enter traffic class 2, and saves the altered policy map to ***running-config***.

```
switch(config)#policy-map type qos Q-PMap_1
switch(config-pmap-Q-PMap_1)#class class-default
switch(config-pmap-c-Q-PMap_1-class-default)#set traffic-class 2
switch(config-pmap-c-Q-PMap_1-class-default)#exit
switch(config-pmap-Q-PMap_1)#exit
switch(config)#show policy-map type qos Q-PMap_1
Service-policy Q-PMap_1

  Class-map: Q-CMap_1 (match-any)
    Match: ipv6 access-group name ACLv6_1
       set cos 7
       set traffic-class 4

  Class-map: class-default (match-any)
       set traffic-class 2

switch(config)#
```

Applying Policy Maps to an Interface

The **service-policy type qos (Interface mode)** command applies a specified policy map to the configuration mode interface.

Example

- These commands apply PMAP-1 policy map to Ethernet interface 8.

```
switch(config)#interface ethernet 8
switch(config-if-Et8)#show active
switch(config-if-Et8)#service-policy input PMAP-1
switch(config-if-Et8)#show active
interface Ethernet8
   service-policy type qos input PMAP-1
switch(config-if-Et8)#
```

26.5.3 Configuring PBR Policies – Trident Platform Switches

Policy Based Routing (PBR) is implemented by creating class maps and policy maps, then applying the policy maps to Ethernet interfaces, port channel interfaces or switch virtual interfaces (SVIs).

Creating PBR Class Maps

PBR policies utilize class maps that are created and modified in class-map configuration mode. The **class-map type pbr** command enters class-map configuration mode.

Example

- This command enters class-map configuration mode to create a PBR class map named CMAP1.

```
switch(config)#class-map type pbr match-any CMAP1
switch(config-cmap-PBR-CMAP1)#
```

A class map contains one or more access control lists (ACLs). The **match (policy-map (pbr))** command assigns an ACL to the class map. Subsequent **match** commands add additional ACLs to the class map. Class maps filter traffic only on ACL permit rules. Deny ACL rules are disregarded; if a class map includes ACLs with deny rules, the configuration reverts to its previous state.

Example

- This command adds the ACL named ACL1 to the class map.

```
switch(config-cmap-PBR-CMAP1)#match ip access-group ACL1
switch(config-cmap-PBR-CMAP1)#
```

Class-map configuration mode is a group-change mode. Changes made in a group-change mode are saved by exiting the mode. The **show active** command displays the saved version of class map.

- The **show active** command indicates that the configuration mode class map is not stored in ***running-config***.

```
switch(config-cmap-PBR-CMAP1)#show active
switch(config-cmap-PBR-CMAP1)#
```

The **exit** command returns the switch to global configuration mode and saves pending class map changes. The **abort** command returns the switch to global configuration mode and discards pending changes.

Example

- This command exits class-map configuration mode and stores pending changes to ***running-config***.

```
switch(config-cmap-PBR-CMAP1)#exit
switch(config)#show class-map type pbr CMAP1
class-map type pbr match-any CMAP1
   10 match ip access-group ACL1
switch(config)#
```

Creating PBR Policy Maps

Policy maps are created and modified in policy-map configuration mode. The **policy-map type pbr** command enters policy-map configuration mode.

Example

- This command enters policy-map configuration mode for creating a PBR policy map named PMAP1.

```
switch(config)#policy-map type pbr PMAP1
switch(config-pmap-PMAP1)#
```

Policy map are edited by adding or removing classes. A class automatically contains its eponymous class map; next-hop commands are added or edited in policy-map-class configuration mode. The **class (policy-map (pbr))** command adds a class to the configuration mode policy map and places the switch in policy-map-class configuration mode, where next-hop commands are added to the class.

Example

- This command adds the CMAP1 class to the policy map and places the switch in policy-map-class configuration mode.

```
switch(config-pmap-PMAP1)#class CMAP1
switch(config-pmap-c-PMAP1-CMAP1)#
```

The **set nexthop (policy-map-class – pbr)** command configures the next hop for data that passes the class map.

- This command configures the policy map to set the next hop to 10.12.0.5 on packets filtered by the class map.

```
switch(config-pmap-c-PMAP1-CMAP1)#set nexthop 10.12.0.5
switch(config-pmap-c-PMAP1-CMAP1)#
```

Policy-map and policy-map-class configuration modes are group-change modes. Changes are saved with the **exit** command or discarded with the **abort** command. The **show active** command displays the currently saved map version.

Example

- These commands exit policy-map-class configuration mode, then exit policy-map configuration mode to save the altered policy map to ***running-config***.

```
switch(config-pmap-c-PMAP1-CMAP1)#exit
switch(config-pmap-PMAP1)#exit
switch(config)#
```

Applying a PBR Policy Map to an Interface

The **service-policy type pbr (Interface mode)** command applies the specified PBR policy map to the configuration mode interface. Only one PBR service policy is supported per interface.

- These commands apply the PMAP1 PBR policy map to Ethernet interface 8.

```
switch(config)#interface ethernet 8
switch(config-if-Et8)#service-policy type pbr input PMAP1
switch(config-if-Et8)#
```

Hardware Decapsulation

When hardware decapsulation takes place, PBR policy maps on Trident platform switches match on inner packet headers (i.e., they match based on the attributes of the decapsulated packet).

26.6 Traffic Management Configuration – Trident-II Platform Switches

Traffic policies are implemented by policy maps, which are applied to the control plane or an interface. Policy maps contain classes, which are composed of class maps and traffic resolution commands. Section 26.1 describes traffic policies.

Trident platform switches support the following traffic policies

- Control plane policies manage control plane traffic.
- QoS traffic policies manage traffic on Ethernet and port channel interfaces.

26.6.1 Configuring Control Plane Traffic Policies – Trident-II Platform Switches

Note On hardware purchased after July 3, 2017, CoPP is not available.

Default control plane traffic policies are implemented automatically without user intervention. These policies are modified by associating traffic resolution commands with static classes that comprise the control plane policy map.

Static Class Maps

Control plane traffic policies utilize static class maps, which are provided by the switch, are not editable, and cannot be deleted.

Editing the Policy Map

The only control plane policy map is **copp-system-policy**, which cannot be deleted. In its default form, **copp-system-policy** consists of the classes listed in Table 26-5. Although the underlying class map of each class cannot be edited, the traffic resolution commands can be adjusted. The default classes cannot be removed from the policy map and their sequence within the policy map is not editable.

Table 26-5 copp-system-policy default classes: Trident-II Platform Switches

Class Name	shape (pps)	bandwidth (pps)
copp-system-acllog	1000	10000
copp-system-arp	1000	10000
copp-system-arpresolver	1000	10000
copp-system-bfd	5000	10000
copp-system-bgp	5000	5000
copp-system-bpdu	5000	5000
copp-system-default	1000	8000
copp-system-glean	1000	10000
copp-system-igmp	1000	10000
copp-system-ipmcmiss	1000	10000
copp-system-ipmcrsvd	1000	10000
copp-system-l3destmiss	1000	10000
copp-system-l3slowpath	1000	10000
copp-system-l3ttl1	1000	10000
copp-system-lacp	5000	5000

Table 26-5 copp-system-policy default classes: Trident-II Platform Switches

Class Name	shape (pps)	bandwidth (pps)
copp-system-lldp	1000	10000
copp-system-mlag	5000	5000
copp-system-selfip	5000	5000
copp-system-selfip-tc6to7	5000	5000
copp-system-sflow	0	25024
copp-system-tc3to5	1000	10000
copp-system-tc6to7	1000	10000
copp-system-urm	1000	10000

Policy maps are modified in policy-map configuration mode. The **policy-map type control-plane** command enters policy-map configuration mode.

Example

- This command enters policy-map configuration mode for editing ***copp-system-policy***.

```
switch(config)#policy-map type control-plane copp-system-policy
switch(config-pmap-copp-system-policy)#
```

The **class (policy-map (control-plane) – Trident-II)** command enters policy-map-class configuration mode, where traffic resolution commands are modified for the configuration mode class.

Example

- This command enters policy-map-class configuration mode for the copp-system-lacp static class.

```
switch(config-pmap-copp-system-policy)#class copp-system-lacp
switch(config-pmap-c-copp-system-policy-copp-system-lacp)#
```

Two traffic resolution commands determine bandwidth parameters for class traffic:

- **bandwidth (policy-map-class (control-plane) – Trident-II)** specifies the minimum bandwidth.
- **shape (policy-map-class (control-plane) – Trident-II)** specifies the maximum bandwidth.

Example

- These commands configure a bandwidth range of 2000 to 4000 packets per seconds (pps) for traffic filtered by the copp-system-lacp class map:

```
switch(config-pmap-c-copp-system-policy-copp-system-lacp)#bandwidth kbps 2000
switch(config-pmap-c-copp-system-policy-copp-system-lacp)#shape kbps 4000
switch(config-pmap-c-copp-system-policy-copp-system-lacp)#
```

Policy-map and policy-map-class configuration modes are group-change modes. Changes are saved with the **exit** command or discarded with the **abort** command. The **show active** command displays the saved version of policy map. The **show pending** command displays the modified policy map.

Example

- These commands exit policy-map-class configuration mode, display the pending policy-map, then exit policy-map configuration mode, which saves the altered policy map to ***running-config***.

```
switch(config-pmap-c-copp-system-policy-copp-system-lacp)#exit
switch(config-pmap-copp-system-policy)#show pending
policy-map type control-plane copp-system-policy
  class copp-system-bpdu

  class copp-system-lldp

  class copp-system-lacp
    shape pps 4000
    bandwidth pps 2000

  class copp-system-arp
             <-------OUTPUT OMITTED FROM EXAMPLE-------->
switch(config-pmap-copp-system-policy)#exit
switch(config)#
```

Applying Policy Maps to the Control Plane

The **copp-system-policy** policy map is always applied to the control plane. No commands are available to add or remove this assignment.

26.7 Traffic Management Configuration Commands

Traffic Policy (Control Plane) Configuration Commands

- class-map type control-plane
- policy-map type control-plane

- bandwidth (policy-map-class (control-plane) – Arad)
- class (policy-map (control-plane) – Arad)
- shape (policy-map-class (control-plane) – Arad)

- bandwidth (policy-map-class (control-plane) – FM6000)
- class (policy-map (control-plane) – FM6000)
- shape (policy-map-class (control-plane) – FM6000)

- bandwidth (policy-map-class (control-plane) – Helix)
- class (policy-map (control-plane) – Helix)
- match (class-map (control-plane) – Helix)
- shape (policy-map-class (control-plane) – Helix)

- bandwidth (policy-map-class (control-plane) – Petra)
- class (policy-map (control-plane) – Petra)
- shape (policy-map-class (control-plane) – Petra)

- bandwidth (policy-map-class (control-plane) – Trident)
- class (policy-map (control-plane) – Trident)
- match (class-map (control-plane) – Trident)
- shape (policy-map-class (control-plane) – Trident)

- bandwidth (policy-map-class (control-plane) – Trident-II)
- class (policy-map (control-plane) – Trident-II)
- match (class-map (control-plane) – Trident-II)
- shape (policy-map-class (control-plane) – Trident-II)

Traffic Policy (PBR) Configuration Commands

- class (policy-map (pbr))
- class-map type pbr
- match (class-map (pbr))
- match (policy-map (pbr))
- platform arad tcam counters feature
- policy-map type pbr
- resequence (class-map (pbr))
- resequence (policy-map (pbr))
- service-policy type pbr (Interface mode)
- set nexthop (policy-map-class – pbr)

Traffic Policy (PDP) Configuration Commands

- policy-map type pdp
- service-policy type pdp (Interface mode)

Traffic Policy (QoS) Configuration Commands

- class-map type qos
- policy-map type qos
- service-policy type qos (Interface mode)

- class (policy-map (qos) – FM6000)
- match (class-map (qos) – FM6000)
- set (policy-map-class (qos) – FM6000)

- class (policy-map (qos) – Helix)
- match (class-map (qos) – Helix)
- set (policy-map-class (qos) – Helix)

- class (policy-map (qos) – Trident)
- match (class-map (qos) – Trident)
- set (policy-map-class (qos) – Trident)

- class (policy-map (qos) – Trident II)
- match (class-map (qos) – Trident II)
- set (policy-map-class (qos) – Trident II)

Traffic Policy Display and Utility Commands

- clear policy-map counters
- show class-map type control-plane
- show class-map type pbr
- show class-map type qos
- show policy-map type control-plane
- show policy-map type pbr
- show policy-map type qos
- show policy-map type qos counters
- show policy-map interface control-plane
- show policy-map interface type qos
- show policy-map interface type qos counters

bandwidth (policy-map-class (control-plane) – Arad)

Note On hardware purchased after July 3, 2017, CoPP is not available.

The **bandwidth** command specifies the minimum bandwidth for traffic filtered by the configuration mode policy map class.

The **no bandwidth** and **default bandwidth** commands remove the minimum bandwidth guarantee for the configuration mode class by deleting the corresponding **bandwidth** command from ***running-config***.

Command Mode

Policy-map-class (control plane) configuration
accessed through **class (policy-map (control-plane) – Arad)**

Command Syntax

```
bandwidth kbps kbits
no bandwidth
default bandwidth
```

Parameters

- *kbits* Minimum data rate (kbits per second). Value ranges from 1 to 10000000.

Related Commands

- **class (policy-map (control-plane) – Arad)** places the switch in policy-map-class (control plane) configuration mode.
- **shape (policy-map-class (control-plane) – Arad)** specifies the maximum bandwidth for traffic defined by the associated class map in its configuration mode policy map class.

Static Classes Default Bandwidth

Arad platform switches define these default bandwidths for control plane static classes:

- copp-system-bgp250•copp-system-l3lpmoverflow250
- copp-system-bpdu1250•copp-system-l3slowpath250
- copp-system-default250•copp-system-l3ttl1250
- copp-system-ipbroadcast250•copp-system-lacp1250
- copp-system-ipmc250•copp-system-linklocal250
- copp-system-ipmcmiss250•copp-system-lldp250
- copp-system-ipunicast250•copp-system-mlag250
- copp-system-l2broadcast250•copp-system-multicastsnoop250
- copp-system-l2unicast250•copp-system-OspfIsis250
- copp-system-l3destmiss250•copp-system-sflow250

Example

- These commands configure the minimum bandwidth of 500 kbps for data traffic specified by the class map copp-system-lldp of the default control-plane policy map.

```
switch(config)#policy-map type control-plane copp-system-policy
switch(config-pmap-copp-system-policy)#class copp-system-lldp
switch(config-pmap-c-copp-system-policy-copp-system-lldp)#bandwidth kbps 500
switch(config-pmap-c-copp-system-policy-copp-system-lldp)#exit
switch(config-pmap-copp-system-policy)#exit
switch(config)#show policy-map interface control-plane copp-system-policy
Service-policy input: copp-system-policy
  Hardware programming status: InProgress
             <-------OUTPUT OMITTED FROM EXAMPLE-------->

  Class-map: copp-system-lldp (match-any)
       shape : 2500 kbps
       bandwidth : 500 kbps
      Out Packets : 0
      Drop Packets : 0

             <-------OUTPUT OMITTED FROM EXAMPLE-------->

switch(config)#
```

bandwidth (policy-map-class (control-plane) – FM6000)

Note On hardware purchased after July 3, 2017, CoPP is not available.

The **bandwidth** command specifies the minimum bandwidth for traffic filtered by the configuration mode policy map class.

The **no bandwidth** and **default bandwidth** commands remove the minimum bandwidth guarantee for the configuration mode class by deleting the corresponding **bandwidth** command from ***running-config***.

Command Mode

Policy-map-class (control plane) configuration
accessed through **class (policy-map (control-plane) – FM6000)**

Command Syntax

```
bandwidth pps packets
no bandwidth
default bandwidth
```

Parameters

- *packets* Minimum data rate (packets per second). Value ranges from 1 to 100000.

Related Commands

- **class (policy-map (control-plane) – FM6000)** places the switch in policy-map-class (control plane) configuration mode.
- **shape (policy-map-class (control-plane) – FM6000)** specifies the maximum bandwidth for traffic defined by the associated class map in its configuration mode policy map class.

Static Classes Default Bandwidth

FM6000 platform switches define these default bandwidths for control plane static classes:

- copp-system-arp1000•copp-system-l3slowpath1000
- copp-system-default1000•copp-system-pim-ptp1000
- copp-system-ipmcrsvd1000•copp-system-ospf-isis1000
- copp-system-ipmcmiss1000•copp-system-selfip5000
- copp-system-igmp1000•copp-system-selfip-tc6to75000
- copp-system-l2rsvd10000•copp-system-sflow1000

Example

- These commands configure the minimum bandwidth of 1000 packets per second for data traffic specified by the class map PMAP-1 in the policy map named copp-system-policy.

```
switch(config)#policy-map type control-plane copp-system-policy
switch(config-pmap-copp-system-policy)#class PMAP-1
switch(config-pmap-c-copp-system-policy-PMAP-1)#bandwidth pps 1000
switch(config-pmap-c-copp-system-policy-PMAP-1)#
```

bandwidth (policy-map-class (control-plane) – Helix)

Note On hardware purchased after July 3, 2017, CoPP is not available.

The **bandwidth** command specifies the minimum bandwidth for traffic filtered by the configuration mode policy map class.

The **no bandwidth** and **default bandwidth** commands remove the minimum bandwidth guarantee for the configuration mode class by deleting the corresponding **bandwidth** command from ***running-config***.

Command Mode

Policy-map-class (control plane) configuration
accessed through **class (policy-map (control-plane) – Helix)**

Command Syntax

```
bandwidth pps packets
no bandwidth
default bandwidth
```

Parameters

- *packets* Minimum data rate (packets per second). Value ranges from 1 to 100000.

Related Commands

- **class (policy-map (control-plane) – Helix)** places the switch in policy-map-class (control plane) configuration mode.
- **shape (policy-map-class (control-plane) – Helix)** specifies the maximum bandwidth for traffic defined by the associated class map in its configuration mode policy map class.

Static Classes Default Bandwidth

Helix platform switches define these default bandwidths for control plane static classes:

- copp-system-acllog1000•copp-system-l3ttl11000
- copp-system-arp1000•copp-system-lacp5000
- copp-system-arpresolver1000•copp-system-lldp1000
- copp-system-bfd5000•copp-system-mlag5000
- copp-system-bgp5000•copp-system-OspfIsis5000
- copp-system-bpdu5000•copp-system-selfip5000
- copp-system-default1000•copp-system-selfip-tc6to75000
- copp-system-glean1000•copp-system-sflow0
- copp-system-igmp1000•copp-system-tc3to51000
- copp-system-ipmcmiss1000•copp-system-tc6to71000
- copp-system-ipmcrsvd1000•copp-system-urm1000
- copp-system-l3destmiss1000•copp-system-vrrp1000
- copp-system-l3slowpath1000

Example

- These commands configure the minimum bandwidth of 500 packets per second for data traffic specified by the class map copp-system-lldp.

```
switch(config)#policy-map type control-plane copp-system-policy
switch(config-pmap-copp-system-policy)#class copp-system-lldp
switch(config-pmap-c-copp-system-policy-copp-system-lldp)#bandwidth pps 500
switch(config-pmap-c-copp-system-policy-copp-system-lldp)#exit
switch(config-pmap-copp-system-policy)#exit
switch(config)#show policy-map interface control-plan copp-system-policy
Service-policy input: copp-system-policy
  Number of units programmed: 4
  Hardware programming status: Successful
             <-------OUTPUT OMITTED FROM EXAMPLE-------->
  Class-map: copp-system-lldp (match-any)
       shape : 10000 pps
       bandwidth : 500 pps
      Out Packets : 304996
      Drop Packets : 0
             <-------OUTPUT OMITTED FROM EXAMPLE-------->
switch(config)#
```

bandwidth (policy-map-class (control-plane) – Petra)

Note On hardware purchased after July 3, 2017, CoPP is not available.

The **bandwidth** command specifies the minimum bandwidth for traffic filtered by the configuration mode policy map class.

The **no bandwidth** and **default bandwidth** commands remove the minimum bandwidth guarantee for the configuration mode class by deleting the corresponding **bandwidth** command from ***running-config***.

Command Mode

Policy-map-class (control plane) configuration
accessed through **class (policy-map (control-plane) – Petra)**

Command Syntax

```
bandwidth kbps kbits
no bandwidth
default bandwidth
```

Parameters

- *kbits* Minimum data rate (kbits per second). Value ranges from 1 to 10000000.

Related Commands

- **class (policy-map (control-plane) – Petra)** places the switch in policy-map-class (control plane) configuration mode.
- **shape (policy-map-class (control-plane) – Petra)** specifies the maximum bandwidth for traffic defined by the associated class map in its configuration mode policy map class.

Static Classes Default Bandwidth

Petra platform switches define these default bandwidths for control plane static classes:

- copp-system-bpdu1250•copp-system-l3destmiss250
- copp-system-default250•copp-system-l3slowpath250
- copp-system-igmp250•copp-system-l3ttl0250
- copp-system-ipbroadcast250•copp-system-l3ttl1250
- copp-system-ipmc250•copp-system-lacp1250
- copp-system-ipmcmiss250•copp-system-lldp250
- copp-system-ipmcrsvd250•copp-system-unicast-arp250
- copp-system-ipunicast250

Guidelines

Petra does not support all discrete rate values. When a specified discrete value is not supported, the switch converts the rate to the next highest discrete value that it supports. The **show** commands displays the converted rate and not the user configured rate.

Example

- These commands configure the minimum bandwidth of 500 kbps for data traffic specified by the class map copp-system-lldp of the default control-plane policy map. Because the switch does not support the discrete value of 500 kbps, it converts the bandwidth up to 651 kbps.

```
switch(config)#policy-map type control-plane copp-system-policy
switch(config-pmap-copp-system-policy)#class copp-system-lldp
switch(config-pmap-c-copp-system-policy-copp-system-lldp)#bandwidth kbps 500
switch(config-pmap-c-copp-system-policy-copp-system-lldp)#exit
switch(config-pmap-copp-system-policy)#exit
switch(config)#show policy-map interface control-plane copp-system-policy
Service-policy input: copp-system-policy
  Hardware programming status: InProgress
             <-------OUTPUT OMITTED FROM EXAMPLE-------->

  Class-map: copp-system-lldp (match-any)
       shape : 2766 kbps
       bandwidth : 651 kbps
      Out Packets : 0
      Drop Packets : 0

             <-------OUTPUT OMITTED FROM EXAMPLE-------->

switch(config)#
```

bandwidth (policy-map-class (control-plane) – Trident)

Note On hardware purchased after July 3, 2017, CoPP is not available.

The **bandwidth** command specifies the minimum bandwidth for traffic filtered by the configuration mode policy map class.

The **no bandwidth** and **default bandwidth** commands remove the minimum bandwidth guarantee for the configuration mode class by deleting the corresponding **bandwidth** command from ***running-config***.

Command Mode

Policy-map-class (control plane) configuration
accessed through **class (policy-map (control-plane) – Trident)**

Command Syntax

```
bandwidth pps packets
no bandwidth
default bandwidth
```

Parameters

- *packets* Minimum data rate (packets per second). Value ranges from 1 to 100000.

Related Commands

- **class (policy-map (control-plane) – Trident)** places the switch in policy-map-class (control plane) configuration mode.
- **shape (policy-map-class (control-plane) – Trident)** specifies the maximum bandwidth for traffic defined by the associated class map in its configuration mode policy map class.

Static Classes Default Bandwidth

Trident platform switches define these default bandwidths for control plane static classes:

- copp-system-arp1000•copp-system-lldp1000
- copp-system-arpresolver1000•copp-system-l3destmiss1000
- copp-system-bpdu5000•copp-system-l3slowpath1000
- copp-system-default1000•copp-system-l3ttl11000
- copp-system-glean1000•copp-system-selfip5000
- copp-system-igmp1000•copp-system-selfip-tc6to75000
- copp-system-ipmcmiss1000•copp-system-sflow0
- copp-system-ipmcrsvd1000•copp-system-tc6to71000
- copp-system-lacp5000•copp-system-tc3to51000

Example

- These commands configure the minimum bandwidth of 1000 packets per second for data traffic specified by the class map PMAP-1 in the policy map named copp-system-policy.

```
switch(config)#policy-map type control-plane copp-system-policy
switch(config-pmap-copp-system-policy)#class PMAP-1
switch(config-pmap-c-copp-system-policy-PMAP-1)#bandwidth pps 1000
switch(config-pmap-c-copp-system-policy-PMAP-1)#
```

bandwidth (policy-map-class (control-plane) – Trident-II)

Note On hardware purchased after July 3, 2017, CoPP is not available.

The **bandwidth** command specifies the minimum bandwidth for traffic filtered by the configuration mode policy map class.

The **no bandwidth** and **default bandwidth** commands remove the minimum bandwidth guarantee for the configuration mode class by deleting the corresponding **bandwidth** command from ***running-config***.

Command Mode

Policy-map-class (control plane) configuration
accessed through **class (policy-map (control-plane) – Trident-II)**

Command Syntax

```
bandwidth pps packets
no bandwidth
default bandwidth
```

Parameters

- *packets* Minimum data rate (packets per second). Value ranges from 1 to 100000.

Related Commands

- **class (policy-map (control-plane) – Trident-II)** places the switch in policy-map-class (control plane) configuration mode.
- **shape (policy-map-class (control-plane) – Trident-II)** specifies the maximum bandwidth for traffic defined by the associated class map in its configuration mode policy map class.

Static Classes Default Bandwidth

Trident-II platform switches define these default bandwidths for control plane static classes:

- copp-system-acllog1000•copp-system-l3slowpath1000
- copp-system-arp1000•copp-system-l3ttl11000
- copp-system-arpresolver1000•copp-system-lacp5000
- copp-system-bfd5000•copp-system-lldp1000
- copp-system-bgp5000•copp-system-mlag5000
- copp-system-bpdu5000•copp-system-selfip5000
- copp-system-default1000•copp-system-selfip-tc6to75000
- copp-system-glean1000•copp-system-sflow0
- copp-system-igmp1000•copp-system-tc3to51000
- copp-system-ipmcmiss1000•copp-system-tc6to71000
- copp-system-ipmcrsvd1000•copp-system-urm1000
- copp-system-l3destmiss1000

Example

- These commands configure the minimum bandwidth of 500 packets per second for data traffic specified by the class map copp-system-lldp.

```
switch(config)#policy-map type control-plane copp-system-policy
switch(config-pmap-copp-system-policy)#class copp-system-lldp
switch(config-pmap-c-copp-system-policy-copp-system-lldp)#bandwidth pps 500
switch(config-pmap-c-copp-system-policy-copp-system-lldp)#exit
switch(config-pmap-copp-system-policy)#exit
switch(config)#show policy-map interface control-plan copp-system-policy
Service-policy input: copp-system-policy
  Number of units programmed: 4
  Hardware programming status: Successful
             <-------OUTPUT OMITTED FROM EXAMPLE-------->
  Class-map: copp-system-lldp (match-any)
       shape : 10000 pps
       bandwidth : 500 pps
      Out Packets : 304996
      Drop Packets : 0
             <-------OUTPUT OMITTED FROM EXAMPLE-------->
switch(config)#
```

class (policy-map (control-plane) – Arad)

Note On hardware purchased after July 3, 2017, CoPP is not available.

The **class** command places the switch in policy-map-class (control plane) configuration mode, which is a group change mode for changing bandwidth and shape parameters associated with a specified class. All changes in a group change mode edit session are pending until the end of the session.

A policy map is an ordered list of classes. The control plane policy map contains 20 static classes. Each class contains an eponymous class map and may contain **bandwidth** and **shape** commands.

- The class map identifies a data stream.
- **Bandwidth** command defines the stream's minimum transmission rate through the control plane.
- **Shape** command defines the stream's maximum transmission rate through the control plane.

Static class maps identify a data stream by definition. Each data packet is managed by commands of the first class whose map matches the packet's content. Dynamic classes are not supported for control plane policing on Arad platform switches.

Each class corresponds to a transmission queue. Queue scheduling is round-robin until ***bandwidth*** rate for a queue is exceeded. Scheduling becomes strict-priority with CPU queue number determining priority until the ***shape*** rate is reached. Packets are dropped after the shape rate is exceeded.

The **exit** command returns the switch to policy-map configuration mode. Saving policy-map-class changes also require an exit from policy-map mode, which saves pending policy-map-class and policy-map changes to ***running-config*** and returns the switch to global configuration mode. The **abort** command discards pending changes, returning the switch to global configuration mode.

The **no class** and **default class** commands remove policy-map-class commands for the specified class assignment from the policy map.

Command Mode

Policy-Map (control plane) configuration
accessed through **policy-map type control-plane** command

Command Syntax

```
class class_name
no class class_name
default class class_name
```

Parameters

- *class_name* name of the class.

Static Classes

Arad platform switches provide the following static control plane classes:

- copp-system-bgp•copp-system-l2broadcast•copp-system-linklocal
- copp-system-bpdu•copp-system-l2unicast•copp-system-lldp
- copp-system-default•copp-system-l3destmiss•copp-system-mlag
- copp-system-ipbroadcast•copp-system-l3lpmoverflow•copp-system-multicastsnoop
- copp-system-ipmc•copp-system-l3slowpath•copp-system-OspfIsis
- copp-system-ipmcmiss•copp-system-l3ttl1•copp-system-sflow
- copp-system-ipunicast•copp-system-lacp

Commands Available in Policy-map-class (control plane) Configuration Mode

- **bandwidth (policy-map-class (control-plane) – Arad)**
- **shape (policy-map-class (control-plane) – Arad)**
- **exit** saves pending class map changes, then returns the switch to global configuration mode.
- **abort** discards pending class map changes, then returns the switch to global configuration mode.

Related Commands

- **policy-map type control-plane** places switch in policy-map (control plane) configuration mode.

Example

- These commands enters policy-map-class configuration mode to modify the shape, bandwidth parameters associated with the static class named ***copp-system-lldp***.

```
switch(config)#policy-map type control-plane copp-system-policy
switch(config-pmap-copp-system-policy)#class copp-system-lldp
switch(config-pmap-c-copp-system-policy-copp-system-lldp)#
```

class (policy-map (control-plane) – FM6000)

Note On hardware purchased after July 3, 2017, CoPP is not available.

The **class** command places the switch in policy-map-class (control plane) configuration mode, which is a group change mode for changing bandwidth and shape parameters associated with a specified class. All changes in a group change mode edit session are pending until the end of the session.

A policy map is an ordered list of classes. The control plane policy map contains 12 static classes. Each class contains an eponymous class map and may contain **bandwidth** and **shape** commands.

- The class map identifies a data stream.
- **Bandwidth** command defines the stream's minimum transmission rate through the control plane.
- **Shape** command defines the stream's maximum transmission rate through the control plane.

Static class maps identify a data stream by definition. Each data packet is managed by commands of the first class whose map matches the packet's content. Dynamic classes are not supported for control plane policing on FM6000 platform switches.

Each class corresponds to a transmission queue. Queue scheduling is round-robin until ***bandwidth*** rate for a queue is exceeded. Scheduling becomes strict-priority with CPU queue number determining priority until the ***shape*** rate is reached. Packets are dropped after the shape rate is exceeded.

The **exit** command returns the switch to policy-map configuration mode. Saving policy-map-class changes also require an exit from policy-map mode, which saves pending policy-map-class and policy-map changes to ***running-config*** and returns the switch to global configuration mode. The **abort** command discards pending changes, returning the switch to global configuration mode.

The **no class** and **default class** commands remove policy-map-class commands for the specified class assignment from the policy map. The class is removed from the policy map if it is a dynamic class.

Command Mode

Policy-Map (control plane) configuration
accessed through **policy-map type control-plane** command

Command Syntax

```
class class_name
no class class_name
default class class_name
```

Parameters

- *class_name* name of the class.

Static Classes

FM6000 platform switches provide the following static control plane classes:

- copp-system-arp•copp-system-igmp•copp-system-PimPtp
- copp-system-default•copp-system-l2rsvd•copp-system-selfip
- copp-system-ipmcmiss•copp-system-l3slowpath•copp-system-selfip-tc6to7
- copp-system-ipmcrsvd•copp-system-OspfIsis•copp-system-sflow

Commands Available in Policy-map-class (control plane) Configuration Mode

- **bandwidth (policy-map-class (control-plane) – FM6000)**
- **shape (policy-map-class (control-plane) – FM6000)**

- **exit** saves pending class map changes, then returns the switch to global configuration mode.
- **abort** discards pending class map changes, then returns the switch to global configuration mode.

Related Commands

- **policy-map type control-plane** places switch in policy-map (control plane) configuration mode.

Example

- These commands enters policy-map-class configuration mode to modify the shape, bandwidth parameters associated with the static class named ***copp-system-arp***.

```
switch(config)#policy-map type control-plane copp-system-policy
switch(config-pmap-copp-system-policy)#class copp-system-arp
switch(config-pmap-c-copp-system-policy-copp-system-arp)#
```

class (policy-map (control-plane) – Helix)

Note On hardware purchased after July 3, 2017, CoPP is not available.

The **class** command places the switch in policy-map-class (control plane) configuration mode, which is a group change mode for changing bandwidth and shape parameters associated with a specified class. All changes in a group change mode edit session are pending until the end of the session.

A policy map is an ordered list of classes. The control plane policy map contains 23 static classes. Each class contains an eponymous class map and may contain **bandwidth** and **shape** commands.

- The class map identifies a data stream.
- **Bandwidth** command defines the stream's minimum transmission rate through the control plane.
- **Shape** command defines the stream's maximum transmission rate through the control plane.

Static class maps identify a data stream by definition. Each data packet is managed by commands of the first class whose map matches the packet's content. Dynamic classes are not supported for control plane policing on Helix platform switches.

Each class corresponds to a transmission queue. Queue scheduling is strict-priority; CPU queue number determines priority until the ***shape*** rate is reached. Packets are dropped after the shape rate is exceeded.

The **exit** command returns the switch to policy-map configuration mode. Saving policy-map-class changes also require an exit from policy-map mode, which saves pending policy-map-class and policy-map changes to ***running-config*** and returns the switch to global configuration mode. The **abort** command discards pending changes, returning the switch to global configuration mode.

The **no class** and **default class** commands remove policy-map-class commands for the specified class assignment from the policy map.

Command Mode

```
Policy-Map (control plane) configuration
accessed through policy-map type control-plane command
```

Command Syntax

```
class class_name
no class class_name
default class class_name
```

Parameters

- *class_name* name of the class.

Static Classes

Helix platform switches provide the following static control plane classes:

- copp-system-acllog•copp-system-ipmcmiss•copp-system-OspfIsis
- copp-system-arp•copp-system-ipmcrsvd•copp-system-selfip
- copp-system-arpresolver•copp-system-l3destmiss•copp-system-selfip-tc6to7
- copp-system-bfd•copp-system-l3slowpath•copp-system-sflow
- copp-system-bgp•copp-system-l3ttl1•copp-system-tc3to5
- copp-system-bpdu•copp-system-lacp•copp-system-tc6to7
- copp-system-default•copp-system-lldp•copp-system-urm
- copp-system-glean•copp-system-lldp•copp-system-vrrp

- copp-system-igmp•copp-system-lldp

Commands Available in Policy-map-class (control plane) Configuration Mode

- **bandwidth (policy-map-class (control-plane) – Helix)**
- **shape (policy-map-class (control-plane) – Helix)**
- **exit** saves pending class map changes, then returns the switch to global configuration mode.
- **abort** discards pending class map changes, then returns the switch to global configuration mode.

Related Commands

- **policy-map type control-plane** places switch in policy-map (control plane) configuration mode.

Example

- These commands enters policy-map-class configuration mode to modify the shape, bandwidth parameters associated with the static class named ***copp-system-arp***.

```
switch(config)#policy-map
switch(config)#policy-map type control-plane copp-system-policy
switch(config-pmap-copp-system-policy)#class copp-system-lldp
switch(config-pmap-c-copp-system-policy-copp-system-lldp)#
```

class (policy-map (control-plane) – Petra)

Note On hardware purchased after July 3, 2017, CoPP is not available.

The **class** command places the switch in policy-map-class (control plane) configuration mode, which is a group change mode for changing bandwidth and shape parameters associated with a specified class. All changes in a group change mode edit session are pending until the end of the session.

A policy map is an ordered list of classes. The control plane policy map contains 15 static classes. Each class contains an eponymous class map and may contain **bandwidth** and **shape** commands.

- The class map identifies a data stream.
- **Bandwidth** command defines the stream's minimum transmission rate through the control plane.
- **Shape** command defines the stream's maximum transmission rate through the control plane.

Static class maps identify a data stream by definition. Each data packet is managed by commands of the first class whose map matches the packet's content. Dynamic classes are not supported for control plane policing on Petra platform switches.

Each class corresponds to a transmission queue. Queue scheduling is round-robin until ***bandwidth*** rate for a queue is exceeded. Scheduling becomes strict-priority with CPU queue number determining priority until the ***shape*** rate is reached. Packets are dropped after the shape rate is exceeded.

The **exit** command returns the switch to policy-map configuration mode. Saving policy-map-class changes also require an exit from policy-map mode, which saves pending policy-map-class and policy-map changes to ***running-config*** and returns the switch to global configuration mode. The **abort** command discards pending changes, returning the switch to global configuration mode.

The **no class** and **default class** commands remove policy-map-class commands for the specified class assignment from the policy map.

Command Mode

Policy-Map (control plane) configuration
accessed through **policy-map type control-plane** command

Command Syntax

```
class class_name
no class class_name
default class class_name
```

Parameters

- *class_name* name of the class.

Static Classes

Petra platform switches provide the following static control plane classes:

- copp-system-bpdu•copp-system-ipmcmiss•copp-system-l3ttl0
- copp-system-default•copp-system-ipmcrsvd•copp-system-l3ttl1
- copp-system-igmp•copp-system-ipunicast•copp-system-lacp
- copp-system-ipbroadcast•copp-system-l3destmiss•copp-system-lldp
- copp-system-ipmc•copp-system-l3slowpath•copp-system-unicast-arp

Commands Available in Policy-map-class (control plane) Configuration Mode

- **bandwidth (policy-map-class (control-plane) – Petra)**

- **shape (policy-map-class (control-plane) – Petra)**
- **exit** saves pending class map changes, then returns the switch to global configuration mode.
- **abort** discards pending class map changes, then returns the switch to global configuration mode.

Related Commands

- **policy-map type control-plane** places switch in policy-map (control plane) configuration mode.

Example

- These commands enters policy-map-class configuration mode to modify the shape, bandwidth parameters associated with the static class named ***copp-system-lldp***.

```
switch(config)#policy-map
switch(config)#policy-map type control-plane copp-system-policy
switch(config-pmap-copp-system-policy)#class copp-system-lldp
switch(config-pmap-c-copp-system-policy-copp-system-lldp)#
```

class (policy-map (control-plane) – Trident)

Note On hardware purchased after July 3, 2017, CoPP is not available.

The **class** command places the switch in policy-map-class (control plane) configuration mode, which is a group change mode for changing bandwidth and shape parameters associated with a specified class. The command adds the specified class to the policy map if it was not previously included. All changes in a group change mode edit session are pending until the end of the session.

A policy map is an ordered list of classes. The control plane policy map contains 18 static classes and up to 30 dynamic classes. Dynamic classes contain an eponymous class map. All classes may contain **bandwidth** and **shape** commands.

- The class map identifies a data stream.
- **Bandwidth** command defines the stream's minimum transmission rate through the control plane.
- **Shape** command defines the stream's maximum transmission rate through the control plane.

Dynamic class maps identify a data stream with an ACL assigned by **match (class-map (control-plane) – Trident)**. Static class maps identify a data stream by definition. Each data packet is managed by commands of the first class whose map matches the packet's content.

Static classes are provided with the switch and cannot be removed from the policy map or modified by the **class** command. Dynamic classes are user defined and added to the policy map by this command. Dynamic classes are always placed in front of the static classes. Bandwidth and shape parameters are editable for all classes.

Each class corresponds to a transmission queue. Queue scheduling is round-robin until ***bandwidth*** rate for a queue is exceeded. Scheduling becomes strict-priority with CPU queue number determining priority until the ***shape*** rate is reached. Packets are dropped after the shape rate is exceeded.

The **exit** command returns the switch to policy-map configuration mode. Saving policy-map-class changes also require an exit from policy-map mode, which saves pending policy-map-class and policy-map changes to ***running-config*** and returns the switch to global configuration mode. The **abort** command discards pending changes, returning the switch to global configuration mode.

The **no class** and **default class** commands remove policy-map-class commands for the specified class assignment from the policy map. The class is removed from the policy map if it is a dynamic class.

Command Mode

Policy-Map (control plane) configuration
accessed through **policy-map type control-plane** command

Command Syntax

```
class class_name [PLACEMENT]
no class class_name [PLACEMENT]
default class class_name [PLACEMENT]
```

Parameters

- *class_name* name of the class.
- ***PLACEMENT*** Specifies the class's map placement. Configurable only for dynamic classes.
  - <no parameter> New classes are placed between the dynamic and static classes. Previously defined classes retain their current policy map placement.
  - **insert-before** *dynamic_class* Class is inserted in front of the specified dynamic class.

Static Classes

Trident switches provide the following static control plane classes:

- copp-system-acllog•copp-system-ipmcmiss•copp-system-lldp
- copp-system-arp•copp-system-ipmcrsvd•copp-system-selfip
- copp-system-arpresolver•copp-system-l3destmiss•copp-system-selfip-tc6to7
- copp-system-bpdu•copp-system-l3slowpath•copp-system-sflow
- copp-system-glean•copp-system-l3ttl1•copp-system-tc3to5
- copp-system-igmp•copp-system-lacp•copp-system-tc6to7

Commands Available in Policy-map-class (control plane) Configuration Mode

- **bandwidth (policy-map-class (control-plane) – Trident)**
- **shape (policy-map-class (control-plane) – Trident)**
- **exit** saves pending class map changes, then returns the switch to global configuration mode.
- **abort** discards pending class map changes, then returns the switch to global configuration mode.

Related Commands

- **class-map type control-plane** places switch in class-map (control-plane) configuration mode.
- **policy-map type control-plane** places switch in policy-map (control plane) configuration mode.

Example

- These commands add CM-1 class to the copp-system-policy policy map.

```
switch(config)#policy-map type control-plane copp-system-policy
switch(config-pmap-copp-system-policy)#class CM-1
switch(config-pmap-c-copp-system-policy-CM-1)#
```

class (policy-map (control-plane) – Trident-II)

Note On hardware purchased after July 3, 2017, CoPP is not available.

The **class** command places the switch in policy-map-class (control plane) configuration mode, which is a group change mode for changing bandwidth and shape parameters associated with a specified class. All changes in a group change mode edit session are pending until the end of the session.

A policy map is an ordered list of classes. The control plane policy map contains 23 static classes. Each class contains an eponymous class map and may contain **bandwidth** and **shape** commands.

- The class map identifies a data stream.
- **Bandwidth** command defines the stream's minimum transmission rate through the control plane.
- **Shape** command defines the stream's maximum transmission rate through the control plane.

Static class maps identify a data stream by definition. Each data packet is managed by commands of the first class whose map matches the packet's content. Dynamic classes are not supported for control plane policing on Trident-II platform switches.

Each class corresponds to a transmission queue. Queue scheduling is strict-priority; CPU queue number determines priority until the ***shape*** rate is reached. Packets are dropped after the shape rate is exceeded.

The **exit** command returns the switch to policy-map configuration mode. Saving policy-map-class changes also require an exit from policy-map mode, which saves pending policy-map-class and policy-map changes to ***running-config*** and returns the switch to global configuration mode. The **abort** command discards pending changes, returning the switch to global configuration mode.

The **no class** and **default class** commands remove policy-map-class commands for the specified class assignment from the policy map.

Command Mode

Policy-Map (control plane) configuration
accessed through **policy-map type control-plane** command

Command Syntax

```
class class_name
no class class_name
default class class_name
```

Parameters

- *class_name* name of the class.

Static Classes

Trident-II platform switches provide the following static control plane classes:

- copp-system-acllog•copp-system-igmp•copp-system-mlag
- copp-system-arp•copp-system-ipmcmiss•copp-system-selfip
- copp-system-arpresolver•copp-system-ipmcrsvd•copp-system-selfip-tc6to7
- copp-system-bfd•copp-system-l3destmiss•copp-system-sflow
- copp-system-bgp•copp-system-l3slowpath•copp-system-tc3to5
- copp-system-bpdu•copp-system-l3ttl1•copp-system-tc6to7
- copp-system-default•copp-system-lacp•copp-system-urm

- copp-system-glean•copp-system-lldp

Commands Available in Policy-map-class (control plane) Configuration Mode

- **bandwidth (policy-map-class (control-plane) – Trident-II)**
- **shape (policy-map-class (control-plane) – Trident-II)**
- **exit** saves pending class map changes, then returns the switch to global configuration mode.
- **abort** discards pending class map changes, then returns the switch to global configuration mode.

Related Commands

- **policy-map type control-plane** places switch in policy-map (control plane) configuration mode.

Example

- These commands enters policy-map-class configuration mode to modify the shape, bandwidth parameters associated with the static class named ***copp-system-arp***.

```
switch(config)#policy-map
switch(config)#policy-map type control-plane copp-system-policy
switch(config-pmap-copp-system-policy)#class copp-system-lldp
switch(config-pmap-c-copp-system-policy-copp-system-lldp)#
```

class (policy-map (pbr))

The **class (policy-map (pbr))** command places the switch in policy-map-class (pbr) configuration mode, which is a group change mode that modifies the specified class of the configuration mode Policy-Based Routing (PBR) policy map. The command adds the class to the policy map if it was not previously included in the policy map. All changes in a group change mode edit session are pending until the mode is exited, and can be canceled by using the **abort** command.

A PBR policy map is an ordered list of classes. Each class contains an eponymous class map and can contain set commands to specify next hop. Classes without set commands translate to no action being performed on that class of packets.

- The class map identifies a data stream through ACLs. Class maps are configured in class-map (pbr) configuration mode.
- **Set** commands can be used to specify the next hop for a given class. **Set** commands are configured in policy-map-class (pbr) configuration mode.

PBR policy maps can also contain one or more raw match statements which filter incoming traffic without using ACLs. Data packets are managed by commands of the first class or raw match statement matching the packet's contents.

The **exit** command returns the switch to policy-map (pbr) configuration mode. However, saving policy-map-class changes also requires an exit from policy-map (pbr) configuration mode. This saves all pending policy map and policy-map-class changes to ***running-config*** and returns the switch to global configuration mode. The **abort** command discards pending changes, returning the switch to global configuration mode.

The **no class** and **default class** commands remove the class assignment from the configuration mode policy map by deleting the corresponding **class** configuration from ***running-config***.

Command Mode

Policy-Map (pbr) Configuration
accessed through **policy-map type pbr**

Command Syntax

```
[sequence_number] class class_name
no [sequence_number] class class_name
default [sequence_number] class class_name
no sequence_number
default sequence_number
```

Parameters

- *sequence_number* Sequence number (1 to 4294967295) assigned to the rule. If no number is entered, the number is derived by adding 10 to the number of the policy map's last numbered line. To increase the distance between existing entries, use the **resequence** command.
- *class_name* name of the class.

Commands Available in Policy-map-class (pbr) Configuration Mode

- **set nexthop (policy-map-class – pbr)** sets next hop for the class.
- **exit** saves pending class changes and returns switch to policy-map (pbr) configuration mode.
- **abort** discards pending class changes and returns switch to policy-map (pbr) configuration mode.

Related Commands

- **class-map type pbr** places switch in class-map (pbr) configuration mode.
- **policy-map type pbr** places switch in policy-map (pbr) configuration mode

Examples

- These commands add the ***CMAP1*** class map to the ***PMAP1*** policy map, then place the switch in policy-map-class configuration mode where the next hops can be assigned to the class. Changes will not take effect until both modes are exited.

```
switch(config)#policy-map type pbr PMAP1
switch(config-pmap-PMAP1)#class CMAP1
switch(config-pmap-c-PMAP1-CMAP1)#
```

class (policy-map (qos) – FM6000)

The **class** command places the switch in policy-map-class (qos) configuration mode, which is a group change mode that modifies the specified class of the configuration mode policy map. The command adds the class to the policy map if it was not previously included in the policy map. All changes in a group change mode edit session are pending until the end of the session.

A policy map is an ordered list of classes. Each class contains an eponymous class map and at least one set command:

- The class map identifies a data stream through an ACL. Class maps are configured in class-map (qos) configuration mode.
- **Set** commands either modify a packet's content (CoS or DSCP fields) or assigns it to a traffic class queue. **Set** commands are configured in policy-map-class (qos) configuration mode.

  Data packets are managed by commands of the first class whose map matches the packet's content.

The **exit** command returns the switch to policy-map configuration mode. However, saving policy-map-class changes also require an exit from policy-map mode. This saves all pending policy map and policy-map-class changes to ***running-config*** and returns the switch to global configuration mode. The **abort** command discards pending changes, returning the switch to global configuration mode.

The **no class** and **default class** commands remove the class assignment from the configuration mode policy map by deleting the corresponding **class** configuration from ***running-config***.

Command Mode

Policy-Map (qos) Configuration
accessed through **policy-map type qos**

Command Syntax

```
class class_name [PLACEMENT]
no class class_name [PLACEMENT]
default class class_name [PLACEMENT]
```

Parameters

- *class_name* name of the class.
- ***PLACEMENT*** Specifies the map placement within the list of class maps.
  - <no parameter> Class is placed at the top of the list.
  - **insert-before** *existing_class* Class is inserted in front of the specified class.

Commands Available in Policy-map-class (qos) Configuration Mode

- **set (policy-map-class (qos) – FM6000)**
- **exit** saves pending class changes and returns switch to policy-map (qos) configuration mode.
- **abort** discards pending class changes and returns switch to policy-map (qos) configuration mode.

Related Commands

- **class-map type qos** places switch in class-map (qos) configuration mode.
- **policy-map type qos** places switch in policy-map (qos) configuration mode

Example

- These commands add the ***CMAP_1*** class map to the ***PMAP_1*** policy map, then places the switch in policy-map-class configuration mode.

```
switch(config)#policy-map type qos PMAP-1
switch(config-pmap-PMAP-1)#class CMAP-1
switch(config-pmap-c-PMAP-1-CMAP-1)#
```

class (policy-map (qos) – Helix)

The **class** command places the switch in policy-map-class (qos) configuration mode, which is a group change mode that modifies the specified class of the configuration mode policy map. The command adds the class to the policy map if it was not previously included in the policy map. All changes in a group change mode edit session are pending until the end of the session.

A policy map is an ordered list of classes. Each class contains an eponymous class map and at least one set command:

- The class map identifies a data stream through an ACL. Class maps are configured in class-map (qos) configuration mode.
- **Set** commands either modify a packet's content (CoS or DSCP fields) or assigns it to a traffic class queue. **Set** commands are configured in policy-map-class (qos) configuration mode.

  Data packets are managed by commands of the first class whose map matches the packet's content.

The **exit** command returns the switch to policy-map configuration mode. However, saving policy-map-class changes also require an exit from policy-map mode. This saves all pending policy map and policy-map-class changes to ***running-config*** and returns the switch to global configuration mode. The **abort** command discards pending changes, returning the switch to global configuration mode.

The **no class** and **default class** commands remove the class assignment from the configuration mode policy map by deleting the corresponding **class** configuration from ***running-config***.

Command Mode

Policy-Map (qos) Configuration
accessed through **policy-map type qos** command

Command Syntax

```
class class_name [PLACEMENT]
no class class_name [PLACEMENT]
default class class_name [PLACEMENT]
```

Parameters

- *class_name* name of the class.
- ***PLACEMENT*** Specifies the map placement within the list of class maps.
  - <no parameter> Class is placed at the top of the list.
  - **insert-before** *existing_class* Class is inserted in front of the specified class.

Commands Available in Policy-map-class (qos) Configuration Mode

- **set (policy-map-class (qos) – Helix)**
- **exit** saves pending class changes and returns switch to policy-map (qos) configuration mode.
- **abort** discards pending class changes and returns switch to policy-map (qos) configuration mode.

Related Commands

- **class-map type qos** places switch in class-map (qos) configuration mode.
- **policy-map type qos** places switch in policy-map (qos) configuration mode

Example

- These commands add the ***CMAP_1*** class map to the ***PMAP_1*** policy map, then places the switch in policy-map-class configuration mode.

```
switch(config)#policy-map type qos PMAP-1
switch(config-pmap-PMAP-1)#class CMAP-1
switch(config-pmap-c-PMAP-1-CMAP-1)#
```

class (policy-map (qos) – Trident)

The **class** command places the switch in policy-map-class (qos) configuration mode, which is a group change mode that modifies the specified class of the configuration mode policy map. The command adds the class to the policy map if it was not previously included in the policy map. All changes in a group change mode edit session are pending until the end of the session.

A policy map is an ordered list of classes. Each class contains an eponymous class map and at least one set command:

- The class map identifies a data stream through an ACL. Class maps are configured in class-map (qos) configuration mode.
- **Set** commands either modify a packet's content (CoS or DSCP fields) or assigns it to a traffic class queue. **Set** commands are configured in policy-map-class (qos) configuration mode.

  Data packets are managed by commands of the first class whose map matches the packet's content.

The **exit** command returns the switch to policy-map configuration mode. However, saving policy-map-class changes also require an exit from policy-map mode. This saves all pending policy map and policy-map-class changes to ***running-config*** and returns the switch to global configuration mode. The **abort** command discards pending changes, returning the switch to global configuration mode.

The **no class** and **default class** commands remove the class assignment from the configuration mode policy map by deleting the corresponding **class** configuration from ***running-config***.

Command Mode

Policy-Map (qos) Configuration
accessed through **policy-map type qos** command

Command Syntax

```
class class_name [PLACEMENT]
no class class_name [PLACEMENT]
default class class_name [PLACEMENT]
```

Parameters

- *class_name* name of the class.
- ***PLACEMENT*** Specifies the map placement within the list of class maps.
  - <no parameter> Class is placed at the top of the list.
  - **insert-before** *existing_class* Class is inserted in front of the specified class.

Commands Available in Policy-map-class (qos) Configuration Mode

- **set (policy-map-class (qos) – Trident)**
- **exit** saves pending class changes and returns switch to policy-map (qos) configuration mode.
- **abort** discards pending class changes and returns switch to policy-map (qos) configuration mode.

Related Commands

- **class-map type qos** places switch in class-map (qos) configuration mode.
- **policy-map type qos** places switch in policy-map (qos) configuration mode

Example

- These commands add the ***CMAP_1*** class map to the ***PMAP_1*** policy map, then places the switch in policy-map-class configuration mode.

```
switch(config)#policy-map type qos PMAP-1
switch(config-pmap-PMAP-1)#class CMAP-1
switch(config-pmap-c-PMAP-1-CMAP-1)#
```

class (policy-map (qos) – Trident II)

The **class** command places the switch in policy-map-class (qos) configuration mode, which is a group change mode that modifies the specified class of the configuration mode policy map. The command adds the class to the policy map if it was not previously included in the policy map. All changes in a group change mode edit session are pending until the end of the session.

A policy map is an ordered list of classes. Each class contains an eponymous class map and at least one set command:

- The class map identifies a data stream through an ACL. Class maps are configured in class-map (qos) configuration mode.
- **Set** commands either modify a packet's content (CoS or DSCP fields) or assigns it to a traffic class queue. **Set** commands are configured in policy-map-class (qos) configuration mode.

  Data packets are managed by commands of the first class whose map matches the packet's content.

The **exit** command returns the switch to policy-map configuration mode. However, saving policy-map-class changes also require an exit from policy-map mode. This saves all pending policy map and policy-map-class changes to ***running-config*** and returns the switch to global configuration mode. The **abort** command discards pending changes, returning the switch to global configuration mode.

The **no class** and **default class** commands remove the class assignment from the configuration mode policy map by deleting the corresponding **class** configuration from ***running-config***.

Command Mode

Policy-Map (qos) Configuration
accessed through **policy-map type qos** command

Command Syntax

```
class class_name [PLACEMENT]
no class class_name [PLACEMENT]
default class class_name [PLACEMENT]
```

Parameters

- *class_name* name of the class.
- ***PLACEMENT*** Specifies the map placement within the list of class maps.
  - <no parameter> Class is placed at the top of the list.
  - **insert-before** *existing_class* Class is inserted in front of the specified class.

Commands Available in Policy-map-class (qos) Configuration Mode

- **set (policy-map-class (qos) – Trident II)**
- **exit** saves pending class changes and returns switch to policy-map (qos) configuration mode.
- **abort** discards pending class changes and returns switch to policy-map (qos) configuration mode.

Related Commands

- **class-map type qos** places switch in class-map (qos) configuration mode.
- **policy-map type qos** places switch in policy-map (qos) configuration mode

Example

- These commands add the ***CMAP_1*** class map to the ***PMAP_1*** policy map, then places the switch in policy-map-class configuration mode.

```
switch(config)#policy-map type qos PMAP-1
switch(config-pmap-PMAP-1)#class CMAP-1
switch(config-pmap-c-PMAP-1-CMAP-1)#
```

class-map type control-plane

The **class-map type control-plane** command places the switch in Class-Map (control plane) configuration mode, which is a group change mode that modifies a control-plane dynamic class map. A dynamic class map is a data structure that uses access control lists (ACLs) to define a data stream by specifying characteristics of data packets that comprise that stream. Control-plane policy maps use class maps to specify which control plane traffic is controlled by policy map criteria.

The **exit** command saves pending class map changes to ***running-config*** and returns the switch to global configuration mode. Class map changes are also saved by entering a different configuration mode. The **abort** command discards pending changes and returns the switch to global configuration mode.

The **no class-map type control-plane** and **default class-map type control-plane** commands delete the specified class map by removing the corresponding **class-map type control-plane** command and its associated configuration.

Command Mode

Global Configuration

Command Syntax

```
class-map type control-plane match-any class_name
no class-map type control-plane [match-any] class_name
default class-map type control-plane [match-any] class_name
```

Parameters

- *class_name* Name of class map.

Commands Available in Class-Map (Control Plane) Configuration Mode

- **match (class-map (control-plane) – Trident)**

Related Commands

- **policy-map type control-plane**
- **class (policy-map (control-plane) – Trident)**
- **class-map type qos**

Example

- This command creates the control plane class map named CP-MAP-1 and places the switch in class-map configuration mode.

```
switch(config)#class-map type control-plane match-any CP-CMAP-1
switch(config-cmap-CP-CMAP-1)#
```

class-map type pbr

The **class-map type pbr** command places the switch in class-map (pbr) configuration mode for the specified class map, and creates the class map if one does not already exist. Class-map (pbr) configuration mode is a group change mode that modifies a class map for policy-based routing (PBR). PBR class maps contain one or more **match** statements which filter incoming traffic using ACLs. PBRs can then use these class maps to set next-hop IP addresses for the traffic that matches them. (Classes without set commands translate to no action being performed on that class of packets.)

The **exit** command saves pending class map changes to ***running-config***, then returns the switch to global configuration mode. Class map changes are also saved by directly entering a different configuration mode. The **abort** command discards pending changes and returns the switch to global configuration mode.

The **no class-map type pbr** and **default class-map type pbr** commands delete the specified class map by removing the corresponding **class-map type pbr** command and its associated configuration.

Command Mode

Global Configuration

Command Syntax

```
class-map type pbr match-any map_name
no class-map type pbr match-any map_name
default class-map type pbr match-any map_name
```

Parameters

- *map_name* Name of class map.

Commands Available in Class-Map (PBR) configuration mode

- **match (class-map (pbr))**
- **resequence (class-map (pbr))**

Related Commands

- **policy-map type pbr**
- **class (policy-map (pbr))**

Example

- This command creates the PBR class map named MAP1 and places the switch in class-map (pbr) configuration mode where match criteria can be configured for the class.

```
switch(config)#class-map type pbrmatch-any MAP1
switch(config-cmap-MAP1)#
```

class-map type qos

The **class-map type qos** command places the switch in Class-Map (qos) configuration mode, which is a group change mode that modifies a QoS dynamic class map. A dynamic class map is a data structure that uses access control lists (ACLs) to define a data stream by specifying characteristics of data packets that comprise that stream. QoS policy maps use class maps to specify the traffic (to which the policy map is assigned) that is transformed by policy map criteria.

The **exit** command saves pending class map changes to ***running-config***, then returns the switch to global configuration mode. Class map changes are also saved by entering a different configuration mode. The **abort** command discards pending changes and returns the switch to global configuration mode.

The **no class-map type qos** and **default class-map type qos** commands delete the specified class map by removing the corresponding **class-map type qos** command and its associated configuration. The **class-map** and **class-map type qos** commands are equivalent.

Command Mode

Global Configuration

Command Syntax

```
class-map [type qos] match-any class_name
no class-map [type qos] [match-any] class_name
default class-map [type qos] [match-any] class_name
```

class-map *map_name* and **class-map type qos** *map_name* are identical commands.

Parameters

- *class_name* Name of class map.

Commands Available in Class-Map (QoS) Configuration Mode

- **match (class-map (qos) – FM6000)**
- **match (class-map (qos) – Trident)**

Related Commands

- **policy-map type qos**
- **class (policy-map (qos) – FM6000)**
- **class (policy-map (qos) – Trident)**

Example

- This command creates the QoS class map named MAP-1 and places the switch in class-map configuration mode.

```
switch(config)#class-map type qos match-any MAP-1
switch(config-cmap-MAP-1)#
```

clear policy-map counters

The **clear policy-map** command resets the specified policy map counters to zero. Policy map counters record the quantity of packets that are filtered by the ACLs that comprise a specified policy map.

Command Mode

Privileged EXEC

Command Syntax

```
clear policy-map INTERFACE_NAME counters MAP_NAME
```

Parameters

- ***INTERFACE_NAME*** Interface for which command clears table counters. Options include:
  - **interface control-plane** Control plane.
- ***MAP_NAME*** Policy map for which command clears counters. Options include:
  - **copp-system-policy** Name of only policy map supported for the control plane.

match (class-map (control-plane) – Helix)

Note On hardware purchased after July 3, 2017, CoPP is not available.

The **match** command assigns an ACL to the configuration mode class map. A class map can contain only one ACL. Class maps only use permit rules to filter data; deny rules are ignored. The command accepts IPv4 and IPv4 standard ACLs.

A class map is assigned to a policy map by the **class (policy-map (control-plane) – Helix)** command.

Class map (control plane) configuration mode is a group change mode. **Match** statements are not saved to ***running-config*** until the edit session is completed by exiting the mode.

The **no match** and **default match** commands remove the **match** statement from the configuration mode class map by deleting the corresponding command from ***running-config***.

Command Mode

Class-Map (control plane) configuration
accessed through **class-map type control-plane** command

Command Syntax

```
match ip access-group list_name
no match ip access-group list_name
default match ip access-group list_name
```

Parameters

- *list_name* name of ACL assigned to class map.

Related Commands

- **class-map type control-plane** places the switch in Class-Map configuration mode.
- **exit** saves pending class map changes, then returns the switch to global configuration mode.
- **abort** discards pending class map changes, then returns the switch to global configuration mode.
- **class (policy-map (control-plane) – Helix)** assigns a class map to a policy map.

Guidelines

Static class maps cannot be modified by this command.

Match statements are saved to ***running-config*** only upon exiting Class-Map (control plane) configuration mode.

Example

- These commands add the IP ACL ***list_1*** to the ***map_1*** class map, then saves the command by exiting class-map mode.

```
switch(config)#class-map type control-plane map_1
switch(config-cmap-map_1)#match ip access-group list_1
switch(config-cmap-map_1)#exit
switch(config)#
```

match (class-map (control-plane) – Trident)

Note On hardware purchased after July 3, 2017, CoPP is not available.

The **match** command assigns an ACL to the configuration mode class map. A class map can contain only one ACL. Class maps only use permit rules to filter data; deny rules are ignored. The command accepts IPv4, IPv6, IPv4 standard, and IPv6 standard ACLs.

A class map is assigned to a policy map by the **class (policy-map (control-plane) – Trident)** command.

Class map (control plane) configuration mode is a group change mode. **Match** statements are not saved to ***running-config*** until the edit session is completed by exiting the mode.

The **no match** and **default match** commands remove the **match** statement from the configuration mode class map by deleting the corresponding command from ***running-config***.

Command Mode

Class-Map (control plane) configuration
accessed through **class-map type control-plane** command

Command Syntax

```
match IP_VERSION access-group list_name
no match IP_VERSION access-group list_name
default match IP_VERSION access-group list_name
```

Parameters

- ***IP_VERSION*** IP version of the specified ACL. Options include:
  - **ip** IPv4.
  - **ipv6** IPv6.
- *list_name* name of ACL assigned to class map.

Related Commands

- **class-map type control-plane** places the switch in Class-Map configuration mode.
- **exit** saves pending class map changes, then returns the switch to global configuration mode.
- **abort** discards pending class map changes, then returns the switch to global configuration mode.
- **class (policy-map (control-plane) – Trident)** assigns a class map to a policy map.

Guidelines

Static class maps cannot be modified by this command.

Match statements are saved to ***running-config*** only upon exiting Class-Map (control plane) configuration mode.

Example

- These commands add the IPv4 ACL names ***list_1*** to the ***map_1*** class map, then saves the command by exiting class-map mode.

```
switch(config)#class-map type control-plane map_1
switch(config-cmap-map_1)#match ip access-group list_1
switch(config-cmap-map_1)#exit
switch(config)#
```

match (class-map (control-plane) – Trident-II)

Note On hardware purchased after July 3, 2017, CoPP is not available.

The **match** command assigns an ACL to the configuration mode class map. A class map can contain only one ACL. Class maps only use permit rules to filter data; deny rules are ignored. The command accepts IPv4 and IPv4 standard ACLs.

A class map is assigned to a policy map by the **class (policy-map (control-plane) – Trident-II)** command.

Class map (control plane) configuration mode is a group change mode. **Match** statements are not saved to ***running-config*** until the edit session is completed by exiting the mode.

The **no match** and **default match** commands remove the **match** statement from the configuration mode class map by deleting the corresponding command from ***running-config***.

Command Mode

Class-Map (control plane) configuration
accessed through **class-map type control-plane** command

Command Syntax

```
match ip access-group list_name
no match ip access-group list_name
default match ip access-group list_name
```

Parameters

- *list_name* name of ACL assigned to class map.

Related Commands

- **class-map type control-plane** places the switch in Class-Map configuration mode.
- **exit** saves pending class map changes, then returns the switch to global configuration mode.
- **abort** discards pending class map changes, then returns the switch to global configuration mode.
- **class (policy-map (control-plane) – Trident-II)** assigns a class map to a policy map.

Guidelines

Static class maps cannot be modified by this command.

Match statements are saved to ***running-config*** only upon exiting Class-Map (control plane) configuration mode.

Example

- These commands add the IP ACL ***list_1*** to the ***map_1*** class map, then saves the command by exiting class-map mode.

```
switch(config)#class-map type control-plane map_1
switch(config-cmap-map_1)#match ip access-group list_1
switch(config-cmap-map_1)#exit
switch(config)#
```

match (class-map (pbr))

The **match** command assigns ACLs to the configuration mode Policy-Based Routing (PBR) class map. The command accepts IPv4, IPv4 standard, IPv6 and IPv6 standard ACLs.

Class map (pbr) configuration mode is a group change mode. **Match** statements are not saved to ***running-config*** until the edit session is completed by exiting the mode.

The **no match** and **default match** commands remove the **match** statement from the configuration mode class map by deleting the corresponding command from ***running-config***.

Important! PBR ACLs use only permit rules to filter data; if there are deny rules in an ACL used by PBR, the configuration will be reverted.

Command Mode

Class-map (pbr) configuration
accessed through **class-map type pbr** command

Command Syntax

```
[sequence_number] match {ip|ipv6} access-group list_name
no [sequence_number] match {ip|ipv6} access-group list_name
default [sequence_number] match {ip|ipv6} access-group list_name
no sequence_number
default sequence_number
```

Parameters

- *sequence_number* Sequence number (1 to 4294967295) assigned to the rule. If no number is entered, the number is derived by adding 10 to the number of the class map's last numbered line. To increase the distance between existing entries, use the **resequence** command.
- *list_name* name of ACL assigned to class map.

Related Commands

- **class-map type pbr** places the switch in class-map configuration mode.
- **exit** saves pending class map changes, then returns the switch to global configuration mode.
- **abort** discards pending class map changes, then returns the switch to global configuration mode.
- **class (policy-map (pbr))** assigns a class map to a policy map.

Example

- These commands add the IPv4 ACL named ***list1*** to the ***map1*** class map, then save the change by exiting class-map mode.

```
switch(config)#class-map type pbr map1
switch(config-cmap-map1)#match ip access-group list1
switch(config-cmap-map1)#exit
switch(config)#
```

match (class-map (qos) – FM6000)

The **match** command assigns an ACL to the configuration mode class map. A class map can contain only one ACL. Class maps only use permit rules to filter data; deny rules are ignored. The command accepts IPv4 and IPv4 standard ACLs.

Class map (qos) configuration mode is a group change mode. **Match** statements are not saved to ***running-config*** until the edit session is completed by exiting the mode.

The **no match** and **default match** commands remove the **match** statement from the configuration mode class map by deleting the corresponding command from ***running-config***.

Command Mode

Class-map (qos) configuration
accessed through **class-map type qos** command

Command Syntax

```
match IP_VERSION access-group list_name
no match IP_VERSION access-group list_name
default match IP_VERSION access-group list_name
```

Parameters

- ***IP_VERSION*** IP version of the specified ACL. Options include:
  - **ip** IPv4.
- *list_name* name of ACL assigned to class map.

Related Commands

- **class-map type qos** places the switch in Class-Map configuration mode.
- **exit** saves pending class map changes, then returns the switch to global configuration mode.
- **abort** discards pending class map changes, then returns the switch to global configuration mode.
- **class (policy-map (qos) – FM6000)** assigns a class map to a policy map.

Example

- These commands add the IPv4 ACL named ***list_1*** to the ***map_1*** class map, then saves the command by exiting class-map mode.

```
switch(config)#class-map type qos map_1
switch(config-cmap-map_1)#match ip access-group list_1
switch(config-cmap-map_1)#exit
switch(config)#
```

match (class-map (qos) – Helix)

The **match** command assigns an ACL to the configuration mode class map. A class map can contain only one ACL. Class maps only use permit rules to filter data; deny rules are ignored. The command accepts IPv4, IPv4 standard, IPv6, and IPv6 standard ACLs.

Class map (qos) configuration mode is a group change mode. **Match** statements are not saved to ***running-config*** until the edit session is completed by exiting the mode.

The **no match** and **default match** commands remove the **match** statement from the configuration mode class map by deleting the corresponding command from ***running-config***.

Command Mode

Class-Map (qos) configuration
accessed through **class-map type qos** command

Command Syntax

```
match IP_VERSION access-group list_name
no match IP_VERSION access-group list_name
default match IP_VERSION access-group list_name
```

Parameters

- ***IP_VERSION*** IP version of the specified ACL. Options include:
  - **ip** IPv4.
  - **ipv6** IPv6.
- *list_name* name of ACL assigned to class map.

Related Commands

- **class-map type qos** places the switch in Class-Map configuration mode.
- **exit** saves pending class map changes, then returns the switch to global configuration mode.
- **abort** discards pending class map changes, then returns the switch to global configuration mode.
- **class (policy-map (qos) – Helix)** assigns a class map to a policy map.

Example

- These commands add the IPv4 ACL named ***list_1*** to the ***map_1*** class map, then saves the command by exiting class-map mode.

```
switch(config)#class-map type qos map_1
switch(config-cmap-map_1)#match ip access-group list_1
switch(config-cmap-map_1)#exit
switch(config)#
```

match (class-map (qos) – Trident)

The **match** command assigns an ACL to the configuration mode class map. A class map can contain only one ACL. Class maps only use permit rules to filter data; deny rules are ignored. The command accepts IPv4, IPv4 standard, IPv6, and IPv6 standard ACLs.

Class map (qos) configuration mode is a group change mode. **Match** statements are not saved to ***running-config*** until the edit session is completed by exiting the mode.

The **no match** and **default match** commands remove the **match** statement from the configuration mode class map by deleting the corresponding command from ***running-config***.

Command Mode

Class-Map (qos) configuration
accessed through **class-map type qos** command

Command Syntax

```
match IP_VERSION access-group list_name
no match IP_VERSION access-group list_name
default match IP_VERSION access-group list_name
```

Parameters

- ***IP_VERSION*** IP version of the specified ACL. Options include:
  - **ip** IPv4.
  - **ipv6** IPv6.
- *list_name* name of ACL assigned to class map.

Related Commands

- **class-map type qos** places the switch in Class-Map configuration mode.
- **exit** saves pending class map changes, then returns the switch to global configuration mode.
- **abort** discards pending class map changes, then returns the switch to global configuration mode.
- **class (policy-map (qos) – Trident)** assigns a class map to a policy map.

Example

- These commands add the IPv4 ACL named ***list_1*** to the ***map_1*** class map, then saves the command by exiting class-map mode.

```
switch(config)#class-map type qos map_1
switch(config-cmap-map_1)#match ip access-group list_1
switch(config-cmap-map_1)#exit
switch(config)#
```

match (class-map (qos) – Trident II)

The **match** command assigns an ACL to the configuration mode class map. A class map can contain only one ACL. Class maps only use permit rules to filter data; deny rules are ignored. The command accepts IPv4, IPv4 standard, IPv6, and IPv6 standard ACLs.

Class map (qos) configuration mode is a group change mode. **Match** statements are not saved to ***running-config*** until the edit session is completed by exiting the mode.

The **no match** and **default match** commands remove the **match** statement from the configuration mode class map by deleting the corresponding command from ***running-config***.

Command Mode

Class-Map (qos) configuration
accessed through **class-map type qos** command

Command Syntax

```
match IP_VERSION access-group list_name
no match IP_VERSION access-group list_name
default match IP_VERSION access-group list_name
```

Parameters

- ***IP_VERSION*** IP version of the specified ACL. Options include:
  - **ip** IPv4.
  - **ipv6** IPv6.
- *list_name* name of ACL assigned to class map.

Related Commands

- **class-map type qos** places the switch in Class-Map configuration mode.
- **exit** saves pending class map changes, then returns the switch to global configuration mode.
- **abort** discards pending class map changes, then returns the switch to global configuration mode.
- **class (policy-map (qos) – Trident)** assigns a class map to a policy map.

Example

- These commands add the IPv4 ACL named ***list_1*** to the ***map_1*** class map, then saves the command by exiting class-map mode.

```
switch(config)#class-map type qos map_1
switch(config-cmap-map_1)#match ip access-group list_1
switch(config-cmap-map_1)#exit
switch(config)#
```

match (policy-map (pbr))

The **match** command creates a policy map clause entry that specifies one filtering condition. When a packet matches the filtering criteria, its next hop is set as specified. When a packet's properties do not equal the statement parameters, the packet is evaluated against the next clause or class map in the policy map, as determined by sequence number. If all clauses fail to set a next hop for the packet, the packet is routed according to the FIB.

The **no match** and **default match** commands remove the **match** statement from the configuration mode policy map by deleting the corresponding command from ***running-config***.

Command Mode

Policy-Map (pbr) Configuration
accessed through **policy-map type pbr** command

Command Syntax

```
[sequence_number] match ip SOURCE_ADDR DEST_ADDR [set nexthop [recursive]
NH-addr_1 [NH-addr_2] ... [NH-addr_n]]

no match ip SOURCE_ADDR DEST_ADDR [set nexthop [recursive] NH-addr_1 [NH-addr_2]
... [NH-addr_n]]

no SEQ_NUM

default match match ip SOURCE_ADDR DEST_ADDR [set nexthop [recursive] NH-addr_1
[[NH-addr_2] ... [NH-addr_n]]

default SEQ_NUM
```

Parameters

- *sequence_number* Sequence number assigned to the rule. If no number is entered, the number is derived by adding 10 to the number of the policy map's last numbered line. To increase the distance between existing entries, use the **resequence** command.
- ***SOURCE_ADDR*** and ***DEST_ADDR*** source and destination address filters. Options include:
  - *network_addr* subnet address (CIDR or address-mask).
  - **any** packets from or to all addresses are matched.
  - **host** *ip_addr* IP address (dotted decimal notation).

  Source and destination subnet addresses support discontiguous masks.
- **recursive** enables recursive next hop resolution.
- *NH_addr* IP address of next hop. If multiple addresses are entered, they are treated as an ECMP group.

Related Commands

- **policy-map type pbr** enters policy-map (PBR) configuration mode.

Example

- These commands create a match rule in policy map "PMAP1" which sets the next hop to 192.168.3.5 for packets received from 172.16.0.0/12 regardless of their destination, then exit the mode to save the changes.

```
switch(config)#policy-map type pbr PMAP1
switch(config-pmap-PMAP1)#match ip 172.16.0.0/12 any set nexthop 192.163.3.5
switch(config-pmap-PMAP1)#exit
switch(config)#
```

platform arad tcam counters feature

The **platform arad tcam counters feature** command enables incrementing PBR hardware counters corresponding to ACL. If counters for PBR are enabled, then counters for ACL will be automatically disabled in all cases. If counters for ACL are enabled, then counters for PBR will be automatically disabled in all cases.

The **no platform arad tcam counters feature** command disables PBR/ACL counters selection. The **default platform arad tcam counters feature** commands resets the default behavior.

Command Mode

Global Configuration

Command Syntax

```
platform arad tcam counters feature [OPTIONS]
no platform arad tcam counters feature [OPTIONS]
default platform arad tcam counters feature [OPTIONS]
```

Parameters

- ***OPTIONS*** Assign the TCAM counters feature. Options include:
  - **pbr** assign the TCAM counters feature PBR hardware counters.
  - **acl** assign the TCAM counters feature ACL hardware counters.

Example

- This command enables incrementing ACL hardware counters selection.

```
switch(config)#platform arad tcam counters feature acl
switch(config)#
```

- This command disables incrementing ACL hardware counters selection.

```
switch(config)#no platform arad tcam counters feature acl
switch(config)#
```

policy-map type control-plane

Note On hardware purchased after July 3, 2017, CoPP is not available.

The **policy-map type control-plane** command places the switch in Policy-Map (control plane) configuration mode, which is a group change mode that modifies a control-plane policy map. A policy map is a data structure that consists of class maps that identify a specific data stream and specify bandwidth and shaping parameters that controls its transmission. Control plane policy maps are applied to the control plane to manage traffic.

The **copp-system-policy** policy map is supplied with the switch and is always applied to the control plane. **Copp-system-policy** is the only valid control plane policy map.

The **exit** command saves pending policy map changes to ***running-config*** and returns the switch to global configuration mode. Policy map changes are also saved by entering a different configuration mode. The **abort** command discards pending changes, returning the switch to global configuration mode.

The **no policy-map type control-plane** and **default policy-map type control-plane** commands delete the specified policy map by removing the corresponding **policy-map type control-plane** command and its associated configuration.

Command Mode

Global Configuration

Command Syntax

```
policy-map type control-plane copp-system-policy
no policy-map type control-plane copp-system-policy
default policy-map type control-plane copp-system-policy
```

copp-system-policy is supplied with the switch and is the only valid control plane policy map.

Commands Available in Policy-Map Configuration Mode

- **class (policy-map (control-plane) – FM6000)**
- **class (policy-map (control-plane) – Trident)**

Related Commands

- **class-map type control-plane** enters control-plane class-map configuration mode for modifying a control-plane dynamic class map.

  Only Helix and Trident platform switches support dynamic classes for control plane policing.

Example

- This command places the switch in policy-map configuration mode to edit the copp-system-policy policy map.

```
switch(config)#policy-map type control-plane copp-system-policy
switch(config-pmap-copp-system-policy)#
```

policy-map type pbr

The **policy-map type pbr** command places the switch in policy-map (pbr) configuration mode, which is a group change mode that modifies a Policy-Based Routing (PBR) policy map. The command also creates the specified policy map if it does not already exist. A PBR policy map is a data structure that consists of class maps that identify specific packets and the next hops for those packets. Policy maps are applied to Ethernet or port channel interfaces to manage traffic.

The **exit** command saves pending policy map changes to ***running-config*** and returns the switch to global configuration mode. Policy map changes are also saved by entering a different configuration mode. The **abort** command discards pending changes, returning the switch to global configuration mode.

The **no policy-map type pbr** and **default policy-map type pbr** commands delete the specified policy map by removing the corresponding **policy-map type pbr** command and its associated configuration.

Command Mode

Global Configuration

Command Syntax

```
policy-map type pbr map_name
no policy-map type pbr map_name
default policy-map type pbr map_name
```

Parameters

- *map_name* Name of policy map.

Commands Available in Policy-Map configuration mode

- **class (policy-map (pbr))**
- **match (policy-map (pbr))**

Related Commands

- **class-map type pbr**
- **service-policy type pbr (Interface mode)**

Example

- This command creates the PBR policy map named PMAP1 and places the switch in policy-map configuration mode.

```
switch(config)#policy-map type pbr PMAP1
switch(config-pmap-PMAP1)#
```

policy-map type pdp

The **policy-map type pdp** command places the switch in policy-map (pdp) configuration mode, which is a group change mode that modifies a Per-port Denial-of-service Protection (PDP) policy map. The command also creates the specified policy map if it does not already exist.

The **exit** command saves pending policy map changes to ***running-config*** and returns the switch to global configuration mode. Policy map changes are also saved by entering a different configuration mode. The **abort** command discards pending changes, returning the switch to global configuration mode.

The **no policy-map type pdp** and **default policy-map type pdp** commands delete the specified policy map by removing the corresponding **policy-map type pdp** command and its associated configuration.

Note PDP is available only in EOS versions 4.19.0F and above.

Command Mode

Global Configuration

Command Syntax

```
policy-map type pdp [shared] map_name
no policy-map type pdp [shared] map_name
default policy-map type pdp [shared] map_name
```

Parameters

- **shared** optional keyword designating a shared policy map.
- *map_name* name of policy map.

Example

- This command creates the unshared PDP policy map named PMAP1 and places the switch in policy-map configuration mode.

```
switch(config)#policy-map type pdp PMAP1
   switch(config-pmap-PMAP1)#
```

policy-map type qos

The **policy-map type qos** command places the switch in Policy-Map (qos) configuration mode, which is a group change mode that modifies a QoS policy map. A policy map is a data structure that consists of class maps that identify a specific data stream and shaping parameters that controls its transmission. Policy maps are applied to Ethernet or port channel interfaces to manage traffic.

The **exit** command saves pending policy map changes to ***running-config*** and returns the switch to global configuration mode. Policy map changes are also saved by entering a different configuration mode. The **abort** command discards pending changes, returning the switch to global configuration mode.

The **no policy-map type qos** and **default policy-map type qos** commands delete the specified policy map by removing the corresponding **policy-map type qos** command and its associated configuration. The **policy-map** and **policy-map type qos** commands are equivalent.

Command Mode

Global Configuration

Command Syntax

```
policy-map [type qos] map_name
no policy-map [type qos] map_name
default policy-map [type qos] map_name
```

policy-map *map_name* and **policy-map type qos** *map_name* are identical commands.

Parameters

- *map_name* Name of policy map.

Commands Available in Policy-Map Configuration Mode

- **class (policy-map (qos) – FM6000)**
- **class (policy-map (qos) – Trident)**

Related Commands

- **class-map type qos**
- **service-policy type qos (Interface mode)**

Example

- This command creates the QoS policy map named PMAP-1 and places the switch in policy-map configuration mode.

```
switch(config)#policy-map PMAP-1
switch(config-pmap-PMAP-1)#
```

resequence (class-map (pbr))

The **resequence** command assigns sequence numbers to rules in the configuration mode class map. Command parameters specify the number of the first rule and the numeric interval between consecutive rules. Once changed, rule numbers persist unless changed again using the **resequence** command, but the interval used for numbering new rules reverts to 10 on exiting class-map (pbr) configuration mode.

Maximum rule sequence number is 4294967295.

Command Mode

Class-Map (PBR) Configuration
accessed through **class-map type pbr** command

Command Syntax

resequence [*start_num* [*inc_num*]]

Parameters

- *start_num* sequence number assigned to the first rule. Default is 10.
- *inc_num* numeric interval between consecutive rules. Default is 10.

Example

- The **resequence** command renumbers the rules in CMAP1, starting the first command at number 100 and incrementing subsequent lines by 20.

```
switch(config)#class-map type pbr match-any CMAP1
switch(config-cmap-CMAP1)#show active
class-map type pbr match-any CMAP1
   10 match ip access-group group1
   20 match ip access-group group2
   30 match ip access-group group3
switch(config-cmap-CMAP1)#resequence 100 20
switch(config-cmap-CMAP1)#exit
switch(config)#class-map type pbr match-any CMAP1
switch(config-cmap-CMAP1)#show active
class-map type pbr match-any CMAP1
        100 match ip access-group group1
        120 match ip access-group group2
        140 match ip access-group group3
```

resequence (policy-map (pbr))

The **resequence** command assigns sequence numbers to rules in the configuration mode policy map. Command parameters specify the number of the first rule and the numeric interval between consecutive rules. Once changed, rule numbers persist unless changed again using the **resequence** command, but the interval used for numbering new rules reverts to 10 on exiting policy-map (pbr) configuration mode.

Maximum rule sequence number is 4294967295.

Command Mode

Policy-Map (PBR) Configuration
accessed through **policy-map type pbr** command

Command Syntax

resequence [*start_num* [*inc_num*]]

Parameters

- *start_num* sequence number assigned to the first rule. Default is 10.
- *inc_num* numeric interval between consecutive rules. Default is 10.

Example

- The **resequence** command renumbers the rules in PMAP1, starting the first command at number 100 and incrementing subsequent lines by 20.

```
switch(config)#policy-map type pbr PMAP1
switch(config-pmap-PMAP1)#show active
policy-map type pbr PMAP1
   10 class CMAP1
      set nexthop 172.16.1.1
   20 class CMAP2
      set nexthop 172.16.2.2
   30 class CMAP3
      set nexthop 172.16.3.3
switch(config-pmap-PMAP1)#resequence 100 20
switch(config-pmap-PMAP1)#exit
switch(config)#policy-map type pbr PMAP1
switch(config-pmap-PMAP1)#show active
class-map type pbr PMAP1
   100 class CMAP1
      set nexthop 172.16.1.1
   120 class CMAP2
      set nexthop 172.16.2.2
   140 class CMAP3
      set nexthop 172.16.3.3
switch(config-pmap-PMAP1)#
```

service-policy type pbr (Interface mode)

The **service-policy pbr** command applies the specified Policy-Based Routing (PBR) policy map to the configuration mode interface. A PBR policy map is a data structure that consists of class maps that identify specific packets and the next hops for those packets. Policy maps are applied to Ethernet or port channel interfaces to manage traffic. Only one service policy is supported per interface.

The **no service-policy pbr and default service-policy pbr** commands remove the service policy assignment from the configuration mode interface by deleting the corresponding **service-policy pbr** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration
Interface-Port-Channel Configuration
Interface-VLAN Configuration

Command Syntax

```
service-policy type pbr TRAFFIC_DIRECTION map_name
no service-policy pbr TRAFFIC_DIRECTION map_name
default service-policy pbr TRAFFIC_DIRECTION map_name
```

Parameters

- ***TRAFFIC_DIRECTION*** IP address or peer group name. Values include:
  - **input** Policy map applies to inbound packet streams.
- *map_name* Name of policy map.

Guidelines

A policy map that is attached to a port channel interface takes precedence for member interfaces of the port channel over their individual Ethernet interface configuration. Members that are removed from a port channel revert to the policy map implementation specified by its Ethernet interface configuration.

Related Commands

- **policy-map type pbr**

Example

- This command applies the PBR policy map “PMAP1” to Ethernet interface 8.

```
switch#config
switch(config)#interface ethernet 8
switch(config-if-Et8)#service-policy type pbr input PMAP1
switch(config-if-Et8)#
```

service-policy type pdp (Interface mode)

The **service-policy pdp** command assigns the specified Per-port Denial-of-service (PDP) policy map to the configuration mode interface.

The **no service-policy pdp** and **default service-policy pdp** commands remove the service policy assignment from the configuration mode interface by deleting the corresponding **service-policy pdp** command from ***running-config***.

Note PDP is available only in EOS versions 4.19.0F and above.

Command Mode

Interface-Ethernet Configuration

Command Syntax

```
service-policy type pdp input map_name
no service-policy pdp input [map_name]
default service-policy pdp input [map_name]
```

Parameters

- *map_name* name of policy map.

Example

- This command assigns the PDP policy map “PMAP1” to Ethernet interface 8.

```
switch#config
switch(config)#interface ethernet 8
switch(config-if-Et8)#service-policy type pdp input PMAP1
   switch(config-if-Et8)#
```

service-policy type qos (Interface mode)

The **service-policy** command applies a specified policy map to the configuration mode interface. A policy map is a data structure that identifies data traffic through class maps, then specifies actions to classify the traffic (by setting the traffic class), mark the traffic (by setting the cos and dscp values), and police the traffic (by setting the police rate) through data packet field modifications.

The **no service-policy and default service-policy** commands remove the service policy assignment from the configuration mode interface by deleting the corresponding **service-policy** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration
Interface-Port-Channel Configuration
Interface-VLAN Configuration

Command Syntax

```
service-policy [type qos] TRAFFIC_DIRECTION map_name
no service-policy [type qos] TRAFFIC_DIRECTION map_name
default service-policy [type qos] TRAFFIC_DIRECTION map_name
```

Parameters

- **type qos** Parameter has no functional effect.
- ***TRAFFIC_DIRECTION*** Direction of data stream to which command applies. Options include:
  - **input** Policy map applies to inbound packet streams.
- *map_name* Name of policy map.

Guidelines

A policy map that is attached to a port channel interface takes precedence for member interfaces of the port channel over their individual Ethernet interface configuration. Members that are removed from a port channel revert to the policy map implementation specified by its Ethernet interface configuration.

DCS-7500E and DCS-7280E limitations:

- A maximum of 31 QoS service policies per chip may be applied on L3 interfaces.
- Applying different QoS service policies to an SVI and its member interfaces causes unpredictable behavior.
- When an SVI on which QoS service policies are applied experiences partial failure due to limited hardware resources, a forwarding agent restart will cause unpredictable behavior.
- Policy-map programming may fail when QoS service policies are applied on two SVIs if an event causes a member interface to switch membership from one to the other. To change the VLAN membership of an interface in this case, remove the interface from one VLAN before adding it to the other.
- Outgoing COS rewrite is not supported.
- QoS policy-map counters are not supported.

DCS-7010, DCS-7050, DCS-7050X, DCS-7250X, and DCS-7300X limitations:

- When the same policy map is applied to multiple SVIs, TCAM resources are not shared.
- A policy map applied to an SVI will result in TCAM allocation on all chips whether SVI members are present or not.
- Applying different QoS service policies to an SVI and its member interfaces causes unpredictable behavior.

Related Commands

- **policy-map type qos**

Example

- This command applies the PMAP-1 policy map to Ethernet interface 8.

```
switch#config
switch(config)#interface ethernet 8
switch(config-if-Et8)#show active
switch(config-if-Et8)#service-policy input PMAP-1
switch(config-if-Et8)#show active
interface Ethernet8
   service-policy type qos input PMAP-1
switch(config-if-Et8)#
```

set (policy-map-class (qos) – FM6000)

The **set** command specifies traffic resolution methods for traffic defined by its associated class map in its configuration mode policy map class. Three set statements are available for each class:

- **cos** Sets the layer 2 class of service field.
- **dscp** Sets the differentiated services code point value in the type of service (ToS) byte.
- **traffic-class** Sets the traffic class queue for data packets.

Each type of set command can be assigned to a class, allowing for the simultaneous modification of both (cos, dscp) fields and assignment to a traffic class.

The **no set** and **default set** commands remove the specified data action from the class map by deleting the associated **set** command from ***running-config***.

Command Mode

Policy-map-class (qos) configuration
accessed through **class (policy-map (qos) – FM6000)** command

Command Syntax

```
set QOS_TYPE value
no set QOS_TYPE
default set QOS_TYPE
```

Parameters

- ***QOS_TYPE*** Specifies the data stream resolution method. Valid options include:
  - **cos** Layer 2 class of service field of outbound packet is modified.
  - **dscp** Differentiated services code point value in the ToS byte is modified.
  - **traffic-class** Data stream is assigned to a traffic class queue.
- *value* Specifies the data field value or traffic class queue. Valid data range depends on ***QOS_TYPE***.
  - *QOS_TYPE is* **cos** Value ranges from 0 to 7.
  - *QOS_TYPE is* **dscp** Value ranges from 0 to 63.
  - *QOS_TYPE is* **traffic-class** Value ranges from 0 to 7.

Related Commands

- **policy-map type qos**
- **class (policy-map (qos) – FM6000)**

Example

- These commands configure the policy map to set the CoS field to 7 to data traffic specified by the class map CMAP-1, then assigns that data to traffic class queue 4.

```
switch(config)#policy-map type qos PMAP-1
switch(config-pmap-PMAP-1)#class CMAP-1
switch(config-pmap-c-PMAP-1-CMAP-1)#set cos 7
switch(config-pmap-c-PMAP-1-CMAP-1)#set traffic-class 4
switch(config-pmap-c-PMAP-1-CMAP-1)#
```

set (policy-map-class (qos) – Helix)

The **set** command specifies traffic resolution methods for traffic defined by its associated class map in its configuration mode policy map class. Three set statements are available for each class:

- **cos** Sets the layer 2 class of service field.
- **dscp** Sets the differentiated services code point value in the type of service (ToS) byte.
- **traffic-class** Sets the traffic class queue for data packets.

Each type of set command can be assigned to a class, allowing for the simultaneous modification of both (cos, dscp) fields and assignment to a traffic class.

The **no set** and **default set** commands remove the specified data action from the class map by deleting the associated **set** command from ***running-config***.

Command Mode

Policy-map-class (qos) configuration
accessed through **class (policy-map (qos) – Helix)** command

Command Syntax

```
set QOS_TYPE value
no set QOS_TYPE
default set QOS_TYPE
```

Parameters

- ***QOS_TYPE*** Specifies the data stream resolution method. Valid options include:
  - **cos** Layer 2 class of service field of outbound packet is modified.
  - **dscp** Differentiated services code point value in the ToS byte is modified.
  - **traffic-class** Data stream is assigned to a traffic class queue.
- *value* Specifies the data field value or traffic class queue. Valid data range depends on QOS type.
  - *QOS_TYPE is* **cos** Value ranges from 0 to 7.
  - *QOS_TYPE is* **dscp** Value ranges from 0 to 63.
  - *QOS_TYPE is* **traffic-class** Value ranges from 0 to 7.

Related Commands

- **policy-map type qos**
- **class (policy-map (qos) – Helix)**

Example

- These commands configure the policy map to set the CoS field to 7 to data traffic specified by the class map CMAP-1, then assigns that data to traffic class queue 4.

```
switch(config)#policy-map type qos PMAP-1
switch(config-pmap-PMAP-1)#class CMAP-1
switch(config-pmap-c-PMAP-1-CMAP-1)#set cos 7
switch(config-pmap-c-PMAP-1-CMAP-1)#set traffic-class 4
switch(config-pmap-c-PMAP-1-CMAP-1)#
```

set (policy-map-class (qos) – Trident)

The **set** command specifies traffic resolution methods for traffic defined by its associated class map in its configuration mode policy map class. Three set statements are available for each class:

- **cos** Sets the layer 2 class of service field.
- **dscp** Sets the differentiated services code point value in the type of service (ToS) byte.
- **traffic-class** Sets the traffic class queue for data packets.

Each type of set command can be assigned to a class, allowing for the simultaneous modification of both (cos, dscp) fields and assignment to a traffic class.

The **no set** and **default set** commands remove the specified data action from the class map by deleting the associated **set** command from ***running-config***.

Command Mode

Policy-map-class (qos) configuration
accessed through **class (policy-map (qos) – Trident)** command

Command Syntax

```
set QOS_TYPE value
no set QOS_TYPE
default set QOS_TYPE
```

Parameters

- ***QOS_TYPE*** Specifies the data stream resolution method. Valid options include:
  - **cos** Layer 2 class of service field of outbound packet is modified.
  - **dscp** Differentiated services code point value in the ToS byte is modified.
  - **traffic-class** Data stream is assigned to a traffic class queue.
- *value* Specifies the data field value or traffic class queue. Valid data range depends on QOS type.
  - *QOS_TYPE is* **cos** Value ranges from 0 to 7.
  - *QOS_TYPE is* **dscp** Value ranges from 0 to 63.
  - *QOS_TYPE is* **traffic-class** Value ranges from 0 to 7.

Related Commands

- **policy-map type qos**
- **class (policy-map (qos) – Trident)**

Example

- These commands configure the policy map to set the CoS field to 7 to data traffic specified by the class map CMAP-1, then assigns that data to traffic class queue 4.

```
switch(config)#policy-map type qos PMAP-1
switch(config-pmap-PMAP-1)#class CMAP-1
switch(config-pmap-c-PMAP-1-CMAP-1)#set cos 7
switch(config-pmap-c-PMAP-1-CMAP-1)#set traffic-class 4
switch(config-pmap-c-PMAP-1-CMAP-1)#
```

set (policy-map-class (qos) – Trident II)

The **set** command specifies traffic resolution methods for traffic defined by its associated class map in its configuration mode policy map class. Three set statements are available for each class:

- **cos** Sets the layer 2 class of service field.
- **dscp** Sets the differentiated services code point value in the type of service (ToS) byte.
- **traffic-class** Sets the traffic class queue for data packets.

Each type of set command can be assigned to a class, allowing for the simultaneous modification of both (cos, dscp) fields and assignment to a traffic class.

The **no set** and **default set** commands remove the specified data action from the class map by deleting the associated **set** command from ***running-config***.

Command Mode

Policy-map-class (qos) configuration
accessed through **class (policy-map (qos) – Trident)** command

Command Syntax

```
set QOS_TYPE value
no set QOS_TYPE
default set QOS_TYPE
```

Parameters

- ***QOS_TYPE*** Specifies the data stream resolution method. Valid options include:
  - **cos** Layer 2 class of service field of outbound packet is modified.
  - **dscp** Differentiated services code point value in the ToS byte is modified.
  - **traffic-class** Data stream is assigned to a traffic class queue.
- *value* Specifies the data field value or traffic class queue. Valid data range depends on QOS type.
  - *QOS_TYPE is* **cos** Value ranges from 0 to 7.
  - *QOS_TYPE is* **dscp** Value ranges from 0 to 63.
  - *QOS_TYPE is* **traffic-class** Value ranges from 0 to 7.

Related Commands

- **policy-map type qos**
- **class (policy-map (qos) – Trident)**

Example

- These commands configure the policy map to set the CoS field to 7 to data traffic specified by the class map CMAP-1, then assigns that data to traffic class queue 4.

```
switch(config)#policy-map type qos PMAP-1
switch(config-pmap-PMAP-1)#class CMAP-1
switch(config-pmap-c-PMAP-1-CMAP-1)#set cos 7
switch(config-pmap-c-PMAP-1-CMAP-1)#set traffic-class 4
switch(config-pmap-c-PMAP-1-CMAP-1)#
```

set nexthop (policy-map-class – pbr)

The **set nexthop** command specifies the next hop for traffic defined by its associated class map in its configuration mode policy map class.

The **no set nexthop** and **default set nexthop** commands remove the specified action from the class map by deleting the associated **set nexthop** command from ***running-config***.

Command Mode

Policy-map-class (pbr) configuration
accessed through **class (policy-map (pbr))** command

Command Syntax

```
set nexthop [recursive] NH-addr_1 [NH-addr_2] ... [NH-addr_n]
no set nexthop [recursive]
default set nexthop [recursive]
```

Parameters

- **recursive** enables recursive next hop resolution.
- *NH_addr* IP address of next hop. If multiple addresses are entered, they are treated as an ECMP group.

Related Commands

- **policy-map type pbr**
- **class (policy-map (pbr))**

Example

- These commands configure the policy map PMAP1 to set the next hop to 192.168.5.3 for traffic defined by class map CMAP1.

```
switch(config)#policy-map type pbr PMAP1
switch(config-pmap-PMAP1)#class CMAP1
switch(config-pmap-c-PMAP1-CMAP1)#set nexthop 192.168.5.3
switch(config-pmap-c-PMAP1-CMAP1)#
```

set nexthop-group (policy-map-class(pbr) – Arad)

The **set nexthop-group** command specifies a nexthop group as the next hop for traffic defined by its associated class map in its configuration mode policy map class.

The **no set nexthop-group** and **default set nexthop-group** commands remove the specified action from the class map by deleting the associated **set nexthop-group** command from ***running-config***.

Command Mode

Policy-map-class (pbr) configuration
accessed through **class (policy-map (pbr))** command

Command Syntax

```
set nexthop-group group_name
no set nexthop-group
default set nexthop-group
```

Parameters

- *group_name* name of ECMP group to use as next hop.

Related Commands

- **policy-map type pbr**
- **class (policy-map (pbr))**

Example

- These commands configure the policy map PMAP1 to set the next hop to a nexthop group named GROUP1 for traffic defined by class map CMAP1.

```
switch(config)#policy-map type pbr PMAP1
switch(config-pmap-PMAP1)#class CMAP1
switch(config-pmap-c-PMAP1-CMAP1)#set nexthop-group GROUP1
switch(config-pmap-c-PMAP1-CMAP1)#
```

shape (policy-map-class (control-plane) – Arad)

Note On hardware purchased after July 3, 2017, CoPP is not available.

The **shape** command specifies the maximum bandwidth for traffic filtered by the configuration mode policy map class.

The **no shape** and **default shape** commands remove the maximum bandwidth restriction for the configuration mode class by deleting the corresponding **bandwidth** command from ***running-config***.

Command Mode

Policy-map-class (control plane) configuration
accessed through **class (policy-map (control-plane) – Arad)**

Command Syntax

```
shape kbps kbits
no shape
default shape
```

Parameters

- *kbits* Maximum data rate (kbps per second). Value ranges from 1 to 10000000.

Related Commands

- **class (policy-map (control-plane) – Arad)** places the switch in policy-map-class (control plane) configuration mode.
- **bandwidth (policy-map-class (control-plane) – Arad)** specifies the minimum bandwidth for traffic defined by its associated class map in its configuration mode policy map class.

Static Classes Default Shape

Arad platform switches define these default shapes for static classes:

- copp-system-bgp2500•copp-system-l3lpmoverflow2500
- copp-system-bpdu2500•copp-system-l3slowpath2500
- copp-system-default2500•copp-system-l3ttl12500
- copp-system-ipbroadcast2500•copp-system-lacp2500
- copp-system-ipmc2500•copp-system-linklocal2500
- copp-system-ipmcmiss2500•copp-system-lldp2500
- copp-system-ipunicastNO LIMIT•copp-system-mlag2500
- copp-system-l2broadcast2500•copp-system-multicastsnoop2500
- copp-system-l2unicastNO LIMIT•copp-system-OspfIsis2500
- copp-system-l3destmiss2500•copp-system-sflow2500

Example

- These commands configure the maximum bandwidth of 2000 kbps for data traffic specified by the class map copp-system-lldp of the default control-plane policy map.

```
switch(config)#policy-map type control-plane copp-system-policy
switch(config-pmap-copp-system-policy)#class copp-system-lldp
switch(config-pmap-c-copp-system-policy-copp-system-lldp)#shape kbps 2000
switch(config-pmap-c-copp-system-policy-copp-system-lldp)#exit
switch(config-pmap-copp-system-policy)#exit
switch(config)#show policy-map interface control-plane copp-system-policy
Service-policy input: copp-system-policy
             <-------OUTPUT OMITTED FROM EXAMPLE-------->

  Class-map: copp-system-lldp (match-any)
       shape : 200 kbps
       bandwidth : 250 kbps
      Out Packets : 0
      Drop Packets : 0

             <-------OUTPUT OMITTED FROM EXAMPLE-------->

switch(config)#
```

shape (policy-map-class (control-plane) – FM6000)

Note On hardware purchased after July 3, 2017, CoPP is not available.

The **shape** command specifies the maximum bandwidth for traffic filtered by the configuration mode policy map class.

The **no shape** and **default shape** commands remove the maximum bandwidth restriction for the configuration mode class by deleting the corresponding **bandwidth** command from ***running-config***.

Command Mode

Policy-map-class (control plane) configuration
accessed through **class (policy-map (control-plane) – FM6000)**

Command Syntax

```
shape pps packets
no shape
default shape
```

Parameters

- *packets* Minimum data rate (packets per second). Value ranges from 1 to 100000.

Related Commands

- **class (policy-map (control-plane) – FM6000)** places the switch in policy-map-class (control plane) configuration mode.
- **bandwidth (policy-map-class (control-plane) – FM6000)** specifies the minimum bandwidth for traffic defined by its associated class map in its configuration mode policy map class.

Static Classes Default Shape

FM6000 platform switches define these default shapes for static classes:

- copp-system-arp10000•copp-system-l3slowpath10000
- copp-system-default8000•copp-system-pim-ptp10000
- copp-system-ipmcrsvd10000•copp-system-ospf-isis10000
- copp-system-ipmcmiss10000•copp-system-selfip5000
- copp-system-igmp10000•copp-system-selfip-tc6to75000
- copp-system-l2rsvd10000•copp-system-sflow25000

Example

- These commands configure the maximum bandwidth of 5000 packets per second for data traffic specified by the class map PMAP-1 in the policy map named copp-system-policy.

```
switch(config)#policy-map type control-plane copp-system-policy
switch(config-pmap-copp-system-policy)#class PMAP-1
switch(config-pmap-c-copp-system-policy-PMAP-1)#shape pps 5000
switch(config-pmap-c-copp-system-policy-PMAP-1)#
```

shape (policy-map-class (control-plane) – Helix)

Note On hardware purchased after July 3, 2017, CoPP is not available.

The **shape** command specifies the maximum bandwidth for traffic filtered by the configuration mode policy map class.

The **no shape** and **default shape** commands remove the maximum bandwidth restriction for the configuration mode class by deleting the corresponding **bandwidth** command from ***running-config***.

Command Mode

Policy-map-class (control plane) configuration
accessed through **class (policy-map (control-plane) – Helix)**

Command Syntax

```
shape pps packets
no shape
default shape
```

Parameters

- *packets* Minimum data rate (packets per second). Value ranges from 1 to 100000.

Static Classes Default Shape

Trident platform switches define these default shapes for static classes:

- copp-system-acllog10000•copp-system-l3ttl110000
- copp-system-arp10000•copp-system-lacp5000
- copp-system-arpresolver10000•copp-system-lldp10000
- copp-system-bfd10000•copp-system-mlag5000
- copp-system-bgp5000•copp-system-OspfIsis10000
- copp-system-bpdu5000•copp-system-selfip5000
- copp-system-default8000•copp-system-selfip-tc6to75000
- copp-system-glean10000•copp-system-sflow25024
- copp-system-igmp10000•copp-system-tc3to510000
- copp-system-ipmcmiss10000•copp-system-tc6to710000
- copp-system-ipmcrsvd10000•copp-system-urm10000
- copp-system-l3destmiss10000•copp-system-vrrp5000
- copp-system-l3slowpath10000

Related Commands

- **class (policy-map (control-plane) – Helix)** places the switch in policy-map-class (control plane) configuration mode.
- **bandwidth (policy-map-class (control-plane) – Helix)** specifies the minimum bandwidth for traffic defined by its associated class map in its configuration mode policy map class.

Example

- These commands configure the maximum bandwidth of 5000 packets per second for data traffic specified by the copp-system-lldp of the default control-plane policy map

```
switch(config)#policy-map type control-plan copp-system-policy
switch(config-pmap-copp-system-policy)#class copp-system-lldp
switch(config-pmap-c-copp-system-policy-copp-system-lldp)#shape pps 5000
switch(config-pmap-c-copp-system-policy-copp-system-lldp)#exit
switch(config-pmap-copp-system-policy)#exit
switch(config)#show policy-map interface control-plane copp-system-policy
Service-policy input: copp-system-policy
             <-------OUTPUT OMITTED FROM EXAMPLE-------->
  Class-map: copp-system-lldp (match-any)
       shape : 5000 pps
       bandwidth : 500 pps
      Out Packets : 305961
      Drop Packets : 0
             <-------OUTPUT OMITTED FROM EXAMPLE-------->
switch(config)#
```

shape (policy-map-class (control-plane) – Petra)

Note On hardware purchased after July 3, 2017, CoPP is not available.

The **shape** command specifies the maximum bandwidth for traffic filtered by the configuration mode policy map class.

The **no shape** and **default shape** commands remove the maximum bandwidth restriction for the configuration mode class by deleting the corresponding **bandwidth** command from ***running-config***.

Command Mode

Policy-map-class (control plane) configuration
accessed through **class (policy-map (control-plane) – Petra)**

Command Syntax

```
shape kbps kbits
no shape
default shape
```

Parameters

- *kbits* Maximum data rate (kbps per second). Value ranges from 1 to 10000000.

Related Commands

- **class (policy-map (control-plane) – Petra)** places the switch in policy-map-class (control plane) configuration mode.
- **bandwidth (policy-map-class (control-plane) – Petra)** specifies the minimum bandwidth for traffic defined by its associated class map in its configuration mode policy map class.

Static Classes Default Shape

Petra platform switches define these default shapes for static classes:

- copp-system-bpdu2500•copp-system-l3destmiss2500
- copp-system-default2500•copp-system-l3slowpath2500
- copp-system-igmp2500•copp-system-l3ttl02500
- copp-system-ipbroadcast2500•copp-system-l3ttl12500
- copp-system-ipmc2500•copp-system-lacp2500
- copp-system-ipmcmiss2500•copp-system-lldp2500
- copp-system-ipmcrsvd2500•copp-system-unicast-arp2500
- copp-system-ipunicastNo Limit

Guidelines

Petra does not support all discrete rate values. When a specified discrete value is not supported, the switch converts the rate to the next highest discrete value that it supports. The **show** commands displays the converted rate and not the user configured rate.

Example

- These commands configure the maximum bandwidth of 2000 kbps for data traffic specified by the class map copp-system-lldp of the default control-plane policy map. Because the switch does not support the discrete value of 2000 kbps, it converts the bandwidth up to 2115 kbps.

```
switch(config)#policy-map type control-plane copp-system-policy
switch(config-pmap-copp-system-policy)#class copp-system-lldp
switch(config-pmap-c-copp-system-policy-copp-system-lldp)#shape kbps 2000
switch(config-pmap-c-copp-system-policy-copp-system-lldp)#exit
switch(config-pmap-copp-system-policy)#exit
switch(config)#show policy-map interface control-plane copp-system-policy
Service-policy input: copp-system-policy
             <-------OUTPUT OMITTED FROM EXAMPLE-------->

  Class-map: copp-system-lldp (match-any)
       shape : 2115 kbps
       bandwidth : 325 kbps
      Out Packets : 0
      Drop Packets : 0

             <-------OUTPUT OMITTED FROM EXAMPLE-------->

switch(config)#
```

shape (policy-map-class (control-plane) – Trident)

Note On hardware purchased after July 3, 2017, CoPP is not available.

The **shape** command specifies the maximum bandwidth for traffic filtered by the configuration mode policy map class.

The **no shape** and **default shape** commands remove the maximum bandwidth restriction for the configuration mode class by deleting the corresponding **bandwidth** command from ***running-config***.

Command Mode

Policy-map-class (control plane) configuration
accessed through **class (policy-map (control-plane) – Trident)**

Command Syntax

```
shape pps packets
no shape
default shape
```

Parameters

- *packets* Minimum data rate (packets per second). Value ranges from 1 to 100000.

Static Classes Default Shape

Trident platform switches define these default shapes for static classes:

- copp-system-arp10000•copp-system-lldp10000
- copp-system-arpresolver10000•copp-system-l3destmiss10000
- copp-system-bpdu5000•copp-system-l3slowpath10000
- copp-system-default8000•copp-system-l3ttl110000
- copp-system-glean10000•copp-system-selfip5000
- copp-system-igmp10000•copp-system-selfip-tc6to75000
- copp-system-ipmcmiss10000•copp-system-sflow25000
- copp-system-ipmcrsvd10000•copp-system-tc3to510000
- copp-system-lacp5000•copp-system-tc6to710000

Related Commands

- **class (policy-map (control-plane) – Trident)** places the switch in policy-map-class (control plane) configuration mode.
- **bandwidth (policy-map-class (control-plane) – Trident)** specifies the minimum bandwidth for traffic defined by its associated class map in its configuration mode policy map class.

Example

- These commands configure the maximum bandwidth of 5000 packets per second for data traffic specified by the class map PMAP-1 in the policy map named copp-system-policy.

```
switch(config)#policy-map type control-plane copp-system-policy
switch(config-pmap-copp-system-policy)#class PMAP-1
switch(config-pmap-c-copp-system-policy-PMAP-1)#shape pps 5000
switch(config-pmap-c-copp-system-policy-PMAP-1)#
```

shape (policy-map-class (control-plane) – Trident-II)

Note On hardware purchased after July 3, 2017, CoPP is not available.

The **shape** command specifies the maximum bandwidth for traffic filtered by the configuration mode policy map class.

The **no shape** and **default shape** commands remove the maximum bandwidth restriction for the configuration mode class by deleting the corresponding **bandwidth** command from ***running-config***.

Command Mode

Policy-map-class (control plane) configuration
accessed through **class (policy-map (control-plane) – Trident-II)**

Command Syntax

```
shape pps packets
no shape
default shape
```

Parameters

- *packets* Minimum data rate (packets per second). Value ranges from 1 to 100000.

Static Classes Default Shape

Trident-II platform switches define these default shapes for static classes:

- copp-system-acllog10000•copp-system-l3slowpath10000
- copp-system-arp10000•copp-system-l3ttl110000
- copp-system-arpresolver10000•copp-system-lacp5000
- copp-system-bfd10000•copp-system-lldp10000
- copp-system-bgp5000•copp-system-mlag5000
- copp-system-bpdu5000•copp-system-selfip5000
- copp-system-default8000•copp-system-selfip-tc6to75000
- copp-system-glean10000•copp-system-sflow25024
- copp-system-igmp10000•copp-system-tc3to510000
- copp-system-ipmcmiss10000•copp-system-tc6to710000
- copp-system-ipmcrsvd10000•copp-system-urm10000
- copp-system-l3destmiss10000

Related Commands

- **class (policy-map (control-plane) – Trident-II)** places the switch in policy-map-class (control plane) configuration mode.
- **bandwidth (policy-map-class (control-plane) – Trident-II)** specifies the minimum bandwidth for traffic defined by its associated class map in its configuration mode policy map class.

Example

- These commands configure the maximum bandwidth of 5000 packets per second for data traffic specified by the copp-system-lldp of the default control-plane policy map

```
switch(config)#policy-map type control-plan copp-system-policy
switch(config-pmap-copp-system-policy)#class copp-system-lldp
switch(config-pmap-c-copp-system-policy-copp-system-lldp)#shape pps 5000
switch(config-pmap-c-copp-system-policy-copp-system-lldp)#exit
switch(config-pmap-copp-system-policy)#exit
switch(config)#show policy-map interface control-plane copp-system-policy
Service-policy input: copp-system-policy
             <-------OUTPUT OMITTED FROM EXAMPLE-------->
  Class-map: copp-system-lldp (match-any)
       shape : 5000 pps
       bandwidth : 500 pps
      Out Packets : 305961
      Drop Packets : 0
             <-------OUTPUT OMITTED FROM EXAMPLE-------->
switch(config)#
```

show class-map type control-plane

The **show class-map** command displays contents of available control-plane class maps. Control-plane class maps can be added to the copp-system-policy policy map. Control-plane class maps can be static class maps defined by the system or dynamic maps created in class-map-configuration mode.

Dynamic class maps are composed of statements that match IPv4 access control lists. Static class maps are defined by the switch and cannot be altered.

Command Mode

EXEC

Command Syntax

show class-map type control-plane [*MAP_NAME*]

Parameters

- ***MAP_NAME*** Name of class map displayed by the command. Options include:
  - <no parameter> Command displays all control plane class maps.
  - *name_text* Command displays specified control-plane class maps.

    show class-map command displays QoS class maps.

Related Commands

- **show class-map type qos** displays control plane class maps.

Example

- This command displays all control plane class maps.
- This command displays the available control plane class maps.

```
switch>show class-map type control-plane
  Class-map: CM-CP1 (match-any)
    Match: ip access-group name LIST-CP1
  Class-map: copp-system-acllog (match-any)
  Class-map: copp-system-arp (match-any)
  Class-map: copp-system-arpresolver (match-any)
  Class-map: copp-system-bpdu (match-any)
  Class-map: copp-system-glean (match-any)
  Class-map: copp-system-igmp (match-any)
  Class-map: copp-system-ipmcmiss (match-any)
  Class-map: copp-system-ipmcrsvd (match-any)
  Class-map: copp-system-l3destmiss (match-any)
  Class-map: copp-system-l3slowpath (match-any)
  Class-map: copp-system-l3ttl1 (match-any)
  Class-map: copp-system-lacp (match-any)
  Class-map: copp-system-lldp (match-any)
  Class-map: copp-system-selfip (match-any)
  Class-map: copp-system-selfip-tc6to7 (match-any)
  Class-map: copp-system-sflow (match-any)
  Class-map: copp-system-tc3to5 (match-any)
  Class-map: copp-system-tc6to7 (match-any)
switch>
```

show class-map type pbr

The **show class-map** command displays contents of all available Policy-Based Routing (PBR) class maps, or of a specified PBR class map. PBR class maps are used by PBR policy maps. PBR class maps are dynamic maps that are created in class-map-configuration mode. Dynamic class maps are composed of statements that match IPv4 or IPv6 access control lists.

Command Mode

EXEC

Command Syntax

show class-map type pbr [*map_name***]**

Parameters

- *map_name* Name of class map displayed by the command. If no parameter is entered, command show all available PBR class maps.

Related Commands

- **show policy-map type pbr** displays PBR policy maps.

Example

- This command displays the contents of the PBR class map CMAP1.

```
switch>show class-map type pbr CMAP1
  Class-map: CMAP1 (match-any)
    Match: 10 ip access-group PBRgroup1
    Match: 20 ip access-group PBRgroup2
    Match: 30 ip access-group PBRgroup3
switch>
```

show class-map type qos

The **show class-map** command displays contents of all available QoS class maps. QoS class maps are used by QoS policy maps. QoS class maps are dynamic maps that are created in class-map-configuration mode. Dynamic class maps are composed of statements that match IPv4 or IPv6 access control lists.

Command Mode

EXEC

Command Syntax

show class-map [type qos] [*MAP_NAME*]

Parameters

- ***MAP_NAME*** Name of class map displayed by the command.
  - <no parameter> Command displays all QoS class maps.
  - *name_text* Command displays specified QoS class maps.

    show class-map and **show class-map type qos** are identical commands.

Related Commands

- **show class-map type control-plane** displays control plane class maps.

Example

- This command displays the available QoS class maps.

```
switch>show class-map type qos
  Class-map: CM-Q1 (match-any)
    Match: ipv6 access-group name LIST-1
  Class-map: CM-Q2 (match-any)
    Match: ip access-group name LIST-2
switch>
```

show policy-map type control-plane

The **show policy-map control-plane** command displays contents of control plane policy maps. Control-plane policy maps are applied to the control plane; copp-system-policy is the only supported policy map.

Command options filter the output to display contents of all policy maps, contents of a specified policy map, or contents of a single class map within a specified policy map.

Command Mode

EXEC

Command Syntax

show policy-map type control-plane copp-system-policy [C*MAP_NAME*]

Parameters

- ***CMAP_NAME*** Name of class map displayed by the command.
  - <no parameter> Command displays all class maps in specified policy map.
  - *class_name* Command displays specified class map.

Example

- This command displays the contents of the copp-system-bpdu class map in the copp-system-policy policy maps.

```
switch>show policy-map type control-plane copp-system-policy class
copp-system-bpdu
  Class-map: copp-system-bpdu (match-any)
       shape : 5000 pps
       bandwidth : 5000 pps

switch>
```

show policy-map type pbr

The **show policy-map pbr** command displays contents of Policy-Based Routing (PBR) policy maps. PBR policy maps are applied to Ethernet interfaces, port channel interfaces or switch virtual interfaces (SVIs).

Command options filter the output to either display contents of all policy maps, contents of a specified policy map, or summary contents of all or a specified policy map.

Command Mode

EXEC

Command Syntax

show policy-map type pbr [*PMAP_NAME*] [*DATA_LEVEL*]

Parameters

- ***PMAP_NAME*** Name of policy map displayed by the command.
  - <no parameter> Command displays all policy maps.
  - *policy_map* Command displays specified policy map.
- ***DATA_LEVEL*** Type of information the command displays. Values include:
  - <no parameter> Command displays all class maps in specified policy map.
  - **summary** Command displays summary data for the specified policy map.

Example

- This command displays the contents of all PBR policy maps in ***running-config***.

```
switch#show policy-map type pbr
Service policy PMAP1
  Configured on:
  Applied on:
  10: Class-map: CMAP1 (match-any)
    Match: 10 ip access-group PBRgroup1
    Match: 20 ip access-group PBRgroup2
    Match: 30 ip access-group PBRgroup3
    Configured actions: set nexthop 172.16.10.12
  20: Class-map: CMAP2 (match-any)
Match: 10 ip access-group PBRgroup1
    Match: 10 ip access-group PBRgroup4
    Match: 20 ip access-group PBRgroup5
    Configured actions: set nexthop 192.168.15.15
switch#
```

show policy-map type qos

The **show policy-map qos** command displays contents of QoS policy maps. QoS policy maps are applied to Ethernet or port channel interfaces.

Command options filter the output to either display contents of all policy maps, contents of a specified policy map, or contents of a single class map within a specified policy map.

Command Mode

EXEC

Command Syntax

show policy-map [type qos] [P*MAP_NAME* [C*MAP_NAME*]]

Parameters

- ***PMAP_NAME*** Name of policy map displayed by the command.
  - <no parameter> Command displays all policy maps.
  - *policy_map* Command displays specified policy map.
- ***CMAP_NAME*** Name of class map displayed by the command. This option is available only when the command includes a policy map name.
  - <no parameter> Command displays all class maps in specified policy map.
  - *class_name* Command displays specified class map.

Example

- This command displays the contents of all QoS policy maps in ***running-config***.

```
switch#show policy-map type qos
Service-policy input: PMAP-1
  Hardware programming status: Successful

  Class-map: xeter (match-any)
    Match: ip access-group name LIST-1
       set cos 6

  Class-map: class-default (match-any)

Service-policy PMAP-2

  Class-map: class-default (match-any)

switch#
```

show policy-map type qos counters

The **show policy-map counters** command displays the quantity of packets that are filtered by the ACLs that comprise a specified QoS policy map.

Command Mode

EXEC

Command Syntax

show policy-map [type qos] *pmap_name* **[*TRAFFIC*] counters [*INFO_LEVEL*]**

Parameters

- *pmap_name* Name of policy map displayed by the command.
- ***TRAFFIC*** Filters policy maps by the traffic they manage. Options include:
  - <no parameter> Policy maps that manage interface's ingress traffic (same as **input** option).
  - **input** Policy maps that manage interface's ingress traffic.
- ***INFO_LEVEL*** amount of information that is displayed. Options include:
  - <no parameter> displays summarized information about the policy map.
  - **detail** displays detailed policy map information.

show policy-map interface control-plane

The **show policy-map interface control-plane** command displays contents of the control-plane policy map. Control-plane policy maps are applied to the control plane, and copp-system-policy is the only supported policy map.

Command Mode

EXEC

Command Syntax

show policy-map interface control-plane copp-system-policy

Example

- This command displays the contents and throughput of the policy map applied to the control plane.

```
switch>show policy-map interface control-plane copp-system-policy
Service-policy input: copp-system-policy
  Number of units programmed: 1
  Hardware programming status: Successful

  Class-map: copp-system-bpdu (match-any)
       shape : 5000 pps
       bandwidth : 5000 pps
      Out Packets : 2
      Drop Packets : 0

  Class-map: copp-system-lacp (match-any)
       shape : 5000 pps
       bandwidth : 5000 pps
      Out Packets : 0
      Drop Packets : 0
             <-------OUTPUT OMITTED FROM EXAMPLE-------->

switch>
```

show policy-map interface type qos

The **show policy-map interface** command displays contents of the policy maps applied to specified interfaces or to the control plane.

Command Mode

EXEC

Command Syntax

show policy-map interface *INTERFACE_NAME* [type qos] [*TRAFFIC*]

Parameters

- ***INTERFACE_NAME*** Filters policy map list by interfaces. Options include:
  - **ethernet** *e_range* Ethernet ports for which command displays policy maps.
  - **port-channel** *p_range* Port channels for which command displays policy maps.
- ***TRAFFIC*** Filters policy maps by the traffic they manage. Options include:
  - <no parameter> Policy maps that manage interface's ingress traffic (same as **input** option).
  - **input** Policy maps that manage interface's ingress traffic.

Example

- This command displays the policy maps applied to Ethernet interfaces 7 and 8.

```
switch#show policy-map interface ethernet 7-8
Service-policy input: PMAP-1
  Hardware programming status: Successful

  Class-map: cmap-1 (match-any)
    Match: ip access-group name LIST-2
       set cos 6

  Class-map: class-default (match-any)

Service-policy input: PMAP-2
  Hardware programming status: Successful

  Class-map: cmap-2 (match-any)
    Match: ip access-group name LIST-2
       set dscp 10

  Class-map: class-default (match-any)

switch#
```

show policy-map interface type qos counters

The **show policy-map interface** command displays displays the quantity of packets that are filtered by ACLs applied to a interface.

Command Mode

EXEC

Command Syntax

show policy-map [*INTERFACE_NAME*] [type qos] [*TRAFFIC*] counters

Parameters

- ***INTERFACE_NAME*** Filters policy map list by interfaces. Options include:
  - <no parameter> Displays data for all configured interfaces.
  - **interface ethernet** *e_range* Ethernet ports for which command displays policy maps.
  - **interface port-channel** *p_range* Port channels for which command displays policy maps.
- ***TRAFFIC*** Filters policy maps by the traffic they manage. Options include:
  - <no parameter> Policy maps that manage interface's ingress traffic (same as **input** option).
  - **input** Policy maps that manage interface's ingress traffic.

Example

- This command displays the policy maps applied to Ethernet interfaces 7 and 8.

```
switch#show policy-map interface ethernet 7-8
Service-policy input: PMAP-1
  Hardware programming status: Successful

  Class-map: cmap-1 (match-any)
    Match: ip access-group name LIST-2
       set cos 6

  Class-map: class-default (match-any)

Service-policy input: PMAP-2
  Hardware programming status: Successful

  Class-map: cmap-2 (match-any)
    Match: ip access-group name LIST-2
       set dscp 10

  Class-map: class-default (match-any)

switch#
```

Chapter 27

Open Shortest Path First – Version 2

Open Shortest Path First (OSPF) is a link-state routing protocol that operates within a single autonomous system. OSPF version 2 is defined by RFC 2328.

This chapter contains the following sections.

- Section 27.1: OSPFv2 Introduction
- Section 27.2: OSPFv2 Conceptual Overview
- Section 27.3: Configuring OSPFv2
- Section 27.4: OSPFv2 Examples
- Section 27.5: OSPFv2 Commands

27.1 OSPFv2 Introduction

27.1.1 Supported Features

Arista switches support the following OSPFv2 functions:

- A single OSPFv2 instance
- Intra- and inter-area routing
- Type 1 and 2 external routing
- Broadcast and P2P interfaces
- Stub areas
- Not so stubby areas (NSSA) (RFC 3101)
- MD5 Authentication
- Redistribution of static, IP, and BGP routes into OSPFv2 with route map filtering
- Opaque LSAs (RFC 2370)
- Graceful restart (RFC 3623)

27.1.2 Features Not Supported

The following OSPFv2 functions are not supported in the current version:

- NBMA, demand circuit, and P2MP interfaces
- OSPFv2 MIB support

27.2 OSPFv2 Conceptual Overview

27.2.1 Storing Link States

OSPFv2 is a dynamic, link-state routing protocol, where links represent interfaces or routable paths. Dynamic routing protocols calculate the most efficient path between locations based on bandwidth and device status.

A link state advertisement (LSA) is an OSPFv2 packet that communicates a router's topology to other routers. The link state database (LSDB) stores an area's topology database and is composed of LSAs received from other routers. Routers update the LSDB by storing LSA's from other routers.

27.2.2 Topology

An autonomous system (AS) is the IP domain within which a dynamic protocol controls the routing of traffic. In OSPFv2, an AS is composed of areas, which define the LSDB computation boundaries. All routers in an area store identical LSDBs. Routers in different areas exchange updates without storing the entire database, reducing information maintenance on large, dynamic networks.

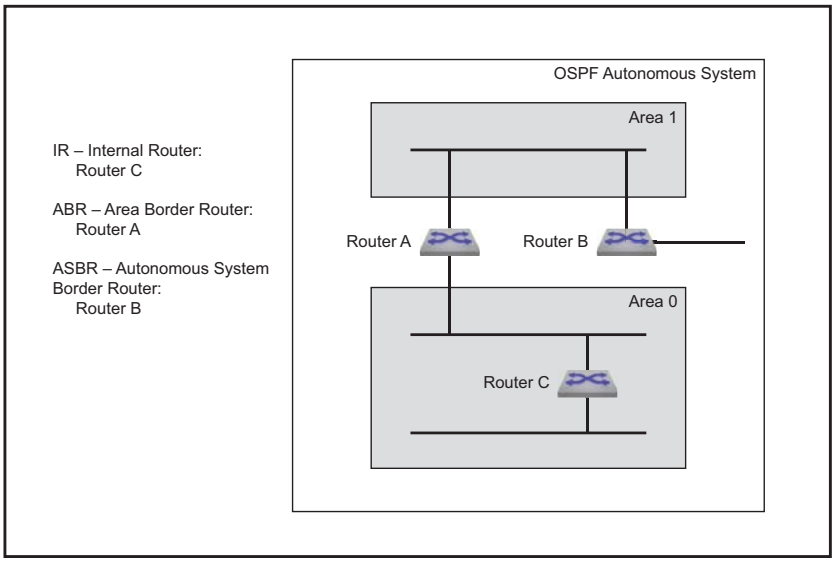
An AS shares internal routing information from its areas and external routing information from other processes to inform routers outside the AS about routes the network can access. Routers that advertise routes on other ASs commit to carry data to the IP space on the route.

OSPFv2 defines these routers:

- Internal router (IR) – a router whose interfaces are contained in a single area. All IRs in an area maintain identical LSDBs.
- Area border router (ABR) – a router that has interfaces in multiple areas. ABRs maintain one LSDB for each connected area.
- Autonomous system boundary router (ASBR) – a gateway router connecting the OSPFv2 domain to external routes, including static routes and routes from other autonomous systems.

Figure 27-1 displays the OSPFv2 router types.

Figure 27-1: OSPFv2 Router Types



OSPFv2 areas are assigned a number between 0 and 4,294,967,295 ($2^{32} – 1$). Area numbers are often expressed in dotted decimal notation, similar to IP addresses.

Each AS has a backbone area, designated as area 0, that connects to all other areas. The backbone receives routing information from all areas, then distributes it to the other areas as required.

OSPFv2 area types include:

- Normal area – accepts intra-area, inter-area, and external routes. The backbone is a normal area.
- Stub area – does not receive router advertisements external to the AS. Stub area routing is based on a default route.
- Not-so-stubby-area (NSSA) – may import external routes from an ASBR, does not receive external routes from the backbone, and does not propagate external routes to other areas.

27.2.3 Link Updates

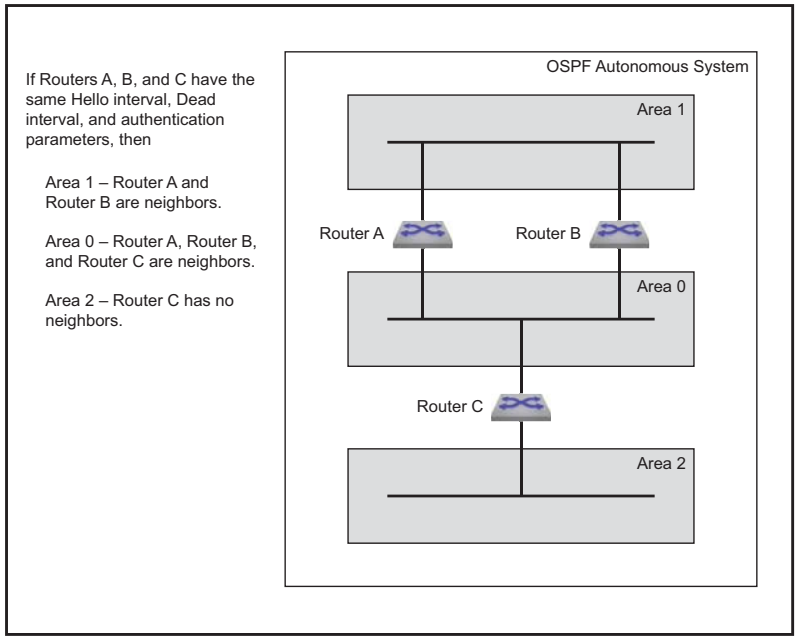
Routers periodically send hello packets to advertise status and establish neighbors. A router's hello packet includes IP addresses of other routers from which it received a hello packet within the time specified by the router dead interval. Routers become neighbors when they detect each other in their hello packets if they:

- share a common network segment.
- are in the same area.
- have the same hello interval, dead interval, and authentication parameters.

Neighbors form adjacencies to exchange LSDB information. A neighbor group uses hello packets to elect a Designated Router (DR) and Backup Designated Router (BDR). The DR and BDR become adjacent to all other neighbors, including each other. Only adjacent neighbors share database information.

Figure 27-2 illustrates OSPFv2 neighbors.

Figure 27-2: OSPFv2 Neighbors



The DR is the central contact for database exchanges. Switches send database information to their DR, which relays the information to the other neighbors. All routers in an area maintain identical LSDBs. Switches also send database information to their BDR, which stores this data without distributing it. If the DR fails, the BDR distributes LSDB information to its neighbors.

OSPFv2 routers distribute LSAs by sending them on all of their active interfaces. The router will generate an LSA for a network defined and active on a passive interface but will not transmit this LSA on the passive interface as no adjacencies are formed.

When a router's LSDB is changed by an LSA, it sends the changes to the DR and BDR for distribution to the other neighbors. Routing information is updated only when the topology changes.

Routers use Dijkstra's algorithm to calculate the shortest path to all known destinations, based on cumulative route cost. The cost of an interface indicates the transmission overhead and is usually inversely proportional to its bandwidth.

27.3 Configuring OSPFv2

These sections describe basic OSPFv2 configuration steps:

- Section 27.3.1: Configuring the OSPFv2 Instance
- Section 27.3.2: Configuring OSPFv2 Areas
- Section 27.3.3: Configuring Interfaces for OSPFv2
- Section 27.3.4: Enabling OSPFv2
- Section 27.3.5: Displaying OSPFv2 Status

27.3.1 Configuring the OSPFv2 Instance

27.3.1.1 Entering OSPFv2 Configuration Mode

The **router ospf** command places the switch in router-ospf configuration mode and creates an OSPFv2 instance if one was not previously created. The switch only supports one OSPFv2 instance and all OSPFv2 configuration commands apply to this instance.

When an OSPFv2 instance is already configured, the command must specify its process ID. Any attempt to define additional instances will fail and generate errors.

The process ID is local to the router and is used to identify the running OSPFv2 process. Neighbor OSPFv2 routers can have different process ID's.

Example

- This command places the switch in router-ospf configuration mode and, if not previously created, creates an OSPFv2 instance with a process ID of 100.

```
switch(config)#router ospf 100
switch(config-router-ospf)#
```

27.3.1.2 Defining the Router ID

The router ID is a 32-bit number assigned to a router running OSPFv2. This number uniquely labels the router within an Autonomous System. Status commands identify the switch through the router ID.

The switch sets the router ID to the first available alternative in the following list:

1. The **router-id** command.
2. The loopback IP address, if a loopback interface is active on the switch.
3. The highest IP address on the router.

Important! When configuring VXLAN on an MLAG, always manually configure the OSPFv2 router ID to prevent the switch from using the common VTEP IP address as the router ID.

The **router-id (OSPFv2)** command configures the router ID for an OSPFv2 instance.

Example

- This command assigns 10.1.1.1 as the OSPFv2 router ID.

```
switch(config-router-ospf)#router-id 10.1.1.1
switch(config-router-ospf)#
```

27.3.1.3 Global OSPFv2 Parameters

These router-ospf configuration mode commands define OSPFv2 behavior.

LSA Overload

The **max-lsa (OSPFv2)** command specifies the maximum number of LSAs allowed in an LSDB database and configures the switch behavior when the limit is approached or exceeded. An LSA overload condition triggers these actions:

- Warning: The switch logs OSPF MAXLSAWARNING if the LSDB contains a specified percentage of the LSA maximum.
- Temporary shutdown: When the LSDB exceeds the LSA maximum, OSPFv2 is disabled and does not accept or acknowledge new LSAs. The switch re-starts OSPFv2 after a specified period.
- Permanent shutdown: The switch permanently disables OSPFv2 after performing a specified number of temporary shutdowns. This state usually indicates the need to resolve a network condition that consistently generates excessive LSA packets.

  OSPFv2 is re-enabled with a **router ospf** command.

The LSDB size restriction is removed by setting the LSA limit to zero.

Example

- This command configures the OSPFv2 maximum LSA count to 20,000 and triggers these actions:
  - The switch logs an OSPF MAXLSAWARNING if the LSDB has 8,000 LSAs (40% of 20,000).
  - The switch temporarily disables OSPFv2 for 10 minutes if the LSDB contains 20,000 LSAs.
  - The switch permanently disables OSPFv2 after four temporary OSPFv2 shutdowns.
  - The shutdown counter resets if the LSDB contains less than 20,000 LSAs for 20 minutes.

```
switch(config-router-ospf)#max-lsa 20000 40 ignore-time 10 ignore-count 4
reset-time 20
switch(config-router-ospf)#
```

Logging Adjacency Changes

The **log-adjacency-changes (OSPFv2)** command configures the switch to log OSPFv2 link-state changes and transitions of OSPFv2 neighbors into the up or down state.

Examples

- This command configures the switch to log transitions of OSPFv2 neighbors into the up or down state.

```
switch(config-router-ospf)#log-adjacency-changes
switch(config-router-ospf)#
```

- This command configures the switch to log all OSPFv2 link-state changes.

```
switch(config-router-ospf)#log-adjacency-changes detail
switch(config-router-ospf)#
```

OSPF RFC Compatibility

RFC 2328 and RFC 1583 specify different methods for calculating summary route metrics. The **compatible (OSPFv2)** command allows the selective disabling of compatibility with RFC 2328.

Example

- This command sets the OSPF compatibility list with RFC 1583.

```
switch(config)#router ospf 6
switch(config-router-ospf)#compatible rfc1583
switch(config-router-ospf)#
```

Intra-Area Distance

The **distance ospf (OSPFv2)** command configures the administrative distance for routes contained in a single OSPFv2 area. Administrative distances compare dynamic routes configured by different protocols. The default administrative distance for intra-area routes is 110.

Example

- This command configures an administrative distance of 95 for OSPFv2 intra-area routes.

```
switch(config-router-ospf)#distance ospf intra-area 95
switch(config-router-ospf)#
```

Passive Interfaces

The **passive-interface <interface> (OSPFv2)** command prevents the transmission of hello packets on the specified interface. Passive interfaces drop all adjacencies and do not form new adjacencies. Passive interfaces send LSAs but do not receive them. The router does not send or process OSPFv2 packets received on passive interfaces. The router advertises the passive interface in the router LSA.

The **no passive-interface** command re-enables OSPFv2 processing on the specified interface.

Examples

- This command configures VLAN 2 as a passive interface.

```
switch(config-router-ospf)#passive-interface vlan 2
switch(config-router-ospf)#
```

- This command configures VLAN 2 as an active interface.

```
switch(config-router-ospf)#no passive-interface vlan 2
switch(config-router-ospf)#
```

Redistributing Connected Routes

Redistributing connected routes causes the OSPFv2 instance to advertise all connected routes on the switch as external OSPFv2 routes. Connected routes are routes that are established when IPv4 is enabled on an interface.

Example

- The **redistribute (OSPFv2) connected** command converts connected routes to OSPFv2 external routes.

```
switch(config-router-ospf)#redistribute connected
switch(config-router-ospf)#
```

Redistributing Static Routes

Redistributing static routes causes the OSPFv2 instance to advertise all static routes on the switch as external OSPFv2 routes. The switch does not support redistributing individual static routes.

Example

- The **redistribute (OSPFv2)** static command converts the static routes to OSPFv2 external routes.

```
switch(config-router-ospf)#redistribute static
switch(config-router-ospf)#
```

- The **no redistribute (OSPFv2)**command stops the advertising of the static routes as OSPFv2 external routes.

```
switch(config-router-ospf)#no redistribute static
switch(config-router-ospf)#
```

Filtering Routes with Distribute Lists

An OSPF distribute list uses a route map or prefix list to filter specific routes from incoming OSPF LSAs; this filtering occurs after SPF calculation. The filtered routes are not installed on the switch, but are still included in LSAs sent by the switch. An OSPF router instance can have one distribute list configured.

If a prefix list is used, destination prefixes that do not match the prefix list will not be installed. If a route map is used, routes may be filtered based on address, next hop, or metric. OSPF external routes may also be filtered by metric type or tag.

The **distribute-list in** command specifies the filter to be used and applies it to the OSPF instance.

Example

- These commands configure a prefix-list named “dist_list1” in OSPF instance 5 to filter certain routes from incoming OSPF LSAs.

```
switch(config)#router ospf 5
switch(config-router-ospf)#distribute-list prefix-list dist_list1 in
switch(config-router-ospf)#
```

27.3.2 Configuring OSPFv2 Areas

OSPFv2 areas are configured through area commands. The switch must be in router-ospf configuration mode, as described in Section 27.3.1.1: Entering OSPFv2 Configuration Mode, to run area commands.

Areas are assigned a 32-bit number that is expressed in decimal or dotted-decimal notation. When an OSPFv2 instance configuration contains multiple areas, the switch only configures areas associated with its interfaces.

27.3.2.1 Configuring the Area Type

The **area (OSPFv2)** command specifies the area type. The switch supports three area types:

- Normal area: Area that accepts intra-area, inter-area, and external routes. The backbone area (area 0) is a normal area.
- Stub area: Area where external routes are not advertised. External routes are reached through a default summary route (0.0.0.0). Networks with no external routes do not require stub areas.
- NSSA (Not So Stubby Area): ASBRs advertise external LSAs directly connected to the area. External routes from other areas are not advertised and are reached through a default summary route.

The default area type is normal.

Examples

- This command configures area 45 as a stub area.

```
switch(config-router-ospf)#area 45 stub
switch(config-router-ospf)#
```

- This command configures area 10.92.148.17 as an NSSA.

```
switch(config-router-ospf)#area 10.92.148.17 NSSA
switch(config-router-ospf)#
```

27.3.2.2 Blocking All Summary Routes from Flooding the NSSA

The **area nssa no-summary (OSPFv2)** command configures the router to not import type-3 summary LSAs into the not-so-stubby area (NSSA) and injects a default summary route (0.0.0.0/0) into the NSSA to reach the inter-area prefixes.

Example

- This command directs the device not to import type-3 summary LSAs into the NSSA area and injects a default summary route (0.0.0.0/0) into the NSSA area.

```
switch(config)# router ospf 6
switch(config-router-ospf)# area 1.1.1.1 nssa no-summary
switch(config-router-ospf)#
```

27.3.2.3 Assigning Network Segments to the Area

Assigning Routes to an Area

The **network area (OSPFv2)** command assigns the specified network segment to an OSPFv2 area. The network can be entered in CIDR notation or by an address and wildcard mask.

The switch zeroes the host portion of the specified network address e.g. 1.2.3.4/24 converts to 1.2.3.0/24 and 1.2.3.4/16 converts to 1.2.0.0/16

Example

- Each of these equivalent commands assign the network segment 10.1.10.0/24 to area 0.

```
switch(config-router-ospf)#network 10.1.10.0 0.0.0.255 area 0
switch(config-router-ospf)#

switch(config-router-ospf)#network 10.1.10.0/24 area 0
switch(config-router-ospf)#
```

  In each case, ***running-config*** stores the command in CIDR (prefix) notation.

Summarizing Routes

By default, ABRs create a summary LSA for each route in an area and advertise them to adjacent routers. The **area range (OSPFv2)** command aggregates routing information, allowing the ABR to advertise multiple routes with one LSA. The **area range** command can be used to suppress route advertisements.

Examples

- Two **network area** commands assign subnets to an area. The **area range** command summarizes the addresses, which the ABR advertises in a single LSA.

```
switch(config-router-ospf)#network 10.1.25.80 0.0.0.240 area 5
switch(config-router-ospf)#network 10.1.25.112 0.0.0.240 area 5
switch(config-router-ospf)#area 5 range 10.1.25.64 0.0.0.192
switch(config-router-ospf)#
```

- The **network area** command assigns a subnet to an area, followed by an **area range** command that suppresses the advertisement of that subnet.

```
switch(config-router-ospf)#network 10.12.31.0 0.0.0.255 area 5
switch(config-router-ospf)#area 5 range 10.12.31.0 0.0.0.255 not-advertise
switch(config-router-ospf)#
```

27.3.2.4 Configuring Area Parameters

These router-ospf configuration mode commands define OSPFv2 behavior in a specified area.

Default Summary Route Cost

The **area default-cost (OSPFv2)** command specifies the cost of the default summary route that ABRs send into a stub area or NSSA. Summary routes, also called inter-area routes, originate in areas different than their destination.

Example

- This command configures a cost of 15 for the default summary route in area 23.

```
switch(config-router-ospf)#area 23 default-cost 15
switch(config-router-ospf)#
```

Filtering Type 3 LSAs

The **area filter (OSPFv2)** command prevents an area from receiving Type 3 (Summary) LSAs from a specified subnet. Type 3 LSAs are sent by ABRs and contain information about one of its connected areas.

Example

- This command prevents the switch from entering Type 3 LSAs originating from the 10.1.1.2/24 subnet into its area 2 LSDB.

```
switch(config-router-ospf)#area 2 filter 10.1.1.2/24
switch(config-router-ospf)#
```

27.3.3 Configuring Interfaces for OSPFv2

OSPFv2 interface configuration commands specify transmission parameters for routed ports and SVIs that handle OSPFv2 packets.

27.3.3.1 Configuring Authentication

OSPFv2 authenticates packets through passwords configured on VLAN interfaces. Interfaces connecting to the same area can authenticate packets if they have the same key. By default, OSPFv2 does not authenticate packets.

OSPFv2 supports simple password and message digest authentication:

- Simple password authentication: A password is assigned to an area. Interfaces connected to the area can authenticate packets by enabling authentication and specifying the area password.
- Message digest authentication: Each interface is configured with a key (password) and key-id pair. When transmitting a packet, the interface generates a string, using the MD5 algorithm, based on the OSPFv2 packet, key, and key ID, then appends that string to the packet.

  Message digest authentication supports uninterrupted transmissions during key changes by allowing each interface to have two keys with different key IDs. When a new key is configured on an interface, the router transmits OSPFv2 packets for both keys. Once the router detects that all neighbors are using the new key, it stops sending the old one.

Implementing authentication on an interface is a two step process:

1. Enabling authentication.
2. Configuring a key (password).

To configure simple authentication on a VLAN interface:

Step 1 Enable simple authentication with the **ip ospf authentication** command.

```
switch(config-if-vl12)#ip ospf authentication
```

Step 2 Configure the password with the **ip ospf authentication-key** command.

```
switch(config-if-vl12)#ip ospf authentication-key 0 code123
```

Running-config stores the password as an encrypted string, using a proprietary algorithm.

To configure Message-Digest authentication on a VLAN interface:

Step 1 Enable Message-Digest authentication with the **ip ospf authentication** command.

```
switch(config-if-vl12)#ip ospf authentication message-digest
```

Step 2 Configure the key ID and password with the **ip ospf message-digest-key** command.

```
switch(config-if-vl12)#ip ospf message-digest-key 23 md5 0 code123
```

Running-config stores the password as an encrypted string, using a proprietary algorithm. The key ID (23) is between keywords **message-digest-key** and **md5**.

27.3.3.2 Configuring Intervals

Interval configuration commands determine OSPFv2 packet transmission characteristics for the specified VLAN interface and are entered in interface-vlan configuration mode.

Hello Interval

The hello interval specifies the period between consecutive hello packet transmissions from an interface. Each OSPFv2 neighbor should specify the same hello interval, which should not be longer than any neighbor's dead interval.

The **ip ospf hello-interval** command configures the hello interval for the configuration mode interface. The default is 10 seconds.

Example

- This command configures a hello interval of 30 seconds for VLAN 2.

```
switch(config-if-Vl2)#ip ospf hello-interval 30
switch(config-if-Vl2)#
```

Dead Interval

The dead interval specifies the period that an interface waits for an OSPFv2 packet from a neighbor before it disables the adjacency under the assumption that the neighbor is down. The dead interval should be configured identically on all OSPFv2 neighbors and be longer than the hello interval of any neighbor.

The **ip ospf dead-interval** command configures the dead interval for the configuration mode interface. The default is 40 seconds.

Example

- This command configures a dead interval of 120 seconds for VLAN 4.

```
switch(config-if-Vl4)#ip ospf dead-interval 120
switch(config-if-Vl4)#
```

Retransmit Interval

Routers that send OSPFv2 advertisements to an adjacent router expect to receive an acknowledgment from that neighbor. Routers that do not receive an acknowledgment will retransmit the advertisement. The retransmit interval specifies the period between retransmissions.

The **ip ospf retransmit-interval** command configures the LSA retransmission interval for the configuration mode interface. The default retransmit interval is 5 seconds.

Example

- This command configures a retransmit interval of 15 seconds for VLAN 3.

```
switch(config-if-Vl3)#ip ospf retransmit-interval 15
switch(config-if-Vl3)#
```

Transmission Delay

The transmission delay is an estimate of the time that an interface requires to transmit a link-state update packet. OSPFv2 adds this delay to the age of outbound packets to more accurately reflect the age of the LSA when received by a neighbor. The default transmission delay is one second.

The **ip ospf transmit-delay** command configures the transmission delay for the configuration mode interface.

Example

- This command configures a transmission delay of 5 seconds for VLAN 6.

```
switch(config-if-Vl6)#ip ospf transmit-delay 5
switch(config-if-Vl6)#
```

27.3.3.3 Configuring Interface Parameters

Interface Cost

The OSPFv2 interface cost (or metric) reflects the overhead of sending packets across the interface. The cost is typically inversely proportional to the bandwidth of the interface. The default cost is 10.

The **ip ospf cost** command configures the OSPFv2 cost for the configuration mode interface.

Example

- This command configures a cost of 15 for VLAN 2.

```
switch(config-if-Vl2)#ip ospf cost 15
switch(config-if-Vl2)#
```

Router Priority

Router priority determines preference during designated router (DR) and backup designated router (BDR) elections. Routers with higher priority numbers have preference over other routers. Routers with a priority of zero cannot be elected as a DR or BDR.

The **ip ospf priority** command configures router priority for the configuration mode interface. The default priority is 1.

Examples

- This command configures a router priority of 15 for VLAN 8.

```
switch(config-if-Vl8)#ip ospf priority 15
switch(config-if-Vl8)#
```

- This command restores the router priority of 1 for VLAN 7.

```
switch(config-if-Vl7)#no ip ospf priority
switch(config-if-Vl7)#
```

27.3.4 Enabling OSPFv2

27.3.4.1 IPv4 Routing

OSPFv2 requires that IPv4 routing is enabled on the switch. When IP routing is not enabled, entering OSPFv2 configuration mode generates a message.

Example

- This message is displayed if, when entering router-ospf configuration mode, IP routing is not enabled.

```
switch(config)#router ospf 100
! IP routing not enabled
switch(config-router-ospf)#
```

- This command enables IP routing on the switch.

```
switch(config)#ip routing
switch(config)#
```

27.3.4.2 Disabling OSPFv2

The switch can disable OSPFv2 operations without disrupting the OSPFv2 configuration.

- **shutdown (OSPFv2)** disables all OSPFv2 activity.
- **ip ospf shutdown** disables OSPFv2 activity on a VLAN interface.

The **no shutdown** and **no ip ospf shutdown** commands resume OSPFv2 activity.

Examples

- This command disables OSPFv2 activity on the switch.

```
switch(config-router-ospf)#shutdown
switch(config-router-ospf)#
```

- This command resumes OSPFv2 activity on the switch.

```
switch(config-router-ospf)#no shutdown
switch(config-router-ospf)#
```

- This command disables OSPFv2 activity on VLAN 5.

```
switch(config-if-Vl5)#ip ospf shutdown
switch(config-if-Vl5)#
```

27.3.5 Displaying OSPFv2 Status

This section describes OSPFv2 **show** commands that display OSPFv2 status. General switch methods that provide OSPFv2 information include pinging routes, viewing route status (**show ip route** command), and viewing the configuration (**show running-config** command).

27.3.5.1 OSPFv2 Summary

The **show ip ospf** command displays general OSPFv2 configuration information and operational statistics.

Example

- This command displays general OSPFv2 information.

```
switch#show ip ospf
Routing Process "ospf 1" with ID 10.168.103.1
 Supports opaque LSA
 Maximum number of LSA allowed 12000
  Threshold for warning message 75%
  Ignore-time 5 minutes, reset-time 5 minutes
  Ignore-count allowed 5, current 0
 It is an area border router
 Hold time between two consecutive SPFs 5000 msecs
 SPF algorithm last executed 00:00:09 ago
 Minimum LSA interval 5 secs
 Minimum LSA arrival 1000 msecs
 Number of external LSA 0. Checksum Sum 0x000000
 Number of opaque AS LSA 0. Checksum Sum 0x000000
 Number of LSA 27.
 Number of areas in this router is 3. 3 normal 0 stub 0 nssa
    Area BACKBONE(0.0.0.0)
        Number of interfaces in this area is 2
        It is a normal area
        Area has no authentication
        SPF algorithm executed 153 times
        Number of LSA 8. Checksum Sum 0x03e13a
        Number of opaque link LSA 0. Checksum Sum 0x000000
    Area 0.0.0.2
        Number of interfaces in this area is 1
        It is a normal area
        Area has no authentication
        SPF algorithm executed 153 times
        Number of LSA 11. Checksum Sum 0x054e57
        Number of opaque link LSA 0. Checksum Sum 0x000000
    Area 0.0.0.3
        Number of interfaces in this area is 1
        It is a normal area
        Area has no authentication
        SPF algorithm executed 5 times
        Number of LSA 6. Checksum Sum 0x02a401
        Number of opaque link LSA 0. Checksum Sum 0x000000
```

The output lists configuration parameters and operational statistics and status for the OSPFv2 instance, followed by a brief description of the areas located on the switch.

27.3.5.2 Viewing OSPFv2 on the Interfaces

The **show ip ospf interface** command displays OSPFv2 information for switch interfaces configured for OSPFv2. Different command options allow the display of either all interfaces or a specified interface. The command can also be configured to display complete information or a brief summary.

Example

- This command displays complete OSPFv2 information for VLAN 1.

```
switch#show ip ospf interface vlan 1
Vlan1 is up, line protocol is up (connected)
  Internet Address 10.168.0.1/24, Area 0.0.0.0
  Process ID 1, Router ID 10.168.103.1, Network Type BROADCAST, Cost: 10
  Transmit Delay is 1 sec, State BDR, Priority 1
  Designated Router is 10.168.104.2
  Backup Designated router is 10.168.103.1
  Timer intervals configured, Hello 10, Dead 40, Retransmit 5
  Neighbor Count is 1
  MTU is 1500
switch#
```

  The display indicates the switch is an ABR by displaying a neighbor count, the Designated Router, and Backup Designated Router.

- This command displays a summary of interface information for the switch.

```
switch#show ip ospf interface brief
Interface    PID   Area            IP Address          Cost  State    Nbrs
Loopback0    1     0.0.0.0         10.168.103.1/24   10    DR       0
Vlan1        1     0.0.0.0         10.168.0.1/24     10    BDR      1
Vlan2        1     0.0.0.2         10.168.2.1/24     10    BDR      1
Vlan3        1     0.0.0.3         10.168.3.1/24     10    DR       0
switch#
```

  Configuration information includes the Process ID (PID), area, IP address, and cost. OSPFv2 operational information includes the Designated Router status and number of neighbors.

27.3.5.3 Viewing the OSPFv2 Database

The **show ip ospf database <link state list>** command displays the LSAs in the LSDB for the specified area. If no area is listed, the command displays the contents of the database for each area on the switch. The database command provides options to display subsets of the LSDB database, a summary of database contents, and the link states that comprise the database.

Examples

- This command displays LSDB contents for area 2.

```
switch#show ip ospf 1 2 database

            OSPF Router with ID(10.168.103.1) (Process ID 1)

                 Router Link States (Area 0.0.0.2)

Link ID         ADV Router      Age         Seq#       Checksum Link count
10.168.103.1   10.168.103.1   00:29:08    0x80000031 0x001D5F 1
10.168.104.2   10.168.104.2   00:29:09    0x80000066 0x00A49B 1

                 Net Link States (Area 0.0.0.2)

Link ID         ADV Router      Age         Seq#       Checksum
10.168.2.1      10.168.103.1    00:29:08    0x80000001 0x00B89D

                 Summary Net Link States (Area 0.0.0.2)

Link ID         ADV Router      Age         Seq#       Checksum
10.168.0.0      10.168.103.1    00:13:20    0x80000028 0x0008C8
10.168.0.0      10.168.104.2    00:09:16    0x80000054 0x00A2FF
10.168.3.0      10.168.104.2    00:24:16    0x80000004 0x00865F
10.168.3.0      10.168.103.1    00:24:20    0x80000004 0x002FC2
10.168.103.0    10.168.103.1    00:14:20    0x80000028 0x0096D2
10.168.103.0    10.168.104.2    00:13:16    0x80000004 0x00364B
10.168.104.0    10.168.104.2    00:08:16    0x80000055 0x002415
10.168.104.0    10.168.103.1    00:13:20    0x80000028 0x00EF6E
switch#
```

- This command displays an LSDB content summary for area 2.

```
switch#show ip ospf 1 2 database database-summary

            OSPF Router with ID(10.168.103.1) (Process ID 1)

Area 0.0.0.2 database summary
  LSA Type      Count
  Router        2
  Network       1
  Summary Net   8
  Summary ASBR  0
  Type-7 Ext    0
  Opaque Area   0
  Subtotal      11

Process 1 database summary
  LSA Type      Count
  Router        2
  Network       1
  Summary Net   8
  Summary ASBR  0
  Type-7 Ext    0
  Opaque Area   0
  Type-5 Ext    0
  Opaque AS     0
  Total         11
switch#
```

- This command displays the router Link States contained in the area 2 LSDB.

```
switch#show ip ospf 1 2 database router

            OSPF Router with ID(10.168.103.1) (Process ID 1)

                 Router Link States (Area 0.0.0.2)

  LS age: 00:02:16
  Options: (E DC)
  LS Type: Router Links
  Link State ID: 10.168.103.1
  Advertising Router: 10.168.103.1
  LS Seq Number: 80000032
  Checksum: 0x1B60
  Length: 36
  Number of Links: 1

    Link connected to: a Transit Network
     (Link ID) Designated Router address: 10.168.2.1
     (Link Data) Router Interface address: 10.168.2.1
      Number of TOS metrics: 0
       TOS 0 Metrics: 10


  LS age: 00:02:12
  Options: (E DC)
  LS Type: Router Links
  Link State ID: 10.168.104.2
  Advertising Router: 10.168.104.2
  LS Seq Number: 80000067
  Checksum: 0xA29C
  Length: 36
  Number of Links: 1

    Link connected to: a Transit Network
     (Link ID) Designated Router address: 10.168.2.1
     (Link Data) Router Interface address: 10.168.2.2
      Number of TOS metrics: 0
       TOS 0 Metrics: 10
switch#
```

27.3.5.4 Viewing OSPFv2 Neighbors

The **show ip ospf neighbor** command displays information about the routers that are neighbors to the switch. Command options allow the display of summary or detailed information about the neighbors for all areas and interfaces on the switch. The command also allows the display of neighbors for individual interfaces or areas. The ***adjacency-changes*** option displays the interface's adjacency changes.

Example

- This command displays the switch's neighbors.

```
switch#show ip ospf neighbor
Neighbor ID     Pri   State        Dead Time   Address         Interface
10.168.104.2      1   FULL/DR      00:00:35    10.168.0.2      Vlan1
10.168.104.2      8   FULL/BDR     00:00:31    10.168.2.2      Vlan2
switch#
```

- This command displays details about the neighbors to VLAN 2.

```
switch#show ip ospf neighbor vlan 2 detail
 Neighbor 10.168.104.2, interface address 10.168.2.2
    In the area 0.0.0.2 via interface Vlan2
    Neighbor priority is 8, State is FULL, 13 state changes
    Adjacency was established 000:01:25:48 ago
    DR is 10.168.2.1 BDR is 10.168.2.2
    Options is E
    Dead timer due in 00:00:34
switch#
```

- This command displays the adjacency changes to VLAN 2.

```
switch#show ip ospf neighbor vlan 2 adjacency-changes
[08-04 08:55:32] 10.168.104.2, interface Vlan2 adjacency established
[08-04 09:58:51] 10.168.104.2, interface Vlan2 adjacency dropped: interface went
down
[08-04 09:58:58] 10.168.104.2, interface Vlan2 adjacency established
[08-04 09:59:34] 10.168.104.2, interface Vlan2 adjacency dropped: interface went
down
[08-04 09:59:42] 10.168.104.2, interface Vlan2 adjacency established
[08-04 10:01:40] 10.168.104.2, interface Vlan2 adjacency dropped: nbr did not
list our router ID
[08-04 10:01:46] 10.168.104.2, interface Vlan2 adjacency established
switch#
```

The **show ip ospf neighbor state** command displays the state information for OSPF neighbors on a per-interface basis.

Examples

- This command displays OSPF information for neighboring routers that are fully adjacent .

```
switch>show ip ospf neighbor state full
Neighbor ID     VRF     Pri   State           Dead Time   Address         Interface
Test1           default    1   FULL/BDR        00:00:35    10.17.254.105  Vlan3912
Test2           default    1   FULL/BDR        00:00:36    10.17.254.29   Vlan3910
Test3            default    1   FULL/DR          00:00:35    10.25.0.1     Vlan101
Test4           default    1   FULL/DROTHER    00:00:36    10.17.254.67   Vlan3908
Test5           default    1   FULL/DROTHER    00:00:36    10.17.254.68   Vlan3908
Test6           default    1   FULL/BDR        00:00:32    10.17.254.66   Vlan3908
Test7           default    1   FULL/DROTHER    00:00:34    10.17.36.4     Vlan3036
Test8           default    1   FULL/BDR        00:00:35    10.17.36.3     Vlan3036
Test9           default    1   FULL/DROTHER    00:00:31    10.17.254.13   Vlan3902
Test10          default    1   FULL/BDR        00:00:37    10.17.254.11   Vlan3902
Test11          default    1   FULL/DROTHER    00:00:33    10.17.254.163  Vlan3925
Test12          default    1   FULL/DR         00:00:37    10.17.254.161  Vlan3925
Test13          default    1   FULL/DROTHER    00:00:31    10.17.254.154  Vlan3923
Test14          default    1   FULL/BDR        00:00:39    10.17.254.156  Vlan3923
Test15          default    1   FULL/DROTHER    00:00:33    10.17.254.35   Vlan3911
Test16          default    1   FULL/DR         00:00:34    10.17.254.33   Vlan3911
Test17           default    1   FULL/DR        00:00:36   10.17.254.138  Ethernet12
Test18          default    1   FULL/DR         00:00:37    10.17.254.2    Vlan3901
switch>
```

The **show ip ospf neighbor summary** command displays a single line of summary information for each OSPFv2 neighbor.

Example

- This command displays the summary information for the OSPFv2 neighbors.

```
switch>show ip ospf neighbor summary
OSPF Router with (Process ID 1) (VRF default)
0 neighbors are in state DOWN
0 neighbors are in state GRACEFUL RESTART
2 neighbors are in state INIT
0 neighbors are in state LOADING
0 neighbors are in state ATTEMPT
18 neighbors are in state FULL
0 neighbors are in state EXCHANGE
0 neighbors are in state 2 WAYS
0 neighbors are in state EXCH START
switch>
```

27.3.5.5 Viewing OSPFv2 Routes

The **show ip routes** command provides an OSPFv2 option.

Examples

- This command displays all of a switch's routes.

```
switch#show ip route
Codes: C - connected, S - static, K - kernel, O - OSPF, B - BGP

Gateway of last resort:
S    0.0.0.0/0 [1/0] via 10.255.255.1

C    10.255.255.0/24 is directly connected, Management1
C    10.168.0.0/24 is directly connected, Vlan1
C    10.168.2.0/24 is directly connected, Vlan2
O    10.168.3.0/24 [110/20] via 10.168.0.1
O    10.168.103.0/24 [110/20] via 10.168.0.1
C    10.168.104.0/24 is directly connected, Loopback0
switch#
```

- This command displays the switch's OSPFv2 routes.

```
switch#show ip route ospf
Codes: C - connected, S - static, K - kernel, O - OSPF, B - BGP

O    10.168.3.0/24 [110/20] via 10.168.0.1
O    10.168.103.0/24 [110/20] via 10.168.0.1
switch#
```

Use the **ping** command to determine the accessibility of a route.

Example

- This command pings an OSPFv2 route.

```
switch#ping 10.168.0.1
PING 10.168.0.1 (10.168.0.1) 72(100) bytes of data.
80 bytes from 10.168.0.1: icmp_seq=1 ttl=64 time=0.148 ms
80 bytes from 10.168.0.1: icmp_seq=2 ttl=64 time=0.132 ms
80 bytes from 10.168.0.1: icmp_seq=3 ttl=64 time=0.136 ms
80 bytes from 10.168.0.1: icmp_seq=4 ttl=64 time=0.137 ms
80 bytes from 10.168.0.1: icmp_seq=5 ttl=64 time=0.136 ms

--- 10.168.0.1 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 7999ms
rtt min/avg/max/mdev = 0.132/0.137/0.148/0.015 ms
switch#
```

27.3.5.6 Viewing OSPFv2 SPF Logs

The **show ip ospf spf-log** command displays when and how long the switch took to run a full SPF calculation for OSPF.

Examples

- This command displays the SPF information for OSPF.

```
switch>show ip ospf spf-log
OSPF Process 172.26.0.22
When      Duration(msec)
13:01:34  1.482
13:01:29  1.547
13:01:24  1.893
13:00:50  1.459
13:00:45  1.473
13:00:40  2.603
11:01:49  1.561
11:01:40  1.463
11:01:35  1.467
11:01:30  1.434
11:00:54  1.456
11:00:49  1.472
11:00:44  1.582
15:01:49  1.575
15:01:44  1.470
15:01:39  1.679
15:01:34  1.601
15:00:57  1.454
15:00:52  1.446
15:00:47  1.603
switch>
```

27.4 OSPFv2 Examples

This section describes the commands required to configure three OSPFv2 topologies.

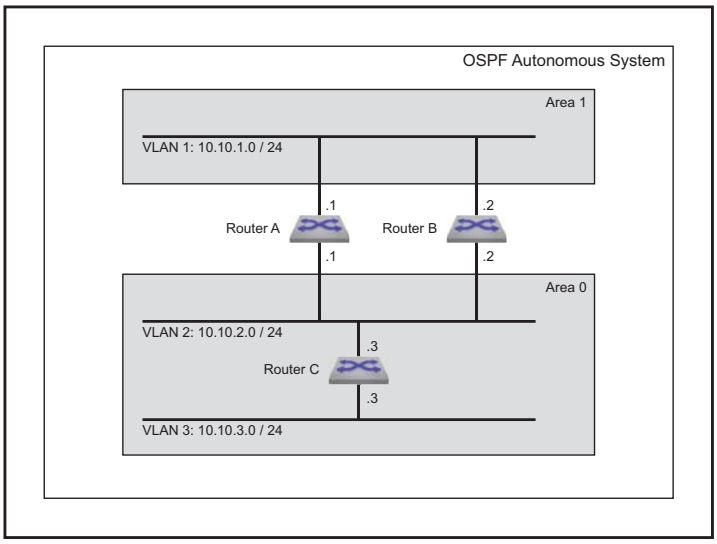
27.4.1 OSPFv2 Example 1

The AS in example 1 contains two areas that are connected through two routers. The backbone area also contains an internal router that connects two subnets.

27.4.1.1 Example 1 Diagram

Figure 27-3 displays the Example 1 topology. Two ABRs connect area 0 and area 1 – Router A and Router B. Router C is an internal router that connects two subnets in area 0.

Figure 27-3: OSPFv2 Example 1



Area 1 Configuration

Area 1 contains one subnet that is accessed by Router A and Router B.

- Router A: The subnet 10.10.1.0/24 is accessed through VLAN 1.
- Router B: The subnet 10.10.1.0/24 is accessed through VLAN 1.
- Each router uses simple authentication, with password abcdefgh.
- Designated Router (DR): Router A.
- Backup Designated Router (BDR): Router B.
- Each router defines an interface cost of 10.
- Router priority is not specified for either router on area 1.

Area 0 ABR Configuration

Area 0 contains one subnet that is accessed by ABRs Router A and Router B.

- Router A: The subnet 10.10.2.0/24 is accessed through VLAN 2.
- Router B: The subnet 10.10.2.0/24 is accessed through VLAN 2.
- Designated Router (DR): Router B.

- Backup Designated Router (BDR): Router A.
- Each router uses simple authentication, with password ijklmnop.
- Each router defines an interface cost of 20.
- Each router defines a retransmit-interval of 10.
- Each router defines a transmit-delay of 2.
- Router priority is specified such that Router B will be elected as the Designated Router.

Area 0 IR Configuration

Area 0 contains one internal router that connects two subnets.

- Router C: The subnet 10.10.2.0/24 is accessed through VLAN 2.
- Router C: The subnet 10.10.3.0/24 is accessed through VLAN 3.
- The subnet 10.10.2.0/24 link is configured as follows:
  - Interface cost of 20.
  - Retransmit-interval of 10.
  - Transmit-delay of 2.
- The subnet 10.10.3.0/24 link is configured as follows:
  - Interface cost of 20.
  - Dead interval of 80 seconds.

27.4.1.2 Example 1 Code

This code configures the OSPFv2 instances on the three switches.

Step 1 Configure the interface addresses.

a Router A interfaces:

```
switch-A(config)#interface vlan 1
switch-A(config-if-vl1)#ip address 10.10.1.1/24
switch-A(config-if-vl1)#interface vlan 2
switch-A(config-if-vl2)#ip address 10.10.2.1/24
```

b Router B interfaces:

```
switch-B(config)#interface vlan 1
switch-B(config-if-vl1)#ip address 10.10.1.2/24
switch-B(config-if-vl1)#interface vlan 2
switch-B(config-if-vl2)#ip address 10.10.2.2/24
```

c Router C interfaces:

```
switch-C(config)#interface vlan 2
switch-C(config-if-vl2)#ip address 10.10.2.3/24
switch-C(config-if-vl2)#interface vlan 3
switch-C(config-if-vl3)#ip address 10.10.3.3/24
```

Step 2 Configure the interface OSPFv2 parameters.

a Router A interfaces:

```
switch-A(config-if-vl2)#interface vlan 1
switch-A(config-if-vl1)#ip ospf authentication-key abcdefgh
switch-A(config-if-vl1)#ip ospf cost 10
switch-A(config-if-vl1)#ip ospf priority 6
switch-A(config-if-vl1)#interface vlan 2
switch-A(config-if-vl2)#ip ospf authentication-key ijklmnop
switch-A(config-if-vl2)#ip ospf cost 20
switch-A(config-if-vl2)#ip ospf retransmit-interval 10
switch-A(config-if-vl2)#ip ospf transmit-delay 2
switch-A(config-if-vl2)#ip ospf priority 4
```

b Router B interfaces:

```
switch-B(config-if-vl2)#interface vlan 1
switch-B(config-if-vl1)#ip ospf authentication-key abcdefgh
switch-B(config-if-vl1)#ip ospf cost 10
switch-B(config-if-vl1)#ip ospf priority 4
switch-B(config-if-vl1)#interface vlan 2
switch-B(config-if-vl2)#ip ospf authentication-key ijklmnop
switch-B(config-if-vl2)#ip ospf cost 20
switch-B(config-if-vl2)#ip ospf retransmit-interval 10
switch-B(config-if-vl2)#ip ospf transmit-delay 2
switch-B(config-if-vl2)#ip ospf priority 6
```

c Router C interfaces:

```
switch-C(config-if-vl3)#interface vlan 2
switch-C(config-if-vl2)#ip ospf cost 20
switch-C(config-if-vl2)#ip ospf retransmit-interval 10
switch-C(config-if-vl2)#ip ospf transmit-delay 2
switch-C(config-if-vl2)#interface vlan 3
switch-C(config-if-vl3)#ip ospf cost 20
switch-C(config-if-vl3)#ip ospf dead-interval 80
```

Step 3 Attach the network segments to the areas.

a Router A interfaces:

```
switch-A(config-if-vl2)#router ospf 1
switch-A(config-router-ospf)#router-id 169.10.0.1
switch-A(config-router-ospf)#network 10.10.1.0/24 area 1
switch-A(config-router-ospf)#network 10.10.2.0/24 area 0
```

b Router B interfaces:

```
switch-B(config-if-vl2)#router ospf 1
switch-B(config-router-ospf)#router-id 169.10.0.2
switch-B(config-router-ospf)#network 10.10.1.0/24 area 1
switch-B(config-router-ospf)#network 10.10.2.0/24 area 0
```

c Router C interfaces:

```
switch-C(config-if-vl3)#router ospf 1
switch-C(config-router-ospf)#router-id 169.10.0.3
switch-C(config-router-ospf)#network 10.10.2.0/24 area 0
switch-C(config-router-ospf)#network 10.10.3.0/24 area 0
```

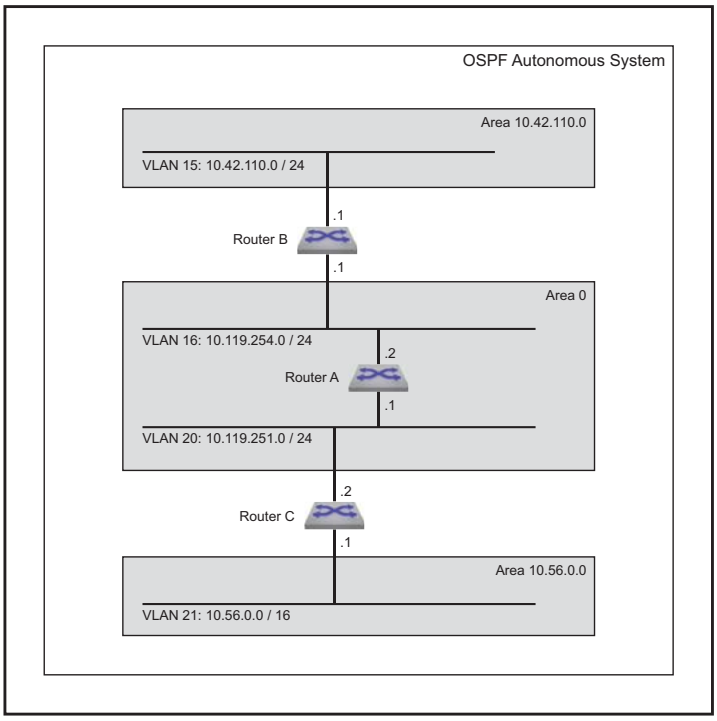
27.4.2 OSPFv2 Example 2

The AS in example 2 contains three areas. Area 0 connects to the other areas through different routers. The backbone area contains an internal router that connects two subnets. Area 0 is normal; the other areas are stub areas.

27.4.2.1 Example 2 Diagram

Figure 27-4 displays the Example 2 topology. One ABR (Router B) connects area 0 and area 10.42.110.0; another ABR (router C) connects area 0 and area 36.56.0.0. Router A is an internal router that connects two subnets in area 0.

Figure 27-4: OSPFv2 Example 2



Area 10.42.110.0 Configuration

Area 10.42.110.0 contains one subnet that is accessed by Router B.

- Router B: The subnet 10.42.110.0 is accessed through VLAN 15.
- Router B uses simple authentication, with password abcdefgh.
- Each router defines a interface cost of 10.

Area 10.56.0.0 Configuration

Area 10.56.0.0 contains one subnet that is accessed by Router C.

- Router C: The subnet 10.56.0.0 is accessed through VLAN 21.
- Router C uses simple authentication, with password ijklmnop.
- Each router defines a interface cost of 20.

Area 0 ABR Configuration

Area 0 contains two subnets. ABR Router B connects one subnet to area 10.42.110.0. ABR Router C connects the other subnet to area 10.56.0.0.

- Router B: The subnet 10.119.254.0/24 is accessed through VLAN 16.
- Router C: The subnet 10.119.251.0/24 is accessed through VLAN 20.

- Designated Router (DR): Router B.
- Backup Designated Router (BDR): Router C.
- Each ABR uses simple authentication, with password ijklmnop
- Each router defines an interface cost of 20.
- Each router defines a retransmit-interval of 10.
- Each router defines a transmit-delay of 2.

Area 0 IR Configuration

Area 0 contains two subnets connected by an internal router.

- Router A: The subnet 10.119.254.0/24 is accessed through VLAN 16.
- Router A: The subnet 10.119.251.0/24 is accessed through VLAN 20.
- The subnet 10.42.110.0 is configured as follows:
  - Interface cost of 10.
- The subnet 10.56.0.0/24 is configured as follows:
  - Interface cost of 20.
  - Retransmit-interval of 10.
  - Transmit-delay of 2.

27.4.2.2 Example 2 Code

Step 1 Configure the interface addresses.

a Router A interfaces:

```
switch-A(config)#interface vlan 16
switch-A(config-if-vl16)#ip address 10.119.254.2/24
switch-A(config-if-vl16)#interface vlan 20
switch-A(config-if-vl20)#ip address 10.119.251.1/24
```

b Router B interfaces:

```
switch-B(config)#interface vlan 15
switch-B(config-if-vl15)#ip address 10.42.110.1/24
switch-B(config-if-vl15)#interface vlan 16
switch-B(config-if-vl16)#ip address 10.119.254.1/24
```

c Router C interfaces:

```
switch-C(config)#interface vlan 20
switch-C(config-if-vl20)#ip address 10.119.251.2/24
switch-C(config-if-vl20)#interface vlan 21
switch-C(config-if-vl21)#ip address 10.56.0.1/24
```

Step 2 Configure the interface OSPFv2 parameters.

a Router A interfaces:

```
switch-A(config-if-vl20)#interface vlan 16
switch-A(config-if-vl16)#ip ospf cost 10
switch-A(config-if-vl16)#interface vlan 20
switch-A(config-if-vl20)#ip ospf cost 20
switch-A(config-if-vl20)#ip ospf retransmit-interval 10
switch-A(config-if-vl20)#ip ospf transmit-delay 2
```

b Router B interfaces:

```
switch-B(config-if-vl16)#interface vlan 15
switch-B(config-if-vl15)#ip ospf authentication-key abcdefgh
switch-B(config-if-vl15)#ip ospf cost 10
switch-B(config-if-vl15)#interface vlan 16
switch-B(config-if-vl16)#ip ospf authentication-key ijklmnop
switch-B(config-if-vl16)#ip ospf cost 20
switch-B(config-if-vl16)#ip ospf retransmit-interval 10
switch-B(config-if-vl16)#ip ospf transmit-delay 2
switch-B(config-if-vl16)#ip ospf priority 6
```

c Router C interfaces:

```
switch-C(config-if-vl21)#interface vlan 20
switch-C(config-if-vl20)#ip ospf authentication-key ijklmnop
switch-C(config-if-vl20)#ip ospf cost 20
switch-C(config-if-vl20)#ip ospf retransmit-interval 10
switch-C(config-if-vl20)#ip ospf transmit-delay 2
switch-C(config-if-vl20)#ip ospf priority 4
switch-C(config-if-vl20)#interface vlan 21
switch-C(config-if-vl21)#ip ospf authentication-key ijklmnop
switch-C(config-if-vl21)#ip ospf cost 20
switch-C(config-if-vl21)#ip ospf dead-interval 80
```

Step 3 Attach the network segments to the areas.

a Router A interfaces:

```
switch-A(config-if-vl20)#router ospf 1
switch-A(config-router-ospf)#router-id 10.24.1.1
switch-A(config-router-ospf)#network 10.119.254.0/24 area 0
switch-A(config-router-ospf)#network 10.119.251.0/24 area 0
switch-A(config-router-ospf)#area 0 range 10.119.251.0 0.0.7.255
```

b Router B interfaces:

```
switch-B(config-if-vl16)#router ospf 1
switch-B(config-router-ospf)#router-id 10.24.1.2
switch-B(config-router-ospf)#area 10.42.110.0 stub
switch-B(config-router-ospf)#network 10.42.110.0/24 area 10.42.110.0
switch-B(config-router-ospf)#network 10.119.254.0/24 area 0
```

c Router C interfaces:

```
switch-C(config-if-vl21)#router ospf 1
switch-C(config-router-ospf)#router-id 10.24.1.3
switch-C(config-router-ospf)area 10.56.0.0 stub 0
switch-C(config-router-ospf)#network 10.119.251.0/24 area 0
switch-C(config-router-ospf)#network 10.56.0.0/24 area 36.56.0.0
```

27.4.3 OSPFv2 Example 3

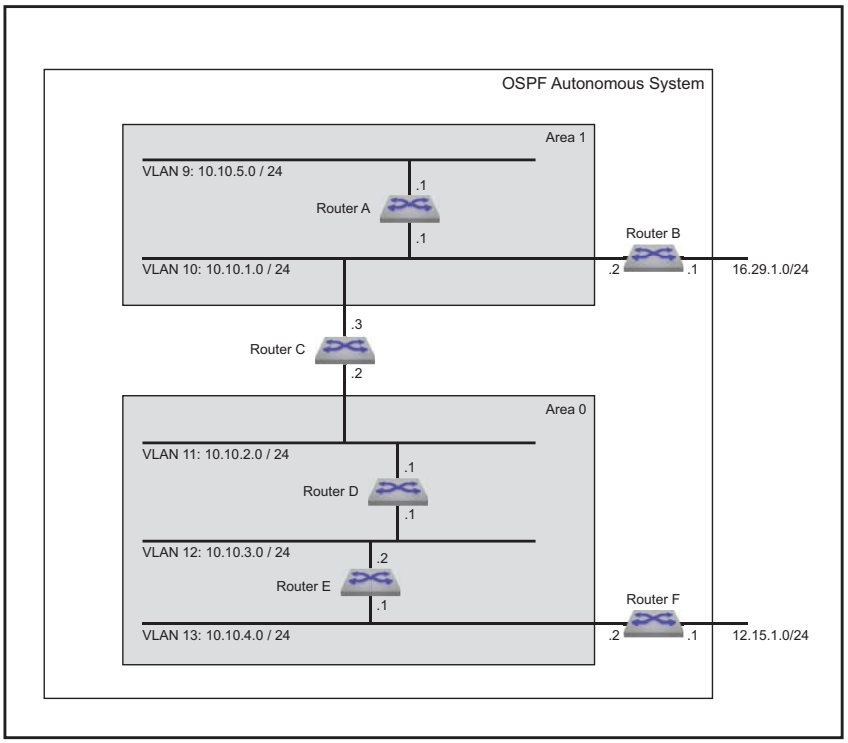
The AS in example 3 contains two areas that connect through one ABR.

- Area 0: Backbone area contains two internal routers that connect three subnets, one ASBR, and one ABR that connects to Area 1.
- Area 1: NSSA contains one internal router, one ASBR, and one ABR that connects to the backbone.

27.4.3.1 Example 3 Diagram

Figure 27-5 displays the Example 3 topology. One ABR connects area 0 and area 1. Router C is an ABR that connects the areas. Router A is an internal router that connects two subnets in area 1. Router D and Router E are internal routers that connect subnets in area 0. Router B and Router F are ASBRs that connect static routes outside the AS to area 1 and area 0, respectively.

Figure 27-5: OSPFv2 Example 3



Area 0 ABR Configuration

ABR Router C connects one area 0 subnet to an area 1 subnet.

- Router C: The subnet 10.10.2.0/24 is accessed through VLAN 11.
- Authentication is not configured on the interfaces.
- All interface OSPFv2 parameters are set to their default values.

Area 0 IR Configuration

Area 0 contains two internal routers, each of which connects two of the three subnets in the area.

- Router D: The subnet 10.10.2.0/24 is accessed through VLAN 11.
- Router D: The subnet 10.10.3.0/24 is accessed through VLAN 12.
- Router E: The subnet 10.10.3.0/24 is accessed through VLAN 12.
- Router E: The subnet 10.10.4.0/24 is accessed through VLAN 13.
- All interface OSPFv2 parameters are set to their default values.

Area 0 ASBR Configuration

ASBR Router F connects one area 0 subnet to an external subnet.

- Router F: The subnet 10.10.4.0/24 is accessed through VLAN 13.
- Router F: The subnet 12.15.1.0/24 is accessed through VLAN 14.
- All interface OSPFv2 parameters are set to their default values.

Area 1 ABR Configuration

ABR Router C connects one area 0 subnet to area 1.

- Router C: The subnet 10.10.1.0/24 is accessed through VLAN 10.
- Authentication is not configured on the interface.
- All interface OSPFv2 parameters are set to their default values.

Area 1 IR Configuration

Area 1 contains one internal router that connects two subnets in the area.

- Router A: The subnet 10.10.1.0/24 is accessed through VLAN 10.
- Router A: The subnet 10.10.5.0/24 is accessed through VLAN 9.
- All interface OSPFv2 parameters are set to their default values.

Area 1 ASBR Configuration

ASBR Router B connects one area 1 subnet to an external subnet.

- Router B: The subnet 10.10.1.0/24 is accessed through VLAN 10.
- Router B: The subnet 16.29.1.0/24 is accessed through VLAN 15.
- All interface OSPFv2 parameters are set to their default values.

27.4.3.2 Example 3 Code

Step 1 Configure the interfaces.

a Router A interfaces:

```
switch-A(config)#interface vlan 10
switch-A(config-if-vl10)#ip address 10.10.1.1/24
switch-A(config-if-vl10)#interface vlan 9
switch-A(config-if-vl11)#ip address 10.10.5.1/24
```

b Router B interfaces:

```
switch-B(config)#interface vlan 10
switch-B(config-if-vl10)#ip address 10.10.1.2/24
switch-B(config-if-vl10)#interface vlan 15
switch-B(config-if-vl18)#ip address 16.29.1.1/24
```

c Router C interfaces:

```
switch-C(config)#interface vlan 10
switch-C(config-if-vl10)#ip address 10.10.1.3/24
switch-C(config-if-vl10)#interface vlan 11
switch-C(config-if-vl11)#ip address 10.10.2.2/24
```

d Router D interfaces:

```
switch-D(config)#interface vlan 11
switch-D(config-if-vl11)#ip address 10.10.2.1/24
switch-D(config)#interface vlan 12
switch-D(config-if-vl12)#ip address 10.10.3.1/24
```

e Router E interfaces:

```
switch-E(config)#interface vlan 12
switch-E(config-if-vl12)#ip address 10.10.3.2/24
switch-E(config)#interface vlan 13
switch-E(config-if-vl13)#ip address 10.10.4.1/24
```

f Router F interfaces:

```
switch-F(config)#interface vlan 13
switch-F(config-if-vl13)#ip address 10.10.4.2/24
switch-F(config)#interface vlan 14
switch-F(config-if-vl14)#ip address 12.15.1.1/24
```

Step 2 Attach the network segments to the areas.

a Router A interfaces:

```
switch-A(config-if-vl10)#router ospf 1
switch-A(config-router-ospf)#router-id 170.21.0.1
switch-A(config-router-ospf)#area 1 NSSA
switch-A(config-router-ospf)#network 10.10.1.0/24 area 1
```

b Router B interfaces:

```
switch-B(config-if-vl10)#router ospf 1
switch-B(config-router-ospf)#router-id 170.21.0.2
switch-B(config-router-ospf)#area 1 NSSA
switch-B(config-router-ospf)#network 10.10.1.0/24 area 1
```

c Router C interfaces:

```
switch-C(config-if-vl11)#router ospf 1
switch-C(config-router-ospf)#router-id 170.21.0.3
switch-C(config-router-ospf)#area 1 NSSA
switch-C(config-router-ospf)#network 10.10.1.0/24 area 1
switch-C(config-router-ospf)#network 10.10.2.0/24 area 0
```

d Router D interfaces:

```
switch-D(config-if-vl12)#router ospf 1
switch-D(config-router-ospf)#router-id 170.21.0.4
switch-D(config-router-ospf)#network 10.10.2.0/24 area 0
switch-D(config-router-ospf)#network 10.10.3.0/24 area 0
```

e Router E interfaces:

```
switch-E(config-if-vl13)#router ospf 1
switch-E(config-router-ospf)#router-id 170.21.0.5
switch-E(config-router-ospf)#network 10.10.3.0/24 area 0
switch-E(config-router-ospf)#network 10.10.4.0/24 area 0
```

f Router F interfaces:

```
switch-F(config-if-vl14)#router ospf 1
switch-F(config-router-ospf)#router-id 170.21.0.6
switch-F(config-router-ospf)#network 10.10.4.0/24 area 0

switch-F(config-router-ospf)#redistribute static
```

27.5 OSPFv2 Commands

Global Configuration Mode

- ip ospf name-lookup
- router ospf

Interface Configuration Mode

- ip ospf authentication
- ip ospf authentication-key
- ip ospf cost
- ip ospf dead-interval
- ip ospf hello-interval
- ip ospf message-digest-key
- ip ospf network point-to-point
- ip ospf priority
- ip ospf retransmit-interval
- ip ospf shutdown
- ip ospf transmit-delay

Router-OSPFv2 Configuration Mode

- adjacency exchange-start threshold (OSPFv2)
- area default-cost (OSPFv2)
- area filter (OSPFv2)
- area nssa (OSPFv2)
- area nssa default-information-originate (OSPFv2)
- area nssa no-summary (OSPFv2)
- area nssa translate type7 always (OSPFv2)
- area range (OSPFv2)
- area stub (OSPFv2)
- auto-cost reference-bandwidth (OSPFv2)
- compatible (OSPFv2)
- default-information originate (OSPFv2)
- distance ospf (OSPFv2)
- log-adjacency-changes (OSPFv2)
- max-lsa (OSPFv2)
- max-metric router-lsa (OSPFv2)
- maximum-paths (OSPFv2)
- network area (OSPFv2)
- no area (OSPFv2)
- passive-interface default (OSPFv2)
- passive-interface <interface> (OSPFv2)
- point-to-point routes (OSPFv2)
- redistribute (OSPFv2)
- router-id (OSPFv2)
- shutdown (OSPFv2)
- timers lsa arrival (OSPFv2)
- timers throttle lsa all (OSPFv2)
- timers throttle spf (OSPFv2)

Display and Clear Commands

- clear ip ospf neighbor

- show ip ospf
- show ip ospf border-routers
- show ip ospf database database-summary
- show ip ospf database <link state list>
- show ip ospf database <link-state details>
- show ip ospf interface
- show ip ospf interface brief
- show ip ospf lsa-log
- show ip ospf neighbor
- show ip ospf neighbor adjacency-changes
- show ip ospf neighbor state
- show ip ospf neighbor summary
- show ip ospf request-list
- show ip ospf retransmission-list
- show ip ospf spf-log

adjacency exchange-start threshold (OSPFv2)

The a**djacency exchange-start threshold** command sets the exchange-start options for an OSPF instance.

The **no adjacency exchange-start threshold** and **default adjacency exchange-start threshold** command resets the default by removing the corresponding a**djacency exchange-start threshold** command from *running-config.*

Command Mode

Router-OSPF Configuration

Command Syntax

```
adjacency exchange-start threshold peers
no adjacency exchange-start threshold
default adjacency exchange-start threshold
```

Parameters

- *peers* Value ranges from 1 4294967295. Default value is 10.

Example

- This command sets the adjacency exchange start threshold to 20045623.

```
switch(config)#router ospf 6
switch(config-router-ospf)#adjacency exchange-start threshold 20045623
switch(config-router-ospf)#
```

area default-cost (OSPFv2)

The **area default-cost** command specifies the cost for the default summary routes sent into a specified area. The default-cost is set to 10.

The **no area default-cost** and **default area default-cost** command resets the default-cost value of the specified area to 10 by removing the corresponding **area default-cost** command from ***running-config***. The **no area (OSPFv2)** command removes all area commands for the specified area from ***running-config***, including the **area default-cost** command.

Command Mode

Router-OSPF Configuration

Command Syntax

```
area area_id default-cost def_cost
no area area_id default-cost
default area area_id default-cost
```

Parameters

- *area_id* area number. <**0** to **4294967295**> or <**0.0.0.0** to **255.255.255.255**>
  running-config stores value in dotted decimal notation.
- *def_cost* Value ranges from 1 to 65535. Default value is 10.

Example

- This command configures a cost of 15 for default summary routes that an ABR sends into area 23.

```
switch(config)#router ospf 6
switch(config-router-ospf)#area 23 default-cost 15
switch(config-router-ospf)#
```

area filter (OSPFv2)

The **area filter** command prevents an area from receiving Type 3 Summary LSAs from a specified subnet.

The **no area filter** and **default area filter** commands remove the specified **area filter** command from ***running-config***. The **no area** command (see **no area (OSPFv2)** removes all area commands for the specified area from ***running-config***, including **area filter** commands.

Command Mode

Router-OSPF Configuration

Command Syntax

```
area area_id filter net_addr
no area area_id filter net_addr
default area area_id filter net_addr
```

Parameters

- *area_id* area number. <**0** to **4294967295**> or <**0.0.0.0** to **255.255.255.255**>
  running-config stores value in dotted decimal notation.
- *net_addr* network IP address. Entry formats include address-prefix (CIDR) and address-mask.
  running-config stores value in CIDR notation.

Example

- This command prevents the switch from entering Type 3 LSAs originating from the 10.1.1.0/24 subnet into its area 2 LSDB.

```
switch(config)#router ospf 6
switch(config-router-ospf)#area 2 filter 10.1.1.0/24
switch(config-router-ospf)#
```

area nssa (OSPFv2)

The **area nssa** command configures an OSPFv2 area as a not-so-stubby area (NSSA). All routers in an AS must specify the same area type for identically numbered areas.

NSSA ASBRs advertise external LSAs that are part of the area, but do not advertise external LSAs from other areas.

Areas are ***normal*** by default; area type configuration is required only for stub NSSA areas. Area 0 is always a normal area and cannot be configured through this command.

The **no area nssa** command configures the specified area as a normal area by removing the specified **area nssa** command from ***running-config***.

Command Mode

Router-OSPF Configuration

Command Syntax

```
area area_id nssa [TYPE]
no area area_id nssa [TYPE]
default area area_id nssa [TYPE]
```

All parameters except *area_id* can be placed in any order.

Parameters

- *area_id*
  - Valid formats: integer <**1** to **4294967295**> or dotted decimal <**0.0.0.1** to **255.255.255.255**>
  - Area 0 (or 0.0.0.0) is not configurable; it is always ***normal***.
  - ***running-config*** stores value in dotted decimal notation.
- ***TYPE*** area type. Values include:
  - <no parameter>
  - **nssa-only**

Example

- This command configures area 3 as a NSSA area.

```
switch(config-router-ospf)#area 3 nssa nssa-only
switch(config-router-ospf)#
```

area nssa default-information-originate (OSPFv2)

The **default area nssa default-information-originate** command sets default route origination for the NSSA, allowing the redistribute policy to advertise a default route if one is present. The resulting OSPF behavior depends on the presence of an installed static default route and on whether static routes are redistributed in OSPF (using the **redistribute (OSPFv2)** command). The **no area nssa default-information-originate** command disables advertisement of the default route for the NSSA regardless of the redistribute policy. See Table 27-1 for details.

Areas are ***normal*** by default; area type configuration is required only for stub and NSSA areas. Area 0 is always a normal area and cannot be configured through this command.

Default route origination is configured differently for different area types and supports three area types:

- Normal areas: advertisement of the default route is configured for all normal areas using the **default-information originate (OSPFv2)** command.
- Stub areas: the default route is automatically advertised in stub areas and cannot be configured.
- Not So Stubby Areas (NSSAs): advertisement of the default route is configured per area using the **area nssa default-information-originate (OSPFv2)** or **area nssa no-summary (OSPFv2)** command.

Table 27-1 Advertisement of Default Route

Static Default Route Installed	Redistribute Static	Command Form	Advertise in ABR	Advertise in ASBR
no	no	**default** or **no**	no	no
no	no	standard	yes	no
no	yes	**default**	yes	yes
no	yes	**no**	no	no
no	yes	standard	yes	no
yes	no	**default** or **no**	no	no
yes	no	standard	yes	yes
yes	yes	**default**	yes	yes
yes	yes	**no**	no	no
yes	yes	standard	yes	yes

Command Mode

Router-OSPF Configuration

Command Syntax

```
area area_id nssa default-information-originate [VALUE][TYPE][EXCL]
no area area_id nssa default-information-originate
default area area_id nssa default-information-originate
```

All parameters except *area_id* can be placed in any order.

Parameters

- *area_id*
  - Valid formats: integer <**1** to **4294967295**> or dotted decimal <**0.0.0.1** to **255.255.255.255**>
  - Area 0 (or 0.0.0.0) is not configurable; it is always ***normal***.
  - ***running-config*** stores value in dotted decimal notation.

- ***VALUE*** Values include:
  - <no parameter> Default value of 1.
  - **metric** *<1-65535>*
- ***TYPE*** Values include:
  - <no parameter>
  - **metric-type** *<1-2>*
- ***EXCL*** Values include:
  - <no parameter>.
  - **nssa-only**

Example

- This command configures area 3 as an NSSA and generates a type 7 default LSA within the NSSA.

```
switch(config-router-ospf)#area 3 nssa default-information-originate nssa-only
switch(config-router-ospf)#
```

area nssa no-summary (OSPFv2)

The **area nssa no-summary** command configures the switch stop importing type-3 summary LSAs into the not-so-stubby area and sets the default summary route into the NSSA in order to reach the inter-area prefixes.

The **no area nssa no-summary** and **default area** `nssa no-summary` commands allow type-3 summary LSAs into the NSSA area.

The **no area nssa** and **default area nssa** commands configure the specified area as a normal area.

Command Mode

Router-OSPF Configuration

Command Syntax

```
area area_id nssa no-summary
no area area_id nssa no-summary
default area area_id nssa no-summary
```

Parameters

- *area_id* area number.
  - Valid formats: integer <**1** to **4294967295**> or dotted decimal <**0.0.0.1** to **255.255.255.255**>
  - Area 0 (or 0.0.0.0) is not configurable; it is always ***normal***.
  - ***running-config*** stores value in dotted decimal notation.

Example

- This command directs the device not to import type-3 summary LSAs into the NSSA area

```
switch(config)# router ospf 6
switch(config-router-ospf)# area 1.1.1.1 nssa no-summary
switch(config-router-ospf)#
```

- This command directs the device to import type-3 summary LSAs into the NSSA area.

```
switch(config)# router ospf 6
switch(config-router-ospf)# no area 1.1.1.1 nssa no-summary
switch(config-router-ospf)#
```

area nssa translate type7 always (OSPFv2)

The **area nssa translate type7 always** command configures the switch to always translate Type-7 link-state advertisement (LSAs) to Type-5 LSAs.

The **no area nssa translate type7 always** and **no area nssa translate type7 always** commands allow LSAs to be translated dynamically by removing the **no area nssa translate type7 always** command from ***running-config***.

Command Mode

Router-OSPF Configuration

Command Syntax

```
area area_id nssa translate type7 always
no area_id nssa translate type7 always
default area_id nssa translate type7 always
```

Parameters

- *area_id* area number.
  - Valid formats: integer <**1** to **4294967295**> or dotted decimal <**0.0.0.1** to **255.255.255.255**>
  - Area 0 (or 0.0.0.0) is not configurable; it is always ***normal***.
  - ***running-config*** stores value in dotted decimal notation.

Example

- This command configures the switch to always translate Type-7 link-state advertisement (LSAs) to Type-5 LSAs.

```
switch(config-router-ospf)#area 3 nssa translate type7 always
switch(config-router-ospf)#
```

area range (OSPFv2)

The **area range** command configures OSPF area border routers (ABRs) to consolidate or summarize routes, to set the cost setting routes, and to suppress summary route advertisements.

The **no area (OSPFv2)** command removes all area commands for the specified area from ***running-config***.

Command Mode

Router-OSPF Configuration

Command Syntax

```
area area_id range net_addr [ADVERTISE_SETTING][COST_SETTING]
no area area_id range net_addr [ADVERTISE_SETTING][COST_SETTING]
default area area_id range net_addr [ADVERTISE_SETTING][COST_SETTING]
```

Parameters

- *area_id* area number. <**0** to **4294967295**> or <**0.0.0.0** to **255.255.255.255**> ***running-config*** stores value in dotted decimal notation.
- *net_addr*
- ***ADVERTISE_SETTING*** Values include
  - <no parameter>
  - **advertise**
  - **not-advertise**
- ***COST_SETTING*** Values include
  - <no parameter>
  - **cost** *range_cost* Value ranges from 1 to 65535.

Examples

- The **network area** commands assign two subnets to an area. The **area range** command summarizes the addresses, which the ABR advertises in a single LSA.

```
switch(config)#router ospf 6
switch(config-router-ospf)#network 10.1.25.80 0.0.0.240 area 5
switch(config-router-ospf)#network 10.1.25.112 0.0.0.240 area 5
switch(config-router-ospf)#area 5 range 10.1.25.64 0.0.0.192
switch(config-router-ospf)#
```

- The **network area** command assigns a subnet to an area, followed by an **area range** command that suppresses the advertisement of that subnet.

```
switch(config-router-ospf)#network 10.12.31.0/24 area 5
switch(config-router-ospf)#area 5 range 10.12.31.0/24 not-advertise
switch(config-router-ospf)#
```

area stub (OSPFv2)

The **area stub** command sets the area type of an OSPF area to ***stub***. All devices in an AS must specify the same area type for identically numbered areas.

The **no area stub** command remove the specified stub area from the OSPFv2 instance by deleting all **area stub** commands from ***running-config*** for the specified area.

The **no area stub command** configure the specified area as a normal area.

Command Mode

Router-OSPF Configuration

Command Syntax

```
area area_id stub [summarize]
no area area_id stub [summarize]
default area area_id stub [summarize]
```

Parameters

- *area_id* area number.
  - Valid formats: integer <**1** to **4294967295**> or dotted decimal <**0.0.0.1** to **255.255.255.255**>
  - Area 0 (or 0.0.0.0) is not configurable; it is always ***normal***.
  - ***running-config*** stores value in dotted decimal notation.
- ***SUMMARIZE*** area type. Values include:
  - <no parameter>
  - **no-summary**

Examples

- These commands configure area 45 as a stub area.

```
switch(config)#router ospf 3
switch(config-router-ospf)#area 45 stub
switch(config-router-ospf)#
```

- This command configures area 10.92.148.17 as a stub area.

```
switch(config-router-ospf)#area 10.92.148.17 stub
switch(config-router-ospf)#
```

auto-cost reference-bandwidth (OSPFv2)

The **auto-cost reference-bandwidth** command is a factor in the formula that calculates the default OSPFv2 cost for Ethernet interfaces.

OSPFv2-cost = (auto-cost value * 1 Mbps) / interface bandwidth

The switch uses a minimum OSPFv2-cost of one. The switch rounds down all non-integer results.

On a 10G Ethernet interface:

- if auto-cost = 100, then OSPFv2-cost = 100 Mbps / 10 Gbps = 0.01, and the default cost is set to 1.
- if auto-cost = 59000, then OSPFv2-cost = 59000 Mbps / 10 Gbps = 5.9, and the default cost is set to 5.

The **no auto-cost reference-bandwidth** and **default auto-cost reference-bandwidth** command removes the **auto-cost reference-bandwidth** command from ***running-config***. When this parameter is not set, the default cost for Ethernet interfaces is the default **ip ospf cost** value of 10.

Command Mode

Router-OSPF Configuration

Command Syntax

```
auto-cost reference-bandwidth rate
no auto-cost reference-bandwidth
default auto-cost reference-bandwidth
```

Parameters

- *rate* Values range from 1 to 4294967 . Default is 100.

Example

To configure a default cost of 20 on 10G Ethernet interfaces:

Step 1 calculate the required auto-cost value:

auto-cost = (OSPFv2-cost*interface bandwidth)/1 Mbps = (20*10000 Mbps) / 1 Mbps = 200000

Step 2 Configure this value as the auto-cost reference-bandwidth.

```
switch(config)#router ospf 6
switch(config-router-ospf)#auto-cost reference-bandwidth 200000
switch(config-router-ospf)#
```

clear ip ospf neighbor

The **clear ip ospf** command clears the neighbors statistics per interface.

Command Mode

Privileged EXEC

Command Syntax

clear ip ospf [*PROCESS_ID*] neighbor[*LOCATION*] [*VRF_INSTANCE*]

Parameters

- ***PROCESS_ID*** OSPFv2 process ID. Values include:
  - <no parameter>
  - <***1*** to ***65535***>
- ***LOCATION*** IP address or interface peer group name. Values include:
  - ***** clears all OSPF IPv4 neighbors.
  - *ipv4_addr*
  - **ethernet** *e_num*
  - **loopback** *l_num*
  - **port-channel** *p_num*
  - **vlan** *v_num*
- ***VRF_INSTANCE*** specifies the VRF instance.
  - **vrf** *vrf_name* configures the vrf_name instance.

Examples

- This command resets all OSPF neighbor statistics.

```
switch#clear ip ospf neighbor *
switch#
```

- This command resets the OSPF neighbor statistics for the specified Ethernet 3 interface.

```
switch#clear ip ospf neighbor ethernet 3
switch##
```

compatible (OSPFv2)

The **compatible** command allows the selective disabling of compatibility with RFC 2328.

The **no compatible** and **default compatible** commands reverts OSPF to RFC 2328 compatible and removes the **compatible** statement from ***running-config***.

Command Mode

Router-OSPF Configuration

Command Syntax

```
compatible rfc1583
no compatible rfc1583
default compatible rfc1583
```

Example

- This command sets the OSPF compatibility list with RFC 1583.

```
switch(config)#router ospf 6
switch(config-router-ospf)#compatible rfc1583
switch(config-router-ospf)#
```

- This command disables RFC 1583 compatibility.

```
switch(config)#router ospf 6
switch(config-router-ospf)# no compatible rfc1583
switch(config-router-ospf)#
```

default-information originate (OSPFv2)

The **default-information originate** command enables default route origination for normal areas. The user user may configure the metric value and metric type used in LSAs. The ***always*** option will cause the ASBR to create and advertise a default route whether or not one is configured.

The **no default-information originate** command prevents the advertisement of the default route, **.** The **default default-information originate** command enables default route origination with default values (metric type 2, metric=1).

Command Mode

Router-OSPF Configuration

Command Syntax

```
default-information originate [FORCE][VALUE][TYPE][MAP]
no default-information originate
default default-information originate
```

All parameters can be placed in any order.

Parameters

- ***FORCE*** advertisement forcing option. Values include:
  - <no parameter>
  - **always**
- ***VALUE*** Values include:
  - <no parameter>
  - **metric** *<1-65535>*
- ***TYPE*** Values include:
  - <no parameter>
  - **metric-type** *<1-2>*
- ***MAP*** sets attributes in the LSA based on a route map. Values include:
  - <no parameter>
  - **route-map** *map_name*.

Examples

- These commands will always advertise the OSPFv2 default route regardless of whether the switch has a default route configured.

```
switch(config)#router ospf 1
switch((config-router-ospf)#default-information originate always
switch(config-router-ospf)#show active
router ospf 1
   default-information originate always
```

- These commands advertise a default route with a metric of 100 and an external metric type of 1 if a default route is configured.

```
switch(config)#router ospf 1
switch((config-router-ospf)#default-information originate metric 100 metric-type
1
```

distance ospf (OSPFv2)

The **distance ospf intra-area** command specifies the administrative distance for routes in a single OSPFv2 area. The default administrative distance for intra-area, inter-area and external routes is 110.

The **no distance ospf intra-area** and **default distance ospf intra-area** commands remove the **distance ospf intra-area** command from ***running-config***, returning the OSPFv2 administrative distance settings to the default value of 110.

Command Mode

Router-OSPF Configuration

Command Syntax

```
distance ospf AREA_TYPE distance
no distance ospf AREA_TYPE
default distance ospf AREA_TYPE
```

Parameters

- ***AREA_TYPE*** specifies routes for which administrative distance is to be set. Values include:
  - **external** .
  - **inter-area**.
  - **intra-area**
- *distance* Values range from 1 to 255. Default value is 110 for all types.

Example

- This command configures a distance of 85 for all OSPFv2 intra-area routes on the switch.

```
switch(config)#router ospf 6
switch(config-router-ospf)#distance ospf intra-area 85
switch(config-router-ospf)#
```

distribute-list in

A distribute list uses a route map or prefix list to filter specific routes from incoming OSPF LSAs. Filtering occurs after SPF calculation. The filtered routes are not installed on the switch, but are still included in LSAs sent by the switch. The **distribute-list in** command creates a distribute list in the configuration mode OSPF instance.

If a prefix list is used, destination prefixes that do not match the prefix list will not be installed. If a route map is used, routes may be filtered based on address, next hop, or metric. OSPF external routes may also be filtered by metric type or tag.

The **no distribute-list in** and **default distribute-list in** commands remove the **distribute-list in** command from ***running-config***.

Command Mode

Router-OSPF Configuration

Command Syntax

```
distribute-list {prefix-list|route-map} list_name in
no distribute-list {prefix-list|route-map}
default distribute-list {prefix-list|route-map}
```

Parameters

- **prefix-list** specifies a prefix-list as the filter.
- **route-map** specifies a route-map as the filter.
- *list_name* the name of the prefix-list or route-map used to filter routes from incoming LSAs.

Example

- These commands configure a prefix list named “dist_list1” in OSPF instance 5 to filter certain routes from incoming OSPF LSAs.

```
switch(config)#router ospf 5
switch(config-router-ospf)#distribute-list prefix-list dist_list1 in
switch(config-router-ospf)#
```

ip ospf authentication

The **ip ospf authentication** command enables OSPFv2 authentication for the configuration mode interface..

The **no ip ospf authentication** and **default ip ospf authentication** commands disable OSPFv2 authentication on the configuration mode interface by removing the corresponding **ip ospf authentication** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration
Interface-Port-Channel Configuration
Interface-VLAN Configuration

Command Syntax

```
ip ospf authentication [METHOD]
no ip ospf authentication
default ip ospf authentication
```

Parameters

- ***METHOD*** OSPFv2 authentication method. Options include:
  - <no parameter>
  - **message-digest**

Examples

- This command enables simple authentication on VLAN 12.

```
switch(config)#interface vlan 12
switch(config-if-vl12)#ip ospf authentication
switch(config-if-vl12)#
```

- This command enables message-digest authentication on VLAN 12.

```
switch(config-if-vl12)#ip ospf authentication message-digest
switch(config-if-vl12)#
```

ip ospf authentication-key

The **ip ospf authentication-key** command configures the OSPFv2 authentication password for the configuration mode interface.

The **no ip ospf authentication-key** and **default ip ospf authentication-key** commands removes the OSPFv2 authentication password from the configuration mode interface by removing the corresponding **ip ospf authentication-key** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration
Interface-Port-Channel Configuration
Interface-VLAN Configuration

Command Syntax

```
ip ospf authentication-key [ENCRYPT_TYPE] key_text
no ip ospf authentication-key
default ip ospf authentication-key
```

Parameters

- ***ENCRYPT_TYPE*** encryption level of the *key_text* parameter. Values include:
  - <no parameter> the *key_text* is in clear text.
  - **0** *key_text* is in clear text. Equivalent to <no parameter>.
  - **7** *key_text* is MD5 encrypted.
- *key_text* the authentication-key password.

Example

- This command specifies a password in clear text.

```
switch(config)#interface vlan 12
switch(config-if-Vl12)#ip ospf authentication-key 0 code123
switch(config-if-Vl12)#show active
interface Vlan12
   ip ospf authentication-key 7 baYllFzVbcx4yHq1IhmMdw==
switch(config-if-Vl12)#
```

Running-config stores the password as an encrypted string.

ip ospf cost

The **ip ospf cost** command configures the OSPFv2 cost for the configuration mode interface. The default cost depends on the interface type:

- Ethernet: determined by the **auto-cost reference-bandwidth (OSPFv2)** command.
- Port channel: 10.
- VLAN: 10.

The **no ip ospf cost** and **default ip ospf cost** commands restore the default OSPFv2 cost for the configuration mode interface by removing the corresponding **ip ospf cost** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration
Interface-Loopback Configuration
Interface-Port-Channel Configuration
Interface-VLAN Configuration

Command Syntax

```
ip ospf cost interface_cost
no ip ospf cost
default ip ospf cost
```

Parameters

- *interface_cost* Value ranges from 1 to 65535; default is 10.

Examples

- This command configures a cost of 15 for VLAN 2.

```
switch(config)#interface vlan 2
switch(config-if-Vl2)#ip ospf cost 15
switch(config-if-Vl2)#
```

ip ospf dead-interval

The **ip ospf dead-interval** command configures the dead interval for the configuration mode interface.

The **no ip ospf dead-interval** and **default ip ospf dead-interval** commands restore the default dead interval of 40 seconds on the configuration mode interface by removing the corresponding **ip ospf dead-interval** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration
Interface-Port-Channel Configuration
Interface-VLAN Configuration

Command Syntax

```
ip ospf dead-interval time
no ip ospf dead-interval
default ip ospf dead-interval
```

Parameters

- *time* Value ranges from 1 to 8192; default is 40.

Example

- This command configures a dead interval of 120 seconds for VLAN 4.

```
switch(config)#interface vlan 4
switch(config-if-Vl4)#ip ospf dead-interval 120
switch(config-if-Vl4)#
```

ip ospf hello-interval

The **ip ospf hello-interval** command configures the OSPFv2 hello interval for the configuration mode interface.

The same hello interval should be specified for Each OSPFv2 neighbor, and should not be longer than any neighbor's dead interval.

The **no ip ospf hello-interval** and **default ip ospf hello-interval** commands restore the default hello interval of 10 seconds on the configuration mode interface by removing the **ip ospf hello-interval** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration
Interface-Port-Channel Configuration
Interface-VLAN Configuration

Command Syntax

```
ip ospf hello-interval time
no ip ospf hello-interval
default ip ospf hello-interval
```

Parameters

- *time* hello interval (seconds). Values range from 1 to 8192; default is 10.

Example

- This command configures a hello interval of 30 seconds for VLAN 2.

```
switch(config)#interface vlan 2
switch(config-if-Vl2)#ip ospf hello-interval 30
switch(config-if-Vl2)#
```

ip ospf message-digest-key

The **ip ospf message-digest-key** command configures a message digest authentication key for the configuration mode interface.

The **no ip ospf message-digest-key** and **default ip ospf message-digest-key** commands remove the message digest authentication key for the specified key ID on the configuration mode interface by deleting the corresponding **ip ospf message-digest-key** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration
Interface-Port-Channel Configuration
Interface-VLAN Configuration

Command Syntax

```
ip ospf message-digest-key key_id md5 ENCRYPT_TYPE key_text
no ip ospf message-digest-key key_id
default ip ospf message-digest-key key_id
```

Parameters

- *key_id* key ID number. Value ranges from 1 to 255.
- ***ENCRYPT_TYPE*** encryption level of the *key_text* parameters. Values include:
  - <no parameter>
  - **0** *key_text*
  - **7** *key_text*
- *key_text* message key (password).

Example

- This command configures ***code123*** as the MD5 key with a corresponding key ID of 23.

```
switch(config)#interface vlan 12
switch(config-if-vl12)#ip ospf message-digest-key 23 md5 0 code123
switch(config-if-vl12)#
```

Running-config stores the password as an encrypted string.

ip ospf name-lookup

The **ip ospf name-lookup** command causes the switch to display DNS names in place of numeric OSPFv2 router IDs in all OSPFv2 show commands, including:

- **show ip ospf**
- **show ip ospf border-routers**
- **show ip ospf database <link state list>**
- **show ip ospf database database-summary**
- **show ip ospf database <link-state details>**
- **show ip ospf interface**
- **show ip ospf neighbor**
- **show ip ospf request-list**
- **show ip ospf retransmission-list**.

The **no ip ospf name-lookup** and **default ip ospf name-lookup** commands remove the **ip ospf name-lookup** command from ***running-config***, restoring the default behavior of displaying OSPFv2 router IDs by their numeric value.

Command Mode

Global Configuration

Command Syntax

```
ip ospf name-lookup
no ip ospf name-lookup
default ip ospf name-lookup
```

Example

- This command programs the switch to display OSPFv2 router IDs by the corresponding DNS name in subsequent show commands.

```
switch(config)#ip ospf lookup
switch(config)#
```

ip ospf network point-to-point

The **ip ospf network point-to-point** command sets the configuration mode interface as a point-to-point link. By default, interfaces are configured as broadcast links.

The **no ip ospf network** and **default ip ospf network** commands set the configuration mode interface as a broadcast link by removing the corresponding **ip ospf network** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration
Interface-Port-Channel Configuration
Interface-VLAN Configuration

Command Syntax

```
ip ospf network point-to-point
no ip ospf network
default ip ospf network
```

Examples

- These commands configure Ethernet interface 10 as a point-to-point link.

```
switch(config)#interface ethernet 10
switch(config-if-Et10)#ip ospf network point-to-point
switch(config-if-Et10)#
```

- This command restores Ethernet interface 10 as a broadcast link.

```
switch(config-if-Et10)#no ip ospf network
switch(config-if-Et10)#
```

ip ospf priority

The **ip ospf priority** command configures OSPFv2 router priority for the configuration mode interface..

The **no ip ospf priority** and **default ip ospf priority** commands restore the default priority (1) on the configuration mode interface by removing the corresponding **ip ospf priority** command from ***running-config.***

Command Mode

Interface-Ethernet Configuration
Interface-Port-Channel Configuration
Interface-VLAN Configuration

Command Syntax

```
ip ospf priority priority_level
no ip ospf priority
default ip ospf priority
```

Parameters

- *priority_level* priority level. Value ranges from 0 to 255. Default value is 1.

Examples

- This command configures a router priority of 15 for VLAN 8.

```
switch(config)#interface vlan 8
switch(config-if-Vl8)#ip ospf priority 15
switch(config-if-Vl8)#
```

- This command restores the router priority of 1 for VLAN 7.

```
switch(config)#interface vlan 7
switch(config-if-Vl7)#no ip ospf priority
switch(config-if-Vl7)#
```

ip ospf retransmit-interval

The **ip ospf retransmit-interval** command configures the link state advertisement retransmission interval for the interface.

The **no ip ospf retransmit-interval** and **default ip ospf retransmit-interval** commands restore the default retransmission interval of 5 seconds on the configuration mode interface by removing the corresponding **ip ospf retransmit-interval** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration
Interface-Port-Channel Configuration
Interface-VLAN Configuration

Command Syntax

```
ip ospf retransmit-interval period
no ip ospf retransmit-interval
default ip ospf retransmit-interval
```

Parameters

- *period* retransmission interval (seconds). Value ranges from 1 to 8192; default is 5.

Example

- This command configures a retransmission interval of 15 seconds for VLAN 3.

```
switch(config)#interface vlan 3
switch(config-if-Vl3)#ip ospf retransmit-interval 15
switch(config-if-Vl3)#
```

ip ospf shutdown

The **ip ospf shutdown** command disables OSPFv2 on the configuration mode interface without disrupting the OSPFv2 configuration. When OSPFv2 is enabled on the switch, the it is also enabled by default on all interfaces.

The OSPFv2 instance is disabled on the entire switch with the **shutdown (OSPFv2)** command.

The **no ip ospf shutdown** and **default ip ospf shutdown** commands enable OSPFv2 on the configuration mode interface by removing the corresponding **ip ospf shutdown** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration
Interface-Port-Channel Configuration
Interface-VLAN Configuration

Command Syntax

```
ip ospf shutdown
no ip ospf shutdown
default ip ospf shutdown
```

Examples

- This command shuts down OSPFv2 activity on VLAN 5.

```
switch(config)#interface vlan 5
switch(config-if-Vl5)#ip ospf shutdown
switch(config-if-Vl5)#
```

- This command resumes OSPFv2 activity on VLAN 5.

```
switch(config-if-Vl5)#no ip ospf shutdown
switch(config-if-Vl5)#
```

ip ospf transmit-delay

The **ip ospf transmit-delay** command configures the transmission delay for OSPFv2 packets over the configuration mode interface.

The **no ip ospf transmit-delay** and **default ip ospf transmit-delay** commands restore the default transmission delay (one second) on the configuration mode interface by removing the corresponding **ip ospf transmit-delay** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration
Interface-Port-Channel Configuration
Interface-VLAN Configuration

Command Syntax

```
ip ospf transmit-delay trans
no ip ospf transmit-delay
default ip ospf transmit-delay
```

Parameters

- *trans* LSA transmission delay (seconds). Value ranges from 1 to 8192; default is 1.

Example

- This command configures a transmission delay of 5 seconds for VLAN 6.

```
switch(config)#interface vlan 6
switch(config-if-Vl6)#ip ospf transmit-delay 5
switch(config-if-Vl6)#
```

log-adjacency-changes (OSPFv2)

The **log-adjacency-changes** command enables syslog messages to be sent when it detects OSPFv2 link state changes or when it detects that a neighbor has gone up or down. Log message sending is enabled by default.

The **default log-adjacency-changes** command restores the default state by removing the **log-adjacency-changes** statement from ***running-config***.

The default option (sending a message only when a neighbor goes up or down) is active when running-config does not contain any form of the command. Entering the command in any form replaces the previous command state in ***running-config***.

The **no log-adjacency-changes** disables link state change syslog reporting.

The **default log-adjacency-changes** command restores the default state by removing the **log-adjacency-changes detail** or **no log-adjacency-changes** statement from ***running-config***.

Command Mode

Router-OSPF Configuration

Command Syntax

```
log-adjacency-changes
log-adjacency-changes detail
no log-adjacency-changes
default log-adjacency-changes
```

Examples

- This command configures the switch to send a syslog message when a neighbor goes up or down.

```
switch(config)#router ospf 6
switch(config-router-ospf)#log-adjacency-changes
switch(config-router-ospf)#
```

After entering the command, **show active** does not display a **log-adjacency-changes** statement.

```
switch(config-router-ospf)#show active
router ospf 1
switch(config-router-ospf)#
```

- This command configures the switch to send a syslog message when it detects any link state change.

```
switch(config-router-ospf)#log-adjacency-changes detail
switch(config-router-ospf)#
```

After entering the command, **show active** displays a **log-adjacency-changes detail** command.

```
switch(config-router-ospf)#show active
router ospf 1
   log-adjacency-changes detail
switch(config-router-ospf)#
```

max-lsa (OSPFv2)

The **max-lsa** command specifies the number of LSAs allowed in the LSDB. Setting the limit to zero removes the LSDB restriction and disables LSA overload actions. Actions triggered by LSDB overload conditions include:

- Warning
- Temporary shutdown
- Permanent shutdown

The **no max-lsa** and **default max-lsa** commands restore all LSA overload parameters to their default settings by placing the **max-lsa 12000** statement in ***running-config***.

Command Mode

Router-OSPF Configuration

Command Syntax

```
max-lsa lsa_num [WARNING] [IGNORE_TIME] [IGNORE_COUNT] [RESET]
no max-lsa
default max-lsa
```

Parameters

- *lsa_num* maximum number of LSAs. Value ranges from 0 to 100,000.
  - **0** disables overload protection
  - ***1*** to ***100000*** Default value is 12,000.
- ***WARNING*** warning threshold, as a percentage of the maximum number of LSAs (% of *lsa_num*).
  - <no parameter> Default of 75%.
  - *percent* Ranges from 25 to 99.
- ***IGNORE_TIME*** temporary shutdown period (minutes). Options include:
  - <no parameter> Default value of 5 minutes.
  - **ignore-time** *period* Value ranges from 1 to 60.
- ***IGNORE_COUNT*** number of temporary shutdowns required to trigger a permanent shutdown.
  - <no parameter> Default value of 5.
  - **ignore-count** *episodes* Ranges from 1 to 20.
- ***RESET*** period of not exceeding LSA limit required to reset temporary shutdown counter to zero.
  - <no parameter> Default value of 5 minutes
  - **reset-time** *r_period* Ranges from 1 to 60.

Example

- This command defines an LSA limit of 8,000 and other parameters.

```
switch(config-router-ospf)#max-lsa 8000 40 ignore-time 6 ignore-count 3
reset-time 20
```

max-metric router-lsa (OSPFv2)

The **max-metric router-lsa** command configures OSPF to include the maximum value in LSA metric fields to keep other network devices from using the switch as a preferred intermediate SPF hop.

The **no max-metric router-lsa** and **default max-metric router-lsa** commands disable the advertisement of a maximum metric.

Command Mode

Router-OSPF Configuration

Command Syntax

```
max-metric router-lsa [EXTERNAL][STUB][STARTUP][SUMMARY]
no max-metric router-lsa [EXTERNAL][STUB][STARTUP][SUMMARY]
default max-metric router-lsa [EXTERNAL][STUB][STARTUP][SUMMARY]
```

All parameters can be placed in any order.

Parameters

- ***EXTERNAL*** advertised metric value. Values include:
  - <no parameter> Default value of 1.
  - **external-lsa**
  - **external-lsa** *<1* to *16777215>* Default value is ***0xFF0000***.
- ***STUB*** advertised metric type. Values include:
  - <no parameter> Default value of 2.
  - **include-stub**
- ***STARTUP*** limit scope of LSAs. Values include:
  - <no parameter>
  - **on-startup**
  - **on-startup wait-for-bgp**
  - **on-startup** *<5* to *86400>*

    wait-for-bgp or an **on-start** time value is not included in **no** and **default** commands.
- ***SUMMARY*** advertised metric value. Values include:
  - <no parameter>
  - **summary-lsa**
  - **summary-lsa** ***<1*** to ***16777215>***

Example

- This command configures OSPF to to include the maximum value in LSA metric fields until BGP has converged:

```
switch(config-router-ospf)#max-metric router-lsa on-startup wait-for-bgp
switch(config-router-ospf)#
```

maximum-paths (OSPFv2)

The **maximum-paths** command controls the number of parallel routes that OSPFv2 supports. The default maximum is 16 paths.

The **no maximum-paths** and **default maximum-paths** commands restore the maximum number of parallel routes that OSPFv2 supports on the switch to the default value of 16 by placing the **maximum-paths 16** statement in ***running-config***.

Command Mode

Router-OSPF Configuration

Command Syntax

```
maximum-paths paths
no maximum-paths
default maximum-paths
```

Parameters

- *paths* maximum number of parallel routes.

  Value ranges from 1 to the number of interfaces available per ECMP group, which is platform dependent.

  - Arad: Value ranges from 1 to 128. Default value is 128.
  - FM6000: Value ranges from 1 to 32. Default value is 32.
  - PetraA: Value ranges from 1 to 16. Default value is 16.
  - Trident: Value ranges from 1 to 32. Default value is 32.
  - Trident-II: Value ranges from 1 to 128. Default value is 128.

Example

- This command configures the maximum number of OSPFv2 parallel paths to 12.

```
switch(config)#router ospf 6
switch(config-router-ospf)#maximum-paths 12
switch(config-router-ospf)#
```

network area (OSPFv2)

The **network area** command assigns the specified IPv4 subnet to an OSPFv2 area.

The **no network area** and **default network area** commands delete the specified network area assignment by removing the corresponding **network area** command from ***running-config***.

Command Mode

Router-OSPF Configuration

Command Syntax

network *ipv4_subnet* **area** *area_id*
no network *ipv4_subnet* **area** *area_id*
default network *ipv4_subnet* **area** *area_id*

Parameters

- *ipv4_subnet* IPv4 subnet. Entry formats include address-prefix (CIDR) or address-wildcard mask.
  running-config stores value in CIDR notation.
- *area_id* area number. <**0** to **4294967295**> or <**0.0.0.0** to **255.255.255.255**>
  Running-config stores value in dotted decimal notation.

Example

- These equivalent commands each assign the subnet 10.1.10.0/24 to area 0.

```
switch(config-router-ospf)#network 10.1.10.0 0.0.0.255 area 0
switch(config-router-ospf)#

switch(config-router-ospf)#network 10.1.10.0/24 area 0
switch(config-router-ospf)#
```

no area (OSPFv2)

The **no area <type>** command removes the corresponding **area <type>** command from ***running-config***:

- **no/default area nssa translate type7 always** commands remove the **translate type7 always** parameter without changing the area type.
- **no/default area nssa , no/default area stub**, and **no/default area stub no-summary** commands restore the area's type to ***normal***.
- **no/default area default-information-originate** command removes all area commands for the specified area from ***running-config***
- **no/default area** command removes all area commands for the specified area from ***running-config***
- **no/default area** command removes all area commands for the specified area from ***running-config.***

Command Mode

Router-OSPF Configuration

Command Syntax

```
no area area_id [TYPE]
default area area_id [TYPE]
```

Parameters

- *area_id* area number.
  - Valid formats: integer <**1** to **4294967295**> or dotted decimal <**0.0.0.1** to **255.255.255.255**>
  - Area 0 (or 0.0.0.0) is not configurable; it is always ***normal***.
  - ***Running-config*** stores value in dotted decimal notation.
- ***TYPE*** area type. Values include:
  - **nssa**
  - **nssa translate type7 always**
  - **stub**
  - **stub no-summary**

Examples

- These commands remove area 1 from the running configuration.

```
switch(config)#router ospf 6
switch(config-router-ospf)# no area 1
switch(config-router-ospf)#
```

- These commands remove area 10.92.148.17 as an NSSA.

```
switch(config-router-ospf)#no area 10.92.148.17 nssa
switch(config-router-ospf)#
```

passive-interface default (OSPFv2)

The **passive-interface default** command configures all interfaces as OSPFv2 passive by default. The switch advertises the passive interface as part of the router LSA.

The **passive-interface <interface> (OSPFv2)** configures the OSPFv2 active-passive status for a specific interface:

- When ***passive-interface default*** is not set, all interfaces are OSPFv2 active by default and passive interfaces are denoted by **passive-interface <interface>** statements in ***running-config***.
- When ***passive-interface default*** is set, all interfaces are OSPFv2 passive by default and active interfaces are denoted by **no passive-interface <interface>** statements in ***running-config***.

The **no passive-interface** and **default passive-interface** commands configures all interfaces as OSPFv2 active by default by removing the **passive-interface default** statement from ***running-config***.

Command Mode

Router-OSPF Configuration

Command Syntax

```
passive-interface default
no passive-interface default
default passive-interface default
```

Examples

- This command configures the default interface setting as OSPFv2 passive. This command also removes all **passive-interface <interface>** statements from ***running-config***.

```
switch(config)#router ospf 6
switch(config-router-ospf)#passive-interface default
switch(config-router-ospf)#
```

- This command configures the default interface setting as OSPFv2 active. This command also removes all **no passive-interface <interface>** statements from ***running-config***.

```
switch(config-router-ospf)#no passive-interface default
switch(config-router-ospf)#
```

passive-interface <interface> (OSPFv2)

The **passive-interface** command disables OSPFv2 on an interface range. The switch advertises the passive interface as part of the LSA.

The default OSPFv2 interface activity is configured by the **passive-interface default (OSPFv2)** command:

- When ***passive-interface default*** is not set, all interfaces are OSPFv2 active by default and passive interfaces are denoted by **passive-interface <interface>** statements in ***running-config***.
- When ***passive-interface default*** is set, all interfaces are OSPFv2 passive by default and active interfaces are denoted by **no passive-interface <interface>** statements in ***running-config***.

The **no passive-interface** command enables OSPFv2 on the specified interface range. The **default passive-interface** command sets the interface to the default interface activity setting by removing the corresponding **passive-interface** or **no passive-interface** statement from ***running-config***.

Command Mode

Router-OSPF Configuration

Command Syntax

```
passive-interface INTERFACE_NAME
no passive-interface INTERFACE_NAME
default passive-interface INTERFACE_NAME
```

Parameters

- ***INTERFACE_NAME*** interface to be configured. Options include:
  - **ethernet** *e_range*
  - **port-channel** *p_range*
  - **vlan** *v_range*
  - **vxlan** *vx_range*

Examples

- These commands configure Ethernet interfaces 2 through 5 as passive interfaces.

```
switch(config)#router ospf 6
switch(config-router-ospf)#passive-interface ethernet 2-5
switch(config-router-ospf)#
```

- This command configures VLAN interfaces 50-54, 61, 68, and 102-120 as passive interfaces.

```
switch(config-router-ospf)#passive-interface vlan 50-54,61,68,102-120
switch(config-router-ospf)#
```

- This command configures VLAN 2 as an active interface.

```
switch(config-router-ospf)#no passive-interface vlan 2
switch(config-router-ospf)#
```

point-to-point routes (OSPFv2)

The point-to-point routes command enables the switch to maintain a local routing information base (RIB) to store information it learns from its neighbors.

The **point-to-point routes** and **default point-to-point routes** commands program the switch to include point-to-point links in its RIB by removing the **no point-to-point routes** command from ***running-config***.

Command Mode

Router-OSPF Configuration

Command Syntax

```
point-to-point routes
no point-to-point routes
default point-to-point routes
```

Examples

- This command configures the switch to optimize the local RIB by not including point-to-point routes.

```
switch(config)#router ospf 6
switch(config-router-ospf)#no point-to-point routes
switch(config-router-ospf)#
```

- This command configures the switch to include point-to-point routes.

```
switch(config-router-ospf)#point-to-point routes
switch(config-router-ospf)#
```

redistribute (OSPFv2)

The **redistribute** command enables the advertising of all specified routes on the switch into the OSPFv2 domain as external routes.

The **no redistribute** and **default redistribute** commands remove the corresponding **redistribute** command from ***running-config***, disabling route redistribution for the specified route type.

Command Mode

Router-OSPF Configuration

Command Syntax

```
redistribute ROUTE_TYPE [ROUTE_MAP]
no redistribute ROUTE_TYPE
default redistribute ROUTE_TYPE
```

Parameters

- ***ROUTE_TYPE*** source from which routes are redistributed. Options include:
  - **connected** routes that are established when IPv4 is enabled on an interface.
  - **BGP** routes from a BGP domain.
  - **RIP** routes from a RIP domain.
  - **static** IP static routes.
- ***ROUTE_MAP*** route map that determines the routes that are redistributed. Options include:
  - <no parameter >
  - **route-map** *map_name*

Examples

- The **redistribute static** command starts the advertising of static routes as OSPFv2 external routes.

```
switch(config)#router ospf 6
switch(config-router-ospf)#redistribute static
switch(config-router-ospf)#
```

- The **no redistribute bgp** command stops the advertising of BGP routes as OSPFv2 external routes.

```
switch(config-router-ospf)#no redistribute bgp
switch(config-router-ospf)#
```

router-id (OSPFv2)

The **router-id** command assigns a router ID for an OSPFv2 instance. This number uniquely identifies the router within an Autonomous System. Status commands use the router ID to identify the switch.

The switch sets the router ID to the first available alternative in the following list:

1. The **router-id** command.
2. The loopback IP address, if a loopback interface is configured on the switch.
3. The highest IP address present on the router.

Important! When configuring VXLAN on an MLAG, always manually configure the OSPFv2 router ID to prevent the switch from using the common VTEP IP address as the router ID.

The **no router-id** and **default router-id** commands remove the router ID command from ***running-config***; the switch uses the loopback or highest address as the router ID.

Command Mode

Router-OSPF Configuration

Command Syntax

```
router-id identifier
no router-id [identifier]
default router-id [identifier]
```

Parameters

- *identifier* Value ranges from 0.0.0.0 to 255.255.255.255.

Example

- This command assigns 10.5.4.2 as the router ID for the OSPFv2 instance.

```
switch(config)#router ospf 6
switch(config-router-ospf)#router-id 10.5.4.2
switch(config-router-ospf)#
```

router ospf

The **router ospf** command places the switch in router-ospf configuration mode. The switch will create a process ID for the new instance if one does not already exist. The **exit** command returns the switch to global configuration mode.

The **show ip ospf** command displays the process ID of the OSPFv2 instances configured on the switch.

The **no router ospf** and **default router ospf** commands delete the specified OSPFv2 instance.

Router-ospf configuration mode is not a group change mode; ***running-config*** is changed immediately upon entering commands. Exiting router-ospf configuration mode does not affect ***running-config***. The **exit** command returns the switch to global configuration mode.

Refer to Router-OSPFv2 Configuration Mode (page 1682) for a list of commands available in router-ospf configuration mode.

Command Mode

Global Configuration

Command Syntax

```
router ospf process_id [VRF_INSTANCE]
no router ospf process_id [VRF_INSTANCE]
default router ospf process_id [VRF_INSTANCE]
```

Parameters

- *process_id* OSPFv2 process ID. Values range from 1 to 65535.
- VRF_INSTANCE
  - <no parameter> configures the default VRF instance.
  - **vrf** *vrf_name* configures the vrf_name instance.

Examples

- This command creates an OSPFv2 instance with process ID 145 in the main VRF.

```
switch(config)#router ospf 145
switch(config-router-ospf)#
```

- This command deletes the specified OSPFv2 instance.

```
switch(config)#no router ospf 145
switch(config)#
```

show ip ospf

The **show ip ospf** command displays OSPFv2 routing information

Command Mode

EXEC

Command Syntax

show ip ospf [*PROCESS_ID*] [*VRF_INSTANCE*]

Parameters

- ***PROCESS_ID*** OSPFv2 process ID. Values include:
  - <no parameter>
  - <***1*** to ***65535***>
- ***VRF_INSTANCE*** specifies the VRF instance.
  - <no parameter> configures the default VRF instance.
  - **vrf** *vrf_name* configures the vrf_name instance.

Example

- This command displays configuration parameters, operational statistics, status of the OSPFv2 instance, and a brief description of the areas on the switch.

```
switch>show ip ospf
Routing Process "ospf 1" with ID 10.168.103.1 VRF default
 Supports opaque LSA
 Maximum number of LSA allowed 12000
  Threshold for warning message 75%
  Ignore-time 5 minutes, reset-time 5 minutes
  Ignore-count allowed 5, current 0
 It is an area border router
 Hold time between two consecutive SPFs 5000 msecs
 SPF algorithm last executed 00:00:09 ago
 Minimum LSA interval 5 secs
 Minimum LSA arrival 1000 msecs
 Number of external LSA 0. Checksum Sum 0x000000
 Number of opaque AS LSA 0. Checksum Sum 0x000000
 Number of LSA 27.
 Number of areas in this router is 3. 3 normal 0 stub 0 nssa
    Area BACKBONE(0.0.0.0)
        Number of interfaces in this area is 2
        It is a normal area
        Area has no authentication
        SPF algorithm executed 153 times
        Number of LSA 8. Checksum Sum 0x03e13a
        Number of opaque link LSA 0. Checksum Sum 0x000000
    Area 0.0.0.2
        Number of interfaces in this area is 1
        It is a normal area
        Area has no authentication
        SPF algorithm executed 153 times
        Number of LSA 11. Checksum Sum 0x054e57
        Number of opaque link LSA 0. Checksum Sum 0x000000
    Area 0.0.0.3
        Number of interfaces in this area is 1
        It is a normal area
        Area has no authentication
        SPF algorithm executed 5 times
        Number of LSA 6. Checksum Sum 0x02a401
        Number of opaque link LSA 0. Checksum Sum 0x000000
```

show ip ospf border-routers

The **show ip ospf border-routers** command displays the internal OSPFv2 routing table entries to area border routers (ABRs) and autonomous system boundary routers (ASBRs) for each of the OSPFv2 areas.

Command Mode

EXEC

Command Syntax

show ip ospf border-routers [*VRF_INSTANCE*]

Parameters

- ***VRF_INSTANCE*** specifies the VRF instance.
  - <no parameter> displays information from all VRFs, or from context-active VRF if set.
  - **vrf** *vrf_name* displays information from the specified VRF.

Example

- This command displays the ABRs and ASBRs

```
switch>show ip ospf border-routers
OSPF Process 10.17.0.42, VRF default

Router ID       Area            Type
10.17.0.1      0.0.0.0         ASBR
switch>
```

show ip ospf database database-summary

The **show ip ospf database database-summary** command displays the number of link state advertisements in the OSPFv2 database.

Command Mode

EXEC

Command Syntax

show ip ospf [*AREA*] database database-summary [*VRF_INSTANCE*]

Parameters

- ***VRF_INSTANCE*** specifies the VRF instance.
  - <no parameter> displays information from all VRFs, or from context-active VRF if set.
  - **vrf** *vrf_name* displays information from the specified VRF.
- ***AREA*** areas for which command displays data. Specifying an individual area requires entering the process ID where the area is located. Options include:
  - <no parameter>
  - *process_id*
  - *process_id area_id*
  - *process_id* input range: <1 to 65535>
  - *area_id* input range: <0 to 4294967295> or <0.0.0.0 to 255.255.255.255>

Example

- This command displays the LSDB content summary for area 0.

```
switch>show ip ospf 1 0 database database-summary

  LSA Type      Count
  Router        18
  Network       21
  Summary Net   59
  Summary ASBR  4
  Type-7 Ext    0
  Opaque Area   0
  Type-5 Ext    4238
  Opaque AS     0
  Total         4340

switch>
```

show ip ospf database <link state list>

The **show ip ospf database <link state list>** command displays the OSPFv2 link state advertisements that originate on a specified switch.

Command Mode

EXEC

Command Syntax

show ip ospf [*AREA*] database [*ROUTER*] [*VRF_INSTANCE*]

Parameters

- ***AREA*** areas for which command displays data. Specifying an individual area requires entering the process ID where the area is located. Options include:
  - <no parameter>
  - *process_id*
  - *process_id area_id*
  - *process_id* value ranges from 1 to 65535.
  - *area_id* is entered in decimal or dotted decimal notation.

- ***ROUTER*** router or switch for which the command provides data. Options include:
  - <no parameter>
  - **adv-router** [*a.b.c.d*]
  - **self-originate**
- ***VRF_INSTANCE*** specifies the VRF instance.
  - <no parameter> displays information from all VRFs, or from context-active VRF if set.
  - **vrf** *vrf_name* displays information from the specified VRF.

Example

- This command displays OSPFv2 LSAs that originate at the router with a router ID of 10.26.0.31.

```
switch>show ip ospf database adv-router 10.26.0.31

            OSPF Router with ID(10.26.0.23) (Process ID 1) (VRF default)

10.26.0.31     10.26.0.31     918         0x80002b4a   0x1315   3

                 Type-5 AS External Link States

Link ID         ADV Router      Age         Seq#         Checksum
10.24.238.238   10.26.0.31     678         0x800003d2   0x8acf   0
10.24.238.244   10.26.0.31     678         0x800003d2   0x4e06   0
10.24.238.224   10.26.0.31     678         0x800003d2   0x1751   0
<-------OUTPUT OMITTED FROM EXAMPLE-------->

                    Type 11 Opaque LSDB

Type         Link ID      ADV Router   Age       Seq# Checksum
switch>
```

show ip ospf database <link-state details>

The **show ip ospf database <link-state details>** command displays details of the specified link state advertisements.

Command Mode

EXEC

Command Syntax

show ip ospf [*AREA*] database *LINKSTATE_TYPE* *linkstate_id* **[*ROUTER*] [*VRF_INSTANCE*]**

Parameters

- ***AREA*** areas for which command displays data. Specifying an individual area requires entering the process ID where the area is located. Options include:
  - <no parameter>
  - *process_id*
  - *process_id area_id*
  - *process_id* input range: <1 to 65535>
  - *area_id* input range: <0 to 4294967295> or <0.0.0.0 to 255.255.255.255>
- ***LINKSTATE_TYPE*** link state types. Parameter options include:
  - **detail** Displays all link states.
  - **router**
  - **network**
  - **summary**
  - **asbr-summary**
  - **external**
  - **nssa-external**
  - **opaque-link**
  - **opaque-area**
  - **opaque-as**
- *linkstate_id* Network segment described by the LSA (dotted decimal notation).

  Value depends on the LSA type.
- ***ROUTER*** router or switch for which the command provides data. Options include:
  - <no parameter>
  - **adv-router** [*a.b.c.d*]
  - **self-originate**
- ***VRF_INSTANCE*** parameter has no effect; this command displays information about the specified process and area regardless of VRF.
  - <no parameter> displays information from all VRFs, or from context-active VRF if set.
  - **vrf** *vrf_name* displays information from the specified VRF.

Examples

- This command displays the router link states contained in the area 2 LSDB.

```
switch>show ip ospf 1 2 database router

            OSPF Router with ID(10.168.103.1) (Process ID 1) (VRF default)

                 Router Link States (Area 0.0.0.2)

  LS age: 00:02:16
  Options: (E DC)
  LS Type: Router Links
  Link State ID: 10.168.103.1
  Advertising Router: 10.168.103.1
  LS Seq Number: 80000032
  Checksum: 0x1B60
  Length: 36
  Number of Links: 1

    Link connected to: a Transit Network
     (Link ID) Designated Router address: 10.168.2.1
     (Link Data) Router Interface address: 10.168.2.1
      Number of TOS metrics: 0
       TOS 0 Metrics: 10


  LS age: 00:02:12
  Options: (E DC)
  LS Type: Router Links
  Link State ID: 10.168.104.2
  Advertising Router: 10.168.104.2
  LS Seq Number: 80000067
  Checksum: 0xA29C
  Length: 36
  Number of Links: 1

    Link connected to: a Transit Network
     (Link ID) Designated Router address: 10.168.2.1
     (Link Data) Router Interface address: 10.168.2.2
      Number of TOS metrics: 0
       TOS 0 Metrics: 10
switch>
```

- This command displays link state database (LSDB) contents for area 2.

```
switch>show ip ospf 1 2 database

            OSPF Router with ID(10.168.103.1) (Process ID 1) (VRF default)

                 Router Link States (Area 0.0.0.2)

Link ID         ADV Router      Age         Seq#       Checksum Link count
10.168.103.1    10.168.103.1    00:29:08    0x80000031 0x001D5F 1
10.168.104.2    10.168.104.2    00:29:09    0x80000066 0x00A49B 1

                 Net Link States (Area 0.0.0.2)

Link ID         ADV Router      Age         Seq#       Checksum
10.168.2.1      10.168.103.1    00:29:08    0x80000001 0x00B89D

                 Summary Net Link States (Area 0.0.0.2)

Link ID         ADV Router      Age         Seq#       Checksum
10.168.0.0      10.168.103.1    00:13:20    0x80000028 0x0008C8
10.168.0.0      10.168.104.2    00:09:16    0x80000054 0x00A2FF
10.168.3.0      10.168.104.2    00:24:16    0x80000004 0x00865F
10.168.3.0      10.168.103.1    00:24:20    0x80000004 0x002FC2
10.168.103.0    10.168.103.1    00:14:20    0x80000028 0x0096D2
10.168.103.0    10.168.104.2    00:13:16    0x80000004 0x00364B
10.168.104.0    10.168.104.2    00:08:16    0x80000055 0x002415
10.168.104.0    10.168.103.1    00:13:20    0x80000028 0x00EF6E
switch>
```

show ip ospf interface

The **show ip ospf interface** command displays interface information that is related to OSPFv2.

Command Mode

EXEC

Command Syntax

show ip ospf [*PROCESS_ID*] interface [*INTERFACE_NAME*] [*VRF_INSTANCE*]

Parameters

- ***PROCESS_ID*** OSPFv2 process ID. Values include:
  - <no parameter>
  - <***1*** to ***65535***>
- ***INTERFACE_NAME*** Interface type and number. Values include
  - <no parameter>
  - **ethernet** *e_num*
  - **loopback** *l_num*
  - **port-channel** *p_num*
  - **vlan** *v_num*
- ***VRF_INSTANCE*** specifies the VRF instance.
  - <no parameter> displays information from all VRFs, or from context-active VRF if set.
  - **vrf** *vrf_name* displays information from the specified VRF.

Related Command

show ip ospf interface brief

Example

- This command displays complete OSPFv2 information for VLAN 1.

```
switch>show ip ospf interface vlan 1
Vlan1 is up, line protocol is up (connected)
  Internet Address 10.168.0.1/24, VRF default, Area 0.0.0.0
  Process ID 1, Router ID 10.168.103.1, Network Type BROADCAST, Cost: 10
  Transmit Delay is 1 sec, State BDR, Priority 1
  Designated Router is 10.168.104.2
  Backup Designated router is 10.168.103.1
  Timer intervals configured, Hello 10, Dead 40, Retransmit 5
  Neighbor Count is 1
  MTU is 1500
switch>
```

show ip ospf interface brief

The **show ip ospf interface brief** command displays a summary of OSPFv2 information.

Command Mode

EXEC

Command Syntax

show ip ospf [*PROCESS_ID*] interface brief [*VRF_INSTANCE*]

Parameters

- ***PROCESS_ID*** OSPFv2 process ID. Values include:
  - <no parameter>
  - <***1*** to ***65535***>
- ***VRF_INSTANCE*** specifies the VRF instance.
  - <no parameter> displays information from all VRFs, or from context-active VRF if set.
  - **vrf** *vrf_name* displays information from the specified VRF.

Related Commands

show ip ospf interface

Example

- This command displays a summary of interface information for the switch.

```
switch>show ip ospf interface brief
Interface    PID   Area            IP Address         Cost  State    Nbrs
Loopback0    1     0.0.0.0         10.168.103.1/24    10    DR       0
Vlan1        1     0.0.0.0         10.168.0.1/24      10    BDR      1
Vlan2        1     0.0.0.2         10.168.2.1/24      10    BDR      1
Vlan3        1     0.0.0.3         10.168.3.1/24      10    DR       0
switch>
```

show ip ospf lsa-log

The **show ip ospf lsa-log** command displays log entries when LSA update messages are sent or received for OSPF.

Command Mode

EXEC

Command Syntax

show ip ospf [*PROCESS_ID*] ospf-log

Parameters

- ***PROCESS_ID*** OSPFv2 process ID. Values include:
  - <no parameter>
  - <***1*** to ***65535***>

Examples

- This command displays log entries when LSA update messages are sent or received for OSPF.

```
switch>show ip ospf lsa-log
OSPF Process 3.3.3.3, LSA Throttling Log:
[04:21:09] type 1: 3.3.3.3/32 [3.3.3.3], event 1, backed off, new hold value 2000 msecs
[04:21:08] type 1: 3.3.3.3/32 [3.3.3.3], event 2, backoff restarted, new hold value 900 msecs
[04:21:00] type 1: 3.3.3.3/32 [3.3.3.3], event 1, backed off, new hold value 3000 msecs
[04:21:00] type 1: 3.3.3.3/32 [3.3.3.3], event 4, maxwait value changed, new hold value 3000 msecs
/* Here the maxwait value was changed to 3000 from earlier 32000, this is not part of the log */
[04:20:42] type 1: 3.3.3.3/32 [3.3.3.3], event 1, backed off, new hold value 32000 msecs
[04:20:10] type 1: 3.3.3.3/32 [3.3.3.3], event 1, backed off, new hold value 32000 msecs
[04:19:54] type 1: 3.3.3.3/32 [3.3.3.3], event 1, backed off, new hold value 16000 msecs
[04:19:46] type 1: 3.3.3.3/32 [3.3.3.3], event 1, backed off, new hold value 8000 msecs
[04:19:42] type 1: 3.3.3.3/32 [3.3.3.3], event 1, backed off, new hold value 4000 msecs
[04:19:40] type 1: 3.3.3.3/32 [3.3.3.3], event 1, backed off, new hold value 2000 msecs
[04:19:39] type 1: 3.3.3.3/32 [3.3.3.3], event 2, backoff restarted, new hold value 900 msecs
[04:19:22] type 1: 4.4.4.4/32 [4.4.4.4], event 3, discarded, was early by 995 msecs
[04:19:22] type 1: 3.3.3.3/32 [3.3.3.3], event 0, backoff started, new hold value 1000 msecs
switch>
```

show ip ospf neighbor

The **show ip ospf neighbor** command displays OSPFv2 neighbor information for specified interfaces.

Command Mode

EXEC

Command Syntax

show ip ospf [*PROCESS_ID*] neighbor
[*INTERFACE_NAME*][*NEIGHBOR*][*DATA*][*VRF_INSTANCE*]

Parameters

- ***PROCESS_ID*** OSPFv2 process ID. Values include:
  - <no parameter>
  - <***1*** to ***65535***>
- ***INTERFACE_NAME*** Interface type and number. Values include:
  - <no parameter>
  - **ethernet** *e_num*
  - **loopback** *l_num*
  - **port-channel** *p_num*
  - **vlan** *v_num*
- NEIGHBOR OSPFv2 neighbor. Options include:
  - <no parameter>
  - *ipv4_addr*
- ***DATA*** Type of information the command displays. Values include:
  - <no parameter>
  - **detail**
- ***VRF_INSTANCE*** specifies the VRF instance.
  - <no parameter> displays information from all VRFs, or from context-active VRF if set.
  - **vrf** *vrf_name* displays information from the specified VRF.

Examples

- This command displays the switch's neighbors.

```
switch>show ip ospf neighbor
Neighbor ID    VRF     Pri  State        Dead Time   Address        Interface
10.168.104.2   default  1   FULL/DR      00:00:35    10.168.0.2     Vlan1
10.168.104.2   default  8   FULL/BDR     00:00:31    10.168.2.2     Vlan2
switch>
```

- This command displays details about the neighbors to VLAN 2.

```
switch>show ip ospf neighbor vlan 2 detail
 Neighbor 10.168.104.2, VRF default, interface address 10.168.2.2
    In the area 0.0.0.2 via interface Vlan2
    Neighbor priority is 8, State is FULL, 13 state changes
    Adjacency was established 000:01:25:48 ago
    DR is 10.168.2.1 BDR is 10.168.2.2
    Options is E
    Dead timer due in 00:00:34
switch>
```

show ip ospf neighbor adjacency-changes

The **show ip ospf neighbor adjacency-changes** command displays the OSPFv2 neighbor adjacency change log for specified interfaces.

Command Mode

EXEC

Command Syntax

show ip ospf neighbor [*INTERFACE_NAME*] [*NEIGHBOR*] adjacency-changes [*VRF_INSTANCE*]

Parameters

- ***INTERFACE_NAME*** Interface type and number. Values include:
  - <no parameter>
  - **ethernet** *e_num*
  - **loopback** *l_num*
  - **port-channel** *p_num*
  - **vlan** *v_num*
- NEIGHBOR OSPFv2 neighbor. Options include:
  - <no parameter>
  - *ipv4_addr*
  - *host_name*
- ***VRF_INSTANCE*** specifies the VRF instance.
  - <no parameter> displays information from all VRFs, or from context-active VRF if set.
  - **vrf** *vrf_name* displays information from the specified VRF.

Examples

- This command displays the adjacency changes to VLAN 2.

```
switch>show ip ospf neighbor vlan 2 adjacency-changes
[08-04 08:55:32] 10.168.104.2, interface Vlan2 adjacency established
[08-04 09:58:51] 10.168.104.2, interface Vlan2 adjacency dropped: interface went
down
[08-04 09:58:58] 10.168.104.2, interface Vlan2 adjacency established
[08-04 09:59:34] 10.168.104.2, interface Vlan2 adjacency dropped: interface went
down
[08-04 09:59:42] 10.168.104.2, interface Vlan2 adjacency established
[08-04 10:01:40] 10.168.104.2, interface Vlan2 adjacency dropped: nbr did not
list our router ID
[08-04 10:01:46] 10.168.104.2, interface Vlan2 adjacency established
switch>
```

show ip ospf neighbor state

The **show ip ospf neighbor state** command displays the state information on OSPF neighbors on a per-interface basis.

Command Mode

EXEC

Command Syntax

```
show ip ospf neighbor state STATE_NAME [VRF_INSTANCE]
```

Parameters

- ***STATE_NAME*** Output filtered by the devices OSPF state. Options include valid OSPF states:
  - **2-ways**
  - **attempt**
  - **down**
  - **exch-start**
  - **exchange**
  - **full**
  - **graceful-restart**
  - **init**
  - **loading**
- ***VRF_INSTANCE*** specifies the VRF instance.
  - <no parameter> displays information from all VRFs, or from context-active VRF if set.
  - **vrf** *vrf_name* displays information from the specified VRF.

Examples

- This command displays OSPF information for neighboring routers that are fully adjacent .

```
switch>show ip ospf neighbor state full
Neighbor ID     VRF    Pri   State           Dead Time   Address         Interface
Test1          default    1   FULL/BDR        00:00:35    10.17.254.105  Vlan3912
Test2          default    1   FULL/BDR        00:00:36    10.17.254.29   Vlan3910
Test3           default    1   FULL/DR          00:00:35    10.25.0.1     Vlan101
Test4          default    1   FULL/DROTHER    00:00:36    10.17.254.67   Vlan3908
Test5          default    1   FULL/DROTHER    00:00:36    10.17.254.68   Vlan3908
Test6          default    1   FULL/BDR        00:00:32    10.17.254.66   Vlan3908
Test7          default    1   FULL/DROTHER    00:00:34    10.17.36.4     Vlan3036
Test8          default    1   FULL/BDR        00:00:35    10.17.36.3     Vlan3036
Test9          default    1   FULL/DROTHER    00:00:31    10.17.254.13   Vlan3902
Test10         default    1   FULL/BDR        00:00:37    10.17.254.11   Vlan3902
Test11         default    1   FULL/DROTHER    00:00:33    10.17.254.163  Vlan3925
Test12         default    1   FULL/DR         00:00:37    10.17.254.161  Vlan3925
Test13         default    1   FULL/DROTHER    00:00:31    10.17.254.154  Vlan3923
Test14         default    1   FULL/BDR        00:00:39    10.17.254.156  Vlan3923
Test15         default    1   FULL/DROTHER    00:00:33    10.17.254.35   Vlan3911
Test16         default    1   FULL/DR         00:00:34    10.17.254.33   Vlan3911
Test17          default    1   FULL/DR        00:00:36   10.17.254.138  Ethernet12
Test18         default    1   FULL/DR         00:00:37    10.17.254.2    Vlan3901
switch>
```

show ip ospf neighbor summary

The **show ip ospf neighbor summary** command displays a single line of summary information for each OSPFv2 neighbor.

Command Mode

EXEC

Command Syntax

show ip ospf [*PROCESS_ID*] neighbor summary [*VRF_INSTANCE*]

Parameters

- ***PROCESS_ID*** OSPFv2 process ID. Values include:
  - <no parameter>
  - <***1*** to ***65535***>
- ***VRF_INSTANCE*** specifies the VRF instance.
  - <no parameter> displays information from all VRFs, or from context-active VRF if set.
  - **vrf** *vrf_name* displays information from the specified VRF.

Examples

- This command displays the summary information for the OSPFv2 neighbors.

```
switch>show ip ospf neighbor summary
OSPF Router with (Process ID 1) (VRF default)
0 neighbors are in state DOWN
0 neighbors are in state GRACEFUL RESTART
2 neighbors are in state INIT
0 neighbors are in state LOADING
0 neighbors are in state ATTEMPT
18 neighbors are in state FULL
0 neighbors are in state EXCHANGE
0 neighbors are in state 2 WAYS
0 neighbors are in state EXCH START
switch>
```

show ip ospf request-list

The **show ip ospf request-list** command displays a list of all OSPFv2 link state advertisements (LSAs) requested by a router.

Command Mode

- EXEC

Command Syntax

```
show ip ospf request-list [VRF_INSTANCE]
```

Parameters

- ***VRF_INSTANCE*** specifies the VRF instance.
  - <no parameter> displays information from all VRFs, or from context-active VRF if set.
  - **vrf** *vrf_name* displays information from the specified VRF.

Example

- This command displays an LSA request list.

```
switch>show ip ospf request-list
Neighbor 10.168.104.2 vrf default interface: 10.168.0.2 address vlan1
Type LS ID ADV RTR Seq No Age Checksum
Neighbor 10.168.104.2 vrf default interface: 10.168.2.2 address vlan2
Type LS ID ADV RTR Seq No Age Checksum
switch>
```

show ip ospf retransmission-list

The **show ip ospf retransmission-list** command displays a list of all OSPFv2 link state advertisements (LSAs) waiting to be re-sent.

Command Mode

EXEC

Command Syntax

show ip ospf retransmission-list [*VRF_INSTANCE*]

Parameters

- ***VRF_INSTANCE*** specifies the VRF instance.
  - <no parameter> displays information from all VRFs, or from context-active VRF if set.
  - **vrf** *vrf_name* displays information from the specified VRF.

Example

- This command displays an empty retransmission list.

```
switch>show ip ospf retransmission-list
Neighbor 10.168.104.2 vrf default interface vlan1 address 10.168.0.2
LSA retransmission not currently scheduled. Queue length is 0

Type         Link ID      ADV Router   Age       Seq# Checksum
Neighbor 10.168.104.2 vrf default interface vlan2 address 10.168.2.2
LSA retransmission not currently scheduled. Queue length is 0

Type         Link ID      ADV Router   Age       Seq# Checksum
switch>
```

show ip ospf spf-log

The **show ip ospf spf-log** command displays when and how long the switch took to run a full SPF calculation for OSPF.

Command Mode

EXEC

Command Syntax

show ip ospf [*PROCESS_ID*] ospf-log

Parameters

- ***PROCESS_ID*** OSPFv2 process ID. Values include:
  - <no parameter>
  - <***1*** to ***65535***>

Examples

- This command displays the SPF information for OSPF.

```
switch>show ip ospf spf-log
OSPF Process 172.26.0.22
When      Duration(msec)
13:01:34  1.482
13:01:29  1.547
13:01:24  1.893
13:00:50  1.459
13:00:45  1.473
13:00:40  2.603
11:01:49  1.561
11:01:40  1.463
11:01:35  1.467
11:01:30  1.434
11:00:54  1.456
11:00:49  1.472
11:00:44  1.582
15:01:49  1.575
15:01:44  1.470
15:01:39  1.679
15:01:34  1.601
15:00:57  1.454
15:00:52  1.446
15:00:47  1.603
switch>
```

shutdown (OSPFv2)

The **shutdown** command disables OSPFv2 on the switch. OSPFv2 is disabled on individual interfaces with the **shutdown (OSPFv2)** command.

The **no shutdown** and **default shutdown** commands enable the OSPFv2 instance by removing the **shutdown** statement from the OSPF block in ***running-config***.

Command Mode

Router-OSPF Configuration

Command Syntax

```
shutdown
no shutdown
default shutdown
```

Examples

- This command disables OSPFv2 activity on the switch.

```
switch(config)#router ospf 6
switch(config-router-ospf)#shutdown
switch(config-router-ospf)#
```

- This command resumes OSPFv2 activity on the switch.

```
switch(config-router-ospf)#no shutdown
switch(config-router-ospf)#
```

timers lsa arrival (OSPFv2)

The **timers lsa arrival** command sets the minimum interval for acceptance of identical link-state advertisements (LSAs) from OSPFv2 neighbors.

The **no timers lsa arrival** and **default timers lsa arrival** commands restore the minimum interval to the default of one second by removing the **timers lsa arrival** command from ***running-config***.

Command Mode

Router-OSPF Configuration

Command Syntax

```
timers lsa arrival lsa_time
no timers lsa arrival
default timers lsa arrival
```

Parameters

- *lsa_time* minimum time (in milliseconds) after which the switch will accept an identical LSA from OSPFv2 neighbors. Default is 1000 (1 second).

Example

- This command sets the minimum LSA arrival interval to ten milliseconds.

```
switch(config)#router ospf 6
switch(config-router-ospf)#timers lsa arrival 10
switch(config-router-ospf)#
```

timers throttle lsa all (OSPFv2)

The **timers throttle lsa all** command sets the rate-limiting values for OSPF link-state advertisement generation.

The **no timers throttle lsa all** and **default imers throttle lsa all** commands restore the defaults by removing the t**imers throttle lsa all** command from ***running-config***.

Command Mode

Router-OSPF Configuration

Command Syntax

```
timers throttle lsa all initial_delay min_hold max_wait
no timers throttle lsa all
default timers throttle lsa all
```

Parameters

- *initial_delay* Value ranges from ***0*** to ***600000*** (ms). Default is 1000.
- *min_hold* Value ranges from ***0*** to ***600000*** (ms). Default is 5000.
- *max_wait* Value ranges from ***0*** to ***600000*** (ms). Default is 5000.

Example

- This command sets the rate-limiting values for OSPF link-state advertisemen to 10 milliseconds.

```
switch(config)#router ospf 6
switch(config-router-ospf)#timers throttle lsa all 10
switch(config-router-ospf)#
```

timers throttle spf (OSPFv2)

The purpose of SPF throttling is to delay shortest path first (SPF) calculations when network topology is changing rapidly. The **timers throttle spf** command controls the intervals at which the switch will perform SPF calculations. The command sets three values:

- **Initial delay:** how long the switch waits to perform an SPF calculation after a topology change in a network that has been stable throughout the hold interval. Because a topology change often causes several link state updates to be sent, the initial delay is configured to allow the network to settle before the switch performs an SPF calculation. If an additional topology change occurs during the initial interval, the SPF calculation still takes place after the expiration of the initial delay period and no other change is made to the throttle timers.
- **Hold interval:** this is an additional wait timer which scales to slow SPF calculations during periods of network instability. If a network change occurs during the hold period, an SPF calculation is scheduled to occur at the expiration of the hold interval. Subsequent hold intervals are doubled if further topology changes occur during a hold interval until either the hold interval reaches its configured maximum or no topology change occurs during the interval. If the next topology change occurs after the expiration of the hold interval, the hold interval is reset to its configured value and the SPF calculation is scheduled to take place after the initial delay.
- **Maximum interval:** the maximum time the switch will wait after a topology change before performing an SPF calculation.

The **no timers throttle spf** and **default timers throttle spf** commands restore the default OSPFv2 SPF calculation intervals by removing the **timers throttle spf** command from ***running-config***.

Command Mode

Router-OSPF Configuration

Command Syntax

```
timers throttle spf initial_delay hold_interval max_interval
no timers spf
default timers spf
```

Parameters

- *initial_delay* Initial delay between a topology change and SPF calculation. Value ranges from 0 to 65535000 (ms). Default is 0 (ms).
- *hold_interval* Additional wait time after SPF calculation to allow the network to settle. If a topology change occurs during the hold interval, another SPF calculation is scheduled to occur after the hold interval expires. The next hold interval is doubled if topology changes occur during the hold interval. If doubling exceeds the maximum value, the maximum value is used instead. Value ranges from 0 to 65535000 (ms). Default is 5000 (ms).
- *max_interval* Maximum hold interval before the switch will perform an SPF calculation. Value ranges from 0 to 65535000 (ms). Default is 5000 (ms).

Example

- These commands set the SPF timers on the switch.

```
switch(config)#router ospf 6
switch(config-router-ospf)#timers spf 5 100 20000
switch(config-router-ospf)#
```

Chapter 28

Open Shortest Path First – Version 3

Open Shortest Path First (OSPF) is a link-state routing protocol that operates within a single autonomous system. OSPF version 3 is defined by RFC 5340.

This chapter contains the following sections.

- Section 28.1: OSPFv3 Introduction
- Section 28.2: OSPFv3 Conceptual Overview
- Section 28.3: Configuring OSPFv3
- Section 28.4: OSPFv3 Examples
- Section 28.5: OSPFv3 Commands

28.1 OSPFv3 Introduction

OSPFv3 is based on OSPF version 2 and includes enhancements that utilize IPv6 features. However, OSPFv3 is configured and operates independently of any implementation of OSPFv2 on the switch. OSPFv2 features that OSPFv3 implements include:

- Packet types
- Neighbor discovery and adjacency formation mechanisms
- LSA aging and flooding
- SPF calculations
- DR election procedure
- Multiple area support
- Router-ID (32 bits)

The following list describes the OSPFv3 differences and enhancements from OSPFv2:

- IPv6 128-bit addresses
- Use of link-local addresses
- OSPFv3 runs over links instead of subnets

Arista switches support the following OSPFv3 functions:

- A single OSPFv3 instance for each VRF
- Intra- and inter-area routing
- Type 1 and 2 external routing
- Broadcast and P2P interfaces
- Stub areas
- Redistribution of static and connected routes into OSPFv3

28.2 OSPFv3 Conceptual Overview

28.2.1 Storing Link States

OSPFv3 is a dynamic, link-state routing protocol, where links represent routable paths. Dynamic routing protocols calculate the most efficient path between locations based on bandwidth and device status.

A link state advertisement (LSA) is an OSPFv3 packet that communicates a router's topology to other routers. The link state database (LSDB) stores an area’s topology database and is composed of LSAs received from other routers. Routers update the LSDB by storing LSAs from other routers.

28.2.2 Topology

An autonomous system (AS) is the IP domain within which a dynamic protocol controls the routing of traffic. In OSPFv3, an AS is composed of areas, which define the LSDB computation boundaries. All routers in an area store identical LSDBs. Routers in different areas exchange updates without storing the entire database, reducing information maintenance on large, dynamic networks.

An AS shares internal routing information from its areas and external routing information from other processes to inform routers outside the AS about routes the network can access. Routers that advertise routes on other ASs commit to carry data to the IP space on the route.

OSPFv3 defines these routers:

- Internal router (IR) – a router whose interfaces are contained in a single area. All IRs in an area maintain identical LSDBs.
- Area border router (ABR) – a router that has interfaces in multiple areas. ABRs maintain one LSDB for each connected area.
- Autonomous system boundary router (ASBR) – a gateway router connecting the OSPFv3 domain to external routes, including static routes and routes from other autonomous systems.

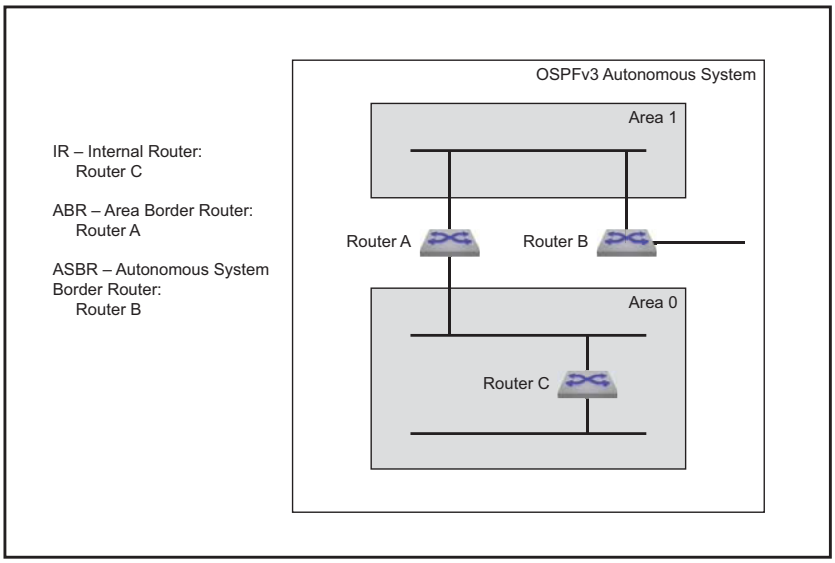
Figure 28-1 displays the OSPFv3 router types.

OSPFv3 areas are assigned a number between 0 and 4,294,967,295. Area numbers are often expressed in dotted decimal notation, similar to IP addresses.

Each AS has a backbone area, designated as area 0, that connects to all other areas. The backbone receives routing information from all areas, then distributes it to the other areas as required.

OSPFv3 area types include:

Figure 28-1: OSPFv3 Router Types



- Normal area – accepts intra-area, inter-area, and external routes. The backbone is a normal area.
- Stub area – does not receive router advertisements external to the AS. Stub area routing is based on a default route.

28.2.3 Link Updates

Routers periodically send hello packets to advertise status and establish neighbors. A router's hello packet includes IP addresses of other routers from which it received a hello packet within the time specified by the router dead interval. Routers become neighbors when they detect each other in their hello packets if they:

- share a common network segment.
- are in the same area.
- have the same hello interval, dead interval, and authentication parameters.

Neighbors form adjacencies to exchange LSDB information. A neighbor group uses hello packets to elect a Designated Router (DR) and Backup Designated Router (BDR). The DR and BDR become adjacent to all other neighbors, including each other. Only adjacent neighbors share database information.

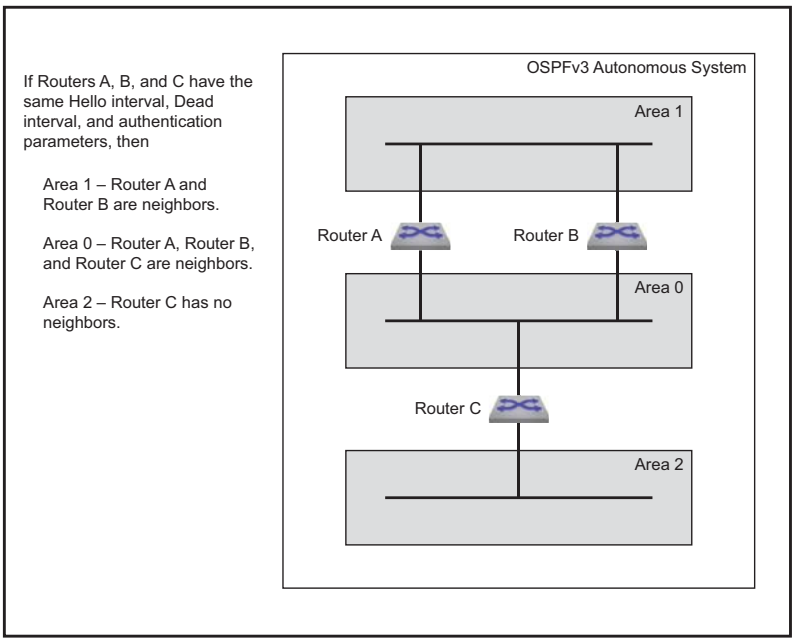
Figure 28-2 illustrates OSPFv3 neighbors.

The DR is the central contact for database exchanges. Switches send database information to their DR, which relays the information to the other neighbors. All routers in an area maintain identical LSDBs. Switches also send database information to their BDR, which stores this data without distributing it. If the DR fails, the BDR distributes LSDB information to its neighbors.

OSPFv3 routers distribute LSAs by sending them on all of their active interfaces. The router does not send hello packets from passive interfaces preventing adjacencies. The router does not process any OSPFv2 packets received on a passive interface.

When a router's LSDB is changed by an LSA, it sends the changes to the BDR and DR for distribution to the other neighbors. Routing information is updated only when the topology changes.

Figure 28-2: OSPFv3 Neighbors



Routing devices use Dijkstra's algorithm to calculate the shortest path to all known destinations, based on cumulative route cost. The cost of an interface indicates the transmission overhead and is usually inversely proportional to its bandwidth.

28.3 Configuring OSPFv3

These sections describe basic OSPFv3 configuration steps:

- Section 28.3.1: Configuring an OSPFv3 Instance
- Section 28.3.2: Configuring OSPFv3 Areas
- Section 28.3.3: Configuring Interfaces for OSPFv3
- Section 28.3.4: Enabling OSPFv3
- Section 28.3.5: Displaying OSPFv3 Status

28.3.1 Configuring an OSPFv3 Instance

28.3.1.1 Entering OSPFv3 Configuration Mode

OSPFv3 configuration commands apply to the specified OSPFv3 instance. To perform OSPFv3 configuration commands, the switch must be in router-OSPFv3 configuration mode. The **ipv6 router ospf** command places the switch in router-OSPFv3 configuration mode, creating an OSPFv3 instance if OSPFv3 was not previously instantiated on the switch. If no VRF is specified, the OSPFv3 instance is in the default VRF. To instantiate or configure OSPFv3 on a non-default VRF, specify that VRF when using the **ipv6 router ospf** command.

The process ID identifies the OSPFv3 instance and is local to the router. Neighbor OSPFv3 routers can have different process IDs. OSPFv3 instances configured in different VRFs on the switch must have different process IDs.

The switch supports one OSPFv3 instance for each VRF. When an OSPFv3 instance already exists, the **ipv6 router ospf** command must specify its process ID (and VRF, if it is not configured in the default VRF). Attempts to define additional instances in the same VRF will generate errors. The **show ipv6 ospf** command displays information about OSPFv3 instances, including their process IDs.

Example

- This command places the switch in router-OSPFv3 configuration mode for the default VRF. If OSPFv3 was not previously instantiated in the default VRF, the command creates an OSPFv3 instance in the default VRF with a process ID of 9.

```
switch(config)#ipv6 router ospf 9
switch(config-router-ospf3)#show active
ipv6 router ospf 9
switch(config-router-ospf3)#
```

28.3.1.2 Defining the Router ID

The router ID is a 32-bit number assigned to a router running OSPFv3. This number uniquely labels the router within an Autonomous System. Status commands identify the switch through the router ID. When configuring OSPFv3 instances in multiple VRFs, each should have a different router ID.

The switch sets the router ID to the first available alternative in the following list:

1. The **router-id** command.
2. The loopback IPv6 address, if a loopback interface is active on the switch.
3. The highest IPv6 address on the router.

Important! When configuring VXLAN on an MLAG, always manually configure the OSPFv3 router ID to prevent the switch from using the common VTEP IP address as the router ID.

The **router-id (OSPFv3)** command configures the router ID for an OSPFv3 instance.

Example

- This command assigns 15.1.1.1 as the OSPFv3 router ID.

```
switch(config-router-ospf3)#router-id 15.21.4.9
switch(config-router-ospf3)#show active
ipv6 router ospf 9
   router-id 15.21.4.9
switch(config-router-ospf3)#
```

28.3.1.3 Global OSPFv3 Parameters

These router-OSPFv3 configuration mode commands define OSPFv3 behavior for the OSPFv3 instance under which they are used.

Logging Adjacency Changes

The **log-adjacency-changes (OSPFv3)** command configures the switch to log OSPFv3 link-state changes and transitions of OSPFv3 neighbors into the up or down state.

Examples

- This command configures the switch to log transitions of OSPFv3 neighbors into the up or down state.

```
switch(config-router-ospf3)#log-adjacency-changes
switch(config-router-ospf3)#
```

- This command configures the switch to log all OSPFv3 link-state changes.

```
switch(config-router-ospf3)#log-adjacency-changes detail
switch(config-router-ospf3)#
```

Intra-Area Distance

The **distance ospf intra-area (OSPFv3)** command configures the administrative distance for routes contained in a single OSPFv3 area. Administrative distances compare dynamic routes configured by different protocols. The default administrative distance for intra-area routes is 10.

Example

- This command configures an administrative distance of 90 for OSPFv3 intra-area routes.

```
switch(config-router-ospf3)#distance ospf intra-area 90
switch(config-router-ospf3)#show active
ipv6 router ospf 9
   distance ospf intra-area 90
switch(config-router-ospf3)#
```

Passive Interfaces

The **passive-interface (OSPFv3)** command prevents the transmission of hello packets on the specified interface. Passive interfaces drop all adjacencies and do not form new adjacencies. Although passive interfaces do not send or receive LSAs, other interfaces may generate LSAs for the network segment. The router does not send OSPFv3 packets from a passive interface or process OSPFv3 packets received on a passive interface. The router advertises the passive interface in the router LSA.

The **no passive-interface** command re-enables OSPFv3 processing on the specified interface.

Examples

- This command configures VLAN 200 as a passive interface.

```
switch(config-router-ospf3)#passive-interface vlan 200
switch(config-router-ospf3)#show active
ipv6 router ospf 9
   passive-interface Vlan200
switch(config-router-ospf3)#
```

- This command configures VLAN 200 as an active interface.

```
switch(config-router-ospf3)#no passive-interface vlan 200
switch(config-router-ospf3)#show active
ipv6 router ospf 9
switch(config-router-ospf3)#
```

Redistributing Connected Routes

Redistributing connected routes causes the OSPFv3 instance to advertise all connected routes on the switch as external OSPFv3 routes. Connected routes are routes that are established when IPv6 is enabled on an interface.

Example

- The **redistribute (OSPFv3) connected** command converts connected routes to OSPFv3 external routes.

```
switch(config-router-ospf3)#redistribute connected
switch(config-router-ospf3)#show active
ipv6 router ospf 9
   redistribute connected
switch(config-router-ospf3)#
```

Redistributing Static Routes

Redistributing static routes causes the OSPFv3 instance to advertise all static routes on the switch as external OSPFv3 routes. The switch does not support redistributing individual static routes.

Example

- The **redistribute (OSPFv3) static** command converts static routes to OSPFv3 external routes.

```
switch(config-router-ospf3)#redistribute static
switch(config-router-ospf3)#show active
ipv6 router ospf 9
   redistribute static
switch(config-router-ospf3)#
```

28.3.2 Configuring OSPFv3 Areas

OSPFv3 areas are configured through area commands. The switch must be in router-OSPFv3 configuration mode, as described in Section 28.3.1.1: Entering OSPFv3 Configuration Mode, to run area commands.

Areas are assigned a 32-bit number that is expressed in decimal or dotted-decimal notation. When an OSPFv3 instance configuration contains multiple areas, the switch only configures areas associated with its interfaces.

28.3.2.1 Configuring the Area Type

The **no area (OSPFv3)** command specifies the area type. The switch supports three area types:

- Normal area: Area that accepts intra-area, inter-area, and external routes. The backbone area (area 0) is a normal area.
- Stub area: Area where external routes are not advertised. External routes are reached through a default summary route (0.0.0.0) inserted into stub areas. Networks with no external routes do not require stub areas.

The default area type is normal.

Example

- These commands configures area 200 as a NSSA area and 300 as a stub area.

```
switch(config)#ipv6 router ospf 9
switch(config-router-ospf3)#area 200 nssa
switch(config-router-ospf3)#area 300 stub
switch(config-router-ospf3)#show active
ipv6 router ospf 9
   area 0.0.0.200
   area 0.0.1.44 stub
switch(config-router-ospf3)#
```

28.3.2.2 Configuring Area Parameters

These router-OSPFv3 configuration mode commands define OSPFv3 behavior in a specified area.

Default Summary Route Cost

The **area default-cost (OSPFv3)** command specifies the cost of the default summary route that ABRs send into a stub area or NSSA. Summary routes, also called inter-area routes, originate in areas different than their destination. When the **area default-cost** command is not configured for an area, the default-cost of that area is set to 10.

Example

- This command configures a cost of 25 for the default summary route in area 0.0.1.194 (450).

```
switch(config-router-ospf3)#area 450 default-cost 25
switch(config-router-ospf3)#show active
ipv6 router ospf 9
   area 0.0.1.194 default-cost 25
```

Area Stub

The **area stub (OSPFv3)** command configures the area type of an OSPFv3 area. All routers in an AS must specify the same area type for identically numbered areas.

Stub areas are areas in which external routes are not advertised. To reach these external routes, the stub area uses a default summary route (0.0.0.0). Networks without external routes do not require stub areas.

Areas are ***normal*** by default; area type configuration is required only for stub NSSA areas. Area 0 is always a normal area and cannot be configured through this command.

Examples

- This command configures area 45 as a stub area.

```
switch(config)#ipv6 router ospf 3
switch(config-router-ospf3)#area 45 stub
switch(config-router-ospf3)#
```

- This command configures area 10.92.148.17 as a stub area.

```
switch(config-router-ospf3)#area 10.92.148.17 stub
switch(config-router-ospf3)#
```

Area Range

The **area range (OSPFv3)** command is used by OSPFv3 area border routers (ABRs) to consolidate or summarize routes, to configure a cost setting for those routes, and to suppress summary route advertisements.

By default, an ABR creates a summary LSA for each route in an area and advertises that LSA to adjacent areas. The **area range (OSPFv3)** command aggregates routing information on area boundaries, allowing the ABR to use one summary LSA to advertise multiple routes.

Examples

- These commands consolidate and summarize routes at an area boundary 1.

```
switch(config)#ipv6 router ospf 1
switch(config-router-ospf3)#area 1 range 2001:0DB8:0:1::/64
switch(config-router-ospf3)#
```

- These commands change the address range status to DoNotAdvertise. Neither one of the individual intra-area routes falling under range or the ranged prefix is advertised as summary LSA.

```
switch(config)#ipv6 router ospf 1
switch(config-router-ospf3)#area 1 range 2001:0DB8:0:1::/64 not-advertise
switch(config-router-ospf3)#
```

28.3.3 Configuring Interfaces for OSPFv3

OSPFv3 interface configuration commands enable OSPFv3 on an interface, assign the interface to an area, and specify transmission parameters for routed ports and SVIs that handle OSPFv3 packets.

28.3.3.1 Assigning an Interface to an Area

The **ipv6 ospf area** command enables OSPFv3 on the configuration mode interface and associates the specified area to the interface. Each routed interface can be associated with one OSPFv3 area; subsequent **ipv6 ospf area** commands that designate a different area on an interface replace any existing command for the interface.

Example

- These commands enable OSPFv3 instance 9 on VLAN interface 200 and associate area 0 to the interface.

```
switch(config)#interface vlan 200
switch(config-if-Vl200)#ipv6 ospf 9 area 0
switch(config-if-Vl200)#show active
interface Vlan200
   ipv6 ospf 9 area 0.0.0.0
switch(config-if-Vl200)#
```

28.3.3.2 Configuring Intervals

Interval configuration commands determine OSPFv3 packet transmission characteristics for a specified VLAN interface. Interval configuration commands are entered in vlan-interface configuration mode.

Hello Interval

The hello interval specifies the period between consecutive hello packet transmissions from an interface. Each OSPFv3 neighbor should specify the same hello interval, which should not be longer than any neighbor's dead interval.

The **ipv6 ospf hello-interval** command configures the hello interval for the configuration mode interface. The default is 10 seconds.

Example

- These commands configure a hello interval of 45 seconds for VLAN 200.

```
switch(config)#interface vlan 200
switch(config-if-Vl200)#ipv6 ospf hello-interval 45
switch(config-if-Vl200)#show active
interface Vlan200
   ipv6 ospf hello-interval 45
switch(config-if-Vl200)#
```

Dead Interval

The dead interval specifies the period that an interface waits for an OSPFv3 packet from a neighbor before it disables the adjacency under the assumption that the neighbor is down. The dead interval should be configured identically on all OSPFv3 neighbors and be longer than the hello interval of any neighbor.

The **ipv6 ospf dead-interval** command configures the dead interval for the configuration mode interface. The default is 40 seconds.

Example

- This command configures a dead interval of 75 seconds for VLAN 200.

```
switch(config)#interface vlan 200
switch(config-if-Vl200)#ipv6 ospf dead-interval 75
switch(config-if-Vl200)#show active
interface Vlan200
   ipv6 ospf dead-interval 75
switch(config-if-Vl200)#
```

Retransmission Interval

Routers that send OSPFv3 advertisements to an adjacent router expect to receive an acknowledgment from that neighbor. Routers that do not receive an acknowledgment will retransmit the advertisement. The retransmission interval specifies the period between retransmissions.

The **ipv6 ospf retransmit-interval** command configures the LSA retransmission interval for the configuration mode interface. The default retransmission interval is 5 seconds.

Example

- This command configures a retransmission interval of 25 seconds for VLAN 200.

```
switch(config)#interface vlan 200
switch(config-if-Vl200)#ipv6 ospf retransmit-interval 25
switch(config-if-Vl200)#show active
interface Vlan200
   ipv6 ospf retransmit-interval 25
switch(config-if-Vl200)#
```

Transmission Delay

The transmission delay is an estimate of the time that an interface requires to transmit a link-state update packet. OSPFv3 adds this delay to the age of outbound packets to more accurately reflect the age of the LSA when received by a neighbor.

The **ipv6 ospf transmit-delay** command configures the transmission delay for the configuration mode interface. The default transmission delay is one second.

Example

- This command configures a transmission delay of 10 seconds for VLAN 200.

```
switch(config)#interface vlan 200
switch(config-if-Vl200)#ipv6 ospf transmit-delay 10
switch(config-if-Vl200)#show active
interface Vlan200
   ipv6 ospf transmit-delay 10
switch(config-if-Vl200)#
```

28.3.3.3 Configuring Interface Parameters

Interface Cost

The OSPFv3 interface cost reflects the overhead of sending packets across the interface. The cost is typically assigned to be inversely proportional to the bandwidth of the interface. The **ipv6 ospf cost** command configures the OSPFv3 cost for the configuration mode interface. The default cost is 10.

Example

- This command configures a cost of 50 for VLAN 200.

```
switch(config)#interface vlan 200
switch(config-if-Vl200)#ipv6 ospf cost 50
switch(config-if-Vl200)#show active
interface Vlan200
   ipv6 ospf cost 50
switch(config-if-Vl200)#
```

Router Priority

Router priority determines preference during designated router (DR) and backup designated router (BDR) elections. Routers with higher priority numbers have preference over other routers. Routers with a priority of zero cannot be elected as a DR or BDR.

The **ipv6 ospf priority** command configures router priority for the configuration mode interface. The default priority is 1.

Example

- This command configures a router priority of 128 for VLAN 200.

```
switch(config)#interface vlan 200
switch(config-if-Vl200)#ipv6 ospf priority 128
switch(config-if-Vl200)#show active
interface Vlan200
   ipv6 ospf priority 128
switch(config-if-Vl200)#
```

28.3.4 Enabling OSPFv3

28.3.4.1 IP Routing

OSPFv3 requires that IPv6 unicast routing is enabled on the switch. When IP routing is not enabled, entering OSPFv3 configuration mode generates a message.

Examples

- This message is displayed if, when entering router-OSPFv3 configuration mode, IPv6 unicast routing is not enabled.

```
switch(config)#ipv6 router ospf 9
! IPv6 routing not enabled
switch(config-router-ospf3)#
```

- This command enables IP routing on the switch.

```
switch(config)#ipv6 unicast-routing
```

28.3.4.2 Disabling OSPFv3

The **shutdown (OSPFv3)** disables OSPFv3 operations on the switch without disrupting the OSPFv3 configuration. To disable OSPFv3 on an interface, remove the **ipv6 ospf area** statement for the corresponding interface.

The **no shutdown** command resumes OSPFv3 activity.

Examples

- This command disables OSPFv3 activity on the switch.

```
switch(config)#ipv6 router ospf 9
switch(config-router-ospf3)#shutdown
switch(config-router-ospf3)#show active
ipv6 router ospf 9
   shutdown
switch(config-router-ospf3)#
```

- This command resumes OSPFv3 activity on the switch.

```
switch(config-router-ospf3)#no shutdown
switch(config-router-ospf3)#show active
ipv6 router ospf 9
switch(config-router-ospf3)#
```

28.3.5 Displaying OSPFv3 Status

This section describes OSPFv3 **show** commands that display OSPFv3 status. General switch methods that provide OSPFv3 information include pinging routes, viewing route status (**show ip route** command), and viewing the configuration (**show running-config** command).

28.3.5.1 OSPFv3 Summary

The **show ipv6 ospf** command displays general OSPFv3 configuration information, operational statistics and status for the OSPFv3 instance, followed by a brief description of the areas configured on the switch.

Example

- This command displays OSPFv3 routing process information.

```
switch>show ipv6 ospf
Routing Process "ospfv3 1" with ID 10.37.0.23 and Instance 0
  It is an autonomous system boundary router and is an area border router
  Hold time between two SPFs is 5
  Minimum LSA interval 5. Minimum LSA arrival 1
  It has 13 fully adjacent neighbors
  Number of areas in this router is 2. 2 normal, 0 stub, 0 nssa
  Graceful restart is enabled
    Grace period is 40
    Strict helper is enabled
  SPF algorithm last executed 00:02:59 ago
  Area 0.0.0.0
    Number of interface in this area is 8
    It is a normal area
  Area 0.0.0.2
    Number of interface in this area is 1
    It is a normal area
```

28.3.5.2 Viewing OSPFv3 on the Interfaces

The **show ipv6 ospf interface** command displays OSPFv3 information for switch interfaces configured for OSPFv3. Different command options allow the display of either all interfaces or a specified interface. The command can also be configured to display complete information or a brief summary.

Example

- This command displays OSPFv3 information for interfaces where OSPFv3 is enabled.

```
switch#show ipv6 ospf interface
Ethernet17 is up
  Interface Address fe80::48c:73ff:fe00:1319%Ethernet12, Area 0.0.0.0
  Network Type Broadcast, Cost 10
  Transmit Delay is 1 sec, State Backup DR, Priority 1
  Designated Router is 10.37.0.37
  Backup Designated Router is 10.37.0.23
  Timer intervals configured, Hello 10, Dead 40, Retransmit 5
  Neighbor Count is 1
Vlan31 is up
  Interface Address fe80::48c:73ff:fe00:1319%Vlan31, Area 0.0.0.0
  Network Type Broadcast, Cost 10
  Transmit Delay is 1 sec, State Backup DR, Priority 1
  Designated Router is 10.37.0.22
  Backup Designated Router is 10.37.0.23
  Timer intervals configured, Hello 10, Dead 40, Retransmit 5
  Neighbor Count is 1
Vlan32 is up
  Interface Address fe80::48c:73ff:fe00:1319%Vlan32, Area 0.0.0.0
  Network Type Broadcast, Cost 10
  Transmit Delay is 1 sec, State DR Other, Priority 1
  Designated Router is 10.37.0.11
  Backup Designated Router is 10.37.0.22
  Timer intervals configured, Hello 10, Dead 40, Retransmit 5
  Neighbor Count is 2
switch#
```

28.3.5.3 Viewing the OSPFv3 Database

The **show ipv6 ospf database <link state list>** command displays the LSAs in the LSDB for the specified area. If no area is listed, the command displays the contents of the database for each area on the switch. The database command provides options to display subsets of the LSDB database, a summary of database contents, and the link states that comprise the database.

Example

- This command displays the OSPFv3 database of link state advertisements (LSAs).

```
switch#show ipv6 ospf database
Routing Process "ospf 9":

  AS Scope LSDB

Type        Link ID     ADV Router  Age       Seq#   Checksum
 AEX        0.0.0.5    10.37.0.37   15 0x80000005   0x00be82
 AEX        0.0.0.9    10.37.0.22 1747 0x8000002b   0x00df56
 AEX        0.0.0.3    10.37.0.46  599 0x8000002d   0x00651d

Area 0.0.0.0 LSDB

Type        Link ID     ADV Router  Age       Seq#   Checksum
 RTR        0.0.0.0    10.37.0.32  234 0x80000031   0x00585a
 NTW       0.0.0.26    10.37.0.32  271 0x80000005   0x005609
 NAP       0.0.0.26    10.37.0.32  274 0x80000005   0x00964c

Interface vlan3911 LSDB

Type        Link ID     ADV Router  Age       Seq#   Checksum
 LNK       0.0.0.38    10.37.0.22  267 0x80000005   0x00a45a
 LNK       0.0.0.23    10.37.0.23  270 0x8000002c   0x005b7e

  Interface vlan3902 LSDB

Type        Link ID     ADV Router  Age       Seq#   Checksum
 LNK       0.0.0.17    10.37.0.11 1535 0x8000002b   0x007120
 LNK       0.0.0.37    10.37.0.22    7 0x8000002b   0x00ce23
 LNK       0.0.0.22    10.37.0.23  250 0x8000002d   0x00c350
            <-------OUTPUT OMITTED FROM EXAMPLE-------->
switch#
```

28.3.5.4 Viewing OSPFv3 Neighbors

The **show ipv6 ospf neighbor** command displays information about the routers that are neighbors to the switch. Command options allow the display of summary or detailed information about the neighbors to all areas and interfaces on the switch. The command also allows for the display of neighbors to individual interfaces or areas. The ***adjacency-changes*** option displays the interface's adjacency changes.

Example

- This command displays the switch's neighbors.

```
switch#show ipv6 ospf neighbor
Routing Process "ospf 9":
Neighbor 10.37.0.37 priority is 1, state is Full
  In area 0.0.0.0 interface et12
  DR is 10.37.0.37 BDR is 10.37.0.23
  Options is 0
  Dead timer is due in 37 seconds
Neighbor 10.37.0.22 priority is 1, state is Full
  In area 0.0.0.0 interface vlan3911
  DR is 10.37.0.22 BDR is 10.37.0.23
  Options is 0
  Dead timer is due in 31 seconds
Neighbor 10.37.0.22 priority is 1, state is Full
  In area 0.0.0.0 interface vlan3902
  DR is 10.37.0.11 BDR is 10.37.0.22
  Options is 0
  Dead timer is due in 31 seconds
Neighbor 10.37.0.22 priority is 1, state is Full
  In area 0.0.0.0 interface vlan3908
  DR is 10.37.0.22 BDR is 10.37.0.21
  Options is 0
  Dead timer is due in 39 seconds

switch#
```

28.3.5.5 Viewing OSPFv3 Routes

The **show ipv6 routes** command provides an OSPFv3 option.

Example

- This command displays the switch's OSPFv3 routes.

```
switch# show ipv6 route ospf
IPv6 Routing Table - 43 entries
Codes: C - connected, S - static, K - kernel, O - OSPF, B - BGP, R - RIP, A -
Aggregate

O   fd7a:3279:81a4:1112::/64 [150/11]
     via fe80::21c:41ff:fe00:d120, Ethernet12
O   fd7a:3279:81a4:1114::/64 [150/11]
     via fe80::21c:41ff:fe00:d120, Ethernet12
O   fd7a:3279:81a4:1124::/64 [10/20]
     via fe80::21c:41ff:fe01:5fe1, Vlan3901
     via fe80::21c:41ff:fe01:5fe1, Vlan3902
     via fe80::21c:41ff:fe01:5fe1, Vlan3908
O   fd7a:3279:81a4:1a00::25/128 [150/11]
     via fe80::21c:41ff:fe00:d120, Ethernet12
O   fd7a:3279:81a4:1a00::28/128 [150/11]
     via fd7a:3279:81a4:fe40::5, Vlan3908
```

28.4 OSPFv3 Examples

This section describes the commands required to configure three OSPFv3 topologies.

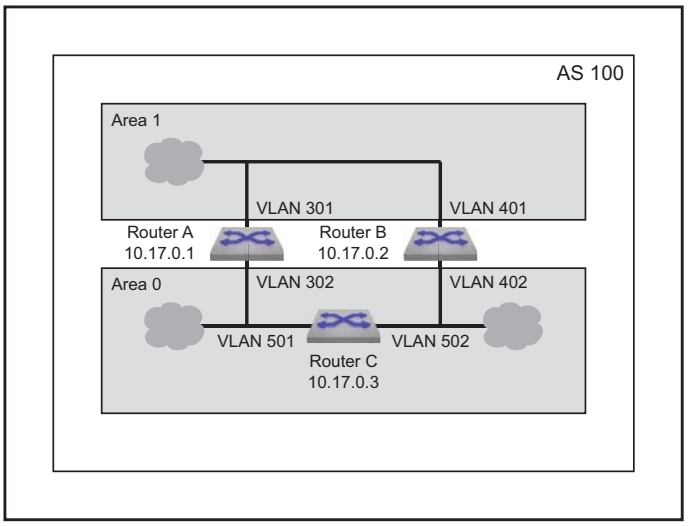
28.4.1 OSPFv3 Example 1

The AS in Example 1 contains two areas that are connected through two routers. The backbone area also contains an internal router that connects two links.

28.4.1.1 Example 1 Diagram

Figure 28-3 displays the Example 1 topology. Two ABRs connect area 0 and area 1 – Router A and Router B. Router C is an internal router that connects two links in area 0. Area 0 is normal; area 1 is stub.

Figure 28-3: OSPFv3 Example 1



Area 1 Configuration

Area 1 contains links to ABRs Router A and Router B.

- Router A is accessed through VLAN 301.
- Router B is accessed through VLAN 401.
- Designated Router (DR): Router A.
- Backup Designated Router (BDR): Router B.
- Each router defines an interface cost of 10.
- Router priority is not specified for either router on area 1.

Area 0 ABR Configuration

Area 0 contains links to ABRs Router A and Router B.

- Router A is accessed through VLAN 302.
- Router B is accessed through VLAN 402.
- Designated Router (DR): Router B.
- Backup Designated Router (BDR): Router A.

- Each router defines an interface cost of 20.
- Each router defines a retransmit-interval of 10.
- Each router defines a transmit-delay of 2.
- Router priority is specified such that Router B will be elected as the Designated Router.

Area 0 IR Configuration

Area 0 contains two links to an internal router.

- Router C is accessed through VLAN 501 and VLAN 502.
- VLAN 501 is configured as follows:
  - Interface cost of 20.
  - Retransmit-interval of 10.
  - Transmit-delay of 2.
- VLAN 502 is configured as follows:
  - Interface cost of 20.
  - Dead interval of 80 seconds.

28.4.1.2 Example 1 Code

This code configures the OSPFv3 instances on the three switches.

Step 1 Configure the areas and router IDs.

a Router A OSPFv3 instance configuration:

```
switch-A(config)#ipv6 router ospf 100
switch-A(config-router-ospfv3)#area 1 stub
switch-A(config-router-ospfv3)#router-id 10.17.0.1
```

b Router B OSPFv3 instance configuration:

```
switch-B(config)#ipv6 router ospf 100
switch-B(config-router-ospfv3)#area 1 stub
switch-B(config-router-ospfv3)#router-id 10.17.0.2
```

c Router C OSPFv3 instance configuration: interfaces:

```
switch-C(config)#ipv6 router ospf 100
switch-C(config-router-ospfv3)#router-id 10.17.0.3
```

Step 2 Configure the interface OSPFv3 area and transmission parameters.

a Router A interfaces:

```
switch-A(config)#interface vlan 301
switch-A(config-if-Vl301)#ipv6 ospf 100 area 1
switch-A(config-if-Vl301)#ipv6 ospf cost 10
switch-A(config-if-Vl301)#ipv6 ospf priority 6
switch-A(config-if-Vl301)#exit
switch-A(config)#interface vlan 302
switch-A(config-if-Vl302)#ipv6 ospf 100 area 0
switch-A(config-if-Vl302)#ipv6 ospf cost 20
switch-A(config-if-Vl302)#ipv6 ospf retransmit-interval 10
switch-A(config-if-Vl302)#ipv6 ospf transmit-delay 2
switch-A(config-if-Vl302)#ipv6 ospf priority 4
```

b Router B interfaces:

```
switch-B(config)#interface vlan 401
switch-B(config-if-Vl401)#ipv6 ospf 100 area 1
switch-B(config-if-Vl401)#ipv6 ospf cost 10
switch-B(config-if-Vl401)#ipv6 ospf priority 4
switch-B(config-if-Vl401)#exit
switch-B(config)#interface vlan 402
switch-B(config-if-Vl402)#ipv6 ospf 100 area 0
switch-B(config-if-Vl402)#ipv6 ospf cost 20
switch-B(config-if-Vl402)#ipv6 ospf retransmit-interval 10
switch-B(config-if-Vl402)#ipv6 ospf transmit-delay 2
switch-B(config-if-Vl402)#ipv6 ospf priority 6
```

c Router C interfaces:

```
switch-C(config)#interface vlan 501
switch-C(config-if-Vl501)#ipv6 ospf 100 area 0
switch-C(config-if-Vl501)#ipv6 ospf cost 20
switch-C(config-if-Vl501)#ipv6 ospf retransmit-interval 10
switch-C(config-if-Vl501)#ipv6 ospf transmit-delay 2
switch-C(config-if-Vl501)#exit
switch-C(config)#interface vlan 502
switch-C(config-if-Vl502)#ipv6 ospf 100 area 0
switch-C(config-if-Vl502)#ipv6 ospf cost 20
switch-C(config-if-Vl502)#ipv6 ospf dead-interval 80
```

28.4.2 OSPFv3 Example 2

The AS in Example 2 contains three areas. Area 0 connects to the other areas through different routers and contains an internal router connecting two links. Area 0 is normal; the other areas are stub areas.

28.4.2.1 Example 2 Diagram

Figure 28-4 displays the Example 2 topology. One ABR (Router B) connects area 0 and area 1; another ABR (router C) connects area 0 and area 2. Router A is an internal router that connects two links in area 0.

Area 1 Configuration

Area 1 contains one link that is accessed by Router B.

- Router B is accessed through VLAN 601.
- The router defines a interface cost of 10.

Area 2 Configuration

Area 2 contains one link that is accessed by Router C.

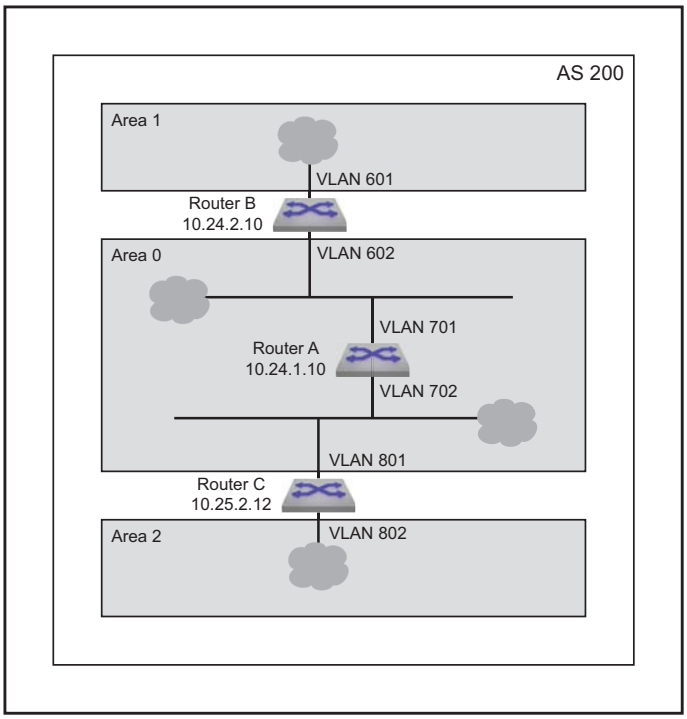
- Router C is accessed through VLAN 802.
- The router defines a interface cost of 20.

Area 0 ABR Configuration

One ABR Router B link connects area 1 to area 0. One ABR Router C link connects area 0 to area 2.

- Router B is accessed through VLAN 602.
- Router C is accessed through VLAN 801.
- Designated Router (DR): Router B.
- Backup Designated Router (BDR): Router C.

Figure 28-4: OSPFv3 Example 2



- Each router defines an interface cost of 20.
- Each router defines a retransmit-interval of 10.
- Each router defines a transmit-delay of 2.

Area 0 IR Configuration

Area 0 contains links connected by an internal router.

- Router A is accessed through VLAN 701 and 702.
- The VLAN 701 link is configured as follows:
  - Interface cost of 10.
- The VLAN 702 link is configured as follows:
  - Interface cost of 20.
  - Retransmit-interval of 10.
  - Transmit-delay of 2.

28.4.2.2 Example 2 Code

Step 1 Configure the areas and router IDs.

a Router A OSPFv3 instance configuration:

```
switch-A(config)#ipv6 router ospf 200
switch-A(config-router-ospfv3)#router-id 10.24.1.10
```

b Router B OSPFv3 instance configuration:

```
switch-B(config)#ipv6 router ospf 200
switch-B(config-router-ospfv3)#area 1 stub
switch-B(config-router-ospfv3)#router-id 10.24.2.10
```

c Router C OSPFv3 instance configuration:

```
switch-C(config)#ipv6 router ospf 200
switch-C(config-router-ospfv3)#area 1 stub
switch-C(config-router-ospfv3)#router-id 10.25.2.12
```

Step 2 Configure the interface OSPFv3 area and transmission parameters.

a Router A interfaces:

```
switch-A(config)#interface vlan 701
switch-A(config-if-Vl701)#ipv6 ospf 200 area 0
switch-A(config-if-Vl701)#ipv6 ospf cost 10
switch-A(config-if-Vl701)#exit
switch-A(config)#interface vlan 702
switch-A(config-if-Vl702)#ipv6 ospf 200 area 0
switch-A(config-if-Vl702)#ipv6 ospf cost 20
switch-A(config-if-Vl702)#ipv6 ospf retransmit-interval 10
switch-A(config-if-Vl702)#ipv6 ospf transmit-delay 2
```

b Router B interfaces:

```
switch-B(config)#interface vlan 601
switch-B(config-if-Vl601)#ipv6 ospf 200 area 1
switch-B(config-if-Vl601)#ipv6 ospf cost 10
switch-B(config-if-Vl601)#exit
switch-B(config)#interface vlan 602
switch-B(config-if-Vl602)#ipv6 ospf 200 area 0
switch-B(config-if-Vl602)#ipv6 ospf cost 20
switch-B(config-if-Vl602)#ipv6 ospf retransmit-interval 10
switch-B(config-if-Vl602)#ipv6 ospf transmit-delay 2
switch-B(config-if-Vl602)#ipv6 ospf priority 6
```

c Router C interfaces:

```
switch-C(config)#interface vlan 801
switch-C(config-if-Vl801)#ipv6 ospf 200 area 0
switch-C(config-if-Vl801)#ipv6 ospf cost 20
switch-C(config-if-Vl801)#ipv6 ospf retransmit-interval 10
switch-C(config-if-Vl801)#ipv6 ospf transmit-delay 2
switch-C(config-if-Vl801)#exit
switch-C(config)#interface vlan 802
switch-C(config-if-Vl802)#ipv6 ospf 200 area 2
switch-C(config-if-Vl802)#ipv6 ospf cost 20
switch-C(config-if-Vl802)#ipv6 ospf dead-interval 80
```

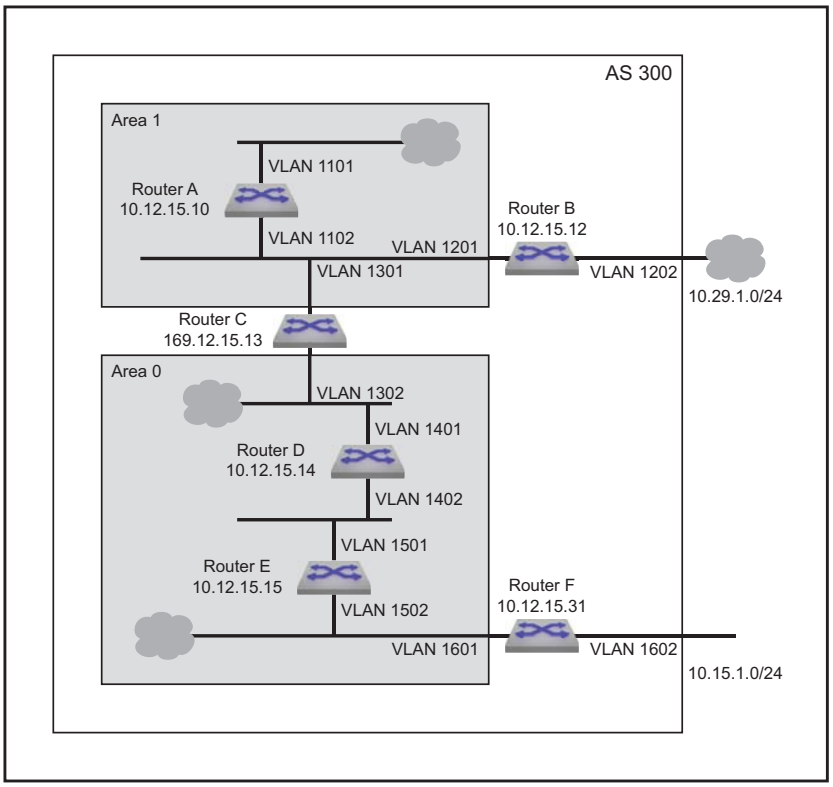
28.4.3 OSPFv3 Example 3

The AS in Example 3 contains two areas that connect through one ABR. Each area also contains an ASBR that connects static routes to the AS.

28.4.3.1 Example 3 Diagram

Figure 28-5 displays the Example 3 topology. One ABR connects area 0 and area 1. Router C is an ABR that connects the areas. Router A is an internal router that connects two links in area 1. Router D and Router E are internal routers that connect links in area 0. Router B and Router F are ASBRs that connect static routes outside the AS to area 1 and area 0, respectively.

Figure 28-5: OSPFv3 Example 3



Area 0 ABR Configuration

ABR Router C connects one area 0 link to an area 1 link.

- Router C is accessed through VLAN 1302.
- All interface OSPFv3 parameters are set to their default values.

Area 0 IR Configuration

Area 0 contains two internal routers, each of which connects two of the three links in the area.

- Router D is accessed through VLAN 1401 and VLAN 1402.
- Router E is accessed through VLAN 1501 and VLAN 1502.
- All interface OSPFv3 parameters are set to their default values.

Area 0 ASBR Configuration

ASBR Router F connects one area 0 link to an external link.

- Router F is accessed through VLAN 1601.
- Router F connects to the external AS through VLAN 1602.
- All interface OSPFv3 parameters are set to their default values.

Area 1 ABR Configuration

ABR Router C connects one area 0 link to an area 1 link.

- Router C is accessed by area 1 through VLAN 1301.

- Router C is accessed by area 0 through VLAN 1302.
- All interface OSPFv3 parameters are set to their default values.

Area 1 IR Configuration

Area 1 contains one internal router that connects two links in the area.

- Router A is accessed through VLAN 1101 and VLAN 1102.
- All interface OSPFv3 parameters are set to their default values.

Area 1 ASBR Configuration

ASBR Router B connects one area 1 link to an external link.

- Router B is access through VLAN 1201.
- Router B connects to the external AS through VLAN 1202.
- All interface OSPFv3 parameters are set to their default values.

28.4.3.2 Example 3 Code

Step 1 Configure the areas and router IDs.

a Router A OSPFv3 instance configuration:

```
switch-A(config)#ipv6 router ospf 300
switch-A(config-router-ospfv3)#router-id 10.12.15.10
switch-A(config-router-ospfv3)#area 1 stub
```

b Router B OSPFv3 instance configuration:

```
switch-B(config)#ipv6 router ospf 300
switch-B(config-router-ospfv3)#router-id 10.12.15.12
switch-B(config-router-ospfv3)#area 1 stub
```

c Router OSPFv3 instance configuration:

```
switch-C(config)#ipv6 router ospf 300
switch-C(config-router-ospfv3)#router-id 10.12.15.13
switch-C(config-router-ospfv3)#area 1 stub
```

d Router D OSPFv3 instance configuration:

```
switch-D(config)#ipv6 router ospf 300
switch-D(config-router-ospfv3)#router-id 10.12.15.14
```

e Router E OSPFv3 instance configuration:

```
switch-E(config)#ipv6 router ospf 300
switch-E(config-router-ospfv3)#router-id 10.12.15.15
```

f Router F OSPFv3 instance configuration:

```
switch-F(config)#ipv6 router ospf 300
switch-F(config-router-ospfv3)#router-id 10.12.15.31
```

Step 2 Configure the interfaces.

a Router A interfaces:

```
switch-A(config)#interface vlan 1101
switch-A(config-if-Vl1101)#ipv6 ospf 300 area 1
switch-A(config-if-Vl1101)#exit
switch-A(config)#interface vlan 1102
switch-A(config-if-Vl1102)#ipv6 ospf 300 area 1
```

b Router B interfaces:

```
switch-B(config)#interface vlan 1201
switch-B(config-if-Vl1201)#ipv6 ospf 300 area 1
switch-B(config-if-Vl1201)#exit
```

c Router C interfaces:

```
switch-C(config)#interface vlan 1301
switch-C(config-if-Vl1301)#ipv6 ospf 300 area 1
switch-C(config-if-Vl1301)#exit
switch-C(config)#interface vlan 1302
switch-C(config-if-Vl1302)#ipv6 ospf 300 area 0
```

d Router D interfaces:

```
switch-D(config)#interface vlan 1401
switch-D(config-if-Vl1401)#ipv6 ospf 300 area 0
switch-D(config-if-Vl1401)#exit
switch-D(config)#interface vlan 1402
switch-D(config-if-Vl1402)#ipv6 ospf 300 area 0
```

e Router E interfaces:

```
switch-E(config)#interface vlan 1501
switch-E(config-if-Vl1501)#ipv6 ospf 300 area 0
switch-E(config-if-Vl1501)#exit
switch-E(config)#interface vlan 1502
switch-E(config-if-Vl1502)#ipv6 ospf 300 area 0
```

f Router F interfaces:

```
switch-F(config)#interface vlan 1601
switch-F(config-if-Vl1601)#ipv6 ospf 300 area 0
switch-F(config-if-Vl1601)#exit
```

28.5 OSPFv3 Commands

Global Configuration Mode

- ipv6 router ospf
- clear ipv6 ospf force-spf

Interface Configuration Mode

- ipv6 ospf area
- ipv6 ospf cost
- ipv6 ospf dead-interval
- ipv6 ospf hello-interval
- ipv6 ospf network
- ipv6 ospf priority
- ipv6 ospf retransmit-interval
- ipv6 ospf transmit-delay

Router-OSPFv3 Configuration Mode

- adjacency exchange-start threshold (OSPFv3)
- area default-cost (OSPFv3)
- area nssa (OSPFv3)
- area nssa default-information-originate (OSPFv3)
- area nssa translate type7 always (OSPFv3)
- area range (OSPFv3)
- area stub (OSPFv3)
- default-information originate (OSPFv3)
- default-metric (OSPFv3)
- distance ospf intra-area (OSPFv3)
- log-adjacency-changes (OSPFv3)
- max-metric router-lsa (OSPFv3)
- maximum-paths (OSPFv3)
- no area (OSPFv3)
- passive-interface (OSPFv3)
- redistribute (OSPFv3)
- router-id (OSPFv3)
- shutdown (OSPFv3)

Display Commands

- show ipv6 ospf
- show ipv6 ospf border-routers
- show ipv6 ospf database
- show ipv6 ospf database<link-state details>
- show ipv6 ospf database <link state list>
- show ipv6 ospf database link
- show ipv6 ospf database link if-name
- show ipv6 ospf database link if-type
- show ipv6 ospf interface
- show ipv6 ospf lsa-log
- show ipv6 ospf neighbor
- show ipv6 ospf neighbor state
- show ipv6 ospf neighbor summary
- show ipv6 ospf spf-log

adjacency exchange-start threshold (OSPFv3)

The **adjacency exchange-start threshold** command sets the exchange-start options for an OSPF instance.

The **no adjacency exchange-start threshold** and **default adjacency exchange-start threshold** command resets the default by removing the corresponding **adjacency exchange-start threshold** command from ***running-config***.

Command Mode

Router-OSPFv3 Configuration

Command Syntax

```
adjacency exchange-start threshold peers
no adjacency exchange-start threshold
default adjacency exchange-start threshold
```

Parameters

- *peers* Value ranges from 1 4294967295. Default value is 10.

Example

- This command sets the adjacency exchange start threshold to 156923.

```
switch(config)#ipv6 router ospf 3
switch(config-router-ospf3)#adjacency exchange-start threshold 156923
switch(config-router-ospf3)#
```

area default-cost (OSPFv3)

The **area default-cost** command sets the cost for the default summary routes sent into an area. When the **area default-cost** command is not configured for an area, the default-cost of that area is set to 10.

The **no area default-cost** and **default area default-cost** command resets the default-cost value of the specified area to 10 by removing the corresponding **area default-cost** command from ***running-config***. The **no area (OSPFv3)** command removes all area commands for the specified area from ***running-config***, including the **area default-cost** command.

Command Mode

Router-OSPFv3 Configuration

Command Syntax

```
area area_id default-cost def_cost
no area area_id default-cost
default area area_id default-cost
```

Parameters

- *area_id* area number. <**0** to **4294967295**> or <**0.0.0.0** to **255.255.255.255**> ***Running-config*** stores value in dotted decimal notation.
- *def_cost* Values range from 1 to 65535.

Example

- These commands configure a cost of 15 for default summary routes that an ABR sends into area 100.

```
switch(config)#ipv6 router ospf 9
switch(config-router-ospf3)#area 100 default 15
switch(config-router-ospf3)#show active
ipv6 router ospf 9
   area 0.0.0.100 default-cost 15
switch(config-router-ospf3)#
```

area nssa (OSPFv3)

The **area nssa** command configures an OSPFv3 area as a not-so-stubby area (NSSA). All routers in an AS must specify the same area type for identically numbered areas.

NSSA ASBRs advertise external LSAs that are part of the area, but do not advertise external LSAs from other areas.

Areas are ***normal*** by default; area type configuration is required only for stub NSSA areas. Area 0 is always a normal area and cannot be configured through this command.

The **no area nssa** command configures the specified area as a normal area by removing the specified **area nssa** command from ***running-config***.

Command Mode

Router-OSPFv3 Configuration

Command Syntax

```
area area_id nssa [TYPE]
no area area_id nssa [TYPE][
default area area_id nssa [TYPE]
```

Parameters

- *area_id*
  - Valid formats: integer <**1** to **4294967295**> or dotted decimal <**0.0.0.1** to **255.255.255.255**>
  - Area 0 (or 0.0.0.0) is not configurable; it is always ***normal***.
  - ***Running-config*** stores value in dotted decimal notation.
- ***TYPE***
- Values include:
  - <no parameter>
  - **nssa-only**

Example

- This command configures area 3 as a NSSA area.

```
switch(config)#ipv6 router ospf 1
switch(config-router-ospf3)#area 3 nssa nssa-only
switch(config-router-ospf3)#
```

area nssa default-information-originate (OSPFv3)

The **area nssa default-information-originate** command sets an area as an NSSA and the generation of a type 7 default LSAis created if a default route exists in the routing table.

The switch supports three area types:

Areas are ***normal*** by default; area type configuration is required only for stub NSSA areas. Area 0 is always a normal area and cannot be configured through this command.

The **no area** and **default area** commands remove the specified area from the OSPFv3 instance by deleting all **area** commands from ***running-config*** for the specified area, including the **area default-cost (OSPFv3)** command.

The **no area stub** and **default area stub** commands configure the specified area as a normal area.

Command Mode

Router-OSPFv3 Configuration

Command Syntax

area *area_id* **nssa default-information-originate [*VALUE*][*TYPE*][*EXCL*]**
no area *area_id* **nssa default-information-originate [*VALUE*][*TYPE*][*EXCL*]**
default area *area_id* **nssa default-information-originate [*VALUE*][*TYPE*][*EXCL*]**

All parameters except *area_id* can be placed in any order.

Parameters

- *area_id*
  - Valid formats: integer <**1** to **4294967295**> or dotted decimal <**0.0.0.1** to **255.255.255.255**>
  - Area 0 (or 0.0.0.0) is not configurable; it is always ***normal***.
  - ***Running-config*** stores value in dotted decimal notation.
- ***VALUE*** Values include:
  - <no parameter>
  - **metric** *<1-65535>*
- ***TYPE*** Values include:
  - <no parameter>
  - **metric-type** *<1-2>*
- ***EXCL*** Values include:
  - <no parameter>
  - **nssa-only**

Example

- These commands sets area 1 as NSSA only and generates a type 7 default LSA if a default route exists in the routing table.

```
switch(config-router-ospf3)#area 3 nssa default-information-originate nssa-only
switch(config-router-ospf3)#
```

- These commands generates a type 7 default route.

```
switch(config-router-ospf3)#area 3 nssa default-information-originate
switch(config-router-ospf3)#
```

area nssa translate type7 always (OSPFv3)

The **area nssa translate type7 always** command configures the switch to always translate Type-7 link-state advertisement (LSAs) to Type-5 LSAs.

The **no area nssa translate type7 always** and **no area nssa translate type7 always** commands allow LSAs to be translated dynamically by removing the **no area nssa translate type7 always** command from ***running-config***.

Command Mode

Router-OSPFv3 Configuration

Command Syntax

```
area area_id nssa translate type7 always
no area_id nssa translate type7 always
default area_id nssa translate type7 always
```

Parameters

- *area_id*
  - Valid formats: integer <**1** to **4294967295**> or dotted decimal <**0.0.0.1** to **255.255.255.255**>
  - Area 0 (or 0.0.0.0) is not configurable; it is always ***normal***.
  - ***Running-config*** stores value in dotted decimal notation.

Example

- These commands configure the switch to always translate Type-7 link-state advertisement (LSAs) to Type-5 LSAs.

```
switch(config)#ipv6 router ospf 3
switch(config-router-ospf3)#area 3 nssa translate type7 always
switch(config-router-ospf)#
```

area range (OSPFv3)

The **area range** command is used by OSPFv3 area border routers to summarize routes.

The **no area range** and **default area range** commands remove the area-range by deleting the corresponding **area range** command from ***running-config***.

Command Mode

Router-OSPFv3 Configuration

Command Syntax

area *area_id* **range** *net_addr* [***ADVERTISE_SETTING***] [***COST_SETTING***]
no area *area_id* **range** *net_addr* [***ADVERTISE_SETTING***] [***COST_SETTING***]
default area *area_id* **range** *net_addr* [***ADVERTISE_SETTING***] [***COST_SETTING***]

Parameters

- *area_id* <**0** to **4294967295**> or <**0.0.0.0** to **255.255.255.255**>
- *net_addr*
- ***ADVERTISE_SETTING*** specifies the LSA advertising activity. Values include
  - <no parameter>
  - **advertise**
  - **not-advertise**
- ***COST_SETTING*** Values include
  - <no parameter>
  - **cost** *range_cost* Value ranges from 1 to 65535.

Examples

- These commands summarize routes at an area boundary 1.

```
switch(config)#ipv6 router ospf 1
switch(config-router-ospf3)#area 1 range 2001:0DB8:0:1::/64
switch(config-router-ospf3)#
```

- These commands modify the address range status to DoNotAdvertise.

```
switch(config)# ipv6 router ospf 1
switch(config-ospf6-router)# area 1 range 2001:0DB8:0:1::/64 not-advertise
switch(config-ospf6-router)#
```

area stub (OSPFv3)

The **area stub** command configures the area type of an OSPFv3 area.

Areas are ***normal*** by default;.

The **no area stub** command configures the specified area as a normal area.

Command Mode

Router-OSPFv3 Configuration

Command Syntax

```
area area_id stub
no area area_id stub
default area area_id stub
```

Parameters

- *area_id*
  - Valid formats: integer <**1** to **4294967295**> or dotted decimal <**0.0.0.1** to **255.255.255.255**>
  - Area 0 (or 0.0.0.0) is not configurable; it is always ***normal***.
  - ***Running-config*** stores value in dotted decimal notation.

Examples

- This command configures area 45 as a stub area.

```
switch(config)#ipv6 router ospf 3
switch(config-router-ospf3)#area 45 stub
switch(config-router-ospf3)#
```

- This command configures area 10.92.148.17 as a stub area.

```
switch(config-router-ospf3)#area 10.92.148.17 stub
switch(config-router-ospf3)#
```

clear ipv6 ospf force-spf

The **clear ipv6 ospf force-spf** command starts the SPF algorithm without clearing the OSPF database.

Command Mode

Privileged EXEC

Command Syntax

clear ipv6 ospf force-spf [***VRF_INSTANCE***]

Parameters

- VRF_INSTANCE Values include:
  - <no parameter> Action is performed in the default VRF.
  - **vrf** *vrf_name* Action is performed in the specified VRF.

Example

- This command restarts the SPF algorithm in the default VRF without first clearing the OSPFv3 database.

```
switch(config)#clear ipv6 ospf force-spf
switch(config)#
```

default-information originate (OSPFv3)

The **default-information originate** command generates a default external route into an OSPF domain.

The **no default-information originate** and **default default-information originate** command removes the configuration from the ***running-config***.

Command Mode

Router-OSPFv3 Configuration

Command Syntax

```
default-information originate [DURATION][VALUE][TYPE][MAP]
no default-information originate
default default-information originate
```

All parameters can be placed in any order.

Parameters

- ***DURATION*** Values include:
  - <no parameter>
  - **always**
- ***VALUE*** Values include:
  - <no parameter>
  - **metric** *<1-65535>*
- ***TYPE*** Values include:
  - <no parameter>
  - **metric-type** *<1-2>*
- ***MAP*** Values include:
  - <no parameter>
  - **route-map** *map_name*

Examples

- These commands will advertise the OSPFv3 default route regardless of whether the switch has a default route configured.

```
switch(config)#ipv6 router ospf 1
switch(config-router-ospf3)#default-information originate always
switch(config-router-ospf3)#show active
ipv6 router ospf 1
   default-information originate always
```

- These commands configures OSPF area 1 as metric of 100 for the default route with an external metric type of Type 1.

```
switch(config)#ipv6 router ospf 1
switch(config-router-ospf3)#default-information originate metric 100 metric-type
1
switch(config-router-ospf3)#show active
ipv6 router ospf 1
   default-information originate metric 100 metric-type 1
switch(config-router-ospf3)#
```

default-metric (OSPFv3)

The **default-metric** command sets default metric value for routes redistributed into the OSPFv3 domain.

The **no default-metric** and **default default-metric** commands restores the default metric to its default value of 10 by removing the **default-metric** command from ***running-config***.

Command Mode

Router-OSPFv3 Configuration

Command Syntax

```
default-metric def_metric
no default-metric
default default-metric
```

Parameters

- *def_metric* Values range from 1 to 65535. Default value is 10.

Example

- These commands configure a default metric of 30 for routes redistributed into OSPFv3.

```
switch(config)#ipv6 router ospf 9
switch(config-router-ospf3)#default-metric 30
switch(config-router-ospf3)#show active
ipv6 router ospf 9
   default-metric 30
switch(config-router-ospf3)#
```

distance ospf intra-area (OSPFv3)

The **distance ospf intra-area** command sets the administrative distance for routes in a single OSPFv3 area. The default is 110.

The **no distance ospf intra-area** and **default distance ospf intra-area** commands remove the **distance ospf intra-area** command from ***running-config***, returning the OSFPv3 intra-area distance setting to the default value of 110.

Command Mode

Router-OSPFv3 Configuration

Command Syntax

```
distance ospf intra-area distance
no distance ospf intra-area
default distance ospf intra-area
```

Parameters

- *distance* Values range from 1 to 255. Default is 110.

Example

- This command configures a distance of 90 for all OSPFv3 intra-area routes on the switch.

```
switch(config)#ipv6 router ospf 9
switch(config-router-ospf3)#distance ospf intra-area 90
switch(config-router-ospf3)#show active
ipv6 router ospf 9
   distance ospf intra-area 90
switch(config-router-ospf3)#
```

ipv6 ospf area

The **ipv6 ospf area** command enables OSPFv3 on the interface and associates the area to the interface.

OSPFv3 areas are configured in by **no area (OSPFv3)** commands in router-OSPFv3 configuration mode

The **no ipv6 ospf area** and **default ipv6 ospf area** commands disable OSPFv3 on the configuration mode interface by removing the corresponding **ipv6 ospf area** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration
Interface-Loopback Configuration
Interface-Port-Channel Configuration
Interface-VLAN Configuration

Command Syntax

```
ipv6 ospf process_id area area_id
no ipv6 ospf process_id [area area_id]
default ipv6 ospf process_id [area area_id]
```

Parameters

- *process_id* Values range from 1 to 65535.
- *area_id*
  - Valid formats: integer <**0** to **4294967295**> or dotted decimal <**0.0.0.0** to **255.255.255.255**>
  - ***Running-config*** stores value in dotted decimal notation.

Example

- These commands enable OSPFv3 on VLAN interface 200 and associates area 0 to the interface.

```
switch(config)#interface vlan 200
switch(config-if-Vl200)#ipv6 ospf 9 area 0
switch(config-if-Vl200)#show active
interface Vlan200
   ipv6 ospf 9 area 0.0.0.0
switch(config-if-Vl200)#
```

ipv6 ospf cost

The **ipv6 ospf cost** command sets the OSPFv3 cost for the interface. The default OSPFv3 cost is 10.

The **no ipv6 ospf cost** and **default ipv6 ospf cost** commands restore the default cost of 10 for the configuration mode interface by removing the corresponding **ipv6 ospf cost** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration
Interface-Loopback Configuration
Interface-Port-Channel Configuration
Interface-VLAN Configuration

Command Syntax

```
ipv6 ospf cost interface_cost
no ipv6 ospf cost
default ipv6 ospf cost
```

Parameters

- *interface_cost* Value ranges from 1 to 65535; default is 10.

Example

- This command configures a cost of 50 for VLAN 200.

```
switch(config)#interface vlan 200
switch(config-if-Vl200)#ipv6 ospf cost 50
switch(config-if-Vl200)#show active
interface Vlan200
   ipv6 ospf cost 50
switch(config-if-Vl200)#
```

ipv6 ospf dead-interval

The **ipv6 ospf dead-interval** command sets the OSPFv3 dead interval.

The **no ipv6 ospf dead-interval** and **default ipv6 ospf dead-interval** commands restore the default dead interval of 40 seconds on the configuration mode interface by removing the corresponding **ipv6 ospf dead-interval** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration
Interface-Loopback Configuration
Interface-Port-Channel Configuration
Interface-VLAN Configuration

Command Syntax

```
ipv6 ospf dead-interval time
no ipv6 ospf dead-interval
default ipv6 ospf dead-interval
```

Parameters

- *time* Value ranges from 1 to 65535; default is 40.

Example

- This command configures a dead interval of 75 seconds for VLAN 200.

```
switch(config)#interface vlan 200
switch(config-if-Vl200)#ipv6 ospf dead-interval 75
switch(config-if-Vl200)#show active
interface Vlan200
   ipv6 ospf dead-interval 75
switch(config-if-Vl200)#
```

ipv6 ospf hello-interval

The **ipv6 ospf hello-interval** command sets the OSPFv3 hello interval. The hello interval is the period between the transmission of consecutive hello packets.

Each OSPFv3 neighbor should be the same hello interval and should not be longer than any neighbor's dead interval.

The **no ipv6 ospf hello-interval** and **default ipv6 ospf hello-interval** commands restore the default hello interval of 10 seconds on the configuration mode interface by removing the **ipv6 ospf hello-interval** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration
Interface-Loopback Configuration
Interface-Port-Channel Configuration
Interface-VLAN Configuration

Command Syntax

```
ipv6 ospf hello-interval time
no ipv6 ospf hello-interval
default ipv6 ospf hello-interval
```

Parameters

- *time* Values range from 1 to 65535; default is 10.

Example

- These commands configure a hello interval of 45 seconds for VLAN 200.

```
switch(config)#interface vlan 200
switch(config-if-Vl200)#ipv6 ospf hello-interval 45
switch(config-if-Vl200)#show active
interface Vlan200
   ipv6 ospf hello-interval 45
switch(config-if-Vl200)#
```

ipv6 ospf network

The **ipv6 ospf network** command sets the configuration mode interface as a point-to-point link. By default, interfaces are set as broadcast links.

The **no ipv6 ospf network** and **default ipv6 ospf network** commands set the configuration mode interface as a broadcast link by removing the corresponding **ipv6 ospf network** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration
Interface-Loopback Configuration
Interface-Port-Channel Configuration
Interface-VLAN Configuration

Command Syntax

```
ipv6 ospf network point-to-point
no ipv6 ospf network
default ipv6 ospf network
```

Examples

- These commands configure VLAN interface 200 as a point-to-point link.

```
switch(config)#interface vlan 200
switch(config-if-Vl200)#ipv6 ospf network point-to-point
switch(config-if-Vl200)#show active
interface Vlan200
   ipv6 ospf network point-to-point
switch(config-if-Vl200)#
```

- This command restores Ethernet interface 10 as a broadcast link.

```
switch(config)#interface vlan 200
switch(config-if-Vl200)#no ipv6 ospf network
switch(config-if-Vl200)#show active
interface Vlan200
switch(config-if-Vl200)#
```

ipv6 ospf priority

The **ipv6 ospf priority** command configures the OSPFv3 router priority.

The **no ipv6 ospf priority** and **default ipv6 ospf priority** commands restore the default priority (1) on the interface by removing the corresponding **ipv6 ospf priority** command from ***running-config.***

Command Mode

Interface-Ethernet Configuration
Interface-Loopback Configuration
Interface-Port-Channel Configuration
Interface-VLAN Configuration

Command Syntax

```
ipv6 ospf priority priority_level
no ipv6 ospf priority
default ipv6 ospf priority
```

Parameters

- *priority_level* Settings range from 0 to 255.

Example

- This command configures a router priority of 128 for VLAN 200.

```
switch(config)#interface vlan 200
switch(config-if-Vl200)#ipv6 ospf priority 128
switch(config-if-Vl200)#show active
interface Vlan200
   ipv6 ospf priority 128
switch(config-if-Vl200)#
```

ipv6 ospf retransmit-interval

The **ipv6 ospf retransmit-interval** command configures the link state advertisement retransmission interval.

The **no ipv6 ospf retransmit-interval** and **default ipv6 ospf retransmit-interval** commands restore the default retransmission interval of 5 seconds on the configuration mode interface by removing the corresponding **ipv6 ospf retransmit-interval** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration
Interface-Loopback Configuration
Interface-Port-Channel Configuration
Interface-VLAN Configuration

Command Syntax

```
ipv6 ospf retransmit-interval period
no ipv6 ospf retransmit-interval
default ipv6 ospf retransmit-interval
```

Parameters

- *period* Value ranges from 1 to 65535; default is 5.

Example

- This command configures a retransmission interval of 25 seconds for VLAN 200.

```
switch(config)#interface vlan 200
switch(config-if-Vl200)#ipv6 ospf retransmit-interval 25
switch(config-if-Vl200)#show active
interface Vlan200
   ipv6 ospf retransmit-interval 25
switch(config-if-Vl200)#
```

ipv6 ospf transmit-delay

The **ipv6 ospf transmit-delay** command configures the transmission delay for OSPFv3 packets.

The **no ipv6 ospf transmit-delay** and **default ipv6 ospf transmit-delay** commands restore the default transmission delay of one second on the configuration mode interface by removing the corresponding **ipv6 ospf transmit-delay** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration
Interface-Loopback Configuration
Interface-Port-Channel Configuration
Interface-VLAN Configuration

Command Syntax

```
ipv6 ospf transmit-delay trans
no ipv6 ospf transmit-delay
default ipv6 ospf transmit-delay
```

Parameters

- *trans* Value ranges from 1 to 65535; default is 1.

Example

- This command configures a transmission delay of 10 seconds for VLAN 200.

```
switch(config)#interface vlan 200
switch(config-if-Vl200)#ipv6 ospf transmit-delay 10
switch(config-if-Vl200)#show active
interface Vlan200
   ipv6 ospf transmit-delay 10
switch(config-if-Vl200)#
```

ipv6 router ospf

The **ipv6 router ospf** command places the switch in router-OSPFv3 configuration mode and creates and OSPFv3 instance if one does not already exist. Note that each OSPFv3 instance on the switch must have a unique process ID. A router ID for the new instance will be created if one does not already exist.

The **show ipv6 ospf** command displays the router ID of each OSPFv3 instance configured on the switch.

The **no ipv6 router ospf** and **default ipv6 router ospf** command deletes the OSPFv3 instance.

Refer to the **Router-OSPFv3 Configuration Mode** command for a list of commands available in router-OSPFv3 configuration mode.

Command Mode

Global Configuration

Command Syntax

```
ipv6 router ospf process_id [VRF_INSTANCE]
no ipv6 router ospf process_id [VRF_INSTANCE]
default ipv6 router ospf process_id [VRF_INSTANCE]
```

Parameters

- *process_id* Values range from 1 to 65535.
- VRF_INSTANCE Values include:
  - <no parameter> OSPF instance is in the default VRF.
  - **vrf** *vrf_name* OSPF instance is the specified VRF.

Examples

- This command creates an OSPFv3 instance in the default VRF with process ID 9.

```
switch(config)#ipv6 router ospf 9
switch(config-router-ospf3)#show active
ipv6 router ospf 9
switch(config-router-ospf3)#
```

- This command deletes the OSPFv3 instance.

```
switch(config)#no ipv6 router ospf 9
switch(config)#
```

log-adjacency-changes (OSPFv3)

The **log-adjacency-changes** command enables syslog messages to be sent when it detects OSPFv3 link state changes or when it detects that a neighbor has gone up or down. Log message sending is enabled by default.

The **default log-adjacency-changes** command restores the default state by removing the **log-adjacency-changes** statement from ***running-config***.

The default option (sending a message only when a neighbor goes up or down) is active when running-config does not contain any form of the command. Entering the command in any form replaces the previous command state in ***running-config***.

The **no log-adjacency-changes** disables link state change syslog reporting.

The **default log-adjacency-changes** command restores the default state by removing the **log-adjacency-changes detail** or **no log-adjacency-changes** statement from ***running-config***.

Command Mode

Router-OSPFv3 Configuration

Command Syntax

```
log-adjacency-changes [INFO_LEVEL]
no log-adjacency-changes
default log-adjacency-changes
```

Parameters

- ***INFO_LEVEL*** Options include
  - <no parameter> Sends messages when a neighbor goes up or down.
  - **detail** Sends messages for all neighbor state changes.

Example

- This command configures the switch to send a syslog message when a neighbor state changes.

```
switch(config)#ipv6 router ospf 9
switch(config-router-ospf3)#log-adjacency-changes
switch(config-router-ospf3)#show active
ipv6 router ospf 9
   log-adjacency-changes
switch(config-router-ospf3)#
```

max-metric router-lsa (OSPFv3)

The **max-metric router-lsa** command configures OSPF to include the maximum value in LSA metric fields to keep other network devices from using the switch as a preferred intermediate SPF hop.

The **no max-metric router-lsa** and **default max-metric router-lsa** commands disable the advertisement of a maximum metric.

Command Mode

Router-OSPFv3 Configuration

Command Syntax

```
max-metric router-lsa [EXTERNAL][STUB][STARTUP][SUMMARY]
no max-metric router-lsa [EXTERNAL][STUB][STARTUP][SUMMARY]
default max-metric router-lsa [EXTERNAL][STUB][STARTUP][SUMMARY]
```

All parameters can be placed in any order.

Parameters

- ***EXTERNAL*** Values include:
  - <no parameter> Default value of 1.
  - **external-lsa**
  - **external-lsa** ***<1*** to ***16777215>*** The default value is 0xFF0000.
- ***STUB*** Values include:
  - <no parameter> Default value of 2.
  - **include-stub**
- ***STARTUP*** Values include:
  - <no parameter>
  - **on-startup**
  - **on-startup wait-for-bgp**
  - **on-startup** *<5* to *86400>*

    wait-for-bgp or an **on-start** time value is not included in **no** and **default** commands.
- ***SUMMARY*** Values include:
  - <no parameter> Metric is set to the default value of 1.
  - **summary-lsa**
  - **summary-lsa** ***<1*** to ***16777215>***

Example

- This command configures OSPFv3 to include the maximum value in LSA metric fields until BGP has converged:

```
switch(config-router-ospf3)#max-metric router-lsa on-startup wait-for-bgp
switch(config-router-ospf3)#
```

maximum-paths (OSPFv3)

The **maximum-paths** command sets the maximum number of parallel routes that OSPFv3 supports on the switch.

The **no maximum-paths** command restores the maximum number of parallel routes that OSPFv3 supports on the switch to the default value of 16 by removing the maximum-paths command from ***running-config***.

Command Mode

Router-OSPFv3 Configuration

Command Syntax

```
maximum-paths paths
no maximum-paths
default maximum-paths
```

Parameters

- *paths* Value range is platform dependent:
  - Arad: Value ranges from 1 to 128. Default value is 128.
  - FM6000: Value ranges from 1 to 32. Default value is 32.
  - PetraA: Value ranges from 1 to 16. Default value is 16.
  - Trident: Value ranges from 1 to 32. Default value is 32.
  - Trident-II: Value ranges from 1 to 128. Default value is 128.

Example

- This command configures the maximum number of OSPFv3 parallel paths to 12.

```
switch(config)#ipv6 router ospf 9
switch(config-router-ospf3)#maximum-paths 12
switch(config-router-ospf3)#
```

no area (OSPFv3)

The **no area** command removes all area configuration commands for the specified OSPFv3 area. Commands removed by the **no area** command include:

- area
- nssa
- range
- stub

Area settings can be removed individually; refer to the command description page of the desired command for details.

Command Mode

Router-OSPFv3 Configuration

Command Syntax

```
no area area_id [TYPE]
default area area_id [TYPE]
```

Parameters

- *area_id* area number.
  - Valid formats: integer <**1** to **4294967295**> or dotted decimal <**0.0.0.1** to **255.255.255.255**>
  - Area 0 (or 0.0.0.0) is not configurable; it is always ***normal***.
  - ***Running-config*** stores value in dotted decimal notation.
- ***TYPE*** area type. Values include:
  - **nssa**
  - **nssa translate type7 always**
  - **stub**
  - **stub no-summary**

Example

- These commands remove the area 1 stub configuration.

```
switch(config)#ipv6 router ospf 9
switch(config-router-ospf3)# no area 1 stub
switch(config-router-ospf3)#
```

passive-interface (OSPFv3)

The **passive-interface** command disables OSPF on an interface range. All interfaces are active by default.

The **no passive-interface** and **default passive-interface** commands enable OSPFv3 on the specified interface range by removing the corresponding **passive-interface** statements from ***running-config***.

Command Mode

Router-OSPFv3 Configuration

Command Syntax

```
passive-interface INTERFACE_NAME
no passive-interface INTERFACE_NAME
default passive-interface INTERFACE_NAME
```

Parameters

- ***INTERFACE_NAME*** Options include:
  - **ethernet** *e_range*
  - **loopback** *l_range*
  - **management** *m_range*
  - **port-channel** *p_range*
  - **vlan** *v_range*
  - **vxlan** *vx_range*
  - **default**

  Valid *e_range*, *l_range*, *m_range*, *p_range v_range*, and *vx_range* formats include number, range, or comma-delimited list of numbers and ranges.

Example

- This command configures VLAN interfaces 101 through 103 as passive interfaces.

```
switch(config)#ipv6 router ospf 9
switch(config-router-ospf3)#passive-interface vlan 101-103
switch(config-router-ospf3)#show active
ipv6 router ospf 9
   passive-interface Vlan101
   passive-interface Vlan102
   passive-interface Vlan103
switch(config-router-ospf3)#
```

redistribute (OSPFv3)

The **redistribute** command enables the advertising of all specified routes into the OSPFv3 domain as external routes.

The **no redistribute** and **default redistribute** commands remove the corresponding **redistribute** command from ***running-config***, disabling route redistribution for the specified route type.

Command Mode

Router-OSPFv3 Configuration

Command Syntax

```
redistribute ROUTE_TYPE [ROUTE_MAP]
no redistribute ROUTE_TYPE
default redistribute ROUTE_TYPE
```

Parameters

- ***ROUTE_TYPE*** Options include:
  - **BGP**
  - **connected**
  - **static**
- ***ROUTE_MAP*** Options include:
  - **route-map** *map_name*

Example

- The **redistribute static** command starts the advertising of static routes as OSPFv3 external routes.

```
switch(config)#ipv6 router ospf 9
switch(config-router-ospf3)#redistribute static
switch(config-router-ospf3)#show active
ipv6 router ospf 9
   redistribute connected
   redistribute static
switch(config-router-ospf3)#
```

router-id (OSPFv3)

The **router-id** command assigns the router ID for an OSPFv3 instance. The switch sets the router ID to the first available alternative in the following list:

1. The **router-id** command
2. The loopback IP address
3. The highest IP address present on the device

Important! When configuring VXLAN on an MLAG, always manually configure the OSPFv3 router ID to prevent the switch from using the common VTEP IP address as the router ID.

The **no router-id** and **default router-id** commands remove the router ID command from ***running-config.***

Command Mode

Router-OSPFv3 Configuration

Command Syntax

```
router-id identifier
no router-id
default router-id
```

Parameters

- *identifier* Value ranges from 0.0.0.0 to 255.255.255.255 (dotted decimal notation).

Example

- This command assigns 10.10.1.4 as the router ID for the OSPFv3 instance.

```
switch(config)#ipv6 router ospf 9
switch(config-router-ospf3)#router-id 10.10.1.4
switch(config-router-ospf3)#show active
ipv6 router ospf 9
   router-id 15.10.1.4
switch(config-router-ospf3)#
```

show ipv6 ospf

The **show ipv6 ospf** command displays information about OSPFv3 routing.

Command Mode

EXEC

Command Syntax

show ipv6 ospf [***VRF_INSTANCE***]

Parameters

- VRF_INSTANCE Values include:
  - <no parameter> Displays information for all VRFs.
  - **vrf** *vrf_name* Displays information for the specified VRF.

Example

- This command displays OSPFv3 routing information for all VRFs.

```
switch>show ipv6 ospf
Routing Process "ospfv3 1" with ID 10.37.0.23 and Instance 0 VRF default
  It is an autonomous system boundary router and is an area border router
  Minimum LSA arrival interval 1000 msecs
  Initial LSA throttle delay 1000 msecs
  Minimum hold time for LSA throttle 5000 msecs
  Maximum wait time for LSA throttle 5000 msecs
  It has 13 fully adjacent neighbors
  Number of areas in this router is 2. 2 normal, 0 stub, 0 nssa
  Initial SPF schedule delay 0 msecs
  Minimum hold time between two consecutive SPFs 5000 msecs
  Current hold time between two consecutive SPFs 5000 msecs
  Maximum wait time between two consecutive SPFs 5000 msecs
  SPF algorithm last executed 00:00:14 ago
  No scheduled SPF
  Adjacency exchange-start threshold is 20
  Maximum number of next-hops supported in ECMP is 128
  Area 0.0.0.0
    Number of interface in this area is 1
    It is a normal area
  Area 0.0.0.2
    Number of interface in this area is 1
    It is a normal area
switch>
```

show ipv6 ospf border-routers

The **show ipv6 ospf border-routers** command displays the OSPF routing table entries.

Command Mode

EXEC

Command Syntax

show ipv6 ospf border-routers [***VRF_INSTANCE***]

Parameters

- VRF_INSTANCE Values include:
  - <no parameter> Displays information for all VRFs.
  - **vrf** *vrf_name* Displays information for the specified VRF.

Example

- This command displays the ABRs and ASBRs configured in the switch in all VRFs.

```
switch>show ipv6 ospf border-routers
Routing Process "ospf 9", VRF default
 Router 10.37.0.32 area 0.0.0.0 ASBR
 Router 10.37.0.18 area 0.0.0.0 ASBR
 Router 10.37.0.22 area 0.0.0.0 ASBR ABR
 Router 10.37.0.31 area 0.0.0.0 ASBR ABR
 Router 10.37.0.58 area 0.0.0.0 ASBR
 Router 10.37.0.37 area 0.0.0.0 ASBR
 Router 10.37.0.22 area 0.0.0.2 ASBR ABR
 Router 10.37.0.31 area 0.0.0.2 ASBR ABR
switch>
```

show ipv6 ospf database

The **show ipv6 ospf database** command displays data from the OSPF database. The switch can return link state data for a single VRF or for all VRFs on the switch.

Command Mode

EXEC

Command Syntax

show ipv6 ospf database [***VRF_INSTANCE***]

Parameters

- VRF_INSTANCE Values include:
  - <no parameter> Displays information for all VRFs.
  - **vrf** *vrf_name* Displays information for the specified VRF.

Example

- This command displays OSPF database information for VRF blue.

```
switch>show ipv6 ospf database vrf blue
Codes: AEX - AS External, GRC - Grace,
 IAP - Inter Area Prefix, IAR - Inter Area Router,
 LNK - Link, NAP - Intra Area Prefix,
 NSA - Not So Stubby Area, NTW - Network,
 RTR - Router
Routing Process "ospf 9", VRF blue
AS Scope LSDB
switch>
```

show ipv6 ospf database<link-state details>

The **show ipv6 ospf database <link-state details>** command displays detailed information about the specified link state advertisements. The switch can return link state data about a single area or for all areas on the switch.

Command Mode

EXEC

Command Syntax

show ipv6 ospf database [*FILTER*] *LINK_TYPE* [*LINKSTATE_ID*][*ROUTER*][*DATA_LEVEL*]

Parameters

- ***FILTER*** filters the output of the command by specifying areas. Options include:
  - **area *<A.B.C.D>***
  - **area backbone**
- ***LINK_TYPE*** Parameter options include:
  - **router**
  - **network**
  - **inter-area-prefix**
  - **inter-area-router**
  - **intra-area-prefix**
  - **nssa**
- ***LINKSTATE_ID*** Options include:
  - <no parameter>
  - ***<A.B.C.D>***
- ***ROUTER*** Options include:
  - <no parameter>
  - **adv-router** [*a.b.c.d*]
  - **self-originate**
- ***DATA_LEVEL*** Options include:
  - <no parameter>
  - **detail**

Example

- This command displays the OSPF database summary.

```
switch>show ipv6 ospf database detail
Codes: AEX - AS External, GRC - Grace,
       IAP - Inter Area Prefix, IAR - Inter Area Router,
       LNK - Link, NAP - Intra Area Prefix,
       NSA - Not So Stubby Area, NTW - Network,
       RTR - Router

Routing Process "ospf 9":

  AS Scope LSDB

LSA Type: AEX
    Link State ID: 0.0.0.1
    Advertising Router: 10.21.4.9
    Age: 1123
    Sequence Number: 0x80000001
    Checksum: 0x009c89
    Length: 40
    Metric Type: 2
    Metric: 1
    External Route Tag: 0
  Prefix
      Prefix: fd7a:629f:52a4:1::
      Length: 64
      Options: (null)
      Metric: 0

Area 0.0.1.44 LSDB

LSA Type: LNK
    Link State ID: 0.0.0.14
    Advertising Router: 10.26.0.11
    Age: 1285
    Sequence Number: 0x800000c1
    Checksum: 0x00629b
    Length: 56
    Option Priority: 16777235
    Link Local Addr: fe80::21c:73ff:fe0b:a80e
    Number of Prefixes: 1

  Prefix
      Prefix: fd7a:629f:52a4:fe08::
      Length: 64
      Options: (null)
      Metric: 0

LSA Type: LNK
    Link State ID: 0.0.0.34
    Advertising Router: 10.26.0.22
    Age: 1042
    Sequence Number: 0x800000c2
    Checksum: 0x00bd9f
    Length: 56
    Option Priority: 16777235
    Link Local Addr: fe80::21c:73ff:fe01:5fe1
    Number of Prefixes: 1
```

```
  Prefix
      Prefix: fd7a:629f:52a4:fe08::
      Length: 64
      Options: (null)
      Metric: 0

LSA Type: LNK
    Link State ID: 0.0.0.15
    Advertising Router: 10.26.0.23
    Age: 1128
    Sequence Number: 0x800000c7
    Checksum: 0x00d4ab
    Length: 56
    Option Priority: 16777235
    Link Local Addr: fe80::21c:73ff:fe00:1319
    Number of Prefixes: 1

  Prefix
      Prefix: fd7a:629f:52a4:fe08::
      Length: 64
      Options: (null)
      Metric: 0

  Interface vlan3925 LSDB



LSA Type: LNK
    Link State ID: 0.0.0.153
    Advertising Router: 10.27.0.52
    Age: 1186
    Sequence Number: 0x800009b6
    Checksum: 0x002f27
    Length: 56
    Option Priority: 16777235
    Link Local Addr: fe80::21c:73ff:fe17:3906
    Number of Prefixes: 1

  Prefix
      Prefix: fd7a:629f:52a4:fe67::
      Length: 64
      Options: (null)
      Metric: 0


  Interface lo0 LSDB


switch>
```

show ipv6 ospf database <link state list>

The **show ipv6 ospf database <link state list>** command displays the OSPF link state advertisements that originate on a switch.

Command Mode

EXEC

Command Syntax

show ipv6 ospf database [*FILTER*] [*LINKSTATE_ID*] [*ROUTER*] [*DATA_LEVEL*]

Parameters

- ***FILTER*** filters the output of the command by specifying areas. Options include:
  - <no parameter>
  - **area** *<A.B.C.D>*
  - **area backbone**
  - **as**
  - **as external**
- ***LINKSTATE_ID*** Options include:
  - <no parameter>
  - ***<A.B.C.D>***
- ***ROUTER*** Options include:
  - <no parameter>
  - **adv-router** [*a.b.c.d*]
  - **self-originate**
- ***DATA_LEVEL*** Options include:
  - <no parameter>
  - **detail**

Example

- This command displays the OSPFv3 database of link state advertisements.

```
switch>show ipv6 ospf database 10.26.0.23
Codes: AEX - AS External, GRC - Grace,
       IAP - Inter Area Prefix, IAR - Inter Area Router,
       LNK - Link, NAP - Intra Area Prefix,
       NSA - Not So Stubby Area, NTW - Network,
       RTR - Router

Routing Process "ospf 9":

  AS Scope LSDB

Type        Link ID     ADV Router  Age       Seq#   Checksum
 AEX        0.0.0.5    10.37.0.37   15 0x80000005   0x00be82
 AEX        0.0.0.9    10.37.0.22 1747 0x8000002b   0x00df56
 AEX        0.0.0.3    10.37.0.46  599 0x8000002d   0x00651d
             <-------OUTPUT OMITTED FROM EXAMPLE-------->

Area 0.0.0.0 LSDB

Type        Link ID     ADV Router  Age       Seq#   Checksum
 RTR        0.0.0.0    10.37.0.32  234 0x80000031   0x00585a
 NTW       0.0.0.26    10.37.0.32  271 0x80000005   0x005609
 NAP       0.0.0.26    10.37.0.32  274 0x80000005   0x00964c
             <-------OUTPUT OMITTED FROM EXAMPLE-------->

Interface vlan3911 LSDB

Type        Link ID     ADV Router  Age       Seq#   Checksum
 LNK       0.0.0.38    10.37.0.22  267 0x80000005   0x00a45a
 LNK       0.0.0.23    10.37.0.23  270 0x8000002c   0x005b7e

  Interface vlan3902 LSDB

Type        Link ID     ADV Router  Age       Seq#   Checksum
 LNK       0.0.0.17    10.37.0.11 1535 0x8000002b   0x007120
 LNK       0.0.0.37    10.37.0.22    7 0x8000002b   0x00ce23
 LNK       0.0.0.22    10.37.0.23  250 0x8000002d   0x00c350
             <-------OUTPUT OMITTED FROM EXAMPLE-------->

switch>
```

show ipv6 ospf database link

The **show ipv6 ospf database link** command displays details of the specified link state advertisements. The switch can return link state data about a single area or for all areas on the switch.

Command Mode

EXEC

Command Syntax

show ipv6 ospf database link [*LINKSTATE_ID*] [*ROUTER*] [*DATA_LEVEL*]

Parameters

- ***LINKSTATE_ID*** Options include:
  - <no parameter>
  - <A.B.C.D>
- ***ROUTER*** Options include:
  - <no parameter>
  - **adv-router** [*a.b.c.d*]
  - **self-originate**
- ***DATA_LEVEL*** Options include:
  - <no parameter>
  - **detail**

Example

- This command displays information about the Open Shortest Path First (OSPF).

```
switch> show ipv6 ospf database link
Codes: AEX - AS External, GRC - Grace,
       IAP - Inter Area Prefix, IAR - Inter Area Router,
       LNK - Link, NAP - Intra Area Prefix,
       NSA - Not So Stubby Area, NTW - Network,
       RTR - Router

Routing Process "ospf 9":

switch>
```

show ipv6 ospf database link if-name

The **show ipv6 ospf database link** command displays link state advertisement details. The switch can return link state data about a single area or for all areas on the switch.

Command Mode

EXEC

Command Syntax

show ipv6 ospf database link if-name [*INTF_ID*] [*LS_ID*] [*ROUTER*] [*DATA_LEVEL*]

Parameters

- ***INTF_ID*** Options include:
  - **ethernet** *e_range* Ethernet interface list.
  - **loopback** *l_range* Loopback interface list.
  - **management** *m_range* Management interface list.
  - **port-channel** *p_range* Channel group interface list.
  - **vlan** *v_range* VLAN interface list.
  - **vxlan** *vx_range* VXLAN interface list.

  Valid *range* formats include number, range, or comma-delimited list of numbers and ranges.
- ***LS_ID*** Options include:
  - <no parameter>
  - ***<A.B.C.D>***
- ***ROUTER*** Options include:
  - <no parameter>
  - **adv-router** [*a.b.c.d*]
  - **self-originate**
- ***DATA_LEVEL*** Options include:
  - <no parameter>
  - **detail**

Example

- This command displays information for Ethernet 4/1 link state advertisements.

```
switch>show ipv6 ospf database link if-name ethernet 4/1
Codes: AEX - AS External, GRC - Grace,
       IAP - Inter Area Prefix, IAR - Inter Area Router,
       LNK - Link, NAP - Intra Area Prefix,
       NSA - Not So Stubby Area, NTW - Network,
       RTR - Router

Routing Process "ospf 1":

switch>
```

show ipv6 ospf database link if-type

The **show ipv6 ospf database link** command displays information of the link state advertisements. The switch can return link state data about a single area or for all areas on the switch.

Command Mode

EXEC

Command Syntax

show ipv6 ospf database link if-type [*INTF_TYPE*] [*LS_ID*] [*ROUTER*] [*DATA_LEVEL*]

Parameters

- INTF_TYPE
  - **broadcast**
  - **nbma**
  - **p2mp**
  - **p2p**
- ***LS_ID*** Options include:
  - <no parameter>
  - ***<A.B.C.D>***
- ***ROUTER*** Options include:
  - <no parameter>
  - **adv-router** [*a.b.c.d*]
  - **self-originate**
- ***DATA_LEVEL*** Options include:
  - <no parameter>
  - **detail**

Example

- This command displays LSA information for the interfaces configured for broadcast transmissions.

```
switch>show ipv6 ospf database link if-type broadcast
Codes: AEX - AS External, GRC - Grace,
       IAP - Inter Area Prefix, IAR - Inter Area Router,
       LNK - Link, NAP - Intra Area Prefix,
       NSA - Not So Stubby Area, NTW - Network,
       RTR - Router

Routing Process "ospf 1":

  Interface et4 LSDB

Type        Link ID     ADV Router  Age       Seq#   Checksum
 LNK       0.0.0.61    10.26.0.49  1378 0x80000027   0x00f8b0
 LNK       0.0.0.20    10.26.0.23  1371 0x80000027   0x005423

  Interface et7 LSDB

Type        Link ID     ADV Router  Age       Seq#   Checksum
 LNK       0.0.0.61    10.26.0.50  1298 0x80000028   0x005e0d
 LNK       0.0.0.38    10.26.0.23  1291 0x80000028   0x00ce8d

  Interface vlan3901 LSDB

Type        Link ID     ADV Router  Age       Seq#   Checksum
 LNK       0.0.0.36    10.26.0.22   216 0x800000b0   0x00c2b1
 LNK       0.0.0.19    10.26.0.23   231 0x800000b0   0x00cfca

switch>
```

show ipv6 ospf interface

The **show ipv6 ospf interface** command displays OSPFv3 information on interfaces where OSPFv3 is enabled.

Command Mode

EXEC

Command Syntax

show ipv6 ospf interface [***VRF_INSTANCE***]

Parameters

- VRF_INSTANCE Values include:
  - <no parameter> Displays information for all VRFs.
  - **vrf** *vrf_name* Displays information for the specified VRF.

Example

- This command displays OSPFv3 information for interfaces where OSPFv3 is enabled.

```
switch>show ipv6 ospf interface
Ethernet17 is up
 Interface Address fe80::48c:73ff:fe00:1319, VRF default, Area 0.0.0.0
 Network Type Broadcast, Cost 10
 Transmit Delay is 1 sec, State Backup DR, Priority 1
 Designated Router is 10.37.0.37
 Backup Designated Router is 10.37.0.23
 Timer intervals configured, Hello 10, Dead 40, Retransmit 5
 Neighbor Count is 1
 Options are R E V6
Vlan31 is up
 Interface Address fe80::48c:73ff:fe00:1319, VRF default, Area 0.0.0.0
 Network Type Broadcast, Cost 10
 Transmit Delay is 1 sec, State Backup DR, Priority 1
 Designated Router is 10.37.0.22
 Backup Designated Router is 10.37.0.23
 Timer intervals configured, Hello 10, Dead 40, Retransmit 5
 Neighbor Count is 1
 Options are R E V6
Vlan32 is up
 Interface Address fe80::48c:73ff:fe00:1319, VRF default, Area 0.0.0.0
 Network Type Broadcast, Cost 10
 Transmit Delay is 1 sec, State DR Other, Priority 1
 Designated Router is 10.37.0.11
 Backup Designated Router is 10.37.0.22
 Timer intervals configured, Hello 10, Dead 40, Retransmit 5
 Neighbor Count is 2
 Options are R E V6
switch>
```

show ipv6 ospf lsa-log

The **show ipv6 ospf lsa-log** command displays log entries when LSA update messages are sent or received for OSPFv3.

Command Mode

EXEC

Command Syntax

show ipv6 ospf [*PROCESS_ID*] lsa-log [*VRF_INSTANCE*]

Parameters

- ***PROCESS_ID*** OSPFv3 process ID. Values include:
  - <no parameter> Displays information for all process IDs.
  - <***1*** to ***65535***> Displays information for the specified process ID.
- VRF_INSTANCE Values include:
  - <no parameter> Displays information for all VRFs.
  - **vrf** *vrf_name* Displays information for the specified VRF.

Examples

- This command displays log entries when LSA update messages are sent or received for OSPFv3.

```
switch>show ipv6 ospf lsa-log
OSPF3 Process 3.3.3.3, VRF default, LSA Throttling Log:
[04:21:09] type 1: 3.3.3.3/32 [3.3.3.3], event 1, backed off, new hold value 2000 msecs
[04:21:08] type 1: 3.3.3.3/32 [3.3.3.3], event 2, backoff restarted, new hold value 900 msecs
[04:21:00] type 1: 3.3.3.3/32 [3.3.3.3], event 1, backed off, new hold value 3000 msecs
[04:21:00] type 1: 3.3.3.3/32 [3.3.3.3], event 4, maxwait value changed, new hold value 3000
msecs
/* Here the maxwait value was changed to 3000 from earlier 32000, this is not part of the log */
[04:20:42] type 1: 3.3.3.3/32 [3.3.3.3], event 1, backed off, new hold value 32000 msecs
[04:20:10] type 1: 3.3.3.3/32 [3.3.3.3], event 1, backed off, new hold value 32000 msecs
[04:19:54] type 1: 3.3.3.3/32 [3.3.3.3], event 1, backed off, new hold value 16000 msecs
[04:19:46] type 1: 3.3.3.3/32 [3.3.3.3], event 1, backed off, new hold value 8000 msecs
[04:19:42] type 1: 3.3.3.3/32 [3.3.3.3], event 1, backed off, new hold value 4000 msecs
[04:19:40] type 1: 3.3.3.3/32 [3.3.3.3], event 1, backed off, new hold value 2000 msecs
[04:19:39] type 1: 3.3.3.3/32 [3.3.3.3], event 2, backoff restarted, new hold value 900 msecs
[04:19:22] type 1: 4.4.4.4/32 [4.4.4.4], event 3, discarded, was early by 995 msecs
[04:19:22] type 1: 3.3.3.3/32 [3.3.3.3], event 0, backoff started, new hold value 1000 msecs
switch>
```

show ipv6 ospf neighbor

The **show ipv6 ospf neighbor** command displays OSPFv3 neighbor information.

Command Mode

EXEC

Command Syntax

show ipv6 ospf neighbor [***VRF_INSTANCE***]

Parameters

- VRF_INSTANCE Values include:
  - <no parameter> Displays information for all VRFs.
  - **vrf** *vrf_name* Displays information for the specified VRF.

Example

- This command displays the switch's neighbors.

```
switch>show ipv6 ospf neighbor
Routing Process "ospf 9":
Neighbor 10.37.0.37 VRF default priority is 1, state is Full
 In area 0.0.0.0 interface et12
 DR is 10.37.0.37 BDR is 10.37.0.23
 Options is 0
 Dead timer is due in 37 seconds
Neighbor 10.37.0.22 VRF default priority is 1, state is Full
 In area 0.0.0.0 interface vlan3911
 DR is 10.37.0.22 BDR is 10.37.0.23
 Options is 0
 Dead timer is due in 31 seconds
Neighbor 10.37.0.11 VRF default priority is 1, state is Full
 In area 0.0.0.0 interface vlan3902
 DR is 10.37.0.11 BDR is 10.37.0.22
 Options is 0
 Dead timer is due in 33 seconds
Neighbor 10.37.0.22 VRF default priority is 1, state is Full
 In area 0.0.0.0 interface vlan3902
 DR is 10.37.0.11 BDR is 10.37.0.22
 Options is 0
 Dead timer is due in 31 seconds
Neighbor 10.37.0.22 VRF default priority is 1, state is Full
 In area 0.0.0.0 interface vlan3923
 DR is 10.37.0.22 BDR is 10.37.0.46
 Options is 0
 Dead timer is due in 31 seconds
Neighbor 10.37.0.22 VRF default priority is 1, state is Full
 In area 0.0.0.0 interface vlan3908
 DR is 10.37.0.22 BDR is 10.37.0.21
 Options is 0
 Dead timer is due in 39 seconds
Neighbor 10.37.0.22 VRF default priority is 1, state is Full
 In area 0.0.0.2 interface vlan3992
 DR is 10.37.0.22 BDR is 10.37.0.23
 Options is 0
 Dead timer is due in 39 seconds
switch>
```

show ipv6 ospf neighbor state

The **show ipv6 ospf neighbor state** command displays the state information on OSPF neighbors on a per-interface basis.

Command Mode

EXEC

Command Syntax

show ipv6 ospf neighbor state *STATE_NAME* [***VRF_INSTANCE***]

Parameters

- ***STATE_NAME*** Values include:
  - **2-ways**
  - **attempt**
  - **down**
  - **exch-start**
  - **exchange**
  - **full**
  - **restart**
  - **init**
  - **loading**
- VRF_INSTANCE Values include:
  - <no parameter> Displays information for all VRFs.
  - **vrf** *vrf_name* Displays information for the specified VRF.

Examples

- This command displays OSPF information for neighboring devices that are adjacent .

```
switch>show ipv6 ospf neighbor state full
Routing Process "ospf 3":
switch>
```

show ipv6 ospf neighbor summary

The **show ipv6 ospf neighbor summary** command displays a single line of state information for each OSPFv3 neighbor.

Command Mode

EXEC

Command Syntax

show ipv6 ospf neighbor summary [***VRF_INSTANCE***]

Parameters

- VRF_INSTANCE Values include:
  - <no parameter> Displays information for all VRFs.
  - **vrf** *vrf_name* Displays information for the specified VRF.

Examples

- This command shows the summary information for the OSPFv3 neighbors.

```
switch>show ipv6 ospf neighbor summary
Routing Process "ospf 1":
      3 neighbors are in state Down
      3 neighbors are in state Full
      5 neighbors are in state Init
      0 neighbors are in state Loading
      0 neighbors are in state Attempt
      3 neighbors are in state Restarting
      0 neighbors are in state Exchange
      3 neighbors are in state 2 Ways
      0 neighbors are in state Exch Start
switch>
```

show ipv6 ospf spf-log

The **show ipv6 ospf spf-log** command displays when and how long the switch took to run a full SPF calculation for OSPFv3.

Command Mode

EXEC

Command Syntax

show ipv6 ospf [*PROCESS_ID*] spf-log [*VRF_INSTANCE*]

Parameters

- ***PROCESS_ID*** OSPFv3 process ID. Values include:
  - <no parameter> Displays information for all process IDs.
  - <***1*** to ***65535***> Displays information for the specified process ID.
- VRF_INSTANCE Values include:
  - <no parameter> Displays information for all VRFs.
  - **vrf** *vrf_name* Displays information for the specified VRF.

Examples

- This command displays the SPF information for OSPFv3 in all VRFs.

```
switch>show ipv6 ospf spf-log
OSPF3 Process 172.26.0.22, VRF default
TIME          EVENT                     REASON
04:54:52.070  SPF ran for 0.70 ms
04:54:52.070  Scheduled after 0 ms      Router LSA generation
04:54:39.151  SPF ran for 0.71 ms
04:54:39.151  Scheduled after 0 ms      Router LSA generation
04:54:12.071  SPF ran for 0.56 ms
04:54:12.070  Scheduled after 0 ms      Router LSA generation
04:54:04.153  SPF ran for 0.29 ms
04:53:59.153  Scheduled after 4999 ms   Router LSA generation
04:53:59.153  SPF ran for 0.25 ms
04:53:59.151  Scheduled after 0 ms      Router LSA generation
04:53:33.081  SPF ran for 0.3 ms
04:53:33.081  Scheduled after 0 ms      ECMP max nexthop cfg change
switch>
```

shutdown (OSPFv3)

The **shutdown** command disables OSPFv3 on the switch.

OSPFv3 is disabled by default on individual interfaces and enabled through **ipv6 ospf area** commands.

The **no shutdown** and **default shutdown** commands enable the OSPFv3 instance by removing the **shutdown** statement from the OSPFv3 block in ***running-config***.

Command Mode

Router-OSPFv3 Configuration

Command Syntax

```
shutdown
no shutdown
default shutdown
```

Example

- These commands disable OSPFv3 activity on the switch.

```
switch(config)#ipv6 router ospf 9
switch(config-router-ospf3)#shutdown
switch(config-router-ospf3)#show active
ipv6 router ospf 9
   shutdown
switch(config-router-ospf3)#
```

Chapter 29

Border Gateway Protocol (BGP)

Border Gateway Protocol (BGP) exchanges routing information among neighboring routers in different Autonomous Systems (AS). Arista switches use BGP version 4+, incorporating the multiprotocol extensions defined by RFC 4760 so that BGP can carry both IPv4 and IPv6 routes simultaneously over a single BGP peering.

This chapter contains the following sections.

- Section 29.1: BGP Conceptual Overview
- Section 29.2: Configuring BGP
- Section 29.3: BGP Examples
- Section 29.4: BGP Commands

Arista switches support these BGP functions:

- A single BGP instance
- Simultaneous internal (iBGP) and external (eBGP) peering
- Multiprotocol BGP
- BGP confederations

29.1 BGP Conceptual Overview

BGP is a protocol that exchanges routing information among neighboring routers in different autonomous systems through TCP sessions.

BGP neighbors (peers) communicate through a TCP session on port 179. They are established by manual configuration commands (static peers) or by creating a peer group listen range and accepting incoming peering requests in that range (dynamic peers). Internal BGP (iBGP) peers operate within a single autonomous system (AS). External BGP (eBGP) peers operate between autonomous systems. Border routers are on AS boundaries and exchange information with other autonomous systems; the primary function of border routers is distributing routes. Internal routers do not distribute route updates that they receive.

BGP defines a state machine for establishing connections. BGP routers maintain a state variable for each peer-to-peer session to track connection status. The state machine consists of these states:

- **Idle**: The router initializes BGP resources, refuses inbound BGP connection attempts, initiates a TCP connection to the peer, then transitions to the ***Connect*** state.
- **Connect**: The router waits for the TCP connection to complete, then sends an OPEN message to the peer and transitions to the ***OpenSent*** state if successful. If unsuccessful, it sets the ***ConnectRetry*** timer and transitions to the ***Active*** state upon expiry.
- **Active**: The router sets the ***ConnectRetry*** timer to zero and returns to the ***Connect*** state.
- **OpenSent:** The router waits for an OPEN message from the peer. After receiving a valid message, it transitions to the ***OpenConfirm*** state.
- **OpenConfirm**: The router waits for a keepalive message from its peer. If the message is received prior to a timeout expiry, the router transitions to the ***Established*** state. If the timeout expires or an error condition exists, the router transitions to the ***Idle*** state.
- **Established**: Peers exchange UPDATE messages about routes they advertise. If an UPDATE message contains an error, the router sends a NOTIFICATION message and transitions to the ***Idle*** state.

During established BGP sessions, routers exchange UPDATE messages about the destinations to which they offer connectivity. The route description includes the destination prefix, prefix length, autonomous systems in the path, the next hop, and information that affects the acceptance policy of the receiving router. UPDATE messages also list destinations to which the router no longer offers connectivity.

BGP detects and eliminates routing loops while making routing policy decisions by using the network topology as defined by AS paths and path attributes.

29.1.1 Multiprotocol BGP

Multiprotocol BGP facilitates the advertisement of network routes and switch capabilities to neighbors from multiple address families over a single BGP peering. The switch supports IPv4 unicast and IPv6 unicast address families.

Neighbors negotiate to select an address family when establishing a connection. The peer session is based on this address family, which identifies the following:

- The set of network layer protocols to which the address carried in the Next Hop field must belong.
- The encoding format of the next hop address.
- The semantics of Network Layer Reachability Information (NLRI).

29.1.2 BGP Confederations

BGP confederations divide an autonomous system (AS) into subsystems (sub-ASs), each identified by a unique sub-AS number, while still appearing externally as a single AS.

29.1.3 QoS Control of Neighbor Discovery and ARP Packets

To help prevent BGP sessions from being affected by dropped neighbor discovery and ARP packets, some Arista switches assign those packets to a higher priority output queue when they are being software forwarded. This helps minimize hardware drops from competition with data plane packets traffic congestion.

The switch platforms which use this feature are:

- DCS-7500E
- DCS-7250X
- DCS-7300X
- DCS-7010X
- DCS-7050X

29.1.4 Best-Path Selection

Routing information received via the BGP protocol often contains more than one route to the same destination: the BGP best-past selection algorithm determines which of these routes will be installed in the routing table. The following criteria are evaluated in order; at each step, if there is a tie for best path the next criterion is applied. If there is still a tie at the end of the process, BGP installs the route received from the peer with the lowest address. When equal cost multi path (ECMP) routing is enabled, multiple paths to a single destination may be installed in the IP routing table.

Route preferences can be shaped through configuration choices as described in Configuring Best-path Selection. The steps in Arista's best-path selection process are described below.

Step 1 Select the route that has the highest weight value. The weight of paths learned from a specific neighbor can be adjusted using the **neighbor weight** or **neighbor route-map (BGP)** command. Weight is only significant locally, and is not communicated in update messages.

Step 2 Select the route that has the highest LOCAL_PREF value.

Step 3 If **bgp bestpath as-path ignore** is disabled (the default), select the route that has the shortest AS path, excluding confederation segments; otherwise skip ahead to the next step.

Step 4 Select the route that has the lower ORIGIN value. The lowest ORIGIN value is IGP (Interior Gateway Protocol), which is better than EGP (Exterior Gateway Protocol), which is better than INCOMPLETE.

Note Exterior Gateway Protocol (EGP), as described in RFC 904, is the predecessor of BGP and is included in the selection algorithm for legacy reasons. It is rarely if ever used, and should not be confused with eBGP (see step 6 below).

Step 5 Select the route that has the lower MED value. The MED values are comparable if **bgp always-compare-med** is enabled or if the routes have the same neighbor AS. The neighbor AS of a path is determined as follows:

a If the router is in a confederation: if **bgp bestpath med confed** is configured, the confederation ID is used for the neighbor AS. Otherwise, the MED values of the routes are not comparable.

b If the router is not in a confederation: the first AS number in the first AS_SEQUENCE segment is used as the neighbor AS. If such a segment does not exist, the local router AS number is used.

Step 6 Prefer routes received from external peers (eBGP routes) over those received from internal peers (iBGP routes).

Step 7 Select the route that has the lowest IGP cost to the BGP NEXT_HOP. The IGP cost is determined using the metric of the resolving route for the BGP nexthop.

Step 8 If **bgp bestpath as-path multipath-relax** is enabled (default), then skip ahead to the next step; otherwise the AS_PATH path lengths are compared. The path length is determined by adding the total number of AS and segment types in the AS_PATH field. Prefer the route with the shorter path length. If path lengths are equal, then prefer the route with the first smaller non-matching AS, and then prefer the route with the first smaller non-matching segment type (sub-AS).

Step 9 If the selection process reaches this stage, then the routes being considered are eligible for multipathing.

a If multipath routes are being evaluated and **bgp bestpath ecmp-fast** is enabled (default) and none of the **bgp bestpath tie-break** configurations is enabled (default), then there is no preferred route in the multipath group and routes are not evaluated further. The effective ordering of routes in the multipath group is implementation-dependent and routes that were added to the multipath group first are preferred over routes that are added to the multipath group later. The path selection process ends here.

b If multipath routes are being evaluated and **bgp bestpath ecmp-fast** is not enabled, continue to step 10.

Step 10 If the routes are eligible for multipathing and none of the **bgp bestpath tie-break** configurations is enabled (default), skip ahead to step 11. Otherwise, continue as follows:

a If **bgp bestpath tie-break router-id** is enabled, prefer the route with the lowest ROUTER_ID. If the route is a reflected route (i.e., if it contains route reflector attributes), use the ORIGINATOR_ID as the ROUTER_ID for comparison.

b If **bgp bestpath tie-break cluster-list-length** is enabled, prefer the route with the shortest CLUSTER_LIST length. The cluster list length is assumed to be 0 if the route doesn't carry a CLUSTER_LIST attribute.

c If there isn't a preferred route yet, go to step 13.

Step 11 Prefer the route with the lowest ROUTER_ID. If the route is a reflected route (i.e., if it contains route reflector attributes), use the ORIGINATOR_ID as the ROUTER_ID for comparison.

Step 12 Prefer the route with the shortest CLUSTER_LIST length. The cluster list length is assumed to be 0 if the route doesn't carry a CLUSTER_LIST attribute.

Step 13 Prefer the route received from the lowest peer address.

Step 14 If the peer address is the same for routes under comparison and **bgp additional-path receive** is enabled (the default), then the routes should differ in the received path IDs. Prefer the route with the lower received path ID. The path selection process ends here.

29.2 Configuring BGP

These sections describe basic BGP configuration steps:

- Section 29.2.1: Configuring BGP Instances
- Section 29.2.2: Configuring BGP Neighbors
- Section 29.2.3: Configuring Routes
- Section 29.2.4: Configuring Address Families
- Section 29.2.5: Configuring Best-path Selection
- Section 29.2.6: BGP Confederations
- Section 29.2.7: BGP Operational Commands

29.2.1 Configuring BGP Instances

29.2.1.1 Creating an Instance and Entering BGP Configuration Mode

The switch supports one BGP instance, which is associated with a specified autonomous system (AS). To other BGP peers, the AS number uniquely identifies the network to which the switch belongs. Arista switches support four-byte AS numbers as described in RFC 4893. Four-byte AS number capability is communicated to BGP peers in OPEN messages. When communicating with a BGP peer which does not support four-byte AS numbers, the switch will replace AS numbers greater than 65535 with the well-known two-byte AS number 23456 (also called AS_TRANS), and encode the actual four-byte AS numbers using the AS4_PATH attribute.

The switch must be in router-BGP configuration mode to run BGP configuration commands. The **router bgp** command places the switch in router-BGP configuration mode and creates a BGP instance if one was not previously created. BGP configuration commands apply globally to the BGP instance.

Example

- This command places the switch in router-BGP configuration mode. It also creates a BGP instance in AS 50 if an instance was not previously created.

```
switch(config)#router bgp 50
switch(config-router-bgp)#
```

When a BGP instance exists, the **router bgp** command must include its autonomous system. Any attempt to create a second instance results in an error message.

Example

- This command attempts to open a BGP instance with a different AS number from that of the existing instance. The switch displays an error and stays in global configuration mode.

```
switch(config)#router bgp 100
% BGP is already running with AS number 50
switch(config)#
```

29.2.1.2 Entering BGP VRF Configuration Mode

IPv6 VRF support in EOS allows application of a BGP configuration to a single VRF instance, overriding global commands. To apply VRF-specific BGP configuration, use the **vrf** command to enter BGP VRF configuration mode. IPv6 BGP VRF configuration is performed in the VRF submode of the router-bgp configuration mode.

Examples

- This command places the switch in BGP VRF configuration mode for VRF “purple.” Commands issued in this mode will override global BGP configuration for the specified VRF instance.

```
switch(config-router-bgp)#vrf purple
```

- This command activates IPv6 address-family support for the IPv6 neighbor 2001:0DB8:8c01::1 in the red VRF.

```
switch(config)#router bgp 1
switch(config-router-bgp)#vrf red
switch(config-router-bgp-vrf-red)#router-id 1.1.1.1
switch(config-router-bgp-vrf-red)#neighbor 2001:0DB8:8c01::1 remote-as 16
switch(config-router-bgp-vrf-red)#address-family ipv6
switch(config-router-bgp-vrf-red-af)#neighbor 2001:0DB8:8c01::1 activate
```

29.2.2 Configuring BGP Neighbors

29.2.2.1 Establishing BGP Neighbors

BGP neighbors, or peers, are established by configuration commands that initiate a TCP connection. BGP supports two types of neighbors:

- Internal neighbors are in the same autonomous system.
- External neighbors are in different autonomous systems.

BGP neighbors can be either static or dynamic:

- Static neighbors are established by manually configuring the connection.
- Dynamic neighbors are established by creating a listen range and accepting incoming connections from neighbors in that address range.

Static neighbors may belong to a static peer group, allowing them to be configured as a group. Configuration applied to an individual member of a static peer group overrides the group configuration for that peer. Dynamic neighbors *must* belong to a dynamic peer group, and can only be configured as a group.

Static BGP Neighbors

The **neighbor remote-as** command connects the switch with a peer, establishing a static neighbor.

Once established, a static neighbor may be added to an existing peer group. Any configuration applied to the peer group then is inherited by the neighbor, unless a conflicting configuration has been entered for that peer. Settings applied to a member of the peer group override group settings.

Note To establish a BGP session, there must be an IPv4 router ID configured in the same VRF or at least one L3 interface with an IPv4 address in the same VRF. If the VRF contains no L3 interfaces with IPv4 addresses (e.g., in an IPv6-only environment), configure an appropriate router ID using the **router-id (BGP)** command.

Example

- These commands establish an internal BGP connection with the peer at 10.1.1.14.

```
switch(config)#router bgp 50
switch(config-router-bgp)#neighbor 10.1.1.14 remote-as 50
switch(config-router-bgp)#
```

- These commands establish an external BGP connection with the peer at 192.168.2.5.

```
switch(config)#router bgp 50
switch(config-router-bgp)#neighbor 192.168.2.5 remote-as 100
switch(config-router-bgp)#
```

Dynamic BGP Neighbors

The **bgp listen range** command specifies a range of IPv4 addresses from which the switch will accept incoming dynamic BGP peering requests, and creates the named dynamic peer group to which those peers belong. Dynamic BGP neighbors are peers which have not been manually established, but are accepted into a dynamic peer group when the switch receives a peering request from them.

Dynamic peers cannot be configured individually, but inherit any configuration that is applied to the peer group to which they belong. Peering relationships with dynamic peers are terminated if the peer group is deleted.

Example

- These commands create a peer group called “brazil” which accepts dynamic peering requests from the 192.168.2.0/24 subnet.

```
switch(config)#router bgp 50
switch(config-router-bgp)#bgp listen range 192.168.2.0/24 peer-group brazil
remote-as 50
switch(config-router-bgp)#
```

Displaying Neighbor Connections

The **show ip bgp summary** and **show ip bgp neighbors** commands display neighbor connection status.

Example

- This command indicates the connection state with the peer at 192.168.2.5 is ***Estab*** (established). The peer is an external neighbor because it is in AS 100 and the local server is in AS 50.

```
switch>show ip bgp summary
BGP summary information for VRF default
BGP router identifier 192.168.104.2, local AS number 50
Neighbor Status Codes: m - Under maintenance
  Neighbor     V  AS  MsgRcvd  MsgSent  InQ OutQ  Up/Down State  PfxRcd PfxAcc
  192.168.2.5  4  100     198      281    0    0 03:11:31 Estab  12     12
switch>
```

Static BGP Peer Groups

A static BGP peer group is a collection of BGP neighbors which can be configured as a group. Once a static peer group is created, the group name can be used as a parameter in neighbor configuration commands, and the configuration will be applied to all members of the group. Neighbors added to the group will inherit any settings already created for the group. Static peer group members may also be configured individually, and the settings of an individual neighbor in the peer group override group settings for that neighbor.

When the **default** form of a BGP configuration command is entered for a member of a static peer group, the peer inherits that configuration from the peer group.

A static peer group is created with the **neighbor peer-group (create)** command, or by using the **bgp listen range** command to accept dynamic peering requests. Once a static peer group has been created, static neighbors can be manually added to the group by using the **neighbor peer-group (neighbor assignment)** command. The **no neighbor peer-group (neighbor assignment)** command removes a neighbor from a static peer group.

The **no neighbor peer-group (create)** command will delete a static peer group. When a peer group is deleted, the members of that group revert to their individual configurations, or to the system default for any attributes that have not been specifically configured for that peer.

Examples

- These commands create a peer group named “akron.”

```
switch(config)#router bgp 50
switch(config-router-bgp)#neighbor akron peer-group
switch(config-router-bgp)#
```

- This command adds the neighbors at 1.1.1.1 and 2.2.2.2 to peer group akron.

```
switch(config-router-bgp)#neighbor 1.1.1.1 peer-group akron
switch(config-router-bgp)#neighbor 2.2.2.2 peer-group akron
switch(config-router-bgp)#
```

- These commands configure the members of peer group akron, but cause the neighbor at 1.1.1.1 to use the system default value for out-delay.

```
switch(config-router-bgp)#neighbor akron remote-as 109
switch(config-router-bgp)#neighbor akron out-delay 101
switch(config-router-bgp)#neighbor akron maximum-routes 12000
switch(config-router-bgp)#no neighbor 1.1.1.1 out-delay
switch(config-router-bgp)#
```

Dynamic BGP Peer Groups

A dynamic BGP peer group is a collection of BGP neighbors in a specified address range which have made peering requests to the switch. Members of a dynamic peer group cannot be configured individually but must be configured as a group. Once a dynamic peer group is created, the group name can be used as a parameter in neighbor configuration commands, and the configuration will be applied to all members of the group. Neighbors joining the group will also inherit any settings already created for the group.

A dynamic peer group is created with the **bgp listen range** command, which identifies a range of IPv4 addresses from which the switch will accept incoming dynamic BGP peering requests, and names the dynamic peer group to which those peers will belong. To delete a dynamic peer group, use the **no** or **default** form of the **bgp listen range** command. All peering relationships with group members are terminated when the dynamic peer group is deleted.

Example

- These commands create a dynamic peer group called “brazil” in AS 5 which accepts peering requests from the 192.0.2.0/24 subnet.

```
switch(config)#router bgp 1
switch(config-router-bgp)#bgp listen range 192.0.2.0/24 peer-group brazil
remote-as 5
```

29.2.2.2 Special Considerations for IPv6

BGP predates the use of IPv6, and BGP configuration assumes IPv4 connections by default. The following additional steps are used to configure IPv6 BGP neighbors.

Note To establish a BGP session, there must be an IPv4 router ID configured in the same VRF or at least one L3 interface with an IPv4 address in the same VRF. If the VRF contains no L3 interfaces with IPv4 addresses (e.g., in an IPv6-only environment), configure an appropriate router ID using the **router-id (BGP)** command.

Activating IPv6 Neighbors

By default, the switch does not negotiate or advertise IPv6 BGP routes. In order to establish a session with an IPv6 neighbor, it must be made active in the IPv6 address family. The **bgp default ipv6-unicast** command causes the switch to send IPv6 capability messages and all network advertisements with IPv6 prefixes to all BGP neighbors. The **neighbor activate** command issued in IPv6 address family configuration mode does the same for a single BGP neighbor.

Examples

- These commands make all BGP neighbors active in the IPv6 address family.

```
switch(config)#router bgp 11
switch(config)#address-family ipv6
switch(config-router-bgp-af)#bgp default ipv6-unicast
switch(config-router-bgp-af)#exit
switch(config-router-bgp)#
```

- These commands make the BGP neighbor at 2001:0DB8:8c01::1 active in the IPv6 address family.

```
switch(config)#router bgp 11
switch(config)#address-family ipv6
switch(config-router-bgp-af)#neighbor 2001:0DB8:8c01::1 activate
switch(config-router-bgp-af)#exit
switch(config-router-bgp)#
```

Sending IPv4 NLRIs over IPv6 Connections

The switch supports the exchange of IPv4 NLRIs with IPv6 neighbors. To enable this feature for all IPv6 neighbors, use the command **bgp default ipv4-unicast transport ipv6** in IPv4 address family configuration mode. To enable it for a single IPv6 neighbor, use the **neighbor activate** command for that neighbor in IPv4 address family configuration mode.

To send IPv4 NLRIs to IPv6 neighbors, the IPv4 next-hop address must also be communicated. To explicitly configure an IPv4 next hop to send to a specific IPv6 neighbor, use the **neighbor local-v4-addr** command. In some network configurations, the switch can also be configured to automatically determine the best IPv4 next-hop address for an individual IPv6 neighbor or for all neighbors in the VRF using the **neighbor auto-local-addr** command.

Examples

- These commands permit IPv4 NLRI transport over all IPv6 connections by making the IPv4 address family active on IPv6 BGP neighbors, then configure the switch to automatically select a local IPv4 address to be sent in NLRIs to the IPv6 neighbors in a peer group called “indianapolis”.

```
switch(config)#router bgp 11
switch(config-router-bgp)#address-family ipv4
switch(config-router-bgp-af)#bgp default ipv4-unicast transport ipv6
switch(config-router-bgp-af)#exit
switch(config-router-bgp)#neighbor indianpolis auto-local-addr
switch(config-router-bgp)#
```

- These commands permit IPv4 NLRI transport with the IPv6 neighbor at 2001:0DB8:8c01::1 using a local IPv4 address of 10.7.5.11.

```
switch(config)#router bgp 11
switch(config-router-bgp)#address-family ipv4
switch(config-router-bgp-af)#neighbor 2001:0DB8:8c01::1 activate
switch(config-router-bgp-af)#exit
switch(config-router-bgp)#neighbor 2001:0DB8:8c01::1 local-v4-addr 10.7.5.11
switch(config-router-bgp)#
```

29.2.2.3 Maintaining Neighbor Connections

BGP neighbors maintain connections by exchanging keepalive, UPDATE, and NOTIFICATION messages. Neighbors that do not receive a message from a peer within a specified period (***hold time***) close the BGP session with that peer. Hold time is typically three times the period between scheduled keepalive messages. The default keepalive period is 60 seconds; default hold time is 180 seconds.

The **timers bgp** command configures the hold time and keepalive period. A peer retains its BGP connections indefinitely when its hold time is zero.

Example

- This command sets the keepalive period to 15 seconds and the hold time to 45 seconds.

```
switch(config-router-bgp)#timers bgp 15 45
switch(config-router-bgp)#
```

The **show ip bgp neighbors** command displays the hold time.

Example

- This command indicates the BGP hold time is 45 seconds.

```
switch>show ip bgp neighbors 10.100.100.2
BGP neighbor is 10.100.100.2, remote AS 100
BGP version 4, remote router ID 192.168.100.13, VRF default
  Negotiated BGP version 4
  Last read 00:00:05, last write 00:00:05
  Hold time is 45, keepalive interval is 15 seconds              <= hold time
  Configured hold time is 45, keepalive interval is 15 seconds
  Connect timer is inactive
  Idle-restart timer is inactive
  BGP state is Established, up for 04:44:05
  Number of transitions to established: 11
  Last state was OpenConfirm
  Last event was RecvKeepAlive
  Last sent notification:Cease/administrative reset, Last time 04:44:09
  Last rcvd notification:Cease/peer de-configured, Last time   2d02h, First time
7d08h, Repeats 1
  Neighbor Capabilities:
    Multiprotocol IPv4 Unicast: advertised and received and negotiated
    Four Octet ASN: advertised and received
              <-------OUTPUT OMITTED FROM EXAMPLE-------->
switch>
```

29.2.2.4 Neighbor – Route Configuration

Maximum Routes

The **neighbor maximum-routes** command determines the number of BGP routes the switch accepts from a specified neighbor. The switch disables peering with the neighbor when this number is exceeded.

Example

- This command configures the switch to accept 15,000 routes from the peer at 192.168.18.24.

```
switch(config-router-bgp)#neighbor 192.168.18.24 maximum-routes 15000
switch(config-router-bgp)#
```

Route Reflection

Participating BGP routers within an AS communicate eBGP-learned routes to all of their peers; they do not re-advertise iBGP-learned routes within the AS to prevent routing loops. Although a fully meshed network topology ensures that all AS members share routing information, this topology can result in high volumes of iBGP messages when scaled. Alternatively, one or more routers are configured as route reflectors in larger networks.

A route reflector re-advertises routes learned through iBGP to a group of BGP neighbors within the AS, replacing the function of a fully meshed topology. The **neighbor route-reflector-client** command configures the switch to act as a route reflector and configures the specified neighbor as a client. The **bgp client-to-client reflection** command enables client-to-client reflection.

When using route reflectors, an AS is divided into clusters. A cluster contains at least one route reflector and a group of clients to which they re-advertise route information. A cluster may contain multiple route reflectors to provide redundancy protection. Each reflector has a cluster ID. When the cluster has a single route reflector, the cluster ID is its router ID. When a cluster has multiple route reflectors, a 4-byte cluster ID is assigned to all route reflectors in the cluster, allowing them to recognize updates from other cluster reflectors. The **bgp cluster-id** command configures the cluster ID in a cluster with multiple route reflectors.

Example

- These commands configure the switch as a route reflector and the neighbor at 172.72.14.5 as one of its clients, and set the cluster ID to 172.22.30.101.

```
switch(config-router-bgp)#neighbor 172.72.14.5 route-reflector-client
switch(config-router-bgp)#bgp cluster-id 172.22.30.101
switch(config-router-bgp)#
```

Usually the clients of a route reflector are not interconnected, and any routes learned by a client are mirrored to other clients and re-advertised within the AS by the route reflector. If the clients of a route reflector are fully meshed, routes received from a client do not need to be mirrored to other clients. In this case, client-to-client reflection should be disabled (**no bgp client-to-client reflection**).

Route Preference

The primary function of external peers is to distribute routes they learn from their peers. Internal peers receive route updates without distributing them. External peers receive route updates, then distribute them to internal and external peers.

Local preference is a metric that iBGP sessions use to select an external route. Preferred routes have the highest local preference value. UPDATE packets include this metric in the LOCAL_PREF field.

The **neighbor export-localpref** command specifies the LOCAL_PREF that the switch sends to an internal peer. The command overrides previously assigned preferences and has no effect on external peers.

Example

- This command configures the switch to enter 200 in the LOCAL_PREF field of UPDATE packets it sends to the peer at 10.1.1.45.

```
switch(config-router-bgp)#neighbor 10.1.1.45 export-localpref 200
switch(config-router-bgp)#
```

The **neighbor import-localpref** command assigns a local preference to routes received through UPDATE packets from an external peer. This command has no effect when the neighbor is an internal peer.

Example

- This command configures the switch to assign the local preference of 50 for routes advertised from the peer at 172.16.5.2.

```
switch(config-router-bgp)#neighbor 172.16.5.2 import-localpref 50
switch(config-router-bgp)#
```

The **show ip bgp** command displays the LOCAL_PREF value for all listed routes.

Example

- This command indicates the route to network 10.10.20.0/24 has a local preference of 400.

```
switch#show ip bgp
BGP routing table information for VRF default
Router identifier 192.168.100.23, local AS number 64512
Route status codes: s - suppressed, * - valid, > - active, # - not installed, E
- ECMP head, e - ECMP
                    S - Stale, c - Contributing to ECMP, b - backup, L -
labeled-unicast
Origin codes: i - IGP, e - EGP, ? - incomplete
AS Path Attributes: Or-ID - Originator ID, C-LST - Cluster List, LL Nexthop -
Link Local Nexthop

        Network           Next Hop         Metric  LocPref Weight  Path
 * >Ec  10.10.20.0/24     192.168.31.3     0       400     0       64521 i
switch#
```

Graceful Restart

Graceful BGP restart allows a BGP speaker with separate control plane and data plane processing to continue forwarding traffic during a BGP restart. Its neighbors (receiving speakers) may retain routing information from the restarting speaker while a BGP session with it is being re-established, reducing route flapping.

Arista switches can act as helpers (receiving speakers) for graceful BGP restart with neighbors that advertise graceful restart capability.

Graceful restart helper mode is enabled by default, but can be turned off globally with the **no graceful-restart-helper** command. Per-peer configuration takes precedence over the global configuration.

Example

- This command disables graceful restart helper mode for all BGP peers.

```
switch(config-router-bgp)#no graceful-restart-helper
switch(config-router-bgp)#
```

- This command disables graceful restart helper mode for the neighbor at 192.168.32.5 regardless of global configuration.

```
switch(config-router-bgp)#no neighbor 192.168.32.5 graceful-restart-helper
switch(config-router-bgp)#
```

Peers with graceful restart capability advertise a restart time value as an estimate of the time it will take them to restart a BGP session. When a BGP session with a restarting speaker goes down, the switch (receiving speaker) marks routes from that peer as stale and starts the restart timer. If the session with the peer is not re-established before the restart time runs out, the switch deletes the stale routes from that peer. If the session is re-established within that time, the stale path timer is started. If the stale paths are not updated by the restarting speaker before the stale path time runs out, they are deleted. The maximum time these stale paths will be retained after the BGP session is re-established is 300 seconds by default, but can be configured using the **graceful-restart stalepath-time** command.

Example

- This command configures BGP to discard stale paths from a restarting peer 500 seconds after the BGP session with that peer is re-established.

```
switch(config-router-bgp)#graceful-restart stalepath-time 500
switch(config-router-bgp)#
```

29.2.2.5 Filtering Routes

Route Maps

Route maps are used in BGP to directly filter IPv4 unicast routes. The **neighbor route-map (BGP)** command applies a route map to inbound or outbound BGP routes. To display the route maps associated with a specific BGP neighbor, use the **show ip bgp neighbors** command.

AS Path Access Lists

An AS path access list is a named list of permit and deny statements which use regular expressions to filter BGP routes based on their AS path attribute. AS path access lists are created using the **ip as-path access-list** command, and are applied using a route map ***match*** clause with the name of the access list as a parameter.

Example

- These commands create an AS path access list identifying routes which pass through AS 3, create a route map which references the access list, assign the routes it filters to community 300, and apply the route map to the neighbor at 192.68.14.5 to assign a community value of 300 to inbound routes received from that neighbor.

Step 1 Create the AS path access list.

```
switch(config)#ip as-path access-list as_list3 permit _3_
```

Step 2 Create a route map that matches the AS path access list and sets the community value.

```
switch(config)#route-map MAP_3 permit
switch(config-route-map-MAP_3)#match as-path as_list3
switch(config-route-map-MAP_3)#set community 300
switch(config-route-map-MAP_3)#exit
```

Step 3 Apply the route map to the neighbor.

```
switch(config)#router bgp 1
switch(config-router-bgp)#neighbor 192.68.14.5 route-map MAP_3 in
switch(config-router-bgp)#
```

BGP Communities

A BGP community is a group of subnet address prefixes that share a common identifying attribute. Communities simplify routing policies by consolidating IP network spaces into logical entities that BGP speakers can address to accept, prefer, and distribute routing information.

The BGP community attribute is a 32 bit value formatted as follows:

- an integer between 0 and 4294967040.
- AA:NN, where AA specifies an Autonomous System number (0-4294967295) and NN specifies a community number (0-65535) within the AS.

These four community attribute values, and the associated BGP speaker actions, are predefined:

- **no-export**: speaker does not advertise the routes beyond the BGP domain.
- **no-advertise**: speaker does not advertise the routes to any BGP peers.
- **local-as**: speaker does not advertise route to any external peers.
- **internet**: speaker advertises the route to the Internet community. By default, this includes all prefixes.

Community values are assigned to a set of subnet prefixes through route map **set** commands. Route map **match** commands subsequently use community values to filter routes. The switch uses the following **ip community-list** commands to filter community routes into a BGP domain:

- **ip community-list standard**
- **ip community-list expanded**
- **ip extcommunity-list standard**
- **ip extcommunity-list expanded**

  Standard community lists refer to route maps by name or number. Expanded community lists reference route maps through regular expressions.

Example

- These commands assign two network subnets to a prefix list, assign a community number to the prefix list members, then utilize that community in an **ip community-list** command to permit the routes into the BGP domain.

Step 1 Compose the IP prefix list.

```
switch(config)#ip prefix-list PL_1 permit 10.1.2.5/24
switch(config)#ip prefix-list PL_1 permit 10.2.5.1/28
switch(config)#
```

Step 2 Create a route map that matches the IP prefix list and sets the community value.

```
switch(config)#route-map MAP_1 permit
switch(config-route-map-MAP_1)#match ip address prefix-list PL_1
switch(config-route-map-MAP_1)#set community 500
switch(config-route-map-MAP_1)#exit
```

Step 3 Create a community list that references the community.

```
switch(config)#ip community-list standard CL_1 permit 500
switch(config)#
```

BGP extended communities identify routes for VRFs. Extended community clauses utilize route target (rt) and site of origin options (soo):

- **route targets** identify sites that may receive appropriately tagged routes.
- **site of origin** identifies the site where the router learned the route.

29.2.3 Configuring Routes

29.2.3.1 Advertising Routes

A BGP neighbor advertises routes it can reach through UPDATE packets. The **network (BGP)** command specifies a prefix that the switch advertises as a route originating from its AS.

The configuration clears the host portion of addresses entered in **network** commands. For example, 192.0.2.4/24 is stored as 192.0.2.0/24.

Example

- This command configures the switch to advertise the 10.5.8.0/24 network.

```
switch(config-router-bgp)#network 10.5.8.0/24
switch(config-router-bgp)#
```

By default, BGP will advertise only those routes that are active in the switch's RIB. This can contribute to dropped traffic. If a preferred route is available through another protocol (like OSPF), the BGP route will become inactive and not be advertised; if the preferred route is lost, there is no available route to the affected peers. Advertising inactive BGP routes minimizes traffic loss by providing alternative routes.

The **bgp advertise-inactive** command causes BGP to advertise inactive routes to BGP neighbors. Inactive route advertisement is configured globally, but the global setting can be overridden on a per-VRF basis.

Examples

- This command configures the switch to advertise routes learned through BGP even if they are not active on the switch.

```
switch(config-router-bgp)#bgp advertise-inactive
switch(config-router-bgp)#
```

- This command overrides inactive route advertisement for VRF "purple."

```
switch(config-router-bgp)#vrf purple
switch(config-router-bgp-vrf-purple)#no bgp advertise-inactive
switch(config-router-bgp-vrf-purple)#
```

29.2.3.2 BGP Route Aggregation

Aggregation combines the characteristics of multiple routes into a single route for advertisement by the BGP speaker. Aggregation can reduce the amount of information that a BGP speaker is required to store and transmit when advertising routes to other BGP speakers. Aggregation options affect the attributes associated with the aggregated route, the advertisement of the contributor routes that comprise the aggregate, and which contributor routes are included.

Aggregate routes are created with the **aggregate-address** command, which takes an IP subnet as an argument; any routes configured on the switch that lie within that subnet then become contributors to the aggregate. Note that on Arista switches the BGP aggregate route will become active if there are any available contributor routes on the switch, regardless of the originating protocol. This includes routes configured statically.

BGP speakers display aggregate routes that they create as null routes (with one exception: if all the contributors to the aggregate have the same BGP path attributes, then the BGP aggregate copies those attributes and is no longer a null route). Aggregate routes are advertised into the BGP autonomous system and redistributed automatically, and their redistribution cannot be disabled. BGP neighbors display inbound aggregate routes as normal BGP routes. Null routes are displayed with the **show ip route** command; normal BGP routes (and null aggregate routes) are displayed with the **show ip bgp** and **show ip route** commands.

Aggregation Options

The **aggregate-address** command provides the following aggregate route options:

- AS_PATH attribute inclusion: the ***as-set*** option controls the aggregate route's AS_PATH and ATOMIC_AGGREGATE attribute contents. AS_PATH identifies the autonomous systems through which UPDATE message routing information passes. ATOMIC_AGGREGATE indicates that the route is an aggregate or summary of more specific routes.

  When the command includes ***as-set***, the aggregate route's AS_SET attribute contains the AS numbers of contributor routes. This can help BGP neighbors to prevent loops by rejecting aggregate routes that include their AS number in the AS_SET.

  When the command does not include ***as-set***, the aggregate route's ATOMIC_AGGREGATE attribute is set and the AS_PATH attribute does not include AS numbers of contributing routes.
- Attribute assignment: The ***attribute-map*** option assigns attributes contained in set commands in a specified route map's lowest sequence with any set command to the aggregated route, overriding the automatic determination of the aggregate route's attributes by the switch.
- Route suppression: The ***summary-only*** option suppresses the advertisement of the contributor routes that comprise the aggregate.
- Contributor filtering: The ***match-map*** option uses a route map to filter out contributor routes that would otherwise be included in the aggregate.

Example

- These commands create an aggregate route (10.16.48.0/20) from four contributor routes (10.16.48.0/23, 10.16.50.0/23, 10.16.52.0/23, and 10.16.54.0/23). The aggregate route includes the AS_PATH information from the contributor routes.

```
switch(config)#router bgp 1
switch(config-router-bgp)#aggregate-address 10.16.48.0/20 as-set
switch(config-router-bgp)#exit
switch(config)#
```

- These commands create an aggregate route and use a route map to add a local-preference attribute to the route.

```
switch(config)#route-map map1 permit 10
switch(config-route-map-map1)#set local-preference 40
switch(config-route-map-map1)#exit
switch(config)#router bgp 1
switch(config-router-bgp)#aggregate-address 10.16.48.0/20 attribute-map map1
switch(config-router-bgp)#exit
switch(config)#
```

- These commands create an aggregate route and use a route map to allow only those contributors which match a specified prefix list to be included in the aggregate route.

```
switch(config)#route-map matchmap permit 10
switch(config-route-map-matchmap)#match ip address prefix-list agglist
switch(config-route-map-matchmap)#exit
switch(config)#router bgp 1
switch(config-router-bgp)#aggregate-address 1.1.0.0/16 match-map matchmap
```

29.2.4 Configuring Address Families

The switch determines the network prefixes that peering sessions advertise and the BGP neighbor addresses that receive advertisements through address family activity configuration.

An address family is a data structure that defines route advertising status to BGP neighbor addresses. Each BGP neighbor address is assigned an activity level for each address family on the switch. The switch sends capability and network prefix advertisements to neighbor addresses that are active within specified address families:

- IPv4 address family: switch advertises IPv4 capability and network commands with IPv4 prefixes to neighbor addresses configured as ***IPv4 address family active***.
- IPv6 address family: switch advertises IPv6 capability and network commands with IPv6 prefixes to neighbor addresses configured as ***IPv6 address family active***.

29.2.4.1 Neighbor Address Family Configuration

Address family activity levels for neighbor addresses are configured through **bgp default** and **neighbor activate** commands.

- The **bgp default** command specifies the default activity level of BGP neighbor addresses for a specified address family.
- The **neighbor activate** command specifies deviations from default address family activity level for a specified BGP neighbor address.

Default neighbor activation

The **bgp default** command configures the default address family activity level of all configured BGP neighbor addresses. The switch advertises the following to ***address family active*** addresses:

- IPv4 address family active: IPv4 capability and all network advertisements with IPv4 prefixes.
- IPv6 address family active: IPv6 capability and all network advertisements with IPv6 prefixes.

These commands configure default address family activity levels for configured BGP neighbor addresses:

- **bgp default ipv4-unicast** all BGP neighbor addresses are IPv4 address family active (this is the switch default).
- **no bgp default ipv4-unicast** no BGP neighbor addresses are IPv4 address family active.
- **bgp default ipv6-unicast** all BGP neighbor addresses are IPv6 address family active.
- **no bgp default ipv6-unicast** no BGP neighbor addresses are IPv6 address family active (this is the switch default).
- **bgp default ipv4-unicast transport ipv6** all BGP neighbor addresses are IPv4 address family active and IPv6 neighbors can receive IPv4 NLRIs.

Note If it is necessary to exchange IPv4 NLRIs over an IPv6 connection, the IPv4 address family must be activated on the IPv6 neighbor. To do this for all IPv6 neighbors, use the command **bgp default ipv4-unicast transport ipv6**. For an individual neighbor, use the **neighbor activate** command for the IPv6 neighbor in the IPv4 address-family configuration mode as described below.

Activating Individual Neighbor Addresses

The **address-family** command places the switch in address family mode to configure the address family activity level of individual BGP neighbor addresses. The switch supports these address families:

- ipv4-unicast
- ipv6-unicast

Running-config displays address family commands in sub-blocks of the BGP configuration. The **neighbor activate** command is available in each address family configuration mode and defines the configuration mode address family activity level of a specified configured BGP neighbor address. Addresses are assigned one of the following states by the activate command:

- **neighbor activate** configures the address as active in the configuration mode address family.
- **no neighbor activate** configures the address as not active in the configuration mode address family.

The switch sends the following announcements to addresses that are active in an address family:

- IPv4 address family: IPv4 capability and all network routes with IPv4 prefixes.
- IPv6 address family: IPv6 capability and all network routes with IPv6 prefixes.

The **neighbor route-map (BGP)** command applies a route map to inbound or outbound BGP routes. In address-family mode, the route map is applied to routes corresponding to the configuration mode address family. When a route map is applied to outbound routes, the switch advertises only routes matching at least one section of the route map. One outbound and one inbound route map can be applied to a neighbor for each address family. Applying a route map to a route replaces the previous corresponding route map assignment.

Network Route Advertising in Address Families

The **network (BGP)** command specifies a network for advertisement through UPDATE packets to BGP peers. The command is available in Router-BGP and Router-BGP-Address-Family configuration modes; the mode in which the command is issued does not affect the command's execution.

- Commands with an IPv4 address are advertised to peers that are IPv4 address family-active.
- Commands with an IPv6 address are advertised to peers that are IPv6 address family-active.

Examples

- These commands instantiate BGP, configure three neighbors, and configure 2 network routes.

  The default activity level for IPv4 and IPv6 address families is set to the default; all neighbor addresses are IPv4 address family active and IPv6 address family not active. IPv4 capability and network routes with IPv4 prefixes are advertised to all neighbor IPv4 addresses.

```
switch(config)#router bgp 9
switch(config-router-bgp)#neighbor 172.21.14.8 remote-as 15
switch(config-router-bgp)#neighbor 172.23.18.6 remote-as 16
switch(config-router-bgp)#neighbor 2001:0DB8:8c01::1 remote-as 16
switch(config-router-bgp)#network 172.18.23.9/24
switch(config-router-bgp)#network 2001:0DB8:de29::/64
switch(config-router-bgp)#
```

- These commands instantiate BGP on the switch, set IPv4 default activity level (***not active***), set IPv6 default activity level (***active***), and configure three neighbor addresses and two network route prefixes.

  IPv6 capability and network routes with IPv6 prefixes are advertised to all neighbor addresses.

```
switch(config)#router bgp 10
switch(config-router-bgp)#bgp default ipv6-unicast
switch(config-router-bgp)#no bgp default ipv4-unicast
switch(config-router-bgp)#neighbor 172.21.14.8 remote-as 15
switch(config-router-bgp)#neighbor 172.23.18.6 remote-as 16
switch(config-router-bgp)#neighbor 2001:0DB8:8c01::1 remote-as 16
switch(config-router-bgp)#network 172.18.23.9/24
switch(config-router-bgp)#network 2001:0DB8:de29::/64
switch(config-router-bgp)#
```

- These commands configure three neighbors, two network routes, and the default activity level for each address family (***not active***), and specify neighbor addresses for each address family that is active.

```
switch(config)#router bgp 11
switch(config-router-bgp)#neighbor 172.21.14.8 remote-as 15
switch(config-router-bgp)#neighbor 172.23.18.6 remote-as 16
switch(config-router-bgp)#neighbor 2001:0DB8:8c01::1 remote-as 16
switch(config-router-bgp)#network 172.18.23.9/24
switch(config-router-bgp)#network 2001:0DB8:de29::/64
switch(config-router-bgp)#no bgp default ipv4-unicast
switch(config-router-bgp)#no bgp default ipv6-unicast
switch(config-router-bgp)#address-family ipv4
switch(config-router-bgp-af)#neighbor 172.21.14.8 activate
switch(config-router-bgp-af)#neighbor 172.23.18.6 activate
switch(config-router-bgp-af)#exit
switch(config-router-bgp)#address-family ipv6
switch(config-router-bgp-af)#neighbor 2001:0DB8:8c01::1 activate
switch(config-router-bgp-af)#exit
switch(config-router-bgp)#
```

- These commands permit IPv4 NLRI transport over all IPv6 connections by making the IPv4 address family active on IPv6 BGP neighbors.

```
switch(config)#router bgp 11
switch(config)#address-family ipv4
switch(config-router-bgp-af)#bgp default ipv4-unicast transport ipv6
switch(config-router-bgp-af)#exit
switch(config-router-bgp)#
```

29.2.5 Configuring Best-path Selection

The best-path selection algorithm (described under Best-Path Selection) determines which of multiple paths to the same destination received by BGP will be added to the IP routing table. To shape route preferences and influence best-path selection, use the following commands in router-BGP configuration mode.

- **bgp always-compare-med** configures the switch to always consider the multi-exit discriminator (MED) value when comparing paths (disabled by default).
- **bgp bestpath as-path ignore** configures the switch to ignore the length of the autonomous system (AS) path when comparing routes (disabled by default).
- **bgp bestpath as-path multipath-relax** used in equal-cost multi path (ECMP configuration) and enabled by default; the **no** form of the command configures the switch to consider paths unequal if their AS paths have different contents.

- **bgp bestpath ecmp-fast** the **no** form of this command causes the switch to ignore order of arrival in evaluating paths within an ECMP group.
- **bgp bestpath med confed** causes comparison of multi-exit discriminator (MED) values in routes originating within the same confederation as the switch and received from confederation peers (disabled by default).
- **bgp bestpath med missing-as-worst** configures the switch to treat a missing MED as having the highest (least preferred) value (disabled by default). This command overrides the **missing-as-worst** setting of the **bgp bestpath med confed** command.
- **bgp bestpath tie-break cluster-list-length** configures the switch to prefer the multipath route with the shortest CLUSTER_LIST length in case of a tie in step 10 of the selection process (disabled by default).
- **bgp bestpath tie-break router-id** configures the switch to prefer the multipath route with the lowest ROUTER_ID in case of a tie in step 10 (disabled by default).

29.2.5.1 Displaying Reasons for Best-path Selection

To see the reasons why certain routes were excluded by the best-path selection process, use the **detail** option of the **show ip bgp** command. Enter the prefix to which BGP has selected a best path, and the output will display all learned paths. Paths which were not selected as best will display the reason they were not selected after the label **not best**.

The reason will be listed as one of the following:

- Path weight
- Local preference
- AS path length
- Origin
- Path MED
- eBGP path preferred
- IGP cost
- AS path details
- ECMP-Fast configured
- Router ID
- Originator ID
- Router ID tie-break configured
- Cluster list length
- Cluster list length tie-break configured
- Peer IP address
- Path ID
- Redistributed route exists
- Unknown
- Another route from the same AS is a better BGP route
- Peer not ready
- Unusuable

Example

- This command displays the reasons why three routes to 172.16.0.0/24 were rejected by the best-path algorithm. The reason for rejection is preceded by the label **Not best:**

```
switch>#show ip bgp 172.16.0.0/24 detail
BGP routing table information for VRF default
Router identifier 192.168.100.18, local AS number 64524
Route status: [a.b.c.d] - Route is  queued for advertisement to peer.
BGP routing table entry for 204.1.47.220/30
 Paths: 4 available
  64512 64550 65100
    192.168.14.2 from 192.168.14.2 (192.168.100.21)
     Origin IGP, metric 0, localpref 100, weight 0, received 19:15:29 ago, valid,
external, ECMP head, ECMP, best, ECMP contributor
      Rx SAFI: Unicast
  64512 64550 65100
    192.168.24.2 from 192.168.24.2 (192.168.100.22)
     Origin IGP, metric 0, localpref 100, weight 0, received 19:15:29 ago, valid,
external, ECMP, ECMP contributor
      Rx SAFI: Unicast
      Not best: ECMP-Fast configured
  64512 64550 65100
    192.168.34.2 from 192.168.34.2 (192.168.100.23)
     Origin IGP, metric 0, localpref 100, weight 0, received 19:15:29 ago, valid,
external, ECMP, ECMP contributor
      Rx SAFI: Unicast
      Not best: Redistributed route exists
  64512 64550 65100
    192.168.44.2 from 192.168.44.2 (192.168.100.24)
     Origin IGP, metric 0, localpref 100, weight 0, received 19:15:29 ago, valid,
external, ECMP, ECMP contributor
      Rx SAFI: Unicast
      Not best: eBGP path preferred
Not advertised to any peer
switch>
```

29.2.6 BGP Confederations

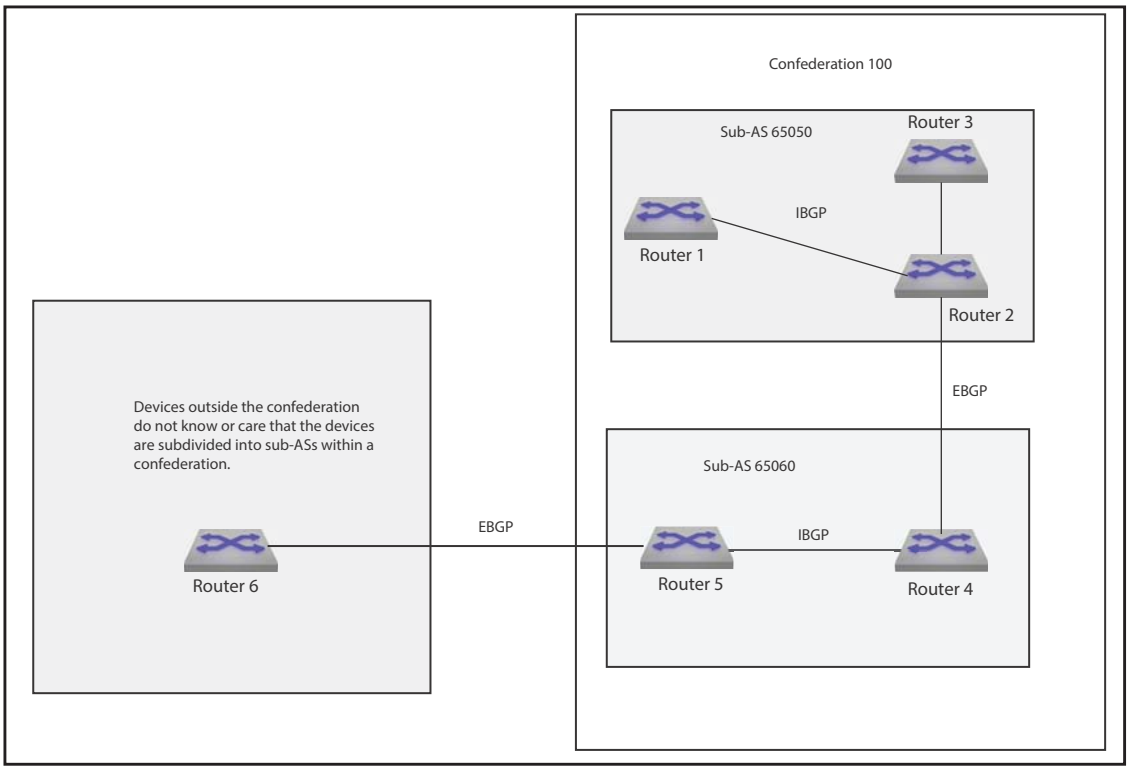
BGP confederations allow you to break an autonomous system (AS) into multiple sub-ASs, and then to group the sub-ASs as a confederation.

The sub-ASs exchange iBGP routing information (next-hop, local-preference and MED), but communicate via eBGP.

To configure a BGP confederation, perform these configuration tasks on each BGP device within the confederation:

- Configure the local AS number. The local AS number indicates membership in a sub-AS. All BGP devices with the same local AS number are members of the same sub-AS. BGP devices use the local AS number when communicating with other BGP4 devices in the confederation.
- Configure the confederation ID. The confederation ID is the AS number by which BGP devices outside the confederation recognize the confederation. A BGP device outside the confederation is not aware of, and does not care that BGP devices are in multiple sub-ASs. A BGP device uses the confederation ID to communicate with devices outside the confederation. The confederation ID must differ from the sub-AS numbers.
- Configure the list of the sub-AS numbers that are members of the confederation. Devices in a sub-AS exchange information via iBGP, while devices in different sub-ASs use eBGP.

Figure 29-1: BGP Confederation Example



Example

- The **router bgp** command enables BGP and configures the router in sub-autonomous system 100. The **bgp confederation identifier** command specifies confederation 65050 belongs to autonomous system 100.

  The neighbors from other autonomous systems within the confederation are treated as special eBGP peers when using the **bgp confederation peers** command.

```
switch(config)#router bgp 100
switch(config-router-bgp)#bgp confederation identifier 65050
switch(config-router-bgp)#bgp confederation peers 65060
switch(config-router-bgp)#
```

- The Arista EOS will group the maximum ranges together. In this example, peers 65032 and 65036 are not included in BGP confederation 65050.

```
switch(config)#router bgp 100
switch(config-router-bgp)#bgp confederation identifier 65050
switch(config-router-bgp)#bgp confederation peers 65060
switch(config-router-bgp)#no bgp confederation peers 65032, 65036
switch(config-router-bgp)#
```

29.2.7 BGP Operational Commands

29.2.7.1 Shutdown

The **shutdown (BGP)** command disables BGP operations without disrupting the BGP configuration. The **no router bgp** command disables BGP and removes the BGP configuration.

The **no shutdown** command resumes BGP activity.

Example

- This command disables BGP activity on the switch.

```
switch(config-router-bgp)#shutdown
switch(config-router-bgp)#
```

- This command resumes BGP activity on the switch.

```
switch(config-router-bgp)#no shutdown
switch(config-router-bgp)#
```

29.2.7.2 Clearing the Routing Table and Resetting BGP Sessions

Changes to a route map do not take effect until the BGP process is forced to recognize the changes. The **clear ip bgp** command clears all BGP learned routes from the routing table, reads routes from designated peers, and sends routes required by those peers. Routes that are read or sent are processed through any modified route map or AS-path access list.

The **clear ip bgp *** command clears the BGP sessions with all BGP peers. To reset the session with a specific peer, enter the peer's IP address in place of the asterisk.

Example

- This command removes all BGP learned routes from the routing table.

```
switch#clear ip bgp
switch#
```

29.3 BGP Examples

This section describes the commands required to configure an iBGP and an eBGP topology.

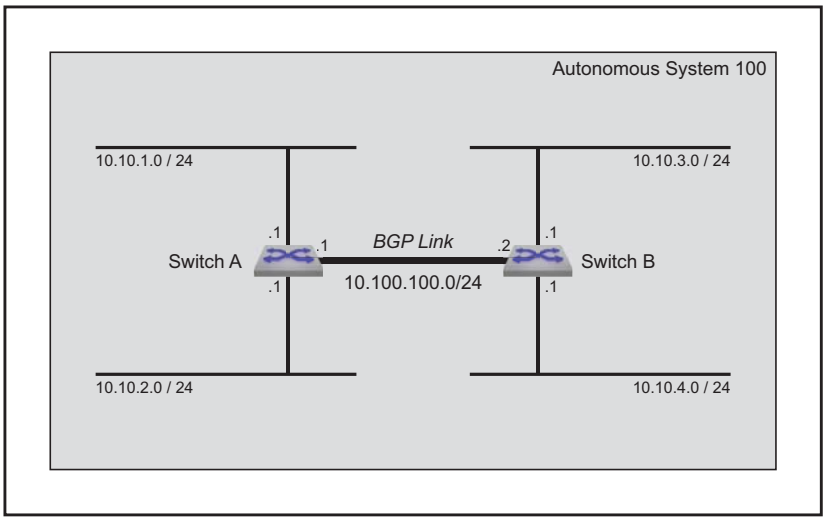
29.3.1 Example 1

Example 1 features an internal BGP link that connects peers in AS 100.

29.3.1.1 Diagram

Figure 29-2 displays BGP Example 1. The BGP link establishes iBGP neighbors in AS 100. Each switch advertises two subnets. In UPDATE packets sent by Switch A, the LOCAL_PREF field is 150. In UPDATE packets sent by Switch B, the LOCAL_PREF field is 75.

Figure 29-2: BGP Example 1



29.3.1.2 Code

This code configures the Example 1 BGP instance on both switches.

Step 1 Configure the neighbor addresses.

a Specify the neighbor to Switch A.

```
switchA(config)#router bgp 100
switchA(config-router-bgp)#neighbor 10.100.100.2 remote-as 100
```

b Specify the neighbor to Switch B.

```
switchB(config)#router bgp 100
switchB(config-router-bgp)#neighbor 10.100.100.1 remote-as 100
```

Step 2 Configure the routes to be advertised.

a Advertise Switch A's routes.

```
switchA(config-router-bgp)#network 10.10.1.0/24
switchA(config-router-bgp)#network 10.10.2.0/24
```

b Advertise Switch B's routes.

```
switchB(config-router-bgp)#network 10.10.3.0/24
switchB(config-router-bgp)#network 10.10.4.0/24
```

Step 3 Configure the LOCAL_PREF.

```
switchA(config-router-bgp)#neighbor 10.100.100.2 export-localpref 150
switchB(config-router-bgp)#neighbor 10.100.100.1 export-localpref 75
```

Step 4 Modify the hold time and keepalive interval.

```
switchA(config-router-bgp)#timer bgp 30 90
switchB(config-router-bgp)#timer bgp 30 90
```

29.3.2 Example 2

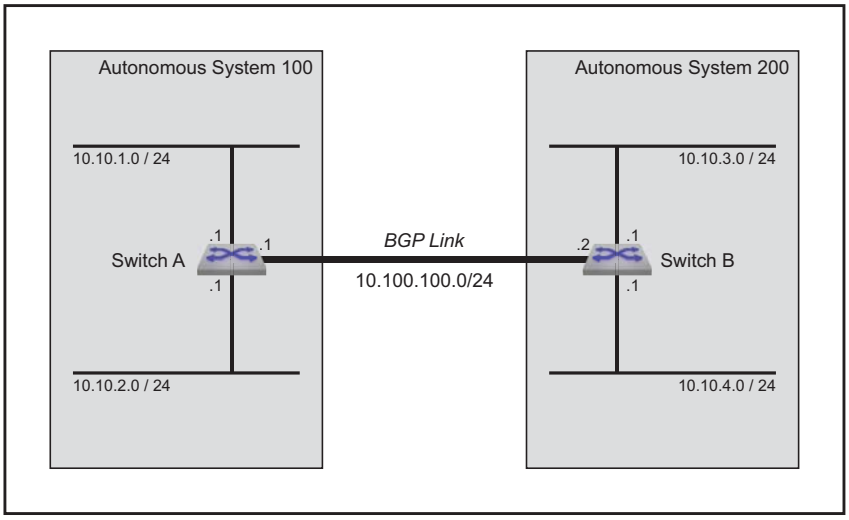
Example 2 creates an external BGP link that connects routers in AS 100 and AS 200.

29.3.2.1 Diagram

Figure 29-3 displays BGP Example 2. The BGP link connects a switch in AS 100 to a switch in AS 200. Each switch advertises two subnets.

Switch A assigns a local preference of 150 to networks advertised by Switch B. Switch B assigns a local preference of 75 to networks advertised by Switch A.

Figure 29-3: BGP Example 2



29.3.2.2 Code

This code configures the Example 2 BGP instance on both switches.

Step 1 Configure the neighbor addresses.

a Specify the neighbor to Switch A.

```
switchA(config)#router bgp 100
switchA(config-router-bgp)#neighbor 10.100.100.2 remote-as 200
```

b Specify the neighbor to Switch B.

```
switchB(config)#router bgp 200
switchB(config-router-bgp)#neighbor 10.100.100.1 remote-as 100
```

Step 2 Configure the routes to be advertised.

a Advertise Switch A's routes.

```
switchA(config-router-bgp)#network 10.10.1.0/24
switchA(config-router-bgp)#network 10.10.2.0/24
```

b Advertise Switch B's routes.

```
switchB(config-router-bgp)#network 10.10.3.0/24
switchB(config-router-bgp)#network 10.10.4.0/24
```

Step 3 Assign local preference values to routes received from their respective peers.

```
switchA(config-router-bgp)#neighbor 10.100.100.2 import-localpref 150
switchB(config-router-bgp)#neighbor 10.100.100.2 import-localpref 75
```

Step 4 Modify the hold timer and keepalive interval.

```
switchA(config-router-bgp)#timer bgp 30 90
switchB(config-router-bgp)#timer bgp 30 90
```

29.4 BGP Commands

Global Configuration Commands

- router bgp
- ip as-path access-list
- ip as-path regex-mode
- ip community-list expanded
- ip community-list standard
- ip extcommunity-list expanded
- ip extcommunity-list standard

Router-BGP Configuration Mode (Includes Address-Family Mode)

- address-family
- aggregate-address
- bgp advertise-inactive
- bgp always-compare-med
- bgp bestpath as-path ignore
- bgp bestpath as-path multipath-relax
- bgp bestpath ecmp-fast
- bgp bestpath med confed
- bgp bestpath med missing-as-worst
- bgp bestpath tie-break cluster-list-length
- bgp bestpath tie-break router-id
- bgp client-to-client reflection
- bgp cluster-id
- bgp confederation identifier
- bgp confederation peers
- bgp default
- bgp enforce-first-as
- bgp listen limit
- bgp listen range
- bgp log-neighbor-changes
- bgp redistribute-internal (BGP)
- distance bgp
- graceful-restart stalepath-time
- graceful-restart-helper
- maximum paths (BGP)
- no neighbor
- neighbor activate
- neighbor allowas-in
- neighbor auto-local-addr
- neighbor default-originate
- neighbor description
- neighbor ebgp-multihop
- neighbor enforce-first-as
- neighbor export-localpref
- neighbor graceful-restart-helper
- neighbor import-localpref
- neighbor local-as
- neighbor local-v4-addr

- neighbor local-v6-addr
- neighbor maximum-routes
- neighbor next-hop-peer
- neighbor next-hop-self
- neighbor out-delay
- neighbor password
- neighbor peer-group (create)
- neighbor peer-group (neighbor assignment)
- neighbor remote-as
- neighbor remove-private-as
- neighbor route-map (BGP)
- neighbor route-reflector-client
- neighbor send-community
- neighbor shutdown
- neighbor soft-reconfiguration
- neighbor timers
- neighbor transport connection-mode
- neighbor update-source
- neighbor weight
- network (BGP)
- redistribute (BGP)
- router-id (BGP)
- shutdown (BGP)
- timers bgp
- vrf

Clear Commands – Privileged EXEC Mode

- clear ip bgp
- clear ip bgp neighbor *
- clear ipv6 bgp
- clear ipv6 bgp neighbor *

Display Commands – EXEC Mode

- show bgp instance
- show ip as-path access-list
- show ip bgp
- show ip bgp community
- show ip bgp neighbors
- show ip bgp neighbors (route type)
- show ip bgp neighbors (route-type) community
- show ip bgp neighbors regexp
- show ip bgp paths
- show ip bgp peer-group
- show ip bgp regexp
- show ip bgp summary
- show ip community-list
- show ip extcommunity-list
- show ipv6 bgp
- show ipv6 bgp community
- show ipv6 bgp neighbors

- show ipv6 bgp neighbors (route type)
- show ipv6 bgp neighbors (route type) community
- show ipv6 bgp neighbors regexp
- show ipv6 bgp regexp
- show ipv6 bgp summary

address-family

The **address-family** command places the switch in address-family configuration mode to configure the address family setting of addresses configured as BGP neighbors. Address-family configuration mode is not a group change mode; ***running-config*** is changed immediately after commands are executed. The **exit** command does not affect the configuration.

The switch supports these address families:

- ipv4-unicast
- ipv6-unicast

Running-config displays address family commands in sub-blocks of the BGP configuration. The following commands are available in address family configuration mode:

- **neighbor activate** configures the address as active in the configuration mode address family.
- **no neighbor activate** configures the address as not active in the configuration mode address family.
- **neighbor default-originate** advertises a default route to the specified BGP neighbor.
- **neighbor route-map (BGP)** applies a route map to the specified BGP route.
- **network (BGP)** specifies a network for advertisement through UPDATE packets to BGP peers.

The **no address-family** and **default address-family** commands delete the specified address-family from ***running-config*** by removing all commands previously configured in the corresponding address-family mode.

The **exit** command returns the switch to router-BGP configuration mode.

Command Mode

Router-BGP Configuration

Command Syntax

```
bgp ADDRESS_TYPE
no bgp ADDRESS_TYPE
default bgp ADDRESS_TYPE
```

Parameters

- ***ADDRESS_FAMILY*** Address family affected by subsequent commands. Options include:
  - **ipv4** IPv4 unicast
  - **ipv6** IPv6 unicast

Example

- These commands enter address family mode for IPv6-unicast, insert a command, then exit the mode:

```
switch(config)#router bgp 1
switch(config-router-bgp)#address-family ipv6
switch(config-router-bgp-af)#neighbor 172.10.1.1 activate
switch(config-router-bgp-af)#exit
switch(config-router-bgp)#
```

aggregate-address

The **aggregate-address** command creates an aggregate route in the Border Gateway Protocol (BGP) database. Aggregate routes combine the characteristics of multiple routes into a single route that the switch advertises. Aggregation can reduce the amount of information that a BGP speaker is required to store and transmit when advertising routes to other BGP speakers. Aggregate routes are advertised only after they are redistributed.

The advertised address of the aggregate is entered as an IP subnet; any routes configured on the switch that lie within that subnet then become contributors to the aggregate. Note that on Arista switches the BGP aggregate route will become active if there are any available contributor routes on the switch, regardless of the originating protocol. This includes routes configured statically.

Important! Aggregate routes are redistributed automatically, and their redistribution cannot be disabled.

Command options affect the attributes associated with the aggregated route, the advertisement of the contributor routes that comprise the aggregate, and which contributor routes are included.

Command options affect the following aggregate routing attributes:

- AS_PATH attribute inclusion: the ***as-set*** option controls the aggregate route's AS_PATH and ATOMIC_AGGREGATE attribute contents. AS_PATH identifies the autonomous systems through which UPDATE message routing information passes. ATOMIC_AGGREGATE indicates that the route is an aggregate or summary of more specific routes.

  When the command includes ***as-set***, the aggregate route's AS_SET attribute contains the AS numbers of contributor routes. This can help BGP neighbors to prevent loops by rejecting aggregate routes that include their AS number in the AS_SET.

  When the command does not include ***as-set***, the aggregate route's ATOMIC_AGGREGATE attribute is set and the AS_PATH attribute does not include AS numbers of contributing routes.
- Attribute assignment: The ***attribute-map*** option assigns attributes contained in set commands in a specified route map's lowest sequence with any set command to the aggregated route, overriding the automatic determination of the aggregate route's attributes by the switch.
- Route suppression: The ***summary-only*** option suppresses the advertisement of the contributor routes that comprise the aggregate.
- Contributor filtering: The ***match-map*** option uses a route map to filter out contributor routes that would otherwise be included in the aggregate.

The **no aggregate-address** and **default aggregate-address** commands remove the corresponding **aggregate-address** command from ***running-config***.

Command Mode

Router-BGP Configuration

Command Syntax

```
aggregate-address AGGREGATE_NET [AS_SET][SUMMARY][ATTRIBUTE_MAP][MATCH_MAP]
no aggregate-address AGGREGATE_NET
default aggregate-address AGGREGATE_NET
```

Parameters

- ***AGGREGATE_NET*** aggregate route IP address. Options include:
  - *netv4_addr* IPv4 subnet address (CIDR or address-mask notation).
  - *netv6_addr* IPv6 subnet address (CIDR notation).

- ***AS_SET*** controls AS_PATH attribute values associated with aggregate route. Options include:
  - <no parameter> ATOMIC_AGGREGATE attribute is set. Route contains no AS_PATH data.
  - **as-set** route includes AS_PATH information from contributor routes as AS_SET attributes.
- ***SUMMARY*** controls advertisement of contributor routes. Options include:
  - <no parameter> contributor and aggregate routes are advertised.
  - **summary-only** contributor routes are not advertised.
- ***ATTRIBUTE_MAP*** controls attribute assignments to the aggregate route. Options include:
  - <no parameter> attribute values are not assigned to route.
  - **attribute-map** *map_name* assigns attribute values in set commands of the map's permit clauses. Deny clauses and match commands in permit clauses are ignored.
- ***MATCH_MAP*** filters contributors to the aggregate route. Options include:
  - <no parameter> no contributors are filtered.
  - **match-map** *map_name* filters contributor routes using the named match-map.

Examples

- These commands create an aggregate route (10.16.48.0/20) from the contributor routes 10.16.48.0/23, 10.16.50.0/23,10.16.52.0/23, and 10.16.54.0/23. The aggregate route includes the AS_PATH information from the contributor routes.

```
switch(config)#router bgp 1
switch(config-router-bgp)#aggregate-address 10.16.48.0/20 as-set
switch(config-router-bgp)#exit
switch(config)#
```

- These commands create an aggregate route and use a route map to add a local-preference attribute to the route.

```
switch(config)#route-map map1 permit 10
switch(config-route-map-map1)#set community 45
switch(config-route-map-map1)#exit
switch(config)#router bgp 1
switch(config-router-bgp)#aggregate-address 10.16.48.0/20 attribute-map map1
switch(config-router-bgp)#exit
switch(config)#
```

- These commands create an aggregate route and use a route map to allow only those contributors which match a specified prefix list to be included in the aggregate route.

```
switch(config)#route-map matchmap permit 10
switch(config-route-map-matchmap)#match ip address prefix-list agglist
switch(config-route-map-matchmap)#exit
switch(config)#router bgp 1
switch(config-router-bgp)#aggregate-address 1.1.0.0/16 match-map matchmap
```

bgp advertise-inactive

By default, BGP will advertise only those routes that are active in the switch's RIB. This can contribute to dropped traffic. If a preferred route is available through another protocol (like OSPF), the BGP route will become inactive and not be advertised; if the preferred route is lost, there is no available route to the affected peers. Advertising inactive BGP routes minimizes traffic loss by providing alternative routes.

The **bgp advertise-inactive** command configures BGP to advertise inactive routes to BGP neighbors. Inactive route advertisement is configured globally, but the global setting can be overridden on a per-VRF basis.

The **no bgp advertise-inactive** and **default bgp advertise-inactive** commands restore the default BGP behavior (advertising only active routes) by removing the corresponding **bgp advertise-inactive** command from ***running-config***.

Command Mode

Router-BGP Configuration

Command Syntax

```
bgp advertise-inactive
no bgp advertise-inactive
default bgp advertise-inactive
```

Example

- These commands configure BGP to advertise inactive routes.

```
switch(config)#router bgp 64500
switch(config-router-bgp)#bgp advertise-inactive
switch(config-router-bgp)#
```

bgp always-compare-med

The **bgp always-compare-med** command configures the switch to always consider multi-exit discriminator (MED) values (also known as “metric”) in best-path selection. By default, this function is disabled, and MED values are compared only if two paths have the same neighbor AS.

When there are two or more links between autonomous systems, MED values may be set by a router in the originating AS to give preferences to certain routes. In comparing MED values, the lower value is preferred.

The **no bgp always-compare-med** and **default bgp always-compare-med** commands restore the default behavior of comparing MED values only on paths with the same neighbor AS.

Command Mode

Router-BGP Configuration

Command Syntax

```
bgp always-compare-med
no bgp always-compare-med
default bgp always-compare-med
```

Related Commands

- **bgp bestpath as-path ignore**
- **bgp bestpath as-path multipath-relax**
- **bgp bestpath ecmp-fast**
- **bgp bestpath med confed**
- **bgp bestpath med missing-as-worst**
- **bgp bestpath tie-break cluster-list-length**
- **bgp bestpath tie-break router-id**

Example

- These commands configure BGP to always consider MED values in best-path comparisons.

```
switch(config)#router bgp 64500
switch(config-router-bgp)#bgp always-compare-med
switch(config-router-bgp)#
```

bgp bestpath as-path ignore

The **bgp bestpath as-path ignore** command configures BGP to ignore the length of the autonomous system (AS) path when comparing routes. This behavior is disabled by default. Normally, the switch compares AS paths as the third step in the best-path selection process (see Best-Path Selection), preferring the route with the shorter AS path.

The **no bgp bestpath as-path ignore** and **default bgp bestpath as-path ignore** commands restore the default behavior of considering AS path length in route comparisons.

Command Mode

Router-BGP Configuration

Command Syntax

```
bgp bestpath as-path ignore
no bgp bestpath as-path ignore
default bgp bestpath as-path ignore
```

Related Commands

- **bgp always-compare-med**
- **bgp bestpath as-path multipath-relax**
- **bgp bestpath ecmp-fast**
- **bgp bestpath med confed**
- **bgp bestpath med missing-as-worst**
- **bgp bestpath tie-break cluster-list-length**

  bgp bestpath tie-break router-id

Example

- These commands configure BGP to ignore AS path lengths when comparing routes.

```
switch(config)#router bgp 64500
switch(config-router-bgp)#bgp bestpath as-path ignore
switch(config-router-bgp)#
```

bgp bestpath as-path multipath-relax

The **bgp bestpath as-path multipath-relax** command allows multiple eBGP routes to a destination to be considered equal in ECMP if their AS paths are the same length despite having different autonomous systems in those paths. The **no bgp bestpath as-path multipath-relax** command configures best-path selection to consider two paths *unequal* if their AS path contents are different, and prefers the first path received.

Multipath-relax is enabled by default. The **bgp bestpath as-path multipath-relax** and **default bgp bestpath as-path multipath-relax** commands restore the default behavior by removing the corresponding **no bgp bestpath as-path multipath-relax** command from ***running-config***.

For BGP to support equal cost multipath (ECMP) routing, the **maximum paths (BGP)** command must be issued in router-BGP configuration mode.

Command Mode

Router-BGP Configuration

Command Syntax

```
bgp bestpath as-path multipath-relax
no bgp bestpath as-path multipath-relax
default bgp bestpath as-path multipath-relax
```

Related Commands

- **bgp always-compare-med**
- **bgp bestpath as-path ignore**
- **bgp bestpath ecmp-fast**
- **bgp bestpath med confed**
- **bgp bestpath med missing-as-worst**
- **bgp bestpath tie-break cluster-list-length**

  bgp bestpath tie-break router-id

Example

- These commands configure BGP best-path selection to consider routes unequal if the contents of their AS paths differ.

```
switch(config)#router bgp 64500
switch(config-router-bgp)#no bgp bestpath as-path multipath-relax
switch(config-router-bgp)#
```

bgp bestpath ecmp-fast

By default, within an ECMP group the BGP best-path selection process prefers the active path (the first path received by the switch) unless a relevant tie-breaker is enabled. The **no bgp bestpath ecmp-fast** command causes the best-path selection process to ignore order of arrival and continue evaluating paths on other criteria.

The **bgp bestpath ecmp-fast** and **default bgp bestpath ecmp-fast** commands restore the default behavior by removing the corresponding **no bgp bestpath ecmp-fast** command from ***running-config***.

Command Mode

Router-BGP Configuration

Command Syntax

```
bgp bestpath ecmp-fast
no bgp bestpath ecmp-fast
default bgp bestpath ecmp-fast
```

Related Commands

- **bgp always-compare-med**
- **bgp bestpath as-path ignore**
- **bgp bestpath as-path multipath-relax**
- **bgp bestpath med confed**
- **bgp bestpath med missing-as-worst**
- **bgp bestpath tie-break cluster-list-length**

  bgp bestpath tie-break router-id

Example

- These commands configure BGP to ignore order of arrival in best-path comparisons of paths within an ECMP group.

```
switch(config)#router bgp 64500
switch(config-router-bgp)#no bgp bestpath ecmp-fast
switch(config-router-bgp)#
```

bgp bestpath med confed

By default, paths originating within the same confederation as the switch and received from confederation peers do not have their multi-exit discriminator (MED) values compared as part of the best-path selection process. The **bgp bestpath med confed** command causes comparison of MED values in such routes. To ensure that MED values are considered in the best-path selection process for all routes received, use the **bgp always-compare-med** command.

The **no bgp bestpath med confed** and **default bgp bestpath med confed** commands restore the default behavior by removing the corresponding **bgp bestpath ecmp-fast** command from ***running-config***.

Command Mode

Router-BGP Configuration

Command Syntax

```
bgp bestpath med confed [missing-as-worst]
no bgp bestpath med confed [missing-as-worst]
default bgp bestpath med confed [missing-as-worst]
```

Related Commands

- **bgp always-compare-med**
- **bgp bestpath as-path ignore**
- **bgp bestpath as-path multipath-relax**
- **bgp bestpath ecmp-fast**
- **bgp bestpath med missing-as-worst**
- **bgp bestpath tie-break cluster-list-length**
  bgp bestpath tie-break router-id

Parameters

missing-as-worst By default, best-path selection considers a missing MED value to be 0, so paths with missing MED values will be preferred. This option reverses the behavior in comparisons of routes originating within the same confederation as the switch, treating a missing MED as having the highest (least preferred) value.

Note The **bgp bestpath med missing-as-worst** command controls how best-path selection treats missing MED values for *all* routes received, and, if configured, overrides the **missing-as-worst** option of this command.

Example

- These commands configure the BGP best-path selection process to consider MED values in comparisons between routes originating within the same confederation as the switch.

```
switch(config)#router bgp 64500
switch(config-router-bgp)#bgp bestpath med confed
switch(config-router-bgp)#
```

bgp bestpath med missing-as-worst

By default, BGP best-path selection considers a missing MED value to be 0, so paths with missing MED values will be preferred. The **bgp bestpath med missing-as-worst** command reverses the behavior, treating a missing MED as having the highest (least preferred) value.

The **no bgp bestpath med missing-as-worst** and **default bgp bestpath med missing-as-worst** commands restore the default behavior (giving preference to missing MED values) by removing the corresponding **bgp bestpath med missing-as-worst** command from ***running-config***.

Note This command overrides the **missing-as-worst** setting of the **bgp bestpath med confed** command.

Command Mode

Router-BGP Configuration

Command Syntax

```
bgp bestpath med missing-as-worst
no bgp bestpath med missing-as-worst
default bgp bestpath med missing-as-worst
```

Related Commands

- **bgp always-compare-med**
- **bgp bestpath as-path ignore**
- **bgp bestpath as-path multipath-relax**
- **bgp bestpath ecmp-fast**
- **bgp bestpath med confed**
- **bgp bestpath tie-break cluster-list-length**

  bgp bestpath tie-break router-id

Example

- These commands configure the BGP best-path selection process to consider a missing MED value to be considered highest (least preferred) in MED comparisons for all routes received.

```
switch(config)#router bgp 64500
switch(config-router-bgp)#bgp bestpath med missing-as-worst
switch(config-router-bgp)#
```

bgp bestpath tie-break cluster-list-length

The **bgp bestpath tie-break cluster-list-length** command causes the best-path selection process to prefer the multipath route with the shortest CLUSTER_LIST length in case of a tie in step 10. The cluster list length is assumed to be 0 if the route doesn't carry a CLUSTER_LIST attribute.

The **no bgp bestpath tie-break cluster-list-length** and **default bgp bestpath tie-break cluster-list-length** commands restore the default behavior by removing the associated **bgp bestpath tie-break cluster-list-length** command from ***running-config***.

Command Mode

Router-BGP Configuration

Command Syntax

```
bgp bestpath tie-break cluster-list-length
no bgp bestpath tie-break cluster-list-length
default bgp bestpath tie-break cluster-list-length
```

Related Commands

- **bgp always-compare-med**
- **bgp bestpath as-path ignore**
- **bgp bestpath as-path multipath-relax**
- **bgp bestpath ecmp-fast**
- **bgp bestpath med confed**
- **bgp bestpath med missing-as-worst**

  bgp bestpath tie-break router-id

Example

- These commands configure the BGP selection process to prefer the multipath multipath route with the shortest CLUSTER_LIST length in case of a tie.

```
switch(config)#router bgp 64500
switch(config-router-bgp)#bgp bestpath tie-break cluster-list-length
switch(config-router-bgp)#
```

bgp bestpath tie-break router-id

The **bgp bestpath tie-break router-id** command causes the best-path selection process to prefer the multipath route with the lowest ROUTER_ID in case of a tie in step 10. If the route is a reflected route (i.e., if it contains route reflector attributes), the process will use the ORIGINATOR_ID as the ROUTER_ID for comparison. This behavior is disabled by default.

The **no bgp bestpath tie-break router-id** and **default bgp bestpath tie-break router-id** commands restore the default behavior by removing the associated **bgp bestpath tie-break router-id** command from ***running-config***.

Command Mode

Router-BGP Configuration

Command Syntax

```
bgp bestpath tie-break router-id
no bgp bestpath tie-break router-id
default bgp bestpath tie-break router-id
```

Related Commands

- **bgp always-compare-med**
- **bgp bestpath as-path ignore**
- **bgp bestpath as-path multipath-relax**
- **bgp bestpath ecmp-fast**
- **bgp bestpath med confed**
- **bgp bestpath med missing-as-worst**
- **bgp bestpath tie-break cluster-list-length**

Example

- These commands configure the best-path selection process to prefer the multipath route with the lowest ROUTER_ID in case of a tie.

```
switch(config)#router bgp 64500
switch(config-router-bgp)#bgp bestpath tie-break router-id
   switch(config-router-bgp)#
```

bgp client-to-client reflection

By default, routes received from a route reflector client and selected as best routes are propagated to all BGP peers, including other route reflector clients. If the clients are fully meshed, however, routes received from a client do not need to be mirrored to other clients. In this case, client-to-client reflection should be disabled.

The **no bgp client-to-client reflection** command disables client-to-client reflection.

The **bgp client-to-client reflection** and **default bgp client-to-client reflection** commands restore the default behavior by removing the **no bgp client-to-client reflection** command from ***running-config*.**

Command Mode

Router-BGP Configuration

Command Syntax

```
bgp client-to-client reflection
no bgp client-to-client reflection
default bgp client-to-client reflection
```

Example

- This command disables client-to-client reflection on the switch.

```
switch(config)#router bgp 1
switch(config-router-bgp)#no bgp client-to-client reflection
switch(config-router-bgp)#
```

bgp cluster-id

When using route reflectors, an AS is divided into clusters. A cluster consists of one or more route reflectors and a group of clients to which they re-advertise route information, and for redundancy a single cluster may contain multiple route reflectors. Each route reflector has a cluster ID. If the cluster has only one route reflector the cluster ID is its router ID, but if a cluster has multiple route reflectors a 4-byte cluster ID must be assigned to all route reflectors in the cluster. All must be configured with the same cluster ID to allow them to identify updates from the cluster's other route reflectors.

The **bgp cluster-id** command configures the cluster ID in a cluster with multiple route reflectors.

The **no bgp cluster-id** and **default bgp cluster-id** commands remove the cluster ID by removing the corresponding **bgp cluster-id** command from ***running-config***. Do not remove the cluster ID if there are multiple route reflectors in the cluster.

Command Mode

Router-BGP Configuration

Command Syntax

```
bgp cluster-id ID_NUM
no bgp cluster-id
default bgp cluster-id
```

Parameters

- ***ID_NUM*** cluster ID shared by all route reflectors in the cluster (32-bit dotted-decimal notation). Options include:
  - ***0.0.0.1*** to ***255.255.255.255*** valid cluster ID number.
  - ***0.0.0.0*** removes the cluster-ID from the switch. Equivalent to **no bgp cluster-id** command.

Example

- This command sets the cluster ID for the switch to 172.22.30.101.

```
switch(config)#router bgp 1
switch(config-router-bgp)#bgp cluster-id 172.22.30.101
switch(config-router-bgp)#
```

bgp confederation identifier

The **bgp confederation identifier** command configures the confederation identifier. Confederation can reduce the number of iBGP connections in a large AS domain. The AS domain is divided into several smaller sub-ASs, and each sub-AS remains fully connected. Devices in a sub-AS exchange information via iBGP, while devices in different sub-ASs use eBGP

The **no bgp confederation identifier** and **default bgp confederation identifier** commands remove the **bgp confederation identifier** command from ***running-config***.

Command Mode

Router-BGP Configuration

Command Syntax

```
bgp confederation identifier as_number
no bgp confederation identifier
default bgp confederation identifier
```

Parameters

- *as_number* the ID of BGP AS confederation. Value ranges from 1 to 4294967295.

Example

- This command sets the BGP confederation identifier to 9.

```
switch(config)#router bgp 1
switch(config-router-bgp)#bgp confederation identifier 9
switch(config-router-bgp)#
```

bgp confederation peers

The **bgp confederation peers** command configures a confederation consisting of sub-ASs.

Before this command is executed, the confederation ID should be configured by the **bgp confederation identifier** command. Otherwise this configuration is invalid. The configured ASs in this command are inside the confederation and each AS uses a fully meshed network. The confederation appears as a single AS to the devices outside it.

The **no bgp confederation peers** and **default bgp confederation peers** commands delete the specified sub-AS from the confederation by removing the corresponding **bgp confederation peers** command from ***running-config***.

Command Mode

Router-BGP Configuration

Command Syntax

```
bgp confederation peers as_range
no bgp confederation peers as_range
default bgp confederation peers as_range
```

Parameters

- *as_range* the Sub-AS number.

  as_range formats include number (from 1 to 4294967295), number range, or comma-delimited list of numbers and ranges.

Example

- This command configures the confederation that contains AS 1000 and 1002.

```
switch(config)#router bgp 1
switch(config-router-bgp)#bgp confederation peers 1000 1002
switch(config-router-bgp)#
```

bgp default

The **bgp default** command configures the default address family activation level of all addresses configured as BGP neighbors. The switch sends the following announcements to addresses active in an address family:

- IPv4 address family: IPv4 capability and all network advertisements with IPv4 prefixes.
- IPv6 address family: IPv6 capability and all network advertisements with IPv6 prefixes.

The following commands configure default address family activation levels for addresses configured as BGP neighbors:

- **bgp default ipv4-unicast** all addresses are IPv4 address family active.
- **no bgp default ipv4-unicast** all addresses are not IPv4 address family active.
- **bgp default ipv6-unicast** all addresses are IPv6 address family active
- **no bgp default ipv6-unicast** all addresses are not IPv6 address family active.
- **bgp default ipv4-unicast transport ipv6** all BGP neighbor addresses are IPv4 address family active and IPv6 neighbors can receive IPv4 NLRIs.

Note If it is necessary to exchange IPv4 NLRIs over an IPv6 connection, the IPv4 address family must be activated on the IPv6 neighbor. To do this for all IPv6 neighbors, use the command **bgp default ipv4-unicast transport ipv6**. For an individual neighbor, use the **neighbor activate** command for the IPv6 neighbor in the IPv4 address-family configuration mode as described below.

The activation state of an individual BGP neighbor address is configured by the **neighbor activate** commands. The **neighbor activate** command overrides the address's default activation state for the address family configuration mode in which the command is issued:

- **neighbor activate**: the specified address is active.
- **no neighbor activate**: the specified address is not active.

The *default-default address family* activation state defines address family activation level of all addresses configured as BGP neighbors when ***running-config*** does not contain any **bgp default** commands. The default state of the BGP default activation level varies by address family.

- **IPv4 address family** all BGP addresses are IPv4 address family active.
- **IPv6 address family** all BGP addresses are not IPv6 address family active.

The **default bgp default** command restores the default-default activation setting for BGP neighbor addresses in the specified address family:

- **default bgp ipv4-unicast** is equivalent to **bgp ipv4-unicast**
- **default bgp ipv6-unicast** is equivalent to **no bgp ipv6-unicast**

Command Mode

Router-BGP Configuration

Command Syntax

```
bgp default ADDRESS_FAMILY
no bgp default ADDRESS_FAMILY
default bgp default ADDRESS_FAMILY
```

Parameters

- ***ADDRESS_FAMILY*** BGP address family. Options include:

- **ipv4-unicast** IPv4-unicast peering sessions.
- **ipv6-unicast** IPv6-unicast peering sessions.

Example

- These commands configure the switch to configure all BGP neighbor addresses as IPv4 address family active and IPv6 address family active.

```
switch(config)#router bgp 1
switch(config-router-bgp)#bgp default ipv4-unicast
switch(config-router-bgp)#bgp default ipv6-unicast
switch(config-router-bgp)#show active
router bgp 65533
   bgp log-neighbor-changes
   distance bgp 20 200 200
   neighbor 172.23.254.2 remote-as 65533
   neighbor 172.41.254.78 remote-as 65534
   neighbor 2001:0DB8:52a4:fe01::2 remote-as 65533
   neighbor 2001:0DB8:52a4:fe4c::1 out-delay 10
switch(config-router-bgp)#
```

The show active command does not display the **bgp default ipv4-unicast** command because it is the default setting for IPv4 peering sessions.

bgp enforce-first-as

The **bgp enforce-first-as** command causes a forced comparison of the first autonomous system (AS) in the AS path of eBGP routes received from BGP neighbors to the configured remote external peer autonomous system number (ASN). Updates from eBGP peers that do not include that ASN as the first item in the AS path (in the AS_PATH attribute) are discarded.

This behavior is enabled by default upon BGP configuration, and disabled globally by the **no** form of this command. To configure enforce-first-as for an individual neighbor or peer group, use the **neighbor enforce-first-as** command.

Command Mode

Router-BGP Configuration

Command Syntax

```
bgp enforce-first-as
default bgp enforce-first-as
no bgp enforce-first-as
```

Example

- This command configures BGP to enforce the first AS globally.

```
switch(config-router-bgp)#bgp enforce-first-as
switch(config-router-bgp)#
```

bgp listen limit

The **bgp listen limit** command limits the number of dynamic BGP peers allowed on the switch.

The **no bgp listen limit** and **default bgp listen limit** commands restore the default limit of dynamic BGP peers by removing the **bgp listen limit** command from ***running-config***.

Command Mode

Router-BGP Configuration

Command Syntax

```
bgp listen limit maximum
no bgp listen limit
default bgp listen limit
```

Parameters

- *maximum* the maximum number of dynamic BGP peers to be allowed on the switch. Values range from 1 to 1000; default value is 100.

Example

- This command sets the maximum number of dynamic BGP peers allowed on the switch to 200.

```
switch(config)#router bgp 1
switch(config-router-bgp)#bgp listen limit 200
switch(config-router-bgp)#
```

bgp listen range

The **bgp listen range** command identifies a range of IPv4 and IPv6 addresses from which the switch will accept incoming dynamic BGP peering requests, and creates the named dynamic peer group to which those peers will belong. To create a static peer group, use the **neighbor peer-group (create)** command.

Neighbors in a dynamic peer group are configured as a group and cannot be configured individually. Once a dynamic peer group is created with this command, the following ***neighbor*** commands can use the peer group name as a parameter:

- **neighbor ebgp-multihop**
- **neighbor import-localpref**
- **neighbor maximum-routes**
- **neighbor route-map (BGP)**
- **neighbor timers**
- **neighbor update-source**

The **no bgp listen range** and **default bgp listen range** commands remove the dynamic peer group by deleting the corresponding command from ***running-config***. To remove a static peer group, use the **no neighbor** command. All peering relationships with group members are terminated when the dynamic peer group is deleted.

Command Mode

Router-BGP Configuration

Command Syntax

```
bgp listen range NET_ADDRESS peer-group group_name remote-as as_number
no bgp listen range NET_ADDRESS peer-group group_name
default bgp listen range NET_ADDRESS peer-group group_name
```

Parameters

- ***NET_ADDRESS*** IPv4 address range. Entry options include:
  - *IPv4 subnet* IPv4 subnet (CIDR notation).
  - *IPv4_address* **mask** *subnet* IPv4 subnet (dotted decimal notation).
  - *IPv6_prefix* IPv6 subnet (dotted decimal notation).
- *group_name* name of the peer group.
- *as_number* the autonomous system to which the peer group belongs.

Example

- These commands create a dynamic peer group called “brazil” in AS 5 which accepts peering requests from the 192.168.6.0/24 subnet.

```
switch(config)#router bgp 1
switch(config-router-bgp)#bgp listen range 192.168.6.0/24 peer-group brazil
remote-as 5
switch(config-router-bgp)#
```

bgp log-neighbor-changes

The **bgp log-neighbor-changes** command configures the switch to generate a log message when a BGP peer enters or exits the Established state. This is the default behavior.

The **no bgp log-neighbor-changes** command disables the generation of these log messages. The **default bgp log-neighbor-changes** command enables the generation of these log messages.

Command Mode

Router-BGP Configuration

Command Syntax

```
bgp log-neighbor-changes
no bgp log-neighbor-changes
default bgp log-neighbor-changes
```

Example

- These commands configure the switch to generate a message when a BGP peer enters or exits the ***established*** state.

```
switch(config)#router bgp 1
switch(config-router-bgp)#bgp log-neighbor-changes
switch(config-router-bgp)#
```

bgp redistribute-internal (BGP)

The **bgp redistribute-internal** command enables the redistribution of iBGP routes into an interior gateway protocol (IGP).

The **no bgp redistribute-internal** and **default bgp redistribute-internal** commands disable route redistribution from the specified domain by removing the corresponding **bgp redistribute-internal** command from ***running-config***.

Command Mode

Router-BGP Configuration
Router-BGP Configuration-Address-Family

Command Syntax

```
bgp redistribute internal
no bgp redistribute internal
default bgp redistribute internal
```

Example

- This command redistributes internal BGP routes.

```
switch(config)#router bgp 9
switch(config-router-bgp)#bgp redistribute-internal
switch(config-router-bgp)#
```

clear ip bgp

The **clear ip bgp** command removes learned BGP routes from the routing table, reads all routes from designated peers, and sends routes to those peers as required.

- using hard reset terminates current BGP sessions and recreates the local routing information base.
- using soft reset maintains current BGP sessions and reconfigures the local routing information base using stored routes.

Routes that are read or sent are processed through modified route maps or AS-path access lists. The command can also clear the switch's BGP sessions with its peers.

After a route map is modified, the changes do not take effect until the BGP process is forced to recognize the changes. Use the **clear ip bgp** command after changing any of these BGP attributes:

- access lists
- weights
- distribution lists
- timers
- administrative distance
- route maps

Command Mode

Privileged EXEC

Command Syntax

clear ip bgp [*ACTION*] [*RESET_TYPE*] [*DATA_FLOW*] [*VRF_INSTANCE*]

Parameters

- ***ACTION*** the entity upon which the clearing action is taken. Options include:
  - <no parameter> clears the routing table, then reads in routes from designated peers.
  - ***** clears all BGP IPv4 sessions with the switch's peers.
  - *ipv4_addr* resets the IPv4 session with the peer at the specified IPv4 address.
  - *ipv6_addr* resets the IPv4 session with the peer at the specified IPv6 address.
- ***RESET_TYPE*** reconfiguration type. Options include:
  - <no parameter> hard reset.
  - **soft** soft reset.
- ***DATA_FLOW*** restricts hard reset to inbound or outbound routes. Soft reset is bidirectional.
  - <no parameter> inbound and outbound routes are reset.
  - **in** inbound routes are reset.
  - **out** outbound routes are reset.
- ***VRF_INSTANCE*** specifies VRF instances.
  - <no parameter> clears routing table for context-active VRF.
  - **vrf** *vrf_name* clears routing table for the specified VRF.
  - **vrf all** clears routing table for all VRFs.
  - **vrf default** clears routing table for default VRF.

Examples

- This command removes all BGP learned routes from the routing table:

```
switch#clear ip bgp
switch#
```

- This command clears all of the switch's BGP IPv4 peering sessions:

```
switch#clear ip bgp *
switch#
```

clear ip bgp neighbor *

The **clear ip bgp neighbor *** command clears BGP neighbors belonging to the IPv4 transport address family. To clear BGP neighbors in the IPv6 transport address family, use the **clear ipv6 bgp neighbor *** command.

Command Mode

Privileged EXEC

Command Syntax

```
clear ip bgp neighbor * [VRF_INSTANCE]
```

Parameters

- ***VRF_INSTANCE*** specifies VRF instance for which IPv4 transport address family BGP neighbors will be cleared. Options include:
  - <no parameter> clears IPv4 BGP neighbors in the context-active VRF.
  - **vrf** *vrf_name* clears IPv4 BGP neighbors in the specified VRF.
  - **vrf all** clears IPv4 BGP neighbors in the all VRFs.
  - **vrf default** clears IPv4 BGP neighbors in the default VRF.

Examples

- This command clears all IPv4 BGP neighbors in the context-active VRF.

```
switch#clear ip bgp neighbor *
switch#
```

- This command clears all IPv4 BGP neighbors in VRF “purple.”

```
switch#clear ip bgp neighbor * vrf purple
switch#
```

clear ipv6 bgp

The **clear ipv6 bgp** command removes learned BGP routes from the routing table, reads all routes from designated peers, and sends routes to those peers as required.

- using hard reset terminates current BGP sessions and recreates the local routing information base.
- using soft reset maintains current BGP sessions and reconfigures the local routing information base using stored routes.

Routes that are read or sent are processed through modified route maps or AS-path access lists. The command can also clear the switch's BGP sessions with its peers.

After a route map is modified, the changes do not take effect until the BGP process is forced to recognize the changes. Use the **clear ipv6 bgp** command after changing any of these BGP attributes:

- access lists
- weights
- distribution lists
- timers
- administrative distance
- route maps

Command Mode

Privileged EXEC

Command Syntax

clear ipv6 bgp [*ACTION*][*RESET_TYPE*][*DATA_FLOW*][*VRF_INSTANCE*]

Parameters

- ***ACTION*** the entity upon which the clearing action is taken. Options include:
  - <no parameter> clears the routing table, then reads in routes from designated peers.
  - ***** clears all BGP IPv6 sessions with the switch's peers.
  - *ipv4_addr* resets IPv6 session with peer at specified IPv4 address.
  - *ipv6_addr* resets IPv6 session with peer at specified IPv6 address.
- ***RESET_TYPE*** reconfiguration type. Options include:
  - <no parameter> hard reset.
  - **soft** soft reset.
- ***DATA_FLOW*** restricts reset to inbound or outbound routes.
  - <no parameter> inbound and outbound routes are reset.
  - **in** inbound routes are reset.
  - **out** outbound routes are reset.
- ***VRF_INSTANCE*** specifies VRF instances.
  - <no parameter> clears routing table for context-active VRF.
  - **vrf** *vrf_name* clears routing table for the specified VRF.
  - **vrf all** clears routing table for all VRFs.
  - **vrf default** clears routing table for default VRF.

Examples

- This command removes all BGP IPv6 learned routes from the routing table:

```
switch#clear ipv6 bgp
switch#
```

- This command clears all of the switch's BGP IPv6 peering sessions:

```
switch#clear ip bgp *
switch#
```

clear ipv6 bgp neighbor *

The **clear ipv6 bgp neighbor *** command clears BGP neighbors belonging to the IPv6 transport address family. To clear BGP neighbors in the IPv4 transport address family, use the **clear ip bgp neighbor *** command.

Command Mode

Privileged EXEC

Command Syntax

clear ipv6 bgp neighbor * [*VRF_INSTANCE*]

Parameters

- ***VRF_INSTANCE*** specifies VRF instance for which IPv6 transport address family BGP neighbors will be cleared. Options include:
  - <no parameter> clears IPv6 BGP neighbors in the context-active VRF.
  - **vrf** *vrf_name* clears IPv6 BGP neighbors in the specified VRF.
  - **vrf all** clears IPv6 BGP neighbors in the all VRFs.
  - **vrf default** clears IPv6 BGP neighbors in the default VRF.

Examples

- This command clears all IPv6 BGP neighbors in the context-active VRF.

```
switch#clear ipv6 bgp neighbor *
switch#
```

- This command clears all IPv6 BGP neighbors in VRF “purple.”

```
switch#clear ipv6 bgp neighbor * vrf purple
switch#
```

distance bgp

The **distance bgp** command assigns an administrative distance to routes that the switch learns through BGP. Routers use administrative distances to select a route when two protocols provide routing information to the same destination. Distance values range from 1 to 255; lower distance values correspond to higher reliability. BGP routing tables do not include routes with a distance of 255.

The distance command assigns distance values to external, internal, and local BGP routes:

- **external**: Best-path routes learned from a neighbor external to the autonomous system. Default distance is 200.
- **internal**: Internal routes are routes learned from a BGP entity within the same autonomous system. Default distance is 200.
- **local**: Local routes are networks listed with a network router configuration command for that router or for networks that are redistributed from another process. Default distance is 200.

The **no distance bgp** and **default distance bgp** commands restore the default administrative distances by removing the **distance bgp** command from ***running-config***.

Command Mode

Router-BGP Configuration

Command Syntax

```
distance bgp external_dist [INTERNAL_LOCAL]
no distance bgp
default distance bgp
```

Parameters

- *external_dist* distance assigned to external routes. Values range from 1 to 255.
- ***INTERNAL_LOCAL*** distance assigned to internal and local routes. Values for both routes range from 1 to 255. Options include:
  - <no parameter> *external_dist* value is assigned to internal and local routes.
  - *internal_dist local_dist* values assigned to internal (*internal_dist)* and local (*local_dist*) routes.

Example

- This command assigns an administrative distance of 150 to external routes, 200 to internal, and 150 to local routes.

```
switch(config)#router bgp 1
switch(config-router-bgp)#distance bgp 150 200 150
switch(config-router-bgp)#
```

graceful-restart stalepath-time

The **graceful-restart stalepath-time** command specifies the maximum time that stale routes from a restarting BGP neighbor will be retained after a BGP session is re-established with that peer.

The **no graceful-restart stalepath-time** and **default graceful-restart stalepath-time** commands restore the default value of 300 seconds by deleting the **graceful-restart stalepath-time** statement from ***running-config***.

Command Mode

Router-BGP Configuration

Command Syntax

```
graceful-restart stalepath-time interval
no graceful-restart stalepath-time
default graceful-restart stalepath-time
```

Parameters

- *interval* Maximum period (in seconds) that stale routes from a restarting BGP neighbor will be retained after the BGP session is re-established. Value ranges from ***1*** to ***3600*** (60 minutes). Default is 300.

Example

- These commands configure the stale path retention interval to 15 minutes.

```
switch(config)#router bgp 1
switch(config-router-bgp)#graceful-restart stalepath-time 900
switch(config-router-bgp)#
```

graceful-restart-helper

The **graceful-restart helper** command enables BGP graceful restart helper mode on the switch for all BGP neighbors. When graceful restart helper mode is enabled, the switch will retain routes from neighbors which are capable of graceful restart while those neighbors are restarting BGP. Graceful restart is enabled by default. To configure graceful restart helper mode for a specific neighbor or peer group, use the **neighbor graceful-restart-helper** command. Individual neighbor configuration takes precedence over the global configuration.

The **no graceful-restart helper** command disables graceful restart helper mode on the switch. The **default graceful-restart helper** command enables graceful restart helper mode by removing the corresponding **no graceful-restart helper** command from ***running-config***.

Command Mode

Router-BGP Configuration

Command Syntax

```
graceful-restart helper
no graceful-restart helper
default graceful-restart helper
```

Example

- These commands disable graceful restart helper mode on the switch.

```
switch(config)#router bgp 1
switch(config-router-bgp)#no graceful-restart-helper
switch(config-router-bgp)#
```

ip as-path access-list

The **ip as-path access-list** command creates an access list to filter BGP route updates. If access list ***list_name*** does not exist, this command creates it. If it already exists, this command appends statements to the list.

The **no ip as-path access-list** and **default ip as-path access-list** commands delete the named access list.

Command Mode

Global Configuration

Command Syntax

```
ip as-path access-list list_name FILTER_TYPE regex ORIGIN
no ip as-path access-list list_name
default ip as-path access-list list_name
```

Parameters

- *list_name* the name of the AS path access list.
- ***FILTER_TYPE*** access resolution of the specified AS path. Options include:
  - **permit** access is permitted.
  - **deny** access is denied.
- *regex* a regular expression describing the AS path being filtered. Regular expressions are pattern matching strings that are composed of text characters and operators.
- ***ORIGIN*** the origin of the path information. Values include:
  - *<no parameter>* sets the origin to **any**.
  - **any** any BGP origin.
  - **egp** EGP origin.
  - **igp** IGP origin.
  - **incomplete** incomplete origin.

Example

- These commands create an AS path access list named “list1” which allows all BGP routes except those originating in AS 3.

```
switch(config)#ip as-path access-list list1 deny _3$
switch(config)#ip as-path access-list list1 permit .*
switch(config)#
```

ip as-path regex-mode

The **ip as-path regex-mode** command specifies how the switch will evaluate regular expressions describing AS paths in ACLs. When the regex mode is set to **asn**, AS numbers in the ACL are interpreted as AS numbers; only complete AS number matches in the AS path return a match. When it is set to **string**, AS numbers in the ACL are interpreted as strings; both complete AS number matches and longer AS numbers that include the target string return a match. The default mode is **asn**.

For example, **asn** mode will return “false” and **string** mode will return “true” when searching for “10” in an AS path of “100 200”.

The **no ip as-path regex-mode** and **default ip as-path regex-mode** commands restore the regex mode to **asn** by removing the **ip as-path regex-mode** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
ip as-path regex-mode MODE_SETTING
no ip as-path regex-mode
default ip as-path regex-mode
```

Parameters

- ***MODE_SETTING*** Specifies how regular expressions describing AS paths in AS path ACLs will be evaluated. Options include:
  - **asn** AS numbers in the ACL are interpreted as AS numbers; only complete AS number matches in the AS path return a match.
  - **string** AS numbers in the ACL are interpreted as strings; both complete AS number matches and longer AS numbers that include the target string return a match.

Example

- This command sets the regex mode to **string**.

```
switch(config)#ip as-path regex-mode string
switch(config)#
```

ip community-list expanded

The **ip community-list expanded** command creates and configures a BGP access list based on BGP communities. A BGP community access list filters route maps that are configured as BGP communities. The command uses regular expressions to name the communities specified by the list.

The **no ip community-list expanded** and **default ip community-list expanded** commands delete the specified community list by removing the corresponding **ip community-list expanded** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
ip community-list expanded listname FILTER_TYPE R_EXP
no ip community-list expanded listname
default community-list expanded listname
```

Parameters

- *listname* name of the community list. Valid input is text.
- ***FILTER_TYPE*** access resolution of the specified community. Options include:
  - **permit** access is permitted.
  - **deny** access is denied.
- ***R_EXP*** list of communities, formatted as a regular expression. Regular expressions are pattern matching strings that are composed of text characters and operators.

Example

- This command creates a BGP community list that permits routes from networks 20-24 and 30-34 in autonomous system 10.

```
switch(config)#ip community-list expanded list_2 permit 10:[2-3][0-4]_
switch(config)#
```

ip community-list standard

The **ip community-list standard** command creates and configures a BGP access list based on BGP communities. A BGP community list filters route maps that are configured as BGP communities.

The **no ip community-list standard** and **default ip community-list standard** commands delete the specified community list by removing the corresponding **ip community-list standard** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
ip community-list standard listname FILTER_TYPE COMM_1 [COMM_2...COMM_n]
no ip community-list standard listname
default ip community-list standard listname
```

Parameters

- *listname* name of the community list. Valid input is text.
- ***FILTER_TYPE*** access resolution of the specified community. Options include:
  - **permit** access is permitted.
  - **deny** access is denied.
- ***COMM_x*** community number or name, as specified in the route map that sets the community list number.
  - *aa:nn* AS and network number, separated by colon. Each value ranges from 1 to 4294967295.
  - *number* community number. Values range from 1 to 4294967040.
  - **internet** advertises route to Internet community.
  - **local-as** advertises route only to local peers.
  - **no-advertise** does not advertise route to any peer.
  - **no-export** advertises route only within BGP AS boundary.

Example

- This command creates a BGP community list (named list_9) that denies members of route maps configured as AS-network number 100:250.

```
switch(config)#ip community-list standard list_9 deny 100:250
switch(config)#
```

ip extcommunity-list expanded

The **ip extcommunity-list expanded** command creates an extended community list to filter VRF routes. The command uses regular expressions to name the communities specified by the list.

- **route targets** identify sites that may receive appropriately tagged routes.
- **site of origin** identifies the site where the router learned the route.

The **no ip extcommunity-list expanded** and **default ip extcommunity-list expanded** commands delete the specified extended community list by removing the corresponding **ip community-list expanded** statement from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
ip extcommunity-list expanded listname FILTER_TYPE R_EXP
no ip extcommunity-list expanded listname
default ip extcommunity-list expanded listname
```

Parameters

- *listname* name of the extended community list. Valid input is text.
- ***FILTER_TYPE*** access resolution of the specified extended community list. Options include:
  - **permit** access is permitted.
  - **deny** access is denied.
- ***R_EXP*** list of communities, formatted as a regular expression. Regular expressions are pattern matching strings that are composed of text characters and operators.
  - Expressions beginning ***RT:*** match the ***route target*** extended community attribute option.
  - Expressions beginning ***SoO:*** match the ***site of origin*** extended community attribute option.

  RT: and ***SoO:*** are case sensitive.

Example

- This command creates a BGP extended community list that denies routes from route target networks 20-24 and 30-34 in autonomous system 10.

```
switch(config)#ip extcommunity-list expanded list_1 deny RT:10:[2-3][0-4]_
switch(config)#
```

ip extcommunity-list standard

The **ip extcommunity-list standard** command creates an extended community list to filter VRF routes.

- **Route Target (rt)** identify sites that may receive appropriately tagged routes.
- **Site of Origin (soo)** identifies sites where the switch learned the route.

The **no ip extcommunity-list standard** and **default ip extcommunity-list standard** commands delete the specified extended community list by removing the corresponding **ip extcommunity-list standard** statement from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
ip extcommunity-list standard listname FILTER_TYPE COMM_1 [COMM_2...COMM_n]
no ip extcommunity-list standard listname
default ip extcommunity-list standard listname
```

Parameters

- *listname* name of the extended community list. Valid input is text.
- ***FILTER_TYPE*** access resolution of the specified extended community list. Options include:
  - **permit** access is permitted.
  - **deny** access is denied.
- ***COMM_x*** extended community attribute. Options include:
  - **rt** *aa:nn* route target, as specified by autonomous system:network number
  - **rt** *ip_addr:nn* route target, as specified by ip address:network number
  - **soo** *aa:nn* site of origin, as specified by autonomous system:network number
  - **soo** *ip_addr:nn* site of origin, as specified by ip address:network number

Example

- This command creates a BGP extended community list that denies routes from route target 100:250.

```
switch(config)#ip extcommunity-list standard list_9 deny rt 100:250
switch(config)#
```

maximum paths (BGP)

The **maximum-paths** command controls the maximum number of parallel eBGP routes that the switch supports. The default maximum is one route. The command provides an ECMP (equal cost multiple paths) parameter that controls the number of equal-cost paths that the switch stores in the routing table for each route.

For paths to be consider equal, they must have the same weight, local preference, AS-path length, and origin. To require that they also have the same multi-exit discriminator (MED) value, use the **bgp always-compare-med** command. To require that their AS paths have the same contents, use the **no bgp bestpath as-path multipath-relax** command.

The **no maximum-paths** and **default maximum-paths** commands restore the default values of the maximum number of parallel routes and the maximum number of ECMP paths by removing the corresponding command from ***running-config***.

Command Mode

Router-BGP Configuration

Command Syntax

```
maximum-paths paths [ecmp ecmp_paths]
no maximum-paths
default maximum-paths
```

Parameters

- *paths* maximum number of parallel routes. Default value is 1.
- *ecmp_paths* maximum number of ECMP paths for each route. Default is maximum value.

  Value for each parameter ranges from 1 to the number of interfaces available per ECMP group, which is platform dependent.

  - Arad: Value ranges from 1 to 128. Default value is 128.
  - FM6000: Value ranges from 1 to 32. Default value is 32.
  - PetraA: Value ranges from 1 to 16. Default value is 16.
  - Trident: Value ranges from 1 to 32. Default value is 32.
  - Trident-II: Value ranges from 1 to 128. Default value is 128.

Examples

- This command configures the maximum number of BGP parallel paths to 12 without changing the default ECMP value.

  ```
  switch(config)#router bgp 1
  switch(config-router-bgp)#maximum-paths 12
  ! Warning: maximum-paths will take effect after BGP restart.
  switch(config-router-bgp)#
  ```

- This command configures the maximum number of eBGP parallel routes to 2, with a maximum of 4 equal cost (ECMP) paths for each route.

  ```
  switch(config)#router bgp 1
  switch(config-router-bgp)#maximum-paths 2 ecmp 4
  ! Warning: maximum-paths will take effect after BGP restart.
  switch(config-router-bgp)#
  ```

neighbor activate

The **neighbor activate** command defines the configuration mode address family activation state of a specified address that is configured as a BGP neighbor. The switch sends the following announcements to addresses active in an address family:

- IPv4 address family: IPv4 capability and all network advertisements with IPv4 prefixes.
- IPv6 address family: IPv6 capability and all network advertisements with IPv6 prefixes.

The **bgp default** command configures the default address family activation state of addresses configured as BGP neighbors. The **neighbor activate** and **no neighbor activate** commands override the neighbor's default activation state within the configuration mode address family.

- **neighbor activate**: the specified address is active in the address family.
- **no neighbor activate**: the specified address is not active in the address family.

The **default neighbor activate** command removes the corresponding **neighbor activate** or **no neighbor activate** command from ***running-config***, restoring the default address family activation state for the specified neighbor address.

Command Mode

Router-BGP Configuration-Address-Family Configuration

Command Syntax

```
neighbor NEIGHBOR_ID activate
no neighbor NEIGHBOR_ID activate
default neighbor NEIGHBOR_ID activate
```

Parameters

- ***NEIGHBOR_ID*** IP address or peer group name. Values include:
  - *ipv4_addr* neighbor's IPv4 address.
  - *ipv6_addr* neighbor's IPv6 address.
  - *group_name* peer group name.

Limitations

The switch supports the advertisement of networks with IPv6 prefixes to IPv4 transport neighbors. The switch does not support the advertisement of networks with IPv4 prefixes to IPv6 transport neighbors.

Example

- The two neighbor activation commands activate the advertising of specified neighbors during IPv4 peering sessions. The **show active** command displays the result of the previous commands.

```
switch(config)#router bgp 1
switch(config-router-bgp)#no address-family ipv4
switch(config-router-bgp-af)#neighbor 172.41.18.15 activate
switch(config-router-bgp-af)#neighbor 172.49.22.6 activate
switch(config-router-bgp-af)#no neighbor 172.15.21.18 activate
switch(config-router-bgp-af)#show active
   address-family ipv4
      no neighbor 172.15.21.18 activate
      neighbor 172.49.22.6 activate
      neighbor 172.41.18.15 activate
switch(config-router-bgp-af)#exit
switch(config-router-bgp)#
```

neighbor allowas-in

The **neighbor allowas-in** command configures the switch to permit the advertisement of prefixes containing duplicate autonomous switch numbers (ASNs). This command programs the switch to ignore its ASN in the AS path of routes and allow them into the routing domain. This function is disabled by default.

The **no neighbor allowas-in** command applies the system default configuration.

The **default neighbor allowas-in** command applies the system default configuration for individual neighbors and applies the peer group's setting for neighbors that are members of a peer group.

The **no neighbor** command removes all configuration commands for the neighbor at the specified address.

Command Mode

Router-BGP Configuration

Command Syntax

```
neighbor NEIGHBOR_ID allowas-in [asn_quantity]
no neighbor NEIGHBOR_ID allowas-in
default neighbor NEIGHBOR_ID allowas-in
```

Parameters

- ***NEIGHBOR_ID*** IP address or peer group name. Values include:
  - *ipv4_addr* neighbor's IPv4 address.
  - *ipv6_addr* neighbor's IPv6 address.
  - *group_name* peer group name.
- *asn_quantity* Number of switches (ASN) allowed in path. Values range from 1 to 10. Default is 3.

Example

- This command activates the allowas-in function for the neighbor at 192.168.1.30.

```
switch(config)#router bgp 1
switch(config-router-bgp)#neighbor 192.168.1.30 allowas-in
switch(config-router-bgp)#
```

neighbor auto-local-addr

The **neighbor auto-local-addr** command configures the switch to automatically determine the local address to be used for the non-transport address family in NLRIs sent to the specified neighbor or peer group. This allows IPv4 NLRIs to be carried over IPv6 transport, or IPv6 NLRIs to be carried over IPv4 transport.

The **no neighbor auto-local-addr** command applies the system default configuration.

The **default neighbor auto-local-addr** command applies the system default configuration for individual neighbors, and applies the peer group's setting for neighbors that are members of a peer group.

Note While this feature works well in eBGP deployments in which the pairing routers are directly connected and have matching IP address configurations, multi-hop eBGP or iBGP deployments may require manual local address configuration.

To explicitly configure a local address for the non-transport address family for a specific neighbor or peer group, use the **neighbor local-v4-addr** command for IPv6 neighbors, or the **neighbor local-v6-addr** for IPv4 neighbors.

Command Mode

Router-BGP Configuration

Command Syntax

```
neighbor NEIGHBOR_ID auto-local-addr
no neighbor NEIGHBOR_ID auto-local-addr
default neighbor NEIGHBOR_ID auto-local-addr
```

Parameters

- ***NEIGHBOR_ID*** IP address or peer group name. Values include:
  - *neighbor_addr* neighbor's IP address.
  - *group_name* peer group name.

Example

- For the IPv6 neighbor at 2001:0DB8:c2a4:1761::2, these commands configure the switch to automatically determine the IPv4 NLRI value to be sent during peering sessions.

```
switch(config)#router bgp 1
switch(config-router-bgp)#neighbor 2001:0DB8:c2a4:1761::2 auto-local-addr
switch(config-router-bgp)#
```

neighbor default-originate

The **neighbor default-originate** command advertises a default route to a BGP neighbor or peer group. This default route overrides the default route advertised by any other means to the specified neighbor or peer group. However, the update generated by **neighbor default-originate** is not processed by neighbor route-map out policies.

If a route map is specified in this command, its set clauses are used to modify attributes of the exported default route, but its match clauses are not used to conditionally advertise the route. The default route is always advertised to the specified neighbor.

The **no neighbor default-originate** command applies the system default configuration.

The **default neighbor default-originate** command applies the system default configuration for individual neighbors and applies the peer group's setting for neighbors that are members of a peer group.

The **no neighbor** command removes all configuration commands for the neighbor at the specified address.

Command Mode

Router-BGP Configuration
Router-BGP Configuration-Address-Family

Command Syntax

```
neighbor NEIGHBOR_ID default-originate [MAP]
no neighbor NEIGHBOR_ID default-originate
default neighbor NEIGHBOR_ID default-originate
```

Parameters

- ***NEIGHBOR_ID*** IP address or peer group name. Values include:
  - *ipv4_addr* neighbor's IPv4 address.
  - *ipv6_addr* neighbor's IPv6 address.
  - *group_name* peer group name.
- ***MAP*** specifies route map that modifies attributes of the exported default route. Options include:
  - <no parameter> attributes are not modified by a route map.
  - **route-map** *map_name* attributes set by specified route map are assigned to the exported default route.

Example

- These commands advertise a default route to the BGP neighbor at 192.168.14.5.

```
switch(config)#router bgp 9
switch(config-router-bgp)#neighbor 192.168.14.5 default-originate
switch(config-router-bgp)#
```

neighbor description

The **neighbor description** command associates descriptive text with the specified peer or peer group.

The **no neighbor description** command removes the text association from the specified peer or peer group.

The **default neighbor description** command removes the text association from the specified peer for individual neighbors, and applies the peer group's description to neighbors that are members of a peer group.

The **no neighbor** command removes all configuration commands for the neighbor at the specified address or for the specified peer group.

Command Mode

Router-BGP Configuration

Command Syntax

```
neighbor NEIGHBOR_ID description description_string
no neighbor NEIGHBOR_ID description
default neighbor NEIGHBOR_ID description
```

Parameters

- ***NEIGHBOR_ID*** IP address or peer group name. Options include:
  - *ipv4_addr* neighbor's IPv4 address.
  - *ipv6_addr* neighbor's IPv6 address.
  - *group_name* peer group name.
- *description_string* text string to be associated with the neighbor or peer group.

Example

- This command associates the string PEER_1 with the peer located at 192.168.1.30.

```
switch(config)#router bgp 1
switch(config-router-bgp)#neighbor 192.168.1.30 description PEER_1
switch(config-router-bgp)#
```

neighbor ebgp-multihop

The **neighbor ebgp-multihop** command programs the switch to accept and attempt BGP connections to the external peers residing on networks not directly connected to the switch. The command does not establish the multihop if the only route to the peer is the default route (0.0.0.0).

The **no neighbor ebgp-multihop** command applies the system default configuration.

The **default neighbor ebgp-multihop** command applies the system default configuration for individual neighbors, and applies the peer group's setting for neighbors that are members of a peer group.

The **no neighbor** command removes all configuration commands for the neighbor at the specified address.

Command Mode

Router-BGP Configuration

Command Syntax

```
neighbor NEIGHBOR_ID ebgp-multihop [hop_number]
no neighbor NEIGHBOR_ID ebgp-multihop
default neighbor NEIGHBOR_ID ebgp-multihop
```

Parameters

- ***NEIGHBOR_ID*** IP address or peer group name. Values include:
  - *ipv4_addr* neighbor's IPv4 address.
  - *ipv6_addr* neighbor's IPv6 address.
  - *group_name* peer group name.
- *hop_number* time-to-live (hops). Values range from 1 to 255. Default value is 255.

Example

- This command programs the switch to accept and attempt BGP connections to the external peer located at 192.168.1.30, setting the hop limit to 32.

```
switch(config)#router bgp 1
switch(config-router-bgp)#neighbor 192.168.1.30 ebgp-multihop 32
switch(config-router-bgp)#
```

neighbor enforce-first-as

The **neighbor enforce-first-as** command causes a forced comparison of the first autonomous system (AS) in the AS path of eBGP routes received from a specified BGP peer or peer group to the configured remote external peer autonomous system number (ASN). Updates from the specified eBGP peers that do not include an ASN as first AS path (in the AS_PATH attribute) are discarded.

This behavior is enabled globally by default upon BGP configuration, and disabled for the specified neighbor or peer group by the **no** form of the command. To configure enforce-first-as globally, use the **bgp enforce-first-as** command.

Command Mode

Router-BGP Configuration

Command Syntax

```
neighbor NEIGHBOR_ID enforce-first-as
default neighbor NEIGHBOR_ID enforce-first-as
no neighbor NEIGHBOR_ID enforce-first-as
```

Parameters

- ***NEIGHBOR_ID*** IP address or peer group name. Values include:
  - *ipv4_addr* neighbor's IPv4 address.
  - *ipv6_addr* neighbor's IPv6 address.
  - *group_name* peer group name.

Example

- This command disables BGP enforce-first-as for the neighbors in peer group "region-3".

```
switch(config-router-bgp)#no neighbor region-3 enforce-first-as
switch(config-router-bgp)#
```

neighbor export-localpref

The **neighbor export-localpref** command determines the LOCAL_PREF value that is sent in BGP UPDATE packets to the specified peer or peer group. This command has no effect on external peers.

The **no neighbor export-localpref** command resets the LOCAL_PREF value to the system default of 100 in packets sent to the specified peer or peer group.

The **default neighbor export-localpref** command resets the LOCAL_PREF value to the system default of 100 for individual neighbors, and applies the peer group's setting for neighbors that are members of a peer group.

The **no neighbor** command removes all configuration commands for the neighbor at the specified address or the specified peer group.

Command Mode

Router-BGP Configuration

Command Syntax

```
neighbor NEIGHBOR_ID export-localpref preference
no neighbor NEIGHBOR_ID export-localpref
default neighbor NEIGHBOR_ID export-localpref
```

Parameters

- ***NEIGHBOR_ID*** IP address or peer group name. Values include:
  - *ipv4_addr* neighbor's IPv4 address.
  - *ipv6_addr* neighbor's IPv6 address.
  - *group_name* peer group name.
- *preference* preference value. Values range from 0 to 4294967295.

Example

- This command configures the switch to fill the LOCAL_PREF field with 200 in UPDATE packets that it sends to the peer located at 10.1.1.45.

```
switch(config)#router bgp 1
switch(config-router-bgp)#neighbor 10.1.1.45 export-localpref 200
switch(config-router-bgp)#
```

neighbor graceful-restart-helper

The **neighbor graceful-restart helper** command enables BGP graceful restart helper mode for the specified BGP neighbor or peer group. When graceful restart helper mode is enabled, the switch will retain routes from neighbors which are capable of graceful restart while those neighbors are restarting BGP. Graceful restart is enabled by default for all BGP neighbors. To configure graceful restart helper mode for all BGP neighbors, use the **graceful-restart-helper** command. Individual neighbor configuration takes precedence over the global configuration.

The **no neighbor graceful-restart helper** command disables graceful restart helper mode for the specified BGP neighbor or peer group. The **default neighbor graceful-restart helper** command enables graceful restart helper mode for the specified BGP neighbor or peer group by removing the corresponding **no neighbor graceful-restart helper** command from ***running-config***.

Command Mode

Router-BGP Configuration

Command Syntax

```
neighbor NEIGHBOR_ID graceful-restart helper
no neighbor NEIGHBOR_ID graceful-restart helper
default neighbor NEIGHBOR_ID graceful-restart helper
```

Parameters

- ***NEIGHBOR_ID*** IP address or peer group name. Values include:
  - *ipv4_addr* neighbor's IPv4 address.
  - *ipv6_addr* neighbor's IPv6 address.
  - *group_name* peer group name.

Example

- These commands disable graceful restart helper mode for the neighbor at 192.168.12.1.

```
switch(config)#router bgp 1
switch(config-router-bgp)#no neighbor 192.168.12.1 graceful-restart-helper
switch(config-router-bgp)#
```

neighbor import-localpref

The **neighbor import-localpref** command determines the local preference assigned to routes received from the specified external peer or peer group. This command has no effect on routes received from internal peers.

The **no neighbor import-localpref** command resets the local preference to the default of 100 for routes received from the specified peer or peer group.

The **default neighbor import-localpref** command resets the local preference to the default of 100 for individual neighbors, and applies the peer group's setting for neighbors that are members of a peer group.

The **no neighbor** command removes all configuration commands for the neighbor at the specified address.

Command Mode

Router-BGP Configuration

Command Syntax

```
neighbor NEIGHBOR_ID import-localpref preference
no neighbor NEIGHBOR_ID import-localpref
default neighbor NEIGHBOR_ID import-localpref
```

Parameters

- ***NEIGHBOR_ID*** IP address or peer group name. Values include:
  - *ipv4_addr* neighbor's IPv4 address.
  - *ipv6_addr* neighbor's IPv6 address.
  - *group_name* peer group name.
- *preference* preference value. Values range from 0 to 4294967295.

Example

- This command configures the switch to assign a local preference of 50 to routes received from the peer located at 192.168.1.30.

```
switch(config)#router bgp 1
switch(config-router-bgp)#neighbor 192.168.1.30 import-localpref 50
switch(config-router-bgp)#
```

neighbor local-as

The **neighbor local-as** command enables AS_PATH attribute modification for received eBGP routes, allowing the switch to appear as a member of a different AS to external peers. Arista switches do not prepend the local AS number to routes received from the eBGP neighbor; currently, we implement the command only as **neighbor local-as no-prepend replace-as**.

The **no neighbor local-as** command disables AS_PATH modification for the specified peer or peer group. The **default neighbor local-as** command disables AS_PATH modification for individual neighbors, and applies the peer group's setting for neighbors that are members of a peer group.

Command Mode

Router-BGP Configuration

Command Syntax

```
neighbor NEIGHBOR_ID local-as as_id no-prepend replace-as
no neighbor NEIGHBOR_ID local-as
default neighbor NEIGHBOR_ID local-as
```

Parameters

- ***NEIGHBOR_ID*** IP address or peer group name. Values include:
  - *ipv4_addr* neighbor's IPv4 address.
  - *ipv6_addr* neighbor's IPv6 address.
  - *group_name* peer group name.
- *as_id* AS number that is sent in outbound routing updates in place of the actual AS of the switch. Values range from 1 to 4294967295.

  This parameter cannot be set to the switch's AS number or to any AS number in the peer's network.

Example

- For the neighbor at 10.13.64.1, this command removes AS 300 from outbound routing updates and replaces it with AS 600.

```
switch(config)#router bgp 300
switch(config-router-bgp)#neighbor 10.13.64.1 local-as 600 no-prepend replace-as
switch(config-router-bgp)#
```

neighbor local-v4-addr

The **neighbor local-v4-addr** command specifies the next-hop value that the switch sends as the IPv4 NLRI value to neighbors with whom IPv6 transport peering is established.

The **no neighbor local-v4-addr** command applies the system default configuration.

The **default neighbor local-v4-addr** command applies the system default configuration for individual neighbors, and applies the peer group's setting for neighbors that are members of a peer group.

To configure the switch to automatically determine the IPv4 address to be sent as the next-hop in IPv4 NLRIs to an IPv6 neighbor, use the **neighbor auto-local-addr** command.

Command Mode

Router-BGP Configuration

Command Syntax

```
neighbor NEIGHBOR_ID local-v4-addr ipv4_local
no neighbor NEIGHBOR_ID local-v4-addr
default neighbor NEIGHBOR_ID local-v4-addr
```

Parameters

- ***NEIGHBOR_ID*** IP address or peer group name. Values include:
  - *ipv6_addr* neighbor's IPv6 address.
  - *group_name* peer group name.
- *ipv4_local* Next hop address.

Example

- For the neighbor at 2001:0DB8:c2a4:1761::2, these commands specify an IPv4 NLRI value of 10.7.5.11 to be sent during IPv6 transport peering sessions.

```
switch(config)#router bgp 1
switch(config-router-bgp)#neighbor 2001:0DB8:c2a4:1761::2 local-v4-addr
10.7.5.11
switch(config-router-bgp)#
```

neighbor local-v6-addr

The **neighbor local-v6-addr** command specifies the next-hop value that the switch sends as the IPv6 NLRI value to neighbors with whom IPv4 transport peering is established.

In IPv6 peering sessions, the switch sends the global IPv6 address of the interface that is used to transmit BGP updates.

The **no neighbor local-v6-addr** command applies the system default configuration.

The **default neighbor local-v6-addr** command applies the system default configuration for individual neighbors, and applies the peer group's setting for neighbors that are members of a peer group.

Command Mode

Router-BGP Configuration

Command Syntax

```
neighbor NEIGHBOR_ID local-v6-addr ipv6_local
no neighbor NEIGHBOR_ID local-v6-addr
default neighbor NEIGHBOR_ID local-v6-addr
```

Parameters

- ***NEIGHBOR_ID*** IP address or peer group name. Values include:
  - *ipv4_addr* neighbor's IPv4 address.
  - *group_name* peer group name.
- *ipv6_local* Next hop address (A:B:C:D:E:F:G:H).

Example

- For the neighbor at 10.7.5.11, these commands specify an IPv6 NLRI value that is sent during IPv4 transport peering sessions.

```
switch(config)#router bgp 1
switch(config-router-bgp)#neighbor 10.7.5.11 local-v6-addr
2001:0DB8:c2a4:1761::2
switch(config-router-bgp)#show active
router bgp 1
   bgp log-neighbor-changes
   bgp default ipv6-unicast
   neighbor 10.7.5.11 local-v6-addr 2001:0DB8:c2a4:1761::2
switch(config-router-bgp)#
```

neighbor maximum-routes

The **neighbor maximum-routes** command determines the number of BGP routes the switch accepts from a specified neighbor and defines an action when the limit is exceeded. The default value is 12,000. To remove the maximum routes limit, select a limit of zero.

When the number of routes received from a peer exceeds the limit, the switch generates an error message. This command can also configure the switch to disable peering with the neighbor. In this case, the neighbor state is reset only through a **clear ip bgp** command.

The **no neighbor maximum-routes** command applies the system default maximum-routes value of 12,000 for the specified peer.

The **default neighbor maximum-routes** command applies the system default value for individual neighbors, and applies the peer group's setting for neighbors that are members of a peer group.

The **no neighbor** command removes all configuration commands for the neighbor at the specified address.

Command Mode

Router-BGP Configuration

Command Syntax

```
neighbor NEIGHBOR_ID maximum-routes quantity [ACTION]
no neighbor NEIGHBOR_ID maximum-routes
default neighbor NEIGHBOR_ID maximum-routes
```

Parameters

- ***NEIGHBOR_ID*** IP address or peer group name. Values include:
  - *ipv4_addr* neighbor's IPv4 address.
  - *ipv6_addr* neighbor's IPv6 address.
  - *group_name* peer group name.
- *quantity* maximum number of routes. Values include:
  - 0 the switch does not define a route limit.
  - 1 to 4294967294 maximum number of routes.
- ***ACTION*** switch action when the route limit is exceeded. Values include:
  - <no parameter> peering is disabled and an error message is generated.
  - **warning-only** peering is not disabled, but an error message is generated.

Example

- This command configures the switch to accept 15000 routes for the neighbor at 10.3.16.210. If the neighbor exceeds 15000 routes, the switch disables peering with the neighbor.

```
switch(config)#router bgp 1
switch(config-router-bgp)#neighbor 110.3.16.210 maximum-routes 15000
switch(config-router-bgp)#
```

neighbor next-hop-peer

The **neighbor next-hop-peer** command configures the switch to list the peer address as the next hop in routes that it receives from the specified peer BGP-speaking neighbor or members of the specified peer group. This command overrides the next hop for all routes received from this neighbor or peer group.

The **no neighbor next-hop-peer** command applies the system default (no next-hop override) for the specified peer.

The **default neighbor next-hop-peer** command applies the system default for individual neighbors and applies the peer group's setting for neighbors that are members of a peer group.

The **no neighbor** command removes all configuration commands for the neighbor at the specified address or the specified peer group.

Command Mode

Router-BGP Configuration

Command Syntax

```
neighbor NEIGHBOR_ID next-hop-peer
no neighbor NEIGHBOR_ID next-hop-peer
default neighbor NEIGHBOR_ID next-hop-peer
```

Parameters

- ***NEIGHBOR_ID*** IP address or peer group name. Values include:
  - *ipv4_addr* neighbor's IPv4 address.
  - *ipv6_addr* neighbor's IPv6 address.
  - *group_name* peer group name.

Example

- This command configures the peer address of 10.3.2.24 as the next hop for routes advertised to the switch from the peer BGP neighbor.

```
switch(config)#router bgp 9
switch(config-router-bgp)#neighbor 10.3.2.24 next-hop-peer
switch(config-router-bgp)#
```

neighbor next-hop-self

The **neighbor next-hop-self** command configures the switch to list its address as the next hop in routes that it advertises to the specified BGP-speaking neighbor or neighbors in the specified peer group. This is used in networks where BGP neighbors do not directly access all other neighbors on the same subnet.

The **no neighbor next-hop-self** command applies the system default (no next-hop override) for the specified peer.

The **default neighbor next-hop-self** command applies the system default for individual neighbors and applies the peer group's setting for neighbors that are members of a peer group.

The **no neighbor** command removes all configuration commands for the neighbor at the specified address or for the specified peer group.

Command Mode

Router-BGP Configuration

Command Syntax

```
neighbor NEIGHBOR_ID next-hop-self
no neighbor NEIGHBOR_ID next-hop-self
default neighbor NEIGHBOR_ID next-hop-self
```

Parameters

- ***NEIGHBOR_ID*** IP address or peer group name. Values include:
  - *ipv4_addr* neighbor's IPv4 address.
  - *ipv6_addr* neighbor's IPv6 address.
  - *group_name* peer group name.

Example

- This command configures the switch as the next hop for the peer at 10.4.1.30.

```
switch(config)#router bgp 1
switch(config-router-bgp)#neighbor 10.4.1.30 next-hop-self
switch(config-router-bgp)#
```

neighbor out-delay

The **neighbor out-delay** command sets the period of time that a route update for the specified neighbor must be in the routing table before the switch exports it to BGP. The out delay interval is used for bundling routing updates.

The **no neighbor out-delay** command applies the system default (out-delay value of zero) for the specified peer.

The **default neighbor out-delay** command applies the system default for individual neighbors and applies the peer group's setting for neighbors that are members of a peer group.

The **no neighbor** command removes all configuration commands for the specified neighbor.

Command Mode

Router-BGP Configuration

Command Syntax

```
neighbor NEIGHBOR_ID out-delay delay_time
no neighbor NEIGHBOR_ID out-delay
default neighbor NEIGHBOR_ID out-delay
```

Parameters

- ***NEIGHBOR_ID*** IP address or peer group name. Values include:
  - *ipv4_addr* neighbor's IPv4 address.
  - *ipv6_addr* neighbor's IPv6 address.
  - *group_name* peer group name.
- *delay_time* the out delay period (seconds). Values range from 0 to 600. Default value is 0.

Example

- This command sets the out delay period to 5 seconds for the connection with the peer at 10.24.15.9.

```
switch(config)#router bgp 1
switch(config-router-bgp)#neighbor 10.24.15.9 out-delay 5
switch(config-router-bgp)#
```

neighbor password

The **neighbor password** command enables authentication on a TCP connection with a BGP peer. The plain-text version of the password is a string, up to 8 bytes in length. Peers must use the same password to ensure proper communication.

Running-config displays the encrypted version of the password. The encryption scheme is not strong by cryptographic standards; encrypted passwords should be treated in the same manner as plain-text passwords.

The **no neighbor password** command applies the system default for the specified peer, removing the neighbor password from the configuration and disabling authentication with the specified peer.

The **default neighbor password** command applies the system default for individual neighbors and applies the peer group's setting for neighbors that are members of a peer group.

The **no neighbor password** and **default neighbor password** commands remove the neighbor password from the configuration, disabling authentication with the specified peer.

The **no neighbor** command removes all configuration commands for the neighbor at the specified address.

Command Mode

Router-BGP Configuration

Command Syntax

```
neighbor NEIGHBOR_ID password [ENCRYPT_LEVEL] key_text
no neighbor NEIGHBOR_ID password
default neighbor NEIGHBOR_ID password
```

Parameters

- ***NEIGHBOR_ID*** IP address or peer group name. Values include:
  - *ipv4_addr* neighbor's IPv4 address.
  - *ipv6_addr* neighbor's IPv6 address.
  - *group_name* peer group name.
- ***ENCRYPT_LEVEL*** the encryption level of the *key_text* parameter. Values include:
  - <no parameter> indicates the *key_text* is in clear text.
  - **0** indicates *key_text* is in clear text. Equivalent to the <no parameter> case.
  - **7** indicates *key_text* is md5 encrypted.
- *key_text* the password.

Example

- This command specifies a password in clear text.

```
switch(config)#router bgp 1
switch(config-router-bgp)#neighbor 10.25.25.13 password 0 code123
switch(config-router-bgp)#
```

Running-config stores the password as an encrypted string.

neighbor peer-group (create)

Peer groups allow the user to apply settings to a group of BGP neighbors simultaneously. Once a peer group is created, the group name can be used as a parameter in neighbor configuration commands, and the configuration will be applied to all members of the group. Settings applied to an individual neighbor in the peer group override group settings.

The **neighbor peer-group (create)** command is used to create static BGP peer groups. Static peer groups are peer groups whose members are added manually. To assign BGP neighbors to a static peer group, use the **neighbor peer-group (neighbor assignment)** command. To create a dynamic peer group, use the **bgp listen range** command.

The **no neighbor peer-group (create)** and **default neighbor peer-group (create)** commands remove the specified static peer group from ***running-config***. When a static peer group is deleted, the neighbors that were members of that peer group lose any configuration that was inherited from the peer group. The **no bgp listen range** command removes a dynamic peer group.

The **no neighbor** command removes all configuration commands for the specified neighbor.

Command Mode

Router-BGP Configuration

Command Syntax

```
neighbor group_name peer-group
no neighbor group_name peer-group
default neighbor group_name peer-group
```

Parameters

- *group_name* peer group name.

Examples

- These commands create a BGP peer group called bgpgroup1, assign several neighbors to the group, apply a route map and adjust the configuration for one group member.

```
switch(config)#router bgp 9
switch(config-router-bgp)#neighbor bgpgroup1 peer-group
switch(config-router-bgp)#neighbor 10.1.1.1 peer-group bgpgroup1
switch(config-router-bgp)#neighbor 10.2.2.2 peer-group bgpgroup1
switch(config-router-bgp)#neighbor 10.3.3.3 peer-group bgpgroup1
switch(config-router-bgp)#neighbor bgpgroup1 route-map corporate in
switch(config-router-bgp)#neighbor 10.3.3.3 maximum-routes 5000
switch(config-router-bgp)#show active
router bgp 9
bgp log-neighbor-changes
   neighbor bgpgroup1 peer-group
   neighbor bgpgroup1 route-map corporate in
   neighbor bgpgroup1 maximum-routes 12000
   neighbor 10.1.1.1 peer-group bgpgroup1
   neighbor 10.2.2.2 peer-group bgpgroup1
   neighbor 10.3.3.3 peer-group bgpgroup1
   neighbor 10.3.3.3 maximum-routes 5000
switch(config-router-bgp)#
```

- This command removes peer group “bgpgroup1” from ***running-config***. The group members remain, but all settings that group members inherited from the peer group are removed.

```
switch(config-router-bgp)#no neighbor bgpgroup1 peer-group
switch(config-router-bgp)#show active
router bgp 9
bgp log-neighbor-changes
   neighbor 10.1.1.1 maximum-routes 12000
   neighbor 10.2.2.2 maximum-routes 12000
   neighbor 10.3.3.3 maximum-routes 5000
switch(config-router-bgp)#
```

neighbor peer-group (neighbor assignment)

Peer groups allow the user to apply settings to a group of BGP neighbors simultaneously. Once a peer group is created, the group name can be used as a parameter in neighbor configuration commands, and the configuration will be applied to all members of the group. Settings applied to an individual neighbor in the peer group override group settings.

The **neighbor peer-group (neighbor assignment)** command is used to assign BGP neighbors to an existing static peer group. To create a static peer group, use the **neighbor peer-group (create)** command. A neighbor can only belong to one peer group, so issuing this command for a neighbor that is already a member of another group will remove it from that group.

The **no neighbor peer-group** and **default neighbor peer-group** commands remove the specified neighbor from all peer groups. When a neighbor is removed from a peer group, the neighbor retains the configuration inherited from the peer group.

The **no neighbor** command removes all configuration commands for the specified neighbor.

Command Mode

Router-BGP Configuration

Command Syntax

```
neighbor NEIGHBOR_ADDR peer-group group_name
no neighbor NEIGHBOR_ADDR peer-group
default neighbor NEIGHBOR_ADDR peer-group
```

Parameters

- ***NEIGHBOR_ADDR*** Address of a neighbor being added to peer group. Values include:
  - *ipv4_addr* neighbor's IPv4 address.
  - *ipv6_addr* neighbor's IPv6 address.
- *group_name* peer group name.

Examples

- These commands create a BGP peer group called bgpgroup1, assign several neighbors to the group, and apply a route map.

```
switch(config)#router bgp 9
switch(config-router-bgp)#neighbor bgpgroup1 peer-group
switch(config-router-bgp)#neighbor 10.1.1.1 peer-group bgpgroup1
switch(config-router-bgp)#neighbor 10.2.2.2 peer-group bgpgroup1
switch(config-router-bgp)#neighbor 10.3.3.3 peer-group bgpgroup1
switch(config-router-bgp)#neighbor bgpgroup1 route-map corporate in
switch(config-router-bgp)#
```

- This command removes the neighbor at 1.1.1.1 from the peer group. All settings that neighbor 10.1.1.1 inherited from the peer group are maintained.

```
switch(config-router-bgp)#no neighbor 10.1.1.1 peer-group
switch(config-router-bgp)#
```

neighbor remote-as

The **neighbor remote-as** command configures the expected AS number for a neighbor (peer). This configuration is required to establish a neighbor connection. Internal neighbors have the same AS number; external neighbors have different AS numbers.

Note To establish a BGP session, there must be an IPv4 router ID configured in the same VRF or at least one L3 interface with an IPv4 address in the same VRF. If the VRF contains no L3 interfaces with IPv4 addresses (e.g., in an IPv6-only environment), configure an appropriate router ID using the **router-id (BGP)** command.

The **no neighbor remote-as** command applies the system default for the specified peer or peer group.

The **default neighbor remote-as** command applies the system default for individual neighbors and applies the peer group's setting for neighbors that are members of a peer group.

The **no neighbor** command removes all configuration commands for the neighbor at the specified address.

Command Mode

Router-BGP Configuration

Command Syntax

```
neighbor NEIGHBOR_ID remote-as as_id
no neighbor NEIGHBOR_ID remote-as
default neighbor NEIGHBOR_ID remote-as
```

Parameters

- ***NEIGHBOR_ID*** IP address or peer group name. Values include:
  - *ipv4_addr* neighbor's IPv4 address.
  - *ipv6_addr* neighbor's IPv6 address.
  - *group_name* peer group name.
- *as_id* Autonomous system (AS) of the peer. Values range from 1 to 4294967295.

Example

- This command establishes a BGP connection with the router at 10.4.3.10 in AS 300.

```
switch(config)#router bgp 9
switch(config-router-bgp)#neighbor 10.4.3.10 remote-as 300
switch(config-router-bgp)#
```

neighbor remove-private-as

The **neighbor remove-private-as** command removes private autonomous system numbers from outbound routing updates for external BGP (eBGP) neighbors. When the autonomous system path includes both private and public autonomous system numbers, the ***REMOVAL*** parameter specifies how the private autonomous system number is removed.

The **no neighbor remove-private-as** command applies the system default (preserves private AS numbers) for the specified peer.

The **default neighbor remove-private-as** command applies the system default for individual neighbors and applies the peer group's setting for neighbors that are members of a peer group.

The **no neighbor** command removes all configuration commands for the neighbor at the specified address.

Command Mode

Router-BGP Configuration

Command Syntax

```
neighbor NEIGHBOR_ID remove-private-as [REMOVAL]
no neighbor NEIGHBOR_ID remove-private-as
default neighbor NEIGHBOR_ID remove-private-as
```

Parameters

- ***NEIGHBOR_ID*** IP address or peer group name. Values include:
  - *ipv4_addr* neighbor's IPv4 address.
  - *ipv6_addr* neighbor's IPv6 address.
  - *group_name* peer group name.
- ***REMOVAL*** Specifies removal of private autonomous AS number when path includes both private and public numbers. Values include:
  - <no parameter> private AS numbers is not removed.
  - all removes all private AS numbers from AS path in outbound updates.
  - all replace-as all private AS numbers in AS path are replaced with router's local AS number.

Example

- These commands program the switch to remove private AS numbers from outbound routing updates for the eBGP neighbor at 10.5.2.11.

```
switch(config)#router bgp 9
switch(config-router-bgp)#neighbor 10.5.2.11 remove-private-as
switch(config-router-bgp)#
```

- This command replaces all private AS numbers in the AS path with the switch's local AS number.

```
switch(config)#router bgp 9
switch(config-router-bgp)#neighbor 10.5.2.11 remove-private-as all replace-as
switch(config-router-bgp)#
```

neighbor route-map (BGP)

The **neighbor route-map** command applies a route map to inbound or outbound BGP routes. When a route map is applied to outbound routes, the switch will advertise only routes matching at least one section of the route map. Only one outbound route map and one inbound route map can be applied to a given neighbor. A new route map applied to a neighbor will replace the previous route map.

The command is available in Router-BGP and Router-BGP-Address-Family configuration modes. The mode in which the command is executed determines the scope of the command:

- In Router-BGP mode, the route map is applied to specified neighbor in all peering sessions where it is advertised.
- In Router-BGP-Address-Family mode, the route map is applied to the neighbors only in peering sessions corresponding to the configuration mode address family.

The **no neighbor route-map** command discontinues the application of the specified route map for the specified neighbor and direction. Removing a route map from one direction does not remove it from the other if it has been applied to both.

The **default neighbor route-map** command applies the system default (no route map) for individual neighbors, and applies the peer group's setting for neighbors that are members of a peer group.

Command Mode

Router-BGP Configuration
Router-BGP Configuration-Address-Family

Command Syntax

```
neighbor NEIGHBOR_ID route-map map_name DIRECTION
no neighbor NEIGHBOR_ID route-map map_name DIRECTION
default neighbor NEIGHBOR_ID route-map map_name DIRECTION
```

Parameters

- ***NEIGHBOR_ID*** IP address or peer group name. Values include:
  - *ipv4_addr* neighbor's IPv4 address.
  - *ipv6_addr* neighbor's IPv6 address.
  - *group_name* peer group name.
- *map_name* name of a route map.
- ***DIRECTION*** routes to which the route map is applied. Options include:
  - **in** route map is applied to inbound routes.
  - **out** route map is applied to outbound routes.

Example

- This command applies a route map named ***inner-map*** to a BGP inbound route from 10.5.2.11

```
switch(config)#router bgp 9
switch(config-router-bgp)#neighbor 10.5.2.11 route-map inner-map in
switch(config-router-bgp)#
```

neighbor route-reflector-client

Participating BGP routers within an AS communicate eBGP-learned routes to all of their peers, but to prevent routing loops they must not re-advertise iBGP-learned routes within the AS. To ensure that all members of the AS share the same routing information, a fully meshed network topology (in which each member router of the AS is connected to every other member) can be used, but this topology can result in high volumes of iBGP messages when it is scaled. Instead, in larger networks one or more routers can be configured as route reflectors.

A route reflector is configured to re-advertise routes learned through iBGP to a group of BGP neighbors within the AS (its clients), eliminating the need for a fully meshed topology.

The **neighbor route-reflector-client** command configures the switch to act as a route reflector and configures the specified neighbor as one of its clients. Additional clients are specified by re-issuing the command.

The **no neighbor route-reflector-client** and **default neighbor route-reflector-client** commands disable route reflection by deleting the **neighbor route-reflector-client** command from ***running-config***.

Command Mode

Router-BGP Configuration

Command Syntax

```
neighbor NEIGHBOR_ID route-reflector-client
no neighbor NEIGHBOR_ID route-reflector-client
default neighbor NEIGHBOR_ID route-reflector-client
```

Parameters

- ***NEIGHBOR_ID*** IP address of neighbor. Values include:
  - *ipv4_addr* neighbor's IPv4 address.
  - *ipv6_addr* neighbor's IPv6 address.
  - *group_name* peer group name.

Related Commands

- **bgp client-to-client reflection**

Example

- This command configures the switch as a route reflector and the neighbor at 10.5.2.1 as one of its clients.

```
switch(config)#router bgp 9
switch(config-router-bgp)#neighbor 10.5.2.11 route-reflector-client
switch(config-router-bgp)#
```

neighbor send-community

The **neighbor send-community** command configures the switch to send community attributes to the specified BGP neighbor.

The **no neighbor send-community** command applies the system default (not sending community attributes) for the specified peer.

The **default neighbor send-community** command applies the system default for individual neighbors and applies the peer group's setting for neighbors that are members of a peer group.

Command Mode

Router-BGP Configuration

Command Syntax

```
neighbor NEIGHBOR_ID send-community
no neighbor NEIGHBOR_ID send-community
default neighbor NEIGHBOR_ID send-community
```

Parameters

- ***NEIGHBOR_ID*** IP address or peer group name. Values include:
  - *ipv4_addr* neighbor's IPv4 address.
  - *ipv6_addr* neighbor's IPv6 address.
  - *group_name* peer group name.

Example

- This command configures the switch to send community attributes to the neighbor at address 10.5.2.23.

```
switch(config)#router bgp 9
switch(config-router-bgp)#neighbor 10.5.2.23 send-community
switch(config-router-bgp)#
```

neighbor shutdown

The **neighbor shutdown** command disables the specified neighbor. Disabling a neighbor also terminates all of its active sessions and removes associated routing information.

The **no neighbor shutdown** command enables the specified peer.

The default neighbor shutdown command enables individual neighbors and applies the peer group's setting for neighbors that are members of a peer group.

Command Mode

Router-BGP Configuration

Command Syntax

```
neighbor NEIGHBOR_ID shutdown
no neighbor NEIGHBOR_ID shutdown
default neighbor NEIGHBOR_ID shutdown
```

Parameters

- ***NEIGHBOR_ID*** IP address or peer group name. Values include:
  - *ipv4_addr* neighbor's IPv4 address.
  - *ipv6_addr* neighbor's IPv6 address.
  - *group_name* peer group name.

Example

- This command disables the neighbor at 10.5.2.23.

```
switch(config)#router bgp 9
switch(config-router-bgp)#neighbor 10.5.2.23 shutdown
switch(config-router-bgp)#
```

neighbor soft-reconfiguration

By default, inbound BGP routes that are filtered out by the import policy are still stored on the switch. Because all routes are retained, this allows policies to be changed without resetting BGP sessions. It also allows the switch to display all advertised routes when the **show ip bgp neighbor advertised-routes** command is issued.

The **no neighbor soft-reconfiguration** command configures the switch to discard information about routes received from the specified neighbor or group that fail the import policy.

The **neighbor soft-reconfiguration** command restores the system default behavior (retaining routes from the specified neighbor or group regardless of import policy).

The default neighbor soft-reconfiguration command applies the system default (retaining all routes) for individual neighbors and applies the peer group's setting for neighbors that are members of a peer group.

Command Mode

Router-BGP Configuration

Command Syntax

```
neighbor NEIGHBOR_ID soft-configuration inbound [SCOPE]
no neighbor NEIGHBOR_ID soft-configuration inbound
default neighbor NEIGHBOR_ID soft-configuration inbound
```

Parameters

- ***NEIGHBOR_ID*** IP address or peer group name. Values include:
  - *ipv4_addr* neighbor's IPv4 address.
  - *ipv6_addr* neighbor's IPv6 address.
  - *group_name* peer group name.
- ***SCOPE*** determines how routes including the switch's AS number are handled. Values include:
  - <no parameter> routes including the switch's AS number are discarded.
  - **all** routes including the switch's AS number are retained.

Example

- This command configures the switch to discard information about routes from the neighbor at 10.5.2.23 which are filtered out by the switch's import policies.

```
switch(config)#router bgp 9
switch(config-router-bgp)#no neighbor 10.5.2.23 soft-reconfiguration inbound
switch(config-router-bgp)#
```

neighbor timers

The **neighbor timers** command configures the BGP keepalive and hold times for a specified peer connection. The **timers bgp** command configures the times on all peer connections for which an individual command is not specified.

- Keepalive time is the period between the transmission of consecutive keepalive messages.
- Hold time is the period the switch waits for a KEEPALIVE or UPDATE message before it disables peering.

The hold time must be at least 3 seconds and should be three times longer than the keepalive setting.

The **no neighbor timers** command applies the system default for the specified peer or group (the timers specified by the **timers bgp** command).

The default neighbor timers command applies the system default for individual neighbors and applies the peer group's setting for neighbors that are members of a peer group.

The **no neighbor** command removes all configuration commands for the neighbor at the specified address.

Command Mode

Router-BGP Configuration

Command Syntax

```
neighbor NEIGHBOR_ID timers keep_alive hold_time
no neighbor NEIGHBOR_ID timers
default neighbor NEIGHBOR_ID timers
```

Parameters

- ***NEIGHBOR_ID*** IP address or peer group name. Values include:
  - *ipv4_addr* neighbor's IPv4 address.
  - *ipv6_addr* neighbor's IPv6 address.
  - *group_name* peer group name.
- *keep_alive* keepalive period, in seconds. Values include
  - **0** keepalive messages are not sent
  - **1** to **3600** keepalive time (seconds).
- *hold_time* hold time. Values include
  - **0** peering is not disabled by timeout expiry; keepalive packets are not sent.
  - **3** to **7200** hold time (seconds).

Example

- This command sets the keepalive time to 30 seconds and the hold time to 90 seconds for the connection with the peer at 10.24.15.9.

```
switch(config)#router bgp 9
switch(config-router-bgp)#neighbor 10.24.15.9 timers 30 90
switch(config-router-bgp)#
```

neighbor transport connection-mode

The **neighbor transport connection-mode** command sets the TCP connection for the specified BGP neighbor or peer group to passive mode. When the peer's transport connection mode is set to passive, it accepts TCP connections for BGP but does not initiate them.

The **no neighbor transport connection-mode** command sets the specified BGP neighbor or peer group to active connection mode. BGP peers in active mode can both accept and initiate TCP connections for BGP. This is the default behavior.

The **default neighbor transport connection-mode** command restores the default connection mode. The default mode is "active" for individual BGP peers, or the mode inherited from the peer group for peer group members.

Command Mode

Router-BGP Configuration

Command Syntax

```
neighbor NEIGHBOR_ID transport connection-mode passive
no neighbor NEIGHBOR_ID transport connection-mode
default neighbor NEIGHBOR_ID transport connection-mode
```

Parameters

- ***NEIGHBOR_ID*** IP address or peer group name. Values include:
  - *ipv4_addr* neighbor's IPv4 address.
  - *ipv6_addr* neighbor's IPv6 address.
  - *group_name* peer group name.

Example

- These commands configure the neighbor at IP address 10.2.2.14 to not initiate TCP connections for BGP peering.

```
switch(config)#router bgp 300
switch(config-router-bgp)#neighbor 10.2.2.14 transport connection-mode passive
switch(config-router-bgp)#
```

neighbor update-source

The **neighbor update-source** command specifies the interface that BGP sessions use for TCP connections. By default, BGP sessions use the neighbor's closest interface (also known as the best local address).

The **no neighbor update-source** command applies the system default (using best local address for TCP connections) for the specified peer or group.

The default neighbor update-source command applies the system default for individual neighbors and applies the peer group's setting for neighbors that are members of a peer group.

The **no neighbor** command removes all configuration commands for the neighbor at the specified address.

Command Mode

Router-BGP Configuration

Command Syntax

```
neighbor NEIGHBOR_ID update-source INTERFACE
no neighbor NEIGHBOR_ID update-source
default neighbor NEIGHBOR_ID update-source
```

Parameters

- ***NEIGHBOR_ID*** IP address or peer group name. Values include:
  - *ipv4_addr* neighbor's IPv4 address.
  - *ipv6_addr* neighbor's IPv6 address.
  - *group_name* peer group name.
- ***INTERFACE*** Interface type and number. Options include:
  - **ethernet** *e_num* Ethernet interface specified by *e_num*.
  - **loopback** *l_num* loopback interface specified by *l_num*.
  - **management** *m_num* management interface specified by *m_num*.
  - **port-channel** *p_num* port channel interface specified by *p_num*.
  - **vlan** *v_num* VLAN interface specified by *v_num*.

Example

- This command configures the switch to use Ethernet interface 10 for TCP connections for the neighbor at 10.2.2.14.

```
switch(config)#router bgp 9
switch(config-router-bgp)#neighbor 10.2.2.14 update-source ethernet 10
switch(config-router-bgp)#
```

neighbor weight

The **neighbor weight** command assigns a weight attribute value to paths from the specified neighbor. Weight is the first parameter that the BGP best-path selection algorithm considers. When multiple paths to a destination prefix exist, the best-path selection algorithm prefers the path with the highest weight. Other attributes are used only when all paths to the prefix have the same weight.

Weight values range from 0 to 65535 and are not propagated to other switches through route updates. The default weight for paths that the router originates is 32768; the default weight for routes received through BGP is 0.

A path's BGP weight is also configurable through route maps. Weight values set through route map commands have precedence over neighbor weight command values.

The **no neighbor weight** command applies the system default (32768 for router-originated paths, 0 for routes received through BGP) for the specified peer or group.

The **default neighbor weight** command applies the system default for individual neighbors, and applies the peer group's setting for neighbors that are members of a peer group.

The **no neighbor** command removes all configuration commands for the neighbor at the specified address.

Command Mode

Router-BGP Configuration

Command Syntax

```
neighbor NEIGHBOR_ID weight weight_value
no neighbor NEIGHBOR_ID weight
default neighbor NEIGHBOR_ID weight
```

Parameters

- ***NEIGHBOR_ID*** IP address or peer group name. Values include:
  - *ipv4_addr* neighbor's IPv4 address.
  - *ipv6_addr* neighbor's IPv6 address.
  - *group_name* peer group name.
- *weight_value* weight value. Values range from 1 to 65535.

Example

- This command specifies a weight of 4000 for all paths from the neighbor at 10.1.2.5

```
switch(config)#router bgp 9
switch(config-router-bgp)#neighbor 10.1.2.5 weight 4000
switch(config-router-bgp)#
```

network (BGP)

The **network** command specifies a network for advertisement through UPDATE packets to BGP peers. The configuration zeros the host portion of the specified network address; for example, 192.0.2.4/24 is stored as 192.0.2.0/24. A route map option is available for assigning attributes to the network.

The command is available in Router-BGP and Router-BGP-Address-Family configuration modes. The mode in which the command is issued does not affect the command. The scope of the command depends on the specified network address:

- Commands with an IPv4 address are advertised to peers activated in the IPv4 address family.
- Commands with an IPv6 address are advertised to peers activated in the IPv6 address family.

The **no network** and **default network** commands remove the network from the routing table, preventing its advertisement.

Command Mode

Router-BGP Configuration
Router-BGP Configuration-Address-Family

Command Syntax

```
network NET_ADDRESS [ROUTE_MAP]
no network NET_ADDRESS
default network NET_ADDRESS
```

Parameters

- ***NET_ADDRESS*** IP address range. Entry options include:
  - *ipv4_subnet* IPv4 subnet (CIDR notation).
  - *ipv4_addr* **mask** *subnet* IPv4 subnet (address-mask notation).
  - *ipv6_prefix* neighbor's IPv6 prefix (CIDR notation).
- ***ROUTE_MAP*** specifies route map that assigns attribute values to the network. Options include:
  - <no parameter> attributes are not assigned through a route map.
  - **route-map** *map_name* attributes listed by specified route map are assigned to the network.

Example

- This command enables BGP advertising for the network located at 10.1.2.5. The configuration stores the network as 10.1.2.5.

```
switch(config)#router bgp 9
switch(config-router-bgp)#network 10.1.2.5
switch(config-router-bgp)#
```

no neighbor

The **no neighbor** command removes all neighbor configuration commands for the specified neighbor. Commands removed by the **no neighbor** command include:

- **neighbor description**
- **neighbor ebgp-multihop**
- **neighbor export-localpref**
- **neighbor import-localpref**
- **neighbor local-as**
- **neighbor maximum-routes**
- **neighbor next-hop-peer**
- **neighbor next-hop-self**
- **neighbor out-delay**
- **neighbor password**
- **neighbor peer-group (create)**
- **neighbor peer-group (neighbor assignment)**
- **neighbor remote-as**
- **neighbor remove-private-as**
- **neighbor route-map (BGP)**
- **neighbor route-reflector-client**
- **neighbor send-community**
- **neighbor timers**
- **neighbor update-source**

Neighbor settings can be removed individually; refer to the command description page of the desired command for details. Neighbor settings for a peer group must be removed individually.

Command Mode

Router-BGP Configuration

Command Syntax

```
no neighbor NEIGHBOR_ID
default neighbor NEIGHBOR_ID
```

Parameters

- ***NEIGHBOR_ID*** IP address. Options include:
  - *ipv4_addr* neighbor's IPv4 address.
  - *ipv6_addr* neighbor's IPv6 address.

Example

- This command removes all neighbor configuration commands for the neighbor at 10.1.1.1.

```
switch(config)#router bgp 9
switch(config-router-bgp)#no neighbor 10.1.1.1
switch(config-router-bgp)#
```

redistribute (BGP)

The **redistribute** command enables the redistribution of specified routes to the BGP domain.

The **no redistribute** and **default redistribute** commands disable route redistribution from the specified domain by removing the corresponding **redistribute** command from ***running-config***.

Important! Aggregate routes are redistributed automatically, and their redistribution cannot be disabled.

Command Mode

Router-BGP Configuration

Command Syntax

```
redistribute ROUTE_TYPE [ROUTE_MAP]
no redistribute ROUTE_TYPE
default redistribute ROUTE_TYPE
```

Parameters

- ***ROUTE_TYPE*** source from which routes are redistributed. Options include:
  - **connected** routes that are established when IP is enabled on an interface.
  - **match nssa-external** all OSPF NSSA external routes.
  - **match nssa-external 1** type 1 OSPF NSSA external routes.
  - **match nssa-external 2** type 2 OSPF NSSA external routes.
  - **ospf** routes from an OSPF domain.
  - **ospf match external** routes external to the AS, but imported from OSPF.
  - **ospf match internal** OSPF routes that are internal to the AS.
  - **ospf match nssa-external** all OSPF NSSA external routes.
  - **ospf match nssa-external 1** type 1 OSPF NSSA external routes.
  - **ospf match nssa-external 2** type 2OSPF NSSA external routes.
  - **ospf3** routes from an OSPFv3 domain.
  - **ospf3 match external** routes external to the AS, but imported from OSPFv3.
  - **ospf3 match internal** OSPFv3 routes that are internal to the AS.
  - **rip** routes from a RIP domain.
  - **static** IP static routes.
- ***ROUTE_MAP*** route map that determines the routes that are redistributed. Options include:
  - <no parameter> all routes are redistributed.
  - **route-map** *map_name* only routes in the specified route map are redistributed.

Example

- This command redistributes OSPF routes into the BGP domain.

```
switch(config)#router bgp 9
switch(config-router-bgp)#redistribute OSPF
switch(config-router-bgp)#
```

router-id (BGP)

The **router-id** command sets the local router BGP router ID.

When no ID has been specified, the local router ID is set to the following:

- The loopback IP address when a single loopback interface is configured.
- The loopback with the highest IP address when multiple loopback interfaces are configured.
- The highest IP address on a physical interface when no loopback interfaces are configured.

Important! The router-id must be specified if the switch has no IPv4 addresses configured.

The **no router-id** and **default router-id** commands remove the **router-id** command from ***running-config***.

Command Mode

Router-BGP Configuration

Command Syntax

```
router-id id_num
no router-id [id_num]
default router-id [id_num]
```

Parameters

- *id_num* router ID number (32-bit dotted decimal notation).

Example

- This command configures the fixed router ID address of 10.10.4.11

```
switch(config)#router bgp 9
switch(config-router-bgp)#router-id 10.10.4.11
switch(config-router-bgp)#
```

router bgp

The **router bgp** command places the switch in router-BGP configuration mode. If BGP was not previously instantiated, this command creates a BGP instance with the specified AS number. Router-BGP configuration mode is not a group change mode; ***running-config*** is changed immediately after commands are executed. The **exit** command does not affect the configuration.

When a BGP instance exists, the command must include the AS number of the existing BGP instance. Running this command with a different AS number generates an error message.

The **no router bgp** and **default router bgp** commands delete the BGP instance.

Refer to Router-BGP Configuration Mode (Includes Address-Family Mode) (page 1843) for a list of commands available in router-BGP configuration mode.

The **exit** command returns the switch to global configuration mode.

Command Mode

Global Configuration

Command Syntax

```
router bgp as_id
no router bgp
default router bgp
```

Parameters

- *as_id* Autonomous system (AS) number. Values range from 1 to 4294967295.

Examples

- This command creates a BGP instance with AS number 64500.

```
switch(config)#router bgp 64500
switch(config-router-bgp)#
```

- This command attempts to open a BGP instance with a different AS number from that of the existing instance. The switch displays an error and stays in global configuration mode.

```
switch(config)#router bgp 64501
% BGP is already running with AS number 64500
switch(config)#
```

- This command exits BGP configuration mode.

```
switch(config-router-bgp)#exit
switch(config)#
```

- This command deletes the BGP instance.

```
switch(config)#no router bgp
switch(config)#
```

show bgp instance

The **show bgp instance** command displays summary Border Gateway Protocol (BGP) information about the BGP instance in the specified VRF or in all VRFs.

Command Mode

EXEC

Command Syntax

show bgp instance [*VRF_INSTANCE*]

Parameters

- ***VRF_INSTANCE*** specifies VRF instances.
  - <no parameter> displays BGP information for the context-active VRF.
  - **vrf** *vrf_name* displays BGP information for the specified VRF.
  - **vrf all** displays BGP information for all VRFs.
  - **vrf default** displays BGP information for the default VRF.

Examples

- This command displays information about the BGP instance in the context-active VRF.

```
switch>show bgp instance
BGP instance information for VRF purple
BGP Local AS:  64497, Router ID: 1.2.3.5
Total peers:              5
Configured peers:         3
  UnConfigured peers:     2
  Disabled peers:         0
  Established peers:      3
Graceful restart helper mode enabled
End of rib timer timeout: 00:05:00
BGP Convergence timer is inactive
BGP Convergence information:
    BGP has converged:   no
    Outstanding EORs:    0,     Outstanding Keepalives: 0
    Convergence timeout: 00:10:00
switch>
```

- This command displays information about the BGP instance in the default VRF.

```
switch>show bgp instance vrf default
BGP instance information for VRF default
BGP Local AS:  64503, Router ID: 1.2.3.5
Total peers:              1
Configured peers:         1
  UnConfigured peers:     0
  Disabled peers:         0
  Established peers:      0
Graceful restart helper mode enabled
End of rib timer timeout: 00:05:00
BGP Convergence timer is inactive
BGP Convergence information:
    BGP has converged:   no
    Outstanding EORs:    0,     Outstanding Keepalives: 0
    Convergence timeout: 00:10:00
switch>
```

show ip as-path access-list

The **show ip as-path access-list** command displays BGP filters on the switch. Specifying an access list displays the statements from that access list. Entering the command without parameters displays the statements from all access lists on the switch.

Command Mode

EXEC

Command Syntax

show ip as-path access-list [list_name]

Parameters

- *list_name* the name of an AS path access list.

Example

- This command displays the contents of the AS path access list named “list1.”

```
switch>show ip as-path access-list list1
ip as-path access-list list1 deny _3$
ip as-path access-list list1 permit .*
switch>
```

show ip bgp

The **show ip bgp** command displays Border Gateway Protocol (BGP) IPv4 routing table entries. The output format depends on the command parameters:

- Data block format displays comprehensive information for each specified BGP routing table entry.
- Tabular format displays routing table entries in a tabular format for the specified IPv4 addresses.

Command Mode

EXEC

Command Syntax

show ip bgp [*FILTER*] [*VRF_INSTANCE*]

Parameters

- ***FILTER*** routing table entries that the command displays. Values include*:*
  - <no parameter> displays all routing table entries. Tabular format.
  - **detail** displays all routing table entries. Data block format.
  - *ipv4_addr* IPv4 host address. Data block format.
  - *ipv4_subnet* IPv4 subnet address. (CIDR notation). Data block format.
  - *ipv4_subnet* **detail** IPv4 subnet address. (CIDR notation). Data block format.
  - *ipv4_subnet* **longer-prefixes** IPv4 subnet address. (CIDR notation). Tabular format.
  - *ipv4_subnet* **longer-prefixes detail** IPv4 subnet address. (CIDR notation). Data block format.
- ***VRF_INSTANCE*** specifies VRF instances.
  - <no parameter> displays routing table for context-active VRF.
  - **vrf** *vrf_name* displays routing table for the specified VRF.
  - **vrf all** displays routing table for all VRFs.
  - **vrf default** displays routing table for default VRF.

Examples

- This command displays the BGP routing table.

```
switch>show ip bgp
BGP routing table information for VRF default
Router identifier 192.168.100.23, local AS number 64512
Route status codes: s - suppressed, * - valid, > - active, # - not installed, E
- ECMP head, e - ECMP
                    S - Stale, c - Contributing to ECMP, b - backup, L -
labeled-unicast
Origin codes: i - IGP, e - EGP, ? - incomplete
AS Path Attributes: Or-ID - Originator ID, C-LST - Cluster List, LL Nexthop -
Link Local Nexthop

         Network             Next Hop           Metric  LocPref Weight  Path
 * >Ec   121.1.110.0/24      192.168.31.3       0       100     0       64521 i
 *  ec   121.1.110.0/24      192.168.31.1       0       100     0       64521 i
 * >Ec   192.168.100.18/32   192.168.34.1       0       100     0       64524 i
 *  ec   192.168.100.18/32   192.168.34.3       0       100     0       64524 i
 * >     192.168.100.23/32   -                  0       0       -       i
 * >     192.168.100.31/32   192.168.113.1      0       100     0       64550 i
 * >     192.168.100.100/32  -                  0       0       -       i
 * >     192.168.121.0/24    192.168.113.1      0       100     0       64550 i
switch>
```

- This command displays the BGP routing table in the 10.17.48.0/23 network.

```
switch>show ip bgp 10.17.48.0/23
BGP routing table information for VRF default
Router identifier 192.168.100.23, local AS number 64512
BGP routing table entry for 10.17.48.0/23
 Paths: 2 available
  64521
    192.168.31.3 from 192.168.31.3 (192.168.100.12)
     Origin IGP, metric 0, localpref 100, weight 0, received 04:26:59 ago, valid,
external, ECMP head, ECMP, best, ECMP contributor
      Community: 1:10
      Rx SAFI: Unicast
  64521
    192.168.31.1 from 192.168.31.1 (192.168.100.11)
     Origin IGP, metric 0, localpref 100, weight 0, received 03:13:08 ago, valid,
external, ECMP, ECMP contributor
      Community: 2:10 3:20
      Rx SAFI: Unicast
tg299.19:30:29#
tg299.19:30:32#show ip bgp 121.1.114.0/24 detail
BGP routing table information for VRF default
Router identifier 192.168.100.23, local AS number 64512
Route status: [a.b.c.d] - Route is  queued for advertisement to peer.
BGP routing table entry for 121.1.114.0/24
 Paths: 2 available
  64521
    192.168.31.3 from 192.168.31.3 (192.168.100.12)
     Origin IGP, metric 0, localpref 100, weight 0, received 04:27:04 ago, valid,
external, ECMP head, ECMP, best, ECMP contributor
      Community: 1:10
      Rx SAFI: Unicast
  64521
    192.168.31.1 from 192.168.31.1 (192.168.100.11)
     Origin IGP, metric 0, localpref 100, weight 0, received 03:13:13 ago, valid,
external, ECMP, ECMP contributor
      Community: 2:10 3:20
      Rx SAFI: Unicast
      Not best: ECMP-Fast configured
 Advertised to 8 peers:
    192.168.31.1       192.168.32.1       192.168.32.3
    192.168.33.1       192.168.33.3       192.168.34.1
    192.168.34.3       192.168.113.1
switch>
```

- This command displays the BGP routing table for the 172.16.0.0/24 network, including the reasons why three routes were rejected by the best-path algorithm. The reason for rejection is preceded by the label **Not best:**

```
switch>#show ip bgp 172.16.0.0/24 detail
BGP routing table information for VRF default
Router identifier 192.168.100.18, local AS number 64524
Route status: [a.b.c.d] - Route is  queued for advertisement to peer.
BGP routing table entry for 204.1.47.220/30
 Paths: 4 available
  64512 64550 65100
    192.168.14.2 from 192.168.14.2 (192.168.100.21)
     Origin IGP, metric 0, localpref 100, weight 0, received 19:15:29 ago, valid,
external, ECMP head, ECMP, best, ECMP contributor
      Rx SAFI: Unicast
  64512 64550 65100
    192.168.24.2 from 192.168.24.2 (192.168.100.22)
     Origin IGP, metric 0, localpref 100, weight 0, received 19:15:29 ago, valid,
external, ECMP, ECMP contributor
      Rx SAFI: Unicast
      Not best: ECMP-Fast configured
  64512 64550 65100
    192.168.34.2 from 192.168.34.2 (192.168.100.23)
     Origin IGP, metric 0, localpref 100, weight 0, received 19:15:29 ago, valid,
external, ECMP, ECMP contributor
      Rx SAFI: Unicast
      Not best: Redistributed route exists
  64512 64550 65100
    192.168.44.2 from 192.168.44.2 (192.168.100.24)
     Origin IGP, metric 0, localpref 100, weight 0, received 19:15:29 ago, valid,
external, ECMP, ECMP contributor
      Rx SAFI: Unicast
      Not best: eBGP path preferred
Not advertised to any peer
   switch>
```

show ip bgp community

The **show ip bgp community** command displays Border Gateway Protocol (BGP) routing table entries, filtered by community.

Command Mode

EXEC

Command Syntax

show ip bgp community [*COMM_1 ... COMM_n*][*MATCH_TYPE*][*DATA_OPTION*][*VRF_INSTANCE*]

Parameters

- ***COMM_x*** community number or name, as specified in the route map that sets the community list number.
  - *aa:nn* AS and network number, separated by colon. Each value ranges from 1 to 4294967295.
  - *comm_num* community number. Values range from 1 to 4294967040.
  - **internet** advertises route to Internet community.
  - **local-as** advertises route only to local peers.
  - **no-advertise** does not advertise the route to any peer.
  - **no-export** advertises route only within BGP AS boundary.
- ***MATCH_TYPE*** Routes are filtered based on their communities.
  - <no parameter> routes must match at least one community in the list
  - **exact** route must match all communities and include no other communities.
- ***DATA_OPTION*** Type of information the command displays. Values include:
  - <no parameter> Displays table of the routing entry line items.
  - **detail** Displays data block for each routing table entry.
- ***VRF_INSTANCE*** specifies VRF instances.
  - <no parameter> displays routing table for context-active VRF.
  - **vrf** *vrf_name* displays routing table for the specified VRF.
  - **vrf all** displays routing table for all VRFs.
  - **vrf default** displays routing table for default VRF.

Example

- This command displays the BGP routing table entries for a specified community.

```
switch>show ip bgp community 65533:100 exact detail
BGP routing table information for VRF default
Router identifier 192.168.100.18, local AS number 64524
BGP routing table entry for 172.16.110.0/24
 Paths: 5 available
  64512 64521
    192.168.14.2 from 192.168.14.2 (192.168.100.21)
     Origin IGP, metric 0, localpref 100, weight 0, received 20d21h ago, valid,
external, ECMP head, ECMP, best, ECMP contributor
      Community: 65533:100
      Rx SAFI: Unicast
switch>
```

show ip bgp neighbors

The **show ip bgp neighbors** command displays Border Gateway Protocol (BGP) and TCP session data for a specified IPv4 BGP neighbor, or for all IPv4 BGP neighbors if an address is not included.

Command Mode

EXEC

Command Syntax

```
show ip bgp neighbors [NEIGHBOR_ADDR][VRF_INSTANCE]
```

Parameters

- ***NEIGHBOR_ADDR*** location of the neighbors. Options include:
  - <no parameter> command displays information for all IPv4 BGP neighbors.
  - *ipv4_addr* command displays information for specified neighbor.
- ***VRF_INSTANCE*** specifies VRF instances.
  - <no parameter> displays routing table for context-active VRF.
  - **vrf** *vrf_name* displays routing table for the specified VRF.
  - **vrf all** displays routing table for all VRFs.
  - **vrf default** displays routing table for default VRF.

Related Command

- **show ip bgp neighbors (route type)**
- **show ip bgp neighbors (route-type) community**

Example

- This command displays information for the neighbor at 10.0.2.6

```
switch>show ip bgp neighbors 10.0.2.6
BGP neighbor is 10.0.2.6, remote AS 64496, external link
Description: lf283-Ethernet3/5/3
  BGP version 4, remote router ID 172.16.100.13, VRF default
  Negotiated BGP version 4
  Last read 00:00:05, last write 00:00:05
  Hold time is 180, keepalive interval is 60 seconds
  Configured hold time is 180, keepalive interval is 60 seconds
  Connect timer is inactive
  Idle-restart timer is inactive
  BGP state is Established, up for 04:44:05
  Number of transitions to established: 11
  Last state was OpenConfirm
  Last event was RecvKeepAlive
  Last sent notification:Cease/administrative reset, Last time 04:44:09
 Last rcvd notification:Cease/peer de-configured, Last time   2d02h, First time
7d08h, Repeats 1
  Neighbor Capabilities:
    Multiprotocol IPv4 Unicast: advertised and received and negotiated
    Four Octet ASN: advertised and received
    Route Refresh: advertised and received and negotiated
    Send End-of-RIB messages: advertised and received and negotiated
    Additional-paths recv capability:
      IPv4 Unicast: advertised
    Additional-paths send capability:
      IPv4 Unicast: received
  Restart timer is inactive
  End of rib timer is inactive
  Message statistics:
    InQ depth is 0
    OutQ depth is 0
                         Sent      Rcvd
    Opens:                 11        12
    Notifications:          9         2
    Updates:              732        57
    Keepalives:         14451     14451
    Route-Refresh:          2         0
    Total messages:     15205     14522
  Prefix statistics:
                         Sent      Rcvd
    IPv4 Unicast:       12056        12
    IPv6 Unicast:           0         0
  Inbound updates dropped by reason:
    AS path loop detection: 0
    Enforced First AS: 0
    Malformed MPBGP routes: 0
    Originator ID matches local router ID: 0
    Nexthop matches local IP address: 0
    Unexpected IPv6 nexthop for IPv4 routes: 0
    Nexthop invalid for single hop eBGP: 0
  Inbound paths dropped by reason:
    IPv4 labeled-unicast NLRIs dropped due to excessive labels: 0
  Outbound paths dropped by reason:
    IPv4 local address not available: 0
    IPv6 local address not available: 0
Local AS is 64512, local router ID 192.168.100.23
```

```
TTL is 1
Local TCP address is 10.0.2.5, local port is 179
Remote TCP address is 10.0.2.6, remote port is 52206
Auto-Local-Addr is disabled
Bfd is enabled and state is Up
TCP Socket Information:
  TCP state is ESTABLISHED
  Recv-Q: 0/32768
  Send-Q: 0/32768
  Outgoing Maximum Segment Size (MSS): 8948
  Total Number of TCP retransmissions: 0
  Options:
    Timestamps enabled: yes
    Selective Acknowledgments enabled: yes
    Window Scale enabled: yes
    Explicit Congestion Notification (ECN) enabled: no
  Socket Statistics:
    Window Scale (wscale): 7,7
    Retransmission Timeout (rto): 204.0ms
    Round-trip Time (rtt/rtvar): 7.5ms/3.0ms
    Delayed Ack Timeout (ato): 40.0ms
    Congestion Window (cwnd): 10
    TCP Throughput: 95.45 Mbps
    Advertised Recv Window (rcv_space): 17896
switch>
```

show ip bgp neighbors (route type)

The **show ip bgp neighbors (route type)** command displays information for next hop routes to a specified IPv4 neighbor. The **show ip bgp neighbors (route-type) community** command displays the same information for routes filtered by communities.

The output format depends on the selected ***FILTER*** parameter:

- Data block format displays comprehensive information for each specified route.
- Tabular format displays routing table entries in tabular format for the specified IP addresses.

Commands that do not include a route type revert to the **show ip bgp neighbors** command.

Command Mode

EXEC

Command Syntax

```
show ip bgp neighbors neighbor_addr HOPDIRECT [FILTER] [VRF_INSTANCE]
show ip bgp neighbors neighbor_addr [ROUTE_TYPE] HOPDIRECT
show ip bgp neighbors neighbor_addr [ROUTE_TYPE] HOPDIRECT detail
```

Related Command

- **show ip bgp neighbors**
- **show ip bgp neighbors (route-type) community**

Parameters

- *neighbor_addr* location of the neighbor.
- ROUTE_TYPE filters route on route type. Options include:
  - **ipv4 unicast** displays IPv4 unicast routes.
  - **ipv6 unicast** displays IPv6 unicast routes.
- ***HOPDIRECT*** filters route on the basis of direction from neighbor. Options include:
  - **advertised-routes** displays routes advertised to the specified neighbor.
  - **received-routes** displays routes received from the specified neighbor (accepted and rejected).
  - **routes** displays routes received and accepted from specified neighbor.
- ***FILTER*** routing table entries that the command displays. Values include*:*
  - <no parameter> displays all routing table entries. Tabular format.
  - **detail** displays all routing table entries. Data block format.
  - *ipv4_addr* host IPv4 address. Data block format.
  - *ipv4_subnet* subnet address. (CIDR notation). Data block format.
  - *ipv4_subnet* **longer-prefixes** subnet address. (CIDR notation). Tabular format.
- ***VRF_INSTANCE*** specifies VRF instances.
  - <no parameter> displays routing table for context-active VRF.
  - **vrf** *vrf_name* displays routing table for the specified VRF.
  - **vrf all** displays routing table for all VRFs.
  - **vrf default** displays routing table for default VRF.

Example

- This command displays information for routes advertised to the neighbor at 10.17.254.78

```
switch>show ip bgp neighbors 10.17.254.78 advertised-routes
BGP routing table information for VRF default
Router identifier 192.168.100.18, local AS number 64524
Route status codes: s - suppressed, * - valid, > - active, # - not installed, E
- ECMP head, e - ECMP
                    S - Stale, c - Contributing to ECMP, b - backup, L -
labeled-unicast, q - Queued for advertisement
Origin codes: i - IGP, e - EGP, ? - incomplete
AS Path Attributes: Or-ID - Originator ID, C-LST - Cluster List, LL Nexthop -
Link Local Nexthop

        Network             Next Hop           Metric  LocPref Weight Path
 * >    121.1.140.0/24      192.168.44.3       -       -       -      64524 i
 * >    121.1.141.0/24      192.168.44.3       -       -       -      64524 i
 * >    121.1.142.0/24      192.168.44.3       -       -       -      64524 i
 * >    121.1.143.0/24      192.168.44.3       -       -       -      64524 i
 * >    121.1.144.0/24      192.168.44.3       -       -       -      64524 i
 * >    121.1.145.0/24      192.168.44.3       -       -       -        645
switch>
```

show ip bgp neighbors (route-type) community

The **show ip bgp neighbors (route type) community** command displays information for next hop routes to a specified neighbor. Routes are filtered by community.

The **show ip bgp neighbors (route type)** command displays the same information for routes filtered by IP addresses and subnets.

Command Mode

EXEC

Command Syntax

```
show ip bgp neighbors addr RTE community CM_1 [CM_2
..CM_n][MATCH][INFO][VRF_INST]
```

Related Command

- **show ip bgp neighbors**
- **show ip bgp neighbors (route type)**

Parameters

- *addr* Neighbor location (IPv4 address). Dotted decimal notation.
- ***RTE*** type of route that the command displays. Options include:
  - **advertised-routes** displays routes advertised to the specified neighbor.
  - **received-routes** displays routes received from the specified neighbor (accepted and rejected).
  - **routes** displays routes received and accepted from specified neighbor.
- ***CM_x*** community number or name, as specified in the route map that sets the community list number. The command must list at least one of the following community identifiers:.
  - *aa:nn* AS and network number, separated by colon. Each value ranges from 1 to 4294967295.
  - *comm_num* community number. Values range from 1 to 4294967040.
  - **internet** advertises route to Internet community.
  - **local-as** advertises route only to local peers.
  - **no-advertise** does not advertise route to any peer.
  - **no-export** advertises route only within BGP AS boundary.
- ***MATCH*** Routes are filtered based on their communities.
  - <no parameter> routes must match at least one community in the list
  - **exact** route must match all communities and include no other communities.
- ***INFO*** Type of information the command displays. Values include:
  - <no parameter> Displays table of routing entry line items.
  - **detail** Displays data block for each routing table entry.
- ***VRF_INST*** specifies VRF instances.
  - <no parameter> displays routing table for context-active VRF.
  - **vrf** *vrf_name* displays routing table for the specified VRF.
  - **vrf all** displays routing table for all VRFs.
  - **vrf default** displays routing table for default VRF.

show ip bgp neighbors regexp

The **show ip bgp neighbors regexp** command displays information for next hop routes to a specified IPv4 neighbor that match the AS path attributes specified in the given regular expression.

Command Mode

EXEC

Command Syntax

show ip bgp neighbors *addr* ***RTE regexp*** *as_paths* [***VRF_INST***]

Related Command

- **show ip bgp neighbors**
- **show ip bgp neighbors (route type)**

Parameters

- *addr* Neighbor location (IPv4 address). Dotted decimal notation.
- ***RTE*** type of route that the command displays. Options include:
  - **advertised-routes** displays routes advertised to the specified neighbor.
  - **received-routes** displays routes received from the specified neighbor (accepted and rejected).
  - **routes** displays routes received and accepted from specified neighbor.
- *as_paths* list of AS paths, formatted as a regular expression. Regular expressions are pattern matching strings that are composed of text characters and operators.
- ***VRF_INST*** specifies VRF instances.
  - <no parameter> displays routing table for context-active VRF.
  - **vrf** *vrf_name* displays routing table for the specified VRF.
  - **vrf all** displays routing table for all VRFs.
  - **vrf default** displays routing table for default VRF.

Example

- This command displays information for routes advertised to the neighbor at 10.14.4.4 which include AS number 64512 in their AS paths.

```
switch>show ip bgp neighbors 10.14.4.4 advertised-routes regexp _64512_
BGP routing table information for VRF default
Router identifier 192.168.100.18, local AS number 64524
Route status codes: s - suppressed, * - valid, > - active, # - not installed, E
- ECMP head, e - ECMP
                    S - Stale, c - Contributing to ECMP,  b - backup, L =
labeled-unicast, q - Queued for advertisement
Origin codes: i - IGP, e - EGP, ? - incomplete
AS Path Attributes: Or-ID - Originator ID, C-LST - Cluster List, LL Nexthop -
Link Local Nexthop

        Network           Next Hop       Metric  LocPref Weight  Path
 * >Ec  10.1.0.0/24       192.168.14.2   0       100     0       64512 64521 i
 *  ec  172.16.110.0/24   192.168.24.2   0       100     0       64512 64521 i
 *  ec  172.28.0.0/24     192.168.34.2   0       100     0       64512 64521 i

switch>
```

show ip bgp paths

The **show ip bgp paths** command displays all BGP paths in the database.

Command Mode

EXEC

Command Syntax

show ip bgp paths [*VRF_INSTANCE*]

Parameters

- ***VRF_INSTANCE*** specifies VRF instances.
  - <no parameter> displays routing table for context-active VRF.
  - **vrf** *vrf_name* displays routing table for the specified VRF.
  - **vrf all** displays routing table for all VRFs.
  - **vrf default** displays routing table for default VRF.

Display Values

- **Refcount**: Number of routes using a listed path.
- **Metric**: The path's Multi Exit Discriminator (MED).
- **Path**: The route's AS path and its origin code.

  The MED (the path's external metric) provides information to external neighbors about the preferred path into an AS that has multiple entry points. Lower MED values are preferred.

Example

- This command displays the BGP paths in the switch's database.

```
switch>show ip bgp paths
Refcount Metric     Path
59       0          i (HashID 1)
2        0          64512 ? (HashID 758)
4        0          64512 i (HashID 316)
4        0          64512 64550 i (HashID 676)
4        0          64512 64550 i (HashID 722)
10       0          64512 i (HashID 594)
24000    0          64512 64550 65100 i (HashID 724)
22       0          64512 64521 i (HashID 494)
24000    0          64512 64550 65100 i (HashID 723)
24000    0          64512 64550 65100 i (HashID 680)
24       0          64512 64522 i (HashID 764)
24       0          64512 64522 i (HashID 763)
2        0          64512 64523 i (HashID 738)
22       0          64512 64523 i (HashID 735)
2        0          64512 64523 i (HashID 733)
2        0          64512 64521 i (HashID 709)
switch>
```

show ip bgp peer-group

The **show ip bgp peer-group** command displays the BGP version, address family and group members for all BGP peer groups defined on the switch.

Command Mode

EXEC

Command Syntax

show ip bgp peer-group [*GROUP*][*VRF_INSTANCE*]

Parameters

- ***GROUP*** peer group for which command displays information. Options include:
  - <no parameter> command displays information for all peer groups.
  - *group_name* name of peer group for which command displays information.
- ***VRF_INSTANCE*** specifies VRF instances.
  - <no parameter> displays routing table for context-active VRF.
  - **vrf** *vrf_name* displays routing table for the specified VRF.
  - **vrf all** displays routing table for all VRFs.
  - **vrf default** displays routing table for default VRF.

Example

- This command displays BGP peer group information for all peer groups on the switch.

```
switch> show ip bgp peer-group
BGP peer-group is fabric, remote AS 65501, peer-group internal
  BGP version 4
  Static peer-group members:
    VRF default:
      192.168.17.0, state: Active
        Negotiated MP Capabilities:
            IPv4 Unicast: No
            IPv4 Multicast: No
            IPv4 MplsLabel: No
            IPv6 Unicast: No
            IPv6 MplsLabel: No
            L2VPN EVPN: No
      192.168.17.2, state: Active
        Negotiated MP Capabilities:
            IPv4 Unicast: No
            IPv4 Multicast: No
            IPv4 MplsLabel: No
            IPv6 Unicast: No
            IPv6 MplsLabel: No
            L2VPN EVPN: No
BGP peer-group is Evpn_RR, remote AS 65501, peer-group internal
  BGP version 4
  Static peer-group members:
    VRF default:
      192.168.0.9, state: Active
        Negotiated MP Capabilities:
            IPv4 Unicast: No
            IPv4 Multicast: No
            IPv4 MplsLabel: No
            IPv6 Unicast: No
            IPv6 MplsLabel: No
            L2VPN EVPN: No
      192.168.0.14, state: Active
        Negotiated MP Capabilities:
            IPv4 Unicast: No
            IPv4 Multicast: No
            IPv4 MplsLabel: No
            IPv6 Unicast: No
            IPv6 MplsLabel: No
            L2VPN EVPN: No
BGP peer-group is Mlag, remote AS 65501, peer-group internal
  BGP version 4
  Static peer-group members:
    VRF default:
      192.168.255.7, state: Established
        Negotiated MP Capabilities:
            IPv4 Unicast: Yes
            IPv4 Multicast: No
            IPv4 MplsLabel: No
            IPv6 Unicast: No
            IPv6 MplsLabel: No
            L2VPN EVPN: No
switch>
```

show ip bgp regexp

The **show ip bgp regexp** command displays Border Gateway Protocol (BGP) IPv4 routing table entries that match the AS path attributes specified in the given regular expression.

Command Mode

EXEC

Command Syntax

show ip bgp regexp *as_paths* **[*VRF_INSTANCE*]**

Parameters

- *as_paths* list of AS paths, formatted as a regular expression. Regular expressions are pattern matching strings that are composed of text characters and operators.
- ***VRF_INSTANCE*** specifies the VRF instance of the BGP routing table to be displayed.
  - <no parameter> displays routing table for context-active VRF.
  - **vrf** *vrf_name* displays routing table for the specified VRF.
  - **vrf all** displays routing table for all VRFs.
  - **vrf default** displays routing table for default VRF.

Examples

- This command displays information about the BGP IPv4 routes in the context-active VRF that pass through AS 64511.

```
switch>show ip bgp regexp _64511_
BGP routing table information for VRF default
Router identifier 10.254.81.1, local AS number 64496
Route status codes: s - suppressed, * - valid, > - active, E - ECMP head, e - ECMP
                    S - Stale
Origin codes: i - IGP, e - EGP, ? - incomplete
AS Path Attributes: Or-ID - Originator ID, C-LST - Cluster List, LL Nexthop -
Link Local Nexthop

      Network             Next Hop         Metric  LocPref Weight Path
 * >E 10.5.128.0/20       10.254.1.28      0       100     0      64510 64511 i
 *    10.5.128.0/20       10.254.1.0       0       100     0      64510 64511 i
 *    10.5.128.0/20       10.254.1.2       0       100     0      64510 64511 i
 *  e 10.5.128.0/20       10.254.1.4       0       100     0      64511
switch>
```

show ip bgp summary

The **show ip bgp summary** command displays BGP path, prefix, and attribute information for all BGP neighbors.

Command Mode

EXEC

Command Syntax

```
show ip bgp summary [VRF_INSTANCE]
```

Parameters

- ***VRF_INSTANCE*** specifies VRF instances.
  - <no parameter> displays routing table for context-active VRF.
  - **vrf** *vrf_name* displays routing table for the specified VRF.
  - **vrf all** displays routing table for all VRFs.
  - **vrf default** displays routing table for default VRF.

Display Values

Header Row

- **BGP router identifier**: The router identifier – loopback address or highest IP address.
- **Local AS Number**: AS number assigned to switch

Neighbor Table Columns

- **(First) Neighbor**: Neighbor's IP address.
- **(Second) V**: BGP version number.
- **(Third) AS**: Neighbor's AS number.
- **(Fourth) MsgRcvd:** Messages received from the neighbor.
- **(Fifth) MsgSent:** Messages sent to neighbor.
- **(Sixth) InQ:** Messages queued from neighbor.
- **(Seventh) OutQ:** Messages queued to send neighbor.
- **(Eighth) Up/Down:** Period the BGP session has been Established, or its current status.
- **(Ninth) State**: State of the BGP session and the number of routes received from a neighbor.

  After the maximum number of routes are received, the ninth field displays **PfxRcd**, and the connection becomes Idle. Maximum number of routes is set using the **maximum paths (BGP)** command.

Example

- This command displays the status of the switch's BGP connections.

```
switch>show ip bgp summary
BGP summary information for VRF default
Router identifier 0.0.2.1, local AS number 1
Neighbor Status Codes: m - Under maintenance
  Neighbor         V  AS           MsgRcvd   MsgSent  InQ OutQ  Up/Down State
PfxRcd PfxAcc
  1.0.1.2        4  2             395      424    0    0 00:16:37 Estab  5     5
  1.0.2.2        4  2             385      426    0    0 00:25:00 Estab  5     5
  1.0.3.2        4  2             392      420    0    0 00:25:02 Estab  5     5
  1.0.4.2        4  2             391      406    0    0 00:25:02 Estab  5     5
switch>
```

show ip community-list

The **show ip community-list** command displays the BGP community lists configured on the switch.

Command Mode

EXEC

Command Syntax

show ip community-list [*COMMUNITY_LIST*]

Parameters

- ***COMMUNITY_LIST*** community list for which command displays information.
  - <no parameter> command displays information for all community lists.
  - *listname* name of the community list (text string).

Example

- This command displays the BGP paths in the switch's database.

```
switch>show ip community-list hs-comm-list
ip community-list standard hs-comm-list permit 0:10
switch>
```

show ip extcommunity-list

The **show ip extcommunity-list** command displays the BGP extended community lists configured on the switch.

Command Mode

EXEC

Command Syntax

show ip extcommunity-list [*COMMUNITY_LIST*]

Parameters

- ***COMMUNITY_LIST*** extended community list for which command displays information.
  - <no parameter> command displays information for all extended community lists.
  - *listname* name of the extended community list (text string).

Example

- This command displays the extended extcommunity lists on the switch.

```
switch>show ip extcommunity-list
ip extcommunity-list standard hs-extcomm-list permit rt 3050:20
ip extcommunity-list standard hs-extcomm-list permit soo 172.17.52.2:30
ip extcommunity-list standard hs-extcomm-list permit rt 3050:70000
switch>
```

show ipv6 bgp

The **show ipv6 bgp** command displays IPv6 Border Gateway Protocol (BGP) routing table entries. The output format depends on the command parameters:

The output format depends on the command parameters:

- Data block format displays comprehensive information for each specified BGP routing table entry.
- Tabular format displays routing table entries in tabular format for the specified IP addresses.

Output produced by the ***longer-prefixes*** option includes the specified route and all more specific routes.

Command Mode

EXEC

Command Syntax

show ipv6 bgp [*FILTER*][*VRF_INSTANCE*]

Parameters

- ***FILTER*** routing table entries that the command displays. Values include*:*
  - <no parameter> displays all routing table entries. Tabular format.
  - **detail** displays all routing table entries. Data block format.
  - *ipv6_addr* IPv6 host address. Data block format.
  - *ipv6_prefix* IPv6 prefix address. (CIDR notation). Data block format.
  - *ipv6_prefix* **detail** IPv6 prefix address. (CIDR notation). Data block format.
  - *ipv6_prefix* **longer-prefixes** IPv6 prefix address. (CIDR notation). Tabular format.
  - *ipv6_prefix* **longer-prefixes detail** IPv6 prefix address. (CIDR notation). Data block format.
- ***VRF_INSTANCE*** specifies VRF instances.
  - <no parameter> displays routing table for context-active VRF.
  - **vrf** *vrf_name* displays routing table for the specified VRF.
  - **vrf all** displays routing table for all VRFs.
  - **vrf default** displays routing table for default VRF.

Examples

- This command displays the routing data blocks for a specified IPv6 prefix.

```
switch>show ipv6 bgp 2001:0DB8:5804:1134::/64 longer-prefixes
Route status codes: s - suppressed, * - valid, > - active, e - ECMP

     Network            Next Hop         R Metric  LocPref Path
* > 2001:0DB8:5804:1134::/64 -               u 0       0       ? (Id 1)
*   2001:0DB8:5804:1134::/64 2001:0DB8:5804:fe4c::1 u 0       100     (65533)
65534 i (Id 597)
```

- This command displays the routing table for a specified IPv6 prefix.

```
switch>show ipv6 bgp 2001:0DB8:5804:1134::/64
BGP routing table entry for 2001:0DB8:5804:1134::/64
 Paths: 2 available
  Local
    - from - (172.26.0.22)
      Origin INCOMPLETE, metric 0, localpref 0, valid, local, best
  (65533) 65534
2001:0DB8:5804:fe4c::1 from 2001:0DB8:5804:fe4c::1 (172.26.0.34)
      Origin IGP, metric 0, localpref 100, valid, external
switch>
```

show ipv6 bgp community

The **show ipv6 bgp community** command displays IPv6 Border Gateway Protocol (BGP) routing table entries, filtered by community.

Command Mode

EXEC

Command Syntax

```
show ipv6 bgp community [COMM_1 ... COMM_n][MATCH_TYPE][INFO][VRF_INSTANCE]
```

Parameters

- ***COMM_x*** community number or name, as specified in the route map that sets the community list number.
  - *aa:nn* AS and network number, separated by colon. Each value ranges from 1 to 4294967295.
  - *comm_num* community number. Values range from 1 to 4294967040.
  - **internet** advertises route to Internet community.
  - **local-as** advertises route only to local peers.
  - **no-advertise** does not advertise route to any peer.
  - **no-export** advertises route only within BGP AS boundary.
- ***MATCH_TYPE*** Routes are filtered based on their communities.
  - <no parameter> routes must match at least one community in the list
  - **exact** route must match all communities and include no other communities.
- ***INFO*** Type of information the command displays. Values include:
  - <no parameter> Displays table of the routing entry line items.
  - **detail** Displays data block for each routing table entry.
- ***VRF_INSTANCE*** specifies VRF instances.
  - <no parameter> displays routing table for context-active VRF.
  - **vrf** *vrf_name* displays routing table for the specified VRF.
  - **vrf all** displays routing table for all VRFs.
  - **vrf default** displays routing table for default VRF.

show ipv6 bgp neighbors

The **show ipv6 bgp neighbors** command displays IPv6 Border Gateway Protocol (BGP) and TCP session data for a specified neighbor. Command displays data for all neighbors if an address is not included.

Command Mode

EXEC

Command Syntax

show ipv6 bgp neighbor [*NEIGHBOR_ADDR*][*VRF_INSTANCE*]

Parameters

- ***NEIGHBOR_ADDR*** location of the neighbors. Options include:
  - <no parameter> command displays information for all neighbors.
  - *ipv6_addr* command displays information for specified neighbor.
- ***VRF_INSTANCE*** specifies VRF instances.
  - <no parameter> displays routing table for context-active VRF.
  - **vrf** *vrf_name* displays routing table for the specified VRF.
  - **vrf all** displays routing table for all VRFs.
  - **vrf default** displays routing table for default VRF.

Example

- This command displays information for the neighbor at 2001:0DB8:52a4:fe01::2

```
switch>show ipv6 bgp neighbors 2001:0DB8:52a4:fe01::2
BGP neighbor is 2001:0DB8:52a4:fe01::2, remote AS 65536
Description: v6-bgp-to-magensium
BGP version is 4, remote router ID 172.26.0.23
Negotiated version is 4
TTL is 0
holdtime is 180
restart-time is 0
Restarting: no
Current state is Established
Updates received: 256
Updates sent: 4787
Total messages received: 11097
Total messages sent: 15250
Last state was OpenConfirm
Last event was RecvKeepAlive
Last error code was 0
Last error subcode was 0
Established time: 652492 seconds
Number of transitions to established: 1
Local TCP address is 2001:0DB8:52a4:fe01::1
Local AS is 65533
Local router ID is 172.26.0.22
Capabilities                      Snt   Rcv   Neg
-----------------------------------------------
Multiprotocol IPv4 Unicast        yes   yes   yes
Graceful Restart IPv4 Unicast     no    no    no
Multiprotocol IPv4 Multicast      no    no    no
Graceful Restart IPv4 Multicast   no    no    no
Multiprotocol IPv6 Unicast        yes   yes   yes
Graceful Restart IPv6 Unicast     no    no    no
Multiprotocol IPv4 VPN            no    no    no
Graceful Restart IPv4 VPN         no    no    no
Route Refresh                     yes   yes   yes
Send End-of-RIB messages          no    no    no
Dynamic Capabilities              no    no    no
```

show ipv6 bgp neighbors (route type)

The **show ipv6 bgp neighbors (route type)** command displays information for next hop routes to a specified IPv6 BGP neighbor. The **show ipv6 bgp neighbors (route type) community** command displays the same information for routes filtered by communities. Commands that do not include a route type revert to the **show ipv6 bgp neighbors** command.

The output format depends on the selected ***FILTER*** parameter:

- Data block format displays comprehensive information for each specified route.
- Tabular format displays routing table entries in tabular format for the specified IP addresses.

Output produced by the ***longer-prefixes*** option includes the specified route and all more specific routes.

Command Mode

EXEC

Command Syntax

show ipv6 bgp neighbors *neighbor_addr* ***ROUTE_TYPE*** [***FILTER***] [***VRF_INSTANCE***]

Parameters

- *neighbor_addr* location of the neighbor.
- ***ROUTE_TYPE*** type of route that the command displays. Options include:
  - **advertised-routes** displays routes advertised to the specified neighbor.
  - **received-routes** displays routes received from the specified neighbor (accepted and rejected).
  - **routes** displays routes received and accepted from specified neighbor.
- ***FILTER*** routing table entries that the command displays. Options include*:*
  - <no parameter> displays all routing table entries. Tabular format.
  - **detail** displays all routing table entries. Data block format.
  - *ipv6_addr* IPv6 host address. Data block format.
  - *ipv6_prefix* IPv6 prefix address (CIDR notation). Data block format.
  - *ipv6_prefix* **longer-prefixes** IPv6 prefix address. (CIDR notation). Tabular format.
- ***VRF_INSTANCE*** specifies VRF instances.
  - <no parameter> displays routing table for context-active VRF.
  - **vrf** *vrf_name* displays routing table for the specified VRF.
  - **vrf all** displays routing table for all VRFs.
  - **vrf default** displays routing table for default VRF.

Example

- This command displays information for routes advertised to the neighbor at 2001:0DB8:52a4:1::/64

```
switch>show ipv6 bgp neighbors 2001:0DB8:52a4:fe01::2 routes 2001:0DB8:52a4::/48
longer-prefixes
Route status codes: s - suppressed, * - valid, > - active, e - ECMP

     Network             Next Hop         R Metric  LocPref Path
* 2001:0DB8:52a4:1::/64 2001:0DB8:52a4:fe61::2 u 0       100     (65533) ? (Id
7) Rt-ID: 172.26.0.23
* 2001:0DB8:52a4:1001::/64 2001:0DB8:52a4:fe61::2 u 0      100     (65533) ? (Id
7) Rt-ID: 172.26.0.23
*    2001:0DB8:52a4:1a00::23/128 2001:0DB8:52a4:fe50::2 u 0      100     (65533)
? (Id 11) Rt-ID: 172.26.0.23
* >  2001:0DB8:52a4:fe70::/64 2001:0DB8:52a4:fe50::2 u 0       100     (65533)
65534 i (Id 59) Rt-ID: 172.26.0.23
*    2001:0DB8:52a4:fee4::/62 2001:0DB8:52a4:fe08::3 u 0      100     (65533) ?
(Id 24) Rt-ID: 172.26.0.23
switch>
```

show ipv6 bgp neighbors (route type) community

The **show ipv6 bgp neighbors (route type) community** command displays information for next hop routes to a specified IPv6 BGP neighbor. Routes are filtered by community.

The **show ipv6 bgp neighbors (route type)** command displays the same information for routes filtered by IP addresses and prefixes.

Command Mode

EXEC

Command Syntax

show ipv6 bgp neighbors *adr* ***RTE*** **community** ***CM_1***
[*CM_2..CM_n*][*MATCH*][*INFO*][*VRF_INST*]

Parameters

- *adr* Neighbor location (IPv6 address). Dotted decimal notation.
- ***RTE*** type of route that the command displays. Options include:
  - **advertised-routes** displays routes advertised to the specified neighbor.
  - **received-routes** displays routes received from the specified neighbor (accepted and rejected).
  - **routes** displays routes received and accepted from specified neighbor.
- ***CM_x*** community number or name, as specified in the route map that sets the community list number. The command must list at least one of the following community identifiers:.
  - *aa:nn* AS and network number, separated by colon. Each value ranges from 1 to 4294967295.
  - *comm_num* community number. Values range from 1 to 4294967040.
  - **internet** advertises route to Internet community.
  - **local-as** advertises route only to local peers.
  - **no-advertise** does not advertise route to any peer.
  - **no-export** advertises route only within BGP AS boundary.
- ***MATCH*** Routes are filtered based on their communities.
  - <no parameter> routes must match at least one community in the list
  - **exact** route must match all communities and include no other communities.
- ***INFO*** Type of information the command displays. Values include:
  - <no parameter> Displays table of the routing entry line items.
  - **detail** Displays data block for each routing table entry.
- ***VRF_INST*** specifies VRF instances.
  - <no parameter> displays routing table for context-active VRF.
  - **vrf** *vrf_name* displays routing table for the specified VRF.
  - **vrf all** displays routing table for all VRFs.
  - **vrf default** displays routing table for default VRF.

Example

- This command displays the BGP routes in the 2001:0DB8:523c:fe4c::1 network that are assigned the community of 65533:100.

```
switch>show ipv6 bgp neighbors 2001:0DB8:523c:fe4c::1 advertised-routes
community 65533:100
Route status codes: s - suppressed, * - valid, > - active, e - ECMP

     Network            Next Hop         R Metric  LocPref Path
* >  10.17.254.0/30     10.17.254.1     u 1       0       65533 i (Id 0)
switch>
```

show ipv6 bgp neighbors regexp

The **show ipv6 bgp neighbors regexp** command displays information for next hop routes to a specified IPv6 neighbor that match the AS path attributes specified in the given regular expression.

Command Mode

EXEC

Command Syntax

show ipv6 bgp neighbors *addr* ***RTE regexp*** *as_paths* [***VRF_INST***]

Related Command

- **show ip bgp neighbors**
- **show ip bgp neighbors (route type)**

Parameters

- *addr* Neighbor location (IPv6 address).
- ***RTE*** type of route that the command displays. Options include:
  - **advertised-routes** displays routes advertised to the specified neighbor.
  - **received-routes** displays routes received from the specified neighbor (accepted and rejected).
  - **routes** displays routes received and accepted from specified neighbor.
- *as_paths* list of AS paths, formatted as a regular expression. Regular expressions are pattern matching strings that are composed of text characters and operators.
- ***VRF_INST*** specifies VRF instances.
  - <no parameter> displays routing table for context-active VRF.
  - **vrf** *vrf_name* displays routing table for the specified VRF.
  - **vrf all** displays routing table for all VRFs.
  - **vrf default** displays routing table for default VRF.

Examples

- This command displays information for routes advertised to the neighbor at 2001:0DB8:254:177::15 which include AS number 64502 in their AS paths.

```
switch>show ipv6 bgp neighbors 2001:0DB8:254:177::15 received-routes
BGP routing table information for VRF default
Router identifier 10.254.83.8, local AS number 64496
Route status codes: s - suppressed, * - valid, > - active, E - ECMP head, e - ECMP
                    S - Stale
Origin codes: i - IGP, e - EGP, ? - incomplete
AS Path Attributes: Or-ID - Originator ID, C-LST - Cluster List, LL Nexthop -
Link Local Nexthop

      Network             Next Hop         Metric  LocPref Weight Path
*     2001:0DB8:52a4:1::/64 2001:0DB8:52a4:fe61::2 u 0       100     (64502)
*     2001:0DB8:52a4:1001::/64 2001:0DB8:52a4:fe61::2 u 0       100     (64502)
*     2001:0DB8:52a4:1a00::23/128 2001:0DB8:52a4:fe50::2 u
0       100     (64502)
*     2001:0DB8:52a4:fe70::/64 2001:0DB8:52a4:fe50::2 u 0       100     (64502)
64511 i
*     2001:0DB8:52a4:fee4::/62 2001:0DB8:52a4:fe08::3 u 0       100     (64502)
```

show ipv6 bgp regexp

The **show ipv6 bgp regexp** command displays Border Gateway Protocol (BGP) IPv6 routing table entries that match the AS path attributes specified in the given regular expression.

Command Mode

EXEC

Command Syntax

show ipv6 bgp regexp *as_paths* [***VRF_INSTANCE***]

Parameters

- *as_paths* list of AS paths, formatted as a regular expression. Regular expressions are pattern matching strings that are composed of text characters and operators.
- ***VRF_INSTANCE*** specifies the VRF instance of the BGP routing table to be displayed.
  - <no parameter> displays routing table for context-active VRF.
  - **vrf** *vrf_name* displays routing table for the specified VRF.
  - **vrf all** displays routing table for all VRFs.
  - **vrf default** displays routing table for default VRF.

Examples

- This command displays information about the BGP IPv6 routes in the context-active VRF that pass through AS 64511.

```
switch>show ipv6 bgp regexp _64511_
BGP routing table information for VRF default
Router identifier 22.0.0.3, local AS number 64496
Route status codes: s - suppressed, * - valid, > - active, E - ECMP head, e - ECMP
                    S - Stale
Origin codes: i - IGP, e - EGP, ? - incomplete
AS Path Attributes: Or-ID - Originator ID, C-LST - Cluster List, LL Nexthop -
Link Local Nexthop

      Network             Next Hop         Metric  LocPref Weight Path
* >  2001:0DB8:0:1::/64     2001:0DB8::3       0       100     0      64511 i
* >  2001:0DB8:0:2::/64     2001:0DB8::3       0       100     0      64511 i
* >  2001:0DB8:0:3::/64     2001:0DB8::3       0       100     0      64511 i
* >  2001:0DB8:0:4::/64     2001:0DB8::3       0       100     0      64511 i
* >  2001:0DB8:0:5::/64     2001:0DB8::3       0       100     0      64511 i
* >  2001:0DB8:0:6::/64     2001:0DB8::3       0       100     0      64511 i
* >  2001:0DB8:0:7::/64     2001:0DB8::3       0       100     0      64511 i
* >  2001:0DB8:0:8::/64     2001:0DB8::3       0       100     0      64511 i
* >  2001:0DB8:0:9::/64     2001:0DB8::3       0       100     0      64511 i
```

show ipv6 bgp summary

The **show ipv6 bgp summary** command displays BGP path, prefix, and attribute information for all BGP IPv6 neighbors.

Command Mode

EXEC

Command Syntax

show ipv6 bgp summary [*VRF_INSTANCE*]

Parameters

- ***VRF_INSTANCE*** specifies VRF instances.
  - <no parameter> displays routing table for context-active VRF.
  - **vrf** *vrf_name* displays routing table for the specified VRF.
  - **vrf all** displays routing table for all VRFs.
  - **vrf default** displays routing table for default VRF.

Display Values

Header Row

- **BGP router identifier**: The router identifier: loopback address or highest IP address.
- **Local AS Number**: AS number assigned to switch

Neighbor Table Columns

- **(First) Neighbor**: Neighbor's IP address.
- **(Second) V**: BGP version number.
- **(Third) AS**: Neighbor's AS number.
- **(Fourth) MsgRcvd:** Messages received from the neighbor.
- **(Fifth) MsgSent:** Messages sent to neighbor.
- **(Sixth) InQ:** Messages queued from neighbor.
- **(Seventh) OutQ:** Messages queued to send neighbor.
- **(Eighth) Up/Down:** Period the BGP session has been Established, or its current status.
- **(Ninth) State**: State of the BGP session and the number of routes received from a neighbor.

  After the maximum number of routes are received, the ninth field displays **PfxRcd**, and the connection becomes Idle. Maximum number of routes is set using the **maximum paths (BGP)** command.

Example

- This command displays the status of the switch's BGP connections.

```
switch>show ipv6 bgp summary
BGP router identifier 10.26.0.22, local AS number 65533
Neighbor         V  AS      MsgRcvd   MsgSent  InQ OutQ  Up/Down State  PfxRcd
2001:0DB8:52a4:fe4c::1 4  65534      6030      6029    0    0    2d13h Estab  8
2001:0DB8:52a4:fe01::2 4  65533      6212      6294    0    0    3d08h Estab  818
switch>
```

shutdown (BGP)

The **shutdown** command disables BGP on the switch without modifying the BGP configuration.

The **no shutdown** and **default shutdown** commands enable the BGP instance by removing the **shutdown** command from ***running-config***.

Command Mode

Router-BGP Configuration

Command Syntax

```
shutdown
no shutdown
default shutdown
```

Examples

- This command disables BGP on the switch.

```
switch(config)#router bgp 9
switch(config-router-bgp)#shutdown
switch(config-router-bgp)#
```

- This command enables BGP on the switch.

```
switch(config)#router bgp 9
switch(config-router-bgp)#no shutdown
switch(config-router-bgp)#
```

timers bgp

The **timers bgp** command configures the BGP keepalive and hold times.Timer settings apply to each peer connection. The **neighbor timers** command configures the times on a specified peer connection.

- Keepalive time: period between the transmission of consecutive keepalive messages.
- Hold time: period the switch waits for a keepalive or UPDATE message before it disables peering.

The hold time must be at least 3 seconds and should be three times longer than the keepalive setting.

The **no timers bgp** and **default timers bgp** commands return the time settings to their default values by removing the **timers bgp** command from ***running-config***.

- keepalive: 60 seconds
- hold time: 180 seconds

Command Mode

Router-BGP Configuration

Command Syntax

```
timers bgp keep_alive hold_time
no timers bgp
default timers bgp
```

Parameters

- *keep_alive* keepalive period, in seconds. Values include
  - **0** keepalive messages are not sent
  - **1** to **3600** keepalive time (seconds).
- *hold_time* hold time. Values include
  - **0** peering is not disabled by timeout expiry; keepalive packets are not sent.
  - **3** to **7200** hold time (seconds).

Example

- This command sets the keepalive time to 30 seconds and the hold time to 90 seconds.

```
switch(config)#router bgp 9
switch(config-router-bgp)#timers bgp 30 90
switch(config-router-bgp)#
```

vrf

The **vrf** command places the switch in BGP VRF configuration mode for the specified VRF. Commands issued in this mode will override global BGP configuration for the specified VRF.

Command Mode

Router-BGP Configuration

Command Syntax

vrf *vrf_instance*

Parameters

- *vrf_instance* VRF to be configured.

Example

- These commands place the switch in BGP VRF configuration mode for VRF “purple.”

```
switch(config)#router bgp 9
switch(config-router-bgp)#vrf purple
switch(config-router-bgp-vrf-purple)#
```

Chapter 30

Routing Information Protocol

This chapter contains the following sections.

- Section 30.1: RIP Conceptual Overview
- Section 30.2: Running RIP on the Switch
- Section 30.3: Configuring RIP on Multiple VRFs
- Section 30.4: RIP Commands

30.1 RIP Conceptual Overview

Routing Information Protocol (RIP) is a routing protocol typically used as an interior gateway protocol (IGP). RIP uses hop counts only to determine the shortest path to a destination. To avoid loops, RIP limits its paths to a maximum of 15 hops, making it an ineffective protocol for large networks. RIP Version 2 supports Classless Inter-Domain Routing (CIDR) and uses IP multicasting at address 224.0.0.9 to share the routing table with adjacent routers.

RIP sends updates whenever there is a change in the network topology and periodic updates when there are no changes. Receiving switches update their routing table whenever the update includes topology changes. Because RIP transmits the entire routing table every 30 seconds, RIP updates can generate heavy traffic loads in large or complicated networks.

Each switch also sends a list of distance-vectors to each of its neighbors periodically. The distance-vector is the metric RIP uses to express the cost of a route, and it describes the number of hops required to reach a destination. Each hop is typically assigned a hop count value of 1, and the router adds 1 to the metric when it receives a routing update and adds the network to its routing table.

To remove dead routes from its routing table, RIP marks a route for deletion if the router does not receive an advertisement for it within the expiration interval, then removes it from the routing table after the deletion interval.

30.2 Running RIP on the Switch

30.2.1 Accessing RIP Configuration Mode and Enabling RIP

30.2.1.1 RIP Configuration Mode

The **router rip** command places the switch in router-RIP configuration mode to configure the Routing Information Protocol (RIP) routing.

Example

- This command places the switch in router-rip configuration mode.

```
switch(config)#router rip
switch(config-router-rip)#
```

Using the **router rip** command puts the switch in router-RIP configuration mode, but does not enable RIP on the switch.

30.2.1.2 Enabling RIP

Routing Information Protocol (RIP) is disabled on the switch by default. The **no shutdown (RIP)** command in router-RIP configuration mode will enable RIP.

Example

- This command enables RIP on the switch.

```
switch(config-router-rip)#no shutdown
switch(config-router-rip)#
```

Issuing this command enables RIP, but to send and receive RIP route updates and to route packets via RIP you must also specify interfaces on which RIP will run by using the **network (RIP)** command.

30.2.1.3 Disabling RIP

You can disable RIP in two ways. The **shutdown (RIP)** command disables RIP on the switch but maintains all user-entered router-RIP configuration statements in the ***running-config***. The **no router rip** command disables RIP and removes all user-entered router-RIP configuration statements from the ***running-config***.

Examples

- This command disables RIP on the switch and removes all user-entered router-RIP configuration.

```
switch(config)#no router rip
switch(config)#
```

- This command disables RIP on the switch, but preserves all user-entered router-RIP configuration.

```
switch(config-router-rip)#shutdown
switch(config-router-rip)#
```

30.2.2 Configuring RIP

Issuing the **no shutdown (RIP)** command in router-RIP configuration mode enables RIP, but to run RIP on an interface you must specify a RIP network by using the **network (RIP)** command.

You can also configure the redistribution of routes learned from other protocols, set the default metric and administrative distance for redistributed routes, configure the timing of various RIP events, and configure specific interfaces to send RIP update packets by broadcast instead of multicast.

30.2.2.1 Specifying RIP Networks

The **network (RIP)** command identifies networks on which RIP will run and also specifies which routes RIP will accept into its routing table. You can issue the **network (RIP)** command multiple times to build up a list of RIP networks. No RIP networks are configured by default, so in order to route packets and send and receive RIP updates you must specify one or more RIP networks.

To disable RIP on a specific network, use the **no network (RIP)** command.

Examples

- This command enables RIP on 10.168.1.1/24

```
switch(config-router-rip)#network 10.168.1.1/24
switch(config-router-rip)#
```

- This command disables RIP on 10.168.1.1/24

```
switch(config-router-rip)#no network 10.168.1.1/24
switch(config-router-rip)#
```

30.2.2.2 Redistributing Routes Learned from Other Protocols into RIP

To enable route import from a specified protocol into RIP, use the **redistribute (RIP)** command. Additionally, you can apply a route map to the incoming routes to filter which routes are added to the RIP routing table. All connected routes are redistributed into RIP by default.

Example

- This command redistributes all routes learned from OSPF into RIP.

```
switch(config-router-rip)#redistribute OSPF
switch(config-router-rip)#
```

30.2.2.3 Configuring RIP Timers

When RIP is running on the switch, it sends unsolicited route updates and deletes expired routes at regular intervals. To configure the timing of those events, use the **timers basic (RIP)** command. The command takes three parameters: the update interval, the route expiration time, and the route deletion time.

The update interval is the amount of time in seconds that the switch waits between sending unsolicited RIP route updates to its neighbors. The route expiration time is how long the switch waits before marking an unadvertised route for deletion (the counter resets whenever an advertisement for the route is received). And the route deletion time is how long the switch waits between marking a route for deletion and removing it from the routing table. During the deletion interval, the switch continues to forward packets on the route.

Example

- This command sets the update interval to 60 seconds, expiration time to 90 seconds, and deletion time to 150 seconds.

```
switch(config-router-rip)#timers basic 60 90 150
switch(config-router-rip)#
```

30.2.2.4 Configuring an Interface to Transmit Broadcast RIP Updates

By default, the switch uses RIP version 2 and multicasts RIP update packets from all participating interfaces. To reconfigure a specific interface to send updates as broadcast packets, use the **ip rip v2-broadcast** command in the configuration mode for the interface.

Example

- The following commands configure RIP version 2 broadcasting on interface Ethernet 5.

```
switch(config)#interface ethernet5
switch(config-if-Et5)#ip rip v2-broadcast
switch(config-if-Et5)#exit
switch(config)#
```

30.2.3 Displaying RIP Information

30.2.3.1 Displaying RIP Routes

To see a listing of the RIP routes in the switch's routing table, use the **show ip rip database** command. (You can also display similar information using the RIP option in the **show ip route** command)

Examples

- This command displays all active rip routes.

```
switch>show ip rip database
10.168.11.0/24 directly connected, Et4
10.168.13.0/24
[1] via 10.168.14.2, 00:00:25, Et4
[2] via 10.168.15.2, 00:00:20, Et1
10.168.13.0/24
[1] via 10.168.14.2, 00:00:25, Et3
```

- This command submits a query for RIP route information for a network..

```
switch>show ip rip database 10.168.13.0/16
10.168.13.0/24
[1] via 10.168.14.2, 00:00:25, Et4
[2] via 10.168.15.2, 00:00:20, Et1
```

30.2.3.2 Displaying RIP Route Gateways

To see information about the switch's RIP route gateways, use the **show ip rip neighbors** command. The output displays the IPv4 address, the last heard time of the gateway, and characteristic flags applying to the gateway.

Example

- This command displays information about all the gateways of RIP routes..

```
switch>show ip rip neighbors
Gateway     Last-Heard      Bad-Packets     Bad-Routes    Flags
10.2.12.33  00:00:15                                      SRC, TRSTED,
                                                          ACCPTED, RJCTED,
                                                          Q_RJCTED, AUTHFAIL
```

30.3 Configuring RIP on Multiple VRFs

VRF support for Routing Information Protocol (RIP) allows instances of RIP on multiple non-default VRFs on the same router. By default, all interfaces belong to the default VRF until VRF forwarding is executed.

The **vrf definition** and **vrf forwarding** commands configure a non-default VRF, enable routing in it, and configure the network command under the configuration router RIP for the prefix to which the interface belongs.

The **router rip vrf** command places the switch in router-RIP configuration mode to configure an RIP routing instance in the non-default VRF.

Examples

- This command configures a non-default VRF and enables unicast routing in it.

```
switch(config)#vrf definition test
switch(config-vrf-test)#rd 1:1
switch(config-vrf-test)#exit
switch(config)#ip routing vrf test
switch(config)#
```

- This command configures a RIP instance in the non-default VRF.

```
switch(config)#router rip vrf test
switch(config-router-rip-router-rip-vrf-test)#no shutdown
switch(config-router-rip)#exit
switch(config)#
```

- This command configures an interface as part of a non-default VRF, by configuring the network command under the configuration router RIP for the prefix to which the interface belongs.

```
switch(config)#interface Ethernet 3 / 1
switch(config-if-Et3/1)#no switchport
switch(config-if-Et3/1)#ip address 1.0.0.1/24
switch(config-if-Et3/1)#vrf forwarding test
switch(config-if-Et3/1)#network 1.0.0.1
switch(config-if-Et3/1)#exit
switch(config)#
```

30.4 RIP Commands

Global Configuration Commands

- router rip
- router rip vrf

Interface Configuration Commands

- ip rip v2-broadcast

Router-RIP Configuration Mode

- default-metric
- distance (RIP)
- distribute-list (RIP)
- network (RIP)
- redistribute (RIP)
- shutdown (RIP)
- timers basic (RIP)

VRF Configuration Mode

- vrf definition
- vrf forwarding

Display Commands – EXEC Mode

- show ip rip database
- show ip rip neighbors

default-metric

The **default-metric** command specifies the metric value assigned to RIP routes learned from other protocols. All routes imported into RIP receive the default metric unless a matching route-map exists for the route. The route metric of 0 is assigned to redistributed connected and static routes. Default-metric values range from 0 to 16 with a default value of 1.

The **no default-metric** and **default default-metric** commands remove the **default-metric** command from ***running-config*** and returns the default-metric value to its default value of 1.

Command Mode

Router-RIP Configuration

Command Syntax

```
default-metric metric_value
no default-metric
default default-metric
```

Parameters

- *metric_value* default metric value assigned. Values range from 0 to 16; default is 1.

Example

- This command sets the default metric value to five.

```
switch(config)#router rip
switch(config-router-rip)#default-metric 5
switch(config-router-rip)#
```

distance (RIP)

The **distance** command assigns an administrative distance to routes that the switch learns through RIP. Routers use administrative distances to select a route when two protocols provide routing information to the same destination. Distance values range from 1 to 255; lower distance values correspond to higher reliability. The default RIP distance value is 120.

The **no distance** and **default distance** commands restore the administrative distance default value of 120 by removing the **distance** command from ***running-config***.

Command Mode

Router-RIP Configuration

Command Syntax

```
distance distance_value
no distance
default distance
```

Parameters

- *distance_value* distance assigned to RIP routes. Values range from 1 to 255.

Examples

- These commands assign an administrative distance of 75 to RIP routes.

```
switch(config)#router rip
switch(config-router-rip)#distance 75
switch(config-router-rip)#
```

distribute-list (RIP)

The **distribute-list** command allows users to filter out routes that are received or sent out. The distribute-list command influences which routes the router installs into its routing table and advertises to its neighbors.

Configuration Notes:

- Only one inbound distribute-list is allowed per interface.
- Only one outbound distribute-list is allowed per interface.
- Only one globally-defined inbound distribute-list is allowed.
- Only one globally-defined outbound distribute-list is allowed.
- Not all match clauses in a route-map are supported using RIP routes filtering. These match clauses for distribute-lists are supported:
  - match ip address access-list
  - match ip address prefix-list
- The **distribute-list** command does not enforce the specified route-map to contain only supported match clauses.
- Permit or deny can be specified in both prefix/access list and route-map configurations. The following rules apply when filtering routes:
  - Routes permitted by the prefix/access lists are treated as matched.
  - Matched routes are filtered based on the permit or deny option configured for the route-map clause.
  - Unmatched routes are further evaluated by the next route-map clause.
  - If a route does not match any clause in a route-map, it is denied.
  - If the route-map given in the **distribute-list** command is not configured, then all routes are permitted.
  - When multiple inbound (or outbound) distribute-lists are configured, only the most specific one is applied.

The **no distribute-list** and **default distribute-list** commands remove the corresponding **distribute-list** command from ***running-config***.

Command Mode

Router-RIP Configuration

Command Syntax

```
distribute-list DIRECTION MAP [INTF]
no distribute-list DIRECTION MAP [INTF]
default distribute-list DIRECTION MAP [INTF]
```

Parameters

- ***DIRECTION*** direction specifies if distribute-list is applied on inbound or outbound traffic. Valid options include:
  - **in** specifies inbound as the direction the distribute-list is applied.
  - **out** specifies outbound as the direction the distribute-list is applied.
- ***MAP*** specifies route map that assigns attribute values to the network. Options include:
  - <no parameter> attributes are not assigned through a route map.
  - **route-map** *map_name* attributes listed by specified route map are assigned to the network.

- ***INTF*** interface to be configured. Options include:
  - **ethernet** *e_num* Ethernet interface.
  - **loopback** *l_num* Loopback interface.
  - **port-channel** *p_num* Port channel interface.
  - **vlan** *v_num* VLAN interface.

Example

- The folllowing commands demonstrate that an access-list or prefix-list can be used within a route-map for use in a distribute-list.

```
switch(config)#ip prefix-list 8to24 seq 5 permit 0.0.0.0/0 ge 8 le 24
switch(config)#route-map myRouteMap permit 10
switch(config-route-map-myRouteMap)#match ip address prefix-list 8to24
switch(config-route-map-myRouteMap)#exit
switch(config)#
switch(config)#router rip
switch(config-router-rip)#distribute-list in route-map myRouteMap
switch(config-router-rip)#
```

- These commands suppress routes advertised on a particular interface.

```
switch(config)#ip prefix-list 2 seq 10 deny 30.1.1.0/24
switch(config)#route-map myRmOut permit 10
switch(config-route-map-myRmOut)#match ip address prefix-list 2
switch(config-route-map-myRouteMap)#exit
switch(config)#router rip
switch(config-router-rip)# distribute-list out route-map myRmOut
```

ip rip v2-broadcast

The **ip rip v2-broadcast** command specifies the transmission of Routing Information Protocol (RIP) Version 2 update packets from the configuration mode interface as broadcast to 255.255.255.255.

The **no rip v2-broadcast** and **default rip v2-broadcast commands** specify the transmission of update packets as multicast to 224.0.0.9 if the configuration mode interface is multicast capable. Updates are broadcast if the interface is not multicast capable.

Command Mode

Interface-Ethernet Configuration
Interface-Port-Channel Configuration
Interface-VLAN Configuration

Command Syntax

```
ip rip v2-broadcast
no ip rip v2-broadcast
default ip rip v2-broadcast
```

Example

- The following example configures version 2 broadcasting on interface Ethernet 5.

```
switch(config)#interface ethernet 5
switch(config-if-Et5)#ip rip v2-broadcast
switch(config-if-Et5)#exit
switch(config)#
```

network (RIP)

The **network** command specifies which network the switch runs Routing Information Protocol (RIP), and also specifies which routes will be accepted into the RIP routing table. Multiple network commands can be issued to create a network list on which RIP runs.

The switch enables RIP on all interfaces in the specified network.

The **no network** and **default network** commands disable RIP on the specified network by removing the corresponding **network** command from ***running-config***.

Command Mode

Router-RIP Configuration

Command Syntax

```
network NETWORK_ADDRESS
no network NETWORK_ADDRESS
default network NETWORK_ADDRESS
```

Parameters

- ***NETWORK_ADDRESS*** network IP address. Entry formats include the following:
  - *ipv4_subnet* IPv4 subnet (CIDR notation).
  - *ipv4_addr* **mask** *wildcard_mask* IP address and wildcard-mask.

Examples

- This command enables RIP on 10.168.1.1/24

```
switch(config)#router rip
switch(config-router-rip)#network 10.168.1.1/24
switch(config-router-rip)#
```

- This command also enables RIP on 10.168.1.1/24

```
switch(config-router-rip)#network 10.168.1.1 mask 0.0.0.255
switch(config-router-rip)#
```

redistribute (RIP)

The **redistribute** command enables the importing of routes from a specified routing domain to RIP.

- **connected** by default, RIP redistributes all connected routes that are established when IP is enabled on an interface. The route-map parameter facilitates the exclusion of connected routes from redistribution by specifying a route map that denies the excluded routes.
- **BGP**, **OSPF**, and **IP static routes** by default, routes are not redistributed. The redistribution command without the route-map parameter faciltates the redistribution of all routes from the specified source.

The **no redistribute** and **default redistribute** commands reset the default route redistribution setting by removing the **redistribute** statement from ***running-config***.

Command Mode

Router-RIP Configuration

Command Syntax

```
redistribute connected ROUTE_MAP
redistribute ROUTE_TYPE [ROUTE_MAP]
no redistribute connected ROUTE_MAP
no redistribute ROUTE_TYPE
default redistribute connected ROUTE_MAP
default redistribute ROUTE_TYPE
```

Parameters

- ***ROUTE_TYPE*** source from which routes are redistributed. Options include:
  - **BGP** routes from a BGP domain.
  - **OSPF** routes from an OSPF domain.
  - **OSPF match external** Routes external to RIP, but imported from OSPF.
  - **OSPF match internal** OSPF routes that are internal to the AS.
  - **static** IP static routes.
- ***ROUTE_MAP*** route map that determines the routes that are redistributed. Options include:
  - <no parameter> all routes are redistributed.
  - **route-map** *map_name* only routes in the specified route map are redistributed.

Example

- These commands redistribute OSPF routes into RIP.

```
switch(config)#router rip
switch(config-router-rip)#redistribute OSPF
switch(config-router-rip)#
```

router rip

The **router rip** command places the switch in router-rip configuration mode to configure the Routing Information Protocol (RIP) routing process. Router-rip configuration mode is not a group change mode; ***running-config*** is changed immediately upon command entry. The **exit** command does not affect ***running-config***.

The **no router rip** and **default router rip** commands disable RIP and remove all user-entered ***router-rip*** configuration statements from ***running-config***. To disable RIP without removing configuration statements, use the **shutdown (RIP)** command.

The **exit** command returns the switch to global configuration mode.

Command Mode

Global Configuration

Command Syntax

```
router rip
no router rip
default router rip
```

Commands Available in router-rip Configuration Mode

- **default-metric**
- **distance (RIP)**
- **network (RIP)**
- **redistribute (RIP)**
- **shutdown (RIP)**
- **timers basic (RIP)**

Example

- This command places the switch in router-rip configuration mode.

```
switch(config)#router rip
switch(config-router-rip)#
```

router rip vrf

The **router rip** command places the switch in router-RIP configuration mode to configure an RIP routing instance in the non-default VRF.

The **no router rip vrf** and **default router rip vrf** commands disable an RIP routing instance in the non-default VRF, and remove all user-entered ***router-rip*** configuration statements from ***running-config***. To disable RIP without removing configuration statements, use the **shutdown (RIP)** command.

The **exit** command returns the switch to global configuration mode.

Command Mode

Global Configuration

Command Syntax

```
router rip vrf [RIP_INSTANCE]
no router rip vrf [RIP_INSTANCE]
default router rip vrf [RIP_INSTANCE]
```

Parameters

- ***RIP_INSTANCE*** configure a RIP VRF instance in the non-default VRF*:*

Example

- This command configures a RIP instance in the non-default VRF.

```
switch(config)#router rip vrf test
switch(config-router-rip-router-rip-vrf-test)#no shutdown
switch(config-router-rip)#exit
switch(config)#
```

- This command disables a RIP instance in the non-default VRF.

```
switch(config)#no router rip vrf test
switch(config)#
```

show ip rip database

The **show ip rip database** command displays information about routes in the Routing Information Base. The default command displays active routes and learned routes not used in deference to higher priority routes from other protocols.

Thiscommand has the following forms:

- default (no arguments): information about all RIP routes.
- IPv4 address and mask: information about the referenced addresses
- **active:** information about routes not superceded by routes from other protocols.

Command Mode

EXEC

Command Syntax

show ip rip database [*FILTER*]

Parameters

- ***FILTER*** routing table entries that the command displays. Values include*:*
  - <no parameter> displays all routing table entries
  - **active** displays all active routing table entries.
  - *net_addr* subnet address (CIDR or address-mask). Command displays entries in this subnet.

Examples

- This command displays all active rip routes.

```
switch>show ip rip database active
10.168.11.0/24 directly connected, Et4
10.168.13.0/24
[1] via 10.168.14.2, 00:00:25, Et4
[2] via 10.168.15.2, 00:00:20, Et1
10.168.13.0/24
[1] via 10.168.14.2, 00:00:25, Et3
```

- This command submits a query for RIP route information for a network.

```
switch>show ip rip database 10.168.13.0/16
10.168.13.0/24
[1] via 10.168.14.2, 00:00:25, Et4
[2] via 10.168.15.2, 00:00:20, Et1
```

- This command returns information for all RIP routes.

```
switch>show ip rip database
10.1.0.0/255.255.255.0
[1] via 10.8.31.15, 00:00:21, Et2, holddown
10.2.0.0/255.255.255.0
[1] via 10.8.31.15, 00:00:21, Et2, holddown
10.3.0.0/255.255.255.0
[1] via 10.8.31.15, 00:00:21, Et2, inactive
10.212.0.0/255.255.255.0
[1] via 10.8.31.15, 00:00:21, Et2, active
10.214.0.0/255.255.255.0
[1] via 10.8.12.17, 00:00:30, Et4, active
```

show ip rip neighbors

The **show ip rip neighbors** command displays information about all RIP route gateways. The output displays the IPv4 address, the last heard time of the gateway, and characteristic flags applying to the gateway.

Command Mode

EXEC

Command Syntax

show ip rip neighbors

Example

- The **show ip rip neighbors** query displays information about all the gateways of RIP routes.

```
switch>show ip rip neighbors
Gateway     Last-Heard      Bad-Packets     Bad-Routes    Flags
10.2.12.33  00:00:15                                      SRC, TRSTED,
                                                          ACCPTED, RJCTED,
                                                          Q_RJCTED, AUTHFAIL
```

shutdown (RIP)

The **shutdown** command disables RIP on the switch without modifying the RIP configuration. RIP is disabled by default.

The **no shutdown** command enables RIP. The **default shutdown** command disables RIP.

Command Mode

```
Router-RIP Configuration
```

Command Syntax

```
shutdown
no shutdown
default shutdown
```

Examples

- This command disables RIP on the switch.

```
switch(config)#router rip
switch(config-router-rip)#shutdown
switch(config-router-rip)#
```

- This command enables RIP on the switch.

```
switch(config-router-rip)#no shutdown
switch(config-router-rip)#
```

timers basic (RIP)

The **timers basic** command configures the update interval, the expiration time, and the deletion time for routes received and sent through RIP. The command requires value declaration of all values.

- The update time is the interval between unsolicited route responses.
- The expiration time is initialized when a route is established and any time an update is received for the route.
- The deletion time is initialized when the expiration time elapses and the route is invalid. It is retained in the routing table until deletion time expiry.

The **no timers basic** and **default timers basic** commands return the timer values to their default values by removing the **timers-basic** command from ***running-config***.

Command Mode

Router-RIP Configuration

Command Syntax

```
timers basic update_time expire_time deletion_time
no timers basic
default timers basic
```

Parameters

- *update_time* Default is 30 seconds
- *expire_time* Default is 180 seconds.
- *deletion_time* Default is 120 seconds.

Parameter values are in seconds and range from ***5*** to ***2147483647***.

Example

- This command sets the update (60 seconds), expiration (90 seconds), and deletion (150 seconds) times.

```
switch(config)#router rip
switch(config-router-rip)#timers basic 60 90 150
switch(config-router-rip)#
```

vrf definition

The **vrf definition** command configure a non-default VRF, enable routing in it, and configure the network command under the configuration router RIP for the prefix to which the interface belongs.

The **no vrf definition** and **default vrf definition** commands disable a non-default VRF.

The **exit** command returns the switch to global configuration mode.

Command Mode

Global Configuration

Command Syntax

```
vrf definition [RIP_INSTANCE]
no vrf definition [RIP_INSTANCE]
default vrf definition [RIP_INSTANCE]
```

Parameters

- ***RIP_INSTANCE*** configure the definition of a RIP VRF instance in the non-default VRF*:*

Examples

- This command configures a non-default VRF and enables unicast routing in it.

```
switch(config)#vrf definition test
switch(config-vrf-test)#rd 1:1
switch(config-vrf-test)#exit
switch(config)#ip routing vrf test
switch(config)#
```

- This command disables a non-default VRF.

```
switch(config)#no vrf definition test
switch(config)#
```

vrf forwarding

The **vrf forwarding** command configures a non-default VRF, enables routing in it, and configures the network command under the configuration router RIP for the prefix to which the interface belongs. By default, all interfaces belong to the default VRF until VRF forwarding is executed.

The **exit** command returns the switch to global configuration mode.

Command Mode

Global Configuration

Command Syntax

```
vrf forwarding [RIP_INSTANCE]
```

Parameters

- ***RIP_INSTANCE*** configure forwarding of a RIP VRF instance in the non-default VRF.

Example

- This command configures an interface as part of a non-default VRF, by configuring the network command under the configuration router RIP for the prefix to which the interface belongs.

```
switch(config)#interface Ethernet 3 / 1
switch(config-if-Et3/1)#no switchport
switch(config-if-Et3/1)#ip address 1.0.0.1/24
switch(config-if-Et3/1)#vrf forwarding test
switch(config-if-Et3/1)#network 1.0.0.1
switch(config-if-Et3/1)#exit
switch(config)#
```

Chapter 31

IS-IS

Intermediate System-to-Intermediate System (IS-IS) intra-domain routing information exchange protocol is designed by the International Organization for Standardization to support connectionless networking. This protocol is a dynamic routing protocol.

This chapter contains the following sections.

- Section 31.1: IS-IS Introduction
- Section 31.2: IS-IS
- Section 31.3: IS-IS Command Descriptions

31.1 IS-IS Introduction

IS-IS is a link state protocol, which uses the shortest path first (SPF) algorithm. IS-IS and the OSPF protocol are similar in many aspects. As an interior gateway protocol (IGP), IS-IS runs inside an AS. To enable IS-IS, you must instantiate an IS-IS routing instance and assign it to an interface.

31.2 IS-IS

These sections describe IS-IS configuration tasks:

- Section 31.2.1: Enabling IS-IS
- Section 31.2.2: IS-IS Optional Global Parameters
- Section 31.2.3: IS-IS Interface Optional Parameters
- Section 31.2.4: Disabling IS-IS
- Section 31.2.5: Verifying IS-IS

31.2.1 Enabling IS-IS

For the normal operation of the IS-IS protocol, the **router isis** command must be used to enable the IS-IS instance. Then the **net** command is used to set a Network Entity Title (NET) for the device. Next you must configure at least one **address-family**. Lastly, the **isis enable** command is used to enable IS-IS on the desired interface. The IS-IS protocol is enabled upon the completion of these configurations.

To enable IS-IS, the following tasks must be performed in the global configuration mode.

- Section 31.2.1.1: Enable IS-IS Globally and Specify an IS-IS Instance
- Section 31.2.1.2: Configure the Network Entity Title (NET)
- Section 31.2.1.4: Enable IS-IS on a Specified Interface
- Section 31.2.1.3: Set the Address Family Configuration

31.2.1.1 Enable IS-IS Globally and Specify an IS-IS Instance

The switch supports only one IS-IS routing instance. The routing instance uniquely identifies the switch to other devices. IS-IS configuration commands apply globally to the IS-IS instance.

The switch must be in router IS-IS configuration mode to run IS-IS configuration commands. The **router isis** command places the switch in router IS-IS configuration mode.

Example

- These commands place the switch in router IS-IS configuration mode. It also creates an IS-IS routing instance named Osiris.

```
switch(config)#router isis Osiris
switch(config-router-isis)#
```

31.2.1.2 Configure the Network Entity Title (NET)

After creating an IS-IS routing instance, you should also configure the Network Entity Title (NET) with the **net** command. The NET defines the current IS-IS area address and the system ID of the device.

Example

- These commands define the current IS-IS area address and the system ID of the device.

```
switch(config)#router isis Osiris
switch(config-router-isis)# net 49.0001.1010.1040.1030.00
```

31.2.1.3 Set the Address Family Configuration

The **address-family** command allows you to enable the address families that IS-IS will route and also enter a configuration sub-mode to configure settings that are distinct to that address family. Currently Arista does not support per address family options. The address families supported are IPv4 unicast and IPv6 unicast.

Example

- These commands enable and enter the address family mode for IPv4 unicast.

```
switch(config)#router isis Osiris
switch(config-router-isis)#address-family ipv4 unicast
switch(config-router-isis-af)#
```

31.2.1.4 Enable IS-IS on a Specified Interface

After enabling IS-IS, you need to specify on which interface IS-IS will be run with the **isis enable** command.

Example

- These commands enable IS-IS on the specified interface Ethernet 4.

```
switch(config-router-isis)#interface ethernet 4
switch(config-if-Eth4)#isis enable 4
```

31.2.2 IS-IS Optional Global Parameters

After globally enabling IS-IS, the following global parameters can be configured.

- Section 31.2.2.1: Set the Router Type
- Section 31.2.2.2: Configure IS-IS to Redistribute Routes of Other Protocols
- Section 31.2.2.3: Set the Overload Bit
- Section 31.2.2.4: Configure IS-IS full MD5
- Section 31.2.2.6: Enable Logging for Peer Changes
- Section 31.2.2.7: Set the IS-IS hostname

31.2.2.1 Set the Router Type

The **is-type** command sets the routing level for an IS-IS instance.

Example

- These commands specify level-2 for the IS-IS instance.

```
switch(config)#router isis Osiris
switch(config-router-isis)#is-type level-2
switch(config-router-isis)#
```

31.2.2.2 Configure IS-IS to Redistribute Routes of Other Protocols

To redistribute static and/or connected routes into IS-IS, use the **redistribute (IS-IS)** command.

Example

- These commands redistribute connected routes into the IS-IS domain.

```
switch(config)#router isis Osiris
switch(config-router-isis)#redistribute connected
switch(config-router-isis)#
```

31.2.2.3 Set the Overload Bit

The **set-overload-bit** command used without the on-startup option, informs other devices not to use the local router to forward transit traffic. When used with the on-startup option, the overload bit is set for the interval specified after startup.

In scenarios when Border Gateway Protocol (BGP) routes are resolved using Interior Gateway Protocol (IGP), if the transit router reboots and becomes available again, IGP will consider the transit router for an optimal path again. After rebooting, it will black hole traffic until the transit router learns the external destination reachability information via BGP.

Examples

- These commands configure the switch and sets **the overload bit to 120 seconds after startup.**

```
switch(config)#router isis Osiris
switch(config-router-isis)#set-overload-bit on-startup 120
switch(config-router-isis)#
```

- These commands configure the overload bit until BGP converges. If BGP fails to converge within the set timeout default period, then the overload bit gets cleared.

```
switch(config)#router isis Osiris
switch(config-router-isis)#set-overload-bit on-startup wait-for-bgp
switch(config-router-isis)#set-overload-bit on-startup wait-for-bgp timeout 750
switch(config-router-isis)#
```

31.2.2.4 Configure IS-IS full MD5

To configure authentication for the IS-IS instance causing LSPs, CSNPs and PSNPs to be authenticated, use the **authentication mode** and **authentication key** commands. To configure authentication on the interface, causing IS-IS Hellos to be authenticated, use the **isis authentication mode.** and **isis authentication key** commands.

Two forms of authentication are supported by the IS-IS routing protocol: Clear text authentication and MD5 authentication. The difference between the two forms of authentication is in the level of security provided. In the case of clear text authentication, the password is specified as text in the authentication TLV, making it possible for an attacker to break authentication by sniffing and capturing IS-IS PDUs on the network.

HMAC MD5 authentication provides much stronger authentication by computing the message digest (on the IS-IS PDU contents) using the secret key to produce a hashed message authentication code (HMAC). Different modes of authentication can be specified on the interface, which authenticates IIH PDUs (IS-IS hello PDUs), and globally in the router IS-IS mode, in which the LSPs, CSNPs and PSNPs are authenticated. Area wide and domain wide authentication can be specified for L1 and L2 routers respectively.

Example

- These commands configure authentication for the IS-IS instance causing LSPs, CSNPs and PSNPs to be authenticated.

```
switch(config)#router isis 1
switch(config-router-isis)#authentication mode md5
switch(config-router-isis)#authentication key secret
switch(config-router-isis)#
```

- These commands configure authentication on the interface causing IS-IS Hellos to be authenticated.

```
switch(config)#interface Ethernet 3/6
switch(config-if-Et3/6)#isis authentication mode text
switch(config-if-Et3/6)#isis authentication key 7 cAm28+9a/xPi04o7hjd8Jw==
switch(config-if-Et3/6)#
```

31.2.2.5 Set the SPF Interval

The **spf-interval** command configures the shortest path first (SPF) timer. IS-IS runs SPF calculations following a change in the network topology or the link state database. The SPF timer defines the minium interval between two successive IS-IS SPF calculations.

Example

- These commands configures the SPF interval to 50 seconds.

```
switch(config)#router isis Osiris
switch(config-router-isis)#spf-interval 50
switch(config-router-isis)#
```

31.2.2.6 Enable Logging for Peer Changes

The **log-adjacency-changes (IS-IS)** command configures the switch to send syslog messages when it detects IS-IS neighbor adjacency state changes.

Example

- These commands configure the switch to send a syslog message when a neighbor goes up or down.

```
switch(config)#router isis Osiris
switch(config-router-isis)#log-adjacency-changes
switch(config-router-isis)#
```

31.2.2.7 Set the IS-IS hostname

The **is-hostname** command configures the use of a human readable string to represent the symbolic name of an IS-IS router, and map the IS-IS system IDs and IS-IS hostnames. It also changes the output of IS-IS show commands, to show the IS-IS hostname in place of system IDs if the corresponding IS-IS hostname is known. However, syslogs still use IS-IS system IDs and not the IS-IS hostname.

By default if there's a hostname configured on the switch, it is used as the IS-IS hostname. It is also possible to unconfigure an assigned hostname for IS-IS using the **no is-hostname** command. When the IS-IS hostname is removed, the switch goes back to using the switch's hostname as the IS-IS hostname.

Examples

- These commands configure the IS-IS hostname to the symbolic name foobar for the IS-IS router.

```
switch(config)#router isis inst1
switch(config-router-isis)#is-hostname ishost1
switch(config-router-isis)#
```

- These commands unconfigure the IS-IS hostname of the symbolic name foobar for the IS-IS router.

```
switch(config)#router isis inst1
switch(config-router-isis)#no is-hostname ishost1
switch(config-router-isis)#
```

31.2.2.8 Configuring IS-IS Multi-Topology

The **multi-topology** command configures IS-IS Multi-Topology (MT) support (disabled by default), enabling an IS-IS router to compute a separate topology for IPv4 and IPv6 links in the network. With MT configured, not all the links in a network need to support both IPv4 and IPv6. Some can support IPv4 or IPv6 individually. The IPv4 SPF will install IPv4 routes using the IPv4 topology, and similarly the IPv6 SPF will install IPv6 routes using the IPv6 topology. Without MT support, all links in an IS-IS network need to support the same set of address families.

When MT is enabled, and each link has a separate IPv4 metric and IPv6 metric.

The **isis ipv6 metric** command configures the IPv6 metric.

The **isis network** command configures the IPv4 or IPv6 address family individually on an interface with both IPv4 and IPv6 addresses.

The address families that are enabled on an interface are based on the global address families enabled in router IS-IS configuration mode, and the addresses configured on the interface. To enable a particular address family on an interface, it needs to have an address configured in that address family. In the case where both IPv4 and IPv6 address families are enabled in router IS-IS configuration mode, then if an interface has IPv4 and IPv6 addresses, both IPv4 and IPv6 address families are enabled on that interface. In the case of an interface with only an IPv4 address family, the IPv4 address family is enabled on that interface. Where an interface only has an IPv6 address family, the IPv6 address family is enabled on that interface. Finally, where only the IPv6 address family is enabled in router IS-IS config mode and MT is enabled, then the IPv6 address family is enabled on all interfaces which have an IPv6 address configured.

Examples

- These commands configure MT for the IS-IS router.

```
switch(config)#router isis 1
switch(config-router-isis)#address-family ipv6 unicast
switch(config-router-isis-af)#multi-topology
switch(config-router-isis-af)#
```

- These commands unconfigure MT for the IS-IS router.

```
switch(config)#router isis 1
switch(config-router-isis)#address-family ipv6 unicast
switch(config-router-isis-af)#no multi-topology
switch(config-router-isis-af)#
```

- These commands configure the IPv6 metric.

```
switch(config)#interface Ethernet 5/6
switch(config-if-Et5/6)#isis ipv6 metric 30
switch(config-if-Et5/6)#
```

- These commands configure the IPv4 address family on an interface with both IPv4 and IPv6 addresses.

```
switch(config)#interface Ethernet1
switch(config-if-Et1)#isis multi-topology address-family ipv4 unicast
switch(config-if-Et1)#
```

- These commands configure the IPv6 address family on an interface with both IPv4 and IPv6 addresses.

```
switch(config)#interface Ethernet1
switch(config-if-Et1)#isis multi-topology address-family ipv6 unicast
switch(config-if-Et1)#
```

- These commands configure both the IPv4 and IPv6 address families on an interface.

```
switch(config)#interface Ethernet1
switch(config-if-Et1)#no isis multi-topology address-family unicast
switch(config-if-Et1)#
```

31.2.3 IS-IS Interface Optional Parameters

After globally enabling IS-IS, the following parameters can be configured on individual interfaces.

- Section 31.2.3.1: Set the Hello Packet Interval
- Section 31.2.3.2: Configure the Hello Multiplier for the Interface
- Section 31.2.3.3: Configure the IS-IS Metric
- Section 31.2.3.4: Set the LSP Interval

- Section 31.2.3.5: Set the IS-IS Priority
- Section 31.2.3.6: Configure an Interface as Passive
- Section 31.2.3.7: Configure BFD support for IS-IS for IPv4

31.2.3.1 Set the Hello Packet Interval

The **isis hello-interval** command periodically sends hello packets to maintain adjacency through the transmitting/receiving of the hello packets. The hello packet interval can be modified.

Example

- These commands configure a hello interval of 60 seconds for Ethernet 4.

```
switch(config)#interface ethernet 4
switch(config-if-Et4)#isis hello-interval 60
switch(config-if-Et4)#
```

31.2.3.2 Configure the Hello Multiplier for the Interface

The switch maintains the adjacency by sending/receiving hello packets. When receiving no hello packets from the peer within a time interval, the local switch considers the neighbors invalid.

The **isis hello-multiplier** command calculates the hold time announced in hello packets by multiplying this number with the configured **isis hello-interval**.

Example

- These commands configure a hello multiplier of 45 for Ethernet 4.

```
switch(config)#interface ethernet 4
switch(config-if-Et4)#isis hello-interval 60
switch(config-if-Et4)#isis hello-multiplier 45
switch(config-if-Et4)#
```

31.2.3.3 Configure the IS-IS Metric

The **isis metric** command sets cost for sending information over a specific interface. At present only wide metrics are supported.

Example

- These commands configure a metric cost of 30 for sending information over Ethernet 5.

```
switch(config)#interface ethernet 5
switch(config-if-Et5)#isis metric 30
switch(config-if-Et5)#
```

31.2.3.4 Set the LSP Interval

The **isis lsp-interval** command configures the minimum interval between successive LSP transmissions on an interface.

Example

- This command sets the LSP interval on interface Ethernent 5 to 600 milliseconds.

```
switch(config)#interface ethernet 5
switch(config-if-Et5)# isis lsp-interval 600
switch(config-if-Et5)#
```

31.2.3.5 Set the IS-IS Priority

The **isis priority** command determines which device will be the Designated Intermediate System (DIS). The device with the highest priority will become the DIS.

Example

- These commands configure a device priority of 60 on interface Ethernet 5.

```
switch(config)#interface ethernet 5
switch(config-if-Et5)#isis priority 60
switch(config-if-Et5)#
```

31.2.3.6 Configure an Interface as Passive

The **isis passive** and **passive-interface (IS-IS)** commands configure IS-IS interfaces as passive to interfaces where adjacencies are wanted. The interface does not send or receive IS-IS packets on an interface configured as passive.

Examples

- These commands configure Ethernet interface 10 as a passive interface. The switch neither sends IS-IS packets received on passive interfaces.

```
switch(config)#interface ethernet 10
switch(config-if-Etl0)#isis passive
switch(config-if-Etl0)#
```

- These commands configure Ethernet interface 10 as a passive interface in the router IS-IS mode.

```
switch(config)#router isis Osiris
switch(config-router-isis)#passive-interface interface ethernet 10
switch(config-router-isis)#
```

31.2.3.7 Configure BFD support for IS-IS for IPv4

The **bfd all-interfaces** and **isis bfd** commands configure Bidirectional Forwarding Detection (BFD), a low overhead protocol designed to provide rapid detection of failures at any protocol layer in the path between adjacent forwarding engines over any media. BFD is supported for IS-IS IPv4 routes.

Examples

- These commands enable BFD for all the interfaces on which IS-IS is enabled. By default BFD is disabled on all the interfaces.

```
switch(config)#router isis 1
switch(config-router-isis)#address-family ipv4
switch(config-router-af)#bfd all-interfaces
switch(config-router-af)#
```

- These commands enable BFD on IS-IS interfaces.

```
switch(config)#interface Ethernet 5/6
switch(config-if-Et5/6)#isis bfd
switch(config-if-Et5/6)#
```

31.2.4 Disabling IS-IS

An IS-IS instance can be shut down globally, or the IS-IS protocol can be disabled on individual interfaces.

The **shutdown (IS-IS)** command shuts down an IS-IS instance globally.

Example

- These commands disable IS-IS globally without modifying the IS-IS configuration.

```
switch(config)#router isis Osiris
switch(config-router-isis)#shutdown
switch(config-router-isis)#
```

The **no isis enable** command disables IS-IS on an interface.

Example

- These commands disable IS-IS on interface Ethernet 4.

```
switch(config-router-isis)#interface ethernet 4
switch(config-if-Eth4)#no isis enable
```

31.2.5 Verifying IS-IS

The following tasks verify the IS-IS peer and connection configuration:

- Section 31.2.5.1: Verify the Link State Database
- Section 31.2.5.2: Verify the Interface Information for the IS-IS Instance
- Section 31.2.5.3: Verify the IS-IS Neighbor Information
- Section 31.2.5.4: Verify IS-IS Instance Information

31.2.5.1 Verify the Link State Database

To display the link state database of IS-IS, use the **show isis database** command.

Example

- This command displays the IS-IS link state database.

```
switch>show isis database

ISIS Instance: Osiris
  ISIS Level 2 Link State Database
    LSPID                 Seq Num   Cksum  Life  IS Flags
    1212.1212.1212.00-00  4         714    1064  L2 <>
    1212.1212.1212.0a-00  1         57417  1064  L2 <>
    2222.2222.2222.00-00  6         15323  1116  L2 <>
    2727.2727.2727.00-00  10        15596  1050  L2 <>
    3030.3030.3030.00-00  12        62023  1104  L2 <>
    3030.3030.3030.c7-00  4         53510  1104  L2 <>
switch>
```

31.2.5.2 Verify the Interface Information for the IS-IS Instance

To display interface information related to the IS-IS instance, use the **show isis interface** command.

Example

- This command displays IS-IS interface information.

```
switch>show isis interface

ISIS Instance: Osiris
  Interface Vlan20:
    Index: 59 SNPA: 0:1c:73:c:5:7f
    MTU: 1497 Type: broadcast
    Level 2:
      Metric: 10, Number of adjacencies: 2
      LAN-ID: 1212.1212.1212, Priority: 64
      DIS: 1212.1212.1212, DIS Priority: 64
  Interface Ethernet30:
    Index: 36 SNPA: 0:1c:73:c:5:7f
    MTU: 1497 Type: broadcast
    Level 2:
      Metric: 10, Number of adjacencies: 1
      LAN-ID: 3030.3030.3030, Priority: 64
      DIS: 3030.3030.3030, DIS Priority: 64
switch>
```

31.2.5.3 Verify the IS-IS Neighbor Information

To display general information for IS-IS neighbors that the device sees, use **show isis neighbors**.

Example

- This command displays information for IS-IS neighbors that the device sees.

```
switch>show isis neighbor

Inst Id   System Id          Type Interface      SNPA            State Hold time
10         2222.2222.2222       L2   Vlan20          2:1:0:c:0:0       UP    30
10         1212.1212.1212       L2   Vlan20          2:1:0:d:0:0       UP    9
10         3030.3030.3030       L2   Ethernet30      2:1:0:b:0:0       UP    9
switch>
```

31.2.5.4 Verify IS-IS Instance Information

To display the system ID, Type, Interface, IP address, State and Hold information for IS-IS instances, use the **show isis summary** command.

Example

- This command displays general information about IS-IS instances.

```
switch>show isis summary
ISIS Instance: Osiris
  System ID: 1010.1040.1030, administratively enabled, attached
  Internal Preference: Level 1: 115, Level 2: 115
  External Preference: Level 1: 115, Level 2: 115
  IS-Type: Level 2, Number active interfaces: 1
  Routes IPv4 only
  Last Level 2 SPF run 2:32 minutes ago
  Area Addresses:
    10.0001
  level 2: number dis interfaces: 1, LSDB size: 1
switch>
```

31.3 IS-IS Command Descriptions

Global Configuration Commands

- router isis

Interface Configuration Commands

- authentication mode
- authentication key
- isis authentication mode
- isis authentication key
- isis enable
- isis bfd
- isis hello-interval
- isis hello-multiplier
- isis ipv6 metric
- isis lsp-interval
- isis metric
- isis multi-topology
- isis network
- isis passive
- isis priority

Router IS-IS Configuration Mode (Includes Address-Family Mode)

- address-family
- is-hostname
- bfd all-interfaces
- is-type
- log-adjacency-changes (IS-IS)
- multi-topology
- net
- passive-interface (IS-IS)
- redistribute (IS-IS)
- set-overload-bit
- shutdown (IS-IS)
- spf-interval

Display Commands – EXEC Mode

- show isis database
- show isis hostname
- show isis interface
- show isis neighbors
- show isis summary
- show isis topology

address-family

The **address-family** command places the switch in address-family configuration mode.

Address-family configuration mode is not a group change mode; ***running-config*** is changed immediately after commands are executed. The **exit** command does not affect the configuration.

The switch supports these address families:

- ipv4-unicast
- ipv6-unicast

The **no address-family** and **default address-family** commands delete the specified address-family from ***running-config*** by removing all commands previously configured in the corresponding address-family mode.

The **exit** command returns the switch to router IS-IS configuration mode.

Command Mode

Router-IS-IS Configuration

Command Syntax

```
isis ADDRESS_FAMILY [TRANSMISSION]
no isis ADDRESS_FAMILY
default isis ADDRESS_FAMILY
```

Parameters

- ***ADDRESS_FAMILY*** Options include:
  - **ipv4** IPv4 unicast
  - **ipv6** IPv6 unicast
- ***MODE*** Options include:
  - <no parameter> Defaults to unicast.
  - **unicast** All IPv4 or IPv6 addresses are active.

Example

- These commands enter the address family mode for IPv4 unicast.

```
switch(config)#router isis Osiris
switch(config-router-isis)#address-family ipv4 unicast
switch(config-router-isis-af)#
```

- To exit from the IPv4 IS-IS unicast address family configuration mode, enter the following command.

```
switch(config)#router isis Osiris
switch(config-router-isis)#address-family ipv4 unicast
switch(config-router-isis-af)#exit
switch(config-router-isis)#
```

bfd all-interfaces

The **bfd all-interfaces** command places the switch in address-family configuration mode.

The **bfd all-interfaces** and **isis bfd** commands configure Bidirectional Forwarding Detection (BFD), a low overhead protocol designed to provide rapid detection of failures at any protocol layer in the path between adjacent forwarding engines over any media. BFD is supported for IS-IS IPv4 routes.

Command Mode

Router-Address-Family Configuration

Command Syntax

```
bfd all-interfaces
```

Example

- These commands enable BFD for all the interfaces on which IS-IS is enabled. By default BFD is disabled on all the interfaces.

```
switch(config)#router isis 1
switch(config-router-isis)#address-family ipv4
switch(config-router-af)#bfd all-interfaces
switch(config-router-af)#
```

authentication mode

The **authentication mode** command configures authentication for the IS-IS instance causing LSPs, CSNPs and PSNPs to be authenticated.

The **no authentication mode** and **default authentication mode** commands disables authentication for the IS-IS instance.

Command Mode

ISIS-Router Configuration

Command Syntax

```
authentication mode <md5 | text> [LAYER_VALUE]
no authentication mode <md5 | text> [LAYER_VALUE]
default authentication mode <md5 | text> [LAYER_VALUE]
```

Parameters

- *LAYER_VALUE* layer value*;* options include:
  - **level-1**
  - **level-2**

Examples

- These commands configure authentication for the IS-IS instance causing LSPs, CSNPs and PSNPs to be authenticated.

```
switch(config)#router isis 1
switch(config-router-isis)#authentication mode md5
switch(config-router-isis)#
```

authentication key

The **authentication key** command configures the authentication key for the IS-IS instance causing LSPs, CSNPs and PSNPs to be authenticated.

The **no authentication key** and **default authentication key** commands disables the authentication key for the IS-IS instance.

Command Mode

ISIS-Router Configuration

Command Syntax

```
authentication key [0 | 7] [LAYER_VALUE]
no authentication key [0 | 7] [LAYER_VALUE]
default authentication key [0 | 7] [LAYER_VALUE]
```

Parameters

- *LAYER_VALUE* layer value; options include:
  - **level-1**
  - **level-2**

Examples

- These commands configure authentication for the IS-IS instance causing LSPs, CSNPs and PSNPs to be authenticated.

```
switch(config)#router isis 1
switch(config-router-isis)#authentication key secret
switch(config-router-isis)#
```

isis authentication mode

The **isis authentication mode** command configures authentication on the interface causing IS-IS Hellos to be authenticated.

The **no isis authentication mode** and **default isis authentication mode** commands disables authentication for the IS-IS instance.

Command Mode

Interface-Ethernet Configuration

Command Syntax

```
isis authentication mode <md5 | text> [LAYER_VALUE]
no isis authentication mode <md5 | text> [LAYER_VALUE]
default isis authentication mode <md5 | text> [LAYER_VALUE]
```

Parameters

- *LAYER_VALUE* layer value; options include:
  - **level-1**
  - **level-2**

Examples

- These commands configure authentication on the interface causing IS-IS Hellos to be authenticated.

```
switch(config)#interface Ethernet 3/6
switch(config-if-Et3/6)#isis authentication mode text
switch(config-if-Et3/6)#isis authentication key 7 cAm28+9a/xPi04o7hjd8Jw==
switch(config-if-Et3/6)#
```

isis authentication key

The **isis authentication key** command configures the authentication key on the interface causing IS-IS Hellos to be authenticated.

The **no isis authentication mode** and **default isis authentication mode** commands disables the authentication key for the IS-IS instance.

Command Mode

Interface-Ethernet Configuration

Command Syntax

```
isis authentication key [0 | 7] [LAYER_VALUE]
no isis authentication key [0 | 7] [LAYER_VALUE]
default isis authentication key [0 | 7] [LAYER_VALUE]
```

Parameters

- *LAYER_VALUE* layer value; options include:
  - **level-1**
  - **level-2**

Examples

- These commands configure authentication on the interface causing IS-IS Hellos to be authenticated.

```
switch(config)#interface Ethernet 3/6
switch(config-if-Et3/6)#isis authentication mode text
switch(config-if-Et3/6)#isis authentication key 7 cAm28+9a/xPi04o7hjd8Jw==
switch(config-if-Et3/6)#
```

isis enable

The **isis enable** command activates the corresponding IS-IS routing instance on the configuration mode interface. By default, the IS-IS routing instance is not enabled on an interface.

The **no isis enable** and **default isis enable** commands disable IS-IS on the configuration mode interface by removing the corresponding **isis enable** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration
Interface-Loopback Configuration
Interface-Port-channel Configuration
Interface-VLAN Configuration

Command Syntax

```
isis enable instance_id
no isis enable
default isis enable
```

Parameters

- *instance_id* IS-IS instance name.

Examples

- These commands enable the IS-IS protocol on the interface Ethernet 4.

```
switch(config)#router isis Osiris
switch(config-router-isis)# net 49.0001.1010.1040.1030.00
switch(config-router-isis)#interface ethernet 4
switch(config-if-Eth4)#isis enable Osiris
```

- These commands disable the IS-IS protocol on the interface Ethernet 4.

```
switch(config)#interface ethernet 4
switch(config-if-Eth4)# no isis enable
```

isis bfd

The **isis bfd** command activates the corresponding IS-IS routing instance on the configuration mode interface. By default, the IS-IS routing instance is not enabled on an interface.

The **no isis enable** and **default isis enable** commands disable IS-IS on the configuration mode interface by removing the corresponding **isis enable** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration

Command Syntax

```
isis bfd
no isis bfd
default isis bfd
```

Examples

- These commands enable BFD on IS-IS interfaces.

```
switch(config)#interface Ethernet 5/6
switch(config-if-Et5/6)#isis bfd
switch(config-if-Et5/6)#
```

isis hello-interval

The **isis hello-interval** command sends Hello packets from applicable interfaces to maintain the adjacency through the transmitting and receiving of Hello packets. The Hello packet interval can be modified.

The **no isis hello-interval** and **default isis hello-interval** commands restore the default hello interval of 10 seconds on the configuration mode interface by removing the **isis hello-interval** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration
Interface-Loopback Configuration
Interface-Port-channel Configuration
Interface-VLAN Configuration

Command Syntax

```
isis hello-interval time
no isis hello-interval
default isis hello-interval
```

Parameters

- *time* Values range from 1 to 300; default is 10.

Examples

- These commands configure a hello interval of 45 seconds for VLAN 200.

```
switch(config)#interface vlan 200
switch(config-if-Vl200)#isis hello-interval 45
switch(config-if-Vl200)#
```

- These commands remove the configured hello interval of 45 seconds from VLAN 200.

```
switch(config)#interface vlan 200
switch(config-if-Vl200)#no isis hello-interval
switch(config-if-Vl200)#
```

- These commands configure a hello interval of 60 seconds for Ethernet 5.

```
switch(config)#interface ethernet 5
switch(config-if-Et5)#isis hello-interval 60
switch(config-if-Et5)#
```

- These commands remove the configured hello interval of 60 seconds from Ethernet 5.

```
switch(config)#interface ethernet 5
switch(config-if-Et5)#no isis hello-interval
switch(config-if-Et5)#
```

isis hello-multiplier

The **isis hello-multiplier** command specifies the number of IS-IS hello packets missed by a neighbor before the the adjacency is considered down.

The **no isis hello-multiplier** and **default isis hello-multiplier** commands restore the default hello interval of 3 on the configuration mode interface by removing the **isis hello-multiplier** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration
Interface-Loopback Configuration
Interface-Port-channel Configuration
Interface-VLAN Configuration

Command Syntax

```
isis hello-multiplier factor
no isis hello-multiplier
default isis hello-multiplier
```

Parameters

- *factor* Values range from 3 to 100; default is 3

Examples

- These commands configure a hello multiplier of 4 for VLAN 200.

```
switch(config)#interface vlan 200
switch(config-if-Vl200)#isis hello-multiplier 4
switch(config-if-Vl200)#
```

- These commands remove the configured hello multiplier of 4 from VLAN 200.

```
switch(config)#interface vlan 200
switch(config-if-Vl200)# no isis hello-multiplier
switch(config-if-Vl200)#
```

- These commands configure a hello multiplier of 45 for Ethernet 5.

```
switch(config)#interface ethernet 5
switch(config-if-Et5)#isis hello-multiplier 45
switch(config-if-Et5)#
```

- These commands remove the configured hello multiplier of 45 from Ethernet 5.

```
switch(config)#interface ethernet 5
switch(config-if-Et5)#no isis hello-multiplier
switch(config-if-Et5)#
```

isis ipv6 metric

The **isis ipv6 metric** command configures the IPv6 metric.

The **no isis ipv6 metric** and **default isis ipv6 metric** commands restore the default metric of 10 on the configuration mode interface.

Command Mode

Interface-Ethernet Configuration

Command Syntax

```
isis ipv6 metric metric_value
no isis ipv6 metric
default isis ipv6 metric
```

Parameters

- *metric_value* Values range from 1 to 16777214; default is 10.

Examples

- These commands configure the IPv6 metric.

```
switch(config)#interface Ethernet 5/6
switch(config-if-Et5/6)#isis ipv6 metric 30
switch(config-if-Et5/6)#
```

isis lsp-interval

The **isis lsp-interva**l command sets the interval at which IS-IS sends link-state information on the interface.

The **no isis lsp-interva**l and **default isis lsp-interva**l commands restores the default setting of 33 ms. by removing the **isis lsp-interva**l command from ***running-config***.

Command Mode

Interface-Ethernet Configuration
Interface-Loopback Configuration
Interface-Port-channel Configuration
Interface-VLAN Configuration

Command Syntax

```
isis lsp-interval period
no isis lsp-interval
default isis lsp-interval
```

Parameters

- *period* Value ranges from 1 through 3000. Default interval is 33 ms.

Examples

- This command sets the LSP interval on interface Ethernet 5 to 600 milliseconds.

```
switch(config)#interface ethernet 5
switch(config-if-Et5)# isis lsp-interval 600
switch(config-if-Et5)#
```

- This command removes the LSP interval on interface Ethernet 5.

```
switch(config)#interface ethernet 5
switch(config-if-Et5)# no isis lsp-interval
switch(config-if-Et5)#
```

isis metric

The **isis metric** command sets cost for sending information over an interface.

The **no isis metric** and **default isis metric** commands restores the default metric to its default value of 10 by removing the **isis metric** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration
Interface-Loopback Configuration
Interface-Port-channel Configuration
Interface-VLAN Configuration

Command Syntax

```
isis metric metric_cost
no isis metric
default isis metric
```

Parameters

- *metric_cost* Values range from 1 to 1677214. Default value is 10.

Examples

- These commands configure a metric cost of 30 for sending information over Ethernet 5.

```
switch(config)#router isis Osiris
switch(config-router-isis)#interface ethernet 5
switch(config-if-Et5)#isis metric 30
switch(config-if-Et5)#
```

- These commands remove the configured metric cost of 30 from Ethernet 5.

```
switch(config)#router isis Osiris
switch(config-router-isis)#interface ethernet 5
switch(config-if-Et5)#no isis metric
switch(config-if-Et5)#
```

isis multi-topology

The **isis multi-topology** command configures the IPv4 or IPv6 address family individually on an interface with both IPv4 and IPv6 addresses.

The **no isis multi-topology** and **default isis multi-topology** commands restores the default interface to both IPv4 and IPv6 address families.

Command Mode

Interface-Ethernet Configuration

Command Syntax

```
isis multi-topology address-family ipv4 unicast
no isis multi-topology address-family ipv4 unicast
default isis multi-topology address-family ipv4 unicast
```

Examples

- These commands configure the IPv4 address family on an interface with both IPv4 and IPv6 addresses.

```
switch(config)#interface Ethernet 5/6
switch(config-if-Et5/6)#isis multi-topology address-family ipv4 unicast
switch(config-if-Et5/6)#
```

- These commands configure the IPv6 address family on an interface with both IPv4 and IPv6 addresses.

```
switch(config)#interface Ethernet 5/6
switch(config-if-Et5/6)#isis multi-topology address-family ipv6 unicast
switch(config-if-Et5/6)#
```

- These commands configure both the IPv4 and IPv6 address families on an interface.

```
switch(config)#interface Ethernet 5/6
switch(config-if-Et5/6)#no isis multi-topology address-family unicast
switch(config-if-Et5/6)#
```

isis network

The **ip isis network** command sets the configuration mode interface as a point-to-point link. By default, interfaces are configured as broadcast links.

The **no ip isis network** and **default ip isis network** commands set the configuration mode interface as a broadcast link by removing the corresponding **ip isis network** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration
Interface-Loopback Configuration
Interface-Port-Channel Configuration
Interface-VLAN Configuration

Command Syntax

```
ip isis network point-to-point
no ip isis network
default ip isis network
```

Examples

- These commands configure Ethernet interface 10 as a point-to-point link.

```
switch(config)#interface ethernet 10
switch(config-if-Et10)# isis network point-to-point
switch(config-if-Et10)#
```

- This command restores Ethernet interface 10 as a broadcast link.

```
switch(config-if-Et10)#no isis network
switch(config-if-Et10)#
```

isis passive

The **isis passive** command disables IS-IS on an interface configured as passive. The switch won't send or process IS-IS packets received on passive interfaces. The switch will continue to advertise the IP address in the LSP.

The **no passive** command enables IS-IS on the interface. The **default passive** command sets the interface to the default interface activity setting by removing the corresponding **passive** or **no passive** statement from ***running-config***.

Command Mode

Interface-Ethernet Configuration
Interface-Loopback Configuration
Interface-Port-Channel Configuration
Interface-VLAN Configuration

Command Syntax

```
isis passive
no isis passive
default isis passive
```

Examples

- These commands configure Ethernet interface 10 as a passive interface.

```
switch(config)#router isis Osiris
switch(config-router-isis)#interface ethernet 10
switch(config-if-Et10)# isis passive
switch(config-if-Et10)#
```

- This command restores Ethernet interface 10 as a broadcast link.

```
switch(config-if-Et10)#no isis passive
switch(config-if-Et10)#
```

isis priority

The **isis priority** command sets IS-IS priority for the interface.

The default priority is 64. The network device with the highest priority will be elected as the designated intermediate router to send link-state advertisements for that network.

The **no isis priority** and **default isis priority** commands restore the default priority (64) on the configuration mode interface*.*

Command Mode

Interface-Ethernet Configuration
Interface-Loopback Configuration
Interface-Port-channel Configuration
Interface-VLAN Configuration

Command Syntax

```
isis priority priority_level
no isis priority
default isis priority
```

Parameters

- *priority_level* Value ranges from 0 to 127. Default value is 64.

Examples

- These commands configure a IS-IS priority of 60 on interface Ethernet 5.

```
switch(config)#router isis Osiris
switch(config-router-isis)#interface ethernet 5
switch(config-if-Et5)#priority 60
switch(config-if-Et5)#
```

- These commands restores the default IS-IS priority of 64 from interface Ethernet 5.

```
switch(config)#router isis Osiris
switch(config-router-isis)#interface ethernet 5
switch(config-if-Et5)# no priority
switch(config-if-Et5)#
```

- These commands configure the switch with a priority of 64 for VLAN 7.

```
switch(config)#interface vlan 7
switch(config-if-Vl7)#isis priority 64
switch(config-if-Vl7)#
```

- These command restores the default IS-IS priority of 64 for VLAN 7.

```
switch(config)#interface vlan 7
switch(config-if-Vl7)#no isis priority
switch(config-if-Vl7)#
```

is-hostname

The **is-hostname** command configures the use of a human readable string to represent the symbolic name of an IS-IS router, and map the IS-IS system IDs and IS-IS hostnames. It also changes the output of IS-IS show commands, to show the IS-IS hostname in place of system IDs if the corresponding IS-IS hostname is known. However, syslogs still use IS-IS system IDs and not the IS-IS hostname.

By default if there's a hostname configured on the switch, it is used as the IS-IS hostname. It is also possible to unconfigure an assigned hostname for IS-IS using the **no is-hostname** command. When the IS-IS hostname is removed, the switch goes back to using the switch's hostname as the IS-IS hostname.

Command Mode

Router-IS-IS Configuration

Command Syntax

```
is-hostname
no is-hostname
```

Example

- These commands configure the IS-IS hostname to the symbolic name ishost1 for the IS-IS router.

```
switch(config)#router isis inst1
switch(config-router-isis)#is-hostname ishost1
switch(config-router-isis)#
```

- These commands unconfigure the IS-IS hostname of the symbolic name ishost1 for the IS-IS router.

```
switch(config)#router isis inst1
switch(config-router-isis)#no is-hostname ishost1
switch(config-router-isis)#
```

is-type

The **is-type** command configures the routing level for an IS-IS instance.

An IS-IS router can be configured as Level-1-2 which can form adjacencies and exchange routing information with both Level-1 and Level-2 routers. A Level-1-2 router can be configured to transfer routing information from Level-1 to Level-2 areas and vice versa (via route leaking). By default, all routes from Level-1 area are always leaked into Level-2 network.

Command Mode

Router-IS-IS Configuration

Command Syntax

is-type *LAYER_VALUE*

Parameters

- ***LAYER_VALUE*** layer value options include:
  - **level-1**
  - **level-1-2**
  - **level-2**

Example

- These commands configure Level 1-2 routing on interface Ethernet 5.

```
switch(config)#router isis Osiris
switch(config-router-isis)#is-type level-1-2
switch(config-router-isis)#
```

- These commands configure Level 2 routing on interface Ethernet 5.

```
switch(config)#router isis Osiris
switch(config-router-isis)#is-type level-2
switch(config-router-isis)#
```

log-adjacency-changes (IS-IS)

The **log-adjacency-changes** command sets the switch to send syslog messages when it detects link state changes or when it detects that a neighbor state has changed.

The default option is active when ***running-config*** does not contain any form of the command. Entering the command in any form replaces the previous command state in ***running-config***.

Command Mode

Router-IS-IS Configuration

Command Syntax

```
log-adjacency-changes
no log-adjacency-changes
default log-adjacency-changes
```

Examples

- These commands configure the switch to send a syslog message when a neighbor state changes.

```
switch(config)#router isis Osiris
switch(config-router-isis)#log-adjacency-changes
switch(config-router-isis)#
```

- These commands configure not to log the peer changes.

```
switch(config)#router isis Osiris
switch(config-router-isis)#no log-adjacency-changes
switch(config-router-isis)#
```

multi-topology

The **multi-topology** command configures IS-IS Multi-Topology (MT) support (disabled by default), enabling an IS-IS router to compute a separate topology for IPv4 and IPv6 links in the network. With MT configured, not all the links in a network need to support both IPv4 and IPv6. Some can support IPv4 or IPv6 individually. The IPv4 SPF will install IPv4 routes using the IPv4 topology, and similarly the IPv6 SPF will install IPv6 routes using the IPv6 topology. Without MT support, all links in an IS-IS network need to support the same set of address families. When MT is enabled, and each link has a separate IPv4 metric and IPv6 metric.

The **no multi-topology** and **default multi-topology** commands restores the default interface to both IPv4 and IPv6 address families.

Command Mode

Interface-Ethernet Configuration

Command Syntax

```
multi-topology
no multi-topology
default multi-topology
```

Examples

- These commands configure MT for the IS-IS router.

```
switch(config)#router isis 1
switch(config-router-isis)#address-family ipv6 unicast
switch(config-router-isis-af)#multi-topology
switch(config-router-isis-af)#
```

- These commands unconfigure MT for the IS-IS router.

```
switch(config)#router isis 1
switch(config-router-isis)#address-family ipv6 unicast
switch(config-router-isis-af)#no multi-topology
switch(config-router-isis-af)#
```

net

The **net** command configures the name of Network Entity Title of the IS-IS instance. By default, no NET is defined.

The **no net** and **default net** commands removes the NET from ***running-config.***

Command Mode

Router-IS-IS Configuration

Command Syntax

net *mask_hex*
no net
default net

Parameters

- *maxk_hex* mask value. Format is hh.hhhh.hhhh.hhhh.hhhh.hhhh.hhhh.hhhh.hhhh.00.

Examples

- These commands specify the NET as 49.0001.1010.1040.1030.00, in which the system ID is 1010.1040.1030, area ID is 49.0001.

```
switch(config)#router isis Osiris
switch(config-router-isis)# net 49.0001.1010.1040.1030.00
switch(config-router-isis)#
```

- These commands remove NET 49.0001.1010.1040.1030.00 from ***running-config.***

```
switch(config)#router isis Osiris
switch(config-router-isis)# no net 49.0001.1010.1040.1030.00
switch(config-router-isis)#
```

passive-interface (IS-IS)

The **passive-interface** command disables IS-IS on a passive interface. The switch will continue to advertise the IP address in the LSP.

The **no passive** command enables IS-IS on the interface. The **default passive** command sets the interface to the default interface activity setting by removing the corresponding **passive** or **no passive** statement from ***running-config***.

Command Mode

Router-IS-IS Configuration

Command Syntax

```
passive-interface INTERFACE_NAME
no passive-interface INTERFACE_NAME
default passive-interface INTERFACE_NAME
```

Parameters

- ***INTERFACE_NAME*** Options include:
  - **ethernet** *e_range* Ethernet interface list.
  - **loopback** *l_range* Loopback interface list.
  - **port-channel** *p_range* Channel group interface list.
  - **vlan** *v_range* VLAN interface list.

  Valid *e_range*, *l_range*, *p_range*, and *v_range* formats include number, range, or comma-delimited list of numbers and ranges.

Examples

- These commands configure Ethernet interface 10 as a passive interface.

```
switch(config)#router isis Osiris
switch(config-router-isis)# passive-interface interface ethernet 10
```

- This command restores Ethernet interface 10 as an active interface.

```
switch(config-if-Et10)#no isis passive
switch(config-if-Et10)#
```

redistribute (IS-IS)

The **redistribute** command redistributes IS-IS connected or static routes. To disable the redistribution, use the **no redistribute** command.

The **no redistribute** and **default redistribute** commands disable route redistribution from the specified domain by removing the corresponding **redistribute** statement from ***running-config***.

Command Mode

Router-IS-IS Configuration

Command Syntax

```
redistribute ROUTE_TYPE
no redistribute ROUTE_TYPE
default redistribute ROUTE_TYPE
```

Parameters

- ***ROUTE_TYPE*** Options include:
  - **connected**
  - **static**

Examples

- These commands redistribute connected routes into the IS-IS domain.

```
switch(config)#router isis Osiris
switch(config-router-isis)#redistribute connected
switch(config-router-isis)#
```

- These commands redistribute static routes into the IS-IS domain.

```
switch(config)#router isis Osiris
switch(config-router-isis)#redistribute static
switch(config-router-isis)#
```

- These commands remove the redistributed connected routes from ***running-config***.

```
switch(config)#router isis Osiris
switch(config-router-isis)#no redistribute connected
switch(config-router-isis)#
```

- These commands remove the redistributed static routes from *running-config*.

```
switch(config)#router isis Osiris
switch(config-router-isis)#no redistribute static
switch(config-router-isis)#
```

router isis

The **router isis** command places the switch in router ISIS configuration mode.

Router ISIS configuration mode is not a group change mode; ***running-config*** is changed immediately after commands are executed. The **exit** command does not affect the configuration.

The **no router isis** command deletes the IS-IS instance.

The **exit** command returns the switch to global configuration mode.

Command Mode

Global Configuration

Command Syntax

```
router isis instance_name [VRF_INSTANCE]
no router isis instance_name
default router isis instance_name
```

Parameters

- *instance_name* routing instance.
- ***VRF_INSTANCE***
  - <no parameter>
  - **vrf** *vrf_name*

Examples

- These commands places the switch in router IS-IS mode and creates an IS-IS routing instance named Osiris.

```
switch(config)#router isis Osiris
switch(config-router-isis)#
```

- This command attempts to open an instance with a different routing instance name from that of the existing instance. The switch displays an error and stays in global configuration mode.

```
switch(config)#router isis Osiris
% More than 1 ISIS instance is not supported
switch(config)#
```

- This command deletes the IS-IS instance.

```
switch(config)#no router isis Osiris
switch(config)#
```

set-overload-bit

The **set-overload-bit** command used without the on-startup option will inform other devices not to use this switch in SPF computation. When used with the on-startup parameter, the overload bit is set for the interval after startup.

The **no set-overload-bit** and **default set-overload-bit** commands removes the corresponding **set-overload-bit** command from ***running-config***.

Command Mode

Router-IS-IS Configuration

Command Syntax

```
set-overload-bit TIMING
no set-overload-bit
default set-overload-bit
```

Parameters

- ***TIMING*** Options include:
  - <no parameter>
  - **on-startup** <***1*** to ***3600***>

Example

- These commands configure the switch to sets **the overload bit to 120 seconds after startup.**

```
switch(config)#router isis Osiris
switch(config-router-isis)#set-overload-bit on-startup 120
switch(config-router-isis)#
```

- These commands remove the configured **overload bit of 120 seconds from the *running-config*.**

```
switch(config)#router isis Osiris
switch(config-router-isis)#no set-overload-bit on-startup
switch(config-router-isis)#
```

show isis database

The **show isis database** command displays the link state database of IS-IS. The default command displays active routes and learned routes.

Command Mode

EXEC

Command Syntax

```
show isis database [INSTANCES] [INFO_LEVEL]
show isis database [INFO_LEVEL] VRF_INSTANCE
```

Parameters

- ***INSTANCES*** Options include:
  - <no parameter>
  - *instance_name*
- ***INFO_LEVEL*** Options include:
  - <no parameter>
  - **detail**
- ***VRF_INSTANCE*** specifies the VRF instance.
  - <no parameter>
  - **vrf** *vrf_name*

Display Values

- ISIS Instance
- LSPID
- Seq Num
- Cksum
- Life
- IS

Examples

- This command displays general information about the link state database of IS-IS.

```
switch>show isis database

ISIS Instance: Osiris
  ISIS Level 2 Link State Database
    LSPID                 Seq Num   Cksum  Life  IS Flags
    1212.1212.1212.00-00  4         714    1064  L2 <>
    1212.1212.1212.0a-00  1         57417  1064  L2 <>
    2222.2222.2222.00-00  6         15323  1116  L2 <>
    2727.2727.2727.00-00  10        15596  1050  L2 <>
    3030.3030.3030.00-00  12        62023  1104  L2 <>
    3030.3030.3030.c7-00  4         53510  1104  L2 <>
switch>
```

- This command displays detailed information about the link state database of IS-IS.

```
switch>show isis database detail

ISIS Instance: Osiris
  ISIS Level 2 Link State Database
    LSPID                 Seq Num   Cksum  Life  IS Flags
    1212.1212.1212.00-00  4         714    1060  L2 <>
      Area address: 49.0001
      Interface address: 10.1.1.2
      Interface address: 2002::2
      IS Neighbor:   1212.1212.1212.0a Metric: 10
      Reachability: 10.1.1.0/24 Metric: 10 Type: 1
      Reachability: 2002::/64 Metric: 10 Type: 1
    1212.1212.1212.0a-00  1         57417  1060  L2 <>
      IS Neighbor:   2727.2727.2727.00 Metric: 0
      IS Neighbor:   2222.2222.2222.00 Metric: 0
      IS Neighbor:   1212.1212.1212.00 Metric: 0
    2222.2222.2222.00-00  6         15323  1112  L2 <>
      Area address: 49.0001
      Interface address: 10.1.1.1
      Interface address: 10.1.1.3
      Interface address: 2002::3
      IS Neighbor:   1212.1212.1212.0a Metric: 10
      Reachability: 10.1.1.0/24 Metric: 10 Type: 1
      Reachability: 10.1.1.0/24 Metric: 10 Type: 1
      Reachability: 2002::/64 Metric: 10 Type: 1
    2727.2727.2727.00-00  10        15596  1046  L2 <>
      Area address: 49.0001
      Interface address: 10.1.1.1
      Interface address: 30.1.1.1
      Interface address: 2002::1
      Interface address: 2001::1
      IS Neighbor:   1212.1212.1212.0a Metric: 10
      IS Neighbor:   3030.3030.3030.c7 Metric: 10
      Reachability: 10.1.1.0/24 Metric: 10 Type: 1
      Reachability: 30.1.1.0/24 Metric: 10 Type: 1
      Reachability: 2002::/64 Metric: 10 Type: 1
      Reachability: 2001::/64 Metric: 10 Type: 1
    3030.3030.3030.00-00  12        62023  1100  L2 <>
      Area address: 49.0001
      Interface address: 30.1.1.2
      Interface address: 2001::2
      IS Neighbor:   3030.3030.3030.c7 Metric: 10
      Reachability: 12.1.1.0/24 Metric: 1 Type: 1
      Reachability: 110.1.1.0/24 Metric: 0 Type: 1
      Reachability: 30.1.1.0/24 Metric: 10 Type: 1
      Reachability: 2001::/64 Metric: 10 Type: 1
    3030.3030.3030.c7-00  4         53510  1100  L2 <>
      IS Neighbor:   2727.2727.2727.00 Metric: 0
      IS Neighbor:   3030.3030.3030.00 Metric: 0
switch>
```

show isis hostname

The **show isis hostname** command displays mapping between the System ID and IS-IS hostname.

Command Mode

EXEC

Command Syntax

show isis hostname

Examples

- This command mapping between the System ID and IS-IS hostnames host1 and host2.

```
switch>show isis hostname
ISIS Instance: 1 VRF: default
Level System ID Hostname
L1 1111.1111.1001 host1
L1 1111.1111.1002 host2
```

show isis interface

The **show isis interface** command displays interface information for the IS-IS instance.

Command Mode

EXEC

Command Syntax

```
show isis interface [INSTANCES] [INTERFACE_NAME] [INFO_LEVEL]
show isis interface [INTERFACE_NAME] [INFO_LEVEL] VRF_INSTANCE
```

Parameters

- ***INSTANCES*** Options include:
  - <no parameter>
  - *instance_name*
- ***INTERFACE_NAME*** Values include
  - <no parameter> all interfaces.
  - **ethernet** *e_num* Ethernet interface specified by *e_num*.
  - **loopback** *l_num* Loopback interface specified by *l_num*.
  - **management** *m_num* Management interface specified by *m_num*.
  - **port-channel** *p_num* Port channel interface specified by *p_num*.
  - **vlan** *v_num* VLAN interface specified by *v_num*.
  - **vxlan** *vx_num* VXLAN interface specified by *vx_num*.
- ***INFO_LEVEL*** Options include:
  - <no parameter>
  - **detail**
- ***VRF_INSTANCE*** specifies the VRF instance.
  - <no parameter>
  - **vrf** *vrf_name*

Display Values

- ISIS Instance
- System ID
- Index
- MTU
- Metric
- LAN-ID
- DIS
- Type
- Interface
- SNPA
- State
- Hold time

Examples

- This command displays general IS-IS information for instance Osiris.

```
switch>show isis interface

ISIS Instance: Osiris
  Interface Vlan20:
    Index: 59 SNPA: 0:1c:73:c:5:7f
    MTU: 1497 Type: broadcast
    Level 2:
      Metric: 10, Number of adjacencies: 2
      LAN-ID: 1212.1212.1212, Priority: 64
      DIS: 1212.1212.1212, DIS Priority: 64
  Interface Ethernet30:
    Index: 36 SNPA: 0:1c:73:c:5:7f
    MTU: 1497 Type: broadcast
    Level 2:
      Metric: 10, Number of adjacencies: 1
      LAN-ID: 3030.3030.3030, Priority: 64
                  DIS: 3030.3030.3030, DIS Priority: 64
```

- This command displays detailed IS-IS information for instance Osiris.

```
switch>show isis interface detail

ISIS Instance: Osiris
  Interface Vlan20:
    Index: 59 SNPA: 0:1c:73:c:5:7f
    MTU: 1497 Type: broadcast
    Level 2:
      Metric: 10, Number of adjacencies: 2
      LAN-ID: 1212.1212.1212, Priority: 64
      DIS: 1212.1212.1212, DIS Priority: 64
    Adjacency 2222.2222.2222:
      State: UP, Level: 2 Type: Level 2 IS
      Hold Time: 30, Supported Protocols: ipv4, ipv6
      SNPA: 2:1:0:c:0:0, Priority: 64
      IPv4 Interface Address: 10.1.1.3
      IPv6 Interface Address: fe80::1:ff:fe0c:0
      Areas:
        49.0001
    Adjacency 1212.1212.1212:
      State: UP, Level: 2 Type: Level 2 IS
      Hold Time: 9, Supported Protocols: ipv4, ipv6
      SNPA: 2:1:0:d:0:0, Priority: 64
      IPv4 Interface Address: 10.1.1.2
      IPv6 Interface Address: fe80::1:ff:fe0d:0
      Areas:
        49.0001
  Interface Ethernet30:
    Index: 36 SNPA: 0:1c:73:c:5:7f
    MTU: 1497 Type: broadcast
    Level 2:
      Metric: 10, Number of adjacencies: 1
      LAN-ID: 3030.3030.3030, Priority: 64
      DIS: 3030.3030.3030, DIS Priority: 64
    Adjacency 3030.3030.3030:
      State: UP, Level: 2 Type: Level 2 IS
      Hold Time: 9, Supported Protocols: ipv4, ipv6
      SNPA: 2:1:0:b:0:0, Priority: 64
      IPv4 Interface Address: 30.1.1.2
      IPv6 Interface Address: fe80::1:ff:fe0b:0
      Areas:
        49.0001
```

show isis neighbors

The **show isis neighbors** command displays IS-IS information.

Command Mode

EXEC

Command Syntax

```
show isis neighbors [INSTANCES] [INFO_LEVEL]
show isis neighbors [INFO_LEVEL] VRF_INSTANCE
```

Parameters

- ***INSTANCES*** Options include:
  - <no parameter>
  - *instance_name*
- ***INFO_LEVEL*** Options include:
  - <no parameter>
  - **detail**
- ***VRF_INSTANCE*** specifies the VRF instance.
  - <no parameter>
  - **vrf** *vrf_name*

Display Values

- Inst. ID
- System ID
- Type
- Interface
- SNPA
- State
- Hold time
- Area Address

Examples

- This command displays general information about the IS-IS neighbors.

```
switch(config)#show isis neighbors

Inst Id   System Id          Type Interface      SNPA            State Hold time
10         2222.2222.2222       L2   Vlan20          2:1:0:c:0:0       UP    30
10         1212.1212.1212       L2   Vlan20          2:1:0:d:0:0       UP    9
10         3030.3030.3030       L2   Ethernet30      2:1:0:b:0:0       UP    9
switch(config)#
```

- This command displays detailed information about the IS-IS neighbors.

```
switch(config)#show isis neighbors detail

Inst Id   System Id          Type Interface      SNPA            State Hold time
10        2222.2222.2222       L2   Vlan20          2:1:0:c:0:0       UP    26
  Area Address(es): 49.0001
  SNPA: 2:1:0:c:0:0
  Advertised Hold Time: 30
  State Changed: -
  LAN Priority: 64
  IPv4 Interface Address: 10.1.1.3
  IPv6 Interface Address: fe80::1:ff:fe0c:0
  Interface name: Vlan20
10        1212.1212.1212       L2   Vlan20          2:1:0:d:0:0       UP    7
  Area Address(es): 49.0001
  SNPA: 2:1:0:d:0:0
  Advertised Hold Time: 9
  State Changed: -
  LAN Priority: 64
  IPv4 Interface Address: 10.1.1.2
  IPv6 Interface Address: fe80::1:ff:fe0d:0
  Interface name: Vlan20
switch(config)#
```

show isis summary

The **show isis summary** command displays information for IS-IS instances.

Command Mode

EXEC

Command Syntax

show isis [***INSTANCES***] **summary**
show isis summary ***VRF_INSTANCE***

Parameters

- ***INSTANCES*** Options include:
  - <no parameter>
  - *instance_name*
- ***VRF_INSTANCE*** specifies the VRF instance.
  - <no parameter>
  - **vrf** *vrf_name*

Display Values

- System ID
- Internal Preference
- External Preference
- IS-Type
- Area Addresses
- level 2

Example

- This command displays general information about IS-IS instances.

```
switch>show isis summary
ISIS Instance: Osiris
  System ID: 1010.1040.1030, administratively enabled, attached
  Internal Preference: Level 1: 115, Level 2: 115
  External Preference: Level 1: 115, Level 2: 115
  IS-Type: Level 2, Number active interfaces: 1
  Routes IPv4 only
  Last Level 2 SPF run 2:32 minutes ago
  Area Addresses:
    10.0001
  level 2: number dis interfaces: 1, LSDB size: 1
switch>
```

show isis topology

The **show isis topology** command displays a list of all connected devices in all areas.

Command Mode

EXEC

Command Syntax

```
show isis topology
show isis INSTANCES topology
show isis topology VRF_INSTANCE
```

Parameters

- ***INSTANCES*** Options include:
  - <no parameter>
  - *instance_name*
- ***VRF_INSTANCE*** specifies the VRF instance.
  - <no parameter>
  - **vrf** *vrf_name*

Display Values

- System Id
- Metric
- Next-Hop
- Interface
- SNPA

Examples

- This command displays forwarding state for ports mapped to all VLANs.

```
switch>show isis topology

ISIS Instance: Osiris VRF: default
  ISIS IP paths to level-2 routers
    System Id            Metric   Next-Hop             Interface       SNPA
    00e0.52b5.7800       20       10.110.2.1
            1/7      00e0.22b5.5843
switch>
```

shutdown (IS-IS)

The **shutdown** command disables IS-IS on the switch without modifying the IS-IS configuration.

The **no shutdown** and **default shutdown** commands enable the IS-IS instance by removing the **shutdown** command from ***running-config***.

Command Mode

Router-IS-IS Configuration

Command Syntax

```
shutdown
no shutdown
default shutdown
```

Examples

- These commands disable IS-IS on the switch.

```
switch(config)#router isis Osiris
switch(config-router-isis)#shutdown
switch(config-router-isis)#
```

- This command enables IS-IS on the switch.

```
switch(config)#router isis Osiris
switch(config-router-isis)#no shutdown
switch(config-router-isis)#
```

spf-interval

The **spf-interval** command sets the shortest path first (SPF) timer. The SPF timer defines the interval between IS-IS path calculations. The default is two seconds.

The **no spf-interval** and **default spf-interval** commands restore the default maximum IS-IS path calculation interval to two seconds by removing the **spf-interval** command from ***running-config***.

Command Mode

Router-IS-IS Configuration

Command Syntax

```
spf-interval period
no spf-interval
default spf-interval
```

Parameters

- *period* Value ranges from 1 through 300. Default interval is 2 seconds.

Examples

- These commands configures the SPF interval to 50 seconds.

```
switch(config)#router isis Osiris
switch(config-router-isis)#spf-interval 50
switch(config-router-isis)#
```

- These commands remove the SPF interval.

```
switch(config)#router isis Osiris
switch(config-router-isis)#no spf-interval
switch(config-router-isis)#
```

Chapter 32

Multiprotocol Label Switching (MPLS)

Tunneling protocols encapsulate packets of a different protocol as the payload of a larger frame for delivery within networks utilizing the encapsulating protocol. Tunneling facilitates the delivery of payload over an incompatible delivery network and creates a secure path through an untrusted network. Protocols that this chapter describes include MPLS, Decap Groups, and Nexthop Groups.

Sections in this chapter include:

- Section 32.1: MPLS
- Section 32.2: Decap Groups
- Section 32.3: Nexthop Groups
- Section 32.4: MPLS Command Descriptions

32.1 MPLS

These sections describe the Arista MPLS implementation:

- Section 32.1.1: MPLS Description
- Section 32.1.2: MPLS Configuration

32.1.1 MPLS Description

32.1.1.1 MPLS Overview

Multiprotocol Label Switching (MPLS) is a networking process that replaces complete network addresses with short path labels for directing data packets to network nodes. The labels identify virtual links (paths) between distant nodes rather than endpoints. MPLS is scalable and protocol-independent. Data packets are assigned labels, which are used to determine packet forwarding destinations without examining the packet.

Arista switches utilize MPLS to improve efficiency and control from servers through data centers and to the WAN. The MPLS implementation supports static MPLS tunneling that is manually configured on each switch or established over a network by an SDN controller. The configuration is specified by a set of rules that filter packets based on matching criteria. Each rule applies MPLS-related actions to packets that match the rule's criteria. Each rule includes a metric that the switch uses to select an action when multiple rules match a packet.

32.1.1.2 MPLS Implementation

MPLS static rule parameters contain the following:

- A 20-bit value that is compared to the top header label of each MPLS packet. Other rule parameters may be applied to packets whose top label match this value.
- A nexthop location that specifies the packet's next destination (IPv4 or IPv6) and the interface through which the switch forwards the packet.
- An MPLS label stack management action that is performed on filtered packets:
  - pop-payload: removes the top label from stack; this terminates an LSP (label-switched path).
  - swap-label: replaces top label with a specified new label; this passes a packet along an LSP.
- A rule metric that the switch uses to select a rule when multiple rules match an MPLS packet.

Packets that do not match any MPLS rules are dropped.

32.1.2 MPLS Configuration

MPLS routing is enabled through the **mpls ip** command.

- This command enables MPLS routing.
-

```
switch(config)#mpls ip
switch(config)#show running-config
```

Example

```
mpls ip
!

end
switch(config)#
```

MPLS rules are created by the **mpls static** command. MPLS static rules identify a set of MPLS packets by a common top label and defines the method of handling these packets.

Examples

- These commands create an MPLS rule that matches packets with a top label value of 3400 and causes the removal of the top label from the header stack. The nexthop destination of the IPv4 payload is IP address 10.14.4.4 through Ethernet interface 3/3/3. This rule has a metric value of 100.

```
switch(config)#mpls static top-label 3400 ethernet 3/3/3 10.14.4.4 pop
payload-type ipv4
switch(config)#show running-config

!
mpls static top-label 3400 Ethernet3/3/3 10.14.4.4 pop payload-type ipv4
!

end
switch(config)#
```

- These commands create a backup rule that forwards the packet through Ethernet interface 4/3. This rule's metric value of 150 assigns it backup status prior to the first rule.

```
switch(config)#mpls static top-label 3400 ethernet 4/3 10.14.4.4 pop payload-type
ipv4 metric 150
switch(config)#show running-config

!
mpls static top-label 3400 Ethernet4/3 10.14.4.4 pop payload-type ipv4 metric 150
mpls static top-label 3400 Ethernet3/3/3 10.14.4.4 pop payload-type ipv4
!

end
switch(config)#
```

- These commands create an MPLS rule that forwards the packet to the nexthop address through any interface.

```
switch(config)#mpls static top-label 4400 10.15.46.45 pop payload-type ipv4
switch(config)#show running-config
             <-------OUTPUT OMITTED FROM EXAMPLE-------->
!
mpls static top-label 3400 Ethernet4/3 10.14.4.4 pop payload-type ipv4 metric 150
mpls static top-label 3400 Ethernet3/3/3 10.14.4.4 pop payload-type ipv4
mpls static top-label 4400 10.15.46.45 pop payload-type ipv4
!

end
switch(config)#
```

The switch's MPLS static rule configuration for specified routes and rules is displayed by **show mpls route**.

Example

- This command displays the MPLS rule configuration.

```
switch>show mpls config route
In-Label  Out-Label  Metric  Payload  NextHop
3400      pop        100     ipv4     10.14.4.4,Et3/3/3
3400      pop        150     ipv4     10.14.4.4,Et4/3
switch>
```

Statistics about the configuration and implementation of MPLS rules are displayed by the **show mpls route summary** command.

Example

- This command displays a summary of MPLS rule implementation.

```
switch>show mpls route summary
Number of Labels: 1 (1 unprogrammed)
Number of adjacencies in hardware: 0
Number of backup adjacencies: 2
switch>
```

32.1.2.1 Egress IPv4/IPv6 over MPLS ACLs

IPv4/IPv6 over MPLS packets are now eligible for ACLs at the egress stage by default, applicable only to IPv4/IPv6 over MPLS packets that are MPLS label popped (i.e. if the label is at the bottom of stack). The user can override this behavior if required, thereby disabling egress ACLs for certain MPLS labels by configuration. No special configuration is required to enable egress ACLs on IPv4/IPv6 over MPLS packets.

Example

- This command disables egress ACLs for MPLS top-label 12000 on the egress interface 120.1.1.1 nexthop address.

```
switch(config)#no mpls static top-label 12000 120.1.1.1 pop payload-type ipv6
switch(config)#
```

- This command enables egress ACLs for MPLS top-label 12000 on the egress interface 120.1.1.1 nexthop address.

```
switch(config)#mpls static top-label 12000 120.1.1.1 pop payload-type ipv6
switch(config)#
```

32.2 Decap Groups

These sections describe the Decap groups:

- Section 32.2.1: Decap Groups Description
- Section 32.2.2: Decap Groups Configuration

32.2.1 Decap Groups Description

The decap group is a data structure that receives encapsulated packets and extracts the payload. The switch then processes or forwards the extracted payload as required. Although packets cannot be transmitted through decap groups, nexthop groups can be used to create a packet's reverse path. Decap groups support payload extraction of packets received from GRE and IP-in-IP tunnels.

Decap groups have these limitations:

- Tunnels are terminated using destination IP address; source IP address has no influence.
- Packets matching a decap group are processed through their ingress interface and VLAN
- Ingress ACL filter each decap group packet’s outer header.
- Packet counters are not available.
- VRF is not supported.

Decap groups are defined by their tunnel type and decap IP address:

- **Tunnel type** specifies the tunnel protocol that the switch uses to extract payload.
- **Decap IP address** specifies the IP address where the switch receives decap group packets.

Decap groups support Generic Routing Encapsulation (GRE) tunnels.

Decap groups support Generic Routing Encapsulation (GRE) and IP-in-IP tunnels.

32.2.2 Decap Groups Configuration

Decap groups are configured in decap-group configuration mode. Decap-group configuration mode is not a group change mode; ***running-config*** is changed immediately upon entering commands. Exiting decap-group configuration mode does not affect ***running-config***. The **exit** command returns the switch to global configuration mode.

- The static CLI entry for the incoming label is specified by the **mpls static** command.
- The tunnel type is specified by the **tunnel type (Decap Group)** command.
- The Decap IP address is specified by the **tunnel decap-ip (Decap Group)** command.

Decap groups do not define a default destination address or tunnel type and is not functional until both parameters are configured. A decap group can contain only one **tunnel decap-ip** statement; a subsequent command replaces the previously configured statements.

Example

- This command defines a static CLI entry for the incoming-label.

```
switch(config)# #mpls static top-label 3400 ethernet 3/3/3 10.14.4.4 pop
payload-type ipv4
```

- This command creates a decap group named DC-1 and configures the group to terminate packets from GRE tunnel packets with the destination IP address of 10.14.3.2

```
switch(config)#ip decap-group DC-1
switch(config-dg-DC-1)#tunnel type gre
switch(config-dg-DC-1)#tunnel decap-ip 10.14.3.2
switch(config-dg-DC-1)#show active
 ip decap-group DC-1
   tunnel type gre
   tunnel decap-ip 10.14.3.2
switch(config-dg-DC-1)#end
switch(config)#
```

32.3 Nexthop Groups

32.3.1 Nexthop Group Description

Each routing table entry provides the next hop address to its specified destination. A nexthop address is the address of the next device on the path to the entry's specified destination.

A nexthop group is a data structure that defines a list of nexthop addresses and a tunnel type for packets routed to the specified address. When an IP route statement specifies a nexthop group as the nexthop address, the switch configures a static route with a nexthop group member as the nexthop address and encapsulates packets forwarded to that address as required by the group's tunnel type.

The nexthop group size is a configurable parameter that specifies the number of entries that the group contains. Group entries that are not explicitly configured are filled with drop routes. The switch uses ECMP hashing to select the address within the nexthop group when forwarding packets. When a packet's hash selects a drop route, the packet is dropped.

Nexthop groups are supported on Trident platform switches and subject to the following restrictions:

- Each switch can support 512 IPv4 or IPv6 Tunnels
- Nexthop groups can contain 256 nexthops.
- The switch supports 1024 nexthop groups.
- Multiple routes can share a tunnel.
- Tunnels do not support IP multicast packets.

Nexthop groups support IP-in-IP tunnels. The entry IP address family within a particular nexthop group cannot be mixed, i.e. either they are all IPv4 or they are all IPv6 entries.

32.3.2 Nexthop Group Configuration

Nexthop groups are configured and modified in nexthop-group configuration mode. After a group is created, it is associated to a static route through an **ip route nexthop-group** statement.

These tasks are required to configure a nexthop group and apply it to a static route.

- Creating and Editing Nexthop Groups
- Configuring a Group's Encapsulation Parameters
- Configuring the Group's Size
- Creating Nexthop Group Entries
- Displaying Nexthop Groups
- Applying a Nexthop Group to a Static Route

Creating and Editing Nexthop Groups

Nexthop groups are created by a **nexthop-group** command that specifies a group that isn't already configured. The switch enters nexthop-group configuration mode for the new group. Nexthop-group mode is also accessible for modifying existing groups. When in nexthop-group configuration mode, the **show active** command displays the group's configuration.

Example

- This command creates a nexthop group named NH-1.

```
switch(config)#nexthop-group NH-1
switch(config-nexthop-group-NH-1)#
```

- These commands enter nexthop-group configuration mode for the group named NH3, then displays the previously configured group parameters.

```
switch(config)#nexthop-group NH3
switch(config-nexthop-group-NH3)#show active
 nexthop-group NH3
   size 4
   ttl 10
   entry 0 tunnel-destination 10.14.21.3
   entry 1 tunnel-destination 10.14.21.5
   entry 2 tunnel-destination 10.14.22.5
   entry 3 tunnel-destination 10.14.22.6
switch(config-nexthop-group-NH3)#
```

Configuring a Group's Encapsulation Parameters

Packets in static routes that are associated with the nexthop group are encapsulated to support the group's tunnel type. Nexthop groups support IP-in-IP tunnels. The group also defines the source IP address and TTL field contents that are included in the packet encapsulation.

Example

- This command configures the TTL setting to 32 for nexthop group NH-1 encapsulation packets.

```
switch(config)#nexthop-group NH-1
switch(config-nexthop-group-NH-1)#ttl 32
switch(config-nexthop-group-NH-1)#show active
 nexthop-group NH-1
   size 128
   ttl 32
switch(config-nexthop-group-NH-1)#
```

The address is inserted in the encapsulation source IP fields is specified by **tunnel-source (Nexthop Group).**

Example

- These commands create loopback interface 100, assign an IP address to the interface, then specifies that address as the tunnel source for packets designated by nexthop-group NH-1.

```
switch(config)#interface loopback 100
switch(config-if-Lo100)#ip address 10.1.1.1/32
switch(config-if-Lo100)#exit
switch(config)#nexthop-group NH-1
switch(config-nexthop-group-NH-1)#tunnel-source intf loopback 100
switch(config-nexthop-group-NH-1)#show active
 nexthop-group NH-1
   size 256
   ttl 32
   tunnel-source intf Loopback100
switch(config-nexthop-group-NH-1)#
```

Configuring IP-in-IP Encapsulation

Through IP-in-IP encapsulation, IP packets matching a static Nexthop-Group route are encapsulated within an IP-in-IP tunnel and forwarded.

Example

- This command configures a static Nexthop-Group route and an IP-in-IP Nexthop-Group for IP-in-IP encapsulation.

```
switch(config)#ip route 124.0.0.1/32 nexthop-group abc
switch(config)#nexthop-group abc type ip-in-ip
switch(config-nexthop-group-abc)#size 512
switch(config-nexthop-group-abc)#tunnel-source 1.1.1.1
switch(config-nexthop-group-abc)#entry 0 tunnel-destination 1.1.1.2
switch(config-nexthop-group-abc)#entry 1 tunnel-destination 10.1.1.1
switch(config-nexthop-group-abc)#ttl 64
switch(config-nexthop-group-abc)#
```

Configuring the Group's Size

The group's size specifies the number of entries in the group. A group can contain up to 256 entries, which is the default size. The group's size is specified by **size (Nexthop Group).**

Example

- This command configures the nexthop group NH-1 to contain 128 entries.

```
switch(config)#nexthop-group NH-1
switch(config-nexthop-group-NH-1)#size 128
switch(config-nexthop-group-NH-1)#show active
 nexthop-group NH-1
   size 128
   ttl 64
switch(config-nexthop-group-NH-1)#
```

Creating Nexthop Group Entries

Each entry specifies a nexthop address that is used to forward packets. A nexthop group contains one entry statement for each nexthop address. The group's size specifies the number of entry statements the group may contain. Each entry statement is assigned an index number to distinguish it from other entries within the group; entry index numbers range from zero to the group size minus one.

Nexthop group entries are configured by **entry (Nexthop Group).**

Example

- These commands set the nexthop group size at four entries, then create three entries. Packets that are hashed to the fourth entry are dropped.

```
switch(config)#nexthop-group NH-1
switch(config-nexthop-group-NH-1)#size 4
switch(config-nexthop-group-NH-1)#entry 0 tunnel-destination 10.13.4.4
switch(config-nexthop-group-NH-1)#entry 1 tunnel-destination 10.15.4.22
switch(config-nexthop-group-NH-1)#entry 2 tunnel-destination 10.15.5.37
switch(config-nexthop-group-NH-1)#show active
 nexthop-group NH-1
   size 4
   ttl 64
   entry 0 tunnel-destination 10.13.4.4
   entry 1 tunnel-destination 10.15.4.22
   entry 2 tunnel-destination 10.15.5.37
switch(config-nexthop-group-NH-1)#
```

- These commands configure a nexthop group with three IPv6 nexthop entries.

```
switch(config)#nexthop-group nhg-v6-mpls type ip
switch(config-nhg-v6-mpls)#size 3
switch(config-nhg-v6-mpls)#entry 0 nexthop 2002::6401:1
switch(config-nhg-v6-mpls)#entry 1 nexthop 2002::6404:1
switch(config-nhg-v6-mpls)#entry 2 nexthop 2002::6404:2
switch(config-nhg-v6-mpls)#
```

- These commands configure an IPv4 route to point to the nexthop group `nhg-v6-mpls`. (Both IPv4 routes and IPv6 routes can point to this nexthop group.)

```
switch#ip route 100.5.0.0/16 Nexthop-Group nhg-v6-mplsp
switch#
```

Displaying Nexthop Groups

The **show nexthop-group** command displays a group's configured parameters.

Example

- This command displays the properties of the nexthop group named NH-1.

```
switch>show nexthop-group NH-1
Name             Id       type     size   ttl    sourceIp
NH-1             4        ipInIp   256    64     0.0.0.0
switch>
```

Applying a Nexthop Group to a Static Route

The **ip route nexthop-group** associates a nexthop group with a specified destination address and configures the encapsulation method for packets tunneled to that address.

Example

- This command creates a static route in the default VRF, using the nexthop group of NH-1 to determine the next hop address.

```
switch(config)#ip route 10.17.252.0/24 nexthop-group NH-1
switch(config)#
```

The **show ip route** command displays the routing table for a specified VRF. Routes that utilize a nexthop group entry are noted with a route type code of NG.

Example

- This command displays a routing table that contains a static route with its nexthop specified by a nexthop group.

```
switch>show ip route
Codes: C - connected, S - static, K - kernel,
       O - OSPF, IA - OSPF inter area, E1 - OSPF external type 1,
       E2 - OSPF external type 2, N1 - OSPF NSSA external type 1,
       N2 - OSPF NSSA external type2, B I - iBGP, B E - eBGP,
       R - RIP, I - ISIS, A B - BGP Aggregate, A O - OSPF Summary,
       NG - Nexthop Group Static Route

Gateway of last resort is not set

 C      10.3.3.1/32 is directly connected, Loopback0
 C      10.9.1.0/24 is directly connected, Ethernet51/3
 C      10.10.10.0/24 is directly connected, Ethernet51/1
 S      10.20.0.0/16 [20/0] via 10.10.10.13, Ethernet51/1
 C      10.10.11.0/24 is directly connected, Ethernet3
 NG     10.10.3.0/24 [1/0] via ng-test1, 5
 C      10.17.0.0/20 is directly connected, Management1
 S      10.17.0.0/16 [1/0] via 10.17.0.1, Management1
 S      10.18.0.0/16 [1/0] via 10.17.0.1, Management1
 S      10.19.0.0/16 [1/0] via 10.17.0.1, Management1
 S      10.20.0.0/16 [1/0] via 10.17.0.1, Management1
 S      10.22.0.0/16 [1/0] via 10.17.0.1, Management1

switch>
```

32.4 MPLS Command Descriptions

MPLS Commands

- mpls ip
- mpls static
- show mpls route
- show mpls route summary

Decap Group Commands

- ip decap-group
- tunnel decap-ip (Decap Group)
- tunnel type (Decap Group)

Nexthop Group Commands

- entry (Nexthop Group)
- ip route nexthop-group
- nexthop-group
- show nexthop-group
- size (Nexthop Group)
- ttl (Nexthop Group)
- tunnel-source (Nexthop Group)

entry (Nexthop Group)

The **entry** command defines a nexthop entry in the configuration mode nexthop group. Each nexthop entry specifies a nexthop IP address for static routes to which the nexthop group is assigned. The group size (**size (Nexthop Group)**) specifies the quantity of entries a group contains. Each entry is created by an individual command. Entries within a group are distinguished by an index number.

The **no entry** and **default entry** commands delete the specified nexthop group entry, as referenced by index number, by removing the corresponding **entry** statement from ***running-config.***

Command Mode

Nexthop-group Configuration

Command Syntax

```
entry index tunnel-destination ipv4_address
no entry index
default entry index
```

Parameters

- *index* Entry index. Values range from ***0*** to ***group-size – 1***.
- *ipv4_address* Nexthop IPv4 address.

group-size is the group's entry capacity, as specified by the **size (Nexthop Group)** command.

Example

- These commands sets the nexthop group size at 4 entries, then creates three nexthop entries. Packets that are hashed to the fourth entry are dropped.

```
switch(config)#nexthop-group NH-1
switch(config-nexthop-group-NH-1)#size 4
switch(config-nexthop-group-NH-1)#entry 0 tunnel-destination 10.13.4.4
switch(config-nexthop-group-NH-1)#entry 1 tunnel-destination 10.15.4.22
switch(config-nexthop-group-NH-1)#entry 2 tunnel-destination 10.15.5.37
switch(config-nexthop-group-NH-1)#show active
 nexthop-group NH-1
   size 4
   ttl 64
   entry 0 tunnel-destination 10.13.4.4
   entry 1 tunnel-destination 10.15.4.22
   entry 2 tunnel-destination 10.15.5.37
switch(config-nexthop-group-NH-1)#
```

ip decap-group

The **ip decap-group** command places the switch in decap-group configuration mode, through which decap groups are created or modified. A decap group is a data structure that defines a method of extracting the payload from an encapsulated packet that the switch receives on a specified IP address.

Decap groups do not specify a default IP address group or tunnel type. These parameters must be explicitly configured before a decap group can function.

Decap-group configuration mode is not a group change mode; ***running-config*** is changed immediately upon entering commands. Exiting decap-group configuration mode does not affect ***running-config***. The **exit** command returns the switch to global configuration mode.

The **no ip decap-group** and **default ip decap-group** commands delete previously configured commands in the specified **decap-group** mode.

Command Mode

Global Configuration

Command Syntax

```
ip decap-group group_name
no ip decap-group group_name
default ip decap-group group_name
```

Parameters

- *group_name* Decap group name.

Commands Available in Decap-group Configuration Mode

- **tunnel decap-ip (Decap Group)** specifies the IP address of packets handled by the decap group.
- **tunnel type (Decap Group)** specifies the tunnel protocol for extracting payload.

Examples

- This command creates a decap group named DC-1.

```
switch(config)#ip decap-group DC-1
switch(config-dg-DC-1)#
```

- This command exits decap-group mode for the DC-1 decap group.

```
switch(config-dg-DC-1)#exit
switch(config)#
```

- This command delete the decap group named DC-1.

```
switch(config)#no ip decap-group DC-1
switch(config)#
```

ip route nexthop-group

The **ip route nexthop-group** command creates a static route. The destination is a network segment. The nexthop address is one of the IP addresses that comprise the specified nexthop group. Packets forwarded as a result of this command are encapsulated as specified by the tunnel-type parameter of the specified nexthop group.

When multiple routes exist to a destination prefix, the route with the lowest administrative distance takes precedence. When a route created through this command has the same administrative distance as another static route (ECMP), the route that was created earliest has preference; ***running-config*** stores static routes in the order that they are created.

By default, the administrative distance assigned to static routes is 1. Assigning a higher administrative distance to a static route configures it to be overridden by dynamic routing data. For example, a static route with a distance value of 200 is overridden by OSPF intra-area routes, which have a default distance of 110.

The **no ip route nexthop-group** and **default ip route nexthop-group** commands delete the specified route by removing the corresponding **ip route nexthop-group** command from ***running-config***. **Ip route nexthop-group** statements for an IP address in multiple VRFs must be removed separately.

A **no ip route** or **default ip route** command without a nexthop parameter deletes all corresponding **ip route nexthop-group** statements. Deleting a user-defined VRF also deletes its static routes.

Command Mode

Global Configuration

Command Syntax

```
ip route [VRF_INST] dest_net nexthop-group nhgp_name [dist] [TAG_OPTION]
[RT_NAME]
no ip route [VRF_INST] dest_net [nexthop-group nhgroup_name] [distance]
default ip route [VRF_INST] dest_net [nexthop-group nhgroup_name] [distance]
```

Parameters

- ***VRF_INST*** Specifies the VRF instance being modified.
  - <no parameter> Changes are made to the default VRF.
  - **vrf** *vrf_name* Changes are made to the specified VRF.
- *dest_net* Destination IPv4 subnet (CIDR or address-mask notation).
- *nhgp_name* Name of nexthop group.
- *dist* Administrative distance assigned to route. Options include:
  - <no parameter> Route assigned default administrative distance of one.
  - <1-255> The administrative distance assigned to route.
- ***TAG_OPTION*** static route tag. Options include:
  - <no parameter> Assigns default static route tag of 0.
  - **tag** *t_value* Static route tag value. *t_value* ranges from ***0*** to ***4294967295***.
- ***RT_NAME*** Associates descriptive text to the route. Options include:
  - <no parameter> No text is associated with the route.
  - **name** *descriptive_text* The specified text is assigned to the route.

Related Commands

- **ip route** creates a static route that specifies the nexthop address without using nexthop groups.

Example

- This command creates a static route in the default VRF, using the nexthop group of NH-1 to determine the next hop address.

```
switch(config)#ip route 10.17.252.0/24 nexthop-group NH-1
switch(config)#
```

mpls ip

The **mpls ip** command enables MPLS routing. Multiprotocol Label Switching (MPLS) is a networking process that avoids complex lookups in a routing table by replacing complete network addresses with short path labels for directing data packets to network nodes. MPLS data paths are serviced through a tunnel encapsulation data structure that adds four-byte label headers to packets.

The **no mpls ip** and **default mpls ip** commands disable MPLS routing by removing the **mpls ip** command from ***running-config***. When MPLS routing is disabled, routed MPLS packets are dropped and all MPLS routes and adjacencies are removed. MPLS routing is disabled by default.

Command Mode

Global Configuration

Command Syntax

```
mpls ip
no mpls ip
default mpls ip
```

Example

- This command enables MPLS routing. Previous commands enabled IP routing and configured MPLS static routes.

```
switch(config)#mpls ip
switch(config)#show running-config
! Command: show running-config

!
ip routing
!
mpls ip
!
mpls static top-label 3400 10.14.4.4 pop payload-type ipv4
mpls static top-label 4400 10.15.46.45 pop payload-type ipv4
!

!
end
switch(config)#
```

- This command disables MPLS routing.

```
switch(config)#no mpls ip
switch(config)#show running-config
! Command: show running-config
              <-------OUTPUT OMITTED FROM EXAMPLE-------->
!
ip routing
!
mpls static top-label 3400 10.14.4.4 pop payload-type ipv4
mpls static top-label 4400 10.15.46.45 pop payload-type ipv4
!

!
end
switch(config)#
```

mpls static

The **mpls static** command creates an MPLS rule that specifies the method of handling of inbound MPLS traffic. Multiprotocol Label Switching (MPLS) is a networking process that replaces complete network addresses with short path labels for directing data packets to network nodes.

Static rules specify these parameters:

- MPLS filter: The top-label parameter specifies the 20-bit value that the MPLS packet's top header label must match to be handled by the rule.
- Nexthop location: Specifies the destination nexthop address (IPv4 or IPv6) and the interface through which the switch forwards the packet.
- MPLS action: Specifies the MPLS label stack management action performed on the packet:
  - pop-payload: removes the top label from stack; this terminates an LSP (label-switched path).
  - swap-label: replaces top label with a specifed new label; this passes a packet along an LSP.
- Rule priority: Specifies the rule to be used when an MPLS packet matches multiple rules.

The **no mpls static** and **default mpls static** commands delete the specified MPLS rule from ***running-config***.

- Commands that include only a top label tag remove all MPLS rules with the matching top label.
- Commands with no ***PRIORITY*** parameter remove all matching routes of every metric value.

Command Mode

Global Configuration

Command Syntax

```
mpls static top-label top_tag [DEST_INTF] NEXTHOP_ADDR ACTION [PRIORITY]
no mpls static top-label top_tag
no mpls static top-label top_tag [DEST_INTF] NEXTHOP_ADDR ACTION [PRIORITY]
default mpls static top-label top_tag
default mpls static top-label top_tag [DEST_INTF] NEXTHOP_ADDR ACTION [PRIORITY]
```

Parameters

- *top_tag* Top header's label field contents. Value ranges from ***0*** to ***1048575*** (20 bits).
- ***DEST_INTF*** Specifies interface through which ***NEXTHOP_ADDR*** is accessed. Options include:
  - <no parameter> Any interface.
  - **ethernet** *e_num* Ethernet interface specified by *e_num*.
  - **loopback** *l_num* Loopback interface specified by *l_num*.
  - **management** *m_num* Management interface specified by *m_num*.
  - **port-channel** *p_num* Port-channel interface specified by *p_num*.
  - **vlan** *v_num* VLAN interface specified by *v_num*.
  - **vxlan** *vx_num* VXLAN interface specified by *vx_num*.
- ***NEXTHOP_ADDR*** Nexthop address for MPLS for filtered MPLS packets. Options include:
  - *ipv4_addr* IPv4 address.
  - *ipv6_addr* IPv6 address.
- ***ACTION*** MPLS header stack management action performed on packet. Options include:
  - **pop payload-type ipv4** Removes top layer from stack. Payload is handled as IPv4 packet.
  - **pop payload-type ipv6** Removes top layer from stack. Payload is handled as IPv6 packet.

- **swap-label** <***0*** to ***1048575***> Replaces header label with specified label value (20 bits).

- ***PRIORITY*** Specifies rule priority when multiple rules match a packet. Options include:
  - <no parameter> Assigns a metric value of 100 to the rule.
  - **metric** <***1*** to ***255***> Lower values denote higher priority. Value ranges from ***1*** to ***255***.

Parameters

The mpls static command does not support push label actions.

Example

- These commands create an MPLS rule that matches packets with a top label value of 3400 and causes the removal of the top label from the header stack. The nexthop destination of the IPv4 payload is IP address 10.14.4.4 through Ethernet interface 3/3/3. This rule has a metric value of 100.

```
switch(config)#mpls static top-label 3400 ethernet 3/3/3 10.14.4.4 pop
payload-type ipv4
switch(config)#show running-config

!
mpls static top-label 3400 Ethernet3/3/3 10.14.4.4 pop payload-type ipv4
!

end
switch(config)#
```

- These commands create a backup rule that forwards the packet through Ethernet interface 4/3. This rule's metric value of 150 assigns it backup status prior to the first rule.

```
switch(config)#mpls static top-label 3400 ethernet 4/3 10.14.4.4 pop payload-type
ipv4 metric 150
switch(config)#show running-config

!
mpls static top-label 3400 Ethernet4/3 10.14.4.4 pop payload-type ipv4 metric 150
mpls static top-label 3400 Ethernet3/3/3 10.14.4.4 pop payload-type ipv4
!
             <-------OUTPUT OMITTED FROM EXAMPLE-------->
end
switch(config)#
```

- These commands create an MPLS rule that forwards the packet to the nexthop address through any interface.

```
switch(config)#mpls static top-label 4400 10.15.46.45 pop payload-type ipv4
switch(config)#show running-config
             <-------OUTPUT OMITTED FROM EXAMPLE-------->
!
mpls static top-label 3400 Ethernet4/3 10.14.4.4 pop payload-type ipv4 metric 150
mpls static top-label 3400 Ethernet3/3/3 10.14.4.4 pop payload-type ipv4
mpls static top-label 4400 10.15.46.45 pop payload-type ipv4
!

end
switch(config)#
```

nexthop-group

The **nexthop-group** command places the switch in nexthop-group configuration mode, through which nexthop groups are created or modified. The command also specifies the tunnel protocol for extracting payload from encapsulated packets that arrive through an IP address upon which the group is applied.

A nexthop group is a data structure that defines a list of nexthop addresses and the encapsulation process for packets routed to the specified address. The command either accesses an existing nexthop group configuration or creates a new group if it specifies a non-existent group. Supported tunnel protocols include IP ECMP and IP-in-IP.

Nexthop-group configuration mode is not a group change mode; ***running-config*** is changed immediately upon entering commands. Exiting nexthop-group configuration mode does not affect ***running-config***. The **exit** command returns the switch to global configuration mode.

The **no nexthop-group** and **default nexthop-group** commands delete previously configured commands in the specified **nexthop-group** mode. When the command does not specify a group, it removes all nexthop-groups. When the command specifies a tunnel type without naming a group, it removes all nexthop-groups of the specified type.

Command Mode

Global Configuration

Command Syntax

```
nexthop-group group_name type TUNNEL_TYPE
no nexthop-group [group_name] [type TUNNEL_TYPE]
default nexthop-group [group_name] [type TUNNEL_TYPE]
```

Parameters

- *group_name* Nexthop group name.
- ***TUNNEL_TYPE*** Tunnel protocol of the nexthop-group. Options include:
  - **ip** ECMP nexthop.
  - **ip-in-ip** IP in IP tunnel.

Commands Available in Nexthop-group Configuration Mode

- **entry (Nexthop Group)**
- **size (Nexthop Group)**
- **ttl (Nexthop Group)**
- **tunnel-source (Nexthop Group)**

Restrictions

Tunnel type availability varies by switch platform.

Examples

- This command creates a nexthop group named NH-1 that specifies ECMP nexthops.

```
switch(config)#nexthop-group NH-1 type ip
switch(config-nexthop-group-NH-1)#
```

- This command exits nexthop-group mode for the NH-1 nexthop group.

```
switch(config-nexthop-group-NH-1)#exit
switch(config)#
```

show mpls route

The **show mpls config route** command displays the switch's MPLS static rule configuration for the specified routes and rules.

Command Mode

EXEC

Command Syntax

show mpls [*INFO_LEVEL*] route [*header_label*]

Parameters

- ***INFO_LEVEL*** Specifies the filters that are used to select the routes to display. Options include:
  - <no parameter> displays routes published by the forwarding agent.
  - **config** displays all configured routes.
  - **lfib** displays routes stored to the Label Forwarding Information Base (LFIB)
- *header_label* Filters routes by MPLS top header label. Options include:
  - <no parameter> Displays routes for all header values.
  - <***0*** to ***1048575***> Specifies header for which command displays information.

Example

- This command displays the MPLS rule configuration.

```
switch>show mpls config route
In-Label  Out-Label  Metric  Payload  NextHop
3400      pop        100     ipv4     10.14.4.4,Et3/3/3
3400      pop        150     ipv4     10.14.4.4,Et4/3
switch>
```

show mpls route summary

The **show mpls route summary** command displays statistics about the configuration and implementation of MPLS rules.

Command Mode

EXEC

Command Syntax

show mpls route summary

Example

- This command displays a summary of MPLS rule implementation.

```
switch>show mpls route summary
Number of Labels: 1 (1 unprogrammed)
Number of adjacencies in hardware: 0
Number of backup adjacencies: 2
switch>
```

show nexthop-group

The **show nexthop-group** command displays properties of the specified nexthop group.

Command Mode

EXEC

Command Syntax

show *nhgroup_name* [***VRF_INST***]

Parameters

- *nhgroup_name* Name of the group displayed by command.
- ***VRF_INST*** specifies the VRF instance for which data is displayed.
  - <no parameter> context-active VRF.
  - **vrf** *vrf_name* specifies name of VRF instance. System default VRF is specified by **default**.

Related Commands

- **nexthop-group** places the switch in nexthop-group configuration mode to create a new group or modify an existing group.

Example

- This command displays the properties of the nexthop group named NH-1.

```
switch>show nexthop-group NH-1
Name             Id       type     size   ttl    sourceIp
NH-1             4        ipInIp   256    64     0.0.0.0
switch>
```

size (Nexthop Group)

The **size** command configures the quantity of nexthop entries in the configuration mode nexthop group. Each entry specifies a nexthop IP address for static routes to which the group is assigned. Entries are configured with the **entry (Nexthop Group)** command. The default size is 256 entries.

The **no size** and **default size** commands restore the size of the configuration mode nexthop group to its default of 256 by removing the corresponding **size** command from ***running-config***.

Command Mode

Nexthop-group Configuration

Command Syntax

```
size entry_size
no size entry_size
default size entry_size
```

Parameters

- *entry_size* Group size (entries). Value ranges from 1 to 255. Default value is 256.

Example

- This command configures the nexthop group NH-1 to contain 128 entries.

```
switch(config)#nexthop-group NH-1
switch(config-nexthop-group-NH-1)#size 128
switch(config-nexthop-group-NH-1)#show active
 nexthop-group NH-1
   size 128
   ttl 64
switch(config-nexthop-group-NH-1)#
```

ttl (Nexthop Group)

The **ttl** command specifies the number entered into the TTL (time to live) encapsulation field of packets that are transmitted to the address designated by the configuration mode nexthop group. The default TTL value is 64.

The **no ttl** and **default ttl** commands restore the default TTL value written into TTL fields for the configuration mode nexthop group by deleting the corresponding **ttl** command from ***running-config***.

Command Mode

Nexthop-group Configuration

Command Syntax

```
ttl hop_expiry
no ttl hop_expiry
default ttl hop_expiry
```

Parameters

- *hop_expiry* Period that the packet remains valid (seconds or hops) Value ranges from 1 to 64.

Restrictions

This command is available only to Nexthop groups for tunnels of type ***IP-in-IP***, ***GRE***, ***MPLS***, and ***MPLS over GRE***.

Related Commands

- **nexthop-group** places the switch in Nexthop-group configuration mode.

Example

- This command configures the ttl setting to 32 for nexthop group NH-1 packets.

```
switch(config)#nexthop-group NH-1
switch(config-nexthop-group-NH-1)#ttl 32
switch(config-nexthop-group-NH-1)#show active
 nexthop-group NH-1
   size 128
   ttl 32
switch(config-nexthop-group-NH-1)#
```

- This command restores the default ttl setting for nexthop group NH-1 packets.

```
switch(config-nexthop-group-NH-1)#no ttl
switch(config-nexthop-group-NH-1)#show active
 nexthop-group NH-1
   size 128
   ttl 64
switch(config-nexthop-group-NH-1)#
```

tunnel decap-ip (Decap Group)

The **tunnel decap-ip** command specifies the IP address of packets that are handled by the configuration mode decap group. A decap group is a data structure that defines a method of extracting the payload from an encapsulated packet that the switch receives on a specified IP address.

Decap groups do not define a default decap-ip address. A decap group is not functional until an IP address is specified. Decap groups can contain only one tunnel decap-ip statement; subsequent commands replace any previously configured statements.

Command Mode

Decap-Group Configuration

Command Syntax

tunnel decap-ip *ipv4_address*

Parameters

- *ipv4_addr* An IPv4 address.

Related Commands

- **ip decap-group** places the switch in decap-group configuration mode.
- **tunnel type (Decap Group)** specifies the tunnel protocol for extracting payload.

Guidelines

A decap group does not specify a default IP address group or tunnel type. These parameters must be explicitly configured before a decap group can function.

Example

- These commands configure 10.14.3.2 as the decap-IP address for the DC-1 decap group.

```
switch(config)#ip decap-group DC-1
switch(config-dg-DC-1)#tunnel decap-ip 10.14.3.2
switch(config-dg-DC-1)#show active
 ip decap-group DC-1
   tunnel decap-ip 10.14.3.2
switch(config-dg-DC-1)#
```

tunnel-source (Nexthop Group)

The **tunnel-source** command specifies the address that is entered into the source IP address encapsulation field of packets that are transmitted as designated by the configuration mode nexthop group. The command may directly specify an IP address or specify an interface from which an IP address is derived. The default source address IP address is 0.0.0.0.

The **no tunnel-source** and **default tunnel-source** commands remove the source IP address setting from the configuration mode nexthop group by deleting the **tunnel-source** command from ***running-config***.

Command Mode

Nexthop-group Configuration

Command Syntax

```
tunnel-source SOURCE
no tunnel-source SOURCE
default tunnel-source SOURCE
```

Parameters

- ***SOURCE*** IP address or derivation interface. Options include:
  - *ipv4_addr* An IPv4 address.
  - **intf ethernet** *e_num* Ethernet interface specified by *e_num*.
  - **intf loopback** *l_num* Loopback interface specified by *l_num*.
  - **intf management** *m_num* Management interface specified by *m_num*.
  - **intf port-channel** *p_num* Port-channel interface specified by *p_num*.
  - **intf vlan** *v_num* VLAN interface specified by *v_num*.

Restrictions

This command is available only to Nexthop groups for tunnels of type ***ip-in-ip***.

Related Commands

- **nexthop-group** places the switch in Nexthop-group configuration mode.

Example

- These commands create loopback interface 100, assign an IP address to the interface, then specifies that address as the tunnel source for packets designated by nexthop-group NH-1.

```
switch(config)#interface loopback 100
switch(config-if-Lo100)#ip address 10.1.1.1/32
switch(config-if-Lo100)#exit
switch(config)#nexthop-group NH-1
switch(config-nexthop-group-NH-1)#tunnel-source intf loopback 100
switch(config-nexthop-group-NH-1)#show active
 nexthop-group NH-1
   size 256
   ttl 64
   tunnel-source intf Loopback100
switch(config-nexthop-group-NH-1)#show nexthop-group NH-1
Name             Id       type     size   ttl    sourceIp
NH-1             2        ipInIp   256    64     10.1.1.1
switch(config-nexthop-group-NH-1)#
```

tunnel type (Decap Group)

The **tunnel type** command specifies the tunnel protocol for extracting payload from encapsulated packets that arrive on the IP address specified for the configuration mode decap group. Supported tunnel protocols include GRE (General Routing Encapsulation) and IP-in-IP.

Decap groups do not define a default tunnel type. A decap group is not functional until an IP address is specified. Decap groups can contain only one tunnel decap-ip statement; subsequent commands replace any previously configured statements.

Command Mode

Decap-group Configuration

Command Syntax

```
tunnel type gre
```

Related Commands

- **ip decap-group** places the switch in decap-group configuration mode.
- **tunnel decap-ip (Decap Group)** specifies the IP address of packets handled by the decap group.

Guidelines

A decap group does not specify a default IP address group or tunnel type. These parameters must be explicitly configured before a decap group can function.

Example

- This command configures decap group DC-1 to terminate packets from GRE tunnel packets.

```
switch(config)#ip decap-group DC-1
switch(config-dg-DC-1)#tunnel type gre
switch(config-dg-DC-1)#show active
 ip decap-group DC-1
   tunnel type gre
switch(config-dg-DC-1)#
```

Chapter 33

Bidirectional Forwarding Detection

This chapter describes bidirectional forwarding detection (BFD) and how it is configured in relation to various protocols. Sections in this chapter include:

- Section 33.1: Introduction
- Section 33.2: BFD Configuration
- Section 33.3: BFD Command Descriptions

33.1 Introduction

In networks without data link signaling, connection failures are usually detected by the hello mechanisms of routing protocols. Detection can take over a second, and reducing detection time by increasing the rate at which hello packets are exchanged can create an excessive burden on the participating CPUs.

BFD is a low-overhead, protocol-independent mechanism which adjacent systems can use instead for faster detection of faults in the path between them. BFD is strictly a failure-detection mechanism, and does not discover neighbors or reroute traffic.

BFD is a simple mechanism which detects the liveness of a connection between adjacent systems, allowing it to quickly detect failure of any element in the connection. It does not operate independently, but only as an adjunct to routing protocols. The routing protocols are responsible for neighbor detection, and create BFD sessions with neighbors by requesting failure monitoring from BFD.

Once a BFD session is established with a neighbor, BFD exchanges control packets to verify connectivity and informs the requesting protocol of failure if a specified number of successive packets are not received. The requesting protocol is then responsible for responding to the loss of connectivity.

Routing protocols using BFD for failure detection continue to operate normally when BFD is enabled, including the exchange of hello packets.

The basic behavior of BFD is defined in RFC 5880.

33.1.1 BFD Modes

BFD functions in asynchronous or demand mode, and also offers an echo function. EOS supports asynchronous mode and the echo function.

33.1.1.1 Asynchronous Mode

In asynchronous mode, BFD control packets are exchanged by neighboring systems at regular intervals. If a specified number of sequential packets are not received, BFD declares the session to be down.

33.1.1.2 Demand Mode

In demand mode, once the BFD session is established, the participating systems can request that BFD packets not be sent, then request an exchange of packets only when needed to verify connectivity. EOS does not support demand mode.

33.1.2 Echo Function

When the echo function is in use, echo packets are looped back through the hardware forwarding path of the neighbor system without involving the CPU. Failure is detected by an interruption in the stream of echoed packets. The minimum reception rate for BFD control packets from the neighbor is also changed automatically when the echo function is operational, because liveness detection is supplied by the echo packets.

While BFD control messages are transmitted to port 3786, BFD echo messages use UDP port 3785 for both source and destination.

33.1.3 BFD on Port Channels

On port channels, the BFD per-link feature can be used to add resiliency to the port channel's BFD sessions. When BFD per-link is enabled, BFD considers the port channel "up" as long as any link in the port channel is functioning properly.

Important! BFD per-link and BFD echo are mutually exclusive on a port channel. If both are configured, BFD per-link takes precedence.

33.2 BFD Configuration

To use BFD as the failure detection mechanism for a routing protocol, it must be enabled for each participating protocol.

These sections describe BFD configuration tasks:

- Section 33.2.1: Configuring BFD on an Interface
- Section 33.2.2: Configuring BFD on a Port Channel
- Section 33.2.3: Configuring the Echo Function
- Section 33.2.4: Configuring BFD for PIM
- Section 33.2.5: Configuring BFD for BGP
- Section 33.2.6: Configuring BFD for VRRP
- Section 33.2.7: Configuring BFD for OSPF
- Section 33.2.8: Displaying BFD Neighbor Information

33.2.1 Configuring BFD on an Interface

The transmission rate for BFD control packets, the minimum rate at which control packets are expected from the peer, and the multiplier (the number of packets that must be missed in succession before BFD declares the session to be down) can all be configured per interface. The values configured apply to all BFD sessions that pass through the interface.

The default values for these parameters are:

- **transmission rate** 300 milliseconds
- **minimum receive rate** 300 milliseconds
- **multiplier** 3

To configure different values for these parameters on an interface, use the **bfd interval** command.

For BFD to function as a failure detection mechanism, it must be enabled for each participating protocol.

Example

- These commands set the transmit and receive intervals to 200 milliseconds and the multiplier to 2 for all BFD sessions passing through Ethernet interface 3/20.

```
switch(config)#interface ethernet 3/20
switch(config-if-Et3/20)#bfd interval 200 min_rx 200 multiplier 2
switch(config-if-Et3/20)#
```

33.2.2 Configuring BFD on a Port Channel

Basic BFD parameters are configured on a port channel as described in Configuring BFD on an Interface above.

Additionally, BFD can be configured in per-link mode on a port channel so that the port channel will be considered up as long as any link in the channel is up.

To enable BFD per-link on a port channel, use the bfd per-link command.

Important! BFD per-link and BFD echo are mutually exclusive on a port channel. If both are configured, BFD per-link takes precedence.

Example

- These commands enabled BFD per-link on port channel 5. BFD echo will be disabled on the port channel.

```
switch(config)#interface port-channel 5
switch(config-if-Po5)#bfd per-link
switch(config-if-Po5)#
```

33.2.3 Configuring the Echo Function

The echo function is disabled by default, and is enabled on an interface using the **bfd echo** command.

When the BFD echo function is enabled, a “slow-timer” value replaces the minimum receive interval value in BFD packets sent from the switch. The default value is 2000 milliseconds. To configure a different value for the slow-timer, use the **bfd slow-timer** command.

Important! BFD per-link and BFD echo are mutually exclusive on a port channel. If both are configured, BFD per-link takes precedence.

Examples

- These commands enable the BFD echo function on Ethernet interface 5. If a slow-timer value has been configured on the switch, the minimum receive rate expected from the BFD neighbor will be reset to that value; otherwise, the minimum receive rate will be set to 2000 milliseconds.

```
switch(config)#interface ethernet 5
switch(config-if-Et5)#bfd echo
switch(config-if-Et5)#
```

- This command configures BFD to expect control packets from the peer every 10000 milliseconds when the BFD echo function is enabled.

```
switch(config)#bfd slow-timer 10000
switch(config)#
```

33.2.4 Configuring BFD for PIM

The **ip pim bfd** command enables or disables bidirectional forwarding detection (BFD) globally for all protocol independent multicast (PIM) neighbors.

To enable or disable PIM BFD on a specific interface, use the **ip pim bfd-instance** command. The interface-level configuration supercedes the global setting.

Example

- This command enables PIM BFD globally on the switch, enabling it on all interfaces where it is not explicitly disabled.

```
switch(config)#ip pim bfd
switch(config)#
```

- These commands configure VLAN interface 200 to use BFD for PIM connection failure detection regardless of the global PIM BFD configuration..

```
switch(config)#interface vlan 200
switch(config-if-VL200)#ip pim bfd-instance
switch(config-if-VL200)#
```

33.2.5 Configuring BFD for BGP

To enable or disable bidirectional forwarding detection (BFD) for border gateway protocol (BGP) connections with a BGP neighbor or peer group, use the **neighbor fall-over bfd** command.

Example

- These commands enable BFD failure detection for BGP connections with the neighbor at 10.13.64.1.

```
switch(config)#router bgp 300
switch(config-router-bgp)#neighbor 10.13.64.1 fall-over bfd
switch(config-router-bgp)#
```

33.2.6 Configuring BFD for VRRP

To enable or disable bidirectional forwarding detection (BFD) for virtual router redundancy protocol (VRRP), use the **vrrp bfd ip** command.

When enabled, BFD provides failure detection for a 2-router VRRP system. When the master is configured with the physical IP address of the backup router, and the backup is configured with the address of the master, a BFD session is established between them. If the BFD session goes down, the backup router immediately assumes the master role.

VRRP master advertisement packets are still sent even when the BFD session is established to accommodate VRRP systems involving more than two routers.

Example

- These commands enable BFD on Ethernet interface 3/20 for VRRP ID 15 with a connection to a router at IP address 192.168.2.1.

```
switch(config)#interface ethernet 3/20
switch(config-if-Et3/20)#vrrp 15 bfd ip 192.168.2.1
switch(config-if-Et3/20)#
```

33.2.7 Configuring BFD for OSPF

To enable or disable BFD globally for all OSPF neighbors, use the **bfd all-interfaces** command in OSPF configuration mode.

To enable or disable BFD for OSPF on a specific interface, use the **ip ospf bfd** command. The interface-level configuration supercedes the global setting.

Example

- These commands enable BFD in OSPF instance 100 for all OSPF neighbors on BFD-enabled interfaces except those connected to interfaces on which OSPF BFD has been explicitly disabled.

```
switch(config)#router ospf 100
switch(config-router-ospf)#bfd all-interfaces
switch(config-router-ospf)#
```

- This command enables OSPF BFD on Ethernet interface 3/21.

```
switch(config)#interface ethernet 3/21
switch(config-if-Et3/21)#ip ospf bfd
switch(config-if-Et3/21)#
```

33.2.8 Displaying BFD Neighbor Information

Use the **show bfd neighbors** command to display information about bidirectional forwarding detection (BFD) neighbors.

Example

- This command displays general information about BFD neighbors.

```
switch>show bfd neighbors
DstAddr       MyDisc  YoDisc  If            LUp     LDown  Ldiag           Stat
e

10.168.1.56   16      13      et52_1(81)    17151450 0      No Diagnostic    Up

10.168.1.58   17      14      et52_2(65)    17151883 0      No Diagnostic    Up

10.168.1.24   18      15      et51_1(73)    17152175 0      No Diagnostic    Up

              <-------OUTPUT OMITTED FROM EXAMPLE-------->
```

- This command displays detailed information about BFD neighbors.

```
switch>show bfd neighbors detail
Peer Addr 10.168.1.56, Intf Ethernet52/1, State Up
VRF default, LAddr 10.168.1.57, LD/RD 16/13
Last Up 17151450
Last Down 0
Last Diag: No Diagnostic
TxInt: 300, RxInt: 300, Multiplier: 3
Received RxInt: 300, Received Multiplier: 3
Rx Count: 433987, Tx Count: 433829
Detect Time: 900
Registered protocols: bgp

Peer Addr 10.168.1.58, Intf Ethernet52/2, State Up
VRF default, LAddr 10.168.1.59, LD/RD 17/14
Last Up 17151883
Last Down 0
Last Diag: No Diagnostic
TxInt: 300, RxInt: 300, Multiplier: 3
Received RxInt: 300, Received Multiplier: 3
Rx Count: 434235, Tx Count: 434050
Detect Time: 900
Registered protocols: bgp
```

33.3 BFD Command Descriptions

BFD Configuration Commands

- bfd interval
- bfd echo
- bfd slow-timer

BFD Display Commands

- show bfd neighbors

PIM-BFD Configuration Commands

- ip pim bfd
- ip pim bfd-instance

BGP-BFD Configuration Commands

- neighbor fall-over bfd

VRRP-BFD Configuration Commands

- vrrp bfd ip

OSPF-BFD Configuration Commands

- bfd all-interfaces
- ip ospf bfd

bfd all-interfaces

The **bfd all-interfaces** command globally configures OSPF to use bidirectional forwarding detection (BFD). When this command is issued, BFD sessions will be established with all OSPF neighbors connected to BFD-enabled interfaces unless OSPF BFD has been disabled on a participating interface using the **ip ospf bfd** command. BFD is globally disabled in OSPF by default.

For OSPF BFD to function on an interface, BFD must also be enabled and configured on that interface using the **bfd interval** command.

The **no bfd all-interfaces** and **default bfd all-interfaces** commands disable OSPF BFD on all interfaces except those where it has been explicitly enabled using the **ip ospf bfd** command.

Command Mode

Router-OSPF Configuration

Command Syntax

```
bfd all-interfaces
no bfd all-interfaces
default bfd all-interfaces
```

Examples

These commands enable BFD for OSPF instance 100 on all interfaces except those on which OSPF BFD has been explicitly disabled.

```
switch(config)#router ospf 100
switch(config-router-ospf)#bfd all-interfaces
switch(config-router-ospf)#
```

bfd echo

The **bfd echo** command enables the BFD echo function on the configuration mode interface.

The **no bfd echo** and **default bfd echo** commands disable the BFD echo function by removing the corresponding **bfd echo** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration
Interface-Loopback Configuration
Interface-Management Configuration
Interface-Port-channel Configuration
Interface-VLAN Configuration

Command Syntax

```
bfd echo
no bfd echo
default bfd echo
```

Example

- These commands enable the BFD echo function on Ethernet interface 5. If a slow-timer value has been configured on the switch, the minimum receive rate expected from the BFD neighbor will be reset to that value; otherwise, the minimum receive rate will be set to 2000 milliseconds.

```
switch(config)#interface ethernet 5
switch(config-if-Et5)#bfd echo
switch(config-if-Et5)#
```

bfd interval

The **bfd interval** command configures the BFD control packet transmission rate, minimum control packet receive rate, and the number of missed packets that will signal that the session is down. These parameters can be configured globally for the switch or for the configuration mode interface. If a parameter is configured both globally and on the interface, the value configured on the interface takes precedence.

Important! For a BFD session to be established, BFD must be enabled for any routing protocol using BFD for failure detection.

The **no bfd interval** and **default bfd interval** commands return the BFD parameters on the configuration mode interface to default values by removing the corresponding **bfd interval** command from ***running-config***.

Command Mode

Global Configuration
Interface-Ethernet Configuration
Interface-Loopback Configuration
Interface-Management Configuration
Interface-Port-channel Configuration
Interface-VLAN Configuration

Command Syntax

```
bfd interval transmit_rate min_rx receive_minimum multiplier factor
no bfd interval
default bfd interval
```

Parameters

- *transmit_rate* rate in milliseconds at which control packets will be sent. Values range from 50 to 60000; the default value is 300.
- *receive_minimum* rate in milliseconds at which control packets will be expected. Values range from 50 to 60000.
- *factor* number of consecutive missed BFD control packets after which BFD will declare the session as down. Values range from 3 to 50.

Examples

- These commands configure BFD on Ethernet interface 5 to expect packets from the peer every 200 milliseconds and declare the session down after failing to receive 5 consecutive packets. This configuration overrides any values configured globally.

```
switch(config)#interface ethernet 5
switch(config-if-Et5)#bfd interval 200 min_rx 200 multiplier 5
switch(config-if-Et5)#
```

bfd per-link

The **bfd per-link** command enables the BFD per-link function on the port channel being configured. When BFD per-link is enabled, BFD sub-sessions are run on each link of the port channel; BFD considers the port-channel to be “up” as long as any one of the links is live.

Important! BFD per-link and BFD echo are mutually exclusive. If both are enabled on a port channel, BFD per-link takes precedence and BFD echo will be disabled.

The **no bfd per-link** and **default bfd per-link** commands disable the BFD per-link function by removing the corresponding **bfd per-link** command from ***running-config***.

Command Mode

Interface-Port-channel Configuration

Command Syntax

```
bfd per-link
no bfd per-link
default bfd per-link
```

Example

- These commands enable the BFD per-link function on port channel 5. If the BFD echo function has been configured on the port channel, the per-link function will disable BFD echo.

```
switch(config)#interface port-channel 5
switch(config-if-Po5)#bfd per-link
switch(config-if-Po5)#
```

bfd slow-timer

The **bfd slow-timer** command configures the minimum reception rate for BFD control packets which will be used if the BFD echo function is enabled. The default value is 2000 milliseconds.

Important! For a BFD session to be established, BFD must be enabled for any routing protocol using BFD for failure detection.

The **no bfd slow-timer** and **default bfd slow-timer** commands return the BFD slow-timer to the default value of 2000 milliseconds by removing the corresponding **bfd interval** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
bfd slow-timer receive_minimum
no bfd slow-timer
default bfd slow-timer
```

Parameters

- *receive_minimum* rate in milliseconds at which control packets will be expected when the BFD echo function is enabled. Values range from 2000 to 60000; default value is 2000.

Examples

- This command configures BFD to expect control packets from the peer every 10000 milliseconds when the BFD echo function is enabled.

```
switch(config)#bfd slow-timer 10000
switch(config)#
```

ip ospf bfd

The **ip ospf bfd** command enables bidirectional forwarding detection (BFD) for the open shortest path first protocol (OSPF) on the configuration mode interface regardless of the global settings for the OSPF instance. All OSPF neighbors associated with the interface become BFD peers, and OSPF uses BFD for failure detection.

For OSPF BFD to function on an interface, BFD must also be enabled and configured on that interface using the **bfd interval** command.

The **no ip ospf bfd** command disables OSPF BFD on the interface and terminates all BFD sessions with the interface's OSPF peers. The **default ip ospf bfd** command causes the interface to follow global OSPF BFD settings configured by the **bfd all-interfaces** command.

Command Mode

Interface-Ethernet Configuration
Interface-Port-Channel Configuration
Interface-VLAN Configuration

Command Syntax

```
ip ospf bfd
no ip ospf bfd
default ip ospf bfd
```

Example

- These commands enable BFD on Ethernet interface 3/20.

```
switch(config)#interface ethernet 3/20
switch(config-if-Et3/20)#ip ospf bfd
switch(config-if-Et3/20)#
```

- These commands cause Ethernet interface 3/20 to follow the global OSPF BFD configuration.

```
switch(config)#interface ethernet 3/20
switch(config-if-Et3/20)#default ip ospf bfd
switch(config-if-Et3/20)#
```

ip pim bfd

The **ip pim bfd** command enables bidirectional forwarding detection (BFD) globally for use as a failure detection mechanism for protocol independent multicast (PIM) on the switch. To override the global configuration for a specific interface, use the **ip pim bfd-instance** command. All PIM interfaces will use the global setting if they are not individually configured.

When PIM BFD is enabled, a BFD session is created for each PIM neighbor and used to detect a loss of connectivity with the neighbor. PIM hello packets are still exchanged with PIM neighbors when BFD is enabled.

The **no ip pim bfd** and **default ip pim bfd** commands disable PIM BFD globally by deleting the **ip pim bfd** statement from ***running-config***. When this is done, only interfaces with PIM BFD explicitly enabled will use PIM BFD.

Command Mode

Global Configuration

Command Syntax

```
ip pim bfd
no ip pim bfd
default ip pim bfd
```

Example

- This command enables PIM BFD globally on the switch, enabling it on all interfaces where it is not explicitly disabled.

```
switch(config)#ip pim bfd
switch(config)#
```

ip pim bfd-instance

The **ip pim bfd-instance** command enables bidirectional forwarding detection (BFD) on the configuration mode interface as a failure detection mechanism for protocol-independent multicast (PIM). To enable PIM BFD globally on the switch, use the **ip pim bfd** command. Interface-level settings override the global setting.

When PIM BFD is enabled, a BFD session is created for each PIM neighbor and used to detect a loss of connectivity with the neighbor. PIM hello packets are still exchanged with PIM neighbors when BFD is enabled.

The **no ip pim bfd-instance** disables PIM BFD on the configuration mode interface regardless of global settings. The **default ip pim bfd-instance** command causes the configuration mode interface to follow the global setting for PIM BFD by removing the corresponding **ip pim bfd-instance** statement from ***running-config***.

Command Mode

Interface-Ethernet Configuration
Interface-Port-Channel Configuration
Interface-VLAN Configuration

Command Syntax

```
ip pim bfd-instance
no ip pim bfd-instance
default ip pim bfd-instance
```

Example

- These commands configure VLAN interface 200 to use BFD for PIM connection failure detection regardless of the global PIM BFD configuration.

```
switch(config)#interface vlan 200
switch(config-if-VL200)#ip pim bfd-instance
switch(config-if-VL200)#
```

neighbor fall-over bfd

The **neighbor fall-over bfd** command enables bidirectional forwarding detection (BFD) for use as a failure detection mechanism for border gateway protocol (BGP) connections to the specified BGP neighbor or peer group.

Once a BFD session is established with a BGP neighbor, if the BFD session goes down the status of the BGP session is changed to “down” as well.

The **no neighbor fall-over bfd** and **default neighbor fall-over bfd** commands disable BFD for BGP connections to the specified neighbor or peer group by removing the corresponding **neighbor fall-over bfd** command from ***running-config***.

Command Mode

Router-BGP Configuration

Command Syntax

```
neighbor NEIGHBOR_ID fall-over bfd
no neighbor NEIGHBOR_ID fall-over bfd
default neighbor NEIGHBOR_ID fall-over bfd
```

Parameters

- ***NEIGHBOR_ID*** IP address or peer group name. Values include:
  - *ipv4_addr* neighbor’s IPv4 address.
  - *ipv6_addr* neighbor’s IPv6 address.
  - *group_name* peer group name.

Example

- These commands enable BFD failure detection for BGP connections with the neighbor at 10.13.64.1.

```
switch(config)#router bgp 300
switch(config-router-bgp)#neighbor 10.13.64.1 fall-over bfd
switch(config-router-bgp)#
```

show bfd neighbors

The **show bfd neighbors** command displays information about the neighbors with which the switch currently has a bidirectional forwarding detection (BFD) session.

Command Mode

EXEC

Command Syntax

```
show bfd neighbors [INFO_LEVEL]
```

Parameters

- ***INFO_LEVEL*** amount of information that is displayed. Options include:
  - <no parameter> command displays data block for each specified interface.
  - **detail** command displays table that summarizes interface data.

Display Values

- **DstAddr** IP address of the BFD neighbor.
- **MyDisc** Local discriminator value of the BFD session.
- **YoDisc** Neighbor's discriminator value for the BFD session.
- **If** Interface to which the neighbor is connected.
- **LUp** Last up.
- **LDown** Last down.
- **Ldiag** Diagnostic for the last change in session state.
- **State** State of the BFD session.
- **TxInt** Transmit interval of the local interface.
- **RxInt** Minimum receive interval set on the local interface.
- **Multiplier** Local multiplier (number of packets that must be missed to declare session down).
- **Received RxInt** Minimum receive interval set on the neighbor interface.
- **Received Multiplier** Neighbor's multiplier (number of packets that must be missed to declare session down).
- **Rx Count** BFD control packets transmitted.
- **Tx Count** BFD control packets received.
- **Detect Time** Total time in milliseconds it takes for BFD to detect connection failure.
- **Registered Protocols** Protocols using BFD with this neighbor.

Examples

- This command displays general information about BFD neighbors.

```
switch>show bfd neighbors
DstAddr       MyDisc  YoDisc  If            LUp     LDown   Ldiag            S
tate

10.168.1.56   16      13      et52_1(81)    17151450 0       No
Diagnostic      Up

10.168.1.58   17      14      et52_2(65)    17151883 0       No
Diagnostic      Up

10.168.1.24   18      15      et51_1(73)    17152175 0       No
Diagnostic      Up

10.168.254.6  19      12      vlan4094(26)  17152336 0       No
Diagnostic      Up

10.168.1.26   20      16      et51_2(57)    17152523 0       No
Diagnostic      Up

10.168.1.40   21      12      et50_1(77)    17152966 0       No
Diagnostic      Up

10.168.1.42   22      13      et50_2(61)    17153488 0       No
Diagnostic      Up

10.168.1.8    27      55      et49_1(69)    26710447 0       No
Diagnostic      Up

10.168.1.10   28      56      et49_2(53)    26710847 0       No
Diagnostic      Up
```

- This command displays detailed information about BFD neighbors.

```
switch>show bfd neighbors detail
Peer Addr 10.168.1.56, Intf Ethernet52/1, State Up
VRF default, LAddr 10.168.1.57, LD/RD 16/13
Last Up 17151450
Last Down 0
Last Diag: No Diagnostic
TxInt: 300, RxInt: 300, Multiplier: 3
Received RxInt: 300, Received Multiplier: 3
Rx Count: 433987, Tx Count: 433829
Detect Time: 900
Registered protocols: bgp

Peer Addr 10.168.1.58, Intf Ethernet52/2, State Up
VRF default, LAddr 10.168.1.59, LD/RD 17/14
Last Up 17151883
Last Down 0
Last Diag: No Diagnostic
TxInt: 300, RxInt: 300, Multiplier: 3
Received RxInt: 300, Received Multiplier: 3
Rx Count: 434235, Tx Count: 434050
Detect Time: 900
Registered protocols: bgp
switch>
```

vrrp bfd ip

The **vrrp bfd ip** command enables and configures bidirectional forwarding detection (BFD) for virtual router redundancy protocol (VRRP) on the configuration mode interface.

When enabled, BFD provides failure detection for a 2-router VRRP system. When the master is configured with the physical IP address of the backup router, and the backup is configured with the address of the master, a BFD session is established between them. If the BFD session goes down, the backup router immediately assumes the master role.

VRRP master advertisement packets are still sent even when the BFD session is established to accommodate VRRP systems involving more than two routers.

The **no vrrp bfd ip** and **default vrrp bfd ip** commands disable BFD for VRRP on the configuration mode interface by removing the corresponding **vrrp bfd ip** statement from ***running-config.*** The **no vrrp** command also removes the **vrrp bfd ip** command for the specified virtual router.

Command Mode

Interface-Ethernet Configuration
Interface-Port-Channel Configuration
Interface-VLAN Configuration

Command Syntax

```
vrrp group bfd ip ipv4_address
no vrrp group bfd ip
default vrrp group bfd ip
```

Parameters

- *group* virtual router identifier (VRID). Values range from 1 to 255.
- *ipv4_address* IPv4 address of the other VRRP router. On the master router, enter the physical IP address of the backup; on the backup, enter the physical IP address of the master.

Example

- These commands enable BFD on Ethernet interface 3/20 for VRRP ID 15 with a connection to a router at IP address 192.168.2.1.

```
switch(config)#interface ethernet 3/20
switch(config-if-Et3/20)#vrrp 15 bfd ip 192.168.2.1
switch(config-if-Et3/20)#
```

Chapter 34

Multicast Architecture

IP multicast is the transmission of data packets to multiple hosts through a common IP address. Arista switches support multicast transmissions through IGMP, IGMP Snooping, and PIM-SM. These sections describe the Arista multicast architecture.

- Section 34.1: Introduction is a chapter overview and lists the features supported by Arista switches.
- Section 34.2: Multicast Architecture Description describes multicast data structures
- Section 34.3: Multicast Configuration describes multicast implementation configuration tasks.
- Section 34.4: Multicast Commands contains multicast command descriptions.

34.1 Introduction

Arista switches provide layer 2 multicast filtering and layer 3 routing features for applications requiring IP multicast services. The switches support over a thousand separate routed multicast sessions at wire speed without compromising other Layer 2/3 switching features. Arista switches support IGMP, IGMP snooping, PIM-SM, and MSDP to simplify and scale data center multicast deployments.

Supported Features

Feature support varies by platform; please consult the release notes for multicast support information by platform.

Multicast and unicast use the same routing table. Unicast routes use TCAM resources, which may also impact the maximum number of multicast routes.

Features Not Supported

These multicast functions are not supported by Arista switches include (*,*,G) forwarding or boundary routers, Multicast MIBs, and Router applications joining multicast groups

34.2 Multicast Architecture Description

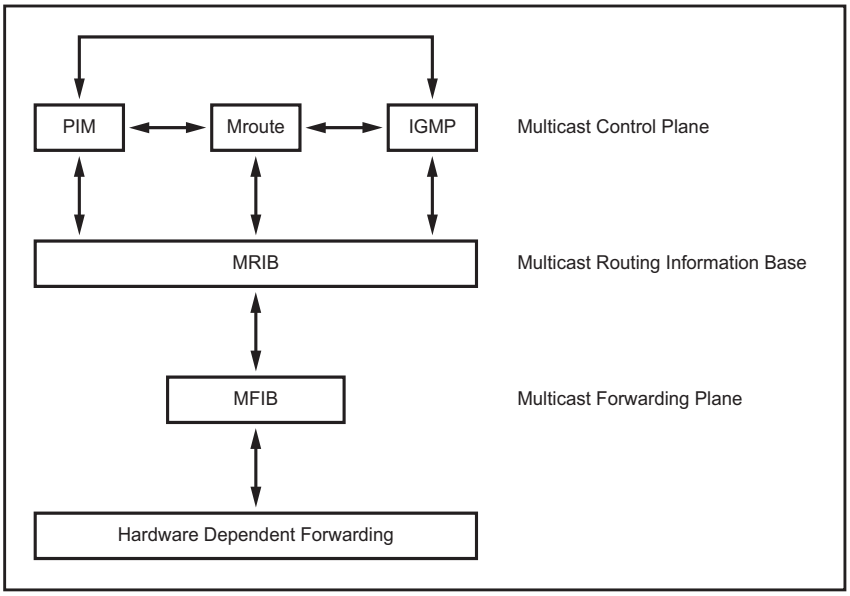
IP multicast is data transmission to a subset of all hosts through a single multicast group address. Multicast packets are delivered using best-effort reliability, similar to unicast packets. Senders use the multicast address as the destination address. Any host, regardless of group membership, can send to a group. However, only group members receive messages sent to a group address.

IP multicast addresses range from 224.0.0.0 to 239.255.255.255. Multicast routing protocol control traffic reserves the address range 224.0.0.0 to 224.0.0.255. The address 224.0.0.0 is never assigned to any group.

Multicast group membership is dynamic; a group's activity level and membership can vary over time. A host can also simultaneously belong to multiple multicast groups.

Figure 34-1 depicts the components that comprise the multicast architecture. This section describes multicast components depicted in the figure.

Figure 34-1: Multicast Architecture



34.2.1 Multicast Control Plane

The Multicast Control Plane builds and maintains multicast distribution trees. It communicates changes in the multicast routing table to the MFIB for multicast forwarding.

- Protocol Independent Multicast (PIM) builds and maintains multicast routing trees using reverse path forwarding (RPF) on a unicast routing table.
- Internet Group Management Protocol (IGMP) identifies multicast group members on subnets directly connected to the switch. Hosts manage multicast group membership with IGMP messages.
- The switch maintains an mroute (multicast routing) table when running PIM to provide forwarding tables used to deliver multicast packets.

  The mroute table stores the states of inbound and outbound interfaces for each source-group pair (S,G). The switch discards and forwards packets on the basis of this state information. Each table entry, referred to as an mroute, corresponds to a unique (S,G) and contains:

  - the multicast group address
  - the multicast source address (or * for all sources)

- the inbound interface
- a list of outbound interfaces

34.2.2 Multicast Forwarding Plane

The Multicast Forwarding Plane consists of the Multicast Forwarding Information Base (MFIB), a forwarding engine that is independent of multicast routing protocols.

MFIB formats PIM and IGMP multicast routes for protocol-independent hardware packet forwarding and adds them to the hardware multicast expansion table (MET) and the hardware FIB.

MFIB uses a core forwarding engine for interrupt-level (fast switching) and process-level (process switching) forwarding. MFIB fast-switches inbound multicast packets that match an MFIB forwarding entry and process-switches packets requiring a forwarding entry if a matching entry does not exist.

34.2.3 Multicast Routing Information Base (MRIB)

The MRIB is the channel between Multicast Control Plane clients and the Multicast Forwarding Plane. The **show ip mroute** displays MRIB entries as (*, G), (S, G), and (*, G/m) multicast entries.

MRIB entries are based on source, group, and group masks. The entries are associated with a list of interfaces whose forwarding state is described with flags. MRIB communication is based on the state change of entry and interface flags. Flags are significant to MRIB clients and not interpreted by MRIB.

34.2.4 Hardware Dependent Forwarding and Fast Dropping

In IP multicast protocols, each (S,G) and (*,G) route corresponds to an inbound reverse path forwarding (RPF) interface. Packets arriving on non-RPF interfaces may require PIM processing, as performed by the CPU subsystem software.

By default, hardware sends all packets arriving on non-RPF interfaces to the CPU subsystem software. However, the CPU can be overwhelmed by non-RPF packets that do not require software processing. The CPU subsystem software prevents CPU overload by creating a fast-drop entry in hardware for inbound non-RPF packets not requiring PIM processing. Packets matching a fast-drop entry are bridged in the ingress VLAN but not sent to the software, avoiding CPU subsystem software overload. Fast-drop entry usage is critical in topologies with persistent RPF failures.

Protocol events, such as links going down or unicast routing table changes, can change the set of packets that can be fast dropped. Packets that were correctly fast dropped before a topology change may require forwarding to the CPU subsystem software after the change. The CPU subsystem software handles fast-drop entries that respond to protocol events so that PIM can process all necessary non-RPF packets.

34.3 Multicast Configuration

This section describes the following configuration tasks:

- Section 34.3.1: Multicast Configuration
- Section 34.3.2: Configuring MFIB
- Section 34.3.3: Configuring Static IP Mroute
- Section 34.3.4: Displaying and Clearing the mroute Table

34.3.1 Multicast Configuration

Enabling Multicast Routing

Enabling IP multicast routing allows the switch to forward multicast packets. The **ip multicast-routing** command enables multicast routing. When multicast routing is enabled, ***running-config*** contains an **ip multicast-routing** statement.

Example

- This command enables multicast routing on the switch.

```
switch(config)#ip multicast-routing
switch(config)#
```

Multicast Boundary Configuration

The multicast boundary specifies subnets where source traffic entering an interface is filtered to prevent the creation of mroute states on the interface. The interface is not included in the outgoing interface list (OIL). Multicast PIM, IGMP and other multicast data cannot cross the boundary, facilitating the use of a multicast group address in multiple administrative domains.

The **ip multicast boundary** command configures the multicast boundary. The multicast boundary can be specified through multiple IPv4 subnets or one standard IPv4 ACL.

Examples

- This command configures the multicast address of 229.43.23.0/24 as a multicast boundary where source traffic is restricted from VLAN interface 300.

```
switch(config)#interface vlan 300
switch(config-if-vl300)#ip multicast boundary 229.43.23.0/24
switch(config-if-vl300)#
```

- These commands create a standard ACL, then implements the ACL in an **ip multicast boundary** command to configure two boundary subnets (225.123.0.0/16 and 239.120.10.0/24).

```
switch(config)#ip access-list standard mbac1
switch(config-std-acl-mbac1)#10 deny 225.123.0.0/16
switch(config-std-acl-mbac1)#20 deny 239.120.10.0/24
switch(config-std-acl-mbac1)#permit any
switch(config-std-acl-mbac1)#exit
switch(config)#interface vlan 200
switch(config-if-Vl200)#ip multicast boundary mbac1
switch(config-if-Vl200)#exit
switch(config)#
```

34.3.2 Configuring MFIB

MFIB formats PIM and IGMP multicast routes for protocol-independent hardware packet forwarding and adds them to the hardware multicast expansion table (MET) and the hardware FIB.

MFIB Polling Interval

The switch records activity levels for multicast routes in the MFIB after polling the corresponding hardware activity bits. The **ip mfib activity polling-interval** command specifies the frequency that the switch polls the hardware activity bits for the multicast routes.

Example

- This command sets the MFIB activity polling period at 15 seconds.

```
switch(config)#ip mfib activity polling-interval 15
switch(config)#
```

MFIB Fastdrops

In IP multicast protocols, every (S,G) or (*,G) route is associated with an inbound RPF (reverse path forwarding) interface. Packets arriving on an interface not associated with the route may need CPU-dependent PIM processing, so packets received by non-RPF interfaces are sent to the CPU by default, causing heavy CPU processing loads.

Multicast routing protocols often do not require non-RPF packets; these packets do not require software processing. The CPU therefore updates the hardware MFIB with a fast-drop entry when it receives a non-RPF interface packet that PIM does not require. Additional packets that match the fast-drop entry are not sent to the system software.

Fastdrop is enabled on all interfaces by default. The **no ip mfib fastdrop** command disables MFIB fast drops for the configuration mode interface.

Example

- This command disables MFIB fast drops for the VLAN interface 120.

```
switch(config)#interface vlan 120
switch(config-if-Vl120)#no ip mfib fastdrop
switch(config-if-Vl120)#
```

The **ip mfib max-fastdrops** command limits the number of fast drop routes that the switch's MFIB table can contain. The default fast drop route limit is 1024.

Example

- This command sets the maximum number of fast drop routes at 2000.

```
switch(config)#ip mfib max-fastdrops 2000
switch(config)#
```

The **clear ip mfib fastdrop** command, in global configuration mode, removes all MFIB fast drop entries on all interfaces.

Example

- This command removes all fast-drop entries from the MFIB table.

```
switch#clear ip mfib fastdrop
switch#
```

The **show ip mfib** command displays information about the routes and interfaces in the IPv4 MFIB

- **show ip mfib** displays MFIB information for hardware-forwarded routes.
- **show ip mfib software** displays MFIB information for software-forwarded routes.

Example

- This command displays MFIB information for hardware-forwarded routes.

```
switch>show ip mfib
Activity poll time: 60 seconds
  239.255.255.250 172.17.26.25
    Vlan26 (iif)
    Vlan2028
    Cpu
      Activity 0:02:11 ago
  239.255.255.250 172.17.26.156
    Vlan26 (iif)
    Vlan2028
    Cpu
      Activity 0:02:11 ago
  239.255.255.250 172.17.26.178
    Vlan26 (iif)
    Vlan2028
    Cpu
      Activity 0:03:37 ago
switch>
```

34.3.3 Configuring Static IP Mroute

The static IP multicast route (or static mroute) interface overrides the interface that is ordinarily selected from the marching route in the unicast routing table, providing a means for breaking dependence on multicast topology for unicast topology. The command **ip mroute** specifies a candidate for the multicast reverse path forwarding (RPF) interface of any (S,G) multicast route, where the source falls within the given source or mask.

34.3.3.1 Selecting RPF interface

Static mroutes are stored in a separate routing table, the Multicast Routing Information Base (MRIB). The RPF interface is selected for a source as follows:

- A route to the source is selected from the unicast Rib (based on the existing unicast Rib lookup algorithm).
- A route to the source is selected from the mrib.
- If only one of the above lookups yields a route, that route is used to select the RPF interface.
- If both of the above lookups yield a route, the admin distances of the two routes is compared. The route with the lower admin distance is used to select the RPF interface.
- Comparing the static mroute and the unicast route, the static mroute is the winner even though both have the same distance.
- Comparing routes from the two Ribs is not driven by specificity. The route with the lower cost is selected even if less specific.
- A connected route in the unicast Rib will always win.
- Static mroutes, by default, have an admin distance of one (1).
- For a static mroute to be considered for selection, the specified interface must be up and the Pim must be enabled on it.

34.3.3.2 Selecting Static Mroutes

The longest match is selected when a source matches multiple static mroutes in the mrib. The order in which static mroutes were configured is not a factor.

Example

- This command selects the longest match when a source matches multiple static mroutes in the mrib.

```
switch(config-as)#ip mroute 10.0.0.0/16 Ethernet 4
switch(config-as)#ip mroute 11.10.1.0/24 Ethernet 5
switch(config-as)#ip mroute 11.10.1.2/32 Ethernet 6
switch(config-as)#
```

Example

- This command includes an administrative distance of 255 on Ethernet 5 with a static mroute.

```
switch(config-as)#ip mroute 10.0.0.0/16 Ethernet 4
switch(config-as)#ip mroute 11.10.1.0/24 Ethernet 5 255
switch(config-as)#ip mroute 11.10.1.2/32 Ethernet 6
switch(config-as)#
```

34.3.4 Displaying and Clearing the mroute Table

The mroute table stores the states of inbound and outbound interfaces for each source-group pair (S,G). The switch discards and forwards packets on the basis of this state information. Each table entry, referred to as an mroute, corresponds to a unique (S,G) and contains:

- the multicast group address
- the multicast source address (or * for all sources)
- the inbound interface
- a list of outbound interfaces

Clearing mroute Entries

The **clear ip mroute** command removes route entries from the mroute table:

- **clear ip mroute** * all entries from the mroute table.
- **clear ip mroute** *gp_ipv4* all entries for the specified multicast group.
- **clear ip mroute** *gp_ipv4* *src_ipv4* all entries for the specified source sending to a specified group.

Examples

- This command removes all route entries from the mroute table.

```
switch#clear ip mroute *
switch#
```

- This command removes entries for source 228.3.10.1 sending to multicast group 224.2.205.42.

```
switch#clear ip mroute 224.2.205.42 228.3.10.1
switch#
```

Displaying the mroute Table

The **show ip mroute count** command displays IP multicast routing table statistics.

Example

- This command displays IP multicast routing table statistics.

```
switch>show ip mroute count
IP Multicast Statistics
1 groups and 1 sources
Multicast routes: 1 (*,G), 1 (S,G)
Average of 1.00 sources per group
Maximum of 1 sources per group:
    228.24.12.1
switch>
```

The **show ip mroute** command displays information from the IP multicast routing table.

- **show ip mroute** displays information for all routes in the table.
- **show ip mroute** *gp_addr* displays information for the specified multicast group.

Example

- This command displays the IP multicast routing table for the multicast group 225.1.1.11

```
switch>show ip mroute 225.1.1.1
PIM Sparse Mode Multicast Routing Table
Flags: E - Entry forwarding on the RPT, J - Joining to the SPT
       R - RPT bit is set, S - SPT bit is set
       W - Wildcard entry, X - External component interest
       I - SG Include Join alert rcvd, P - Ex-Prune alert rcvd
       H - Joining SPT due to policy, D - Joining SPT due to protocol
       Z - Entry marked for deletion
       A - Learned via Anycast RP Router
225.1.1.1
  172.28.1.100, 5d04h, flags: S
    Incoming interface: Vlan281
    Outgoing interface list:
      Port-Channel999
switch>
```

34.4 Multicast Commands

Multicast Configuration Commands (Global)

- ip mfib activity polling-interval
- ip mfib cache-entries unresolved max
- ip mfib max-fastdrops
- ip mfib packet-buffers unresolved max
- ip mroute
- ip multicast multipath none
- ip multicast-routing

Multicast Configuration Commands (Interface)

- ip mfib fastdrop
- ip multicast boundary

Multicast Clear Commands

- clear ip mfib fastdrop
- clear ip mroute

Multicast Display Commands

- show ip mfib
- show ip mfib software
- show ip mroute
- show ip mroute count

clear ip mfib fastdrop

The **clear ip mfib fastdrop** command removes all fast-drop entries from the MFIB table.

Command Mode

Privileged EXEC

Command Syntax

```
clear ip mfib fastdrop
```

Example

- This command removes all fast-drop entries from the MFIB table.

```
switch#clear ip mfib fastdrop
switch#
```

clear ip mroute

The **clear ip mroute** command removes route entries from the mroute table, as follows:

- **clear ip mroute *** all entries from the mroute table.
- **clear ip mroute** *gp_ipv4* all entries for the specified multicast group.
- **clear ip mroute** *gp_ipv4* *src_ipv4* all entries for the specified source sending to a specified group.

Command Mode

Privileged EXEC

Command Syntax

```
clear ip mroute ENTRY_LIST
```

Parameters

- ***ENTRY_LIST*** entries that the command removes from the mroute table. Options include:
  - * all route entries are removed from the table
  - *gp_ipv4* all entries for multicast group *gp_ipv4* (dotted decimal notation).
  - *gp_ipv4* *src_ipv4* all entries for source (*src_ipv4*) sending to group (*gp_ipv4*).

Examples

- This command removes all route entries from the mroute table.

```
switch#clear ip mroute *
switch#
```

- This command removes entries for the source 228.3.10.1 sending to multicast group 224.2.205.42.

```
switch#clear ip mroute 224.2.205.42 228.3.10.1
switch#
```

ip mfib activity polling-interval

The switch records activity levels for multicast routes in the mfib after polling the corresponding hardware activity bits. The **ip mfib activity polling-interval** command specifies the frequency that the switch polls the hardware activity bits for the multicast routes.

The **no ip mfib activity polling-interval** and **default ip mfib activity polling-interval** commands restore the default interval of 60 seconds by removing the **ip mfib activity polling-interval** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
ip mfib activity polling-interval period
no ip mfib activity polling-interval
default ip mfib activity polling-interval
```

Parameters

- *period* interval (seconds) between polls. Values range from 1 to 60. Default is 60.

Example

- This command sets the MFIB activity polling period at 15 seconds.

```
switch(config)#ip mfib activity polling-interval 15
switch(config)#
```

ip mfib cache-entries unresolved max

The **ip mfib cache-entries unresolved max** command specifies the buffer size for storing multicast packets whose routes are not cached and that have not been otherwise dropped. The default buffer size is 4000 packets.

The **ip mfib packet-buffers unresolved max** command configures the number of packets for an individual (S,G) entry that the switch can process before its route is entered into cache. Packets that exceed this limit for an individual route are dropped. The **ip mfib cache-entries unresolved max** is the cumulative limit for all routes; packets that exceed this limit are dropped even if they do not exceed the limit for their routes.

The **no ip mfib cache-entries unresolved max** and **default ip mfib cache-entries unresolved max** commands restore the default unresovled cache-entries buffer size of 4000 packets by removing the **ip mfib cache-entries unresolved max** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
ip mfib cache-entries unresolved max quantity_entries
no ip mfib cache-entries unresolved max
default ip mfib cache-entries unresolved max
```

Parameters

- *quantity_entries* maximum buffer size (packets). Value ranges from 10 to 10000000. Default is 4000.

Example

- This command sets the maximum mfib unresovled cache-entry buffer size at 6000 packets.

```
switch(config)#ip mfib cache-entries unresolved max 6000
switch(config)#
```

ip mfib fastdrop

In IP multicast protocols, every (S,G) or (*,G) route is associated with an inbound RPF (reverse path forwarding) interface. Packets arriving on an interface not associated with the route may need CPU-dependent PIM processing, so packets received by non-RPF interfaces are sent to the CPU by default, causing heavy CPU processing loads.

Multicast routing protocols often do not require non-RPF packets; these packets do not require software processing. The CPU therefore updates the hardware MFIB with a fast-drop entry when it receives a non-RPF interface packet that PIM does not require. Additional packets that match the fast-drop entry are not sent to the system software.

Fastdrop is enabled on all interfaces by default. The **no ip mfib fastdrop** command disables MFIB fast drops for the configuration mode interface.

The **ip mfib fastdrop** and **default ip mfib fastdrop** commands enable MFIB fast drops for the configuration mode interface by removing the corresponding **no ip mfib fastdrop** command from ***running-config***.

The **clear ip mfib fastdrop** command, in global configuration mode, removes all MFIB fast drop entries on all interfaces.

Command Mode

Interface-Ethernet Configuration
Interface-Port-channel Configuration
Interface-VLAN Configuration

Command Syntax

```
ip mfib fastdrop
no ip mfib fastdrop
default ip mfib fastdrop
```

Examples

- This command enables MFIB fast drops for the VLAN interface 120.

```
switch(config)#interface vlan 120
switch(config-if-Vl120)#ip mfib fastdrop
switch(config-if-Vl120)#
```

ip mfib max-fastdrops

The **ip mfib max-fastdrops** command limits the number of fast drop routes that the switch's MFIB table can contain.

The **no ip mfib max-fastdrops** and **default ip mfib max-fastdrops** commands restore the default fast drop route limit of 1024 by removing the **ip mfib max-fastdrops** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
ip mfib max-fastdrops quantity
no ip mfib mfib max-fastdrops
default ip mfib mfib max-fastdrops
```

Parameters

- *quantity* number of fast-drop routes. Value ranges from 0 to 1000000 (one million). Default is 1024.

Example

- This command sets the maximum number of fast drop routes at 2000.

```
switch(config)#ip mfib max-fastdrops 2000
switch(config)#
```

ip mfib packet-buffers unresolved max

The **ip mfib packet-buffers unresolved max** command specifies the number of (S,G) multicast packets for an individual route that the switch can process before the (S,G) entry is entered into cache. Packets that are received in excess of this limit before the route is programmed into the cache are dropped. By default, the switch processes 3 unresolved packets for an individual route.

The **ip mfib cache-entries unresolved max** command specifies the buffer size for storing multicast packets whose routes are not cached and that have not been otherwise dropped. This command is the cumulative limit for all routes; packets that exceed this limit are dropped even if they do not exceed the limit for their routes.

The **no ip mfib packet-buffers unresolved max** and **default ip mfib packet-buffers unresolved max** commands restore the number of unresolved packets that the switch processes to the default value of 3 packets by removing the **ip mfib packet-buffers unresolved max** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
ip mfib packet-buffers unresolved max quantity_packets
no ip mfib packet-buffers unresolved max
default ip mfib packet-buffers unresolved max
```

Parameters

- *quantity_packets* packets per unresolved route that the switch processes. Value ranges from 3 to 10000000. Default is 3.

Example

- This command programs the switch to process three multicast packets from any route regardless of its entry's presence in the multicast routing cache.

```
switch(config)#ip mfib packet-buffers unresolved max 30
switch(config)#
```

ip multicast boundary

The **ip multicast boundary** command specifies subnets where source traffic entering the configuration mode interface is dropped, preventing the creation of mroute states on the interface. The interface is not included in the outgoing interface list (OIL). The multicast boundary can be specified through multiple IPv4 subnets or one standard IPv4 ACL.

Multicast PIM, IGMP and other multicast data cannot cross the boundary, facilitating the use of a multicast group address in multiple administrative domains.

The **no ip multicast boundary** and **default ip multicast boundary** commands delete the specified subnet restriction by removing the corresponding **ip multicast boundary** command from ***running-config***. When these commands do not specify a subnet address, all **ip multicast boundary** statements for the configuration mode interface are removed.

Command Mode

Interface-Ethernet Configuration
Interface-Port-channel Configuration
Interface-VLAN Configuration

Command Syntax

```
ip multicast boundary SUB_NET [TCAM]
no ip multicast boundary [SUB_NET]
default ip multicast boundary [SUB_NET]
```

Parameters

- ***SUB_NET*** the subnet address configured as the multicast boundary. Options include:
  - *net_addr* multicast subnet address (CIDR or address mask).
  - *acl_name* standard access control list (ACL) that specifies the multicast group addresses.
- ***TCAM*** specifies address inclusion in the routing table. Options include:
  - <no parameter> boundaries ((S,G) entries) are added to routing table.
  - **out** boundaries are not added to routing table.

Guidelines

When **out** is selected, the first inbound data packet corresponding to the ***SUB_NET*** may be sent to the CPU. In response, the packet is dropped and the boundary prefix is added to the hardware table. In this scenario, the mroute entry is added only when data traffic is received.

Restrictions

Only one command that specifies an ACL can be assigned to an interface. Commands that specify an ACL and a subnet cannot be simultaneously assigned to an interface.

Examples

- This command configures the multicast address of 229.43.23.0/24 as a multicast boundary where source traffic is restricted from VLAN interface 300.

```
switch(config)#interface vlan 300
switch(config-if-vl300)#ip multicast boundary 229.43.23.0/24
switch(config-if-vl300)#
```

- These commands create a standard ACL, then implements ACL in an ip multicast boundary command to configure two boundary subnets (225.123.0.0/16 and 239.120.10.0/24).

```
switch(config)#ip access-list standard mbac1
switch(config-std-acl-mbac1)#10 deny 225.123.0.0/16
switch(config-std-acl-mbac1)#20 deny 239.120.10.0/24
switch(config-std-acl-mbac1)#permit any
switch(config-std-acl-mbac1)#exit
switch(config)#interface vlan 200
switch(config-if-Vl200)#ip multicast boundary mbac1
switch(config-if-Vl200)#exit
switch(config)#
```

ip mroute

The **ip mroute** command specifies a candidate for the multicast reverse path forwarding (RPF) interface of any (S,G) multicast route, where the source falls within the given source or mask. Static mroutes are stored in a separate routing table, the Multicast Routing Information Base (MRIB).

Command Mode

Global Configuration

Command Syntax

```
ip mroute [<source-prefix>|<source-address> <mask>]
[<rpf-interface>|<rpf-neighbor>]
[admin distance]
no ip mroute [<source-prefix>|<source-address> <mask>]
[<rpf-interface>|<rpf-neighbor>]
default ip mroute [<source-prefix>|<source-address> <mask>]
[<rpf-interface>|<rpf-neighbor>]
```

Parameters

- ***source-prefix*** specifies the source prefix.
- ***source-address*** specifies the source address.
- ***mask*** specifies the address mask.
- ***rpf-interface*** specifies the multicast RPF interface.
- ***rpf-neighbor*** specifies the multicast RPF neighbor.

Examples

- This command selects the longest match when a source matches multiple static mroutes in the MRIB.

```
switch(config-as)#ip mroute 10.0.0.0/16 Ethernet 4
switch(config-as)#ip mroute 11.10.1.0/24 Ethernet 5
switch(config-as)#ip mroute 11.10.1.2/32 Ethernet 6
switch(config-as)#
```

- This command includes an administrative distance of 255 on Ethernet 5 with static mroute.

```
switch(config-as)#ip mroute 10.0.0.0/16 Ethernet 4
switch(config-as)#ip mroute 11.10.1.0/24 Ethernet 5 255
switch(config-as)#ip mroute 11.10.1.2/32 Ethernet 6
switch(config-as)#
```

ip multicast multipath none

The **ip multicast multipath none** command routes multicast ECMP traffic to the neighbor with the highest IPv4 address. By default, multicast traffic is load balanced by distributing packets over all ECMP links.

The **no ip multicast multipath none** and **default ip multicast multipath** commands restore the default behavior of randomly distributing multicast traffic over all ECMP links by removing the **ip multicast multipath none** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
ip multicast multipath none
no ip multicast multipath none
default ip multicast multipath none
```

Example

- This command configures the switch to route multicast traffic through the ECMP link to the neighbor with the highest IP address.

```
switch(config)#ip multicast multipath none
switch(config)#
```

ip multicast-routing

The **ip multicast-routing** command allows the switch to forward multicast packets. Multicast routing is disabled by default.

The **no ip multicast-routing** and **default ip multicast-routing** commands disables multicast routing removing the **ip multicast-routing** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
ip multicast-routing
no ip multicast-routing
default ip multicast-routing
```

Example

- This command enables multicast routing on the switch.

```
switch(config)#ip multicast-routing
switch(config)#
```

show ip mfib

The **show ip mfib** command displays information about the interfaces and the hardware-forwarded routes included in the IPv4 Multicast Forwarding Information Base (MFIB). Use the **show ip mfib software** command for software-forwarded routes.

Parameter options are available to filter output by group address or group and source addresses.

Command Mode

EXEC

Command Syntax

show ip mfib [*ROUTE*]

Parameters

- ***ROUTE*** routes displayed, filtered by multicast group and source IP addresses:
  - <no parameter> all multicast messages of the specified group are fast-switched.
  - *group_addr* multicast group IPv4 address.
  - *group_addr source address* two IPv4 addresses: multicast group and source addresses.

Example

- This command displays MFIB information for hardware-forwarded routes.

```
switch>show ip mfib
Activity poll time: 60 seconds
  239.255.255.250 172.17.26.25
    Vlan26 (iif)
    Vlan2028
    Cpu
      Activity 0:02:11 ago
  239.255.255.250 172.17.26.156
    Vlan26 (iif)
    Vlan2028
    Cpu
      Activity 0:02:11 ago
  239.255.255.250 172.17.26.190
    Vlan26 (iif)
    Vlan2028
    Cpu
      Activity 0:02:11 ago
  239.255.255.250 172.17.26.209
    Vlan26 (iif)
    Vlan2028
    Cpu
      Activity 0:02:11 ago
  239.255.255.250 172.17.26.223
    Vlan26 (iif)
    Vlan2028
    Cpu
      Activity 0:03:37 ago
switch>
```

show ip mfib software

The **show ip mfib software** command displays information about the interfaces and the software-forwarded routes included in the IPv4 Multicast Forwarding Information Base (MFIB). Use the **show ip mfib** command for hardware-forwarded routes.

Parameter options are available to filter output by group address or group and source address.

Command Mode

EXEC

Command Syntax

```
show ip mfib software [INFO_LEVEL][ROUTE]
```

Parameters

- ***INFO_LEVEL*** specifies the type of information displayed. Options include
  - <no parameter> Output displays packet reception counters.
  - **detail** Output displays packet reception counters and packet queued/dropped counters.
- ***ROUTE*** routes displayed, filtered by multicast group and source IP addresses:
  - <no parameter> all multicast messages of the specified group are fast-switched.
  - *group_addr* multicast group IPv4 address.
  - *group_addr source address* two IPv4 addresses: multicast group and source addresses.

Example

- This command displays MFIB information for software-forwarded routes.

```
switch>show ip mfib software
239.255.255.250 172.17.41.150
        Vlan3040 (iif)
        Packets Received: 18
        Bytes Received  : 9147
        RPF Failures    : 0
239.255.255.250 172.17.41.120
        Vlan3040 (iif)
        Packets Received: 6
        Bytes Received  : 966
        RPF Failures    : 0
switch>
```

- This command displays MFIB information for software-forwarded routes.

```
switch>show ip mfib software detail
239.255.255.250 172.17.41.150
        Vlan3040 (iif)
        Packets Received: 18
        Bytes Received  : 9147
        RPF Failures    : 0
        Packets Queued/Dropped : 0 / 0
239.255.255.250 172.17.41.120
        Vlan3040 (iif)
        Packets Received: 6
        Bytes Received  : 966
        RPF Failures    : 0
        Packets Queued/Dropped : 0 / 0
switch>
```

show ip mroute

The **show ip mroute** command displays information from the IP multicast routing table.

- **show ip mroute** displays information for all routes in the table.
- **show ip mroute** *gp_addr* displays information for the specified multicast group.

Command Mode

EXEC

Command Syntax

```
show ip mroute
show ip mroute gp_addr
```

Parameters

- *gp_addr* group IP address (dotted decimal notation).

Example

- This command displays the IP multicast routing table for the multicast group 225.1.1.11

```
switch>show ip mroute 225.1.1.1
PIM Sparse Mode Multicast Routing Table
Flags: E - Entry forwarding on the RPT, J - Joining to the SPT
       R - RPT bit is set, S - SPT bit is set
       W - Wildcard entry, X - External component interest
       I - SG Include Join alert rcvd, P - Ex-Prune alert rcvd
       H - Joining SPT due to policy, D - Joining SPT due to protocol
       Z - Entry marked for deletion
       A - Learned via Anycast RP Router
225.1.1.1
  172.28.1.100, 5d04h, flags: S
    Incoming interface: Vlan281
    Outgoing interface list:
      Port-Channel999
switch>
```

show ip mroute count

The **show ip mroute count** command displays IP multicast routing table statistics.

The **show ip mroute** command displays information from the IP multicast routing table.

Command Mode

EXEC

Command Syntax

show ip mroute count

Example

- This command displays IP multicast routing table statistics.

```
switch>show ip mroute count
IP Multicast Statistics
1 groups and 1 sources
Multicast routes: 1 (*,G), 1 (S,G)
Average of 1.00 sources per group
Maximum of 1 sources per group:
    228.24.12.1
switch>
```

Chapter 35

IGMP and IGMP Snooping

IP multicast is the transmission of data packets to multiple hosts through a common IP address. Networks use Internet Group Management Protocol (IGMP) to control the flow of layer 3 multicast traffic. Hosts request and maintain multicast group membership through IGMP messages. IGMP snooping is a layer 2 optimization for the layer 3 IGMP protocol that extracts lists of hosts receiving multicast group traffic by monitoring IGMP network packets.

These sections describe the Arista IGMP and IGMP snooping implementation.

- Section 35.1: Introduction lists supported IGMP and IGMP snooping features.
- Section 35.2: IGMP Protocols describes IGMP and IGMP snooping.
- Section 35.3: Configuring IGMP describes IGMP configuration tasks.
- Section 35.4: Configuring IGMP Snooping describes IGMP snooping configuration tasks.
- Section 35.5: IGMP Host Proxy describes IGMP host proxy configuration tasks.
- Section 35.6: IGMP and IGMP Snooping Commands lists IGMP and IGMP snooping commands.

35.1 Introduction

35.1.1 Supported Features

For a list of the IGMP features that each Arista switch platform supports, referred to the supported features table here: https://www.arista.com/en/support/product-documentation/supported-features.

35.2 IGMP Protocols

35.2.1 IGMP

Networks use Internet Group Management Protocol (IGMP) to control the flow of layer 3 multicast traffic. Hosts request and maintain multicast group membership through IGMP messages. Multicast routers use IGMP to maintain a membership list of active multicast groups for each attached network.

- IGMP version 1 is defined in RFC 1112. Hosts can join multicast groups without a method to leave a group. Routers use a timeout-based process to determine when hosts lose interest in a group.
- IGMP version 2 is defined in RFC 2236. Version 2 adds leave messages that hosts use to terminate group membership.
- IGMP version 3 is defined in RFC 4604. Version 3 allows hosts to specify IP addresses within a group from where they receive traffic. Traffic from all other group addresses is blocked from the host.

With respect to each of its attached networks, a multicast router is either a querier or non-querier. Each physical network contains only one querier. A network with more than one multicast router designates the router with the lowest IP address as its querier.

Queriers solicit group membership information by periodically sending General Query messages. Queriers also receive unsolicited messages from hosts joining or leaving a multicast group. When a querier receives a message from a host, it updates its membership list for the group referenced in the message and the network where the message originated.

Queriers forward multicasts from remote sources only to networks as specified by its membership list. If a querier does not receive a report from a network host for a specific group, it removes the corresponding entry from the table and discontinues forwarding multicasts for that group on the network. Queriers also send group-specific queries after receiving a leave request from a host to determine if the network still contains active multicast group members. If it does not receive a membership report during the period defined by the ***last member query response interval***, the querier removes the group-network entry from the membership list.

When a host receives a General Query, it responds with Membership Report messages for each of its multicast groups within the interval specified by the Max Response Time field in the query. IGMP suppresses multiple messages from different hosts on a network for the same group. Hosts send unsolicited Membership reports to join a multicast group and send leave messages to exit a group.

35.2.2 IGMP Snooping

IGMP snooping is a layer 2 switch process that extracts lists of hosts receiving multicast group traffic by monitoring IGMP network packets. The switch uses these lists to avoid flooding hosts with extraneous multicast traffic by sending group packets only to group members. Besides preventing local hosts from receiving traffic for groups they did not join, snooping prunes multicast traffic from links that do not contain IGMP clients.

When snooping is enabled, a switch examines IGMP packets sent between hosts connected to network switches and multicast routers (mrouters). When a switch finds an IGMP report from a multicast group recipient, it adds the recipient's port to the group multicast list. When the switch receives an IGMP leave, it removes the recipient's port from the list. Groups are removed upon the group timer expiry. When the switch finds an IGMP query packet or PIM hello packet from a multicast router, it adds the router's port to the port list for all multicast groups.

Snooping Querier

Snooping requires an IGMP querier in the network to create multicast group tables. An IGMP snooping querier performs the multicast router (mrouter) role when the network does not have a router. When the snooping querier is enabled on a VLAN, the switch periodically broadcasts IGMP queries and listens for IGMP Reports that indicate host group memberships.

Networks that contain multiple snooping queriers elect one as the querier, based on IP address. When IGMP snooping querier is enabled on a VLAN, the switch performs as a querier only when it is elected or it is the only snooping querier on the network.

L2 Report Flooding

L2 report flooding is an IGMP snooping feature that forwards membership report messages to specified ports. Relying on a single switch to maintain and send report messages can degrade performance. L2 report flooding addresses this by facilitating report message forwarding through any network port. This allows switches to bypass the querier when forwarding multicast traffic to its interested ports.

35.3 Configuring IGMP

This section describes the following configuration tasks:

- Section 35.3.1: Enabling IGMP
- Section 35.3.2: Configuring IGMP Settings

35.3.1 Enabling IGMP

Enabling PIM also enables IGMP on that interface. When the switch fills the multicast routing table, it only adds interfaces when the interface receives join messages from downstream devices or when the interface is directly connected to a member of the IGMP group.

By default, PIM and IGMP are disabled on an interface. The command enables PIM and IGMP on the configuration mode interface.

Example

- This command enables PIM and IGMP on VLAN interface 8.

```
switch(config)#interface vlan 8
switch(config-if-Vl8)#ip pim sparse-mode
switch(config-if-Vl8)#
```

35.3.2 Configuring IGMP Settings

An interface that runs IGMP uses default protocol settings unless otherwise configured. The switch provides commands that alter startup query, last member query, and normal query settings.

IGMP Version

The switch supports IGMP versions 1 through 3. The **ip igmp version** command configures the IGMP version on the configuration mode interface. Version 3 is the default IGMP version.

Example

- This command configures IGMP version 3 on VLAN interface 4

```
switch(config)#interface vlan 4
switch(config-if-Vl4)#ip igmp version 3
switch(config-if-Vl4)#
```

Startup Query

Membership queries are sent at an increased frequency immediately after an interface starts up to quickly establish the group state. Query count and query interval commands adjust the period between membership queries for a specified number of messages.

The **ip igmp startup-query-interval** command specifies the interval between membership queries that an interface sends immediately after it starts up. The **ip igmp startup-query-count** command specifies the number of queries that the switches sends from the interface at the startup interval rate.

Example

- These commands define a startup interval of 15 seconds for the first 10 membership queries sent from VLAN interface 12.

```
switch(config)#interface vlan 12
switch(config-if-Vl12)#ip igmp startup-query-interval 150
switch(config-if-Vl12)#ip igmp startup-query-count 10
switch(config-if-Vl12)#
```

Membership Queries

The router with the lowest IP address on a subnet sends membership queries as the IGMP querier. When a membership query is received from a source with a lower IP address, the router resets its query response timer. Upon timer expiry, the router begins sending membership queries. If the router subsequently receives a membership query originating from a lower IP address, it stops sending membership queries and resets the query response timer.

The **ip igmp query-interval** command configures the frequency at which the active interface, as an IGMP querier, sends membership query messages.

The **ip igmp query-max-response-time** command configures the time that a host has to respond to a membership query.

Example

- These commands define a membership query interval of 75 seconds and a query response timer reset value of 45 seconds for queries sent from VLAN interface 15.

```
switch(config)#interface vlan 15
switch(config-if-Vl15)#ip igmp query-interval 75
switch(config-if-Vl15)#ip igmp query-max-response-time 450
switch(config-if-Vl15)#
```

Last Member Query

When the querier receives an IGMP leave message, it verifies the group has no remaining hosts by sending a set of group-specific queries at a specified interval. If the querier does not receive a response to the queries, it removes the group state and discontinues multicast transmissions.

The **ip igmp last-member-query-count** (LMQC) command specifies the number of query messages the router sends in response to a group-specific or group-source-specific leave message.

The **ip igmp last-member-query-interval** command configures the transmission interval for sending group-specific or group-source-specific query messages to the active interface.

Example

- These commands program the switch to send 3 query messages, one every 25 seconds, when VLAN interface 15 receives an IGMP leave message.

```
switch(config)#interface vlan 15
switch(config-if-Vl15)#ip igmp last-member-query-interval 250
switch(config-if-Vl15)#ip igmp last-member-query-count 3
switch(config-if-Vl15)#
```

Static Groups

The **ip igmp static-group** command configures the configuration mode interface as a static member of the multicast group at the specified address. The router forwards multicast group packets through the interface without otherwise appearing or acting as a group member. No interface is a static member of a multicast group by default.

Note To become a static member of a multicast group, the switch must be the PIM designated router (DR) for the network. If it is not, you can use the **ip pim dr-priority** command to make it the DR by configuring its PIM DR value to be the highest on the network.

Example

- These commands configure VLAN interface 15 as the PIM designated router, then configure it as a static member of the multicast group at address 231.1.1.15 for multicast data packets that originate at 10.1.1.1.

```
switch(config)#interface vlan 15
switch(config-if-Vl15)#ip pim dr-priority 5000
switch(config-if-Vl15)#ip igmp static-group 231.1.1.45 10.1.1.1
switch(config-if-Vl15)#
```

35.4 Configuring IGMP Snooping

This section describes the following configuration tasks:

- Section 35.4.1: Enabling Snooping
- Section 35.4.2: Configuring Snooping Parameters
- Section 35.4.3: Snooping Querier
- Section 35.4.4: IGMP Snooping L2 Report Flooding
- Section 35.4.5: IGMP Snooping Filters

35.4.1 Enabling Snooping

The switch provides two control settings for snooping IGMP packets:

- Global settings control the availability of IGMP snooping on the switch. Snooping is globally enabled by default.
- Per-VLAN settings control IGMP on individual VLANs. If snooping is enabled on the VLAN, it follows the global snooping state.

The **ip igmp snooping** command controls the global snooping setting. The **ip igmp snooping vlan** command configures snooping on individual VLANs.

Examples

- This command globally enables snooping on the switch.

```
switch(config)#ip igmp snooping
switch(config)#
```

- This command disables snooping on VLANs 2 through 4.

```
switch(config)#no ip igmp snooping vlan 2-4
switch(config)#
```

35.4.2 Configuring Snooping Parameters

Specifying a Static Multicast Router Connection

The **ip igmp snooping vlan mrouter** command statically configures a port that connects to a multicast router to join all multicast groups. The port to the router must be in the specified VLAN range.

Snooping may not always be able to locate the IGMP querier. This command is for IGMP queriers that are known to connect through the network to a port on the switch.

Example

- This command configures the static connection to a multicast router through Ethernet port 3.

```
switch(config)#ip igmp snooping vlan 2 mrouter interface ethernet 3
switch(config)#
```

Adding a Port to a Multicast Group

The **ip igmp snooping vlan static** command adds an a port to a multicast group. The IP address must be an unreserved IPv4 multicast address. The interface to the port must be in the specified VLAN range.

Example

- This command configures the static connection to a multicast group at 237.2.1.4 through Ethernet port 3.

```
switch(config)#ip igmp snooping vlan 7 static 237.2.1.4 interface ethernet 3
switch(config)#
```

Robustness Variable

The robustness variable specifies the number of unacknowledged snooping queries that a switch sends before removing the recipient from the group list.

The **ip igmp snooping robustness-variable** command configures the robustness variable for all snooping packets sent from the switch. The default value is 2.

Example

- This command sets the robustness-variable value to 3.

```
switch(config)#ip igmp snooping robustness-variable 3
switch(config)#
```

Configuring Interface Startup Initial Query Times

The **ip igmp snooping interface-restart-query** command configures the interface startup initial query times in milliseconds. If nothing is configured, a default value of 2000 milliseconds is used. Issuing the command replaces any values already configured. Multiple values may be input in a single command; this makes the mechanism more resilient in the case of dropped packets.

Examples

- This command configures interfaces to send IGMP queries at 1000, 2000, and 4000 milliseconds (i.e., 1 second, 2 seconds, and 4 seconds) after an interface restart or spanning tree change.

```
switch(config)#ip igmp snooping interface-restart-query 1000 2000 4000
switch(config)#
```

Example

- This command configures interfaces to send a single IGMP querie 5000 milliseconds (i.e., 5 seconds) after an interface restart or spanning tree change.

```
switch(config)#ip igmp snooping interface-restart-query 5000
switch(config)#
```

35.4.3 Snooping Querier

The IGMP snooping querier supports snooping by sending layer 2 membership queries to hosts attached to the switch. Note that if IGMP snooping is enabled, QoS will not apply to IGMP packets.

35.4.3.1 Enabling the Snooping Querier

Enabling the snooping querier on an interface requires the explicit configuration of a global querier address or a local querier address for the interface. See Section 35.4.3.2.

The switch provides two control settings for controlling the snooping querier:

- The global setting controls the querier on VLANs for which there is no snooping querier command.
- VLAN querier settings take precedence over the global querier setting.

The **ip igmp snooping querier** command controls the global querier setting. When enabled globally, the querier is controlled on individual VLANs through the **ip igmp snooping vlan querier** command.

The **ip igmp snooping vlan querier** command controls the querier for the specified VLANs. VLANs follow the global querier setting unless overridden by one of these commands:

- **ip igmp snooping vlan querier** enables the querier on specified VLANs.
- **no ip igmp snooping vlan querier** disables the querier on specified VLANs.

Example

- These commands globally enables the snooping querier on the switch, explicitly disables snooping on VLANs 1-4, and explicitly enables snooping on VLANs 5-8.

```
switch(config)#ip igmp snooping querier
switch(config)#no ip igmp snooping vlan 1-4 querier
switch(config)#ip igmp snooping vlan 5-8 querier
switch(config)#
```

- This command removes the querier setting for VLANs 3-6:

```
switch(config)#default ip igmp snooping vlan 3-6 querier
switch(config)#
```

Globally Set the Snooping Querier Version

The **ip igmp snooping querier version** command configures the IGMP snooping querier version. Version 3 is the default IGMP snooping version.

Example

- This command gloablley configures IGMP snooping querier version 2.

```
switch(config)#ip igmp snooping querier version 2
switch(config)#
```

The **ip igmp snooping vlan querier version** command configures IGMP globally on the VLAN. Version 3 is the default IGMP snooping version.

Example

- This command configures IGMP snooping vlan querier version VLAN 5.

```
switch(config)#ip igmp snooping vlan 5 querier version 2
switch(config)#
```

35.4.3.2 Configuring Snooping Querier Parameters

Querier Address

The switch provides two IP addresses for setting the querier source:

- The global address is used by VLANs for which there is no querier address command.
- VLAN querier address settings take precedence over the global querier address.

The snooping querier address specifies the source IP address for IGMP snooping query packets that the switch transmits. The source address is also used to elect a snooping querier when the subnet contains multiple snooping queriers.

The default global querier address is not defined. When the configuration includes a snooping querier, a querier address must be defined globally or for each interface that enables a querier.

The **ip igmp snooping querier address** command sets the global querier source IP address for the switch. VLANs use the global address unless overwritten with the **ip igmp snooping vlan querier address** command. The default global address is not defined.

The **ip igmp snooping vlan querier address** command sets the source IP address for query packets transmitted from the specified VLAN. This command overrides the **ip igmp snooping querier address** for the specified VLAN.

Examples

- This command sets the source IP address for query packets that the switch transmits to 10.1.1.41

```
switch(config)#ip igmp snooping querier address 10.1.1.41
switch(config)#
```

- This command sets the source IP address for query packets that VLAN 2 transmits to 10.14.1.1.

```
switch(config)#ip igmp snooping vlan 2 querier address 10.14.1.1
switch(config)#
```

Membership Query Interval

The query interval is the period (seconds), between IGMP Membership Query message transmissions. The interval ranges from 5 to 3600 seconds.

The **ip igmp snooping querier query-interval** command specifies the global query interval for packets the switch sends as a snooper querier. . The default global setting is 125 seconds.

The **ip igmp snooping vlan querier query-interval** command specifies the query interval for packets sent from the snooping querier to the specified VLAN, overriding the global setting. VLANs that do not specify a query interval use the global setting.

Examples

- This command sets a query interval of 150 seconds for queries transmitted from VLANs for which a query interval is not configured.

```
switch(config)#ip igmp snooping querier query-interval 150
switch(config)#
```

- This command sets the query interval of 240 seconds for queries transmitted from VLAN 2.

```
switch(config)#ip igmp snooping vlan 2 querier query-interval 240
switch(config)#
```

Membership Query Response Interval

The Max Response Time field, in Membership Query messages, specifies the longest time a host can wait before responding with a Membership Report message. In all other messages, the sender sets the field to zero and the receiver ignores it. The switch provides two values for setting this field:

- The global value is used by VLANs for which there is no Max Response Time command.
- VLAN values take precedence over the global value for the specified VLAN.

The **ip igmp snooping querier max-response-time** command specifies the global Max Response Time value used in snooping query packets transmitted from the switch. Values range from 1 to 25 seconds with a default of 10 seconds. VLANs use the global setting unless overwritten with the **ip igmp snooping vlan querier max-response-time** command.

The **ip igmp snooping vlan querier max-response-time** command configures the Max Response Time field contents for packets transmitted from the specified VLAN, overriding the global setting.

Examples

- This command sets the maximum response time of 15 seconds for queries transmitted from VLANs for which a maximum response time is not configured.

```
switch(config)#ip igmp snooping querier max-response-time 15
switch(config)#
```

- This command sets a maximum response time of 5 seconds for queries that VLAN 2 transmits.

```
switch(config)#ip igmp snooping vlan 2 querier max-response-time 5
switch(config)#
```

Last Member Query

When the querier receives an IGMP leave message, it verifies the group has no remaining hosts by sending a set of group-specific queries at a specified interval. If the querier does not receive a response to the queries, it removes the group state and discontinues multicast transmissions.

The switch provides two values for setting this field:

- The global value is used by VLANs for which there is no last-member-query-interval defined.
- VLAN values take precedence over the global value for the specified VLAN.

The **ip igmp snooping querier last-member-query-interval** command specifies the global last-member-query-interval used in snooping query packets transmitted from the switch. This value is used for VLANs that do not have a value specified. Values range from 1 to 25 seconds with a global default of one second.

The **ip igmp snooping vlan querier last-member-query-interval** command configures the last-member-query-interval field contents for packets transmitted from the specified VLAN, overriding the global setting.

Example

- This command sets the global snooping querier last-member-query-interval to five seconds and the VLAN 10 last-member-query-interval to 12 seconds.

```
switch(config)#ip igmp snooping querier last-member-query-interval 5
switch(config)#ip igmp snooping vlan 10 querier last-member-query-interval 12
switch(config)#
```

Interface Restart Query Spoofing

When the port status (link status or spanning tree status) changes, an IGMP general query is spoofed based on the information of the last known IGMP querier. This facilitates faster network convergence time.

By default, interfaces wait 2000 milliseconds before sending the spoofed IGMP query. To configure the delay before the spoofed query is sent, use the **ip igmp snooping interface-restart-query** command. This setting is applied to all ports.

Example

- This command configures the switch to send general IGMP queries at 100 milliseconds, 200 milliseconds, and 300 milliseconds after interface restart or spanning tree status change.

```
switch(config)# ip igmp snooping interface-restart-query 100 200 300
```

35.4.4 IGMP Snooping L2 Report Flooding

L2 report flooding is an IGMP snooping feature that forwards membership report messages to specified ports. Report flooding is disabled by default and must be enabled globally before it can be enabled on individual interfaces.

The list of ports that can forward membership report messages must be explicitly configured. Commands are available to define lists of ports that are valid for all VLANs and port lists that are valid for specified VLAN ranges. Ports can forward membership reports only if they are configured to handle VLAN traffic, regardless of any report flooding configuration settings.

Enabling L2 Report Flooding

These commands enable L2 report flooding:

- **ip igmp snooping report-flooding** enables report flooding globally.
- **ip igmp snooping vlan report-flooding** enables report flooding on a specified VLAN range.

Example

- These commands enable L2 report flooding globally, and on VLANs 201-205.

```
switch(config)#ip igmp snooping report-flooding
switch(config)#ip igmp snooping vlan 201-205 report-flooding
switch(config)#
```

Configuring Forwarding Ports

These commands specify the ports that forward membership report messages:

- **ip igmp snooping report-flooding switch-port** configures ports globally.
- **ip igmp snooping vlan report-flooding switch-port** configures ports for a specified VLAN range.

Example

- These commands enable Ethernet ports 5-9 to forward reports on all VLANs and ports 12-15 on VLANs 201-205.

```
switch(config)#ip igmp snooping report-flooding switch-port ethernet 5-9
switch(config)#ip igmp snooping vlan 201-205 report-flooding switch-port ethernet
12-15
switch(config)#
```

35.4.5 IGMP Snooping Filters

IGMP Snooping filters assign IGMP profiles to Ethernet and port channel interfaces to control the multicast groups that the interfaces can join. An IGMP profile specifies a filter type and a list of address ranges. The address ranges comprise the multicast groups covered by the profile. The filter type determines an interface's accessibility to the multicast groups:

- Permit filters define the multicast groups the interface can join.
- Deny filters define the multicast groups the interface cannot join.

Profiles are created in IGMP-profile configuration mode, then applied to an interface in interface configuration mode.

The **ip igmp profile** command places the switch in IGMP profile configuration mode. The **permit / deny** and **range** commands specify the profile's filter type and address range. A profile may contain multiple range statements to define a discontiguous address range.

Example

- These commands create an IGMP profile named list_1 by entering IGMP-profile configuration mode, configure the profile to permit multicast groups 231.22.24.0 through 231.22.24.127, and return the switch to global configuration mode.

```
switch(config)#ip igmp profile list_1
switch(config-igmp-profile-list_1)#permit
switch(config-igmp-profile-list_1)#range 231.22.24.0 231.22.24.127
switch(config-igmp-profile-list_1)#exit
switch(config)#
```

The **ip igmp snooping filter** command applies an IGMP profile to the configuration mode interface.

Example

- These commands apply the ***list_1*** snooping profile to Ethernet interface 7.

```
switch(config)#interface ethernet 7
switch(config-if-Et7)#ip igmp snooping filter list_1
switch(config-if-Et7)#
```

35.4.5.1 Verifying IGMP Snooping

Show commands are available to display various configurations and IGMP snooping status. IGMP snooping that are viewable include:

- show ip igmp snooping
- show ip igmp snooping counters
- show ip igmp snooping querier
- show ip igmp snooping querier counters
- show ip igmp snooping querier membership

IGMP Snooping Status

The **show ip igmp snooping** command displays the switch's IGMP snooping configuration.

Example

- This command displays the switch's IGMP snooping configuration.

```
switch>show ip igmp snooping
   Global IGMP Snooping configuration:
-----------------------------------------
IGMP snooping                  : Enabled
Robustness variable            : 2

Vlan 1 :
----------
IGMP snooping                  : Enabled
Multicast router learning mode : pim-dvmrp

Vlan 20 :
----------
IGMP snooping                  : Enabled
Multicast router learning mode : pim-dvmrp

Vlan 2028 :

switch>
```

IGMP Snooping Counters

The **show ip igmp snooping counters** command displays the number of IGMP messages sent and received through each switch port. The display table sorts the messages by type.

Example

- This command displays the number of messages received on each port.

```
switch>show ip igmp snooping counters
                        Input                   |           Output
Port    Queries Reports  Leaves  Others  Errors|Queries Reports  Leaves  Others
-------------------------------------------------------------------------------
Cpu       15249  106599       4  269502       0   30242  102812     972    3625
Et1           0       0       0       0       0       0       0       0       0
Et2           0       6       1      26       0    5415       0       0     731
Et3           0   10905     222    1037       0   15246       0       0    1448
Et4           0   44475      21     288       0   15247       0       0    2199
Et5           0     355       0      39       0   15211       0       0    2446
Et6           0     475      13       0       0   15247       0       0    2487
Et7           0       0       0     151       0   15247       0       0    2336
Et8           0     578       6      75       0    2859       0       0     931
Et9           0       0       0      27       0   15247       0       0    2460
Et10          0   12523     345      54       0   15247       0       0    2433
Et11          0       0       0       0       0       0       0       0       0
Et12          0    4509      41      22       0   15247       0       0    2465
Et13          0     392      29     119       0   15247       0       0    2368
Et14          0      88       3       6       0   15247       0       0    2481
Et15          0   16779     556      72       0   15117       0       0      66
Et16          0    2484      13      66       0   15247       0       0    2421
Et17          0       0       0       0       0       0       0       0       0
Et18          0      20       6     160       0    3688       0       0     803
Et19          0    4110      17       0       0   15247       0       0    2487
Et20          0       0       0       0       0       0       0       0       0
Et21          0       0       0       0       0       0       0       0       0
Et22          0       0       0      52       0   15247       0       0    2435
Et23          0    5439     181     138       0   15247       0       0    2349
Et24          0    2251      21       4       0   15247       0       0    2483
Po1       45360  540670    8853  464900       0   15249  224751     618    2576
Po2           0  101399      58      17       0   15120       0       0    1121
Switch        0       0       0       0       0       0       0       0       0
```

IGMP Snooping Querier

The **show ip igmp snooping querier** command displays snooping querier configuration and status information. Command provides options to only include specific VLANs.

Example

- This command displays the querier IP address, version, and port servicing each VLAN.

```
switch>show ip igmp snooping querier
Vlan  IP Address       Version  Port
--------------------------------------
1     172.17.0.37      v2       Po1
20    172.17.20.1      v2       Po1
26    172.17.26.1      v2       Cpu
2028  172.17.255.29    v2       Po1
switch>
```

IGMP Snooping Querier Counters

The **show ip igmp snooping querier counters** command displays the counters from the querier, as learned through Internet Group Management Protocol (IGMP).

Example

- This command displays the counters from the querier.

```
switch>show ip igmp snooping querier counters
---------------------------------------------------------------------------
Vlan: 1    IP Addr: 100.0.0.1       Op State: Querier     Version:  v3

v1 General Queries Sent         :0
v1 Queries Received             :0
v1 Reports Received             :0
v2 General Queries Sent         :1
v2 Queries Received             :0
v2 Reports Received             :25
v2 Leaves Received              :0
v3 General Queries Sent         :655
v3 GSQ Queries Sent             :0
v3 GSSQ Queries Sent            :8
v3 Queries Received             :654
v3 Reports Received             :2385
Error Packets                   :0
Other Packets                   :0
switch>
```

IGMP Snooping Querier Membership

The **show ip igmp snooping querier membership** command displays the membership from the querier, as learned through Internet Group Management Protocol (IGMP).

Example

- This command displays the membership from the querier fro VLAN 1.

```
switch>show ip igmp snooping querier membership
---------------------------------------------------------------------------
Vlan: 1    Elected: 100.0.0.1       QQI: 125  QRV: 2  QRI: 10  GMI: 260

Groups           Mode  Ver  Num of Sources
---------------------------------------------------------------------------
10.0.0.2        EX    v3   0 []
10.0.0.3        IN    v3   2 [ 3.3.3.3, 3.3.3.4 ]
10.0.0.4        EX    v3   0 []
10.0.0.13       EX    v3   0 []
10.0.0.22       EX    v3   0 []
10.0.0.1        IN    v3   3 [ 5.6.7.9, 5.6.7.8, ... ]
switch>
```

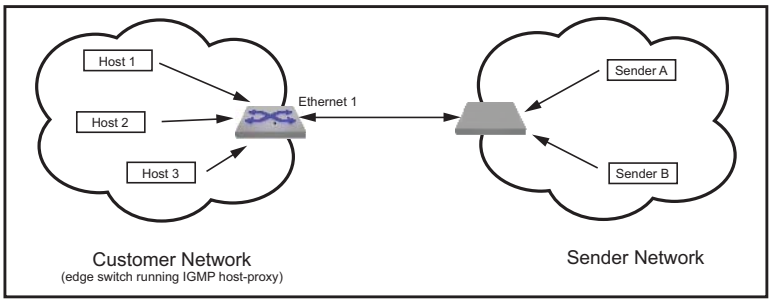
35.5 IGMP Host Proxy

Interfaces on the switch can be configured to serve as IGMP host proxies. An IGMP host proxy exchanges IGMP reports (joins/leaves) between networks whose connection does not support PIM along network boundaries.

35.5.1 IGMP Host Proxy Description

Figure 35-1 displays a typical IGMP host-proxy implementation. The customer network connects to the sender network through the edge switch's Ethernet 1 interface, which is configured as an IGMP host proxy. PIM is enabled within the sender and customer networks but not on the connection between the networks.

Figure 35-1: IP IGMP Host Proxy Implementation



The IGMP proxy agent sends unsolicited IGMP joins when a (S,G) or (*,G) entry arrives in the multicast routing table (mroute table). Subsequently, IGMP reports are sent when queries or group-specific queries arrive on the host proxy interface. When the customer network is void of active listeners, the connection eventually expires and the senders stop transmitting to the network.

IGMP host proxy requires the following:

- PIM MBR must be enabled on the interface.
- IP IGMP and IP multicast must be enabled.
- The switch must be an RP or in each host's RP path.
- Fast drop entries are required when there are no interested listeners for the group.

IGMP host proxy is configurable to filter for specific multicast groups and sources.

35.5.2 IGMP Host Proxy Configuration

Enabling IGMP Host Proxy

The IGMP host proxy service is configured on an interface using the **ip igmp host-proxy** command. When the host proxy is configured, it sends reports for (S,G) entries in the multicast routing (mroute) table if these are the only routes there; if there are any (*,G) entries, it sends reports only for these. To send reports for a specific group even when there is no (*, G) entry in the mroute table for that group, include the group address in the **ip igmp host-proxy** command. Multiple **ip igmp host-proxy statements** are required to specify multiple groups. The interval between IGMP reports is configured by **ip igmp host-proxy report-interval**.

Host Proxy IGMP Version and Source Filtering

IGMP host proxies can be configured with IGMP versions 1, 2, or 3, and use version 3 by default. When the host-proxy IGMP version is set to 3, the proxy can explicitly include or exclude source addresses. Otherwise, include/exclude configuration for source addresses is ignored. The IGMP version of unsolicited reports is specified with the **ip igmp host-proxy version** command. Reports that are triggered by IGMP queries, however, are sent in the same IGMP version as the received query. (An interface may also have a different IGMP version configured on it for other purposes using the **ip igmp version** command.)

Using ACLs

IGMP host proxy can also be enabled for the addresses defined by an ACL; if one or more groups are configured in addition to ACLs, the groups are processed first. Implicit deny in the ACL is ignored, but if the ACL includes an explicit deny rule, then the interface sends joins only to groups configured directly on the interface or included in a permit ACL. Deny rules take precedence over permit rules. If a group is configured with no filters and a host-proxy is configured with an ACL with rules having filters for the group, or configured with groups and source filters, then the filters are applied to the group.

Disabling Host Proxy or Removing an Individual Group or Source

The **no igmp host-proxy** command can be entered with group or source parameters to remove the specified group or source from the list. Entering the **no igmp host-proxy** command without specifying group or source disables the forwarding of all IGMP reports on the interface.

Examples

- These commands enable IGMP host proxy on Ethernet interface 17 for all multicast group addresses.

```
switch(config)#interface ethernet 17
switch(config-if-Et17)#ip igmp host-proxy
switch(config-if-Et17)#show active
 interface Ethernet17
   ip igmp host-proxy
switch(config-if-Et17)#
```

- These commands enable IGMP host proxy on Ethernet interface 18 for the multicast group at 231.10.10.1. The list of source addresses is not restricted.

```
switch(config)#interface ethernet 18
switch(config-if-Et18)#ip igmp host-proxy 231.10.10.1
switch(config-if-Et18)#show active
 interface Ethernet18
   ip igmp host-proxy 231.10.10.1
switch(config-if-Et18)#
```

- These commands enable IGMP host proxy on Ethernet interface 17 for the multicast group at 231.10.10.2. The list of source addresses only excludes 10.4.4.1 and 10.4.5.2.

```
switch(config)#interface ethernet 19
switch(config-if-Et19)#ip igmp host-proxy 231.10.10.2 exclude 10.4.4.1
switch(config-if-Et19)#ip igmp host-proxy 231.10.10.2 exclude 10.4.5.2
switch(config-if-Et19)#show active
 interface Ethernet19
   ip igmp host-proxy 231.10.10.2 exclude 10.4.5.2
   ip igmp host-proxy 231.10.10.2 exclude 10.4.4.1
switch(config-if-Et19)#
```

- These commands enable IGMP host proxy on Ethernet interface 16 for the multicast group at 231.10.10.3. The list of source address for this group only includes 10.5.5.1 and 10.5.5.2

```
switch(config)#interface ethernet 16
switch(config-if-Et16)#ip igmp host-proxy 231.10.10.3 include 10.5.5.1
switch(config-if-Et16)#ip igmp host-proxy 231.10.10.3 include 10.5.5.2
switch(config-if-Et16)#show active
 interface Ethernet16
   ip igmp host-proxy 231.10.10.3 include 10.5.5.2
   ip igmp host-proxy 231.10.10.3 include 10.5.5.1
   ip igmp host-proxy 231.10.10.3
switch(config-if-Et16)#
```

- These commands configure an IGMP host proxy interval of five seconds on port channel 100.

```
switch(config)#interface port-channel 100
switch(config-if-Po100)#ip igmp host-proxy report-interval 5
switch(config-if-Po100)#show active
 interface Port-Channel100
   ip igmp host-proxy
   ip igmp host-proxy report-interval 5
switch(config-if-Po100)#
```

- This command enables IGMP host proxy on Ethernet interface 17 for the group address(es) specified in ACL “acl1”.

```
switch(config-if-Et17)#ip igmp host-proxy access-list acl1
switch(config-if-Et17)#show active
 interface Ethernet17
   ip igmp host-proxy
   ip igmp host-proxy access-list acl2
switch(config-if-Et17)#
```

35.6 IGMP and IGMP Snooping Commands

IGMP Configuration Commands (Interface Configuration Mode)

- ip igmp last-member-query-count
- ip igmp last-member-query-interval
- ip igmp query-interval
- ip igmp query-max-response-time
- ip igmp router-alert
- ip igmp startup-query-count
- ip igmp startup-query-interval
- ip igmp static-group
- ip igmp static-group acl
- ip igmp static-group range
- ip igmp version

IGMP Clear Commands

- clear ip igmp group
- clear ip igmp statistics

IGMP Display Commands

- show ip igmp groups
- show ip igmp groups count
- show ip igmp interface
- show ip igmp static-groups
- show ip igmp static-groups acl
- show ip igmp static-groups group
- show ip igmp statistics

IGMP Snooping Configuration Commands (Global Configuration Mode)

- ip igmp profile
- ip igmp snooping
- ip igmp snooping querier
- ip igmp snooping querier address
- ip igmp snooping querier last-member-query-count
- ip igmp snooping querier last-member-query-interval
- ip igmp snooping querier max-response-time
- ip igmp snooping querier query-interval
- ip igmp snooping querier startup-query-count
- ip igmp snooping querier startup-query-interval
- ip igmp snooping querier version
- ip igmp snooping report-flooding
- ip igmp snooping report-flooding switch-port
- ip igmp snooping restart query-interval
- ip igmp snooping robustness-variable
- ip igmp snooping vlan
- ip igmp snooping vlan immediate-leave
- ip igmp snooping vlan max-groups
- ip igmp snooping vlan mrouter
- ip igmp snooping vlan querier
- ip igmp snooping vlan querier address
- ip igmp snooping vlan querier last-member-query-count

- ip igmp snooping vlan querier last-member-query-interval
- ip igmp snooping vlan querier max-response-time
- ip igmp snooping vlan querier query-interval
- ip igmp snooping vlan querier startup-query-count
- ip igmp snooping vlan querier startup-query-interval
- ip igmp snooping vlan querier version
- ip igmp snooping vlan report-flooding
- ip igmp snooping vlan report-flooding switch-port
- ip igmp snooping vlan static

IGMP Configuration Commands (Interface Configuration Mode)

- ip igmp snooping filter

IGMP Snooping Clear Commands

- clear ip igmp snooping counters

IGMP Snooping Display Commands

- show ip igmp profile
- show ip igmp snooping
- show ip igmp snooping counters
- show ip igmp snooping counters ethdev-pams
- show ip igmp snooping groups
- show ip igmp snooping groups count
- show ip igmp snooping mrouter
- show ip igmp snooping querier
- show ip igmp snooping querier counters
- show ip igmp snooping querier membership
- show ip igmp snooping report-flooding

IGMP Profile Configuration Mode Commands

- permit / deny
- range

IGMP Host Proxy Commands

- ip igmp host-proxy
- ip igmp host-proxy report-interval
- ip igmp host-proxy version
- show ip igmp host-proxy config-sanity
- show ip igmp host-proxy interface

clear ip igmp group

The **clear ip igmp group** command deletes IGMP cache entries as follows:

- **clear ip igmp group** all entries from the IGMP cache.
- **clear ip igmp group** *gp_addr* all entries for a specified multicast group.
- **clear ip igmp group interface** *int_id* all entries that include a specified interface.
- **clear ip igmp group** *gp_addr* **interface** *int_id* all entries for a specified interface in a specified group.

Command Mode

Privileged EXEC

Command Syntax

clear ip igmp group [*gp_addr***] [interface *INT_ID*]**

Parameters

- *gp_addr* multicast group IP address (dotted decimal notation).
- ***INT_ID*** interface name. Options include:
  - **ethernet** *e_num* Ethernet interface specified by *e_num*.
  - **loopback** *l_num* Loopback interface specified by *l_num*.
  - **management** *m_num* Management interface specified by *m_num*.
  - **port-channel** *p_num* Port-channel interface specified by *p_num*.
  - **vlan** *v_num* VLAN interface specified by *v_num*.
  - **vxlan** *vx_num* VXLAN interface specified by *vx_num*.

Examples

- This command deletes all IGMP cache entries for the multicast group 231.23.23.14.

```
switch#clear ip igmp group 231.23.23.14
switch#
```

- This command deletes IGMP cache entries for Ethernet interface 16 in multicast group 226.45.10.45.

```
switch#clear ip igmp group 226.45.10.45 interface ethernet 16
switch#
```

clear ip igmp snooping counters

The **clear ip igmp snooping counters** command resets the snooping message counters for the specified interface. The snooping counters for all interfaces are reset if the command does not include an interface name.

The **show ip igmp snooping counters** command displays the counter contents. See the **show ip igmp snooping counters** command description for a list of available snooping counters.

Command Mode

Privileged EXEC

Command Syntax

clear ip igmp snooping counters [*INT_NAME*]

Parameters

- ***INT_NAME*** interface name. Formats include:
  - **ethernet** *e_num* Ethernet interface specified by *e_num.*
  - **port-channel** *p_num* Port-channel interface specified by *p_num*.
  - **switch** virtual interface to an L2 querier.

Example

- This command clears the snooping counters for messages received on Ethernet interface 15.

```
switch(config)#clear ip igmp snooping counters ethernet 15
switch(config)#
```

clear ip igmp statistics

The **clear ip igmp statistics** command resets IGMP transmission statistic counters for the specified interface.

Command Mode

Privileged EXEC

Command Syntax

clear ip igmp statistics [*INTF_ID*]

Parameters

- ***INTF_ID*** interface name. Options include:
  - <no parameter> all interfaces.
  - **interface ethernet** *e_num* Ethernet interface specified by *e_num*.
  - **interface loopback** *l_num* Loopback interface specified by *l_num*.
  - **interface management** *m_num* Management interface specified by *m_num*.
  - **interface port-channel** *p_num* Port-channel interface specified by *p_num*.
  - **interface vlan** *v_num* VLAN interface specified by *v_num*.
  - **interface xlan** *vx_num* VXLAN interface specified by *vx_num*.

Examples

- This command resets IGMP transmission statistic counters on Ethernet 1 interface.

```
switch#clear ip igmp statistics interface ethernet 1
switch#
```

ip igmp host-proxy

The **ip igmp host-proxy** command enables the IGMP host proxy service on the configuration mode interface. The IGMP host proxy performs IGMP joins and leaves between networks that are directly connected by an exchange that does not support PIM on the network boundary.

The IGMP host proxy sends unsolicited IGMP join reports when an (S,G) or (*,G) entry arrives in the multicast routing (mroute) table. Reports are subsequently sent upon the arrival of queries on the interface. The interval between IGMP reports is configured through **ip igmp host-proxy report-interval**.

The **ip igmp host-proxy** command can also specify a group address; this ensures that reports are generated for the specified group even if there is no (*,G) entry in the mroute table for that group. Multiple **ip igmp host-proxy** statements are required to specify multiple groups.

When the host proxy IGMP version is set to 3 using the **ip igmp host-proxy version** command, the **ip igmp host-proxy** command can also include or exclude source addresses. These options are ignored when the interface runs host proxy IGMP version 1 or 2. Note that the IGMP version set using the **ip igmp version** command does not affect host proxy behavior.

An ACL can also be used in place of a group address by using the **access-list** option. If one or more groups are configured in addition to ACLs, the groups are processed first. Implicit deny in the ACL is ignored, but if the ACL includes an explicit deny rule, then the interface sends joins only to groups configured directly on the interface or included in a permit ACL. Deny rules take precedence over permit rules. If a group is configured with no filters and a host-proxy is configured with an ACL with rules having filters for the group, or configured with groups and source filters, then the filters are applied to the group.

The **no ip igmp host-proxy** and **default ip igmp host-proxy** commands remove the corresponding **ip igmp host-proxy** command from ***running-config***. When these commands do not include a group address, all **ip igmp host-proxy** statements are deleted. When inclusion or exclusion parameters are not specified, all statements with the specified group address are deleted.

Command Mode

Interface-Ethernet Configuration
Interface-Port-Channel Configuration
Interface-VLAN Configuration

Command Syntax

```
ip igmp host-proxy [GROUP_ADDRESS [SOURCE_ADDRESS]]|[access-list acl]
no ip igmp host-proxy [GROUP_ADDRESS [SOURCE_ADDRESS]]
default ip igmp host-proxy [GROUP_ADDRESS [SOURCE_ADDRESS]]
```

Parameters

- ***GROUP_ADDRESS*** IPv4 address of group address for which host proxy sends reports.
  - <*no parameter*> only groups for which there is a (*,G) entry in the mroute table.
  - *ipv4_address* IP address of multicast group (dotted decimal notation). This ensures that reports are generated for this group even if it does not have a (*,G) entry in the mroute table.
- ***SOURCE_ADDRESS*** IP address of a host that originates multicast data packets.
  - <*no parameter*> Proxy sends report for all received or configured groups regardless of source address.
  - **exclude** *ipv4_address* Proxy does not send reports for specified source address.
  - **include** *ipv4_address* Proxy always sends reports for specified source address.

Commands that list at least one parameter must specify a group address.
Parameters may be listed in any order.
When a command specifies include and exclude parameters, the exclude parameter is ignored.

- **access-list** *acl* specifies an access control list (ACL); a join is sent for all groups and/or sources obtained by processing the rules from all configured ACLs.

Guidelines

Multiple statements for a group address may be configured. The effect of entering a command depends on previously entered commands. The following describes command combination:

- **ip igmp host-proxy**: IGMP host proxy is enabled for all multicast groups and their source addresses. When enabled for all group addresses, the source address list cannot be restricted.
- **ip igmp host-proxy** *group_ipv4*: IGMP host proxy is enabled for a specified multicast group. The list of source addresses for this group is not restricted. Enabling host proxy for another group address requires another **ip igmp host-proxy** command.
- **ip igmp host-proxy** *group_ipv4* **exclude** *source_ipv4*: IGMP host proxy is enabled for the specified multicast group. Sources for this group include all addresses not in an exclude statement. Multiple source addresses for the group are excluded by multiple statements.
- **ip igmp host-proxy** *group_ipv4* **include** *source_ipv4*: IGMP host proxy is enabled for the specified group address for only the specified source address. Additional statements are required to include other source addresses for the group. The presence of one include parameter invalidates all exclude statements for the specified multicast group.
- **ip igmp host-proxy access-list** *acl*: IGMP host proxy is enabled for the addresses defined by the specified ACL. If one or more groups are configured in addition to ACLs, the groups are processed first. If the ACL has a “deny all” rule for a group, then this filter takes precedence over configurations with include /exclude keywords or permit/deny rules for that group. If a group is configured with no filters and a host-proxy is configured with an ACL with rules having filters for the group, or configured with groups and source filters, then the filters are applied to the group.

Example

- This command enables IGMP host proxy on Ethernet interface 17 for all multicast group addresses.

```
switch(config)#interface ethernet 17
switch(config-if-Et17)#ip igmp host-proxy
switch(config-if-Et17)#show active
 interface Ethernet17
   ip igmp host-proxy
switch(config-if-Et17)#
```

- This command enables IGMP host proxy on Ethernet interface 17 for the multicast group at 231.10.10.1. The list of source addresses is not restricted.

```
switch(config-if-Et17)#ip igmp host-proxy 231.10.10.1
switch(config-if-Et17)#show active
 interface Ethernet17
   ip igmp host-proxy 231.10.10.1
switch(config-if-Et17)#
```

- These commands enable IGMP host proxy on Ethernet interface 17 for the multicast group at 231.10.10.2. The list of source addresses only excludes 10.4.4.1 and 10.4.5.2.

```
switch(config-if-Et17)#ip igmp host-proxy 231.10.10.2 exclude 10.4.4.1
switch(config-if-Et17)#ip igmp host-proxy 231.10.10.2 exclude 10.4.5.2
switch(config-if-Et17)#show active
 interface Ethernet17
   ip igmp host-proxy 231.10.10.2 exclude 10.4.5.2
   ip igmp host-proxy 231.10.10.2 exclude 10.4.4.1
   ip igmp host-proxy 231.10.10.1
switch(config-if-Et17)#
```

- These commands enable IGMP host proxy on Ethernet interface 17 for the multicast group at 231.10.10.3. The list of source address for this group only includes 10.5.5.1 and 10.5.5.2

```
switch(config-if-Et17)#ip igmp host-proxy 231.10.10.3 include 10.5.5.1
switch(config-if-Et17)#ip igmp host-proxy 231.10.10.3 include 10.5.5.2
switch(config-if-Et17)#show active
 interface Ethernet17
   ip igmp host-proxy 231.10.10.3 include 10.5.5.2
   ip igmp host-proxy 231.10.10.3 include 10.5.5.1
   ip igmp host-proxy 231.10.10.3
   ip igmp host-proxy 231.10.10.2 exclude 10.4.5.2
   ip igmp host-proxy 231.10.10.2 exclude 10.4.4.1
   ip igmp host-proxy 231.10.10.1
switch(config-if-Et17)#
```

- This command enables IGMP host proxy on Ethernet interface 17 for the group address(es) specified in ACL “acl1”

```
switch(config-if-Et17)#ip igmp host-proxy access-list acl1
switch(config-if-Et17)#show active
 interface Ethernet17
   ip igmp host-proxy
   ip igmp host-proxy access-list acl2
switch(config-if-Et17)#
```

ip igmp host-proxy report-interval

The **ip igmp host-proxy report-interval** command configures the period between unsolicited join reports that the switch sends as an IGMP host proxy from the configuration mode interface to a sender network after a (S,G) or (*,G) entry arrives in the multicast route (mroute) table. When the interface receives a query in response, this interval is set to the **ip igmp last-member-query-interval**. This command also enables the host proxy on the configuration mode interface if it was not previously enabled.

The **no ip igmp host-proxy report-interval** and **default ip igmp host-proxy report-interval** commands reset the query interval to the default value of one second by removing the corresponding **ip igmp host-proxy report-interval** command from ***running-config***. The **no ip igmp host-proxy** and **default ip igmp host-proxy** commands also remove the corresponding **report-interval** command.

Command Mode

Interface-Ethernet Configuration
Interface-Port-Channel Configuration
Interface-VLAN Configuration

Command Syntax

```
ip igmp host-proxy report-interval period
no ip igmp host-proxy report-interval
default ip igmp host-proxy report-interval
```

Parameters

- *period* transmission interval (seconds) between consecutive reports.

  Value range: 1 (one second) to 31744 (8 hours, 49 minutes, 4 seconds). Default is 1 (one second).

Example

- These commands configures a IGMP host proxy interval of five seconds on port channel 100.

```
switch(config)#interface port-channel 100
switch(config-if-Po100)#ip igmp host-proxy report-interval 5
switch(config-if-Po100)#show active
 interface Port-Channel100
   ip igmp host-proxy
   ip igmp host-proxy report-interval 5
switch(config-if-Po100)#
```

ip igmp host-proxy version

The **ip igmp host-proxy version** command configures the version number to be used in unsolicited reports when the interface is serving as an IGMP host proxy. To configure the IGMP version used by the interface for other purposes, use the **ip igmp version** command instead.

The **no ip igmp host-proxy version** and **default ip igmp host-proxy version** commands reset the version to the default value of 3.

Command Mode

Interface-Ethernet Configuration
Interface-Port-Channel Configuration
Interface-VLAN Configuration

Command Syntax

```
ip igmp host-proxy version version_number
no ip igmp host-proxy version
default ip igmp host-proxy version
```

Parameters

- *version_number* values range from 1-3; default value is 3.

Example

- These commands configure the IGMP host proxy version on port channel interface 100 to 2.

```
switch(config)#interface port-channel 100
switch(config-if-Po100)#ip igmp host-proxy version 2
switch(config-if-Po100)#
```

ip igmp last-member-query-count

The **ip igmp last-member-query-count** command specifies the number of query messages the switch sends in response to a group-specific or group-source-specific leave message.

After receiving a message from a host leaving a group, the switch sends query messages at intervals specified by **ip igmp last-member-query-interval**. If the switch does not receive a response to the queries after sending the number of messages specified by this parameter, it stops forwarding messages to the host.

Setting the last member query count (LMQC) to 1 causes the loss of a single packet to stop traffic forwarding. While the switch can start forwarding traffic again after receiving a response to the next general query, the host may not receive that query for a period defined by **ip igmp query-interval**.

The **no ip igmp last-member-query-count** and **default ip igmp last-member-query-count** commands reset the LMQC to the default value by removing the corresponding **ip igmp last-member-query-count** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration
Interface-Port-Channel Configuration
Interface-VLAN Configuration

Command Syntax

```
ip igmp last-member-query-count number
no ip igmp last-member-query-count
default ip igmp last-member-query-count
```

Parameters

- *number* query message quantity. Values range from 1 to 3. Default is 2.

Example

- This command configures the last-member-query-count to 3 on VLAN interface 4.

```
switch(config)#interface vlan 4
switch(config-if-Vl4)#ip igmp last-member-query-count 3
switch(config-if-Vl4)#
```

ip igmp last-member-query-interval

The **ip igmp last-member-query-interval** command configures the switch's transmission interval for sending group-specific or group-source-specific query messages from the configuration mode interface.

When a switch receives a message from a host that is leaving a group it sends query messages at intervals set by this command. The **ip igmp startup-query-count** specifies the number of messages that are sent before the switch stops forwarding packets to the host.

If the switch does not receive a response after this period, it stops forwarding traffic to the host on behalf of the group, source, or channel.

The **no ip igmp last-member-query-interval** and **default ip igmp last-member-query-interval** commands reset the query interval to the default value of one second by removing the **ip igmp last-member-query-interval** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration
Interface-Port-Channel Configuration
Interface-VLAN Configuration

Command Syntax

```
ip igmp last-member-query-interval period
no ip igmp last-member-query-interval
default ip igmp last-member-query-interval
```

Parameters

- *period* transmission interval (deciseconds) between consecutive group-specific query messages.

  Value range: 10 (one second) to 317440 (8 hours, 49 minutes, 4 seconds). Default is 10 (one second).

Example

- This command configures the last member query interval of 6 seconds for VLAN interface 4.

```
switch(config)#interface vlan 4
switch(config-if-Vl4)#ip igmp last-member-query-interval 60
switch(config-if-Vl4)#
```

ip igmp profile

The **ip igmp profile** command places the switch in IGMP-profile configuration mode to configure an IGMP profile. IGMP profiles control the multicast groups that an interface can join.

Profiles consist of the filter type and an address range:

- Filter types specify accessibility to the listed address range:
  - Permit filters define the multicast groups the interface can join.
  - Deny filters define the multicast groups the interface cannot join.

  Profiles are deny filters by default.
- Address ranges specify a list of addresses and ranges:
  - In permit filters, permitted groups are specified by the address range.
  - In deny filters, all groups are permitted except those specified by the address range.

Implementing IGMP filtering affects IGMP report forwarding as follows:

- IGMPv2: Report is forwarded to mrouters for permitted groups and dropped for disallowed groups.
- IGMPv3: There may be multiple group records in a report.
  - No groups are allowed: The report is dropped.
  - All groups are allowed: The report is forwarded to mrouter ports as normal.
  - Some groups are allowed: A revised report is forwarded to mrouter ports.

    The revised report includes records for the allowed group addresses with the same source MAC and IP addresses.

The **no ip igmp profile** and **default ip igmp profile** commands delete the specified IGMP profile from ***running-config***.

IGMP-profile configuration mode is not a group change mode; ***running-config*** is changed immediately upon entering commands. Exiting IGMP-profile configuration mode does not affect the configuration. The **exit** command returns the switch to global configuration mode.

Command Mode

Global Configuration

Command Syntax

```
ip igmp profile profile_name
no ip igmp profile profile_name
default ip igmp profile profile_name
```

Parameters

- *profile_name* name of the IGMP profile.

Commands Available in igmp-profile Configuration Mode

- **permit / deny**
- **range**

Related Commands

- **ip igmp snooping filter** applies an IGMP snooping filter to a configuration mode interface.

Example

- These commands enter IGMP-profile configuration mode and configure the profile as a permit list.

```
switch(config)#ip igmp profile list_1
switch(config-igmp-profile-list_1)#permit
switch(config-igmp-profile-list_1)#
```

ip igmp query-interval

The **ip igmp query-interval** command configures the frequency at which the configuration mode interface, as an IGMP querier, sends host-query messages.

An IGMP querier sends host-query messages to discover the multicast groups that have members on networks attached to the interface. The switch implements a default query interval of 125 seconds.

The **no ip igmp query-interval** and **default ip igmp query-interval** commands reset the IGMP query interval to the default value of 125 seconds by removing the **ip igmp query-interval** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration
Interface-Port-Channel Configuration
Interface-VLAN Configuration

Command Syntax

```
ip igmp query-interval period
no ip igmp query-interval
default ip igmp query-interval
```

Parameters

- *period* interval (seconds) between IGMP query messages. Values range from 1 to 3175 (52 minutes, 55 seconds). Default is 125.

Example

- This command configures the query-interval of 2 minutes, 30 seconds for VLAN interface 4.

```
switch(config)#interface vlan 4
switch(config-if-Vl4)#ip igmp query-interval 150
switch(config-if-Vl4)#
```

ip igmp query-max-response-time

The **ip igmp query max-response-time** command configures the ***query-max-response-time*** variable for the configuration mode interface. This variable is used to set the Max Response Time field in outbound Membership Query messages. Max Response Time specifies the maximum period a recipient can wait before responding with a Membership Report.

The router with the lowest IP address on a subnet sends membership queries as the IGMP querier. When a membership query is received from a source with a lower IP address, the router resets its query response timer. Upon timer expiry, the router begins sending membership queries. If the router subsequently receives a membership query originating from a lower IP address, it stops sending membership queries and resets the query response timer.

The **no ip igmp query-max-response-time** and **default ip igmp query-max-response-time** commands restore the default query-max-response-time of 10 seconds for the configuration mode interface by removing the corresponding **ip igmp query max-response-time** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration
Interface-Port-Channel Configuration
Interface-VLAN Configuration

Command Syntax

```
ip igmp query-max-response-time period
no ip igmp query-max-response-time
default ip igmp query-max-response-time
```

Parameters

- *period* maximum response time (deciseconds). Values range from 1 to 31744 (52 minutes, 54 seconds). Default is 100 (ten seconds).

Example

- This command configures the query-max-response-time of 18 seconds for VLAN interface 4.

```
switch(config)#interface vlan 4
switch(config-if-Vl4)#ip igmp query-max-response-time 180
switch(config-if-Vl4)#
```

ip igmp router-alert

The **ip igmp router-alert** command configures the switch disposition of inbound IGMP packets to the configuration mode interface based on the presence of the router-alert option in the IP header. By default, the port accepts all IGMP packets that arrive on the local subnet and rejects all other packets that arrive without the router-alert option.

The command provides three IGMP packet disposition options:

- mandatory: packets are accepted only when router-alert is present.
- optional: packets are accepted regardless of router-alert presence.
- optional connected: packets are accepted from the same subnet; other packets require router-alert.

The **no ip igmp router-alert** and **default ip igmp router-alert** commands reset the default setting of ***optional connected*** on the configuration mode interface by removing the corresponding **ip igmp router-alert** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration
Interface-Port-Channel Configuration
Interface-VLAN Configuration

Command Syntax

```
ip igmp router-alert DISPOSITION
no ip igmp router-alert
default ip igmp router-alert
```

Parameters

- ***DISPOSITION*** IGMP packet disposition method. Options include:
  - **mandatory** Rejects packets if router-alert is not present.
  - **optional** Accepts packets regardless of router-alert presence.
  - **optional connected** Accepts packets from same subnet. Other packets require router-alert.

Example

- This command configures the switch to accept IGMP packets on Ethernet interface 8 only if the IP header contains router alert.

```
switch(config)#interface ethernet 8
switch(config-if-Et8)#ip igmp router-alert mandatory
switch(config-if-Et8)#show active
interface Ethernet8
   load-interval 60
   ip igmp router-alert mandatory
switch(config-if-Et8)#
```

ip igmp snooping

The **ip igmp snooping** command enables snooping globally. By default, global snooping is enabled.

When global snooping is enabled, **ip igmp snooping vlan** enables or disables snooping on individual VLANs. When global snooping is disabled, snooping cannot be enabled on individual VLANs.

QoS cannot be used for IGMP packets when IGMP snooping is enabled.

The **no ip igmp snooping** command disables global snooping. The **default ip igmp snooping** command restores the global snooping default setting of enabled by removing the **ip igmp snooping** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
ip igmp snooping
no ip igmp snooping
default ip igmp snooping
```

Example

- This command globally enables snooping on the switch.

```
switch(config)#ip igmp snooping
switch(config)#
```

ip igmp snooping filter

The **ip igmp snooping filter** command applies the specified IGMP snooping profile to the configuration mode interface. An IGMP snooping profile specifies the multicast groups that an interface may join. Profiles consist of the filter type and an address range:

- Filter type: Specifies accessibility to the listed address range:
  - Permit filters define the multicast groups the interface can join.
  - Deny filters define the multicast groups the interface cannot join.
- Address range: Specifies a list of addresses and ranges.
  - In permit filters, the permitted groups are specified by the address range.
  - In deny filters, all groups are permitted except those specified by the address range.

An interface without a snooping profile assignment may join any multicast group.

Snooping profiles are configured in IGMP-profile configuration mode (**ip igmp profile**).

The **no ip igmp snooping filter** and **default ip igmp snooping filter** commands restore the default setting of allowing an interface to join any multicast group by deleting the corresponding **ip igmp snooping filter** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration
Interface-Port-Channel Configuration

Command Syntax

```
ip igmp snooping filter profile_name
no ip igmp snooping filter [profile_name]
default ip igmp snooping filter [profile_name]
```

Parameters

- *profile_name* name of profile assigned to interface.

Example

- This command applies the ***list_1*** snooping profile to Ethernet interface 7.

```
switch(config)#interface ethernet 7
switch(config-if-Et7)#ip igmp snooping filter list_1
switch(config-if-Et7)#
```

ip igmp snooping interface-restart-query

The **ip igmp snooping interface-restart-query** command configures the interface startup initial query time used for IGMP query spoofing. When an interface restarts or there is a change to the spanning tree, the interface will send general IGMP queries after this interval. The query is based on the information of the last known IGMP querier, and serves to facilitate faster network convergence times.

Multiple values can be configured with a single command; issuing the command again replaces any previously configured value(s).

The **no ip igmp snooping interface-restart-query** and **default ip igmp snooping interface-restart-query** commands restore the default setting of 2000 milliseconds by deleting the corresponding **ip igmp snooping interface-restart-query** command from ***running-config***.

Command Mode

General Configuration

Command Syntax

```
ip igmp snooping interface-restart-query query_time
no ip igmp snooping interface-restart-query
default ip igmp snooping interface-restart-query
```

Parameters

- *query_time* interval (in milliseconds) after an interface restart or spanning tree change at which the interface will send general IGMP queries. Values range from 100 to 50000 milliseconds; default is 2000.

Example

- This command configures interfaces to send IGMP queries at 100, 200, and 300 milliseconds after an interface restart or spanning tree change.

```
switch(config)#ip igmp snooping interface-restart-query 100 200 300
switch(config)#
```

ip igmp snooping querier

The **ip igmp snooping querier** command enables the snooping querier globally, which controls the querier for VLANs that are not configured with a snooping querier command. The **ip igmp snooping vlan querier** command controls the querier on individual VLANs.

The IGMP snooping querier supports snooping by sending layer 2 membership queries to hosts attached to the switch. The snooping querier is functional on VLANs where hosts receive IP multicast traffic without access to a network IP multicast router. A snooping querier avoids flooding multicast packets in the VLAN by querying for hosts and routers.

The IGMP snooping querier is functional on VLANs that meet these criteria:

- Snooping is enabled.
- The corresponding SVI (VLAN interface) is active.
- The VLAN's querier IP address or the global querier IP address is configured.

The **no ip igmp snooping querier** and **default ip igmp snooping querier** commands disable the snooping querier globally by removing the **ip igmp snooping querier** statement from ***running-config***. The snooping querier is globally disabled by default.

Command Mode

Global Configuration

Command Syntax

```
ip igmp snooping querier
no ip igmp snooping querier
default ip igmp snooping querier
```

Guidelines

- Enabling a querier after it was disabled is equivalent to establishing a new querier.
- Changing the querier's IP address is equivalent to establishing a new querier.

Example

- This command globally enables the snooping querier on the switch.

```
switch(config)#ip igmp snooping querier
switch(config)#
```

ip igmp snooping querier address

The **ip igmp snooping querier address** command sets the global querier source IP address, which specifies the source address for packets transmitted from VLANs for which a querier address (**ip igmp snooping vlan querier address**) is not configured. To use a snooping querier, an address must be explicitly configured globally or for the VLAN.

The switch does not define a default global querier address.

The **no ip igmp snooping querier address** and **default ip igmp snooping querier address** commands remove the global querier address command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
ip igmp snooping querier address ipv4_address
no ip igmp snooping querier address
default ip igmp snooping querier address
```

Parameters

- *ipv4_address* source IPv4 address.

Example

- This command sets the source IP address to 10.1.1.41 for query packets transmitted from the switch.

```
switch(config)#ip igmp snooping querier address 10.1.1.41
switch(config)#
```

ip igmp snooping querier last-member-query-count

The **ip igmp snooping querier last-member-query-count** command configures the global ***IGMP snooping querier last member query count*** (LMQC) value. LMQC specifies the number of query messages the switch sends in response to group-specific or group-source-specific leave messages it receives from a host; the transmission frequency is specified by ***IGMP snooping querier last member query interval***. The switch stops forwarding messages to the host if it does not receive a response to these query messages.

Setting LMQC to 1 causes the loss of one packet to stop traffic forwarding. While the switch can start forwarding traffic again after receiving a response to the next general query, the host may not receive that query for a period defined by **ip igmp snooping querier query-interval*.***

VLANs use the global value when they are not assigned a value (**ip igmp snooping vlan querier last-member-query-count**) . VLAN commands take precedence over the global value. The default global value is specified by the robustness variable (**ip igmp snooping robustness-variable**).

The **no igmp snooping querier last-member-query-count** and **default igmp snooping querier last-member-query-count** commands reset the LMQC to the default value by removing the corresponding **ip igmp snooping querier last-member-query-count** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
ip igmp snooping querier last-member-query-count number
no ip igmp snooping querier last-member-query-count
default ip igmp snooping querier last-member-query-count
```

Parameters

- *number* query message quantity. Value ranges from 1 to 3. Default is set by robustness-variable.

Example

- This command configures the global last-member-query-count to 3.

```
switch(config)#ip igmp snooping querier last-member-query-count 3
switch(config)#show ip igmp snooping querier status
   Global IGMP Querier status
-----------------------------------------------------
admin state                     : Disabled
source IP address               : 0.0.0.0
query-interval (sec)            : 125.0
max-response-time (sec)         : 10.0
querier timeout (sec)           : 255.0
last-member-query-interval (sec) : 1.0
last-member-query-count         : 3
startup-query-interval (sec)    : 31.25 (query-interval/4)
startup-query-count             : 2 (robustness)

Vlan Admin    IP             Query    Response Querier Operational Ver
     State                   Interval Time     Timeout State
-------------------------------------------------------------------------
1    Disabled 0.0.0.0        125.0    10.0     255.0   Non-Querier v2
100  Disabled 0.0.0.0        125.0    10.0     255.0   Non-Querier v2
101  Disabled 0.0.0.0        125.0    10.0     255.0   Non-Querier v2
switch(config)#
```

ip igmp snooping querier last-member-query-interval

The **ip igmp snooping querier last-member-query-interval** command sets the global IGMP snooping last member query interval. The default interval is one second.

A multicast host sends an IGMP leave report when it leaves a group. To determine if the host was the last group member, the leave message recipient sends an IGMP query. The last-member-query-interval determines when the group record is deleted if no subsequent reports are received.

VLANs not assigned a ***last member query interval*** value (**ip igmp snooping vlan querier last-member-query-interval**) use the global value. VLAN commands take precedence over the global value.

The **no ip igmp snooping querier last-member-query-interval** and **default ip igmp snooping querier last-member-query-interval** commands reset the **last-member-query-interval** value the default interval of one second by removing the **ip igmp snooping querier last-member-query-interval** statement from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
ip igmp snooping querier last-member-query-interval period
no ip igmp snooping querier last-member-query-interval
default ip igmp snooping querier last-member-query-interval
```

Parameters

- *period* last member query interval (seconds). Value ranges from 1 to 25. Default is one second.

Related Commands

- **ip igmp snooping vlan querier last-member-query-interval** assign a last member query interval value to the specified VLANs.

Example

- This command sets the IGMP snooping querier last-member-query-interval to five seconds.

```
switch(config)#ip igmp snooping querier last-member-query-interval 5
switch(config)#
```

ip igmp snooping querier max-response-time

The **ip igmp snooping querier max-response-time** command specifies the global ***max-response-time*** value. The switch uses ***max-response-time*** to set the Max Response Time field in outbound Membership Query messages. Max Response Time specifies the maximum period a recipient can wait before responding with a Membership Report.

VLANs not assigned a ***max-response-time*** value (**ip igmp snooping vlan querier max-response-time**) use the global value. VLAN commands take precedence over the global value.

Values range from 1 to 25 seconds. The default global value is 10 seconds.

The **no ip igmp snooping querier max-response-time** and **default ip igmp snooping querier max-response-time** commands restore the global ***max-response-time*** default value by removing the **ip igmp snooping querier max-response-time** statement from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
ip igmp snooping querier max-response-time resp_sec
no ip igmp snooping querier max-response-time
default ip igmp snooping querier max-response-time
```

Parameters

- *resp_sec* ***max-response-time*** value (seconds). Values range from 1 to 25. Default (global) is 10.

Example

- This command sets the global max-response-time to 15 seconds.

```
switch(config)#ip igmp snooping querier max-response-time 15
switch(config)#
```

ip igmp snooping querier query-interval

The **ip igmp snooping querier query-interval** command sets the global query interval. This command also sets the query-interval of IGMP Snooping when using IGMP version 2. Values range from 5 to 3600 seconds. The default global value is 125 seconds. The query interval is the period between IGMP Membership Query messages sent from the querier. The global value specifies the query interval for VLANs with no query-interval command.

VLANs not assigned a ***query interval*** value (**ip igmp snooping vlan querier query-interval**) use the global value. VLAN commands take precedence over the global value.

The **no ip igmp snooping querier query-interval** and **default ip igmp snooping querier query-interval** commands reset the global query-interval value to 125 seconds by removing the **ip igmp snooping querier query-interval** statement from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
ip igmp snooping querier query-interval query_sec
no ip igmp snooping querier query-interval
default ip igmp snooping querier query-interval
```

Parameters

- *query_sec* query interval (seconds). Values range from 5 to 3600. Default (global) is 125.

Example

- This command sets the global query interval to 150 seconds.

```
switch(config)#ip igmp snooping querier query-interval 150
switch(config)#
```

ip igmp snooping querier startup-query-count

The **ip igmp snooping querier startup-query-count** command configures the global ***startup query count*** value. The ***startup query count*** specifies the number of query messages that the querier sends on a VLAN during the ***startup query interval*** (**ip igmp snooping querier startup-query-interval**).

When snooping is enabled, the group state is more quickly established by sending query messages at a higher frequency. The ***startup-query-interval*** and ***startup-query-count*** parameters define the startup period by defining the number of queries to be sent and transmission frequency for these messages.

VLANs use the global ***startup query count*** value when they are not assigned a value (**ip igmp snooping vlan querier startup-query-count**). VLAN commands take precedence over the global value. The default global value is specified by the robustness variable (**ip igmp snooping robustness-variable**).

The **no ip igmp snooping querier startup-query-count** and **default ip igmp snooping querier startup-query-count** commands restore the default ***startup-query-count*** value by removing the corresponding **ip igmp snooping querier startup-query-count** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
ip igmp snooping querier startup-query-count number
no ip igmp snooping querier startup-query-count
default ip igmp snooping querier startup-query-count
```

Parameters

- *number* global startup query count. Value ranges from 1 to 3.

Example

- These commands configure the global startup query count value of 2, then displays the status of the snooping querier.

```
switch(config)#ip igmp snooping querier startup-query-count 2
switch(config)#show ip igmp snooping querier status
   Global IGMP Querier status
-------------------------------------------------------
admin state                     : Disabled
source IP address               : 0.0.0.0
query-interval (sec)            : 125.0
max-response-time (sec)         : 10.0
querier timeout (sec)           : 255.0
last-member-query-interval (sec) : 1.0
last-member-query-count         : 2 (robustness)
startup-query-interval (sec)    : 31.25 (query-interval/4)
startup-query-count             : 2

Vlan Admin    IP             Query    Response Querier Operational Ver
     State                   Interval Time     Timeout State
-----------------------------------------------------------------------
1    Disabled 0.0.0.0        125.0    10.0     255.0   Non-Querier v2
100  Disabled 0.0.0.0        125.0    10.0     255.0   Non-Querier v2
101  Disabled 0.0.0.0        125.0    10.0     255.0   Non-Querier v2
switch(config)#
```

ip igmp snooping querier startup-query-interval

The **ip igmp snooping querier startup-query-interval** command configures the global startup query interval value. The ***startup query interval*** specifies the period between query messages that the querier sends upon startup.

When snooping is enabled, the group state is more quickly established by sending query messages at a higher frequency. The ***startup-query-interval*** and ***startup-query-count*** parameters define the startup period by defining the number of queries to be sent and transmission frequency for these messages.

VLANs use the global ***startup query interval*** value when they are not assigned a value (**ip igmp snooping vlan querier startup-query-interval**). VLAN commands take precedence over the global value. The default global value equals the query interval divded by four. (**ip igmp snooping querier query-interval**).

The **no ip igmp snooping querier startup-query-interval** and **default ip igmp snooping querier startup-query-interval** commands restore the default method of specifying the startup query interval by removing the corresponding **ip igmp snooping querier startup-query-interval** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
ip igmp snooping querier startup-query-interval period
no ip igmp snooping querier startup-query-interval
default ip igmp snooping querier startup-query-interval
```

Parameters

- *period* startup query interval (seconds). Value ranges from 1 to 3600 (1 hour).

Example

- This command configures the startup query count of one minute for VLAN interface 4.

```
switch(config)#ip igmp snooping querier startup-query-interval 40
switch(config)#show ip igmp snooping querier status
   Global IGMP Querier status
-------------------------------------------------------
admin state                     : Enabled
source IP address               : 0.0.0.0
query-interval (sec)            : 125.0
max-response-time (sec)         : 10.0
querier timeout (sec)           : 255.0
last-member-query-interval (sec) : 1.0
last-member-query-count         : 2 (robustness)
startup-query-interval (sec)    : 40.0
startup-query-count             : 2

Vlan Admin    IP             Query    Response Querier Operational Ver
     State                   Interval Time     Timeout State
-----------------------------------------------------------------------
1    Enabled  0.0.0.0        125.0    10.0     255.0   Non-Querier v3
100  Enabled  0.0.0.0        125.0    10.0     255.0   Non-Querier v3
101  Enabled  0.0.0.0        125.0    10.0     255.0   Non-Querier v3
switch(config)#
```

ip igmp snooping querier version

The **ip igmp snooping querier version** command configures the Internet Group Management Protocol (IGMP) snooping querier version on the configuration mode interfaces. Version 3 is the default IGMP version.

IGMP is enabled by the **ip pim sparse-mode** command. The **ig igmp snooping querier version** command does not affect the IGMP enabled status.

The **no ip igmp snooping querier version** and **default ip igmp snooping querier version** commands restore the configuration mode to IGMP version 3 by removing the **ip igmp snooping querier version** statement from ***running-config.***

Command Mode

Global Configuration

Command Syntax

```
ip igmp snooping querier version version_number
no ip igmp snooping querier version
default ip igmp snooping querier version
```

Parameters

- *version_number* IGMP version number. Value ranges from 1 to 3. Default value is 3.

Example

- This command configures IGMP snooping querier version 2.

```
switch(config)#ip igmp snooping querier version 2
switch(config)#
```

- This command restores the IGMP snooping querier to version 3.

```
switch(config)# no ip igmp snooping querier version
switch(config)#
```

ip igmp snooping report-flooding

The **ip igmp snooping report-flooding** command globally enables L2 report flooding on the switch. When report flooding is globally enabled, the **ip igmp snooping vlan report-flooding** configures a VLAN range to forward membership report messages to specified ports. When report flooding is not globally enabled, L2 report flooding cannot be enabled on individual VLANs.

L2 report flooding is an IGMP snooping feature that forwards membership report messages to specified ports. Relying on a single switch to maintain and send report messages can result in performance issues. L2 report flooding addresses this by facilitating report message transmissions through any network port. This allows switches to bypass the querier when forwarding multicast traffic to its interested ports.

The **no ip igmp snooping report-flooding** and **default ip igmp snooping report-flooding** commands disable global L2 report flooding by removing **ip igmp report flooding** from ***running-config***. L2 report flooding is disabled by default.

Command Mode

Global Configuration

Command Syntax

```
ip igmp snooping report-flooding
no ip igmp snooping report-flooding
default ip igmp snooping report-flooding
```

Related Commands

- **ip igmp snooping vlan report-flooding** enables L2 report flooding on a specified VLAN range.

Example

- This command globally enables the snooping L2 report-flooding.

```
switch(config)#ip igmp snooping report-flooding
switch(config)#
```

ip igmp snooping report-flooding switch-port

The **ip igmp snooping report-flooding switch-port** command specifies Ethernet ports or port channels that can forward IGMP membership report messages for all VLANs where L2 report flooding is enabled. Ports that are connected to multicast routers or queriers continue to forward traffic as previously specified and are not affected by L2 report flooding commands.

L2 report flooding is an IGMP snooping feature that forwards membership report messages to specified ports. The **ip igmp snooping vlan report-flooding switch-port** command configures a list of forwarding ports for a specified VLAN range.

The **no ip igmp snooping report-flooding switch-port** and **default ip igmp snooping report-flooding switch-port** commands remove the specified ports from the global report flooding port list by deleting the corresponding **ip igmp snooping report-flooding switch-port** command from ***running-config*.**

Command Mode

Global Configuration

Command Syntax

```
ip igmp snooping report-flooding switch-port INTERFACE
no ip igmp snooping report-flooding switch-port INTERFACE
default ip igmp snooping report-flooding switch-port INTERFACE
```

Parameters

- ***INTERFACE*** Membership report message forwarding is enabled on these ports:
  - **ethernet** *e_range* where *e_range* is the number, range, or list of ethernet ports
  - **port-channel** *p_range* where *p_range* is the number, range, or list of channel ports

Related Commands

- **ip igmp snooping report-flooding** globally enables L2 report flooding.
- **ip igmp snooping vlan report-flooding switch-port** specifies a port list for a VLAN range.

Example

- This command configures Ethernet ports 7-9 for report message forwarding for any VLAN where L2 report flooding is enabled.

```
switch(config)#ip igmp snooping report-flooding switch-port ethernet 7-9
switch(config)#
```

ip igmp snooping restart query-interval

The **ip igmp snooping restart query-interval** command sets the query interval for all VLANs during an IGMP snooping restart. By default, the query interval during an IGMP snooping restart is a VLAN's configured query interval divided by five. This accelerates the transmission of robustness queries to establish the IGMP snooping state more quickly. However, some large scale configurations may not be able to process all of the queries at this query interval rate. The restart query interval, when configured, is valid for all VLANs.

The **no ip igmp snooping resrtart query-interval** and **default ip igmp snooping restart query-interval** commands removes the global restart query interval by deleting the **ip igmp snooping restart query-interval** statement from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
ip igmp snooping restart query-interval query_sec
no ip igmp snooping restart query-interval
default ip igmp snooping restart query-interval
```

Parameters

- *query_sec* query interval (seconds). Values range from 2 to 400. Default (global) is 125.

Example

- This command sets the global query interval to 35 seconds.

```
switch(config)#ip igmp snooping restart query-interval 35
switch(config)#
```

ip igmp snooping robustness-variable

The **ip igmp snooping robustness-variable** command configures the robustness variable for snooping packets sent from any VLAN. Values range from 1 to 3 with a default of 2.

The robustness variable specifies the number of unacknowledged snooping queries that a switch sends before removing the recipient from the group list.

The **no ip igmp snooping robustness-variabl**e and **default ip igmp snooping robustness-variabl**e commands reset the robustness variable to 2 by removing the **ip igmp snooping robustness-variable** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
ip igmp snooping robustness-variable robust_value
no ip igmp snooping robustness-variable
default ip igmp snooping robustness-variable
```

Parameters

- *robust_value* robustness variable. Values range from 1 to 3. Default is 2.

Example

- This command sets the robustness-variable value to 3.

```
switch(config)#ip igmp snooping robustness-variable 3
switch(config)#
```

ip igmp snooping vlan

The **ip igmp snooping vlan** command enables snooping on the specified VLANs if snooping is globally enabled. IGMP snooping is globally enabled by default. The **ip igmp snooping** command enables snooping globally.

Note that if IGMP snooping is enabled, QoS will not apply to IGMP packets.

The **no ip igmp snooping vlan** command disables snooping on the specified VLANs.

The **default ip igmp snooping vlan** command returns the snooping setting for the specified VLANs to enabled by removing the corresponding **ip igmp snooping vlan** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
ip igmp snooping vlan v_range
no ip igmp snooping vlan v_range
default ip igmp snooping vlan v_range
```

Parameters

- *v_range* VLANs upon which snooping is enabled. Formats include a number, a number range, or a comma-delimited list of numbers and ranges. Numbers range from 1 to 4094.

Example

- This command disables snooping on VLANs 2 through 4.

```
switch(config)#no ip igmp snooping vlan 2-4
switch(config)#
```

ip igmp snooping vlan immediate-leave

The **ip igmp snooping vlan immediate-leave** command enables fast-leave processing on specified VLANs. When fast-leave processing is enabled, the removal of a VLAN interface's multicast group entry from the IGMP table is not preceded by an IGMP group-specific query to the interface. The switch removes an interface from the forwarding table when it detects an IGMP leave message on the interface. IGMP fast-leave processing is enabled on all VLANs by default.

The **no ip igmp snooping vlan immediate-leave** command disables fast-leave processing on the specified VLANs. The **default ip igmp snooping vlan immediate-leave** command restores fast-leave processing on the specified VLANs by removing the corresponding **no ip igmp snooping vlan immediate-leave** statement from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
ip igmp snooping vlan v_range immediate-leave
no ip igmp snooping vlan v_range immediate-leave
default ip igmp snooping vlan v_range immediate-leave
```

Parameters

- *v_range* VLAN IDs. Formats include a number, number range, or comma-delimited list of numbers and ranges. Numbers range from 1 to 4094.

Example

- This command enables IGMP fast-leave processing on VLAN 10.

```
switch(config)#ip igmp snooping vlan 10 immediate-leave
switch(config)#
```

ip igmp snooping vlan max-groups

The **ip igmp snooping vlan max-groups** command configures the quantity of multicast groups that the specified VLAN's forwarding table can contain. After the limit is reached, attempts to join new groups are ignored. There is no default limit.

The **no ip igmp snooping vlan max-groups** and **default ip igmp snooping vlan max-groups** removes the maximum group limit by deleting the **ip igmp snooping vlan max-groups** statement from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
ip igmp snooping vlan v_range max-groups quantity
no ip igmp snooping vlan v_range max-groups
default ip igmp snooping vlan v_range max-groups
```

Parameters

- *v_range* VLAN IDs. Formats include a number, number range, or comma-delimited list of numbers and ranges. Numbers range from 1 to 4094.
- *quantity* maximum number of groups that can access the VLAN. Value ranges from 0 to 65534.

Examples

- This command limits the number of multicast groups that hosts on VLAN 6 can simultaneously access to 25.

```
switch(config)#ip igmp snooping vlan 6 max-groups 25
switch(config)#
```

- This command allows each VLAN between 8 and 15 to receive multicast packets from 30 groups.

```
switch(config)#ip igmp snooping vlan 8-15 max-groups 30
switch(config)#
```

- This command removes the maximum group restriction from all VLAN interfaces between 1 and 50.

```
switch(config)#no ip igmp snooping vlan 1-50 max-groups
switch(config)#
```

ip igmp snooping vlan mrouter

The **ip igmp snooping vlan mrouter** command adds a multicast router as a static port to the specified VLANs. The router port must be in the specified VLAN range.

Snooping may not always be able to locate the IGMP querier. This command should specify IGMP queriers that are known to connect to the network through a port on the switch.

The **no ip igmp snooping vlan mrouter** and **default ip igmp snooping vlan mrouter** commands remove the specified static port configuration by deleting the corresponding **ip igmp snooping vlan mrouter** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
ip igmp snooping vlan v_range mrouter interface STATIC_INT
no ip igmp snooping vlan v_range mrouter interface STATIC_INT
default ip igmp snooping vlan v_range mrouter interface STATIC_INT
```

Parameters

- *v_range* VLAN IDs. Formats include a number, number range, or comma-delimited list of numbers and ranges. Numbers range from 1 to 4094.
- ***STATIC_INT*** interface the command configures as a static port. Selection options include:
  - **ethernet** *e_range* where *e_range* is the number, range, or list of ethernet ports
  - **port-channel** *p_range* where *p_range* is the number, range, or list of channel ports

  The ***STATIC_INT*** interface must route traffic through a VLAN specified within *v_range.*

Example

- This command configures the static connection to a multicast router through Ethernet port 3.

```
switch(config)#ip igmp snooping vlan 2 mrouter interface ethernet 3
switch(config)#
```

ip igmp snooping vlan querier

The **ip igmp snooping vlan querier** command controls the querier for the specified VLANs. VLANs follow the global querier setting unless overridden by one of these commands:

- **ip igmp snooping vlan querier** enables the querier on specified VLANs.
- **no ip igmp snooping vlan querier** disables the querier on specified VLANs.

VLAN querier commands take precedence over the global querier setting. The **ip igmp snooping querier** controls the querier for VLANs with no snooping querier command.

The IGMP snooping querier supports snooping by sending layer 2 membership queries to hosts attached to the switch. The snooping querier is functional on VLANs where hosts receive IP multicast traffic without access to a network IP multicast router. A snooping querier avoids flooding multicast packets in the VLAN by querying for hosts and routers.

The IGMP snooping querier is functional on VLANs that meet these criteria:

- Snooping is enabled.
- The corresponding SVI (VLAN interface) is active.
- The VLAN's querier IP address or the global querier IP address is configured.

The **default ip igmp snooping vlan querier** command restores the usage of the global setting for the specified VLAN by removing the corresponding **ip igmp snooping vlan querier** or **no ip igmp snooping vlan querier** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
ip igmp snooping vlan v_range querier
no ip igmp snooping vlan v_range querier
default ip igmp snooping vlan v_range querier
```

Parameters

- *v_range* VLAN IDs. Formats include a number, a number range, or a comma-delimited list of numbers and ranges. Numbers range from 1 to 4094.

Examples

- These commands globally enable the snooping querier on the switch, explicitly disable snooping on VLANs 1-3, and explicitly enable snooping on VLANs 4-6.

```
switch(config)#ip igmp snooping querier
switch(config)#no ip igmp snooping vlan 1-3 querier
switch(config)#ip igmp snooping vlan 4-6 querier
```

After running these commands, the running-config file contains these lines, which indicate that the snooping querier is enabled on VLANs 4-6.

```
switch(config)#show running-config
             <-------OUTPUT OMITTED FROM EXAMPLE-------->
no ip igmp snooping vlan 1 querier
no ip igmp snooping vlan 2 querier
no ip igmp snooping vlan 3 querier
ip igmp snooping vlan 4 querier
ip igmp snooping vlan 5 querier
ip igmp snooping vlan 6 querier
ip igmp snooping querier
             <-------OUTPUT OMITTED FROM EXAMPLE-------->
```

- This command removes the querier setting for VLANs 2-5:

  ```
  switch(config)#default ip igmp snooping vlan 2-5 querier
  ```

  When executed after the previous commands, the snooping querier is disabled explicitly on VLANs 1-2, enabled implicitly on VLANs 3-6, and enabled explicitly on VLANs 7-8, as shown by ***running-config***:

  ```
                <-------OUTPUT OMITTED FROM EXAMPLE-------->
  no ip igmp snooping vlan 1 querier
  ip igmp snooping vlan 6 querier
  ip igmp snooping querier
                <-------OUTPUT OMITTED FROM EXAMPLE-------->
  ```

- This command sets the global snooping querier to disabled by removing the global querier setting from ***running-config***:

  ```
  switch(config)#no ip igmp snooping querier
  switch(config)#
  ```

  When executed after the previous commands, the snooping querier is disabled explicitly on VLANs 1-2, disabled implicitly on VLANs 3-6 and enabled explicitly on VLANs 7-8, as shown by ***running-config***.

  ```
                <-------OUTPUT OMITTED FROM EXAMPLE-------->
  no ip igmp snooping vlan 1 querier
  ip igmp snooping vlan 6 querier
                <-------OUTPUT OMITTED FROM EXAMPLE-------->
  ```

ip igmp snooping vlan querier address

The **ip igmp snooping vlan querier address** command sets the source address for query packets sent from specified VLANs. VLANs not assigned an address use the global address (**ip igmp snooping querier address**). VLAN querier address commands take precedence over the global address.

To use a snooping querier, an address must be explicitly configured globally or for the querier's VLAN.

The **no ip igmp vlan snooping querier address** and **default ip igmp snooping vlan querier address** commands reset the specified VLAN to use the global address by removing the corresponding **ip igmp snooping vlan querier address** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
ip igmp snooping vlan v_range querier address ipv4_address
no ip igmp snooping vlan v_range querier address
default ip igmp snooping vlan v_range querier address
```

Parameters

- *v_range* VLAN IDs. Formats include a number, number range, or comma-delimited list of numbers and ranges. Numbers range from 1 to 4094.
- *ipv4_address* source IPv4 address.

Example

- This command sets the source IPv4 address of 10.14.1.1. for query packets transmitted from VLAN 2.

```
switch(config)#ip igmp snooping vlan 2 querier address 10.14.1.1
switch(config)#
```

ip igmp snooping vlan querier last-member-query-count

The **ip igmp snooping vlan querier last-member-query-count** command specifies an ***IGMP snooping querier last member query count*** (LMQC) value for the specified VLANs. LMQC specifies the number of query messages the switch sends in response to group-specific or group-source-specific leave messages it receives from a host; the transmission frequency is specified by ***IGMP snooping querier last member query interval***. The switch stops forwarding messages to the host if it does not receive a response to these query messages.

VLANs not assigned an LMQC value use the global value (**ip igmp snooping querier last-member-query-count**). VLAN commands take precedence over the global command.

Setting the last member query count (LMQC) to 1 causes the loss of a single packet to stop traffic forwarding. While the switch can start forwarding traffic again after receiving a response to the next general query, the host may not receive that query for a period defined by **ip igmp snooping querier query-intervall**.

The **no igmp snooping vlan querier last-member-query-count** and **default igmp snooping vlan querier last-member-query-count** commands reset the specified VLAN to use the global LMQC by removing the corresponding **ip igmp snooping vlan querier last-member-query-count** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
ip igmp snooping vlan v_range querier last-member-query-count number
no ip igmp snooping vlan v_range querier last-member-query-count
default ip igmp snooping vlan v_range querier last-member-query-count
```

Parameters

- *v_range* VLAN IDs. Formats include a number, number range, or comma-delimited list of numbers and ranges. Numbers range from 1 to 4094.
- *number* query message quantity. Value ranges from 1 to 3.

Example

- This command configures the last-member-query-count to 1 on VLAN interface 3.

```
switch(config)#ip igmp snooping vlan 3 querier last-member-query-count 1
switch(config)#
```

ip igmp snooping vlan querier last-member-query-interval

The **ip igmp snooping vlan querier last-member-query-interval** command configures ***last-member-query-interval*** for packets sent from the specified VLANs. VLANs not assigned a value use the global setting (**ip igmp snooping querier last-member-query-interval**). VLAN commands take precedence over the global value. The global default is one second.

A multicast host sends an IGMP leave report when it leaves a group. To determine if the host was the last group member, the leave message recipient sends an IGMP query. The last-member-query-interval determines when the group record is deleted if no subsequent reports are received.

The **no ip igmp snooping vlan querier last-member-query-interval** and **default ip igmp snooping vlan querier last-member-query-interval** commands reset the specified VLAN to use the global ***last-member-query-interval*** by removing the corresponding **ip igmp snooping vlan querier last-member-query-interval** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
ip igmp snooping vlan v_range querier last-member-query-interval period
no ip igmp snooping vlan v_range querier last-member-query-interval
default ip igmp snooping vlan v_range querier last-member-query-interval
```

Parameters

- *v_range* VLAN IDs. Formats include a number, number range, or comma-delimited list of numbers and ranges. Numbers range from 1 to 4094.
- *period* last member query interval (seconds). Value ranges from 1 to 25.

Example

- This command sets the last-member-query-interval for VLAN 10 to 12 seconds.

```
switch(config)#ip igmp snooping vlan 10 querier last-member-query-interval 12
switch(config)#
```

ip igmp snooping vlan querier max-response-time

The **ip igmp snooping vlan querier max-response-time** command configures ***max-response-time*** for packets sent from the specified VLANs. VLANs not assigned a value use the global setting (**ip igmp snooping querier max-response-time**). VLAN commands take precedence over the global value. The global default is 10 seconds.

Switches use ***max-response-time*** to set the Max Response Time field in outbound Membership Query messages. Max Response Time specifies the maximum period a recipient can wait before responding with a Membership Report.

The **no ip igmp snooping vlan querier max-response-time** and **default ip igmp snooping vlan querier max-response-time** commands reset the specified VLAN to use the global ***max-response-time*** by removing the corresponding **ip igmp snooping vlan querier max-response-time** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
ip igmp snooping vlan v_range querier max-response-time resp_sec
no ip igmp snooping vlan v_range querier max-response-time
default ip igmp snooping vlan v_range querier max-response-time
```

Parameters

- *v_range* VLAN ID. Formats include a number, number range, or comma-delimited list of numbers and ranges. Numbers range from 1 to 4094.
- *resp_sec* ***max-response-time*** value (seconds). Values range from 1 to 25.

Example

- This command sets the max-response-time for VLAN 2 to 5 seconds.

```
switch(config)#ip igmp snooping vlan 2 querier max-response-time 5
switch(config)#
```

ip igmp snooping vlan querier query-interval

The **ip igmp snooping vlan querier query-interval** command sets the query interval for the specified VLAN. VLANs not assigned a value use the global value (**ip igmp snooping querier query-interval**). VLAN commands have precedence over the global value. The query interval is the period between IGMP Membership Query messages sent from the querier.

The **no ip igmp snooping vlan querier query-interval** and **default ip igmp snooping vlan querier query-interval** commands reset the specified VLAN to use the global value by removing the corresponding **ip igmp snooping vlan querier query-interval** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
ip igmp snooping vlan v_range querier query-interval query_sec
no ip igmp snooping vlan v_range querier query-interval
default ip igmp snooping vlan v_range querier query-interval
```

Parameters

- *v_range* VLAN IDs. Formats include a number, number range, or comma-delimited list of numbers and ranges. Numbers range from 1 to 4094.
- *query_sec* query interval (seconds). Values range from 5 to 3600. Default (global) is 125.

Example

- This command sets the query interval for VLAN 10 to 240 seconds.

```
switch(config)#ip igmp snooping vlan 10 querier query-interval 240
switch(config)#
```

ip igmp snooping vlan querier startup-query-count

The **ip igmp snooping vlan querier startup-query-count** command specifies the startup query count value for the specified VLANs. The ***startup query count*** specifies the number of query messages that the querier sends on a VLAN during the ***startup query interval*** (**ip igmp snooping vlan querier startup-query-interval**).

When an interface starts running IGMP, it can establish the group state more quickly by sending query messages at a higher frequency. The ***startup-query-interval*** and ***startup-query-count*** parameters define the startup period and the query message transmission frequency during that period.

VLANs not assigned a ***startup query count*** value use the global value (**ip igmp snooping querier startup-query-count**). VLAN commands take precedence over the global command.

The **no ip igmp snooping vlan querier startup-query-count** and **default ip igmp snooping vlan querier startup-query-count** commands restore the default condition of using the global ***startup query count*** value by removing the corresponding **ip igmp snooping vlan querier startup-query-count** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
ip igmp snooping vlan v_range querier startup-query-count number
no ip igmp snooping vlan v_range querier startup-query-count
default ip igmp snooping vlan v_range querier startup-query-count
```

Parameters

- *v_range* VLAN IDs. Formats include a number, number range, or comma-delimited list of numbers and ranges. Numbers range from 1 to 4094.
- *number* startup query count. Value ranges from 1 to 3.

Example

- This command configures the startup query count of 3 for VLAN 100.

```
switch(config)#ip igmp snooping vlan 100 querier startup-query-count 3
switch(config)#
```

ip igmp snooping vlan querier startup-query-interval

The **ip igmp snooping vlan querier startup-query-interval** command specifies the ***startup query interval*** value for the specified VLANs. The ***startup query interval*** specifies the period between query messages that the querier sends upon startup.

When snooping is enabled, the group state is more quickly established by sending query messages at a higher frequency. The ***startup-query-interval*** and ***startup-query-count*** parameters define the startup period by defining the number of queries to be sent and transmission frequency for these messages.

VLANs not assigned a ***startup query interval*** value use the global value (**ip igmp snooping querier startup-query-count**). VLAN commands take precedence over the global command.

The **no ip igmp snooping vlan querier startup-query-interval** and **default ip igmp snooping vlan querier startup-query-interval** commands restore the default condition of using the global ***startup query interval*** value by removing the corresponding **ip igmp snooping vlan querier startup-query-interval** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
ip igmp snooping vlan v_range querier startup-query-interval period
no ip igmp snooping vlan v_range querier startup-query-interval
default ip igmp snooping vlan v_range querier startup-query-interval
```

Parameters

- *v_range* VLAN IDs. Formats include a number, number range, or comma-delimited list of numbers and ranges. Numbers range from 1 to 4094.
- *period* startup query interval (seconds). Value ranges from 1 to 3600 (1 hour). Default is 31.

Example

- This command configures the startup query count of one minute for VLAN interface 100.

```
switch(config)#ip igmp snooping vlan 100 querier startup-query-interval 60
switch(config)#
```

ip igmp snooping vlan querier version

The **ip igmp snooping vlan querier version** command configures the Internet Group Management Protocol (IGMP) snooping querier function on the VLAN. Version 3 is the default IGMP snooping version.

IGMP is enabled by the **ip pim sparse-mode** command. The **ig igmp snooping vlan querier version** command does not affect the IGMP enabled status.

The **no ip igmp snooping vlan querier version** and **default ip igmp snooping vlan querier version** commands restore the configuration mode interface to IGMP snooping VLAN querier version 3 by removing the **ip igmp snooping vlan querier version** statement from ***running-config.***

Command Mode

Global Configuration

Command Syntax

```
ip igmp snooping vlan v_range querier version version_number
no ip igmp snooping vlan v_range querier version
default ip igmp snooping vlan v_range querier version
```

Parameters

- *v_range* VLAN ID. Formats include a number, number range, or comma-delimited list of numbers and ranges. Numbers range from 1 to 4094.
- *version_number* IGMP version number. Value ranges from 1 to 3. Default value is 3.

Example

- The example sets the querier version to 2 on vlan 5.

```
switch(config)#ip igmp snooping vlan 5 querier version 2
switch(config)#
```

- This command restores IGMP snooping querier version 3 to VLAN 5.

```
switch(config)# no ip igmp snooping vlan 5 querier version
switch(config)#
```

ip igmp snooping vlan report-flooding

The **ip igmp snooping vlan immediate-leave** command enables L2 report flooding on the specified VLANs if report flooding is globally enabled. When L2 report flooding is not globally enabled, this command has no effect. The **ip igmp snooping report-flooding** command globally enables L2 report flooding.

L2 report flooding is an IGMP snooping feature that forwards membership report messages to specified ports. Relying on a single switch to maintain and send report messages can degrade performance. L2 report flooding addresses this by facilitating report message forwarding through any network port. This allows switches to bypass the querier when forwarding multicast traffic to its interested ports.

Two commands specify the ports that forward reports:

- **ip igmp snooping vlan report-flooding switch-port** for a VLAN range.
- **ip igmp snooping report-flooding switch-port** for all VLANs where report flooding is enabled.

The **no ip igmp snooping vlan immediate-leave** and **default ip igmp snooping vlan immediate-leave** commands disable L2 report flooding for the specified VLAN by removing the corresponding **ip igmp snooping vlan immediate-leave** statement from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
ip igmp snooping vlan v_range report-flooding
no ip igmp snooping vlan v_range report-flooding
default ip igmp snooping vlan v_range report-flooding
```

Parameters

- *v_range* VLAN IDs. Formats include a number, number range, or comma-delimited list of numbers and ranges. Numbers range from 1 to 4094.

Related Commands

- **ip igmp snooping report-flooding** globally enables L2 report flooding.

Example

- These commands enable L2 report flooding globally and on VLANs 201 through 205.

```
switch(config)#ip igmp snooping report-flooding
switch(config)#ip igmp snooping vlan 201-205 report-flooding
switch(config)#
```

ip igmp snooping vlan report-flooding switch-port

The **ip igmp snooping vlan report-flooding switch-port** command configures Ethernet ports or port channels to forward IGMP membership report messages for a specified VLAN range where L2 report flooding is enabled. Ports that are connected to multicast routers or queriers continue to forward traffic as previously specified and are not affected by L2 report flooding commands.

L2 report flooding is an IGMP snooping feature that forwards membership report messages to specified ports. The **ip igmp snooping report-flooding switch-port** command configures a list of forwarding ports for all VLANs where L2 report flooding is enabled.

The **no ip igmp snooping vlan report-flooding switch-port** and **default ip igmp snooping vlan report-flooding switch-port** commands remove the listed ports from the specified report flooding port list by deleting the corresponding **ip igmp snooping vlan report-flooding switch-port** statements from ***running-config*.**

Command Mode

Global Configuration

Command Syntax

```
ip igmp snooping vlan v_range report-flooding switch-port INTERFACE
no ip igmp snooping vlan v_range report-flooding switch-port INTERFACE
default ip igmp snooping vlan v_range report-flooding switch-port INTERFACE
```

Parameters

- *v_range* VLAN IDs. Formats include a number, number range, or comma-delimited list of numbers and ranges. Numbers range from 1 to 4094.
- ***INTERFACE*** Membership report message forwarding is enabled on these ports:
  - **ethernet** *e_range* where *e_range* is the number, range, or list of ethernet ports
  - **port-channel** *p_range* where *p_range* is the number, range, or list of channel ports

Related Commands

- **ip igmp snooping report-flooding** globally enables L2 report flooding.
- **ip igmp snooping vlan report-flooding switch-port** specifies a port list for a VLAN range.
- **ip igmp snooping report-flooding switch-port** specifies a port list for all VLANs.

Example

- These commands globally enable L2 report flooding, enable flooding on VLANs 201 through 205, and specify Ethernet ports 8-10 as the report flooding port list for VLANS 201-205.

```
switch(config)#ip igmp snooping report-flooding
switch(config)#ip igmp snooping vlan 201-205 report-flooding
switch(config)#ip igmp snooping vlan 201-205 report-flooding switch-port ethernet
8-10
switch(config)#
```

ip igmp snooping vlan static

The **ip igmp snooping static** command adds ports as static members to a multicast group. The ports must be in the specified VLAN range.

The **no ip igmp snooping static** and **default ip igmp snooping static** commands remove the specified ports from the multicast group by deleting the corresponding ip igmp snooping static statements from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
ip igmp snooping vlan v_num static ipv4_addr interface STATIC_INT
no ip igmp snooping vlan v_num static ipv4_addr interface STATIC_INT
default ip igmp snooping vlan v_num static ipv4_addr interface STATIC_INT
```

Parameters

- *v_num* VLAN number. Value ranges from 1 to 4094.
- *ipv4_addr* multicast group IPv4 address.
- ***STATIC_INT*** interface the command configures as the static group member. Options include:
  - **ethernet** *e_range*, where *e_range* is the number, range, or list of Ethernet ports
  - **port-channel** *p_range*, where *p_range* is the number, range, or list of channel ports

Example

- This command configures the static connection to a multicast group at 237.2.1.4 through Ethernet port 3.

```
switch(config)#ip igmp snooping vlan 7 static 237.2.1.4 interface ethernet 3
switch(config)#
```

ip igmp startup-query-count

The **ip igmp startup-query-count** command specifies the number of query messages that an interface sends during the startup interval defined by **ip igmp startup-query-interval**.

When an interface starts running IGMP, it can establish the group state more quickly by sending query messages at a higher frequency. The ***startup-query-interval*** and ***startup-query-count*** parameters define the startup period and the query message transmission frequency during that period.

The **no ip igmp startup-query-count** and **default ip igmp startup-query-count** commands restore the default ***startup-query-count*** value of 2 for the configuration mode interface by removing the corresponding **ip igmp startup-query-count** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration
Interface-Port-Channel Configuration
Interface-VLAN Configuration

Command Syntax

```
ip igmp startup-query-count number
no ip igmp startup-query-count
default ip igmp startup-query-count
```

Parameters

- *number* quantity of queries. Values range from 1 to 65535. Default is 2.

Example

- This command configures the startup query count of 10 for VLAN interface 4.

```
switch(config)#interface vlan 4
switch(config-if-Vl4)#ip igmp startup-query-count 10
switch(config-if-Vl4)#
```

ip igmp startup-query-interval

The **ip igmp startup-query-interval** command specifies the configuration mode interface's IGMP startup period, during which query messages are sent at an accelerated rate.

When an interface starts running IGMP, it can establish the group state quicker by sending query messages at a higher frequency. The ***startup-query-interval*** and ***startup-query-count*** parameters define the startup period and the query message transmission frequency during that period.

The **no ip igmp startup-query-interval** and **default ip igmp startup-query-interval** commands restore the configuration mode interface's default IGMP ***startup-query-interval*** of 31 seconds by removing the corresponding **ip igmp startup-query-interval** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration
Interface-Port-Channel Configuration
Interface-VLAN Configuration

Command Syntax

```
ip igmp startup-query-interval period
no ip igmp startup-query-interval
default ip igmp startup-query-interval
```

Parameters

- *period* startup query interval, in deciseconds. Value ranges from 10 (one second) to 317440 (8 hours, 49 minutes, 4 seconds). Default is 31 seconds.

Example

- This command configures the startup query count of one minute for VLAN interface 4.

```
switch(config)#interface vlan 4
switch(config-if-Vl4)#ip igmp startup-query-interval 600
switch(config-if-Vl4)#
```

ip igmp static-group

The **ip igmp static-group** command configures the configuration mode interface as a static member of a specified multicast group. This allows the router to forward multicast group packets through the interface without otherwise appearing or acting as a group member. By default, static group memberships are not configured on any interfaces.

If the command includes a source address, only multicast group messages received from the specified host address are fast-switched. Otherwise, all multicast messages of the specified group are fast-switched.

Note To become a static member of a multicast group, the switch must be the PIM designated router (DR) for the network. If it is not, you can use the **ip pim dr-priority** command to make it the DR by configuring its PIM DR value to be the highest on the network.

The **no ip igmp static-group** and **default ip igmp static-group** commands remove the configuration mode interface's group membership by removing the corresponding **ip igmp static-group** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration
Interface-Port-Channel Configuration
Interface-VLAN Configuration

Command Syntax

```
ip igmp static-group group_address [SOURCE_ADDRESS]
no ip igmp static-group group_address [SOURCE_ADDRESS]
default ip igmp static-group group_address [SOURCE_ADDRESS]
```

Parameters

- *group_address* IPv4 address of multicast group for which the interface fast-switches packets.
- ***SOURCE_ADDRESS*** IP address of host that originates multicast data packets.
  - <no parameter> all multicast messages of the specified group are fast-switched.
  - *ipv4_address* source IP address (dotted decimal notation).

Related Commands

- **ip igmp static-group acl** configures the configuration mode interface as a static member of the multicast groups specified by an IP access control list (ACL).
- **ip igmp static-group range** configures the configuration mode interface as a static member of multicast groups specified by an address range.

  One **ip igmp static-group range** command is equivalent to multiple **ip igmp static-group** commands.

Example

- These commands configure VLAN interface 15 as the PIM designated router, then configure it as a static member of the multicast group at address 231.1.1.15 for multicast data packets that originate at 10.1.1.1.

```
switch(config)#interface vlan 15
switch(config-if-Vl15)#ip pim dr-priority 5000
switch(config-if-Vl15)#ip igmp static-group 231.1.1.45 10.1.1.1
switch(config-if-Vl15)#
```

ip igmp static-group acl

The **ip igmp static-group acl** command configures the configuration mode interface as a static member of the multicast groups specified by an IP access control list (ACL). This command is a variant of the **ip igmp static-group** command that uses ACL rules to specify a set of source-multicast group address pairs instead of specifying a single pair. Multiple static-group ACLs can be assigned to an interface. Static groups can be assigned manually and through ACLs simultaneously.

Access control lists that this command references must contain rules of the following format.

- **permit <protocol><source><destination>**, where
  - **<protocol>** has no effect on the static group.
  - **<source>** address of host originating multicast data packets. Must be a host address.
  - **<destination>** multicast group IP address or subnet. Must be a valid multicast address.

An ACL can contain multiple rules. An ACL can be applied to an interface only when all of its rules comply to the specified restrictions. The **show ip igmp static-groups acl** displays the source-multicast group pairs that the specified list configures and lists issues with illegal rules.

The **no ip igmp static-group acl** and **default ip igmp static-group acl** commands remove the specified static group ACL command from ***running-config***.

Command Mode

Interface-Ethernet Configuration
Interface-Port-Channel Configuration
Interface-VLAN Configuration

Command Syntax

```
ip igmp static-group acl list_name
no ip igmp static-group acl list_name
default ip igmp static-group acl list_name
```

Parameters

- *list_name* ACL that specifies multicast group addresses for which interface fast-switches packets.

Example

- This command configures VLAN interface 4 as a static member of the multicast group specified by the ACL named ***LIST_1***.

```
switch(config)#interface vlan 4
switch(config-if-Vl4)#ip igmp static-group acl LIST_1
switch(config-if-Vl4)#
```

ip igmp static-group range

The **ip igmp static-group range** command configures the configuration mode interface as a static member of multicast groups specified by an address range. This allows the router to forward multicast group packets through the interface without otherwise appearing or acting as a group member. By default, no static group memberships are configured on interfaces.

This command is a variant of the **ip igmp static-group** command that allows the assignment of a subnet range of source addresses or a subnet range of multicast groups. A single **ip igmp static-group range** command is the equivalent of multiple **ip igmp static-group** commands, each of which can only assign a single multigroup-source pair to an interface. Running-config converts the range command to the equivalent list of **ip igmp static-group** commands.

If the command includes a source address range, only multicast group messages received from the range are fast-switched. Otherwise, all multicast messages of the specified group are fast-switched.

The **no ip igmp static-group range** and **default ip igmp static-group range** commands remove the specified range of static group statements from ***running-config***. The **no ip igmp static-group** and **default ip igmp static-group** commands can remove an individual static-group command that was initially added to ***running-config*** by an **ip igmp static-group range** command.

Command Mode

Interface-Ethernet Configuration
Interface-Port-Channel Configuration
Interface-VLAN Configuration

Command Syntax

```
ip igmp static-group range GROUP_ADDR [SOURCE_ADDR]
no ip igmp static-group range GROUP_ADDR [SOURCE_ADDR]
default ip igmp static-group range GROUP_ADDR [SOURCE_ADDR]
```

Parameters

- ***GROUP_ADDR*** address of multicast group for which the interface fast-switches packets.
  - *gp_ipv4_addr* multicast group IPv4 address.
  - *gp_ipv4_subnet* IPv4 subnet address of multicast groups (CIDR or address-mask).
- ***SOURCE_ADDR*** IP address of a host range that originates multicast data packets.
  - <no parameter> all multicast messages of the specified range are fast-switched.
  - **source** *sr_ipv4_address* source IPv4 address (dotted decimal notation).
  - **source** *sr_ ipv4_subnet* IPv4 subnet address of source hosts (CIDR or address- mask).

Warning A command cannot specify a subnet address for both multicast group and source.

Examples

- This command configures VLAN interface 4 as a static member of the multicast group range 241.1.4.1/24 for data packets that originate at 10.1.1.1.

```
switch(config)#interface vlan 4
switch(config-if-Vl4)#ip igmp static-group range 239.1.4.1/24 source 10.1.1.1
switch(config-if-Vl4)#
```

- This command attempts to configure VLAN interface 4 as a static member of the multicast group range 241.1.4.1/24 for data packets that originate at the 10.1.1.1/29 subnet. Because the range and source cannot both be subnets, this command generates an error message.

```
switch(config-if-Vl4)#ip igmp static-group range 239.1.1.1/29 source 16.1.1.1/29
% Error: cannot specify source range with group range
switch(config-if-Vl4)#
```

ip igmp version

The **ip igmp version** command configures the Internet Group Management Protocol (IGMP) version on the configuration mode interface. Version 3 is the default IGMP version.

IGMP is enabled by the **ip pim sparse-mode** command. The ig igmp version command does not affect the IGMP enabled status.

The **no ip igmp version** and **default ip igmp version** commands restore the configuration mode interface to IGMP version 3 by removing the **ip igmp version** statement from ***running-config.***

Command Mode

Interface-Ethernet Configuration
Interface-Port-Channel Configuration
Interface-VLAN Configuration

Command Syntax

```
ip igmp version version_number
no ip igmp version
default ip igmp version
```

Parameters

- *version_number* IGMP version number. Value ranges from 1 to 3.

Example

- This command configures IGMP version 3 on VLAN interface 4.

```
switch(config)#interface vlan 4
switch(config-if-Vl4)#ip igmp version 3
switch(config-if-Vl4)#
```

permit / deny

The **permit** command configures the configuration mode IGMP profile as a permit list. Applying a permit list to an interface restricts that interface from joining any multicast group not included in the list.

IGMP profiles are deny lists by default. When applied to an interface, a deny list allows the interface to join any multicast group that is not included in the list.

The **deny** command restores the IGMP list to its default type by removing the corresponding **permit** statement from ***running-config***.

The **range** command adds and removes address ranges from the configuration mode profile.

Command Mode

IGMP-profile Configuration

Command Syntax

```
permit
deny
```

Related Commands

- **ip igmp profile** places the switch in IGMP-profile configuration mode.

Example

- These commands enter IGMP profile configuration mode and configure the profile as a permit list.

```
switch(config)#ip igmp profile list_1
switch(config-igmp-profile-list_1)#permit
switch(config-igmp-profile-list_1)#
```

range

The **range** command specifies an address range for the configuration mode IGMP profile. A permit range specifies the groups that an interface is permitted to join. A deny range specifies the groups that an interface is not permitted to join. The **permit / deny** command specifies the range type.

A profile may contain multiple range statements to define a discontiguous address range.

The **no range** and **default range** commands remove the specified address range from a previous specified list.

Command Mode

IGMP-profile Configuration

Command Syntax

```
range init_address [UPPER_RANGE]
no range init_address [UPPER_RANGE]
default range init_address [UPPER_RANGE]
```

Parameters

- *init_address* IP address of lower boundary of the address range (dotted decimal notation).
- ***UPPER_RANGE*** sets the upper boundary of the address range. Options include
  - <no parameter> upper boundary is equal to lower boundary: range consists of one address.
  - *range_address* IP address of upper boundary.

  All addresses must be multicast addresses (10.0.0.0 to 239.255.255.255).

Related Commands

- **ip igmp profile** places the switch in IGMP-profile configuration mode.

Example

- These commands enter IGMP profile configuration mode, configure the profile as a permit list, and define the permit address list of 232.1.1.0 to 232.1.1.255 and 233.1.1.10.

```
switch(config)#ip igmp profile list_1
switch(config-igmp-profile-list_1)#permit
switch(config-igmp-profile-list_1)#232.1.1.0 232.1.1.255
switch(config-igmp-profile-list_1)#233.1.1.10
switch(config-igmp-profile-list_1)#
```

show ip igmp groups

The **show ip igmp groups** command displays multicast groups that have receivers directly connected to the switch, as learned through Internet Group Management Protocol (IGMP).

- **show ip igmp groups** all multicast groups.
- **show ip igmp groups** *group_addr* listed multicast group.
- **show ip igmp groups interface** *int_name* all multicast groups on specified interfaces
- **show ip igmp groups** *group_addr* **interface** *int_name* listed multicast group on specified interface.

Command Mode

EXEC

Command Syntax

show ip igmp groups *GROUP_LIST* [*DATA*]

Parameters

- ***GROUP_LIST*** list of groups for which the command displays information. Options include:
  - <no parameter> all multicast groups.
  - *group_addr* single multicast group address (dotted decimal notation).
  - **interface ethernet** *e_num* all multicast groups on specified Ethernet interface.
  - **interface loopback** *l_num* all multicast groups on specified Loopback interface.
  - **interface management** *m_num* all multicast groups on specified Management interface.
  - **interface port-channel** *p_num* all multicast groups on specified Port-Channel Interface.
  - **interface vlan** *v_num* all multicast groups on specified VLAN interface.
  - **interface vxlan** *vx_num* all multicast groups on specified VXLAN interface.
- ***DATA*** specifies the type of information displayed. Options include:
  - <no parameter> provides uptime, expiration, and address of reporter.
  - **detail** also include group mode and group source list.

Example

- This command displays multicast groups with receivers directly connected to the switch.

```
switch>show ip igmp groups
NOTE: static-group information not shown below.  Use the
      'show ip igmp static-groups' command.
IGMP Connected Group Membership
Group Address    Interface                Uptime    Expires   Last Reporter
10.12.1.1       Vlan162                 11d01h    00:02:57  172.17.2.110
10.12.1.2       Vlan162                 11d01h    00:02:57  172.17.2.110
10.12.1.3       Vlan162                 11d01h    00:02:57  172.17.2.110
10.12.1.4       Vlan162                 11d01h    00:02:57  172.17.2.110
10.12.1.5       Vlan162                 11d01h    00:02:57  172.17.2.110
switch>
```

show ip igmp groups count

The **show ip igmp groups count** command displays the number of multicast groups that are joined across the specified interfaces.

Command Mode

EXEC

Command Syntax

show ip igmp groups [*GROUP_LIST*] count

Parameters

- ***INTERF*** Specifies the interface for which the command displays information. Options include:
  - <no parameter> all IGMP interfaces.
  - **interface ethernet** *e_num* Ethernet interface.
  - **interface loopback** *l_num* Loopback interface.
  - **interface management** *m_num* Management interface.
  - **interface port-channel** *p_num* Port-Channel Interface.
  - **interface vlan** *v_num* VLAN interface.
  - **interface vxlan** *vx_num* VXLAN interface.

Example

- This command displays the number of multicast groups joined across all interfaces.

```
switch>show ip igmp groups count
Number of total groups joined across all IGMP interfaces: 5
switch>
```

- This command displays the number of multicast groups joined on Ethernet 3/4 interface.

```
switch>show ip igmp groups interface ethernet 3/4 count
Number of groups joined on Ethernet3/4: 2
switch>
```

show ip igmp host-proxy config-sanity

The **show ip igmp host-proxy config-sanity** command displays diagnostic information about an IGMP host proxy configuration.

Command Mode

EXEC

Command Syntax

show ip igmp host-proxy config-sanity

Example

- This command displays IGMP host proxy configuration diagnostic information.

```
switch>show ip igmp host-proxy config-sanity
DISCLAIMER:
Below are only hints of potential IGMP Host-Proxy misconfiguration.
They do not necessary imply that there is a real problem.

No IGMP Host-Proxy misconfiguration hints found
switch>
```

show ip igmp host-proxy interface

The **show ip igmp host-proxy interface** command displays IGMP host proxy configuration information. Command filters allow the list to provide data for a specified interface.

Command Mode

EXEC

Command Syntax

show ip igmp host-proxy interface *GROUP_LIST* [*DATA*]

Parameters

- ***GROUP_LIST*** Filters data to include only a specified list of groups for which the command displays information:
  - <no parameter> reports host proxy configuration status on all interfaces.
  - **ethernet** *e_num* reports status on specified Ethernet interface.
  - **port-channel** *p_num* reports status on specified Port-Channel Interface.
  - **vlan** *v_num* reports status on specified VLAN interface.
- ***DATA*** specifies the type of information displayed. Options include
  - <no parameter> indicates interfaces where host proxy is configured.
  - **detail** data includes multicast group and source addresses.

Example

- This command displays host proxy information for all switch interfaces.

```
switch(config-if-Po100)#show active
 interface Port-Channel100
   ip igmp host-proxy 234.4.4.4 exclude 0.0.0.0
   ip igmp host-proxy 234.10.4.4 include 10.14.3.3
   ip igmp host-proxy 234.10.4.4 include 10.14.3.2
   ip igmp host-proxy 234.10.4.4
   ip igmp host-proxy report-interval 4
switch(config-if-Po100)#show ip igmp host-proxy interface
-------------------------------------------------
Interface: Port-Channel100
IGMP Host-Proxy configured.
-------------------------------------------------
switch(config-if-Po100)#
```

- This command displays host proxy information for all switch interfaces, including multicast groups and source interfaces for which the interface is proxying.

```
switch>show ip igmp host-proxy interface detail
-------------------------------------------------
Interface: Port-Channel100
IGMP Host-Proxy configured.

Group: 234.4.4.4
Exclude source list: 0.0.0.0

Group: 234.10.4.4
Include source list: 10.14.3.3  10.14.3.2
-------------------------------------------------
switch>
```

show ip igmp interface

The **show ip igmp interface** command displays multicast information about the specified interface.

Command Mode

EXEC

Command Syntax

show ip igmp interface [*INT_NAME*]

Parameters

- ***INT_NAME*** Interface type and number. Values include
  - **<no parameter>** Displays information for all interfaces.
  - **ethernet** *e_num* Ethernet interface specified by *e_num*.
  - **loopback** *l_num* Loopback interface specified by *l_num*.
  - **management** *m_num* Management interface specified by *m_num*.
  - **port-channel** *p_num* Port-Channel Interface specified by *p_num*.
  - **vlan** *v_num* VLAN interface specified by *v_num*.
  - **vxlan** *vx_num* VXLAN interface specified by *vx_num*.

Example

- This command displays multicast related information about VLAN 26.

```
switch>show ip igmp interface vlan 26
Vlan26 is up
  Interface address: 172.17.26.1/23
  IGMP on this interface: enabled
  Multicast routing on this interface: enabled
  Multicast TTL threshold: 1
  Current IGMP router version: 2
  IGMP query interval: 125 seconds
  IGMP max query response time: 100 deciseconds
  Last member query response interval: 10 deciseconds
  Last member query response count: 2
  IGMP querier: 172.17.26.1
  Robustness: 2
  Require router alert: enabled
  Startup query interval: 312 deciseconds
  Startup query count: 2
  General query timer expiry: 00:00:22
  Multicast groups joined:
    239.255.255.250

switch>
```

show ip igmp profile

The **show ip igmp profile** command displays the contents of the specified IGMP profile. IGMP snooping filters use an IGMP profile to control the multicast groups that an interface can join.

Command Mode

EXEC

Command Syntax

show ip igmp snooping [*PROFILES*]

Parameters

- ***PROFILES*** IGMP profiles for which command displays contents. Options include:
  - <no parameter> displays all IGMP profiles.
  - *profile_name* displays specified profile.

Example

- This command displays the IGMP profiles configured on the switch.

```
switch>show ip igmp profile
IGMP Profile list_1
     permit
     range 229.1.24.0 229.1.25.255
IGMP Profile list_2
     range 234.1.1.0 234.1.255.255
switch>
```

show ip igmp snooping

The **show ip igmp snooping** command displays the switch's IGMP snooping configuration.

Command Mode

EXEC

Command Syntax

show ip igmp snooping [*VLAN_ID*]

Parameters

- ***VLAN_ID*** specifies VLANs for which command displays information. Options include:
  - <no parameter> displays information for all VLANs.
  - **vlan** *v_num* displays information for specified VLAN.

Example

- This command displays the switch's IGMP snooping configuration.

```
switch>show ip igmp snooping
   Global IGMP Snooping configuration:
-----------------------------------------
IGMP snooping                  : Enabled
Robustness variable            : 2

Vlan 1 :
----------
IGMP snooping                  : Enabled
Multicast router learning mode : pim-dvmrp

Vlan 20 :
----------
IGMP snooping                  : Enabled
Multicast router learning mode : pim-dvmrp

Vlan 26 :
----------
IGMP snooping                  : Enabled
Multicast router learning mode : pim-dvmrp

Vlan 2028 :
----------
IGMP snooping                  : Enabled
Multicast router learning mode : pim-dvmrp

switch>
```

show ip igmp snooping counters

The **show ip igmp snooping counters** command displays the number of IGMP messages sent and received through each switch port. The display table sorts the messages by type.

Command Mode

EXEC

Command Syntax

show ip igmp snooping counters [*DATA_TYPE*][*DATA_LEVEL*]

Parameters

- ***DATA_TYPE*** Information displayed by the command. Options include:
  - <no parameter> displays transmission counters.
  - **errors** displays error counters.
- ***DATA_LEVEL*** specifies the type of information displayed. Options include:
  - <no parameter> number of packets on physical ports.
  - **detail** number of packets on physical ports.

Example

- This command displays the number of messages received on each port.

```
switch>show ip igmp snooping counters
                        Input                    |            Output
Port    Queries Reports  Leaves  Others  Errors|Queries Reports  Leaves  Others
-------------------------------------------------------------------------------
Cpu       15249  106599       4  269502       0   30242  102812     972    3625
Et1           0       0       0       0       0       0       0       0       0
Et2           0       6       1      26       0    5415       0       0     731
Et3           0   10905     222    1037       0   15246       0       0    1448
Et4           0   44475      21     288       0   15247       0       0    2199
Et5           0     355       0      39       0   15211       0       0    2446
Et6           0     475      13       0       0   15247       0       0    2487
Et7           0       0       0     151       0   15247       0       0    2336
Et8           0     578       6      75       0    2859       0       0     931
Et9           0       0       0      27       0   15247       0       0    2460
Et10          0   12523     345      54       0   15247       0       0    2433
Et11          0       0       0       0       0       0       0       0       0
Et12          0    4509      41      22       0   15247       0       0    2465
Et13          0     392      29     119       0   15247       0       0    2368
Et14          0      88       3       6       0   15247       0       0    2481
Et15          0   16779     556      72       0   15117       0       0      66
Et16          0    2484      13      66       0   15247       0       0    2421
Et17          0       0       0       0       0       0       0       0       0
Et18          0      20       6     160       0    3688       0       0     803
Et19          0    4110      17       0       0   15247       0       0    2487
Et20          0       0       0       0       0       0       0       0       0
Et21          0       0       0       0       0       0       0       0       0
Et22          0       0       0      52       0   15247       0       0    2435
Et23          0    5439     181     138       0   15247       0       0    2349
Et24          0    2251      21       4       0   15247       0       0    2483
Po1       45360  540670    8853  464900       0   15249  224751     618    2576
Po2           0  101399      58      17       0   15120       0       0    1121
Switch        0       0       0       0       0       0       0       0       0
```

show ip igmp snooping counters ethdev-pams

The **show ip igmp snooping counters** command displays the number of dropped IGMP packets messages sent and received through each switch port at the kernel level. The display table sorts the messages by type.

Command Mode

EXEC

Command Syntax

show ip igmp snooping counters ethdev-pams

Example

- This command displays the number of messages dropped at the kernel level.

```
switch>show ip igmp snooping counters ethdev-pams
  IntfName   rxErrors   txErrors    txDrops
       et9          1          0          0
      et18          1          0          0
     mlag9          1          0          0
     mlag8          1          0          0
      et17          1          0          0
       po1          1          0          0
       po2          1          0          0
      et15          1          0          0
       et6          1          0          0
    mlag10          1          0          0
      et16          1          0          0
     mlag7          1          0          0
      et11          1          0          0
     mlag5          1          0          0
     mlag4          1          0          0
       cpu          1          0          0
      et13          1          0          0
switch>
```

show ip igmp snooping groups

The **show ip igmp snooping groups** command displays IGMP snooping statistics. Available information includes the physical ports that send and receive information, the time when multicast data was originally and most recently heard on the ports, and the version number of the IGMP messages. Command provides options that restrict the output to specific VLANs, ports, and groups.

Command Mode

EXEC

Command Syntax

show ip igmp snooping groups [*VLAN_ID*][*PORT_INT*][*GROUPS*][*DATA*]

Parameters

- ***VLAN_ID*** specifies VLAN for which command displays information. Options include:
  - <no parameter> displays information for all VLANs.
  - **vlan** *v_num* displays information for VLAN *v_num* (1 to 4094).
- ***PORT_INT*** specifies physical ports for which command displays information. Options include:
  - <no parameter> displays information for all physical ports.
  - **interface ethernet** *e_range*, where *e_range* is the number, range, or list of Ethernet ports.
  - **interface port-channel** *p_range*, where *p_range* is the number, range, or list of channel ports.
- ***GROUPS*** specifies the multicast groups. Options include:
  - <no parameter> all multicast groups on all specified ports.
  - *mgroup_address* multicast group specified by IPv4 address (dotted decimal notation).
  - **dynamic** multicast groups learned through IGMP.
  - **user** multicast groups manually added.
- ***DATA*** specifies the type of information displayed. Options include:
  - <no parameter> VLAN number and port-list for each group.
  - **detail** port-specific information for each group, including transmission times and expiration.

Example

- This command displays the port lists for all multicast groups.

```
switch>show ip igmp snooping groups
Vlan  Group            Type     Version             Port-List
------------------------------------------------------------------------------
-
1     239.255.255.250  -        -                   Po1, Po2
26    239.255.255.250  -        -                   Cpu, Et3, Et4, Et10, Et23,
                                                    Et27
switch>
```

- This command displays detailed port information of all multicast groups.

```
switch>show ip igmp snooping groups detail
Vlan Group           IP              First     Last    Expire    Ver Filter Port
                                      Heard     Heard                Mode
-------------------------------------------------------------------------------
-
1    239.255.255.250 172.17.3.73     2536:15   0:47    3:33      v2  0      Po2
1    239.255.255.250 172.17.0.37     31532:48  0:18    1:27      -   -      Po1
26   239.255.255.250 172.17.26.189   5:07      0:52    3:28      v2  0      Et3
26   239.255.255.250 172.17.26.182   17:34     3:02    1:18      v2  0      Et3
26   239.255.255.250 172.17.26.245   1046:47   0:57    3:23      v2  0      Et4
26   239.255.255.250 172.17.26.184   27:41     0:53    3:27      v2  0      Et10
26   239.255.255.250 172.17.26.161   9:16      0:56    3:24      v2  0      Et23
26   239.255.255.250 172.17.26.62    90:24     0:50    3:30      v2  0      Et27
26   239.255.255.250 172.17.26.1     31532:52  0:04    1:41      -   -      Cpu
switch>
```

- This command displays the port lists for all dynamic multicast groups.

```
switch>show ip igmp snooping groups dynamic
Vlan  Group            Type     Version             Port-List
-------------------------------------------------------------------------------
-
1     239.255.255.250  -        -                   Po1, Po2
26    239.255.255.250  -        -                   Cpu, Et3, Et4, Et10, Et23,
                                                    Et27, Et34
switch>
```

- This command displays the detailed port information for all dynamic multicast groups.

```
switch>show ip igmp snooping groups dynamic detail
Vlan Group           IP              First     Last    Expire    Ver Filter Port
                                      Heard     Heard                Mode
-------------------------------------------------------------------------------
-
1    239.255.255.250 172.17.3.73     2539:16   1:37    2:43      v2  0      Po2
1    239.255.255.250 172.17.0.37     31535:49  0:19    1:26      -   -      Po1
26   239.255.255.250 172.17.26.189   8:08      3:53    0:27      v2  0      Et3
26   239.255.255.250 172.17.26.182   20:35     1:49    2:31      v2  0      Et3
26   239.255.255.250 172.17.26.245   1049:48   1:46    2:34      v2  0      Et4
26   239.255.255.250 172.17.26.184   30:42     1:44    2:36      v2  0      Et10
26   239.255.255.250 172.17.26.161   12:17     3:57    0:23      v2  0      Et23
26   239.255.255.250 172.17.26.143   1:53      1:53    2:27      v2  0      Et23
26   239.255.255.250 172.17.26.62    93:25     1:48    2:32      v2  0      Et27
26   239.255.255.250 172.17.26.164   0:32      0:31    3:49      v2  0      Et34
26   239.255.255.250 172.17.26.1     31535:53  0:05    1:40      -   -      Cpu
switch>
```

- This command displays the port lists for all static (user configured) multicast groups.

```
switch>show ip igmp snooping groups user
Vlan  Group            Type     Version             Port-List
-------------------------------------------------------------------------------
-
1     239.255.255.250  -        -                   Po1, Po2
26    239.255.255.250  -        -                   Cpu, Et3, Et4, Et10, Et23,
                                                    Et27, Et34
switch>
```

- This command displays detailed port information for all user configured (static) multicast groups.

```
switch>show ip igmp snooping groups user detail
Vlan Group           IP              First     Last    Expire    Ver Filter Port
                                      Heard     Heard                 Mode
-------------------------------------------------------------------------------
-
1    239.255.255.250 172.17.3.73     2539:50   0:06    4:14      v2  0      Po2
1    239.255.255.250 172.17.0.37     31536:23  0:23    1:22      -   -      Po1
26   239.255.255.250 172.17.26.182   21:09     0:21    3:59      v2  0      Et3
26   239.255.255.250 172.17.26.245   1050:22   0:17    4:03      v2  0      Et4
26   239.255.255.250 172.17.26.184   31:16     0:17    4:03      v2  0      Et10
26   239.255.255.250 172.17.26.161   12:51     0:17    4:03      v2  0      Et23
26   239.255.255.250 172.17.26.143   2:27      2:27    1:53      v2  0      Et23
26   239.255.255.250 172.17.26.62    93:59     0:22    3:58      v2  0      Et27
26   239.255.255.250 172.17.26.164   1:06      0:21    3:59      v2  0      Et34
26   239.255.255.250 172.17.26.1     31536:27  0:09    1:36      -   -      Cpu
switch>
```

- This command displays detailed port information for multicast group 239.255.255.253 on VLAN 10.

```
switch>show ip igmp snooping groups vlan 10 239.255.255.253 detail
Vlan Group           IP              First     Last    Expire    Ver Filter Port
                                      Heard     Heard                 Mode
-------------------------------------------------------------------------------
-
10   239.255.255.253 10.255.255.246  7177:16   0:08    2:07      v2  0      Po7
10   239.255.255.253 10.255.255.247  7177:20   0:03    2:12      v2  0      Po7
10   239.255.255.253 10.255.255.248  7177:16   0:06    2:09      v2  0      Po7
10   239.255.255.253 10.255.255.254  7177:56   0:07    1:38      -   -      Cpu
switch>
```

show ip igmp snooping groups count

The **show ip igmp snooping groups count** command displays the number of multicast groups on the switch. Command provides options to only include specific VLANs and ports.

Command Mode

EXEC

Command Syntax

show ip igmp snooping groups [*VLAN_ID*][*PORT_INT*] count [*DATA*]

Parameters

- ***VLAN_ID*** specifies VLAN for which command displays information. Options include:
  - <no parameter> all VLANs.
  - **vlan** *v_num* specified VLAN.
- ***PORT_INT*** specifies physical ports for which command displays information. Options include:
  - <no parameter> all physical ports.
  - **interface ethernet** *e_range* specified Ethernet ports.
  - **interface port-channel** *p_range* specified port channels.

  Valid *e_range* and *p_range* formats include number, number range, or comma-delimited list of numbers and ranges.
- ***DATA*** specifies the type of information displayed. Options include:
  - <no parameter> number of multicast group on specified VLAN and ports.
  - **detail** number of multicast group on specified VLAN and ports.

Example

- This command displays the number of multicast groups on the switch.

```
switch>show ip igmp snooping groups count
Total number of multicast groups: 2
switch>
```

show ip igmp snooping mrouter

The **show ip igmp snooping mrouter** command displays the status of dynamic and static multicast router ports. Command provides options to include only specific VLANs.

Command Mode

EXEC

Command Syntax

show ip igmp snooping mrouter [*VLAN_ID*] [*DATA*]

Parameters

- ***VLAN_ID*** specifies VLAN for which command displays information. Options include:
  - <no parameter> all VLANs.
  - **vlan** *v_num* specified VLAN.
- ***DATA*** specifies the type of information displayed. Options include:
  - <no parameter> displays VLAN number and port-list for each group.
  - **detail** displays port-specific data for each group; includes transmission times and expiration.

Examples

- This command displays port information of each multicast router on all VLANs.

```
switch>show ip igmp snooping mrouter
Vlan    Interface-ports
---------------------------------------------------------
1       Po1(dynamic)
20      Po1(dynamic)
26      Cpu(dynamic)
2028    Cpu(dynamic), Po1(dynamic)
switch>
```

- This command displays multicast router information for each port.

```
switch>show ip igmp snooping mrouter detail
Vlan   Intf      Address           FirstHeard LastHeard  Expires    Type
------------------------------------------------------------------------------
1      Po1       172.17.0.37       31549:12   0:12       1:33       pim
20     Po1       172.17.20.1       7066:51    0:19       1:26       pim
26     Cpu       172.17.26.1       31549:16   0:28       1:17       pim
2028   Po1       172.17.255.29     31549:10   0:18       1:27       pim
2028   Cpu       172.17.255.30     31549:14   0:28       1:17       pim
switch>
```

show ip igmp snooping querier

The **show ip igmp snooping querier** command displays snooping querier configuration and status information. Command provides options to only include specific VLANs.

Command Mode

EXEC

Command Syntax

show ip igmp snooping querier [*STATUS*][*VLAN_ID*][*DATA*]

Parameters

- ***STATUS*** specifies the type of information displayed. Options include:
  - <no parameter> querier IP address, port, and IGMP version.
  - **status** querier configuration parameters.
- ***VLAN_ID*** specifies VLANs for which command displays information. Options include:
  - <no parameter> all VLANs.
  - **vlan** *v_num* specified VLAN.
- ***DATA*** specifies the type of information displayed. Options include:
  - <no parameter> displays VLAN number and port-list for each group.
  - **detail** displays port-specific data for each group; includes transmission times and expiration.

Example

- This command displays the querier IP address, version, and port servicing each VLAN.

```
switch>show ip igmp snooping querier
Vlan  IP Address       Version  Port
---------------------------------------
1     172.17.0.37      v2       Po1
20    172.17.20.1      v2       Po1
26    172.17.26.1      v2       Cpu
2028  172.17.255.29    v2       Po1
switch>
```

- This command displays the querier configuration parameters for each VLAN.

```
switch>show ip igmp snooping querier status
   Global IGMP Querier status
-----------------------------------
admin state              : Enabled
source IP address        : 0.0.0.0
query-interval (sec)     : 125.0
max-response-time (sec)  : 10.0
querier timeout (sec)    : 130.0

Vlan Admin    IP             Query    Response Querier Operational
     State                   Interval Time     Timeout State
-----------------------------------------------------------------------
1    Enabled  0.0.0.0         125.0    10.0     130.0   Non-Querier
4    Enabled  0.0.0.0         125.0    10.0     130.0   Non-Querier
20   Enabled  0.0.0.0         125.0    10.0     130.0   Non-Querier
22   Enabled  0.0.0.0         125.0    10.0     130.0   Non-Querier
28   Enabled  0.0.0.0         125.0    10.0     130.0   Non-Querier
```

show ip igmp snooping querier counters

The **show ip igmp snooping querier counters** command displays the counters from the querier, as learned through Internet Group Management Protocol (IGMP).

Command Mode

EXEC

Command Syntax

show ip igmp querier counters [*VLAN_ID*]

Parameters

- ***VLAN_ID*** specifies VLANs for which command displays information. Options include:
  - <no parameter> displays information for all VLANs.
  - **vlan** *v_num* displays information for specified VLAN.

Example

- This command displays the counters from the querier.

```
switch>show ip igmp snooping querier counters
------------------------------------------------------------------------
Vlan: 1    IP Addr: 100.0.0.1       Op State: Querier     Version:  v3

v1 General Queries Sent         :0
v1 Queries Received             :0
v1 Reports Received             :0
v2 General Queries Sent         :1
v2 Queries Received             :0
v2 Reports Received             :25
v2 Leaves Received              :0
v3 General Queries Sent         :655
v3 GSQ Queries Sent             :0
v3 GSSQ Queries Sent            :8
v3 Queries Received             :654
v3 Reports Received             :2385
Error Packets                   :0
Other Packets                   :0
switch>
```

show ip igmp snooping querier membership

The **show ip igmp snooping querier membership** command displays the membership from the querier, as learned through Internet Group Management Protocol (IGMP).

Command Mode

EXEC

Command Syntax

show ip igmp querier membership [*VLAN_ID* [*GROUP_LIST*]]

Parameters

- ***VLAN_ID*** specifies VLANs for which command displays information. Options include:
  - <no parameter> displays information for all VLANs.
  - **vlan** *v_num* displays information for specified VLAN.
- ***GROUP_LIST*** list of groups for which the command displays information. Options include:
  - <no parameter> all multicast groups within specified VLAN.
  - **group** *ipv4_addr* single multicast group address (dotted decimal notation).

Example

- This command displays the membership from the querier fro VLAN 1.

```
switch>show ip igmp snooping querier membership
-------------------------------------------------------------------------
Vlan: 1    Elected: 10.0.0.1       QQI: 125  QRV: 2  QRI: 10  GMI: 260

Groups           Mode  Ver  Num of Sources
-------------------------------------------------------------------------
10.0.0.2        EX    v3   0 []
10.0.0.3        IN    v3   2 [ 3.3.3.3, 3.3.3.4 ]
10.0.0.4        EX    v3   0 []
10.0.0.13       EX    v3   0 []
10.0.0.22       EX    v3   0 []
10.0.0.1        IN    v3   3 [ 5.6.7.9, 5.6.7.8, ... ]
switch>
```

show ip igmp snooping report-flooding

The **show ip igmp snooping report-flooding** command displays IGMP snooping L2 report flooding configuration and status information. Command provides options to only include specific VLANs.

Command Mode

EXEC

Command Syntax

```
show ip igmp snooping report-flooding [VLAN_ID][DATA]
```

Parameters

- ***VLAN_ID*** specifies VLANs for which command displays information. Options include:
  - <no parameter> all VLANs.
  - **vlan** *v_num* specified VLAN.
- ***DATA*** specifies the type of information displayed. Options include:
  - <no parameter> displays VLAN number and port-list for each group.
  - **detail** displays port-specific data for each group; includes transmission times and expiration.

show ip igmp static-groups

The **show ip igmp static-groups** command displays information about all configured IGMP multicast static groups. IGMP multicast static groups are assigned with the **ip igmp static-group** command.

Command Mode

EXEC

Command Syntax

show ip igmp static-groups [*INFO_LEVEL*] [interface *INT_NAME*]

Parameters

- ***INFO_LEVEL*** specifies the type of information displayed. Options include
  - <no parameter> VLAN number and port-list for each group.
  - **detail** port-specific information for each group, including transmission times and expiration.
- ***INT_NAME*** Interface type and number. Values include
  - <no parameter> static groups on all interfaces.
  - **ethernet** *e_num* Ethernet interface specified by *e_num*.
  - **loopback** *l_num* Loopback interface specified by *l_num*.
  - **management** *m_num* Management interface specified by *m_num*.
  - **port-channel** *p_num* Port-Channel Interface specified by *p_num*.
  - **vlan** *v_num* VLAN interface specified by *v_num*.
  - **vxlan** *vx_num* VXLAN interface specified by *vx_num*.

Related Commands

- **show ip igmp static-groups acl**
- **show ip igmp static-groups group**

Examples

- This command displays information about all multicast static groups.

```
switch>show ip igmp static-groups
Interface Vlan281:
        Manually configured groups:
Interface Port-Channel999:
        Manually configured groups:
switch>
```

- This command displays information about the multicast static groups on VLAN interface 21.

```
switch>show ip igmp static-groups interface vlan 21
Interface Vlan281:
        Manually configured groups:
switch>
```

show ip igmp static-groups acl

The **show ip igmp static-groups acl** command displays information about the IGMP multicast static groups that are configured by the specified access control list (ACL). The command also displays problems with an ACL that prevent its assignment to an interface.

Command Mode

EXEC

Command Syntax

show ip igmp static-groups acl

Example

The following **show ip igmp static-group acl** command example references these ACLs:

```
ip access-list 1
   10 permit igmp host 10.1.1.1 10.1.1.0/29
   20 permit igmp host 10.1.1.2 10.1.1.0/29
!
ip access-list 2
   10 permit igmp 10.1.1.0/29 host 10.1.1.1
!
ip access-list 3
   10 deny igmp host 10.1.1.1 255.1.1.0/29
!
ip access-list 4
   10 permit igmp host 10.1.1.1 10.1.1.0/29
   20 permit igmp 10.1.1.0/29 host 10.1.1.1
```

- This command displays static group configuration data about the various ACLs.

```
switch>show ip igmp static-group acl 1
acl 1
        ( 10.1.1.1, 10.1.1.0/29 )
        ( 10.1.1.2, 10.1.1.0/29 )
Interfaces using this ACL for static groups:
        Ethernet12
switch>show ip igmp static-group acl 2
acl 2
        Seq no 30: source address must be a single host or *, not a range
Interfaces using this ACL for static groups:
        Ethernet8
switch>show ip igmp static-group acl 3
acl 4
        Seq no 10: action must be 'permit'
Interfaces using this ACL for static groups:
         none
switch>show ip igmp static-group acl 4
acl 5
        ( 10.1.1.1, 10.1.1.0/29 )
        Seq no 20: source address must be a single host or *, not a range
Interfaces using this ACL for static groups:
         none
switch>
```

show ip igmp static-groups group

The **show ip igmp static-groups group** command displays information about all specified IGMP multicast static groups. IGMP multicast static groups are assigned with the **ip igmp static-group** command.

Command Mode

EXEC

Command Syntax

```
show ip igmp static-groups group [GROUP_LIST]
```

Parameters

- ***GROUP LIST*** Groups for which command displays information
  - <no parameter> all multicast groups.
  - *group_address* single multicast group address (dotted decimal notation).

Related Commands

- **show ip igmp static-groups**

show ip igmp statistics

The **show ip igmp statistics** command displays IGMP transmission statistics for the specified interface.

Command Mode

EXEC

Command Syntax

show ip igmp statistics [*INTERFACE_ID*]

Parameters

- ***INTERFACE_ID*** Specifies tnterface for which command returns data. Options include:
  - <no parameter> all interfaces.
  - **interface ethernet** *e_num* Ethernet interface specified by *e_num*.
  - **interface loopback** *l_num* Loopback interface specified by *l_num*.
  - **interface management** *m_num* Management interface specified by *m_num*.
  - **interface port-channel** *p_num* Port-Channel Interface specified by *p_num*.
  - **interface vlan** *v_num* VLAN interface specified by *v_num*.
  - **interface vxlan** *vx_num* VXLAN interface specified by *vx_num*.

Example

- This command displays IGMP transmission statistics for ethernet 1 interface.

```
switch>show ip igmp statistics interface ethernet 1
IGMP counters for Ethernet1:
  V1 queries sent: 0
  V2 queries sent: 0
  V3 queries sent: 3
  Total general queries sent: 3
  V3 group specific queries sent: 0
  V3 group-source specific queries sent: 0
  V1 queries received: 0
  V2 queries received: 0
  V3 queries received: 0
  V1 reports received: 0
  V2 reports received: 0
  V3 reports received: 14
  V2 leaves received: 0
  Error Packets received: 0
  Other Packets received: 0
switch>
```

Chapter 36

Protocol Independent Multicast

Protocol Independent Multicast (PIM) distributes multicast data using routes gathered by other protocols. PIM Sparse Mode (PIM-SM) is designed for networks where multicast group recipients are sparsely distributed, including wide-area and inter-domain networks. These sections describe the Arista PIM implementation:

- Section 36.1: Introduction provides an overview of PIM.
- Section 36.2: Configuring PIM describes configuration tasks that implement PIM.
- Section 36.3: Multicast Example provides a multicast implementation scenario.
- Section 36.4: PIM Commands contains PIM command descriptions.

36.1 Introduction

Protocol Independent Multicast (PIM) distributes multicast data using routes gathered by other protocols. PIM Sparse Mode (PIM-SM), defined in RFC 4601, is a multicast routing protocol designed for networks where multicast group recipients are sparsely distributed, including wide-area and inter-domain networks.

PIM builds and maintains multicast routing trees using reverse path forwarding (RPF) on a unicast routing table. PIM can use routing tables consisting of OSPF, BGP, RIP, and static routes. All sources send traffic to the multicast group through shared trees that have a common root node called the Rendezvous Point (RP). Each host (senders and receivers) is associated with a Designated Router (DR) that acts for all directly connected hosts in PIM-SM transactions.

36.1.1 Protocol Overview

PIM uses an MRIB that is populated from the unicast table. The MRIB provides the next-hop router along a multicast-capable path to each destination subnet. This determines the next-hop neighbor for sending PIM Join or Prune messages.

PIM establishes multicast routes through three phases:

- Establishing the RP Tree
- Eliminating Encapsulation
- Establishing the Shortest Path Tree (SPT)

Establishing the RP Tree (Phase 1)

The RP tree is a distribution network that all sources share to deliver multicast data. The root of the RP tree is the Rendezvous Point.

The process starts when a receiver requests multicast data from a group (G). The receiver's DR sends a PIM (*,G) Join message toward the multicast group's RP. As the message travels towards the RP, it instantiates the multicast (*,G) state in each router on the path. After many receivers join the group, the Join messages converge on the RP to form the RP tree.

The DR resends Join messages periodically, while it has a receiver in the group, to prevent state timeout expiry in the routers along the path. When all receivers on a DR's subnet leave a group, the DR sends a (*,G) Prune message towards the RP to remove the state from the routers.

A multicast sender transmits multicast data to the RP through its DR. The DR encapsulates the multicast packets and sends them as unicast packets. The RP extracts the native multicast packet and sends it to the RP tree towards the group members.

Eliminating Multicast Encapsulation (Phase 2)

Data encapsulation, while initially required before the multicast path is established, is inefficient because it requires the transmission of data that is extraneous to multicast. Phase 2 establishes states in the routers that support the transmission of native multicast packets.

When the RP receives an encapsulated packet from source S on group G, it sends a source-specific (S,G) join message towards the source. As the message travels towards S, it instantiates the (S,G) state on each router in the path. This state is used only to forward packets for group G from source S. Data packets on the (S,G) path are also routed into the RP tree when they encounter an (*,G) router.

When the RP starts receiving native packets from the sources, it sends a Register-Stop message to the source's DR, halting packet encapsulation. At this time, traffic flows natively from the source along a source-specific tree to the RP, then along the shared RP tree to the receivers.

Establishing the Shortest Path Tree (Phase 3)

The third phase establishes the shortest path from the multicast source to all receivers.

When a multicast packet arrives at the receiver, its router (typically the DR) sends a Join message towards the source to instantiate the (S,G) state in all routers along its path. The message eventually reaches either the source's subnet or a router that already has an (S,G) state. This causes data to flow from the source to the receiver following the (S,G) path. At this time, the receiver is receiving data from the Shortest Path Tree (SPT) and the RP Tree (RPT).

The DR (or upstream router) eliminates the data transmission along the RPT by sending a Prune message (S,G,rpt) towards the RP. The message travels hop-by-hop, instantiating the state on each router in the path, continues until it reaches the RP or a router that needs traffic from S for other receivers.

36.1.2 Rendezvous Points (RP)

An RP is a router that is configured as the root of a non-source-specific distribution tree for a multicast group. Join messages for receivers to a group are sent towards the RP. Data from senders is sent to the RP, allowing receivers to discover sender identity and begin receiving group traffic. Paths through RP routers are temporary; when traffic volume reaches a sufficient level, the receiver joins a source-specific tree and the path through the RP is dropped.

The switch supports two methods of mapping RPs to multicast groups:

- Static: RPs are statically configured through a CLI statement.

- Dynamic: RPs are dynamically selected by a bootstrap router from a set of candidate RPs.

While dynamic RP mappings have priority over static maps by default, a static RP can be configured to override dynamic mappings.

Section 36.2.2 describes the configuration of rendezvous points.

36.2 Configuring PIM

The following sections describe the configuration of static RPs, dynamic RPs, and anycast-RPs. RP implementation is defined through the following RFCs:

- RFC 5059: Bootstrap Router (BSR) for Protocol Independent Multicast (PIM).
- RFC 6226: PIM Group-to-Rendezvous-Point Mapping.

This section describes the following configuration tasks:

- Section 36.2.1: Enabling PIM
- Section 36.2.2: Rendezvous Points (RPs)
- Section 36.2.3: Hello Messages
- Section 36.2.4: Designated Router Election
- Section 36.2.5: Join-Prune Messages

36.2.1 Enabling PIM

By default, PIM is disabled on an interface. The **ip pim sparse-mode** command enables PIM on the configuration mode interface.

Example

- This command enables PIM and IGMP on VLAN interface 8.

```
switch(config-if-Vl8)#ip pim sparse-mode
switch(config-if-Vl8)#
```

36.2.2 Rendezvous Points (RPs)

Networks that run PIM sparse mode require a rendezvous point (RP). The switch supports dynamic RPs, static RPs, and anycast-RP.

Configuring Static RPs

The **ip pim rp-address** command configures a static RP, providing an option to override dynamic RPs.

Examples

- This command creates a static RP at 10.17.255.83 that maps to all multicast groups (224/4) and override dynamic RPs.

```
switch(config)#ip pim rp-address 10.17.255.83 override
switch(config)#
```

- This command creates a static RP at 10.21.18.23 that maps to the multicast groups at 238.1.12.0/24.

```
switch(config)#ip pim rp-address 10.21.18.23 238.1.12.0/24
switch(config)#
```

Configuring Dynamic RPs

Dynamic RP selection is implemented through a bootstrap router (BSR), which is a PIM router within the PIM domain that selects RPs from a list of candidates. A subset of PIM routers within the domain are configured as candidate bootstrap routers (C-BSRs). Through the exchange of bootstrap messages (BSMs), the C-BSRs elect the BSR, which then uses BSMs to inform all domain routers of its status.

The BSR holdtime defines the timeout period that an elected BSR remains valid after the receipt of a BSM and is also used in dynamic RP configuration. Holdtime is designated by the BSR router and communicated to other routers through BSMs.

Another subset of domain PIM routers are configured as candidate RPs (C-RPs). The BSR creates a set of qualifying RPs from the list of C-RPs, then distributes the group-to RP mapping set to all domain routers through BSMs. Each PIM router, after receiving this set, uses a standard algorithm defined in RFC 6226 to select one RP per multicast group.

The **ip pim bsr-candidate** command configures the switch as a candidate BSR router (C-BSR). Command parameters specify the switch's BSR address, the interval between BSM transmissions, and the switch's BSR priority rating. Priority ratings range from 0 to 255 with a default of 128. Higher numbers denote higher priority during BSR elections.

Example

- These commands configure the switch as a BSR candidate, using the IP address assigned to VLAN interface 24 as its BSR address. The BSM transmission interval is set to 30 seconds and the priority is set to 192.

```
switch(config)#ip pim bsr-candidate vlan 24 priority 192 interval 30
switch(config)#
```

The **ip pim bsr-holdtime** command specifies the value the switch inserts in the ***holdtime*** field of bootstrap messages (BSMs) that it sends. This value becomes the holdtime for the PIM domain if the switch is elected as the BSR.

Example

- This command specifies 75 seconds as the value that the switch inserts into BSM holdtime fields.

```
switch(config)#ip pim bsr-holdtime 75
switch(config)#
```

The **ip pim rp-candidate** command configures the switch as a candidate rendezvous point (C-RP) . The BSR selects a multicast group's dynamic RP set from the list of C-RPs. Command parameters specify the switch's RP address, C-RP advertisement interval, and priority rating. The priority rating is used by the BSR when selecting RPs. The C-RP advertisement interval specifies the period between successive C-RP advertisement message transmissions to the BSR.

Running-config may contain multiple **ip pim rp-candidate** statements to support multiple multicast groups:

- All commands must specify the same interface. Issuing a command with an interface that differs from existing commands removes all existing commands from ***running-config***.
- ***Running-config*** stores the ***interval*** setting in a separate statement that applies to all **rp-candidate** statements. Commands that specify an interval that differs from the previously configured value place the new value in ***running-config***. This new value applies to all **rp-candidate** statements.

Example

- This command configures a switch as a candidate RP for the multicast group 235.1.1.0/24, with a priority of 48 and a RP advertisement interval of 45 seconds.

```
switch(config)#ip pim rp-candidate vlan 24 235.1.1.0/24 priority 48 interval 45
switch(config)#
```

By default, the switch transmits bootstrap router messages (BSMs) over all PIM-SM enabled interfaces. The **ip pim bsr-border** command prevents the switch from transmitting BSMs over the configuration mode interface.

Example

- This command enables **ip pim bsr-border** in VLAN 10.

```
switch(config)#interface vlan 10
switch(config-if-Vl10)#ip pim bsr-border
switch(config-if-Vl10)#
```

Anycast-RP

PIM Anycast-RP defines a single RP address that is on multiple devices. An anycast-RP set consists of the routers configured with the same anycast-RP address. Anycast-RP provides redundancy protection and load balancing. The anycast-RP set supports all multicast groups.

The **ip pim anycast-rp** command configures the switch as a member of an anycast-RP set and establishes a communication link with another member of the set.

Example

- These commands configure a switch (IP address 10.1.1.14) into an anycast-RP set with an RP address of 10.17.255.2. The anycast-RP set contains three other routers, located at 10.1.2.14, 10.1.3.14, and 10.1.4.14. It sets the number of unacknowledged register messages it sends to each router at 15.

```
switch(config)#ip pim anycast-rp 10.17.255.2 10.1.1.14 register-count 15
switch(config)#ip pim anycast-rp 10.17.255.2 10.1.2.14 register-count 15
switch(config)#ip pim anycast-rp 10.17.255.2 10.1.3.14 register-count 15
switch(config)#ip pim anycast-rp 10.17.255.2 10.1.4.14 register-count 15
```

36.2.3 Hello Messages

Multicast routers send PIM router query (Hello) messages to determine the designated router (DR) for each subnet. The DR then sends registration messages to the RP.

The **ip pim query-interval** command specifies the transmission interval between PIM hello messages originating from the specified VLAN interface.

Example

- This command configures 45 second intervals between hello messages originating from VLAN interface 4.

```
switch(config-if-Vl4)#ip pim query-interval 45
switch(config-if-Vl4)#
```

36.2.4 Designated Router Election

PIM uses these criteria for electing designated routers (DR):

- If one router does not advertise a dr-priority value, the router with the highest IP address becomes the Designated Router.
- If all routers advertise a dr-priority value, the router with the highest dr-priority value becomes the Designated Router.

The **ip pim dr-priority** command sets the DR priority value that the switch advertises. If running-config does not contain a **ip pim dr-priority** statement, the switch does not advertise a dr-priority value.

Examples

- This command configures the dr-priority value of 15 on VLAN interface 4.

```
switch(config-if-Vl4)#ip pim dr-priority 15
switch(config-if-Vl4)#
```

- This command removes the **ip-pim dr-priority** statement (VLAN interface 4) from ***running-config***.

```
switch(config-if-Vl4)#no ip pim dr-priority
switch(config-if-Vl4)#
```

36.2.5 Join-Prune Messages

Join/prune messages are sent by the PIM designated router (DR) toward the rendezvous point (RP). These messages inform other PIM routers about clients that want to become receivers (Join) or stop being receivers (Prune) for the groups.

The **ip pim join-prune-interval** command specifies the period between join/prune messages that the switch originates from the specified VLAN interface and sends to the upstream RPF neighbor.

Example

- This command configures 75 second intervals between join/prune messages originating from VLAN interface 4.

```
switch(config-if-Vl4)#ip pim join-prune-interval 75
switch(config-if-Vl4)#
```

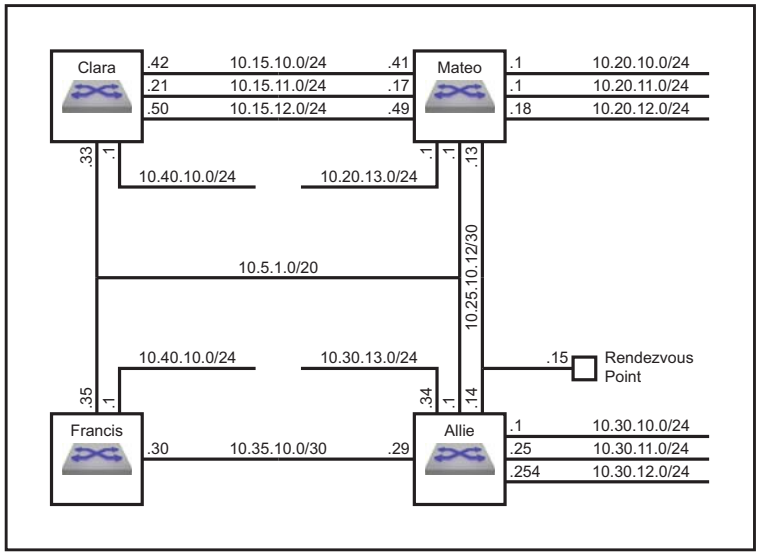
36.3 Multicast Example

This section provides an example network that implements multicast and includes the required commands.

36.3.1 Diagram

Figure 36-1 displays the multicast network example. The network contains four routers. Multicast routing is enabled on two switches. One switch has its IGMP Snooping Querier enabled.

Figure 36-1: Multicast Example



The example multicast network implements these multicast parameters:

Rendezvous Point Address: 10.25.10.15

Switch Clara

- IGMP Snooping : disabled
- Subnet Summary:
  - 10.40.10.0/24: VLAN 11
  - 10.15.10.0/24: VLAN 12
  - 10.15.11.0/24: VLAN 13
  - 10.15.12.0/24: VLAN 14
  - 10.5.1.0/20: VLAN 10

Switch Mateo

- IGMP Snooping : disabled
- Subnet Summary:
  - 10.20.13.0/24: VLAN 18
  - 10.20.10.0/24: VLAN 15
  - 10.20.11.0/24: VLAN 16

- 10.20.12.0/24: VLAN 17
- 10.15.10.0/24: VLAN 12
- 10.15.11.0/24: VLAN 13
- 10.15.12.0/24: VLAN 14
- 10.25.10.12/30: VLAN 19
- 10.5.1.0/20: VLAN 10

Switch Allie

- IGMP Snooping : enabled
- Multicast Routing: enabled
- Querier: enabled
- Rendezvous Point Address: 10.25.10.15
- MFIB activity polling interval: 5 second
- Subnet Summary:
  - 10.30.13.0/24: VLAN 23
  - 10.30.10.0/24: VLAN 20 – PIM-SM enabled
  - 10.30.11.0/24: VLAN 21 – PIM-SM enabled
  - 10.30.12.0/24: VLAN 22
  - 10.25.10.12/30: VLAN 19
  - 10.35.10.0/30: VLAN 24 – PIM-SM enabled
  - 10.5.1.0/20: VLAN 10 – PIM-SM enabled

Switch Francis

- IGMP Snooping : enabled
- Multicast Routing: enabled
- Subnet Summary:
  - 10.40.10.0/24: VLAN 25 – PIM-SM enabled
  - 10.35.10.0/30: VLAN 24 – PIM-SM enabled
  - 10.5.1.0/20: VLAN 10

36.3.2 Example

This example configures multicasting.

Step 1 Configure the interface addresses

a Router Clara interfaces

```
Clara(config)#interface vlan 11
Clara(config-if-vl11)#ip address 10.40.10.1/24
Clara(config-if-vl11)#interface vlan 12
Clara(config-if-vl12)#ip address 10.15.10.42/24
Clara(config-if-vl12)#interface vlan 13
Clara(config-if-vl13)#ip address 10.15.11.21/24
Clara(config-if-vl13)#interface vlan 14
Clara(config-if-vl14)#ip address 10.15.12.50/24
Clara(config-if-vl14)#interface vlan 10
Clara(config-if-vl10)#ip address 10.5.1.33/20
Clara(config-if-vl10)#router ospf 1
Clara(config-router-ospf)#redistribute static
```

b Router Mateo interfaces

```
Mateo(config)#interface vlan 18
Mateo(config-if-vl18)#ip address 10.20.13.1/24
Mateo(config-if-vl18)#interface vlan 15
Mateo(config-if-vl15)#ip address 10.20.10.1/24
Mateo(config-if-vl15)#interface vlan 16
Mateo(config-if-vl16)#ip address 10.20.11.1/24
Mateo(config-if-vl16)#interface vlan 17
Mateo(config-if-vl17)#ip address 10.20.12.16/24
Mateo(config-if-vl17)#interface vlan 12
Mateo(config-if-vl12)#ip address 10.15.10.41/24
Mateo(config-if-vl12)#interface vlan 13
Mateo(config-if-vl13)#ip address 10.15.11.17/24
Mateo(config-if-vl13)#interface vlan 14
Mateo(config-if-vl14)#ip address 10.15.12.49/24
Mateo(config-if-vl14)#interface vlan 19
Mateo(config-if-vl19)#ip address 10.25.10.13/30
Mateo(config-if-vl19)#interface vlan 10
Mateo(config-if-vl10)#ip address 10.5.1.1/20
Mateo(config-if-vl10)#router ospf 1
Mateo(config-router-ospf)#redistribute static
```

c Router Allie interfaces

```
Allie(config)#interface vlan 23
Allie(config-if-vl23)#ip address 10.30.13.34/24
Allie(config-if-vl23)#interface vlan 20
Allie(config-if-vl20)#ip address 10.30.10.1/24
Allie(config-if-vl20)#interface vlan 21
Allie(config-if-vl21)#ip address 10.30.11.25/24
Allie(config-if-vl21)#interface vlan 22
Allie(config-if-vl22)#ip address 10.30.12.254/24
Allie(config-if-vl22)#interface vlan 19
Allie(config-if-vl19)#ip address 10.25.10.14/30
Allie(config-if-vl19)#interface vlan 24
Allie(config-if-vl24)#ip address 10.35.10.29/30
Allie(config-if-vl24)#interface vlan 10
Allie(config-if-vl10)#ip address 10.5.1.1/20
Allie(config-if-vl10)#router ospf 1
Allie(config-router-ospf)#redistribute static
```

d Router Francis interfaces

```
Francis(config)#interface vlan 25
Francis(config-if-vl25)#ip address 10.40.10.1/24
Francis(config-if-vl25)#interface vlan 24
Francis(config-if-vl24)#ip address 10.35.10.30/24
Francis(config-if-vl24)#interface vlan 10
Francis(config-if-vl10)#ip address 10.5.1.35/24
Francis(config-if-vl10)#router ospf 1
Francis(config-router-ospf)#redistribute static
```

Step 2 Configure the interface multicast parameters

a Router Allie interfaces

```
Allie(config-router-ospf)#interface vlan 20
Allie(config-if-vl20)#ip pim sparse-mode
Allie(config-if-vl20)#interface vlan 21
Allie(config-if-vl21)#ip pim sparse-mode
Allie(config-if-vl21)#interface vlan 24
Allie(config-if-vl24)#ip pim sparse-mode
Allie(config-if-vl24)#interface vlan 10
Allie(config-if-vl10)#ip pim sparse-mode
```

b Router Francis interfaces

```
Francis(config-router-ospf)#interface vlan 25
Francis(config-if-vl25)#ip pim sparse-mode
Francis(config-if-vl25)#interface vlan 24
Francis(config-if-vl24)#ip pim sparse-mode
```

Step 3 Configure the router multicast parameters

a Router Clara parameters

```
Clara(config-router-ospf)#exit
Clara(config)#no ip igmp snooping
```

b Router Mateo interfaces

```
Mateo(config-router-ospf)#exit
Mateo(config)#no ip igmp snooping
```

c Router Allie interfaces

```
Allie(config-if-vl10)#exit
Allie(config)#ip multicast-routing
Allie(config)#ip mfib activity polling-interval 5
Allie(config)#ip pim rp-address 10.25.10.15
```

d Router Francis interfaces

```
Francis(config-if-vl24)#exit
Francis(config)#ip multicast-routing
Francis(config)#ip pim rp-address 10.25.10.15
```

36.4 PIM Commands

PIM Configuration Commands (Global)

- ip pim anycast-rp
- ip pim bsr-candidate
- ip pim bsr-holdtime
- ip pim log-neighbor-changes
- ip pim register-source
- ip pim rp-address
- ip pim rp-candidate
- ip pim sparse-mode sg-expiry-timer
- ip pim spt-threshold
- ip pim spt-threshold group-list
- ip pim ssm range

PIM Configuration Commands (Interface)

- ip pim border-router
- ip pim bsr-border
- ip pim dr-priority
- ip pim join-prune-interval
- ip pim neighbor-filter
- ip pim query-interval
- ip pim sparse-mode

PIM Display Commands

- show ip pim bsr
- show ip pim config-sanity
- show ip pim interface
- show ip pim neighbor
- show ip pim protocol counters
- show ip pim register-source
- show ip pim rp
- show ip pim rp-candidate
- show ip pim rp-hash
- show ip pim upstream joins

ip pim anycast-rp

The **ip pim anycast-rp** command configures the switch as a member of an anycast-RP set and establishes a communication link with another member of the set.

The **no ip pim anycast-rp** and **default ip pim anycast-rp** commands remove the corresponding **ip pim anycast-rp** commands from ***running-config***. When the **no** and **default** commands do not include a peer address, all commands for the specified rp address are removed.

Command Mode

Global Configuration

Command Syntax

```
ip pim anycast-rp rp_addr peer_addr [REGISTER]
no ip pim anycast-rp rp_addr [peer_addr]
default ip pim anycast-rp rp_addr [peer_addr]
```

Parameters

- *rp_addr* Rendezvous point IP address (dotted decimal notation).
- *peer_addr* IP address of an anycast-RP set member (dotted decimal notation).
- ***REGISTER*** Number of unacknowledged register messages the switch sends to the peer router.
  - <No parameter> register count is set to default value of 10.
  - **register-count** *r_num* where *r_num* is an integer that ranges from 1 to 4294967295.
  - **register-count infinity**

Example

- These commands configure a switch (IP address 10.1.1.14) into an anycast-RP set with an RP address of 10.17.255.2. The anycast-RP set contains three other routers, located at 10.1.2.14, 10.1.3.14, and 10.1.4.14. It sets the number of unacknowledged register messages it sends to each router at 15.

```
switch(config)#ip pim anycast-rp 10.17.255.2 10.1.1.14 register-count 15
switch(config)#ip pim anycast-rp 10.17.255.2 10.1.2.14 register-count 15
switch(config)#ip pim anycast-rp 10.17.255.2 10.1.3.14 register-count 15
switch(config)#ip pim anycast-rp 10.17.255.2 10.1.4.14 register-count 15
```

ip pim border-router

The **ip pim border-router** command configures the configuration mode interface as a PIM multicast border router (MBR). A PIM MBR interface allows multicast traffic from sources that are outside of the PIM domain.

This command does not control the transmission or reception of PIM protocol packets by the interface.

Sources learned through an MBR interface are treated as local sources (directly connected to the switch). The border-bit is set in all PIM register messages sent for these sources.

Important! Configuration as an MBR and configuration in PIM sparse mode must be mutually exclusive. Ensure that PIM sparse mode is not configured by issuing the **no ip pim sparse-mode** command on the interface before issuing this command.

The **no ip pim border-router** and **default ip pim border-router** commands removes the PIM MBR configuration for the configuration mode interface by removing the corresponding **ip pim border-router** statement from ***running-config***.

Command Mode

Interface-Ethernet Configuration
Interface-Port-Channel Configuration
Interface-VLAN Configuration

Command Syntax

```
ip pim border-router
no ip pim border-router
default ip pim border-router
```

Example

- These commands configure VLAN interface 200 as a PIM MBR, then display its status.

```
switch(config)#interface vlan 200
switch(config-if-VL200)#ip address 10.44.2.1/24
switch(config-if-VL200)#no ip pim sparse-mode
switch(config-if-VL200)#ip pim border-router
switch(config-if-VL200)#show active
interface Vlan200
   ip address 10.44.2.1/24
   ip pim border-router
switch(config-if-VL200)#exit
switch(config)#show ip pim interface
Address    Interface   Mode    Neighbor   Hello DR   DR
Address  PktsQed  PktsDropped
                               Count      Intvl Pri
10.44.2.1  Vlan200     mbr     0          30    1    10.44.2.1   0        0
switch(config)#
```

ip pim bsr-border

The **ip pim bsr-border** command prevents the switch from sending bootstrap router messages (BSMs) over the configuration mode interface. By default, BSMs are transmitted over all PIM-SM enabled interfaces.

The **no ip pim bsr-border** and **default ip pim bsr-border** commands restore the transmission of BSMs over the configuration mode interface by removing the corresponding **ip pim bsr-border** statement from ***running-config***.

Command Mode

Interface-Ethernet Configuration
Interface-Port-Channel Configuration
Interface-VLAN Configuration

Command Syntax

```
ip pim bsr-border
no ip pim bsr-border
default ip pim bsr-border
```

Example

- This command prevents the switch from sending BSMs from VLAN interface 10.

```
switch(config)#interface vlan 10
switch(config-if-Vl10)#ip pim bsr-border
switch(config-if-Vl10)#
```

ip pim bsr-candidate

The **ip pim bsr-candidate** command configures the switch as a candidate BSR router (C-BSR). A BSR is a PIM router within the PIM domain through which dynamic RP selection is implemented. The BSR selects RPs from a list of candidate RPs and exchange bootstrap messages (BSM) with all routers in the domain. The BSR is elected from one of the C-BSRs through an exchange of BSMs.

A subset of PIM routers within the domain are configured as candidate bootstrap routers (C-BSRs). Through the exchange of bootstrap messages (BSMs), the C-BSRs elect the BSR, which then uses BSMs to inform all domain routers of its status.

Command parameters specify the switch's BSR address, the interval between BSM transmissions, the length of the hash mask, and the priority assigned to the switch when electing a BSR.

Entering an **ip pim bsr-candidate** command replaces any previously configured **bsr-candidate** command. If the new command does not specify a priority, hash mask length, or interval, the previously configured values persist in ***running-config***.

The **no ip pim bsr-candidate** and **default ip pim bsr-candidate** commands remove the corresponding **ip pim bsr-candidate** commands from ***running-config***. The **no** and **default** commands restore the priority, hash mask length, and interval parameters to their default values.

Command Mode

Global Configuration

Command Syntax

```
ip pim bsr-candidate INTERFACE [HASHMASK_LENGTH] [INTERVAL_PERIOD]
[PRIORITY_NUM]
no ip pim bsr-candidate [priority] [interval]
default ip pim bsr-candidate [priority] [interval]
```

Parameters

- ***INTERFACE*** Switch uses IP address of specified interface as its BSR address. Options include:
  - **ethernet** *e_num* Ethernet interface specified by *e_num*.
  - **loopback** *l_num* Loopback interface specified by *l_num*.
  - **management** *m_num* Management interface specified by *m_num*.
  - **port-channel** *p_num* Port-Channel Interface specified by *p_num*.
  - **vlan** *v_num* VLAN interface specified by *v_num*.
- ***HASHMASK_LENGTH*** Length (in bits) of the hash mask.
  - <no parameter> hash mask remains unchanged from previous setting.
  - **hashmask** <***0*** - ***32***> hash mask length (in bits). Default value is 30.
- ***INTERVAL_PERIOD*** Period between the transmission of BSMs (seconds). Default value is 60.
  - <no parameter> interval remains unchanged from previous setting.
  - **interval** <***10*** - ***536870906***> transmission interval in seconds.
- ***PRIORITY_NUM*** BSR election priority rating. Larger numbers denote higher priority. Default value is 64.
  - <no parameter> priority remains unchanged from previous setting.
  - **priority** <***0*** - ***255***> priority rating.

Example

- This command configures the switch as a BSR candidate, using the IP address assigned to VLAN interface 24 as its BSR address. The BSM transmission interval is set to 30 seconds and the priority is set to 192.

```
switch(config)#ip pim bsr-candidate vlan 24 priority 192 interval 30
switch(config)#
```

ip pim bsr-holdtime

The **ip pim bsr-holdtime** command specifies the value the switch inserts in the ***holdtime*** parameter field in bootstrap messages (BSM) that it sends. The BSR holdtime defines the timeout period that an elected BSR remains valid after the receipt of a BSM and is also used in dynamic RP configuration. BSR holdtime is designated by the BSR router and communicated to other routers through BSMs.

The **no ip pim bsr-holdtime** and **default ip pim bsr-holdtime** commands restore the default holdtime parameter field insertion value of 130 seconds by removing the **ip pim dr-priority** statement from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
ip pim bsr-holdtime period
no ip pim bsr-holdtime
default ip pim bsr-holdtime
```

Parameters

- *period* BSR holdtime (seconds). Value ranges from 12 to 1073741823 (1.073 billion). Default is 130.

Example

- This command specifies 75 seconds as the value that the switch inserts into BSM holdtime fields.

```
switch(config)#ip pim bsr-holdtime 75
switch(config)#
```

ip pim dr-priority

PIM uses these criteria for electing designated routers (DR):

- If one router does not advertise a dr-priority value, the router with the highest IP address becomes the Designated Router.
- If all routers advertise a dr-priority value, the router with the highest dr-priority value becomes the Designated Router.

The **ip pim dr-priority** command sets the dr-priority value that the configuration mode interface advertises. By default, the interface does not advertise a dr-priority value.

The **no ip pim dr-priority** and **default ip pim dr-priority** commands force the use of IP addresses to elect the designated router by removing the corresponding **ip pim dr-priority** statement from ***running-config***.

Command Mode

Interface-Ethernet Configuration
Interface-Port-Channel Configuration
Interface-VLAN Configuration

Command Syntax

```
ip pim dr-priority level
no ip pim dr-priority [level]
default ip pim dr-priority [level]
```

Parameters

- *level* DR selection priority rating. Value ranges from 0 to 4294967295.

Examples

- This command configures the dr-priority value of 15 on VLAN interface 4.

```
switch(config)#interface vlan 4
switch(config-if-Vl4)#ip pim dr-priority 15
switch(config-if-Vl4)#
```

- This command removes the **ip-pim dr-priority** statement from ***running-config***.

```
switch(config-if-Vl4)#no ip pim dr-priority
switch(config-if-Vl4)#
```

ip pim join-prune-interval

The **ip pim join-prune-interval** command specifies the period between join/prune messages that the configuration mode interface originates and sends to the upstream RPF neighbor.

The **no ip pim join-prune-interval** and **default ip pim join-prune-interval** commands restore the default join/prune interval of 60 seconds for the configuration mode interface by removing the corresponding **ip pim join-prune-interval** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration
Interface-Port-Channel Configuration
Interface-VLAN Configuration

Command Syntax

```
ip pim join-prune-interval period
no ip pim join-prune-interval [period]
default ip pim join-prune-interval [period]
```

Parameters

- *period* join/prune interval (seconds). Value ranges from 1 to 18724. Default is 60.

Example

- This command configures 75-second intervals between join/prune messages originating from VLAN interface 4.

```
switch(config)#interface vlan 4
switch(config-if-Vl4)#ip pim join-prune-interval 75
switch(config-if-Vl4)#
```

ip pim log-neighbor-changes

The **ip pim log-neighbor-changes** command configures the switch to generate a log message when a neighbor entry is added or removed from the PIM Neighbor table. This function is enabled by default.

The **no ip pim log-neighbor-changes** command disables log message generation based on changes to the PIM Neighbor table; this command is stored in the ***running-config***. The **ip pim log-neighbor-changes** and **default ip pim log-neighbor-changes** commands restore the default setting of generating log messages by deleting the **no ip pim log-neighbor-changes** statement from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
ip pim log-neighbor-changes
no ip pim log-neighbor-changes
default ip pim log-neighbor-changes
```

Examples

- This command configures the switch to stop generating log messages based on PIM Neighbor table changes.

```
switch(config)#no ip pim log-neighbor-changes
switch(config)#
```

- This command configures the switch to generate log messages when a neighbor entry is added or removed from the PIM Neighbor table.

```
switch(config)#ip pim log-neighbor-changes
switch(config)#
```

ip pim neighbor-filter

The **ip pim neighbor-filter** command configures the configuration mode interface to filter PIM control packets on the basis of neighbor addresses listed in a specified standard access list.

The **no ip pim neighbor-filter** and **default ip pim neighbor-filter** commands disable the configuration mode interface from filtering PIM control packets by removing the corresponding **ip pim neighbor-filter** command from ***running-config***.

Command Mode

```
Interface-Ethernet Configuration
Interface-Port-Channel Configuration
Interface-VLAN Configuration
```

Command Syntax

```
ip pim neighbor-filter access_list
no ip pim neighbor-filter
default ip pim neighbor-filter
```

Parameters

- *access_list* name of the standard IP access list.

Example

- This command configures the IP access list named filter_1 to filter neighbor PIM control messages for VLAN 4.

```
switch(config)#ip access-list standard filter_1
switch(config-std-acl-filter_1)#permit 10.13.24.9/24
switch(config-std-acl-filter_1)#exit
switch(config)#interface vlan 4
switch(config-if-Vl4)#ip pim neighbor-filter filter_1
switch(config-if-Vl4)#
```

ip pim query-interval

The **ip pim query-interval** command specifies the transmission interval between PIM hello messages originating from the configuration mode interface.

The **no ip pim query-interval** and **default ip pim query-interval** commands restore the default query interval of 30 seconds for the configuratiom mode interface by removing the corresponding **ip pim query-interval** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration
Interface-Port-Channel Configuration
Interface-VLAN Configuration

Command Syntax

```
ip pim query-interval period
no ip pim query-interval [period]
default ip pim query-interval [period]
```

Parameters

- *period* query interval (seconds). Value ranges from 1 to 1000000 (1 million). Default is 30.

Example

- This command configures 45 second intervals between hello messages originating from VLAN interface 4.

```
switch(config)#interface vlan 4
switch(config-if-Vl4)#ip pim query-interval 45
switch(config-if-Vl4)#
```

ip pim register-source

The **ip pim register-source** command programs the switch to fill the source field in all outbound PIM SM register packets with the IP address of a specified interface or the incoming interface of the group specified by the message. By default, the source field is filled with the IP address from the interface associated with the best route to the RP.

The **no ip pim register-source** and **default ip pim register-source** commands restore the default method of filling the register packet source field by removing the **ip pim register-source** statement from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
ip pim register-source INT_NAME
no ip pim register-source
default ip pim register-source
```

Parameters

- ***INT_NAME*** Interface type and number. Values include:
  - **ethernet** *e_num* Ethernet interface specified by *e_num*.
  - **loopback** *l_num* Loopback interface specified by *l_num*.
  - **management** *m_num* Management interface specified by *m_num*.
  - **port-channel** *p_num* Port channel interface specified by *p_num*.
  - **vlan** *v_num* VLAN interface specified by *v_num*.

Example

- This command programs the switch to fill the source field of outbound PIM SM register packets with the IP address of loopback interface 2.

```
switch(config)#ip pim register-source loopback 2
switch(config)#
```

ip pim rp-address

The **ip pim rp-address** command configures the address of a Protocol Independent Multicast (PIM) static rendezvous point (RP) for a specified multicast subnet. If the command does not specify a subnet, the static RP maps to all multicast groups (224/4). Dynamic RPs override static RPs unless the static RP is given priority by using the ***override*** option of this command.

Multicast groups use RPs to connect sources and receivers. A PIM domain requires that all routers have consistently configured RP addresses.

The switch uses multiple **ip pim rp-address** commands to configure multiple RPs or to assign multiple subnets to an RP. When the address of a multicast group falls within multicast subnets configured by multiple **ip pim rp-address** commands, the group's RP address is selected by comparing the commands' multicast subnet size.

- Different size subnets: group uses command with the largest subnet.
- Same size subnets: group uses command as determined by hash algorithm.

The **no ip pim rp-address** and **default ip pim rp-address** commands remove the corresponding **ip pim rp-address** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
ip pim rp-address rp_addr [MULTICAST_SUBNET] [HASHMASK_LENGTH] [BSR_OVERRIDE]
[PRIORITY_NUM]
no ip pim rp-address rp_addr [MULTICAST_SUBNET]
default ip pim rp-address rp_addr [MULTICAST_SUBNET]
```

Parameters

- *rp_addr* Rendezvous point IP address (dotted decimal notation).
- ***MULTICAST_SUBNET*** Multicast IP address space (CIDR or address-mask).
  - <no parameter> Default multicast group IP address of 224/4.
  - *gp_addr* Multicast group IP address (CIDR or address-mask).
  - **access-list** *acl_name* Standard access control list that specifies the multicast group address.
  - *acl_name* Standard access control list that specifies the multicast group address.
- ***HASHMASK_LENGTH*** Length (in bits) of the hash mask.
  - <no parameter> hash mask remains unchanged from previous setting.
  - **hashmask** <***0*** - ***32***> hash mask length (in bits). Default value is 30.
- ***BSR_OVERRIDE*** Configures priority relative to dynamic RPs selected by BSR.
  - <no parameter> Dynamic RPs have priority over specified RP.
  - **override** RP has priority over dynamic RPs.
- ***PRIORITY_NUM*** BSR election priority rating. Larger numbers denote higher priority. Default value is 64.
  - <no parameter> priority remains unchanged from previous setting.
  - **priority** <***0*** - ***255***> priority rating.

Example

- This command configures 10.17.255.2 as a static RP for all multicast groups.

```
switch(config)#ip pim rp-address 10.17.255.2
switch(config)#
```

ip pim rp-candidate

The **ip pim rp-candidate** command configures the switch as a candidate rendezvous point (C-RP). The BSR selects a multicast group's dynamic RP set from the list of C-RPs in the PIM domain. The command specifies the interface (used to derive the RP address), C-RP advertisement interval, and priority rating. The BSR selects the RP set by comparing C-RP priority ratings. The C-RP advertisement interval specifies the period between successive C-RP advertisement message transmissions to the BSR.

Running-config supports multiple multicast groups through multiple **ip pim rp-candidate** statements:

- All commands must specify the same interface. Issuing a command with an interface that differs from existing commands removes all existing commands from ***running-config***.
- ***Running-config*** stores the ***interval*** setting in a separate statement that applies to all **rp-candidate** statements. When a command specifies an interval that differs from the previously configured value, the new value replaces the old value and applies to all configured **rp-candidate** statements. The default ***interval*** value is 60 seconds.

The **no ip pim rp-candidate** and **default ip pim rp-candidate** commands remove **ip pim rp-candidate** from ***running-config*** for the specified group. When these commands do not specify a multicast group, all **rp-candidate** statements are removed from ***running-config***.

The **no ip pim rp-candidate interval** and **default ip pim rp-candidate interval** commands restore the interval setting to the default value of 60 seconds. The **no ip pim rp-candidate priority** and **default ip pim rp-candidate priority** commands restore the priority setting to the default value of 0.

Command Mode

Global Configuration

Command Syntax

The ***INTERFACE*** parameter is always listed first. All other parameters can be placed in any order.

```
ip pim rp-candidate INTERFACE [GROUP_ADDR][PRIORITY_NUM][INTERVAL_PERIOD]
no ip pim rp-candidate [INTERFACE] [GROUP_ADDR]
no ip pim rp-candidate [INTERFACE] interval
no ip pim rp-candidate [INTERFACE] priority
default ip pim rp-candidate [INTERFACE] [GROUP_ADDR]
default ip pim rp-candidate [INTERFACE] interval
default ip pim rp-candidate [INTERFACE] priority
```

Parameters

- ***INTERFACE*** Switch uses IP address of specified interface as its C-RP address. Options include:
  - **ethernet** *e_num* Ethernet interface specified by *e_num*.
  - **loopback** *l_num* Loopback interface specified by *l_num*.
  - **management** *m_num* Management interface specified by *m_num*.
  - **port-channel** *p_num* Port-Channel Interface specified by *p_num*.
  - **vlan** *v_num* VLAN interface specified by *v_num*.
  - **vxlan** *vx_num* VXLAN interface specified by *vx_num*.
- ***GROUP_ADDR*** address of multicast group for which candidate is configured. Options include:
  - <no parameter> default multicast group (224.0.0.0/4).
  - *net_addr* multicast IPv4 subnet address (CIDR or address mask).

- **access-list** *acl_name* standard access control list that specifies the multicast group address.

- ***PRIORITY_NUM*** RP selection priority rating. Smaller numbers denote higher priority.
  - <no parameter> priority rating is set to the default value of 0.
  - **priority** <***0*** - ***255***> priority rating.
- ***INTERVAL_NUM*** Period between consecutive RP-advertisement message transmissions (seconds). Value also applies to previously configured rp-candidate statements.
  - <no parameter> interval remains unchanged from previous setting.
  - **interval** <***10*** - ***16383***> transmission interval.

Example

- This command configures a switch as a candidate RP for the multicast group 235.1.1.0/24 with a priority of 48 and a RP advertisement interval of 45 seconds. The switch advertises the IP address assigned to VLAN 24 as its RP address.

```
switch(config)#ip pim rp-candidate vlan 24 235.1.1.0/24 priority 48 interval 45
switch(config)#
```

ip pim sparse-mode

The **ip pim sparse-mode** command enables PIM and IGMP (router mode) on the configuration mode interface.

Important! PIM sparse mode and multicast border router (MBR) must be mutually exclusive on an interface. If the interface is configured as an MBR, do not enable PIM sparse mode on the interface.

The **no ip pim sparse-mode**, **no ip pim, default ip pim sparse-mode**, and **default ip pim** commands restore the default PIM and IGMP (router mode) settings of ***disabled*** on the configuration mode interface by removing the **ip pim sparse-mode** statement from ***running-config***.

Command Mode

Interface-Ethernet Configuration
Interface-Port-Channel Configuration
Interface-VLAN Configuration

Command Syntax

```
ip pim sparse-mode
no ip pim
no ip pim sparse-mode
default ip pim
default ip pim sparse-mode
```

Example

- This command enables PIM sparse mode on VLAN 4 interface.

```
switch(config)#interface vlan 4
switch(config-if-Vl4)#ip pim sparse-mode
switch(config-if-Vl4)#
```

ip pim sparse-mode sg-expiry-timer

The **ip pim sparse-mode sg-expiry-timer** command configures the (S, G) expiry timer interval for PIM-SM (S, G) multicast routes. The command does not apply to (*, G) mroutes.

The **no ip pim sparse-mode sg-expiry-timer** and **default ip pim sparse-mode sg-expiry-timer** commands restore the default setting of 210 seconds by removing the **ip pim sparse-mode sg-expiry-timer** statement from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
ip pim sparse-mode sg-expiry-timer period
no ip pim sparse-mode sg-expiry-timer
default ip pim sparse-mode sg-expiry-timer
```

Parameters

- *period* expiry timer interval (seconds). Value ranges from 120 (two minutes) to 65535 (18 hours, 12 minutes, 15 seconds). Default is 210 (three minutes, 30 seconds).

Example

- This command configures 2 minutes 30 seconds as the (S,G) expiry timer interval.

```
switch(config)#ip pim sparse-mode sg-expiry-timer 150
switch(config)#
```

ip pim spt-threshold

The **ip pim spt-threshold** command determines when the switch joins the shortest path source tree (SPT) for all IPv4 multicast groups. Setting the SPT-threshold is valid for switches configured as PIM leaf routers.

- When ***running-config*** does not list this command, the switch joins the SPT immediately after receiving the first PIM packet from a new source. The switch joins the SPT by sending a PIM join message toward the source.
- When ***running-config*** lists this command with a value of infinity, the switch never joins the SPT.

The **ip pim spt-threshold group-list** command configures the spt-threshold action for IPv4 multicast groups that match a specified access control list (ACL).

The **no ip pim spt-threshold** and **default ip pim spt-threshold** commands restore the default value of 0 by removing the **ip pim spt-threshold infinity** command from running-config.

Command Mode

Global Configuration

Command Syntax

```
ip pim spt-threshold JOIN
no ip pim spt-threshold
default ip pim spt-threshold
```

Parameters

- ***JOIN*** specifies switch's use of the short path tree (SPT). Options include:
  - **0** The switch immediately joins the SPT. This is the default value.
  - **infinity** The switch never joins the SPT.

Examples

- This command configures the switch to never join the SPT.

```
switch(config)#ip pim spt-threshold infinity
switch(config)#
```

- These equivalent commands restore the default value by removing the **ip pim spt-threshold** statement from running-config.

```
switch(config)#ip pim spt-threshold 0
switch(config)#

switch(config)#no ip pim spt-threshold
switch(config)#
```

ip pim spt-threshold group-list

The **ip pim spt-threshold group-list** command configures the spt-threshold action for IPv4 multicast groups that match a specified access control list (ACL).

- When ***running-config*** does not list this command, the switch joins the shortest path tree (SPT) immediately after receiving the first PIM packet from a new source. The switch joins the SPT by sending PIM join message toward the source.
- When running-config lists this command with a value of infinity, the switch never joins the SPT.

The action for all groups that are not specified by an ACL is configured with the global **ip pim spt-threshold** command.

The **no ip pim spt-threshold** and **default ip pim spt-threshold** commands remove the corresponding **ip pim spt-threshold group-list** command from ***running-config***. All groups specified by ACLs removed by this command revert to using the global **ip pim spt-threshold** command unless covered by another configured group-list command.

Command Mode

Global Configuration

Command Syntax

```
ip pim spt-threshold JOIN group-list acl_name
no ip pim spt-threshold JOIN group-list acl_name
default ip pim spt-threshold JOIN group-list acl_name
```

Parameters

- ***JOIN*** specifies switch's use of the short path tree (SPT) for specified groups. Options include:
  - **0** The switch immediately joins the SPT. This is the default value.
  - **infinity** The switch never joins the SPT.
- *acl_name* name of access control list.

Examples

- This command configures the switch to never join the SPT except for multcast groups matched by the ACL group-1.

```
switch(config)#ip pim spt-threshold infinity
switch(config)#ip pim spt-threshold 0 group-list group-1
switch(config)#
```

ip pim ssm range

The **ip pim ssm range** command defines the source specific multicast (SSM) range of IP multicast addresses.

SSM is a multicast packet delivery method where only packets originating from a specific source address requested by a receiver are routed to that receiver. SSM explicitly excludes the use of (*,G) join for applicable multicast groups. Source-specific multicast differs from any-source multicast (ASM), where a receiver expresses interest in traffic to a multicast address, then receives traffic from all multicast sources sending to that address.

The **no ip pim ssm range** and **default ip pim ssm range** commands remove the SSM IP multicast address range by deleting the **ip pim ssm range** statement from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
ip pim ssm range [ACCESS_RANGE]
no ip pim ssm range
default ip pim ssm range
```

Parameters

- ***ACCESS_RANGE*** specifies the SSM IP multicast address range. Options include:
  - *acl_name* sets the SSM range to address set specified by the standard ACL.
  - **standard** sets the SSM range to 232/8.

Examples

- This command configures the SSM address range to 232/8.

```
switch(config)#ip pim ssm range standard
switch(config)#
```

- These commands configure the SSM address range to those permitted by the LIST_1 standard ACL. The ACL permits the subnet address range 233.0.0.0/24.

```
switch(config)#ip access-list standard LIST_1
switch(config-std-acl-LIST_1)#permit 233.0.0.0/24
switch(config-std-acl-LIST_1)#exit
switch(config)#ip pim ssm range LIST_1
switch(config)#
```

show ip pim bsr

The **show ip pim bsr** command displays the switch's bootstrap router (BSR) information.

Command Mode

EXEC

Command Syntax

show ip pim bsr [***GROUP_FILTER***]

Parameters

- ***GROUP_FILTER*** specifies groups for which command displays information.
  - <no parameter> Displays data for all groups.
  - *net_addr* Displays message for specified group address. (CIDR or address mask).

Example

- This command configures the switch's BSR information.

```
switch>show ip pim bsr
PIMv2 Bootstrap information
This system is the Bootstrap Router (BSR)
  BSR address: 10.1.1.1
  Uptime:      00:14:42, BSR Priority: 0, Hash mask length: 30
  Next bootstrap message in 00:00:05
```

show ip pim config-sanity

The **show ip pim config-sanity** command displays diagnostic information about the switch's PIM configuration.

Command Mode

EXEC

Command Syntax

show ip pim config-sanity

Example

- This command displays PIM configuration diagnostic information.

```
switch>show ip pim config-sanity
DISCLAIMER: Below are only hints of potential PIM misconfiguration.
They do not necessary imply that there is a real problem.

The interfaces with PIM which are down: Vl4

switch>
```

show ip pim interface

The **show ip pim interface** command displays information about interfaces configured for PIM.

Command Mode

EXEC

Command Syntax

show ip pim interface [*INT_NAME*] [*INFO_LEVEL*]

Parameters

- ***INT_NAME*** Interface type and number. Values include
  - <no parameter> displays information for all interfaces.
  - **ethernet** *e_num* Ethernet interface specified by *e_num*.
  - **port-channel** *p_num* Port-Channel Interface specified by *p_num*.
  - **vlan** *v_num* VLAN interface specified by *v_num*.
  - **vxlan** *vx_num* VXLAN interface specified by *vx_num*.
- ***INFO_LEVEL*** specifies level of information detail provided by the command.
  - <no parameter> table of basic configuration information.
  - **detail** list of complete configuration information.

Examples

- This command displays information about all interfaces on which PIM is enabled.

```
switch>show ip pim interface
Address      Interface      Mode      Neighbor   Hello DR   DR Address    PktsQed
PktsDropped
                                        Count      Intvl Pri
10.17.254.30   Vlan3910       sparse    1        30   1    10.17.254.30   0
0
10.17.254.162  Vlan3925       sparse    2        30   1    10.17.254.163  0
0
10.17.254.106  Vlan3912       sparse    1        30   1    10.17.254.106  0
0
10.17.254.137  Ethernet12     sparse    1        30   1    10.17.254.138  0
0
switch>
```

- This command displays detailed PIM information for VLAN 26 interface.

```
switch>show ip pim interface vlan 26 detail
Interface address is 172.17.26.1
Vif number is 1
PIM: enabled
  PIM version: 2, mode: sparse
  PIM DR: 172.17.26.1 (this system)
  PIM DR Priority: 1
  PIM neighbor count: 0
  PIM Hello Interval: 30 seconds
  PIM Hello Priority: 1
  PIM Hello Lan Delay: 500 milliseconds
  PIM Hello Override Interval: 2500 milliseconds
  PIM Hello Lan Prune Delay in use
  PIM Hello Generation ID: 0x4a05aa0
  PIM Hello Generation ID is not required
  PIM Triggered Hello Delay: 5 seconds
  PIM Join-Prune Interval: 60 seconds
  PIM State-Refresh processing: disabled
  PIM State-Refresh Interval: unknown seconds
  PIM Graft Retry Interval: unknown seconds
  PIM domain border: disabled
switch>
```

show ip pim neighbor

The **show ip pim neighbor** command displays information about Protocol Independent Multicast (PIM) neighbors discovered by hello messages.

Command Mode

EXEC

Command Syntax

show ip pim neighbor [*INT_NAME*] [*BFD_DATA*]

Parameters

- ***INT_NAME*** Interface type and number. Values include
  - <no parameter> displays information for all interfaces.
  - **ethernet** *e_num* Ethernet interface specified by *e_num*.
  - **loopback** *l_num* Loopback interface specified by *l_num*.
  - **management** *m_num* Management interface specified by *m_num*.
  - **port-channel** *p_num* Port-Channel Interface specified by *p_num*.
  - **vlan** *v_num* VLAN interface specified by *v_num*.
  - **vxlan** *vx_num* VXLAN interface specified by *vx_num*.
- ***BFD_DATA*** Specifies inclusion of BFD data.
  - <no parameter> BFD data is not displayed.
  - **bfd** BFD data is displayed.

Example

- This command displays information about neighbor PIM routers.

```
switch>show ip pim neighbor
PIM Neighbor Table
Neighbor Address   Interface       Uptime     Expires    Mode
10.17.255.2      Vlan2028        21d22h     00:01:31   sparse

switch>
```

- This command displays information about neighbor PIM routers and the status of BFD.

```
switch>show ip pim neighbor bfd
PIM Neighbor Table
Flags: U - BFD is enabled and is UP
       I - BFD is enabled and is INIT
       D - BFD is enabled and is DOWN
       N - Not running BFD

Neighbor Address   Interface       Uptime     Expires    Mode     Flags
10.17.255.2      Vlan2028        21d22h     00:01:31   sparse    U

switch>
```

show ip pim protocol counters

The **show ip pim protocol** command displays statistics about Protocol Independent Multicast (PIM) control messages sent and received by the switch.

Command Mode

EXEC

Command Syntax

show ip pim protocol counters [***INT_NAME***]

Parameters

- ***INT_NAME*** Interface type and number. Values include
  - <no parameter> displays information for all interfaces.
  - **ethernet** *e_num* Ethernet interface specified by *e_num*.
  - **loopback** *l_num* Loopback interface specified by *l_num*.
  - **management** *m_num* Management interface specified by *m_num*.
  - **port-channel** *p_num* Port-Channel Interface specified by *p_num*.
  - **vlan** *v_num* VLAN interface specified by *v_num*.
  - **vxlan** *vx_num* VXLAN interface specified by *vx_num*.

Example

- This command displays statistics about inbound and outbound PIM control messages.

```
switch>show ip pim protocol counters
PIM Control Counters
                          Received         Sent      Invalid
Assert                           0           37            0
Bootstrap Router                 0            0            0
CRP Advertisement                0            0            0
Graft                            0            0            0
Graft Ack                        0            0            0
Hello                        63168       126355            0
J/P                         275714       143958            0
Join                             0            0            0
Prune                            0            0            0
Register                         0        13643            0
Register Stop                11839            0            0
State Refresh                    0            0            0

switch>
```

show ip pim register-source

The **show ip pim register-source** command displays the name of the interface from where the switch derives the IP address that it uses to fill the source field in all outbound PIM SM register packets. The **ip pim register-source** command specifies this interface.

By default, the source field is filled with the IP address from the interface associated with the best route to the RP. The **show ip pim register-source** command does not return a value when the source field is filled with the default value.

Command Mode

EXEC

Command Syntax

```
show ip pim register-source
```

Example

- This command displays the register-source interface.

```
switch>show ip pim register-source
Ethernet22
switch>
```

show ip pim rp

The **show ip pim rp** command displays the status and multicast group of each cached rendezvous point (RP).

Command Mode

EXEC

Command Syntax

```
show ip pim rp
```

Example

- This command displays the cached RPs.

```
switch>show ip pim rp
show ip pim rp
The PIM RP Set
Group: 224.0.0.0/4
  RP: 10.1.2.3
    Uptime: 00:05:12, Expires: never, Priority: 1 Override: 1
```

show ip pim rp-candidate

The **show ip pim rp-candidate** command displays the rendezvous point (RP) that is used for a specified multicast group.

Command Mode

EXEC

Command Syntax

```
show ip pim rp-candidate
```

Example

- This command displays the switch's candidate-RP information.

```
switch>show ip pim rp-candidate
Candidate RP information
  Candidate RP Address: 10.0.12.2
  CRP Holdtime: 150 seconds
    Group 224.2.0.0/16 Priority 2
```

show ip pim rp-hash

The **show ip pim rp-hash** displays the group to RP-hash mappping for the specified group and the list of qualifying candidate RPs.

Command Mode

EXEC

Command Syntax

show ip pim rp-hash *ipv4_addr* **[*INFO_LEVEL*]**

Parameters

- *ipv4_addr* multicast group IPv4 address.
- ***INFO_LEVEL*** specifies level of information detail provided by the command.
  - <no parameter> RP-hash map and list of candidate RPs.
  - **detail** includes data about the selected RP.

Example

- This command displays the RP that the switch uses for multicast group 224.1.0.0.

```
switch>show ip pim rp-hash 224.1.0.0
RP 10.1.2.3
```

show ip pim upstream joins

The **show ip pim upstream joins** command displays the join messages that the switch is scheduled to send.

Command Mode

EXEC

Command Syntax

show ip pim upstream joins [*JOIN_ADDRESSES*] [*NEIGHBOR_FILTER*]

Parameters

- ***JOIN_ADDRESSES*** Filters messages by source and group addresses.
  - <no parameter> displays all join messages.
  - *source_addr* displays all join messages for specified source group IPv4 address.
  - *group_addr* displays all join messages for specified multicast IPv4 address.
  - *source_addr group_addr* displays join message with specified source and group addresses.
  - *group_addr source_addr* displays join message with specified group and source addresses.

    group_addr must be a valid multicast IPv4 address.
- ***NEIGHBOR_FILTER*** specifies neighbors for which command provides data.
  - <no parameter> Displays messages for all neighbors.
  - **neighbor** *neighbor_addr* Displays message for specified neighbor address.

Example

- This command displays the list of join messages the switch is scheduled to send. The example only displays the first two messages.

```
switch>show ip pim upstream joins

------------- show ip pim upstream joins -------------

Neighbor address: 10.1.1.1
Via interface: 10.1.1.2
Next message in 1 seconds
  Group: 10.10.10.3
    Joins:
      10.25.1.1/32 SPT
    Prunes:
      No prunes included
Neighbor address: 10.1.1.6
Via interface: 10.1.1.5
Next message in 1 seconds
  Group: 10.14.1.69
    Joins:
      10.105.14.3/32 SPT
    Prunes:
      No prunes included
switch>
```

Chapter 37

Multicast Source Discovery Protocol

Multicast Source Discovery Protocol (MSDP) describes a topology that connects multiple IPv4 Protocol Independent Multicast Sparse-Mode (PIM-SM) domains. Each PIM-SM domain uses its independent Rendezvous Point (RP) without depending on RPs in other domains.

These sections describe the Arista MSDP implementation.

- Section 37.1: MSDP Introduction is an overview and lists features supported by Arista switches.
- Section 37.2: MSDP Description describes the MSDP protocol.
- Section 37.3: MSDP Configuration describes configuration tasks that implement MSDP.
- Section 37.4: MSDP Commands contains MSDP command descriptions.

37.1 MSDP Introduction

Arista switches support these MSDP features:

- Basic MSDP speaker functions.
- MSDP peer configuration: description, connect-source interface, keepalive time, and hold time.
- ACL filtering of inbound and outbound Source-Active (SA) messages.
- Mesh groups.
- Display of peer status.
- Display of filtered SA messages received from MSDP peers.

These MSDP features are not supported:

- MSDP is not supported with Anycast-RP (RFC4610).
- IP packet encapsulation.

37.2 MSDP Description

The Multicast Source Discovery Protocol (MSDP) defines a topology connecting Protocol Independent Multicast sparse mode (PIM-SM) domains. MSDP provides interdomain access to multicast sources in all domains by enabling all rendezvous points (RPs) to discover multicast sources outside of their domains. RPs also use MSDP to announce sources that are sending to a multicast group.

37.2.1 MSDP Speakers

An MSDP speaker is a router in a PIM-SM domain that has MSDP peering sessions with MSDP peers in other domains. An MSDP peering session is a TCP connection through which peers exchange MSDP control information. An MSDP peer is a router that is connected to the speaker though a peering session.

PIM uses MSDP to register a local source with remote domain RPs through Source Active (SA) messages, which originate at the local domain's RP. Receivers in remote PIM-SM domains depend only on RPs in their domains to learn of multicast data sources in other domains. Multicast data is subsequently delivered from a source to receivers in different domains through a PIM-SM source tree.

Section 37.3.1: MSDP Speaker Configuration describes the process of configuring MSDP speakers.

37.2.2 Network Configuration

The TCP connections between RPs are defined either through an underlying unicast routing table or by configuring a default MSDP peer. A typical MSDP configuration utilizes a BGP specified routing table. SA messages are MSDP control messages that peers exchange during peering sessions.

37.2.2.1 Source Active Messages

A Source Active (SA) message is a message that an RP creates and sends to MSDP peers when it learns of a new multicast source through a PIM register message. RPs that intend to originate or receive SA messages must establish MSDP peering with other RPs, either directly or through intermediate MSDP peers. An RP that is not a DR on a shared network should only originate SAs in response to register messages it receives from the DR. It does not originate SA's for directly connected sources in its domain.

SA messages contain the following fields:

- Source address of the data source.
- Group address that receives data sent by the source.
- IP address of the RP.

Th SA Cache is the repository of SA messages received by the MSDP speaker. The switch always stores received SA messages. Section 37.3.4: Managing the SA Cache describes procedures that limit the size of the SA cache and options for displaying the cache.

37.2.2.2 Reverse Path Forwarding

Reverse path forwarding (RPF) is a multicast packet transport technique that ensures loop-free packet forwarding by using a router's unicast routing table. Traffic forwarding is based on source addresses instead of destination addresses. RPF is implemented as defined in RFC 3618.

Packet forwarding is based on the packet's unicast reverse path. An RPF router prevents network loops by only forwarding a packet when it enters through the interface holding its source routing entry.

When a multicast packet enters a router's interface, the router checks the reverse path of the packet by examining the list of networks that are reachable through the input interface. If the list contains a matching routing entry for the multicast packet's source IP address, the packet is forwarded to all other interfaces that are participants in the multicast group. Otherwise, the packet is dropped.

RPF requires that the unicast routing table is correct and converged. It also assumes that the use of symmetric forward and reverse paths between router and sender. RPF fails on uni-directional links.

Section 37.3.3.1: Displaying RPF Peers describes commands that display RPF peers.

37.2.2.3 Default MSDP Peers

The default peer is the MSDP peer from which the MSDP speaker accepts SA messages. If there is only one MSDP peer, all of its SA messages will be accepted. When multiple default peers are configured the switch uses the first default peer to appear in ***running-config***. Default MSDP peers invalidate the use of RPF over unicast routing tables.

Each default peer may be associated with a prefix list. The prefix list specifies the RPs from where the speaker accepts SA messages. When ***running-config*** contains multiple default peers with prefix lists, an SA is accepted from the first default peer in ***running-config*** whose prefix list contains the RP in the SA. The speaker accepts all remaining SAs from the first default peer that is not associated with a prefix list.

Section 37.3.3.2: Configuring the Default Peer describes commands that configure default peers.

37.2.3 MSDP Exchange Processes

37.2.3.1 Control Information Exchange

An RP originates an SA message when a source registers with the RP to send data to a multicast group. RPs periodically originate SA messages while its registered sources send data to maintain messages in SA caches of its MSDP peers. RPs that have no registered sources periodically send keepalive messages to maintain TCP connections with its peers.

MSDP defines the following timers that specify the transmission frequency of control messages:

- SA Advertisement Time: Duration of SA Advertisement intervals. An RP sends periodic SA messages to reference each registered source once per interval. SA advertisement time is 60 seconds.
- Keepalive Time: Period between the transmission of consecutive keepalive messages. Default keepalive time is 60 seconds. Minimum keepalive time is one second.
- Hold Timer: Period an MSDP speaker maintains a peer TCP connection after receiving an SA or keepalive message from the peer. Default time is 75 seconds. Minimum hold time is three seconds.

37.2.3.2 MSDP Data Exchange

This sequence describes the exchange of multicast data across PIM domains through MSDP:

Step 1 When a source's first data packet is registered by the first hop router, the RP extracts the data from the packet and forwards it down the shared tree in the PIM domain.

Step 2 The RP informs MSDP peers of the new source by sending a Source-Active (SA) message that identifies the source, the recipient group, and the RP's address or originator ID.

Step 3 Upon receiving the SA message, an MSDP peer which is the RP for a multicast tree that includes members interested in the the multicast sends a PIM join message (S,G) toward the data source.

Step 4 The PIM designated router (DR) sends subsequent data encapsulated in PIM register messages directly to the remote domain's RP when the source becomes active.

Step 5 If the source times out, this process repeats when the source goes active again.

37.3 MSDP Configuration

These sections describe the configuration of the switch as an MSDP speaker and the establishment of MSDP peering sessions.

- Section 37.3.1: MSDP Speaker Configuration
- Section 37.3.2: Establishing MSDP Peers
- Section 37.3.3: MSDP Network Configuration
- Section 37.3.4: Managing the SA Cache

MSDP requires that TCP port 639 (MSDP) is open on the control plane. The default control-plane ACL includes a permit rule that allows TCP packets access through the MSDP port.

37.3.1 MSDP Speaker Configuration

The switch is configured as an MSDP speaker when MSDP is enabled. MSDP is enabled by configuring an MSDP peer. Section 37.3.2.1 describes the process of configuring an MSDP peer.

Source-Address (SA) messages that an MSDP speaker originates contain the speaker's rendezvous point (RP) address, as configured through PIM statements and processes. MSDP provides a method of assigning an originator ID address, which the speaker uses in place of its RP address when advertising SA messages. The **ip msdp originator-id** command configures the switch to set the RP address to the specified interface's IP address in SA messages that it originates as an MSDP speaker.

Only RPs originate SA messages and only for its registered sources. RPs do not originate periodic SA messages for sources in other PIM domains. MSDP speakers that are not RPs do not originate periodic SA messages. Intermediate MSDP speakers forward SA messages received from other domains. Intermediate speakers are not required to be RPs.

Example

- This command configures the switch to use the IP address assigned to loopback interface 100 as the RP address in SA messages that it originates.

```
switch(config)#ip msdp originator-id loopback 100
switch(config)#
```

37.3.2 Establishing MSDP Peers

These sections describe MSDP Peer configuration tasks.

- Section 37.3.2.1: Configuring an MSDP Peer
- Section 37.3.2.2: Mesh Groups
- Section 37.3.2.3: Filtering SA Messages
- Section 37.3.2.4: Keepalive, Hold Time, and Reset Time Configuration
- Section 37.3.2.5: Displaying Peer Information

37.3.2.1 Configuring an MSDP Peer

The switch attempts to establish MSDP peering sessions through IP addresses configured as MSDP peers. The **ip msdp peer** command configures a specified address as an MSDP peer and enables the switch as an MSDP speaker if no other peers are configured. The peering session with the device at the specified network is established over a TCP connection. The command can specify an interface through which the switch establishes the TCP session. When the command does not specify an interface, the connection is established through an interface determined by existing routing algorithms.

To display MSDP peer information, enter **show ip msdp peer.**

Example

- These commands assign an IP address to loopback interface 100, then configure 10.4.4.12 as an MSDP peer and establishes the TCP peer session through the loopback.

```
switch(config)#interface loopback 100
switch(config-if-Lo100)#ip address 10.6.8.6/24
switch(config-if-Lo100)#exit
switch(config)#ip msdp peer 10.4.4.12 connect-source loopback 100
switch(config)#show ip msdp peer
MSDP Peer 10.4.4.12
Connection status:
   State: Connect
   Connection Source: Loopback100 ( 10.6.8.6 )

switch(config)#
```

To associate descriptive text with the specified MSDP peer, enter **ip msdp description**.

Example

- These commands associate the string NORTH with the MSDP peer located at 10.4.4.12.

```
switch(config)#ip msdp description 10.4.4.12 NORTH
switch(config)#show ip msdp peer
MSDP Peer 10.4.4.12
Description: NORTH
Connection status:
   State: Connect
   Connection Source: Loopback100 ( 10.6.8.6 )

switch(config)#
```

To close the peering session with the specified MSDP peer, enter **ip msdp shutdown**. This terminates the TCP connection between the switch and the peer. The peer remains configured and the peer session can be resumed by removing the **ip msdp shutdown** command from ***running-config***.

Examples

- This command closes the peering session with the MSDP peer at 10.4.4.12.

```
switch(config)#ip msdp shutdown 10.4.4.12
switch(config)#show ip msdp peer
MSDP Peer 10.4.4.12
Description: NORTH
Connection status:
   State: Disabled
   Connection Source: Loopback100 ( 10.6.8.6 )

switch(config)#
```

- This command reopens the peering seesion with the peer at 10.4.4.12.

```
switch(config)#no ip msdp shutdown 10.4.4.12
switch(config)#show ip msdp peer
MSDP Peer 10.4.4.12
Description: NORTH
Connection status:
   State: Connect
   Connection Source: Loopback100 ( 10.6.8.6 )

switch(config)#
```

37.3.2.2 Mesh Groups

Each node in a fully meshed network is directly connected to every other node in the network. Each peer in a fully meshed MSDP speaker network can be configured as a member of a mesh group. SA messages received from a mesh group peer are not forwarded to other members of the mesh group.

To configure the specified MSDP peer connection as an MSDP mesh group member, enter **ip msdp mesh-group**. An MSDP peer can be assigned to multiple mesh groups. Multiple peer connections can be assigned to the same mesh group.

To display the mesh group membership of configured MSDP peers, enter **show ip msdp mesh-group**.

Example

- These commands configure the MSDP peer connection to 10.1.1.14 as a member of the AREA-1 mesh group, then displays members of mesh groups to which configured MSDP peers belong.

```
switch(config)#ip msdp mesh-group AREA-1 10.1.1.14
switch(config)#show ip msdp mesh-group
Mesh Group: AREA-1
        10.1.1.14
Mesh Group: tier_01
        10.24.18.13
Mesh Group: tier_02
        10.26.101.18
switch(config)#
```

37.3.2.3 Filtering SA Messages

The switch can filter Source-Active (SA) messages that it sends and receives with access control lists (ACLs). The commands accept standard and extended ACLs. The address field in standard ACLs filter an SA message on its group address.

The **ip msdp sa-filter in** command assigns an ACL to filter inbound SA messages from a specified MSDP peer connection. The switch only accepts SA messages from the peer that pass the ACL. The switch accepts all SA messages from peers that are not assigned an input ACL. A peer can be assigned only one input filter ACL. Subsequent **ip msdp sa-filter in** commands for a peer replace the existing command.

The **ip msdp sa-filter out** command assigns an ACL as a filter for outbound SA messages to a specified MSDP peer connection. The switch only sends SA messages to the peer that pass the ACL. The switch sends all specified SA messages to peers not assigned an output filter ACL. A peer can be assigned only one output ACL. Subsequent **ip msdp sa-filter out** commands for a peer replace the existing command.

Example

- These commands assign the IP ACLs named LIST-IN as the inbound SA message filter and LIST-OUT as the outbound SA message filter for the MSDP peer connection to 10.4.4.12.

```
switch(config)#ip msdp sa-filter in 10.4.4.12 list LIST-IN
switch(config)#ip msdp sa-filter out 10.4.4.12 list LIST-OUT
switch(config)#show ip msdp peer
MSDP Peer 10.4.4.12
Connection status:
   State: Listen
   Connection Source: Loopback100 ( 10.6.8.6 )
SA Filtering:
Input Filter: LIST-IN
Output Filter: LIST-OUT

switch(config)#
```

37.3.2.4 Keepalive, Hold Time, and Reset Time Configuration

To configure the MSDP keepalive and hold time intervals for a specified MSDP peer connection, enter **ip msdp keepalive**.

- Keepalive time interval is the period between the transmission of consecutive keepalive messages. The default keepalive time interval is 60 seconds.
- Hold time interval is the period the switch waits for a KEEPALIVE or UPDATE message before it disables peering. The default hold time interval is 75 seconds.

The hold time interval must be longer than or equal to the keepalive time interval.

Example

- This command sets the keepalive time to 45 seconds and the hold time to 80 seconds for the MSDP peer connection to 10.4.4.12.

```
switch(config)#ip msdp keepalive 10.4.4.12 45 80
switch(config)#
```

To specify the period that the switch waits after an MSDP peering session is reset before attempting to reestablish the session, enter **ip msdp timer**. The default period is 30 seconds.

Example

- This command configures the switch to wait 45 seconds after an MSDP peering session is reset before attempting to reestablish the session.

```
switch(config)#ip msdp timer 45
switch(config)#
```

37.3.2.5 Displaying Peer Information

To display the MSDP peers, enter **show ip msdp summary**. The command also displays the operational status of each peer and the number of messages from the peers in the SA cache.

Example

- This command displays the configured peers, the status of the peers, and the number of SA messages received from those peers.

```
switch>show ip msdp summary
MSDP Peer Status Summary
Peer Address      State   SA Count
192.168.3.18      Up      0
192.168.3.16      Up      0
192.168.3.37      Listen  0
192.168.3.46      Up      0
192.168.3.47      Up      0
switch>
```

37.3.3 MSDP Network Configuration

37.3.3.1 Displaying RPF Peers

The switch uses the unicast routing table to define TCP connections between RPs by selecting the next hop peer toward the originating RP of an SA message as the reverse path forwarding (RPF) peer. The switch forwards SA messages that it receives from the RPF peer to all other MSDP peers. The switch rejects SA messages that it receives from non-RPF peers.

To display MSDP information for the peer from which the switch accepts SA messages for a specified rendezvous point (RP), enter **show ip msdp rpf-peer**.

Example

- This command displays MSDP information for the peer from which the switch accepts SA messages for the RP at 10.5.29.4.

```
switch>show ip msdp rpf-peer 10.5.29.4
Rpf Peer is 10.5.29.4 for RP 10.5.29.4
switch>
```

37.3.3.2 Configuring the Default Peer

The default peer is the MSDP peer from which the MSDP speaker is configured to accept all SA messages. A default peer may be associated with a prefix list. The prefix list specifies the RPs from where the speaker accepts SA messages.

The switch can designate multiple default peers:

- Switch defines one peer: A default peer statement is not required; the switch accepts SA traffic from the configured peer.
- Switch defines one default peer (no prefix list): The switch accepts all SA messages from only the default peer.
- Switch defines multiple default peers (no prefix lists): The switch accepts all SA messages from only the first default peer listed in ***running-config***. Other listed default peers take effect only if the peer named in the first default-peer statement is not accessible.
- First default-peer statement includes a prefix list: The switch accepts all SA messages from the default peer whose originating RP is covered in the prefix list. The disposition of SA messages originating from other RPs is determined by subsequent **ip msdp default-peer** statements.

To configure the specified MSDP peer connection as a default peer on the switch, enter **ip msdp default-peer**. The default peer address must be a previously configured MSDP peer (**ip msdp peer**).

Example

- These commands configure an MSDP peer.

```
switch(config)#ip msdp peer 10.5.2.2
switch(config)#ip msdp default-peer 10.5.2.2
switch(config)#
```

37.3.4 Managing the SA Cache

The switch stores Source Active (SA) messages after forwarding the information. This allows new group members to learn about the source before the next SA message is received. The caching action is not configurable and cannot be disabled. The **ip msdp cache-sa-state** command is included to maintain compatibility with other devices. The command has no effect on switch operations.

SA messages have an expiration period of 90 seconds and remain in the SA cache until they expire. A peer's SA limit defines the number of SA messages the switch stores from the peer. The switch does not store SA messages from a peer whose SA limit is reached until its cached messages start expiring.

37.3.4.1 Limiting SA Cache Contents

To configure the maximum number of SA messages from a specified MSDP peer that the switch stores in the SA cache, enter **ip msdp sa-limit**. The default limit of SA messages that the switch can store from a specified peer is 40000.

Example

- This command sets the SA limit of 500 for the MSDP peer at 10.1.1.5.

```
switch(config)#ip msdp sa-limit 10.1.1.5 500
switch(config)#
```

The maximum number of SA messages that the switch can store in the SA cache for a specified multicast group address is configured by the **ip msdp group-limit** command. The default limit of SA messages that the switch can store from a specified group is 40000.

Example

- This command sets the maximum number of 1000 SAs for multicast group 225.13.15.8/29

```
switch(config)#ip msdp group-limit 1000 source 225.13.15.8/29
switch(config)#
```

The maximum number of rejected SA messages that the switch can store in the SA cache is configured by the **ip msdp rejected-limit** command. The default limit of rejected SA messages that the switch can store is 40000.

Example

- This command sets 5000 as the maximum number of rejected SAs that the SA cache can contain.

```
switch(config)#ip msdp rejected-limit 5000
switch(config)#
```

Contents of the SA message cache are removed by the **clear ip msdp sa-cache** command. The command provides options for removing all cache contents or only contents of a specific multicast group.

Example

- This command deletes all SA message cache contents.

```
switch#clear ip msdp sa-cache
switch#
```

37.3.4.2 Displaying SA Cache Contents

SA message cache contents are displayed by the **show ip msdp sa-cache** command. Filter options provided by the command for displaying partial cache contents include:

- multicast group address: multicast group
- source address and group address

The command can also display unexpired SAs rejected by ACL filters or cache limit exceeded conditions.

Example

- This command displays the contents of the SA message cache.

```
switch>show ip msdp sa-cache
MSDP Source Active Cache
(10.61.71.29, 234.1.4.2), RP 10.5.29.4, heard from 10.5.29.4
(10.51.71.23, 234.1.4.1), RP 10.5.29.4, heard from 10.5.29.4
(10.53.71.27, 234.1.4.2), RP 10.3.25.4, heard from 10.3.25.4
(10.10.101.24, 234.1.4.1), RP 10.2.44.4, heard from 10.2.44.4
(10.10.151.22, 234.1.4.1), RP 10.1.12.4, heard from 10.1.12.4
switch>
```

Information about specified MSDP peers, including SAs accepted from the peer is displayed by the **show ip msdp peer** command.

Example

- This command displays data for the peer at 10.2.42.4, including SAs accepted from the peer.

```
switch>show ip msdp peer 10.2.42.4 accepted-sas
MSDP Peer 10.2.42.4
Connection status:
  State: Up
  Connection Source: Loopback4 ( 10.2.43.4 )
SA Filtering:
Input Filter: allow-multicast-for-msdp
Output Filter: allow-multicast-for-msdp
SAs accepted:
(10.62.79.30, 234.1.4.2), RP 10.2.42.4
(10.61.79.29, 234.1.4.1), RP 10.2.42.4
(10.62.79.30, 234.1.4.1), RP 10.2.42.4
switch>
```

The SA cache for the local PIM domain is displayed by the **show ip msdp pim sa-cache** command.

Example

- This command displays the SA cache for the local PIM domain.

```
switch>show ip msdp pim sa-cache
MSDP Source Active Messages for local Pim RP
(10.51.71.23, 234.1.4.1), RP 10.2.43.4
(10.20.91.26, 234.1.4.1), RP 10.2.43.4
(10.20.91.26, 234.1.4.2), RP 10.2.43.4
(10.20.91.24, 234.1.4.1), RP 10.2.43.4
switch>
```

37.3.4.3 Verifying Consistency Between the SA Cache and the Routing Table

To check the consistency between the multicast routing table and the MSDP Source-Address (SA) caches, enter **show ip msdp sanity.** When the command detects inconsistencies, it displays the cache entries that are not in the table.

Example

- This command displays a sanity check that detects inconsistencies between the SA cache and the multicast routing table.

```
switch>show ip msdp sanity
PIM SA cache entries not in the MRT
Msdp-learnt MRT entries not in the SA cache
SA cache entries not in the MRT
(192.168.3.8, 224.1.154.1)
(192.168.3.35, 224.1.167.1)
(192.168.3.16, 224.1.226.1)
(192.168.3.12, 224.1.182.1)
(192.168.3.33, 224.1.150.1)
May-Notify-MSDP entries not in the PIM SA cache
(need not be an error condition)
4.1), RP 10.2.42.4
switch>
```

37.4 MSDP Commands

MSDP Configuration Commands (Global)

- ip msdp cache-sa-state
- ip msdp default-peer
- ip msdp description
- ip msdp group-limit
- ip msdp keepalive
- ip msdp mesh-group
- ip msdp originator-id
- ip msdp peer
- ip msdp rejected-limit
- ip msdp sa-filter in
- ip msdp sa-filter out
- ip msdp sa-limit
- ip msdp shutdown
- ip msdp timer

MSDP SA Cache Commands

- clear ip msdp sa-cache

MSDP Display Commands

- show ip msdp mesh-group
- show ip msdp peer
- show ip msdp pim sa-cache
- show ip msdp rpf-peer
- show ip msdp sa-cache
- show ip msdp sanity
- show ip msdp summary

clear ip msdp sa-cache

The **clear ip msdp sa-cache** command removes contents of the Source-Active (SA) message cache. The command provides these filter options for removing partial cache contents:

- contents of a multicast group by specifying its group address
- all cache contents

Command Mode

Privileged EXEC

Command Syntax

clear ip msdp sa-cache [*ADDRESS_FILTER*]

Parameters

- ***ADDRESS_FILTER*** IPv4 address used to select table entries for removal.
  - <no parameter> All SA messages
  - *grp_addr* Multicast group address (IPv4 address).

    grp_addr must be a valid multicast address.

Example

- This command deletes all SA message cache contents.

```
switch#clear ip msdp sa-cache
switch#
```

ip msdp cache-sa-state

The switch stores Source Active (SA) messages after forwarding the information it contains to the next MSDP peer. This allows new group members to learn about the source before the next SA message is received. The caching action is not configurable and cannot be disabled.

The **ip msdp cache-sa-state** command is included to maintain compatibility with other devices. The command has no effect on switch operations.

Command Mode

Global Configuration

Command Syntax

```
ip msdp cache-sa-state
```

ip msdp default-peer

The **ip msdp default-peer** command configures the specified MSDP peer connection as a default peer on the switch. The default peer configuration defines the peers from which the switch accepts Source-Active (SA) messages. When the command includes a *prefix list* parameter, the specified peer is the default peer for only SA messages originating from rendezvous points (RPs) covered by prefix list entries. The default peer address must be a previously configured MSDP peer (**ip msdp peer**).

Default peers provide an alternative to reverse packet forwarding (RPF) typically used by MSDP to specify the peers from which a switch accepts SA messages. However, RPF requires a unicast routing table that is correct and converged. RPF also assumes symmetric forward and reverse paths between router and sender. RPF fails on uni-directional links. Default MSDP peers invalidate the use of RPF over unicast routing tables.

The switch can designate multiple default peers:

- Switch defines one peer: A default peer statement is not required; the switch accepts SA traffic from the configured peer.
- Switch defines one default peer (no prefix list): The switch accepts all SA messages from only the default peer.
- Switch defines multiple default peers (no prefix lists): The switch accepts all SA messages from only the first default peer listed in ***running-config***. Other listed default peers are used only when peers listed before them in ***running-config*** are not accessible.
- First default-peer statement includes a prefix list: The switch accepts all SA messages from the default peer whose originating RP is covered in the prefix list. The disposition of SA messages originating from other RPs is determined by subsequent **ip msdp default-peer** statements.

The **no ip msdp default-peer** and **default ip default-peer** commands remove the corresponding **ip msdp default-peer** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
ip msdp default-peer peer_id [PREFIX]
no ip msdp default-peer peer_id
default ip msdp default-peer peer_id
```

Parameters

- *peer_id* MSDP peer (IPv4 address).
- ***PREFIX*** List of RPs from the SA messages originate for whiich the default peer is valid.
  - <no parameter> default peer is valid for SAs from all originating RPs.
  - **prefix-list** *list_name* name of the prefix list that defines affected originating RP prefixes.

Example

- These commands configure two MSDP peers.

```
switch(config)#ip msdp peer 10.5.2.2
switch(config)#ip msdp peer 10.6.2.2
switch(config)#ip msdp default-peer 10.5.2.2
switch(config)#
```

ip msdp description

The **ip msdp description** command associates descriptive text with the specified MSDP peer.

The **no ip msdp description** and **default ip msdp description** commands remove the text association from the specified peer.

Command Mode

Global Configuration

Command Syntax

```
ip msdp peer_id description description_string
no ip msdp peer_id description
default ip msdp peer_id description
```

Parameters

- *peer_id* MSDP peer (IPv4 address).
- *description_string* text string that is associated with neighbor.

Example

- These commands associate the string NORTH with the MSDP peer located at 10.4.4.12.

```
switch(config)#ip msdp description 10.4.4.12 NORTH
switch(config)#show ip msdp peer
MSDP Peer 10.4.4.12
Description: NORTH
Connection status:
   State: Connect
   Connection Source: Loopback100 ( 10.6.8.6 )

switch(config)#
```

ip msdp group-limit

The **ip msdp group-limit** command specifies the maximum number of Source-Active (SA) messages that the switch allows in the SA cache for a specified multicast group address.

SA messages have an expiration period of 90 seconds and remain in the SA cache until they expire. The switch does not accept SA messages for a group whose cache limit is reached until its cached messages start expiring.

The **no ip msdp group-limit** and **default ip msdp group-limit** command removes the maximum group limit for the specified prefix by removing the corresponding **ip msdp group-limit** statement from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
ip msdp group-limit quantity source src_subnet
no ip msdp group-limit quantity source src_subnet
default ip msdp group-limit quantity source src_subnet
```

Parameters

- *quantity* maximum number of groups that can access the interface. Value ranges from 1 to 40000.
- *src_subnet* Source IPv4 subnet (CIDR or address-mask notation).

Example

- This command sets the maximum number of 1000 SAs for multicast group 10.13.15.8/29.

```
switch(config)#ip msdp group-limit 1000 source 10.13.15.8/29
switch(config)#
```

ip msdp keepalive

The **ip msdp keepalive** command configures the MSDP keepalive and hold time intervals for a specified MSDP peer connection.

- Keepalive time interval is the period between the transmission of consecutive keepalive messages. The default keepalive time interval is 60 seconds.
- Hold time interval is the period the switch waits for a KEEPALIVE or UPDATE message before it disables peering. The default hold time interval is 75 seconds.

The **no ip msdp keepalive** and **default ip msdp keepalive** commands restore the default keepalive and hold time intervals for the specified MSDP peer connection by removing the corresponding **ip msdp keepalive** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
ip msdp keepalive peer_id keep_alive hold_time
no ip msdp keepalive peer_id
default ip msdp keepalive peer_id
```

Parameters

- *peer_id* MSDP peer address (IPv4 address).
- *keep_alive* keepalive period (seconds). Value ranges from 1 to 65535. Default value is 60.
- *hold_time* hold time (seconds). Value ranges from 1 to 65535. Deafult value is 75.

Restrictions

The hold time interval must be longer than or equal to the keepalive time interval.

Example

- This command sets the keepalive time to 45 seconds and the hold time to 80 seconds for the connection with the MSDP peer at 10.4.4.12.

```
switch(config)#ip msdp keepalive 10.4.4.12 45 80
switch(config)#
```

ip msdp mesh-group

The **ip msdp mesh-group** command configures the specified MSDP peer connection as an MSDP mesh group member. A peer can be assigned to multiple mesh groups. Multiple MSDP peers can be assigned to a common mesh group.

An MSDP mesh group is a network of MSDP speakers where each speaker directly connects to every other speaker. The switch does not forward Source-Active (SA) messages that it receives from a mesh group peer to other peers of the same group.

The **no ip msdp mesh-group** and **default ip msdp mesh-group** commands delete the specified peer connection from a mesh group by removing the corresponding **ip msdp mesh-group** command from running-config. **C**ommands that do not include a specific MSDP peer delete all configured connections from the specified mesh group.

Command Mode

Global Configuration

Command Syntax

```
ip msdp mesh-group group_name peer_id
no ip msdp mesh-group group_name [peer_id]
default ip msdp mesh-group group_name [peer_id]
```

Parameters

- *group_name* name of mesh group.
- *peer_id* MSDP peer address (IPv4 address).

Related Commands

- **show ip msdp mesh-group** displays mesh group membership of MSDP peers.

Example

- These commands configure the MSDP peer connection to 10.1.1.14 a member of AREA-1 mesh group, then displays members of mesh groups to which configured MSDP peers belong.

```
switch(config)#ip msdp mesh-group AREA-1 10.1.1.14
switch(config)#show ip msdp mesh-group
Mesh Group: AREA-1
        10.1.1.14
switch(config)#
```

ip msdp originator-id

The **ip msdp originator-id** command configures an originator ID to replace the rendezvous point (RP) address in source-address (SA) messages that it originates as an MSDP speaker.

SA messages that an MSDP speaker originates contain the speaker's rendezvous point (RP) address, as configured through PIM statements and processes. An originator ID is an alternative IPv4 address that a speaker uses in place of its RP address when advertising SA messages. This command configures the switch to use the specified interface's IP address as the RP address in SA messages that it originates.

The **no ip msdp originator-id** and **default ip msdp originator-id** commands configure the switch to use its RP address in SA messages that it sends by removing the **ip msdp originator-id** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
ip msdp originator-id INTERFACE
no ip msdp originator-id INTERFACE
default ip msdp originator-id INTERFACE
```

Parameters

- ***INTERFACE*** Specifies the interface from which the IP address is derived. Options include:
  - **ethernet** *e_num* Ethernet interface.
  - **loopback** *l_num* Loopback interface.
  - **management** *m_num* Management interface.
  - **port-channel** *p_num* Port-Channel Interface.
  - **vlan** *v_num* VLAN interface.
  - **vxlan** *vx_num* VXLAN interface.

Example

- This command configures the switch to use the IP address assigned to loopback 100 as the RP address in SA messages that it originates.

```
switch(config)#ip msdp originator-id loopback 100
switch(config)#
```

ip msdp peer

The **ip msdp peer** command configures the specified address as an MSDP peer and enables MSDP on the switch if it was not previously enabled. The peering session with the device at the specified network is established over a TCP connection.

The command can specify an interface through which the TCP connection is established. When the command does not specify an interface, the connection is established through an interface determined by existing routing algorithms.

The **no ip msdp peer** and **default ip msdp peer** commands remove the specified MSDP peer configuration by deleting the corresponding **ip msdp peer** command from ***running-config***. MSDP is disabled when the last **ip msdp peer** command is removed.

Command Mode

Global Configuration

Command Syntax

```
ip msdp peer peer_id [CONNECTION]
no ip msdp peer peer_id
default ip msdp peer peer_id
```

Parameters

- *peer_id* MSDP peer address (IPv4 address).
- ***CONNECTION*** interface through which TCP session connects. Options include:
  - <no parameter> determined through previously configured protocol.
  - **connect-source ethernet** *e_num* Ethernet interface.
  - **connect-source loopback** *l_num* Loopback interface.
  - **connect-source management** *m_num* Management interface.
  - **connect-source port-channel** *p_num* Port-Channel Interface.
  - **connect-source vlan** *v_num* VLAN interface.
  - **connect-source vxlan** *vx_num* VXLAN interface.

Example

- These commands assign an IP address to loopback interface 100, then configure 10.4.4.12 as an MSDP peer and establishes the TCP peer session through the loopback.

```
switch(config)#interface loopback 100
switch(config-if-Lo100)#ip address 10.6.8.6/24
switch(config-if-Lo100)#exit
switch(config)#ip msdp peer 10.4.4.12 connect-source loopback 100
switch(config)#show ip msdp peer
MSDP Peer 10.4.4.12
Connection status:
   State: Connect
   Connection Source: Loopback100 ( 10.6.8.6 )

switch(config)#
```

ip msdp rejected-limit

The **ip msdp rejected-limit** command specifies the maximum number of rejected Source-Active messages that the switch allows in the SA cache.

SA messages have an expiration period of 90 seconds. They remain in the SA cache during this time. The default limit of rejected SA messages that the switch can store is 40000.

The **no ip msdp rejected-limit** and **default ip msdp rejected-limit** commands restore the rejected SA limit of 40000 by removing the **ip msdp rejected-limit** statement from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
ip msdp rejected-limit quantity
no ip msdp rejected-limit
default ip msdp rejected-limit
```

Parameters

- *quantity* maximum rejected SA messages the SA cache can store. Value ranges from 0 to 40000.

Example

- This command sets 5000 as the maximum number of rejected SAs that the SA cache can contain.

```
switch(config)#ip msdp rejected-limit 5000
switch(config)#
```

ip msdp sa-filter in

The **ip msdp sa-filter in** command assigns an IP access control list (ACL) as a filter for inbound Source-Active (SA) messages from the specified MSDP peer connection. The switch only accepts SA messages from the specified peer that are accepted by the assigned ACL. The switch accepts all SA messages from the peer when an ACL is not assigned as an inbound filter.

Only one ACL can be assigned as an inbound filter to an MSDP peer. Any subsequent **ip msdp sa-filter in** commands for the peer replace the existing command.

The **no ip msdp sa-filter in** and **default ip msdp sa-filter in** commands remove the ACL assignment as an inbound filter by removing the corresponding **ip msdp sa-filter in** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
ip msdp sa-filter in peer_id list list_name
no ip msdp sa-filter in peer_id
default ip msdp sa-filter in peer_id
```

Parameters

- *peer_id* MSDP peer address (IPv4 address).
- *list_name* name of ACL that filters SA messages.

Related Commands

- **ip msdp sa-filter out** assigns an IP ACL to filter outbound SA messages to a specified MSDP peer.

Guidelines

The command accepts standard and extended ACLs. The address field in a standard ACLs filters an SA message on its group address.

Example

- These commands create an IP ACL named LIST-IN as the inbound SA message filter for the MSDP peer connection to 10.4.4.12. The ACL permits SAs from the multicast group 239.14.4.2/28.

```
switch(config)#ip access-list LIST-IN
switch(config-acl-LIST-IN)#permit ip any  239.14.4.2/28
switch(config-acl-LIST-IN)#exit
switch(config)#ip msdp sa-filter in 10.4.4.12 list LIST-IN
switch(config)#show ip msdp peer
MSDP Peer 10.4.4.12
Connection status:
   State: Listen
   Connection Source: Loopback100 ( 10.6.8.6 )
SA Filtering:
Input Filter: LIST-IN

switch(config)#
```

ip msdp sa-filter out

The **ip msdp sa-filter out** command assigns an IP access control list (ACL) as a filter for outbound Source-Active (SA) messages to the specified MSDP peer connection. The switch only sends SA messages to the specified peer that are accepted by the assigned ACL. The switch sends all SA messages to the peer when an ACL is not assigned as an output filter to the peer.

Only one ACL can be assigned as an outbound filter to an MSDP peer. Any subsequent **ip msdp sa-filter out** commands for the peer replace the existing command.

The **no ip msdp sa-filter out** and **default ip msdp sa-filter out** commands remove the ACL assignment as an outbound filter by removing the corresponding **ip msdp sa-filter out** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
ip msdp sa-filter out peer_id list list_name
no ip msdp sa-filter out peer_id
default ip msdp sa-filter out peer_id
```

Parameters

- *peer_id* MSDP peer address (IPv4 address).
- *list_name* name of ACL that filters SA messages.

Related Commands

- **ip msdp sa-filter in** assigns an IP ACL to filter inbound SA messages from a specified MSDP peer.

Guidelines

The command accepts standard and extended ACLs. The address field in a standard ACLs filters an SA message on its group address.

Example

- These commands assign the IP ACL named LIST-OUT as the outbound SA message filter for the MSDP peer connection to 10.4.4.12.

```
switch(config)#ip access-list LIST-OUT
switch(config-acl-LIST-OUT)#permit ip any  239.14.4.2/28
switch(config-acl-LIST-OUT)#exit
switch(config)#ip msdp sa-filter out 10.4.4.12 list LIST-OUT
switch(config)#show ip msdp peer
MSDP Peer 10.4.4.12
Connection status:
   State: Listen
   Connection Source: Loopback100 ( 10.6.8.6 )
SA Filtering:
Output Filter: LIST-OUT

switch(config)#
```

ip msdp sa-limit

The **ip msdp sa-limit** command specifies the maximum number of Source-Active messages from a specified MSDP peer that the switch allows in the SA cache. SA messages have an expiration period of 90 seconds, during which time they remain in the SA cache. The switch does not accept SA messages from a peer after the peer's SA limit is reached. By default, The limit to the number of SA messages that the switch can store from a specified peer is 40000, by default.

The **no ip msdp sa-limit** and **default ip msdp sa-limit** commands restore the SA limit of 40000 for the specified MSDP peer by removing the corresponding **ip msdp sa-limit** statement from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
ip msdp sa-limit peer_id quantity
no ip msdp sa-limit peer_id
default ip msdp sa-limit peer_id
```

Parameters

- *peer_id* MSDP peer (IPv4 address).
- *quantity* maximum number of SA messages that the switch can store. Value ranges from 0 to 40000.

Example

- This command sets the SA limit of 500 for the MSDP peer at 10.1.1.5

```
switch(config)#ip msdp sa-limit 10.1.1.5 500
switch(config)#
```

ip msdp shutdown

The **ip msdp shutdown** command closes the peering session with the specified MSDP peer by terminating the TCP connection between the switch and the peer. The connection is not resumed until the shutdown command is removed from ***running-config***.

The **no ip msdp shutdown** and **default ip msdp shutdown** commands estanblish an MSDP peering session with the specified peer by removing the corresponding **ip msdp shutdown** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
ip msdp peer_id shutdown
no ip msdp peer_id shutdown
default ip msdp peer_id shutdown
```

Parameters

- *peer_id* MSDP peer (IPv4 address).

Examples

- This command closes the peering session with the MSDP peer at 10.4.4.12.

```
switch(config)#ip msdp shutdown 10.4.4.12
switch(config)#show ip msdp peer
MSDP Peer 10.4.4.12
Description: NORTH
Connection status:
   State: Disabled
   Connection Source: Loopback100 ( 10.6.8.6 )

switch(config)#
```

- This command reopens the peering seesion with the peer at 10.4.4.12.

```
switch(config)#no ip msdp shutdown 10.4.4.12
switch(config)#show ip msdp peer
MSDP Peer 10.4.4.12
Description: NORTH
Connection status:
   State: Connect
   Connection Source: Loopback100 ( 10.6.8.6 )

switch(config)#
```

ip msdp timer

The **ip msdp timer** command specifies the period that the switch waits after an MSDP peering session is reset before trying to reestablish the session. The default period is 30 seconds.

The **no ip msdp timer** and **default ip msdp timer** commands reset the timer interval to the default period of 30 seconds by removing the **ip msdp timer** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
ip msdp timer connect_retry
no ip msdp timer connect_retry
default ip msdp timer connect_retry
```

Parameters

- *connect_retry* Reconnect period (seconds). Value ranges from 1 to 65535. Default is 30.

Example

- This command configures the switch to wait 45 seconds after an MSDP peering session is reset before attempting to reestablish the session.

```
switch(config)#ip msdp timer 45
switch(config)#
```

show ip msdp mesh-group

The **show ip msdp mesh-group** command displays the mesh group membership of MSDP peers that are configured on the switch. An MSDP mesh group is a network of MSDP speakers where each speaker is directly connected to every other speaker. The switch does not forward Source-Active (SA) messages that it receives from a mesh group peer to other peers of the same group.

Command Mode

EXEC

Command Syntax

```
show ip msdp mesh-group
```

Related Commands

- **ip msdp mesh-group** configures the MSDP peer connection as an MSDP mesh group member.

Example

- This command displays the mesh group membership of configured MSDP peers.

```
switch>show ip msdp mesh-group
Mesh Group: tier_01
        10.24.18.13
Mesh Group: tier_02
        10.26.101.18
switch(config)#
```

show ip msdp peer

The **show ip msdp peer** command displays information about specified MSDP peers. The command includes an optional parameter for displaying SAs accepted from the peer.

Command Mode

EXEC

Command Syntax

show ip msdp peer [*PEER_ADDR*] [*SA_ACCEPT*]

Parameters

- ***PEER_ADDR*** Peers for which command displays information.
  - <no parameter> All peers configured on the switch.
  - *ipv4_addr* Address of specified MSDP peer.
- ***SA_ACCEPT*** Command displays SAs accepted from the specified peers.
  - <no parameter> Accepted SAs are not displayed.
  - **accepted-sas** Accepted SAs are displayed.

Example

- This command displays MSDP information concerning the peer located at 10.2.42.4, including SAs that the switch accepted from this peer.

```
switch>show ip msdp peer 10.2.42.4 accepted-sas
MSDP Peer 10.2.42.4
Connection status:
  State: Up
  Connection Source: Loopback4 ( 10.2.43.4 )
SA Filtering:
Input Filter: allow-multicast-for-msdp
Output Filter: allow-multicast-for-msdp
SAs accepted:
(10.62.79.30, 234.1.4.2), RP 10.2.42.4
(10.61.79.29, 234.1.4.1), RP 10.2.42.4
(10.62.79.30, 234.1.4.1), RP 10.2.42.4
switch>
```

show ip msdp pim sa-cache

The **show ip msdp pim sa-cache** command displays the SA cache for the local PIM domain configured on the switch. An SA cache is a table of Source-Active messages that are generated or accepted by the PIM domain.

Command Mode

EXEC

Command Syntax

show ip msdp pim sa-cache

Example

- This command displays the SA cache for the local PIM domain.

```
switch>show ip msdp pim sa-cache
MSDP Source Active Messages for local Pim RP
(10.51.71.23, 234.1.4.1), RP 10.2.43.4
(10.20.91.26, 234.1.4.1), RP 10.2.43.4
(10.51.71.23, 234.1.4.2), RP 10.2.43.4
(10.20.91.21, 234.1.4.1), RP 10.2.43.4
(10.51.79.23, 234.1.4.1), RP 10.2.43.4
(10.20.91.24, 234.1.4.2), RP 10.2.43.4
(10.51.79.23, 234.1.4.2), RP 10.2.43.4
(10.20.91.21, 234.1.4.2), RP 10.2.43.4
(10.20.91.26, 234.1.4.2), RP 10.2.43.4
(10.20.91.24, 234.1.4.1), RP 10.2.43.4
switch>
```

show ip msdp rpf-peer

The **show ip msdp rpf-peer** command displays MSDP information for the peer from which the switch accepts SA messages for a specified rendezvous point (RP).

The switch examines the BGP routing table to determine the next hop peer toward the originating RP of an SA message. This next hop peer is the reverse path forwarding (RPF) peer. Because the switch receives SA messages from the RPF peer, it forwards the message to all other MSDP peers. The switch rejects identical SA messages that it receives from a non-RPF peer.

Command Mode

EXEC

Command Syntax

show ip msdp peer *rp_addr*

Parameters

- *rp_addr* PIM RP IPv4 address.

Example

- This command displays MSDP information for the peer from which the switch accepts SA messages for the RP at 10.5.29.4.

```
switch>show ip msdp rpf-peer 10.5.29.4
Rpf Peer is 10.5.29.4 for RP 10.5.29.4
switch>
```

show ip msdp sa-cache

The **show ip msdp sa-cache** command displays contents of the Source-Active (SA) message cache. The command provides these filter options for displaying partial cache contents:

- multicast group address: multicast group
- source address and group address

The command can also display unexpired SAs that were rejected by ACL filters or cache limit exceeded conditions.

Command Mode

EXEC

Command Syntax

show ip msdp sa-cache [*ADDRESS_FILTER*] [*CONTENTS*]

Parameters

- ***ADDRESS_FILTER*** IPv4 address used to filter SA messages.
  - <no parameter> All SA messages.
  - *grp_addr* Multicast group address (IPv4 address).
  - *src_addr* *grp_addr* Source and multicast group addresses (two IPv4 addresses).

    grp_addr must be a valid multicast address.
- ***CONTENTS*** type of SAs that the command displays.
  - <no parameter> Displays contents of SA Cache.
  - **rejected** Displays rejected SAs in addition to the SA cache contents.

Example

- This command displays the contents of the SA message cache.

```
switch>show ip msdp sa-cache
MSDP Source Active Cache
(10.61.71.29, 234.1.4.2), RP 10.5.29.4, heard from 10.5.29.4
(10.51.71.23, 234.1.4.1), RP 10.5.29.4, heard from 10.5.29.4
(10.61.79.29, 234.1.4.2), RP 10.5.29.4, heard from 10.5.29.4
(10.53.71.27, 234.1.4.2), RP 10.3.25.4, heard from 10.3.25.4
(10.10.101.24, 234.1.4.1), RP 10.2.44.4, heard from 10.2.44.4
(10.10.151.22, 234.1.4.2), RP 10.1.12.4, heard from 10.1.12.4
(10.61.71.29, 234.1.4.1), RP 10.5.29.4, heard from 10.5.29.4
(10.20.91.21, 234.1.4.1), RP 10.2.44.4, heard from 10.2.44.4
(10.61.79.29, 234.1.4.1), RP 10.2.42.4, heard from 10.2.42.4
(10.53.79.27, 234.1.4.2), RP 10.3.25.4, heard from 10.3.25.4
(10.10.151.28, 234.1.4.2), RP 10.3.25.4, heard from 10.3.25.4
(10.52.79.25, 234.1.4.2), RP 10.2.44.4, heard from 10.2.44.4
(10.52.71.25, 234.1.4.2), RP 10.2.44.4, heard from 10.2.44.4
(10.20.91.24, 234.1.4.1), RP 10.5.29.4, heard from 10.5.29.4
(10.10.151.22, 234.1.4.1), RP 10.1.12.4, heard from 10.1.12.4
switch>
```

show ip msdp sanity

The **show ip msdp sanity** command performs a consistency check between the multcast routing table and the MSDP Source-Address (SA) caches. When the command detects inconsistencies, it displays the cache entries that are not in the table.

Command Mode

EXEC

Command Syntax

```
show ip msdp sanity
```

Example

- This command displays a sanity check that detects no inconsistencies between the SA cache and the multicast routing table.

```
switch>show ip msdp sanity
PIM SA cache entries not in the MRT
Msdp-learnt MRT entries not in the SA cache
SA cache entries not in the MRT
May-Notify-MSDP entries not in the PIM SA cache
(need not be an error condition)
switch>
```

- This command displays inconsistencies between the SA cache and the multicast routing table.

```
switch>show ip msdp sanity
PIM SA cache entries not in the MRT
Msdp-learnt MRT entries not in the SA cache
SA cache entries not in the MRT
(192.168.3.8, 224.1.154.1)
(192.168.3.35, 224.1.167.1)
(192.168.3.16, 224.1.226.1)
(192.168.3.19, 224.1.246.1)
(192.168.3.17, 224.1.204.1)
(192.168.3.12, 224.1.182.1)
(192.168.3.33, 224.1.150.1)
(192.168.3.26, 224.1.198.1)
(192.168.3.33, 224.1.195.1)
(192.168.3.4, 224.1.246.1)
(192.168.3.37, 224.1.188.1)
(192.168.3.12, 224.1.245.1)
(192.168.3.31, 224.1.206.1)
(192.168.3.35, 224.1.178.1)
(192.168.3.6, 224.1.155.1)
May-Notify-MSDP entries not in the PIM SA cache
(need not be an error condition)
4.1), RP 10.2.42.4
switch>
```

show ip msdp summary

The **show ip msdp summary** command displays a list of peer addresses, the operational status of the peer, and the number of Source-Active messages in the SA cache from that peer.

Command Mode

EXEC

Command Syntax

show ip msdp summary

Example

- This command displays the configured peers, the status of the peers, and the number of SA message received from those peers.

```
switch>show ip msdp summary
MSDP Peer Status Summary
Peer Address      State   SA Count
192.168.3.18      Up      0
192.168.3.16      Up      0
192.168.3.37      Listen  0
192.168.3.46      Up      0
192.168.3.47      Up      0
switch>
```

Chapter 38

Audio Video Bridging (AVB)

Arista switches support Audio Video Bridging (AVB) and the associated protocols. This chapter describes AVB concepts and the implementation of associated protocols.

Sections in this chapter include:

- Section 38.1: AVB Overview
- Section 38.2: AVB Protocols
- Section 38.3: AVB Configuration
- Section 38.4: AVB Command Descriptions

38.1 AVB Overview

Audio Video Bridging (AVB) is a protocol set that provides precision time synchronization, admission control, queuing reservation, and guaranteed bandwidth of professional grade quality audio and video across an IP network.

Supported AVB protocols include:

- Generalized Precision Time Protocol (gPTP)
- Multiple Stream Reservation Protocol (MSRP)
- Multiple VLAN Registration Protocol (MVRP)

These AVB features are supported on Arista 7280, 7150 Series, and 7500E Series switches:

- gPTP with hardware time stamping
- gPTP Grandmaster function
- MSRP protocol on Ethernet interfaces: stream admission control and propagation
- Control plane protection for PTP and MSRP control frames
- MVRP
- Traffic classes 2 and 3 for AVB traffic
- Traffic shaping on egress ports

These AVB features are not available on Arista switches:

- MSRP protocol on LAGs
- MSRP co-ordination with gPTP; streams are allowed even when gPTP is not in sync
- MMRP
- Signaling message support in gPTP
- Running peer delay mechanism on STP blocked ports
- Grandmaster-specific state machines (gPTP)

38.2 AVB Protocols

This section describes supported AVB protocols:

- Section 38.2.1: gPTP
- Section 38.2.2: MVRP
- Section 38.2.3: MSRP
- Section 38.2.4: MRP

38.2.1 gPTP

Generalized Precision Time Protocol (gPTP) is a network time synchronization standard for bridged Local Area Networks based on the IEEE 1588v2 Precision Time Protocol and supports the AVB protocol standards. Time synchronization in a gPTP domain is conducted the same way as in a PTP 1588 domain. A grandmaster is selected through the best grand master clock algorithm and distributes timing synchronization information to all directly attached peers. This information is propagated across the network to provide a common time reference to all Audio and Video end stations.

38.2.2 MVRP

Multiple VLAN Registration Protocol (MVRP) is an application of Multiple Registration Protocol used by AVB endpoints to dynamically register and unregister VLANs on an interface.

When an interface wishes to join a VLAN advertised by an MSRP talker (to receive a stream), MVRP sends a Join message. On receiving the Join message, the interface is added to the VLAN. If the VLAN does not already exist, MVRP dynamically creates the VLAN and propagates it through the network.

MVRP events post Syslog messages, with the severity level of **INFO** for each message.

- **MVRP_VLAN_JOIN**

  MVRP VLAN Join received/transmitted on an interface.

- **MVRP_VLAN_LV**

  MVRP VLAN Leave received/transmitted on an interface.

- **MVRP_ERROR**

  MVRP Join was discarded due to an error.

38.2.3 MSRP

Multiple Stream Registration Protocol (MSRP) is a signaling protocol that allows end stations (nodes) to reserve network resources and ensure QoS for communicating with other end stations.

MSRP nodes are specified as talkers or listeners:

- Talker nodes transmit multimedia streams to other nodes in the AVB network.
- Listener nodes receive multimedia streams from the AVB talker nodes.

MSRP is implemented by the switch on individual interfaces. MSRP is active when it is enabled on at least one interface, and stopped when it is disabled on all interfaces. MSRP uses Multiple Registration Protocol (MRP) to facilitate attribute registrations and distribution across connected end points in a LAN environment.

MSRP events post Syslog messages, with the severity level of **INFO** for each message.

- **MSRP_SR_CLASS_TRANSITION**

  MSRP SR Class state transition occurred on an interface.

- **MSRP_TALKER_ADV_JOIN**

  Talker Advertise Join message for a stream was transmitted/received on an interface.

- **MSRP_TALKER_FAIL_JOIN**

  Talker Failed Join message for a stream was transmitted/received on an interface.

- **MSRP_LISTENER_JOIN**

  Listener Join message for a stream was transmitted/received on an interface.

- **MSRP_DOMAIN_JOIN**

  Domain Join message was transmitted/received on an interface.

- **MSRP_TALKER_ADV_LV**

  Talker Advertise Leave message for a stream was transmitted/received on an interface.

- **MSRP_TALKER_FAIL_LV**

  Talker Failed Leave message for a stream was transmitted/received on an interface.

- **MSRP_LISTENER_LV**

  Listener Leave message for a stream was transmitted/received on an interface.

- **MSRP_DOMAIN_LV**

  Domain Leave message was transmitted/received on an interface.

- **MSRP_BW_ALLOC_SUCCESS**

  MSRP Bandwidth was allocated for a listener on an interface.

- **MSRP_BW_ALLOC_FAIL**

  MSRP Bandwidth could not be allocated for a listener on an interface.

- **MSRP_BW_DEALLOC**

  MSRP Bandwidth was de-allocated for a listener on an interface.

- **MSRP_ERROR**

  MSRP Join was discarded because of an error.

38.2.4 MRP

Multiple Registration Protocol (MRP) protocol includes MSRP and MVRP, and allows participants in an MRP application to register attributes with participants in a Bridged Local Area Network (BLAN).

38.3 AVB Configuration

This section describes the AVB configuration:

- Section 38.3.1: Enabling gPTP
- Section 38.3.2: Enabling MSRP
- Section 38.3.3: Displaying MSRP Configuration and Status
- Section 38.3.4: Enabling MVRP
- Section 38.3.5: Displaying MVRP Configuration and Status

38.3.1 Enabling gPTP

Configure gPTP on the switch though the ptp mode command. PTP is enabled on individual interfaces with the ptp enable command.

Example

- These commands configure gPTP on the switch and enable PTP on Ethernet interfaces 41-45.

```
switch(config)#ptp mode gptp
switch(config)#interface ethernet 41-45
switch(config-if-Et41-45)#ptp enable
switch(config)#show running-config
             <-------OUTPUT OMITTED FROM EXAMPLE-------->
!
ptp mode gptp
!
             <-------OUTPUT OMITTED FROM EXAMPLE-------->
end
switch(config-if-Et41-45)#show active
interface Ethernet41
   speed forced 10000full
   msrp
   ptp enable
interface Ethernet42
   speed forced 10000full
   msrp
   ptp enable
interface Ethernet43
   speed forced 10000full
   msrp
   ptp enable
interface Ethernet44
   speed forced 10000full
   ptp enable
interface Ethernet45
   ptp enable
switch(config-if-Et41-45)#
```

38.3.2 Enabling MSRP

MSRP is enabled on an interface with the **msrp** command.

Example

- These commands enable MSRP on Ethernet interfaces 41-43.

```
switch(config)#interface ethernet 41-43
switch(config-if-Et41-43)#msrp
switch(config-if-Et41-43)#show active
interface Ethernet41
   speed forced 10000full
   msrp
interface Ethernet42
   speed forced 10000full
   msrp
interface Ethernet43
   speed forced 10000full
   msrp
switch(config-if-Et41-43)#
```

38.3.3 Displaying MSRP Configuration and Status

MSRP configuration information and status is displayed with the **show msrp** command.

Example

- This command displays the MSRP status for Ethernet interfaces 41-43.

```
switch(config)#show msrp interfaces ethernet 41-43

  MSRP Global Status : Enabled
  Max Frame Size : 1522
  Max Fan-In Ports : No limit

                            Delta
  Class Supported Priority Bandwidth
  ----- --------- -------- ---------
      A         Y        3       75%
      B         Y        2        0%

Legend
------
Adv     : Talker Advertise          Fail    : Talker Fail
AskFail : Listener Asking Failed    Rdy     : Listener Ready
RdyFail : Listener Ready Failed

       Admin    Sr                            Talkers    Listeners    Bandwidth
  Port State    Pvid  Class  Oper State      Adv  Fail  Rdy  AskFail  Allocated
  ---- -------- ----- ------ --------------- ---- ----- ---- -------- ---------
  Et41 Active      5      A  Boundary          1     0    1        0    200kbps
                          B  Core              0     0    0        1    100kbps
  Et42 Active      3      A  Core              0     0    1        1     50kbps
                          B  WaitingForPeer    1     0    0        0     20kbps
  Et43 Disabled    3

switch(config)#
```

Stream data is available for the talker and listener. The **show msrp interfaces** command displays the status and configuration information for each stream.

Examples

- This command displays data for listener station streams on Ethernet interfaces 1 and 2.

```
switch(config)#show msrp interfaces ethernet 1-2
MSRP Global Status : Enabled
  Max Frame Size : 1522
  Max Fan-In Ports : No limit

                             Delta
  Class Supported Priority Bandwidth
  ----- --------- -------- ---------
      A         Y        3       75%
      B         Y        2        0%

Legend
------
Adv     : Talker Advertise          Fail    : Talker Fail
AskFail : Listener Asking Failed    Rdy     : Listener Ready
RdyFail : Listener Ready Failed

                                  Listeners
Port     Stream Id            Dec       Dir
-------- -------------------- --------- ------
Et1      0000.0000.0000.002a  AskFail   Tx
         0000.0000.0000.029a  RdyFail   Rx
         0000.0000.0000.038f  AskFail   Rx
Et2      0000.0000.0000.002a  AskFail   Rx
         0000.0000.0000.029a  RdyFail   Rx
         0000.0000.0000.038f  AskFail   Tx

switch(config)#
```

- This command displays data for talker station streams on Ethernet interfaces 1 and 2.

```
switch(config)#show msrp interfaces ethernet 1-2 talkers
Legend
------
Adv     : Talker Advertise          Fail    : Talker Fail

                                           Talkers
  Port     Stream Id            Dec       Dir    FailCode
  -------- -------------------- --------- ------ ---------
  Et1      0000.0000.0000.002a  Adv       Rx     --
           0000.0000.0000.038f  Fail      Tx     7

  Et2      0000.0000.0000.002a  Adv       Tx     --
           0000.0000.0000.038f  Adv       Rx     7

switch(config)#
```

38.3.4 Enabling MVRP

MVRP is disabled by default. To enable MVRP on an interface, use the **mvrp** command. MVRP is enabled globally if it is enabled on at least one interface.

Example

- These commands enable MVRP on Ethernet interface 34.

```
switch(config)#interface ethernet 34
switch(config-if-Et34)#mvrp
switch(config-if-Et34)#
```

38.3.5 Displaying MVRP Configuration and Status

MVRP configuration information and status are displayed with the **show mvrp** command.

Example

- This command displays the MVRP status for Ethernet interfaces 30 through 40.

```
switch(config)#show mvrp interfaces Ethernet 30-40

  MVRP Global Status : Enabled

  Port      Admin State   Registered Vlans      Declared Vlans
  --------  ------------  --------------------  --------------------
  Et30      Disabled
  Et31      Disabled
  Et32      Disabled
  Et33      Disabled
  Et34      Active
  Et35      Disabled
  Et36      Disabled
  Et37      Disabled
  Et38      Disabled
  Et39      Disabled
  Et40      Disabled
switch(config)#
```

38.4 AVB Command Descriptions

MSRP Commands

- msrp
- msrp streams load-file
- show msrp
- show msrp interfaces
- show msrp streams

MRP Commands

- mrp leave-all-timer
- mrp leave-timer

MVRP Commands

- mvrp
- show mvrp

msrp

MSRP enables Multiple Stream Registration Protocol (MSRP), which is a signaling protocol that provides nodes with the ability to reserve network resources to ensure Quality of Service (QoS) between talker and listener endpoints. The Stream Reservation Protocol (SRP) utilizes MSRP to reserve bandwidth for data streams, and configure a complete path between endpoints.

The **msrp** command enables MSRP on the configuration mode interface. If MSRP was not previously enabled on any interface, the MSRP agent is launched by this command.

The **no msrp** and **default msrp** commands disable MSRP on the configuration mode interface, and removes the corresponding **msrp** command from ***running-config***. The command stops the MSRP agent when MSRP is no longer enabled on any interface.

Command Mode

Interface-Ethernet Configuration

Command Syntax

```
msrp
no msrp
default msrp
```

Example

- These commands enable MSRP on Ethernet interface 3/3/3. Because it was not previously enabled on any other interface, the command launches the MSRP agent.

```
switch(config)#interface ethernet 3/3/3
switch(config-if-Et3/3/3)#msrp
Launching MSRP Agent
switch(config-if-Et3/3/3)#show active
 interface Ethernet3/3/3
   msrp
switch(config-if-Et3/3/3)#
```

- These commands disable the MSRP agent on Ethernet interface 3/3/3. Because it is not enabled on any other interface, the command stops the MSRP agent.

```
switch(config-if-Et3/3/3)#no msrp
Stopping MSRP agent
switch(config-if-Et3/3/3)#show active
 interface Ethernet3/3/3
switch(config-if-Et3/3/3)#msrp
```

mrp leave-all-timer

The **mrp leave-all-timer** command specifies the mrp leave all timer interval for the configuration mode interface.

When starting MRP, a participant starts its LeaveAll timer. Upon timer expiry, it sends a LeaveAll message and restarts its timer. When other participants receive the message, they register their attributes and restart their leave-all timers.

The default leave-all timer interval is a randomly selected value from 10 to 15 seconds. Under normal conditions, this value should not be adjusted.

The **no mrp leave-all-timer** and **default mrp leave-all-timer** commands restore the default leave-all timer interval on the configuration mode interface by removing the corresponding **mrp leave-all-timer** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration

Command Syntax

```
mrp leave-all-timer period
no mrp leave-all-timer
default mrp leave-all-timer
```

Parameters

- *period* leave all timer interval (seconds). Values range from 10 to 60. Default value is a randomly selected value from 10 to 15.

Example

- This command sets the MRP leave-all timer interval on Ethernet interface 17 to twelve seconds.

```
switch(config)#interface ethernet 17
switch(config-if-Et17)#mrp leave-all-timer 12
switch(config-if-Et17)#
```

mrp leave-timer

The leave-timer controls the deregistration of attributes. If an MRP participant needs other participants to unregister their attributes, it sends a Leave message. When receiving a Leave message, the Leave-timer starts and unregisters the attributes if it doesn't receive Join messages for the attributes before the Leave-timer expires.

The **mrp leave-timer** command specifies the mrp leave-timer interval for the configuration mode interface. The default leave-timer interval is 0.6 seconds. Under normal operation conditions, this value should not be adjusted.

The **no mrp leave-timer** and **default mrp leave-timer** commands restore the default leave-timer interval of 0.6 seconds on the configuration mode interface by removing the corresponding **mrp leave-timer** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration

Command Syntax

```
mrp leave-timer period
no mrp leave-timer
default mrp leave-timer
```

Parameters

- *period* leave all timer (seconds). Values range from 0.6 to 30.

Example

- This command sets the MRP leave timer interval on Ethernet interface 17 to 0.8 seconds.

```
switch(config)#interface ethernet 17
switch(config-if-Et17)#mrp leave-timer 0.8
switch(config-if-Et17)#
```

msrp streams load-file

The load-file for MSRP streams provides a file that contains an alias that can be substituted in the name (stream-id) of a string.

The **msrp streams load-file** command allows users to include a line in the file (example: ***0102.0304.0506 XYZW4*** or ***0102.0304 XYZW5***) that causes the bytes to be replaced with the accompanying string in **show msrp streams** commands.

The **no msrp streams load-file** and **default msrp streams load-file** commands remove the alias assignment by removing the corresponding **msrp streams load-file** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
msrp streams load-file [FILE TYPE]
no msrp streams load-file
default msrp streams load-file
```

Parameters

- ***FILE TYPE*** The options include:
  - **certificate:** device name, directory, or file name
  - **extension:** device name, directory, or file name
  - **file:** device name, directory, or file name
  - **flash:** device name, directory, or file name
  - **ftp:** device name, directory, or file name
  - **http:** device name, directory, or file name
  - **https:** device name, directory, or file name
  - **scp:** device name, directory, or file name
  - **sftp:** device name, directory, or file name
  - **sslkey:** device name, directory, or file name
  - **system:** device name, directory, or file name
  - **terminal:** device name, directory, or file name
  - **tftp:** device name, directory, or file name
  - **usb1:** device name, directory, or file name

Example

- This command indicates that the file named file1 contains the alias names that is used in MSRP stream names.

```
switch(config)#msrp streams load-file file1
switch(config)#
```

mvrp

MVRP dynamically registers and unregisters VLANs on an interface. When an interface wishes to join a VLAN advertised by an MSRP talker (to receive a stream), MVRP sends a Join message. On receipt of the Join message, the interface is added to the VLAN. If the VLAN does not already exist, MVRP dynamically creates the VLAN and propagates it through the network.

The **mvrp** command enables Multiple VLAN Registration Protocol (MVRP) on the configuration mode interface. If MVRP was not previously enabled on any interface, MVRP is also enabled globally by this command.

The **no mvrp** and **default mvrp** commands disable MVRP on the configuration mode interface by removing the corresponding **mvrp** command from ***running-config***. These commands also disable MVRP globally when MVRP is no longer enabled on any interface.

Command Mode

Interface-Ethernet Configuration

Command Syntax

```
mvrp
no mvrp
default mvrp
```

Example

- These commands enable MVRP on Ethernet interface 34. If MVRP was not previously enabled on any other interface, these commands also enable MVRP globally.

```
switch(config)#interface ethernet 34
switch(config-if-Et34)#mvrp
switch(config-if-Et34)#
```

show msrp

The **show msrp** command displays MSRP operational information for the specified interfaces.

Command Mode

EXEC

Command Syntax

show msrp [*INTERFACE_NAME*]

Parameters

- ***INTERFACE_NAME*** Interface type and number. Values include
  - <no parameter> all Ethernet interfaces.
  - **interfaces ethernet** *e_range* Ethernet interface list.

  Valid *e_range* formats include number, range, or comma-delimited list of numbers and ranges.

Example

- This command displays the MSRP status for Ethernet interfaces 41 through 43.

```
switch(config)#show msrp interfaces ethernet 41-43

  MSRP Global Status : Enabled
  Max Frame Size : 1522
  Max Fan-In Ports : No limit

  Class Supported Priority Delta Bandwidth
  ----- --------- -------- ---------
      A         Y        3       75%
      B         Y        2        0%

Legend
------
Adv     : Talker Advertise          Fail    : Talker Fail
AskFail : Listener Asking Failed    Rdy     : Listener Ready
RdyFail : Listener Ready Failed

       Admin    Sr                            Talkers     Listeners    Bandwidth
  Port State    Pvid  Class  Oper State      Adv  Fail  Rdy  AskFail  Allocated
  ---- -------- ----- ------ --------------- ---- ----- ---- -------- ---------
  Et41 Active      5      A  Boundary          1     0    1        0    200kbps
                          B  Core              0     0    0        1    100kbps
  Et42 Active      3      A  Core              0     0    1        1     50kbps
                          B  WaitingForPeer    1     0    0        0     20kbps
  Et43 Disabled    3

switch(config)#
```

show msrp interfaces

The **show msrp interfaces** command displays station stream information for the specified station type, interfaces and streams.

Command Mode

EXEC

Command Syntax

show msrp interfaces [*INTERFACE_NAME*] *STATION_TYPE_NAME* [*STREAMS*]

Parameters

- ***INTERFACE_NAME*** Interface type and number. Values include:
  - <no parameter> all Ethernet interfaces.
  - **ethernet** *e_range* Ethernet interface list.
- ***STATION_TYPE*** Endpoint type. Values include:
  - **talker** Command displays data for talker station streams.
  - **listeners** Command displays data for listener station streams.
  - **streams** Command displays data for talker and listener station streams.
- ***STREAMS*** Streams for which command displays information. Options include:
  - <no parameter> all streams.
  - **stream-id** *hex_string* specifies the stream command for which command displays information.

  Valid *e_range* formats include number, range, or comma-delimited list of numbers and ranges.

  Valid *hex_string* formats include <H>, <H.H>, <H.H.H>, or <H.H.H.H>, where H is a four-digit hex number that ranges from ***0*** to ***FFFF***.

Example

- This command displays data for listener station streams on Ethernet interfaces 1 and 2.

```
switch(config)#show msrp interfaces ethernet 1-2 listeners
Legend
------
AskFail : Listener Asking Failed    Rdy     : Listener Ready
RdyFail : Listener Ready Failed

                                  Listeners
  Port     Stream Id            Dec       Dir
  -------- -------------------- --------- ------
  Et1      0000.0000.0000.002a  AskFail   Tx
           0000.0000.0000.029a  RdyFail   Rx
           0000.0000.0000.038f  AskFail   Rx

  Et2      0000.0000.0000.002a  AskFail   Rx
           0000.0000.0000.029a  RdyFail   Rx
           0000.0000.0000.038f  AskFail   Tx

switch(config)#
```

- This command displays data for talker station streams on Ethernet interfaces 1 and 2.

```
switch(config)#show msrp interfaces ethernet 1-2 talkers
Legend
------
Adv     : Talker Advertise          Fail    : Talker Fail

                                          Talkers
  Port     Stream Id            Dec       Dir    FailCode
  -------- -------------------- --------- ------ ---------
  Et1      0000.0000.0000.002a  Adv       Rx     --
           0000.0000.0000.038f  Fail      Tx     7

  Et2      0000.0000.0000.002a  Adv       Tx     --
           0000.0000.0000.038f  Adv       Rx     7

switch(config)#
```

show msrp streams

The **show msrp streams** command displays configuration and status information on the specified MSRP streams.

Command Mode

EXEC

Command Syntax

show msrp streams [*STREAM_NAME*] [*INFO_LEVEL*]

Parameters

- ***STREAMS*** Streams for which command displays information. Options include:
  - <no parameter> all streams.
  - **stream-id** *hex_string* specifies the stream command for which command displays information.

  Valid *hex_string* formats include <H>, <H.H>, <H.H.H>, or <H.H.H.H>, where H is a four-digit hex number that ranges from ***0*** to ***FFFF***.
- ***INFO_LEVEL*** type of information that the command displays. Options include:
  - <no parameter> command displays stream identification information.
  - **detail** command displays identification and transmission characteristics.
  - **propagation** command displays ingress and egress port information.

Examples

- This command displays stream identification information.

```
switch(config)#show msrp interfaces streams
Legend
------
Adv     : Talker Advertise          Fail    : Talker Fail

  Stream Id            DMAC                Port     Dec    Vlan Class Bandwidth
  -------------------- ------------------- -------- ------ ---- ----- ----------
  0000.0000.0000.002a  00:11:22:33:44:55   Et1      Adv      24     A   8000kbps
  0000.0000.0000.029a  22:33:44:55:66:77   --       --     4095     A      0kbps
  0000.0000.0000.038f  11:22:33:44:55:66   Et2      Adv     119     B   1000kbps

switch(config)#
```

- This command displays stream identification and status information.

```
switch(config)#show msrp streams detail
Legend
------
Adv     : Talker Advertise          Fail    : Talker Fail

Stream Id            DMAC                Port     Dec    Vlan Class Bandwidth
-------------------- ------------------- -------- ------ ---- ----- ----------
  0000.0000.0000.002a  00:11:22:33:44:55 Et1      Adv    24   A     8000kbps
                Latency (nsec): 800
                Max Frame Size: 1522
                Max Interval Frames: 2

  0000.0000.0000.029a  22:33:44:55:66:77 --       --     4095 A     0kbps
                Latency (nsec): 0
                Max Frame Size: 1100
                Max Interval Frames: 3

switch(config)#
```

- This command displays stream ingress and egress port information.

```
switch(config)#show msrp streams propagation
Legend
------
Adv     : Talker Advertise          Fail    : Talker Fail
AskFail : Listener Asking Failed    Rdy     : Listener Ready
RdyFail : Listener Ready Failed

Stream Id            DMAC                Port     Dec    Vlan Class Bandwidth
-------------------- ------------------- -------- ------ ---- ----- ----------
0000.0000.0000.002a  00:11:22:33:44:55   Et1      Adv    24   A     8000kbps

    Talker Propagation:
         Ingress      Ingress      Propagated     Propagated     Egress
         Dec          Port         Dec            Port           Dec
         --------     --------     ----------     ----------     --------
         Adv         --> Et1         --> Adv       --> Et2        --> Adv
                                                    Et4        --> Adv
                                                    Et5        --> Adv

    Listener Propagation:
         Egress       Egress       Propagated     Listener       Ingress
         Dec          Port         Dec            Port           Dec
         --------     --------     ----------     ----------     --------
         AskFail  <--  Et1         <-- AskFail   <-- Et2       <-- AskFail
                                   AskFail  <-- Et4       <-- AskFail
                                   AskFail  <-- Et5       <-- AskFail

0000.0000.0000.029a  22:33:44:55:66:77   --       --     4095  A    0kbps

    Talker Propagation:
         Ingress      Ingress      Propagated     Propagated     Egress
         Dec          Port         Dec            Port           Dec
         --------     --------     ----------     ----------     --------

    Listener Propagation:
         Egress       Egress       Propagated     Listener       Ingress
         Dec          Port         Dec            Port           Dec
         --------     --------     ----------     ----------     --------
                                   RdyFail   <-- Et1       <-- RdyFail
                                   RdyFail   <-- Et2       <-- RdyFail

0000.0000.0000.038f  11:22:33:44:55:66   Et2      Adv     119   B   1000kbps

    Talker Propagation:
         Ingress      Ingress      Propagated     Propagated     Egress
         Dec          Port         Dec            Port           Dec
         --------     --------     ----------     ----------     --------
         Adv        -->  Et2         --> Fail      --> Et1       --> Fail

    Listener Propagation:
         Egress       Egress       Propagated     Listener       Ingress
         Dec          Port         Dec            Port           Dec
         --------     --------     ----------     ----------     --------
         AskFail  <--  Et2        <--  AskFail   <-- Et1      <-- Rdy

switch(config)#
```

show mvrp

The **show mvrp** command displays MVRP operational information for the specified interfaces.

Command Mode

EXEC

Command Syntax

show MVRP [*INTERFACE_NAME*]

Parameters

- ***INTERFACE_NAME*** Interface type and number. Values include
  - <no parameter> all Ethernet interfaces.
  - **interfaces ethernet** *e_range* Ethernet interface list.

  Valid *e_range* formats include number, range, or comma-delimited list of numbers and ranges.

Example

- This command displays the MVRP status for Ethernet interfaces 30 through 40.

```
switch(config)#show mvrp interfaces Ethernet 30-40

MVRP Global Status : Enabled

Port      Admin State   Registered Vlans      Declared Vlans
--------  ------------  --------------------  --------------------
Et30      Disabled
Et31      Disabled
Et32      Disabled
Et33      Disabled
Et34      Active
Et35      Disabled
Et36      Disabled
Et37      Disabled
Et38      Disabled
Et39      Disabled
Et40      Disabled

switch(config)#
```

Chapter 39

SNMP

This chapter describes the Arista switch SNMP agent and contains these sections:

- Section 39.1: SNMP Introduction
- Section 39.2: SNMP Conceptual Overview
- Section 39.3: Configuring SNMP
- Section 39.4: SNMP Commands

39.1 SNMP Introduction

Arista Networks switches support many standard SNMP MIBs, making it easier to integrate these platforms into existing network management infrastructures. With only a few configurations, many public domain and commercially available network management tools can quickly manage Arista switches out of the box. Support of SNMP V2 groups and views and V3 security allow network managers to tune switch monitoring to match the administration policy of the IT organization.

39.2 SNMP Conceptual Overview

Simple Network Management Protocol (SNMP) is a protocol that provides a standardized framework and a common language to monitor and manage network devices.

39.2.1 SNMP Structure

The SNMP framework has three parts:

- **SNMP manager**: The SNMP manager controls and monitors network host activities and is typically part of a Network Management System (NMS).
- **SNMP agent**: The SNMP agent is the managed device component that manages and reports device information to the manager.
- **Management Information Base (MIB)**: The MIB stores network management information.

The agent and MIB reside on the switch. Enabling the SNMP agent requires the definition of the manager-agent relationship. The agent contains MIB variables whose values the manager can request or change. The agent gathers data from the MIB and responds to requests for information. For a list of supported MIBs, please refer to the release notes for a specific EOS version.

This chapter discusses enabling the SNMP agent on an Arista switch and controlling notification transmissions from the agent. Information on using SNMP management systems is available in the appropriate documentation for the corresponding NMS application.

39.2.2 SNMP Notifications

SNMP notifications are messages, sent by the agent, informing of an event or a network condition. A ***trap*** is an unsolicited notification. An ***inform*** (or inform request) is a trap that includes a request for a confirmation that the message is received. Events that a notification can indicate include improper user authentication, restart, and connection losses.

For a list of supported traps, please refer to the release notes for a specific EOS version.

39.2.3 SNMP Versions

Arista switches support the following SNMP versions:

- **SNMPv1**: The Simple Network Management Protocol, defined in RFC 1157. Security is based on community strings.
- **SNMPv2c**: Community-string based Administrative Framework for SNMPv2, defined in RFC 1901, RFC 1905, and RFC 1906. Security is based on SNMPv1.
- **SNMPv3**: Version 3, as defined in RFCs 2273 to 2275.

39.3 Configuring SNMP

This section describes the steps that configure the switch SNMP agent to communicate with an SNMP manager, including the following:

- **Enabling and Disabling SNMP**
- **Configuring Community Access Control**
- **Configuring SNMP Parameters**
- **Configuring the Agent to Send Notifications**
- **Extending the SNMP Agent Through Run Time Scripts**

39.3.1 Enabling and Disabling SNMP

SNMP is enabled globally by issuing any **snmp-server community** or **snmp-server user** command. The **no snmp-server** command disables SNMP agent operation by removing all non-default **snmp-server** commands from ***running-config***.

39.3.2 Enabling SNMP in a VRF

By default, SNMP is enabled only in the **default** VRF. The switch can only send SNMP traps and informs if the host that has been configured to receive them is accessible through an interface in a VRF in which SNMP has been enabled.

To enable or disable SNMP in a VRF, use the **snmp-server vrf** command.

39.3.3 Configuring Community Access Control

SNMP community strings serve as passwords that permit an SNMP manager to access the agent on the switch. A Network Management System (NMS) can access the switch only if its community string matches at least one of the switch's community strings.

The **snmp-server community** command configures the community string.

Example

- This command adds the community string ***ab_1*** to provide read-only access to the switch agent.

```
switch(config)#snmp-server community ab_1 ro
switch(config)#
```

Community statements can reference views to limit MIB objects that are available to a manager. A view is a community string object that specifies a subset of MIB objects. The **snmp-server view** command configures the community string.

Example

- These commands create a view that includes all objects in the ***system*** group except for those in ***system.2***.

```
switch(config)#snmp-server view sys-view system include
switch(config)#snmp-server view sys-view system.2 exclude
switch(config)#
```

- This command adds the community string ***lab_1*** to provide read-only access to the switch agent for the previously defined view.

```
switch(config)#snmp-server community lab_1 sys-view
switch(config)#
```

39.3.4 Configuring SNMP Parameters

This section describes these SNMP parameter configuration tasks:

- **Configuring the Engine ID**
- **Configuring the Group**
- **Configuring the User**
- **Configuring the Host**
- **Enabling Link Trap Generation**
- **Configuring the Chassis-id String**
- **Configuring the Contact String**
- **Configuring the Location String**

Configuring the Engine ID

The **snmp-server engineID remote** command configures the the name of a Simple Network Management Protocol (SNMP) engine located on a remote device. Use the **snmp-server engineID local** command for the local engine.

A remote agent's engine ID must be configured before remote users for that agent are configured. User authentication and privacy digests are derived from the engine ID and user passwords. The configuration command fails if the remote engine ID is not configured first.

Important! When the remote engine ID is changed, all user passwords associated with the engine must be reconfigured.

Example

- This command configures DC945798CAB4 as the name of the remote SNMP engine located at 12.23.104.25, UDP port 162

```
switch(config)#snmp-server engineID remote 10.23.104.25 udp-port DC945798CA
switch(config)#
```

Configuring the Group

An SNMP group grants specific levels of SNMP access to group users. The **snmp-server group** command configures a new SNMP group.

Example

- This command configures ***normal_one*** as an SNMPv3 group (authentication and encryption) that provides access to the ***all-items*** read view.

```
switch(config)#snmp-server group normal_one v3 priv read all-items
switch(config)#
```

Configuring the User

Members of SNMP groups are called “users.” The **snmp-server user** command allows a new user to be added an SNMP group and configures that user’s parameters. Remote users are configured by specifying the IP address or port number that accesses the user’s SNMP agent.

Example

- This command configures the local SNMPv3 user ***tech-1*** as a member of the SNMP group ***tech-sup***.

```
switch(config)#snmp-server user tech-1 tech-sup v3
switch(config)#
```

- This command configures the remote SNMPv3 user ***tech-2*** as a member of the SNMP group ***tech-sup***. The remote user is on the agent located at 13.1.1.4.

```
switch(config)#snmp-server user tech-2 tech-sup remote 13.1.1.4 v3
switch(config)#
```

Configuring the Host

The **snmp-server host** command configures an SNMP host (to which SNMP traps will be sent). The **snmp-server host** command sets the community string if it was not previously configured.

Example

- This command adds a v2c inform notification recipient at 12.15.2.3 using the community string ***comm-1***.

```
switch(config)#snmp-server host 12.15.2.3 informs version 2c comm-1
switch(config)#
```

Enabling Link Trap Generation

The **snmp trap link-status** command enables SNMP link trap generation on the configuration mode interface. SNMP link trap generation is enabled by default. If SNMP link trap generation was previously disabled, this command removes the corresponding **no snmp link-status** statement from the configuration. The **show snmp trap** command displays the SNMP link trap generation information.

Example

- This command disables SNMP link trap generation on the Ethernet 5 interface.

```
switch(config-if-Et5)#no snmp trap link-status
switch(config-if-Et5)#
```

Specifying the Source Interface

The **snmp-server source-interface** command specifies the interface from where an SNMP trap originates. The **show snmp source-interface** command displays the interface of the IP address for SNMP traps.

Example

- This command configures the Ethernet 1 interface as the source of SNMP traps and informs.

```
switch(config)#snmp-server source-interface ethernet 1
switch(config)#
```

Configuring the Chassis-id String

The chassis ID string is typically set to the serial number of the switch. The SNMP manager uses this string to associate all data retrieved from the switch with a unique identifying label. Under normal operating conditions, editing the chassis ID string contents is unnecessary.

The **snmp-server chassis-id** command configures the chassis ID string. The default chassis ID string is the serial number of the switch. The **show snmp** command displays the chassis ID.

Example

- This command configures *xyz-1234* as the chassis-ID string, then displays the result.

```
switch(config)#snmp-server chassis-id xyz-1234
switch(config)#show snmp
    Chassis: xyz-1234                              <---chassis ID
8 SNMP packets input
    0 Bad SNMP version errors
    0 Unknown community name
    0 Illegal operation for community name supplied
    0 Encoding errors
    8 Number of requested variables
    0 Number of altered variables
    4 Get-request PDUs
    4 Get-next PDUs
    0 Set-request PDUs
21 SNMP packets output
    0 Too big errors
    0 No such name errors
    0 Bad value errors
    0 General errors
    8 Response PDUs
    0 Trap PDUs
SNMP logging: enabled
    Logging to taccon.162
SNMP agent enabled
switch(config)#
```

Configuring the Contact String

The SNMP contact string is information text that typically displays the name of a person or organization associated with the SNMP agent.

The **snmp-server contact** command configures the system contact string. The contact string is displayed by the **show snmp** and **show snmp contact** commands.

Example

- These commands configure *Bonnie H at 3-1470* as the contact string.

```
switch(config)#snmp-server contact Bonnie H at 3-1470
switch(config)#
```

Configuring the Location String

The location string typically provides information about the physical location of the SNMP agent. The **snmp-server location** command configures the system location string. By default, the system location string is not set.

Example

- These commands configure *lab-25* as the location string.

```
switch(config)#snmp-server location lab_25
switch(config)#show snmp location
Location: lab_25
switch(config)#
```

39.3.5 Configuring the Agent to Send Notifications

The following steps are mandatory when setting up the SNMP agent to send notifications:

Step 1 Configure the remote engine ID.

Step 2 Configure the group.

Step 3 Configure the user.

Step 4 Configure the host.

Step 5 Enable link trap generation on the interfaces.

Section 39.3.4 describes each of these tasks.

39.3.6 Extending the SNMP Agent Through Run Time Scripts

The switch supports the execution of user supplied scripts to service portions of the OID space.

Scripts run under one of two operational modes:

- Normal: scripts run over an indefinite period to process subsequent objects after the initial request. Maintaining an executing script avoids startup and connection delay each time an object requires processing.
- One-shot mode: scripts process a single object, then terminates execution.

Normal extension scripts are conceptually multithreaded: one thread collects data and the other thread is ready to communicate with snmpd. One-shot scripts process a single object, running once and exiting. Startup and data collection overhead is required for each request. In both modes, the SNMP server is blocked from serving other requests when waiting for script responses.

The **snmp-server extension** command configures the execution of user supplied scripts to service portions of the OID space.

Example

- This command specifies the file ***example.sh***, located in flash as the script file that services the specified OID space in normal mode.

```
switch(config)#snmp-server extension .1.3.6.1.4.1.8072.2 flash:example.sh
switch(config)#
```

39.3.6.1 Normal Script Behavior

The first time the SNMP server requires a script result, it launches it with no arguments. The server communicates with the script through stdin/stdout. Before each request, the script is the string ***PING\n*** on **stdin**. The expected response is printing ***PONG\n*** to **stdout**.

GET and GETNEXT Requests

For GET and GETNEXT requests, the script is passed two lines on stdin, the command (get or getnext) and the requested OID. The expected response from the script is the printing of three lines to stdout: , the TYPE, the OID for the result varbind, and the VALUE itself.

Table 39-1 lists legal TYPE values and resulting VALUE encodings. If the command does not return an appropriate varbind, it should print print "NONE\n" to stdout and continue running; this results in an SNMP ***noSuchName*** error or a ***noSuchInstance*** exception.

Table 39-1 Extension Script Type and Encoding

Type string	SNMP type	Encoding for script
integer	Integer32	integer
unsigned	Unsigned32	integer
gauge	Gauge32	integer
counter	Counter32	integer
counter64	Counter64	integer
timetick	TimeTicks	integer
ipaddress	IpAddress	a.b.c.d
objectid	ObjectID	1.3.6.1.42.99.2468
octet	OctetString	hexadecimal string
opaque	Opaque	hexadecimal string
string	OctetString	ascii string

SET

For SET requests, script is passed three lines on stdin: the command (set), and the requested OID, and the type and value, both on the same line. If the assignment is successful, the expected script response is to print **DONE\n** to stdout. Indicated errors by writing one of the error strings described in Table 39-2 In each case, the command should continue running.

authorization-error	no-access	too-big
bad-value	no-creation	undo-failed
commit-failed	no-such-name	wrong-type
gen-error	not-writable	wrong-length
inconsistent-name	read-only	wrong-encoding
inconsistent-value	resource-unavailable	wrong-value

Table 39-2 Set Request Error Strings

39.3.6.2 One Shot Script Behavior

The command should exit after it finishes processing a single object.

GET and GETNEXT

For each GET or GETNEXT request, the script is invoked once for each OID in the space that it serves. It receives two arguments: -g for GET or -n for GETNEXT, and the requested OID.

The expected script response is the response varbind as three separate lines printed to stdout: the result OID, the type, and the value.

If the command does not return an appropriate varbind, then the script should exit without producing any output. This results in an SNMP ***noSuchName*** error, or a ***noSuchInstance exception***.

Possible reasons that a command would not return an appropriate varbind includes:

- The specified OID didn't correspond to a valid instance for a GET request.
- There were no following instances for a GETNEXT.

SET

A SET request results in the command being called with the arguments: -s, OID, TYPE and VALUE, where TYPE is a listed token Table 39-1, indicating the type of the value passed as the third parameter.

When the assignment is successful, the script exits without producing any output. Errors are indicated by writing just the error name (Table 39-2); the agent generates the appropriate error response.

39.4 SNMP Commands

Global Configuration Commands

- no snmp-server
- snmp-server chassis-id
- snmp-server community
- snmp-server contact
- snmp-server enable traps
- snmp-server engineID local
- snmp-server engineID remote
- snmp-server extension
- snmp-server group
- snmp-server host
- snmp-server location
- snmp-server source-interface
- snmp-server user
- snmp-server view
- snmp-server vrf

Interface Configuration Commands

- snmp trap link-status

Display Commands

- show snmp
- show snmp chassis
- show snmp community
- show snmp contact
- show snmp engineID
- show snmp group
- show snmp host
- show snmp location
- show snmp mib
- show snmp source-interface
- show snmp trap
- show snmp user
- show snmp view

no snmp-server

The **no snmp-server** and **default snmp-server** commands disable Simple Network Management Protocol (SNMP) agent operation by removing all **snmp-server** commands from ***running-config***.

SNMP is enabled with any **snmp-server community** or **snmp-server user** command.

Command Mode

Global Configuration

Command Syntax

```
no snmp-server
default snmp-server
```

Example

- This command disables SNMP agent operation on the switch.

```
switch(config)#no snmp-server
switch(config)#
```

show snmp

The **show snmp** command displays SNMP information including the SNMP counter status and the chassis ID string.

Example

EXEC

Command Syntax

show snmp

Example

- This command displays SNMP counter status, the chassis ID, the previously configured location string, logging status and destination, and the VRFs in which the SNMP agent is operating.

```
switch>show snmp
Chassis: JFL08320162
Location: 5470ga.dc
2329135 SNMP packets input
    0 Bad SNMP version errors
    0 Unknown community name
    0 Illegal operation for community name supplied
    0 Encoding errors
    38132599 Number of requested variables
    0 Number of altered variables
    563934 Get-request PDUs
    148236 Get-next PDUs
    0 Set-request PDUs
2329437 SNMP packets output
    0 Too big errors
    0 No such name errors
    0 Bad value errors
    0 General errors
    2329135 Response PDUs
    0 Trap PDUs
SNMP logging: enabled
    Logging to 172.22.22.20.162
SNMP agent configured in VRFs: default
SNMP agent enabled in default VRF
switch>
```

show snmp chassis

The **show snmp chassis** command displays the Simple Network Management Protocol (SNMP) server serial number or the chassis ID string configured by the **snmp-server chassis-id** command.

Example

EXEC

Command Syntax

```
show snmp chassis
```

Example

- This command displays the chassis ID string.

```
switch>show snmp chassis
Chassis: JFL08320162
switch>
```

show snmp community

The **show snmp community** command displays the Simple Network Management Protocol (SNMP) community access strings configured by the **snmp-server community** command.

Example

EXEC

Command Syntax

show snmp community

Example

- This command displays the list of community access strings configured on the switch.

```
switch>show snmp community

Community name: public
switch>
```

show snmp contact

The **show snmp contact** command displays the Simple Network Management Protocol (SNMP) system contact string configured by the **snmp-server contact** command. The command has no effect if a contact string was not previously configured.

Example

EXEC

Command Syntax

show snmp contact

Example

- This command displays the contact string contents.

```
switch>show snmp contact
Contact: John Smith
switch>
```

show snmp engineID

The **show snmp engineID** command displays the local SNMP engine information configured on the switch.

Example

EXEC

Command Syntax

show snmp engineID

Example

- This command displays the ID of the local SNMP engine.

```
switch>show snmp engineid
Local SNMP EngineID: f5717f001c730436d700
switch>
```

show snmp group

The **show snmp group** command shows the names of configured SNMP groups along with the security model, and view status of each group.

Example

EXEC

Command Syntax

show snmp group [*GROUP_LIST*]

Parameters

- ***GROUP_LIST*** the name of the group.
  - <no parameter> displays information about all groups.
  - *group_name* the name of the group.

Field Descriptions

- **groupname** name of the SNMP group.
- **security model** security model used by the group: ***v1***, ***v2c***, or ***v3***.
- **readview** string identifying the group's read view. Refer to **show snmp view**.
- **writeview** string identifying the group's write view.
- **notifyview** string identifying the group's notify view.

Example

- This command displays the groups configured on the switch.

```
switch>show snmp group
groupname : normal                          security model:v3 priv
readview  : all                             writeview: <no writeview specified>
notifyview: <no notifyview specified>

switch>
```

show snmp host

The **show snmp host** command displays information for Simple Network Management Protocol notification. Details include IP address and port number of the Network Management System, notification type, and SNMP version.

Example

EXEC

Command Syntax

show snmp host

Field Descriptions

- **Notification host** IP address of the host.
- **udp-port** port number.
- **type** notification type.
- **user** access type of the user.
- **security model** SNMP version used.
- **traps** details of the notification.

Example

- This command displays the hosts configured on the switch.

```
switch>show snmp host
Notification host: 172.22.22.20    udp-port: 162   type: trap
user: public                       security model: v2c

switch>
```

show snmp location

The **show snmp location** command displays the Simple Network Management Protocol (SNMP) system location string. The **snmp-server location** command configures system location details. The command has no effect if a location string was not previously configured.

Example

EXEC

Command Syntax

show snmp location

Example

- This command displays the location string contents.

```
switch>show snmp location
Location: santa clara
switch>
```

show snmp mib

The **show snmp mib** command displays values associated with specified MIB object identifiers (OIDs) that are registered on the switch.

Example

EXEC

Command Syntax

show snmp mib *OBJECTS*

Parameters

- **OBJECTS** object identifiers for which the command returns data. Options include:
  - **get** *oid_1* [*oid_2 ... oid_x*] values associated with each listed OID.
  - **get-next** *oid_1* [*oid_2 ... oid_x*] values associated with subsequent OIDs relative to listed OIDs.
  - **table** *oid* table associated with specified OID.
  - **translate** *oid* object name associated with specified OID.
  - **walk** *oid* objects below the specified subtree.

Example

- This command uses the get option to retrieve information about the sysORID.1 OID.

```
switch#show snmp mib get sysORID.1
SNMPv2-MIB::sysORID[1] = OID: TCP-MIB::tcpMIB
```

- This commnd uses the get-next option to retrieve information about the OID that is after sysORID.8.

```
switch#show snmp mib get-next sysORID.8
SNMPv2-MIB::sysORDescr[1] = STRING: The MIB module for managing TCP
implementations
```

show snmp source-interface

The **show snmp source-interface** command displays the interface whose IP address is the source address for SNMP traps.

Example

EXEC

Command Syntax

```
show snmp source-interface
```

Example

- This command displays the source interface for the SNMP notifications.

```
switch>show snmp source-interface
SNMP source interface: Ethernet1
switch>
```

show snmp trap

The **show snmp trap** command displays the SNMP trap generation information.

Example

EXEC

Command Syntax

show snmp trap

Example

- This command displays the SNMP traps configured on the switch.

```
switch>show snmp trap
  Type                         Name                                 Enabled
---------------------------- ------------------------------------ -------------
  entity                       entConfigChange                      Yes (default)
  entity                       entStateOperDisabled                 Yes (default)
  entity                       entStateOperEnabled                  Yes (default)
  lldp                         lldpRemTablesChange                  Yes (default)
  msdpBackwardTransition       msdpBackwardTransition               Yes
  msdpEstablished              msdpEstablished                      Yes
  snmp                         linkDown                             Yes
  snmp                         linkUp                               Yes
  snmpConfigManEvent           aristaConfigManEvent                 Yes (default)
  switchover                   aristaRedundancySwitchOverNotif      Yes
  test                         aristaTestNotification               Yes
switch>
```

show snmp user

The **show snmp user** command shows information about Simple Network Management Protocol (SNMP) users. Information that the command displays about each user includes their SNMP version, the engine ID of the host where they reside, and security information.

Example

EXEC

Command Syntax

show snmp user [*USER_LIST*]

Parameters

- ***USER_LIST*** the name of the group.
  - <no parameter> displays information about all users.
  - *user_name* specifies name of displayed user.

Example

- This command displays information about the users configured on the switch.

```
switch>show snmp user

User name: test

Security model: v3
Engine ID: f5717f001c73010e0900
Authentication protocol: SHA
Privacy protocol: AES-128
Group name: normal
switch>
```

show snmp view

The **show snmp view** command displays the information of a Simple Network Management Protocol configuration and the associated MIB. SNMP views are configured with the **snmp-server view** command.

Example

EXEC

Command Syntax

show snmp view [*VIEW_LIST*]

Parameters

- ***VIEW_LIST*** the name of the view.
  - <no parameter> displays information about all views.
  - *view_name* the name of the view.

Field Descriptions

- ***First column*** view name.
- ***Second column*** name of the MIB object or family.
- ***Third column*** inclusion level of the specified family within the view.

Example

- These commands configure an SNMP view, then displays that view.

```
switch(config)#snmp-server view sys-view system include
switch(config)#snmp-server view sys-view system.2 exclude
switch(config)#show snmp view
sys-view system - included
sys-view system.2 - excluded
```

snmp-server chassis-id

The **snmp-server chassis-id** command configures the chassis ID string. The default chassis ID string is the serial number of the switch. The **show snmp** command displays the chassis ID.

The **no snmp-server chassis-id** and **default snmp-server chassis-id** commands restore the default chassis ID string by removing the **snmp-server chassis-id** command from the configuration.

Command Mode

Global Configuration

Command Syntax

```
snmp-server chassis-id id_text
no snmp-server chassis-id
default snmp-server chassis-id
```

Parameters

- *id_text* chassis ID string

Example

- These commands configure *xyz-1234* as the chassis-id string, then display the result.

```
switch(config)#snmp-server chassis-id xyz-1234
switch(config)#show snmp
Chassis: xyz-1234                                    <---chassis ID
8 SNMP packets input
    0 Bad SNMP version errors
    0 Unknown community name
    0 Illegal operation for community name supplied
    0 Encoding errors
    8 Number of requested variables
    0 Number of altered variables
    4 Get-request PDUs
    4 Get-next PDUs
    0 Set-request PDUs
21 SNMP packets output
    0 Too big errors
    0 No such name errors
    0 Bad value errors
    0 General errors
    8 Response PDUs
    0 Trap PDUs
SNMP logging: enabled
    Logging to taccon.162
SNMP agent enabled
switch(config)#
```

snmp-server community

The **snmp-server community** command configures the community string. SNMP community strings serve as passwords that permit an SNMP manager to access the agent on the switch. The Network Management System (NMS) must define a community string that matches at least one of the switch community strings to access the switch.

The **no snmp-server community** and **default snmp-server community** commands remove the community access string from the configuration.

Command Mode

Global Configuration

Command Syntax

```
snmp-server community string_text [MIB_VIEW][ACCESS][ACL_NAMES]
no snmp-server community string_text
default snmp-server community string_text
```

Parameters

- *string_text* community access string.
- **MIB_VIEW** community access availability. Options include:
  - <no parameter> community string allows access to all objects.
  - **view** *view_name* community string allows access only to objects in the *view_name* view.
- **ACCESS** community access availability. Options include:
  - <no parameter> read-only access (default setting).
  - **ro** read-only access.
  - **rw** read-write access.
- **ACL_NAMES** community access availability. Options include:
  - <no parameter> community string allows access to all objects.
  - *list_v4* IPv4 ACL list.
  - **ipv6** *list_v6* IPv6 ACL list.
  - **ipv6** *list_v6 list_v4* IPv4 and IPv6 ACL list.

Example

- This command adds the community string lab_1 to provide read-only access to the switch agent.

```
switch(config)#snmp-server community lab_1 ro
switch(config)#
```

snmp-server contact

The **snmp-server contact** command configures the system contact string. The contact is displayed by the **show snmp** and **show snmp contact** commands.

The **no snmp-server contact** and **default snmp-server contact** commands remove the **snmp-server contact** command from the configuration.

Command Mode

Global Configuration

Command Syntax

```
snmp-server contact contact_string
no snmp-server contact
default snmp-server contact
```

Parameters

- *contact_string* system contact string.

Example

- These commands configure *Bonnie H* as the contact string, then display the result.

```
switch(config)#snmp-server contact Bonnie H
switch(config)#show snmp
Chassis: xyz-1234
Contact: Bonnie H.
8 SNMP packets input
    0 Bad SNMP version errors
    0 Unknown community name
    0 Illegal operation for community name supplied
    0 Encoding errors
    8 Number of requested variables
    0 Number of altered variables
    4 Get-request PDUs
    4 Get-next PDUs
    0 Set-request PDUs
24 SNMP packets output
    0 Too big errors
    0 No such name errors
    0 Bad value errors
    0 General errors
    8 Response PDUs
    0 Trap PDUs
SNMP logging: enabled
    Logging to taccon.162
SNMP agent enabled
switch(config)#
```

snmp-server enable traps

The **snmp-server enable traps** command enables both Simple Network Management Protocol (SNMP) traps and SNMP inform requests; use the **snmp-server host** command to specify which will receive SNMP notifications. Sending notifications requires at least one **snmp-server host** command.

The **snmp-server enable traps** and **no snmp-server enable traps** commands, without a MIB parameter, specify the default notification trap generation setting for all MIBs. These commands, when specifying a MIB, control notification generation for the specified MIB. The **default snmp-server enable traps** command resets notification generation to the default setting for the specified MIB.

Command Mode

Global Configuration

Command Syntax

```
snmp-server enable traps [trap_type]
no snmp-server enable traps [trap_type]
default snmp-server enable traps [trap_type]
```

Parameters

- ***trap_type*** controls the generation of informs or traps for the specified MIB:
  - <no parameter> controls notifications for MIBs not covered by specific commands.
  - **entity** controls entity-MIB modification notifications.
  - **lldp** controls LLDP notifications.
  - **msdpBackwardTransition** controls msdpBackwardTransition notifications.
  - **msdpEstablished** controls msdpEstablished notifications.
  - **snmp** controls SNMP-v2 notifications.
  - **switchover** controls switchover notifications.
  - **snmpConfigManEvent** controls snmpConfigManEvent notifications.
  - **test** controls test traps.

Examples

- These commands enables notification generation for all MIBs except spanning tree.

```
switch(config)#snmp-server enable traps
switch(config)#no snmp-server enable traps spanning-tree
switch(config)#
```

- This command enables spanning-tree MIB notification generation, regardless of the default setting.

```
switch(config)#snmp-server enable traps spanning-tree
switch(config)#
```

- This command resets the spanning-tree MIB notification generation to follow the default setting.

```
switch(config)#default snmp-server enable traps spanning-tree
switch(config)#
```

- This command enables switchover MIB notification generation, regardless of the default setting.

```
switch(config)#snmp-server enable traps switchover
switch(config)#
```

- This command resets the switchover MIB notification generation to follow the default setting.

```
switch(config)# default snmp-server enable traps switchover
switch(config)#
```

snmp-server engineID local

The **snmp-server engineID local** command configures the name for the local Simple Network Management Protocol (SNMP) engine. The default SNMP engineID is generated by the switch and is used when an engineID is not configured with this command. The **show snmp engineID** command displays the default or configured engine ID.

SNMPv3 authenticates users through security digests (MD5 or SHA) that are based on user passwords and the local engine ID. Passwords entered on the CLI are similarly converted, then compared to the user's security digest to authenticate the user.

Important! Changing the local engineID value invalidates SNMPv3 security digests, requiring the reconfiguration of all user passwords.

The **no snmp-server engineID** and **default snmp-server engineID** commands restore the default engineID by removing the **snmp-server engineID** command from the configuration**.**

Command Mode

Global Configuration

Command Syntax

```
snmp-server engineID local engine_hex
no snmp-server engineID local
default snmp-server engineID
```

Parameters

- *engine_hex* the switch's name for the local SNMP engine (hex string).

  The string must consist of at least ten characters with a maximum of 64 characters.

Example

- This command configures DC945798CAB4 as the name of the local SNMP engine.

```
switch(config)#snmp-server engineID local DC945798CAB4
switch(config)#
```

snmp-server engineID remote

The **snmp-server engineID remote** command configures the name of a Simple Network Management Protocol (SNMP) engine located on a remote device. The switch generates a default engineID; use the **show snmp engineID** command to view the configured or default engineID.

An SNMPv3 inform requires a remote engine ID to compute the security digest that authenticates and encrypts data transmitted to remote users. SNMPv3 authenticates users with MD5 or SHA through the engine ID and user passwords. CLI passwords are similarly authenticated.

Important! Changing the engineID value invalidates SNMPv3 security digests, requiring the reconfiguration of all user passwords.

The **no snmp-server engineID remote** and **default snmp-server engineID remote** commands remove the **snmp-server engineID remote** command from the configuration**.**

Command Mode

Global Configuration

Command Syntax

```
snmp-server engineID remote engine_addr [PORT] engine_hex
no snmp-server engineID remote engine_addr [PORT]
default snmp-server engineID remote engine_addr [PORT]
```

Parameters

- *engine_addr* location of remote engine (IP address or host name).
- ***PORT*** udp port location of the remote engine. Options include:
  - <No parameter> port number 161 (default).
  - **udp-port** *port_num* port number. Ranges from 0 to 65535.
- *engine_hex* the switch's name for the remote SNMP engine (hex string).

  The string must have at least ten characters and can contain a maximum of 64 characters.

Example

- This command configures DC945798CA as the engineID of the remote SNMP engine located at 10.23.10.25, UDP port 162.

```
switch(config)#snmp-server engineID remote 10.23.10.25 udp-port 162 DC945798CA
switch(config)#
```

snmp-server extension

The **snmp-server extension** command configures the execution of user supplied scripts to service portions of the OID space.

The **no snmp-server extension** and **default snmp-server extension** commands deletes the **snmp-server extension** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

snmp-server extension *OID_space* ***FILE_PATH*** [***DURATION***]

Parameters

- *OID_space* OID branch serviced by the script, in numerical format.
- ***FILE_PATH*** path and name of the script file. Options include:
  - **file:** file is located in the switch file directory.
  - **flash:** file is located in flash memory.
- ***DURATION*** the execution scope of the script.
  - <no parameter> script runs after initial request to process subsequent requests.
  - **one-shot** script processes a single object (runs once), then terminates.

Examples

- This command specifies the file ***example.sh***, located in flash, as the script file that services the listed OID space.

```
switch(config)#snmp-server extension .1.3.6.1.4.1.8072.2 flash:example.sh
```

snmp-server group

The **snmp-server group** command configures a new Simple Network Management Protocol (SNMP) group or modifies an existing group. An SNMP group is a data structure that user statements reference to map SNMP users to SNMP contexts and views, providing a common access policy to the specified users.

An SNMP context is a collection of management information items accessible by an SNMP entity. Each item of may exist in multiple contexts. Each SNMP entity can access multiple contexts. A context is identified by the EngineID of the hosting device and a context name.

The **no snmp-server group** and **default snmp-server group** commands delete the specified group by removing the corresponding **snmp-server group** command from the configuration.

Command Mode

Global Configuration

Command Syntax

```
snmp-server group group_name VERSION [CNTX] [READ] [WRITE] [NOTIFY]
no snmp-server group group_name VERSION
default snmp-server group group_name VERSION
```

Parameters

- *group_name* the name of the group.
- ***VERSION*** the security model utilized by the group.
  - **v1** SNMPv1. Uses a community string match for authentication.
  - **v2c** SNMPv2c. Uses a community string match for authentication.
  - **v3 no auth** SNMPv3. Uses a username match for authentication.
  - **v3 auth** SNMPv3. HMAC-MD5 or HMAC-SHA authentication.
  - **v3 priv** SNMPv3. HMAC-MD5 or HMAC-SHA authentication. AES or DES encryption.
- ***CNTX*** associates the SNMP group to an SNMP context.
  - <no parameter> command does not associate group with an SNMP context.
  - **context** *context_name* associates group with context specified by *context_name.*
- ***READ*** specifies read view for SNMP group.
  - <no parameter> command does not specify read view.
  - **read** *read_name* read view specified by *read_name (*string – maximum 64 characters).
- ***WRITE*** specifies write view for SNMP group.
  - <no parameter> command does not specify write view.
  - **write** *write_name* write view specified by *write_name (*string – maximum 64 characters).
- ***NOTIFY*** specifies notify view for SNMP group.
  - <no parameter> command does not specify notify view.
  - **notify** *notify_name* notify view specified by *notify_name (*string – maximum 64 characters).

Example

- This command configures normal_one as SNMP version 3 group (authentication and encryption) that provides access to the ***all-items*** read view.

```
switch(config)#snmp-server group normal_one v3 priv read all-items
switch(config)#
```

snmp-server host

The **snmp-server host** command command configures an SNMP host (to which SNMP traps will be sent) and sets the community string if it was not previously configured. The host is denoted by host location and community string. The command also specifies the type of SNMP notifications that are sent: a ***trap*** is an unsolicited notification; an ***inform*** is a trap that includes a request for a confirmation that the message is received.

The configuration can contain multiple statements to the same host location with different community strings. For instance, a configuration can simultaneously contain all of the following:

- **snmp-server host host-1 version 2c comm-1**
- **snmp-server host host-1 informs version 2c comm-2**
- **snmp-server host host-1 version 2c comm-3 udp-port 666**
- **snmp-server host host-1 version 3 auth comm-3**

The **no snmp-server host** and **default snmp-server host** commands remove the specified host by deleting the corresponding **snmp-server host** statement from the configuration. When removing a statement, the host (address and port) and community string must be specified.

Command Mode

Global Configuration

Command Syntax

```
snmp-server host host_id [VRF_INST][MESSAGE][VERSION] comm_str [PORT]
no snmp-server host host_id [VRF_INST][MESSAGE][VERSION] comm_str [PORT]
default snmp-server host host_id [VRF_INST][MESSAGE][VERSION] comm_str [PORT]
```

Parameters

- *host_id* hostname or IP address of the SNMP host.
- ***VRF_INST*** specifies the VRF instance being modified.
  - <no parameter> changes are made to the default VRF.
  - **vrf** *vrf_name* changes are made to the specified user-defined VRF.
- ***MESSAGE*** message type that is sent to the host.
  - <no parameter> sends SNMP traps to host (default).
  - **informs** sends SNMP informs to host.
  - **traps** sends SNMP traps to host.
- ***VERSION*** SNMP version. Options include:
  - <no parameter> SNMPv2c (default).
  - **version 1** SNMPv1; option not available with informs.
  - **version 2c** SNMPv2c.
  - **version 3 noauth** SNMPv3; enables user-name match authentication.
  - **version 3 auth** SNMPv3; enables MD5 and SHA packet authentication.
  - **version 3 priv** SNMPv3. HMAC-MD5 or HMAC-SHA authentication. AES or DES encryption.
- *comm_str* community string to be sent with the notification as a password.

  Arista recommends setting this string separately before issuing the **snmp-server host** command. To set the community string separately, use the **snmp-server community** command.
- ***PORT*** port number of the host.

- <no parameter> socket number set to 162 (default)
- **udp-port** *p-name* socket number specified by *p-name*

Guidelines

The switch can only send SNMP traps and informs if the host that has been configured to receive them is accessible through an interface in a VRF in which SNMP has been enabled. SNMP is enabled by default only in the **default** VRF. Enable or disable SNMP in a VRF with the **snmp-server vrf** command.

Example

- This command adds a version 2c inform notification recipient.

```
switch(config)#snmp-server host 10.15.2.3 informs version 2c comm-1
switch(config)#
```

snmp-server location

The **snmp-server location** command configures the system location string. By default, no system location string is set.

The **no snmp-server location** and **default snmp-server location** commands delete the location string by removing the **snmp-server location** command from the configuration.

Command Mode

Global Configuration

Command Syntax

```
snmp-server location node_locate
no snmp-server location
default snmp-server location
```

Parameters

- *node_locate* system location information (string).

Example

- These commands configure *lab-east* as the location string.

```
switch(config)#snmp-server location lab_east
```

snmp-server source-interface

The **snmp-server source-interface** command specifies the interface where SNMP originates informs and traps.

The **no snmp-server source-interface** and **default snmp-server source-interface** commands remove the inform or trap source assignment by removing the **snmp-server source-interface** command from **running-config**.

Command Mode

Global Configuration

Command Syntax

```
snmp-server source-interface INTERFACE
no snmp-server source-interface
default snmp-server source-interface
```

Parameters

- ***INTERFACE*** Interface type and number. Values include:
  - **ethernet** *e_num* Ethernet interface specified by *e_num*.
  - **loopback** *l_num* Loopback interface specified by *l_num*.
  - **management** *m_num* Management interface specified by *m_num*.
  - **port-channel** *p_num* Port-Channel Interface specified by *p_num*.
  - **vlan** *v_num* VLAN interface specified by *v_num*.

Example

- This command configures the Ethernet 1 interface as the source of SNMP traps and informs.

```
switch(config)#snmp-server source-interface ethernet 1
```

snmp-server user

The **snmp-server user** command adds a user to a Simple Network Management Protocol (SNMP) group or modifies an existing user's parameters.

To configure a user, the IP address or port number of the device where the user's remote SNMP agent resides must be specified. A user's authentication come from the engine ID and the user's password. Remote user configuration commands fail if the remote engine ID is not configured first.

The **no snmp-server user** and **default snmp-server user** commands remove the user from an SNMP group by removing the user command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
snmp-server user user_name group_name [AGENT] VERSION [ENGINE][SECURITY]
no snmp-server user user_name group_name [AGENT] VERSION
default snmp-server user user_name group_name [AGENT] VERSION
```

Parameters

- *user_name* name of user.
- *group_name* name of group to which user is being added.
- *AGENT* Options include:
  - <no parameter> local SNMP agent.
  - **remote** *addr* [**udp-port** *p_num*] remote SNMP agent location.

    addr denotes the IP address*; p_num* denotes the udp port socket*.* (default port is 162).
- ***VERSION*** SNMP version; options include:
  - **v1** SNMPv1.
  - **v2c** SNMPv2c.
  - **v3** SNMPv3 .
- ***ENGINE*** engine ID used to localize passwords. Available only if ***VERSION*** is **v3**.
  - <no parameter> Passwords localized by SNMP copy specified by *agent*.
  - **localized** *engineID* octet string of engineID.
- ***SECURITY*** Specifies authentication and encryption levels. Available only if ***VERSION*** is **v3**. Encryption is available only when authentication is configured.
  - <no parameter> no authentication or encryption.
  - **auth** ***a_meth*** *a_pass* [**priv** ***e_meth*** *e_pass*] authentication parameters.

    a-meth authentication method: options are **md5** (HMAC-MD5-96) and **sha** (HMAC-SHA-96).

    a-pass authentication string for users receiving packets.

    e-meth encryption method: Options are **aes** (AES-128) and **des** (CBC-DES).

    e-pass encryption string for the users sending packets.

Example

- This command configures the remote SNMP user ***tech-1*** to the ***tech-sup*** SNMP group.

```
switch(config)#snmp-server user tech-1 tech-sup remote 10.1.1.2 v3
```

snmp-server view

The **snmp-server view** command defines a view.

An SNMP view defines a subset of objects from an MIB. Every SNMP access group specifies views, each associated with read or write access rights, to allow or limit the group's access to MIB objects.

The **no snmp-server view** command deletes a view entry by removing the corresponding **snmp-server view** command from the ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
snmp-server view view_name family_name INCLUSION
no snmp-server view view_name [family_name]
snmp-server view view_name [family_name]
```

Parameters

- *view_name* Label for the view record that the command updates. Other commands reference the view with this label.
- *family_name* name of the MIB object or family.

  MIB objects and MIB subtrees can be identified by name or by the numbers representing the position of the object or subtree in the MIB hierarchy.
- ***INCLUSION*** inclusion level of the specified family within the view. Options include:
  - **include** view includes the specified subtree.
  - **exclude** view excludes the specified subtree.

Example

- These commands create a view named ***sys-view*** that includes all objects in the ***system*** subtree except for those in ***system.2***.

```
switch(config)#snmp-server view sys-view system include
switch(config)#snmp-server view sys-view system.2 exclude
```

snmp-server vrf

The **snmp-server vrf** command enables SNMP in the specified VRF. By default, SNMP is enabled only in **default** VRF.

- User-defined VRFs: The **no snmp-server vrf** command disables SNMP in the specified VRF by removing the corresponding **snmp-server vrf** command from the ***running-config***.
- Default VRF: The **no snmp-server vrf** command disables SNMP in the VRF by adding a **no snmp-server vrf default** statement to ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
snmp-server vrf vrf_name
no snmp-server vrf vrf_name
default snmp-server vrf vrf_name
```

Parameters

- *vrf_name* The VRF in which SNMP is enabled. The keyword **default** specifies the default VRF.

Guidelines

The switch can only send SNMP traps and informs if the host that has been configured to receive them is accessible through an interface in a VRF in which SNMP has been enabled. SNMP is enabled by default only in the **default** VRF. Enable or disable SNMP in a VRF with the **snmp-server vrf** command.

Example

- These commands disable SNMP in the default VRF, then enable it in the user-defined VRFs named “magenta” and “columbia.”

```
switch(config)#no snmp-server vrf default
switch(config)#snmp-server vrf magenta
switch(config)#snmp-server vrf columbia
switch(config)#
```

snmp trap link-status

The **snmp trap link-status** command enables Simple Network Management Protocol (SNMP) link-status trap generation on the configuration mode interface. The generation of link-status traps is enabled by default. If SNMP link-trap generation was previously disabled, this command removes the corresponding **no snmp link-status** statement from the configuration to re-enable link-trap generation.

The **no snmp trap link-status** command disables SNMP link trap generation on the configuration mode interface.

The **snmp trap link-status** and **default snmp trap link-status** commands restore the default behavior by removing the **no snmp trap link-status** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration
Interface-Loopback Configuration
Interface-Management Configuration
Interface-Port-channel Configuration
Interface-VLAN Configuration
Interface-VXLAN Configuration

Command Syntax

```
snmp trap link-status
no snmp trap link-status
default snmp trap link-status
```

Guidelines

The switch can only send SNMP traps and informs if the host that has been configured to receive them is accessible through an interface in a VRF in which SNMP has been enabled. SNMP is enabled by default only in the **default** VRF. Enable or disable SNMP in a VRF with the **snmp-server vrf** command.

Example

- This command disables SNMP link trap generation on the Ethernet 5 interface.

```
switch(config-if-Et5)#no snmp trap link-status
switch(config-if-Et5)#
```

Chapter 40

Latency Analyzer (LANZ)

Arista Networks' Latency Analyzer (LANZ) is a family of EOS features that provide enhanced visibility into network dynamics, particularly in areas related to the delay packets experience through the network. The LANZ feature is available on the FM6000, Petra, Arad, Trident II, and Tomahawk switch platforms.

This chapter describes the purpose, behavior, and configuration of LANZ features. Topics covered by this chapter include:

- Section 40.1: Introduction to LANZ
- Section 40.2: LANZ Overview
- Section 40.3: Configuring LANZ
- Section 40.4: LANZ Commands

40.1 Introduction to LANZ

LANZ tracks interface congestion and queuing latency with real-time reporting. With LANZ application layer event export, external applications can predict impending congestion and latency. This enables the application layer to make traffic routing decisions with visibility into the network layer.

With LANZ, network operations teams and administrators have near real-time visibility into the network, enabling early detection of microbursts. LANZ continually monitors congestion, allowing for rapid detection of congestion and sending of application layer messages.

40.2 LANZ Overview

LANZ monitors output queue lengths to provide congestion information for individual interfaces. This allows for more detailed analysis of congestion events, and allows identification of potential latency problems before they arise. On some platforms, LANZ also monitors global buffer usage.

Output queues for each port are monitored, and information about queue congestion events can be accessed in the form of syslog messages, reports, or streaming.

40.2.1 LANZ Monitoring Mechanism

LANZ provides congestion data by continuously monitoring each port's output queue lengths. When the length of an output queue exceeds the upper threshold for that port, LANZ generates an over-threshold event. LANZ continues to report an over-threshold state every 800 microseconds until all queue lengths for that port pass below the lower threshold.

40.2.2 LANZ Logging

Over-threshold events generated by LANZ can be logged as syslog messages. Log messages are generated for events on all ports, at a maximum rate of one message per second per interface. The interval between messages can be configured globally.

Log messages indicate the time of the event, the interface affected, the threshold set for that interface, and the actual number of entries in the port's queue.

40.2.3 LANZ Reporting

Detailed LANZ data can be viewed through the CLI or exported as a CSV-formatted report.

A circular FIFO event buffer is dynamically shared by all interfaces. When an interface begins generating LANZ over-threshold events it can fill all available buffer space. However, each interface is guaranteed sufficient resources for a minimum of 500 entries.

40.2.4 LANZ Streaming

On some platforms, external client applications can also receive congestion event information as a data stream. The switch can stream LANZ data to up to 100 clients via TCP through port 50001. Streamed data is in Google protocol buffer format, and includes both over-threshold events and LANZ configuration information.

40.2.5 Platforms

The LANZ feature is available on the FM6000, Petra, Arad, Trident II, and Tomahawk switch platforms. To determine the switch platform from the CLI, enter **show platform ?** at the prompt.

Settings and capabilities differ slightly between the platforms:

- The Petra and Arad chips measure threshold values in bytes; FM6000, Trident II, and Tomahawk chips measure threshold values in segments.
- FM6000, Trident II, and Tomahawk chips allow configuration of both upper and lower threshold values.
- FM6000, Trident II, and Tomahawk chips support LANZ data streaming.
- Only FM6000 chips support global buffer monitoring.
- While the FM6000, Trident II, and Tomahawk chips monitor congestion events for all queues, the Petra and Arad chips only monitor the most congested queues.

40.3 Configuring LANZ

LANZ is disabled by default and must be enabled to function. Upper and lower queue-length thresholds can be defined for individual interfaces.

These sections describe the basic LANZ configuration steps:

- Section 40.3.1: Enabling and Disabling LANZ
- Section 40.3.2: Setting LANZ Congestion Thresholds
- Section 40.3.3: Setting LANZ Traffic Sampling
- Section 40.3.4: Logging LANZ Congestion Events
- Section 40.3.5: Viewing LANZ Data
- Section 40.3.6: Streaming LANZ Data

40.3.1 Enabling and Disabling LANZ

For the switch to collect and display latency information, LANZ must be enabled. The **queue-monitor length (global configuration mode)** command enables LANZ with the current settings, or with the default settings if none have been configured. LANZ is disabled by default.

When LANZ is enabled, the switch monitors queue lengths on all ports and queue length data is available in the following forms:

- syslog data (see **queue-monitor length log**)
- CLI display or CSV-format output (see **show queue-monitor length**)
- data stream (see **queue-monitor streaming**)

To disable LANZ globally, enter the **no queue-monitor length** command in global configuration mode. Disabling LANZ globally also discards LANZ log data, but retains settings. To disable LANZ on an individual interface, enter the **no queue-monitor length** command in interface ethernet configuration mode.

Examples

- This command enables LANZ on the switch.

```
switch(config)#queue-monitor length
```

- This command disables LANZ on the switch.

```
switch(config)#no queue-monitor length
```

- These commands disable LANZ on Ethernet interface 7.

```
switch(config)#interface ethernet 7
switch(config-if-Et7)#no queue-monitor length
```

40.3.2 Setting LANZ Congestion Thresholds

When LANZ is enabled on the switch, it generates over-threshold events when queue lengths on any monitored interface exceed the upper threshold value and continues generating them until all the queue lengths on that interface drop back below the lower threshold.

40.3.2.1 Congestion Thresholds on FM6000, Trident II, and Tomahawk Switches

Queue lengths are measured in 480-byte segments on FM6000 switches and in 208-byte segments on Trident II and Tomahawk switches. The default threshold values are 512 segments and 256 segments. To change the threshold values for a specific interface, use the **queue-monitor length thresholds (FM6000, Trident II, and Tomahawk)** command.

FM6000 switches can also monitor global buffer usage. Global buffers are measured in 160-byte segments; the default threshold values are 10940 segments and 4376 segments. To enable global buffer monitoring, use the **queue-monitor length global-buffer** command. To change the threshold values for global buffer usage monitoring on the switch, use the **queue-monitor length global-buffer thresholds** command.

Examples

- These commands set the upper and lower queue-length thresholds on Ethernet interface 5 to 300 segments and 200 segments.

```
switch(config)#interface ethernet 5
switch(config-if-Et5)#queue-monitor length thresholds 300 200
switch(config-if-Et5)#
```

- These commands enable global buffer monitoring on the switch and set the upper and lower thresholds to 9000 segments and 4000 segments.

```
switch(config)#queue-monitor length global-buffer
switch(config)#queue-monitor length global-buffer thresholds 9000 4000
switch(config)#
```

40.3.2.2 Congestion Thresholds on Petra and Arad Switches

Queue lengths are measured in bytes. The top threshold value can be between 2 and 52428800 bytes (the default value is 52428800 bytes). To change the upper threshold value for a specific interface, use the **queue-monitor length threshold (Arad and Petra)** command.

Example

- These commands set the upper queue-length threshold on Ethernet interface 5 to 2614400 bytes.

```
switch(config)#interface ethernet 5
switch(config-if-Et5)#queue-monitor length thresholds 2614400
switch(config-if-Et5)#
```

40.3.3 Setting LANZ Traffic Sampling

Traffic experiencing congestion can be configured to automatically send congested traffic to either the CPU or an Ethernet egress interface destination, once a queue threshold is crossed, by enabling LANZ mirroring through the command **queue-monitor length mirror**. The CPU or an egress interface mirror destination is then configured through the command **queue-monitor length mirror destination**. LANZ traffic sampling includes exporting congested traffic to a packet capture device or another tool for analysis, or directly to the switch CPU for inspection through the command **tcpdump queue-monitor**.

Example

- This command enables LANZ traffic sampling.

```
switch(config)#queue-monitor length mirror
switch(config)#
```

- This command disables LANZ traffic sampling.

```
switch(config)#no queue-monitor length mirror
switch(config)#
```

Examples

- This command configures LANZ traffic sampling for a CPU interface mirror destination.

```
switch(config)#queue-monitor length mirror destination cpu
switch(config)#
```

- This command configures LANZ traffic sampling for an Ethernet interface mirror destination for ports 1 through 5.

```
switch(config)#queue-monitor length mirror destination Ethernet 1-5
switch(config)#
```

- This command configures LANZ traffic sampling for an Ethernet interface mirror destination for ports 6, 10, and 12 through 14.

```
switch(config)#queue-monitor length mirror destination Ethernet 6,10,12-14
switch(config)#
```

Example

- This command inspects traffic on the switch.

```
switch(config)#tcpdump queue-monitor
tcpdump: WARNING: lanz: no IPv4 address assigned
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on lanz, link-type EN10MB (Ethernet), capture size 65535 bytes
...
0 packets captured
0 packets received by filter
0 packets dropped by kernel
switch(config)#
```

40.3.4 Logging LANZ Congestion Events

To generate syslog messages when queue lengths on an interface exceed its upper threshold, enable logging with the **queue-monitor length log** command. When logging is enabled, a log message is generated each time one or more queues on an interface exceed the upper threshold value for that interface (see **queue-monitor length threshold (Arad and Petra)** or **queue-monitor length thresholds (FM6000, Trident II, and Tomahawk)**). Once an interface is over threshold, additional messages are generated at a maximum rate of one per ***interval*** as long as the queue length remains above the lower threshold for that interface. No syslog message is generated when queue length drops back under threshold.

Queue length information is not included in log messages, but can be accessed by displaying LANZ data or exporting reports.

On FM6000 platforms, log messages can also be created whenever global buffer usage exceeds its upper threshold value (see **queue-monitor length global-buffer thresholds**). To enable global buffer monitoring, use the **queue-monitor length global-buffer** command. To log over-threshold events for the global buffer, use the **queue-monitor length global-buffer log** command.

Examples

- This command enables queue-length over-threshold logging with a minimum interval of 10 seconds between messages for a given interface.

```
switch(config)#queue-monitor length log 10
```

- This command disables queue-length over-threshold logging on the switch.

```
switch(config)#queue-monitor length log 0
```

- This is an example of a queue-length log message.

```
Oct 27 12:48:22 switch QUEUE_MONITOR-6-LENGTH_OVER_THRESHOLD: Interface
Ethernet6 queue length is over threshold of 512, current length is 1024.
```

- This command enables global buffer over-threshold logging on the switch with a minimum interval of 60 seconds between messages.

```
switch(config)#queue-monitor length global-buffer log 60
```

40.3.5 Viewing LANZ Data

LANZ status, and the data stored in the LANZ data buffer, can be viewed using the CLI. Output varies by switch platform, and can be limited to a specified number of records.

40.3.5.1 Viewing LANZ Data on Petra and Arad Platform Switches

When LANZ is enabled on a Petra or Arad platform switch, the **show queue-monitor length** command displays a report of recent over-threshold events for a range of interfaces or for all interfaces. By default, the command displays data for all interfaces, limited to the last 1000 records, with the most recent events listed first. To view a subset of the LANZ data, limited to a specified number of records, use the **show queue-monitor length limit** command.

Example

- This command displays the last 100 records for Ethernet interfaces 6 through 8.

```
switch#show queue-monitor length ethernet 6-8 limit 100
Report generated at 2010-01-01 12:56:13

Time                         Interface   Queue length (segments, 1 to 512 bytes)
-------------------------------------------------------------------------------
0:00:07.43393 ago            Et6         1049
0:00:39.22856 ago            Et7         2039
1 day, 4:33:23.12345 ago     Et6         1077
```

To view the current LANZ configuration for the switch and for each interface, use the **show queue-monitor length status** command.

Example

- This command displays LANZ configuration and status information.

```
switch(config)#show queue-monitor length status
Per-Interface Queue Length Monitoring
-------------------------------------
Queue length monitoring is enabled
Maximum queue length in bytes : 52428800
Port threshold in bytes:
Port     High threshold
Et3/1           5242880
Et3/2           5242880
Et3/3           5242880
Et3/4           5242880
Et3/5           5242880

                <-------OUTPUT OMITTED FROM EXAMPLE-------->
```

40.3.5.2 Viewing LANZ Data on FM6000, Trident II, and Tomahawk Platform Switches

When LANZ is enabled on an FM6000, Trident II, or Tomahawk platform switch, the **show queue-monitor length** command displays a report of recent over-threshold events for a range of interfaces or for all interfaces. By default, the command displays data for all interfaces, limited to the last 1000 records, with the most recent events listed first. To view a subset of the LANZ data, limited to a specified number of records, use the **show queue-monitor length limit** command.

Example

- This command displays the last 100 records for Ethernet interfaces 6 through 8.

```
switch#show queue-monitor length ethernet 6-8 limit 100
Report generated at 2010-01-01 12:56:13

Time                          Interface   Queue length (segments, 1 to 512 bytes)
------------------------------------------------------------------------------
0:00:07.43393 ago             Et6         1049
0:00:39.22856 ago             Et7         2039
1 day, 4:33:23.12345 ago      Et6         1077
```

To view the current LANZ configuration for the switch and for each interface, use the **show queue-monitor length status** command.

Example

- This command displays LANZ configuration and status information.

```
switch(config)#show queue-monitor length status
queue-monitor length enabled
Global Buffer Monitoring
------------------------
Global buffer monitoring is enabled
Segment size in bytes :   160
Total buffers in segments : 36864
High threshold : 10940
Low threshold :  4376


Per-Interface Queue Length Monitoring
-------------------------------------
Queue length monitoring is enabled
Segment size in bytes :   480
Maximum queue length in segments :  3647
Port thresholds in segments:
Port     High threshold  Low threshold
Et1                512            256
Et2                512            256
Et3                512            256
Et4                512            256
Et5                512            256
              <-------OUTPUT OMITTED FROM EXAMPLE-------->
```

To view all available LANZ records, use the **show queue-monitor length all** command.

Example

- This command displays all available LANZ records.

```
switch>show queue-monitor length all

Report generated at 2013-04-01 13:23:13
E-End, U-Update, S-Start, TC-Traffic Class
GH-High, GU-Update, GL-Low
Segment size for E, U and S congestion records is 480 bytes
Segment size for GL, GU and GH congestion records is 160 bytes
* Max queue length during period of congestion
+ Period of congestion exceeded counter
-------------------------------------------------------------------------------
-
Type    Time                    Intf    Congestion     Queue       Time of Max
                                (TC)    duration       length      Queue length
                                        (usecs)        (segments)  relative to
                                                                   congestion
                                                                   start
                                                                   (usecs)
-------------------------------------------------------------------------------
-
E  0:00:00.07567 ago         Et22(7)   >=71 mins      20*         30us
GU 0:00:00.15325 ago         N/A       N/A            5695        N/A
U  0:00:00.19859 ago         Et4(1)    N/A            5693        N/A
GU 0:00:00.95330 ago         N/A       N/A            5696        N/A
U  0:00:00.99859 ago         Et4(1)    N/A            5695        N/A
E  0:00:01.28821 ago         Et44(1)   9672us         2502*       7294us
S  0:00:01.17591 ago         Et22(7)   N/A            26          N/A
U  0:00:03.08248 ago         Et44(1)   N/A            50          N/A
S  12days,8:56:44.07567 ago  Et44(1)   N/A            20          N/A
switch>
```

On the FM6000, Trident II, and Tomahawk platforms, information is also available for the number of dropped packets (see **show queue-monitor length drops**) and transmission latency (see **show queue-monitor length tx-latency**); global buffer usage can also be viewed on FM6000 platforms (see **show queue-monitor length global-buffer**).

40.3.6 Streaming LANZ Data

To support analysis of latency conditions, the switch can be configured to stream LANZ congestion and configuration data. The switch streams LANZ data via TCP in Google protocol buffer format through port 50001 and through the management interface.

You must create a client application to receive the streaming data. By default, the switch will accept up to 10 client connections for streaming LANZ data. This limit can be configured up to a maximum of 100. Maximum connections can be configured when LANZ is disabled.

40.3.6.1 Enabling and Disabling LANZ Data Streaming

LANZ data streaming is disabled by default. To enable streaming, issue the **no shutdown (queue-monitor-streaming configuration)** command in queue-monitor streaming configuration mode. To disable streaming, use the **shutdown (queue-monitor-streaming configuration)** command.

When streaming is disabled, a message is sent to any connected clients and the connections are closed.

To ensure client access to LANZ data, add a rule to any relevant ACL permitting traffic destined for the LANZ port (50001) before initiating a client connection for streaming from a remote host. A static rule (sequence number 130) in the default control plane ACL permits LANZ traffic, but a similar rule must be added to any user-created ACL.

Examples

- These commands enable the streaming of LANZ data from the switch.

```
switch(config)#queue-monitor streaming
switch(config-qm-streaming)#no shutdown
switch(config-qm-streaming)#
```

- These commands disable LANZ data streaming.

```
switch(config)#queue-monitor streaming
switch(config-qm-streaming)#shutdown
switch(config-qm-streaming)#
```

40.3.6.2 Configuring Maximum Connections

By default, the switch will accept a maximum of 10 client connections for LANZ data streaming. This maximum can be configured using the **max-connections** command. If a client connects to the switch after the limit has been reached, an error message is sent and the connection is closed.

Example

- This command sets the maximum number of client connections for LANZ data streaming to 50.

```
switch(config-qm-streaming)#max-connections 50
```

40.3.6.3 LANZ Streaming Messages

When streaming is enabled, LANZ sends a message whenever a congestion event or a configuration event occurs. The messages are streamed in Google protocol buffer format.

Configuration Messages

A configuration message is sent whenever a change is made to the LANZ configuration settings on the switch. The switch also sends a configuration message when a new client connection is established.

The configuration message includes the following information:

- **timestamp** time of change in configuration in tens of microseconds (UTC).
- **lanzVersion** LANZ feature version.
- **numOfPorts** number of ports in the switch.
- **segmentSize** segment size.
- **maxQueueSize** maximum queue size in segments.
- **qLenInterval** frequency of updates.
- **intfName** name of the port.
- **switchId** ID of the chip on a multi-chip system.
- **portId** ID of the port.
- **internalPort** “true” if it is an internal port.
- **highThreshold** higher threshold value.
- **lowThreshold** lower threshold value.

Congestion Messages

A congestion message is sent whenever LANZ generates an over-threshold event.

The congestion message includes the following information:

- **timestamp** time of congestion in micro-seconds (UTC).
- **intfName** name of the port.
- **switchId** ID of the chip on a multi-chip system.
- **portId** ID of the port.
- **queueSize** queue size in segments at time of congestion.

40.3.6.4 Creating the LANZ Client

For a client device to receive streaming data from the LANZ server, it must be running a client application designed to receive LANZ data. Client programs must be based on the Google protocol buffer schema file describing the structure of the congestion and configuration messages which LANZ streams.

Google Protocol Buffers

Google protocol buffers provide an efficient mechanism for serializing LANZ data for streaming. A protocol buffer package is needed in order to run a LANZ client.

The latest version of the Google protocol buffer source code is available at this address: http://code.google.com/p/protobuf/downloads/list

LANZ Message Schema

LANZ client applications must be designed based on the LANZ protocol buffer schema, which defines the format and contents of the streamed messages. The schema file is shown below, and is also available on the Arista FTP site at this address:
https://www.arista.com/support/download/Extensions/Lanz.proto

```
package LanzProtobuf;

message ConfigRecord {
    required uint64 timestamp = 1; // Time of change in configuration in
    micro-seconds (UTC)
    required uint32 lanzVersion = 2; // LANZ feature version
    required uint32 numOfPorts = 3; // Num of ports in the switch
    required uint32 segmentSize = 4; // Segement size
    required uint32 maxQueueSize = 5; // Maximum queue size in segments
    optional uint32 qLenInterval = 10; // Frequency of update
    message PortConfigRecord {
        required string intfName = 1; // Name of the port
        required uint32 switchId = 2; // Id of the chip on a multi-chip system
        required uint32 portId = 3; // Id of the port
        required bool internalPort = 4; // 'True' if it's an internal port
        required uint32 highThreshold = 5; // Higher threshold
        required uint32 lowThreshold = 6; // Lower threshold
    }

    repeated PortConfigRecord portConfigRecord = 6; // Lanz config details of each
    port
}

message CongestionRecord {
    required uint64 timestamp = 1; // Time of congestion in micro-seconds (UTC)
    required string intfName = 2; // Name of the port
    required uint32 switchId = 3; // Id of the chip on a multi-chip system
    required uint32 portId = 4; // Id of the port
    required uint32 queueSize = 5; // Queue size in segments at time of congestion
}

message ErrorRecord {
    required uint64 timestamp = 1; // Time of event in micro-seconds (UTC)
    required string errorMessage = 2; // Text message
}

message LanzRecord {
    optional ConfigRecord configRecord = 1;
    optional CongestionRecord congestionRecord = 2;
    optional ErrorRecord errorRecord = 3;
}
```

Implementation Procedure

The following steps create and install a functional client to receive streamed LANZ data. This procedure assumes a functional Python programming environment.

Step 1 Download the example client from the Arista FTP server at this address:
https://www.arista.com/support/download/Extensions/lanz_client.py

Step 2 Decompress the GPB archive to a directory.

Step 3 Run the GPB C++ compilation and install. With default flags using GCC on *nix platforms, this will produce a binary called “protoc” in your /usr/local/bin directory.

Step 4 From the archive root, **cd** to python, and run the following commands:

python setup.py build

python setup.py test

Step 5 Next, use the protoc compiler to convert the Lanz.proto file into a Python program called Lanz_pb2.py, used by the client. The command to do so is:

protoc --python_out=. Lanz.proto

The **--python_out=.** flag drops the compiled Python program in the directory where you ran the command.

Step 6 Run **lanz_client.py -h** to activate the LANZ client.

40.4 LANZ Commands

LANZ Commands: Global Configuration

- clear queue-monitor length statistics
- queue-monitor length (global configuration mode)
- queue-monitor length global-buffer
- queue-monitor length global-buffer log
- queue-monitor length global-buffer thresholds
- queue-monitor length log
- queue-monitor length mirror
- queue-monitor length mirror destination
- queue-monitor streaming
- tcpdump queue-monitor

LANZ Commands: Interface Ethernet Configuration Mode

- queue-monitor length threshold (Arad and Petra)
- queue-monitor length thresholds (FM6000, Trident II, and Tomahawk)

LANZ Commands: Queue-Monitor Streaming Configuration Mode

- max-connections
- shutdown (queue-monitor-streaming configuration)

LANZ Display Commands

- show queue-monitor length
- show queue-monitor length all
- show queue-monitor length cpu
- show queue-monitor length csv
- show queue-monitor length drops
- show queue-monitor length ethernet
- show queue-monitor length global-buffer
- show queue-monitor length limit
- show queue-monitor length drops
- show queue-monitor length statistics
- show queue-monitor length status
- show queue-monitor length tx-latency

clear queue-monitor length statistics

The **clear queue-monitor length statistics** command resets the occurrences of all over-threshold events on the switch including global buffer information (if supported).

Command Mode

Privileged EXEC

Command Syntax

```
clear queue-monitor length statistics
```

Example

- This command resets all over-threshold events and global buffer information on an FM6000 switch.

```
switch#clear queue-monitor length statistics
switch#
```

max-connections

The **max-connections** command sets the maximum number of client connections the switch accepts for streaming LANZ data. The default maximum is 10 connections. To stream LANZ data, you must use the **queue-monitor streaming** command to enable LANZ data streaming.

Command Mode

Queue-Monitor-Streaming Configuration

Command Syntax

max-connections *connections*

Parameters

- *connections* maximum number of simultaneous LANZ streaming client connections the switch will accept. Values range from 1 through 100.

Related Commands

- **queue-monitor streaming** places the switch in queue-monitor-streaming configuration mode.

Examples

- This command sets the maximum number of client connections the switch accepts for LANZ data streaming to 50.

```
switch(config-qm-streaming)#max-connections 50
switch(config-qm-streaming)#
```

queue-monitor length (global configuration mode)

The **queue-monitor length (global configuration mode)** command enables LANZ with the current settings, or with the default settings if LANZ has not yet been configured. LANZ is disabled by default.

When LANZ is enabled, the switch monitors queue lengths on all ports and generates over-threshold events when an output queue becomes congested. Over-threshold event data is available in the following forms:

- syslog data (see **queue-monitor length log**)
- CLI display or CSV-format output (see **show queue-monitor length**)
- data stream (see **queue-monitor streaming**)

The **no queue-monitor length** and **default queue-monitor length** commands entered in global configuration mode disable LANZ and discard LANZ log data, but retain settings. LANZ settings include:

- logging settings (see **queue-monitor length log**)
- queue length thresholds (see **queue-monitor length threshold (Arad and Petra)** or **queue-monitor length thresholds (FM6000, Trident II, and Tomahawk)**)
- data streaming settings (see **queue-monitor streaming**)

Command Mode

Global Configuration

Command Syntax

```
queue-monitor length
no queue-monitor length
default queue-monitor length
```

Examples

- This command enables LANZ on the switch.

```
switch(config)#queue-monitor length
switch(config)#
```

- This command disables LANZ on the switch.

```
switch(config)#no queue-monitor length
switch(config)#
```

queue-monitor length threshold (Arad and Petra)

The **queue-monitor le**ngth threshold command sets the queue length threshold to define “congested” on the command-mode interface for purposes of LANZ reporting. If LANZ is enabled (see **queue-monitor length (global configuration mode)**), an over-threshold event is generated when one or more queues on the interface exceed the upper threshold, and over-threshold events continue to be generated until all queue lengths on the interface drop below the lower threshold. (To log these events, use the **queue-monitor length log** command.)

Entering the **no queue-monitor length** command in interface configuration mode disables LANZ on the interface. Entering either the **queue-monitor length threshold** command or the **default queue-monitor length threshold** command enables LANZ on the interface by removing the **no queue-monitor length** command from the configuration.

The **no queue-monitor length threshold** and **default queue-monitor length threshold** commands erase custom queue length threshold settings for the interface.

Command Mode

Interface-Ethernet Configuration

Command Syntax

```
queue-monitor length threshold upper_limit
no queue-monitor length
default queue-monitor length
```

Parameters

- *upper_limit* is the queue length in bytes that triggers an over-threshold event. Values range from 2 to 52428800 bytes. Default setting is 52428800.

Guidelines

Queue length is measured in bytes. Only the upper threshold is configurable, and it is set at a default value of 52428800 bytes.

Examples

- These commands set the upper queue-length threshold on Ethernet interface 3/30 to 40000000 bytes.

```
switch(config)#interface ethernet 3/30
switch(config-if-Et3/30)#queue-monitor length threshold 40000000
switch(config-if-Et3/30)#
```

- These commands reset the upper queue-length threshold on Ethernet interface 3/30 to its default value of 52428800 bytes.

```
switch(config)#interface ethernet 3/30
switch(config-if-Et3/30)#default queue-monitor length threshold
switch(config-if-Et3/30)#
```

queue-monitor length thresholds (FM6000, Trident II, and Tomahawk)

The **queue-monitor length thresholds** command sets queue length thresholds to define “congested” on the command-mode interface for purposes of LANZ reporting. If LANZ is enabled (see **queue-monitor length (global configuration mode)**), an over-threshold event is generated when one or more queues on the interface exceed the upper threshold, and over-threshold events continue to be generated until all queue lengths on the interface drop below the lower threshold. (To log these events, use the **queue-monitor length log** command.)

Entering the **no queue-monitor length** command in interface configuration mode disables LANZ on the interface. Entering either the **queue-monitor length** command or the **default queue-monitor length** command in interface configuration mode enables LANZ on the interface by removing the **no queue-monitor length** command from the configuration.

The **no queue-monitor length thresholds** and **default queue-monitor length thresholds** commands in interface configuration mode both erase custom queue length threshold settings for the interface.

Command Mode

Interface-Ethernet Configuration

Command Syntax

```
queue-monitor length thresholds upper_limit lower_limit
no queue-monitor length
default queue-monitor length
```

Parameters

- *upper_limit* queue length in segments that triggers an over-threshold event. Must be higher than *lower_limit*. The minimum value is 2. The maximum is the largest number of segments which can be queued before packets are dropped, and varies based on factors including flow control state and private buffer settings. Default setting is 512.
- *lower_limit* lower queue length threshold in segments. When logging is enabled, an over-threshold interface continues generating over-threshold events until all its queues drop back below this length. Must be lower than *upper_limit*. Values range from 1 to 4806. Default setting is 256.

Guidelines

Queue lengths are measured in 480-byte segments on FM6000 switches and in 208-byte segments on Trident II and Tomahawk switches. Default upper threshold is 512 segments and lower threshold is 256 segments. Both upper and lower thresholds are configurable.

Examples

- These commands set the upper and lower queue-length thresholds on Ethernet interface 5 to 300 segments and 200 segments.

```
switch(config)#interface ethernet 5
switch(config-if-Et5)#queue-monitor length thresholds 300 200
switch(config-if-Et5)#
```

- These commands reset the upper and lower queue-length thresholds on Ethernet interface 5 to their default values.

```
switch(config)#interface ethernet 5
switch(config-if-Et5)#default queue-monitor length thresholds
switch(config-if-Et5)#
```

- These commands disable LANZ on Ethernet interface 5.

```
switch(config)#interface ethernet 5
switch(config-if-Et5)#no queue-monitor length
switch(config-if-Et5)#
```

queue-monitor length global-buffer

The **queue-monitor length global-buffer** command includes global buffer usage in LANZ reporting.

When global buffer reporting is enabled, over-threshold events are generated when global buffer usage exceeds the upper threshold. To set the threshold value, use the **queue-monitor length global-buffer thresholds** command. Usage data may be viewed using the **show queue-monitor length global-buffer** command. To view status and threshold information, use the **show queue-monitor length status** command.

Global buffer usage is measured in segments of 160 bytes.

The **no queue-monitor length global-buffer** and **default queue-monitor length global-buffer** commands disable global buffer usage reporting by removing the corresponding queue-monitor length global-buffer command from running-config.

Command Mode

Global Configuration

Command Syntax

```
queue-monitor length global-buffer
no queue-monitor length global-buffer
default queue-monitor length global-buffer
```

Guidelines

This command is available on FM6000 platform switches.

Examples

- This command enables global buffer monitoring on the switch.

```
switch(config)#queue-monitor length global-buffer
switch(config)#
```

- This command disables global buffer monitoring on the switch.

```
switch(config)#no queue-monitor length global-buffer
switch(config)#
```

queue-monitor length global-buffer log

The **queue-monitor length global-buffer log** command enables logging of global buffer over-threshold events. When logging is enabled, a log message is generated each time the contents of the global buffer exceed the upper threshold value set for the switch (see **queue-monitor length global-buffer thresholds**). Once the global buffer is over the threshold, additional messages are generated at a maximum rate of one per ***interval*** as long as the buffer value remains above the lower threshold for the switch.

Global buffer logging is disabled by default.

Log messages do not include buffer usage or congestion information. To view this information, use the **show queue-monitor length global-buffer** command.

The **no queue-monitor length global-buffer log** and **default queue-monitor length global-buffer log** commands disable global buffer logging by removing the corresponding **queue-monitor length global-buffer log** command from ***running-config.*** The **queue-monitor length global-buffer log** command with an interval value of 0 also disables global buffer logging.

Command Mode

Global Configuration

Command Syntax

```
queue-monitor length global-buffer log interval
no queue-monitor length global-buffer log
default queue-monitor length global-buffer log
```

Parameters

- *interval* minimum interval in seconds between logged messages.
  - 0 global buffer logging is disabled on the switch (the default setting).
  - **1 to 65535** minimum logging interval (in seconds).

Guidelines

This command is available on FM6000 platform switches.

Examples

- This command enables global buffer logging with a minimum interval of 10 seconds between messages.

  ```
  switch(config)#queue-monitor length global-buffer log 10
  ```

- This command disables global buffer logging on the switch.

  ```
  switch(config)#no queue-monitor length global-buffer log
  ```

queue-monitor length global-buffer thresholds

The **queue-monitor length global-buffer thresholds** command sets global buffer thresholds for the switch. An over-threshold event is generated when usage of the global buffer exceeds the upper threshold, and over-threshold events continue to be generated until usage drops below the lower threshold. (To log these events, use the **queue-monitor length global-buffer log** command.)

The **no queue-monitor length global-buffer** and **default queue-monitor length global-buffer** commands disable global buffer reporting.

The **no queue-monitor length global-buffer thresholds** and **default queue-monitor length global-buffer thresholds** commands erase custom global buffer threshold settings.

Command Mode

Global Configuration

Command Syntax

```
queue-monitor length global-buffer thresholds max_segments min_segments
no queue-monitor length global-buffer log
default queue-monitor length global-buffer log
```

Parameters

- *max_segments* upper threshold in 160-byte segments. Value ranges from 2 to 36864. Default is 10940.
- *min_segments* lower threshold in 160-byte segments. Value ranges from 1 to 36864. Default is 4376.

Examples

- This command sets the upper and lower global buffer thresholds to 9000 segments and 3000 segments.

```
switch(config)#queue-monitor length global-buffer thresholds 9000 3000
switch(config)#
```

- This command resets the upper and lower global buffer thresholds to their default values.

```
switch(config)#no queue-monitor length global-buffer thresholds 9000 3000
switch(config)#
```

queue-monitor length log

The **queue-monitor length log** command enables logging of queue-length over-threshold events when LANZ is enabled on the switch (see **queue-monitor length (global configuration mode)**). When logging is enabled, a log message is generated each time one or more queues on an interface exceed the upper threshold value for that interface (see **queue-monitor length threshold (Arad and Petra)**). Once an interface is over threshold, additional messages are generated at a maximum rate of one per ***interval*** as long as the queue length remains above the lower threshold for that interface. No syslog message is generated when queue length drops back under threshold.

Logging is disabled by default.

Log messages do not include queue length information. To view queue length information, use the **show queue-monitor length** command.

The **queue-monitor length log** command with an interval value of 0 disables event logging.

Command Mode

Global Configuration

Command Syntax

queue-monitor length log interval

Parameters

- *interval* minimum interval in seconds between logged messages from a single interface.
  - 0 queue-length logging is disabled on the switch (the default setting).
  - **1 to 65535** minimum logging interval (in seconds).

Examples

- This command enables over-threshold logging with a minimum interval of 10 seconds between messages for a given interface.

  ```
  switch(config)#queue-monitor length log 10
  ```

- This command disables queue-length over-threshold logging on the switch.

  ```
  switch(config)#queue-monitor length log 0
  ```

- This is an example of a queue-length log message.

  ```
  Oct 27 12:48:22 switch QUEUE_MONITOR-6-LENGTH_OVER_THRESHOLD: Interface
  Ethernet6 queue length is over threshold of 512, current length is 1024.
  ```

queue-monitor length mirror

The **queue-monitor length mirror** command enables LANZ mirroring. As a result, traffic experiencing congestion can be configured to automatically send congested traffic to either the CPU or an Ethernet egress interface destination, once a queue threshold is crossed (see **queue-monitor length mirror destination**).

Command Mode

Global Configuration

Command Syntax

queue-monitor length mirror

Example

- This command enables LANZ traffic sampling.

```
switch(config)#queue-monitor length mirror
switch(config)#
```

- This command disables LANZ traffic sampling.

```
switch(config)#no queue-monitor length mirror
switch(config)#
```

queue-monitor length mirror destination

The **queue-monitor length mirror destination** command results in automatically sending traffic experiencing congestion to either the CPU or an Ethernet egress interface destination, once a queue threshold is crossed. Before using this command, first enable LANZ mirroring through the command **queue-monitor length mirror**.

Command Mode

Global Configuration

Command Syntax

queue-monitor length mirror destination cpu | Ethernet <ports 1-24>

Parameters

- *ports* any combination of Ethernet ports 1 through 24.

Examples

- This command configures LANZ traffic sampling for a CPU interface mirror destination.

```
switch(config)#queue-monitor length mirror destination cpu
switch(config)#
```

- This command configures LANZ traffic sampling for an Ethernet interface mirror destination for ports 3, 11, and 15 through 20.

```
switch(config)#queue-monitor length mirror destination Ethernet 3,11,15-20
switch(config)#
```

queue-monitor streaming

The **queue-monitor streaming** command places the switch in queue-monitor-streaming configuration mode. Queue-monitor-streaming configuration mode is not a group change mode; ***running-config*** is changed immediately upon command entry. The exit command does not affect ***running-config***.

To enable LANZ data streaming on the switch, use the **no shutdown (queue-monitor-streaming configuration)** command.

The **exit** command returns the switch to global configuration mode.

Command Mode

Global Configuration

Command Syntax

```
queue-monitor streaming
```

Commands Available in queue-monitor streaming Configuration Mode

- **max-connections**
- **shutdown (queue-monitor-streaming configuration)**

Example

- This command places the switch in queue-monitor streaming configuration mode.

```
switch(config)#queue-monitor streaming
switch(config-qm-streaming)#
```

show queue-monitor length

The **show queue-monitor length** command displays a report of recent over-threshold events for all interfaces, limited to the last 1000 records, with the newest events listed first.

LANZ must be enabled to use this command (see **queue-monitor length (global configuration mode)**). If LANZ is disabled, the command displays "queue-monitor is disabled."

To limit the output to a specified number of seconds and/or records, use the **show queue-monitor length limit** command.

Command Mode

EXEC

Command Syntax

```
show queue-monitor length
```

Example

- This command displays the last 1000 LANZ records on a Petra or Arad platform switch.

```
switch>show queue-monitor length
Report generated at 2010-01-01 12:56:13
Time                          Interface   Queue length (segments, 1 to 512 bytes)
-------------------------------------------------------------------------------
0:00:07.43393 ago             Et6         1049
0:00:39.22856 ago             Et7         2039
1 day, 4:33:23.12345 ago      Et6         1077
switch>
```

- This command displays the last 1000 LANZ records on an FM6000, Trident II, or Tomahawk platform switch.

```
switch>show queue-monitor length
Report generated at 2017-03-10 14:57:12
E-End, U-Update, S-Start, TC-Traffic Class
Segment size for E, U and S congestion records is 208 bytes
* Max queue length during period of congestion
+ Period of congestion exceeded counter
-----------------------------------------------------------------------------------------------
Type    Time                  Interface      Congestion     Queue      Time of Max    Fabric
                              (TC)           duration       length     Queue length   Peer
                                             (usecs)        (segments) relative to
                                                                       congestion
                                                                       start
                                                                       (usecs)
-----------------------------------------------------------------------------------------------
 E   0:08:04.45352 ago        Et23/3(13)     22704753       5743*      0
 U   0:08:07.10807 ago        Et23/3(13)     N/A            5742       N/A
 U   0:08:12.10808 ago        Et23/3(13)     N/A            5742       N/A
 U   0:08:17.10809 ago        Et23/3(13)     N/A            5742       N/A
 U   0:08:22.10810 ago        Et23/3(13)     N/A            5742       N/A
 U   0:08:27.10810 ago        Et23/3(13)     N/A            5742       N/A
 U   0:08:27.11311 ago        Et23/3(13)     N/A            5741       N/A
 U   0:08:27.11811 ago        Et23/3(13)     N/A            5742       N/A
 U   0:08:27.12312 ago        Et23/3(13)     N/A            5742       N/A
 U   0:08:27.12812 ago        Et23/3(13)     N/A            5743       N/A
 U   0:08:27.13315 ago        Et23/3(13)     N/A            5743       N/A
 U   0:08:27.13816 ago        Et23/3(13)     N/A            5743       N/A
 U   0:08:27.14319 ago        Et23/3(13)     N/A            5742       N/A
 U   0:08:27.14822 ago        Et23/3(13)     N/A            5743       N/A
 U   0:08:27.15322 ago        Et23/3(13)     N/A            5742       N/A
 S   0:08:27.15828 ago        Et23/3(13)     N/A            2064       N/A
switch>
```

show queue-monitor length all

The **show queue-monitor length all** command displays all available over-threshold event records on the switch including global buffer information, with the most recent events listed first.

LANZ must be enabled to use this command (see **queue-monitor length (global configuration mode)**). If LANZ is disabled, the command displays “queue-monitor is disabled.”

Command Mode

EXEC

Command Syntax

```
show queue-monitor length all
```

Guidelines

This command is available on FM6000 platform switches.

Example

- This command displays all available LANZ records from the switch.

```
switch>show queue-monitor length all

Report generated at 2013-04-01 13:23:13
E-End, U-Update, S-Start, TC-Traffic Class
GH-High, GU-Update, GL-Low
Segment size for E, U and S congestion records is 480 bytes
Segment size for GL, GU and GH congestion records is 160 bytes
* Max queue length during period of congestion
+ Period of congestion exceeded counter
--------------------------------------------------------------------------------
-
Type    Time                    Intf    Congestion      Queue       Time of Max
                                (TC)    duration        length      Queue length
                                        (usecs)         (segments)  relative to
                                                                    congestion
                                                                    start
                                                                    (usecs)
--------------------------------------------------------------------------------
-
E  0:00:00.07567 ago         Et22(7)   >=71 mins       20*         30us
GU 0:00:00.15325 ago         N/A       N/A             5695        N/A
U  0:00:00.19859 ago         Et4(1)    N/A             5693        N/A
GU 0:00:00.95330 ago         N/A       N/A             5696        N/A
U  0:00:00.99859 ago         Et4(1)    N/A             5695        N/A
E  0:00:01.28821 ago         Et44(1)   9672us          2502*       7294us
S  0:00:01.17591 ago         Et22(7)   N/A             26          N/A
U  0:00:03.08248 ago         Et44(1)   N/A             50          N/A
S  12days,8:56:44.07567 ago  Et44(1)   N/A             20          N/A
switch>
```

show queue-monitor length cpu

The **show queue-monitor length cpu** command displays LANZ data for CPU ports on the switch. On Trident II and Tomahawk platforms, the “Interface” column identifies the CPU port by its card slot and chip index.

LANZ must be enabled to use this command (see **queue-monitor length (global configuration mode)**). If LANZ is disabled, the command displays “queue-monitor is disabled.”

Command Mode

EXEC

Command Syntax

show queue-monitor length cpu

Examples

- This command displays LANZ data for CPU ports on a Trident II or Tomahawk switch.

```
switch>show queue-monitor length cpu
Report generated at 2017-03-10 15:24:11
E-End, U-Update, S-Start, TC-Traffic Class
Segment size for E, U and S congestion records is 208 bytes
* Max queue length during period of congestion
+ Period of congestion exceeded counter
-------------------------------------------------------------------------------------------------
Type    Time                  Interface      Congestion     Queue      Time of Max    Fabric
                              (TC)           duration       length     Queue length   Peer
                                             (usecs)        (segments) relative to
                                                                       congestion
                                                                       start
                                                                       (usecs)
-------------------------------------------------------------------------------------------------
 E   0:02:24.19153 ago        Cpu0/0(39)     16669811       271*       0
 U   0:02:25.81126 ago        Cpu0/0(39)     N/A            270        N/A
 U   0:02:30.81126 ago        Cpu0/0(39)     N/A            270        N/A
 U   0:02:35.81128 ago        Cpu0/0(39)     N/A            270        N/A
 U   0:02:40.81129 ago        Cpu0/0(39)     N/A            270        N/A
 U   0:02:40.81630 ago        Cpu0/0(39)     N/A            270        N/A
 U   0:02:40.82130 ago        Cpu0/0(39)     N/A            270        N/A
 U   0:02:40.82631 ago        Cpu0/0(39)     N/A            270        N/A
 U   0:02:40.83132 ago        Cpu0/0(39)     N/A            270        N/A
 U   0:02:40.83632 ago        Cpu0/0(39)     N/A            270        N/A
 U   0:02:40.84132 ago        Cpu0/0(39)     N/A            270        N/A
 U   0:02:40.84633 ago        Cpu0/0(39)     N/A            270        N/A
 U   0:02:40.85133 ago        Cpu0/0(39)     N/A            270        N/A
 U   0:02:40.85634 ago        Cpu0/0(39)     N/A            270        N/A
 S   0:02:40.86134 ago        Cpu0/0(39)     N/A            271        N/A
```

- This command displays LANZ data for CPU ports on an FM6000 switch.

```
switch>show queue-monitor length cpu
Report generated at 2016-02-09 17:46:16
E-End, U-Update, S-Start, TC-Traffic Class
GH-High, GU-Update, GL-Low
Segment size for E, U and S congestion records is 480 bytes
Segment size for GL, GU and GH congestion records is 160 bytes
* Max queue length during period of congestion
+ Period of congestion exceeded counter
--------------------------------------------------------------------------------
-
Type    Time                  Intf    Congestion     Queue       Time of Max
                              (TC)    duration       length      Queue length
                                      (usecs)        (segments)  relative to
                                                                 congestion
                                                                 start
                                                                 (usecs)
--------------------------------------------------------------------------------
-
 U   0:00:03.41073 ago        Cpu(11)   N/A           205         N/A
 U   0:00:08.41093 ago        Cpu(11)   N/A           205         N/A
 U   0:00:13.41111 ago        Cpu(11)   N/A           205         N/A
```

show queue-monitor length csv

The **show queue-monitor length csv** command displays LANZ records in comma-separated value (CSV) format with the oldest samples displayed first.

LANZ must be enabled to use this command (see **queue-monitor length (global configuration mode)**). If LANZ is disabled, the command displays “queue-monitor is disabled.”

Command Mode

EXEC

Command Syntax

```
show queue-monitor length csv
```

Example

- This command displays LANZ records in CSV format.

```
switch>show queue-monitor length csv

Report generated at 2016-02-09 22:57:50
Type,Time,Interface,Duration(usecs),Queue-Length,Time-Of-Max-Queue(usecs),Laten
cy(usecs),Tx-Drops
 S,2016-02-09 22:53:05.70596,Et29(11),N/A,2590,N/A,60.088,0
 U,2016-02-09 22:53:05.71098,Et29(11),N/A,2590,N/A,60.088,216555
 U,2016-02-09 22:53:05.71600,Et29(11),N/A,2590,N/A,60.088,215546
switch>
```

show queue-monitor length drops

The **show queue-monitor length drops** command displays a report of cumulative transmission drop totals for a range of interfaces or for all interfaces. Output can be limited to a specified number of seconds or records. The most recent events are listed first. By default, the command displays data for all interfaces, limited to the last 1000 records. Newest events are listed first.

LANZ must be enabled to use this command (see **queue-monitor length (global configuration mode)**). If LANZ is disabled, the command displays "queue-monitor is disabled."

Command Mode

EXEC

Command Syntax

```
show queue-monitor length [INTERFACES] [FACTOR] drops
```

Parameters

- *INTERFACES* interface type and number for report. Values include:
  - <no parameter> displays information for all interfaces.
  - **ethernet** *e-range* e-range formats include a number, number range, or comma-delimited list of numbers and ranges
- ***FACTOR*** limiting parameter for report. Values include:
  - <no parameter> displays the last 1000 records.
  - **limit** *number* **samples** displays the last *number* records.
  - **limit** *number* **seconds** displays all records generated during the last *number* seconds.

    Value of *number* ranges from 1 to 1000000.

Guidelines

This command is available on FM6000, Trident II, and Tomahawk platform switches.

Example

- This command displays the last 10 records of transmission drop data on a Trident II or Tomahawk switch.

```
switch>show queue-monitor length limit 10 samples drops

Report generated at 2017-03-10 15:25:34
Time                                    Interface(TC)  TX Drops
------------------------------------------------------------------
0:03:47.06553 ago                       Cpu0/0(39)     17419818
0:03:48.68527 ago                       Cpu0/0(39)     53773707
0:03:53.68527 ago                       Cpu0/0(39)     53773770
0:03:58.68528 ago                       Cpu0/0(39)     53773763
0:04:03.68529 ago                       Cpu0/0(39)     53878
0:04:03.69030 ago                       Cpu0/0(39)     53777
0:04:03.69530 ago                       Cpu0/0(39)     53917
0:04:03.70031 ago                       Cpu0/0(39)     53880
0:04:03.70532 ago                       Cpu0/0(39)     53786
0:04:03.71032 ago                       Cpu0/0(39)     53782
switch>
```

show queue-monitor length ethernet

The **show queue-monitor length ethernet** command displays a report of recent over-threshold events for a range of interfaces, with the newest events listed first.

LANZ must be enabled to use this command (see **queue-monitor length (global configuration mode)**). If LANZ is disabled, the command displays "queue-monitor is disabled."

Command Mode

EXEC

Command Syntax

show queue-monitor length ethernet *e-range*

Parameters

e-range the range of interfaces to be included in the report; formats include a number, number range, or comma-delimited list of numbers and ranges

Examples

- This command displays the last 1000 records for Ethernet interfaces 6 through 8 on a Petra or Arad platform switch.

```
switch>show queue-monitor length ethernet 6-8
Report generated at 2010-01-01 12:56:13
Time                        Interface   Queue length (segments, 1 to 512 bytes)
--------------------------------------------------------------------------------
0:00:07.43393 ago           Et6         1049
0:00:39.22856 ago           Et7         2039
1 day, 4:33:23.12345 ago    Et6         1077
switch>
```

- This command displays the last 1000 records for Ethernet interface on an FM6000 platform switch.

```
switch>show queue-monitor length ethernet 9
Report generated at 2013-04-03 08:45:03
E-End, U-Update, S-Start, TC-Traffic Class
GH-High, GU-Update, GL-Low
Segment size for E, U and S congestion records is 480 bytes
Segment size for GL, GU and GH congestion records is 160 bytes
* Max queue length during period of congestion
+ Period of congestion exceeded counter
--------------------------------------------------------------------------------
Type    Time                   Intf    Congestion     Queue       Time of Max
                               (TC)    duration       length      Queue length
                                       (usecs)        (segments)  relative to
                                                                  congestion
                                                                  start
                                                                  (usecs)
--------------------------------------------------------------------------------
 E  0:00:03.32391 ago         Et9(1)    21044358      4808*       6208
 U  0:00:04.36722 ago         Et9(1)    N/A           4804        N/A
 U  0:00:09.36695 ago         Et9(1)    N/A           4806        N/A
 U  0:00:14.36668 ago         Et9(1)    N/A           4807        N/A
 U  0:00:19.36642 ago         Et9(1)    N/A           4806        N/A
 U  0:00:24.36614 ago         Et9(1)    N/A           4807        N/A
 U  0:00:24.36623 ago         Et9(1)    N/A           4805        N/A
 U  0:00:24.36631 ago         Et9(1)    N/A           4805        N/A
switch>
```

- This command displays the last 1000 records for Ethernet interface 23/3 on a Trident II or Tomahawk platform switch.

```
switch>show queue-monitor length Ethernet 23/3
Report generated at 2017-03-10 15:38:01
E-End, U-Update, S-Start, TC-Traffic Class
Segment size for E, U and S congestion records is 208 bytes
* Max queue length during period of congestion
+ Period of congestion exceeded counter
-------------------------------------------------------------------------------------------------
Type    Time                 Interface      Congestion     Queue      Time of Max    Fabric
                             (TC)           duration       length     Queue length   Peer
                                            (usecs)        (segments) relative to
                                                                      congestion
                                                                      start
                                                                      (usecs)
-------------------------------------------------------------------------------------------------
 E   0:00:29.49376 ago      Et23/3(1)      22388908       7879*      365268
 U   0:00:31.83332 ago      Et23/3(1)      N/A            7877       N/A
 U   0:00:36.83288 ago      Et23/3(1)      N/A            7877       N/A
 U   0:00:41.83289 ago      Et23/3(1)      N/A            7877       N/A
 U   0:00:46.83289 ago      Et23/3(1)      N/A            7877       N/A
 U   0:00:51.83260 ago      Et23/3(1)      N/A            7877       N/A
 U   0:00:51.83760 ago      Et23/3(1)      N/A            7877       N/A
 U   0:00:51.84261 ago      Et23/3(1)      N/A            7878       N/A
 U   0:00:51.84762 ago      Et23/3(1)      N/A            7877       N/A
 U   0:00:51.85263 ago      Et23/3(1)      N/A            7877       N/A
 U   0:00:51.85763 ago      Et23/3(1)      N/A            7877       N/A
 U   0:00:51.86264 ago      Et23/3(1)      N/A            7877       N/A
 U   0:00:51.86765 ago      Et23/3(1)      N/A            7878       N/A
 U   0:00:51.87265 ago      Et23/3(1)      N/A            7877       N/A
 U   0:00:51.87766 ago      Et23/3(1)      N/A            7877       N/A
 S   0:00:51.88267 ago      Et23/3(1)      N/A            1710       N/A
switch>
```

show queue-monitor length global-buffer

The **show queue-monitor length global-buffer** command displays a report of recent high usage, low usage and update events for the global buffer. Newest events are listed first.

LANZ must be enabled to use this command (see **queue-monitor length (global configuration mode)**). If LANZ is disabled, the command displays “queue-monitor is disabled.”

Command Mode

EXEC

Command Syntax

```
show queue-monitor length global-buffer
```

Guidelines

This command is available on FM6000 platform switches.

Example

- This command displays the global buffer event records for the switch.

```
switch>show queue-monitor length global buffer
Report generated at 2013-04-01 14:30:07
GH-High, GU-Update, GL-Low
Segment size = 160 bytes
* Max buffer usage during period of congestion
------------------------------------------------------------------------------
-
Type        Time                        Buffer      Congestion     Time of Max
                                        usage       duration       buffer usage
                                        (segments)  (usecs)        relative to
                                                                   GH (usecs)
------------------------------------------------------------------------------
-
GE 0:04:04.49547 ago                    3121*       20786516       3418
GU 0:04:05.27967 ago                    3120        N/A            N/A
GU 0:04:10.27968 ago                    3120        N/A            N/A
GU 0:04:25.28163 ago                    3118        N/A            N/A
GU 0:04:25.28173 ago                    3118        N/A            N/A
GU 0:04:25.28182 ago                    2963        N/A            N/A
GU 0:04:25.28192 ago                    1916        N/A            N/A
GS 0:04:25.28201 ago                    913         N/A            N/A
switch>
```

show queue-monitor length limit

The **show queue-monitor length limit** command displays a report of recent over-threshold events for a range of interfaces or for all interfaces, limited by a specified number of records.

LANZ must be enabled to use this command (see **queue-monitor length (global configuration mode)**). If LANZ is disabled, the command displays "queue-monitor is disabled."

Command Mode

EXEC

Command Syntax

show queue-monitor length limit [INTERFACES] *number*

Parameters

- *INTERFACES* interface type and number for report. Values include:
  - <no parameter> displays information for all interfaces.
  - **ethernet** *e-range* e-range formats include a number, number range, or comma-delimited list of numbers and ranges
- *number* number of records to display. Values range from 1 to 1000000.

Example

- This command displays the last 100 records for Ethernet interfaces 6 through 8.

```
switch>#show queue-monitor length ethernet 6-8 limit 100 samples
Report generated at 2010-01-01 12:56:13
Time                        Interface   Queue length (segments, 1 to 512 bytes)
--------------------------------------------------------------------------------
0:00:07.43393 ago           Et6         1049
0:00:39.22856 ago           Et7         2039
1 day, 4:33:23.12345 ago    Et6         1077
switch>
```

show queue-monitor length statistics

The **show queue-monitor length statistics** command displays LANZ statistics for all interfaces, showing the traffic class and number of recorded congestion events for each interface.

LANZ must be enabled to use this command (see **queue-monitor length (global configuration mode)**). If LANZ is disabled, the command displays “queue-monitor is disabled.”

Command Mode

EXEC

Command Syntax

```
show queue-monitor length statistics
```

Example

- This command displays LANZ statistics for all interfaces on the switch.

```
switch>show queue-monitor length statistics

Report generated at 2016-02-09 22:59:56
Interface      Traffic Class  Count
---------------------------------------------------------------
Et29            11             1
```

show queue-monitor length status

The **show queue-monitor length status** command displays the current LANZ configuration for the switch and for each interface.

Command Mode

EXEC

Command Syntax

```
show queue-monitor length status
```

Guidelines

On FM6000 platform switches, this command includes status information about global buffer monitoring.

Examples

- This command displays the current LANZ configuration on a Petra or Arad device with default settings.

```
switch(config)#show queue-monitor length status
Per-Interface Queue Length Monitoring
-------------------------------------
Queue length monitoring is enabled
Maximum queue length in bytes : 52428800
Port threshold in bytes:
Port     High threshold
Et3/1           5242880
Et3/2           5242880
Et3/3           5242880
Et3/4           5242880
Et3/5           5242880

              <-------OUTPUT OMITTED FROM EXAMPLE-------->
```

- This command displays the current LANZ configuration on an FM6000 device with default settings.

```
switch(config)#show queue-monitor length status
queue-monitor length enabled
Global Buffer Monitoring
------------------------
Global buffer monitoring is enabled
Segment size in bytes :   160
Total buffers in segments : 36864
High threshold : 10940
Low threshold :  4376


Per-Interface Queue Length Monitoring
-------------------------------------
Queue length monitoring is enabled
Segment size in bytes :   480
Maximum queue length in segments :  3647
Port thresholds in segments:
Port     High threshold  Low threshold
Et1                512            256
Et2                512            256
Et3                512            256
Et4                512            256
Et5                512            256
             <-------OUTPUT OMITTED FROM EXAMPLE-------->
```

- This command displays the current LANZ configuration on a Trident II or Tomahawk device with default settings.

```
switch#show queue-monitor length status
queue-monitor length enabled
queue-monitor length packet sampling is disabled
queue-monitor length update interval in micro seconds:  5000000
Per-Interface Queue Length Monitoring
-------------------------------------
Queue length monitoring is enabled
Segment size in bytes :   208
Maximum queue length in segments : 16382
Port thresholds in segments:
Port     High threshold  Low threshold
Cpu                512            256
Et1/1              512            256
Et1/2              512            256
Et1/3              512            256
Et1/4              512            256
Et2/1              512            256
Et2/2              512            256
Et2/3              512            256
             <-------OUTPUT OMITTED FROM EXAMPLE-------->
```

show queue-monitor length tx-latency

The **show queue-monitor length tx-latency** command displays the latency data of recent LANZ events for a range of interfaces or for all interfaces. Output can be limited to a specified number of seconds or records. The most recent events are listed first. By default, the command displays data for all interfaces, limited to the last 1000 records. Newest events are listed first.

LANZ must be enabled to use this command (see **queue-monitor length (global configuration mode)**). If LANZ is disabled, the command displays "queue-monitor is disabled."

Command Mode

EXEC

Command Syntax

```
show queue-monitor length [INTERFACES] [FACTOR] tx-latency
```

Parameters

- *INTERFACES* interface type and number for report. Values include:
  - <no parameter> displays information for all interfaces.
  - **ethernet** *e-range* e-range formats include a number, number range, or comma-delimited list of numbers and ranges
- ***FACTOR*** limiting parameter for report. Values include:
  - <no parameter> displays the last 1000 records.
  - **limit** *number* **samples** displays the last *number* records.
  - **limit** *number* **seconds** displays all records generated during the last *number* seconds.

    Value of *number* ranges from 1 to 1000000.

Guidelines

This command is available on FM6000, Trident II, and Tomahawk platform switches.

Example

- This command displays transmission latency data for the last 1000 LANZ events on the switch.

```
switch>show queue-monitor length tx-latency

Report generated at 2017-03-10 15:40:02
Time                          Interface(TC)  Tx-Latency (usecs)
-------------------------------------------------------------------
0:02:29.99222 ago             Et23/3(1)      724.868
0:02:32.33178 ago             Et23/3(1)      724.684
0:02:37.33134 ago             Et23/3(1)      724.684
0:02:42.33135 ago             Et23/3(1)      724.684
0:02:47.33135 ago             Et23/3(1)      724.684
0:02:52.33106 ago             Et23/3(1)      730.985
switch>
```

shutdown (queue-monitor-streaming configuration)

The **shutdown** command disables the streaming of LANZ data to external clients. The **no shutdown** command enables LANZ data streaming. Streaming is disabled by default.

Command Mode

Queue-Monitor-Streaming Configuration

Command Syntax

```
shutdown
no shutdown
```

Example

- These commands enable the streaming of LANZ data on the switch.

```
switch(config)#queue-monitor streaming
switch(config-qm-streaming)#no shutdown
switch(config-qm-streaming)#
```

tcpdump queue-monitor

The **tcpdump queue-monitor** command exports congested traffic to a packet capture device or another tool for analysis, or directly to the switch CPU for inspection.

Command Mode

Global Configuration

Command Syntax

```
tcpdump queue-monitor
tcpdump queue-monitor <file | filecount | filter | lookup-names | max-file-size
| packet-count | size | verbose>
```

Parameters

- *file* output file.
  - certificate: certificate file.
  - file: standard file.
  - flash: flash file.
  - sslkey: sslkey file.
  - usb1: usb1 file.
- *filecount* specify the number of output files: 1 to 100.
- *filter* set the filtering expression to select which packets will be dumped.
- *lookup-names* enable reverse DNS lookups.
- *max-file-size* specify the maximum file size by entering 1 to 100 million bytes.
- *packet-count* specify 1 to 10000 packets to capture.
- *size* specify the maximum number of bytes to dump per packet with a size of 1 to 65536 bytes.
- *verbose* enable verbose mode.

Example

- This command inspects traffic on the switch.

```
switch(config)#tcpdump queue-monitor
tcpdump: WARNING: lanz: no IPv4 address assigned
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on lanz, link-type EN10MB (Ethernet), capture size 65535 bytes
...
0 packets captured
0 packets received by filter
0 packets dropped by kernel
switch(config)#
```

Chapter 41

VM Tracer

This chapter describes VM Tracer configuration and usage and contains these sections:

- Section 41.1: VM Tracer Introduction
- Section 41.2: VM Tracer Description
- Section 41.3: VM Tracer Configuration Procedures
- Section 41.4: VM Tracer Configuration Commands

41.1 VM Tracer Introduction

VM Tracer is a switch feature that determines the network configuration and requirements of connected VMware hypervisors. The switch uses VMware's SOAP XML API to discover VMware host server components, including:

- instantiated VMs with their network configuration (VLANs and distributed/virtual Switches).
- server hardware IPMI data which can be shown to the network manager.

VM Tracer also supports adaptive auto-segmentation, which automatically provisions and prunes VLANs from server-switched ports as VMs are instantiated and moved within the data center.

41.2 VM Tracer Description

Cloud operating systems manage large virtualized computing infrastructures, including software and hardware. Cloud operating systems consist of virtual machines and hypervisors:

- A virtual machine (VM) is a software implementation of a computer that operates as running on dedicated physical hardware. Multiple VMs share physical machine resources from a single physical device. Each VM is controlled by its operating system.
- A hypervisor, also called a virtual Machine Manager (VMM), is software that manages multiple operating systems running concurrently on a physical device.

VM Tracer tracks activity of VMs that are controlled by hypervisors connected to the switch's Ethernet or LAG ports. VM Tracer supports vSphere versions 4.0 – 5.5. vSphere features include distributed virtual switches (DVS) and VM movement among VMware servers (VMotion).

vSphere components include:

- ESX and ESXi: hypervisors that run on VMware host server hardware.
- vCenter: centralized tool that manages multiple servers running VMware hypervisors.
- vShield: suite of security applications that support vCenter server integration.
- vShield Manager (VSM): centralized tool that manages vShield access.

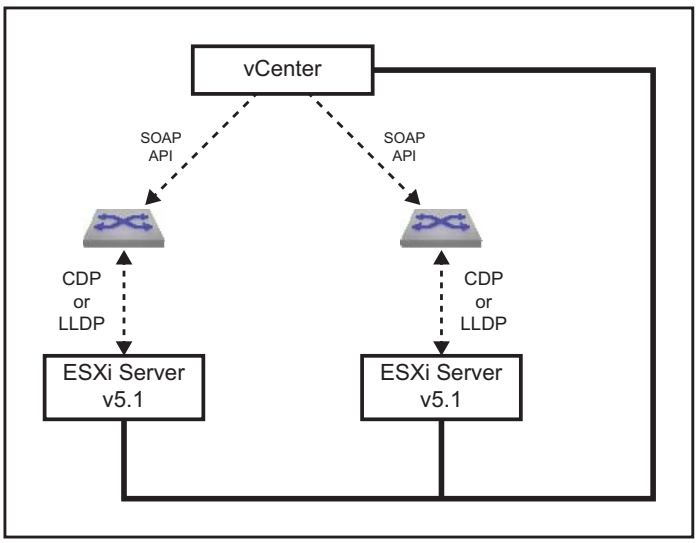
Monitoring VLAN based configurations requires vCenter access. Monitoring VXLAN based configurations requires access to vCenter and vShield Manager. The following sections describe topologies that monitor these networks:

- Section 41.2.1: Monitoring VLAN Based Configurations
- Section 41.2.2: Monitoring VXLAN Based Configurations

41.2.1 Monitoring VLAN Based Configurations

vCenter manages ESX hosts and VMs through a central database. VM Tracer identifies interfaces connected to a specified ESX host and sends discovery packets (CDP or LLDP) on interfaces where VM Tracer is enabled. The ESX host updates the vCenter when it receives a discovery packet. VM Tracer reads this data from the vCenter through a SOAP XML API to associate the ESX host to the connected switch ports. Figure 41-1 displays the network topology of this configuration.

Figure 41-1: VM Tracer Topology – Monitoring VLAN Based Configurations



VM Tracer connects to a maximum of four vCenters through a SOAP (Simple Object Access Protocol) API to discover VMs in the data centers that the vCenters manage. VM Tracer maintains a list of VMs in the data center and gathers network related information about each VM, including the number of Vnics (virtual network interface card), the MAC address of each Vnic, the switch to which it connects, and the host on which it resides. VM Tracer also identifies the host NICs connected to the switch through the bridge MAC address and the interface port name. VM Tracer then searches for VMs on this host and connected to the vswitch or dvswitch whose uplink is mapped to the connected NIC.

For each connected interface, VM Tracer creates a VM Table that lists its active VMs, sorted by Vnic MAC address. Each VM entry includes its name, Vnic name, VLAN, switch name, datacenter name, and portgroup. An entry is deleted when the corresponding VM is removed, moved to a different host, or its Vnic is no longer part of the vswitch or dvswitch. An entry is added when a VM is created or moved to a host connected to the interface. VM Tracer monitors vCenter for VM management updates. If an interface goes down, all VM entries for that interface are removed from the VM Table.

41.2.2 Monitoring VXLAN Based Configurations

Monitoring VXLAN based configurations require access to the vShield Manager (VSM), in addition to the configuration described in Section 41.2.1. Each VM Tracer session can communicate with one VSM through a REST interface over XML and gathers VXLAN information by polling it on a 30 second polling cycle. VXLAN data that the switch receives from the VSM includes:

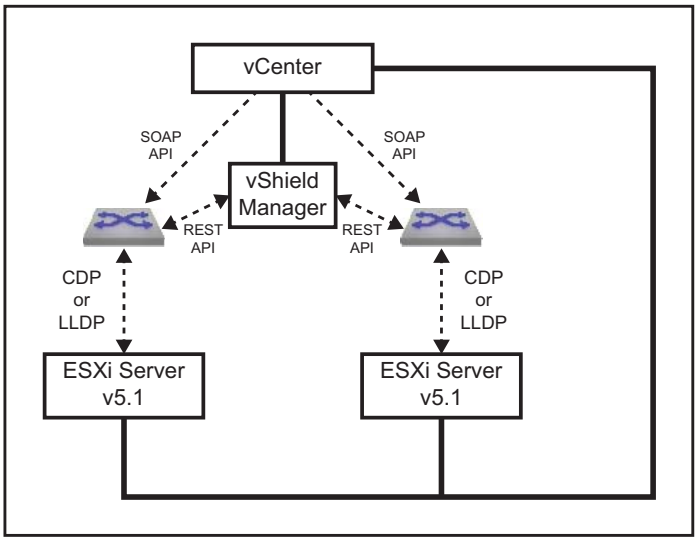
- VNI range.
- VXLAN segment.
- Multicast address range.
- network scope.

  The network scope specifies the virtual address space the VXLAN segments span and is defined by the server group (cluster) collections within the segments, which in turn contain a collection of distributed virtual switches (DVS) from ESX hosts within the clusters.

VM Tracer uses this information to build a network model. Communications with VSM requires a single polling thread that detects network connectivity and constantly updates the local data model.

Figure 41-2 displays the network topology of this configuration.

Figure 41-2: VM Tracer Topology – Monitoring VXLAN Based Configurations



41.3 VM Tracer Configuration Procedures

The following sections describe the session configuration process, configuring the VSM connection for VXLAN based configurations, and the procedure for enabling VM Tracer on individual interfaces. The switch defines vmtracer configuration mode and VMtracer mode:

- vmtracer configuration mode is a command mode for configuring VM Tracer monitoring sessions.
- VMtracer mode is defines an interface state where discovery packets are sent to attached vSwitches.

41.3.1 Configuring vCenter Monitoring Sessions

A VM Tracer session connects the switch to a vCenter server for downloading data about VMs and vSwitches managed by ESX hosts connected to the switch's ports. The switch supports four VM Tracer sessions.

The switch is placed in vmtracer configuration mode to edit session parameters, including the vCenter location and dynamic VLAN usage. Changes take effect by exiting vmtracer mode.

The **vmtracer session** command places the switch in vmtracer configuration mode for a specified session. The command either creates a new session or loads an existing session for editing.

- This command enters vmtracer configuration mode for the system_1 session.

```
switch(config)#vmtracer session system_1
switch(vmtracer-system_1)#
```

In vmtracer configuration mode, the **url (vmtracer mode)**, **username (vmtracer mode)**, and **password (vmtracer mode)** commands specify the location and the account information that authenticates the switch. The url parameter must reference a fully formed secure url.

- These commands specify the IANA url along with the username and password that allow the switch to access the location.

```
switch(vmtracer-system_1)#url https://example.com
switch(vmtracer-system_1)#username a-switch_01
switch(vmtracer-system_1)#password abcde
switch(vmtracer-system_1)#
```

Default session settings allow auto-segmentation, or the dynamic allocation and pruning of VLANs when a VM managed by the ESX host connected to the switch is created, deleted, or moved to a different host. The **autovlan disable** command prevents auto-segmentation, regardless of VM activity. The **allowed-vlan** command specifies the VLANs that may be added when a VM is added or moved. By default, all VLANs are allowed.

- This command disables auto-segmentation.

```
switch(vmtracer-system_1)#autovlan disable
switch(vmtracer-system_1)#
```

- These commands enable auto-segmentation and limit the list of allowed VLANs to VLAN 1-2000.

```
switch(vmtracer-system_1)#no autovlan disable
switch(vmtracer-system_1)#allow-vlan 1-2000
switch(vmtracer-system_1)#
```

The **exit** command returns the switch to Global configuration mode and enables the VM Tracer session. Vmtracer configuration mode can be re-entered for this session to edit session parameters.

- This command exits vmtracer configuration mode.

```
switch(vmtracer-system_1)#exit
switch(config)#
```

The **no vmtracer session** command disables the session and removes it from ***running-config***.

- This command disables and deletes the system_1 VM Tracer session.

```
switch(config)#no vmtracer session system_1
switch(config)#
```

41.3.2 Configuring vShield Monitoring Sessions

To monitor VXLAN based VMware configurations, the switch must communicate with a vShield Manager (VSM). Vmtracer-vxlan configuration mode specifies the location and user account data that allows the switch to access a VSM within the configuration mode vmtracer session.

The switch is placed in vmtracer configuration mode to edit session parameters, including the vCenter location and dynamic VLAN usage. Changes take effect by exiting vmtracer mode.

The **vxlan (vmtracer mode)** command is executed from vmtracer mode for a specified session and places the switch in vmtracer-vxlan configuration mode for that session. Each VM Tracer session can be associated with one vShield instance.

- These commands create the vShield instance for the VMTracer session named vnet-1.

```
switch(config)#vmtracer session vnet-1
switch(config-vmtracer-vnet-1)#vxlan
switch(config-vmtracer-vnet-1-vxlan)#
```

In vmtracer-vxlan configuration mode, the **url (vmtracer-vxlan mode)**, **username (vmtracer-vxlan mode)**, and **password (vmtracer-vxlan mode)** commands specify the vShield server's location and the account information that authenticates the switch to the vShield server. The url parameter must reference a fully formed secure url, such as **https://vcshield.democorp.com/sdk**.

- These commands specify the vShield's url along with the username and password that allow the switch to access the vShield server.

```
switch(config-vmtracer-vnet-1-vxlan)#url https://vshieldserver.company1.org/sdk
switch(config-vmtracer-vnet-1-vxlan)#username a-shield_01
switch(config-vmtracer-vnet-1-vxlan)#password home
switch(config-vmtracer-vnet-1-vxlan)#
```

41.3.3 Enabling vmtracer Mode

VMtracer mode is an interface setting that enables interfaces to send discovery packets to the connected vSwitch. The **vmtracer** command enables VMtracer mode on the configuration mode interface.

- These commands enable VMtracer mode on the Ethernet 3 interface.

```
switch(config)#interface Ethernet3
switch(config-if-Et3)#vmtracer vmware-esx
switch(config-if-Et3)#
```

The **no vmtracer** command disables vmtracer mode on the configuration mode interface.

- This command disables vmtracer mode on the Ethernet 3 interface.

```
switch(config-if-Et3)#no vmtracer vmware-esx
switch(config-if-Et3)#
```

41.3.4 Displaying VM Tracer Data

41.3.4.1 Displaying Session Status

The **show vmtracer session** command displays information about the specified session.

Without the ***detail*** parameter, the command displays connection parameters and status for the vCenter associated to the specified session.

- This command displays connection parameters for the vCenter associated with the system_1 session.

```
switch#show vmtracer session system_1
    vCenter URL https://vmware-vcenter1/sdk
    username arista
    password arista
    Session Status Disconnected
```

With the ***detail*** parameter, the command displays connection status and data concerning messages the vCenter previously received from ESX hosts connected to the switch.

- This command displays connection parameters and message details for the vCenter associated with the system_1 session.

```
switch#show vmtracer session system_1 detail
    vCenter URL https://vmware-vcenter1/sdk
    username arista
    sessionState Connected
    lastStateChange 19 days, 23:03:59 ago
    lastMsgSent CheckForUpdatesMsg
    timeOfLastMsg 19 days, 23:14:09 ago
    resonseTimeForLastMsg 0.0
    numSuccessfulMsg 43183
    lastSuccessfulMsg CheckForUpdatesMsg
    lastSuccessfulMsgTime 19 days, 23:14:19 ago
    numFailedMsg 1076
    lastFailedMsg CheckForUpdatesMsg
    lastFailedMsgTime 19 days, 23:14:09 ago
    lastErrorCode Error -1 fault: SOAP-ENV:Client [no subcode]
    "End of file or no input: Operation interrupted or timed out after 600s send
or     600s receive delay"
    Detail: [no detail]
    CheckForUpdates:
```

41.3.4.2 Displaying VM Interfaces

The **show vmtracer interface** command displays the VM interfaces (Vnics) that are active on switch interfaces where vmtracer mode is enabled. For each Vnic, the command displays the name of the attached VM, the adapter name, its VLAN, the VM power state, and the presence status of its MAC address in the switch's MAC table.

- This command displays the Vnics connected to all VM Tracer-enabled interfaces.

```
switch#show vmtracer interface

Ethernet8 : example.com
    VM Name VM Adapter VLAN Status
    esx3.aristanetworks.com vmk0 0 Up/Down
    vspheremanagement Network adapter 1 0 Up/Down

Ethernet15 : example.om
    VM Name VM Adapter VLAN Status
    Openview Network adapter 1 123 Up/Down
    VmTracerVm Network adapter 1 123 Down/Down

Ethernet23 : example.com
    VM Name VM Adapter VLAN Status

Ethernet24 : example.com
    VM Name VM Adapter VLAN Status
```

41.3.4.3 Displaying VMs

The **show vmtracer vm** command displays VM interfaces (Vnics) accessible to the VM Tracer-enabled interfaces. For each active listed VM, the command displays its name, adapter, and the connected hypervisor.

- This command displays the VMs connected to all VM Tracer-enabled interfaces.

```
switch#show vmtracer vm
    VM Name VM Adapter Interface VLAN
    Openview Network adapter 1 Et15 123
    vspheremanagement Network adapter 1 Et8 0
    VmTracerVm Network adapter 1 Et15 123
    example.com vmk0 Et8 0
```

- This command displays connection data for the VMs connected to all VM Tracer-enabled interfaces.

```
switch#show vmtracer vm detail
VM Name Openview
   intf : Et15
   vnic : Network adapter 1
   mac : 00:0c:29:ae:7e:90
   portgroup : dvPortGroup
   vlan : 123
   switch : vds
   host : example.com
```

41.4 VM Tracer Configuration Commands

Global Configuration Commands

- vmtracer session

Interface Configuration (Ethernet and Port Channel) Commands

- vmtracer

VMTracer Configuration Commands

- allowed-vlan
- autovlan disable
- password (vmtracer mode)
- url (vmtracer mode)
- username (vmtracer mode)
- vxlan (vmtracer mode)

VMTracer-VXLAN Configuration Commands

- password (vmtracer-vxlan mode)
- url (vmtracer-vxlan mode)
- username (vmtracer-vxlan mode)

VM Tracer Display Commands

- show vmtracer all
- show vmtracer interface
- show vmtracer session
- show vmtracer session vcenter
- show vmtracer session vsm
- show vmtracer vm
- show vmtracer vm detail
- show vmtracer vnic counters
- show vmtracer vxlan segment
- show vmtracer vxlan vm

allowed-vlan

The **allowed-vlan** command specifies the VLANs that may be added when a VM is added or moved from the hypervisor connected to the session specified by the vmtracer mode. By default, all VLANs are allowed.

Command Mode

Vmtracer Configuration

Command Syntax

```
allowed-vlan VLAN_LIST
no allowed-vlan vlan
default allowed-vlan vlan
```

Parameters

- ***VLAN_LIST*** The VLAN list or the edit actions to the current VLAN list. Valid *v_range* formats include number, or number range.
  - *v_range* The list consists of the *v_range* VLANs.
  - **add** *v_range* The *v_range* VLANs are added to the current VLAN list.
  - **all** The list consists of all VLANs (1-4094).
  - **except** *v_range* The list consists of all VLANs except for those specified by *v_range*.
  - **none** The list of VLANs is empty.
  - **remove** *v_range* The *v_range* VLANs are removed from the current VLAN list.

Related Commands

- **vmtracer session** places the switch in vmtracer configuration mode.

Examples

- This command sets the list of allowed VLANs to 1 through 2000.

```
switch(vmtracer-system_1)#allow-vlan 1-2000
switch(vmtracer-system_1)#
```

- This command adds VLANs to 2501 through 3000.

```
switch(vmtracer-system_1)#allow-vlan add 2051-3000
switch(vmtracer-system_1)#
```

autovlan disable

Default VM Tracer session settings enable auto provisioning, which allows the dynamic assignment and pruning of VLANs when a VM attached to the ESX connected to the switch is created, deleted, or moved to a different ESX host. The autovlan setting controls auto provisioning.

The **autovlan disable** command disables auto provisioning, which prevents the creation or deletion of VLANs regardless of VM activity. The **allowed-vlan** command specifies the VLANs that may be added when a VM is added or moved. By default, all VLANs are allowed.

The **no autovlan disable** command enables the creation and deletion of VLANs caused by VM activity. This is the default setting.

Command Mode

Vmtracer Configuration

Command Syntax

```
autovlan disable
no autovlan disable
default autovlan disable
```

Related Commands

- **vmtracer session** places the switch in vmtracer configuration mode.

Example

- This command disables dynamic VLAN creation or pruning within the configuration mode VM Tracer session.

```
switch(vmtracer-system_1)#autovlan disable
switch(vmtracer-system_1)#
```

password (vmtracer mode)

The **password** command specifies the token that authorizes the username to the vCenter associated with the VM Tracer mode session.

Command Mode

Vmtracer Configuration

Command Syntax

password [*ENCRYPTION*] [*password*]

Parameters

- ***ENCRYPTION*** encryption level of the password.
  - <no parameter> *password* is a clear-text string.
  - **0** the *password* is a clear-text string. Equivalent to <no parameter>.
  - **7** the *password* is an encrypted string.
- *password* text that authenticates the username.
  - *password* is a clear-text string if ***ENCRYPTION*** specifies clear text.
  - *password* is an encrypted string if ***ENCRYPTION*** specifies an encrypted string.

Related Commands

- **vmtracer session** places the switch in vmtracer configuration mode.

Example

- This command configures ***abode*** as the clear-text string that authorizes the username ***a-switch_01*** located at ***example.com***.

```
switch(vmtracer-system_1)#url https://example.com
switch(vmtracer-system_1)#username a-switch_01
switch(vmtracer-system_1)#password abcde
switch(vmtracer-system_1)#
```

password (vmtracer-vxlan mode)

The **password** command specifies the token that authorizes the username on the vShield Manager (VSM) server located at the URL configured for the configuration mode VM Tracer. The switch uses this account to access VSM information.

The **password** statement is replaced in ***running-config*** for the configuration mode interface by a subsequent **password** command. The statement is removed by deleting the VSM instance through a **no vxlan (vmtracer mode)** command in vmtracer configuration mode.

Command Mode

Vmtracer-vxlan Configuration

Command Syntax

password [*ENCRYPTION*] *password*

Parameters

- ***ENCRYPTION*** encryption level of the password.
  - <no parameter> *password* is a clear-text string.
  - **0** the *password* is a clear-text string. Equivalent to <no parameter>.
  - **7** the *password* is an encrypted string.
- *password* text that autorizes the username.
  - *password* is a clear-text string if ***ENCRYPTION*** specifies clear text.
  - *password* is an encrypted string if ***ENCRYPTION*** specifies an encrypted string.

Related Commands

- **vxlan (vmtracer mode)** places the switch in vmtracer-vxlan configuration mode.

Example

- This command configures ***5678*** as the clear-text string that authorizes the username ***admin*** to the VSM located at ***https://example.com***.

```
switch(config)#vmtracer session vnet-1
switch(config-vmtracer-vnet-1)#vxlan
switch(config-vmtracer-vnet-1-vxlan)#url https://example.com
switch(config-vmtracer-vnet-1-vxlan)#username admin
switch(config-vmtracer-vnet-1-vxlan)#password 5678
switch(config-vmtracer-vnet-1-vxlan)#exit
switch(config-vmtracer-vnet-1)#show active
vmtracer session vnet-1
   allowed-vlan 1-4094
   vxlan
      url https://example.com
      username admin
      password 7 s2Xq4GSBlYU=
switch(config-vmtracer-vnet-1)#
```

show vmtracer all

The **show vmtracer all** command displays VM Tracer data for all switches with the vSphere scope.

Command Mode

EXEC

Command Syntax

```
show vmtracer all
```

Example

- This command displays data for both switches in the vSphere scope.

```
switch>show vmtracer all

Switch : a109(10.10.30.109)
Ethernet49     : 10.102.28.3/10G
   VM Name             VM Adapter          VLAN      Status    State
   ABCD                Network adapter 2   native    Up/--     --

Switch : a164(10.10.30.(172.22.30.164)
Ethernet49     : 10.102.28.3/10G Storage Network/dvUplink1
   VM Name             VM Adapter          VLAN      Status    State
   WXYZ                Network adapter 2   native    Up/--     --
switch>
```

show vmtracer interface

The **show vmtracer interface** command displays the VM interfaces (Vnics) that are active on the VM Tracer enabled interface. For each Vnic, the command displays the name of the attached VM, the adapter name, its VLAN, the VM power state, and the presence status of its MAC address in the switch's MAC table.

Command Mode

EXEC

Command Syntax

show vmtracer interface [*INT_NAME*][*INFO_LEVEL*]

Parameters

- ***INT_NAME*** the interfaces to be configured. Values include:
  - <no parameter> command returns information for all interfaces.
  - **ethernet** *e_range* Ethernet interface range.
  - **port-channel** *p_range* Port Channel interface range.

    Valid *e_range* and *p_range* formats include number, number range, or comma-delimited list of numbers and ranges.
- ***INFO_LEVEL*** specifies information that the command returns.
  - <no parameter> connection parameters and status for VM associated to specified sessions.
  - **detail** connection status and data concerning messages the VM.
  - **host** name of the connected host.

Examples

- This command displays the Vnics connected to all VM Tracer enabled interfaces.

```
switch>show vmtracer interface

Ethernet8 : example.com
    VM Name                     VM Adapter          VLAN          Status
    esx3.aristanetworks.com     vmk0                0             Up/Down
    vspheremanagement           Network adapter 1   0             Up/Down

Ethernet15 : example.com
    VM Name                     VM Adapter          VLAN          Status
    Openview                    Network adapter 1   123           Up/Down
    VmTracerVm                  Network adapter 1   123           Down/Down

Ethernet23 : example.com
    VM Name                     VM Adapter          VLAN          Status
switch>
```

- This command displays the Vnics connected to the Ethernet 8 interface.

```
switch>show vmtracer interface Ethernet8

Ethernet8 : example.com
    VM Name                     VM Adapter          VLAN          Status
    example.com                 vmk0                0             Up/Down
    vspheremanagement           Network adapter 1   0             Up/Down
switch>
```

show vmtracer session

The **show vmtracer session** command displays vCenter and vShield connection information for a specified VM Tracer session.

Command Mode

EXEC

Command Syntax

show vmtracer session [*SESSION_LIST*]

Parameters

- ***SESSION_LIST*** VM Tracer sessions for which the command returns information.
  - <no parameter> all configured VM Tracer sessions.
  - *session_name* name of one VM Tracer session.

Examples

- This command displays connection parameters associated to the abcde session.

```
switch>show vmtracer session abcde

Session abcde
vCenter URL       https://example.com
username                      Administrator
autovlan                      enabled
allowed-vlans     1-4094
sessionState      Connected
VShield URL       https:/vmware-vshield5.1.xyz.abcde.com
username                  admin
sessionState      Connected

switch>
```

show vmtracer session vcenter

The **show vmtracer session vcenter** command displays vCenter information for a specified VM Tracer session.

Command Mode

EXEC

Command Syntax

show vmtracer session *session_name* vcenter [*INFO_LEVEL*]

Parameters

- *session_name* VM Tracer sessions for which the command returns information.
- ***INFO_LEVEL*** specifies information that the command returns.
  - <no parameter> displays connection and status information for the specified vCenter.
  - **detail** displays connection, status, and history information for the specified vCenter.

Examples

- This command displays connection parameters for the vCenter associated to the abcde session.

```
switch>show vmtracer session abcde vcenter

Session abcde
vCenter URL       https://vmware-vcenter5.1/sdk
username                       Administrator
autovlan                       enabled
allowed-vlans     1-4094
sessionState      Connected
switch>
```

- This command displays connection parameters and history details from the vCenter associated to the abcde session.

```
switch>show vmtracer session abcde vcenter detail

Session                        abcde
vCenter URL                    https://vmware-vcenter5.1/sdk
username                       Administrator
autovlan                       enabled
allowed-vlans                  1-4094
SessionState                   Connected
lastStateChange                2:46:50 ago
lastMsgSent                    Query network hint message
timeOfLastMsg                  0:00:20 ago
responseTimeForLastMsg         0.000102301000479
numSuccessfulMsg               998
lastSuccessfulMsg              Query network hint message
lastSuccessfulMsgTime          0:00:20 ago
numFailedMsg                   0
lastFailedMsg                  --
lastFailedMsgTime              never
lastErrorCode                  --
switch>
```

show vmtracer session vsm

The **show vmtracer session vsm** command displays vShield Manager information for a specified VM Tracer session.

Command Mode

EXEC

Command Syntax

show vmtracer session *session_name* vsm [*INFO_LEVEL*]

Parameters

- *session_name* VM Tracer sessions for which the command returns information.
- ***INFO_LEVEL*** specifies information that the command returns.
  - <no parameter> connection and status information for the specified vShield Manager.
  - **detail** connection, status, and history information for the specified vShield Manager..

Examples

- This command displays connection parameters for the vShield Manager associated to the abcde session.

```
switch>show vmtracer session abcde vsm

Session abcde
VShield URL       https://example.com
username                   admin
sessionState      Connected
switch>
```

- This command displays connection parameters and history details from the vShield Manager associated to the abcde session.

```
switch>show vmtracer session abcde vsm detail

Session                       abcde
VShield URL                   https://vmware-vshield5.1/
username                      admin
SessionState                  Connected
LaststateChange               19 days, 23:14:19 ago
LastMsgSent                   /api/2.0/vdn/scopes
timeOfLastMsg                 1 days, 13:22:09 ago
responseTimeForLastMsg        0.3 sec
numSuccessfulMsg              3649
lastSuccessfulMsg             /api/2.0/vdn/scopes
lastSuccessfulMsgTime         0:00:00 ago
numFailedMsg                  1
lastFailedMsg                 /api/2.0/vdn/config/segments
lastFailedMsgTime             10 days, 1:15:29 ago
lastErrorCode                CURLE_COULDNT_RESOLVE_HOST - Couldn't resolve host
                              The given remote host was not resolved.
switch>
```

show vmtracer vm

The **show vmtracer vm** command displays VMs interfaces (Vnics) that are accessible to VM Tracer enabled interfaces. For each active VM, the command displays the name of the VM, its adapter, and the hypervisor to which it connects.

Command Mode

EXEC

Command Syntax

show vmtracer [*INT_NAME*] vm [*VM_LIST*]

Parameters

- ***INT_NAME*** the interfaces name Values include:
  - <no parameter> command returns information for all interfaces.
  - **interface ethernet** *e_range* Ethernet interface range.
  - **interface port-channel** *p_range* Port Channel interface range.

    Valid *e_range* and *p_range* formats include a number, number range, or comma-delimited list of numbers and ranges.
- ***VM_LIST*** The virtual machines for which the command displays information. Options include:
  - <no parameter> command returns information for all present VMs.
  - *vm_name* command returns information only for specified VM.

Related Commands

- **show vmtracer vm detail** displays connection information for one or more specified VMs.

Examples

- This command displays the VMs connected to all VM Tracer enabled interfaces.

```
switch>show vmtracer vm
VM Name             Esx Host                 Interface VLAN   Status

vCenter1            172.22.28.8              Po45      native Down/Down
vCenter2            172.22.28.8              Po45      native Up/Up
vCenter3            172.22.28.8              Po45      11     Down/Down
vCenter4            172.22.28.8              Po45      native Down/Down
VMKernel                                     Po43      native Up/Up
demo vcenter 5 clone                         Po43      native Up/Up
switch>
```

show vmtracer vm detail

The **show vmtracer vm detail** command displays connection data for VMs interfaces (Vnics) that are accessible to VM Tracer enabled interfaces.

Command Mode

EXEC

Command Syntax

show vmtracer vm [*VM_LIST*] detail

Parameters

- ***VM_LIST*** The virtual machines for which the command displays information. Options include:
  - <no parameter> command returns information for all present VMs.
  - *vm_name* command returns information only for specified VM.

Examples

- This command displays connection data for the VMs connected to all VM Tracer enabled interfaces.

```
switch#show vmtracer vm vmcenter1
VM Name  vCenter1 Server App
  Interface   :     Po45
  vNIC        :     Network adapter 1
  MAC         :     00:31:22:8e:b8:41
  Portgroup   :     VM Network
  VLAN        :     native
  Switch      :     Switch2
  Status      :     Down/Down
  Host        :     10.22.18.28
  Data Center :     vcenter-5
switch>
```

- This command displays connection data for the VMs connected to all VM Tracer enabled interfaces.

```
switch>show vmtracer vm detail
VM Name  vCenter1 Server App
  Interface   :     Po45
  vNIC        :     Network adapter 1
  MAC         :     00:31:22:8e:b8:41
  Portgroup   :     VM Network
  VLAN        :     native
  Switch      :     Switch2
  Status      :     Down/Down
  Host        :     10.22.18.28
  Data Center :     vcenter-5

VM Name  vCenter2 Server App
  Interface   :     Po45
  vNIC        :     vmk0
  MAC         :     00:33:23:3c:e1:4e
  Portgroup   :     Management Network
  VLAN        :     native

switch>
```

show vmtracer vnic counters

The **show vmtracer interface vnic counters** command displays input and output packet counts for VM interfaces (Vnics) that are active on the specified interface or VM.

Command Mode

EXEC

Command Syntax

show vmtracer [*ENTITY*] vnic counters

Parameters

- ***ENTITY*** the virtual machine or interface over which statistics are gathered and displayed.
  - <no parameter> command returns information for all active VMs.
  - **interface ethernet** *e_range* Ethernet interface range.
  - **interface port-channel** *p_range* Port Channel interface range.
  - **vm** *vm_name* command returns information for specified VM.

    Valid *e_range* and *p_range* formats include a number, number range, or comma-delimited list of numbers and ranges.

Examples

- This command displays the Vnics connected to Ethernet interface 24.

```
switch>show vmtracer interface ethernet 24 vnic counters
Physical Intf: Ethernet24
Host: 10.17.28.8/site1/dvUplink1
VM Name         vNic                            Input
Pkt/Byte/%              Output Pkt/Byte/%
vCenter1        Network adapter 2          2550/  187175/ 0.6             6/
360/  0.0
vCenter2        Network adapter 2        418615/ 30678024/ 99.4        1904439/
1145654613/100.0
Summary                                  421165/ 30865199/100.0        1904445/
1145654973/100.0
switch>
```

show vmtracer vxlan segment

The **show vmtracer vxlan segment** command displays information about the VXLAN segments that are managed by the connected vShield Manager.

Command Mode

EXEC

Command Syntax

show vmtracer segment *ENTITY*

Parameters

- ***ENTITY*** specifies the information that the command displays. Options include:
  - <no parameter> displays information for VXLAN segments.
  - **pool** displays resource pools available to segments.
  - **pool** *pool_name* displays connection information about the specified pool.
  - **range** displays the VNI range of the managed segments.

Examples

- This command displays the VXLAN segments managed by the VSM.

```
switch>show vmtracer vxlan segment

Name                 VNI     Multicast IP     Network Scope
-------------------------------------------------------------
Eng Wire             5002    237.0.0.1        abcde
HR Wire              5000    237.0.0.2        abcde

switch>
```

- This command displays the resource pools available to the VXLANs.

```
switch>show vmtracer vxlan segment pool

Name                  Description                    Segments
-------------------------------------------------------------------------------
--
abcde                 Spans Cluster 1 and Cluster 2   Eng Wire, HR Wire

switch>
```

- This command displays connection and packet information for the abcde pool.

```
switch>show vmtracer vxlan segment pool abcde

Name:          abcde
Description:   Spans Cluster 1 and Cluster 2
Segments:      Eng Wire, HR Wire

Vxlan Segment  Cluster   Host                  VTEP IP         DVS       VLAN  MTU
Eng Wire       Cluster2  test2.example.com         10.168.200.1/24  dvs-test2
200   1600
Eng Wire       Cluster1  test2.example.com         10.168.100.1/24  dvs-test1
100   1600
HR Wire        Cluster1  test2.example.com         10.168.100.1/24  dvs-test1
100   1600
HR Wire        Cluster2  test2.example.com         10.168.200.1/24  dvs-test2
200   1600
switch>
```

- This command displays the VNI range of the VXLAN segments.

```
switch>show vmtracer vxlan segment range

VNI Range            Multicast IP Range
-------------------------------------------------
5000 - 5024          237.0.0.1 - 237.0.0.117

Name                 VNI     Multicast IP     Network Scope
-----------------------------------------------------------------
HR Wire              5002    237.0.0.1        abcde

Eng Wire             5000    237.0.0.2        abcde
switch>
```

show vmtracer vxlan vm

The **show vmtracer interface vnic counters** command displays the VXLAN segments, their VTEP IP numbers, and their VM endpoints that are managed by the connected vShield Manager.

Command Mode

EXEC

Command Syntax

```
show vmtracer vxlan vm
```

Examples

- This command displays the VM endpoints of the VXLAN segments managed by the VSM.

```
switch>show vmtracer vxlan vm

Vxlan Segment       VTEP IP            VLAN  VMs
Eng Wire            192.168.200.1/24   200   Eng VM3, Eng VM2
Eng Wire            192.168.100.1/24   100   Eng VM1
HR Wire             192.168.100.1/24   100   HR VM2, HR VM1
HR Wire             192.168.200.1/24   200   --
switch>
```

url (vmtracer mode)

The **url** command specifies the vCenter server location that is monitored by the session being edited by the current vmtracer mode. The command must reference a fully formed secure url.

Command Mode

Vmtracer Configuration

Command Syntax

url *url_name*

Parameters

- *url_name* location of the vCenter server. Valid formats include IP address (dotted decimal notation) and fully qualified domain name.

Related Commands

- **vmtracer session** places the switch in vmtracer configuration mode.

Example

- This command specifies the location of the vCenter monitored by the system_1 VM Tracer session.

```
switch(vmtracer-system_1)#url https://example.com
switch(vmtracer-system_1)#
```

url (vmtracer-vxlan mode)

The **url** command specifies the vShield Manager (VSM) server location that is monitored for VXLAN information by the configuration mode VM Tracer session. The command must reference a fully formed secure url.

The **url** statement is replaced in ***running-config*** for the configuration mode session by a subsequent **url** command. The statement is removed by deleting the VSM instance through a **no vxlan (vmtracer mode)** command in vmtracer configuration mode.

Command Mode

Vmtracer-vxlan Configuration

Command Syntax

url *url_name*

Parameters

- *url_name* location of the VSM server. Valid formats include IP address (dotted decimal notation) and fully qualified domain name.

Related Commands

- **vxlan (vmtracer mode)** places the switch in vmtracer-vxlan configuration mode.

Example

- This command configures the location of the VSM monitored by the vnet-1 VM Tracer session.

```
switch(config)#vmtracer session vnet-1
switch(config-vmtracer-vnet-1)#vxlan
switch(config-vmtracer-vnet-1-vxlan)#url https://example.com
switch(config-vmtracer-vnet-1-vxlan)#exit
switch(config-vmtracer-vnet-1)#show active
vmtracer session vnet-1
   allowed-vlan 1-4094
   vxlan
      url https://example.com
switch(config-vmtracer-vnet-1)#
```

username (vmtracer mode)

The **username** command identifies the switch's account name on the vCenter server. The switch uses this user name to access vCenter information.

Command Mode

Vmtracer Configuration

Command Syntax

username *name_string*

Parameters

- *name_string* vCenter account user name. Parameter must match the user name configured on the vCenter.

Related Commands

- **vmtracer session** places the switch in vmtracer configuration mode.

Example

- This command configures the user name for the vCenter associated with the system_1 session. The session uses this user name to log into the vCenter server.

```
switch(vmtracer-system_1)#username a-switch_01
switch(vmtracer-system_1)#
```

username (vmtracer-vxlan mode)

The **username** command identifies the switch's account name on the vShield Manager (VSM) server located at the URL configured for the configuration mode VM Tracer. The switch uses this user name to access VSM information.

The **username** statement is replaced in ***running-config*** for the configuration mode interface by a subsequent **username** command. The statement is removed by deleting the VSM instance through a **no vxlan (vmtracer mode)** command in vmtracer configuration mode.

Command Mode

Vmtracer-vxlan Configuration

Command Syntax

username *name_string*

Parameters

- *name_string* VSM account user name. Parameter must match a user name configured on the VSM.

Related Commands

- **vxlan (vmtracer mode)** places the switch in vmtracer-vxlan configuration mode.

Example

- This command configures the user name of admin for the VSM located at the URL specified by the URL command.

```
switch(config)#vmtracer session vnet-1
switch(config-vmtracer-vnet-1)#vxlan
switch(config-vmtracer-vnet-1-vxlan)#url https://example.com
switch(config-vmtracer-vnet-1-vxlan)#username admin
switch(config-vmtracer-vnet-1-vxlan)#exit
switch(config-vmtracer-vnet-1)#show active
vmtracer session vnet-1
   allowed-vlan 1-4094
   vxlan
      url https://example.com
      username admin
switch(config-vmtracer-vnet-1)#
```

vmtracer

The **vmtracer** command enables vmtracer mode on the configuration mode interface. Interfaces with vmtracer mode enabled send discovery packets to the connected vSwitch.

The **no vmtracer** and **default vmtracer** commands disable vmtracer mode on the configuration mode interface by removing the corresponding **vmtracer** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration
Interface-Port-channel Configuration

Command Syntax

```
vmtracer HOST_TYPE
no vmtracer HOST_TYPE
default vmtracer HOST_TYPE
```

Parameters

- ***HOST_TYPE*** the type of hypervisor that controls the vSwitch to which the interface connects.
  - **vmware-esx** ESX or ESXI hypervisor (VMware).

Examples

- These commands enable vmtracer mode on the Ethernet 3 interface.

```
switch(config)#interface Ethernet 3
switch(config-if-Et3)#vmtracer vmware-esx
switch(config-if-Et3)#
```

- This command disables vmtracer mode on the Ethernet 3 interface.

```
switch(config-if-Et3)#no vmtracer vmware-esx
switch(config-if-Et3)#
```

vmtracer session

The **vmtracer session** command places the switch in vmtracer mode for the specified session. The command creates a new session or loads an existing session for editing.

A VM Tracer session connects the switch to a vCenter server at a specified location, then downloads data about VMs and vSwitches managed by ESX hosts connected to switch ports. The switch supports a maximum of four VM Tracer sessions.

VM Tracer session parameters are configured in vmtracer mode. Parameters configured in vmtracer mode include the vCenter location and dynamic VLAN usage.

The **no vmtracer session** and **default vmtracer session** commands disable the session and remove its configuration from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
vmtracer session name
no vmtracer session name
default vmtracer session name
```

Parameters

- *name* The label assigned to the VM Tracer session.

Commands Available in vmtracer Configuration Mode

- **allowed-vlan**
- **autovlan disable**
- **password (vmtracer mode)**
- **url (vmtracer mode)**
- **username (vmtracer mode)**
- **vxlan (vmtracer mode)**

Examples

- This command enters vmtracer mode for the system_1 session.

```
switch(config)#vmtracer session system_1
switch(vmtracer-system_1)#
```

- This command disables the system_1 VM Tracer session. The system_1 session and all of its parameters are removed from ***running-config***.

```
switch(config)#no vmtracer session system_1
switch(config)#
```

vxlan (vmtracer mode)

The **vxlan** command places the switch in vmtracer-vxlan configuration mode. To monitor VXLAN based VMware configurations, the switch must communicate with a vShield Manager (VSM). Vmtracer-vxlan configuration mode specifies the location and user account data that allows the switch to access a VSM within the configuration mode vmtracer session. Each VM Tracer session can be associated with one vShield instance.

The **no vxlan** and **default interface vxlan** commands delete the vShield instance from the configuration mode vmtracer session by removing all vmtracer-vxlan mode commands from ***running-config***.

Command Mode

Vmtracer Configuration

Command Syntax

```
vxlan
no vxlan
default vxlan
```

Related Commands

- **vmtracer session** places the switch in vmtracer configuration mode.

Commands Available in vmtracer-vxlan Configuration Mode

- **password (vmtracer mode)**
- **url (vmtracer mode)**
- **username (vmtracer mode)**

Example

- These commands create the vShield instance for the VMTracer session named vnet-1.

```
switch(config)#vmtracer session vnet-1
switch(config-vmtracer-vnet-1)#vxlan
switch(config-vmtracer-vnet-1-vxlan)#
```

Chapter 42

Path Tracer

Path Tracer is a network tool that continuously analyzes the network for packet loss and network changes. This chapter describes path tracer concepts and configuration processes.

Sections in this chapter include:

- Section 42.1: Path Tracer Description
- Section 42.2: Path Tracer Configuration
- Section 42.3: Path Tracer Command Descriptions

42.1 Path Tracer Description

42.1.1 Path Tracer Synopsis

Path Tracer is a network tool that continuously analyzes the network for packet loss by sending small test packets (probes) with ranging TTL values to various network destinations. Probes are trapped upon their expiration and recorded by switches and routers along the data path. Aggregated data from multiple probes provide information concerning packet drop frequency along each data path.

Within networks that utilize ECMP and LAG multipaths, a packet's destination is among a criteria set that determines its path and the network devices it encounters. At each multipath decision point, a packet's movement is based on a hash of its header field contents and its ingress layer-2 interface. Factors in a packet's successful delivery include its path and ultimate destination.

Path Tracer implements a distributed version of traceroute, where probe performance measurements are based on sequence number gaps at trapping devices.

Path Tracer provides the following:

- Background monitoring for problem detection.
- Coverage of all first-hop ECMP or LAG members to the probed destination.
- Randomized source port and IP address, providing entropy in the rest of the network links.
- Records recipient L2 interface on trapping device to detect drops affecting a single LAG member.
- Unidirectional path verification without relying on response packets.
- Hardware forwarding verification.

42.1.2 Path Tracer Devices

Path tracer devices are switches and routers along a data path that forwards and processes and path tracer packets (probes). The following terms refer to these devices in terms of their path tracer function.

- **Beacon**: Device that sends or receives probes. A beacon can be a switch, router, or host.

  - An ***egress beacon*** is a device that sends probes.
  - An ***ingress beacon*** is a device that receives probes and records statistics.
- **Path**: The set of devices that probes traverse through the network.
- **Load balancing group**: The set of physical interfaces a packet can egress, based on its IP destination.

42.1.3 Path Tracer Data Structures

Data structures refer to packet formats and stream contents that Path Tracer uses. The following terms describe Path Tracer data structures:

- **Probe**: Text packets transmitted by path tracer beacons and processed by other path tracer devices. Probes are TCP or UDP packets with special payload information that receiving devices use to track network performance.
- **Stream header**: The basic path tracer packet construct. A stream header consists of:
  - Source IP Address
  - Destination IP Address
  - Source Port
  - IP Type
  - Egress Source MAC Address
  - Egress Dest MAC Address
  - Egress Vlan Tag
  - Originating device's egress L3-Interface
  - Originating device's egress L2-Interface

  The first four fields determine the packet's hash. The originating device's egress L2-interface determines the link used by the first packet. The ingress L2-interface at the second hop can be used for hashing. The originating device's L3-interface determines the packets' first-hop direction.
- **Egress header**: Data set used to compute a beacon's set of egress streams. Contains information common to all streams that egress a specific L2-interface. The egress header does not include stream parameters that are not directly tied to an interface, (source IP, source port, IP type).

  The egress header includes:

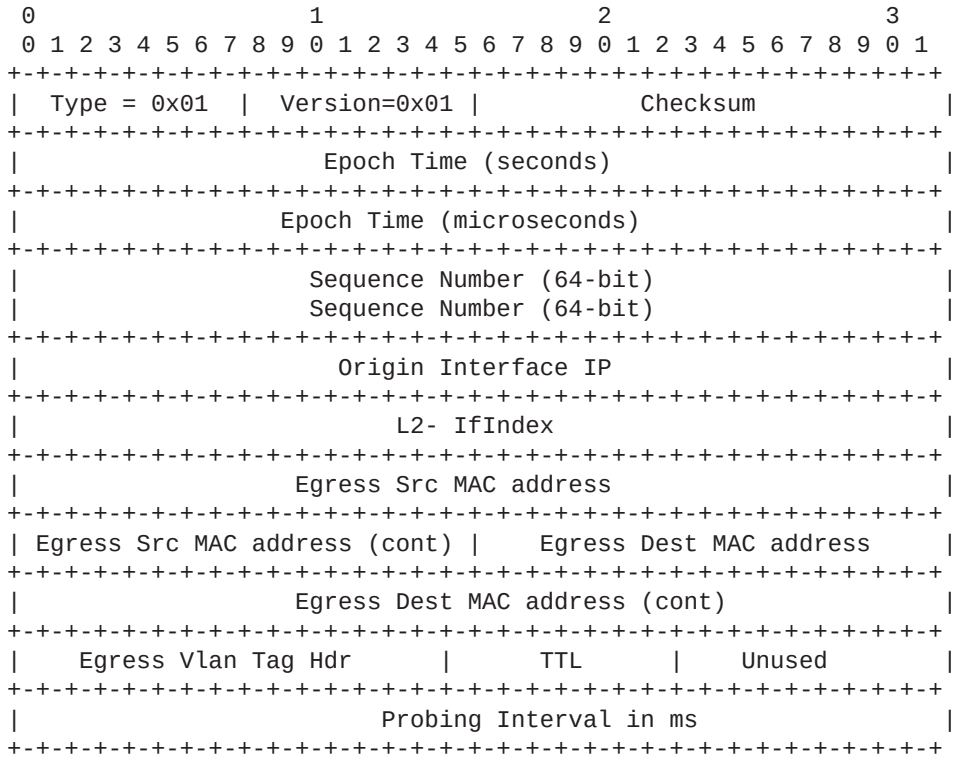
  - Egress Source MAC Address
  - Egress Dest MAC Address
  - Egress Vlan Tag
  - Destination IP Address
  - Original Egress L3-Interface
  - Original Egress L2-Interface
- **Egress stream**: The data set that consists of the stream header and TTL. The egress stream specifies the probe packet's direction (stream header) and network depth (TTL). Probes flow from one or more egress streams at the origin beacon.
- **Ingress stream**: The data set consisting of the stream header, original TTL, and ingress L2-interface, as recorded from arriving probe packets. During stable network periods, a stream's ingress L2-interface remains constant.
- **Destination L4 Port**: Destination TCP or UDP port identifies inbound probe packets. The port must be identical on all beacons through a probe's path. The default destination L4 port 49152.

- **Sequence number**: A 64-bit number that is written into each probe packet to specify its sequence within the stream.
- **Epoch**: Time-based identifier of a probe session. It is refreshed when the sequence number resets.
- **Robustness**: A factor in the calculation of the period that stream data remains in the ingress stream table. The robustness value is 30.

42.1.4 Path Tracer Probe Format

Probe header type (TCP or UDP) and data written in a probe header is determined by the egress stream except for the destination port number. The probe body repeats the TTL from the egress stream, since it is decremented when the packet egresses. The body contains the epoch and sequence numbers.

Probe packets are formatted as follows:



42.1.5 Path Tracer Processes

These sections describe the behavior of switches and routers within their roles as path tracer ingress and egress beacons.

42.1.5.1 Ingress Beacon Behavior

Ingress beacons are network devices that receive and process probes. Ingress beacons trap only probes with an expired TTL (hop limit) or a destination IP that matches the ingress interface.

Each ingress beacon maintains an Ingress Stream Table, which contains the following information for each ingress stream:

- The ingress stream data set.

- The number of probes received, lost, and misordered.

Ingress beacons distinguish probes from other packets by their L4 destination port. Upon identifying a packet that should be trapped, an ingress beacon suspends normal processing of the packet without generating normal responses. The ingress stream data set is generated from the packet header and body. The epoch is compared to the data stream's recorded epoch number. Matching epoch numbers indicate a common probing session sent to each probe. The probe's sequence number is compared to the sequence number in the most recently received packet in the ingress stream. The following are recorded:

- The probe is recorded as ***received***.
- Sequence number gaps are recorded as ***losses***.
- A sequence number that is smaller than the previously received probe is recorded as ***misordered*** if the epochs are identical.
- When epochs differ, the probe is received without recording a loss or misorder.

Ingress Beacons store information locally about received probes. They do not send response packets to the egress beacon.

42.1.5.2 Egress Beacon Behavior

Egress beacons are devices that generate probes. Each egress beacon maintains a path tracer monitor configuration and an egress stream table.

Egress beacon configuration includes the following:

- The source IP addresses or source interfaces whose IP addresses are used as the probe source.
- The L4 source ports to used as the probe source.
- The IP TTL (hop limit) and a corresponding probe transmission rate.
- The destination IP addresses.

The Egress Stream Table contains the following:

- Egress stream data set.
- Epoch and sequence numbers.
- Probe transmission interval and rate (packets per interval).

The egress beacon uses configuration and forwarding information to generate a set of egress streams based on configuration parameters, valid egress physical interfaces, and specified destinations. An egress stream is added for each of the following:

- Each interface or address that is configured as the egress stream source IP address.
- Each port number that is configured as the egress stream source port.
- Each value that is configured as the Egress Stream TTL.
- Each protocol for which the egress stream source IP protocol type is configured.
- Each member of the load balancing group for the specified destination.

Load balancing group members include the set of Ethernet interfaces from where the destination IP address can egress. This set is computed using local forwarding and bridging tables (a switch's FIB and MAC tables or a server's kernel routing tables) and includes:

- Destination IP Address (configured destination).
- Egress Interface IP (each member of ECMP next hops for this destination)
- Egress L2- Interface (each member of any L2 LAGs that act as next hops for this destination).

The egress stream set is computed periodically and is frequently out of date when routing information changes rapidly. When an egress stream is added to the egress stream table, an epoch number is generated based on the current system time and the sequence number is reset to 1. Each egress stream generates probe packets at a specified rate with incrementing sequence numbers. When the sequence number space is exhausted, the epoch number is reset to the current time and the sequence number restarts at 1. Egress beacons can transmit probes in batches if the interval is too short. Each probe must be transmitted from the egress L2- interface specified in the egress stream data set. Beacons do not route packets using their kernel or other routing mechanisms.

42.2 Path Tracer Configuration

These sections describe Path Tracer configuration tasks:

- Section 42.2.1: Enabling Path Tracer
- Section 42.2.2: Path Tracer Beacon Configuration
- Section 42.2.3: Displaying Path Tracer Results

42.2.1 Enabling Path Tracer

Path Tracer parameters are configured in monitor-reachability configuration mode. Monitor-reachability configuration mode is not a group change mode; ***running-config is immediately changed when commands are entered.*** The **monitor reachability** command places the switch in monitor-reachability configuration mode.

Path Tracer is disabled by default and enabled globally on the switch. The **no shutdown (Monitor Reachability)** command enables Path Tracer on the switch.

Example

- These commands enter monitor-reachability configuration mode and enables Path Tracer.

```
switch(config)#monitor reachability
switch(config-mon-reach)#no shutdown
switch(config-mon-reach)#show active
 monitor reachability
   no shutdown
switch(config-mon-reach)#
```

- This command disables Path Tracer.

```
switch(config-mon-reach)#shutdown
switch(config-mon-reach)#show active
switch(config-mon-reach)#
```

- This command exits monitor reachability mode.

```
switch(config-mon-reach)#exit
switch(config)#
```

The **destination port (Monitor Reachability)** command specifies the TCP or UDP port for Path Tracer probes that the switch sends and receives. A probe's destination port must remain consistent throughout its path and not used by any other application.

Example

- This command sets the Path Tracer destination port to 55666.

```
switch(config)#monitor reachability
switch(config-mon-reach)#destination port 55666
switch(config-mon-reach)#show active
 monitor reachability
   destination port 55666
switch(config-mon-reach)#
```

42.2.2 Path Tracer Beacon Configuration

This section describes tasks that configure ingress beacon and egress beacon parameters.

Ingress Beacon Configuration

Enabling Path Tracer also enables the switch as an ingress beacon. Ingress beacon commands limit the quantity of streams that the switch can process and control content removal from the ingress stream table.

Limiting the number of ingress probe streams can diminish the total CPU resources and memory used by Path Tracer when there is a misconfiguration or the switch is receiving more probes than expected. By default, 50,000 streams is the switch maximum. The **probe receiver max-streams** command configures the quantity of ingress probe streams that the switch can simultaneously process.

Example

- This command configures the switch to process a maximum of 10000 ingress streams.

```
switch(config)#monitor reachability
switch(config-mon-reach)#probe receiver max-streams 10000
switch(config-mon-reach)#show active
 monitor reachability
   probe receiver max-streams 10000
switch(config-mon-reach)#
```

By default, data streams remain in the ingress stream period for the period obtained by multiplying the robustness value by the stream's TTL. The **preserve streams** command prevents the removal of stream data from the Ingress Stream Table on the basis of the robustness-TTL expiration period.

Example

- This command prevents the removal of data from the ingress stream table on the basis of robustness-TTL timeout expiry.

```
switch(config)#monitor reachability
switch(config-mon-reach)#preserve-streams
switch(config-mon-reach)#show active
 monitor reachability
   preserve-streams
switch(config-mon-reach)#
```

Egress Beacon and Probe Transmitter Configuration

Egress beacon functions are enabled and configured from probe-xmit configuration, which is a child mode of monitor reachability mode. Probe-xmit mode configures a specified probe transmitter and can be entered multiple times with different transmitter names to create multiple probe transmitters.

Probe-xmit mode is not a group change mode; ***running-config is immediately changed when commands are entered.*** The **probe transmitter** command places the switch in probe-xmit configuration mode for a specified transmitter. The **exit** command returns the switch to monitor-reachability configuration mode.

Probe transmitters are disabled by default and enabled individually from probe-xmit mode. The **no shutdown (Monitor Reachability Probe Transmitter)** command enables Path Tracer on the switch. Even when enabled, a probe transmitter cannot start functioning until a mandatory set of parameters are configured for the transmitter.

Example

- These commands globally enable Path Tracer, then enables the PROBE-1 probe transmitter.

```
switch(config)#monitor reachability
switch(config-mon-reach)#no shutdown
switch(config-mon-reach)#probe transmitter PROBE-1
switch(config-mr-trans-PROBE-1)#no shutdown
switch(config-mr-trans-PROBE-1)#show active
monitor reachability
   no shutdown
   probe transmitter PROBE-1
      no shutdown
switch(config-mr-trans-PROBE-1)#
```

Probe transmission characteristics for probe packets are configured through **hops (Monitor Reachability Probe Transmitter)** commands. Each **hops** command specifies the following:

- The probe's range, in terms of hops as implemented through IP TTL. When a probe's hop limit expires or it reaches the destination IP address, the destination device processes the packet.
- The transmission rate of packets per specified interval. Packet transmissions are uniformly distributed over the specified interval.

A probe transmitter may contain multiple hop commands to facilitate probing at different distances from the switch. Each hop statement in a probe transmitter must specify different hop numbers.

Example

- These commands configure two sets of probe packets: one set transmits probes with a maximum range of 40 hops and the other set transmits probes with a maximum range of 50 hops.

```
switch(config)#monitor reachability
switch(config-mon-reach)#probe transmitter PROBE-X
switch(config-mr-trans-PROBE-X)#hops 40 rate 25 probes every 10 seconds uniform
switch(config-mr-trans-PROBE-X)#hops 50 rate 15 probes every 20 seconds uniform
switch(config-mr-trans-PROBE-X)#show active
monitor reachability
   probe transmitter PROBE-X
      hops 40 rate 25 probes every 20 seconds uniform
      hops 50 rate 15 probes every 20 seconds uniform
      destination ip 10.4.3.3
      destination ip 10.10.5.5/24 subnets 4
switch(config-mr-trans-PROBE-X)#
```

The probe destination is configured though **destination ip (Monitor Reachability Probe Transmitter)**. The **destination ip** command specifies either a single IP address or a set of IP addresses with a common host number and varying subnets.

A probe packet is processed when its TTL (hop limit) expires or it reaches the destination IP address. Each probe transmitter must list at least one destination address and may list multiple addresses through additional **destination ip** commands. The CLI does not accept commands that include an address that was previously specified.

Example

- This command configures the destination IP address a 10.4.3.3.

```
switch(config-mon-reach)#probe transmitter PROBE-X
switch(config-mr-trans-PROBE-X)#destination ip 10.4.3.3
switch(config-mr-trans-PROBE-X)#show active
monitor reachability
   probe transmitter PROBE-X
      destination ip 10.4.3.3
switch(config-mr-trans-PROBE-X)#
```

- This command configures the destination address as 10.10.5.5, 10.10.6.5, 10.10.7.5, and 10.10.8.5

```
switch(config-mr-trans-PROBE-X)#destination ip 10.10.5.5/24 subnets 4
switch(config-mr-trans-PROBE-X)#show active
monitor reachability
   probe transmitter PROBE-X
      destination ip 10.10.5.5/24 subnets 4
switch(config-mr-trans-PROBE-X)#
```

The IP protocol through which probe packets are sent is configured through **ip protocol (Monitor Reachability Probe Transmitter)** commands. UDP is the default protocol.

Example

- These commands configure the PROBE-X probe transmitter to send TCP probe packets.

```
switch(config)#monitor reachability
switch(config-mon-reach)#probe transmitter PROBE-X
switch(config-mr-trans-PROBE-X)#ip protocol tcp
switch(config-mr-trans-PROBE-X)#show active
monitor reachability
   probe transmitter PROBE-X
      hops 40 rate 25 probes every 20 seconds uniform
      hops 50 rate 15 probes every 20 seconds uniform
      ip protocol tcp
      destination ip 10.4.3.3
      destination ip 10.10.5.5/24 subnets 4
switch(config-mr-trans-PROBE-X)#
```

The contents of source IP address and source port fields of probe packets are derived from **source interface (Monitor Reachability Probe Transmitter)** and **source port (Monitor Reachability Probe Transmitter)** commands. The probe transmitter enters one of these source IP addresses and ports in the source fields. Entering multiple addresses and ports in source fields assure the distribution of packets among the available ECMPs.

Example

- These commands configure the probe transmitter to use the IP address 10.25.25.3 as the source interface it probe packets by assigning that address to loopback interface 0, then configuring the loopback interface as the probe transmitter's source interface.

```
switch(config)#interface loopback 0
switch(config-if-Lo0)#ip address 10.25.25.3/28
switch(config-if-Lo0)#exit
switch(config)#monitor reachability
switch(config-mon-reach)#probe transmitter PROBE-X
switch(config-mr-trans-PROBE-X)#source interface loopback 0
switch(config-mr-trans-PROBE-X)#show active
monitor reachability
   probe transmitter PROBE-X
      source interface Loopback0
      destination ip 10.4.3.3
      destination ip 10.10.5.5/24 subnets 4
switch(config-mr-trans-PROBE-X)#
```

- This command configures port transmitters to use 1000, 1010, and 1020 as source ports in its packets.

```
switch(config)#monitor reachability
switch(config-mon-reach)#probe transmitter PROBE-X
switch(config-mr-trans-PROBE-X)#source port sequential start 1000 stride 10 count
3
switch(config-mr-trans-PROBE-X)#show active
monitor reachability
   probe transmitter PROBE-X
      hops 40 rate 25 probes every 20 seconds uniform
      hops 50 rate 15 probes every 20 seconds uniform
      source interface Ethernet7
      source interface Ethernet8
      source interface Ethernet9
      source interface Ethernet10
      source interface Loopback0
      source port sequential start 10000 stride 1000 count 3
      destination ip 10.4.3.3
      destination ip 10.10.5.5/24 subnets 4
switch(config-mr-trans-PROBE-X)#
```

42.2.3 Displaying Path Tracer Results

An ingress beacon captures a probe whose TTL expires or has reached its IP destination. Probes are categorized based on their stream headers; the ingress physical interface is also recorded. Two consecutive probes with the same epoch number and a sequence number gap indicates packet loss that corresponds to the missing sequence numbers. When consecutive probes arrive with the same epoch number and the later packets has the lower sequence number, the switch records a misordering.

Path Tracer information and status is displayed through the **show monitor reachability** command.

Example

- This command displays Path Tracer configuration information.

```
switch#show monitor reachability
Reachability Monitor Configuration
==================================
Reachability Monitor           : Enabled
Ignore checksum                : False
Preserve streams               : True
Allow destination reuse        : False
Maximum ingress streams        : 50000
Destination L4 port            : 49152
Checkpoint interval (s)        : 60.0

Probe Transmitters:
Transmitter one                : Enabled
switch#
```

Probe transmitter configuration and status is displayed through the **show monitor reachability probe-transmitter** command.

Example

- This command displays configuration data for the configured probe transmitters.

```
switch#show monitor reachability probe-transmitter
Probe Transmitter one
=================================
Status: Enabled
IP Protocol: UDP
Probing hop 1 uniformly at rate 1 probes every 1 seconds
Probing hop 2 uniformly at rate 1 probes every 1 seconds
Source Interfaces: Loopback0
Source L4 Ports:   1053, 1181, 2044, 2069, 2738, 2965, 3833, 4505, 4660, 5035,
                   5303, 5713, 5769, 7115, 7244, 8477, 9991, 11516, 13831,
                   14217, 15763, 18541, 19727, 20623, 22674, 22855, 23085,
                   23287, 23641, 24566, 24620, 25462, 25804, 27591, 28266,
                   29970, 32479, 32846, 33172, 33661, 33793, 34803, 35837,
                   39500, 40219, 41086, 41427, 42139, 42355, 42948, 44237,
                   44314, 44552, 44803, 46036, 46555, 47853, 47878, 48376,
                   48386, 50322, 50325, 50420, 50581, 51602, 51942, 52886,
                   54378, 54711, 54907, 56087, 57569, 57825, 57902, 59942,
                   61370, 62817, 63070, 63715, 65340
Destination IPs:   10.0.3.1

switch#
```

Contents of the egress streams table is displayed through the **show monitor reachability egress-streams** command.

Example

- This command displays Path Tracer packets that the switch transmitted.

```
switch#show monitor reachability egress-streams
Report generated at 2013-11-01 17:27:54

Running Counters
Egress Intf IP(L2- IfIndex): 1.2.3.10(32); Egress Intf: Ethernet32;
Dst IP: 10.0.3.1; IP Protocol: UDP; TTL: 1
   Source IP:Port       Probe Rate    Probes TX
-------------------- --------------- ---------
   1.1.5.2:1053         23.438/min    1238707
   1.1.5.2:1181         23.438/min    1238705
   1.1.5.2:2044         23.438/min    1238706
              <-------OUTPUT OMITTED FROM EXAMPLE-------->

   1.1.5.2:65340        23.438/min    1238707

Egress Intf IP(L2- IfIndex): 1.2.3.10(32); Egress Intf: Ethernet32;
Dst IP: 10.0.3.1; IP Protocol: UDP; TTL: 2
   Source IP:Port       Probe Rate    Probes TX
-------------------- --------------- ---------
   1.1.5.2:1053         23.438/min    1238707
   1.1.5.2:1181         23.438/min    1238705
   1.1.5.2:2044         23.438/min    1238705
              <-------OUTPUT OMITTED FROM EXAMPLE-------->
   1.1.5.2:65340        23.438/min    1238709

Egress Intf IP(L2- IfIndex): 1.2.3.10(33); Egress Intf: Ethernet33;
Dst IP: 10.0.3.1; IP Protocol: UDP; TTL: 1
   Source IP:Port       Probe Rate    Probes TX
-------------------- --------------- ---------
   1.1.5.2:1053         23.438/min    1238711
   1.1.5.2:1181         23.438/min    1238705
   1.1.5.2:2044         23.438/min    1238705
              <-------OUTPUT OMITTED FROM EXAMPLE-------->

switch#
```

The **show monitor reachability ingress-streams** command displays contents of the ingress streams table.

Example

- This command displays the ingress stream table.

```
switch#show monitor reachability ingress-streams
Report generated at 2013-11-01 19:03:23

Stream Information
Egress Intf IP(L2- IfIndex): 1.2.3.2(16); Ingress Intf: Ethernet37;
Dst IP: 10.0.2.1; IP Protocol: UDP; TTL: 2
   Source IP:Port      Stream            Last              Last          Current
                        Created           Epoch Change       Probe RX       Rate
-------------------- ----------------- ----------------- -----------------
----------
   1.1.5.3:2181       6 days,          6 days,          6 days,       31.266/min
                        1:36:01 ago        1:36:01 ago        1:35:50 ago
   1.1.5.3:6360       6 days,          6 days,          6 days,       31.266/min
                        1:36:02 ago        1:36:02 ago        1:35:48 ago
   1.1.5.3:6713       36 days,         9 days,          0:00:01 ago   23.438/min
                        18:38:41 ago       22:19:08 ago
        <-------OUTPUT OMITTED FROM EXAMPLE-------->
switch#
```

Statistics for all received ingress stream packets is displayed through the **show monitor reachability interface-ttl-statistics** command. Data is categorized by interface-TTL combination.

Example

- This command displays ingress stream statistics for each interface-TTL combination.

```
switch#show monitor reachability interface-ttl-statistics
Report generated at 2013-11-01 19:38:20

Running Counters
  Ingress Intf   TTL    Probes RX     Probes Lost    Probes Misordered    Probes
Duplicated
--------------- ------ ------------ ------------- --------------------
----------------
  Ethernet32      1      198824703     3              0                    0
  Ethernet32      2      98225716      38811183       0                    81
  Ethernet33      1      198824644     2              0                    0
  Ethernet33      2      99744621      38354746       0                    84
  Ethernet34      1      198824653     2              0                    0
  Ethernet34      2      117370842     32965473       0                    97
       <-------OUTPUT OMITTED FROM EXAMPLE-------->
  Ethernet47      2      84025148      1471809        0                    0

switch#
```

Statistics for all received ingress stream packets is displayed through the **show monitor reachability probe-statistics** command. Data is categorized by source IP address and port for each ingress interface.

Example

- This command displays the cumulative statistics concerning the ingress probes it has processed.

```
switch#show monitor reachability probe-statistics
Report generated at 2013-11-01 19:55:18

Running Counters
Egress Intf IP(L2- IfIndex): 1.2.3.2(16); Ingress Intf: Ethernet37;
Dst IP: 10.0.2.1; IP Protocol: UDP; TTL: 2
Source IP:Port  Probes RX   Probes   Probes       Probes      Last Loss/Misorder
                             Lost     Misordered   Duplicated   Last Duplicate
--------------- ---------- -------- ------------ ------------ ------------------
10.1.5.3:2181    5           2        0            0           6 days,
                                                               2:27:50 ago
                                                               never
10.1.5.3:6360    7           1        0            0           6 days,
                                                               2:27:43 ago
                                                               never
10.1.5.3:6713    531701      333710   0            1           3 days,
                                                               13:12:58 ago
                                                               31 days,
                                                               22:46:09 ago
              <-------OUTPUT OMITTED FROM EXAMPLE-------->
switch#
```

42.3 Path Tracer Command Descriptions

Global Configuration Commands

- monitor reachability

Clear Commands

- clear monitor reachability probe-statistics

Display Commands

- show monitor reachability
- show monitor reachability egress-streams
- show monitor reachability ingress-streams
- show monitor reachability interface-ttl-statistics
- show monitor reachability probe-statistics
- show monitor reachability probe-transmitter

Monitor Reachability Configuration Mode Commands

- destination port (Monitor Reachability)
- preserve streams
- probe receiver max-streams
- probe transmitter
- shutdown (Monitor Reachability)

Monitor Reachability Probe Transmitter Configuration Mode Commands

- destination ip (Monitor Reachability Probe Transmitter)
- hops (Monitor Reachability Probe Transmitter)
- ip protocol (Monitor Reachability Probe Transmitter)
- shutdown (Monitor Reachability Probe Transmitter)
- source interface (Monitor Reachability Probe Transmitter)
- source port (Monitor Reachability Probe Transmitter)

clear monitor reachability probe-statistics

The **clear monitor reachability probe-statistics** command removes all entries from the probe statistics table and resets the ingress stream table.

Command Mode

Privileged EXEC

Command Syntax

```
clear monitor reachability probe-statistics
```

Example

- This command clears the probe statistics and ingress stream tables.

```
switch#clear monitor reachability probe-statistics
switch#
```

destination ip (Monitor Reachability Probe Transmitter)

The **destination ip** command specifies the destination of packets sent by the configuration mode probe transmitter. The command specifies either a single IP address or a set of IP addresses with a common host number and varying subnets.

A probe packet is process when its TTL (hop limit) expires or it reaches the destination IP address. Each probe transmitter must list at least one destination address and may list multiple addresses through additional commands. A command that includes an address that was previously specified is not accepted.

The **no destination ip** and **default destination ip** commands remove the specified destination IP address from the configuration probe transmitter by removing the corresponding destination IP command from ***running-config***.

Command Mode

Mon-reach-xmit Configuration

Command Syntax

```
destination ip DESTINATION_ADDR
no destination ip DESTINATION_ADDR
default destination ip DESTINATION_ADDR
```

Parameters

- ***DESTINATION_ADDR*** Probe transmitter destination IP addresses. Options include:
  - *ipv4_addr* Individual IP address.
  - *ipv4_prefix* **subnets** <***1*** to ***65535***> IPv4 address followed by number of subnets.

Related Commands

- **probe transmitter** places the switch in monitor-reachability-xmit configuration mode.

Examples

- This command configure the destination IP address as 10.4.3.3.

```
switch(config-mon-reach)#probe transmitter PROBE-X
switch(config-mr-trans-PROBE-X)#destination ip 10.4.3.3
switch(config-mr-trans-PROBE-X)#show active
monitor reachability
   probe transmitter PROBE-X
      destination ip 10.4.3.3
switch(config-mr-trans-PROBE-X)#
```

- This command configures the destination address as 10.10.5.5, 10.10.6.5, 10.10.7.5, and 10.10.8.5

```
switch(config-mr-trans-PROBE-X)#destination ip 10.10.5.5/24 subnets 4
switch(config-mr-trans-PROBE-X)#show active
monitor reachability
   probe transmitter PROBE-X
      destination ip 10.4.3.3
      destination ip 10.10.5.5/24 subnets 4
switch(config-mr-trans-PROBE-X)#
```

destination port (Monitor Reachability)

The **destination port** command specifies the destination L4 (TCP or UDP) port for Path Tracer probes that the switch sends and receives. The default path tracer port is 49152. A probe's destination port must remain consistent throughout its path and not used by any other application.

The **no destination port** and **default destination port** commands restore the default destination L4 port to 49152 by removing the **destination port** command from ***running-config***.

Command Mode

Mon-reach Configuration

Command Syntax

```
destination port port_id
no destination port
default destination port
```

Parameters

- *port_id* UDP or TCP port number. Value ranges from 0 to 65535.

Related Commands

- **monitor reachability** places the switch in monitor-reachability configuration mode.

Example

- This command sets the Path Tracer destination port to 55666.

```
switch(config)#monitor reachability
switch(config-mon-reach)#destination port 55666
switch(config-mon-reach)#show active
 monitor reachability
   destination port 55666
switch(config-mon-reach)#
```

- This command resets the path tracer destination port to the default value of 49152.

```
switch(config-mon-reach)#no destination port
switch(config-mon-reach)#show active
switch(config-mon-reach)#
```

hops (Monitor Reachability Probe Transmitter)

The **hops** command configures probe transmission characteristics for the configuration mode probe transmitter by specifying the network distance (measured in hops) and packet rate. The command configures the following egress stream packet transmission parameters:

- The distance is measured in hops as implemented through IP TTL. When a probe's hop limit expires or it reaches the destination IP address, the device that received the packet processes it.
- The packet rate defines the number of packets sent per specified interval. Packet transmissions are uniformly distributed over the specified interval.

A probe transmitter may contain multiple hop commands to facilitate reachability probing at different distances from the switch. Multiple hop statements within a probe transmitter cannot specify the same number of hops.

The **no hops** and **default hops** commands remove the probe transmission characteristics configuration by removing the corresponding **hops** command from ***running-config***.

Command Mode

Mon-reach-xmit Configuration

Command Syntax

```
hops distance_hops rate probes_quantity probes every INTERVAL uniform
no hops distance_hops
default hops distance_hops
```

Parameters

- *distance_hops* specifies the probe's hop limit (TTL). Value ranges from 1 to 255.
- *probes_quantity* quantity of packets sent per transmission period. Value ranges from 1 to 65535.
- ***INTERVAL*** transmission period. Options specify a programmable number of seconds or minutes. Quantity value for each time unit ranges from 1 to 65535.
  - *quantity* **seconds** Interval length, as measured in seconds.
  - *quantity* **minutes** Interval length, as measured in minutes.

Related Commands

- **probe transmitter** places the switch in monitor-reachability-xmit configuration mode.

Examples

- These commands configure two set of probe packets: one set transmits probes with a maximum range of 40 hops and the other set transmits probes with a maximum range of 50 hops.

```
switch(config)#monitor reachability
switch(config-mon-reach)#probe transmitter PROBE-X
switch(config-mr-trans-PROBE-X)#hops 40 rate 25 probes every 10 seconds uniform
switch(config-mr-trans-PROBE-X)#hops 50 rate 15 probes every 20 seconds uniform
switch(config-mr-trans-PROBE-X)#show active
monitor reachability
   probe transmitter PROBE-X
      hops 40 rate 25 probes every 20 seconds uniform
      hops 50 rate 15 probes every 20 seconds uniform
      destination ip 10.4.3.3
      destination ip 10.10.5.5/24 subnets 4
switch(config-mr-trans-PROBE-X)#
```

ip protocol (Monitor Reachability Probe Transmitter)

The **ip protocol** command specifies the IP protocol that the switch uses to send probe packet through the configuration mode probe transmitter. The default protocol is UDP.

The **no ip protocol** and **default ip protocol** commands restore the default protocol for the configuration mode probe transmitter by removing the corresponding **ip protocol** statement from ***running_config***.

Command Mode

Mon-reach-xmit Configuration

Command Syntax

```
ip protocol PROT_TYPE
no ip protocol
default ip protocol
```

Parameters

- ***PROT_TYPE*** Specifies the IP protocol. Options include:
  - **tcp** TCP packets.
  - **udp** UDP packets.

Related Commands

- **probe transmitter** places the switch in monitor-reachability-xmit configuration mode.

Example

- These commands configure the PROBE-X probe transmitter to send TCP probe packets.

```
switch(config)#monitor reachability
switch(config-mon-reach)#probe transmitter PROBE-X
switch(config-mr-trans-PROBE-X)#ip protocol tcp
switch(config-mr-trans-PROBE-X)#show active
monitor reachability
   probe transmitter PROBE-X
      hops 40 rate 25 probes every 20 seconds uniform
      hops 50 rate 15 probes every 20 seconds uniform
      ip protocol tcp
      destination ip 10.4.3.3
      destination ip 10.10.5.5/24 subnets 4
switch(config-mr-trans-PROBE-X)#
```

monitor reachability

The **monitor reachability** command places the switch in monitor-reachability configuration mode, which is used to specify Path Tracer configuration parameters. Path Tracer is a network monitor and analysis tool that continuously and actively probes the network for packet loss.

Monitor-reachability configuration mode is not a group change mode; ***running-config*** is changed immediately upon entering commands. Exiting monitor-reachability configuration mode does not affect ***running-config***. The **exit** command returns the switch to global configuration mode.

The **no monitor reachability** and **default monitor reachability** commands remove previously configured **monitor reachability** commands from *running-config*.

Command Mode

Global Configuration

Command Syntax

```
monitor reachability
no monitor reachability
default monitor reachability
```

Commands Available in Monitor-reachability Configuration Mode

- **destination port (Monitor Reachability)**
- **preserve streams**
- **probe transmitter**
- **probe receiver max-streams**
- **shutdown (Monitor Reachability)**

Examples

- These commands place the switch in monitor-reachability configuration mode and globally enable Path Tracer.

```
switch(config)#monitor reachability
switch(config-mon-reach)#no shutdown
switch(config-mon-reach)#show active
 monitor reachability
   no shutdown
switch(config-mon-reach)#
```

- This command exits monitor reachability mode.

```
switch(config-mon-reach)#exit
switch(config)#
```

- This command deletes all previously configured monitor-reachability commands.

```
switch(config)#no monitor reachability
switch(config)#monitor reachability
switch(config-mon-reach)#show active
switch(config-mon-reach)#
```

preserve streams

The **preserve streams** command prevents the expiration of streams from the Ingress Stream Table. By default, data streams remain in the ingress stream period for the period obtained by multiplying the robustness value by the stream's TTL.

The **no preserve streams** and **default preserve streams** commands restore the default behavior of removing streams from the ingress stream table upon robustness-TTL timeout expiry by removing the **preserve streams** command from ***running-config***.

Command Mode

Mon-reach Configuration

Command Syntax

```
preserve streams
no preserve streams
default preserve streams
```

Related Commands

- **monitor reachability** places the switch in monitor-reachability configuration mode.

Example

- This command prevents the expiration of streams from the Ingress Stream Table.

```
switch(config)#monitor reachability
switch(config-mon-reach)#preserve-streams
switch(config-mon-reach)#show active
 monitor reachability
   preserve-streams
switch(config-mon-reach)#
```

probe receiver max-streams

The **probe receiver max-streams** command configures the quantity of ingress probe streams that the switch can simultaneously process. Limiting the number of ingress probe streams can diminish the total CPU resources and memory used by Path Tracer when there is a misconfiguration or the switch is receiving more probes than expected. The default maximum is 50,000 streams.

The **no probe receiver max-streams** and **default probe receiver max-streams** commands restore the max-hops setting to its default value of 50000 by removing the **probe receiver max-streams** command from ***running-config***.

Command Mode

Mon-reach Configuration

Command Syntax

```
probe receiver max-streams data_streams
no probe receiver max-streams
default probe receiver max-streams
```

Parameters

- *data_streams* Maximum quantity of ingress probe streams. Value ranges from 0 to 50000. Default is 50000.

Related Commands

- **monitor reachability** places the switch in monitor-reachability configuration mode.

Example

- This command configures a ingress maximum of 10000 probe streams.

```
switch(config)#monitor reachability
switch(config-mon-reach)#probe receiver max-streams 10000
switch(config-mon-reach)#show active
 monitor reachability
   probe receiver max-streams 10000
switch(config-mon-reach)#
```

probe transmitter

The **probe transmitter** command is a monitor reachability command that places the switch in probe-xmit mode for configuring and enabling a Path Tracer probe transmitter. The command either accesses an existing probe transmitter configuration or creates a new instance if it specifies a new transmitter.

Probe-xmit configuration mode is not a group change mode; ***running-config*** is changed immediately upon entering commands. Exiting probe-xmit mode does not affect ***running-config***. The **exit** command returns the switch to monitor-reachability configuration mode.

The **no probe transmitter** and **default probe transmitter** commands delete the specified probe transmitter and its configuration statements.

Command Mode

Mon-reach Configuration

Command Syntax

```
probe transmitter transmitter_name
no probe transmitter transmitter_name
default probe transmitter transmitter_name
```

Parameters

- *transmitter_name* Probe transmitter name.

Related Commands

- **monitor reachability** places the switch in monitor-reachability configuration mode.

Commands Available in Monitor-reachability- Configuration Mode

- **destination ip (Monitor Reachability Probe Transmitter)**
- **hops (Monitor Reachability Probe Transmitter)**
- **ip protocol (Monitor Reachability Probe Transmitter)**
- **shutdown (Monitor Reachability Probe Transmitter)**
- **source port (Monitor Reachability Probe Transmitter)**

Examples

- These commands create a Path Tracer probe transmitter named PROBE-1.

```
switch(config)#monitor reachability
switch(config-mon-reach)#probe transmitter PROBE-1
switch(config-mr-trans-PROBE-1)#show active
monitor reachability
   probe transmitter PROBE-1
switch(config-mr-trans-PROBE-1)#
```

- These commands exit probe-xmit configuration and monitor reachability configuration modes.

```
switch(config-mr-trans-PROBE-1)#exit
switch(config-mon-reach)#exit
switch(config)#
```

- These commands delete the PROBE-1 probe transmitter.

```
switch(config)#monitor reachability
switch(config-mon-reach)#no probe transmitter PROBE-1
switch(config-mon-reach)#
```

show monitor reachability

The **show monitor reachability** command displays path tracer configuration information.

Command Mode

Privileged EXEC

Command Syntax

```
show monitor reachability
```

Example

- This command displays Path Tracer configuration information.

```
switch#show monitor reachability
Reachability Monitor Configuration
==================================
Reachability Monitor           : Enabled
Ignore checksum                : False
Preserve streams               : True
Allow destination reuse        : False
Maximum ingress streams        : 50000
Destination L4 port            : 49152
Checkpoint interval (s)        : 60.0

Probe Transmitters:
Transmitter one                : Enabled
switch#
```

show monitor reachability egress-streams

The **show monitor reachability egress-streams** command displays contents of the egress streams table.

Command Mode

Privileged EXEC

Command Syntax

```
show monitor reachability egress-streams DATA_TYPE OUTPUT
```

Parameters

- ***DATA_TYPE*** specifies information about the egress beacon.
  - <no parameter> displays beacon's interface and IP information.
  - **layer2** displays beacon's interface, IP, and MAC address information.
- ***OUTPUT*** Specifies format of output data.
  - <no parameter> beacon data is summarized, followed by list of packets.
  - **verbose** each entry displays packet and beacon information. Items listed in tabular format.
  - **verbose csv** Same information as verbose, formatted into csv table.

Example

- This command displays Path Tracer packets that the switch transmitted.

```
switch#show monitor reachability egress-streams
Report generated at 2013-11-01 17:27:54

Running Counters
Egress Intf IP(L2- IfIndex): 1.2.3.10(32); Egress Intf: Ethernet32;
Dst IP: 10.0.3.1; IP Protocol: UDP; TTL: 1
   Source IP:Port       Probe Rate    Probes TX
------------------- ---------------- ---------
   1.1.5.2:1053         23.438/min    1238707
   1.1.5.2:1181         23.438/min    1238705
   1.1.5.2:2044         23.438/min    1238706
              <-------OUTPUT OMITTED FROM EXAMPLE-------->

   1.1.5.2:65340        23.438/min    1238707

Egress Intf IP(L2- IfIndex): 1.2.3.10(32); Egress Intf: Ethernet32;
Dst IP: 10.0.3.1; IP Protocol: UDP; TTL: 2
   Source IP:Port       Probe Rate    Probes TX
------------------- ---------------- ---------
   1.1.5.2:1053         23.438/min    1238707
   1.1.5.2:1181         23.438/min    1238705
   1.1.5.2:2044         23.438/min    1238705
              <-------OUTPUT OMITTED FROM EXAMPLE-------->
   1.1.5.2:65340        23.438/min    1238709

Egress Intf IP(L2- IfIndex): 1.2.3.10(33); Egress Intf: Ethernet33;
Dst IP: 10.0.3.1; IP Protocol: UDP; TTL: 1
   Source IP:Port       Probe Rate    Probes TX
------------------- ---------------- ---------
   1.1.5.2:1053         23.438/min    1238711
   1.1.5.2:1181         23.438/min    1238705
   1.1.5.2:2044         23.438/min    1238705
              <-------OUTPUT OMITTED FROM EXAMPLE-------->

switch#
```

show monitor reachability ingress-streams

The **show monitor reachability ingress-streams** command displays contents of the ingress streams table.

Command Mode

Privileged EXEC

Command Syntax

show monitor reachability ingress-streams *DATA_TYPE STREAM_FILTER OUTPUT*

Parameters

- ***DATA_TYPE*** specifies information about the ingress beacon.
  - <no parameter> displays beacon's interface and IP information.
  - **layer2** displays beacon's interface, IP, and MAC address information.
- ***STREAM_FILTER*** Filters the packets for which data is displayed.
  - <no parameter> All packets are displayed.
  - **anomaly** Only misordered and loss packets are displayed.
  - **loss** Only lost packets are displayed.
- ***OUTPUT*** Specifies format of output data.
  - <no parameter> beacon data is summarized, followed by list of packets.
  - **verbose** each entry displays packet and beacon information. Items listed in tabular format.
  - **verbose csv** Same information as verbose, formatted into csv table.

Example

- This command displays the ingress stream table.

```
switch#show monitor reachability ingress-streams
Report generated at 2013-11-01 19:03:23

Stream Information
Egress Intf IP(L2- IfIndex): 1.2.3.2(16); Ingress Intf: Ethernet37;
Dst IP: 10.0.2.1; IP Protocol: UDP; TTL: 2
   Source IP:Port      Stream            Last             Last          Current
                        Created           Epoch Change       Probe RX       Rate
-------------------- ----------------- ----------------- -----------------
----------
  10.1.5.3:2181       6 days,          6 days,          6 days,       31.266/min
                        1:36:01 ago        1:36:01 ago        1:35:50 ago
  10.1.5.3:6360       6 days,          6 days,          6 days,       31.266/min
                        1:36:02 ago        1:36:02 ago        1:35:48 ago
  10.1.5.3:6713       36 days,         9 days,          0:00:01 ago   23.438/min
                        18:38:41 ago       22:19:08 ago
      <-------OUTPUT OMITTED FROM EXAMPLE-------->
switch#
```

show monitor reachability interface-ttl-statistics

The **show monitor reachability interface-ttl-statistics** command displays aggregated data concerning ingress stream packets. Cumulative information is presented for each interface-TTL combination.

Command Mode

Privileged EXEC

Command Syntax

show monitor reachability interface-ttl-statistics *STREAM_FILTER*

Parameters

- ***STREAM_FILTER*** Filters the packets for which data is displayed.
  - <no parameter> All packets are used.
  - **anomaly** Only misordered and loss packets are used.
  - **loss** Only lost packets are used.

Example

- This command displays ingress stream statistics for each interface-TTL combination.

```
switch#show monitor reachability interface-ttl-statistics
Report generated at 2013-11-01 19:38:20

Running Counters
  Ingress Intf   TTL    Probes RX     Probes Lost    Probes Misordered    Probes
Duplicated
--------------- ------ ------------ ------------- --------------------
----------------
  Ethernet32      1      198824703     3              0                    0
  Ethernet32      2      98225716      38811183       0                    81
  Ethernet33      1      198824644     2              0                    0
  Ethernet33      2      99744621      38354746       0                    84
  Ethernet34      1      198824653     2              0                    0
  Ethernet34      2      117370842     32965473       0                    97
       <-------OUTPUT OMITTED FROM EXAMPLE-------->
  Ethernet47      2      84025148      1471809        0                    0

switch#
```

show monitor reachability probe-statistics

The **show monitor reachability probe-statistics** command displays statistics concerning the ingress packets the switch has captured and processed. Data is categorized by source IP address and port for each ingress interface.

Command Mode

Privileged EXEC

Command Syntax

show monitor reachability probe-statistics ***DATA_TYPE STREAM_FILTER OUTPUT***

Parameters

- ***DATA_TYPE*** specifies information about the ingress beacon.
  - <no parameter> displays beacon's interface and IP information.
  - **layer2** displays beacon's interface, IP, and MAC address information.
- ***STREAM_FILTER*** Filters the packets for which data is displayed.
  - <no parameter> All packets are displayed.
  - **anomaly** Only misordered and loss packets are displayed.
  - **loss** Only loss packets are displayed.
- ***OUTPUT*** Specifies format of output data.
  - <no parameter> beacon data is summarized, followed by list of packets.
  - **verbose** each entry displays packet and beacon information. Items listed in tabular format.
  - **verbose csv** Same information as verbose, formatted into csv table.

Example

- This command displays the cumulative statistics concerning the ingress probes it has processed.

```
switch#show monitor reachability probe-statistics
Report generated at 2013-11-01 19:55:18

Running Counters
Egress Intf IP(L2- IfIndex): 1.2.3.2(16); Ingress Intf: Ethernet37;
Dst IP: 10.0.2.1; IP Protocol: UDP; TTL: 2
Source IP:Port  Probes RX   Probes   Probes       Probes      Last Loss/Misorder
                             Lost     Misordered   Duplicated   Last Duplicate
--------------- ---------- -------- ------------ ------------ ------------------
10.1.5.3:2181    5           2        0            0           6 days,
                                                               2:27:50 ago
                                                               never
10.1.5.3:6360    7           1        0            0           6 days,
                                                               2:27:43 ago
                                                               never
10.1.5.3:6713    531701      333710   0            1           3 days,
                                                               13:12:58 ago
                                                               31 days,
                                                               22:46:09 ago
                <-------OUTPUT OMITTED FROM EXAMPLE-------->
switch#
```

show monitor reachability probe-transmitter

The **show monitor reachability probe-transmitter** command displays configured parameters for the specified probe transmitter.

Command Mode

Privileged EXEC

Command Syntax

show monitor reachability probe-transmitter *XMITTER_LIST*

Parameters

- ***XMITTER_LIST*** Source interface for the mirroring session.
  - <no parameter> Displays all configured probe transmitters.
  - *xmitter_1* Name of probe transmitter that is displayed.
  - *xmitter_1 xmitter_2 ... xmitter_N* List of probe transmitters that the command displays.

Example

- This command displays configuration data for the configured probe transmitters.

```
switch#show monitor reachability probe-transmitter
Probe Transmitter one
=================================
Status: Enabled
IP Protocol: UDP
Probing hop 1 uniformly at rate 1 probes every 1 seconds
Probing hop 2 uniformly at rate 1 probes every 1 seconds
Source Interfaces: Loopback0
Source L4 Ports:   1053, 1181, 2044, 2069, 2738, 2965, 3833, 4505, 4660, 5035,
                   5303, 5713, 5769, 7115, 7244, 8477, 9991, 11516, 13831,
                   14217, 15763, 18541, 19727, 20623, 22674, 22855, 23085,
                   23287, 23641, 24566, 24620, 25462, 25804, 27591, 28266,
                   29970, 32479, 32846, 33172, 33661, 33793, 34803, 35837,
                   39500, 40219, 41086, 41427, 42139, 42355, 42948, 44237,
                   44314, 44552, 44803, 46036, 46555, 47853, 47878, 48376,
                   48386, 50322, 50325, 50420, 50581, 51602, 51942, 52886,
                   54378, 54711, 54907, 56087, 57569, 57825, 57902, 59942,
                   61370, 62817, 63070, 63715, 65340
Destination IPs: 10.0.3.1

switch#
```

shutdown (Monitor Reachability)

The **shutdown** command globally disables the Path Tracer on the switch. Globally disabling path tracer disables all probe transmitters on the switch. When path tracer is globally enabled, probe transmitters are individually enabled through **shutdown (Monitor Reachability Probe Transmitter)** commands.

By default, path tracer is disabled. After entering monitor-reachability configuration mode, a **no shutdown** command is required to enable path tracer. When path tracer is globally disabled, all path tracer probe transmitter are disabled.

The **shutdown** and **default shutdown** commands disable path tracer by removing the **no shutdown** command from ***running-config***

Command Mode

Mon-reach Configuration

Command Syntax

```
shutdown
no shutdown
default shutdown
```

Related Commands

- **monitor reachability** places the switch in monitor-reachability configuration mode.

Examples

- These commands globally enable path tracer.

```
switch(config)#monitor reachability
switch(config-mon-reach)#no shutdown
switch(config-mon-reach)#show active
 monitor reachability
   no shutdown
switch(config-mon-reach)#
```

- This command globally disables path tracer.

```
switch(config-mon-reach)#shutdown
switch(config-mon-reach)#show active
switch(config-mon-reach)#
```

shutdown (Monitor Reachability Probe Transmitter)

The **shutdown** command disables the configuration mode probe transmitter. A probe transmitter is a routine that defines the parameters of an egress stream and, when enabled, facilitates the transmission of the probes that comprise the stream.

When Path Tracer is globally disabled (**shutdown (Monitor Reachability)**, all probe transmitters are disabled. When Path Tracer is globally enabled, probe transmitters are individually enabled through **shutdown (Monitor Reachability Probe Transmitter)** commands.

By default, probe transmitters are disabled. After entering probe-xmit configuration mode, a **no shutdown** command is required to enable the probe transmitter..

The **shutdown** and **default shutdown** commands disable the configuration mode probe transmitter by removing the corresponding **no shutdown** command from ***running-config.***

Command Mode

Mon-reach-xmit Configuration

Command Syntax

```
shutdown
no shutdown
default shutdown
```

Related Commands

- **probe transmitter** places the switch in monitor-reachability-xmit configuration mode.

Examples

- These commands globally enable the Path Tracer, then enables the PROBE-1 probe transmitter..

```
switch(config)#monitor reachability
switch(config-mon-reach)#no shutdown
switch(config-mon-reach)#probe transmitter PROBE-1
switch(config-mr-trans-PROBE-1)#no shutdown
switch(config-mr-trans-PROBE-1)#show active
monitor reachability
   no shutdown
   probe transmitter PROBE-1
      no shutdown
switch(config-mr-trans-PROBE-1)#
```

source interface (Monitor Reachability Probe Transmitter)

The **source interface** command specifies the interface from where the configuration mode probe transmitter's source IP addresses are derived. The probe transmitter enters one of these source IP addresses in the source IP address field of packets that it sends. Multiple source interface statements may be configured for a probe transmitter. Using multiple source addresses can assure the distribution of packets among the available ECMPs paths.

The **no source interface** and **default source interface** commands remove the specified source interface assignments from the configuration mode transmitter by removing the corresponding **source interface** command from ***running-config***.

Command Mode

Mon-reach-xmit Configuration

Command Syntax

```
source interface INT_NAME
no source interface INT_NAME
default source interface INT_NAME
```

Parameters

- ***INT_NAME*** Interface type and number. Options include:
  - **ethernet** *e_range* Ethernet interface range specified by *e_range*.
  - **loopback** *l_range* Loopback interface specified by *l_range*.
  - **management** *m_range* Management interface range specified by *m_range*.
  - **port-channel** *p_range* Port-Channel Interface range specified by *p_range*.
  - **vlan** *v_range* VLAN interface range specified by *v_range*.

  Valid parameter formats include number, number range, or comma-delimited list of numbers and ranges.

Example

- These commands configure the probe transmitter to use the IP address 10.25.25.3 as the source interface by assigning that address to loopback interface 0, then configuring the loopback interface as the probe transmitter's source interface.

```
switch(config)#interface loopback 0
switch(config-if-Lo0)#ip address 10.25.25.3/28
switch(config-if-Lo0)#exit
switch(config)#monitor reachability
switch(config-mon-reach)#probe transmitter PROBE-X
switch(config-mr-trans-PROBE-X)#source interface loopback 0
switch(config-mr-trans-PROBE-X)#show active
monitor reachability
   probe transmitter PROBE-X
      source interface Loopback0
      destination ip 10.4.3.3
      destination ip 10.10.5.5/24 subnets 4
switch(config-mr-trans-PROBE-X)#
```

source port (Monitor Reachability Probe Transmitter)

The **source port** command specifies the configuration mode probe transmitter's source port values. The probe transmitter enters one of these source port values in the source port field of packets that it sends. Although only one source port statement can be configured for a probe transmitter, a statement may specify multiple explicitly or randomly selected ports. Using multiple source ports can assure the distribution of packets among the available ECMPs paths. The default setting is one randomly selected port.

The **no source port** and **default source port** commands restore the default setting of one random source port numbers by removing the corresponding **source port** command from ***running-config***.

Command Mode

Mon-reach-xmit Configuration

Command Syntax

```
source port [PORT_ROSTER]
no source port
default source port
```

Parameters

- ***PORT_ROSTER*** set of source port numbers. Options include:
  - <no parameter> one random port. Same as ***random count 1***.
    - **list** *port_1* **[***port_2* **...** *port_n***]** a list of port number. Values of *port_x* range from 0 to 65535.
    - **random count** *port_quantity* a set of randomly selected port numbers. The set size is specified by *port_quantity*. Value of *port_quantity* ranges from 1 to 65535.
    - **sequential start** *port_1* **stride** *port_interval* **count** *port_quantity* A set of explicitly defined port values. The set size is specified by *port_quantity*. The first port is specified by *port_1*. Successive port numbers are derived by adding *port_interval* to the port numbers, starting with *port_1*.

Related Commands

- **probe transmitter** places the switch in monitor-reachability-xmit configuration mode.

Example

- This command configures the port transmitters to use 1000, 1010, and 1020 as source ports in its packets.

```
switch(config)#monitor reachability
switch(config-mon-reach)#probe transmitter PROBE-X
switch(config-mr-trans-PROBE-X)#source port sequential start 1000 stride 10 count
3
switch(config-mr-trans-PROBE-X)#show active
monitor reachability
   probe transmitter PROBE-X
      hops 40 rate 25 probes every 20 seconds uniform
      hops 50 rate 15 probes every 20 seconds uniform
      source interface Ethernet7
      source interface Ethernet8
      source interface Ethernet9
      source interface Ethernet10
      source interface Loopback0
      source port sequential start 10000 stride 1000 count 3
      destination ip 10.4.3.3
      destination ip 10.10.5.5/24 subnets 4
switch(config-mr-trans-PROBE-X)#
```

Chapter 43

MapReduce Tracer

This chapter describes Arista's implementation of MapReduce Tracer, including configuration instructions and command descriptions. Topics covered by this chapter include:

- Section 43.1: MapReduce Tracer Introduction
- Section 43.2: MapReduce Tracer Configuration
- Section 43.3: Displaying MapReduce Tracer Results
- Section 43.4: MapReduce Tracer Command Descriptions

43.1 MapReduce Tracer Introduction

MapReduce Tracer is a network tool that monitors Hadoop nodes that are directly connected to Arista switches. MapReduce Tracer requires the following:

- Hadoop clusters are deployed with a L3 design.
- The top of rack switch is the default gateway to all attached TaskTracker nodes.
- JobTracker RPC ports do not require authentication.
- Nodes cannot simultaneously belong to multiple Hadoop clusters.
- All TaskTrackers within a cluster are accessed through a common HTTP access port.
- The switch's DNS or static host configuration facilitates TaskTracker name resolution.

Map Reduce Tracer supports these Hadoop releases:

- Apache 0.20.205
- Apache 1.2.1
- Cloudera 3u6
- Cloudera 4.1.3
- Cloudera 4.3.0
- HortonWorks 1.3
- Cloudera 4.5.0

These sections briefly describe Hadoop, Hadoop data structures, and MapReduce Tracer.

43.1.1 Hadoop Description

Apache Hadoop is an open-source, Java-based software framework that supports large dataset storage and processing in a distributed computational environment. Hadoop is licensed under Apache License 2.0 and developed through a global community.

Hadoop facilitates application execution on systems composed of thousands of nodes utilizing petabytes of data. Its distributed file system facilitates rapid data transfer among nodes and supports continued operations when individual nodes fail or become inaccessible.

Hadoop Distributed File System (HDFS) is a distributed file-system that stores data on the commodity machines to provide high aggregate bandwidth across the cluster.

43.1.2 Hadoop Cluster Structure

A cluster is a group of servers that function as a single system to provide high availability through load balancing and parallel processing. A Hadoop cluster is a type of computational cluster designed for storing and analyzing large amounts of unstructured data in a distributed computing environment.

Typical Hadoop clusters include one master and multiple worker nodes. The master node consists of a TaskTracker, JobTracker, NameNode and DataNode. Worker nodes include a TaskTracker and DataNode.

43.1.3 Map Reduce

MapReduce is an algorithm that Hadoop implements to process large datasets by distributing parallel tasks to nodes within a cluster. The MapReduce program includes a Map procedure that filters data and a Reduce procedure that processes the data.

MapReduce manages task and data distribution to cluster nodes such that tasks are executed in parallel, and data transfers between cluster components support redundancy and fault tolerance.

The MapReduce engine consists of one JobTracker and multiple TaskTrackers – all nodes within the Hadoop cluster. The JobTracker receives MapReduce jobs from a client application and manages the completion of these jobs by submitting tasks to available TaskTracker nodes. If a TaskTracker fails to perform the assigned task, the JobTracker reschedules that part of the job to another node.

43.1.4 MapReduce Tracer Function

MapReduce Tracer is a feature that tracks and interacts with Hadoop nodes directly connected to Arista switches in a cluster. It communicates with a JobTracker to obtain a list of all nodes in a cluster and then queries JobTracker and TaskTrackers on these nodes for information regarding the jobs they are running and progress of those jobs. This creates a map of TaskTrackers with kinds of jobs they are running. Commands are available to display this data in tables through the CLI and EAPI.

MapReduce Tracer monitors only nodes that connect directly to the switch in L3 networks. Directly connected nodes use the top-of-rack switch as their default gateway and the switch learns ARP entries for these nodes. The list of nodes provided by JobTracker is filtered by tracking ARP entries to remove nodes that are not directly accessible.

MapReduce Tracer creates a database of nodes from the filtered list. After the database is created, the switch queries the JobTracker and TaskTrackers to obtain the following:

- The number of monitored Hadoop nodes.
- The list of monitored nodes, including their IP addresses.
- Jobs that the TaskTrackers are running.
- JobTracker and TaskTracker statistics.

MapReduce Tracer can simultaneously monitor multiple clusters. This means the directly connected TaskTracker nodes can belong to different clusters. A maximum of 5 clusters are supported per switch.

43.2 MapReduce Tracer Configuration

MapReduce Tracer configuration commands are structured into two configuration levels:

- Monitor-hadoop configuration mode is a child of global configuration mode and controls global MapReduce Tracer settings.
- Monitor-hadoop-cluster configuration mode is a child of Monitor-hadoop configuration mode and defines polling configurations that monitor individual Hadoop clusters.

These sections describe MapReduce Tracer configuration processes:

- Section 43.2.1: MapReduce Tracer Global Configuration
- Section 43.2.2: Hadoop Cluster Access Configuration

MapReduce Tracer functions after it is enabled globally. Each polling configuration can be individually enabled after the feature is enabled globally.

43.2.1 MapReduce Tracer Global Configuration

MapReduce Tracer global parameters are configured in Monitor-hadoop configuration mode. Tasks performed from this mode include specifying connection parameters to Hadoop clusters and globally enabling MapReduce Tracer.

Entering Monitor-Hadoop Configuration Mode

Monitor-hadoop configuration mode is entered by **monitor hadoop**. Monitor-hadoop configuration mode is not a group change mode; statements are stored in the ***running-config*** when they are entered through the CLI. The **exit** command returns the switch to global configuration mode.

Examples

- These commands place the switch in monitor-hadoop configuration mode.

```
switch(config)#monitor hadoop
switch(config-monitor-hadoop)#
```

- This command exits monitor-hadoop mode.

```
switch(config-monitor-hadoop)#exit
switch(config)#
```

- This command deletes all previously configured monitor-hadoop configuration mode commands.

```
switch(config)#no monitor hadoop
switch(config)#
```

Globally Enabling MapReduce Tracer

MapReduce Tracer is globally enabled by **no shutdown (Monitor-Hadoop)**. MapReduce Tracer is globally disabled by default.

Example

- These commands globally enable MapReduce Tracer.

```
switch(config)#monitor hadoop
switch(config-monitor-hadoop)#no shutdown
switch(config-monitor-hadoop)#show active
 monitor hadoop
   no shutdown
switch(config-monitor-hadoop)#
```

Creating a Cluster Monitor

A cluster monitor is created by entering monitor-hadoop-cluster mode with **cluster (Monitor Hadoop).** Each monitor is labeled with a cluster ID and probes one Hadoop cluster. When the command specifies a monitor with a previously defined cluster ID, subsequent commands edit that monitor's parameters. A monitor with a new cluster ID is created by a command that specifies a nonexistent cluster ID.

Example

- These commands enter monitor-hadoop-cluster mode to edit a cluster monitor. The monitor's cluster-id is CL2.

```
switch(config-monitor-hadoop)#cluster CL2
switch(config-monitor-hadoop-CL2)#show active
monitor hadoop
   cluster CL2
switch(config-monitor-hadoop-CL2)#
```

43.2.2 Hadoop Cluster Access Configuration

Cluster monitors are configured in monitor-hadoop-cluster configuration mode. Each monitor corresponds to a hadoop cluster through these configurable parameters:

- JobTracker access parameters (address, port number, and username)
- TaskTracker access port
- Polling interval
- Cluster description
- Enabled setting

The minimum explicit configuration includes JobTracker address and username; default values are defined for all other parameters. By default, cluster monitors are disabled.

The **cluster (Monitor Hadoop)** command places the switch in monitor-hadoop-cluster mode for the specified monitor, where a cluster's connection parameters are specified. Monitor-hadoop-cluster mode is not a group change mode.

A cluster monitor is enabled by **no shutdown (Monitor Hadoop Cluster)** when MapReduce Tracer is globally enabled.

43.2.2.1 JobTracker Configuration

A cluster's JobTracker is located on the master node and schedules work to the cluster's TaskTracker nodes. The **jobtracker (Monitor Hadoop Cluster)** command specifies connection parameters to the monitored cluster.

JobTracker parameters include its node location (IPv4 address or hostname), RPC port, and username. The default RPC port is 8021. Location and username parameters do not have default values and must be explicitly configured.

Example

- For the CL2 monitor, these commands configure connection parameters to a JobTracker node at ***10.4.4.4*** with the username ***account1***. The default RPC port (8021) is implicitly specified.

```
switch(config)#monitor hadoop
switch(config-monitor-hadoop)#cluster CL2
switch(config-monitor-hadoop-CL2)#jobtracker host 10.4.4.4 username account1
switch(config-monitor-hadoop-CL2)#show active
monitor hadoop
   cluster CL2
      jobtracker host 10.4.4.4 user account1
switch(config-monitor-hadoop-CL2)#
```

43.2.2.2 TaskTracker Configuration

The **tasktracker (Monitor Hadoop Cluster)** command specifies the HTTP port that access TaskTrackers of the Hadoop cluster probed by the configuration mode monitor. The switch compiles a list of the cluster's TaskTracker addresses by periodically polling the cluster's JobTracker.

The default TaskTracker HTTP port is 50060.

Examples

- For the CL2 monitor, these commands configure a TaskTracker access port of 51000.

```
              switch(config)#monitor hadoop
switch(config-monitor-hadoop)#cluster CL2
switch(config-monitor-hadoop-CL2)#tasktracker http-port 51000
switch(config-monitor-hadoop-CL2)#show active
monitor hadoop
   cluster CL2
      tasktracker http-port 51000
switch(config-monitor-hadoop-CL2)#
```

- These commands restore the default TaskTracker HTTP access port address of 50060.

```
switch(config-monitor-hadoop-CL2)#no tasktracker http-port
switch(config-monitor-hadoop-CL2)#show active
monitor hadoop
   cluster CL2
switch(config-monitor-hadoop-CL2)#show active all
monitor hadoop
   cluster CL2
      jobtracker rpc-port 8021
      tasktracker http-port 50060
      interval 10
      shutdown
switch(config-monitor-hadoop-CL2)#
```

43.2.2.3 Polling Interval Configuration

When the monitor configuration is complete, the switch polls the cluster's JobTracker to maintain the list of active TaskTracker nodes associated with the monitored cluster and compile Hadoop job statistics. The **interval (Monitor Hadoop Cluster)** command specifies the interval between polls to the JobTracker of the monitored cluster. The default interval is 10 seconds.

Example

- This command sets the JobTracker polling interval to 25 seconds for the cluster monitored by the CL2 MapReduce Tracer configuration.

```
switch(config)#monitor hadoop
switch(config-monitor-hadoop)#cluster CL2
switch(config-monitor-hadoop-CL2)#interval 25
switch(config-monitor-hadoop-CL2)#show active
monitor hadoop
   cluster CL2
      interval 25
switch(config-monitor-hadoop-CL2)#
```

43.2.3 MapReduce Tracer Example

The commands in this section create monitors that probe two Hadoop clusters, enables each monitor individually, then enables MapReduce Tracer globally. Monitor parameters for the clusters include:

- Cluster ID: CL_1
  - Jobtracker: IP address 10.15.2.2; RPC port 8021; username xyz1
  - TaskTracker: HTTP address 54000
  - JobTracker polling interval: 10 seconds (default)
- Cluster ID: CL_2
  - Jobtracker: IP address 10.21.5.2; RPC port 9521; username qrst4
  - TaskTracker: HTTP address 50060 (default)
  - JobTracker polling interval: 5 seconds

```
switch(config)#monitor hadoop
switch(config-monitor-hadoop)#cluster CL_1

switch(config-monitor-hadoop-CL_1)#jobtracker host 10.15.2.2 username xyz1
switch(config-monitor-hadoop-CL_1)#tasktracker http-port 54000
switch(config-monitor-hadoop-CL_1)#no shutdown
switch(config-monitor-hadoop-CL_1)#exit

switch(config-monitor-hadoop)#cluster CL_2
switch(config-monitor-hadoop-CL_2)#jobtracker host 10.21.5.2 rpc-port 9521
username qrst4
switch(config-monitor-hadoop-CL_2)#interval 5
switch(config-monitor-hadoop-CL_2)#no shutdown
switch(config-monitor-hadoop-CL_2)#exit

switch(config-monitor-hadoop)#no shutdown
switch(config-monitor-hadoop)#show active
 monitor hadoop
   no shutdown
   cluster CL_1
      jobtracker host 10.15.2.2 user xyz1
      tasktracker http-port 54000
      no shutdown
   !
   cluster CL_2
      jobtracker host 10.21.5.2 rpc-port 9521 user qrst4
      interval 5
```

```
      no shutdown

switch(config-monitor-hadoop)#show active all
 monitor hadoop
   no shutdown
   cluster CL_1
      jobtracker host 10.15.2.2 rpc-port 8021 user xyz1
      tasktracker http-port 54000
      interval 10
      no shutdown
   !
   cluster CL_2
      jobtracker host 10.21.5.2 rpc-port 9521 user qrst4
      tasktracker http-port 50060
      interval 5
      no shutdown
switch(config-monitor-hadoop)#exit
switch(config)#
```

43.3 Displaying MapReduce Tracer Results

MapReduce Tracer display commands provide information about the configuration and activity on the monitored clusters.

43.3.1 MapReduce Tracer Status

MapReduce Tracer status is accessed through **show monitor hadoop status**. Status information includes the enabled status and the number of monitored clusters, TaskTrackers, and locally running jobs.

Example

- This command displays MapReduce Tracer status for all connected clusters and TaskTrackers.

```
switch>show monitor hadoop status
Last updated: 2013-10-06 18:14:23
Mapreduce Tracer status:
  Admin status                          : Enabled
  Operational status                    : Enabled
  Number of clusters configured         : 3
  Number of local TaskTrackers          : 4
  Number of jobs running locally        : 4

switch>
```

43.3.2 Cluster Configuration and Connections

The following cluster configuration and connection information is available through these commands:

- Configuration and connection data for all monitored clusters – **show monitor hadoop cluster all**.
- Configuration and connection data for a specified cluster – **show monitor hadoop cluster status**
- Connection and activity information for TaskTrackers in a specified cluster, on a specified node, or accessed through a specified interface – **show monitor hadoop tasktracker status**.

Example

- This command displays configuration and connection data for the ***Cluster0*** cluster.

```
switch>show monitor hadoop cluster Cluster0 status
Last updated: 2013-10-06 18:14:23
Cluster status for cluster:  Cluster0
  Admin status                : Enabled
  JobTracker host             : host0
  JobTracker RPC port         : 9000
  JobTracker user             : user0
  JobTracker polling interval : 100 seconds
  TaskTracker HTTP port       : 8800
  Operational status          : Enabled
  Active TaskTrackers         : 31
  Blacklisted TaskTrackers    : 1
  Decommissioned TaskTrackers : 1
  Tracker expiry interval     : 20.0
  Map slots (used/total)      : 10/100
  Reduce slots (used/total)   : 11/110
  JobTracker heap size        : 1.04GB (max: 2.08GB)
switch>
```

43.3.3 Job Lists

The following commands display rosters of currently running job or jobs that previously ran:

- Jobs running on all monitored Hadoop clusters – **show monitor hadoop**.
- Jobs running on a specified cluster and byte counter data – **show monitor hadoop cluster counters**.
- Jobs that previously ran on a specified cluster – **show monitor hadoop cluster history**.
  - Includes jobs that ran since the monitor was enabled, the switch was reloaded, or the job history was cleared (**clear monitor hadoop job-history**).
- Jobs running on a specified cluster – **show monitor hadoop cluster jobs**.
- Jobs that ran on all configured clusters is accessed through **show monitor hadoop history**.
  - Includes jobs that ran since the monitor was enabled, the switch was reloaded, or the job history was cleared (**clear monitor hadoop job-history**).
- Jobs running on a specified TaskTracker and byte counter data – **show monitor hadoop tasktracker counters**.

Examples

- This command displays the jobs that are running on all monitored clusters.

```
switch>show monitor hadoop
Last updated: 2013-10-06 18:14:23
Currently running jobs: 4
JobId   Job Name         Cluster   Maps(#/%)  Reduces(#/%)  Start Time
------- ---------------- --------- ---------- ------------- -------------------
1       ReallyAVeryLon\  Cluster0  2/12.34%   0/13.45%      2013-10-06 17:56:03
        gNameForAJob1
2       ShortName2       Cluster0  2/24.68%   0/26.90%      2013-10-06 17:37:43
510001  ReallyAVeryLon\  Cluster1  2/12.34%   0/13.45%      2013-10-06 17:56:03
        gNameForAJob11
510002  ShortName12      Cluster1  2/24.68%   0/26.90%      2013-10-06 17:37:43

switch>
```

- This command displays data the jobs that previously ran on connected Hadoop clusters.

```
switch>show monitor hadoop history
Job history for all clusters:
JobId  Job Name         Cluster   Start Time   End Time     Bytes In  Bytes Out
------ ---------------- --------- ----------- ------------ ---------- ---------
2      AReallyBigHist\  Cluster0  2013-10-06   2013-10-09   26.08GB   13.04GB
       oricalJobName              17:41:03     06:47:43
442    AReallyBigHist\  Cluster1  2013-10-06   2013-10-09   26.08GB   13.04GB
       oricalJobName              17:41:03     06:47:43
442    AReallyBigHist\  Cluster1  2013-10-06   2013-10-09   26.08GB   13.04GB
       oricalJobName              17:41:03     06:47:43
2      AReallyBigHist\  Cluster0  2013-10-06   2013-10-09   26.08GB   13.04GB
       oricalJobName              17:41:03     06:47:43
441    HistoryJob1      Cluster1  2013-10-06   2013-10-08   26.08GB   13.04GB
                                  17:57:43     00:31:03
1      HistoryJob1      Cluster0  2013-10-06   2013-10-08   26.08GB   13.04GB
                                  17:57:43     00:31:03

switch>
```

- This command displays jobs running on cluster ***Cluster0*** and byte counters for each job.

```
switch>show monitor hadoop cluster Cluster0 counters
Last updated: 2013-10-06 18:14:23
Counters for currently running jobs on cluster: Cluster0
 JobId   Job Name          User       Bytes In   Bytes Out  Start Time
------- ----------------- ---------- ---------- ---------- -------------------
 2       ShortName2        JobUser2   37.36GB    76.29MB    2013-10-06 17:37:43
 1       ReallyAVeryLon\   JobUser1   37.36GB    76.29MB    2013-10-06 17:56:03
         gNameForAJob1
switch>
```

43.3.4 Job Data

The following commands display information about jobs that are running or previously ran on monitored clusters. Available data include job identifiers, JobTracker ID, start time, stop time, data consumption, and progress statistics.

- Data consumption, start and stop times, and JobTracker ID for a specific job – **show monitor hadoop cluster history jobs**.
- Data consumption, start and stop times, priority, JobTracker ID, and progress statistics for a specified job – **show monitor hadoop cluster jobs <job number>**.
- HDFS (Hadoop Distributed File System) data consumption and and shuffle byte counters for a specified job – **show monitor hadoop cluster jobs counter**.
- Data through and start time for jobs running on all monitored clusters – **show monitor hadoop counters**.
- Progress statistics and start times are available for jobs running on specified TaskTracker – **show monitor hadoop tasktracker jobs**.
- Job progress and byte counts of jobs running on a specified Hadoop cluster – **show monitor hadoop tasktracker running-tasks**.
- Progress statistics, HDFS data consumption, start time, and progress information for the specified task of a running job – **show monitor hadoop tasktracker running-tasks cluster job task**.
- Data consumption and start times for jobs running on a specified TaskTracker – **show monitor hadoop tasktracker counters**.

Examples

- This command displays information about job 1 that ran on cluster Cluster0.

```
switch>show monitor hadoop cluster Cluster0 history job 1
Job history data for job: HistoryJob1
  Cluster                  : Cluster0
  Job Id                   : 1
  JT Id                    : 201310110013
  User                     : HistoryUser1
  Job start time           : 2013-10-06 17:57:43
  Job end time             : 2013-10-08 00:31:03
Per Interface job counters:
Interface       TaskTracker        Bytes In       Bytes Out
--------------- ------------------ -------------- ---------
Ethernet7       TaskTracker2       26.08GB        13.04GB

switch>
```

- This command displays information about job 1 that is running on cluster ***Cluster0***.

```
switch>show monitor hadoop cluster Cluster0 jobs 1
Last updated: 2013-10-06 18:14:23
Information for job: ReallyAVeryLongNameForAJob1 running on cluster: Cluster0
  Cluster                  : Cluster0
  Id                       : 1
  Name                     : ReallyAVeryLongNameForAJob1
  User                     : JobUser1
  Priority                 : veryHigh
  Running state            : running
  Number of map tasks      : 2
  Number of reduce tasks   : 0
  Start time               : 2013-10-06 17:56:03
  Bytes In                 : 37.36GB
  Bytes Out                : 76.29MB
  Map Progress             : 12.34%
  Reduce Progress          : 13.45%
  Cleanup Progress         : 14.56%
  Setup Progress           : 15.67%

switch>
```

- This command displays data for jobs running on ***TaskTracker1***.

```
switch>show monitor hadoop tasktracker host TaskTracker1 jobs
Last updated: 2013-10-06 18:14:23
Running job for TaskTracker: TaskTracker1
JobId  Job Name         Cluster   Maps(#/%)   Reduces(#/%)  Start Time
------ ---------------- --------- ---------- -------------- -------------------
1      ReallyAVeryLon\  Cluster0  2/12.34%    0/13.45%      2013-10-06 17:56:03
       gNameForAJob1
2      ShortName2       Cluster0  2/24.68%    0/26.90%      2013-10-06 17:37:43

switch>
```

43.3.5 TaskTracker Lists

These commands display lists of TaskTrackers that are active on monitored clusters:

- TaskTrackers on a specified cluster – **show monitor hadoop cluster tasktracker**.
- TaskTrackers on all monitored clusters – **show monitor hadoop tasktracker all**.

Example

- This command displays the TaskTrackers on the ***Cluster0*** cluster.

```
switch>show monitor hadoop cluster Cluster0 tasktracker
Last updated: 2013-10-06 18:14:23
Total 2 TaskTrackers on cluster Cluster0:
   Node               IP Address       Interface           Maps    Reduces
------------------ ---------------- ------------------- ---------- -------
   TaskTracker1       10.100.0.1       Ethernet7           4       0
   TaskTracker2       10.100.0.2       Port-Channel7       4       0

switch>
```

43.3.6 TaskTracker Connection and Activity

The following TaskTracker connection and activity data is available through these commands:

- Connection and activity information for TaskTrackers on a specified cluster or accessed through a specified interface – **show monitor hadoop tasktracker status**.
- Data consumption for TaskTrackers connected to monitored clusters – **show monitor hadoop tasktracker all counters**.

Example

- This command displays connection and activity data for TaskTracker on the ***TaskTracker1*** node.

```
switch>show monitor hadoop tasktracker host TaskTracker1 status
Last updated: 2013-10-06 18:14:23
  TaskTracker              : TaskTracker1
  IP Address               : 10.100.0.1
  Interface                : Ethernet7
  State                    : active
  Running jobs             : 2
  Running tasks            : 4
  Map Tasks                : 4
  Reduce Tasks             : 0
  Total bytes read         : 2.08GB
  Total bytes written      : 4.24MB

switch>
```

43.3.7 Data Bursts

The **show monitor hadoop traffic burst** command displays the largest data bursts for jobs running on a specified cluster or accessed through a specified node or interface. A data burst is the data consumed during a polling interval.

Example

- This command displays traffic burst data for all running jobs that are accessible through port channel interface 7.

```
switch>show monitor hadoop traffic burst interface Port-Channel 7
Last updated: 2013-10-06 18:14:23
Bursts on Interface: 'Port-Channel7' in cluster: Cluster0
Top 2 input bursts:
   JobId       Job Name              Burst     Time
----------- -------------------- ------------ -------------------
   1           ShortName             3.07GB    2013-10-06 17:57:43
   2           ReallyAVeryLon\       6.15GB    2013-10-06 17:41:03
               gNameForAJob

Top 2 output bursts:
   JobId       Job Name              Burst     Time
----------- -------------------- ------------ -------------------
   1           ShortName             4.10GB    2013-10-06 17:55:13
   2           ReallyAVeryLon\       8.20GB    2013-10-06 17:36:03
               gNameForAJob
```

43.4 MapReduce Tracer Command Descriptions

Global Configuration Commands

- monitor hadoop

Clear Hadoop Monitor Commands

- clear monitor hadoop burst-counters
- clear monitor hadoop job-history

Display Commands

- show monitor hadoop
- show monitor hadoop cluster all
- show monitor hadoop cluster counters
- show monitor hadoop cluster history
- show monitor hadoop cluster history jobs
- show monitor hadoop cluster jobs
- show monitor hadoop cluster jobs <job number>
- show monitor hadoop cluster jobs counter
- show monitor hadoop cluster status
- show monitor hadoop cluster tasktracker
- show monitor hadoop counters
- show monitor hadoop history
- show monitor hadoop status
- show monitor hadoop tasktracker all
- show monitor hadoop tasktracker all counters
- show monitor hadoop tasktracker counters
- show monitor hadoop tasktracker jobs
- show monitor hadoop tasktracker running-tasks
- show monitor hadoop tasktracker running-tasks cluster job task
- show monitor hadoop tasktracker status
- show monitor hadoop traffic burst

Hadoop Commands

- cluster (Monitor Hadoop)
- shutdown (Monitor-Hadoop)

Hadoop-Cluster Commands

- description (Monitor Hadoop Cluster)
- interval (Monitor Hadoop Cluster)
- jobtracker (Monitor Hadoop Cluster)
- shutdown (Monitor Hadoop Cluster)
- tasktracker (Monitor Hadoop Cluster)

clear monitor hadoop burst-counters

The **clear monitor hadoop burst-counters** command resets MapReduce Tracer burst counters for all jobs running on specified clusters.

Command Mode

Privileged EXEC

Command Syntax

clear monitor hadoop burst-counters [***CLUSTERS***]

Parameters

- ***CLUSTERS*** Hadoop clusters for which command displays data. Options include:
  - <no parameter> all clusters.
  - **cluster** *c_name* Cluster name.

Example

- This command clears the burst counters for all jobs running on CL2 cluster.

```
switch#clear monitor hadoop burst-counters cluster CL2
Cleared burst counters
switch#
```

clear monitor hadoop job-history

The **clear monitor hadoop job-history** command resets the job history database for all specified clusters.

Command Mode

Privileged EXEC

Command Syntax

clear monitor hadoop job-history [***CLUSTERS***]

Parameters

- ***CLUSTERS*** Hadoop clusters for which command displays data. Options include:
  - <no parameter> all clusters.
  - **cluster** *c_name* Cluster name.

Example

- This command clears the job history on the CL2 cluster.

```
switch#clear monitor hadoop job-history cluster CL2
Cleared job history
switch#
```

cluster (Monitor Hadoop)

The **cluster** command is a monitor-hadoop command that places the switch in monitor-hadoop-cluster mode for configuring and enabling a MapReduce Tracer monitor for a Hadoop cluster. The command either accesses an existing monitor configuration or creates a monitor.

Monitor-hadoop-cluster configuration mode is not a group change mode; ***running-config*** is changed immediately upon entering commands. Exiting monitor-hadoop-cluster mode does not affect ***running-config***. The **exit** command returns the switch to monitor-hadoop configuration mode.

The configuration mode monitor is enabled by **no shutdown (Monitor Hadoop Cluster)**. Enabling a monitor also requires that MapReduce Tracer is globally enabled (**no shutdown (Monitor-Hadoop)**.

The **no cluster** and **default cluster** commands remove the specified Hadoop cluster configuration from ***running-config***.

Command Mode

Monitor-hadoop Configuration

Command Syntax

```
cluster cluster_name
no cluster cluster_name
default cluster cluster_name
```

Parameters

- *cluster_name* Hadoop cluster name.

Related Commands

- **monitor hadoop** places the switch in monitor-hadoop configuration mode.

Commands Available in Monitor-hadoop-cluster Configuration Mode

- **description (Monitor Hadoop Cluster)**
- **interval (Monitor Hadoop Cluster)**
- **jobtracker (Monitor Hadoop Cluster)**
- **shutdown (Monitor Hadoop Cluster)**
- **tasktracker (Monitor Hadoop Cluster)**

Examples

- These commands create the CL2 monitor and enters monitor-hadoop-cluster mode for the monitor.

```
switch(config)#monitor hadoop
switch(config-monitor-hadoop)#cluster CL2
switch(config-monitor-hadoop-CL2)#show active
monitor hadoop
   cluster CL2
switch(config-monitor-hadoop-CL2)#
```

- These commands exit monitor-hadoop-cluster mode.

```
switch(config-monitor-hadoop-CL2)#exit
switch(config-monitor-hadoop)#show active
 monitor hadoop
   cluster CL2
switch(config-monitor-hadoop)#
```

- These commands remove the CL2 monitor.

```
switch(config-monitor-hadoop)#no cluster CL2
switch(config-monitor-hadoop)#show active
switch(config-monitor-hadoop)#
```

description (Monitor Hadoop Cluster)

The **description** command adds a text string to the configuration mode MapReduce Tracer cluster monitor. The string has no functional impact on the monitor.

The **no description** and **default description** commands remove the text string from the configuration mode monitor by removing the corresponding **description** command from ***running-config***.

Command Mode

Monitor-hadoop-cluster Configuration

Command Syntax

```
description label_text
no description
default description
```

Parameters

- *label_text* character string assigned to the monitor configuration.

Related Commands

- **cluster (Monitor Hadoop)** places the switch in monitor-hadoop-cluster configuration mode.

Examples

- These commands add description text to the CL2 monitor.

```
switch(config)#monitor hadoop
switch(config-monitor-hadoop)#cluster CL2
switch(config-monitor-hadoop-CL2)#description First Cluster
monitor hadoop
   cluster CL2
      description First Cluster
      jobtracker host 10.3.3.3 user JANE
switch(config-monitor-hadoop-CL2)#
```

interval (Monitor Hadoop Cluster)

The **interval** command specifies the polling interval between queries to the Hadoop cluster JobTracker specifed by configuration mode statements. The switch polls a cluster's JobTracker to update its list of active TaskTracker nodes and the statistics of jobs running in the cluster. This command controls the frequency of these polls. The default interval is 10 seconds.

The **no interval** and **default interval** commands restore the default interval of 10 seconds by removing the **interval** command from ***running-config***.

Command Mode

Monitor-hadoop-cluster Configuration

Command Syntax

```
interval period
no interval
default interval
```

Parameters

- *period* interval (seconds) between JobTracker polls. Value ranges from 1 to 600. Default is 10.

Related Commands

- **cluster (Monitor Hadoop)** places the switch in monitor-hadoop-cluster configuration mode.

Example

- This command sets the JobTracker polling interval to 25 seconds for the CL2 cluster configuration.

```
switch(config)#monitor hadoop
switch(config-monitor-hadoop)#cluster CL2
switch(config-monitor-hadoop-CL2)#interval 25
switch(config-monitor-hadoop-CL2)#show active
monitor hadoop
   cluster CL2
      interval 25
switch(config-monitor-hadoop-CL2)#
```

jobtracker (Monitor Hadoop Cluster)

The **jobtracker** command specifies JobTracker access parameters for the cluster monitored by configuration mode monitor statements. A cluster's JobTracker is located on the master node and schedules work to the cluster's TaskTracker nodes.

Parameters required to communicate with a JobTracker include its node location (IPv4 address or hostname), RPC port, and username. The default RPC port is 8021. Location and username parameters do not have default values and must be explicity configured. A JobTracker command that specifies a partial parameter list modifies the existing corresponding **jobtracker** statement in ***running-config***.

The **no jobtracker** and **default jobtracker** commands perform the following:

- removes the **jobtracker** statement from ***running config*** when it lists all command parameters.
- modifies the existing **jobtracker** statement when it lists a subset of command parameters.

Command Mode

Monitor-hadoop-cluster Configuration

Command Syntax

```
jobtracker [LOCATION] [PORT] [USER]
no jobtracker [LOCATION] [PORT] [USER]
default jobtracker [LOCATION] [PORT] [USER]
```

All parameters can be placed in any order.

Parameters

- ***LOCATION*** Address or hostname of JobTracker node. Options include:
  - <no parameter> location remains undefined or unchanged from a previous configuration.
  - **host** *ipv4_addr* IPv4 address of master (JobTracker) node.
  - **host** *hostname* Hostname of master (JobTracker) node.
- ***PORT*** JobTracker RPC port number. Default value is 8021. Options include:
  - <No parameter> Port number remains unchanged from previous configuration.
  - **rdp-port** *port_num* Port number of master (JobTracker) node. Value ranges from 1 to 65535.
- ***USER*** Username that accesses JobTracker node. Options include:
  - <No parameter> username remains undefined or unchanged from previous configuration.
  - **username** *name_string* JobTracker username.

Related Commands

- **cluster (Monitor Hadoop)** places the switch in monitor-hadoop-cluster configuration mode.

Example

- For the CL2 cluster configuration, these commands establish a connection to the JobTracker node at ***10.4.4.4*** with the username ***account1***. The default RPC port (8021) is implicitly specified.

```
switch(config)#monitor hadoop
switch(config-monitor-hadoop)#cluster CL2
switch(config-monitor-hadoop-CL2)#jobtracker host 10.4.4.4 username account1
switch(config-monitor-hadoop-CL2)#show active
monitor hadoop
   cluster CL2
      jobtracker host 10.4.4.4 user account1
switch(config-monitor-hadoop-CL2)#
```

- These commands modify the JobTracker configuration to specify an RPC port of 9000.

```
switch(config-monitor-hadoop-CL2)#jobtracker rpc-port 9000
switch(config-monitor-hadoop-CL2)#show active
monitor hadoop
   cluster CL2
      jobtracker host 10.4.4.4 rpc-port 9000 user account1
switch(config-monitor-hadoop-CL2)#
```

monitor hadoop

The **monitor hadoop** command places the switch in monitor-hadoop configuration mode for configuring MapReduce Tracer monitors. A MapReduce Tracer monitor interacts with Hadoop cluster nodes that are directly attached to the switch. Tasks that the switch can perform through this interaction include:

- compile a list of nodes in the cluster
- compile a list of jobs the nodes are running
- download progress of the running jobs

Monitor-hadoop configuration mode is not a group change mode; ***running-config*** is changed immediately upon entering commands. Exiting monitor-hadoop configuration mode does not affect ***running-config***. The **exit** command returns the switch to global configuration mode. MapReduce Tracer is enabled in monitor-hadoop mode through the **no shutdown (Monitor-Hadoop)** command.

The **no monitor hadoop** and **default monitor hadoop** commands delete previously configured **monitor hadoop mode** configuration commands.

Command Mode

Global Configuration

Command Syntax

```
monitor hadoop
no monitor hadoop
default monitor hadoop
```

Commands Available in Monitor-hadoop Configuration Mode

- **cluster (Monitor Hadoop)**
- **shutdown (Monitor-Hadoop)**

Examples

- These commands place the switch in monitor-hadoop configuration mode.

```
switch(config)#monitor hadoop
switch(config-monitor-hadoop)#
```

- This command exits monitor-hadoop mode.

```
switch(config-monitor-hadoop)#exit
switch(config)#
```

- This command deletes all previously configured monitor-hadoop configuration mode commands.

```
switch(config)#no monitor hadoop
switch(config)#
```

show monitor hadoop

The **show monitor hadoop** command displays a list of jobs that are running on all monitored Hadoop clusters.

Command Mode

EXEC

Command Syntax

show monitor hadoop

Example

- This command displays the jobs that are running on all monitored clusters.

```
switch>show monitor hadoop
Last updated: 2013-10-06 18:14:23
Currently running jobs: 4
JobId   Job Name         Cluster   Maps(#/%)  Reduces(#/%)  Start Time
------- ---------------- --------- ---------- ------------- -------------------
1       ReallyAVeryLon\  Cluster0  2/12.34%   0/13.45%      2013-10-06 17:56:03
        gNameForAJob1
2       ShortName2       Cluster0  2/24.68%   0/26.90%      2013-10-06 17:37:43
510001  ReallyAVeryLon\  Cluster1  2/12.34%   0/13.45%      2013-10-06 17:56:03
        gNameForAJob11
510002  ShortName12      Cluster1  2/24.68%   0/26.90%      2013-10-06 17:37:43

switch>
```

show monitor hadoop cluster all

The **show monitor hadoop cluster all** command displays configuration and connection information for all monitored Hadoop clusters.

Command Mode

EXEC

Command Syntax

show monitor hadoop cluster all

Example

- This command displays configuration and connection data for all connected Hadoop clusters.

```
switch>show monitor hadoop cluster all
Total number of clusters configured: 3
  Cluster                     : Cluster0
  Admin status                : Enabled
  JobTracker host             : host0
  JobTracker RPC port         : 9000
  JobTracker user             : user0
  JobTracker polling interval : 100 seconds
  TaskTracker HTTP port       : 8800
  Operational status          : Enabled
  Active TaskTrackers         : 31
  Blacklisted TaskTrackers    : 1
  Decommissioned TaskTrackers : 1
  Tracker expiry interval     : 20.0
  Map slots (used/total)      : 10/100
  Reduce slots (used/total)   : 11/110
  JobTracker heap size        : 1.04GB (max: 2.08GB)

  Cluster                     : Cluster1
  Admin status                : Enabled
  JobTracker host             : host1
  JobTracker RPC port         : 9001
  JobTracker user             : user1
  JobTracker polling interval : 101 seconds
  TaskTracker HTTP port       : 8801
  Operational status          : Enabled
  Active TaskTrackers         : 32
  Blacklisted TaskTrackers    : 0
  Decommissioned TaskTrackers : 0
  Tracker expiry interval     : 40.0
  Map slots (used/total)      : 20/200
  Reduce slots (used/total)   : 22/220
  JobTracker heap size        : 2.09GB (max: 4.15GB)

  Cluster                     : Cluster2
  Admin status                : Disabled
  JobTracker host             : host2
  JobTracker RPC port         : 9002
  JobTracker user             : user2
  JobTracker polling interval : 102 seconds
  TaskTracker HTTP port       : 8802
  Operational status          : Disabled
```

show monitor hadoop cluster counters

The **show monitor hadoop cluster counters** command displays a list of jobs running on the specified Hadoop cluster and data consumption associated with these jobs.

Command Mode

EXEC

Command Syntax

show monitor hadoop cluster *c_name* **counters**

Parameters

- *c_name* Cluster name.

Examples

- This command displays jobs running on cluster ***Cluster0***.

```
switch>show monitor hadoop cluster Cluster0 counters
Last updated: 2013-10-06 18:14:23
Counters for currently running jobs on cluster: Cluster0
 JobId   Job Name          User       Bytes In   Bytes Out  Start Time
------- ----------------- ---------- ---------- ---------- -------------------
 2       ShortName2        JobUser2   37.36GB    76.29MB    2013-10-06 17:37:43
 1       ReallyAVeryLon\   JobUser1   37.36GB    76.29MB    2013-10-06 17:56:03
         gNameForAJob1
switch>
```

show monitor hadoop cluster history

The **show monitor hadoop cluster history** command displays all jobs that ran on the specified cluster. The list includes all jobs that ran since the switch was reloaded, the job history was cleared (**clear monitor hadoop job-history**), or MapReduce Tracer was enabled.

Command Mode

EXEC

Command Syntax

show monitor hadoop cluster *c_name* **history**

Parameters

- *c_name* Cluster name.

Examples

- This command displays the jobs that were ran on the cluster named ***Cluster0***.

```
switch>show monitor hadoop cluster Cluster0 history
Jobs history on cluster: Cluster0
 JobId    Job Name           Start Time     End Time       Bytes In   Bytes Out
-------- ------------------ ------------- -------------- ------------ ---------
 2        AReallyBigHist\    2013-10-06     2013-10-09     26.08GB    13.04GB
          oricalJobName      17:41:03       06:47:43
 2        AReallyBigHist\    2013-10-06     2013-10-09     26.08GB    13.04GB
          oricalJobName      17:41:03       06:47:43
 1        HistoryJob1        2013-10-06     2013-10-08     26.08GB    13.04GB
                             17:57:43       00:31:03

switch>
```

show monitor hadoop cluster history jobs

The **show monitor hadoop cluster history jobs** command displays data about the specified job. Hadoop jobs are identified by job number and the cluster that ran the job.

Data that the command returns include job identifiers, JobTracker ID, start and stop times, and data consumption.

Command Mode

EXEC

Command Syntax

show monitor hadoop cluster *c_name* **history jobs** *job_number*

Parameters

- *c_name* Cluster name.
- *job_number* Job number. Value ranges from ***0*** to ***2147483647.***

Examples

- This command displays information about job 1 that ran on cluster Cluster0.

```
switch>show monitor hadoop cluster Cluster0 history job 1
Job history data for job: HistoryJob1
  Cluster                  : Cluster0
  Job Id                   : 1
  JT Id                    : 201310110013
  User                     : HistoryUser1
  Job start time           : 2013-10-06 17:57:43
  Job end time             : 2013-10-08 00:31:03
Per Interface job counters:
Interface       TaskTracker        Bytes In       Bytes Out
--------------- ------------------ -------------- ---------
Ethernet7       TaskTracker2       26.08GB        13.04GB

switch>
```

show monitor hadoop cluster jobs

The **show monitor hadoop cluster jobs** command displays a list of jobs that are running on the specified cluster.

Command Mode

EXEC

Command Syntax

show monitor hadoop cluster *c_name* **jobs**

Parameters

- *c_name* Cluster name.

Examples

- This command displays the list of jobs running on cluster ***Cluster0***.

```
switch>show monitor hadoop cluster Cluster0 jobs
Last updated: 2013-10-06 18:14:23
Currently running jobs on cluster: Cluster0
 JobId    Job Name           User        Maps     Reduces   Start Time
-------- ------------------ ----------- ------- ----------- -------------------
 2        ShortName2         JobUser2    2        0         2013-10-06 17:37:43
 1        ReallyAVeryLon\    JobUser1    2        0         2013-10-06 17:56:03
          gNameForAJob1
switch>
```

show monitor hadoop cluster jobs <job number>

The **show monitor hadoop cluster jobs <job number>** command displays information about the specified job. Hadoop jobs are identified by job ID and the cluster that is running the job.

Data that the command returns include time of update, job identifiers, start times, data consumption, and completion progress.

Command Mode

EXEC

Command Syntax

show monitor hadoop cluster *c_name* **jobs** *job_number*

Parameters

- *c_name* Cluster name.
- *job_number* Job number. Value ranges from ***0*** to ***2147483647.***

Examples

- This command displays information about job 1 that is running on cluster ***Cluster0***.

```
switch>show monitor hadoop cluster Cluster0 jobs 1
Last updated: 2013-10-06 18:14:23
Information for job: ReallyAVeryLongNameForAJob1 running on cluster: Cluster0
  Cluster                  : Cluster0
  Id                       : 1
  Name                     : ReallyAVeryLongNameForAJob1
  User                     : JobUser1
  Priority                 : veryHigh
  Running state            : running
  Number of map tasks      : 2
  Number of reduce tasks   : 0
  Start time               : 2013-10-06 17:56:03
  Bytes In                 : 37.36GB
  Bytes Out                : 76.29MB
  Map Progress             : 12.34%
  Reduce Progress          : 13.45%
  Cleanup Progress         : 14.56%
  Setup Progress           : 15.67%

switch>
```

show monitor hadoop cluster jobs counter

The **show monitor hadoop cluster jobs counter** command displays data consumption and progress statistics for the specified job.

Command Mode

EXEC

Command Syntax

show monitor hadoop cluster *c_name* **jobs** *job_number* **counter**

Parameters

- *c_name* Cluster name.
- *job_number* Job number. Value ranges from ***0*** to ***2147483647.***

Examples

- This command displays byte counters for the job named ***1*** that is running on the cluster named ***Cluster0***.

```
switch>show monitor hadoop cluster Cluster0 jobs 1 counters
Last updated: 2013-10-06 18:14:23
  Cluster                 : Cluster0
  Job Name                : ReallyAVeryLongNameForAJob1
  Job Id                  : 1
Interface         HDFS Bytes Read    HDFS Bytes Written    Reduce Shuffle Bytes
----------------- ------------------ --------------------- --------------------
Port-Channel8     4.14GB             8.48MB                12.72MB
Port-Channel9     6.21GB             12.72MB               19.07MB
Ethernet8         3.10GB             6.36MB                9.54MB
Ethernet9         5.17GB             10.60MB               15.89MB
Port-Channel7     2.07GB             4.24MB                6.36MB
Ethernet10        7.24GB             14.83MB               22.25MB
Port-Channel10    8.28GB             16.95MB               25.43MB
Ethernet7         1.03GB             2.12MB                3.18MB

switch>
```

show monitor hadoop cluster status

The **show monitor hadoop cluster status** command displays configuration and connection information for the specified cluster.

Command Mode

EXEC

Command Syntax

show monitor hadoop cluster *c_name* **status**

Parameters

- *c_name* Cluster name.

Example

- This command displays configuration and connection data for the ***Cluster0*** cluster.

```
switch>show monitor hadoop cluster Cluster0 status
Last updated: 2013-10-06 18:14:23
Cluster status for cluster:  Cluster0
  Admin status                : Enabled
  JobTracker host             : host0
  JobTracker RPC port         : 9000
  JobTracker user             : user0
  JobTracker polling interval : 100 seconds
  TaskTracker HTTP port       : 8800
  Operational status          : Enabled
  Active TaskTrackers         : 31
  Blacklisted TaskTrackers    : 1
  Decommissioned TaskTrackers : 1
  Tracker expiry interval     : 20.0
  Map slots (used/total)      : 10/100
  Reduce slots (used/total)   : 11/110
  JobTracker heap size        : 1.04GB (max: 2.08GB)
switch>
```

show monitor hadoop cluster tasktracker

The **show monitor hadoop cluster tasktracker** command displays a list of TaskTrackers in the specified cluster. The IP address and access interface is included in the table.

Command Mode

EXEC

Command Syntax

show monitor hadoop cluster *c_name* **tasktracker**

Parameters

- *c_name* Cluster name.

Example

- This command displays the TaskTrackers on the ***Cluster0*** cluster.

```
switch>show monitor hadoop cluster Cluster0 tasktracker
Last updated: 2013-10-06 18:14:23
Total 2 TaskTrackers on cluster Cluster0:
   Node               IP Address       Interface           Maps    Reduces
------------------ --------------- ------------------- ---------- -------
   TaskTracker1       10.100.0.1       Ethernet7           4       0
   TaskTracker2       10.100.0.2       Port-Channel7       4       0

switch>
```

show monitor hadoop counters

The **show monitor hadoop counters** command displays byte counter data for all jobs running on clusters for which MapReduce Tracer is configured.

Command Mode

EXEC

Command Syntax

```
show monitor hadoop counters
```

Examples

- This command displays byte counter data for all jobs running on clusters that the switch is accessing through MapReduce Tracer.

```
switch>show monitor hadoop counters
Last updated: 2013-10-06 18:14:23
Counters for running jobs:
JobId    Job Name          Cluster    Bytes In   Bytes Out  Start Time
------- ----------------- ---------- ---------- ----------- -------------------
510002   ShortName12       Cluster1   37.36GB    76.29MB    2013-10-06 17:37:43
510001   ReallyAVeryLon\   Cluster1   37.36GB    76.29MB    2013-10-06 17:56:03
         gNameForAJob11
2        ShortName2        Cluster0   37.36GB    76.29MB    2013-10-06 17:37:43
1        ReallyAVeryLon\   Cluster0   37.36GB    76.29MB    2013-10-06 17:56:03
         gNameForAJob1
switch>
```

show monitor hadoop history

The **show monitor hadoop history** command displays jobs that ran on clusters for which MapReduce Tracer is configured. The list includes all jobs that ran since the switch was reloaded, MapReduce Tracer was enabled, or the job history was cleared (**clear monitor hadoop job-history**).

Command Mode

EXEC

Command Syntax

show monitor hadoop history

Example

- This command displays data that jobs that previously ran on connected Hadoop clusters.

```
switch>show monitor hadoop history
Job history for all clusters:
JobId  Job Name         Cluster   Start Time   End Time     Bytes In  Bytes Out
------ ---------------- --------- ----------- ------------ ---------- ---------
2      AReallyBigHist\  Cluster0  2013-10-06   2013-10-09   26.08GB   13.04GB
       oricalJobName              17:41:03     06:47:43
442    AReallyBigHist\  Cluster1  2013-10-06   2013-10-09   26.08GB   13.04GB
       oricalJobName              17:41:03     06:47:43
442    AReallyBigHist\  Cluster1  2013-10-06   2013-10-09   26.08GB   13.04GB
       oricalJobName              17:41:03     06:47:43
2      AReallyBigHist\  Cluster0  2013-10-06   2013-10-09   26.08GB   13.04GB
       oricalJobName              17:41:03     06:47:43
441    HistoryJob1      Cluster1  2013-10-06   2013-10-08   26.08GB   13.04GB
                                  17:57:43     00:31:03
1      HistoryJob1      Cluster0  2013-10-06   2013-10-08   26.08GB   13.04GB
                                  17:57:43     00:31:03

switch>
```

show monitor hadoop status

The **show monitor hadoop status** command displays system status for MapReduce Tracer.

Command Mode

EXEC

Command Syntax

show monitor hadoop status

Example

- This command displays MapReduce Tracer status for all connected clusters and TaskTrackers.

```
switch>show monitor hadoop status
Last updated: 2013-10-06 18:14:23
Mapreduce Tracer status:
  Admin status                          : Enabled
  Operational status                    : Enabled
  Number of clusters configured         : 3
  Number of local TaskTrackers          : 4
  Number of jobs running locally        : 4

switch>
```

show monitor hadoop tasktracker all

The **show monitor hadoop tasktracker all** command displays a list of TaskTrackers that are on all monitored Hadoop clusters.

Command Mode

EXEC

Command Syntax

```
show monitor hadoop tasktracker all
```

Examples

- This command displays the TaskTrackers of all monitored clusters that are connected to the switch.

```
switch>show monitor hadoop tasktracker all
Last updated: 2013-10-06 18:14:23
All local TaskTrackers:
  Node             Cluster      IP Address     Interface         Maps   Reduces
---------------- ------------ -------------- ----------------- -------- -------
  TaskTracker1     Cluster0     10.100.0.1     Ethernet7         4      0
  TaskTracker3     Cluster1     10.100.0.3     Ethernet8         4      0
  TaskTracker2     Cluster0     10.100.0.2     Port-Channel7     4      0
  TaskTracker4     Cluster1     10.100.0.4     Port-Channel8     4      0

switch>
```

show monitor hadoop tasktracker all counters

The **show monitor hadoop tasktracker all counters** command displays byte counters for the TaskTrackers of all monitored Hadoop clusters.

Command Mode

EXEC

Command Syntax

```
show monitor hadoop tasktracker all counters
```

Examples

- This command displays byte counter data for the TaskTrackers servicing all MapReduce Tracer Hadoop clusters.

```
switch>show monitor hadoop tasktracker all counters
Last updated: 2013-10-06 18:14:23
Counters for all TaskTrackers:
  Node             IP Address     Interface         Bytes Read    Bytes Written
---------------- -------------- ----------------- --------------- -------------
  TaskTracker1     10.100.0.1     Ethernet7         2.08GB        4.24MB
  TaskTracker3     10.100.0.3     Ethernet8         6.23GB        12.72MB
  TaskTracker2     10.100.0.2     Port-Channel7     4.15GB        8.48MB
  TaskTracker4     10.100.0.4     Port-Channel8     8.30GB        16.95MB

switch>
```

show monitor hadoop tasktracker counters

The **show monitor hadoop tasktracker counters** command displays a list of jobs running on the specified TaskTracker and output from byte counters associated with these jobs.

Command Mode

EXEC

Command Syntax

show monitor hadoop tasktracker *NODES* counters

Parameters

- ***NODES*** TaskTracker node access point. Options include:
  - **host** *hostname* Node name.
  - **interface ethernet** *e_range* Ethernet interfaces through which node connects.
  - **interface port-channel** *p_range* Port channel interfaces through which node connects.

Examples

- This command displays the jobs running on the TaskTracker on the ***TaskTracker1*** node.

```
switch>show monitor hadoop tasktracker host TaskTracker1 counters
Last updated: 2013-10-06 18:14:23
Running job for TaskTracker: TaskTracker1
 JobId   Job Name          Cluster    Bytes In   Bytes Out  Start Time
------- ----------------- ---------- ---------- ---------- -------------------
 2       ShortName2        Cluster0   37.36GB    76.29MB    2013-10-06 17:37:43
 1       ReallyAVeryLon\   Cluster0   37.36GB    76.29MB    2013-10-06 17:56:03
         gNameForAJob1

Note: these counters are derived from Hadoop counters and represent approximate
network bandwidth utilization

switch>
```

- This command displays jobs running on TaskTrackers accessed through Ethernet interfaces 7 and 8.

```
switch>show monitor hadoop tasktracker interface Ethernet 7,8 counters
Last updated: 2013-10-06 18:14:23
Running job for TaskTracker: TaskTracker3
JobId    Job Name          Cluster    Bytes In   Bytes Out  Start Time
------- ----------------- ---------- ---------- ----------- -------------------
510002   ShortName12       Cluster1   37.36GB    76.29MB    2013-10-06 17:37:43
510001   ReallyAVeryLon\   Cluster1   37.36GB    76.29MB    2013-10-06 17:56:03
         gNameForAJob11

Note: These counters are derived from Hadoop counters and represent approximate
network bandwidth utilization

Running job for TaskTracker: TaskTracker1
 JobId   Job Name          Cluster    Bytes In   Bytes Out  Start Time
------- ----------------- ---------- ---------- ----------- -------------------
 2       ShortName2        Cluster0   37.36GB    76.29MB    2013-10-06 17:37:43
 1       ReallyAVeryLon\   Cluster0   37.36GB    76.29MB    2013-10-06 17:56:03
         gNameForAJob1

Note: these counters are derived from Hadoop counters and represent approximate
network bandwidth utilization

switch>
```

- This command displays jobs running on TaskTrackers accessed through port channel interface 7.

```
switch>show monitor hadoop tasktracker interface Port-Channel 7 counters
Last updated: 2013-10-06 18:14:23
Running job for TaskTracker: TaskTracker2
 JobId   Job Name          Cluster    Bytes In   Bytes Out  Start Time
------- ----------------- ---------- ---------- ----------- -------------------
 2       ShortName2        Cluster0   37.36GB    76.29MB    2013-10-06 17:37:43
 1       ReallyAVeryLon\   Cluster0   37.36GB    76.29MB    2013-10-06 17:56:03
         gNameForAJob1

Note: these counters are derived from Hadoop counters and represent approximate
network bandwidth utilization

switch>
```

show monitor hadoop tasktracker jobs

The **show monitor hadoop tasktracker jobs** command displays data about the jobs that are running on TaskTrackers located on the specified node or accessed through the listed interfaces.

Including a cluster parameter filters results to include data only from the cluster polled by the specified monitor.

Command Mode

EXEC

Command Syntax

show monitor hadoop tasktracker *NODES* jobs [*CLUSTERS*]

Parameters

- ***NODES*** TaskTracker node access point. Options include:
  - **host** *hostname* Node name.
  - **interface ethernet** *e_range* Ethernet interfaces through which node connects.
  - **interface port-channel** *p_range* Port channel interfaces through which node connects.
- ***CLUSTERS*** Hadoop cluster for which command displays data. Options include:
  - <no parameter> TaskTracker on specified ***NODE*** can be in any cluster.
  - **cluster** *c_name* TaskCluster on specified ***NODE*** must be in named cluster.

Examples

- This command displays data for jobs running on ***TaskTracker1***.

```
switch>show monitor hadoop tasktracker host TaskTracker1 jobs
Last updated: 2013-10-06 18:14:23
Running job for TaskTracker: TaskTracker1
JobId  Job Name         Cluster   Maps(#/%)   Reduces(#/%)  Start Time
------ ---------------- --------- ---------- -------------- -------------------
1      ReallyAVeryLon\  Cluster0  2/12.34%    0/13.45%      2013-10-06 17:56:03
       gNameForAJob1
2      ShortName2       Cluster0  2/24.68%    0/26.90%      2013-10-06 17:37:43

switch>
```

- This command displays data for jobs on TaskTrackers accessed through Ethernet interfaces 7 and 8.

```
switch>show monitor hadoop tasktracker interface Ethernet 7,8 jobs
Last updated: 2013-10-06 18:14:23
Running job for TaskTracker: TaskTracker3
JobId   Job Name         Cluster   Maps(#/%)  Reduces(#/%)  Start Time
------- ---------------- --------- ---------- ------------- -------------------
510001  ReallyAVeryLon\  Cluster1  2/12.34%   0/13.45%      2013-10-06 17:56:03
        gNameForAJob11
510002  ShortName12      Cluster1  2/24.68%   0/26.90%      2013-10-06 17:37:43


Running job for TaskTracker: TaskTracker1
JobId  Job Name         Cluster   Maps(#/%)   Reduces(#/%)  Start Time
------ ---------------- --------- ---------- ------------- -------------------
1      ReallyAVeryLon\  Cluster0  2/12.34%    0/13.45%      2013-10-06 17:56:03
       gNameForAJob1
2      ShortName2       Cluster0  2/24.68%    0/26.90%      2013-10-06 17:37:43


switch>
```

- This command displays data for jobs on TaskTrackers accessed through port channel interface 7.

```
switch>show monitor hadoop tasktracker interface Port-Channel 7 jobs
Last updated: 2013-10-06 18:14:23
Running job for TaskTracker: TaskTracker2
JobId  Job Name         Cluster   Maps(#/%)   Reduces(#/%)  Start Time
------ ---------------- --------- ---------- ------------- -------------------
1      ReallyAVeryLon\  Cluster0  2/12.34%    0/13.45%      2013-10-06 17:56:03
       gNameForAJob1
2      ShortName2       Cluster0  2/24.68%    0/26.90%      2013-10-06 17:37:43


switch>
```

- This command displays data for jobs on TaskTrackers on the Cluster0 cluster that are accessed through Ethernet interfaces 7 and 8.

```
switch>show monitor hadoop tasktracker interface Ethernet 7,8 jobs cluster
Cluster0
Last updated: 2013-10-06 18:14:23
Running job for TaskTracker: TaskTracker1
JobId  Job Name         Cluster   Maps(#/%)   Reduces(#/%)  Start Time
------ ---------------- --------- ---------- ------------- -------------------
1      ReallyAVeryLon\  Cluster0  2/12.34%    0/13.45%      2013-10-06 17:56:03
       gNameForAJob1
2      ShortName2       Cluster0  2/24.68%    0/26.90%      2013-10-06 17:37:43


switch>
```

- This command displays data for jobs on TaskTracker named TaskTracker1 on the Cluster0 cluster.

```
switch>show monitor hadoop tasktracker host TaskTracker1 jobs cluster Cluster0
Last updated: 2013-10-06 18:14:23
Running job for TaskTracker: TaskTracker1
JobId  Job Name         Cluster   Maps(#/%)   Reduces(#/%)  Start Time
------ ---------------- --------- ---------- -------------- -------------------
1      ReallyAVeryLon\  Cluster0  2/12.34%    0/13.45%      2013-10-06 17:56:03
       gNameForAJob1
2      ShortName2       Cluster0  2/24.68%    0/26.90%      2013-10-06 17:37:43

switch>
```

- This command displays data for jobs on TaskTrackers on the Cluster0 cluster that are accessed through port channel interface 7.

```
switch>show monitor hadoop tasktracker interface Port-Channel 7 jobs cluster
Cluster0
Last updated: 2013-10-06 18:14:23
Running job for TaskTracker: TaskTracker2
JobId  Job Name         Cluster   Maps(#/%)   Reduces(#/%)  Start Time
------ ---------------- --------- ---------- -------------- -------------------
1      ReallyAVeryLon\  Cluster0  2/12.34%    0/13.45%      2013-10-06 17:56:03
       gNameForAJob1
2      ShortName2       Cluster0  2/24.68%    0/26.90%      2013-10-06 17:37:43

switch>
```

show monitor hadoop tasktracker running-tasks

The **show monitor hadoop tasktracker running-tasks** command displays progress and byte counts of tasks executed by TaskTrackers located on the specified node or accessed through the listed interfaces.

Including a cluster-ID parameter filters results to include data only from the specified cluster.

Command Mode

EXEC

Command Syntax

show monitor hadoop tasktracker *NODES* running-tasks [*CLUSTERS*] [*JOBS*]

Parameters

- ***NODES*** TaskTracker node access point. Options include:
  - **host** *hostname* Node name.
  - **interface ethernet** *e_range* Ethernet interfaces through which node connects.
  - **interface port-channel** *p_range* Port channel interfaces through which node connects.
- ***CLUSTERS*** Hadoop cluster for which command displays data. Options include:
  - <no parameter> TaskTracker on specified ***NODE*** can be in any cluster.
  - **cluster** *c_name* TaskCluster on specified ***NODE*** must be in named cluster.
- ***JOBS*** Job list. Options include:
  - <no parameter> all jobs.
  - **job** <***0*** to ***2147483647***> Specifies number of single job.

Examples

- This command displays data for tasks running on TaskTracker named ***TaskTracker1***.

```
switch>show monitor hadoop tasktracker host TaskTracker1 running-tasks
Last updated: 2013-10-06 18:14:23
Running tasks for TaskTracker: TaskTracker1 on interface Ethernet7
JobId TaskId  Cluster   Type  Progress  Status   HDFS Read  HDFS Write  Shuffle
----- ------- --------- ----- --------- -------- ---------- ----------- -------
1     2       Cluster0  Map   33.33%    running  2.10MB     2.14MB      2.96MB
2     2       Cluster0  Map   33.33%    running  2.10MB     2.14MB      2.96MB
1     1       Cluster0  Map   50.00%    running  1.05MB     1.07MB      1.48MB
2     1       Cluster0  Map   50.00%    running  1.05MB     1.07MB      1.48MB

switch>
```

- This command displays data for tasks running on the TaskTracker named ***TaskTracker1*** of the ***Cluster0*** cluster.

```
switch>show monitor hadoop tasktracker host TaskTracker1 running-tasks cluster
Cluster0
Last updated: 2013-10-06 18:14:23
Running tasks for TaskTracker: TaskTracker1 on interface Ethernet7
JobId TaskId  Cluster   Type  Progress  Status   HDFS Read  HDFS Write  Shuffle
----- ------- --------- ----- --------- -------- ---------- ----------- -------
1     2       Cluster0  Map   33.33%    running  2.10MB     2.14MB      2.96MB
2     2       Cluster0  Map   33.33%    running  2.10MB     2.14MB      2.96MB
1     1       Cluster0  Map   50.00%    running  1.05MB     1.07MB      1.48MB
2     1       Cluster0  Map   50.00%    running  1.05MB     1.07MB      1.48MB

switch>
```

- This command displays data for tasks running for job ***1*** on the TaskTracker named ***TaskTracker1*** of the ***Cluster0*** cluster.

```
switch>show monitor hadoop tasktracker host TaskTracker1 running-tasks cluster
Cluster0 job 1
Last updated: 2013-10-06 18:14:23
Running tasks for TaskTracker: TaskTracker1 on interface Ethernet7
JobId TaskId  Cluster   Type  Progress  Status   HDFS Read  HDFS Write  Shuffle
----- ------- --------- ----- --------- -------- ---------- ----------- -------
1     2       Cluster0  Map   33.33%    running  2.10MB     2.14MB      2.96MB
1     1       Cluster0  Map   50.00%    running  1.05MB     1.07MB      1.48MB

switch>
```

- This command displays data for tasks running on TaskTrackers accessed through Ethernet interfaces 7 and 8.

```
switch>show monitor hadoop tasktracker interface Ethernet 7,8 running-tasks
Last updated: 2013-10-06 18:14:23
Running tasks for TaskTracker: TaskTracker1 on interface Ethernet7
JobId TaskId  Cluster   Type  Progress  Status   HDFS Read  HDFS Write  Shuffle
----- ------- --------- ----- --------- -------- ---------- ----------- -------
2     2       Cluster0  Map   33.33%    running  2.10MB     2.14MB      2.96MB
1     1       Cluster0  Map   50.00%    running  1.05MB     1.07MB      1.48MB
2     1       Cluster0  Map   50.00%    running  1.05MB     1.07MB      1.48MB

Running tasks for TaskTracker: TaskTracker3 on interface Ethernet8
JobId  TaskId Cluster   Type  Progress  Status   HDFS Read  HDFS Write  Shuffle
------ ------ --------- ----- --------- -------- ---------- ----------- -------
510002 222    Cluster1  Map   33.33%    running  2.10MB     2.14MB      2.96MB
510001 112    Cluster1  Map   33.33%    running  2.10MB     2.14MB      2.96MB
510002 221    Cluster1  Map   50.00%    running  1.05MB     1.07MB      1.48MB
510001 111    Cluster1  Map   50.00%    running  1.05MB     1.07MB      1.48MB

switch>
```

- This command displays data for tasks running on TaskTrackers of ***Cluster0*** cluster that are accessed through Ethernet interfaces 7 and 8.

```
switch>show monitor hadoop tasktracker interface Ethernet 7,8 running-tasks
cluster Cluster0
Last updated: 2013-10-06 18:14:23
Running tasks for TaskTracker: TaskTracker1 on interface Ethernet7
JobId TaskId  Cluster   Type  Progress  Status   HDFS Read  HDFS Write  Shuffle
----- ------- --------- ----- --------- -------- ---------- ----------- -------
1     2       Cluster0  Map   33.33%    running  2.10MB     2.14MB      2.96MB
2     2       Cluster0  Map   33.33%    running  2.10MB     2.14MB      2.96MB
1     1       Cluster0  Map   50.00%    running  1.05MB     1.07MB      1.48MB
2     1       Cluster0  Map   50.00%    running  1.05MB     1.07MB      1.48MB

switch>
```

- This command displays data for tasks running for job 1 on the TaskTrackers of ***Cluster0*** cluster that are accessed through Ethernet interfaces 7 and 8.

```
switch>show monitor hadoop tasktracker interface Ethernet 7,8 running-tasks
cluster Cluster0 job 1
Last updated: 2013-10-06 18:14:23
Running tasks for TaskTracker: TaskTracker1 on interface Ethernet7
JobId TaskId  Cluster   Type  Progress  Status   HDFS Read  HDFS Write  Shuffle
----- ------- --------- ----- --------- -------- ---------- ----------- -------
1     2       Cluster0  Map   33.33%    running  2.10MB     2.14MB      2.96MB
1     1       Cluster0  Map   50.00%    running  1.05MB     1.07MB      1.48MB

switch>
```

- This command displays data for tasks running on TaskTrackers accessed through port channel interfaces 7 and 8.

```
switch>show monitor hadoop tasktracker interface Port-Channel 7-8 running-tasks
Last updated: 2013-10-06 18:14:23
Running tasks for TaskTracker: TaskTracker2 on interface Port-Channel7
JobId TaskId  Cluster   Type  Progress  Status   HDFS Read  HDFS Write  Shuffle
----- ------- --------- ----- --------- -------- ---------- ----------- -------
1     2       Cluster0  Map   33.33%    running  2.10MB     2.14MB      2.96MB
1     1       Cluster0  Map   50.00%    running  1.05MB     1.07MB      1.48MB
2     1       Cluster0  Map   50.00%    running  1.05MB     1.07MB      1.48MB

Running tasks for TaskTracker: TaskTracker4 on interface Port-Channel8
JobId  TaskId Cluster   Type  Progress  Status   HDFS Read  HDFS Write  Shuffle
------ ------ --------- ----- --------- -------- ---------- ----------- -------
510002 222    Cluster1  Map   33.33%    running  2.10MB     2.14MB      2.96MB
510001 112    Cluster1  Map   33.33%    running  2.10MB     2.14MB      2.96MB
510001 111    Cluster1  Map   50.00%    running  1.05MB     1.07MB      1.48MB

switch>
```

- This command displays data for tasks running on TaskTrackers of ***Cluster0*** cluster accessed through port channel interface 7.

```
switch>show monitor hadoop tasktracker interface Port-Channel 7 running-tasks
cluster Cluster0
Last updated: 2013-10-06 18:14:23
Running tasks for TaskTracker: TaskTracker2 on interface Port-Channel7
JobId TaskId  Cluster   Type  Progress  Status   HDFS Read  HDFS Write  Shuffle
----- ------- --------- ----- --------- -------- ---------- ----------- -------
1     2       Cluster0  Map   33.33%    running  2.10MB     2.14MB      2.96MB
1     1       Cluster0  Map   50.00%    running  1.05MB     1.07MB      1.48MB
2     1       Cluster0  Map   50.00%    running  1.05MB     1.07MB      1.48MB

switch>
```

- This command displays data for job 510001 running on TaskTrackers of ***Cluster1*** cluster that are accessed through port channel interface 8.

```
switch>show monitor hadoop tasktracker interface Port-Channel 8 running-tasks
cluster Cluster1 job 510001
Last updated: 2013-10-06 18:14:23
Running tasks for TaskTracker: TaskTracker4 on interface Port-Channel8
JobId  TaskId Cluster   Type  Progress  Status   HDFS Read  HDFS Write  Shuffle
------ ------ --------- ----- --------- -------- ---------- ----------- -------
510001 112    Cluster1  Map   33.33%    running  2.10MB     2.14MB      2.96MB
510001 111    Cluster1  Map   50.00%    running  1.05MB     1.07MB      1.48MB

switch>
```

show monitor hadoop tasktracker running-tasks cluster job task

The **show monitor hadoop tasktracker running-tasks cluster job task** command displays detailed data for the specified task.

Command Mode

EXEC

Command Syntax

show monitor hadoop tasktracker *NODE* running-tasks cluster *name* **job** *jnum* **task** *tnum*

Parameters

- ***NODE*** TaskTracker node access point. Options include:
  - **host** *hostname* Node name.
  - **interface ethernet** *e_range* Ethernet interfaces through which node connects.
  - **interface port-channel** *p_range* Port channel interfaces through which node connects.
- *name* Cluster name.
- *jnum* Job number. Value ranges from ***0*** to ***2147483647***
- *tnum* Task number. Value ranges from ***0*** to ***2147483647***

Examples

- This command displays data for task 1 of job 1 on ***TaskTracker1*** of ***Cluster0***.

```
switch>show monitor hadoop tasktracker host TaskTracker1 running-tasks cluster
Cluster0 job 1 task 1
Last updated: 2013-10-06 18:14:23
Task details for one task as given below:
  TaskTracker name         : TaskTracker1
  Interface                : 'Ethernet7'
  Cluster                  : Cluster0
  Job Id                   : 1
  Task Id                  : 1
  Attempt Id               : 0
  Task type                : Map
  Status                   : running
  State                    : running
  Start time               : 2013-10-06 17:57:43
  Progress                 : 50.00%
  HDFS bytes read          : 1.05MB
  HDFS bytes written       : 1.07MB
  Reduce shuffle bytes     : 1.48MB

switch>
```

- This command displays data for task 1 of job 1 on the ***Cluster0*** TaskTracker that is accessible through Ethernet interface 7.

```
switch>show monitor hadoop tasktracker interface Ethernet 7 running-tasks cluster
Cluster0 job 1 task 1
Last updated: 2013-10-06 18:14:23
Task details for one task as given below:
  TaskTracker name         : TaskTracker1
  Interface                : 'Ethernet7'
  Cluster                  : Cluster0
  Job Id                   : 1
  Task Id                  : 1
  Attempt Id               : 0
  Task type                : Map
  Status                   : running
  State                    : running
  Start time               : 2013-10-06 17:57:43
  Progress                 : 50.00%
  HDFS bytes read          : 1.05MB
  HDFS bytes written       : 1.07MB
  Reduce shuffle bytes     : 1.48MB

switch>
```

- This command displays data for task 111 of job 510001 on the ***Cluster0*** TaskTracker that is accessible through port channel interface 8.

```
switch>show monitor hadoop tasktracker interface Port-Channel 8 running-tasks
cluster Cluster1 job 510001 task 111
Last updated: 2013-10-06 18:14:23
Task details for one task as given below:
  TaskTracker name         : TaskTracker4
  Interface                : 'Port-Channel8'
  Cluster                  : Cluster1
  Job Id                   : 510001
  Task Id                  : 111
  Attempt Id               : 0
  Task type                : Map
  Status                   : running
  State                    : running
  Start time               : 2013-10-06 17:57:43
  Progress                 : 50.00%
  HDFS bytes read          : 1.05MB
  HDFS bytes written       : 1.07MB
  Reduce shuffle bytes     : 1.48MB

switch>
```

show monitor hadoop tasktracker status

The **show monitor hadoop tasktracker status** command displays connection and activity information for the TaskTracker on the specified clusters or accessed through the specified interface. The following command formats display the listed TaskTracker information:

- **show monitor hadoop cluster** *c_name* **tasktracker status**: TaskTrackers on specified cluster.
- **show monitor hadoop tasktracker** *node* **status**: TaskTrackers on specified nodes or interfaces.
- **show monitor hadoop tasktracker all status**: all connected TaskTrackers.

Command Mode

EXEC

Command Syntax

```
show monitor hadoop cluster c_name tasktracker status
show monitor hadoop tasktracker NODES status
show monitor hadoop tasktracker all status
```

Parameters

- *c_name* Cluster name.
- ***NODES*** TaskTracker node access point. Options include:
  - **host** *hostname* Node name.
  - **interface ethernet** *e_range* Ethernet interfaces through which node connects.
  - **interface port-channel** *p_range* Port channel interfaces through which node connects.

Examples

- This command displays connection and activity information for all TaskTrackers connected through Ethernet interfaces 7 and 8.

```
switch>show monitor hadoop tasktracker interface Ethernet7,8 status
Last updated: 2013-10-06 18:14:23
  TaskTracker              : TaskTracker1
  IP Address               : 10.100.0.1
  Interface                : Ethernet7
  State                    : active
  Running jobs             : 2
  Running tasks            : 4
  Map Tasks                : 4
  Reduce Tasks             : 0
  Total bytes read         : 2.08GB
  Total bytes written      : 4.24MB

  TaskTracker              : TaskTracker3
  IP Address               : 10.100.0.3
  Interface                : Ethernet8
  State                    : active
  Running jobs             : 2
  Running tasks            : 4
  Map Tasks                : 4
  Reduce Tasks             : 0
  Total bytes read         : 6.23GB
  Total bytes written      : 12.72MB

switch>
```

- This command displays connection and activity information for all connected TaskTrackers.

```
switch>show monitor hadoop tasktracker all status
Last updated: 2013-10-06 18:14:23
All local TaskTrackers:
  TaskTracker              : TaskTracker4
  IP Address               : 10.100.0.4
  Interface                : Port-Channel8
  State                    : active
  Running jobs             : 2
  Running tasks            : 4
  Map Tasks                : 4
  Reduce Tasks             : 0
  Total bytes read         : 8.30GB
  Total bytes written      : 16.95MB

  TaskTracker              : TaskTracker3
  IP Address               : 10.100.0.3
  Interface                : Ethernet8
  State                    : active
  Running jobs             : 2
  Running tasks            : 4
  Map Tasks                : 4
  Reduce Tasks             : 0
  Total bytes read         : 6.23GB
  Total bytes written      : 12.72MB

  TaskTracker              : TaskTracker2
  IP Address               : 10.100.0.2
  Interface                : Port-Channel7
  State                    : active
  Running jobs             : 2
  Running tasks            : 4
  Map Tasks                : 4
  Reduce Tasks             : 0
  Total bytes read         : 4.15GB
  Total bytes written      : 8.48MB

  TaskTracker              : TaskTracker1
  IP Address               : 10.100.0.1
  Interface                : Ethernet7
  State                    : active
  Running jobs             : 2
  Running tasks            : 4
  Map Tasks                : 4
  Reduce Tasks             : 0
  Total bytes read         : 2.08GB
  Total bytes written      : 4.24MB

switch>
```

- This command displays connection and activity data for TaskTracker on the ***TaskTracker1*** node.

```
switch>show monitor hadoop tasktracker host TaskTracker1 status
Last updated: 2013-10-06 18:14:23
  TaskTracker              : TaskTracker1
  IP Address               : 10.100.0.1
  Interface                : Ethernet7
  State                    : active
  Running jobs             : 2
  Running tasks            : 4
  Map Tasks                : 4
  Reduce Tasks             : 0
  Total bytes read         : 2.08GB
  Total bytes written      : 4.24MB

switch>
```

- This command displays connection and activity data for all TaskTracker connected through Port Channel 7.

```
switch>show monitor hadoop tasktracker interface Port-Channel 7 status
Last updated: 2013-10-06 18:14:23
  TaskTracker              : TaskTracker2
  IP Address               : 10.100.0.2
  Interface                : Port-Channel7
  State                    : active
  Running jobs             : 2
  Running tasks            : 4
  Map Tasks                : 4
  Reduce Tasks             : 0
  Total bytes read         : 4.15GB
  Total bytes written      : 8.48MB

switch>
```

- This command displays connection and activity data for all TaskTrackers on the ***Cluster0*** cluster.

```
switch>show monitor hadoop cluster Cluster0 tasktracker status
Last updated: 2013-10-06 18:14:23
Total 2 TaskTrackers on cluster Cluster0:
  TaskTracker              : TaskTracker2
  IP Address               : 10.100.0.2
  Interface                : Port-Channel7
  State                    : active
  Running jobs             : 2
  Running tasks            : 4
  Map Tasks                : 4
  Reduce Tasks             : 0
  Total bytes read         : 4.15GB
  Total bytes written      : 8.48MB

  TaskTracker              : TaskTracker1
  IP Address               : 10.100.0.1
  Interface                : Ethernet7
  State                    : active
  Running jobs             : 2
  Running tasks            : 4
  Map Tasks                : 4
  Reduce Tasks             : 0
  Total bytes read         : 2.08GB
  Total bytes written      : 4.24MB

switch>
```

show monitor hadoop traffic burst

The **show monitor hadoop traffic burst** command displays the largest data bursts for specified Hadoop cluster jobs. A data burst is the data consumed during a polling interval. The command displays input and output burst:

- Input bursts include bytes written to the host.
- Output bursts include bytes written by the host.

Command Mode

EXEC

Command Syntax

```
show monitor hadoop [CLUSTERS] traffic burst [NODE]
```

Parameters

- ***CLUSTERS*** Hadoop clusters for which command displays data. Options include:
  - <no parameter> all clusters.
  - **cluster** *c_name* Cluster name.
- ***NODES*** TaskTracker node access point. Options include:
  - **host** *hostname* Node name.
  - **interface ethernet** *e_range* Ethernet interfaces through which node connects.
  - **interface port-channel** *p_range* Port channel interfaces through which node connects.

Examples

- This command displays traffic burst data for all running jobs.

```
switch>show monitor hadoop traffic burst
Last updated: 2013-10-06 18:14:23
Bursts on Interface: 'Ethernet7' in cluster: Cluster0
Top 2 input bursts:
   JobId       Job Name              Burst     Time
----------- -------------------- ------------ -------------------
   1           ShortName             3.07GB    2013-10-06 17:57:43
   2           ReallyAVeryLon\       6.15GB    2013-10-06 17:41:03
               gNameForAJob

Top 2 output bursts:
   JobId       Job Name              Burst     Time
----------- -------------------- ------------ -------------------
   1           ShortName             4.10GB    2013-10-06 17:55:13
   2           ReallyAVeryLon\       8.20GB    2013-10-06 17:36:03
               gNameForAJob

Bursts on Interface: 'Port-Channel7' in cluster: Cluster0
Top 2 input bursts:
   JobId       Job Name              Burst     Time
----------- -------------------- ------------ -------------------
   1           ShortName             3.07GB    2013-10-06 17:57:43
   2           ReallyAVeryLon\       6.15GB    2013-10-06 17:41:03
               gNameForAJob

Top 2 output bursts:
   JobId       Job Name              Burst     Time
----------- -------------------- ------------ -------------------
   1           ShortName             4.10GB    2013-10-06 17:55:13
   2           ReallyAVeryLon\       8.20GB    2013-10-06 17:36:03
               gNameForAJob
Bursts on Interface: 'Ethernet8' in cluster: Cluster1
Top 4 input bursts:
   JobId        Job Name              Burst      Time
------------ -------------------- ------------- -------------------
   510001       ShortName             3.07GB     2013-10-06 17:57:43
   510002       ReallyAVeryLon\       6.15GB     2013-10-06 17:41:03
                gNameForAJob
   510003       ShortName             9.22GB     2013-10-06 17:24:23
   510004       ReallyAVeryLon\       12.29GB    2013-10-06 17:07:43
                gNameForAJob

Top 4 output bursts:
   JobId        Job Name              Burst      Time
------------ -------------------- ------------- -------------------
   510001       ShortName             4.10GB     2013-10-06 17:55:13
   510002       ReallyAVeryLon\       8.20GB     2013-10-06 17:36:03
                gNameForAJob
   510003       ShortName             12.29GB    2013-10-06 17:16:53
   510004       ReallyAVeryLon\       16.39GB    2013-10-06 16:57:43
                gNameForAJob

Bursts on Interface: 'Port-Channel8' in cluster: Cluster1
Top 4 input bursts:
   JobId        Job Name              Burst      Time
------------ -------------------- ------------- -------------------
```

```
   510001       ShortName             3.07GB     2013-10-06 17:57:43
   510002       ReallyAVeryLon\       6.15GB     2013-10-06 17:41:03
                gNameForAJob
   510003       ShortName             9.22GB     2013-10-06 17:24:23
   510004       ReallyAVeryLon\       12.29GB    2013-10-06 17:07:43
                gNameForAJob

Top 4 output bursts:
   JobId        Job Name              Burst      Time
------------ -------------------- ------------- -------------------
   510001       ShortName             4.10GB     2013-10-06 17:55:13
   510002       ReallyAVeryLon\       8.20GB     2013-10-06 17:36:03
                gNameForAJob
   510003       ShortName             12.29GB    2013-10-06 17:16:53
   510004       ReallyAVeryLon\       16.39GB    2013-10-06 16:57:43
                gNameForAJob

switch>
```

- This command displays traffic burst for all jobs running on TaskTrackers that are accessible through Ethernet interfaces 7 and 8.

```
switch>show monitor hadoop traffic burst interface Ethernet 7,8
Last updated: 2013-10-06 18:14:23
Bursts on Interface: 'Ethernet7' in cluster: Cluster0
Top 2 input bursts:
   JobId       Job Name              Burst     Time
----------- -------------------- ------------ -------------------
   1           ShortName             3.07GB    2013-10-06 17:57:43
   2           ReallyAVeryLon\       6.15GB    2013-10-06 17:41:03
               gNameForAJob
Top 2 output bursts:
   JobId       Job Name              Burst     Time
----------- -------------------- ------------ -------------------
   1           ShortName             4.10GB    2013-10-06 17:55:13
   2           ReallyAVeryLon\       8.20GB    2013-10-06 17:36:03
               gNameForAJob

Bursts on Interface: 'Ethernet8' in cluster: Cluster1
Top 4 input bursts:
   JobId        Job Name              Burst      Time
------------ -------------------- ------------- -------------------
   510001       ShortName             3.07GB     2013-10-06 17:57:43
   510002       ReallyAVeryLon\       6.15GB     2013-10-06 17:41:03
                gNameForAJob
   510003       ShortName             9.22GB     2013-10-06 17:24:23
   510004       ReallyAVeryLon\       12.29GB    2013-10-06 17:07:43
                gNameForAJob

Top 4 output bursts:
   JobId        Job Name              Burst      Time
------------ -------------------- ------------- -------------------
   510001       ShortName             4.10GB     2013-10-06 17:55:13
   510002       ReallyAVeryLon\       8.20GB     2013-10-06 17:36:03
                gNameForAJob
   510003       ShortName             12.29GB    2013-10-06 17:16:53
   510004       ReallyAVeryLon\       16.39GB    2013-10-06 16:57:43
                gNameForAJob

switch>
```

- This command displays traffic burst data for all running jobs that are accessible through port channel interface 7.

```
switch>show monitor hadoop traffic burst interface Port-Channel 7
Last updated: 2013-10-06 18:14:23
Bursts on Interface: 'Port-Channel7' in cluster: Cluster0
Top 2 input bursts:
   JobId       Job Name              Burst     Time
----------- -------------------- ------------ -------------------
   1           ShortName             3.07GB    2013-10-06 17:57:43
   2           ReallyAVeryLon\       6.15GB    2013-10-06 17:41:03
               gNameForAJob

Top 2 output bursts:
   JobId       Job Name              Burst     Time
----------- -------------------- ------------ -------------------
   1           ShortName             4.10GB    2013-10-06 17:55:13
   2           ReallyAVeryLon\       8.20GB    2013-10-06 17:36:03
               gNameForAJob
```

shutdown (Monitor-Hadoop)

The **shutdown** command globally disables MapReduce Tracer on the switch. Enabling MapReduce Tracer for an individual cluster requires the feature to be globally enabled through this command and enabled for the individual cluster thorough the **shutdown (Monitor Hadoop Cluster)** command. By default, MapReduce Tracer is globally disabled.

The **no shutdown** command globally enables MapReduce Tracer. The **shutdown** and **default shutdown** commands globally disable MapReduce Tracer by removing the corresponding **no shutdown** command from ***running-config***.

Command Mode

Monitor-hadoop Configuration

Command Syntax

```
shutdown
no shutdown
default shutdown
```

Related Commands

- **monitor hadoop** places the switch in monitor-hadoop configuration mode.

Example

- These commands globally enable MapReduce Tracer.

```
switch(config)#monitor hadoop
switch(config-monitor-hadoop)#no shutdown
switch(config-monitor-hadoop)#show active
 monitor hadoop
   no shutdown
switch(config-monitor-hadoop)#
```

- This command globally disables MapReduce Tracer.

```
switch(config-monitor-hadoop)#shutdown
switch(config-monitor-hadoop)#show active
switch(config-monitor-hadoop)#
```

shutdown (Monitor Hadoop Cluster)

The **shutdown** command disables MapReduce Tracer for the configuration mode cluster. Globally disabling MapReduce Tracer (**shutdown (Monitor-Hadoop)**) also disables the function on the individual cluster. Enabling MapReduce Tracer for the cluster requires the function to be enabled globally and for the individual cluster.

The **no shutdown** command configures the MapReduce Tracer setting as ***enabled*** for the configuration mode cluster. The **shutdown** and **default shutdown** commands disable MapReduce Tracer for the cluster by removing the corresponding **no shutdown** command from ***running-config***.

Command Mode

Monitor-hadoop-cluster Configuration

Command Syntax

```
shutdown
no shutdown
default shutdown
```

Related Commands

- **cluster (Monitor Hadoop)** places the switch in monitor-hadoop-cluster configuration mode.

Example

- These commands globally enable MapReduce Tracer, then enables it for the CL2 cluster.

```
switch(config)#monitor hadoop
switch(config-monitor-hadoop)#no shutdown
switch(config-monitor-hadoop)#cluster CL2
switch(config-monitor-hadoop-CL2)#no shutdown
switch(config-monitor-hadoop-CL2)#show active
monitor hadoop
   cluster CL2
      no shutdown
switch(config-monitor-hadoop-CL2)#exit
switch(config-monitor-hadoop)#show active
 monitor hadoop
   no shutdown
   cluster CL2
      no shutdown
switch(config-monitor-hadoop)#
```

- These commands disable MapReduce Tracer for the CL2 cluster. MapReduce Tracer remains globally enabled.

```
switch(config-monitor-hadoop)#cluster CL2
switch(config-monitor-hadoop-CL2)#shutdown
switch(config-monitor-hadoop-CL2)#show active
monitor hadoop
   cluster CL2
switch(config-monitor-hadoop-CL2)#exit
switch(config-monitor-hadoop)#show active
 monitor hadoop
   no shutdown
   cluster CL2
switch(config-monitor-hadoop)#
```

tasktracker (Monitor Hadoop Cluster)

The **tasktracker** command specifies the HTTP port for accessing TaskTrackers of the Hadoop cluster monitored through configuration mode statements. The switch compiles a list of the cluster's TaskTracker addresses by periodically polling the cluster's JobTracker (**jobtracker (Monitor Hadoop Cluster)**). The default TaskTracker HTTP port is 50060.

The **no tasktracker** and **default tasktracker** commands restore the configuration mode TaskTracker HTTP port to 50060 by removing the corresponding **tasktracker** command from ***running-config***.

Command Mode

Monitor-hadoop-cluster Configuration

Command Syntax

```
tasktracker http-port port_number
no tasktracker http-port
default tasktracker http-port
```

Parameters

- *port_num* TaskTracker HTTP port number. Value ranges from 1 to 65535. Default value is 50060.

Related Commands

- **cluster (Monitor Hadoop)** places the switch in monitor-hadoop-cluster configuration mode.

Example

- These commands specify a TaskTracker HTTP port address of 51000.

```
switch(config)#monitor hadoop
switch(config-monitor-hadoop)#cluster CL2
switch(config-monitor-hadoop-CL2)#tasktracker http-port 51000
switch(config-monitor-hadoop-CL2)#show active
monitor hadoop
   cluster CL2
      tasktracker http-port 51000
switch(config-monitor-hadoop-CL2)#
```

- These commands restore the default TaskTracker HTTP port address of 50060.

```
switch(config-monitor-hadoop-CL2)#no tasktracker http-port
switch(config-monitor-hadoop-CL2)#show active
monitor hadoop
   cluster CL2
switch(config-monitor-hadoop-CL2)#show active all
monitor hadoop
   cluster CL2
      jobtracker rpc-port 8021
      tasktracker http-port 50060
      interval 10
      shutdown
switch(config-monitor-hadoop-CL2)#
```

Chapter 44

sFlow

This chapter describes Arista's implementation of sFlow, including configuration instructions and command descriptions. Topics covered by this chapter include:

- Section 44.1: sFlow Conceptual Overview
- Section 44.2: sFlow Configuration Procedures
- Section 44.3: sFlow Configuration Commands

44.1 sFlow Conceptual Overview

44.1.1 sFlow Technology

sFlow is a multi-vendor sampling technology that continuously monitors application level traffic flow at wire speed simultaneously on all interfaces. sFlow provides gigabit speed quantitative traffic measurements without impacting network performance.

sFlow has the following network traffic monitoring characteristics:

- sFlow provides a network view of active route usage that measures network traffic.
- sFlow is scalable to 10 Gb/s without impacting switch performance or the network load.
- sFlow is implemented on a wide range of devices, without requiring additional memory and CPU.
- sFlow is an industry standard.

An sFlow configuration consists of:

- sFlow agents, embedded on network equipment, that monitors traffic and generates data.
- sFlow collectors that receive and analyze sFlow data.

Arista switches include an sFlow agent that monitors ingress data through all Ethernet interfaces.

44.1.1.1 sFlow Agents

The sFlow agent is a software process that runs as part of the network management software within an Arista switch. It combines interface counters and flow samples into sFlow datagrams that are sent to an sFlow collector. Packets typically include flow samples and state information of the forwarding/routing table entries associated with each sample.

The sFlow Agent performs minimal processing when packaging data into datagrams. Immediate data forwarding minimizes agent memory and CPU requirements.

44.1.1.2 sFlow Collector

An sFlow collector is a server that runs software that analyzes and reports network traffic. Collectors receive flow samples and counter samples respectively as sFlow datagrams from sFlow agents. Arista switches reference a collector's IP address and UDP port as a configurable setting through a CLI command. Arista switches do not include sFlow collector software.

44.1.1.3 sFlow Data

The sFlow Agent uses two forms of sampling: statistical packet-based sampling of switched flows and time-based sampling of network interface statistics.

- **Switched flow sampling**: A sample is taken by either copying the packet's header or extracting feature data from the packet.
- **Interface statistics sampling**: Counter sampling extracts statistics by periodically polling each data source on the device.

sFlow implements flow sampling and counter sampling as part of an integrated system. An sFlow datagram incorporates both sample types.

44.1.2 Arista sFlow Implementation

Arista switches provide a single sFlow agent instance that samples ingress traffic from all Ethernet and port channel interfaces. The switch provides two levels of settings for enabling sFlow:

- a global setting that enables packet sampling on the entire switch.
- interface settings that control sampling on individual interfaces when sFlow is globally enabled.

sFlow default settings include:

- global: sFlow is globally disabled.
- Ethernet and port channel interfaces: sFlow is enabled on all interfaces when it is globally ***enabled***.

The switch performs sFlow polling when sFlow is globally enabled. The CLI provides commands that globally disable sampling while counter polling remains enabled. Sample enabling is not controllable on individual interfaces.

The switch sends sFlow datagrams to the collector located at an IP address specified by a global configuration command. If the collector destination is not configured, the switch samples data

without transmitting the resulting datagrams.

Although the CLI enforces the configured sampling rate limit, it may drop samples if it cannot handle the number of samples it receives over a specified period. Under normal operation, the maximum packet sample rate is one per 16384 packets. The CLI allows for higher sampling rates by using the ***dangerous*** keyword.

The following lists describe sFlow's sampling behavior relative to different packet types:

- Packets that are sampled:
  - CPU
  - IP Options and MTU violations
  - Flooded packets
  - Multicast packets
- Packets that are not sampled:
  - LACP frames

- LLDP frames
- STP BPDUs
- IGMP packets
- PAUSE frames
- PIM hello packets
- CRC error frames
- Packets dropped by ACLs or due to VLAN violations

44.1.3 Petra Platform sFlow Implementation

sFlow implementation on Petra platform switches differ from sFlow implementation on other platforms as follows:

- Petra platform ports configured for mirroring cannot support sFlow.

  Ports configured for both sFlow and mirroring ignore sFlow and continue mirroring operations; sFlow configuration commands remain in place and take effect when mirroring is disabled on the port.

- sFlow sampling rate may be affected by congestion of the switch on Petra which can be mitigated by changing the buffer configuration.

  On Petra platforms, ingress buffers are divided into three categories: unicast, multicast and mini-multicast. Ingress packets on interfaces configured for sFlow sampling or ingress mirroring use mini-multicast buffers. If mini-multicast buffers are exhausted, the packets use unicast buffers but are not candidates for sflow sampling or ingress mirroring.

  By default on Petra platforms, EOS allocates 8K mini-multicast buffers and 180K unicast buffers. Up to 64K of the unicast buffers can be reconfigured as mini-multicast buffers, dividing the total buffer pool into 64K mini-multicast buffers and 124K unicast buffers. When implementing sFlow, it is recommended that more buffer space be allocated to mini-multicast buffers using the **platform petraA buffers mini-multicast** command to ensure proper sFlow sampling.

Example

- The following command allocates 64K buffer space to mini-multicast buffers:

```
switch(config)#platform petraA buffers mini-multicast 65536
! Command will cause interfaces to flap (links will go down/up).
Proceed with command? [confirm]y
switch(config)#
```

  The default setting is 8192 (8K). Executing this command disrupts traffic on all switch ports.

44.2 sFlow Configuration Procedures

Implementing sFlow on an Arista switch consists of configuring the following agent parameters:

1. Collector location address.
2. Agent source address.
3. Polling interval.
4. Sampling rate.

(Optionally, sFlow can be configured to include output interface and traffic class information in samples using the **sflow sample** command.)

After configuring the sFlow agent, sampling is initiated by globally enabling sFlow on the switch.

Configuring the collector location

The **sflow destination** command specifies the IP address and UDP port of an sFlow collector. The switch supports multiple collectors.

Example

- This command configures the switch to send sFlow data to collectors at 10.42.15.12, port 6100 and 10.52.12.2 port 6343 (the default sFlow port).

```
switch(config)#sflow destination 10.42.15.12 6100
switch(config)#sflow destination 10.52.12.2
switch(config)#
```

Configuring the agent source address

The **sflow source** command specifies the source address that the switch places in all sFlow datagrams that it sends to the collector. This address is normally set to an IP address configured on the switch.

Example

- This command configures 10.2.9.21 as the sFlow source address.

```
switch(config)#sflow source 10.2.9.21
switch(config)#
```

The **sflow source-interface** command can be alternatively used to specify the interface from which an IP address is derived that the switch places in all sFlow datagrams that it sends to the collector. This address is normally set to an IP address configured on the switch.

Example

- This command configures VLAN interface 25 as the sFlow source interface. The switch enters the IP address for VLAN 25 in the source field of sFlow datagrams.

```
switch(config)#sflow source-interface vlan 25
switch(config)#
```

running-config cannot simultaneously contain **sflow source** and **sflow source-interface** commands.

Configuring the polling interval

The **sflow polling-interval** command specifies the interval for sending counter data to the sFlow collector. The default interval is two seconds.

Example

- This command configures the switch to send sFlow data every ten seconds.

```
switch(config)#sflow polling-interval 10
switch(config)#
```

Configuring the sampling rate and sample contents

The **sflow sample** command sets the packet sampling rate. Packets are sampled at random intervals to avoid inaccurate sampling of periodic events. A rate of 16384 corresponds to an average sample of one per 16,384 packets. The default rate is 1048576.

Example

- This command configures the sFlow sampling rate as 65536 (one per 65,536 packets).

```
switch(config)#sflow sample 65536
switch(config)#
```

The **sflow sample** command can also optionally configure sample packets to include information about the traffic class of the sample. Traffic class is communicated by rewriting the DSCP field in the sample packet.

By default, samples include information about the output interface. To remove this information, use the **[no] sflow sample output interface** command.

- These commands configure sFlow to include traffic class information in samples but to exclude output interface data.

```
switch(config)#no sflow sample output interface
switch(config)#sflow sample rewrite dscp
switch(config)#
```

Enabling sFlow

The **sflow run** command globally enables sFlow on the switch. The **sflow enable** command controls sFlow operation on Ethernet and port channel interfaces when sFlow is globally enabled. The **sflow enable** command has no effect when sFlow is globally disabled.

Example

- These commands enable sFlow on the switch, then disables sFlow on Ethernet interface 10.

```
switch(config)#sflow run
switch(config)#interface ethernet 10
switch(config-if-Et10)#no sflow enable
switch(config)#
```

44.3 sFlow Configuration Commands

This section contains descriptions of sFlow commands.

Global Configuration Commands

- platform petraA buffers mini-multicast
- sflow destination
- sflow polling-interval
- sflow run
- sflow sample
- sflow source
- sflow source-interface

Interface Configuration Commands

- sflow enable

Privileged EXEC Command

- clear sflow counters

sFlow Display Commands

- show sflow
- show sflow interfaces

clear sflow counters

The **clear sflow counters** command resets the global sFlow statistics, which includes the number of samples and sample pool. The hardware trigger count is not reset.

The **show sflow** command displays global sFlow statistics.

Command Mode

Privileged EXEC

Command Syntax

```
clear sflow counters
```

Example

- This command resets the sFlow counters.

```
switch#clear sflow counters
switch#
```

platform petraA buffers mini-multicast

The **platform petraA buffers mini-multicast** command configures the buffer space allocated to mini-multicast use on the Petra chip.

Mini-multicast buffers are used on the Petra platform switches for ingress mirroring and sFlow. On Petra platforms, ingress buffers are divided into three categories: unicast, multicast and mini-multicast. Ingress packets on interfaces configured for sflow sampling or ingress mirroring use mini-multicast buffers. If mini-multicast buffers are exhausted, the packets use unicast buffers but are not candidates for sflow sampling or ingress mirroring.

By default on Petra platforms, EOS allocates 8K mini-multicast buffers and 180K unicast buffers. Up to 64K of the unicast buffers can be reconfigured as mini-multicast buffers, dividing the total buffer pool into 64K mini-multicast buffers and 124K unicast buffers. When implementing sFlow, it is recommended that more buffer space be allocated to mini-multicast buffers to ensure proper sFlow sampling.

Note Executing this command will cause all links on the switch to flap.

Command Mode

Global Configuration

Command Syntax

```
platform petraA buffers mini-multicast num_buf RUNTIME
no platform petraA buffers mini-multicast RUNTIME
default platform petraA buffers mini-multicast RUNTIME
```

Parameters

- *num_buf* number of buffers to allocate to mini-multicast use. Value ranges from 8192 (the default) to 65536.
- ***RUNTIME*** allows execution of the command without confirmation. Options include:
  - <no parameter> CLI prints a warning and asks for confirmation before execution.
  - **now** command is executed immediately.

Example

- The following command allocates 64K of buffer space to mini-multicast buffers.

```
switch(config)#platform petraA buffers mini-multicast 65536
! Command will cause interfaces to flap (links will go down/up).
Proceed with command? [confirm]y
switch(config)#
```

sflow destination

The **sflow destination** command specifies an sFlow collector IP address and UDP port. The switch supports sFlow collector addresses through multiple sFlow destination commands in ***running-config.***

The **no sflow destination** and **default sflow destination** commands remove the specified sFlow collector IP address by deleting the corresponding **sflow destination** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
sflow destination dest_addr [UDP_PORT]
no sflow destination dest_addr [UDP_PORT]
default sflow destination dest_addr [UDP_PORT]
```

Parameters

- *dest_addr* sflow collector's IP address.
- ***UDP_PORT*** sFlow collector's data reception port. Options include:
  - <No parameter> port number 6343 (default).
  - *port_num* port number. Value ranges from 0 to 65535.

Example

- This command configures the switch to send sFlow data to the collector located at 10.42.15.12; the collector receives the data through UDP port 6100.

```
switch(config)#sflow destination 10.42.15.12 6100
switch(config)#
```

sflow enable

The **sflow enable** command enables sFlow on the configuration mode interface when sFlow is globally enabled. By default, sFlow is enabled on all interfaces when sFlow is globally enabled (**sflow run**). The **sflow enable** command is required only when ***running-config*** contains a **no sflow enable** statement for the specified interface.

The **no sflow enable** command disables sFlow on the configuration mode interface. When sFlow is globally disabled, this command persists in ***running-config*** but has no effect on switch operation.

The **default sflow enable** command removes the corresponding **no sflow enable** command from ***running-config,*** enabling sFlow capability on the interface.

Command Mode

Interface-Ethernet Configuration
Interface-Port-Channel Configuration

Command Syntax

```
sflow enable
no sflow enable
default sflow enable
```

Examples

- These commands enable sFlow on the switch and disable sFlow on Ethernet interface 12.

```
switch(config)#sflow run
switch(config)#interface ethernet 12
switch(config-if-Et12)#no sflow enable
switch(config-if-Et12)#
```

- This command removes the **no sflow enable** command for Ethernet interface 12 from ***running-config***, enabling sFlow on the interface whenever sFlow is globally enabled.

```
switch(config-if-Et12)#sflow enable
switch(config-if-Et12)#
```

sflow polling-interval

The **sflow polling-interval** command specifies the counter's polling interval. The switch uses this interval to schedule a port's counter data transmissions to the sFlow collector.

The default interval is two seconds.

The **no sflow polling-interval** and **default sflow polling-interval** commands revert the polling interval to the default of two seconds by removing the **sflow polling-interval** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
sflow polling-interval interval_period
no sflow polling-interval
default sflow polling-interval
```

Parameters

- *interval_period* polling interval (seconds). Value ranges from 0 to 3600 (60 minutes). Default is 2.

Example

- This command configures the switch to send sFlow counter data every ten seconds.

```
switch(config)#sflow polling-interval 10
switch(config)#
```

sflow run

The **sflow run** command globally enables sFlow on the switch. The default sFlow global setting is ***disabled***. sFlow cannot be enabled on individual interfaces when it is globally disabled.

The **sflow enable** interface configuration command controls sFlow operation on individual Ethernet and port channel interfaces when sFlow is globally enabled. When sFlow is enabled globally, sFlow is also enabled on all interfaces by default.

The **no sflow run** and **default sflow run** commands globally disable sFlow on the switch.

Command Mode

Global Configuration

Command Syntax

```
sflow run
no sflow run
default sflow run
```

Examples

- This command enables sFlow on the switch.

```
switch(config)#sflow run
switch(config)#
```

- This command globally disables sFlow.

```
switch(config)#no sflow run
switch(config)#
```

sflow sample

The **sflow sample** command sets the packet sampling rate. Packets are sampled at random intervals to avoid inaccurate sampling of periodic events; the packet sampling rate defines the average number of ingress packets that pass through an interface for every packet that is sampled. A rate of 16384 corresponds to an average sample of one per 16,384 packets. The switch may drop samples if it cannot handle the configured sample rate. Under normal operation, the maximum packet sample rate is one per 16384 packets. Higher sampling rates can be specified with the ***dangerous*** option.

By default, samples include information about the output interface. To remove this information, use the **[no] sflow sample output interface** command.

The **sflow sample** command can also optionally configure sample packets to include information about the traffic class of the sample. Traffic class is communicated by rewriting the DSCP field in the sample packet.

The **no sflow sample** and **default sflow sample** commands reset the packet sampling rate to the default of 1,048,576 and remove output interface and traffic class information from samples by removing the **sflow sample** command from the configuration.

Command Mode

Global Configuration

Command Syntax

```
sflow sample SAMPLE_RATE [rewrite dscp]
no sflow sample
default sflow sample
```

Parameters

- ***SAMPLE_RATE*** size of the packet sample from which one packet is selected. Default sample size is 1048576 packets. Options include:
  - *recommended_rate* Integer between 16384 to 16777215.
  - **dangerous** *any_rate* permits overriding the recommended range of sampling rates. The *any_rate* value range varies by platform:

    fm6000 1 to 65535
    trident 1 to 16777216
    petra 1 and 7895 to 16777216
- **rewrite dscp** configures sFlow to rewrite the DSCP field of sample packets to indicate the traffic class of the original packet.

Examples

- This command configures the sFlow sampling rate as 65536 (one per 65,536 packets).

```
switch(config)#sflow sample 65536
switch(config)#
```

- This command configures the sFlow sampling rate as 256 (one per 256 packets).

```
switch(config)#sflow sample dangerous 256
switch(config)#
```

- This command configures sFlow to include traffic class information in samples.

```
switch(config)#sflow sample rewrite dscp
switch(config)#
```

[no] sflow sample output interface

By default, sFlow samples include information about the output interface of the sampled packet. The **no sflow sample output interface** command prevents sFlow from including that information.

Command Mode

Global Configuration

Command Syntax

no sflow sample output interface

Examples

- This command configures sFlow to *not* include output interface information in samples.

```
switch(config)#no sflow sample output interface
switch(config)#
```

sflow source

The **sflow source** command specifies the address that is listed as the source in all sFlow datagrams that the switch sends to the collector. The source address is normally set to an IP address configured on the switch. This command cannot be used if ***running-config*** contains an **sflow source-interface** command.

The **no sflow source** and **default sflow source** commands remove the **sflow source** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
sflow source source_addr
no sflow source
default sflow source
```

Parameters

- *source_addr* source IP address (dotted decimal notation).

Example

- This command configures 10.2.9.21 as the sFlow source address.

```
switch(config)#sflow source 10.2.9.21
switch(config)#
```

sflow source-interface

The **sflow source-interface** command specifies the interface from which the sFlow source IP address is derived. The switch enters the interface's IP address as the source in sFlow datagrams that it sends to the collector. This command cannot be used if ***running-config*** contains an **sflow source** command.

The **no sflow source-interface** and **default sflow source-interface** commands remove the **sflow source-interface** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
sflow source-interface INT_NAME
no sflow source-interface
default sflow source-interface
```

Parameters

- ***INT_NAME*** Interface type and number. Options include:
  - **interface ethernet** *e_num* Ethernet interface specified by *e_num*.
  - **interface loopback** *l_num* Loopback interface specified by *l_num*.
  - **interface management** *m_num* Management interface specified by *m_num*.
  - **interface port-channel** *p_num* Port-Channel Interface specified by *p_num*.
  - **interface vlan** *v_num* VLAN interface specified by *v_num*.

Example

- This command configures the sFlow source address as the IP address assigned to the loopback 0 interface.

```
switch(config)#sflow source-interface loopback 0
switch(config)#
```

show sflow

The **show sflow** command displays configured sFlow parameters, operational status, and statistics.

The **show sflow interfaces** command displays the interfaces where sFlow is enabled.

Command Mode

EXEC

Command Syntax

show sflow [*INFO_LEVEL*]

Parameters

- ***INFO_LEVEL*** Specifies the information that the command displays: Options include:
  - <no parameter> displays base information
  - **detail** displays base information plus hardware sampling status and number of discarded samples.

Examples

- This command displays the base sFlow information.

```
switch#show sflow
Warning: displaying counters that may be stale
sFlow Configuration
-------------------
Destination IP: 10.67.90.3
Destination Port: 6343 ( default )
Source IP: 0.0.0.0 ( default )
Sample Rate: 16384
Polling Interval (sec): 2.0 ( default )

Status
------
Running: Yes
Polling On: Yes ( default )
Sampling On: Yes ( default )
Send Datagrams: No ( default )
Hardware Sample Rate: 16384

Statistics
----------
Total Packets: 20334189
Number of Samples: 1201
Sample Pool: 19677184
Hardware Trigger: 1205
Number of Datagrams: 356
```

- This command displays the expanded sFlow information.

```
switch#show sflow detail
Warning: displaying counters that may be stale
sFlow Configuration
-------------------
Destination IP: 10.67.90.3
Destination Port: 6343 ( default )
Source IP: 0.0.0.0 ( default )
Sample Rate: 16384
Polling Interval (sec): 2.0 ( default )

Status
------
Running: Yes
Polling On: Yes ( default )
Sampling On: Yes ( default )
Send Datagrams: No ( default )
Hardware Sample Rate: 16384
Hardware Sampling On: No

Statistics
----------
Total Packets: 20334189
Number of Samples: 1201
Sample Pool: 19677184
Hardware Trigger: 1205
Number of Datagrams: 356
Number of Samples Discarded: 0
```

show sflow interfaces

The **show sflow interfaces** command displays the interfaces where sFlow is enabled.

The **show sflow** command displays configured sFlow parameters, operational status, and statistics.

Command Mode

EXEC

Command Syntax

show sflow interfaces

Examples

- This command displays the show sflow interface message when sFlow is globally disabled.

```
switch#show sflow interfaces
sFlow Interface (s):
--------------------
sFlow is not running
```

- This command displays the show sflow interface message when sFlow is globally enabled and enabled on all interfaces.

```
             switch(config)#sflow run
switch(config)#show sflow interfaces
sFlow Interface (s):
--------------------
Ethernet1
Ethernet2
Ethernet3
Ethernet4
Ethernet5
Ethernet6
Ethernet7
Ethernet8
Ethernet9
Ethernet10
Ethernet11
Ethernet12
Ethernet13
Ethernet14
Ethernet15
Ethernet16
Ethernet17
Ethernet18
Ethernet19
Ethernet20
Ethernet21
```

Chapter 45

OpenFlow

This chapter describes Arista's OpenFlow implementation. Sections in this chapter include:

- Section 45.1: OpenFlow Introduction
- Section 45.2: OpenFlow Description
- Section 45.3: OpenFlow Configuration
- Section 45.4: OpenFlow Command Descriptions

45.1 OpenFlow Introduction

Arista EOS supports OpenFlow 1.0 controlled by OpenFlow controllers for filtering and redirecting traffic.

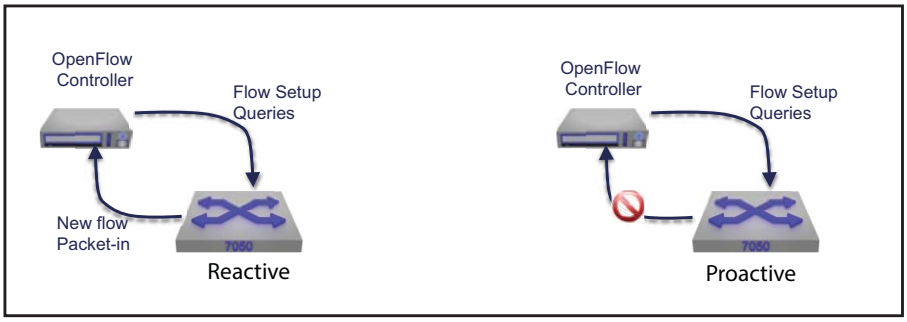
45.2 OpenFlow Description

OpenFlow is a programmable network protocol that manages and directs traffic among Ethernet switches, routers, and wireless access points over the network in support of Software-Defined Networking (SDN) applications. OpenFlow can be used for traffic flow management in metro, WAN, and data center networks, and also security management in enterprise and campus data center applications, and other applications with the appropriate use of OpenFlow controllers.

45.2.1 OpenFlow Controller

The Arista device supports an active controller connection for which the Arista device will initiate (seek) the TCP connection to a given OpenFlow Controller address.

Figure 45-1: Reactive and proactive modes



The controller can be any standard OpenFlow controller.

Switch consists of three parts:

- A flow table, to tell the switch how to process the flow.
- A channel that connects the switch to a remote controller, allowing commands and packets to be sent between a controller and the switch.
- The OpenFlow Protocol, which provides a way for a controller to communicate with a switch.

An OpenFlow-enabled device supports an OpenFlow Client (control plane software), which communicates with an OpenFlow Controller using the OpenFlow protocol. The OpenFlow Controller runs on a server or a server cluster. OpenFlow-enabled devices support the abstraction of a flow table, which is manipulated by the OpenFlow Controller. A flow is a collection of packets where some selected header fields match particular values for those fields. The flow table is sorted by flow priority, which is defined by the controller.

Flow table

Forwarding decisions for incoming packets are decided by a simple lookup on its flow-table entries. Packets that don't match any flow entry are dropped by default. Every flow entry in the flow-table contains:

- Header fields to match against packets: Each entry contains a specific value, or ANY, which matches any value.

Ingress Port	Ether Source	Ether Dst	Ether Type	VLAN Id	IP Proto	Src Dst	Dst Port

- Counters to update for matching packet: These counters are used for statistics purposes, in order to keep track of the number of packets and bytes for each flow and the time that has elapsed since the flow initiation.
- Actions to apply to matching packets: The action specifies the way in which the packets of a flow will be processed. An action can be one of the following: 1) forward the packet to a given port or ports, after optionally rewriting some header fields, 2) drop the packet 3) forward the packet to the controller.

Channel

The channel is the interface that connects each OpenFlow switch to a controller. Through this interface the controller exchanges messages with the switches in order to configure and manage them

45.2.2 OpenFlow Modes

Bind modes

The switch can be configured to divide traffic entering the switch in either of two ways:

- By interface, so that only packets arriving on certain interfaces are processed by OpenFlow (interface bind mode, the default).
- By VLAN, so that only packets associated with certain VLAN IDs are processed by OpenFlow (VLAN bind mode).

Other packets are forwarded normally according to the MAC address table, filtered by ACLs, mirrored to other ports.

Note The hybrid mode of operation is experimental.

The switch can also be configured to apply a limited set of OpenFlow actions to any packets, regardless of ingress interface or VLAN, as well as forward the packets normally (monitor bind mode).

Interface bind mode

When the switch is configured in interface bind mode, the ingress interface of a packet is processed according to entries in the OpenFlow table.

Only interfaces bound to OpenFlow are mapped to OpenFlow ports and exposed to the controller via features reply and port status messages. Output actions in flow table entries and in packet out messages can refer only to mapped ports. Use the **show openflow ports** command to see which interfaces the switch maps to OpenFlow ports and exposes to the controller.

- In OpenFlow configuration mode, use the **bind mode (OpenFlow)** command to select interface bind mode.
- In the OpenFlow configuration mode, use the bind interface command to bind one or more interfaces to OpenFlow.

When an interface is bound to OpenFlow, certain switch functions are disabled on the interface, including spanning tree protocol (STP). The OpenFlow controller and application must ensure that flow table entries do not allow traffic to loop in the network.

Only Ethernet and Port-Channel interfaces can be bound to OpenFlow. If an Ethernet interface is configured as a member of a LAG, attempting to bind the interface to OpenFlow has no effect. However, the Port-Channel interface of which it is a member may itself be bound to OpenFlow.

VLAN bind mode

When a packet arrives at a switch interface, the switch assigns it a VLAN for internal processing, based on the switchport configuration of the ingress interface and on the packet's VLAN tag (if any). If the switch is configured in VLAN bind mode, the internal VLAN determines whether the packet is processed according to entries in the OpenFlow tab and whether the packet is matched by a given entry in the OpenFlow table. After the switch has processed the packet, the switchport configuration of each potential egress interface controls whether the packet is transmitted tagged with the internal VLAN ID, transmitted untagged, or filtered.

Several configuration commands affect whether packets received on a given interface are processed by OpenFlow, and whether packets directed to an interface via an OpenFlow output action are transmitted or filtered:

Use the VLAN configuration mode command to create the VLANs to be accepted by the switch and processed by OpenFlow.

In the interface configuration mode, use switchport commands to configure the interface as either an access port or a trunk port. For an access port, set the VLAN to an OpenFlow VLAN; for a trunk port, configure which OpenFlow VLANs are allowed.

In OpenFlow configuration mode, use the **bind mode (OpenFlow)** command to select VLAN bind mode, and use the **bind vlan (OpenFlow)** command to bind one or more VLANs to OpenFlow.

Untagged packet processing in VLAN bind mode

The OpenFlow protocol also allows a flow table entry to explicitly match untagged packets, or to strip the VLAN tag from matched packets. Since the switch actually assigns a VLAN internally to packets received without a tag, the OpenFlow function on the switch must be configured with a single "native" VLAN ID in order to make sense of such flow entries. When an OpenFlow native VLAN is configured:

- A flow table entry defined to match untagged packets actually matches packets whose internal VLAN is the OpenFlow native VLAN.
- A flow table entry with a strip VLAN tag action actually sets the packet's internal VLAN to the OpenFlow native VLAN.
- Packets sent to the controller via a packet-in message are sent untagged if they are assigned to the native VLAN, and tagged otherwise.
- Untagged packets received from the controller via a packet-out message are assigned to the native VLAN.

In contrast, when no OpenFlow native VLAN is configured:

- Flow table entries defined to match untagged packets or with a strip VLAN tag action are rejected.
- All packets sent to the controller via a packet-in message are sent tagged.
- Untagged packets received from the controller via a packet-out message are dropped.

There is no explicit command to configure the OpenFlow native VLAN. To configure a VLAN as the OpenFlow native VLAN:

- Use the VLAN configuration mode command.
- Every interface handling of OpenFlow traffic, in interface configuration mode, uses switchport commands to configure the interface as either an access port or a trunk port. For an access port, set the access VLAN to N; for a trunk port, either set the native VLAN to N or configure the interface to drop untagged frames.
- In OpenFlow configuration mode, use the **bind vlan (OpenFlow)** command to assign VLAN N to OpenFlow.

Configuring two interfaces as access ports with different OpenFlow-bound VLANs, or as trunk ports with different native OpenFlow-bound VLANs, violates these constraints and causes the OpenFlow function to behave as no OpenFlow native VLAN is configured.

Use the **show openflow** command to see whether an OpenFlow native VLAN has been configured.

Spanning Tree Protocol in VLAN bind mode

STP can operate on OpenFlow-bound VLANs. The switch default STP configuration is one multiple spanning tree (MST) instance containing all VLANs, including OpenFlow-bound VLANs. When STP is configured on OpenFlow-bound VLANs, packets received from or sent to blocked ports are dropped, regardless of the rules defined in the OpenFlow flow table.

For some applications,you may want to disable STP on OpenFlow-bound VLANs. Before doing so, be sure that the OpenFlow controller and application is configured properly to manage multiple redundant paths through the network without allowing traffic to loop.

To ensure proper operation of STP on the switch and to support OpenFlow applications that interoperate with STP, OpenFlow forwards inbound STP packets both to the spanning tree agent on the switch and to the OpenFlow controller as packet-in messages. This behavior overrides any flow table entries that might otherwise match STP packets, and is not configurable.

Monitor bind mode

Unlike interface and VLAN bind modes, monitor bind mode is tailored for specific applications. The switch both forwards traffic normally and selectively mirrors packets under OpenFlow control.

When the switch is configured in monitor bind mode, all traffic entering the switch is forwarded normally, regardless of ingress interface or internal VLAN. All Ethernet and Port-Channel interfaces are mapped to OpenFlow ports and exposed to the controller (except LAG members and mirror destination ports). In this mode, the entire switch is bound to OpenFlow, and OpenFlow processing is applied to packets in addition to the normal forwarding behavior.

Currently the only actions that can be performed on packets in monitor bind mode are:

- Output normally
- Copy to mirror destination port

In monitor bind mode, the default action taken on packets that are not matched by any flow table entry is output normally. The switch rejects flow entries not conforming to these restrictions.

Routing Between the OpenFlow and Non-OpenFlow Domain

The switch can be configured to perform standard IP routing of traffic processed by OpenFlow. From the controller's point of view, the switch appears to have a virtual port 40000 (OpenFlowRouter) in addition to the physical ports.

Packets sent out the OpenFlowRouter port can undergo standard IP routing into a different IP subnet. After routing, those packets can either exit the switch or be processed by OpenFlow again.

45.2.2.1 Port mapping

For switches that support QSFP+ modules, a 40G interface can be configured as four 10G ports. These Ethernet interfaces are mapped to OpenFlow ports according to the formula port = M * 200 + N for EthernetM/N. For example, interface Ethernet1/1 is mapped to OpenFlow port 201; Ethernet1/2 to OpenFlow port 202, Ethernet16/1 to OpenFlow port 3201, Ethernet16/2 to OpenFlow port 3202, and so on.

When IP routing is configured, the OpenFlow Router interface is mapped to OpenFlow port 40000.

Port-Channel (LAG) interfaces are mapped to OpenFlow ports according to the formula port = 40000 + N for Port-ChannelN. For example, interface Port-Channel23 is mapped to OpenFlow port 40023.

The OpenFlow virtual ports all and flood refer to all Ethernet interfaces on the switch, but normal VLAN egress policies apply: a packet tagged with a given OpenFlow-bound VLAN (or untagged, if a native OpenFlow VLAN is configured) will egress a given interface only if the interface is configured to handle traffic for that VLAN. If an interface is not configured to handle traffic for any OpenFlow-bound VLAN, then no packets sent to all or flood will egress on that interface.

45.2.2.2 Queue mapping

All multicast transmit queues that are configured to be mapped from a QoS traffic class are mapped to OpenFlow. OpenFlow-mapped queues can be used by the enqueue action in flow table entries and are included in queue stats reply messages. By default, all the multicast queues 0 to 3 are mapped.

Use the **show qos maps** command to view the current mapping of traffic class to multicast transmit queue, and use the **qos map traffic-class to mc-tx-queue** configuration command to modify it. If no traffic class is mapped to a given multicast transmit queue, the queue will not be mapped to OpenFlow and will be unavailable for use by the enqueue action.

45.2.2.3 Table size

The switch supports one flow table. OpenFlow packet processing is performed in hardware; software forwarding (via the switch CPU) is not supported.

The switch advertises the table size for the l2-match profile. This should be taken as an approximation, as other switch features such as ACLs can consume hardware resources shared with OpenFlow. If the controller attempts to add a flow entry but there are insufficient resources to implement it in hardware, the switch returns an error message.

45.2.2.4 Match fields

A flow table entry can specify an exact value or wildcard for any of the following fields:

- L2 source and destination addresses
- VLAN ID (and untagged packets, if the native OpenFlow VLAN is configured)
- VLAN priority
- L2 frame type
- IPv4 source and destination addresses with subnet masking
- IPv4 TOS/DSCP field
- IPv4 protocol
- TCP/UDP source and destination port numbers

Matching the IPv4 source or destination address within an ARP message is not supported, nor is matching the ARP opcode.

45.2.2.5 Actions

In VLAN and interface bind modes, the following flow entry actions are supported:

- Copy packet on ingress to a mirror destination port (vendor-specific extension)
- Set L2 source and destination addresses
- Set VLAN ID
- Strip VLAN tag (if the native OpenFlow VLAN is configured)
- Set VLAN priority
- Set IPv4 TOS/DSCP

- Output or enqueue to physical port (see Section 45.2.3 for restrictions on multiple output actions)
- Output or enqueue to all or flood (see Section 45.2.3)
- Output to controller (buffering not supported; entire packet contents are always sent)
- Drop (no action)
- Copy packet on egress to a mirror destination port (vendor-specific extension)

In monitor bind mode, only the following actions are supported:

- Copy packet on ingress to a mirror destination port (vendor-specific extension)
- Output per normal forwarding (this action is required in every flow entry)
- Copy packet on egress to a mirror destination port (vendor-specific extension)

45.2.3 OpenFlow Limitations

Consider the following when using OpenFlow:

- OpenFlow is supported on both the 7050 and 7050X series of switches.
- OpenFlow Hybrid mode is not supported.
- Output to an ingress port is silently dropped. Flow table entries with an output to ingress port action are accepted by the switch, but matching packets are not actually forwarded via the ingress port. (But for packet-out, the output to ingress port action is supported.)
- Output/enqueue actions must follow modify actions. The switch will return an error if a modify action follows an output/enqueue action.
- Each action can be performed at most once. The switch will return an error if the same action appears more than once. Output and enqueue actions may appear at most once per port.
- Support output to only one queue. The switch will return an error if multiple enqueue actions appear with different queue ids, or if both enqueue and output actions appear.
- Packet is sent at most once per port even if there are overlapping output or enqueue actions. For example, the switch will accept a rule with actions output to all ports and output to a specific port 12, but will transmit the packet on port 12 only once even though it is contained in both actions.
- Flow entry priority is always respected, even for exact-match flow entries. The switch does not force exact-match flow entries to be processed at the highest priority.
- For packet-out messages, only output actions are supported (to a physical port, or to all, flood, or ingress port). The switch will return an error if a packet-out message is received with any other action.
- The switch-to-controller connection is plain TCP. The switch does not support encrypted TLS connections to the controller.
- Matching source and destination IP and operation code in ARP packets is not supported. Flow entries with matching the ARP Ethernet type are accepted by the switch, but the source and destination IP and protocol (opcode) match field values are ignored (i.e. the fields are wildcarded).
- A flow_mod message with modify or modify_strict command does not modify the cookie value of existing flow entries. If the modify is treated as an add, however, the new entry will be assigned the specified cookie value.
- Matching all 802.3 packets without SNAP headers is not supported. The switch does not treat a dl_type value of 0x5ff as special.
- The port_mod message is not supported. It is not possible to modify the behavior of physical ports via the port_mod message. In particular, the no_flood port_config bit cannot be used to exclude ports from the flood virtual output port set.

- Changing the list of controllers causes the current controller connection to be dropped. When the OpenFlow feature is enabled and the list of controllers is changed in any fashion (e.g. by adding or deleting a controller), the current controller connection will be dropped.
- When adding a large number of flow table entries, add higher-priority entries before lower-priority entries. Due to hardware limitations, the switch will take much longer to add a new flow entry if the table already contains many entries with lower priority.

45.3 OpenFlow Configuration

By default, the OpenFlow feature is disabled on Arista devices. You must first enable the OpenFlow feature on the device.These sections describe OpenFlow configuration tasks:

- Section 45.3.1: Configuration Procedures
- Section 45.3.2: Enabling Basic OpenFlow
- Section 45.3.3: Optional OpenFlow Commands
- Section 45.3.4: Displaying OpenFlow Configurations

45.3.1 Configuration Procedures

Use the OpenFlow configuration mode commands to configure the following basic parameters:

- **openflow**: places the switch in OpenFlow configuration mode.
- **controller (OpenFlow)**: set the controller IP address and port
- **bind interface (OpenFlow)**: bind interfaces to OpenFlow
- **shutdown (Openflow)**: enable or disable OpenFlow

45.3.2 Enabling Basic OpenFlow

Configure the management interface. Assign an IP address to the interface and set the default gateway IP address, allowing the OpenFlow function on the switch establish a TCP connection with the OpenFlow controller.

The following commands turn on OpenFlow pointing to a controller, ready to receive flow setup messages to be programmed in hardware for all traffic.

- The **openflow** command places the switch in OpenFlow configuration mode.

```
switch(config)#openflow
switch(config-OpenFlow)#
```

- The **controller (OpenFlow)** command points to the primary OpenFlow controller. Others can be configured as a standby list.

```
switch(config)#OpenFlow
switch(config-OpenFlow)#controller tcp:15.16.15.16:6633
switch(config-OpenFlow)#
```

- The **bind vlan (OpenFlow)** command dictates what VLAN or interfaces are tied to OpenFlow. Since hybrid mode is not supported, Arista recommends binding all VLANs or all interfaces to OpenFlow.

```
switch(config)#openflow
switch(config-openflow)#controller tcp:1.2.3.4:6633
switch(config-openflow)#bind mode vlan
switch(config-openflow)#bind vlan 1
```

- The **shutdown (Openflow)** command determines if the configuration takes effect or not. The following command enables OpenFlow on the switch.

```
switch(config-OpenFlow)#no shutdown
switch(config-OpenFlow)#
```

45.3.3 Optional OpenFlow Commands

Keepalive

The **keepalive (OpenFlow)** command allows you to set the interval for switch to controller keepalives (default of 10 seconds scales best for large scale multi-node OpenFlow switch networks). After three consecutive reply (from the controller) misses, the switch will try to connect to the second configured controller, if configured.

```
switch(config-OpenFlow)#keepalive
switch(config-OpenFlow)#
```

Profile

The **profile (OpenFlow)** command determines the type of flows. To double flow table size (in case all flows are L2 only), setting a profile of l2-match is best suited. Default is full-match (includes L3/4 field match).

```
switch(config-OpenFlow)#profile l2-match
switch(config-OpenFlow)#
```

Default-action

The **default-action (OpenFlow) c**ommand tells the Arista OpenFlow agent the action that needs to be taken for packets (drop or send-to-controller) that don't match any existing flows programmed locally on the hardware.

```
switch(config-OpenFlow)#default-action drop
switch(config-OpenFlow)#
```

45.3.4 Displaying OpenFlow Configurations

Show commands display the state of OpenFlow running on the Arista switch.

- The **show openflow** command displays the configuration state of the OpenFlow feature and the flows that are actively installed in the hardware of the Arista switch.

```
switch(config)# show openflow
OpenFlow configuration: Enabled
DPID: 0x0000001c73111a92
Description: sw3-Arista
Controllers:
  configured: tcp:172.22.28.228:6633
  connected: tcp:172.22.28.228:6633
  connection count: 3
  keepalive period: 10 sec
Flow table state: Enabled
Flow table profile: full-match
Bind mode: VLAN
  VLANs: 1-2
  native VLAN: 1
IP routing state: Disabled
Shell command execution: Disabled
Total matched: 7977645 packets
```

- The **show openflow flows** command show the default flow that is installed when OpenFlow is enabled.

```
switch(config)# show OpenFlow flows
Flow flow00000000000000000005:
  priority: 100
  cookie: 45035996453121666 (0xa000000ab1ae82)
  match:
    ingress interface: Ethernet3
    Ethernet type: IPv4
    source IPv4 address: 10.0.0.0/255.255.255.0
  actions:
    output interfaces: Ethernet11
  matched: 0 packets, 0 bytes
Flow __default__:
  priority: -1
  cookie: 0 (0x0)
  match:
  actions:
    output to controller
  matched: 5519922 packets, 433188045 bytes
```

45.4 OpenFlow Command Descriptions

OpenFlow Global Configuration Mode

- openflow

Openflow Configuration Commands

- bind interface (OpenFlow)
- bind mode (OpenFlow)
- bind vlan (OpenFlow)
- controller (OpenFlow)
- default-action (OpenFlow)
- description (OpenFlow)
- keepalive (OpenFlow)
- profile (OpenFlow)
- routing recirculation-interface (OpenFlow)
- routing vlan (OpenFlow)
- shell-command allowed (OpenFlow)
- shutdown (Openflow)

OpenFlow Display and Clear Commands

- clear openflow statistics
- show openflow
- show openflow flows
- show openflow ports
- show openflow profiles
- show openflow queues
- show openflow statistics

bind interface (OpenFlow)

When the switch is configured in interface bind mode, the ingress interface of a packet determines whether the packet is processed according to entries in the OpenFlow table or forwarded normally by the switch.

Only interfaces bound to OpenFlow are mapped to OpenFlow ports and exposed to the controller via features reply and port status messages. Output actions in flow table entries and in packet out messages can refer only to mapped ports. Use the **show openflow ports** command to see which interfaces the switch maps to OpenFlow ports and exposes to the controller.

In the OpenFlow configuration mode, use the **bind mode interface** command to select the interface bind mode.

When an interface is bound to OpenFlow, certain switch functions are disabled on the interface, including spanning tree protocol (STP). The OpenFlow controller and application must ensure that flow table entries do not allow traffic to loop in the network.

Only Ethernet and Port-Channel interfaces can be bound to OpenFlow. If an Ethernet interface is configured as a member of a LAG, attempting to bind the interface to OpenFlow has no effect. However, the Port-Channel interface of which it is a member may itself be bound to OpenFlow.

The **no bind interface** and **default bind interface** commands revert the specified list configuration to its default by removing the corresponding **bind interface** command from ***running-config***.

Command Mode

OpenFlow Configuration

Command Syntax

```
bind interface INTF
no bind interface [INTF]
default bind interface [INTF]
```

Parameters

- ***INTF*** Interface that are tied to OpenFlow. Options include:
  - **ethernet** *e_range* Ethernet interfaces specified by *e_range*.
  - **port-channel** *p_range* port channel interfaces specified by *p_range*.

  Valid *e_range* and *p_range* formats include number, range, or comma-delimited list of numbers and ranges.

Example

- This command binds Ethernet 1 to OpenFlow.

```
switch(config)# openflow
switch(config-openflow)#bind interface ethernet 1
```

bind mode (OpenFlow)

The **bind mode** command controls the way packets are divided on ingress between OpenFlow processing and normal switch processing.

The switch can be configured to divide traffic entering the switch in the following ways:

- **Interface bind mode**: Packets entering the switch from certain interfaces are only processed by OpenFlow according to flow table entries; packets entering from other interfaces are forwarded normally. (interface bind mode is the default).
- **VLAN bind mode**: Only packets associated with certain VLAN IDs are processed by OpenFlow.
- **Monitor bind mode**: All packets are forwarded normally, and are also processed by OpenFlow; a restricted set of actions are applied to packets matching a flow table entry.

Other packets are forwarded normally according to the MAC address table, filtered by ACLs, mirrored to other ports.

The switch can also be configured to apply a limited set of OpenFlow actions to any packets, regardless of ingress interface or VLAN, as well as forward the packets normally (monitor bind mode).

The **no bind mode** and **default bind mode** commands revert the specified list configuration to its default by removing the corresponding **bind mode** command from ***running-config***.

Command Mode

Open flow Configuration

Command Syntax

```
bind mode METHOD
no bind mode
default bind mode
```

Parameters

- ***METHOD*** bind interfaces to OpenFlow. Options include:
  - **interface** Only packets arriving on certain interfaces are processed by OpenFlow.
  - **monitor** All packets are forwarded normally, and are also processed by OpenFlow.
  - **vlan** Only packets associated with certain VLAN IDs are processed by OpenFlow.

Example

- In this example, packets received without VLAN tags are assigned to the default VLAN 1 upon entering the switch and are processed by OpenFlow. All VLAN-tagged packets are dropped.

```
switch>enable
switch#configure
switch(config)#interface et1-48
switch(config-if-Et1-48)#switchport mode access
switch(config-if-Et1-48)#switchport access vlan 1
switch(config-if-Et1-48)#exit
switch(config)#openflow
switch(config-openflow)#controller tcp:1.2.3.4:6633
switch(config-openflow)#bind mode vlan
switch(config-openflow)#bind vlan 1
```

bind vlan (OpenFlow)

The **bind vlan** command adds one or more VLAN IDs to the set of VLANs that are processed by OpenFlow in VLAN bind mode. The VLANs must be created separately using the VLAN configuration mode commands.

If you specify a nonexistent VLAN with the **bind vlan** command, the binding will be stored in the running configuration but will not take effect until the VLAN is created.

A range of VLANs may be passed to the **bind vlan** command to add more than one at a time.

The number of VLANs that may be bound to OpenFlow depends on available hardware resources, which are shared with other features including IP routing and ACLs. On the 7050 Series switches the maximum number is 1024.

Use the **show openflow** command to verify which VLANs are bound to OpenFlow; this command reflects the actual hardware state rather than the configuration.

The **no bind vlan** and **default bind vlan** commands removes one or more VLANs from the set of VLANs that are processed by OpenFlow in VLAN bind mode.

Command Mode

OpenFlow Configuration

Command Syntax

```
bind vlan v_range
no spanning-tree vlan [v_range]
default spanning-tree vlan [v_range]
```

Parameters

- *v_range* VLAN list. VLAN numbers range from 1 to 4094.

Examples

- These command bind VLANs 1 and 2 to OpenFlow.

```
switch(config-openflow)#bind mode vlan
switch(config-openflow)#bind vlan 1,2
```

clear openflow statistics

The **clear openflow statistics** command resets the flow statistics for OpenFlow.

Command Mode

Privileged EXEC

Command Syntax

```
clear openflow statistics
```

Example

- This command resets the OpenFlow counters.

```
switch#clear openflow statistics
switch#
```

controller (OpenFlow)

The **controller** command adds the address of an OpenFlow controller to which the switch should connect. The parameter must take the form tcp:1.2.3.4:6633 where 1.2.3.4 is the IP address of the controller and 6633 is the TCP port number.

The **controller** command may be used multiple times to add multiple controllers. The switch will attempt to connect to the first controller in the list of controllers. If the connection attempt fails, or the current connection terminates, the switch will try the next controller in that list, and so on. If the switch cannot connect to the last controller in the list, it will retry with the first controller in the list.

The order in which controllers are added is the order that the switch uses to establish controller connections. This ordering can be seen in the output of the **show openflow** command.

The **no controller** command either removes the specified controller from the list of controllers if a controller address is given as a parameter, or removes all controllers from the list of controllers if **no** parameter is given. If there are no controllers remaining after this command is executed, the OpenFlow function is effectively disabled.

Warning Adding or removing a controller will cause the current controller connection to be dropped. The switch will then attempt to connect to the first controller in the list of controllers, then second controller, and so on.

The **no controller** and **default controller** commands delete s the **controller** statement from ***running-config***.

Command Mode

OpenFlow Configuration

Command Syntax

```
controller tcp:ip_address:tcp_port
no controller tcp:ipaddress:tcp_port default controller tcp:ipaddress:tcp_port
```

Parameters

- *ip_address* ip address used for OpenFlow. Dotted decimal location.
- *tcp_port* name of the TCP port used for OpenFlow. Value ranges from 0 to 65535.

Example

- These commands enable OpenFlow and sets the controller for an OpenFlow instance.

```
switch(config)# openflow
switch(config-OpenFlow)# controller tcp:1.2.3.4:6633
```

default-action (OpenFlow)

The **default-action** command sets the action for the default flow table entry. This entry is automatically added by the switch. It has the lowest priority, and matches packets that are not matched by any other entry.

Use **default-action drop** to change the default entry's action to drop packets instead of sending them to the controller. (Note: In this mode, the switch deviates from the OpenFlow specification.)

The **no default-action** command restores the default entry's action to send packets to the controller.

Command Mode

Openflow Configuration

Command Syntax

```
default-action ACTION_TYPE
no default-action
default default-action
```

Parameters

- ***ACTION_TYPE*** Action for the default flow table entry. Options include:
  - **controller** Sets the default entry's action to send packets to the controller.
  - **drop** Changes the default entry's action to drop packets instead of sending them to the controller.

Example

- This command sets the default entry's action to drop packets instead of sending them to the controller.

```
switch(config)# openflow
switch(config-OpenFlow)# default-action drop
```

description (OpenFlow)

The **description** command allows overriding the switch description string (normally the switch hostname) sent to the controller.

The **no description** and **default description** commands remove the description text for the switch hostname from ***running-config***.

Command Mode

OpenFlow Configuration

Command Syntax

```
description label_text
no description
default description
```

Parameters

- *label_text* character string up to 256 characters assigned to describe the switch.

Examples

- These commands add the description test to the switch

```
switch(config-openflow)#description test
switch(config-openflow)#
```

keepalive (OpenFlow)

The **keepalive** command alters how often the switch sends an OpenFlow echo request to the currently connected controller (every 10 seconds by default). If an echo reply is not received after three successive echo requests, the switch disconnects from the controller. It then attempts to establish a new controller connection depending on the controller configuration.

The **no keepalive** command restores the default keepalive period by removing the **keepalive** command from the ***running-config***.

Command Mode

OpenFlow Configuration

Command Syntax

```
keepalive keep_alive_time
no keepalive
default keepalive
```

Parameters

- *keep_alive_interval_* keepalive period, in seconds. Value ranges from 1 to 100000. Default value is 10 seconds.

Example

- This command sets the keepalive time for OpenFlow to 30 seconds.

```
switch(config)#openflow
switch(config-openflow)#keepalive 30
switch(config-openflow)#
```

openflow

The **openflow** command places the switch in OpenFlow configuration mode.

The **no openflow** and **default openflow** commands delete the openflow configuration mode statements from ***running-config***.

OpenFlow configuration mode is not a group change mode; ***running-config*** is changed immediately upon entering commands. Exiting OpenFlow configuration mode does not affect ***running-config***. The **exit** command returns the switch to global configuration mode.

Command Mode

Global Configuration

Command Syntax

```
openflow
no openflow
default openflow
```

Commands Available in OpenFlow Configuration Mode

- **bind interface (OpenFlow)**
- **bind mode (OpenFlow)**
- **bind vlan (OpenFlow)**
- **controller (OpenFlow)**
- **default-action (OpenFlow)**
- **description (OpenFlow)**
- **keepalive (OpenFlow)**
- **profile (OpenFlow)**
- **routing recirculation-interface (OpenFlow)**
- **routing vlan (OpenFlow)**
- **shell-command allowed (OpenFlow)**
- **shutdown (Openflow)**

Example

- This command places the switch in OpenFlow configuration mode:

```
switch(config)#openflow
switch(config-openflow)#
```

- This command returns the switch to global management mode:

```
switch(config-openflow)#exit
switch(config)#
```

profile (OpenFlow)

The **profile** command sets an alternate flow table profile. Use the **show openflow profiles** command to see the flow table profiles supported by the switch.

The **no profile** and **default profile** commands restores the default flow table profile by removing the profile command from the from ***running-config***.

Command Mode

OpenFlow Configuration

Command Syntax

```
profile FIELD_TYPE
no profile
default profile
```

Parameters

- ***FIELD_TYPe*** Profiles supported by the switch for the active bind mode. Options include:
  - **full-match** Supports matching the full set of OpenFlow match fields.
  - **l2-match** Supports matching only a subset but with a larger maximum number of flow table entries.

Example

- This command advertises the table size for the full-match flow table profile.

```
switch#(config-openflow)# profile full-match
switch#(config-openflow)#
```

routing recirculation-interface (OpenFlow)

The **routing recirculation-interface** command designates a switch interface to recirculate routed OpenFlow traffic for a second pass of processing. Exactly one recirculation interface must be configured to use routing, regardless of the number of VLANs being routed.

Any Ethernet or Port-Channel interface can be used for OpenFlow routing recirculation.

When an interface is configured for OpenFlow routing recirculation:

- The switch programs the hardware into a special MAC loopback mode, so the interface cannot be used to carry normal traffic.
- The link LED turns green and the recirculation function works even if a transceiver is not present or a cable is not inserted.
- The link speed is forced to the maximum.
- Interface configuration commands such as switchport and shutdown are ineffective, although they are preserved in the running configuration and become effective again when the interface is no longer configured for OpenFlow routing recirculation.

The **routing recirculation-interface** and **default routing recirculation-interface** commands revert the t configuration to its default by removing the corresponding **routing recirculation-interface** command from ***running-config***.

Command Mode

OpenFlow Configuration

Command Syntax

```
bind interface INTF
no bind interface [INTF]
default bind interface [INTF]
```

Parameters

- ***INTF*** Options include:
  - **ethernet** *e_range* Ethernet interfaces specified by *e_range*.
  - **port-channel** *p_range* port channel interfaces specified by *p_range*.

  Valid *e_range* and *p_range* formats include number, range, or comma-delimited list of numbers and ranges.

Example

- This command recirculates traffic routed to and from VLAN 1 via the routed transit VLAN 401.

```
switch(config-openflow)#bind mode vlan
switch(config-openflow)#bind vlan 1
switch(config-openflow)#routing recirculation-interface et48
switch(config-openflow)#routing vlan 1 routed-vlan 401
switch(config-openflow)#enable
```

routing vlan (OpenFlow)

The **routing vlan** command enables IP routing of traffic processed by OpenFlow for a specific VLAN.

The **no routing vlan** and **default routing vlan** command disables IP routing of traffic processed by OpenFlow for a VLAN.

Command Mode

OpenFlow Configuration

Command Syntax

```
routing vlan VLAN_ID routed-vlan vlan_transit
no routing vlan VLAN_ID
default routing vlan VLAN_ID
```

Parameters

- ***VLAN_ID*** Options include
  - *v_num* The full form of the command is **routing vlan 123 routed-vlan 456,** where 123 is the VLAN of the OpenFlow traffic to be routed, and 456 is a (non-OpenFlow-bound) VLAN configured for standard IP routing.
  - **untagged** To route untagged OpenFlow traffic. use the command routing vlan untagged routed-vlan 456

Examples

- This command associates the VLAN with an untagged VLAN 22 to match during the OpenFlow pass.

```
switch(config-openflow)# routing vlan untagged routed-vlan 22
```

shell-command allowed (OpenFlow)

The **shell-command allowed** command allows the controller to run shell or CLI vendor extension commands on the switch.

When this extension is enabled, the switch will execute any CLI command sent by the controller, bypassing normal access controls, so enable it only if the controller is trusted.

The **no shell-command allowed** and **default shell-command allowed** commands disables the corresponding **shell-command allowed** from the ***running-config***.

Command Mode

OpenFlow Configuration

Command Syntax

```
shell-command allowed
no shell-command allowed
default shell-command allowed
```

Example

- This command allows the controller to run arbitrary CLI commands on the switch.

```
switch(config)#openflow
switch(config-openflow)#shell-command allowed
switch(config-openflow)#
```

show openflow

The **show openflow** command shows the effective OpenFlow configuration parameters.

Command Mode

EXEC

Command Syntax

```
show openflow
```

Example

- This command displays the actual hardware state of OpenFlow.

```
switch# show openflow
OpenFlow configuration: Enabled
DPID: 0x000000123456789a
Description: My awesome OpenFlow switch
Controllers:
  configured: tcp:1.2.3.4:6633 tcp:5.6.7.8:6633
  connected: tcp:1.2.3.4:6633
  attempted connection count: 24
  successful connection count: 1
  keepalive period: 10 sec
Flow table state: Enabled
Flow table profile: full-match
Bind mode: interface
  interfaces: Ethernet2, Ethernet4, Ethernet6, Ethernet8
IP routing state: Enabled
  recirculation interface: Ethernet44
  VLAN untagged: routed to/from VLAN 3636
Shell command execution: Disabled
Total matched: 4601 packets
switch#
```

show openflow flows

The **show openflow flows** command displays the contents of the flow table, showing each entry with its match rules, actions, packet counters, and timeouts.

The default flow table entry is automatically created by the switch. It always has the lowest priority, and matches packets that are not matched by any other entry. The default entry's action is to send the packet to the controller.

Command Mode

EXEC

Command Syntax

show openflow flows

Example

- This command displays the contents of the flow table.

```
switch# show openflow flows
Flow flow00000000000000000002:
  priority: 0
  cookie: 0 (0x0)
  idle timeout: 60.0 sec
  match:
    ingress interface: Ethernet2
    source Ethernet address: 00:a9:87:65:43:21
    destination Ethernet address: 00:12:34:56:78:9a
    untagged/native VLAN ID
    VLAN PCP: 0
    Ethernet type: IPv4
    source IPv4 address: 10.0.1.1
    destination IPv4 address: 10.0.1.2
    IPv4 TOS: 0
    IPv4 protocol: ICMP
    source TCP/UDP port or ICMP type: 8
    destination TCP/UDP port or ICMP code: 0
  actions:
    output interfaces: OpenFlowRouter
  matched: 4 packets, 408 bytes
Flow __default__:
  priority: -1
  cookie: 0 (0x0)
  match:
  actions:
    output to controller
switch#
```

show openflow ports

The **show openflow ports** command displays the mapping between OpenFlow port number and switch interface.

In interface bind mode, all OpenFlow-bound interfaces (except routed ports and LAG members) are mapped to OpenFlow ports and exposed to the controller.

In VLAN bind mode, Ethernet and Port-Channel interfaces (except routed ports and LAG members) configured to carry traffic for one or more OpenFlow-bound VLANs are mapped to OpenFlow ports and exposed to the controller.

In monitor bind mode, all Ethernet and Port-Channel interfaces (except routed ports and LAG members) are mapped to OpenFlow ports and exposed to the controller.

Command Mode

EXEC

Command Syntax

```
show openflow ports
```

Example

- This command displays which interfaces the switch maps to OpenFlow ports.

```
switch# show openflow ports
Port 1: Ethernet1
Port 15: Ethernet15
switch#
```

show openflow profiles

The **show openflow profiles** command displays the flow table profiles supported by the switch for the active bind mode. For each profile, it shows:

- Which fields can be matched by a flow table entry and which can be wildcarded
- Which actions are supported for matched packets (in monitor bind mode, only normal and mirror actions are supported)
- The maximum number of entries that can be added to the flow table

The hardware resources available to OpenFlow are shared with other switch features like ACLs, so the actual maximum number of flow entries may be lower than the number shown by **show openflow profiles** command.

On Series 7050 switches, two profiles are available: the full-match profile supports matching the full set of OpenFlow match fields with a maximum of 750 flow table entries, while the l2-match profile supports matching only a subset but with a larger maximum number of flow table entries (1500).

Command Mode

EXEC

Command Syntax

```
show openflow profiles
```

Example

- This command displays the flow table profiles.

```
switch#show openflow profiles
full-match:
  Match fields:
    ingress interface
    source Ethernet address
    destination Ethernet address
    VLAN ID
    VLAN PCP
    Ethernet type
    source IPv4 address
    destination IPv4 address
    IPv4 TOS
    IPv4 protocol
    source TCP/UDP port or ICMP type
    destination TCP/UDP port or ICMP code
  Wildcard fields:
    ingress interface
    source Ethernet address
    destination Ethernet address
    VLAN ID
    VLAN PCP
    Ethernet type
    source IPv4 address
    destination IPv4 address
    IPv4 TOS
    IPv4 protocol
    source TCP/UDP port or ICMP type
    destination TCP/UDP port or ICMP code
  Actions:
    copy ingress to mirror dest interfaces
    forward normally
    copy egress to mirror dest interfaces
  Table size: 750 entries max
l2-match:
  Match fields:
    ingress interface
    source Ethernet address
    destination Ethernet address
    VLAN ID
    VLAN PCP
    Ethernet type
  Wildcard fields:
    ingress interface
    source Ethernet address
    destination Ethernet address
    VLAN ID
    VLAN PCP
    Ethernet type
    source IPv4 address
    destination IPv4 address
    IPv4 TOS
    IPv4 protocol
    source TCP/UDP port or ICMP type
    destination TCP/UDP port or ICMP code
  Actions:
    copy ingress to mirror dest interfaces
```

```
    forward normally
    copy egress to mirror dest interfaces
  Table size: 1500 entries max
switch#
```

show openflow queues

The **show openflow queues** command displays the queues exposed to the OpenFlow controller for each switch interface, and packet and byte counters for each queue.

Command Mode

EXEC

Command Syntax

show openflow queues

Example

- This command displays the packet and byte counters for each queue on the active OpenFlow interfaces.

```
switch#show openflow queues
Port 1 (Ethernet1):
  Queue 0: 0 packets (0 bytes) transmitted, 0 dropped
  Queue 1: 0 packets (0 bytes) transmitted, 0 dropped
  Queue 2: 0 packets (0 bytes) transmitted, 0 dropped
  Queue 3: 0 packets (0 bytes) transmitted, 0 dropped
Port 15 (Ethernet15):
  Queue 0: 0 packets (0 bytes) transmitted, 0 dropped
  Queue 1: 0 packets (0 bytes) transmitted, 0 dropped
  Queue 2: 0 packets (0 bytes) transmitted, 0 dropped
  Queue 3: 0 packets (0 bytes) transmitted, 0 dropped
switch#
```

show openflow statistics

The **show openflow statistics** command displays statistics sampled every 5 seconds over the past 5 minutes:

- Number of entries in the flow table
- Number of flow_mod, packet_out and packet_in messages processed in the 5-second interval
- Number of packet_out messages dropped in the 5-second interval (the OpenFlow agent starts dropping packet_out messages when the transmit queue of the controller TCP connection exceeds 50% of capacity)

Command Mode

EXEC

Command Syntax

```
show openflow statistics
```

Example

- This command displays statistics sampled every 5 seconds.

```
switch# show openflow statistics
                      table       messages processed last 5 sec     dropped
                    entries   (flow_mod)(packet_out) (packet_in) last 5 sec
2013-08-16 14:48:06       4           0           0           0           0
2013-08-16 14:48:01       4           2           2           2           0
2013-08-16 14:47:56       0           0           2           2           0
2013-08-16 14:47:51       4           0           0           0           0
2013-08-16 14:47:46       4           0           0           0           0
2013-08-16 14:47:41       4           0           0           0           0
2013-08-16 14:47:36       4           0           0           0           0
2013-08-16 14:47:31       4           2           2           2           0
2013-08-16 14:47:26       0           0           0           0           0
2013-08-16 14:47:21       4           0           0           0           0
2013-08-16 14:47:16       4           0           0           0           0
2013-08-16 14:47:11       4           0           0           0           0
2013-08-16 14:47:06       4           0           0           0           0
2013-08-16 14:47:01       4           2           2           2           0
2013-08-16 14:46:56       4           2           2           2           0
2013-08-16 14:46:51       4           0           0           0           0
2013-08-16 14:46:46       0           0           0           0           0
2013-08-16 14:46:41       4           0           2           2           0
2013-08-16 14:46:36       4           0           2           2           0
2013-08-16 14:46:31       4           0           0           0           0
2013-08-16 14:46:26       4           0           0           0           0
2013-08-16 14:46:21       4           2           2           2           0
2013-08-16 14:46:16       4           2           2           2           0
2013-08-16 14:46:11       4           0           2           2           0
2013-08-16 14:46:06       0           0           0           0           0
2013-08-16 14:46:01       0           0           0           0           0
2013-08-16 14:45:56       0           0           0           0           0
2013-08-16 14:45:51       0           0           0           0           0
2013-08-16 14:45:46       0           0           0           0           0
2013-08-16 14:45:41       0           0           0           0           0
2013-08-16 14:45:36       0           0           0           0           0
2013-08-16 14:45:31       0           0           0           0           0
2013-08-16 14:45:26       0           0           0           0           0
2013-08-16 14:45:21       4           0           0           0           0
switch#
```

shutdown (Openflow)

The **shutdown** command, in OpenFlow mode, disables OpenFlow on the switch. OpenFlow is disabled by default.

The **no shutdown** and **default shutdown** commands re-enable OpenFlow by removing the **shutdown** command from ***running-config***.

Command Mode

Openflow Configuration

Command Syntax

```
shutdown
no shutdown
default shutdown
```

Example

- These commands enable OpenFlow on the switch.

```
switch(config)#openflow
switch(config-openflow)#no shutdown
switch(config-openflow)#
```

- This command disables OpenFlow.

```
switch(config-openflow)#shutdown
```

Chapter 46

DirectFlow

This chapter describes Arista's DirectFlow implementation. Sections in this chapter include:

- Section 46.1: Introduction
- Section 46.2: DirectFlow Configuration
- Section 46.3: DirectFlow Feature Interactions
- Section 46.4: DirectFlow Command Descriptions

46.1 Introduction

Like OpenFlow, DirectFlow exposes the underlying forwarding ASICs capabilities through a programmable interface like EAPI or the standard CLI.

Unlike OpenFlow, DirectFlow works in conjunction with all other aspects of standard L2/L3 bridging or forwarding, and DirectFlow traffic is subject to the standard packet processing pipeline within the ASIC. You can think of DirectFlow as a stage in packet processing that processes traffic after ingress checks and before any egress actions.

This feature enables you to configure flows that consist of a matching criteria and actions, and to modify how traffic is processed (for example, by overriding the L2 lookup decision or rewriting a mac address or VLAN).

Features like MAC learning, STP state checks, ingress or egress VLAN membership checks on ports, ACLs, QoS and other features are all respected by DirectFlow. Traffic that doesn't match any programmed flow is processed normally while traffic that matches programmed flows is now subject to the actions specified in the flows.

DirectFlow and OpenFlow are mutually exclusive and you can run only one of the two at any given time.

How DirectFlow is different from OpenFlow

There is no default flow matching all traffic, traffic not matched by other rules is forwarded as normal.This means the configuration/ controller/ application doesn't consume TCAM space programming flows for normal forwarding.

DirectFlow works with other features and so the user can use ACL, rate limiting, STP etc. in their network as normal and not build all of that into the application used to inject flows.

DirectFlow flows can be configured from the CLI or using EAPI, giving users the option of using flow based forwarding without an external controller. This is especially useful where the number of flows is small and static e.g.to process a small subset of the traffic in a different manner to the normal L2/L3 pipeline.

Unlike OpenFlow which requires the switch support OUTPUT NORMAL or re-circulate a packet in order to send a packet from the OpenFlow domain to non-OpenFlow domain, there is just one domain with DirectFlow.

46.1.1 DirectFlow Flows

Similar to OpenFlow, you can define a relative priority between flows and define idle or hard timeouts for the flow. DirectFlow also enables you to insert a flow entry that matches on specified criteria, and define actions to be taken on traffic that matches the specified matching conditions. You can define flows to match on TCP flags, IPv6 source and destination addresses, input ports, and more.

For more information, see:

- Section 46.1.1.1: DirectFlow Non-persistent Flows
- Section 46.1.1.2: Supported matches
- Section 46.1.1.3: Supported actions

46.1.1.1 DirectFlow Non-persistent Flows

DirectFlow enables you to configure flows that are not visible in the startup or running configurations. This feature is designed to be used for flows that are configured by a custom agent using the EOS SDK or eAPI and age out (expire) after a specified time period.

For example, if you are using a custom agent that reacts to traffic sent to the CPU (the redirect to CPU action), and you want to use a flow that will drop all matching traffic for 5 minutes, the agent can program a non-persistent flow that expires after a hard timeout of 300 seconds.

Using a non-persistent flow for this purpose ensures that other administrator actions (for example, saving the configuration) does not result in the flow being resurrected on startup or reverting to the saved configuration. It also removes the need for the agent to delete the expired flow.

Note By default, all direct flow flows are persistent. You must use the **no persistent** command to configure a non-persistent flow.

46.1.1.2 Supported matches

DirectFlow supports all matches supported on EOS with OpenFlow 1.0.

This includes matches on VLAN, ether type, source or destination MAC address, COS, source or destination IP address, IP protocol, IP TOS, L4 source, destination ports, ICMP type and code).

In addition, DirectFlow also allows matching on:

- TCP flags
- IPv6 source address
- IPv6 destination address
- Traffic injected from the CPU
- Input port

DirectFlow also permits re-using the same flow on multiple input ports, saving valuable TCAM space.

46.1.1.3 Supported actions

DirectFlow supports all actions supported on EOS with OpenFlow 1.0, including:

- Setting the source or destination MAC address

- VLAN
- COS
- IP TOS
- Transmit queue
- Output port list and mirroring traffic pre-modification (ingress mirror) and post-modification (egress mirror).
- Redirect to CPU

  The redirect to CPU action is useful in cases in which a custom agent is running on EOS and you want to trap specific traffic (matching traffic) and send the trapped traffic to the agent. See the example “Redirect to CPU” on page 2573.

46.2 DirectFlow Configuration

Consider the following when using DirectFlow.

- DirectFlow takes effect ONLY after exiting the individual flow configuration sub-mode.
- Match criteria are connected with Boolean AND operators i.e. they must **all** match for the condition to be true and action to be taken.
- CLI is automatically set to match the ethertype to IP if IP fields (such as source or destination address or L4 ports) are chosen as part of other match/ action commands.
- In a single flow, only the following fields can be matched along with IPv6 source and destination addresses:
  - VLAN priority
  - VLAN ID
  - EtherType
  - Source interface
  - Class of Service (CoS)

46.2.1 Commands Used to Enable DirectFlow, Configure and Display Flows

A number of different commands are provided for the DirectFlow feature. The different commands enable you to enter the DirectFlow configuration mode, enable DirectFlow, configure flows, and display configured flows.

Important! ALL match criteria specified in a flow definition must match in the packet for the actions specified to be applied to the traffic.

Enter the DirectFlow configuration mode

The **directflow** command places the switch in DirectFlow configuration mode.

```
switch(config)#directflow
switch(config-directflow)#
```

Enable DirectFlow

The **shutdown (DirectFlow)** command determines if the configuration takes effect or not. To enable DirectFlow, enter the following command.

```
switch(config-directflow)#no shutdown
switch(config-directflow)#
```

Create the flow

The **flow (DirectFlow)** command creates a new flow entry. It must be unique or it will be overwritten by an existing entry.

```
switch(config-directflow)#flow Test-1
switch(config-directflow-Test-1)#
```

Create the DirectFlow match criteria

The **match (DirectFlow-flow mode)** command allows you to configure a rule or a flow which match on L2, L3, L4 fields of a packet and specify a certain action to either modify, drop or redirect the packet.

```
switch(config-directflow)#flow Test1
switch(config-directflow-Test1)#match ethertype ip
switch(config-directflow-Test1)#match source ip 10.10.10.10
```

Action Set

The **action set (DirectFlow-flow mode)** command allows you to configure a packet to be routed out a layer three interface using a DirectFlow entry.

```
switch(config-directflow)#flow Test1
switch(config-directflow-Test1)#action egress mirror ethernet 7
switch(config-directflow-Test1)#action set destination mac 0000.aaaa.bbbb
```

Redirect to CPU

The **action output interface cpu (DirectFlow-flow mode)** command allows you to configure flows so that traffic that matches the matching conditions specified in the flow is redirected to the CPU.

```
switch(config)#directflow
switch(config-directflow)#flow redirect-http-cpu
switch(config-directflow-redirect-http=cpu)#match ip protocol tcp
switch(config-directflow-redirect-http-cpu)#match destination port 80
switch(config-directflow-redirect-http-cpu)#action output interface cpu
```

Configuring a non-persistent flow

Including the **no persistent** command allows you to configure non-persistent direct flow flows.

```
switch (config-directflow)#flow example-non-persistent
switch (config-directflow-example-non-persistent)#match input interface ethernet 25
switch (config-directflow-example-non-persistent)#action drop
switch (config-directflow-example-non-persistent)#no persistent
switch (config-directflow-example-non-persistent)#timeout hard 300
```

Display details for configured flows

The **show directflow flow <flow name> detail** command enables you to display the details of configured flows. You can use this command to verify that a non-persistent flow is deleted after the timeout period configured for the flow has elapsed.

The following example shows the use of this command to view the configuration of a non-persistent flow before the timeout period has elapsed, and a second time, after the timeout period has expired.

The initial use of the command displays the flow configuration (before the timeout expires).

```
switch (config-directflow)#show directflow flows example-non-persistent detail
Flow example-non-persistent: (Flow programmed)
 persistent: False
 priority: 0
 hard timeout: 300
 idle timeout: 0
 match:
  ingress interface:
      Et25
 actions:
  drop
 matched: 0 packets, 0 bytes
```

The **second** use of the command displays the flow details (after the timeout expires). The output shows that the flow is no longer programmed.

```
switch (config-directflow)#show directflow flows example-non-persistent detail
Flow example-non-persistent: (Flow not programmed)
persistent: False
priority: 0
hard timeout: 300
idle timeout: 0
match:
  ingress interface:
      Et25
actions:
  drop
matched: 0 packets, 0 bytes
```

46.3 DirectFlow Feature Interactions

DirectFlow flow entries can have one of the following actions:

- A set of egress ports for sending a matched packet
- Copy to CPU
- Redirect to CPU
- Drop (default)
- No specified action (in this case, the traffic is output normally).

The only exception is the ingress or egress mirroring action, where the DirectFlow entry causes the packet to be mirrored.

When the ingress or egress packets are mirrored, the original traffic is sent out normally.

Bridging Features

- DirectFlow entries have precedence over all entries in the MAC table, including static MAC entries and static MAC drop entries. Packets that do not match DirectFlow entries are forwarded based on the MAC address table.
- VLANs: DirectFlow entries can modify the VLAN of a packet. MAC learning takes place in the original VLAN for DirectFlow entries that modify the VLAN. The modified packet will be subject to VLAN membership checks on the egress port. If a packet has no VLAN tag, DirectFlow assumes it came in on the native VLAN for the ingress interface. A VLAN override causes the packet to obey the VLAN rules on the egress port.
- Q-in-Q: Q-in-Q is supported as DirectFlow entries match only on the outer tag.
- Counters: All packets that match DirectFlow entries cause interface counters to increment as usual.

Spanning Tree

DirectFlow runs alongside MSTP, RSTP, and PVST. DirectFlow entries do not match on packets that ingress an STP discarding port. DirectFlow entries that cause a packet to be forwarded out an STP discarding port will result in the packets being dropped on egress.

When STP is enabled, BPDUs will always be trapped to the CPU. When STP is disabled, BPDUs will be subject to DirectFlow entries and not be copied to the CPU by default.

LLDP, LAGs, and LACP

- LLDP packets are always trapped to the CPU. DirectFlow entries can never match LLDP packets.
- LAGs are fully supported, and can be part of a match criteria and part of an output action to an interface.
- LACP packets are always trapped to CPU. DirectFlow entries can never match LACP packets.

sFlow

sFlow is unaffected by DirectFlow.

IGMP Snooping

IGMP control packets are trapped to the CPU when IGMP Snooping is enabled. DirectFlow entries can match IGMP Snooping control traffic and override the trap to CPU.

Link-local-multicast packets are flooded in hardware in the VLAN via a TCAM entry. DirectFlow entries can match link-local-multicast packets and change the flooding behavior. As DirectFlow entries have to specify output interfaces or drop, the action will conflict and so matching DirectFlow entries will get precedence.

When IGMP snooping is enabled, unknown IPV4 multicast packets are flooded to the multicast-router ports in the VLAN. If DirectFlow entries match unknown IPV4 multicast packets, they will override the flooding behavior.

Data packets in groups under IGMP snooping control are sent to the group members through a MAC table entry. Matching DirectFlow entries override the MAC table entries.

ACLs

DirectFlow entries are lower priority than any configured Port ACLs (ingress). Packets coming in on a port that match DirectFlow entries obey any configured ACL on that port, and will only apply to packets that have a ***permit*** action.

DirectFlow entries are higher priority than any configured RACLs. Packets coming in on an L3 interface that match DirectFlow entries ignore any RACLs configured on that interface.

DirectFlow entries are lower priority than any configured Egress ACLs.

46.3.1 Layer Three Features and DirectFlow

DirectFlow runs alongside IP routing. If a packet is routed out a layer three interface using a DirectFlow entry, the actions associated with the entry will have to specify the new source MAC and destination MAC for the packet, as well as the physical port or LAG. If there are no output ports specified in an entry, packets that match that entry will be dropped.

Unicast Routing

When unicast routing is enabled, DirectFlow entries that match take precedence for all packets that would have been otherwise been routed. The three exceptions are the ingress mirror, egress mirror and copy-to-CPU actions where the packets will be routed normally in addition to the action being performed. Routed packets that do not match DirectFlow entries are forwarding based on the L3 lookup.

Multicast Routing

When multicast routing is enabled, DirectFlow entries that match take precedence for all packets that would have otherwise been multicast routed. The packets are not replicated based on the hardware multicast tables, but are forwarded strictly according to the actions specified by the DirectFlow entry. The entry can specify a set of output interfaces, which will result in the packet being replicated based on the DirectFlow entry.

46.3.2 Displaying DirectFlow Configurations

To **show directflow flows** command displays the contents of the flow table, showing each entry with its match rules, actions, and packet counters.

- This example shows the status of a default (persistent) flow.

```
switch(config-directflow)#show directflow flows
Flow Test1:
 priority: 0
 match:
   ingress interface: Ethernet1
   ethertype ip
   source ip address: 10.10.10.10
 actions:
   output mirror: Ethernet2
 matched: 0 packets, 0 bytes
switch(config-directflow)#
```

- This example shows the status of a non-persistent flow. The flow will be deleted once 5 minutes have elapsed.

```
switch(config-directflow)#show directflow flows example-non-persistent
Flow example-non-persistent:
 persistent: False
 priority: 0
 hard timeout: 300
 idle timeout: 0
 match:
   ingress interface:
       Et25
 actions:
   drop
 matched: 0 packets, 0 bytes
```

46.4 DirectFlow Command Descriptions

DirectFlow Global Configuration Mode

- directflow

DirectFlow Configuration Command

- action drop (DirectFlow-flow mode)
- action mirror (DirectFlow-flow mode)
- action output (DirectFlow-flow mode)
- action output interface cpu (DirectFlow-flow mode)
- action set (DirectFlow-flow mode)
- flow (DirectFlow)
- match (DirectFlow-flow mode)
- priority (DirectFlow-flow mode)
- shutdown (DirectFlow)
- timeout (DirectFlow-flow mode)

DirectFlow and Clear Commands

- show directflow
- show directflow flows

action drop (DirectFlow-flow mode)

The **action drop** command configures packets that match an entry to be dropped.

The **no action drop** and **default action drop** commands remove the statement from the DirectFlow configuration mode.

Command Mode

Directflow-flow Configuration

Command Syntax

```
action drop
no action drop
default action drop
```

Example

- This command sets the action for packets from Test-1 to be dropped.

```
switch(config-directflow-Test-1)#action drop
switch#
```

action mirror (DirectFlow-flow mode)

The **action mirror** command can be used to ingress or egress mirror traffic to a mirror destination. This requires a mirror destination to be setup on the switch. If a packet comes in or goes out an interface that is part of another mirror session, then the destination for that destination as well as the DirectFlow destination will receive a copy of the packet.

The **no action mirror** and **default action mirror** commands remove the statement from DirectFlow configuration mode.

Command Mode

Directflow-flow Configuration

Command Syntax

```
action DIRECTION mirror INT_NAME
no action DIRECTION mirror INT_NAME
default action DIRECTION mirror INT_NAME
```

Parameters

- ***DIRECTION*** transmission direction of traffic to be mirrored.
  - **ingress** mirrors before any rewrites.
  - **egress** mirrors after rewrites.
- ***INT_NAME*** Source interface for the mirroring session.
  - **ethernet** *e_range* Ethernet interfaces specified by *e_range*.
  - **port-channel** *p_range* Port channel interfaces specified by *p_range*.

Example

- This command configures mirror traffic to Ethernet 2.

```
switch(config-directflow)# flow Test1
switch(config-directflow-Test1)#match ethertype ip
switch(config-directflow-Test1)#match source ip 10.10.10.10
switch(config-directflow-Test1)#action egress mirror ethernet 2
switch(config-directflow-Test1)#
```

action output (DirectFlow-flow mode)

The **action output** command configures an Ethernet or port channel interface as the output of a specified port mirroring session.

The **no action output** and **default action output** commands remove the statement from DirectFlow configuration mode.

Command Mode

Directflow-flow Configuration

Command Syntax

```
action output DESTINATION
no action output DESTINATION
default action output DESTINATION
```

Parameters

- ***DESTINATION*** transmission direction of traffic to be mirrored.
  - **all** mirrors transmitted and received traffic.
  - **flood** mirrors received traffic only.
  - **interface ethernet** *e_range* Ethernet interfaces specified by *e_range*.
  - **interface port-channel** *p_range* Port channel interfaces specified by *p_range*.

Example

- This command configures Ethernet interface 7 as the output for the mirroring session.

```
switch(config-directflow-Test1)#action output interface ethernet 7
switch(config-directflow-Test1)#
```

action output interface cpu (DirectFlow-flow mode)

The **action output interface cpu** command configures the action (other commands are used to define the traffic matching conditions).

The **no action output interface cpu** and **default action output** commands remove the statement from DirectFlow configuration mode.

Command Mode

Directflow-flow Configuration

Command Syntax

```
action output DESTINATION
no action output DESTINATION
default action output DESTINATION
```

Parameters

- ***DESTINATION*** transmission direction of traffic to be mirrored.
  - **all** mirrors transmitted and received traffic.
  - **flood** mirrors received traffic only.
  - **interface cpu** Ethernet interfaces specified by *e_range*.

Example

- This command configures Ethernet interface 7 as the output for the mirroring session.

```
switch(config-directflow-Test1)#action output interface ethernet 7
switch(config-directflow-Test1)#
```

- These commands configure the action to redirect traffic matching the flow to the CPU and the matching conditions for the flow.

```
switch (config)#directflow
switch (config-directflow)#flow redirect-http-cpu
switch (config-directflow-redirect-http=cpu)#match ip protocol tcp
switch (config-directflow-redirect-http-cpu)#match destination port 80
switch (config-directflow-redirect-http-cpu)#action output interface cpu
```

action set (DirectFlow-flow mode)

The **action set** command allows you to configure a packet to be routed out a layer three interface using a DirectFlow entry. The actions associated with the entry will have to specify the new source MAC and destination MAC for the packet, as well as the physical port or LAG. If there are no output ports specified in an entry, packets that match that entry will be dropped.

The **no action set** and **default action set** commands remove **action set** statement from DirectFlow configuration mode.

Command Mode

Directflow-flow Configuration

Command Syntax

```
action set CONDITION
no action set CONDITION
default action set CONDITION
```

Parameters

- ***CONDITION*** specifies parameter and value. Options include:
  - **cos** <***0*** to ***7***> cost of service.
  - **destination mac** *mac_addr* Dotted hex notation.
  - **ip tos** <***0*** to ***255***> Type of service.
  - **source mac** *mac_addr* Dotted hex notation.
  - **traffic-class** <***0*** to ***7***> Dotted hex notation.
  - **vlan** <***1*** to ***4094***> Number of VLAN.

  The **no action set** and **default action set** commands require only the ***CONDITION*** type without a specific condition value.

Example

- These commands change the destination MAC of the frame.

```
switch(config-directflow)#flow Test1
switch(config-directflow-Test1)#action egress mirror ethernet 7
switch(config-directflow-Test1)#action set destination mac 0000.aaaa.bbbb
```

directflow

The **directflow** command places the switch in DirectFlow configuration mode.

The **no directflow** and **default directflow** commands delete the DirectFlow configuration mode statements from ***running-config***.

DirectFlow configuration mode is not a group change mode; ***running-config*** is changed immediately upon entering commands. Exiting OpenFlow configuration mode does not affect ***running-config***. The **exit** command returns the switch to global configuration mode.

Command Mode

Global Configuration

Command Syntax

```
directflow
no directflow
default directflow
```

Commands Available in DirectFlow-Flow configuration mode:

- **flow (DirectFlow)**
- **shutdown (DirectFlow)**

Example

- This command places the switch in DirectFlow configuration mode:

```
switch(config)#directflow
switch(config-directflow)#
```

- This command returns the switch to global management mode:

```
switch(config-directflow)#exit
switch(config)#
```

flow (DirectFlow)

The **flow** command places the switch in flow configuration mode.

The **flow** command specifies the name of the flow that subsequent commands modify and creates a newflow definition if it references a nonexistent flow. All changes in a flow configuration mode edit session are pending until the session ends:

- The **exit** command saves pending changes to ***running-config*** and returns the switch to DirectFlow configuration mode. Changes are also saved by entering a different configuration mode.
- The **abort** command discards pending changes, returning the switch to DirectFlow configuration mode.

The **no flow** and **default flow** commands delete the specified role by removing the role and its statements from ***running-config***.

Command Mode

DirectFlow Configuration

Command Syntax

```
flow flow_name
no flow flow_name
default flow flow_name
```

Parameters

- *flow_name* Name of flow.

Commands Available in DirectFlow-Flow configuration mode:

- **action drop (DirectFlow-flow mode)**
- **action mirror (DirectFlow-flow mode)**
- **action output (DirectFlow-flow mode)**
- **action set (DirectFlow-flow mode)**
- **match (DirectFlow-flow mode)**

match (DirectFlow-flow mode)

The **match** command allows you to configure a rule or a flow which could match on L2, L3, L4 fields of a packet and specify a certain action to modify, drop or redirect the packet.

All traffic ingressing on the switch will be matched against the flows installed. In cases where none of the packets match, normal switching or routing behavior will take over. When multiple entries match a packet, precedence is given to the entry that was installed first.

The **no match** and **default match** commands remove the **match** statement from the configuration mode.

Command Mode

Directflow-flow Configuration

Command Syntax

```
match CONDITION
no match CONDITION
default match CONDITION
```

Parameters

- ***CONDITION*** specifies criteria for evaluating a route. Options include:
  - **cos** <***0*** to ***7***> cost of service.
  - **destination ip** *ipv4_sub* destination IPv4 subnet. L3 fields valid only if ethertype is IP (0x0800).
  - **destination mac** *mac_addr* Add to the existing community. Dotted hex notation.
  - **destination mac** *mac_addr* **mask** *mac_mask* Add to the sting community. Dotted hex notation.
  - **destination port** <***0*** to ***65535***> Fields accepted only if protocol is TCP| UDP
  - **ethertype** <***0*** to ***65535***> Layer 4 destination port.
  - **ethertype ARP** Layer 4 destination port.
  - **ethertype IP** Layer 4 destination port.
  - **icmp code** <***0*** to ***255***> Fields accepted only if protocol is ICMP
  - **icmp type** <***0*** to ***255***> Fields accepted only if protocol is ICMP
  - **input interface ethernet** *e_num* Ethernet interface specified by *e_num*.
  - **input interface port-channel** *p_num* Port channel interface specified by *p_num*.
  - **ip protocol** <***0*** to ***255***> Type of service.
  - **ip protocol icmp** L3 fields valid only if ethertype is IP (0x0800).
  - **ip protocol tcp** L3 fields valid only if ethertype is IP (0x0800).
  - **ip protocol udp** L3 fields valid only if ethertype is IP (0x0800).
  - **ip tos** <***0*** to ***255***> L3 fields valid only if ethertype is IP (0x0800).
  - **source ip** *ipv4_subnet* L3 fields valid only if ethertype is IP (0x0800).
  - **source mac** *mac_addr* Add to the existing community. Dotted hex notation.
  - **source mac** *mac_addr* **mask** *mac_mask* Add to the sting community. Dotted hex notation.
  - **source port** <***0*** to ***65535***> Fields accepted only if protocol is TCP| UDP
  - **tcp flag ack** Layer 4 destination port.
  - **tcp flag fin** Layer 4 destination port.

- **tcp flag psh** Layer 4 destination port.
- **tcp flag rst** Layer 4 destination port.
- **tcp flag syn** Layer 4 destination port.
- **tcp flag urg** Layer 4 destination port.
- **tcp flag urg** Layer 4 destination port
- **vlan <*1* to *4094*> mask <*1* to *4095*>** Number of VLAN.

  The **no match** and **default match** commands require only the ***CONDITION*** type without a specific condition value.

Example

- This command creates the rules to match on Ethertype IP and Source IP 10.10.10.10.

```
switch(config-directflow)# flow Test1
switch(config-directflow-Test1)#persistent
switch(config-directflow-Test1)#match ethertype ip
switch(config-directflow-Test1)#match source ip 10.10.10.10
```

priority (DirectFlow-flow mode)

The **priority** command sets the priority for the flow match rules. Each flow-table entry has an optional priority field, with a higher number indicating a higher priority. Flows with the same priority may be loaded in any order, and the order may be changed at any time. If multiple entries match a packet, precedence is given to the entry that was installed first.

Priority numbers range from 0 to 65535. The default is 0. The higher priority rules match first.

The **no priority** and **default priority** commands remove **priority** statement from the DirectFlow configuration mode.

Command Mode

Directflow-flow Configuration

Command Syntax

```
priority priority_value
no priority
default priority
```

Parameters

- *priority_level* priority xxx. Value ranges from 0 to 65535. Default is 0.

Example

- These commands assign the priority of 150 to flow Test-1.

```
switch(config-directflow-Test-1)#priority 150
switch(config-directflow-Test-1)#
```

show directflow

The **show directflow** command shows the effective DirectFlow configuration parameters.

Command Mode

EXEC

Command Syntax

```
show directflow
```

Example

- This command displays the actual hardware state of DirectFlow.

```
switch# show directflow
DirectFlow configuration: Enabled
Total matched: 23 packets
switch#
```

show directflow flows

The **show directflow flows** command displays the contents of the flow table, showing each entry with its match rules, actions, and packet counters.

Command Mode

EXEC

Command Syntax

```
show directflow flows
```

Example

- This command displays the contents of the flow table.

```
switch# show directflow flows
Flow Test-1:
  priority: 0
  match:
    VLAN ID: 0xa/0x1
    Ethernet type: IPv4
    source IPv4 address: 10.10.10.1
  actions:
    set destination Ethernet address to: 00:00:aa:aa:bb:bb
    output interfaces: Port-Channel100
  matched: 0 packets, 0 bytes
switch#
```

shutdown (DirectFlow)

The **shutdown** command, in DirectFlow mode, disables DirectFlow on the switch. DirectFlow is disabled by default.

The **no shutdown** command re-enables DirectFlow.

Command Mode

Directflow Configuration

Command Syntax

```
shutdown
no shutdown
default shutdown
```

Example

- These commands enable DirectFlow on the switch.

```
switch(config)#directflow
switch(config-directflow)#no shutdown
switch(config-directflow)#
```

- This command disables DirectFlow Flow.

```
switch(config-directflow-Test1)#shutdown
```

timeout (DirectFlow-flow mode)

The **timeout** command, in DirectFlow mode, command configures the connection timeout period for connection sessions. The connection timeout period defines the interval between a user's most recently entered command and an automatic connection shutdown. Automatic connection timeout is disabled by setting the idle-timeout to zero, which is the default setting.

Command Mode

Directflow-flow Configuration

Command Syntax

```
no priority
no timeout hard
no timeout idle
```

Parameters

- *idle* session idle timeout length.
  - *0* Automatic connection timeout is disabled
  - *<1-4294967295>* Automatic timeout period (seconds).
- *hard* session hard timeout length.
  - *0* Automatic connection timeout is disabled.
  - *<1-4294967295>* Automatic timeout period (seconds).

Example

- These commands enable a hard timeout period of 5 seconds on the switch.

```
switch(config)#directflow
switch(config-directflow-Test1)#timeout hard 5
switch(config-directflow-Test1)#
```

- These commands enable DirectFlow on the switch.

```
switch(config)#directflow
switch(config-directflow-Test1)#no timeout hard
switch(config-directflow-Test1)#
```