# Arista 7250X & 7300 Switch Architecture (‘A day in the life of a packet’)

The Arista 7250X and 7300 series are purpose built 10GbE/40GbE data center switches in four flexible energy efficient form factors, designed specifically to offer increased performance, scalability and density, without making any compromises to system availability, reliability or functionality.

The 7250X and 7300 series combine cutting edge hardware developments with the award winning EOS software to offer wirespeed layer 2 and layer 3 forwarding, advanced features for software defined cloud networking and support for the next generation of protocols such as VXLAN.

This white paper provides an overview of the switch architecture and the packet forwarding characteristics of the Arista 7250X and 7300 Series with X-Series linecards.

### Switch Overview

The Arista 7250X switch is a fixed configuration system, supporting a high density of 10 or 40 GbE interfaces in a compact power efficient 2RU form factor.

The Arista 7300 series are a purpose built range of modular chassis and a key component of the Arista data center product portfolio. When coupled with the X-Series linecards the 7300 series increases design flexibility and scalability in traditional MoR/EoR, Leaf and Spine and SplineTM deployments.

Both the 7250QX and 7300 Series are designed with a common architecture specifically to meet the challenges of dense 10 Gigabit and 40 Gigabit Ethernet switching. Featuring a flexible combination of both 10GbE and 40GbE interfaces in compact, low latency, energy efficient form factors, scaling from 64 x 40GbE to 512 x 40GbE.

The following table highlights the key technical specifications.

**Table 1: Arista 7250X and 7300 Series overview**

| Characteristic | 7250QX-64 | 7304 | 7308 | 7316 |
|---|---|---|---|---|
| Switch Height (RU) | 2 RU | 8 RU | 13 RU | 21 RU |
| Max SFP+ Ports | - | 192 | 384 | 768 |
| Max QSFP+ Ports | 64 | 128 | 256 | 512 |
| Maximum System Density 10GbE ports | 256 | 512 | 1024 | 2048 |
| Maximum System Density 40GbE ports | 64 | 128 | 256 | 512 |
| Maximum System Throughput (Gbps) | 5.12Tbps | 10Tbps | 20Tbps | 40Tbps |
| Maximum Forwarding Rate (PPS) | 3.84Bpps | 7.5Bpps | 15Bpps | 30Bpps |
| Latency | 550ns – 1800ns | 2us | 2us | 2us |
| Packet Buffer Memory | 48MB | 96MB | 192MB | 384MB |

### Cloud Technology Shift

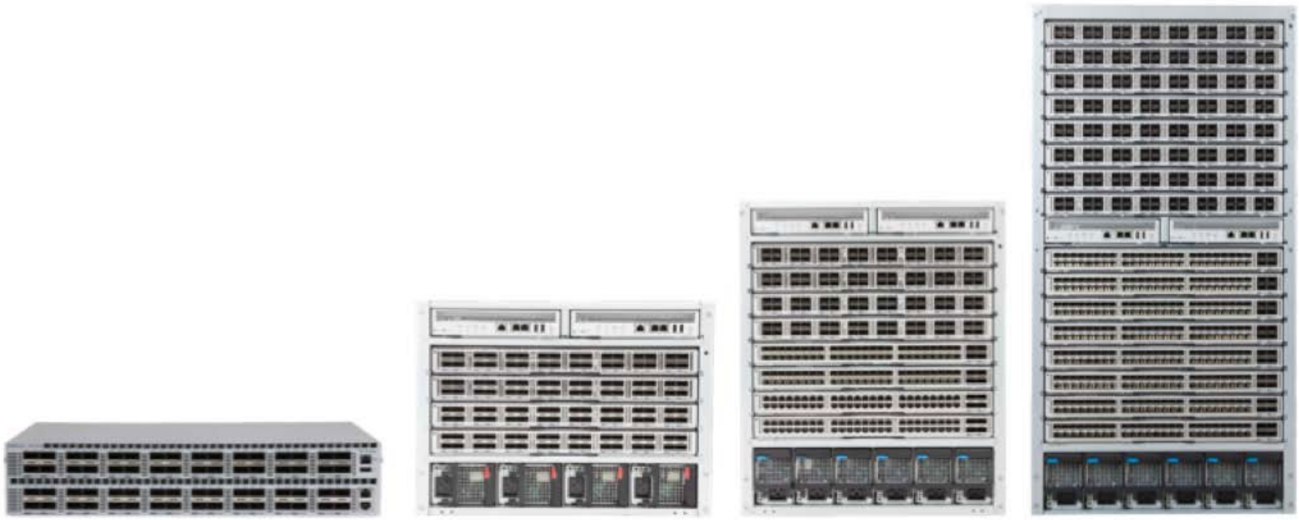

Figure 1: Left to Right: 7250QX-64, 7304, 7308 and 7316

Increased adoption of 10 Gigabit Ethernet servers coupled with applications requiring higher bandwidth is accelerating the need for dense 10 and 40 Gigabit Ethernet switching. The 7250X and 7300 Series meet this demand with a choice of four high density wirespeed layer 2/3 form factors ranging from 2RU to 21RU that offer a flexible combination of 1G, 10G and 40G switching.

The high interface density, coupled with the advanced features and functionality supported natively at the hardware level make the 7250X and 7300 series ideal for a number of deployment scenarios inside the data center, either as part of a 2 tier leaf/spine or a single tier Arista Spline architecture. While 2 tier designs allow greater deployment flexibility for large or hyper-scale deployments, supporting MLAG, ECMP or L2 over L3 approaches, a collapsed leaf and spine or Spline solution offers a compelling option for more condensed deployments. The Spline design collapses both the server leaf and spine switching tiers into a single Spline tier, reducing the number of devices required. A Spline based design will always offer the lowest capital and operating expenses, the lowest latency and the best performance with just two points of management, no ports wasted on switch interconnects, and a design that is inherently non-blocking.

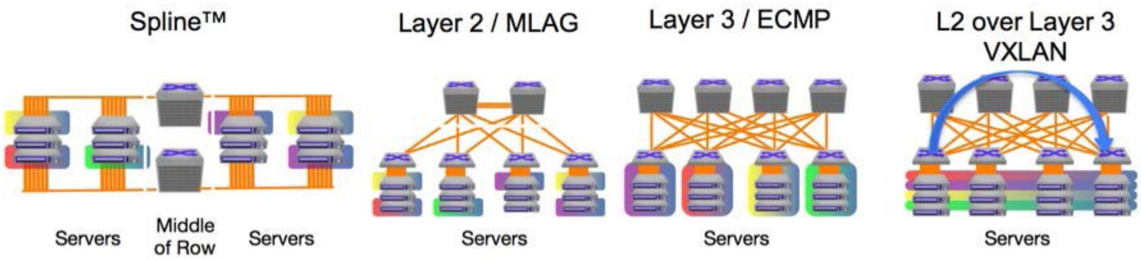

Figure 2: 7250X and 7300 series deployment scenarios

With typical power consumption under 12 watts per 40GbE port the 7250X and 7300 Series provide industry leading power efficiency coupled with power supplies rated at the platinum level with 93% efficiency. All models offer a choice of airflow direction (front to rear or rear to front) to support deployment in either hot aisle / cold aisle environments, traditional middle and end of row designs or at the network spine layer. An optional built-in SSD enables advanced capabilities for example long term logging, data captures and other services that are run

All models within the Arista 7250X and 7300 series share a common system architecture with the Arista 7050X series, built upon the same Switch on Chip (SOC) silicon. As a result all of the 7050X, 7250X and 7300 series share a common set of software and hardware features and key data center capabilities such as flexible table sizes and support for VXLAN VTEP. While the 7050X series use a single SoC per system, the 7250X series switch and 7300X series linecards implement a multi-chip solution to significantly expand the interface density.

Built on top of the same industry defining EOS image that runs on the entire Arista product portfolio, the Arista 7250X and 7300 Series deliver advanced features for big data, high performance compute, cloud, virtualized and traditional network designs.

**Multichip ASIC Configurations**

The 7250X and 7300 series use an optimized 'Internal CLOS' design with multiple Port ASICs interconnected via Fabric ASICs in an efficient non-blocking two-tier design. Internal links are optimized for switch-switch interconnection and integrate seamlessly without the need for multiple planes of management. This approach allows the creation of switching platforms with significantly higher density and forwarding capacity than a single SoC system.

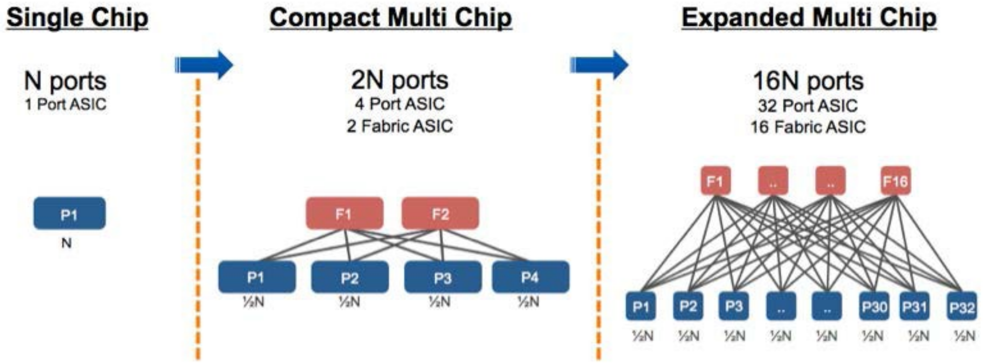

Figure 2: 7250X and 7300 series deployment scenarios

These larger density multi-chip switches enable the design of both larger two tier Leaf/Spine and single tier Spline networks, that offers significant cost reduction in terms of both capital and operational expenses by minimizing the number of tiers required to support a given number of hosts. This in turn reduces the physical and power footprint of the network by decreasing the required number of devices and eliminating external connections between them.

The interconnection between port ASICs on the 7250X and 7300 series platforms are built around a packet based fabric. In such a fabric, a packet is sent over a single fabric link in entirety, avoiding the need to fragment and reassemble. This approach reduces latency to a level similar to that seen in single system top of rack switches, while still providing modular system port density.

Many traditional internal CLOS switches and packet-based fabrics rely on a set of hashing algorithms to evenly distribute frames crossing the fabric. While the efficiency of such algorithms vary widely, no available algorithm claims to support 100% efficiency, which has typically resulted in switching fabric performance significantly less than the theoretical capacity.

This limitation inherent in previous CLOS based systems has been eliminated with the Arista 7250X and 7300 series switches, which leverage Dynamic Load Balancing Fabric (DLBF) technology. DLBF actively monitors internal port-fabric links and both allocates new flows and rebalances existing flows to fabric links with the lowest utilization, providing an efficient and well-balanced distribution of load over all links at all times, without the need for dividing packets into smaller segments, adding additional internal headers, or using fabric over-speed techniques to compensate for the hashing algorithms inherent inefficiency.

To provide deployment flexibility, the fabric used in 7250X and 7300 series can be configured to operate in two modes:

*Performance fabric mode* is aimed at environments where it is anticipated that interface utilization may consistently remain high at up to 100% on most interfaces. Unlike the single chip implementations, a multi-chip switch uses an internal fabric, with a header, where the additional overhead prevents the switch from forwarding at line-rate. Performance mode enables line rate performance by increasing the speed of the fabric interfaces to offset the internal fabric header. As a result of the speed differences between the ingress ports and the fabric links, in performance mode all forwarding takes place in a store-and-forward mode.

*40GbE low latency fabric mode* is provided for environments where minimizing the latency of 40GbE to 40GbE traffic is vital, such as financial trading environments. In this mode the internal fabric links are run at 40Gb, the same speed as the ingress 40GbE interfaces, which enables cut-through switching for all ingress ports running at 40GbE.

## 7250X & 7300 Series Architecture

All stages of the packet forwarding pipeline are performed entirely in the hardware/dataplane. The forwarding pipeline is a closed system integrated on the packet processor (PP) of each SoC. All packet processors are capable of providing both the ingress and egress forwarding pipeline stages for packets that arrive on or are destined to ports located on that packet processor. In multi-chip systems each packet processor can perform local switching for traffic between ports on the same packet processor.

### 7250QX Series

The 7250QX-64 is a 2 RU system, with 64 QSFP+ interfaces supporting up to 64x 40GbE. Each of the 64 QSFP+ interfaces can be used as 40GbE interfaces or each QSFP+ port can be individually enabled as 4 x 10GbE through the use of transceivers and splitter cables or copper cables, providing a maximum of 256 x 10GbE interfaces.

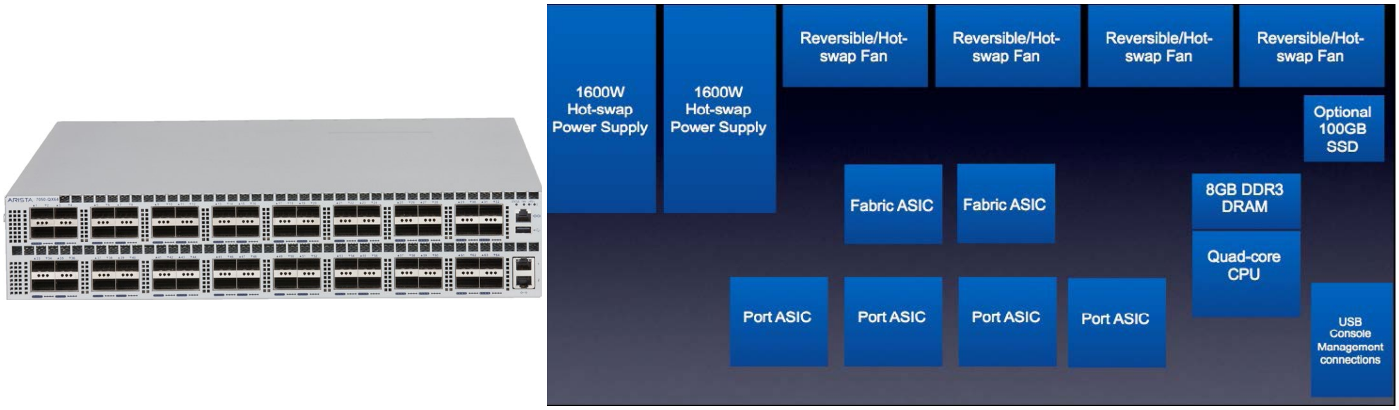

Figure 4: Arista 7250QX-64 (left: Physical Image, right: Logical device layout)

### 7300 Series

The 7300 series are a range of modular systems, with the choice of three 10GbE/40GbE linecards for high performance, low latency and scalable multilayer switching powered by Arista EOS, the worlds most advanced operating system. The high density of the 7300 series makes it a suitable deployment option for traditional middle/end-of-row aggregation, leaf/spine and Spline deployments.

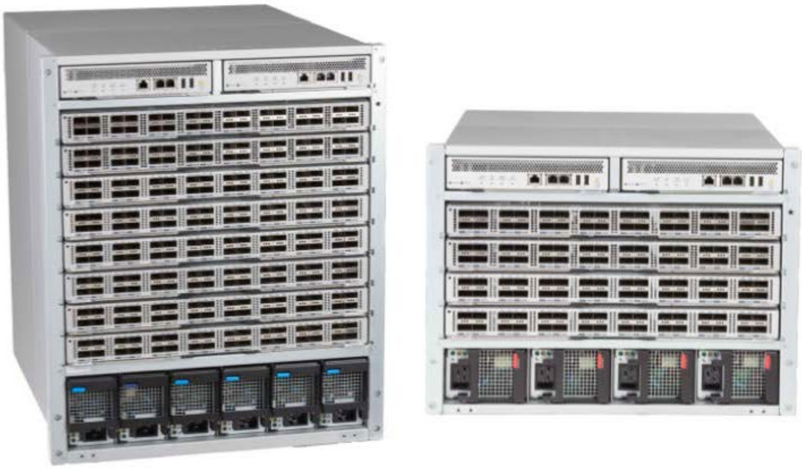

Figure 5: Left to right, DCS-7308-CH, DCS-7304-CH.

The 7300 series chassis is offered in 2 distinct form factors, the 7304 offers 4 linecard slots in an 8RU system and the 7308 offers 8 linecards slots in a 13RU system. Each chassis supports 2 supervisor modules, 4 fabric modules with integrated fans and fully redundant power supplies. The 7300 series supports the 7300X series of linecards which include QSFP+, SFP+ and 10GBASE-T models.

Figure 6: Arista DCS-7300X-32Q-LC

*32 port QSFP+ 40G Linecard for 10/40 GbE*

- 32 40GbE or 128 10GbE ports with QSFP+ optics and breakout cables.
- Choice of Copper, Multimode and Single-mode optics with both 10 and 40G options.
- 1.92Bpps and under 12W per 40G port.

*48 port 1/10G SFP+ and 4 port 40GbE QSFP+*

- Up to 64 10GbE ports per line card
- 48 1/10GbE SFP+ ports
- 4 QSFP+ ports allow flexibility of 4x 40GbE or 16 x 10GbE
- 960Mpps and under 3W per 10G port.

Figure 7: Arista DCS-7300X-64S

Figure 8: Arista DCS-7300X-64T

*F48 port 10GBASE-T for 1/10GbE and 4 port 40GbE QSFP+*

- Up to 64 10G ports per Linecard
- 48 1/10GbE BASE-T ports allow for extended reach UTP connectivity.
- 4 QSFP+ ports allow flexibility of 4 x 40GbE or 16 x 10GbE.
- 960Mpps and under 5W per 10G port.

The industry leading port density of the Arista 7300 series allows a dedicated QSFP+ form factor deployment of up to 256 40GbE interfaces which can be independently configured as up to 1024 10GbE links, or a mixed form factor deployment of up to 384 10GbE SFP+ or 10GBASE-T interfaces and 32 40GbE QSFP+ interfaces. All linecards can be mixed and matched in all three systems in any combination allowing the 7300 series to be customized to meet both the density and interface type requirements of any deployment.

The 7300 series feature a common architecture between all models, ensuring performance and feature parity. The key differences between the different capacity models relates to the number of PSUs (to support the system requirements), line cards and the number of Fabric ASIC per fabric card. All 7300 X-Series linecards share a common architecture avoiding the need to reference multiple data-sheets when developing a mixed deployment. The principle difference between linecard models is the number of port ASICs per card and the number and form factor of front panel interfaces.

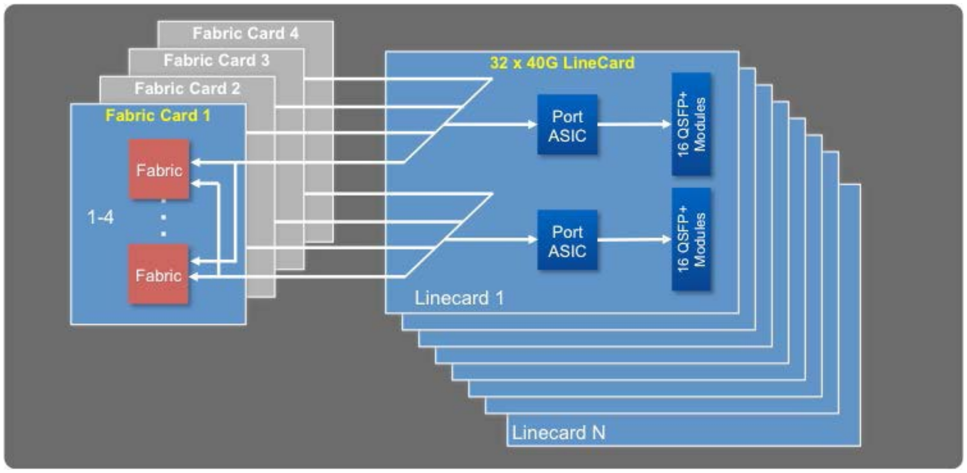

Figure 9: Arista 7300 series logical system overview

## Scaling the Data Plane

In addition to high port density, the Arista 7250X & 7300 series also incorporate significant increases in both forwarding table density and flexibility. While traditional switches statically allocate resources to specific functions such as MAC Address tables, or IPv4 Host routes, recognizing that no two deployments are identical the 7250X and 7300 series support a more flexible approach.

Forwarding table flexibility on the 7250X and 7300X Series linecards is delivered through the Unified Forwarding Table (UFT). Each L2 and L3 forwarding element has a dedicated table and can additionally have the tables sizes augmented by allocating a portion of the UFT. The UFT contains 256K entries from 4 banks, where each bank can be allocated to forwarding tables. Much wider deployment flexibility is achieved by being able to dedicate the entire UFT to expand the MAC address tables in dense L2 environments, or a balanced approach achieved by dividing the UFT between MAC Address and Host route scale. The UFT can also be leveraged to support the expansion of the longest prefix match (LPM tables) – (future).

**Table 2: Arista 7250X & 7300 Series Table Scale with UFT**

| Linecard Port Characteristics | DCS-7250QX-64 | DCS-7304 | DCS-7308 |
|---|---|---|---|
| MAC Address Table | 288K | 288K | 288K |
| IPv4 Host Routes | 208K | 208K | 208K |
| IPv4 LPM Routes | 144K * | 144K * | 144K * |
| IPv4 Multicast Routes | 104K | 104K | 104K |
| IPv6 Host Routes | 104K | 104K | 104K |
| IPv6 LPM Routes | 77K * | 77K * | 77K * |
| IPv6 Multicast Routes | 4000 | 4000 | 4000 |
| Packet Buffers | 48MB | 96MB | 192MB |
| ACLs | 16K Ingress<br>4K Egress | 32K Ingress<br>8K Egress | 64K Ingress<br>16K Egress |

*Roadmap Scale

## Scaling the Control Plane

The central CPU complex on the 7250X and Supervisor Module CPU on the Arista 7300 Series is used exclusively for control-plane and management functions; all data-plane forwarding logic occurs at the packet processor/Port ASIC level.

Arista EOS®, the control-plane software for all Arista switches executes on multi-core x86 CPUs with multiple gigabytes of DRAM. As EOS is multi-threaded, runs on a Linux kernel and is extensible, the large RAM and fast multi-core CPUs provide for operating an efficient control plane with headroom for running 3rd party software, either within the same Linux instance as EOS or within a guest virtual machine.

Out-of-band management is available via a serial console port and/or the 10/100/1000 Ethernet management interface. The 7250X & 7300 Series also offer USB2.0 interfaces that can be used for a variety of functions including the transferring of images or logs.

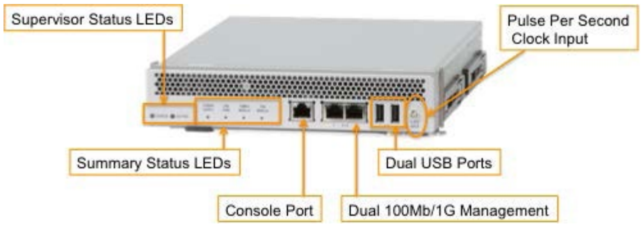

Figure 10: Arista 7300 Supervisor Module.

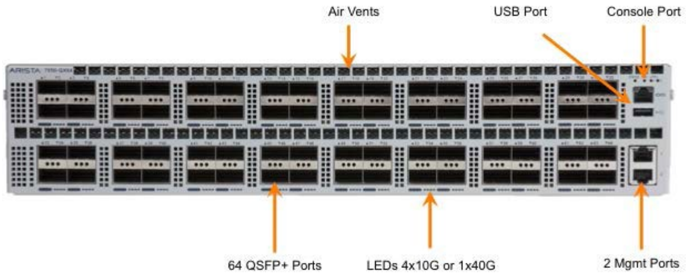

Figure 11: Arista 7250QX

### Multi-Chip Packet Forwarding Pipeline

Each packet processor is a Switch on a Chip (SoC) capable of providing both the ingress and egress forwarding pipeline stages for packets to or from the front panel ports. In a multichip system it is likely one packet processor provides the ingress stages of the forwarding pipeline, and a second packet processor provides the egress stages, however if the ingress and egress ports share a packet processor both sets of actions will take place locally and the forwarding pipeline will mirror the behavior described for 7050X single-chip systems.

The ingress packet processor is responsible for actions such as address learning, VLAN assignment, L2 and L3 forwarding lookups, QoS classification and ingress ACL processing. While the egress packet processor provides packet buffering, packet rewrite, egress ACL processing and multicast replication.

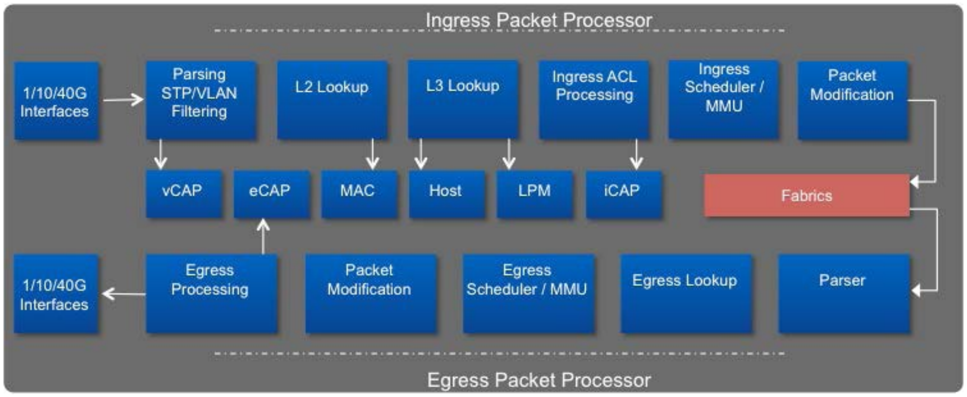

Figure 12: Packet forwarding pipeline stages inside a Arista 7250X / 7300X

## Stage 1: Network Interface (Ingress)

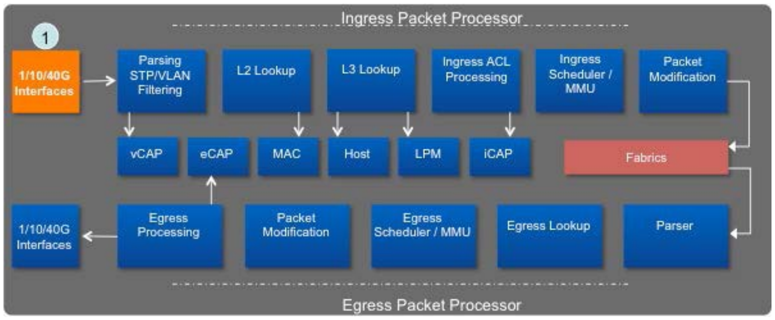

Figure 13: Packet Processor stage 1: Network Interface (Ingress)

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

The PHY layer is responsible for transmission and reception of bit streams across physical connections including encoding, multiplexing, synchronization, clock recovery and serialization of the data on the wire for whatever speed/type of Ethernet interface is configured. Operation of the PHY is in compliance with the IEEE 802.3 standard. The PHY layer transmits/receives the electrical signal to/from the transceiver where the signal is converted to light in the case of an optical port/transceiver. In the case of a copper (electrical) interface, e.g., Direct Attach Cable (DAC), the signals are converted into differential pairs.

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

### Stage 2: Ingress Parser

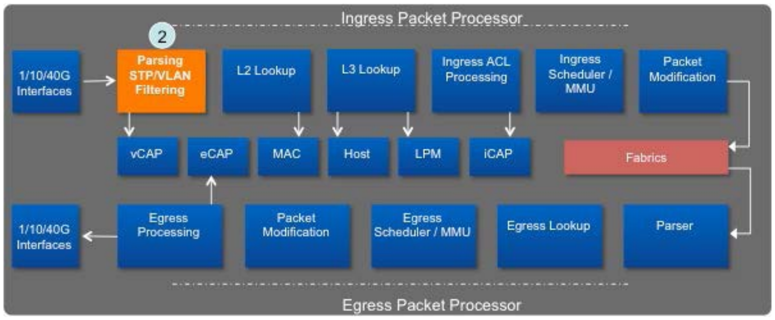

Figure 14: Packet Processor stage 2: Ingress Parser

The Ingress Parser represents the first true block of the forwarding pipeline. While the entire packet is received at the Mac/Phy layer only the packet header is sent through the forwarding pipeline itself.

The first step is to parse the headers of the packet and extract all of the key fields required to make a forwarding decision. The headers extracted by the parser depend on the type of packet being processed. A typical IPv4 packet would extract a variety of L2, L3 and L4 headers including the source MAC address, destination MAC address, Source IP, Destination IP and Port numbers).

The Parser will then determine the VLAN ID of the packet, if the packet arrived on a trunk port this can be determined based on the contents of the VLAN header. If the packet arrived on an access port, or arrived untagged the VLAN ID is determined based on the port configuration.

Once the Parser is aware of the VLAN ID and ingress interface it must verify the STP port state for the receiving VLAN. If the port STP state is discarding or learning, the packet is dropped. If the port STP state is forwarding no action is taken.

As a final ingress check the Parser will compare the packet against any configured Port ACLs by performing a lookup in the vCAP, the first of the three ACL TCAMs. If the packet matches a DENY statement it will be dropped. If the packet matches a PERMIT statement, or no port ACL is applied, the packet is passed to the next block of the pipeline.

### Stage 3: L2 Lookup

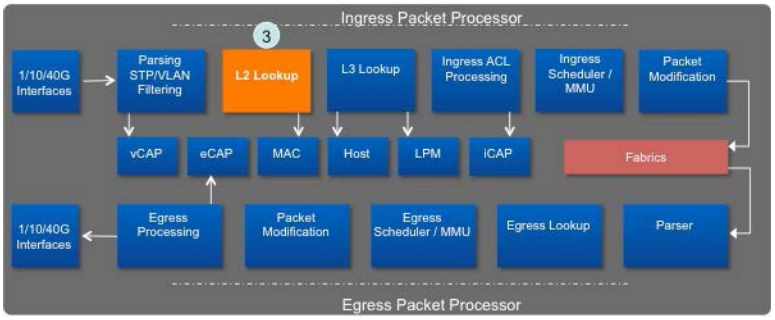

Figure 15: Packet Processor stage 3: L2 Lookup

The L2 Lookup block will access the MAC address-table (an exact-match table) and perform two parallel lookups.

The first lookup is performed with the key (VLAN, source mac-address), to identify if it matches an entry already known to the switch and therefore present in the mac-address table. There are three possible outcomes to this lookup:

- MAC address unknown, trigger a new MAC learn, mapping the source MAC to this port.
- MAC address known but attached to another port, triggering a MAC move and a reset of the entries' age.
- MAC address known and attached to this port, triggering a reset of the entries' age.

The second lookup is performed with the key (VLAN, Destination MAC address) this lookup has four possible outcomes:

- If the destination MAC address is a well known or IEEE MAC, trap the packet to the CPU. The system uses a series of hardware rate-limiters to control the rate at which traffic can be trapped or copied to the CPU.
- If the destination MAC address is either a physical MAC address or a Virtual (VRRP/VARP) MAC address owned by the switch itself, the packet is routed.
- If neither of the above is true but the MAC address-table contains an entry for the destination MAC address, the packet is bridged out of the interface listed within the entry.
- If neither of the above is true and the MAC address-table does not contain an entry for that MAC address, the packet is flooded out of all ports in an STP forwarding state within the ingress VLAN, subject to stormcontrol thresholds.

### Stage 4: L3 Lookup

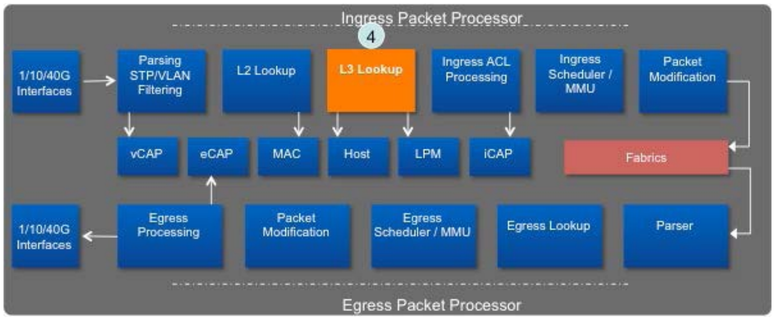

Figure 16: Packet Processor stage 4: L3 Lookup

The L3 Lookup stage performs sequential accesses into two distinct tables, each access includes up to two lookups. The first table is an exact-match table which contains /32 v4 and /128 v6 host routes. The second table is a longest-prefix match (LPM) table which contains all v4 routes and v6 routes shorter than /32 and /128 lengths respectively.

The first lookup into both the host route and LPM tables is based on the key (VRF, Source IP Address), this lookup is designed to verify that the packet was received on the correct interface (the best interface towards the source of the packet), if received on any other interface the packet may be dropped depending on user configuration. This lookup takes place only if uRPF is enabled.

The second lookup takes place initially in the host route table; the lookup is based on the key (VRF, Destination IP address) the purpose is to attempt to find an exact match for the destination IP address. This is typically seen if the destination is a host in a directly connected subnet. If an exact match is found in the host route table the result provides a pointer to an egress packet processor, physical port, an L3 interface and packet rewrite data.

If there is no match for the lookup in the host table, another lookup with an identical key is performed in the LPM table to find the best or longest prefix-match, with a default route being used as a last resort. This lookup has three possible outcomes:

- If there is no match, including no default route, then the packet is dropped.
- If there is a match in the LPM and that match is a directly connected subnet, but there was no entry for the destination in the host route table, the packet is punted to CPU to generate an ARP request.
- If there is a match in the LPM table, and it is not a directly connected subnet it will resolve to a single next-hop which will be located in the Host Route table. This entry provides an egress packet processor, physical port, L3 Interface and packet rewrite data.

The logic for multicast traffic is virtually identical, with multicast routes occupying the same tables as the unicast routes. However instead of providing egress port and rewrite information, the adjacency points to a Multicast ID. The Multicast ID indexes to an entry in the multicast expansion table to provide a list of output interfaces.

### Stage 5: Ingress ACL Processing

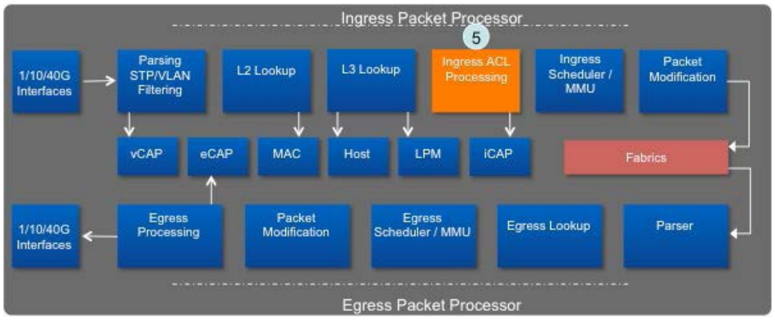

Figure 17: Packet Processor stage 5: Ingress ACL Processing

The Ingress ACL processing block functions as a matching and policy enforcement engine. All policy and matching logic is stored in the iCAP TCAM.

Routed traffic is checked against any router ACLs configured on the ingress direction of the receiving L3 interface. If the packet matches a DENY statement it will be dropped. However if the packet matches a PERMIT statement, or no router ACL is applied to the source interface, the traffic will continue through the forwarding pipeline.

The packet is also checked against any quality of service (QoS) policies contained on the ingress interface, if the packet is matched by a class within a policy-map it is subject to any actions defined within that class. Typical actions include policing/rate-limiting, remarking the CoS/DSCP or manually setting the traffic-class/queue of the packet to influence queuing further in the pipeline.

Finally the Ingress ACL Processing block applies any packet filters, such as storm-control and IGMP Snooping.

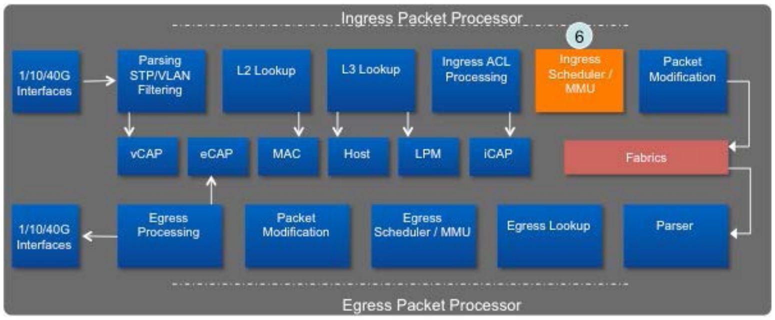

Figure 18: Packet Processor stage 6: Ingress Scheduling Engine

The Ingress Scheduling Engine or Memory Management Unit (MMU) performs the packet buffering and scheduling functions of the ASIC. The ingress scheduler is made up of two components:

- The ingress phase of the ingress MMU allocates available memory segments to packets that must be buffered.
- The egress phase of the ingress MMU replicates and de-queues packets resident in system buffer, making those buffers available once again.

The primary focus of the ingress scheduler is to provide short term buffering to packets in situations where the fabric links are congested. In a typical system that leverages Dynamic Load Balancing Fabric it is expected for ingress queuing to be minimal (See step 8 for additional detail on DLBF).

In the case of multicast traffic the Ingress MMU replicates the packet once for each packet processor that must receive the packet.

## Stage 7: Ingress Packet Modification

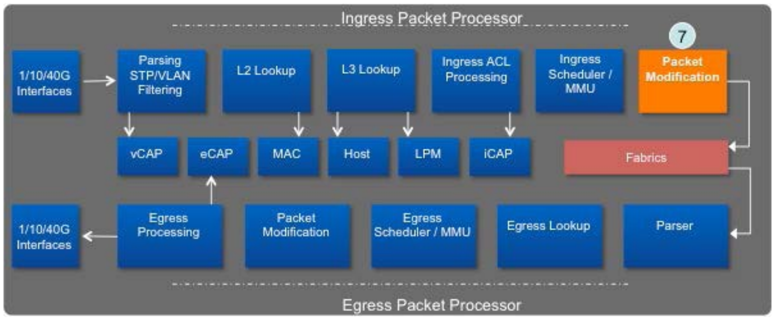

Figure 19: Packet Processor stage 7: Ingress Packet Modification

All previous blocks in the forwarding pipeline performed actions, some of these actions resulted in a requirement to make changes to the packet header, however no actual rewrites took place. Each block in the pipeline appended any changes to the packet header as meta-data.

In the case of unicast packets the modification block takes the meta-data added by previous blocks in the ingress pipeline, and performs the appropriate rewrite of the packet header. The exact data rewritten depends on the packet type and if the packet was routed or bridged, rewritten data typically includes changing the source and destination MAC address and decrementing the TTL for routed traffic and rewriting the CoS value. Multicast packets are not rewritten until the egress packet modification phase.

The ingress packet modification block is also responsible for creating and applying the internal fabric header. The header provides a set of forwarding instructions to the switching fabric and egress packet processors. This approach negates the need for the fabric or egress packet processors to perform a full forwarding lookup on the entire packet header, which in turn reduces both the latency and complexity of the system.

## Stage 8: Switching Fabric

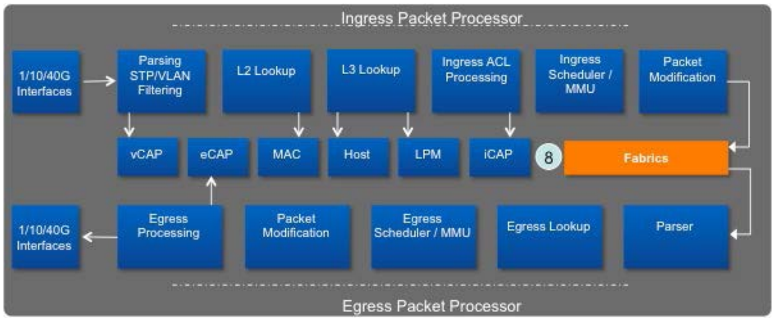

Figure 20: Packet Processor stage 8: Switching Fabric

The switching fabric provides data-plane connectivity between all packet processors in the system. The primary responsibility of the fabric is providing a mechanism to transport packets from the ingress to the egress packet processor based on the internal fabric header. The 7250X and 7300X series uses a packet based fabric where the packet is sent in entirety over a chosen fabric link. In a packet based fabric no cell-division, or super-framing actions are required significantly reducing the latency of the fabric.

The final task of the ingress packet processor is to select which of the available fabric links should be used for forwarding to the egress packet processor, this is achieved through the use of a load distribution algorithm. As with traditional load distribution algorithms a set of header fields are collected from the packet, and a hash key is created. The hash key is run through a hash rotation, and a result is produced. However unlike traditional methods where hash results (or buckets) are arbitrarily allocated across all available interfaces, the 7250X / 7300X use a more advanced and flexible approach called fabric dynamic load balancing, this feature enables efficient mapping and subsequent remapping of flows to the optimum available link.

At the packet processor level each fabric link is tracked and monitored, both in terms of link utilization and queue depth, this data is quantized to identify the optimum interface at a particular point in time. Each new flow received by the packet processor is mapped to the current optimum interface based on the computed hash result.

The rate of a particular traffic flow generally does not remain consistent, especially in the case of TCP communication. DLBF provides a mechanism to periodically re-balance flows, ensuring the distribution remains consistently optimized. Rebalancing is achieved using an inactivity timer, if the time between receiving two packets in the same flow is greater than the inactivity timer, then the flow is rebalanced over the current optimum link. Using the inactivity timer allows effective optimization of the fabric without risking out of order packets.

### Stage 9: Egress Parser

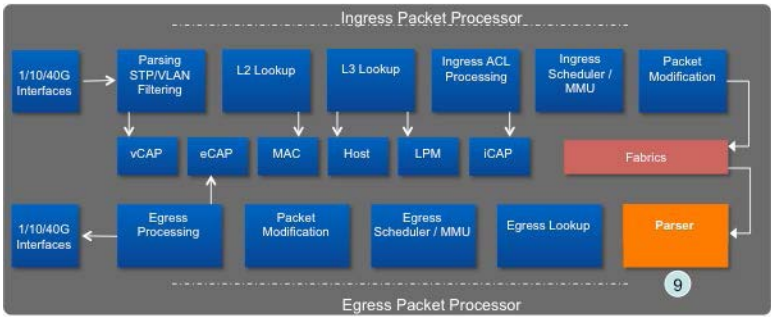

Figure 21: Packet Processor stage 9: Egress Parser

Upon receiving the packet from the fabric the Parser extracts the key fields from two sets of packet headers. The first is the internal fabric header, which contains the instructions required to make an egress forwarding decision. The parser will also extract the key L2, L3 and L4 data from the external packet headers in order to support the egress processing functionality.

### Stage 10: Egress Lookup

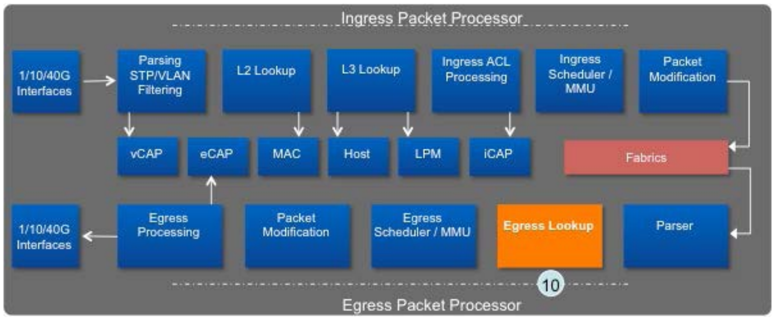

Figure 22: Packet Processor stage 10: Egress Lookup

The egress lookup is made entirely based on the contents of the internal fabric header attached by the ingress packet processor. This negates the need to perform additional accesses to the Host and LPM tables in the egress direction, preserving pipeline bandwidth and maintaining a low latency forwarding model.

The Internal fabric header includes all necessary information as to which port(s) and in which vlan(s) packets should be forwarded and potentially replicated.

## Stage 11: Egress Scheduling Engine

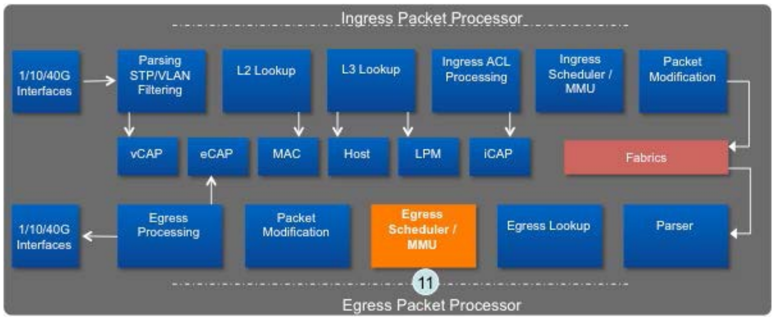

Figure 23: Packet Processor stage 11: Egress Scheduling Engine

Similar to the Ingress Scheduling Engine, the Egress Scheduling Engine performs the packet buffering and scheduling functions of the ASIC, and is divided into the same two phases:

- The ingress phase of the egress MMU allocates available memory segments to packets that must be buffered.
- The egress phase of the egress MMU replicates and de-queues packets resident in system buffer, making those buffers available once again.

However unlike the ingress scheduler which performs buffering only if the fabric links are congested, the egress MMU will provide buffering in response to a congested egress port, it will also implement any configured QoS shaping or queuing policy, in a typical system it is expected the majority of the buffering will take place on the egress scheduler.

Each packet processor has 12MB of on chip packet buffer memory, the egress scheduler/MMU is a logic block discrete from the ingress scheduler; no system memory is shared between packet processors. This memory is divided into fixed segments, 208 bytes in size, this ensures the system a finite but predictable number of packet buffer segments. These segments are then distributed among the various memory pools. There are three types of memory pool:

- Headroom pools, used exclusively for in-flight packets.
- Private Pools, buffers dedicated exclusively to a particular system queue.
- The Shared Pool, a single large pool is available to store packets once a particular system queue's private pool has been exhausted. The shared pool is significantly larger than the headroom or private pools.

If packet buffering is required the ingress phase of the egress MMU ascertains if there is memory available for this packet and in which pool the packet should be stored (based on the system fair-use policy). While a large packet may consume multiple buffer segments it is not possible for multiple packets to be located in a single segment.

Each physical port will have 8 unicast queues which map internally to the 8 supported traffic-classes. Therefore a system queue can be uniquely identified by the combination of Egress Port and Traffic class, or (EP, TC). Each system queue will have a pool of dedicated (private) buffers that cannot be used by any other system queue.

If a packet arrives at the scheduling engine and must be enqueued (i.e. if the egress port is congested), several steps take place. In the first instance the Ingress MMU will attempt to enqueue this packet into the private buffers for the destination system queue.

If the are no private buffers for that (EP,TC) available in the appropriate private pool, two further checks are made:

- Are any packet buffers available in the shared buffer pool?
- Is the system queue occupying less than its permitted maximum number of buffer segments of the shared pool? (i.e. the queue-limit).

If both of the above statements are true, the packet will be en-queued on buffers from the shared pool. If either of the above statements is false the packet will be dropped.

If a packet arrived and there was no congestion the packet it would be held in 'headroom buffers' used exclusively for in-flight packets, the packet would remain here only long enough for the header to pass through the forwarding pipeline and be serialized out of the egress interface.

Once a system queue contains 1 or more segments the egress phase of the egress MMU will attempt to de-queue these segments. The egress MMU will attempt to forward packets out of an Egress Port on a per Traffic Class basis. The rate at which this occurs is based on the queuing configuration and any configured egress packet shaping. By default the MMU will be configured with hierarchical strict priority queues, this ensures packets in traffic-class 5 are processed only when the higher priority classes 6 and 7 are empty, while packets in traffic-class 4 are processed only when classes 5,6 and 7 are empty etc.

In the case of multicast traffic, the egress MMU will replicate the packet once per subscribed VLAN on all local ports that connect to receivers.

### Stage 12: Egress Packet Modification

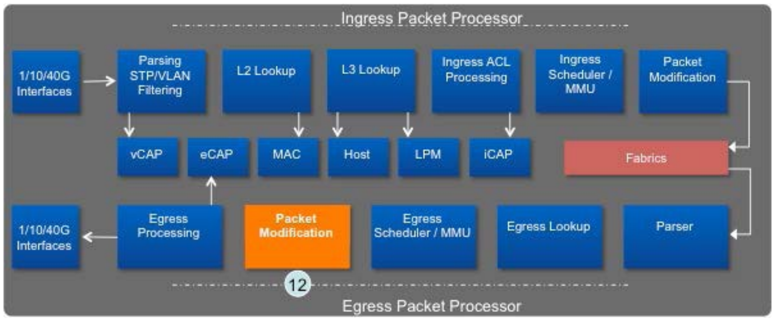

Figure 24: Packet Processor stage 12: Egress Packet Modification

The Egress Packet Modification first determines if a packet is unicast or multicast. If the packet is unicast, all necessary rewrites have already taken place on the ingress packet processor. If the packet is multicast the egress Packet Modification block will make appropriate rewrites, based on if the packet was bridged or routed, such as Source MAC address and TTL.

Finally the Egress Packet Modification block will remove the internal fabric header from all packets, ensuring the format of the packet that leaves the device conforms to IEEE standards.

### Stage 13: Egress Processing

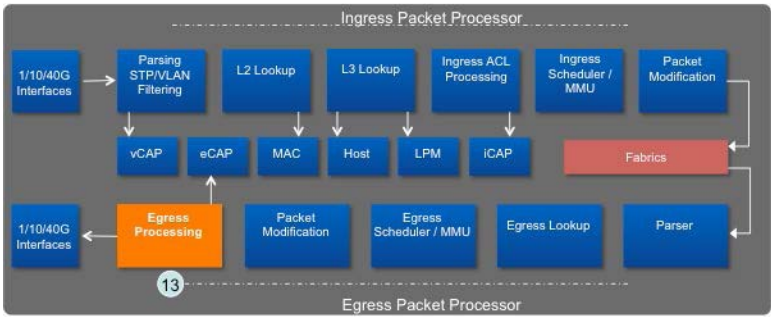

Figure 25: Packet Processor stage 13: Egress Processing

The Egress ACL processing block enables packet-filtering functionality in the egress direction by performing a mask-based lookup in the eCAP, the third of the ACL TCAMs.

### Stage 14: Network Interface (Egress)

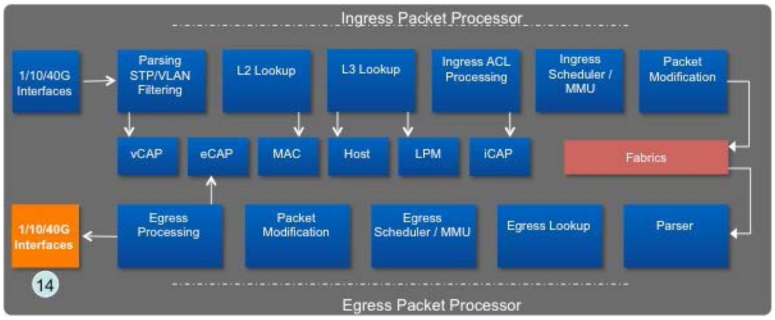

Figure 25: Packet Processor stage 13: Egress Processing

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

**Arista EOS: A Platform for Scale, Stability and Flexibility**

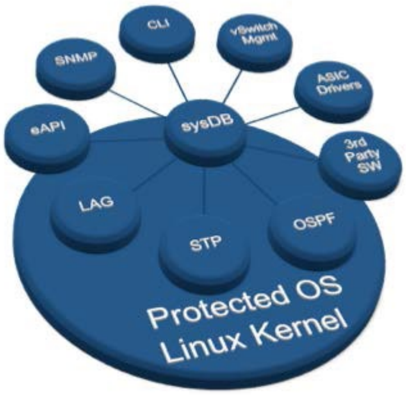

Figure 27: Arista EOS Software Architecture showing some of the Agents

Arista Extensible Operating System, or EOS®, is the most advanced network operating system in the world. It combines modern-day software and O/S architectures, transparently restartable processes, open platform development, a Linux kernel, and a stateful publish/subscribe database model.

At the core of EOS is the System Data Base, or SysDB for short. SysDB is machine generated software code based on the object models necessary for state storage for every process in EOS. All inter-process communication in EOS is implemented as writes to SysDB objects. These writes propagate to subscribed agents, triggering events in those agents. As an example, when a user-level ASIC driver detects link failure on a port it writes this to SysDB, then the LED driver receives an update from SysDB and it reads the state of the port and adjusts the LED status accordingly. This centralized database approach to passing state throughout the system and the automated way SysDB code is generated reduces risk and error, improving software feature velocity and provides flexibility for customers who can use the same APIs to receive notifications from SysDB or customize and extend switch features.

Arista's software engineering methodology also benefits customers in terms of quality and consistency:

- Complete fault isolation in the user space and through SysDB effectively convert catastrophic events to nonevents. The system self-heals from more common scenarios such as memory leaks. Every process is separate, no IPC or shared memory fate sharing, endian-independent, and multi-threaded where applicable.
- No manual software testing. All automated tests run 24x7 and with the operating system running in emulators and on hardware Arista scales protocol and unit testing cost effectively.
- Keep a single system binary across all platforms. This improves the testing depth on each platform, improves time-to-market, and keeps feature and bug compatibility across all platforms.

EOS, and at its core SysDB, provide a development framework that enables the core concept - Extensibility. An open foundation, and best-in-class software development models deliver feature velocity, improved uptime, easier maintenance, and a choice in tools and options.

**Conclusion**

The Arista 7250X & 7300 series switches combined with the Arista leaf-spine or Spline design methodologies provide flexible design solutions for network architects looking to deliver market-leading performance driven networks, which scale from several hundred hosts all the way up several hundred thousand hosts.

The fixed configuration 7250X series delivers up to 64 x 40GbE or 256 x 10GbE ports offering wirespeed performance, with 5.12Tbps or 3.84Mpps of forwarding capacity. While the high density modular 7300 Series delivers up to 256 x 40GbE or 1024 x 10GbE ports with up to 20Tbps or 15Bpps of forwarding capacity. The 7250X & 7300 series both provide the port density, table scale, feature set and forwarding capacity essential in today's datacenter environments.

All Arista products including the 7250X & 7300 Series run the same Arista EOS software binary image, simplifying network administration with a single standard across all switches. Arista EOS is a modular switch operating system with a unique state sharing architecture that cleanly separates switch state from protocol processing and application logic. Built on top of a standard Linux kernel, all EOS processes run in their own protected memory space and exchange state through an in-memory database. This multi-process state sharing architecture provides the foundation for in-service-software updates and self-healing resiliency.

Combining the broad functionality with the diverse form factors make the Arista 7250X and 7300 Series ideal for building reliable, low latency, cost effective and highly scalable datacenter networks, regardless of the deployment size and scale.

**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 & B, 11th Floor
Marathahalli Outer Ring Road
Devarabeesanahalli Village, Varthur Hobli
Bangalore, India 560103

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

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