Introduction

Ethernet networks have evolved significantly since their inception back in the 1980s, with many generational changes to where we are today. Networks are orders of magnitude faster with 10Gbps line rate switching as the norm. Moreover, today's Ethernet networks offer sub-microsecond switch latency, traffic scaling and load balancing across redundant interfaces, and are reducing the management complexities with state-driven architecture and open interfaces for managing hundreds of switches through global, yet highly secured interfaces.

Thousands of man-years of engineering effort have gone into developing features for enterprise switching and routing with each vendor developing their own software stack, thus creating a system where the forwarding logic and the hardware to switch millions of packets per second remains closed.

Software Defined Cloud Networking (SDCN) is a term often used when a controller external to the forwarding logic and the actual switch itself programs the network devices to alter or enhance the flow of traffic.

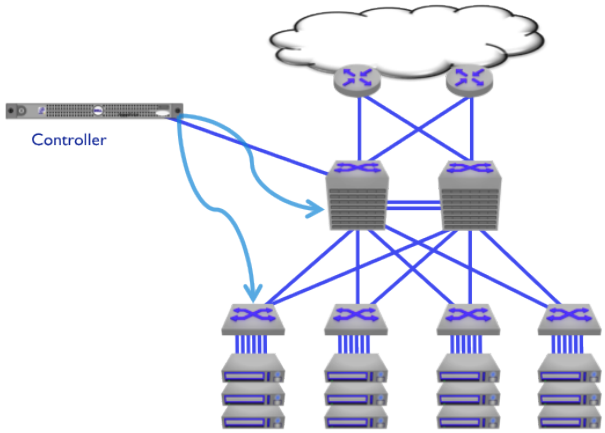

Figure 1: SDCN via a controller

SDCN can be achieved via standard APIs that several hardware and software architectures intend to support.

Need for SDCN

The need for SDCN arises primarily in places where the standard behavior of a switch or a router is not fully optimized for traffic flow on a unique, or per application basis. Moreover, where there is a need for centralized management with a great deal of service abstraction, such as a cloud, all of the infrastructure components are reduced to a few sets of programmable services at the subscriber level.

Consider the case of a data center or cloud operator. In a Data Center, the physical setup of servers, switches and interconnects is well known to the operator. 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, configuration files, etc. are imported into asset tracking and configuration database applications. This database information is very important for pinpointing problems and performing break/fixed tasks in an extremely efficient manner. If any of this data changes either with MAC learning, aging, and ARP refreshes then the asset tracking and database applications need to be updated by the operations team in timely manner. This creates a

human dependency that is often difficult to keep current and accurate.

In such cases, the operator should not have to worry about MAC learning, aging and ARP refresh and uploading any changes into the database. The path from one device to another is known upfront and, if programmed in advance by the operator external to the switch, can make network configuration and troubleshooting a lot easier, as there are far less forwarding table changes, and therefore less synchronization requirements with these asset tracking and database applications. A majority of data center switches across the industry do not allow any forwarding path programmability from the outside. They are closed black box that the vendor controls, with pre-set forwarding path software and hardware algorithms. This is a clear case where an external controller offers value.

Similarly, there are other use cases - in traffic engineering for network taps, adding special headers to overlay layer 2 traffic over layer 3 networks, classifying traffic based on content, monitoring congestion and hash efficiency over a LAG (Link Aggregation) or ECMP (Equal Cost Multi-Pathing) group, etc. Programmable switches, managed by external controllers, can address many of these cases.

Distributed vs Centralized Control

SDCN as a concept is often viewed as the next big thing in networking. However, it needs to be approached with caution as there are many reasons why switches and routers have evolved with many of the controller functions distributed and embedded within each switch node. With protocols such as LACP, OSPF, or BGP, networks have been designed to meet a wide variety of business critical application needs. In addition, when each network device operates independent of the rest, there are high resiliency and hit-less fail over behaviors that offer 24x7x365 uptime. A majority of data centers have become highly dependent on these capabilities. With a centralized controller, there needs to be further evolution to address the resiliency needs. In some cases, even with a well-architected active/active or active/standby external controller, these controller implementations may never achieve the fail-over, or real-time congestion behavior that distributed network forwarding accomplishes.

Advantages of Distributed Control	Advantages of Centralized Control
• Resilient network - independent devices • L2 or L3 with standard protocols • Hardware based learning and forwarding • Well understood troubleshooting tools	• Optimize specific flows • Abstraction of the applications from the underlying infrastructure protocols and address schemes • Invent new protocols without the time lag to implement directly into switches and routers • Single point of management, without proprietary forwarding • Design for ultra-large scale

Best of Both Worlds

Networking is critical to every IT organization building a cloud, large or small. As a result, no one is likely to compromise resiliency over traffic flow optimization. The approach that is well suited for most companies is to let the network layers do their job with standard protocols and to use SDCN to enhance the behavior for their specific use cases. Common use cases for SDCN are noted in later sections of this document.

Can all Switches Support SDCN?

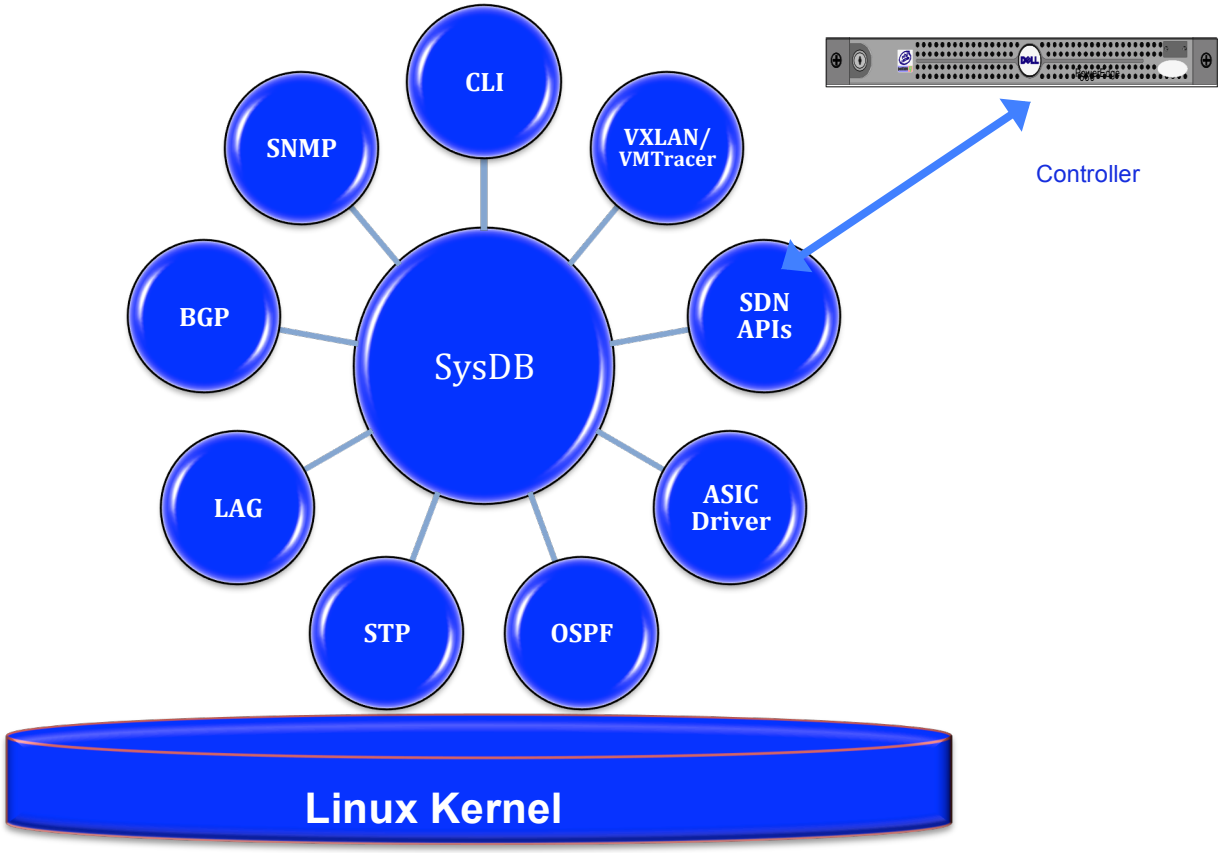

Figure 2: Arista EOS Architecture with API Support is foundation for SDCN

Not all switches are created equal. Network Operating Systems (NOS) require a clean architecture to support both local control and an external controller programming forwarding logic concurrently. Most systems have not been built with these design principals. Processes typically interact through message passing and most bugs arise in this interaction when there are multiple systems (internal and external) to interact with. The interactions required here prevent these systems from scaling.

An operating system that has been built ground up with such interactions factored in can, however, support the needed functionality. In order to build a scalable platform, a database used to read/write all state of the system is required. All processes, including bindings to the SDCN controller through APIs can then transact through the database. A per-event notification scheme can allow the model to scale without causing any inter-process dependencies. Most operating systems designed for networking do not have a database with a message bus and, as a result, are weak starting points for SDCN.

The Four Pillars of SDCN:

At Arista, we believe that Ethernet scaling from 10Gigabits to 40 Gigabits to 100Gbit Ethernet and even Terabits, with well-defined standards and protocols for L2/L3, is the optimal approach for a majority of companies building

clouds. This allows large cloud networks of 10000+ physical and virtual server/storage nodes today, and scaling to 100,000+ nodes in the future without re-inventing the Internet or having to introduce proprietary tags. At VMWorld 2011, VMware announced an exciting technology called VXLAN (specification co-authored by Arista for IETF submission) that enables large-scale cloud networking, reflective of these views. VXLAN embodies several of the SDCN design principals.

It is important to recognize that building such massively scalable and dense clouds is only part of the equation. Application mobility, storage portability, self service provisioning/automation, and dynamic resource optimization create new management and operation challenges different from many of the legacy data centers, including those designed in the late 1990s for web hosting. Arista has identified these challenges and has been solving them methodically, step by step as the four pillars of software defined networking.

Pillar 1: Multi-path Active-Active Data Path leaf-spine Scaling: Scaling Cloud networking across multiple chassis via MLAG (Multi-chassis Link Aggregation Groups) at L2 or ECMP (Equal Cost Multi Pathing) at L3 is a standards based and scalable approach for uncompromised cloud networking. This insures effective use of all available bandwidth in a non-blocking mode, while providing failover and resiliency when any individual chassis or port has an outage condition. Together, they cover all of the important multi-path deployment scenarios in a practical manner while not introducing any proprietary inventions. These technologies currently scale to 50,000+ compute and storage nodes, both physical and virtual.

With the advent of next generation multipath server CPUs, dense virtual machines and storage this type of uncompromised leaf-spine topologies with non-subscribed capacity, uplink, downlink and peer ports becomes paramount. The introduction of tunneling technologies (MAC in IP), make hybrid L2/L3 topologies possible as well. These specific functions are best optimized in hardware and with Arista's Extensible Operating System (EOS™), a modern open software system designed for advanced network operations.

Pillar 2: Single-image L2/3 control plane: Some vendors are trying to recreate three decades of networking control plane architecture work with a non modular, non database centric proprietary starting point in response to Software Defined Networking initiatives. These are multi-year, expensive undertakings with a history of vendor lock-in that disregards the IETF and IEEE work done to-date on Internet protocols. At Arista, we find ourselves debugging these products, box-by-box, as part of our interoperability testing work. Many of these non-Arista switches have poorly documented protocols, are very hard to implement, and have proprietary tools for configuration and management. This is not the answer going forward.

Instead of these touted proprietary "fabric" approaches, standards based L2/L3 IETF control plans specifications plus OpenFlow options (without hype) can be a promising open augmentation for providing single image control planes in the future. OpenFlow 1.1 implementations in the next few years will be based on specific use cases and the instructions the controller could load into the switch. Arista's Zero Touch Provisioning (ZTP) for automating deployments of compute racks as well as Latency Analyzer (LANZ) for detecting application-induced congestion are also examples of innovative operational controls.

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 handle any application anywhere on the network. This drives economy of scale for cloud operators. This network-wide virtualization is an ideal use case in which an external controller abstracts the virtual machine from the network and defines the mobility and optimization policies with a greater

degree of network flexibility than what is currently available today. This requires a tunneling approach and external APIs in which external controllers can define the forwarding path. Arista has been leading this effort with VMware and Microsoft which has resulted in several new IETF endorsed tunneling approaches in which Arista openly embraces including VXLAN from VMware and NV-GRE from Microsoft. The net benefit is much larger mobility domains across the network. This is a key requirement for scaling large clouds. In short, we are seeing a conceptual disaggregation of physical network topology from virtual machine mobility. In many ways, this is analogous to a network wide hypervisor.

Pillar 4: Single Point of Management: Single Point of Management has been tackled in the past by various enterprise stackable technologies that have been platform specific and throughput limited. In theory, it can be layered on top of the traditional control plane and data path of a cloud network. Simply put it is all about coordinating the configurations across multiple otherwise-independent switches. No "fabric" technology required, and no need to turn every switch feature into a distributed systems physics problem. Arista's CloudVision™ is a good standards-based example of this using XMPP or Netconf messaging methods or future APIs based on Openstack or Openflow.

What is clear is that cloud networking requires an appropriate networking software foundation (such as Arista EOS) using well-defined and open controller APIs. This facilitates communication between the network switches centrally, and external controllers, thus allowing higher layer provisioning and automation systems to determine the mobility and forwarding domains of the applications within the cloud. The network becomes far more harmonious with the cloud virtualization and resource optimization tools. The diagram and figure 4 below show many of these principals.

Software Defined Networking Requirement	Arista EOS Pillars - Standards Option
Advanced Cloud Topology	IEEE/IETF Standards, not proprietary tags MLAG/ECMP SW based, TRILL for Future Tools based on Standards (example Arista Cloud Vision)
Cloud Control Plane	Single Image Binary EOS for L2/L3, ZTP, LANZ OpenFlow 1.X Future ONS APIS
Single Pane of Management	Tools based on Standards (example Arista Cloud Vision)
Network Virtualization	Arista VMTracer for VMWare vCloud VXLANS today OpenVirtualization Switch (OVS)Controllers – Future

Figure 3: Four pillars of SDCN

Applications for SDCN:

Network Visibility with VMWare VXLANS/Controllers

In a virtualized data center, virtual machines move from overloaded to underutilized servers. This flexibility improves utilization and reduces costs. However, it complicates debugging. When the application team asks the network team to help debug an application performance problem, merely finding the application can be challenging, particularly if the server virtualization management tools are outside of the network team's control. Network equipment vendors are responding by integrating their products with virtualization controllers to improve visibility. For example, Arista EOS supports VMtracer™, a switch service that automatically locates virtual machines within a virtualized data center, providing the network operator with a list of ESX servers and associated VMs on a per-interface basis as shown in Figure 4.

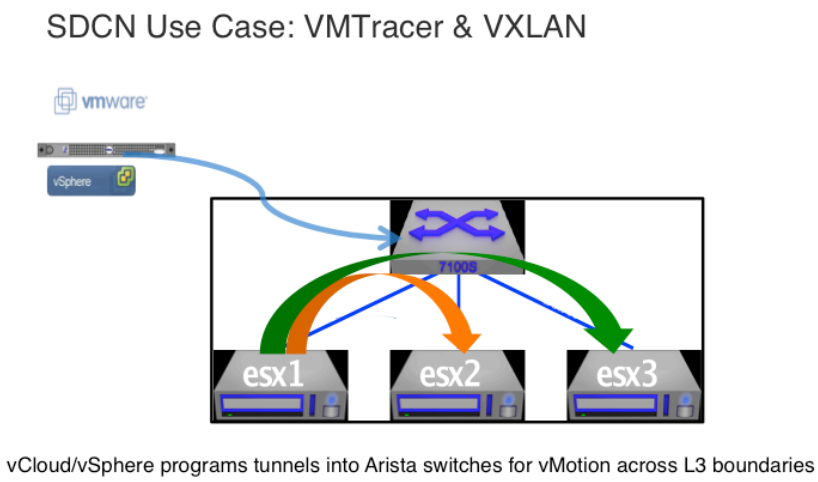

Figure 4: SDCN Use Case

Cloud Control & Automation with ZTP and VMTracer

Operating efficiently at cloud scale requires automating network configuration tasks. For example, you can automatically rack and connect across the switch and compute farms using Arista's Zero Touch Provisioning. Furthermore, as the virtualization controller migrates a virtual machine to a server, it is necessary to provision the VM's VLAN on the server's access switch. Failure to do so results in loss of connectivity for the VM. Arista EOS also supports auto-segmentation, where the switch learns from the controller about the arrival of the VM and automatically provisions the VM's VLAN on trunk links. More importantly, Arista EOS is open, enabling third parties to add software to the switch to integrate its control and/or management planes with any virtualization controller as shown in Figure 5 below.

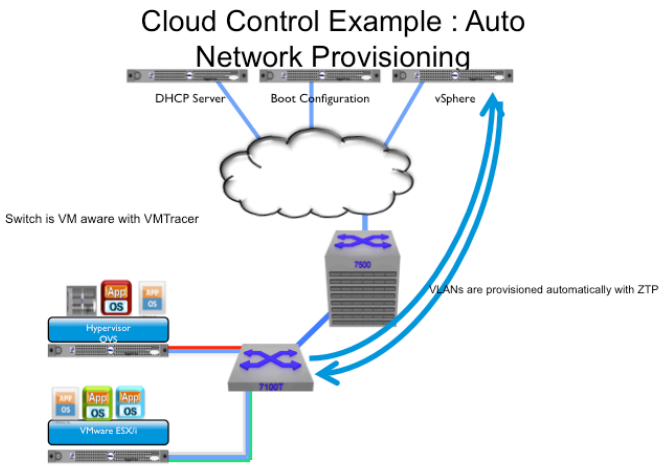

Figure 5: Cloud Control: Auto Network Provisioning

OpenFlow and Open Virtual Switch (OVS)

Contrary to popular opinion, OpenFlow and OVS are not architectures but building blocks that can be used in a Software Defined Network. The OpenFlow 1.0 specification (Arista developed a lab prototype) defines a protocol interaction between a controller (an OpenFlow Controller) and an OpenFlow client/agent residing upon a physical switch. Open Virtual Switch (OVS) is a virtual switch running within a hypervisor similar to VmWare's virtual switch but based upon open source software. An OVS can be controlled by any OpenFlow controller using the OpenFlow protocol.

As emerging building blocks for Software Defined Cloud Networks, both Openflow and OVS are in still early in development. Arista has taken a leading role in working with the Openflow and OVS community. Openflow and OVS implementations can take advantage of EOS extensive APIs and provide a seamless integration. Arista is focused on delivering solutions based upon these technologies and engaging with the Opensource community. While promising, both Openflow and OVS are in v1.0 stage of development need to mature. Early stage deployments for either of these technologies are likely to occur in lab environments, network monitoring (openflow specific flows to hardware probes), and customer proof of concept testing.

Both Openflow and OVS must address encapsulation hardware requirements, feature velocity and coverage, compatibility with non openflow networks, troubleshooting tools, reliability and fault tolerance, and scalability and performance of the controllers. Arista is working to solve these issues with our partners and developing solutions that address these critical requirements.

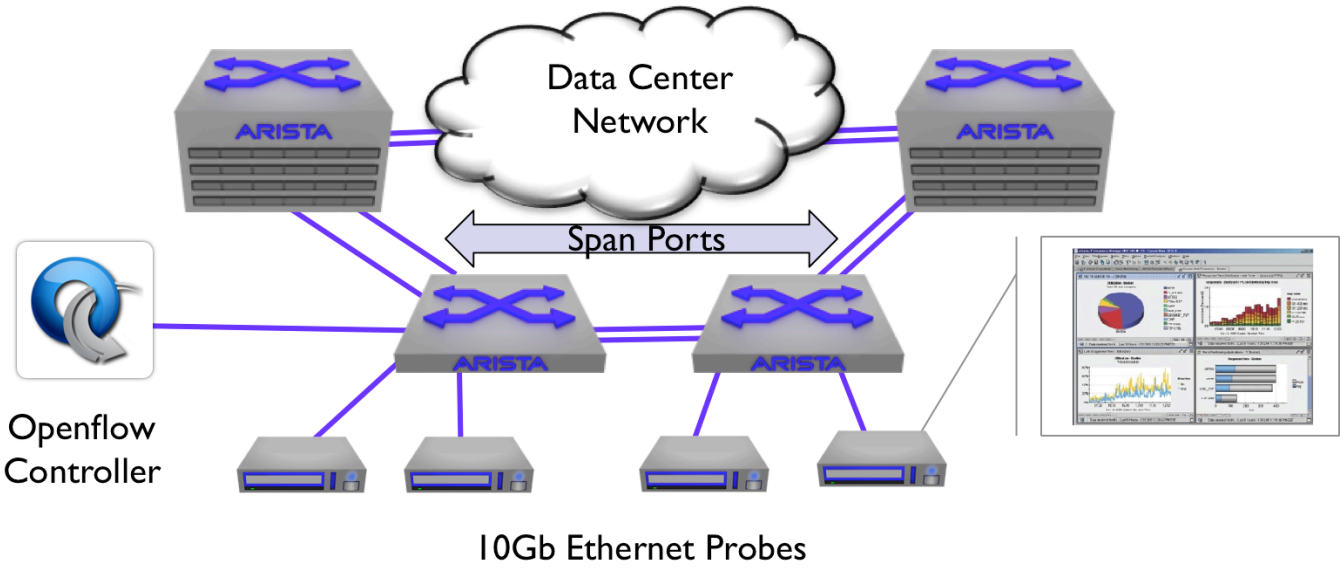

Figure 6: Network Monitoring via Openflow

Summary

Arista's Software Defined Cloud Networking (SDCN) embodies many of the academic and design principals of Software Defined Networks; however, the company takes a more surgical view based on the scalability, virtualization, mobility, and automation needs specific to cloud computing. Ethernet switching is well advanced as it exists today and there are many distributed forwarding capabilities that offer scalability and resiliency at scale. Clearly, cloud technologies and the operational benefits of cloud automation and optimization drive new requirements for external controllers, whether it be for abstracting the services with single points of management, or defining unique forwarding paths for highly customized applications. Arista Networks fully embraces these principals. Arista has defined four pillars - these pillars are based upon a highly modular, resilient, open, state-centric network operating system, commonly referred to as EOS. Developers and end-user customers can already add their own scripts,and management tools into EOS. Arista will continue to build upon this operating system, which is the key building block for SDCN.