Arista 7800R4 Switch Architecture

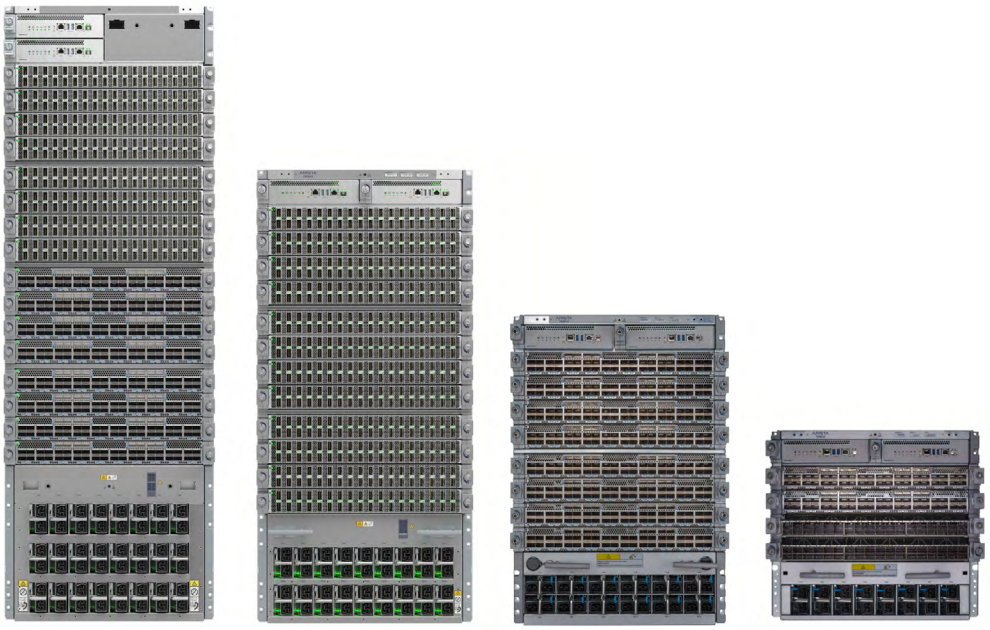


Figure 1.1: 7800R4 platform portfolio

Table of contents

Table of contents

Introduction

The Arista Networks R-series is a purpose-built family of high-performance routing platforms available in both fixed and modular form-factors. The Arista 7800R4 platform is the 7th generation of modular switches within the R-Series family, sharing the same underlying architecture with the R4 fixed configuration 7280R4 and 7020R4 platforms.

The Arista 7800R4 Platform is the next evolution of the R-Series family of modular switches, retaining the consistent architecture of deep buffers, VOQ and non-blocking lossless forwarding of the previous generation. The 7800R4 platform is available in a choice of 4, 8, 12 and 16-slot systems that support line cards providing high density 100G, 400G and 800G with a choice of forwarding table scale and quantum-safe encryption with Arista TunnelSecTM (MACsec, IPsec and VXLANsec) encryption options.

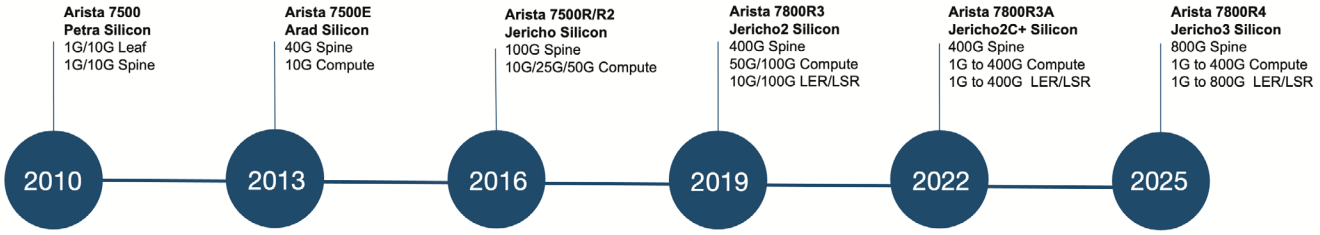


Figure 1.2: Arista R Series modular Platform Generations

As a full featured switching and routing platform, providing a comprehensive feature set, in a variety of dense footprints and interfaces options, the 7800R4 series is the ideal platform for Data Center, AI/ML workloads, rich multimedia, content delivery, 5G Metro Aggregation, and next generation Service Provider edge and core use cases.

This white paper provides an overview of the architecture of the Arista 7800R4 modular platforms, and the differentiational benefits they provide over previous generations.

Arista R4 series: Overview

As the next evolution of Arista's 7800 chassis, the 7800R4 series of modular platforms have been engineered from the ground up for maximum reliability and lowest TCO. Key attributes of the R4 series include:

- Standards-based high-density multi-speed platform with support for a range of interface speeds up to 800G interfaces, providing a universal routing platform high bandwidth use cases, such as AI/ML workloads, Content Delivery Networks (CDNs), Internet Exchanges and next generation 400G/800G Core architectures.
- Hierarchical hybrid deep buffers in a high-radix form-factor make it the ideal platform for workloads where lossless performance and in-cast problems are expected, such as in big data analytics, AI/ML and IP storage.
- Quantum-safe, line rate encryption with Arista TunnelSecTM which delivers MACsec and IPsec encryption for simple, reliable and scalable strong layer 2 and layer 3 security for WAN, data center interconnect and for securing links between tiers in leaf and spine data center designs.
- Fully featured switching and routing EOS feature set for open multi-vendor network architectures. This includes a comprehensive virtualisation solution for the data center using EVPN-VXLAN and carrier grade routing with IP-VPN, VPLS, EVPN-MPLS, Segment Routing and Traffic Engineering.
- Large scale forwarding tables with a programmable packet pipeline, to meet the most demanding switching and routing requirements, with the flexibility to allocate HW resources to ensure optimal utilization and scale.
- Hardware-assisted IEEE 1588 PTP enables accurate timing solutions across Ethernet-based networks, without costly investment in separate timing networks.
- NEBS compliance and DC power supplies designed for service provider environments

- Accelerated sFlow and IPFIX for network forensics
- Streaming network state for advanced analytics with Arista CloudVision®
- Unique monitoring and provisioning features – LANZ, DANZ, AEM, ZTP, VM Tracer, and eAPI

Arista R4 Series Deployment Models

Arista's rich EOS routing stack, in conjunction with the flexible footprint of the 7800R4 platforms, wide range of interface speeds and density, provides operators with a universal routing platform for a multitude of roles across data center, HPC/AI and wide area networks.

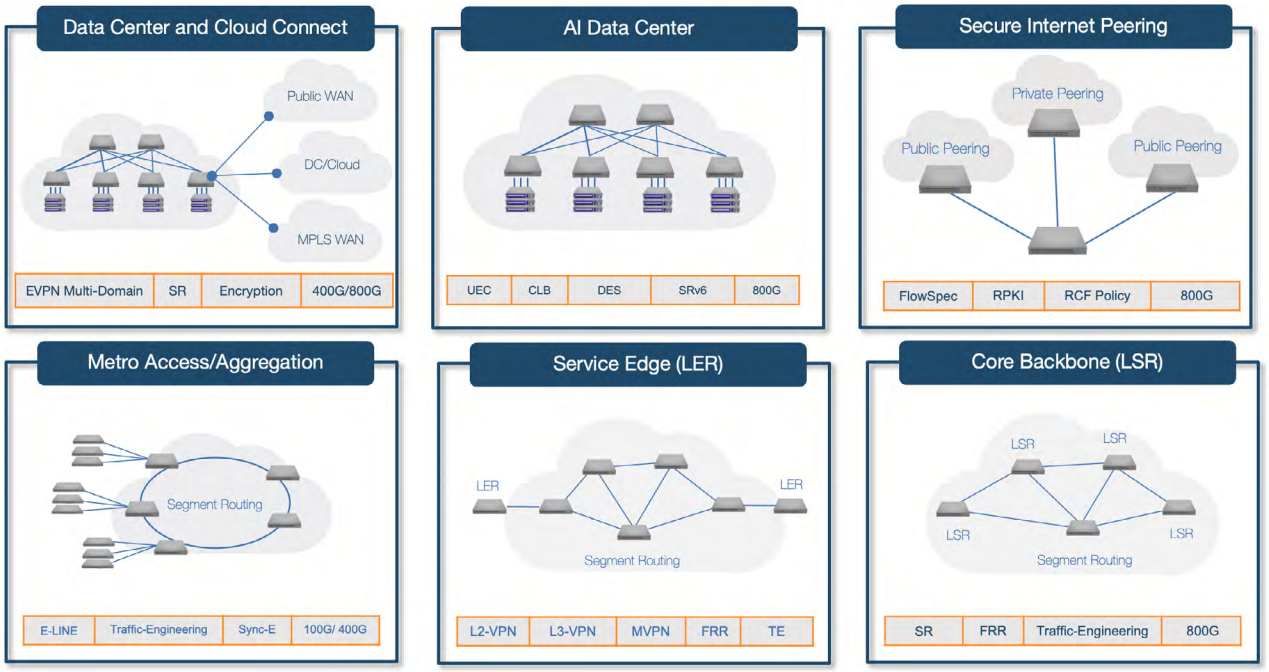


Figure 3.1: Deployment Use cases of the 7280R4 Platform Generations

- **Data Center Leaf Spine:** For virtualized multi-tenant data centers, the R4 Series with Arista's EOS software provides support for a fully featured EVPN-VXLAN software stack; which includes Symmetric/Asymmetric IRB, A-A multi-homing and EVPN-OISM.
- **AI Data Center:** For modern AI/ML workloads which require high-bandwidth, lossless, low-latency, scalable fabrics the R4 Series provides a comprehensive set of features (RoCEv2, DCQCN, PFC, CLB and DLB) to optimize performance, minimize congestion, and enhance overall network reliability of the 400G/800G AI infrastructure.
- **Secure Internet Peering:** To meet the demands for secure high-speed 100G/800G internet peering, the R4 Series provides support for holding multiple copies of the full Internet table in hardware. Alongside a rich security and DDoS mitigation feature set, including large modular ACLs with deep packet inspection, full featured Flowspec stack, and a robust RPKI implementation for secure routing decisions.
- **Next-Generation Service Edge:** To deliver cost efficient revenue generating high-bandwidth VPN services, the R4 Series provides full support for both traditional Ethernet and IP VPN services (IP-VPN, MVPN, VPLS and PW) and next-generation EVPN services (L3VPN, L2VPN, E-tree and OISM).
- **Metro Access and Aggregation:** For 5G backhaul and triple-play business services at the Metro edge, the R4 Series support resilient Ethernet services, including E-LINE, E-LAN, E-Tree services (EVPN, PW, VPWS, L2VPN) in addition to layer 3 VPN services (IP-VPN and EVPN-MPLS). With a rich OAM and PTP/SyncE software stack for timing and performance monitoring of the services.

- **Next-Generation Core:** To construct next-generation provider core networks, R4 platforms with Arista's EOS software provide support for multiple transport models; LDP, mLDP, RSVP-TE (P2P/P2MP) and SR-TE with fast re-route capabilities, while providing a flexible label allocation framework to allow the coexistence and seamless migration of LDP/RSVP-TE deployments to next-generation SR/SR-TE solutions.

Arista EOS state-driven operating system

As Cloud, Enterprise and Provider networks scale to support 400G and 800G port speeds, a more programmable software-driven operational model that is both agile, while cost-efficient is required to ensure operating costs remain constant or decrease as the infrastructure scales. This evolution to a software driven operational model places greater focus than ever before on the architectural design, resilience and programmability of the software running within the infrastructure.

Arista's EOS is a next-generation state-driven modular operating system, designed to address the requirements of very large scale environments. The EOS architecture builds on a standard Linux kernel and runs all processes in their own protected memory space, cleanly separating switch state from protocol processing and application logic through an in-memory database (NetDB).

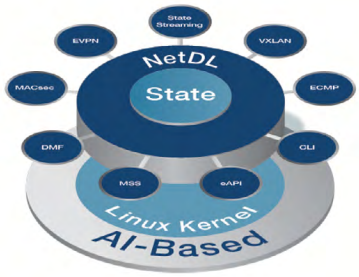


Figure 4.1: EOS state-drive modular operating system

The in-memory NetDB database, machine generated at run time, runs in user space and contains the complete real-time state of the system. Like traditional databases, NetDB does not contain any application logic and is only responsible for keeping state. However, rather than being optimized for transactions, NetDB is designed for synchronizing state among processes, also called 'EOS agents', by notifying interested agents when there is a state change. Each EOS agent subscribes to NetDB to be notified when the state of other related agents change within NetDB, When a state change occurs within an agent, updates are then published to NetDB, which then in turn notifies the subscribed agents interested in the change.

This centralized database approach to passing state throughout the system, and the automated way the NetDB code is generated, reduces system overhead and simplifies inter-process communication to significantly reduce the number of possible code paths, reducing the scope for errors and dramatically improving testability. By removing inter-process dependency and direct communication between agents the architecture also improves software feature velocity and quality, and provides openness for customers wishing to build their own applications, who can use the same in-built APIs to receive notifications from NetDB both for state visibility and feature customization.

Serviceability

The multi-process state sharing architecture of the EOS operating system, provides the foundation for industry leading availability and serviceability on the R-Series platform. In traditional network operating systems, software faults or security patches often require a software reload due to the monolithic architecture of the operating system, resulting in seconds to minutes of downtime. Reconvergence around such issues places additional load on neighboring devices, as topology changes ripple across the network.

With the modular architecture of the EOS operating system, a fault is contained within the agent where the fault originated, in the unlikely event that a fault causes an agent to crash, then the EOS process manager restarts a new instance of the agent. With the separation of state from the processing agents, there is no requirement for the re-starting agent to query or process any older stale state, instead the current state can be pulled directly from the NetDB database. If the fault causes the agent to hang or loop, EOS detects the condition and seamlessly restarts the agent, providing a self-healing architecture. This multi-process state-sharing architecture is also key to reducing maintenance windows by allowing more operational tasks to be performed during normal hours,

without the need for downtime; EOS agents can be patched live and restarted if necessary without disrupting the overall operation of the node.

Programmability

As provider networks linearly scale and new revenue generating services are enabled, continuing to deliver an industry leading return on investment creates the need for a more agile programmable operational model. To deliver this new software-driven operational model cost effectively, EOS is fully programmable across all layers of its software stack – Linux kernel, hardware forwarding tables, node configuration, control plane as well as the management layer via open APIs. The rich set of structured APIs includes:

- Tight dev-ops integrations with Puppet, Chef and Ansible
- eAPI JSON based RPC, providing a REST-like interface for configuration and monitoring, using native CLI commands
- OpenConfig, Go, Python and Ruby based object models
- NETCONF and Restconf transport protocol
- Native Go and Python scripting
- Native Linux APIs and scripting
- Scope to develop native high performance applications using EOS SDK

The open programmability of the EOS software stack, and the storing of state in a common database, NetDB, which is easily accessible through the open APIs, allows the the R-Series platforms to be rapidly and easily integrated with a wide range of third-party and open-source applications for service initiation, traffic engineering, network management, automation, and network orchestration.

Real-time state streaming and analytics

Arista EOS software supports traditional Syslog, SNMP traps and polling mechanisms for collecting and reporting routing state, however these traditional approaches can often be restrictive when trying to monitor protocol state, environmental alarms, memory, buffers etc. in real-time. To provide this level of visibility, Arista's EOS software supports real-time state telemetry. All network state (for example, interface statistics, configuration, protocol, routing table, environmentals etc.) is stored within EOS's state-driven NetDB database, and can be streamed off-box in real-time via gRPC. This open standard interface provides third-party monitoring tools and Arista's CloudVIsion Portal (CVP) an unprecedented level of visibility into the ongoing health and performance of the routing infrastructure.

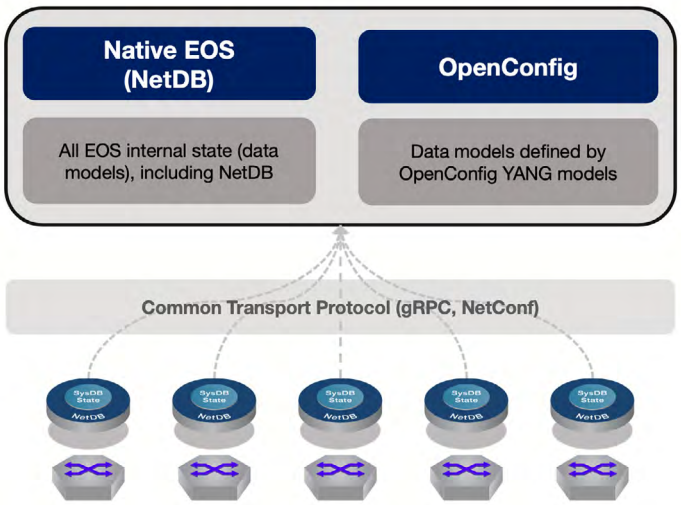


Figure 4.2: EOS real-time state driven telemetry architecture

With state collection, through CloudVision or third-party tools, based on real-time streaming rather than polling, state updates are provided continually as changes occur. This approach is vastly superior to traditional polling models that have a typical granularity in the order of minutes and are often limited due to system control plane capacity. Also unlike traditional pre-defined SNMP MIB-based approaches, the EOS state streaming model is capable of streaming the full state of each node in the infrastructure, this includes details such as configuration, counters, errors, statistics, tables, environmentals, buffer utilization, flow data, and much more. This streaming architecture also forms the basis for EOS's YANG-based OpenConfig data models.

Arista FlexRoute™ and Programmable Pipeline

One of the key characteristics of the Arista R4 Series universal routing platform is its programmable pipeline and flexible resource allocation. The FlexRoute™ Engine is one example of Arista's innovative implementation of the platform's flexible pipeline.

The FlexRoute Engine provides internet scale routing, enabling support for large scale FIB routes on standard data center scale (R4) platforms, scaling to higher capacity on R4K platforms, allowing the platform to hold multiple copies of the full internet table in hardware while ensuring investment protection for future growth. The FlexRoute Engine is a patented algorithmic approach to constructing layer 3 forwarding tables on the R-Series platforms and is a key enabler to building scalable internet peering and VPN services. This scale is achieved with significant power consumption savings over traditional approaches which use longest prefix match (LPM) lookups and external TCAMs. FlexRoute enables the R4 Series to deliver Internet route scale at a higher port density and performance while achieving power and cooling advantages over traditional platforms.

To accommodate the wide range of network roles made possible by the platform's FIB scale and breadth of feature support, the platform's pipeline resources are programmable through the selection of a forwarding profile, allowing the balancing of layer 2 and 3 forwarding entries and lookup resources (ACLs, QoS) across the ingress and egress stages of the pipeline. However, unlike legacy routing silicon architectures, increasing the size of a specific forwarding table does not compromise the performance or latency of the packet processor, which remains at wire-speed.

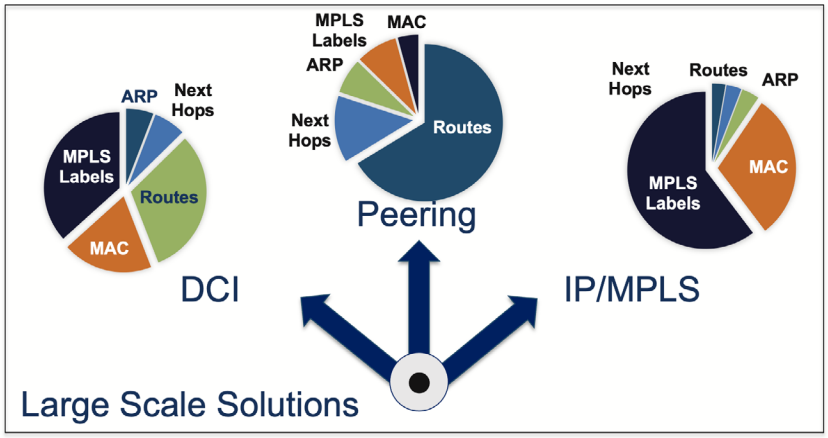


Figure 5.1: Flexible resources enable a wide range of deployment profiles.

The fungible nature of the resources within the R4 Series provides the flexibility operators require to standardize on a common platform across the DC and WAN, with the confidence that the specific scale requirements of a given network role can be appropriately addressed. This removes the need for dedicated platforms for each role within the network, thereby enabling cloud and service providers to streamline their deployments, operational skillsets while simplifying sparing and consolidating testing.

Tunnelsec

Arista's TunnelSec technology, native to the R4K platforms, provides hardware encryption, with support for IPsec, MACsec and Arista's innovative VXLANsec solution. Instead of requiring external encryption processors, TunnelSec encryption is embedded directly into the packet processor, ensuring wire-speed encryption performance for all three schemes, while providing the flexibility to choose the appropriate encryption technology for the deployment scenario within a single platform.

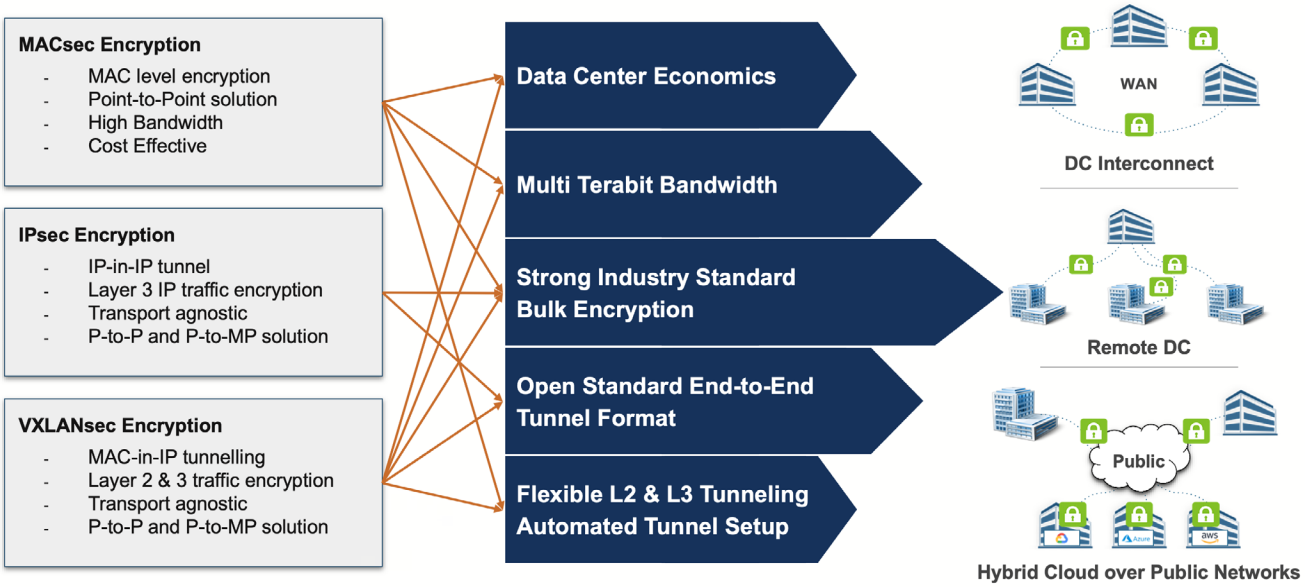


Figure 6.1: TunnelSec Overview

- **MACsec:** R4K platforms provide native 25G to 800G line-rate MACsec (802.1AE) leveraging the quantum-safe AES-256-GCM bulk cipher. MACsec is a point-to-point encryption solution performed at the MAC layer, where security keys are exchanged with the neighboring node of the link, resulting in all data beyond the initial Ethernet header being fully encrypted between the two nodes. Operating at the MAC layer, MACsec is an appropriate fit for providing data encryption on point-to-point links of a WAN backbone, across the leaf spine links of a Data Center fabric and as part of a high-speed point-to-point Data Center Interconnect (DCI) solution.
- **IPsec:** As an alternative to MACsec, R4K platforms also provide support for standards based IPsec (RFC 4303) with 25G to 800G line-rate authentication and encryption using the quantum-safe AES-256-GCM bulk cipher. IPsec is an IP-in-IP tunneling technology, where the original IP payload is encrypted and encapsulated within a new IP packet (IPsec Tunnel node). This allows the encryption end-points of an IPsec solution to be transparently routed across an intermediate backbone. This is a cost effective approach when dedicated dark fiber or dedicated circuits between sites is cost-prohibitive for a MACsec solution. IPsec also offers the additional benefit of native support for both point-to-point and point-to-multipoint encryption topologies. This allows the interconnection of multiple sites in both a full-mesh and hub-and-spoke topology.
- **VXLANsec:** Arista's innovative VXLANsec encryption technology, again based on quantum-safe AES-256-GCM, offers a mechanism to combine both L2 and L3 wide area connectivity securely. VXLANsec is an VXLAN-in-IP tunnelling technology where the original VXLAN header and payload is encrypted and encapsulated within a new IPsec packet. This provides the ability to encrypt all VXLAN traffic when transported between VTEPs. Fully integrated with an EVPN control-plane, the VXLANsec solution can be used within a colo site for securely interconnecting EVPN domains at high-speed across third-party circuits or across sites as part of a multi-tenant Data Center Interconnect (DCI) solution.

Arista 7800R4 - Platform Overview

The 7800R4 Series is available in a choice of 4-, 8-, 12- and 16-slot systems that support line cards providing high density 100G, 400G and 800G with a choice of forwarding table scale and optional TunnelSecTM (MACsec, IPsec and VXLANsec) encryption.

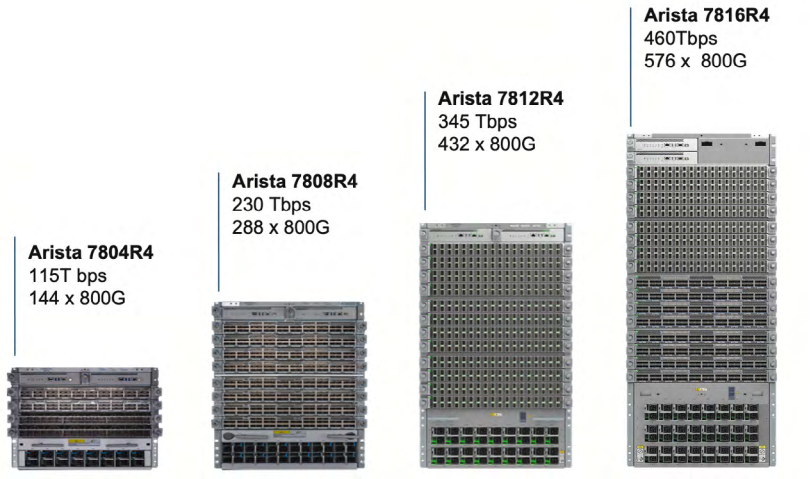


Figure 7.1: 7800R4 modular 4,8,12 and 16-slot chassis

At a system level, each chassis in the family shares a common architecture, supporting redundant, hot-swappable, and field-replaceable components such as supervisor modules, fabric cards, power supply units (PSUs), and fans. With common line cards and software features across the platforms, the 16,12, 8 and 4-slot systems provide the flexibility operators require to standardize on a common platform across the DC and WAN, with the confidence that the specific scale requirements of a given network role can be appropriately addressed. This removes the need for dedicated platforms for each role within the network, thereby enabling cloud and service providers to streamline their deployments, operational skillsets while simplifying sparing and consolidating testing.

The key physical, performance and port density metrics of the four 7800R4 platforms, is outlined in the table below.

Table 7.1: Arista 7800R4
Physical dimensions, port density and forwarding performance

Characteristic	Arista 7804R4	Arista 7808R4	Arista 7812R4	Arista 7816LR4
Chassis Height (RU)	10	16	23	32
line card Module slots	4	8	12	16
Supervisor Module slots	2	2	2	2
Fabric Module slots	6	6	6	6
100G Maximum Density	1152	2304	4608	3456
400G Maximum Density	288	576	864	1152
800G Maximum Density	144	288	432	576
System Usable Capacity (FDX)	115 (230) Tbps	230 (460) Tbps	345 (690) Tbps	460 (920) Tbps

Arista 7800R4 System Components

All 7800R4 modular platforms share a consistent architecture with 32 Tbps of fabric bandwidth and 28.8 Tbps forwarding capacity per slot. Line cards and power supplies are common across systems, while supervisors and fabric modules are system specific. Airflow is always front-to-rear and all data cabling is at the front of the chassis.

Chassis and Midplane-less system

Chassis design and layout are key aspects that enable high bandwidth density, efficient power consumption and reliability. All four sizes of 7800R4 series systems are midplane-less, with line card modules connecting directly to orthogonally oriented fabric modules.

This design alleviates the requirement to route high speed signal traces on a midplane or backplane, reducing trace lengths between system elements and enabling high speed signals to operate more efficiently with high signal integrity. The midplane-less arrangement also eliminates unnecessary components resulting in a more compact, lighter device with improved airflow resulting in lower power consumption and higher long term reliability.

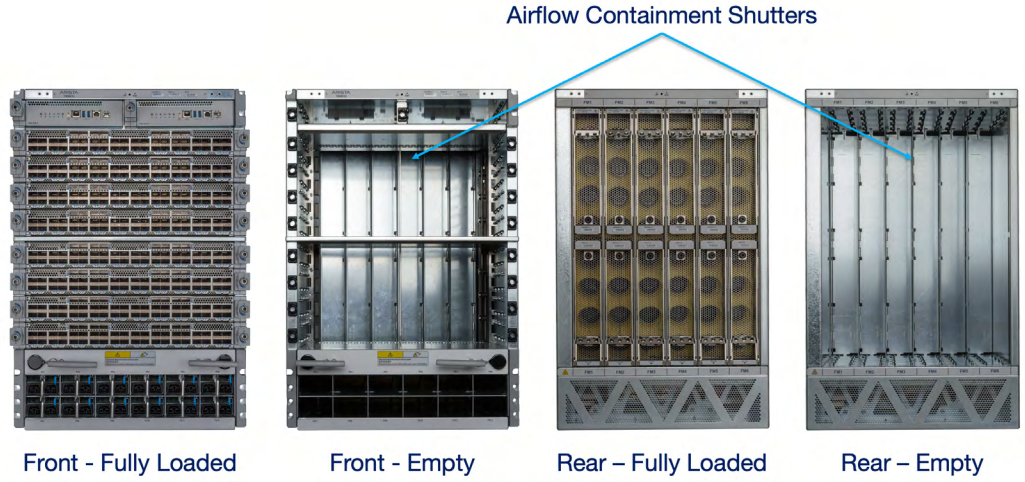


Figure 8.1: Arista DCS-7800R4 mid-plane-less orthogonal fabric architecture

Fabric Modules

7800R4 platforms employ multiple active fabric modules to ensure resilient traffic forwarding between line cards and packet processors. Each fabric module connects to every line card in the chassis, with the overall fabric capacity being the combined throughput of all fabric modules.

This design ensures equal bandwidth and full connectivity to all line cards as well as graceful degradation of the system's throughput in the event of a fabric module failure. If a failure does occur each fabric module is hot swappable and may be inserted or removed while the system is in operation.

Table 8.1: Arista 7800R4 Platform and fabric throughput per slot

Characteristic	Arista 7804R4	Arista 7808R4	Arista 7812R4	Arista 7816LR4
Number of fabric modules	5	5	5	5
Fabric model	DCS-7804R4-FM	DCS-7808R4-FM	DCS-7812R4-FM	DCS-7816LR4-FM
Throughput per fabric module	25.6 Tbps	51.2 Tbps	76.8 Tbps	102.4 Tbps
Fabric throughput per slot	32 Tbps	32 Tbps	32 Tbps	32 Tbps
Front-panel capacity per slot	28.8 Tbps	28.8 Tbps	28.8 Tbps	28.8 Tbps
Throughput per system (FDX)	115 (230) Tbps	230 (460) Tbps	345 (690) Tbps	460 (920) Tbps
Redundancy	Graceful degradation	Graceful degradation	Graceful degradation	Graceful degradation

A single fabric module provides 6.4 Tbps fabric bandwidth (6.4 Tbps transmit and 6.4 Tbps receive, sometimes referred to as 12.8 Tbps Full Duplex) to each line card slot. With all five fabric modules active this provides 32 Tbps of fabric bandwidth to each slot in the chassis, thus providing more bandwidth than required for non-blocking performance on all ports of a 36 x 800G line card (28.8 Tbps).

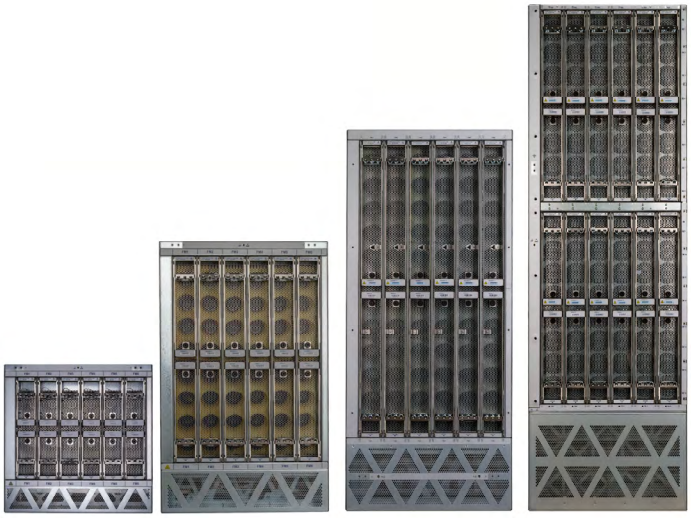


Figure 8.2: Arista DCS-7800R4 Series Fabric/Fan modules

Each fabric module contains multiple field serviceable fan modules composed of dual counter-rotating fans. Fans are connected to two independent fan controllers to increase fault tolerance. To ensure that there is sufficient cooling capacity for line cards and fabric modules, fans are powered on before any active forwarding elements and continue to operate independently even if the fabric silicon is disabled.

To maximize efficiency, cooling is segmented into independent cooling zones, each responsible for four line cards. This scheme allows per-zone control of fan speed, minimizing power utilization for zones with fewer line cards or lower heat loads to maximize efficiency and minimize power consumption.

As part of the forwarding plane, fabric modules form an intrinsic part of the tightly coupled end-to-end scheduling mechanism in the 7800R4 series, including:

- **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 learned, moves or is aged out, the ingress packet processor with ownership of the MAC address will update other packet processors.
- **Data-plane health tracer:** all packet processors within the system send continuous health check messages to all other packet processors, validating all data-plane connectivity paths within the system

Packets crossing the fabric are split into cells before being sprayed across all possible fabric paths before being reassembled by the receiving packet processor. This mechanism eliminates congestion caused by elephant flows or poor flow distribution, which cause hot spots or blocking traditional packet-based fabrics. The forwarding plane is covered in more detail later in this document.

7800R4 Series Line Cards

The 4, 8, 12 and 16-slot 7800R4 modular platforms support a common set of line cards. Line cards utilize on-board packet processors to provide a distributed forwarding plane for switching traffic between packet processors on the same line cards and across line cards.

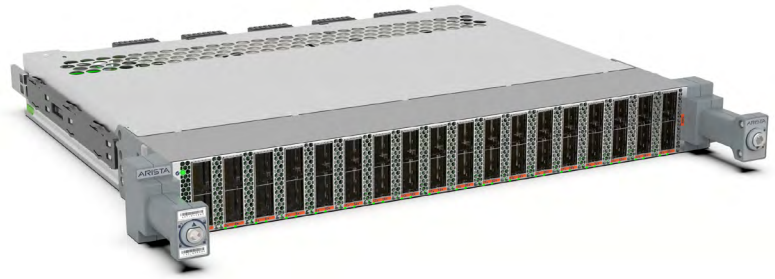


Figure 8.3: Arista DCS-7800R4 line card modules

All packet flow within the system is fully scheduled via the fabric, ensuring fair and deterministic traffic handling for both ports on the same packet processor as well as ports on other processors. Virtual Output Queuing (VOQ) between the input and output interfaces, for both locally switched and non-locally switched packets, ensures there is fairness even when traffic is locally forwarded.

Forwarding between ports on the same packet processor utilizes fabric based scheduling but local payload switching. Forwarding across different packet processors utilizes the fabric for both scheduling and transmission of cellified packets between ingress and egress processors.

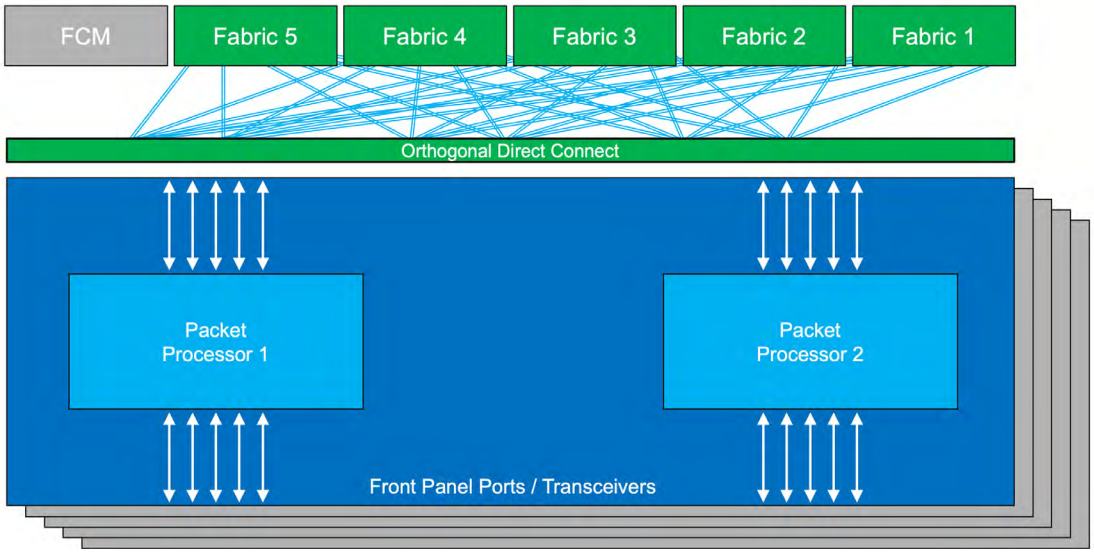


Figure 8.4: Distributed Forwarding within an Arista 7800R4 Series

Supervisor Modules

7800R4 Supervisor modules provide control-plane and management-plane functions. Each system can operate with a single supervisor or a redundant pair, each capable of managing the system. All data-plane forwarding is performed on the line cards and forwarding between line card modules is always via the fabric modules.

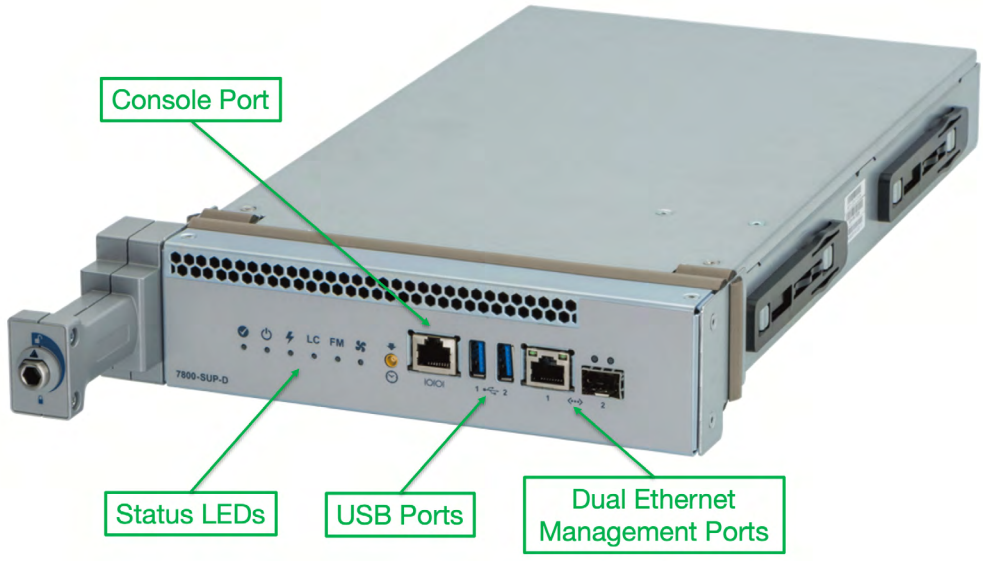


Figure 8.5 : Arista 7800 Series Supervisor Module.

7800R4 Series Supervisors are based on multi-core x86 processors and DRAM appropriately sized for high capacity systems. Each module connects over dedicated high bandwidth interfaces to all line cards and to the adjacent supervisor.

Two form-factors exist, one module common to 7804 and 7808 and one common to 7812 and 7816L systems. The following table details the specification of each module:

Table 8.2: Arista 7800R4 Supervisor models

Supervisor model	Processor	RAM	SSD	7804R4 7808R4	7812R4 7816LR4	Secure Boot Enabled
DCS-7800-SUP1A	6c/12t 1.9 GHz x86	64 GB	256 GB	Yes	–	–
DCS-7800-SUP1S	6c/12t 1.9 GHz x86	64 GB	256 GB	Yes	–	Yes
DCS-7816-SUP	8c/16t 2.0 GHz x86	64 GB	256 GB	–	Yes	–
DCS-7816-SUP1S	8c/16t 2.0 GHz x86	64 GB	256 GB	–	Yes	Yes

Arista's Extensible Operating System (EOS®) is a natively multi-threaded modern operating system that makes efficient use of multi-core CPUs and high performance DRAM. EOS is built on a 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 of the Supervisor modules also provide headroom for running third party software within the same Linux instance as EOS, within a guest virtual machine or within containers. An enterprise-grade SSD provides additional flash storage for logs, VM images or third party software packages.

Out-of-band management is available via a serial console port and/or dual 10/100/1000 Ethernet interfaces (SFP and RJ45 ports are provided). There are two USB 2.0 interfaces that can be used for transferring images/logs or many other uses.

Power Supplies

7800R4 platforms are equipped with common redundant and hot-swappable AC or DC power supplies with an internal variable speed fan. Each dual-input power supply integrates 1+1 grid redundancy in a choice of 3KW HVAC, HVDC or LVDC formats across each of the four chassis platforms. The AC power supply's dual inputs operate in an active/standby configuration, are Titanium climate saver rated and have an efficiency of over 94% with single stage conversion to the internal 12V DC voltage. Hybrid HVAC/DC supplies support wide ranging AC or DC inputs (240-380Vdc), while LVDC supplies are available to suit -40 to -72V deployment.

7800R4 platforms utilize a single internal power domain, so there is no need to allocate specific power supply units to individual power zones. This enables operators to achieve a minimum of N+2 power supply redundancy, with N+N in most common scenarios, as well as balance power feeds across data center power grids.

Table 8.3: Arista 7800R4 PSU capacity

System	Included PSUs	Maximum PSUs	Max Power
7804R	6	8	24 kW
7808R	8	12	36 kW
7812R	10	18	54 kW
7816LR	12	24	72 kW

Arista 7800R4 Forwarding Architecture

The 7800R4 platform is a modular multi-processor architecture, with fully distributed end-to-end scheduling. All packet processors operate in a fully coordinated manner to behave like a single very large monolithic chip that is internally lossless, fair and deterministic. The following sections provide an overview of the forwarding-plane.

Fabric and Queuing Architecture Overview

Three critical mechanisms are combined to create this highly versatile and performant architecture:

Fully Scheduled Fabric

Unlike typical multichip systems that can be thought of as 'an Ethernet network in a box', where there is no robust internal congestion management, the 7800R switch fabric implements a Distributed and Fully Scheduled paradigm.

In order to guarantee lossless transmission of a packet from ingress to egress, a distributed, continuous and self-regulating algorithm runs between all packet processors. In one direction, this algorithm ensures that every egress port knows which ingress ports have waiting traffic. In the other direction the egress processor distributes credits to each waiting ingress processor in a fair and paced way that allows traffic to flow without congesting the fabric or the egress port and avoids the need to drop packets.

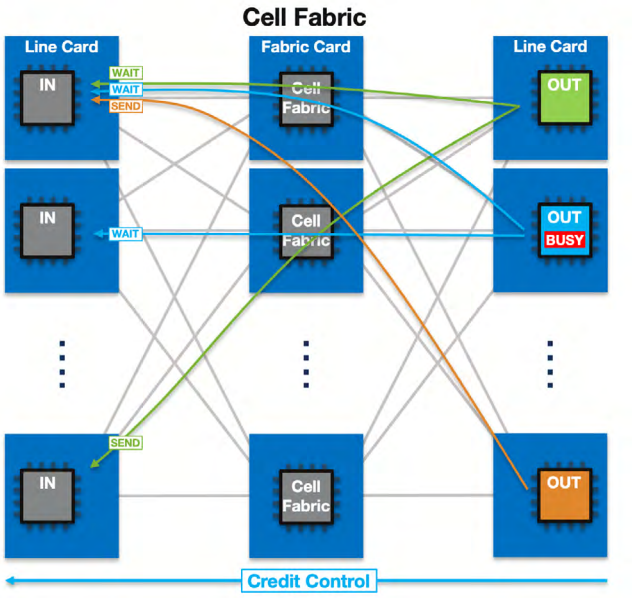


Figure 9.1 : Arista 7800 Fully Scheduled Fabric

Cell Based Transmission

Outside of the switch, Ethernet traffic consists of variable packet sizes that may arrive in long or short bursts and are poorly synchronized to fill empty slots on the wire. These aspects contribute to phenomena known as microbursts where there are instances of extreme contention for a link or interface as well as issues mixed flow sizes (elephants and mice) and with load balancing large flows evenly across bundles of links.

In a simple multi-chip switch design these same congestion events occur internally but may be invisible to the operator - a worst case scenario when trying to optimize for intensive applications or simply troubleshoot performance issues.

The 7800R architecture eliminates this issue entirely by combining the distributed scheduling described above with a cell based internal fabric. Instead of transmitting variable sized packets across the fabric, each packet is segmented into multiple equal sized cells with cells being sprayed across all available parallel paths through the fabric. The egress chip receives the cells and reassembles the packet for transmission to the client.

The segmentation into cells, inbuilt load balancing and multi-pathing ensures it is possible to achieve 100% utilization of all fabric links and therefore 100% perfect load balancing without packet loss while maintaining flow-safe packet ordering. Externally, hosts continue to user standards based Ethernet and IP - there is no requirement for specialized network adapters to benefit from the 7800 architecture.

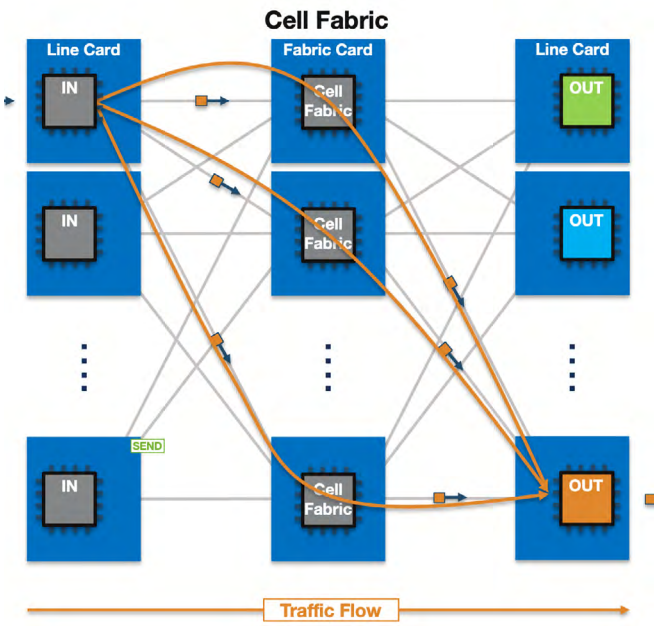


Figure 9.2 : Arista 7800 Cell Based Transmission

Virtual Output Queuing

Classical switching architectures implement packet queues based on output interfaces, meaning multiple traffic inputs contending for a specific output port are always sent to the output port and at that point the decision is made to enqueue or drop the packet. The queueing resources are fixed by the allocation of the output port, meaning the higher the number of contending flows, the less room per flow available in the queue. Finite egress resources naturally lead to drops under congestion.

This "fire and forget" method of shipping traffic from ingress to egress is wasteful in terms of internal crossbar bandwidth (shipping packets across the fabric that will be immediately dropped), exacerbates noisy neighbor and in-cast scenarios and is susceptible to head-of-line blocking (HOLB); ultimately leading to traffic loss and low application performance.

To eliminate these issues, the 7800R Series implements Virtual Output Queuing (VOQ). Instead of relying on a fixed egress side output queue, each input port on every leaf switch or line card has a set of logical virtual queues corresponding to every possible traffic class for every possible output port in the system. These queues are independent by source, thus eliminating HOLB and providing the ability for the device to scale enqueuing of traffic linearly with the number of inputs contending for any given output.

In other words, VOQ provides the ability to handle mixed interface speeds, noisy neighbors and transient high volume traffic bursts by dynamically scaling the instantaneous queue size available to a contention scenario in relation to the breadth of the in-cast itself. VOQ works hand in hand with the fully scheduled, cell based fabric to ensure lossless fabric forwarding.

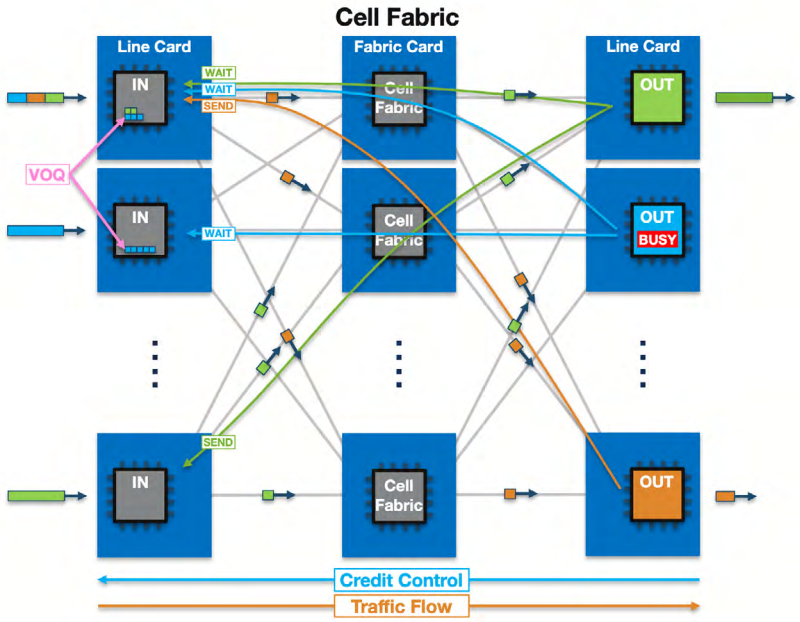


Figure 9.3 : Arista 7800 Virtual Output Queueing

Forwarding Pipeline

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

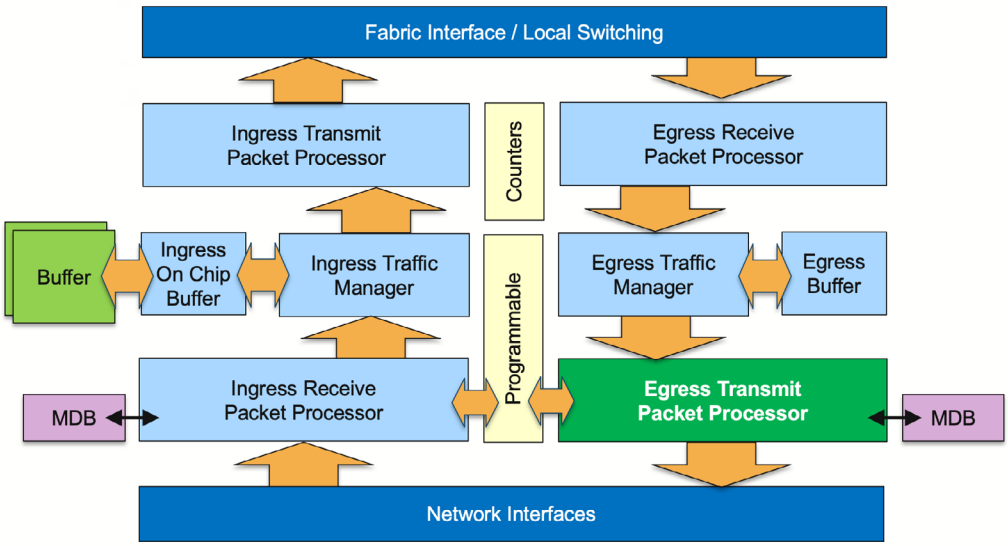


Figure 9.4: Packet forwarding pipeline stages inside a packet processor on an Arista 7280R4 Series

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

Stage 1: Networking 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).

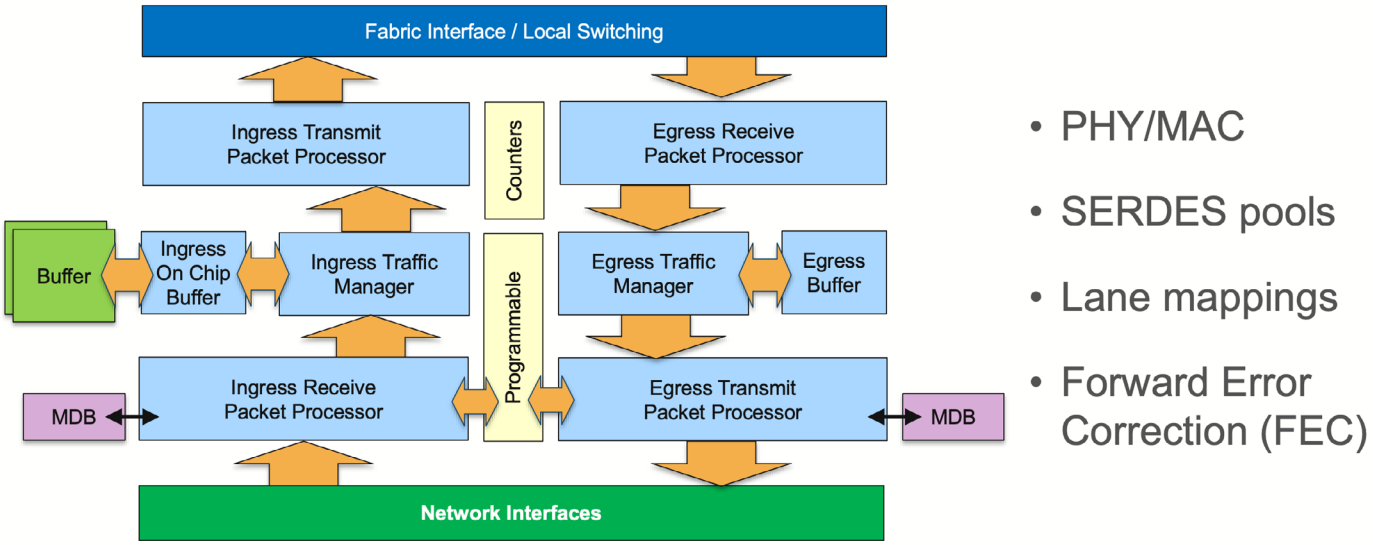


Figure 9.5: Packet Processor stage 1 (ingress): Network Interface

The PHY layer is responsible for the transmission and reception of bitstreams 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 mappings are used to map the physical lanes to logical ports based on the interface type and configuration. If a valid bitstream is received at the PHY then the data is sent to the MAC layer. On input, the MAC layer is responsible for turning the bitstream into frames/packets: checking for errors (FCS, Inter-frame gap, detect frame preamble) and finding the start of frame and end of frame delimiters.

Stage 2: Ingress Receive Packet Processor

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

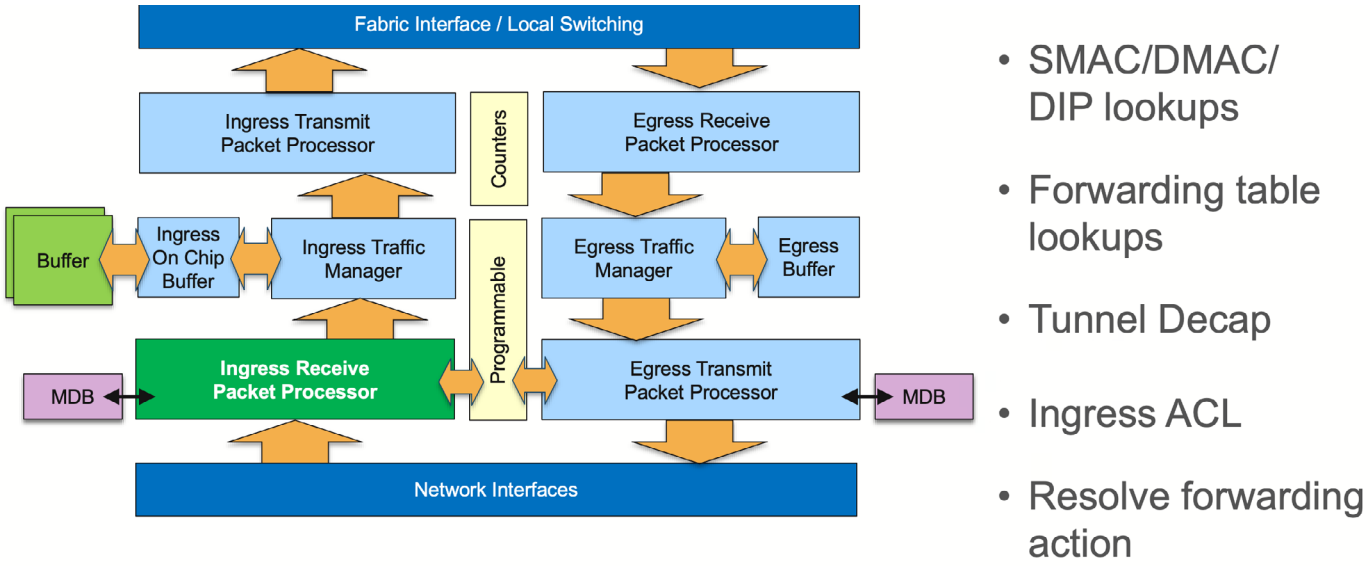


Figure 9.6: Packet Processor stage 2 (ingress): Ingress Receive Packet Processor

Before any forwarding can take place, 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 R4 packet parser also supports multiple tunnel formats (MPLS, IPinIP, GRE, VXLAN, MPLSoGRE, SRv6) as well as 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 forwarding models.

After parsing the relevant encapsulation fields, 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 is used for L2 forwarding (if present) 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. If the DIP matches a subnet local to the switch for which 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. Additionally, 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, the fields which are configured for load balancing calculations are used to derive a single matching entry. The final matching adjacency entry provides details on where to send the packet (output interface and a pointer to the output encapsulation/MAC rewrite on the egress pipeline).

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 hardware-based forwarding path. In cases where software assistance is required (e.g. traffic destined for a L3 subnet for which there is no current ARP entry for the end host, to map the destination IP to a L2 MAC 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.

Counters are available within this stage providing accounting and statistics on ACLs, VLAN and sub-interfaces, as well as a range of tunnel and next-hop group types. The R4 Series systems provide flexibility in how these counter resources can be utilized making them malleable to the specific use case.

Stage 3: Ingress Traffic Manager

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

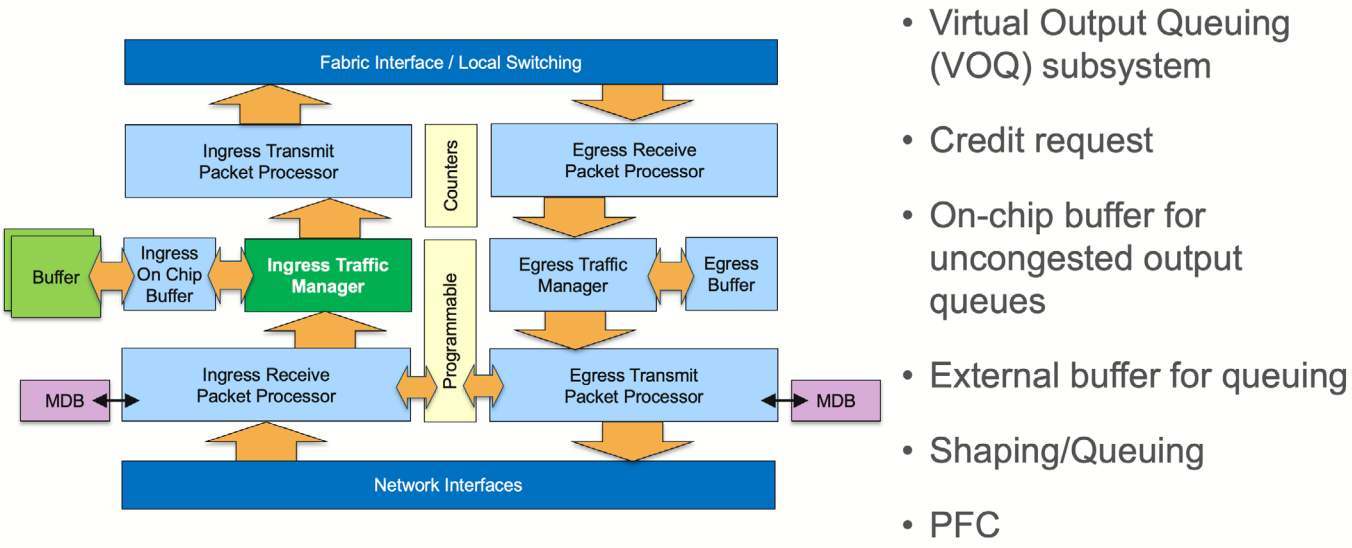


Figure 9.7: Packet Processor stage 3 (ingress): Ingress Traffic Manager

As R4 platforms utilize a Virtual Output Queuing (VOQ) architecture, the majority of the buffering within the switch is on the ingress port. While the physical buffer is on the input packet processor, it represents packets queued for transmission on a specific output port (hence, the term virtual output queuing). VOQ allows buffers for an output port to be balanced across multiple ingress ports. In the event of congestion on the output port, QoS policies can then be implemented in a fair and distributed manner across the contending ingress ports.

When a packet arrives into the Ingress Traffic Manager, a VOQ credit request is forwarded to the Egress Traffic Manager (ETM) 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 ETM of the output port, indicating the Ingress Traffic Manager can forward the frame to the egress port.

While the VOQ request/grant credit cycle is underway, the packet is queued in the processor's input buffer. Ingress buffers are realised as a hierarchy of on-chip memory (up to 256 MB) and external (on-package or discrete) memory (up to 4 GB per packet processor core). Memory is dynamically allocated to store packets while awaiting the VOQ grant. Memory is allocated such that traffic destined to uncongested outputs (egress VOQ is lightly occupied) will go into on-chip memory, while flows experiencing sustained congestion can be moved dynamically to external buffer memory. The hierarchical hybrid buffer architecture ensures maximum throughput and minimum latency for uncongested traffic.

The amount of total on-chip memory and external memory for a specific 7800R4 line card, will depend on the throughput and the number of Jericho3 processors on the line card, this ensures consistent buffer allocation across different line cards regardless of their port-density. The table below outlines the buffer size of each of the Jericho3 processors:

Table 9.1: 7800R4 Buffer size per Jericho3 Processor		
Packet Processor	On-chip Buffer	External Buffer per Core
Jericho 3	128 MB	4G per Core = 16GB total (#4 Cores)
Jericho 3M	128 MB	4G per Core = 8GB total (#2 Cores)

While there is up to 32GB of buffer memory per line card (e.g. two Jericho3 processors on the 7800R4-36PE line card), to avoid ports consuming all available buffers during periods of congestion, a proportion of the available buffers are dynamically pre-allocated across the available VOQs within the 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

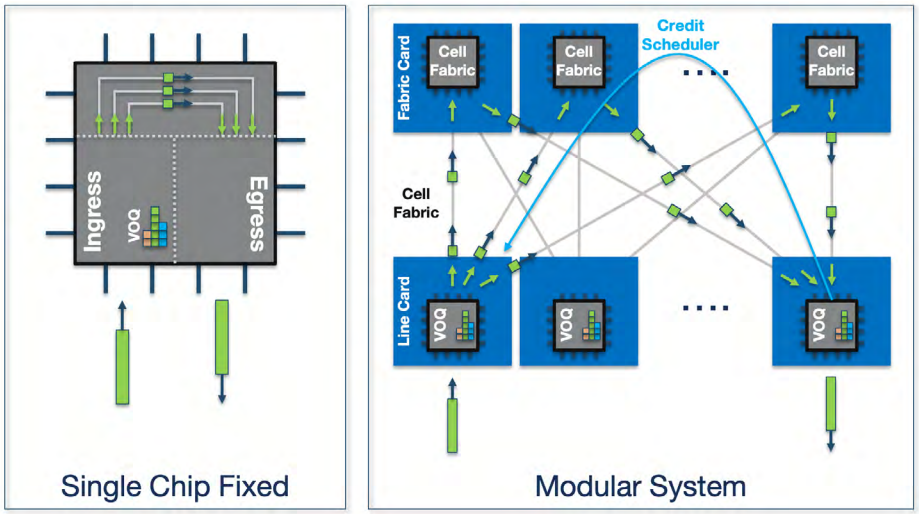


Figure 9.8: Physical Buffer on Ingress allocated as Virtual Output Queues

The dynamic pool allows the majority of the buffer to be used in an intelligent manner based on real-time congestion of the output ports. Individual VOQ limits are applied ensuring a single VOQ doesn't result in excess latency or queuing on a given output port. The default allocations (configurable via the CLI) based on port speed are outlined in the table below. :

Table 9.2: Default buffer allocation per egress port speed		
Output Port Characteristic	Maximum Packet Buffer Depth (MB)	Maximum Packet Buffer Depth (msec)
VOQ for a 25G output port	125 MB	40 msec
VOQ for a 40G output port	400MB	80 msec
VOQ for a 50G output port	400MB	64 msec
VOQ for a 100G output port	700MB	56 msec
VOQ for a 400G output port	2000MB	40 msec
VOQ for a 800G output port	2000MB	20 msec

The VOQ subsystem enables buffers that are dynamic, intelligent and deep so that there is always packet 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.

The VOQ subsystem enables buffers that are dynamic, intelligent and deep so that there is always packet 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 advanced architecture allows any application workload to be deployed – existing or future – and provides the basis for deployment in Content Delivery Networks (CDNs), service providers, internet edge, converged storage, hyper-converged systems, AI data centers, 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 any flow size in between.

7800R4 Hierarchical Packet Buffers

As outlined, the VOQ architecture of the R4 Series utilizes a combination of on-chip buffers (up to 128 MB) and a flexible on-package or discrete packet buffer (up to 4GB per Core processor).

On-chip buffers are used for non-congested forwarding while for sustained periods of congestion, external packet buffers are utilized. If the packets are required to be buffered in the external buffer, they are transmitted directly from the external packet buffer to the egress packet processor pipeline to minimize latency.

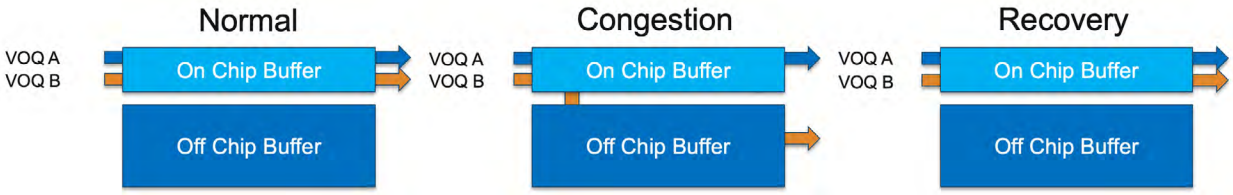


Figure 9.9: Packet buffer memory access

HBM memory is integrated directly into the packet processor package, providing an ultra-low latency connection to the packet processor die and eliminating the need for additional high-speed memory interconnects which would be the case for HMC or GDDR solutions.

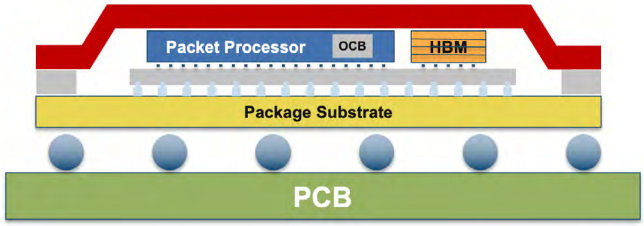


Figure 9.10: HBM memory packaging integration

This direct integration with the packet processor results in upwards of a 43% reduction in power utilization than alternative solutions and helps to reduce the thermal footprint of the switch, in turn reducing the power consumed for cooling.

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 at this stage once the output port has signaled, via a VOQ grant message, that it is the allocated slot for a given input packet processor to transmit the packet.

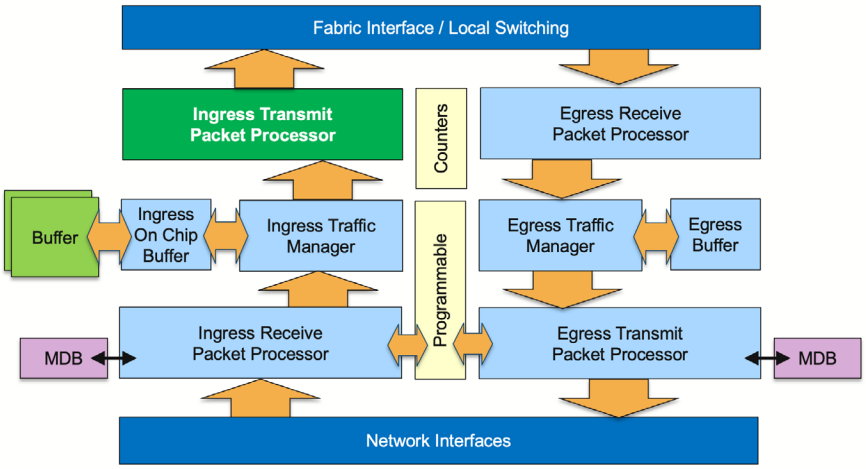


Figure 9.11: Packet Processor stage 4 (ingress): Ingress Transmit Packet Processor

All available paths are used in parallel to transfer the frame or packet to the output packet processor, with the original packet segmented into variable-sized cells if it's required to be forwarded across the fabric links in a multi-chip architecture. Packets destined to ports on the same packet processor are switched locally and do not use fabric bandwidth resources, but aren't processed any differently in terms of the VOQ subsystem. As a packet is only transferred after a VOQ grant is received, there is a guarantee that resources will be available on the egress processor to process the frame.

Segmenting the packet into variable sized cells ensures there are no hot spots on the fabric links, as cells of the packet are evenly balanced across the fabric links. When the packet is segmented into cells for transmission, 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.

Stage 5: Egress Receive Packet Processor

The Egress Receive Packet Processor stage is responsible for reassembling cells (received on the fabric links) back into packets/frames. This is also the stage that takes a multicast packet/frame and replicates it when there are multiple locally attached receivers on this output packet processor (See multicast section for further details).

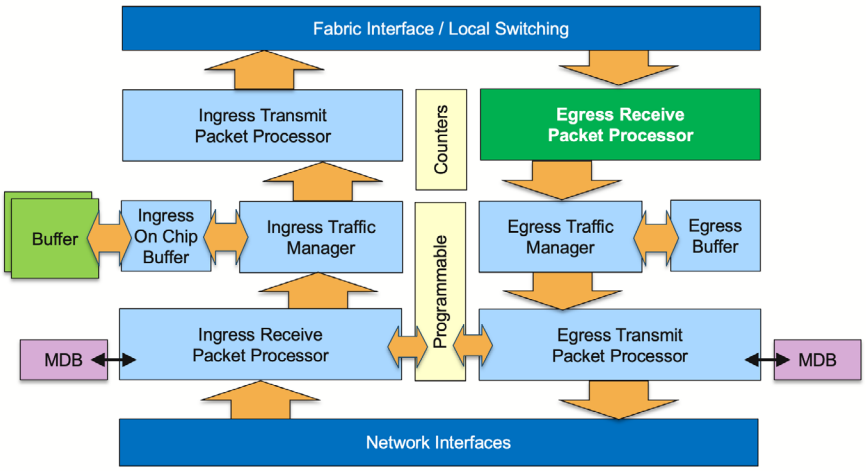


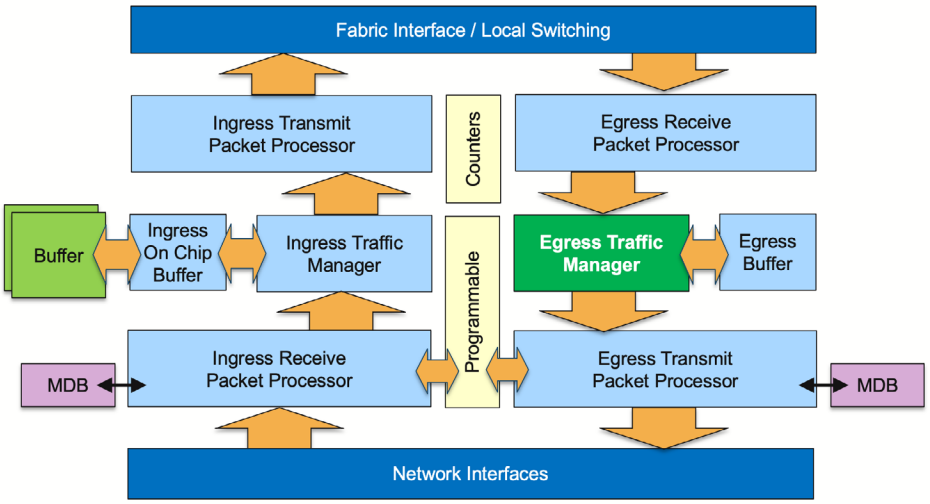
Figure 9.12: Packet Processor stage 5 (egress): Egress Receive Packet Processor

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/fabric links in the system.

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 6: 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.

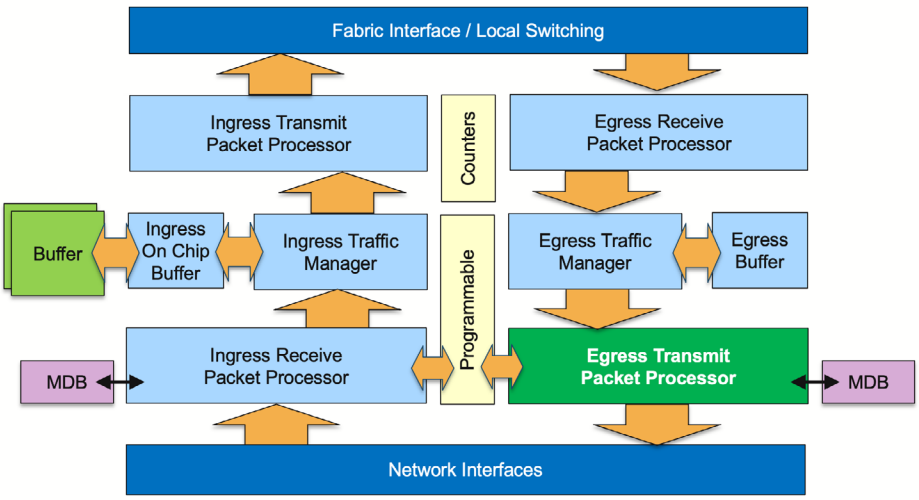


- Manage Egress Queues (unicast & multicast)
- Grant VoQ requests from Ingress
- PFC/ETS traffic scheduling

Figure 9.13: Packet Processor stage 6 (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 fairly balance the service requests 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.

Stage 7: Egress Transmit Packet Processor



- Apply egress packet header rewrites
- TCP ECN marking
- Tunnel Encap
- Egress ACLs

Figure 9.14: Packet Processor stage 7 (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 stage. Decoupling the packet forwarding on ingress from the packet rewrite on egress provides the ability to increase the next-hop and tunnel scale of the system 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 queuing 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.

Stage 8: 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 bitstream in compliance with IEEE 802.3 standards.

Multicast forwarding

The multicast forwarding pipeline of the Jericho3 processor can operate in one of two modes which are configurable in EOS:

Scheduled Ingress Replication: In this mode of operation the packet processor core attached to the multicast source is responsible for replicating the multicast stream to each egress port that has an attached receiver. This means all replication is done on the ingress core, destination egress core processors don't perform any packet replication. In this mode, multicast traffic is scheduled, where a replicated copy is only transmitted to an interested egress port after a VOQ grant has been received.

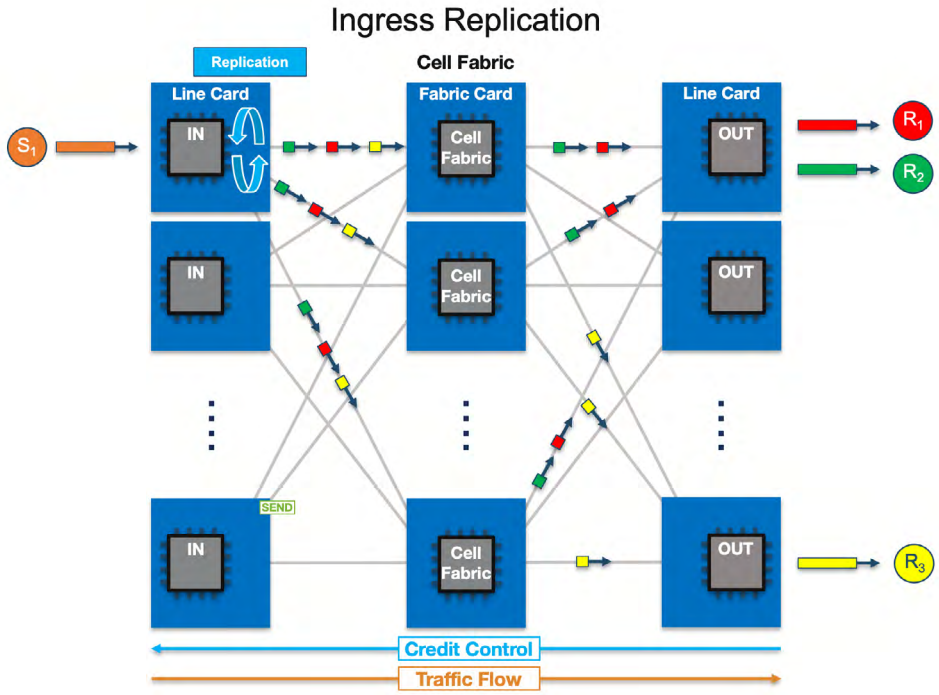


Figure 9.15: Ingress replicated multicast forwarding

Scheduled Ingress Replication with Egress Replication: In this mode of operation the multicast traffic is again fully scheduled, but the replication process is distributed across processors. The packet processor core attached to the multicast source is responsible for replicating the multicast stream to each processor core that has attached interested receivers. This first stage of the replication is scheduled using VOQ credits on the ingress core, where the packet is queued and only sent when a credit is signaled by a receiving processor core. On receiving the multicast packet, the packet is again scheduled on the receiving processor core and replicated to the locally attached receivers port when a VOQ credit for the associated interface is granted.

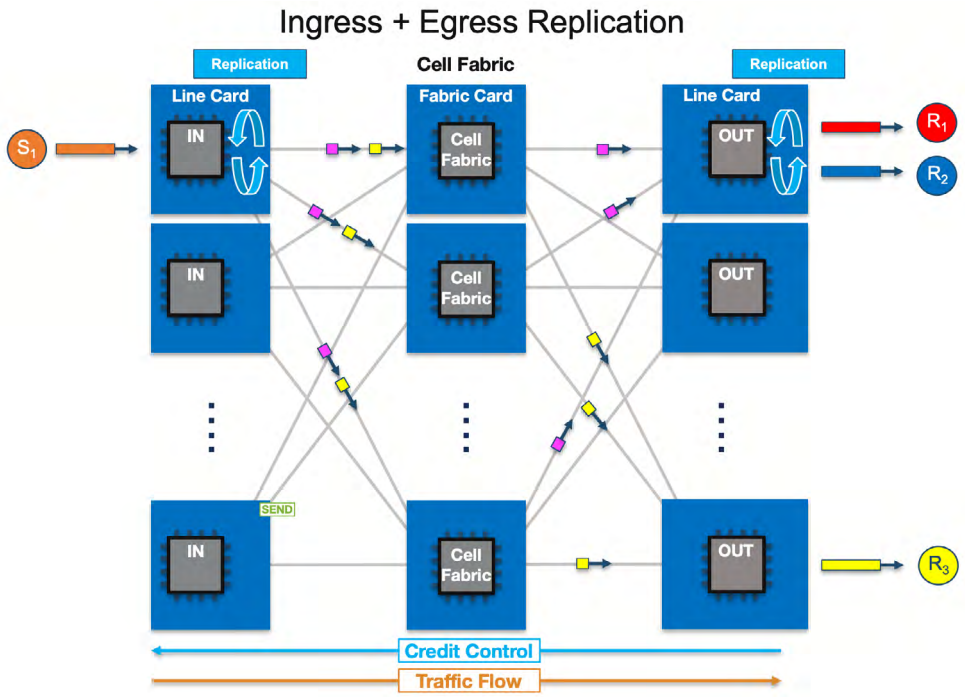


Figure 9.16: Ingress + Egress replicated multicast forwarding

processor core and replicated to the locally attached receivers port when a VOQ credit for the associated interface is granted.

Arista 7800R4: Line Card Overview

Arista 7800R4 line cards utilize the Jericho3 family of packet processors, with the number of packet processors dependent on the throughput and port density of the specific line card.

The packet forwarding architecture of the line cards is common: a group of front-panel ports (different transceiver/port/speed options) connected to a packet processor with connections to the fabric modules.

Each Jericho3 packet processor supports network interface speeds ranging from 25G to 800G for up to 14.4 Tbps (14.4 Tbps transmit and 14.4 Tbps receive) of total network capacity to the front panel ports with an additional 16Tbps (16 Tbps transmit and 16 Tbps receive) of capacity for connectivity to the fabric modules.

The 14.4 Tbps of capacity per packet processor is delivered over a total of 144 x 100G PAM SerDes lanes that can be run at 25G, 50G or 100G. Lanes may be be used as individual interfaces or combined in groups to provide flexible 25G, 50G 100G, 200G, and 800G interfaces.

As there are 144 PAM4 lanes, each Jericho3 packet processor supports up to a maximum of 144 logical or physical interfaces per chip, which defines the maximum possible port density for a given product form factor.

Some line cards employ gearboxes to increase the front panel interface density and maximize the capabilities by converting the 100G PAM4 SerDes lanes to more lanes at lower speeds and different encoding. Gearboxes enable an increased choice of interfaces without requiring additional packet processors, reducing overall system power consumption and heat generation. As the number of physical interfaces and supported breakout options is flexible, EOS provides tools to enable both configuration and analysis of the available port combinations for each platform.

Table 10.1: Arista 7800R4 Series Line card Module Port Characteristics

Output Port Characteristic	Maximum Packet Buffer Depth (MB)	Maximum Packet Buffer Depth (msec)
VOQ for a 10G output port	50 MB	40 msec
VOQ for a 25G output port	125 MB	40 msec

* Maximum port numbers are uni-dimensional, may require the use of break-outs, and are subject to transceiver/cable capabilities.

DCS-7800R4-36PE/DE

The 7800R4-36PE/DE is a 36 x 800G port line card with 28.8Tbps of switching capacity. The non-blocking throughput is achieved with two Jericho3 packet processors on the line card, as outlined in the architectural block diagram shown below.

For flexibility the 7800R4-36PE (OSFP) and 7800R4-36DE (QSFP112-DD) line cards are available in a choice of scale levels; Cluster Computing (R4C), Data Center scale (R4) and Large Scale (R4K), all line cards share a common physical architecture as outlined in the following section. TunnelSec encryption is supported on the R4K models, providing the choice of MACsec, VXLANsec or IPsec on all the 800G interfaces at wire-speed.

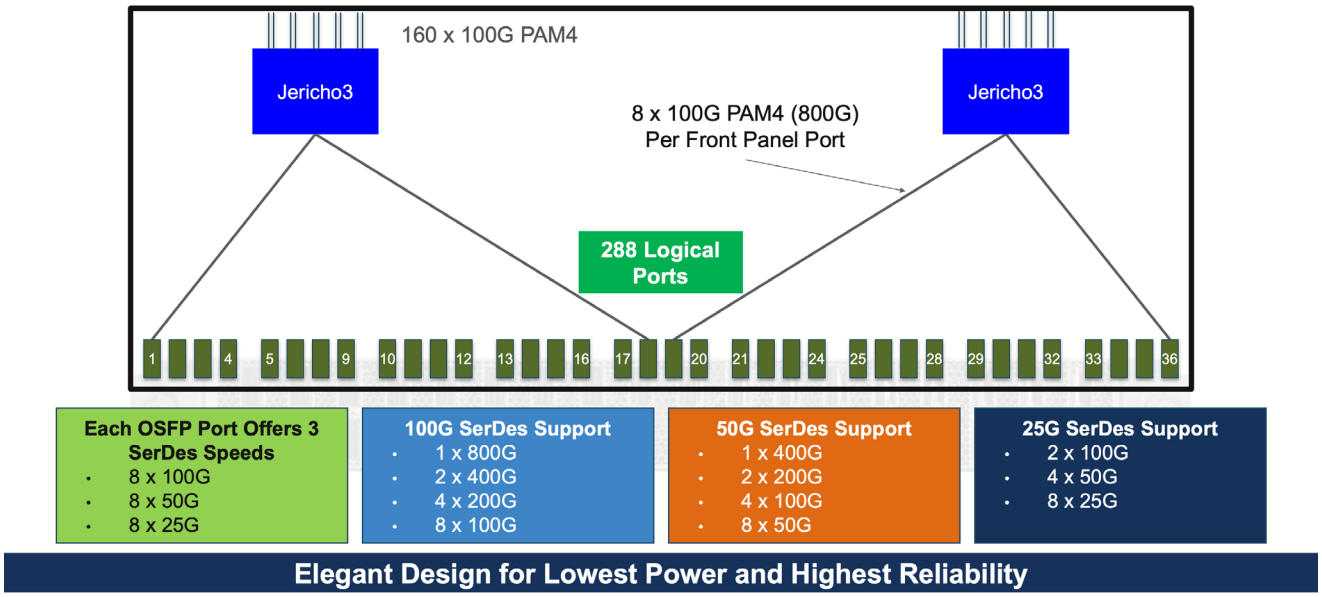


Figure 10.1.1: Arista 7800R4-36PE/DE--LC module architecture

Each of the two packet processors services a group of 18 front panel 800G QSFP-DD/OSFP ports. With each port supporting a range of optics available in the OSFP (7800R4-36PE-LC) or QSFP-DD (7800R4-36DE-LC) form-factors subject to the capabilities of the cable or transceiver.

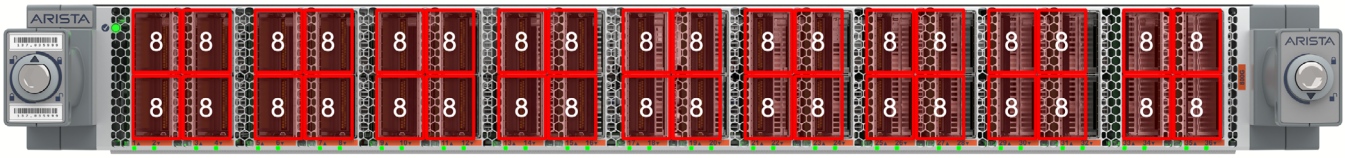
As illustrated in the architecture block diagram, the 800G OSFP/QSFP-DD front panel ports of the line card connect directly to one of the two packet processors via groups of 8 x 100G PAM4 lanes, with a total of 288 x 100G PAM4 lanes provided across the two processors. This design provides the potential for up to 8 logical interfaces per 800G port, with each group of 8 lanes forming a speed group capable of running at a SerDes speed of 25G, 50G or 100G. This allows each 800G port to be independently configured to operate at 1 x 800G-8/400G-8, 2 x 400G-4/200G-4/100G-4, 4 x 200G-2/100G-2/50G-2 or 8 x 100G-1/50G-1/25G-1. This flexibility provides the following connectivity choices on the 7800R4-36PE/DE line cards:

Table 10.1.1: 7800R4-36PE/DE 800G OSFP/QSFP breakout speed options

Interface Type	SerDes Speed	Max per OSFP/QSFP port	Port numbers	Max Density
25G	25G	8	[1-32] / [1-8]	256
50G-2	25G	4	[1-32] / [1,3,5,7]	128
50G-1	50G	8	[1-32] / [1-8]	256
100G-4	25G	2	[1-32] / [1,5]	64
100G-2	50G	4	[1-32] / [1,3,5,7]	128
100G-1	100G	8	[1-32] / [1-8]	256
200G-4	50G	2	[1-32] / [1,5]	64

Contd. Table 10.1.1: 7800R4-36PE/DE 800G OSFP/QSFP breakout speed options				
Interface Type	SerDes Speed	Max per OSFP/QSFP port	Port numbers	Max Density
200G-2	100G	4	[1-32] / [1,3,5,7]	128
400G-8	50G	1	[1-32] / [1]	32
400G-4	100G	2	[1-32] / [1,5]	64
800G-8	100G	1	[1-32] / [1]	32

The 32 x 800G front panel ports are directly connected to the packet processor, via 256 x 100G PAM4 lanes. This means each 800G port is consists of 8 x 100G PAM4 lanes to the chip, enabling each 800G port to be independently configured to operate at 1 x 800G-8/400G-8, 2 x 400G-4/200G-4/100G-4, 4 x 200G-2/100G-2/50G-2 or 8 x 100G-1/50G-1/25G-1. This flexibility provides the following connectivity choices:



Speed Group (Max 1 SerDes Speed)

Port Type	Practical / Max Density	Comment
25G	288	Using all 800G ports with breakouts
50G-2	144	Using all 800G ports with breakouts
50G-1	288	Using all 800G ports with breakouts
100G-4	72	Using all 800G ports with breakouts
100G-1	288	Using all 800G ports with breakouts
200G-4	72	Using all 800G ports with breakouts
200G-2	144	Using all 800G ports with breakouts
400G-8	36	Using all 800G ports with 400G transceivers
400G-4	72	Using all 800G ports with breakouts
800G-8	36	Using all 800G ports

Figure 10.1.2: Arista DCS-7800R34-36PE/DE--LC default breakout capabilities (indicates not all signaling lanes are used)*

The port numbering scheme of the line card is left to right, top to bottom for the 800G ports, meaning the top left 800G QSFP port is port 1 and the bottom right port is port 36. If an individual port is broken-out, then the [1-36] / [1-8] schema is used to identify a specific interface of the physical port.

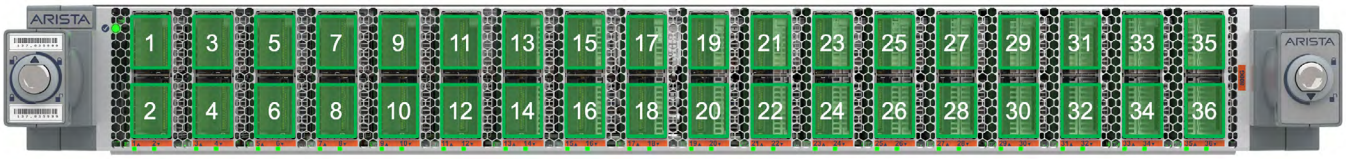


Figure 10.1.3: Arista DCS-7800R34-36PE/DE--LC port numbering scheme

Transceiver Support

Each port is capable of supporting a variety of DAC, AEC, and AOC cables as well as a broad range of compatible optics available in SFP, QSFP or OSFP form-factors. Further details of available transceiver types and any restrictions are available in the Transceiver Guide and Transceiver Datasheet, both published on arista.com.

800G OSFP ports support backwards compatibility with 400G OSFP transceivers, or QSFP28 transceivers through an OSFP-QSFP adapter.

800G QSFP-DD ports support backwards compatibility with 400G QSFP-DD transceivers, or QSFP56/QSFP28 transceivers.

Summary

The R4 series builds on a highly successful architecture, deployed globally in mission critical environments, with a range of solutions supporting 1GbE to 800GbE connectivity, with Internet scale service capabilities and next generation packet processing functionality at the optimum intersection of performance and power utilization, while always prioritizing quality and reliability.

Purpose-built for the highest performance environments including Cloud data center, AI/ML, Content Delivery, Service Provider, IP storage, Enterprise leaf and spine networks, Data Center Interconnect and IP Peering, the 7800R4 series provides operators with a proven, industry leading, platform to evolve their network capabilities and deploy a common architecture in an increasing set of network roles.

Arista's EOS network operating system continues to lead the industry in openness, extensibility and software quality. EOS has been leading the industry in telemetry innovations through the availability of NetDB and enabled operators to truly automate their network deployments through rich programmatic interfaces and support for industry standards such as OpenConfig.

Combined with Arista EOS, the 7800R4, 7280R4 and 7020R4 series are the ideal foundations for high value, large scale, modern networking.

Santa Clara—Corporate Headquarters
5453 Great America Parkway,
Santa Clara, CA 95054

Phone: +1-408-547-5500
Fax: +1-408-538-8920
Email: info@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, 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