

ARISTA WHITE PAPER

Software Driven Cloud Networking

Arista Networks, the leader in high-speed, highly programmable data center switching, has outlined a number of guiding principles for integration with Software Defined Networking (SDN) technologies, including controllers, hypervisors, cloud orchestration middleware, and customized flow-based forwarding agents. These guiding principles leverage proven, scalable, and standards-based control and data plane switching technologies from Arista.

Emerging SDN technologies complement data center switches by automating network policies and provisioning within a broader integrated cloud infrastructure ecosystem. Arista defines the combination of SDN technologies with the Arista Extensible Operating System (Arista EOS[®]) and the Arista EOS+ platform as Software Driven Cloud Networking (SDCN).

The resulting benefits of Arista's SDCN approach include: network applications and open, scalable standards integration with a wide variety of cloud provisioning and orchestration tools; seamless mobility and visibility of multi-tenant virtual machines for open workload mobility; and real-time Network Telemetry data for best-of-breed coupling with cloud operations management tools.

ARISTA

CLOUD TECHNOLOGY SHIFT

Ethernet networks have evolved significantly since their inception in the 1980s, with many evolutionary changes leading to the various switch categories that are available today (see Figure 1). Data center switching has emerged as a unique category, with high-density 10Gbps, 40Gbps, and now 100Gbps port-to-port wire-rate switching as one of the leading Ethernet networking product areas. Beyond these considerable speed progressions, data center switching offers sub-microsecond switch latency, more resilient architectures with wide equal-cost multi-path routing architectures, the integration of network virtualization to support simplified provisioning, and the integration of Network Applications on top of the data center infrastructure to align IT operations with network behavior.

While these state-of-the-art switching features leverage 30 years of progressive hardware and software technology evolution, successful implementation of Arista SDCN requires a fundamental shift from closed, vendor-specific proprietary monolithic device centric operating systems to open, extensible, externally programmable network operating systems. This open extensibility requirement is driven by the guiding principles of cloud data centers in which resources are managed dynamically as one integrated system made up of compute, network, and storage.

Cloud controllers, as middleware orchestration and resource managers to the underlying infrastructure, drive provisioning decisions on workload placement and mobility. This includes where the workload resides at the edge of the network. Every time a workload is moved the network must be updated to ensure proper provisioning of required resources or visibility to the virtual workload. Switches must interface with these controllers in real time, as these workloads are becoming highly mobile, based upon rack level and data center private and hybrid optimization technologies. Arista refers to this as open workload mobility.

Closed network operating systems that are built on older design principles can, at best, offer one-off implementations and struggle to support the growing list of different SDN controller form factors. Arista, on the other hand, is in a unique leadership position—the industry award-winning modular Arista EOS can interact with multiple systems concurrently, handling external controller updates and managing highly distributed switch forwarding states, both in real time. The Arista approach offers the best of both worlds, providing service control to external controllers, while scaling with Leaf/Spine switching architectures for the most demanding carrier-class cloud data centers.

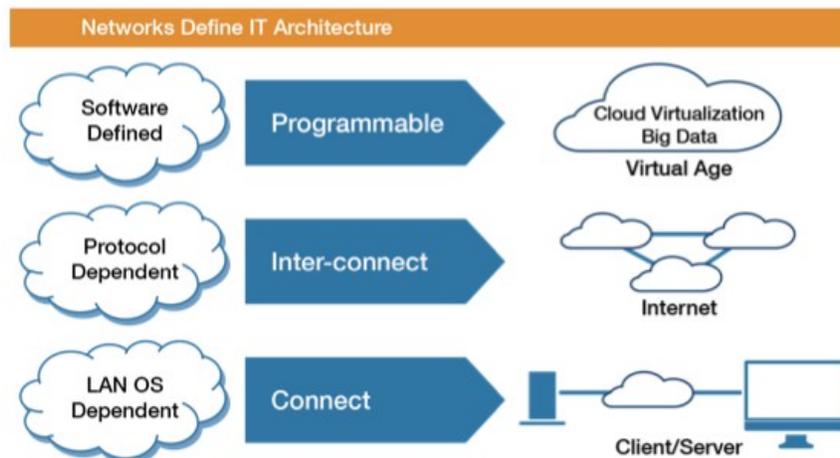


Figure 1: Software driven cloud networking evolution

Table 1: A stack approach to Arista SDCN

| Stack | Examples | Benefits |
|------------------------|---|--|
| Virtual machines | Web app framework | Scale up/down as needed |
| SDN controllers | OpenFlow, OpenStack, vCloud Suite, vSphere | Orchestration, service abstraction and positioning |
| Network virtualization | Scalable, multi-tenant virtual networks | Enables open workload mobility |
| Server hypervisor | x86 bare metal server abstractions | Elastic computing, resource optimization, non-disruptive server upgrades |
| Storage | Network, direct-attached, SSD, Hadoop, big data | Centralized VMDK for app mobility, software patches |
| Cloud-enabled network | Arista EOS | Open and programmable for custom flows, VM mobility, automated tenant onboarding |

On average, with a legacy network operating system it took two to four weeks to configure and release into production a fully integrated data center infrastructure for any new or refreshed application. Much of this two- to four-week time period was based on the administrators coordinating with each other in an ad hoc manner on change control issues.

Cloud data centers run counter to this model, with highly virtualized and elastic workloads, time-of-day application demands, and rapid provisioning requirements that are driven by service-catalog web-facing front ends. Administrators can no longer manually coordinate provisioning events, manually update configuration databases, and fully test the system prior to hosting live in-production environments. Highly virtualized cloud infrastructures drive the need for real-time configurations, switch topology data, the ability to trace virtual machines (VMs) across physical and virtual resources end-to-end, and the ability to change or remap tenants and VMs based on quality of service (QoS) and security policies. Network administrators cannot perform these functions instantaneously, nor can they perform these functions in isolation. Integration with external controllers, cloud orchestration or provisioning middleware, and service took weeks to configure, test, and release into production. Service level agreement (SLA) management tools have become a core cloud infrastructure requirement (see Table 1).

Consider the case of a data center or cloud administrator. The physical attributes of servers, switches, and interconnects are well known to the infrastructure administrators. In many cases, the MAC address of each server, its physical location (including floor, row, and rack information), assigned IP address, physical and logical connections to the switch, and configuration files, are imported into asset tracking and configuration database applications. This database information is important for pinpointing problems and performing break/fix tasks in an efficient manner. In non-virtualized, non-cloud environments, this data is static and easy to maintain by the administrators. In clouds, where servers are virtualized and the placement of these VMs is often changing, there is a need for centralized controllers that can map and update the service policies as a set of explicit instructions to the underlying infrastructure platforms (such as servers, switches, firewalls, and load balancers) based on these location changes.

In large-scale virtualized environments, the operator should not have to worry about MAC learning, aging, Address Resolution Protocol (ARP) refresh, and the uploading of any VM location changes into a configuration database. The path from one device to another should be known within a centralized topology database, with real-time updates. When integrated externally to cloud controllers, the paths make network configuration, customized forwarding, and troubleshooting easier. The majority of data center switches across the industry do not allow any forwarding path programmability from the outside. They are a closed black box that the vendor controls, with pre-set forwarding path software and hardware algorithms. This is a clear case in which an external controller offers value.

Similarly, there are other use cases—in traffic engineering for aggregating Test Access Points (TAPs) to a centralized collection point, adding special headers to overlay Layer 2 traffic onto Layer 3 networks, classifying traffic based on content, monitoring congestion and hash efficiency over a Link Aggregation Group (LAG) or Equal-Cost Multipath (ECMP) group, and so on. Programmable switches, managed by external controllers, can address many of these cases.

SDN OVERLAYS, UNDERLAYS OR BOTH?

Enterprises are a complex mix of legacy and newer cloud-based applications. As the network architecture evolves to support the next phase of cloud and virtual networking needed to bring these worlds together, it is important to look at the two main elements in a Software Defined Network.

OVERLAY SDN

SDN overlays use network virtualization to separate, abstract and decouple the physical topology of networks from a 'logical' or 'virtual' topology by using encapsulated tunneling. This logical network topology is often referred to as an 'Overlay Network'. In these architectures some network features and functions are moved into overlays to control the data, specific flow or forwarding path. This may include:

1. Software overlays to shift management functions from the control plane of the switch to servers
2. Specific network services such as load balancing, access control, fire walling and visibility

Controllers for SDN overlays leverage existing physical networks to deliver functions such as provisioning and visibility.

UNDERLAY SDN

Controllers do not solve the broader set of complex high-performance network issues that exist at L2/L3/L4. To achieve that we need an uncompromised wire-speed SDN data plane as the physical 'Underlay Network.' To best utilize the capabilities of an SDN overlay while providing maximum transparency and performance, the underlying physical network must scale linearly and programmatically interface in a seamless manner with new network virtualization capabilities with very little contention and minimal end-to-end latency. The use of open standards and open APIs makes it possible to be controller agnostic and interoperable with other network infrastructure elements such as Wi-Fi, load balancers and firewalls.

At the core of every cloud, customers demand scalability, resiliency, and 24-hour business-critical uptime every day of the year. Hundreds of switches can easily be located within the same data center and need to instantaneously react to change events anywhere within the topology without dropping packets or creating congestion conditions. To deliver on these requirements, networking platforms have evolved with many of the data plane controller functions distributed and embedded. Link Aggregation Control Protocol (LACP), Open Shortest

Path First (OSPF), ECMP, and Border Gateway Protocol (BGP) are primary examples of standards-based distributed controller functions (which are often referred to as traffic engineering protocols). Because the majority of change events typically occur locally, a distributed approach allows the affected network node to operate independently, reacting and resolving changes within split seconds with near-zero packet drop. This distributed approach provides the high-resiliency behavior that is required for around-the-clock every-day uptime. As a result, networks today are rarely the root cause when there are application outage conditions.

BEST OF BOTH WORLDS: SOFTWARE DRIVEN CLOUD NETWORKING

Networking is critical to every IT organization that is building a cloud, whether the cloud is large or small. As a result, compromising resiliency over traffic flow optimization is unlikely. The approach that is well suited for most companies is to let the network layers perform their intelligent forwarding with standard protocols, and to use Arista SDCN to enhance the behavior of the network with tighter integration at the application layers utilizing open APIs for programmability of the underlay network at every level, be it the control plane, management plane or the data plane.

The more common Arista SDCN use cases include the following:

- Network virtualization for multi-tenant configuration, mobility, and management of VMs
- Customized flows between servers and monitoring/accounting tools (or customizable data taps)
- Service routing to load balancers and firewalls that are located at the Internet edge
- Big Data/Hadoop search placement and real-time diagnostics

Arista SDCN can greatly enhance and automate the operations that are associated with these use cases. Integration with an external controller provides the customized intelligence for mapping, connecting, and tracing highly mobile VMs, while the distributed protocols within the networking devices provide the best-path data forwarding and network resiliency intelligence across large distributed topologies.

The migration to private, public or hybrid clouds is revolutionary in technology, but evolutionary in adoption. Arista endorses a broad spectrum of data, management and control plane capabilities to bring flexibility to our customers as shown in Figure 2.

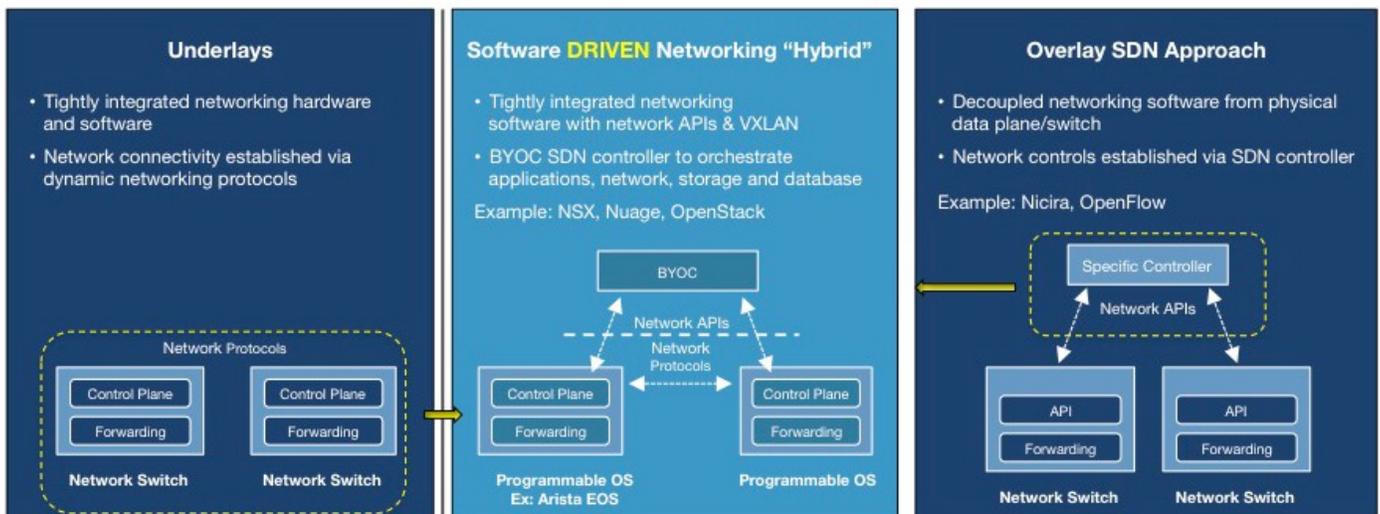


Figure 2: SDN approaches

THE FOUR PILLARS OF ARISTA SDCN

Arista Networks believes that Ethernet scaling from 10Gb to 40Gb to 100Gb Ethernet—and even terabits— with well-defined standards and protocols for Layer 2 and Layer 3 is the optimal approach for a majority of companies that are building clouds. This scaling allows large cloud networks of 10,000 or more physical and virtual server and storage nodes today, scaling to 100,000 or more nodes in the future without reinventing the Internet or having to introduce proprietary APIs.

At VMworld 2012, Arista demonstrated the integration of its highly distributed Layer 2 and Layer 3 Leaf/Spine architecture with VMware's Virtual eXtensible LAN (VXLAN) centrally controlled, overlay transport technologies. This integration offers unsurpassed multi-tenant scalability for up to 16 million logically partitioned VMs within the same Layer 2 broadcast domain. VXLAN embodies several of the Arista SDCN design principles and is a result of an IETF submission by VMware, Arista, and several other companies.

It is important to recognize that building such highly scalable and dense clouds is only part of the equation. Application mobility, storage portability, self-service provisioning and automation, and dynamic resource optimization create new management and operational challenges that are different from many traditional data centers, including those designed in the late 1990s (based on a client/server architecture).

Arista has identified these cloud challenges and has been solving them methodically using the four pillars of Software Driven Cloud Networking (see Table 3):

PILLAR 1: UNIVERSAL CLOUD NETWORK

Scaling cloud networking across multiple chassis via Multi-Chassis Link Aggregation Groups (MLAGs) at Layer 2 or Equal-Cost Multipath (ECMP) at Layer 3 is a standards-based approach for scalable cloud networking. This approach ensures effective use of all available bandwidth in non-blocking mode, while providing failover and resiliency when any individual chassis or port has an outage condition. MLAG and ECMP cover all of the important multipath deployment scenarios in a practical manner, without introducing any proprietary inventions. These technologies currently scale to 100,000 or more physical compute and storage nodes, and to more than one million virtual machines.

The new Arista Spline® architecture introduced with the high-density host-aggregation platform from Arista enables large densities of directly connected hosts to connect to a single-tier or two-tier network.

With the advent of next-generation multi-core server CPUs, as well as dense VMs and storage, this type of uncompromised Leaf/Spine or Spline topology with non-oversubscribed capacity, uplink, downlink, and peer ports is paramount. These technologies are commonly integrated with server link redundancy, both physically and logically. The industry standard is LACP. Arista has completed interoperability, including configuration automation with VMware's vSphere 5.1 release. This interoperability and configuration automation ensures that links are configured correctly for load sharing and redundancy at the virtual network interface card (vNIC) level.

PILLAR 2: SINGLE-IMAGE LAYER 2/3/4 CONTROL PLANE

Some networking vendors are attempting to respond to SDN with three decades of networking control plane architectures that are non-modular, non-database-centric, and proprietary. For these vendors, SDN integration requires multi-year, expensive undertakings. Customers will receive proprietary implementations, with vendor lock-in at the controller level, as well as many of their non-standard distributed forwarding protocols. Arista has seen these issues firsthand. Customers have requested Layer 2 and Layer 3 control interoperability with Arista switches as well as with switches from other vendors. Arista has had to debug many of these non-standard protocols. In

short, the switches from other vendors are very difficult to implement as part of an SDN architecture, and they have proprietary tools for configuration and management. This is not the answer going forward.

Instead of these touted proprietary “fabric” approaches, standards-based Layer 2 and Layer 3 IETF control plane specifications plus OpenFlow options can be a promising open approach to providing single-image control planes across the Arista family of switches. OpenFlow implementations in the next few years will be based on specific use cases and the instructions that the controller could load into the switch. Examples of operational innovations are the Arista Zero Touch Provisioning (ZTP) feature for automating network and server provisioning and the Arista Latency Analyzer (LANZ) product for detecting application-induced congestion.

PILLAR 3: NETWORK-WIDE VIRTUALIZATION

- By decoupling “the physical infrastructure” from applications, network-wide virtualization expands the ability to fully optimize and amortize compute and storage resources with bigger mobility and resource pools. It therefore makes sense to provision the entire network with carefully defined segmentation and security to seamlessly manage any application anywhere on the network. This decoupling drives economies of scale for cloud operators. Network-wide virtualization is an ideal use case in which an external controller abstracts the VM requirements from the network and defines the mobility and optimization policies with a greater degree of network flexibility than what is currently available. This virtualization requires a tunneling approach to provide mobility across Layer 3 domains as well as support for APIs in which external controllers can define the forwarding path. Arista is leading this effort with several major hypervisor offerings. Critical to this integration is Arista CloudVision which emphasizes a multi-vendor open API support including eAPI (RESTful api using JSON), OVSDB, Openstack (Neutron ML2) and Open Management Interface OMI).

PILLAR 4: Network Wide Cloud Automation

Customers who are deploying next-generation data centers are challenged with managing and provisioning hundreds, or possibly thousands, of networking devices and to do so at an increasing pace of change. In today’s networks, more is required than just a handful of CLI-initiated features in order to scale to the demands of managing and automating the network. Application mobility, storage portability, self-service provisioning and automation, and dynamic resource optimization require programmability of the network, and, for many customers, pre-built solutions that are easy to customize to their unique needs. The Arista EOS+ platform and Arista EOS CloudVision provide an open programmable network built on an open network operating system, Arista EOS. Programming the network for business agility by enabling dynamic workload and workflow integration, scalability and visibility. — this is the real SDN and the core of Software Driven Cloud Networking and is delivered via Arista CloudVision.

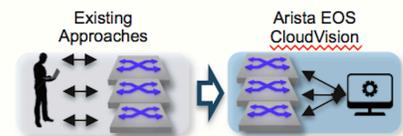
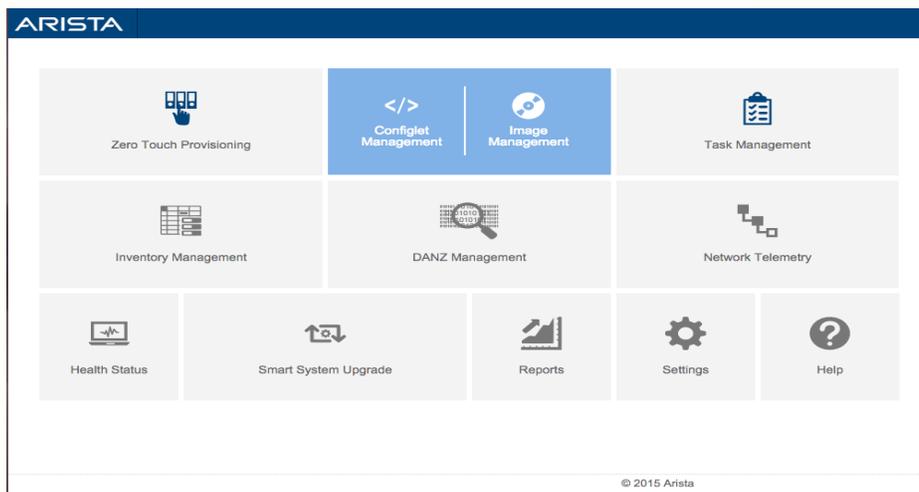
CloudVision

Arista’s CloudVision platform provides a set of services that simplifies monitoring, management and controller integration in the virtualized data center. CloudVision provides standard northward facing APIs supporting commercial or open source controllers. CloudVision’s open api’s ensures systems architects are not locked into potentially proprietary virtualization system architectures. In addition to OVSDB services, CloudVision provides a RESTful, JSON based command line API that allows administrators to craft customized network management and provisioning tools. CloudVision’s open API infrastructure, reduces development costs of orchestration tools. CloudVision communicates with multiple controllers simultaneously to accommodate heterogeneous data centers enabling a common infrastructure to economically serve cloud data center workloads. Cloudvision delivers this common infrastructure and by enabling a single point of network state with no proprietary lock in with open partner integration including Dell ASM, HP OneView, F5, Palo Alto Networks Panorama, Microsoft (Systems Center and Network Controller), Openstack Neutron ML2 (RedHat, Rackspace, Mirantis, SUSE, Vmware), Nuage Networks Virtualized Service Platform, and Infinera Transport SDN.

CloudVision delivers a network services abstraction layer that decouples controllers from the physical data center infrastructure. This abstraction insulates controllers from infrastructure OS dependencies removing switch OS and controller version dependencies and reducing costs associated with controller certification and network maintenance. This includes networked applications for workload mobility, network configuration management and control via smart systems rollback and upgrades, and integrated workflow telemetry. CloudVision's network abstraction dramatically improves controller scaling by up to 10x competitive offerings, with a single touch point to control Arista switches in a cloud data center. CloudVision brings the benefits of a software driven cloud solution by leveraging Arista's open and programmable Extensible Operating System, EOS, to drive down OpEx costs. CloudVision brings the benefits of a software driven cloud solution by leveraging Arista's **network wide state synchronization** to an Extensible Operating System, EOS, **reducing** OpEx costs. **It delivers customers a turnkey solution to cloud-like workload automation and workflow visibility.**

Building on this network abstraction layer, CloudVision also provides a turnkey approach to network-wide automation. From initial device provisioning via the Zero Touch Provisioning (ZTP) feature to ongoing network-wide provisioning of configurations, software versions, and scripts, CloudVision helps to reduce the time and improve the accuracy of change controls over the operational lifecycle. Since CloudVision is built on a network-wide database that includes both realtime and historical state, the operator can actually roll-back their entire network (including configurations and software versions) to a previous known and working state. This is an especially helpful tool for restoring network service should the need arise during maintenance windows. Specifically built to help bring cloud-like automation to the mainstream, these capabilities are pre-built into the CloudVision suite so that customers can take advantage of the functionality without having to do custom integration and scripting themselves.

Arista CloudVision



- Automatically provision switches centrally from a network wide view
- Single tool & view to manage code, config, and scripts
- Manage switch throughout switch's DC life cycle using single provisioning process
- Full configuration coverage, including VXLAN
- Configuration compliance validation

Figure 3: Arista Cloudvision

Arista EOS, the Extensible Operating System

Arista EOS is an inherently open and programmable network operating. It is built on a standard linux distribution, bringing the capability to integrate with the broad ecosystem of linux-based DevOps tools. Arista EOS has a rich set of APIs, using standard and well-known management protocols such as OpenFlow, Extensible Messaging and Presence Protocol (XMPP), System Network Management Protocol (SNMP), and the ability to natively support common scripting languages such as python. Moreover, Arista EOS provides a single point of management and is easily integrated with a variety of cloud stack architectures. No proprietary fabric technology is required, and there is no need to turn every switch feature into a complicated distributed systems problem.

Arista's EOS software architecture is uniquely suited to expanded automation or programming of the network. Arista EOS deploys a memory-resident database, Arista sysDB, for managing the system state—from routing tables to ACLS to counters. The publish-subscribe event model of databases inherently solves the complexity of coordination and communication of state changes among interested processes and applications. Arista EOS maintains its reliability and resiliency even as more functionality is added Arista's development of EOS delivers the first network operating system specifically designed for the demands of today's cloud data centers.

| Legacy Network OS | Arista OS |
|-------------------------------------|---|
| Closed/proprietary | Open, based on standard linux distribution |
| Multiple images and software trains | Single image across the product portfolio |
| Bloated kernel | In-memory, system database for storing state |
| IPC message passing | Scalable, resilient publish-subscribe communication |
| Interrupt driven | Event driven |
| Fragmented strategy | Developer friendly |
| Intended to lock-in | Designed to integrate |

Arista EOS Extensibility

Arista EOS offers extensibility at all layers, Figure 4.

1. Management plane extensibility via APIs, such as EOS API (eAPI) and SNMP. Using simple, well-documented and widely used programming models such as Java-Script Object Notation (JSON), eXtensible Markup Language (XML), Python and XMPP to interact with the EOS management plane, Arista's APIs provide direct programmatic access to management systems such as Dell ASM, HP OneView, EMC Smarts, VMware vCenter/vRealize, IBM Tivoli and Splunk.

2. Control plane extensibility via advanced event management (AEM), a complete event handler subsystem to allow real-time and event-driven programming of the Control plane. Interacting with SysDB, Arista EOS can enable network switch actions on any state change in the system through a set of pre-defined triggers.
3. Data plane extensibility with in-line programmability. Customers looking to tune their application performance on the network can customize traffic flows by filtering and redirecting traffic using industry-standard OpenFlow or controller-less Arista DirectFlow constructs.
4. Virtual Machine extensibility using Arista vEOS and VM Tracer. The Arista vEOS control plane provides the ability to run the same EOS software as a vm inside any virtualized environment. This provides customers the virtual machine flexibility for lab certification efforts or for development purposes.
5. Application level extensibility for third-party development. The Arista EOS applications portal opens up Arista EOS to third-party development via SDK tool kits, scripting and APIs, making possible new types of applications as well as off-the-shelf integration with best-of-breed solutions.
6. Access to all Linux operating system facilities including shell-level access. Arista EOS can be extended with unmodified Linux applications and a growing number of open source management tools to meet the needs of network engineering and operations. Also, Arista EOS provides direct access to the full set of Linux tools such as tcpdump through our full binary Fedora compatibility.

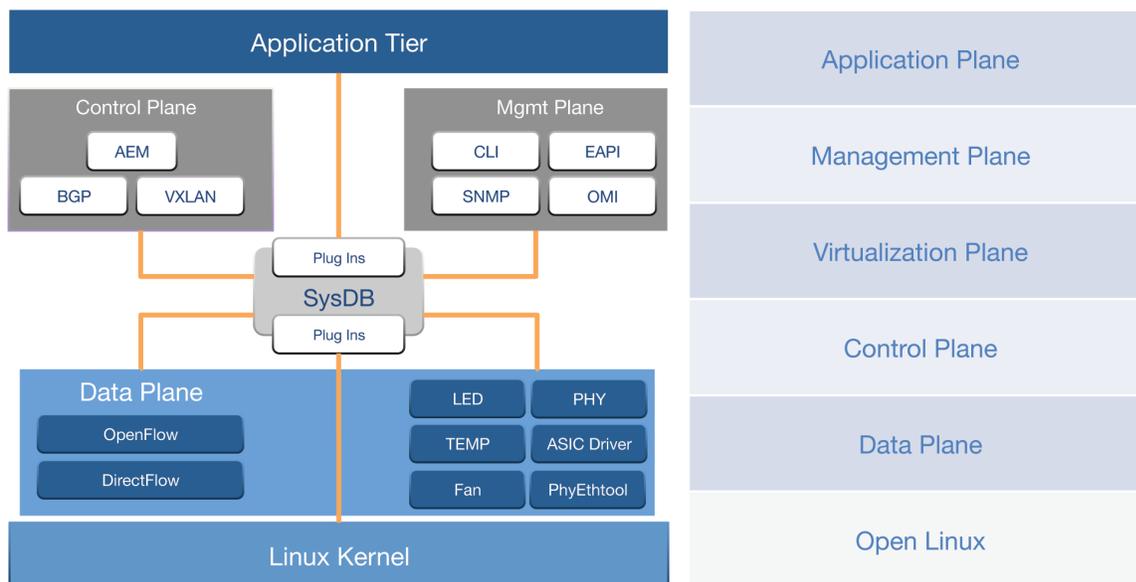


Figure 3: Extensibility at all layers

Arista EOS+ Platform

The Arista EOS+ Platform simplifies the programming of the network through a rich development environment built on the best network operating system and with pre-built applications. EOS+ enables a tight integration into cloud stacks, DevOps, network controllers and application/service workflows. Arista Consulting Services are available to customize applications, integrate the stack, or develop to complete custom requirements.

Arista EOS+ includes

- EOS SDK and eAPI: These programmatic interfaces allow customers to develop applications that integrate directly with the switch operating system by providing a direct interface into Arista EOS SysDB.

- vEOS: A Virtual EOS instance that allows for automated deployment, multi-topology setups, testing and certification in a rapid manner. All EOS features including STP, LACP, BGP, OSPF, JSON, OVSDB, Openstack and full Linux access are available.
- EOS Applications: Turnkey end-to-end software solutions supported by Arista's Technical Assistance Center (TAC) engineers. Examples of EOS Applications include ZTPServer for provisioning and a network telemetry solution for Splunk® Enterprise.
- EOS Consulting: A professional services practice focused on Enterprises and Service Provider customers delivering custom development and integration. EOS Consulting enables rapid prototyping and development providing customization of cloud application workflows, DevOps and third party service integration.

EOS FEATURES AND APPLICATIONS

Leveraging EOS and EOS+, Arista has developed core network features and applications that are designed to go above and beyond the traditional role of the network operating system by integrating three core components:

1. A collection of EOS features purpose-built to align with important IT workflows and operational tasks
2. Integration the key best-of-breed partners to bring the entire ecosystem together
3. Extensibility of the Network Application so that it can be aligned with any networks' operating environment and augment the traditional IT workflows

Arista enables open workload mobility with EOS. Arista EOS connects to the widest variety of network controllers and couple that integration with VM awareness, auto-provisioning, and network virtualization; Arista EOS is then able to deliver the tightest and most open integration with today's orchestration and virtualization platforms. In short, network operators gain the capability of deploying any workload anywhere in the network, with all provisioning happening in seconds, through software configuration and extensible API structures.

Arista Smart System Upgrade (SSU) is a series of patent-pending technologies that enable the network operator to seamlessly align one of the most challenging periods of network operations, the upgrade and change management operation, with the networks' operational behaviors. The network, with SSU, is capable of gracefully exiting the topology, moving workloads off of directly connected hosts, and aging out server load balancer Virtual IPs (VIPs) before any outage is ever seen. The multi-step, multi-hour process many network operators go through to achieve maximum system uptime becomes the default method of operations. SSU has demonstrated interoperability with F5 Load Balancers, VMware vSphere, OpenStack, and more. Smart System Upgrades have been pre-integrated into the CloudVision solution so that customers have a turnkey solution to automate network-wide upgrades and rollback.

Network Telemetry is all about data: generating, collecting, and distributing the data necessary to make well-informed network decisions about where problems may be happening, thus ensuring the data is available and easily reachable and indexed so that these hot spots, or problem areas, are rapidly fixed and troubleshooting is simple and quick. Network Telemetry integrates with Splunk Enterprise and several other log management and rotation/indexing tools.

Table 2: Arista SDCN four networking pillars

| Cloud Networking Requirements | Arista EOS Pillars |
|--|---|
| 1. Highly resilient, link-optimized, scalable topology | IEEE and IETF standard protocols MLAG and ECMP topology protocols |
| 2. Cloud adaptation, control plane | Single binary image for all platforms Zero Touch Provisioning for rapid switch deployment Industry support for OpenFlow and OpenStack |
| 3. Network virtualization | Hardware-based VXLAN, NVGRE VM Tracer for troubleshooting Integration with hypervisor controllers Open workload provisioning and orchestration |
| 4. Open Network Programmability | Extensibility at all layers Well-known interfaces into Arista EOS including Arista CloudVision Standard Linux utilities XMPP, XML, RESTful API, eAPI, EOS SDK Tools, solutions and services for EOS vEOS, EOS Applications, EOS Consulting Services |

USE CASES FOR ARISTA SDCN

NETWORK VIRTUALIZATION

Network virtualization is vital because the network must scale with the number of VMs and tenant partitions, as well as the affinity rules that are associated with mobility, adjacency, and resource and security policies. Moreover, IP mobility where the VM maintains the same IP address, regardless of the Layer 2 or Layer 3 network on which it is placed, whether within the same data center or moved to a different data center, is of significant importance. Additionally, the ability to partition bandwidth from an ad-hoc approach to one that is reservation-based is becoming a true service offering differentiator (see Figure 3).

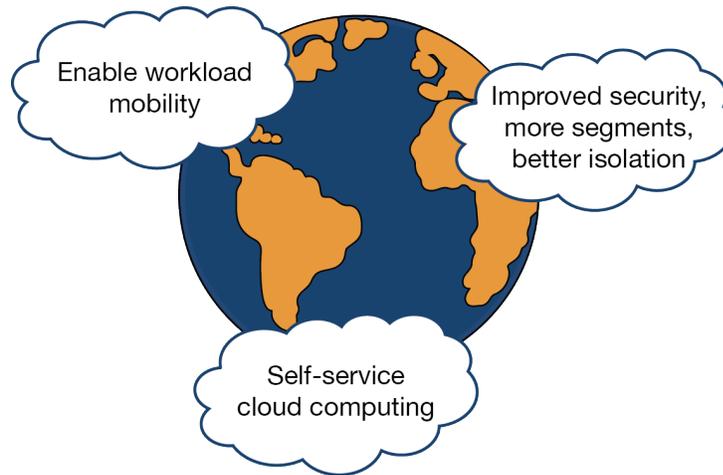


Figure 3: Network virtualization and use cases

There are multiple challenges in virtualizing the network. First, each Leaf/Spine data center switching core must support tenant pools well above the current 4K VLAN limits, as this is a requirement of both the VXLAN and NVGRE protocols used for network virtualization. Second, these switching cores (or bounded Layer 2 and Layer 3 switching domains) must offer large switching tables for scaling to 10,000 physical servers and 100,000 VMs.

Third, the switching core must be easily programmed centrally, with topology, location, resource, and service aware real-time databases. Fourth, the switching core must support the ability to have customized flows programmed within the Ternary Content Addressable Memory (TCAM) from an external controller. Finally, there must be a role-based security configuration model in which only a subset of services is available to the external controller while network compliancy is managed and tightly maintained by the network administrators (and not available to external controllers).

Offering tenant pool expansion above the 4K VLAN limit with overlay tunneling approaches and supporting large host tables, both physically and logically, is very hardware dependent. Switches must support these functions within the switching chips. This is one of the core pillars of Arista cloud-capable switching products—highly scalable, distributed protocols for handling large switching tables with ultra-low-latency efficiencies. Programming switches in real time, from a centralized controller and out to hundreds of switches within the topology, requires a messaging bus approach with a real-time database. This is another core Arista SDCN pillar—Arista EOS leads the industry with open programmatic interfaces, including the ability to run applications that are co-resident within Arista EOS as VMs. Additionally, CloudVision provides an interface to an external controller for programming the forwarding tables (or TCAMs) requires support for OpenFlow and other controller form factors. Again, as a core SDCN pillar, Arista has demonstrated the ability to program the host and flow entries within the switch tables using external controllers (see Figure 4).

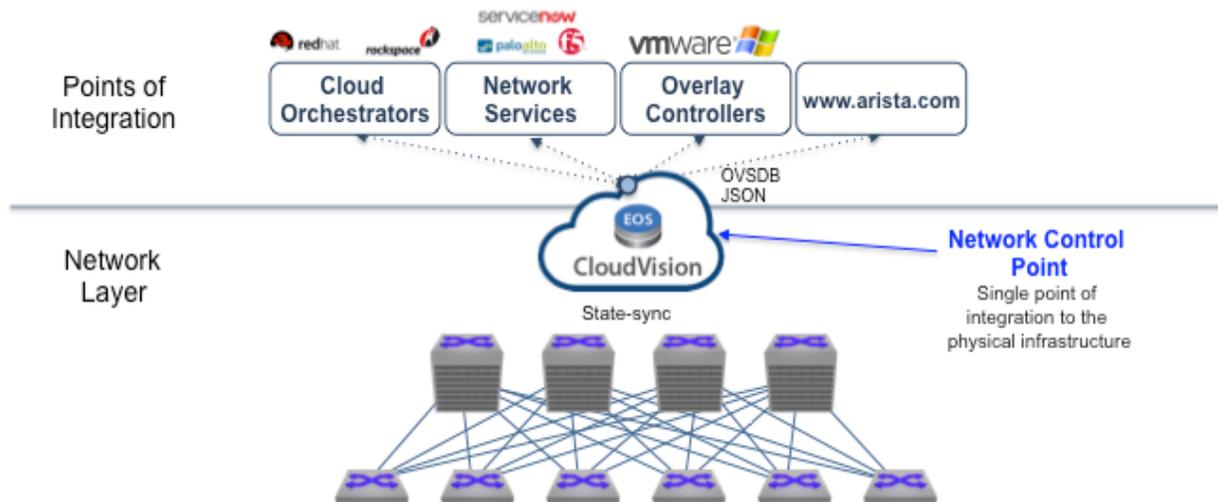


Figure 4: Arista Controller Integration

Arista offers industry-leading forwarding plane tunneling technologies (such as VXLAN) and integrates with network virtualization controllers. Arista EOS is one of the primary enablers for real-time communication event change, notification, and updating with external controllers. From a tracking and troubleshooting perspective, Arista offers its award-winning VM Tracer application. Arista VM Tracer supports standard VLAN multi-tenant virtual switch segmentation and has been extended to also track and trace VMs with VXLAN identities.

CUSTOMIZABLE DATA TAPS

The need for collecting and archiving application traffic has become a fundamental compliance requirement within many vertical markets. Financial transactions, healthcare patient interactions, database requests, call recordings, and call center responses are all becoming audited and recorded events. Moreover, cloud operations managers must collect traffic data from within the cloud infrastructure based on customer SLAs, bandwidth subscription rates, and capacity management.

The network is the ideal place for directing, collecting, filtering, analyzing, and reporting on the majority of these vertical market compliance and SLA management requirements. However, given the volume of traffic, the number of applications and associated VMs for each cloud tenant, and the high-speed data rates, it becomes difficult to capture, collect, and archive every packet that flows across the network. This is a classic data overload problem.

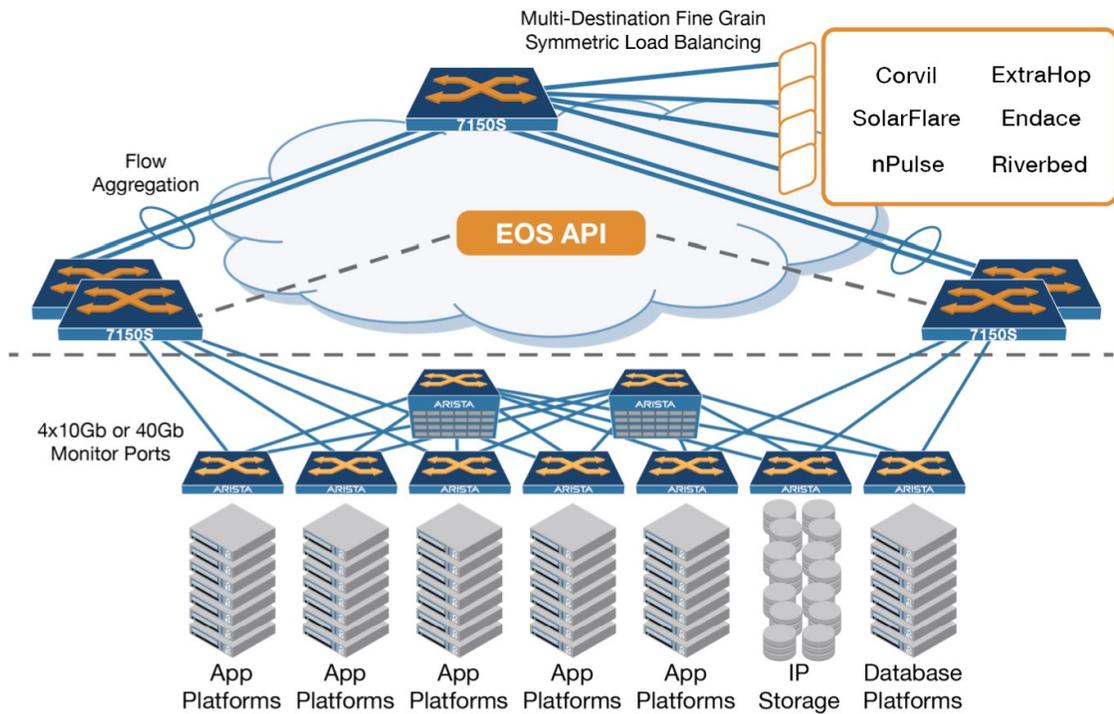


Figure 5: TAP and SPAN aggregation

One approach to reducing the scope of this problem is to provide customized data TAPs (see Figure 5) — specifically, to program the data flows between the endpoints that are generating the traffic and the collector devices that capture, store, analyze, and report on the data. This is an ideal use case for external controllers. The controller offers the mediation layer between the application endpoints. It identifies the endpoints that need traffic captures, the time of day required for collection, and the collection device that is specifically engineered for collecting, filtering, and reporting based on various vertical market compliance regulations. Ideally, the controller is integrated with the Information Technology Infrastructure Library (ITIL)-based service catalog onboarding tenant interface, and, based upon a set of collection options, can capture these compliance requirements as a set of actionable configuration events on a per-VM activation start and stop basis.

The controller communicates the endpoint information to the switch infrastructure every time the VM is started, moved, or stopped. The switch forwarding tables are then uniquely customized for redirecting traffic across non-production traffic ports to the industry-specific collectors (often referred to as tools) as driven by VM activation events. Customized data flows and taps are set up when the VM is started and the location of the physical machine in which it is running is identified. They are removed and reprogrammed when the VM is migrated to another location or taken out of service.

A customized data tap that is integrated with an external controller is a more scalable, effective, and industry-standard approach for monitoring, reporting, and alerting on VM traffic. This is especially true for customers who are scaling to 100,000 or more VMs in large multi-tenant cloud infrastructures. This use case exercises several of the core Arista SDCN pillars, including the need to program the network monitoring flows when a VM is started, moved, or stopped; the ability to mirror, forward, and redirect traffic at line-rate based upon multi-tenant header and packet information; and the ability to detect, in real time, the congested conditions and to send alerts back to the controller for real-time remediation. Arista EOS offers these capabilities today.

CLLOUD SERVICES

Cloud hosting is driving significant technology shifts with network-edge-based application services, including firewalls, load balancers, file compression, and file-caching appliances. These shifts are two-fold. First, many of these services become virtualized, running within the hypervisor that is co-resident and adjacent to the VMs that they are servicing (as opposed to centrally). Second, the services continue to be located at the WAN edge with dedicated appliances, but they need to have dynamic proximity-awareness based on VM mobility changes. In the first scenario, the services are moved together with the VM. In the second scenario, the services need instantaneous updating on one or several edge devices based on the new location of the VM. The second scenario is the most compelling from a controller-to-network packet flow view, because there are topology dependencies.

The control plane of the network holds topology location information and is the first to know, topologically, when a VM is moved from within the topology. While the application services management platforms can also determine the new location of the VM based on integration with external virtualization platforms, the mapping within the topology and where to best provide the services is not immediately known. This can cause an application outage, a client reachability problem, or even an application performance issue for periods of time that are unacceptable.

As an intermediary between the external controller and application services management platform, Arista has developed a JSON (JavaScript Object Notation) Remote Procedure Call (RPC) API that provides instantaneous location information to the cloud application edge services. Arista provides this ability based on the real-time interaction model within Arista EOS and its ability to work in parallel with updating forwarding tables while communicating to multiple external systems, including controllers and other management platforms. Arista has developed this Arista SDCN controller-based application services API by working closely with F5 Networks, Palo Alto Networks, and others.

APACHE HADOOP BIG DATA

While Apache Hadoop is typically being deployed in dedicated racks and is not integrated within the virtualized cloud infrastructure, many customers are building out several Big Data compute racks and are offering these to their business analytics communities as a service. Rather than one individual business community owning these compute racks, they are making this technology available as a utility. Business communities leverage a time-sharing approach, where they are allowed to load their data sets and run their analytics for a dedicated period of time, and are then removed from the cluster based upon another community being in the queue.

Time to job completion is the key SLA requirement because each community only has a given period of time to uncover actionable business data based on structured and unstructured data searches and analytics. The faster that structured and unstructured searches can be completed, the better. The network plays a vital role here because it offers topology location data, which helps in localizing each search closest to where the data is stored. The key technology component is MapReduce and the ability to feed network topology data into these search algorithms. Moreover, handling and reporting on microburst conditions for determining bottlenecks helps with search placement decisions.

Apache Hadoop Big Data requires several cloud networking pillars. Distributed congestion, microburst, and load-balancing control, as determined within the switch control and forwarding planes, are critical to ensuring that no packets are dropped and achieving the best time to completion results. Offering a real-time external interface with topology data, as well as node mapping awareness, fosters Hadoop open-source developer and commercial application (eg, Cloudera) integration with MapReduce technologies. Providing event triggers based on congestion and over-subscription as they happen in real time helps in redirecting searches to other racks where the network has more capacity. These are all components of Arista EOS.

SDN CONTROLLERS

There is a clear and growing need for cloud controllers. Use cases such as VM mobility, multi-tenant traffic isolation, real-time network path tracing, firewall rule updating, and customized data captures are driving the need for greater programmability. Controllers that are external to the forwarding and control plane of the network platforms provide a programmable mediation layer between the VM service requirements and infrastructure in which the VM is hosted. Controllers translate these service requirements into actionable control and forwarding logic to the compute, network, storage, and application service platforms. These infrastructure platforms, including the network switches, take action based on the input coming in from the controller.

Because there is a growing diversity of use cases, onboarding technologies, and user communities (private, public, and hybrid), there is no universal form factor or agreed-upon set of standards for how a controller mediates and interacts. The controller market is in its infancy with startups, open-source offerings, customer-developed offerings, and infrastructure system offerings with proprietary embedded controllers (see Table 3). This requires an open, highly programmable approach in integrating with the various controller form factors and use case implementations. Arista's CloudVision provides a common framework for tight integration with all network virtualization controllers.

Table 3: Arista CloudVision Integration

| Controller | Details |
|--|---|
| OpenFlow | Open Flow 1.3 integration including Open Daylight, NEC. |
| OpenStack | OpenStack Neutron ML2 plug-in; Partners include: VMware , Red Hat, Rackspace, Mirantis, SUSE |
| VMware | Native VMware integration with vSphere 5.0/6.0, vCloud, NSX-MH, NSX-V |
| Microsoft | Integration with Microsoft System Center, Hyper-V, Network Controller and Azure Cloud Services |
| Nuage Networks | Integration with Nuage Networks Virtualized Services Platform |
| Infinera | Integration with Infinera Transport SDN |
| Network Telemetry, Monitoring, Management and Security- F5, Riverbed, Palo Alto Networks, VMWare Vrealize Operations. A10, Fortinet, HP OneView, Dell ASM , EMC SMARTS, SPLUNK, ServiceNow, VM Turbo | Native API calls being developed with key partners; enables network automation for workload mobility and visibility, network management and service insertion including security. |

Arista is focusing its efforts on the controller vendors that best align with these use cases and the markets for which Arista switches are best optimized. The underpinnings of this integration are centered on Arista EOS and the ability to interact with external controllers in real time, while updating the control and forwarding plane across the topology. This integration requires a highly scalable, transaction-based, real-time database and a modern message-passing network operating system architecture. This is a core technology component of Arista EOS.

Arista is integrating with many different controller form factors and industry leaders. This integration includes CloudVision agent based integration with the open source distribution of OpenFlow (specifically version 1.3), and several unique use cases that are again agent (OpenFlow) based, including integration with Open Daylight and NEC controllers.. Moreover, Arista has been an active contributor within the OpenStack Neutron project and has developed a dual stack ML2 driver for unifying physical and virtual network device configurations. OpenStack is compelling for many service providers that want to offer their own customized branded services with an open-source service catalog, provisioning, and operations management architecture and Arista CloudVision has support for Redhat, Rackspace, Mirantis, VMware, SUSE openstack distributions. Finally, Arista has developed a way to extend the capabilities of OpenFlow with controller-less operation using Arista DirectFlow to enable direct CLI and eAPI control over specific flow switching operations. This interface provides machine-to-machine communication for dynamically programming the service path between firewall, load balancing, and other application layer service optimizers.

SUMMARY: THE NETWORK IS THE APPLICATION

The Arista SDCN approach embodies many of the academic and design principles of software-defined networks; however, the company takes a more surgical view based on the scalability, virtualization, mobility, and automation needs that are specific to cloud computing. Ethernet switching is well advanced and there are many distributed forwarding capabilities that offer scalability and resiliency for many of the world's largest data centers.

Clearly, cloud technologies and the operational benefits of cloud automation and optimization drive new requirements for external controllers, whether it is for abstracting the services with single points of management or for defining unique forwarding paths for highly customized applications. Arista fully embraces these principles. Arista has defined four pillars that are based upon a highly modular, resilient, open, state-centric network operating system, Arista EOS, into which developers and end-user customers can add their own scripts and management tools. Arista continues to build upon this operating system, which is the key building block for SDCN. Arista's unique offerings include applications that provide workload mobility, monitoring and visibility, and real-time network telemetry for integration with cloud operations and administration tools.

GLOSSARY

Command-Line Interfaces (CLIs): CLIs are the de-facto standard for configuring, checking, archiving, and obtaining switch status. CLI is a prompt-driven, character-driven, rudimentary programming language and requires a strong technical and syntax understanding of the underlying switch operating system. CLIs are typically used on a per-device basis and offer a fast, direct interface for changing and obtaining feature-by-feature switch information.

System administrators that are technically advanced use CLIs. These administrators have a deep understanding of the capabilities of the switch.

Simple Network Management Protocol (SNMP): SNMP was authored in the late 1980s and is a higher-level, more-abstracted interface for managing switches and routers when compared to CLI. SNMP is the de-facto interface for many GUI-based management applications. SNMP requires an agent (SNMP agent) on the switch device. Agents can support read-only and read-write operations. SNMP agents expose management data, specifically information that is contained within a management information base (MIB). MIBs package a series of low-level information and send that information to centralized management stations that have registered and are authorized to receive MIB data.

Extensible Messaging and Presence Protocol (XMPP): XMPP is an IETF-approved standard for instant messaging and presence technologies. XMPP is gaining traction as a formalized protocol for communicating state information from switches to a centralized control point (controllers). XMPP employs

client/server architecture. The switches communicate to a central controller or controllers, but they do not communicate as peers between each other. There is no one authoritative (server) controller, thus offering various implementations that are well suited for cloud applications. XMPP offers a multi-switch message bus approach for sending CLI commands from a controller to any participating switch or groups of switches.

OpenFlow Protocol: The OpenFlow protocol offers an approach for communicating between switches and a centralized controller or controllers. This protocol, like the other protocols, is TCP/IP-based, with security and encryption definitions. The protocol uses a well-known TCP port (6633) for communicating to the controller. The switch and the controller mutually authenticate by exchanging certificates that are signed by a site-specific private key. The protocol exchanges switch and flow information with a well-defined header field and tags. For more information, please refer to the OpenFlow Switch Specification.

OpenStack: OpenStack is at a broader program level. It goes beyond defining a communication interface and set of standards for communicating with a centralized controller. OpenStack has more than 135 companies that are actively contributing, including representation from server, storage, network, database, virtualization, and application companies. The goal of OpenStack is to enable any public or private organization to offer a cloud computing service on standard hardware. Rackspace Hosting and NASA formally launched OpenStack in 2010. OpenStack is free, modular, open-source software for developing public and private cloud computing fabrics, controllers, automations, orchestrations, and cloud applications.

Virtualization APIs: Several APIs are available within hypervisors and hypervisor management tools for communication with Ethernet switches and centralized controllers. These APIs and tools define affinity rules, resource pools, tenant groups, and business rules for SLAs. Moreover, these tools automate low-level server, network, and storage configurations at a business policy and services level. This automation reduces the points of administration and operation costs every time a new VM is added or changed, when it is operational within a cloud.



Santa Clara—Corporate Headquarters

5453 Great America Parkway

Santa Clara, CA 95054

Tel: 408-547-5500

www.arista.com

Ireland—International Headquarters

4325 Atlantic Avenue

Westpark Business Campus

Shannon, Co. Clare, Ireland

Singapore—APAC Administrative Office

9 Temasek Boulevard

#29-01, Suntec Tower Two

Singapore 038989

Copyright © 2015 Arista Networks, Inc. All rights reserved. CloudVision, and EOS are registered trademarks and Arista Networks is a trademark of Arista Networks, Inc. All other company names are trademarks of their respective holders. Information in this document is subject to change without notice. Certain features may not yet be available. Arista Networks, Inc. assumes no responsibility for any errors that may appear in this document. 06/15