ARISTA WHITE PAPER

Arista 7500R Universal Spine Platform Architecture
('A day in the life of a packet')

Arista Networks' award-winning Arista 7500 Series was introduced in April 2010 as a revolutionary switching platform, which maximized data center performance, efficiency and overall network reliability. It raised the bar for switching performance, being five times faster, one-tenth the power draw and one-half the footprint compared to other modular data center switches.
In 2013, the Arista 7500E Series delivered a three-fold increase in density and performance, with no sacrifices on features and functionality and with complete investment protection. Just three years later and the Arista 7500R Universal Spine platform deliver more than 3.8X increase in performance and density with significant increases in features and functionality, most notable forwarding table sizes that put it in the class of router platforms.
This white paper provides an overview of the switch architecture of the Arista 7500R Universal Spine platform line card and fabric modules, the characteristics of packet forwarding and how the packet processing has evolved to the point where the Arista 7500 Series is a Router.

The Arista 7500R Universal Spine Platform is a family of modular switches available in 4-slot, 8-slot and 12-slot form factors that support a range of line card options.

At a system level, the Arista 7512R with 115 Tbps fabric scales up to 432 x 100GbE or 40GbE ports, 1,728 x 10GbE or 25GbE ports, or 864 x 50GbE in 18 RU providing industry-leading performance and density without compromising on features/functionality or investment protection.

Figure 1: Arista 7500R Universal Spine platform.

Table 1: Arista 7500R Key Port and Forwarding Metrics

Characteristic	Arista 7504R	Arista 7508R	Arista 7512R
Chassis Height (RU)	7 RU	13 RU	18 RU
Line card Module slots	4	8	12
Supervisor Module Slots	2	2	2
10GbE/25GbE Maximum Density	576	1,152	1,728
40GbE/100GbE Maximum Density	144	288	432
System Usable Capacity (Tbps)	37.5 Tbps	75 Tbps	115 Tbps
Maximum forwarding throughput per Line card (Tbps)	3.6 Tbps (DCS-7500R-36CQ-LC) (36 x 100G / 144 x 25G per LC)		
Maximum forwarding throughput per System (Tbps)	28.8 Tbps	57.6 Tbps	86.4 Tbps
Maximum packet forwarding rate per Line card (pps)	4.32Billion pps (36CQ)		
Maximum packet forwarding rate per System (pps)	17.2 Bpps	34.5 Bpps	51.8 Bpps
Maximum Buffer memory / System	96 GB	192 GB	288 GB
Virtual Output Queues / System	More than 2.2 million		

ARISTA

TRUE INVESTMENT PROTECTION

The Arista 7500 Series was first introduced in 2010 providing industry-leading density and performance at a fraction the power consumption compared to other switches. In 2013 the second-generation Arista 7500E delivered a three-fold increase in density and performance with complete investment protection with existing Arista 7500 Series line cards and fabric modules.

With the introduction of the Arista 7500R Universal Spine platform – the third-generation iteration of the Arista 7500 Series – Arista has continued to provide investment protection to existing Arista 7500 Series deployments. Existing chassis, power supplies, fans, 7500E line cards and fabric modules can continue to be used alongside newer 7500R line cards.

For new deployments, backwards interoperability with existing components may appear less important. However what it represents from a feature/functionality perspective is a platform that is feature-rich with proven high quality day one, leveraging thousands of person years of software development on the single OS and platform, enabling much faster qualification.

ARISTA 7500R – ROUTER TABLE SCALE AND FEATURES/FUNCTIONALITY

Historically, network engineers would live by the adage *"switch where you can, route where you must."* Long established as a best practice, it was a reference to switch ports being substantially more cost effective, higher performance and with lower power consumption compared to router ports.

More recent history introduced switches that could provide layer 3 switching – IP routing – but with a distinction that they were still a 'switch' because they lacked many features and functionality, table sizes, buffering or programmability that routers could provide.

Iterations of the packet processor silicon in the Arista 7500 Series – from first-generation in 2010 (Petra / 7500) to second-generation in 2013 (Arad / 7500E) to third-generation in 2016 (Jericho) have essentially been riding Moore's Law. The observation that on average there will be two times more transistors available every two years has held true for decades, and the network silicon within the Arista 7500 Series has used those additional transistors to more than double its performance and density, from 80Gbps per packet processor (8x10G) to 240Gbps (6x40G) to up to 960Gbps.

In addition to more ports and higher performance, forwarding table sizes have continued to increase. Arista's innovative FlexRoute™ Engine enables more than a million IPv4 and IPv6 route prefixes in hardware, beyond what the merchant silicon enables natively and the Arista EOS NetDB™ evolution of SysDB enables increased scale and performance and industry-leading routing convergence, enabling this to be the first switch system that has evolved to truly be called a router.

Internet Edge/Peering Router requirements:

- Large forwarding tables – able to handle all paths of the Internet, with headroom for future growth of both IPv4 and IPv6
- Multi-protocol data-plane forwarding: layer 2, IPv4, IPv6, MPLS underlay with large numbers of tunnels and overlays including GRE and VXLAN
- Control-plane scale to support millions of prefixes and hundreds of peers
- Dynamic Large buffers for in-cast and speed changes with Virtual Output Queuing for fairness
- Flexible packet parser and flexible rewrite engine capable of handling future overlay and tunneling protocols

CHASSIS AND MID-PLANE

All Arista 7500R chassis (4-slot, 8-slot, 12-slot) share a common system architecture with identical fabric bandwidth and forwarding capacity per slot. Line cards, Supervisors and power supplies are common across all systems; the only differences are in the size of the fabric/fan modules, number of line card slots and power supplies. Airflow is always front-to-rear and all data cabling is at the front of the chassis.

Chassis design and layout is a key aspect that enables such high performance per slot: the fabric modules are directly behind line card modules and oriented orthogonal to the line card modules. This design alleviates the requirement to route high speed signal traces on the mid plane of the chassis, reducing the signal trace lengths and enabling more high speed signals to operate at faster speeds by being shorter lengths. This characteristic has enabled Arista to scale the system from 10 Tbps with first generation modules in 2010 up to 115 Tbps today with headroom for even higher performance and density in future.

SUPERVISOR 2 MODULES

Supervisor modules on the Arista 7500R Universal Spine platform are used for control-plane and management-plane functions only. There are two redundant supervisors in the chassis, each capable of managing the system. All data-plane forwarding is performed on line card modules and forwarding between line card modules is always via the fabric modules.

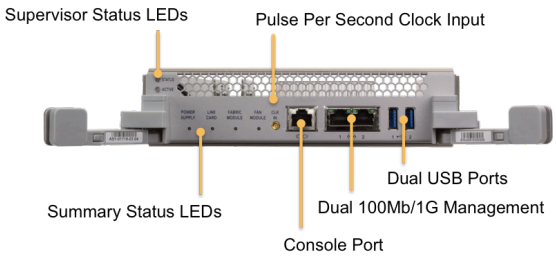


Figure 2: Arista 7500 Series Supervisor 2 Module.

The Arista 7500 Series Supervisor 2 provides a 6-core hyper-threaded Intel Xeon 'Broadwell' CPU with turbo boost frequency to 2.7 GHz and 32GB RAM. Arista's Extensible Operating System (EOS®) makes full use of multiple cores due to its unique multi-process state sharing architecture that separates state information and packet forwarding from protocol processing and application logic. The multi-core CPU and large memory configuration provides headroom for running 3rd party software within the same Linux instance as EOS, within a guest virtual machine or within containers. An optional enterprise-grade SSD provides additional flash storage for logs, VM images or third party packages.

Out-of-band management is available via a serial console port and/or dual 10/100/1000 Ethernet interfaces. There are two USB2.0 interfaces that can be used for transferring images/logs or many other uses. A pulse-per-second clock input is provided for accurate clock synchronization.

There is more than 40 Gbps of inband connectivity from data-plane to control-plane and more than 30 Gbps connectivity between redundant Supervisor modules. Combined, these enable very high performance connectivity for the control-plane to manage and monitor the data-plane as well as replicate state between redundant Supervisors.

DISTRIBUTED PACKET FORWARDING IN THE ARISTA 7500R UNIVERSAL SPINE

DISTRIBUTED DATA-PLANE FORWARDING

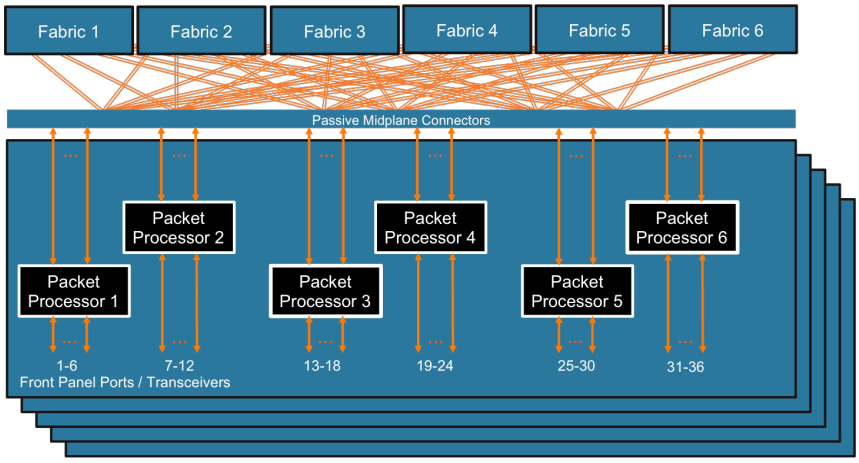


Figure 3: Distributed Forwarding within an Arista 7500R Series

Arista 7500R Universal Spine platform line card modules utilize packet processors to provide distributed data-plane forwarding. Forwarding between ports on the same packet processor utilizes local switching and no fabric bandwidth is used. Forwarding across different packet processors uses all fabric modules in a fully active/active mode. There is always Virtual Output Queuing (VoQ) between input and output, for both locally switched and non-locally switched packets, ensuring there is always fairness even where some traffic is local.

FABRIC MODULES

Within the Arista 7500R up to six fabric modules are utilized in an active/active mode. Each fabric module provides up to 800 Gbps fabric bandwidth full duplex (800 Gbps receive + 800 Gbps transmit) to each line card slot, and with six active fabric modules there is 4.8 Tbps (4.8 Tbps receive + 4.8 Tbps transmit) available. This is more than sufficient capacity to provide N+1 redundancy with the highest-density 36x100G line card module (3.6 Tbps).

Figure 4: Arista DCS-7500R Series Fabric/Fan modules

Packets are transmitted across fabric modules as variable sized cells of up to 256 bytes. Serialization latency of larger frames is amortized via the parallel cell spraying that utilizes all available paths in an active/active manner, preventing hot spots or blocking that can occur with packet-based fabrics.

Besides data-plane packets, the fabric modules are also used for a number of other functions:

- **Virtual Output Queuing (VoQ):** a distributed scheduling mechanism is used within the switch to ensure fairness for traffic flows contending for access to a congested output port. A credit request/grant loop is utilized and packets are queued in physical buffers on ingress packet processors within VoQs until the egress packet scheduler issues a credit grant for a given input packet.
- **Hardware-based distributed MAC learning and updates:** when a new MAC address is learnt, moves or is aged out, the ingress packet processor with ownership of the MAC address will update other packet processors of the update.
- **Data-plane health tracer:** all packet processors within the system send continuous reachability messages to all other packet processors, validating all data-plane connectivity paths within the system.

ARISTA 7500R UNIVERSAL SPINE PLATFORM LINE CARD ARCHITECTURE

All stages associated with packet forwarding are performed in integrated system on chip (SoC) packet processors. A packet processor provides both the ingress and egress packet forwarding pipeline stages for packets that arrive or are destined to ports serviced by that packet processor. Each packet processor can perform local switching for traffic between ports on the same packet processor.

The architecture of a line card (in this case, the 36 port QSFP100 module DCS-7500R-36CQ-LC) is shown below in Figure 5. Each of the six packet processors on the line card services a group of 6 x 100G QSFP100 front panel ports, and is highlighted in a different color.

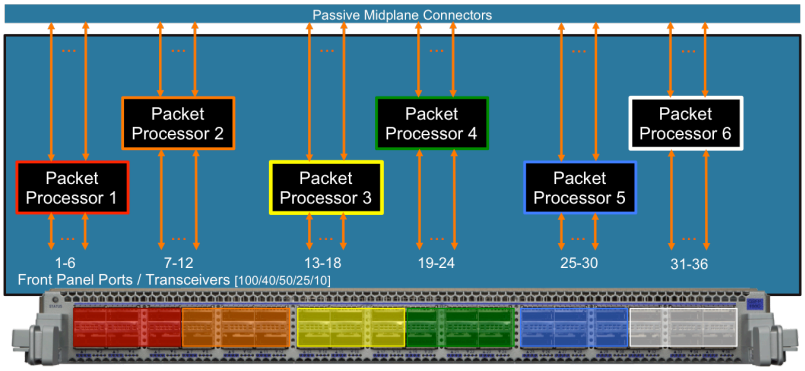


Figure 5: Arista DCS-7500R-36CQ-LC module architecture

In the case of the DCS-7500R-36CQ-LC, each of the 36 ports can operate as either 1 x 100G or breakout to 2 x 50G or 4 x 25G when populated with a QSFP100 transceiver. If populated with a QSFP+ transceiver, each port can operate as either 1 x 40G or breakout to 4 x 10G. All port speed configuration changes are hitless and no ports are disabled or unavailable.

In order to operate a port in breakout mode (e.g. run a 100G port as 4 x 25G or 2 x 50G, or a 40G port as 4 x 10G) the underlying transceiver must enable this. In the case of 25G/50G, SR4 and PSM4 optics support breakout (as they are optically 4x25G in parallel), likewise Direct Attach Copper (DAC) QSFP100 to 4xSFP25 cables provide breakout support. The same holds true for 4x10G breakout of a 40G port, which is available for transceivers and cables that are based on 4x10G parallel signaling.

ARISTA 7500R UNIVERSAL SPINE PLATFORM LINE CARD LAYOUT

Arista 7500R line card modules utilize the same packet processor, with the number of packet processors varied based on the number and type of ports on the module. The packet forwarding architecture of each of these modules is essentially the same: a group of front-panel ports (different transceiver/port/speed options) are connected to a packet processor with connections to the fabric modules.

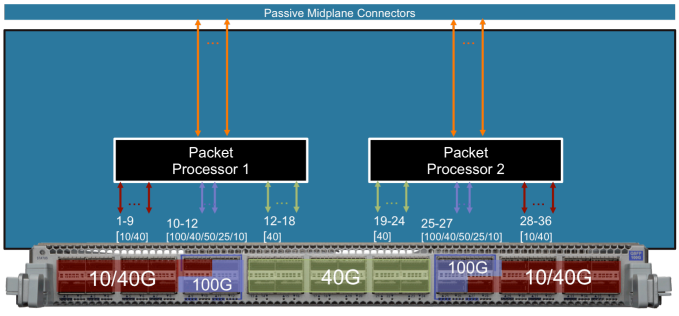


Figure 6: Arista DCS-7500R-36Q-LC (36x40G)

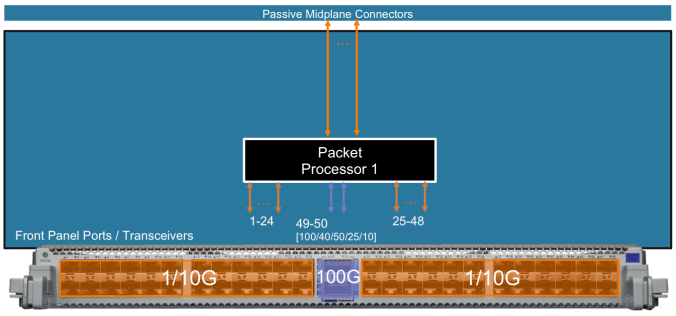


Figure 7: Arista DCS-7500R-48S2CQ-LC (48x10G+2x100G)

As shown in Figure 6, the DCS-7500R-36Q-LC provides up to 36 x 40G ports via QSFP+ transceivers. In addition, 6 of the ports can operate at 100G also (ports 10-12 and 25-27). This line card can be considered to be a 30x40G + 6x40G/100G line card, though its primary function is for 40G connectivity. Up to 24 of the ports can be used in breakout mode for 4x10G (1-12, 25-36).

Figure 7 shows the architecture of the DCS-7500R-48S2CQ-LC module, which provides 48 x 10G/1G ports via SFP+/SFP and 2 x 100G ports. A single packet processor is used for all ports on this line card and the 100G ports support either QSFP100 (100G) or QSFP+ (40G). These ports also support breakout to 2x50G/4x25G/4x10G.

To distinguish between ports that are 100G capable (QSFP100) and those that aren't (QSFP+ only), there is a silkscreen highlight around those ports. Any QSFP+ 40G ports that don't support breakout to 4x10G have a single status LED for those ports.

Also all existing Arista 7500E line card modules are supported alongside Arista 7500R line card modules, providing flexibility of transceiver and optics types while maximizing system port density and investment protection. Table 2 below lists all the line card options and the port speed types available.

Table 2: Arista 7500R Series and Arista 7500E Series Line card Module Port Characteristics

Line card	Port (type)	Interfaces					Port Buffer	Forwarding Rate	Switching Capacity
		10GbE	25GbE	50GbE	40GbE	100GbE			
7500R-36CQ	36 QSFP100	144	144	72	36	36	24GB	4.32Bpps	7.2Tbps
7500R-36Q	36 QSFP+	96	24	12	36	6	8GB	1.44Bpps	3.6Tbps
7500R-48S2CQ	48 SFP+ and 2 QSFP100	56	8	4	2	2	4GB	720Mpps	1.36Tbps
7500E-6CFPX	6 ACO CFP2 (DWDM)	-	-	-	-	6 (tunable λ)	9GB	900Mpps	1.2Tbps
7500E-12CQ	12 QSFP100	48	-	-	12	12	18GB	1.8Mpps	2.4Tbps
7500E-12CM	12 MXP	144	-	-	36	12	18GB	1.8Bpps	2.88Tbps
7500E-36Q	36 QSFP+	144	-	-	36	-	18GB	1.8Bpps	2.88Tbps
7500E-6C2	6 CFP2	60	-	-	12	6	9GB	900Mpps	2.4Tbps
7500E-72S	48 SFP+ and 2 MXP	72	-	-	6	2	9GB	900Mpps	1.44Tbps
7500E-48S	48 SFP+	48	-	-	-	-	9GB	720Mpps	960Gbps
7500E-48T	48 10GBASE-T	48	-	-	-	-	6GB	720Mpps	960Gbps

ARISTA 7500R UNIVERSAL SPINE PLATFORM PACKET FORWARDING PIPELINE

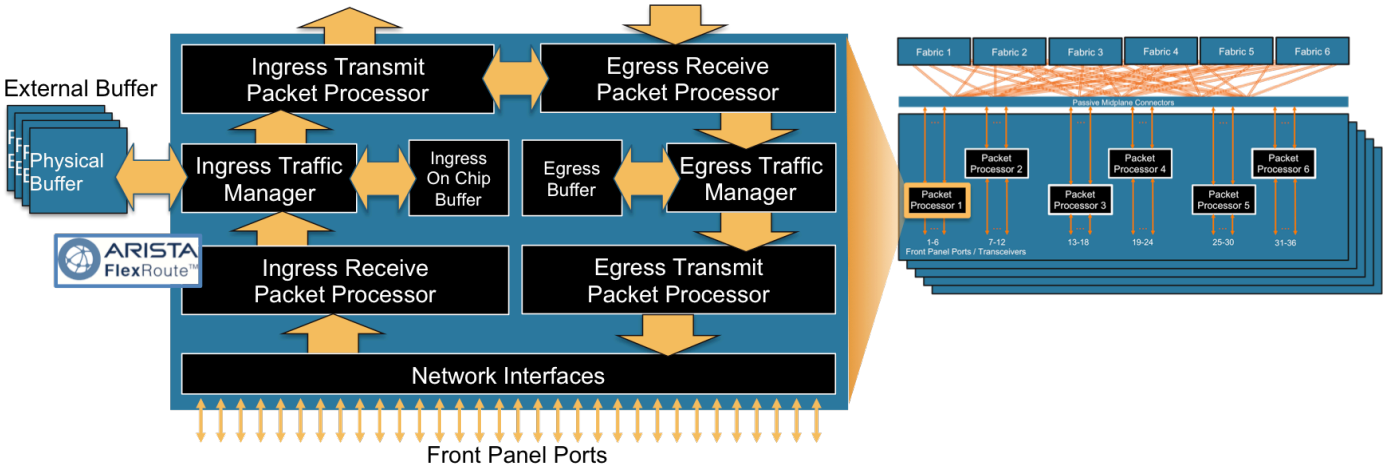


Figure 8: Packet forwarding pipeline stages inside a packet processor on an Arista 7500R line card module

Each packet processor on a line card is a System on Chip (SoC) that provides all the ingress and egress forwarding pipeline stages for packets to or from the front panel input ports connected to that packet processor. Forwarding is always hardware-based and never falls back to software for forwarding.

The steps involved at each of the logical stages of the packet forwarding pipeline are outlined below.

STAGE 1: NETWORK INTERFACE (INGRESS)

When packets/frames enter the switch, the first block they arrive at is the Network Interface stage. This is responsible for implementing the Physical Layer (PHY) interface and Ethernet Media Access Control (MAC) layer on the switch and any Forward Error Correction (FEC).

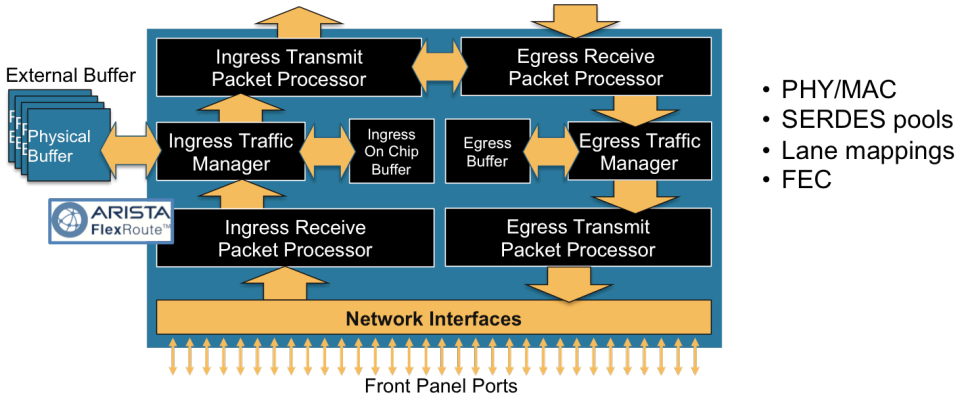


Figure 9: Packet Processor stage 1 (ingress): Network Interface

The PHY layer is responsible for transmission and reception of bit streams across physical connections including encoding, multiplexing, synchronization, clock recovery and serialization of the data on the wire for whatever speed/type Ethernet interface is configured.

Programmable lane mapping are used to map the physical lanes to logical ports based on the interface type and configuration. For example, lane mapping are used on an Arista MXP ports to map physical lanes to 1x100G, 3x40G, 12x10G or combinations thereof based on user configuration. Lane mapping are also used for breakout of 4x25G and 2x50G on 100G ports.

If a valid bit stream is received at the PHY then the data is sent to the MAC layer. On input, the MAC layer is responsible for turning the bit stream into frames/packets: checking for errors (FCS, Inter-frame gap, detect frame preamble) and find the start of frame and end of frame delimiters.

STAGE 2: INGRESS RECIEVE PACKET PROCESSOR

The Ingress Receive Packet Processor stage is responsible for forwarding decision. It is the stage where all forwarding lookups are performed.

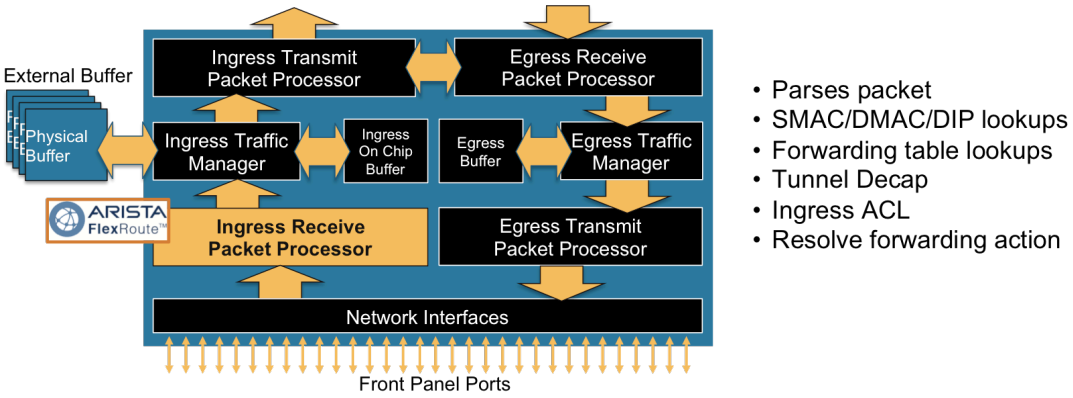


Figure 10: Packet Processor stage 2 (ingress): Ingress Receive Packet Processor

Before any forwarding can take place, first packet or frame headers must be parsed and fields for forwarding decisions extracted. Key fields include L2 Source and Destination MAC addresses [SMAC, DMAC], VLAN headers, Source and Destination IP Addresses [SIP, DIP], class of service (COS), DSCP and so on. The Arista 7500R packet parser supports many tunnel formats (MPLS, IPinIP, GRE, VXLAN, MPLSoGRE) including parsing Ethernet and IP headers under a multi-label stack. The parser is flexible and extensible such that it can support future protocols and new methods of forwarding.

Following parsing, the DMAC is evaluated to see if it matches the device's MAC address for the physical or logical interface. If it's a tunneled packet and is destined to a tunnel endpoint on the device, it is decapsulated within its appropriate virtual routing instance and packet processing continues on the inner packet/frame headers. If it's a candidate for L3 processing (DMAC matches the device's relevant physical or logical MAC address) then the forwarding pipeline continues down the layer 3 (routing) pipeline, otherwise forwarding continues on the layer 2 (bridging) pipeline.

In the layer 2 (bridging) case, the packet processor performs SMAC and DMAC lookup in the MAC table for the VLAN. SMAC lookup is used to learn (and can trigger a hardware MAC-learn or MAC-move update), DMAC (if present) is used for L2 forwarding and if not present will result in the frame being flooded to all ports within the VLAN, subject to storm-control thresholds for the port.

In the layer 3 (routing) case, the packet processor performs a lookup on the Destination IP address (DIP) within the VRF and if there is a match it knows what port to send the frame to and what packet processor it needs to send the frame through to. If the DIP matches a subnet local on this switch but there is no host route entry, the switch will initiate an ARP request to learn the MAC address for where to send the packet. If there is no matching entry at all the packet is dropped. IP TTL decrement also occurs as part of this stage. VXLAN Routing can be performed within a single pass through this stage.

For unicast traffic, the end result from a forwarding lookup match is a pointer to a Forwarding Equivalence Class (FEC) or FEC group (Link Aggregation, Equal Cost Multipathing [ECMP] or Unequal Cost Multipathing [UCMP]). In the case of a FEC group, whatever fields are configured L2/L3/L4 load balancing are used derive a single

matching entry. The final matching adjacency entry provides details on where to send the packet (egress packet processor, output interface and a pointer to the output encapsulation/MAC rewrite on the egress packet processor).

For multicast traffic, the logic is similar except that the adjacency entry provides a Multicast ID, which indicates expansion for both local (ingress) multicast destinations on local ports, whether there are other packet processors that require a copy and if so, what packet processors they are (via multicast expansion in the fabric modules). By default, Arista 7500R Series operates in egress multicast expansion but can be configured for ingress multicast expansion too.

The forwarding pipeline always remains in the hardware data-plane. There are no features that can be enabled that cause the packet forwarding to drop out of the silicon (hardware) data-plane forwarding path. In cases where software assistance is required (e.g. traffic destined within a L3 subnet but for which the switch has not yet seen the end device provide an ARP and doesn't have the L3-to-L2 glue entry), hardware rate limiters and Control-plane Policing are employed to protect the control-plane from potential denial of service attacks.

In parallel with forwarding table lookups there are also Ingress ACL lookups (Port ACLs, Routed ACLs) for applying security and QoS lookups to apply Quality of Service. All lookups are ultimately resolved using strength-based resolution (some actions are complementary and multiple actions are applied, some actions override others) but ultimately the outcome of this stage is a resolved forwarding action.

Flexible Counters within this stage provide accounting and statistics on ACLs, VLAN and sub-interfaces, as well as next hop groups. Counters are updated sub-second via a scalable batch update and is available as telemetry that can be streamed in real-time.

ARISTA FLEXROUTE™ ENGINE

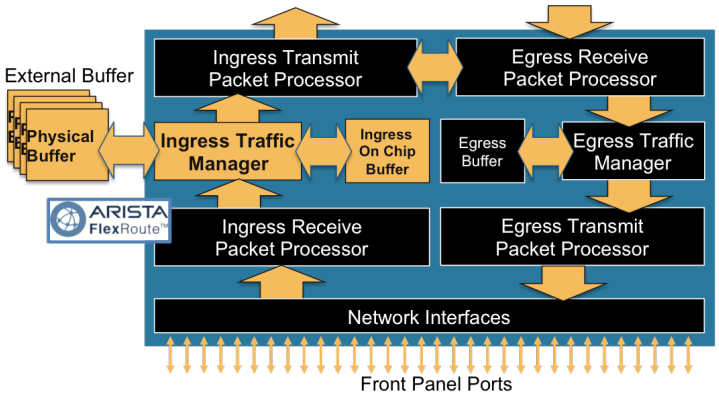
One of the key characteristics of the Arista 7500R Universal Spine platform is the FlexRoute™ Engine, an Arista innovation which enables more than a million IPv4 and IPv6 route prefixes in hardware. This enables deployments in Internet edge/peering use-cases where historically traditional edge routers would have been required. Arista FlexRoute enables large L3 routing tables with significant power consumption savings over existing ways that IP routing longest prefix match lookups are performed. This in turn enables higher port densities and performance with power and cooling advantages due to more efficient transistor count and activity factor reduction compared to alternatives.

Arista's FlexRoute Engine is used for both IPv4 and IPv6 Longest Prefix Match (LPM) lookups without partitioning table resources. It is optimized around the Internet routing table, its prefix distribution and projected growth. FlexRoute enables scale beyond 1 million IPv4 and IPv6 prefixes combined, providing headroom for internet table growth for many years.

In addition to large table support, FlexRoute enables very fast route programming and reprogramming (tens of thousands of prefixes per second), and does so in a manner that is non-disruptive to other prefixes while forwarding table updates are taking place.

STAGE 3: INGRESS TRAFFIC MANAGER

The Ingress Traffic Manager stage is responsible for packet queuing and scheduling.



- Virtual Output Queuing (VoQ) subsystem
- Credit request
- On-chip buffer for uncongested output queues
- External buffer for queuing
- Shaping/Queuing
- PFC

Figure 11: Packet Processor stage 3 (ingress): Ingress Traffic Manager

Arista 7500R Universal Spine platforms utilize Virtual Output Queuing (VoQ) where the majority of the buffering within the switch is on the input line card. While the physical buffer is on the input, it represents packets queued on the output side (hence why its called *virtual* output queuing). VoQ is a technique that allows buffers to be balanced across sources contending for a congested output port and ensures fairness and QoS policies can be implemented in a distributed forwarding system.

When a packet arrives into the Ingress Traffic Manager, a VoQ credit request is forwarded to the egress port processor requesting a transmission slot on the output port. Packets are queued on ingress until such time as a VoQ grant message is returned (from the Egress Traffic Manager on the output port) indicating that the Ingress Traffic Manager can forward the frame to the egress packet processor.

While the VoQ request/grant credit loop is under way, the packet is queued in input buffers. A combination of onboard memory (16MB) and external memory (4GB) per packet processor is used to store packets while waiting their VoQ grant. The memory is used such that traffic destined to uncongested outputs (egress VoQ is empty) will go into onboard memory (head of the queue) otherwise external buffer memory is utilized. External buffer memory is used because it's not feasible to build sufficiently large buffers "on-chip" due to the number of transistors and area that would subsequently consume.

While there is up to 288GB buffer memory per system, the majority of the buffer is allocated in a dynamic manner wherever it is required across potentially millions of VoQs per system:

- ~30% buffer reserved for traffic per Traffic Class per Output Port
- ~15% buffer for multi-destination traffic
- ~55% available as a dynamic buffer pool

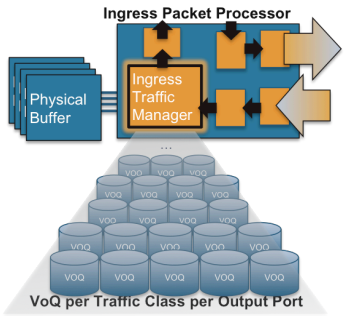


Figure 12: Physical Buffer on Ingress allocated as Virtual Output Queues

The dynamic pool enables the majority of the buffer to be used in an intelligent manner based on real-time contention and congestion on output ports. While there is potentially hundreds of gigabytes of buffer memory, individual VoQ limits are applied such that a single VoQ doesn't result in excess latency or queuing on a given output port. The default allocations (configurable) are as per in Table 4:

Table 4: Default per-VoQ Output Port Limits

Output Port Characteristics	Maximum Packet Queue Depth	Maximum Packet Buffer Depth (MB)	Maximum Packet Buffer Depth (msec)
VoQ for a 100Mbps output port	5,000 packets	1.25 MB	12.5 msec
VoQ for a 1G output port	12,500 packets	12.5 MB	12.5 msec
VoQ for a 10G output port	50,000 packets	50 MB	5 msec
VoQ for a 25G output port	125,000 packets	125 MB	5 msec
VoQ for a 40G output port	200,000 packets	200 MB	5 msec
VoQ for a 50G output port	250,000 packets	250 MB	5 msec
VoQ for a 100G output port	500,000 packets	500 MB	5 msec

Individual queues are configurable with queue depths between 0 and 1.57M packets and 0 bytes to 2.1 gigabytes.

The VoQ subsystem enables buffers that are dynamic, intelligent and deep such that there is always buffer space available for new flows, even under congestion and heavy load scenarios. There is always complete fairness in the system, with QoS policy always enforced in a distributed forwarding system. This enables any application workload to be deployed – existing or future – and provides the basis for deployment in Content Delivery Neworks (CDNs), service providers, internet edge, converged storage, hyper-converged systems, big data/analytics, enterprise and cloud providers. The VoQ subsystem enables maximum fairness and *goodput* for applications with *any* traffic profile, be it any-cast, in-cast, mice or elephant flows, or anything in between.

STAGE 4: INGRESS TRANSMIT PACKET PROCESSOR

The Ingress Transmit Packet Processor stage is responsible for transferring frames from the input packet processor to the relevant output packet processor. Frames arrive here once the output port has signaled via a VoQ grant message that it is the allocated slot for a given input.

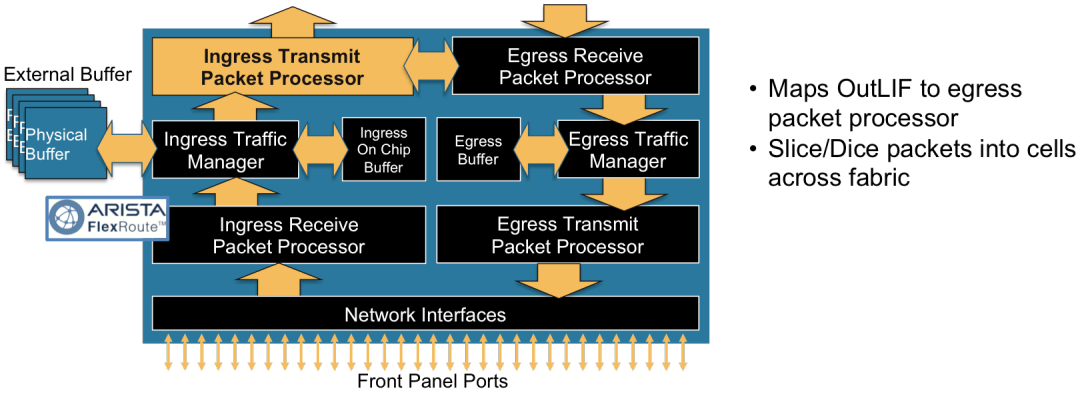


Figure 13: Packet Processor stage 4 (ingress): Ingress Transmit Packet Processor

All available fabric paths are used in parallel to transfer the frame or packet to the output packet processor, with the original packet sliced into variable-sized cells which are forwarded across up to 36 fabric links simultaneously. This mechanism reduces serialization to at most 256 bytes at 25Gbps and ensures there are no hot spots as every flow is always evenly balanced across all fabric paths. Since a packet is only transferred across the fabric once there is a VoQ grant, there is no queuing within the fabric and there are guaranteed resources to be able to process the frame on the egress packet processor.

Each cell has a header added to the front for the receiving packet processor to be able to reassemble and maintain in-order delivery. Forward Error Correction (FEC) is also enabled for traffic across the fabric modules, both to correct errors (if they occur) but also to help monitor data-plane components of the system for any problems.

Packets destined to ports on the same packet processor can be switched locally and bypass using any fabric bandwidth resources, but otherwise aren't processed any differently in terms of the VoQ subsystem.

STAGE 5: FABRIC MODULES

There are 6 fabric modules in the rear of the chassis all operating in an active/active manner. These provide connectivity between all data-plane forwarding packet processors inside the system.

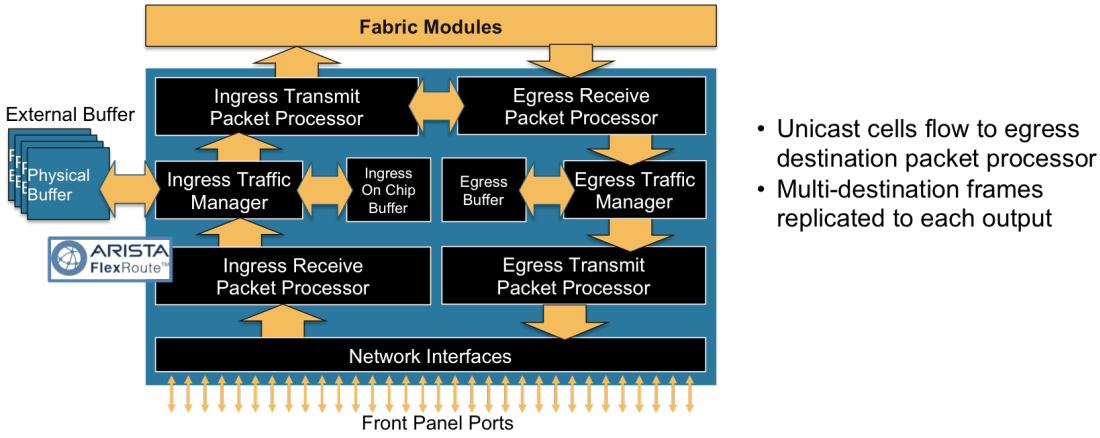


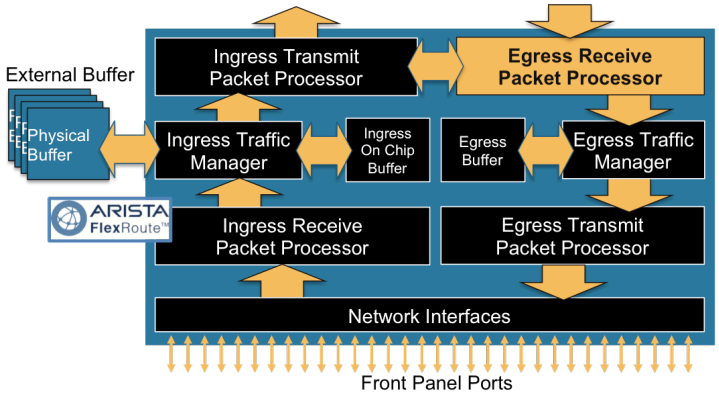
Figure 14: Fabric modules

The fabric modules forward based on cell headers indicating which of the 72 possible output packet processors to send the cell to.

For multi-destination packets such as multicast or broadcast, there is a lookup into a 64K multicast group table that uses a bitmap to indicate which packet processors should receive replication copies of the cell. Note that if there are multiple multicast receivers on an output packet processor, there is only one copy delivered per output packet processor as there is optimal egress multicast expansion inside the system. Control-plane software maintains the multicast group table based on the fan-out of multicast groups across the system. IP multicast groups that share a common set of output packet processors reuse the same fabric Multicast ID.

STAGE 6: EGRESS RECEIVE PACKET PROCESSOR

The Egress Receive Packet Processor stage is responsible for reassembling cells back into packets/frames. This is also the stage that takes a multicast packet/frame and replicates it into multiple output packets/frames if there are multiple receivers on this output packet processor.



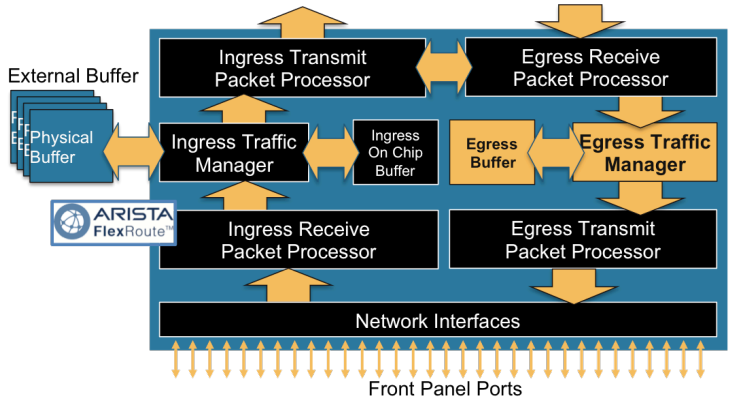
- Reassemble cells back into frames
- Egress multicast expansion

Figure 15: Packet processor stage 6 (egress): Egress Receive Packet Processor

This stage ensures that there is no frame or packet reordering in the system. It also provides the data-plane health tracer, validating reachability messages from all other packet processors across all paths in the system.

STAGE 7: EGRESS TRAFFIC MANAGER

The Egress Traffic Manager stage is responsible for the granting of VoQ credit requests from input packet processors and managing egress queues.



- Manage Egress Queues (unicast & multicast)
- Grant VoQ requests from Ingress
- PFC/ETS traffic scheduling

Figure 16: Packet processor stage 7 (egress): Egress Receive Packet Processor

When an ingress packet processor requests to schedule a packet to the egress packet processor it is the Egress Traffic Manager stage that receives the request. If the output port is not congested then it will grant the request immediately. If there is congestion it will service requests in a fair manner between contending input ports, within the constraints of QoS configuration policy (e.g. output port shaping) while also conforming to PFC/ETS traffic scheduling policies on the output port. Scheduling between multiple contending inputs for the same queue can be configured to weighted fair queuing (WFQ) or round-robin.

The Egress Traffic Manager stage is also responsible for managing egress buffering within the system. There is an additional 6MB on-chip buffer used for egress queuing, which is allocated as 64K packet descriptors. This buffer is mostly reserved for multicast traffic as unicast traffic has a minimal requirement for egress buffering due to the large ingress VoQ buffer and fair adaptive dynamic thresholds are utilized as a pool of buffer for the output ports.

STAGE 8: EGRESS TRANSMIT PACKET PROCESSOR

The Egress Transmit Packet Processor is the last stage of packet processing.

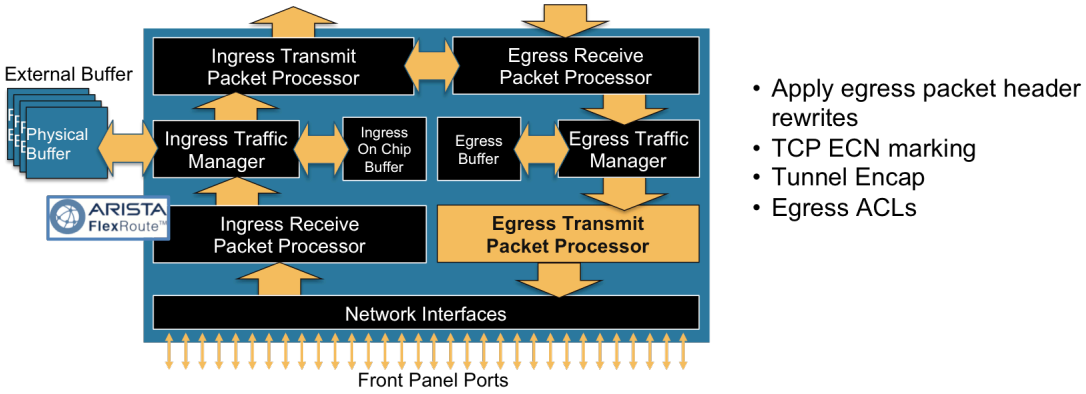


Figure 17: Packet processor stage 8 (egress): Egress Transmit Packet Processor

In this stage, any packet header updates such as updating the next-hop DMAC, dot1q updates and tunnel encapsulation operations are performed based on packet header rewrite instructions passed from the Input Receive Packet Processor. Decoupling the packet forwarding on ingress from the packet rewrite on egress enables larger numbers of next-hops and tunnels as these resources are programmed in a distributed manner.

This stage can also optionally set TCP Explicit Congestion Notification (ECN) bits based on whether there was contention on the output port and the time the packet spent queued within the system from input to output. Flexible Counters are available at this stage and can provide packet and byte counters on a variety of tables.

Egress ACLs are also performed at this stage based on the packet header updates, and once the packet passes all checks, it is transmitted on the output port.

STAGE 9: NETWORK INTERFACE (EGRESS)

Just as packets/frames entering the switch went through the Ethernet MAC and PHY layer with the flexibility of multi-speed interfaces, the same mechanism is used on packet/frame transmission. Packets/frames are transmitted onto the wire as a bit stream in compliance with the IEEE 802.3 standards.

ARISTA EOS: A PLATFORM FOR SCALE, STABILITY AND EXTENSIBILITY

At the core of the Arista 7500R Universal Spine platform is Arista EOS® (Extensible Operating System). Built from the ground up using innovations in core technologies since our founding in 2004, EOS contains more than 8 million lines of code and over 1000 man-years of advanced distributed systems software engineering. EOS is built to be open and standards-based, and its modern architecture delivers better reliability and is uniquely programmable at all system levels.

EOS has been built to address two fundamental issues that exist in cloud networks: the need for non-stop availability and the need for high feature velocity coupled to high quality software. Drawing on our engineers experience in building networking products over more than 30 years, and on the state-of-the-art in open systems technology and distributed systems, Arista started from a clean sheet of paper to build an operating system suitable for the cloud era.

At its foundation, EOS uses a unique multi-process state-sharing architecture where there is separation of state information from packet forwarding and from protocol processing and application logic. In EOS, system state and data is stored and maintained in a highly efficient, centralized System Database (SysDB). The data stored in SysDB is accessed using an automated publish/subscribe/notify model. This architecturally distinct design principle supports self-healing resiliency in our software, easier software maintenance and module independence, higher software quality overall, and faster time-to-market for new features that customers require.

Arista EOS contrasts with the legacy approach to building network operating systems developed in the 1980's that relied upon embedding system state held within each independent process, extensive use of inter-process communications (IPC) mechanisms to maintain state across the system, and manual integration of subsystems without an automated structured core like SysDB. In legacy network operating systems, as dynamic events occur in large networks or in the face of a system process failure and restart, recovery can be difficult if not impossible.

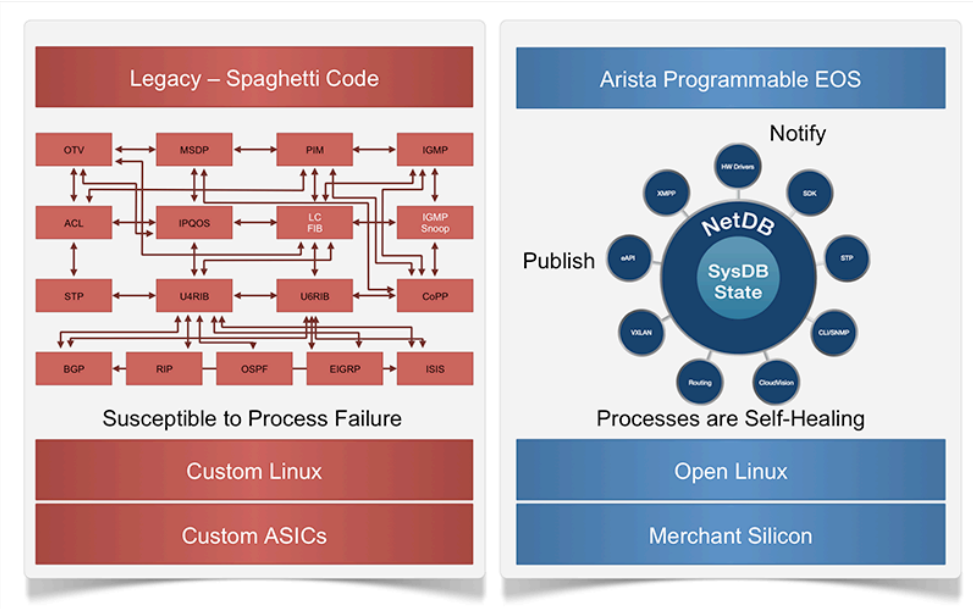


Figure 18: Legacy approaches to network operating systems (left), Arista EOS (right)

Arista took to heart the lessons of the open source world and built EOS on top of an unmodified Linux kernel. We have also maintained full, secured access to the Linux shell and utilities. This allows EOS to preserve the security, feature development and tools of the Linux community on an on-going basis, unlike legacy approaches where the original OS kernel is modified or based on older and less well-maintained versions of Unix. This has made it possible for EOS to natively support things like Docker Containers to simplify the development and deployment of applications on Arista switches. Arista EOS represents a simple but powerful architectural approach that results in a higher quality platform on which Arista is faster to deliver significant new features to customers.

EOS is extensible at every level, with open APIs at every level: management plane, control-plane, data-plane, services-level extensibility, application-level extensibility and with access to all Linux operating system facilities including shell-level access, Arista EOS can be extended with unmodified Linux applications and a growing number of open source management tools to meet the needs of network engineering and operations.

Open APIs such as EOS API (eAPI) and OpenConfig and EOS SDK provide well-documented and widely used programmatic access to configuration, management and monitoring that can stream real-time network telemetry, providing a superior alternative to traditional polling mechanisms.

The NetDB evolution of SysDB extends the core EOS architecture in the following ways:

- NetDB NetTable is the mechanism to hold network state that allows EOS to scale to new limits. It scales the routing stack to hold more than a million routes or tunnels with millisecond convergence. This is critical, as the spine is becoming the new center of the network, the point of transition between the data center and the rest of the world.
- NetDB Network Central enables system state to be streamed and stored as historical data in a central repository such as CloudVision, HBase or other third party systems. This ability to take all of the network state and bring it to one point is crucial for scalable network analysis, debugging, monitoring, forensics and capacity planning. This simplifies workload orchestration and provides a single touch point for third party controllers.
- NetDB Replication enables state streaming to other interested systems in a way that automatically tolerates failures, and adapts the rate of update propagation to match the capability of the receiver to process those updates.

The evolution of SysDB to NetDB builds on the same core principles that have been the foundation of the success of EOS: the openness, the programmability, the quality, and the way that a single build of EOS runs across all of our products.

SYSTEM HEALTH TRACER AND INTEGRITY CHECKS

Just as significant engineering effort has been invested in the software architecture of Arista EOS, the same level of detail has gone into system health and integrity checks within the system. There are numerous subsystems on Arista 7500R Universal Spine platform switches that validate and track the system health and integrity on a continual basis:

- All memories where code executes (control-plane and data-plane) are ECC protected; single bit errors are detected and corrected automatically, double bit errors are detected.
- All data-plane forwarding tables are parity protected with shadow copies kept in ECC protected memory on the control-plane. Continual hardware table validation verifies that the hardware tables are valid and truthful.
- All data-plane packet buffers are protected using CRC32 checking from the time a packet/frame arrives to the time it leaves the switch. The CRC32 is validated at multiple points through the forwarding pipeline to ensure no corruption has happened, or if there has been a problem it can be isolated.
- Forward Error Correction (FEC) is also utilized for traffic across the fabric modules, both to correct errors (if they occur) but also to help monitor data-plane components of the system for problems.
- Data-plane forwarding elements are continually testing and checking reachability with all other forwarding elements

CONCLUSION

Designed for large virtualized and cloud networks the Arista 7500R Series modular switches are the industry's highest performance universal spine switches. They combine 100GbE density with internet scale table sizes and comprehensive L2 and L3 features and proven investment protection.

Santa Clara—Corporate Headquarters
5453 Great America Parkway
Santa Clara, CA 95054
Tel: 408-547-5500
www.arista.com

Ireland—International Headquarters
3130 Atlantic Avenue
Westpark Business Campus
Shannon, Co. Clare
Ireland

Vancouver—R&D Office
9200 Glenlyon Pkwy, Unit 300
Burnaby, British Columbia
Canada V5J 5J8

San Francisco—R&D and Sales Office
1390 Market Street, Suite 800
San Francisco, CA 94102

India—R&D Office
Global Tech Park, Tower A & B, 11th Floor
Marathahalli Outer Ring Road
Devarabeesanahalli Village, Varthur Hobli
Bangalore, India 560103

Singapore—APAC Administrative Office
9 Temasek Boulevard
#29-01, Suntec Tower Two
Singapore 038989

Nashua—R&D Office
10 Tara Boulevard
Nashua, NH 03062