Architecting the Scale-across AI fabric
Expanding the boundaries of AI networking

Driven by the relentless evolution of Artificial Intelligence (AI), frontier training continues to push the development of Large Language Models (LLMs) containing hundreds of billions to trillions of parameters. Scaling this immense processing power has traditionally relied on a centralized approach within a single data center, leveraging Scale-up fabrics for intra-rack XPU communication and Scale-out fabrics to create processing clusters of 1000s of XPUs across racks within a single location. However, as parameter sizes continue to grow, the data center facilities themselves with regards to power, cooling, space, and availability begin to become the limiting factor rather than the XPU or the network. In tandem with AI training challenges, AI inference is undergoing a parallel shift towards distributed execution to enhance user experience, performance and provide data sovereignty. Transitioning from a monolithic, cloud-centric architecture, modern inference architectures are now looking to leverage a distributed compute model spanning hyperscale clouds, regional data centers, and edge nodes. By hosting each stage of the inference pipeline where it is most operationally effective and cost-efficient, organizations avoid the latency and bandwidth overhead of routing every request back to a central AI facility.

This technical whitepaper explores Arista's REACH approach to building an open high-performance intelligent Scale-across fabric. Designed to resolve the distinct demands of both distributed AI training and inference, this industry leading solution leverages Arista's R4 routing portfolio powered by the EOS operating system and automated through Arista's CloudVision platform. This ensures operational and automation consistency with comprehensive end-to-end telemetry across all three fabrics (Scale-up, Scale-out, and Scale-across) and the Frontend network.

The demand for Scale-across fabrics

When designing the Scale-across fabric, operators are looking to accommodate two highly divergent use cases that place contrasting demands on the network. Distributed model training necessitates a robust, high-performance interconnect optimized for lossless, low-latency, and high-throughput data exchange to maintain synchronization between dispersed sites. Conversely, distributed real-time inference shifts the operational focus to strategic compute placement nearer the user and data source. This minimizes user-perceived latency and ensures strict data sovereignty, mandating advanced features such as workload-aware routing, robust encryption for data security and multi-tenancy across shared infrastructure.

Distributed AI training

To effectively distribute the immense computational processing power required for LLM training across XPUs, a hierarchy of nested parallelism strategies, termed hybrid parallelism, is a common approach. Tensor parallelism (TP) partitions the individual layers of the model across XPUs within the same Scale-up domain, generating extremely high-bandwidth, microsecond-sensitive traffic that never needs to leave the domain. Data parallelism (DP) runs multiple independent replicas of the model in parallel across XPUs, each fed different batches of input data. At the end of each step, gradients are synchronised across the XPUs, generating sustained all-reduce traffic patterns. Pipeline parallelism (PP) is the outermost dimension of the hierarchy, splitting the training model by layers into sequential stages. Each stage runs on a different XPU. Input data is divided into micro-batches that flow through the pipeline. While one batch is processed in stage 1, another can be processed in stage 2, maximizing utilization. Since communication between pipeline stages is bidirectional point-to-point exchanges of activations and gradients, and not an All-Reduce/All-Gather collective, sequential stages can be distributed across disparate Scale-out fabrics interconnected via a Scale-across fabric. Consequently, it is these deterministic point-to-point traffic patterns of pipeline parallelism that a Scale-across fabric needs to be specifically engineered to address and facilitate.

Model	Traffic Pattern	Network Requirements	Fabric Component
Tensor Parallelism (TP)	High-Bandwidth flows between XPUs within an AI Server	High-bandwidth and ultra low latency	Scale-up
Data Parallelism (DP)	Highly synchronised, All-Reduce and All-to-All bursty traffic patterns across XPUs within an AI backend fabric	Lossless, non-blocking with consistent low latency to reduce JCT	Scale-out
Pipeline Parallelism (PP)	Deterministic high-bandwidth point-to-point between between pairs of XPUs	Guaranteed bandwidth SLAs, with predictable latency and congestion handling	Scale-out/Scale-across

Table 1: Parallelism models utilised with distributed AI training models

While hierarchical communication collectives can reduce the amount and sensitivity of flows to latency-bound communications, the Scale-across challenges can't all be addressed solely at the communication collective, or rely only on the congestion algorithms of end node's NIC. The underlying network architecture needs to be programmed and tuned to take into account the available bandwidth and path latency with embedded congestion control mechanisms to minimise and prioritize the workloads during transient periods of oversubscription. This all has to be achieved with non-blocking performance with the capability of guaranteeing data integrity as it moves across administrative boundaries.

Distributed AI inference

Distributed inferencing represents a pivotal shift in the AI infrastructure architecture aimed at optimizing performance and user experience, while maintaining data sovereignty. Moving away from a traditional central cloud execution approach to a hierarchical resource model spanning hyperscale clouds, regional data centers, and edge nodes. This design places each phase of the inference pipeline where computation is most effective and cost-efficient, rather than routing every query to a centralized AI data center. Positioning compute resources closer to data sources addresses significant bottlenecks in preprocessing and embedding while minimizing bandwidth overhead. Additionally, distributing computational workloads across smaller facilities helps mitigate facility constraints related to power and cooling.

Inference is highly sensitive to response delay, where one primary performance KPI is Time to First Token (TTFT). The Scale-across fabric is therefore required to become an active component of the AI infrastructure, providing advanced workload-aware routing, that is telemetry driven, ensuring queries are directed to the correct resource pools based on locality and latency. With inference queries and potentially proprietary data now exiting the confines of the AI data center, encryption also becomes a mandatory requirement within each hop of the fabric. To maximize infrastructure efficiency, these capabilities must be delivered within a multi-tenant framework that guarantees strict, clear isolation of distinct customer inference workloads.

Extending the REACH of AI workloads

Scale-across will require the most sophisticated set of software features out of any of the AI fabrics, it demands the throughput characteristics of the Scale-out fabric but across multiple disparate sites. This geographic dispersion alongside the strict performance requirements, introduces a new set of complexities on the platform architecture, and a new set of baseline software features. To build the Scale-across fabric for distributed AI training and inference workloads, Arista has taken a holistic approach, unifying hardware innovation with a rich intelligent software toolkit that will allow operators to design for high scale while maintaining consistent performance, visibility and availability. This is captured in Arista's REACH framework:

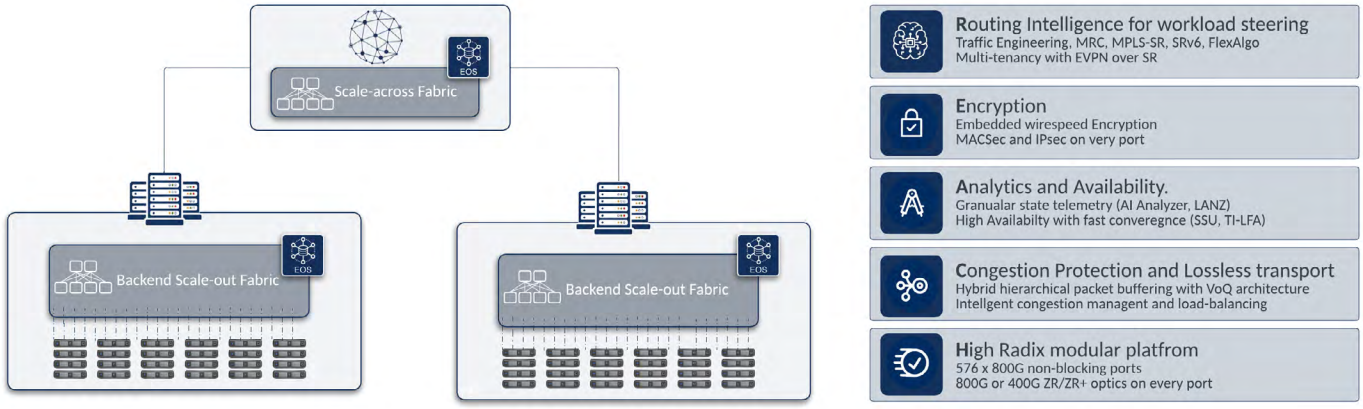

Figure 2.1 Arista's REACH tool-kit for building the Scale-across AI fabric

- **R**outing Intelligence with traffic engineered SLA Paths: Interconnecting geographically dispersed AI data centers for training and resource pools within a distributed inference pipeline, precludes the ability to build end-to-end non-blocking architecture due to the constraints of available fibers and paths. Consequently, with the potential for oversubscription within the topology, the Scale-across fabric needs to provide intelligent workload aware traffic steering that delivers a guaranteed service level agreement (SLA), regarding bandwidth and latency for both training and inference workloads.
- **E**ncryption: As data is now flowing outside the confines of the data center, potentially crossing administration domains, regional and national boundaries, it becomes vulnerable to snooping eyes and malicious actors. Wire-speed, embedded hardware-based encryption becomes mandatory at each hop, on every port of every node within the fabric.
- **A**nalytics & Availability: To ensure workload SLAs can be met there is a need for real-time visibility into the network state; ports, packet queues, traffic flows, buffer utilization, and path latency across the entire Scale-across fabric. This actionable intelligence enables dynamic workload steering along optimal paths during both steady state and failure events. At the same time, when an XPU or resource pool encounters a failure, combining real-time root-cause analysis with swift network recovery using Smart System Upgrades (SSU) creates a closed-loop environment. This proactive and rapid remediation mitigates the impact of failures, helping minimize overall job completion times

- **C**ongestion Management: When AI workloads are placed on the Scale-across fabric, providing guaranteed performance SLAs doesn't eliminate the potential for traffic bursts resulting in transient periods of over-subscription. The network nodes within the fabric, therefore, still need to provide a Congestion Management solution, with hierarchical, deep packet buffering to avoid packet loss during any congestion events. Dropping packets is far more negatively impactful to net job completion time (JCT) than any increased latency due to transient packet buffering. It's basic job (completion) insurance. This ensures against any packet loss under abnormal scenarios for the lowest job completion time (JCT) at global scale.
- **H**igh Radix: A critical aspect of the Scale-across architecture is the need for high-speed interconnectivity across geographically dispersed locations, potentially stretching beyond 100KM+. This necessitates the need for dense 400G and 800G ports with power efficiency to enable the deployment of long-reach ZR and ZR+ optics across all ports. Arista's R4 portfolio of fixed and modular platforms rise to the occasion to not only enable local capacity but also to seamlessly expand AI fabrics to remote data centers.

One Operational and Automation framework

While the Scale-across fabric demands a distinct set of hardware and software capabilities compared to the AI fabrics within the data center, Arista's consistent architectural framework ensures it does not turn into a separate operational silo. This is accomplished by powering the Scale-across fabric with the same EOS operating system and automation toolkit utilized across the frontend network and the Scale-up and Scale-out fabrics within the data center.

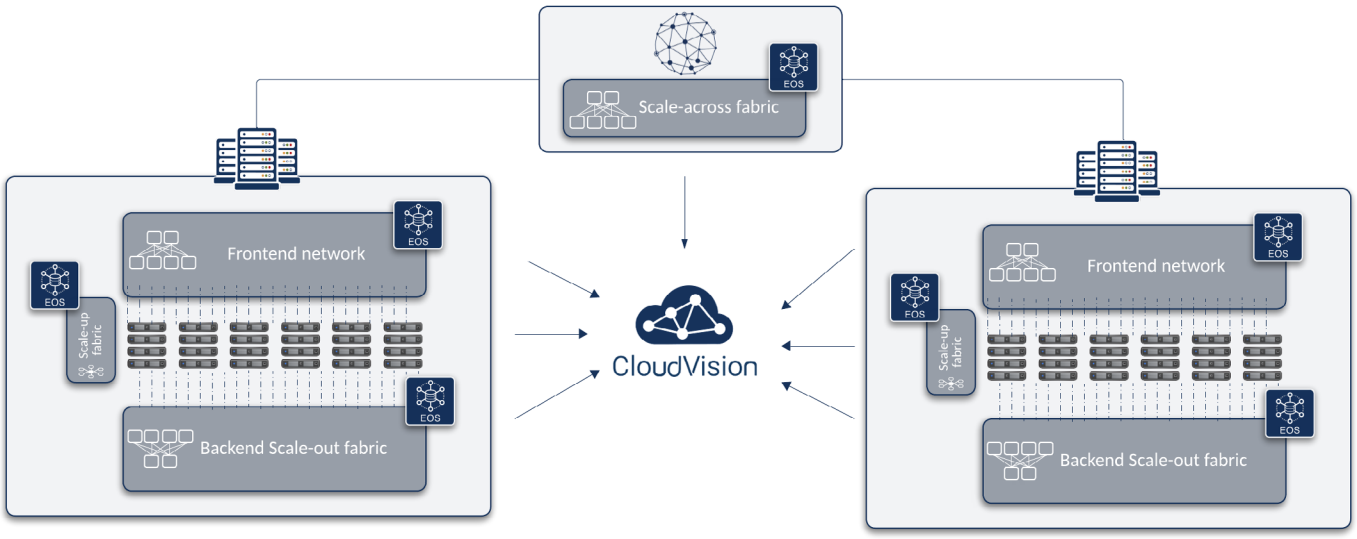

Figure 2.2: Arista's unified operation, automation and visibility framework for the AI infrastructure

This enables the same automation, orchestration, and telemetry tools used within the data center's Frontend, Backend Scale-up, and Scale-out fabrics to be seamlessly extended throughout the Scale-across infrastructure. Consequently, this unified management approach provides end-to-end operational visibility while simplifying management across the entire infrastructure.

Arista's R4 routing platform

The fundamental requirement of the Scale-across fabric is enabling the secure and intelligent forwarding of latency-sensitive, bandwidth-intensive AI workloads between geographically dispersed locations. Arista's R4 routing portfolio is designed from the ground up to meet and exceed these stringent AI demands and is the core hardware platform within its REACH approach to building the Scale-across fabric.

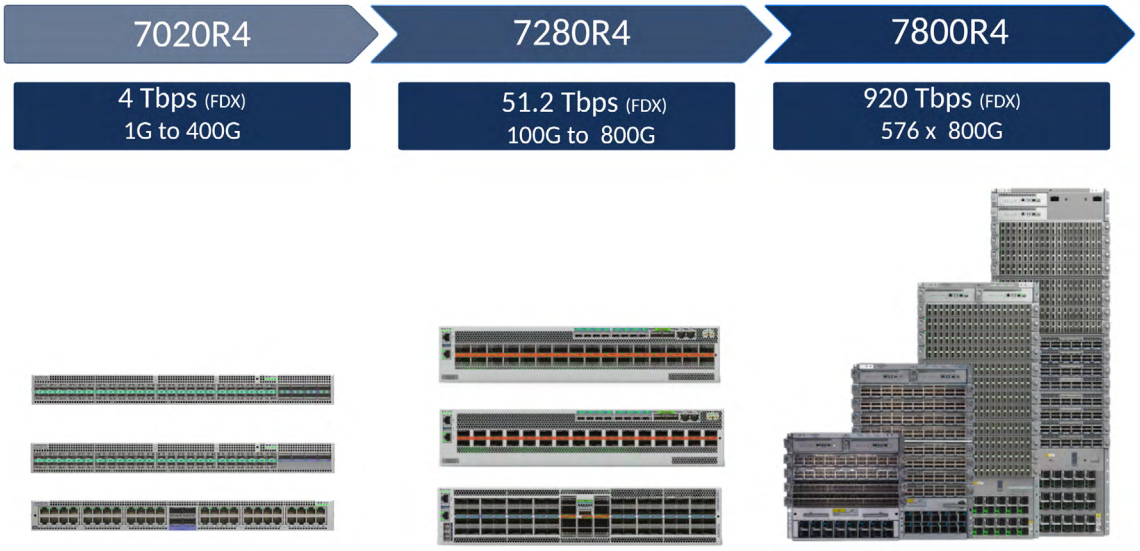

Figure 3.1 : Arista 7800R4/7280R4 and 7020R4 routing portfolio

Arista's R4 series is a portfolio of fixed and modular platforms built on a single ASIC architecture to ensure consistent performance and routing features across the portfolio. The 7280R4 is the fixed configuration platform providing high-density 100G, 400G and 800G connectivity in a 1RU or 2RU form factor with up to 51.2Tbps (FDX) of throughput. These lower radix fixed configuration platforms are the ideal candidates for deployment at the edge or within a POP of distributed AI inference architecture. For higher density Scale-across build-outs the 7800R4 modular platforms (4-slot, 8-slot, 12-slot and 16-slot) built on the same hardware architecture provides up to 920 Tbps (FDX) of throughput with a radix of 576 non-blocking 800G ports (or 1,152 ports of 400 GbE) in a single 16-slot 32RU chassis.

Fully Scheduled Lossless Performance

Engineered specifically for Scale-across deployments, the R4 platform provides the high port radix and throughput required to build and grow the fabric. To handle the demanding workloads and traffic patterns seen within the Scale-across fabric, including bursty flows, fan-in traffic patterns, and transient oversubscription, it leverages a key set of architectural advantages to ensure optimal performance even under the most demanding workloads.

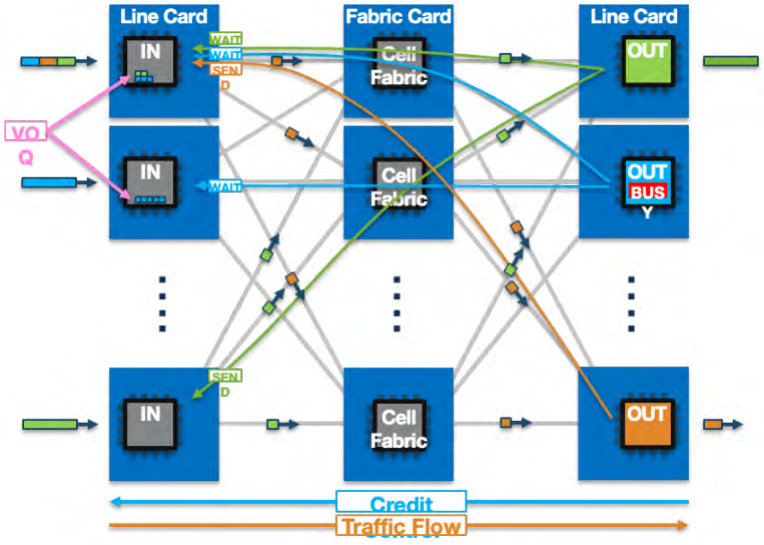

Figure 3.2 : Scheduled VoQ architecture of the Arista R4 routing platform

Virtual Output Queues (VOQ)

Within the Scale-across fabric transient contention events will occur, where workflows ingress across multiple inputs content for the same output port. The queue architecture of the routing platform now becomes a critical component within the Scale-across fabric, in order to avoid packet drops and any resultant latency spikes on the AI workload due to retransmissions.

The R4 platform addresses this design concern by implementing a Virtual Output Queue (VOQ) architecture. Instead of relying on fixed egress output queues, each input port on the platform has a set of logical virtual queues corresponding to every possible traffic class for every possible output port on the platform. These queues are independent, thus eliminating head of line blocking (HOLB), providing the ability for the R4 platform 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.

Fully Scheduled Fabric

To achieve the 800G port density demands of the Scale-across fabric while guaranteeing lossless transmission of packets from ingress to egress, the R4 platforms implement a fully distributed traffic scheduler. A continuous self-regulating distributed algorithm runs between all egress switch chips and all ingress switch chips. In one direction, this algorithm ensures that every egress port knows which ingress ports have waiting traffic. In the other direction the egress chip distributes credits to each waiting ingress chip in a fair and paced manner allowing traffic to flow without congesting the internal fabric of the switch or the egress port thereby avoiding the need to drop packets during a congestion event.

Cell Based Transmission

The final piece of the puzzle addresses efficient and fair utilization of the platform's internal switching fabric. Within the Scale-across fabric, traffic will arrive in uneven bursts, leading to 'microbursts' that cause link contention and load-balancing challenges of the internal switching fabric. In traditional multi-chip switches, these internal congestion events which often lead to packet drops are hidden from an operator, a critical disadvantage when you need to optimize for the strict demands of AI training and inference.

Arista's R4 architecture eliminates this issue entirely by combining the distributed queueing and scheduling described above with a cell based internal switching fabric. Instead of transmitting variable sized packets across the internal fabric, each packet is segmented into multiple equal sized cells which are sprayed across all available parallel paths through the internal fabric. The egress chip receives the cells and reassembles the packet for transmission out of the destination interface. The segmentation into cells, inbuilt load balancing and multi-pathing ensures it is possible to achieve 100% utilization of all internal fabric links and therefore 100% perfect load balancing without introducing packet reordering.

Embedded Encryption

As critical AI workloads are now flowing outside the confines of the data center, potentially crossing administration domains, and regional or national boundaries, data confidentiality becomes a mandatory requirement as traffic is routed through each hop within the Scale-across fabric.

Rather than push the encryption out to the endpoint, which becomes an unnecessary management overhead and an impossible challenge to achieve at scale with today's hardware, the R4 platform provides line-rate embedded MACsec (802.1AE) on every port. MACsec is performed at the interface (MAC layer) level, 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. This makes it the perfect fit for the routing architecture of the Scale-across fabric. Operating at the interface level all traffic on the wire is fully encrypted while working transparently to the upper transport layer which will be used for intelligent traffic steering.

Optical choice is critical

A final, vital component within the overall architecture is the choice and support of optics, as it will profoundly affect the design's cooling needs, power usage, cabling requirements, and the overall cost. The R4 platform addresses this common concern, with each

port flexibly supporting both ultra-low power linear drive pluggable optics (LPO) for connecting local clusters as well as ultra-long reach ZR/ZR+ (100KM+) coherent pluggable optics for interconnecting sites.

Analytics and Observability

A cornerstone of the Scale-across architecture is Arista's EOS operating system, unlocking the intelligent routing functionality of the R4 platforms while providing the granular real-time analytics required to validate AI performance expectations across the fabric.

EOS state-driven operating system

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 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 : Arista's EOS state-driven operating system

Operating purely as a state repository without embedded application logic, the in-memory NetDB database is dedicated exclusively to handle switch state. This means rather than being optimized for transactions, NetDB is designed for synchronizing state among processes. When a state change occurs within an agent, updates are then published to NetDB, which then in turn notifies any subscribed agents interested in the change. This centralized database approach with a publish and subscribe model for sharing state, simplifies inter-process communication to significantly reduce risk and error. This allows the hitless upgrade of individual agents and EOS images (SSU), which is critical to maximise uptime. This state-driven approach of the EOS architecture is also the foundation for providing real-time telemetry and visibility of all platform, software and network state across the Scale-across fabric.

Network Observability

The sensitivity of the AI workloads means even minor inefficiencies within the Scale-across fabric, such as an oversubscribed link, asymmetric path latency, or optic failures, can lead to severe financial consequences. In training environments, idle XPUs directly elevate Job Completion Time (JCT), while in inference deployments, delayed or dropped queries adversely impact essential Time to First Token (TTFT) metrics driving revenue losses.

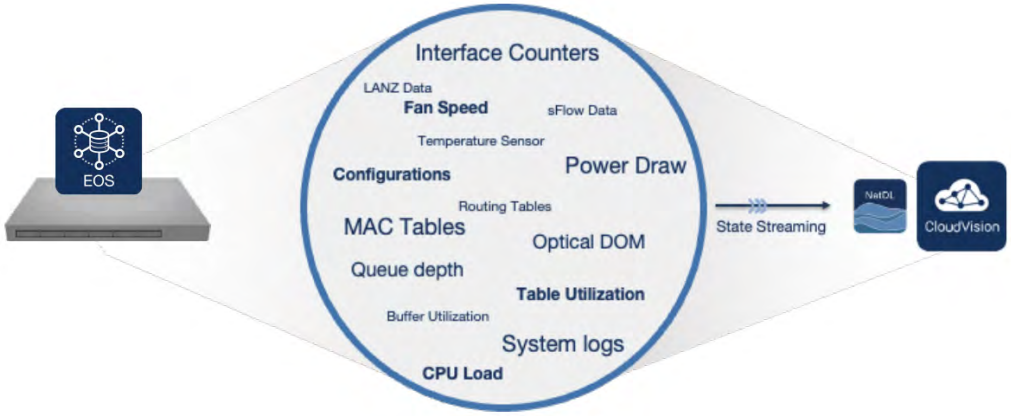

Figure 4.2 : Arista's EOS real-time state telemetry

But you can't optimize what you can't see. Achieving peak fabric performance requires a real-time, network-wide observability model capable of detecting bottlenecks and failures as they arise, enabling workloads to be dynamically reassigned when necessary. This is made possible in the Arista Scale-across fabric due to the state-driven EOS operating system running on each R4 platform. Each node within the fabric streams event-driven network state changes in real time via gNMI. Streaming network, software and platform state changes the moment they happen, provides administrators and orchestration tools immediate, actionable visibility into the overall health of the Scale-across fabric. As highlighted in the EOS toolset below, the state streaming delivers a comprehensive, granular visibility across every layer of the infrastructure. from real-time hardware and software metrics to routing tables changes and protocol updates.

- **Arista EOS AI Analyzer** monitors interface traffic counters at an ultra-granular, 100-microsecond resolution. This hardware-driven capability captures transient bursts and "incast" events, providing Equal-Cost Multi-Path (ECMP) member utilization analysis over sub-millisecond timescales. This level of precision allows operators to observe micro-variations in traffic distribution within the fabric missed by traditional tools.
- **Physical Layer Monitoring**, Layer 1 anomalies often manifest as unstable grey failures, resulting in intermittent packet drops with the fabric. With Arista's EOS interface telemetry and Digital Optical Monitoring (DOM), these grey failures can be proactively isolated and remediated against, with capability to steer traffic around any detected fault.
- **Switch Counter and Queue Monitoring** AI workloads flow through the network on microsecond and millisecond timescales, creating a massive visibility gap when counters are polled on intervals of seconds. Running EOS within the fabric solves this by providing hardware-accelerated monitoring tools that track switch queues and traffic counters at microsecond scales.
- **Latency Analyzer (LANZ)** provides real-time reporting on interface congestion and queuing latency and buffer utilization, allowing operations to shift from reactive troubleshooting to proactive fabric management. Correlating this LANZ telemetry with interface ECN, PFC, and queue drop counters reveals exactly how transient congestion impacts AI job performance within the fabric, enabling precise buffer optimization.

Workload Aware Routing

To provide intelligent workload aware routing, Arista's Scale-across architecture leverages Segment Routing over IPv6 (SRv6) on the R4 platform While an SR-MPLS approach would also be applicable for the fabric, the SRv6 approach can offer a number of benefits in the context of AI workloads.

- Cloud infrastructures natively support an IPv4/IPv6 forwarding plane, rather than MPLS. Therefore to connect to XPU resource pools within the Cloud, an MPLS-SR approach would require the complexity of an additional forwarding model (MPLSoIP) within the architecture. With an SRv6 forwarding plane this complexity can be removed.
- At large scale, the Scale-out fabrics of the AI backend network are transitioning to multi-planar topologies with packet spraying and SRv6 packet programming (i.e. MRC) to achieve optimal traffic load-balancing. Although a separate solution, building the Scale-across fabric with SRv6 offers a level of operational consistency between the two fabrics, while providing the ability to link the fabrics seamlessly in the future.

Segment Routing over an IPv6 forwarding plane (SRv6) provides a programmatic approach to traffic forwarding. End-to-end network paths are encoded in a series of network instructions (termed microSIDs) inside the IPv6 destination address of the packet along with an optional Segment Routing Header (SRH) if longer more explicit paths are required to be defined.

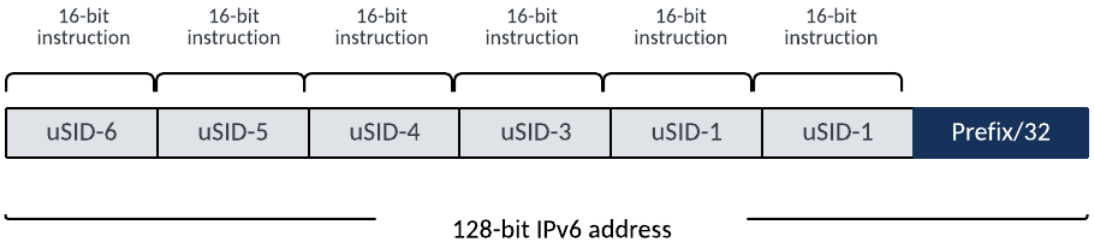

Figure 5.1 : SRv6 packet format with six microSIDs instructions

This programmable approach to traffic forwarding enables the forwarding of traffic at a highly granular level for inter-DC workloads in the case of distributed AI training or between resource pools within a distributed AI inference model, where individual links and nodes in the path can be chosen to meet the exacting latency and bandwidth requirements of the workload.

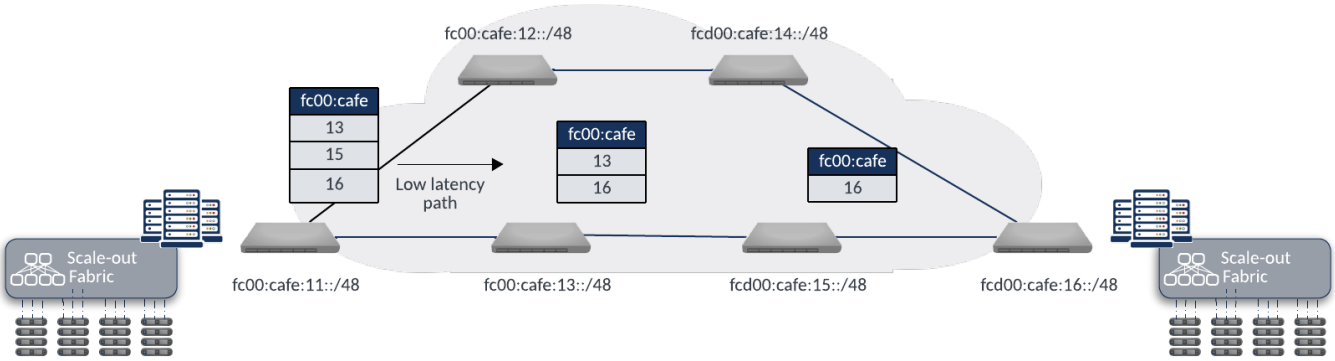

Figure 5.2 : SRv6 programmable workload forwarding with microSIDs

Workload Path Calculation

The calculation of the SRv6 path can be performed locally on the ingress R4 platform or alternatively externally with an off-box orchestrator which can consume the routing state from the R4 node and collect the streamed real-time telemetry from each node within the Scale-across fabric. The telemetry and routing state provides the orchestrator with end-to-end visibility of the fabric, allowing optimal placement of new workloads based on current fabric utilisation. Where there can be a need to co-ordinate the placement of both non-critical AI workloads and real-time AI flows on the fabric, both approaches could be combined.

The actual calculation of the optimal path for a specific workload is not a simple task, the workloads are no longer bound by the predictable constraints of the Scale-out fabric; instead, they now need to navigate multi-hop paths, spanning large geographic distances. Over these links, path bandwidth can vary and oversubscription can occur at any transit node, with the end-to-end latency highly dependent on the physical fiber lengths and the overall node count. To meet these challenges, and provide paths that meet the characteristic requirements of the workload, path selection moves beyond simple shortest-path routing in favor of a holistic, end-to-end view of network topology and real-time state, with the ability to dynamically monitor the latency of each individual link within the fabric.

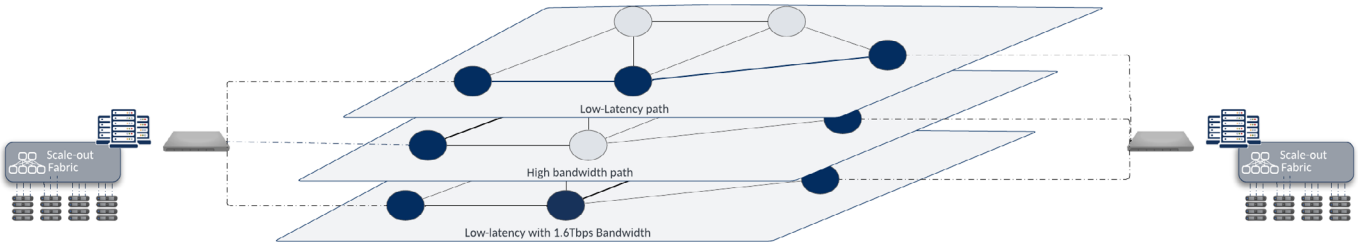

Figure 5.3: Logical network paths within Scale-across fabric

Whether through FlexAlgo definitions (FADs) or policies running on-box or paths calculated by an Orchestrator and pushed to the R4 platform, the SRv6 forwarding plane provides the ability to create multiple logical topologies within the fabric, each with their own unique set of constraints. The logical topologies could be used to define the lowest-latency path, highest-bandwidth paths, or even a unique set of diverse paths to provide workload resiliency. The forwarding requirement of the individual AI workloads can then be mapped to these logical topologies and monitored in real-time via telemetry, to create AI Workload aware forwarding across the fabric.

Fast Reroute

The sensitivity of the AI traffic means the delay incurred by the fabric detecting, re-converging and re-routing around a failure is not always a viable option. To achieve sub-50ms failover the network node needs to locally detect a failure event and re-act rather than wait for the network to slowly re-converge. TI-LFA (Topology Independent Loop Free Alternative) is an embedded component of the SRv6 approach, where the node closest to the failure, termed the PLR (Point of Local Repair) is responsible for detecting a local failure and switching traffic to a precomputed backup path. By locally switching over to a pre-computed backup path, rather than waiting for all nodes in the topology to re-converge, the TI-LFA approach can achieve rapid (~50ms) failover.

Multi-tenancy

The SRv6 forwarding plane can provide a programmable transport model for encapsulating and forwarding native IP AI workloads, but it can also be extended to multi-tenancy across the fabric when required. Multi-tenancy within the architecture is achieved using an EVPN control-plane over the SRv6 forwarding plane, allowing tenants to signal VPN connectivity across locations. This flexibility means single and multi-tenant AI work flows can be delivered within the same Scale-across fabric.

Adaptive congestion management

The traffic engineering (TE) capability of Arista's Scale-across architecture, enables paths to be computed across the fabric to meet the exacting constraints of each workload, however it doesn't negate the possibility of traffic bursts causing temporary periods of congestion on the links. The nodes within the fabric therefore need to collaborate with the traffic engineering approach to manage congestion. This involves providing visibility into active congestion hotspots, prioritizing AI flows, and absorbing traffic bursts to minimize packet drops and the resulting latency from packet retransmission.

The R4 platforms achieve this through the implementation of a hierarchical buffer architecture. In this model, buffer allocation is tunable and dynamic based on the specific requirements of the workload, allowing on-demand access to both a shallow and dynamic deep buffer pipeline. This is achieved through a 2-stage architecture; 1) An on-chip buffer for steady-state forwarding and 2) An on-package deep buffer for congestion management. This means during transient periods of congestion, AI workloads, where lossless transport is critical, can be allocated a higher portion of the buffer space to avoid packet drops. While this can result in a temporary increase in latency, it's a beneficial compromise for AI applications because packet loss, resulting in re-transmissions, will have a more significant effect on the overall job completion time and Time to First Token.

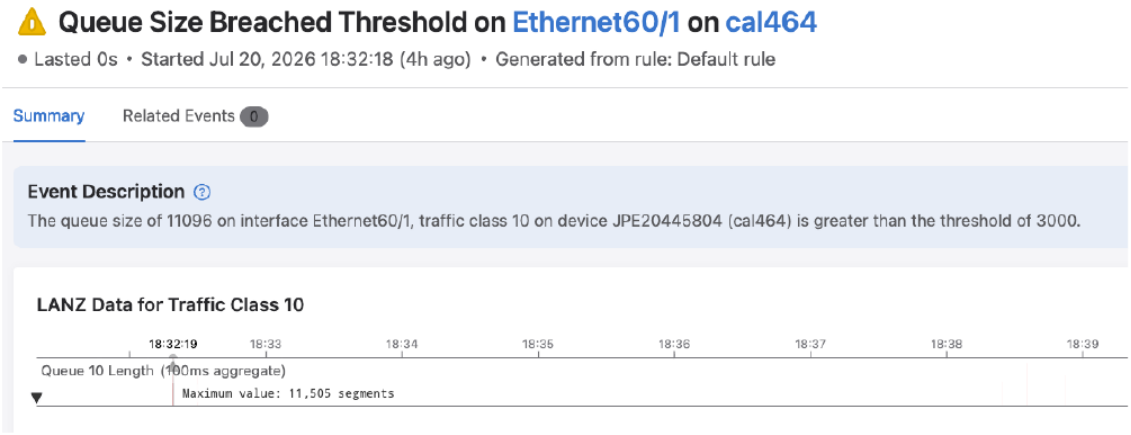

Figure 6.1: Real-time buffer depth visibility with LANZ

While the platform's buffer architecture is designed to absorb brief, transient traffic bursts, sustained or repeated bursts can indicate a more fundamental traffic load issue. This is where the telemetry component of architecture is critical. During congestion events ECN, latency-based ECN events along with LANZ metadata is streamed from the platform; tracking congestion notification events, queue depth, queue latency and any drops within the queue. Providing the visibility of hotspots within the fabric in real-time to allow workloads to be reallocated to less congested paths/links in the fabric.

Summary

Arista's Scale-across architecture is designed from the ground up to meet and exceed the demanding requirements of distributed AI training and inference. By combining the R4 routing platform's high-radix loss-less architecture and embedded encryption with the programmatic precision of an SRv6 forwarding plane, the solution provides workload aware routing across the fabric. Crucially, the solution avoids the creation of operational silos, by running the same EOS operating system deployed within Arista's the frontend network and Scale-up and Scale-out fabrics. Enabling the same automation, orchestration, and rich real-time telemetry tools to be seamlessly extended across entire AI infrastructure.

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