Introduction

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

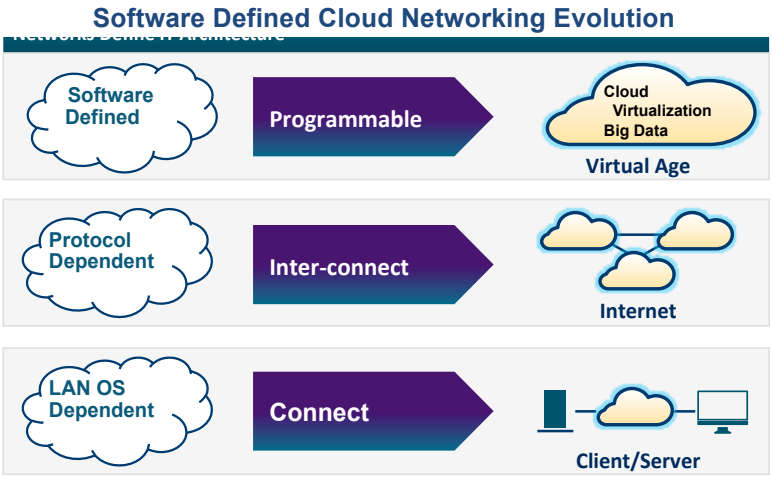
Emerging SDN technologies complement data center switches by automating network policies and provisioning within a broader integrated cloud infrastructure ecosystem. Arista defines the combination of SDN technologies and Arista EOS® (Extensible Operating System) as Software Defined Cloud Networking (SDCN).

Cloud Technology Shift

Ethernet networks have evolved significantly since their inception in the late 1980s, with many evolutionary changes leading to the various switch categories that are available today. Data center switching has emerged as a unique category, with highly dense 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 (measured in nanoseconds), zero-drop packet failover when failing over to redundant links, traffic load balancing for increased asset optimization, and scaling in support of large carrier-class virtualized infrastructures.

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 network operating systems to open, extensible, externally programmable operating systems. This open extensibility requirement is driven by the guiding principals of cloud data centers in which resources are managed dynamically as one integrated system made up of compute, network, and storage. Controllers that are external to the switches drive many of the hosting decisions and, as a result, must interface to the switches at the network edge to ensure appropriate service mappings.

Closed network operating systems that are built on older design principals 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.



Need for Arista SDCN

Prior to the shift toward virtualization and elastic computing, there were highly specialized infrastructure administrators—including server, network, storage, and application specialists—that configured services within their domain statically, based on infrequent change requests that came from the application community. On average, it took two to four weeks to configure, test, 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, updated MAC learning tables, 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 level agreement (SLA) management tools have become a core cloud infrastructure requirement.

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 VMs and the physical 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 location changes of the VMs.

In these 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 closed, a 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 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.

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 & provisioning
Network Virtualization	Scalable, multi-tenant virtual networks	Workload mobility enabled with VXLAN, NVGRE
Server Hypervisor	X86 bare metal server abstractions	Elastic computing, resource optimization, non disruptive server updates, upgrades
Storage	Network, Direct attached, SSD, Hadoop Big Data	Centralized VMDK for app mobility, software patches
Cloud Enabled Network	Arista EOS	Open, programmable, for custom flows, VM mobility, automated tenant onboarding

Distributed or Centralized Control?

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, therefore 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.

Arista SDCN should be approached with careful research because it is not the panacea for all switching and routing control and data plane functions. While SDN is driving open standards for interfacing with networking platforms in a more service-oriented approach with centralized controller architectures, its ability to instantaneously redirect traffic in large topologies is unproven. In some cases, even with a well-architected active/active or active/standby external controller, these controller implementations may never achieve the instantaneous failover or real-time congestion behavior that distributed network forwarding delivers today. As a result, controllers are now being tested in very limited proof-of-concept and production-level data center infrastructures. Conversely, traditional data center switches are deployed across the globe and are relied upon for a majority of the world's most demanding scale-out applications.

SDCN should be approached with careful research, as it is not the panacea for all switching and routing control and data plane functions. While SDN is driving open standards for interfacing with networking platforms in a more service oriented approach with centralized controller architectures, its ability to instantaneously re-direct traffic in large topologies is unproven. In some cases, even with a well-architected active/active or active/standby external controller, these controller implementations may never achieve the instantaneous fail-over, or real-time congestion behavior that distributed network forwarding delivers today. As a result, controllers are being tested today in very limited proof of concept and production level data center infrastructures. Conversely, data center switches are deployed across the globe and are relied upon for a majority of the world's most demanding scale out applications.

Distributed or Centralized Control?

Distributed Control Advantages	Centralized Control Advantages
Resilient self-healing	Active/standby, active/active clustering
Mature Layer 2 and Layer 3 well-known protocols	Faster prototyping based on source and less consensus-open building for agreeing on peer-to-peer protocols
Hardware-optimized learning and forwarding	Customized flows based on broader VM service definitions
Mature troubleshooting best practices	Centralized point of management for reviewing configuration databases
Traffic load balancing and link-level failover	Designed for large scale with co- dependency on network platform interfaces

Best of Both Worlds

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 for their specific use cases.

These are some of the more common Arista SDCN use case:

- 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.

Can All Switches Support Arista SDCN?

An open modular network operating system with the ability to respond in real time to both internal and external control operations is required to support SDN. Unfortunately, not all switch operating systems offer this capability because many of them were architected a decade or two ago, when the need for cloud and the interaction with external controllers was not envisioned. These older operating systems typically interact internally through a proprietary message-passing protocol and externally with non-real-time state information (or application programming interfaces [APIs]). Many configuration, forwarding, race, and state problems arise when multitasking occurs in real time with multiple systems, as in the case of communicating with external controllers while trying to resolve topology changes. The message-passing architectures of these legacy switches prevent these operating systems from quickly and reliably multitasking with external controllers.

A modular network operating system that has been designed with a real-time interaction database, with API-level integration both internally and externally, is a better approach. The system can, therefore, integrate and scale more reliably. In order to build a scalable platform, a database that is used to read and write the state of the system is required. All processes, including bindings through APIs, can then transact through the database in real time, using a listener and subscriber message bus. Multiple systems, both internally and externally, can subscribe, listen, and publish to this message bus. A per-event notification scheme can allow the model to scale without causing any inter-process dependencies.

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, and 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 principals and is a result of an IETF submission by VMware, Arista, and several other companies.

It is important to recognize that building such largely 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 client/server architecture).

Arista has identified these cloud challenges and has been solving them methodically using the four pillars of software-defined networking:

Pillar 1: *Multipath Active-Active Data Path Leaf/Spine Scaling*: 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 50,000 or more compute and storage nodes, both physical and virtual

With the advent of next-generation multi-core server CPUs, as well as dense VMs and storage, this type of uncompromised Leaf/Spine 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 and Layer 3 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 multiyear, expensive undertakings. Customers will receive proprietary implementations with vendor lock-in at the controller level as well as in many of their non-standard distributed forwarding protocols. Arista has seen these issues first-hand. 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 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. This effort has resulted in several new IETF-endorsed tunneling approaches that Arista openly embraces, including VXLAN from VMware and NVGRE from Microsoft. The net benefit is much larger mobility domains across the network. This is a key requirement for scaling large clouds.

Pillar 4: *Single Point of Management*: Customers that are deploying next-generation data centers are challenged with managing and provisioning hundreds (or possibly thousands) of networking devices. Simply put, it is all about coordinating network policies and configurations across multiple otherwise-independent switches. Arista EOS provides a rich set of APIs that use standard and well-known management protocols. 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 has a rich API infrastructure that includes OpenFlow, Extensible Messaging and Presence Protocol (XMPP), System Network Management Protocol (SNMP), and the ability to natively support common scripting languages such as Python. The Arista Extensible API (eAPI) product scales across hundreds of switches and provides an open programmatic interface to network system configuration and status. eAPI integrates directly with Arista EOS SysDB and delivers a standardized way to administer, configure, and manage Arista switches, regardless of switch type or placement within the network.

Arista EOS Application Extensibility

Core to successful implementation of Arista SDCN is the extensibility of Arista networking operating system. While the modularity, distributed scalability, and real-time database interaction capabilities of Arista EOS are mentioned throughout this document, there are other aspects to consider as well. These considerations include the ability to write scripts and load applications (such as third-party RPM Package Managers [RPMs]) directly onto the Linux operating system, and to run these applications as guest VMs. Arista provides a developer's site called "EOS Central" for customers that are interested in this hosting model.

Applications that are loaded into Arista EOS as guest VMs run on the control plane of the switch, which offers various benefits:

- The decoupling of data plane forwarding (or silicon) from the control plane (or software) enables deploying applications on the switch with no impact on network performance.
- The x86 control plane of Arista switches (multicore x86 Xeon-class CPU, with many gigabytes of RAM) running atop Linux enables third-party software to be installed as-is without modification.
- Arista switches optionally ship with an Enterprise-grade solid-state drive (SSD) for additional persistent storage and Arista EOS extensibility, which can be used to access third-party storage via Network File System (NFS) or Common Internet File System (CIFS).

- Arista switches provide scripting and Linux (or bash) shell-level access for automation.

Proof points of these benefits include the ability to run cloud infrastructure automation applications (such as Chef, Puppet, or CFEngine) and network analytics applications (such as Splunk for traffic analysis and visibility).

This table maps Arista SDCN requirements to the capabilities within Arista EOS:

SDCN Four Networking Pillars

Cloud Networking Requirements	Arista's Extensible Operating System Pillars
Highly resilient, link optimized, scalable topology	IEEE/IETF Standard Protocols MLAG/ECMP Topology protocols
Cloud Adaptation/Control Plane	Single binary image for all platforms Zero touch protocol for rapid platform deployment Industry support for Openflow, OpenStack
Network Virtualization	Hardware based VXLAN, NV-GRE VM tracing for troubleshooting Integration with hypervisor controllers
Single Pane of Management	Well known interfaces into EOS including XMPP, XML, RESTful API, E-API, standard Linux utilities

Use Cases for Arista SDCN:

Network Virtualization

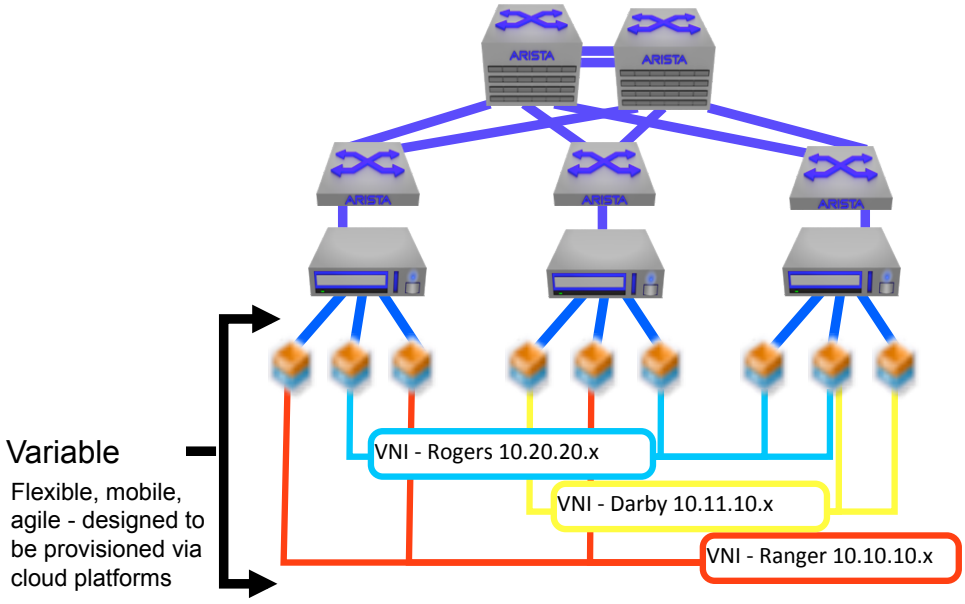
Network virtualization is vital because the network must scale with the number of VMs, tenant partitions, and the affinity rules that are associated with mobility, adjacency, and resource and security policies. Moreover, IP mobility where the VM maintains the same IP address, regardless of the Layer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

Network Virtualization 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, providing 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.

VXLAN Mobility Across Traditional Network Boundaries



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.

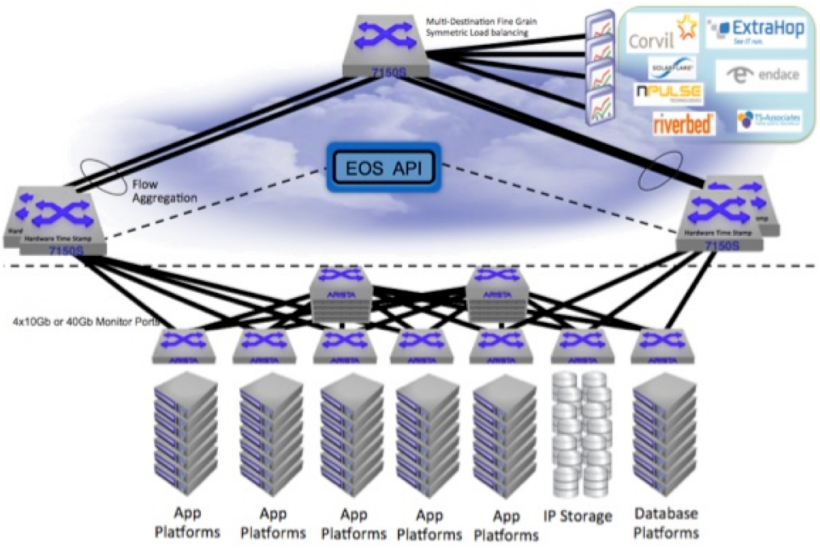
The network is the ideal place for directing, collecting, filtering, analyzing, and reporting on the majority of these vertical market compliance and SLA management requirements. However, given the volume of traffic, the number of applications and associated VMs for each cloud tenant, and the high-speed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

One approach to reducing this problem is to provide customized data taps—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 that 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.

Data Tap Aggregation



Service Routing

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 controllers, 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 an XML-based, RESTful 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.

Hadoop Big Data

While Hadoop Big Data 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, 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.

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 (called 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 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. This requires an open, highly programmable approach in integrating with the various controller form factors and use case implementations.

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.

From an implementation perspective, Arista is integrating with many different controller form factors and industry leaders. This integration includes EOS agent based integration with the open source distribution of

OpenFlow (specifically version 1.0), Floodlight (with Big Switch Networks), and several unique use cases that are again agent (OpenFlow) based including integration with NEC, Aruba, and Microsoft. Moreover, Arista has been active contributor within the OpenStack Quantum project and has developed a dual stack 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. 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.

Arista is Open to All Controllers

OpenFlow integration including BSN, Beacon, Floodlight,NEC integration	OpenStack support with Quantum API integration with partners including Nebula and Piston	Native VMware integration into vSphere and vCloud - VXLAN integration	Native API calls being developed with key partners. Enables network automation through event manager

Summary

The Arista SDCN product 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 that is commonly referred to as 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.

Glossary

Command-Line Interfaces (CLI): 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.

Network Configuration Protocol (NETCONF): NETCONF is an IETF protocol for configuring, changing, and deleting switch settings. NETCONF can also be used for monitoring. NETCONF uses textual data representations that can easily be changed. The NETCONF protocol uses Extensible Mark-up Language (XML) for data encoding because this format is well known. Regarding how to configure, monitor, or change any settings within a switch or router, NETCONF offers the best of both worlds when compared to a CLI or SNMP approach.

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 multiswitch 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, automation, orchestration, and cloud applications.

Virtualization APIs: There are several APIs that 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, therefore reducing the points of administration and operation costs every time a new VM is added or changed, once it is operational within a cloud.