White Paper

Infrastructure SDN with Cariden Technologies

Providing the benefits of software-defined networking in resource-constrained and dynamic demand environments today

Contents

Figures

Tables

Executive Summary

Software-Defined Networking (SDN) is succinctly defined by the Open Networking Foundation as bringing "direct software programmability to networks."[1] This brief definition hides a vast set of functionality and uses, and in practice SDN refers more broadly to logically centralized software control.

Depending on the value proposition, distinct classes of SDN have emerged, and it is useful to view their unique missions and note the services for which they typically show their benefits:

- **Flow Services SDN** addresses the wealth of security and visibility applications that become possible with flow-level programmability
- **Virtualization SDN** provides virtual network connectivity for efficiency and agility
- **Infrastructure SDN** exposes network resources for continual optimization of resources and predictable handling of diverse traffic demands

The platform requirements for an Infrastructure SDN can best be illustrated by example, through two specific use cases: Bandwidth Calendaring and Demand Placement. **Bandwidth Calendaring** provides visibility and reservations of network resources across time; examples that leverage this include data center replication, video streaming and cloud services migration. **Demand Placement** informs applications of availability of network resources in different locations; relevant examples include content cache placement and network-aware DNS. For all of these applications, the benefit is in the ability to run the network at high utilization while maintaining predictability and high availability.

Cariden's vision of abstracting and controlling physical resources through software—realized in the creation of a Network Services OS (NS-OS)—forms the core of an Infrastructure SDN solution. The NS-OS can be described in three layers:

- **Network Controllers** layer that is the equivalent of device drivers in computer operating systems providing I/O and control to specific devices (an OpenFlow controller, if present, would be integrated at this layer)
- **Network Model** layer that provides an abstract view of network resources and reservations across time
- **Services Control** layer that provides intelligence for controlling network paths, network admission and orchestration tasks

Three key, sometimes overlooked, requirements for the NS-OS that it needs to provide are the following:

- **Centralized resource mediation** such that there would be a single authoritative source for keeping track of resources and issuing them to various applications
- **Predictive analytics** (anticipation of network behavior upon failure states or other "what if" scenarios) to enable applications to schedule resilient bandwidth resources into the future
- **Multivendor/layer interface** that provides support for a wide range of infrastructure

Cariden's vision of network abstraction and control fulfills the promise of simplified control and programmability of physical resources within the SDN movement.

[1] See https://www.opennetworking.org/about.

Introduction

Software has always been an important element of networking and is now the focus of innovation in networking like never before. There is a strong movement to create network programmability standards that allow software developers to rely on network resources with the same ease as they do with compute and storage resources. This movement, called Software-Defined Networking (SDN), has taken on a broad mandate.

On one end, SDN is concerned with unbundling network control software from network hardware to allow innovation where needed and commoditization where appropriate. On the other end, SDN is concerned with providing a standardized programming interface for application developers. The field is broad, young and has a lot of buzz so there are many voices and quite a bit still to reconcile en route to clear definitions.

Classifying SDN applications by the source of value they provide is useful in making sense of the many voices around SDN:

- The first source of value is security and other services enabled by flow-level programmability; this is **Flow Services SDN**.
- The second source of value is virtualization, which can be termed **Virtualization SDN**.
- The third source of value is exposing programmability of network resources to software applications: **Infrastructure SDN**.

Figure 1 and the following sections present the main value areas, how they map to various market segments, and the most talked about “killer apps” in each value area.

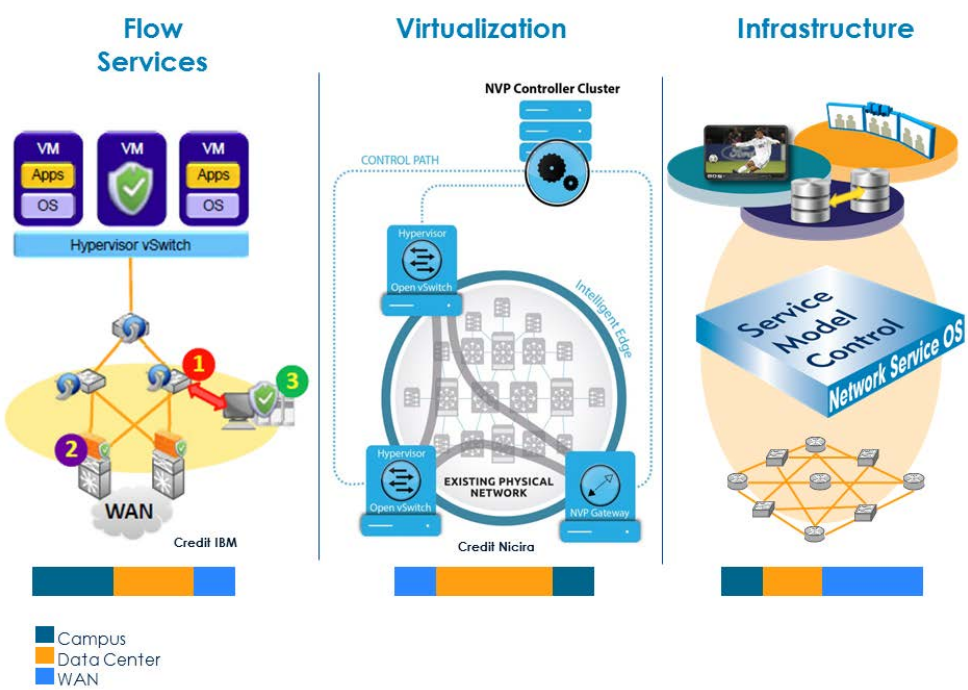

Figure 1.   Classes of SDN

Applications and benefits for each of the SDN classes are summarized in the table below.

Table 1. Applications and Benefits for the SDN Classes

	Flow Services SDN	Virtualization SDN	Infrastructure SDN
Uses	Integrated flow tapping Service pipelining	Cloud services Virtual networks	Bandwidth Calendaring Demand Placement
Benefits	These services would be prohibitively expensive or complicated without Flow Service SDN.	Flexibility and savings in networking similar to what compute virtualization has brought to servers.	Flexibility and savings from application being able to coordinate access and programming of network resources.
Notes	Most useful where user-specific services are defined (campus, data center perimeter, network edge).	A grounds-up attempt to bring together networking and server hypervisor technologies.	Jumbo networking apps (e.g., data center replication) and any that require guarantees (video streaming, cloud services) require visibility and coordination of network resources.

Flow Services SDN

OpenFlow and associated technologies bring a revolutionary ability for software applications to interact with individual flows or an aggregation of flows. This translates to streamlined perimeter security and other applications requiring flow-level visibility.

For example, in the past one would have needed to resort to specialized sFlow and NetFlow instrumentation platforms for flow-level visibility and specialized packet processing platforms for control of the flows (e.g., detecting and scrubbing denial of service attacks). Now, one can program this functionality and a host of others more easily using only the switches at hand.

This powerful innovation is applicable to enterprise[2] and service provider[3] markets, across campus, data center and WAN networks. However, the distinguishing factor of being able to see and control microflows is less significant in the core where operators mostly deal with aggregate flows.

Virtualization SDN

Compute virtualization has revolutionized the use of servers in data centers but the network component of virtualization in data centers has lagged behind. Virtualization SDN aims to make orchestration of moves, adds, and changes as easy to manage for networking equipment (firewalls, VPN aggregators, switches and routers) as their computer-centric forbears achieved for processors, memory, and storage arrays.

According to Martin Casado, CTO of Nicira,[4] SDN natively isolates services and hardware and reassigns them more flexibly and quickly than any other type of network virtualization.[5] As before, there is also an

[2] See *Enterprise Data Center Security with Software-Defined Networking and OpenFlow* at https://amphionforum.com/wp-content/uploads_a/2012/05/Enterprise%20DC%20Security%20with%20SDN%20and%20OpenFlow%20-%20public.pdf.
[3] http://www.cisco.com/en/US/solutions/collateral/ns341/ns973/ns1081/brochure_c02-630395.pdf.
[4] Acquired on by VMware on July 23, 2012.
[5] See http://www.networkworld.com/news/2012/061812-openflow-260157.html?page=2.

advantage of getting *easily scalable and cheaper infrastructure*[6] by separating the hardware and software components of networking.

One of the principles of virtualization is decoupling the virtual and physical aspects of the network. The virtualization layer shifts the headache of dealing with physical network resources from the application point to the virtual switch. The next section is concerned with providing a programmable resource interface to applications or to the virtual switches that support those applications.

Infrastructure SDN

Infrastructure SDN draws value from providing visibility into and programmability of the physical network resources. Infrastructure SDN is key wherever resources are constrained (e.g., because of expense, geographic, or power/space limitations) or there are dynamic changes (failures, traffic changes) not under the control of the operator. The WAN is always expensive and hence resource-constrained. The data center is resource-constrained from an applications point of view when applications wish to place, reserve and control large traffic flows.

Infrastructure SDN is of particular interest to service providers who must bring about operational efficiencies to fight declining ARPU and need and to bring back profit margins through a higher quality of service. It has risen to the level of a matter of survival. "It's going to be a case of *The Service Provider Strikes Back*" noted Matthew Finnie, CTO of Interoute, referring to the prospects of software control for managing the network and creating cloud applications.[7] And Stuart Elby, the VP of Network Architecture and Technology at Verizon, noted that "Like all disruptive technologies, [SDN] will change the economics and also change the playing field" for service providers and their suppliers.[8]

The key drivers for Infrastructure SDN are the benefits achievable from providing information, access and control of network resources to application software that has been operating at arm's length with respect to the network until now. Data center replication software, for example, has not had the benefit of a network application that would easily allow it to reserve large chunks of bandwidth for a limited period of time. Similarly, CDN cache access software has not had the benefit of querying a network application for the best places, from a network perspective, to place content.

Service Providers and SDN

Service providers have an interest in all of the SDN classes discussed above; they expect SDN technology to simplify their networks and to make them more agile for the creation of high value services. As Michael Howard, CEO and Founder of Infonetics, noted, "Carriers are driven to consider SDNs in the expectation of simplifying the way they provision, creating network services and virtual networks in ways not always possible with existing technologies—and they must do it across multivendor equipment."[9]

Figure 2 lists carrier deployment plans for various SDN technologies.

---

[6] See http://nicira.com/en/nicira-virtualizes-the-network-for-att-ebay-fidelity-investments-ntt-and-rackspace.
[7] See *Interoute's SDN Dream* at http://www.lightreading.com/blog.asp?doc_id=221994.
[8] See *Verizon: Putting SDN in its Place* at http://www.lightreading.com/video.asp?doc_id=221051.
[9] See *SDNs, 40G/100G, and MPLS Control Plane Strategies: Global Service Provider Survey* at http://www.infonetics.com/.

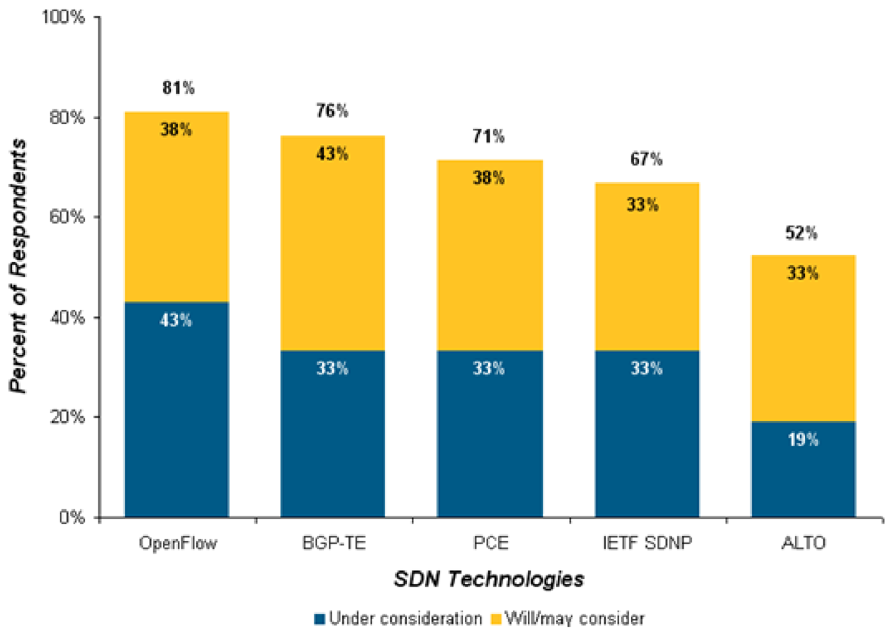

Figure 2. Programmable Network and SDN Deployment Plans (Source: Infonetics, 2012)

Much of the carriers' adoption plans center around Infrastructure SDN: Path Computation Element (PCE) computes and distributes path information to routers along a path; BGP-TE optimizes traffic across autonomous systems;[10] and IETF Software Driven Network Protocol (SDNP) enables existing control planes to more readily adapt to application requirements.[11] OpenFlow adoption can span multiple SDN classes and Application Layer Traffic Optimization (ALTO) plays a role in both Virtualization and Infrastructure SDN, as it facilitates the optimal location and construction of cloud services.[12]

Use Cases for Infrastructure SDN

Describing two of the popular uses of Infrastructure SDN, bandwidth calendaring and demand placement, is an effective way to get insight into a systems view of Infrastructure SDN.

Bandwidth Calendaring

Bandwidth calendaring allows user applications to reserve network resources across time. It is useful, for example, for scheduling data center backups, managing content distribution, and orchestrating cloud applications across the WAN. Google showcased bandwidth calendaring as part of its SDN efforts at the 2012 Open Networking Summit (ONS) and explained how a massive inter-datacenter network runs with commodity hardware, OpenFlow controllers and a bandwidth calendaring application. An excerpt from the Google presentation is shown in Figure 2.

[10] See *Programmable Networking is Safe for Work* at http://opennetsummit.org/talks/ward-wed.pdf.
[11] See presentation for the Open Networking Foundation's SDNP breakout forum at http://www.ietf.org/proceedings/82/slides/sdn-11.pdf.
[12] ALTO is described in RFC 5693 at http://tools.ietf.org/html/rfc5693.

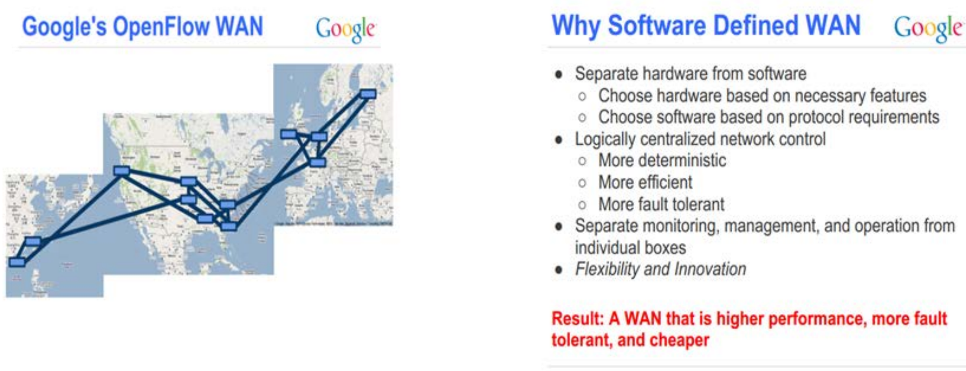

Figure 3. Google's OpenFlow WAN (Source: ONS 2012[13])

Using this new WAN, Google is able to schedule network resources dynamically, determining whether and when to admit new requests. These characteristics are useful in data center replication, cloud migration, and video streaming. A key benefit that Google was looking for—and achieved—when it set up an SDN/OpenFlow backbone was to increase link utilization of the core network from 30%-40% to close to 100%.[14]

Google explained that their solution has helped to improve backbone performance and reduce backbone complexity and cost. One key advantage is centralized traffic engineering (TE), ensuring that routers don't "compete for the best path" after a link failure. They were able to deploy this production-grade centralized TE in two months, and enjoyed hitless (no packet loss) software upgrade.

Stuart Elby of Verizon articulated a different angle on the use of SDN.[15] In addition to the requirements for next generation cloud services—new abstractions, feedback loops, and communication links between the network, service, and application layers—Elby highlighted the need for different network layers to ensure flexibility and speed. Specifically, this speaks to the need for packet and optical equipment, and traditional routers/switches in addition to OpenFlow-enabled switches to participate in the software abstraction.

Demand Placement

Bandwidth Calendaring is a logical first application to explore in Infrastructure SDN since control and service activation are the first applications that come to mind for most people. But much of the value of SDN infrastructure is unlocked just at the level of visibility. At the 2012 MPLS World Congress, John Evans of Cisco explained the value demand placement[16] could provide by locating best positions for content caches for cloud services.[17]

Demand placement provides value similar to traffic engineering without the provider having to implement TE on the underlying equipment.[18] It allows operators to determine the best places to locate sources of traffic (e.g., caches, or servers for cloud applications) before placing these demands on the network.

---

[13] A complete presentation by Urs Hoelzle, including slides, is at www.opennetsummit.org.

[14] For more information, see *Google's software-defined/OpenFlow backbone drives WAN links to 100% utilization* (June 7, 2012) at http://www.networkworld.com/news/2012/060712-google-openflow-vahdat-259965.html.

[15] See *Adoption of SDN: Progress Update* at http://opennetsummit.org/talks/ONS2012/elby-tue-sdn.pdf.

[16] Within this context, the term *demand* refers to a *potential* stream of data, whereas a *flow* would refer to an *actual* stream of data (i.e., once the demand has been admitted, it becomes a flow).

[17] See *Cloud Aware IP/MPLS Networking*, John Evans, Cisco (2012), MPLS World Congress, Paris, February 2012, http://www.uppersideconferences.com/mplsworld2012/mplsworld2012program_day_one.html.

[18] Cisco calls this demand engineering to clarify the parallels to traffic engineering; we call it demand placement because that better matches the functionality of the feature.

Considering projections that Internet traffic will increase four-fold in four years, and that video will be a major driver, selecting the best video cache will be an increasingly valuable use case.[19]

Infrastructure SDN System Diagram

Google did not provide a system diagram for their bandwidth calendaring system. So Figure 4 is an independent attempt here to shed light into the components of a system that would support bandwidth calendaring, demand placement, and Elby's SDN requirements.

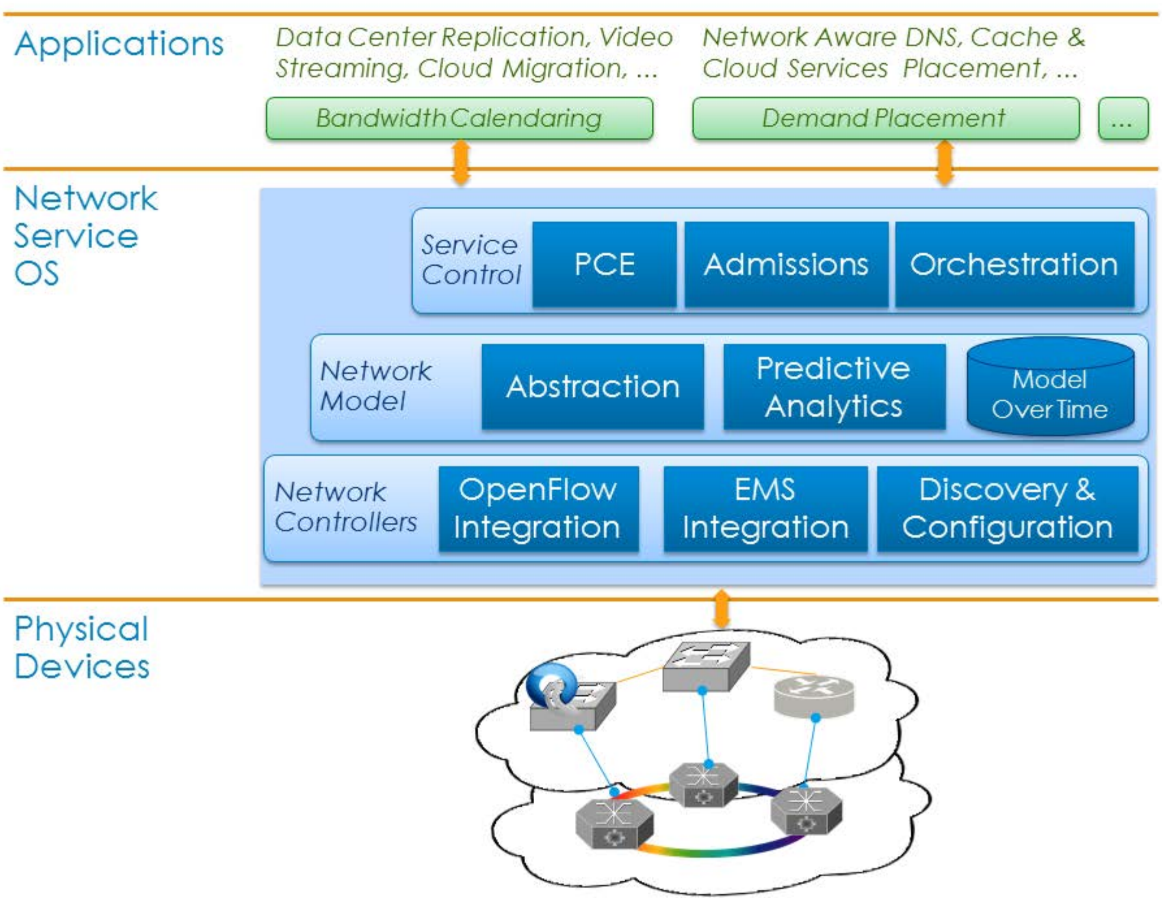

Figure 4. Network Services OS System Diagram

Network Service OS

The Network Service OS (NS-OS) introduced here is made up a Service Control layer, a Network Model layer, and a Network Controllers layer. The **Service Control** layer is the point of interaction for applications requesting network resources. The **Network Model** layer provides the facility for the Service Control layer modules to view an abstraction of the resources in terms of network graph nodes and links, not protocols. The model layer in turn relies on the **Network Controllers** layer that provides a global view of the network, not individual elements.

[19] See Cisco VNI press release from May 30, 2012: http://newsroom.cisco.com/press-release-content?type=webcontent&articleId=888280.

An example will help clarify the various elements. Imagine that an enterprise video conferencing system seeks to reserve network resources for a high-definition video broadcast. The conference scheduling software needs to communicate with the network to ask whether the conferencing demand can be met. This is the work of the **Admissions** module. Admissions may have to consult the **Path Computation Element (PCE)** to design paths that meet latency and resiliency requirements. At the time of establishing the video service, the **Orchestration** module ensures that all the elements required to establish and later disconnect the service (e.g., access controls and specific paths) are properly set up.

Abstracting the service request (bandwidth amount, time and place with latency and resiliency requirements) is sufficient for most user-level applications. However, a central idea behind SDN is to have multiple levels of abstraction appropriate for different kinds of programmability. In particular, Scott Shenker, one of the founders of SDN, talks about a Forwarding Interface, a Global Network View, an Abstract Network Model and a Control Program—which map closely to the layers in Figure 4.[20]

In our previous example, the PCE would be able to decide on a path that meets the video distribution requirements without having to concern itself whether those paths happen to be set up using multicast, point-to-multipoint RSVP or some OpenFlow mechanism.[21] Correspondingly, the **Predictive Analytics** module relieves the PCE and admissions from having to understand the details of predicting failure and future reservation states. Lastly, on the model layer, the **Model Over Time** module is crucial as a central authoritative source of resource reservations at any given time.[22]

Physical Layer: Mixed Technology and Legacy Infrastructure

The mixed technology of the multivendor packet and optical networks point to the need for being able to interface with legacy infrastructure through traditional SNMP/CLI interfaces and element management systems. The various modules in the Network Controllers layer serve to cover the variety of equipment deployed in real networks. This complete knowledge of the infrastructure is needed for Verizon and other established service providers to achieve their SDN goals; it is what Elby meant when he said we need to "incrementally introduce new architecture [and] interact with the large legacy network."[23] One cannot ignore the already deployed infrastructure, and that is what makes SDN a matter of *evolution*, rather than *revolution*.

In service provider networks, performing the discovery and control can be particularly difficult to achieve; it is not only the technical subtleties involved but also the degree of trust to even gather the information.

Requirements for an Infrastructure SDN Platform

Having seen these applications in action, it is worthwhile to consider the requirements of a platform that would support them.   Figure 5 provides a broad overview of this.

[20] See http://opennetsummit.org/talks/shenker-tue.pdf. See especially slides 25-27.

[21] PCEs and even user applications may under special circumstances access lower-level API and even the hardware directly. This is analogous to gaming software making use of specific GPU functionality. It is not the norm but remains possible.

[22] The Predictive Analytics and Model Over Time concepts are not present in the Shenker abstraction models but are easily motivated as part of real world application requirements such as those posed by bandwidth calendaring.

[23] See http://opennetsummit.org/talks/elby-wed.pdf.

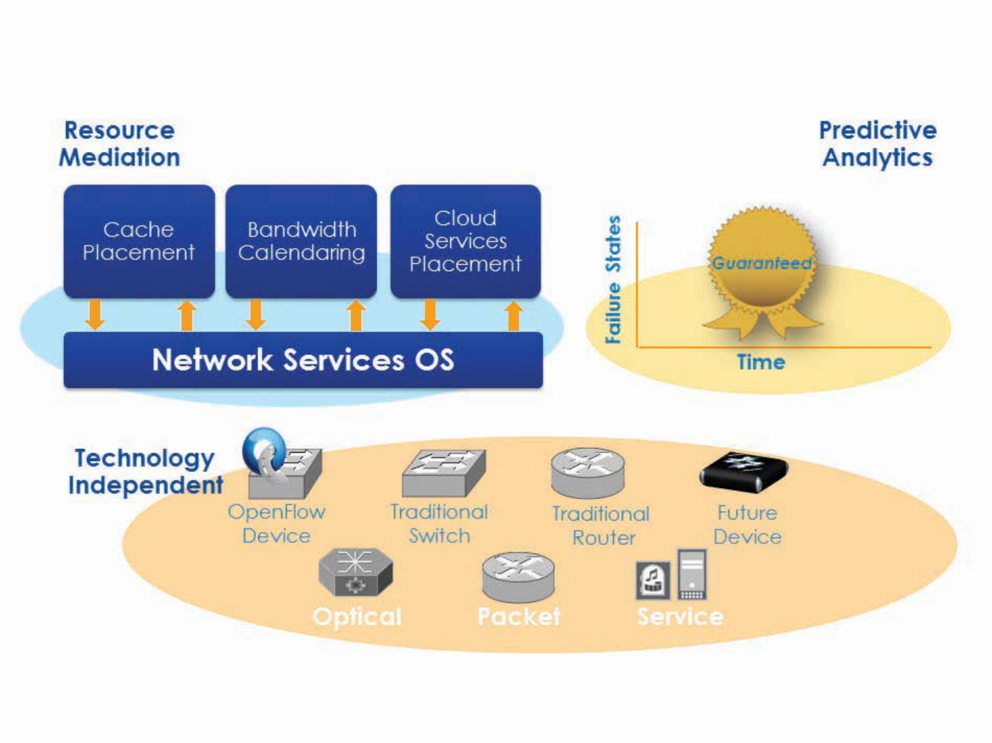

Figure 5. Platform Requirements for an Infrastructure SDN

Resource Mediation Between Applications

Many different applications make requests for network resources. For example, cache location and cloud service placement, as well as bandwidth calendaring, rely on the same network resources. These requests are satisfied by an NS-OS that provides a resource mediation service.

Resource mediation makes it easier for applications to communicate with each other—they can interact in a closed loop through a central routine instead of having to make direct connections (and thus incur the overhead of meshing). The centralized resource model is key to cross-application integrity: the NS-OS tracks not only the state of the network resources but also of all applications. The NS-OS needs to keep a repository of network resources and demands across time that can be queried and updated by all the applications trying to run on the network.

Predictive Analytics

Predictive analytics (determining behavior based on network events such as failures) is required if the service has resiliency requirements or if the solution allows operators to notice and act on trends. In both of these cases, the service admission, path computation and orchestration take place on a potential (failure state) or future state of the network. The NS-OS must provide this common abstraction to the various applications that require it.

Technology Independence

Technology independence is another key. The network might include OpenFlow-enabled switches, traditional switches and routers, and future technologies and network appliances. All of these will be side by side in the network and have to be recognized by the resource mediation layer so that the proper admission decision can be made.

The applications view of network resources goes beyond packets and reaches into optical infrastructure and servers. Sharing information across network layers with protocols can get complicated and applications can benefit from a central authoritative source of information about the whole network.

Role of OpenFlow

OpenFlow remains a "crucial" element of SDN (especially for Flow Services SDN), but as Scott Shenker and others have pointed out, it is just one piece of the overall SDN architecture.[24] The Controller abstraction provides a useful global network view, but user applications like video distribution or cache placement need the higher-level Service abstraction.

Achieving Infrastructure SDN with Cariden

At ONS 2012 (Figure 6), Stuart Elby addressed the issue of how the industry will get to a realization of the SDN vision.

verizon

SDN/OF Gap Analysis

SDN Requirement	Gap with Current Industry
Coordinated resource virtualization, aggregation, orchestration and optimization	Major gap is coordinated network and cloud computing orchestration
Abstractions and programming languages enabling higher-level service composition	Common form of abstract representation for applications, computing resources and networks will be needed.
Use of dynamic multilayer network capabilities and abstraction to achieve pooling, scaling, optimization, and simplification	Global optimization may only be possible with a logically centralized system.
Protocol specifications for major aspects of SDN ecosystem while still leaving room for differentiation and experimentation	OpenFlow only solves part of the lower-layer networking problem.
Operations functions to provision, monitor, diagnose and maintain services that are dynamically composed of network and compute resources.	No systems to enable stable, responsive and robust operation in a production environment.

Figure 6. Gap Analysis from Verizon's ONS 2012 Presentation

At first glance, the analysis can seem discouraging: there are major gaps in coordinating and unifying network and computing infrastructure, limitations to OpenFlow, and lack of production-quality systems. Cariden, however, has a strong foundation for realizing the Network Service OS, which is in trials with its technology-forward customers.

The elements of the NS-OS match strongly with Cariden software, which is deployed in a large number of challenging networks including nine of the eleven Tier-1 ISPs with the specific purpose of managing network infrastructure. For example, in the Controllers Layer, the Cariden MATE Collector and Deploy products form the basis of controllers for traditional switches and routers. Those existing elements are now matched with new development such as Cariden's OpenFlow integration to have a workable technology-independent Collectors layer. This approach in turn allows Cariden's customers to more immediately take advantage of SDN by rolling it in alongside existing infrastructure with minimum pain.

On the Network Model layer, the Cariden MATE resource model has been proven as an industry standard for modeling network resources. The MATE platform has traditionally been used with a human interface

[24] See, in particular, slide 27 at http://opennetsummit.org/talks/shenker-tue.pdf.

(a network planner or engineer as the source of control). More recently, however, Cariden has been exposing a programmable interface that is ready to work as part of a software-driven infrastructure. For example, the modeling API used at the SDN Workshop at Carrier Cloud Summit 2012[25] enabled an admissions module to respond at the rate of 400,000 requests per second while considering failure conditions (predictive analytics) as part of the admission request.

On the Service Control layer, Cariden's offline path computation software is a commercially robust module that has been production hardened over the last decade; an early study presented at NANOG in 2003[26] outlined the benefits. Deutsche Telecom also described detailed usage of its innovative offline path design system using MATE at a subsequent NANOG in 2005.[27] More recently, the London Internet Exchange (LINX) publicly announced the use of Cariden's MATE platform as a path computational element (PCE) for an architectural redesign in preparation for the London Olympics.[28]

Conclusion

The SDN movement has sparked exciting innovations in networking, portending a major shift in focus. The industry is moving steadily from its long array of new protocols to a disciplined investigation into finding the right level of abstraction for the greatest flexibility and agility—at sharply reduced complexity and cost.

Looking at SDN through the prism of value reveals three distinct areas of interest:

1. **Flow Services SDN** that is closely tied to microflow visibility and programming made possible by OpenFlow
2. **Virtualization SDN** that is closely tied to data centers and other multitenant compute-network environments
3. **Infrastructure SDN**—Cariden's focus—that is closely tied to providing visibility and control of network resources to software applications

Cariden enables Infrastructure SDN with its Network Service OS (NS-OS) built on top of its industry standard infrastructure management solutions. The NS-OS comprises a full battery of Network Controllers, an extended Network Model and real-time Service Control.

Cariden is currently engaged with its technology-leading customers in deploying Infrastructure SDN with an eye to making the NS-OS generally available in 2013.

[25] See http://www.cariden.com/images/uploads/DemoSheet.pdf.
[26] See *Metric Based Traffic Optimization* at http://www.cariden.com/packages/cariden/papers/arman-nanog.pdf.
[27] See *IGP Tuning in an MPLS Network* at http://www.cariden.com/packages/cariden/papers/nanog-t-com-horneffer.pdf.
[28] See the press release from June 18, 2012 at http://www.cariden.com/news/item/linx_uses_caridens_pce_functionality_in_preparation_for_summer_olympics.

About Cariden

Cariden Technologies, Inc. is a software company serving telecommunications providers worldwide. Founded in 2001, the company has maintained steady growth and profitability with its industry-standard network planning and analytics software. Networks serving over 85% of U.S. broadband customers have adopted Cariden software, as have most of the global Tier 1 ISPs. Cariden's technology leadership is embodied in the network services OS (NS-OS): the blueprint for Infrastructure SDN where software control is the centerpiece of networking infrastructure.

For further details, please contact us: info@cariden.com and http://www.cariden.com.

Cariden Technologies, Inc.
840 W. California Avenue, Suite 200
Sunnyvale, CA 94086
USA

Main: +1 650 564 9200
Fax: +1 650 564 9500
info@cariden.com