From Static Networks to Software-defined Networking

An Evolution in Process

Do you have something to say about this article?
Visit the *Journal* pages of the ISACA web site *(www.isaca.org/journal)*,find the article and click on the Comments link to share your thoughts.

The networking industry is gradually transforming itself from a hardware-centric approach to a software-defined platform. Although the concept of software-defined networking (SDN) is still considered new and acceptance of it is at a very nascent stage, the life cycle and evolution of the personal computer indicate the benefits of such an architectural model and suggest the unstoppable direction in which the networking industry will eventually go.

SDN is largely considered to be at the conceptual stage. The implementation of SDN is dependent on the network strategy adopted by enterprises. SDN refers to all of the protocols and technologies that work in synchrony to create a global view of the network and provide a centralized, intelligence-based network service, delivery and control.

The Open Networking Foundation (ONF) is the organization that leads the effort of the promotion and adoption of SDN. It does this through open standards development. ONF mentions SDN as an emerging network architecture in which network control is made directly programmable and is decoupled from the forwarding plane.[1] This migration of control, from tightly bound in individual network devices to accessible computing devices, enables the underlying infrastructure to be separated for applications and network services, which allows administrators to manipulate networkwide traffic flow to meet the changing needs of today's business-driven networks.

Nikesh Dubey, CISA, CISM, CRISC, CCISO, CISSP
Is a cybersecurity specialist and governance, risk management and compliance (GRC) expert. He has a wide range of consulting experience in the fields of IS audit, information security and GRC. Working on different continents has given him an opportunity to look closely at the core issues, drivers, expectations and challenges of various enterprises. His previous *ISACA® Journal* article, "Corporate Responsibility—Retaining Top Management Commitment," discussed an innovative way to retain and improve management commitment levels, which is essential for the success of any program. He is currently associated with AGC Networks and can be reached at *nikesh.dubey@agcnetworks.com* or *nikesh.dubey@gmail.com.*

The Challenge

The advent of new technologies, e.g., mobile devices, server and content virtualization, and cloud services, are among the key forces driving the networking industry today. These new technologies have forced the networking industry to take a fresh look at the traditional network architectures currently in use. Many typical networks are hierarchical in nature, built with layers of ethernet switches arranged in a tree-like structure. The key characteristic for traditional networks is that each device has a local control plane and a local data plane. Each device also has its own management planes, e.g., connecting to the device through Telnet, a simple, early network protocol that allows users on one computer to log into another computer that is on the same network.

The process of establishing the network topology using a control plane that runs locally is complex. This complexity results from no single device knowing the entire network. To manage each device, each device must be connected to its data plane individually to make configuration changes or updates, which is not an intelligent approach. The control plane is where the forwarding and routing decisions are made, while the data plane is where the commands of the control plane are executed. This traditional design did meet the needs of a time when client-server computing was dominant. However, such a basic architecture is not well equipped to meet the dynamic computing and storage needs of today's enterprise data centers and evolving technical landscapes due to changing business needs. Drawbacks of traditional networks include their static nature in contrast to the dynamic nature of today's server requirements. The complexities of today's networks make it difficult for IT to apply a consistent set of access. Hence, the traditional policies leave organizations vulnerable to security breaches and regulatory or noncompliance issues. Furthermore, networks must also grow to meet the needs of hundreds or thousands of newly added devices with different performance and service needs. The inability to scale up to meet these demands is a major limitation of traditional static networks. It is also understood that the lack of a standard in this area and

open interfaces often limit the capability of network operators to customize the network to their specific individual environments because they are hindered by the vendors' control of the equipment.

The Genesis of SDN

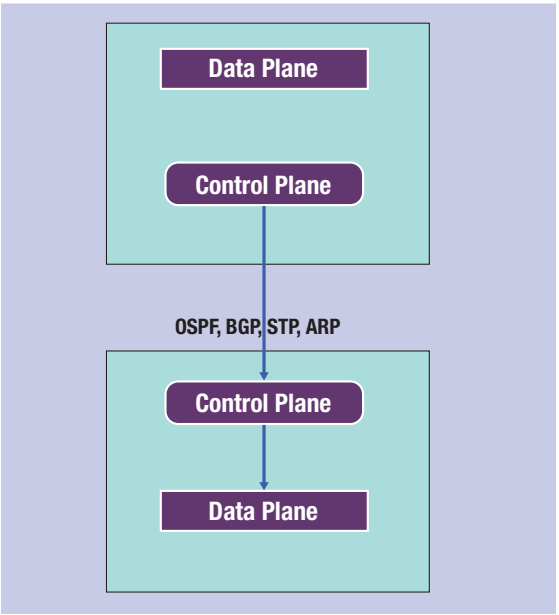
These disconnects between the increasing network industry requirements to support business and the existing static nature of traditional network capabilities have given birth to the concept of SDN. The basis of SDN is the concept of virtualization, which, in its most simplistic form, allows software to run separately from the underlying hardware. Virtualization has made cloud computing a reality today. There are several benefits of virtualization.

Virtualization allows data centers to quickly and dynamically provision IT resources exactly where they are needed. However, to keep up with the speed and complexity of split-second processing, there is a need for the network to also adapt, becoming more flexible and automatically responsive. The idea of virtualization can be applied to the network as well, separating the function of traffic control from the underlying network hardware plane into a centralized network-based intelligence control entity resulting in SDN. Thus, SDN is the natural next step in the evolutionary process of network architecture used today. The networking industry will gradually see a major shift in paradigm from a static, hardware-centric model to an evolving, software-defined model.

A New Approach to Building Networks

Most networks deployed in today's environments require a great deal of manual administration. This is because traditional networks had the device-driven control plane interacting with the device-driven data plane (see **figure 1**), using protocols such as Open Shortest Path First (OSPF), Border Gateway Protocol (BGP), Address Resolution Protocol (ARP) and Spanning Tree Protocol (STP), and this was a limitation both from a technical and management perspective. The limitation arises because to configure and manage such traditional networks, the administrator needs to log into every device for intervention and manage the out-of-box capabilities driven by hardware appliances, which require configuration changes, making it tedious and resource intensive.

Figure 1—Legacy/Traditional Networks

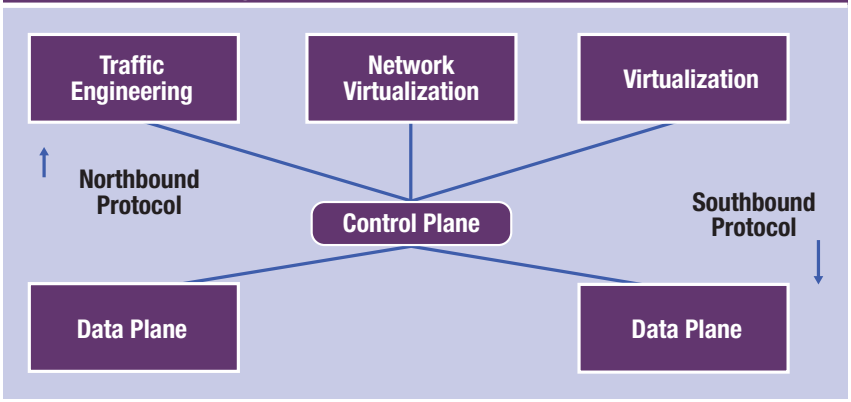


Source: Nikesh Dubey. Reprinted with permission.

However, the growing number of technologies using virtualization, cloud and mobility create more challenging and demanding environments; networks must appropriately support and adapt to these environments and manage their demanding requests in real time. SDN does this by introducing an abstraction layer that logically separates the control and data planes, centralizing the network intelligence layer. It also abstracts the underlying network infrastructure from applications with the objective of dynamically responding to changing network demands using controllable packet/flow processing protocols. This helps the SDN architecture provide networks with the advantages of visualization, traffic engineering and network virtualization.

There are several approaches to implementing SDN, but this article focuses on the most common components and concepts.

Figure 2—Basic SDN Architecture



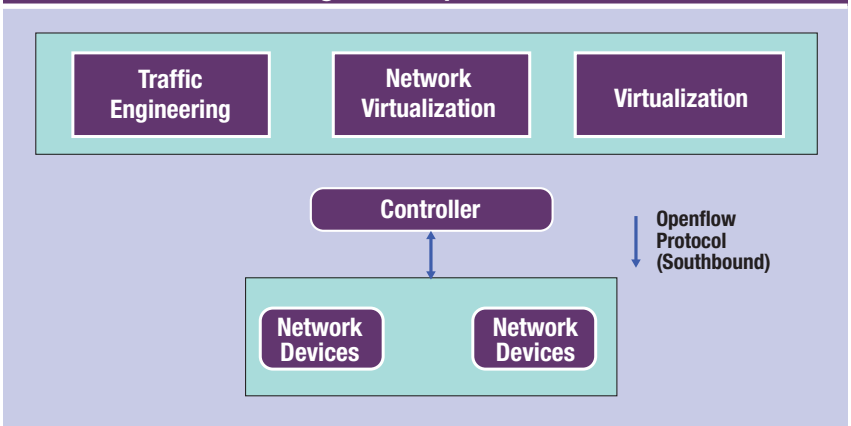
Source: Nikesh Dubey. Reprinted with permission.

Basic SDN Architecture

At a basic level, SDN architecture consists of three layers: the application layer; the control layer or SDN controller; and the data, physical or infrastructural layer (**figure 2**). At the top is the application layer, which includes applications that deliver services, e.g., switch/network virtualization, firewalls and flow balancers. These are abstracted from the bottom layer, which is the underlying data or physical network layer.

In the middle is the control layer or SDN controller, the most important aspect of the SDN architecture. This layer removes the control plane from the physical plane and runs it as software while being integrated with the physical and virtual devices on the network, facilitating optimal network service management.

Figure 3—Open SDN



Source: Nikesh Dubey. Reprinted with permission.

Open SDN

In Open SDN, the goal is to separate the control layer and data layer, creating a common language for programing network switches. The most common example of open SDN is OpenFlow, created by the ONF. SDN actually started with OpenFlow, which is a vendor-neutral communications interface defined in between the control and forwarding planes. OpenFlow internally provides an application program interface (API) or open interface to networking devices. It does not matter which operating system or vendor the networking device is using. With OpenFlow, there is an open interface to managing the device.

Typically, open-source tools are always a risk as they could be vulnerable. Lack of secure coding practices by novice and enthusiastic developers may allow vulnerabilities to creep into their code that may be exploited in the future. Organizations are weary of security issues when it comes to open-source tools. Opening the software's programmable interface to anyone who wants to come in and code makes the code vulnerable, devoid of quality coding practices and open to manipulations in the future. OpenFlow protocol is considered limited with insufficient functionality and scaling problems. **Figure 3** is the architecture of Open SDN.

SDN Using APIs

APIs are an alternate way to provide the abstraction necessary for SDN and provide a highly programmable infrastructure. Programmable APIs provide a channel by which instructions can be sent to a device to program it. Programmers can read API documentation to understand the device and code the appropriate commands into their applications. As SDN has evolved, APIs are considered northbound or southbound, depending on the location where they function in the architecture (**figure 4**). APIs that reside on a controller and are used by applications to send instructions to the controller are called northbound because the communication takes place north of the controller. Examples of northbound APIs are RESTful and Java APIs.[2] These APIs allow the developer to manipulate flow tables and flow entries on networking devices (e.g., routers and switches) without talking to them

directly. The application developer is abstracted from the hardware and does not need to know the details and specific requirements of the switches, routers and other network devices.

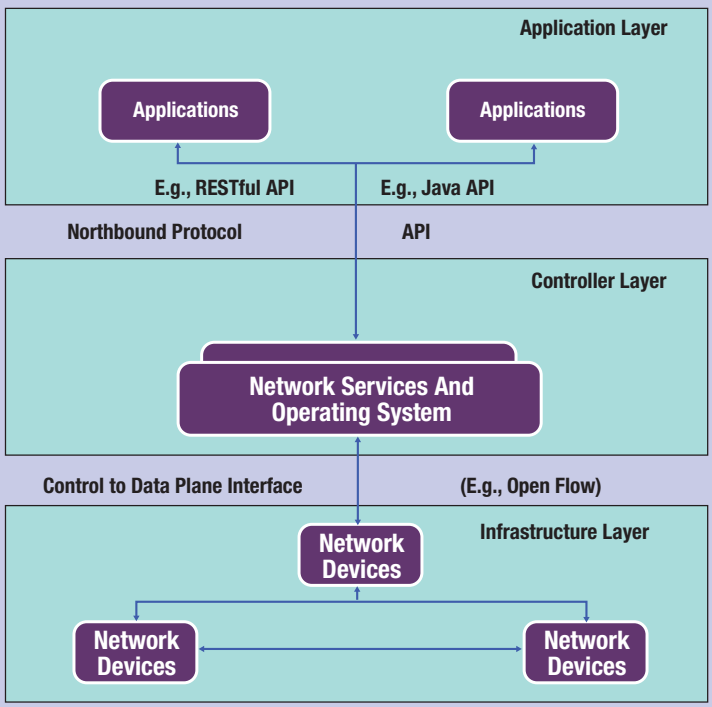
Southbound APIs reside on network devices, such as switches. These are used by the SDN controller to provision the network, with the communication taking place south of the controller. OpenFlow is a prominent southbound protocol. Another example of a southbound protocol is the Network Configuration Protocol (NETCONF).

SDN Using Overlay

The advent of virtualization allowed for the possibility of the network overlay architectures to be created. Overlay networks run as separate virtual networks on top of the physical network infrastructure. When the concept of SDN was envisioned, the platform for leveraging the network overlay architecture already existed.

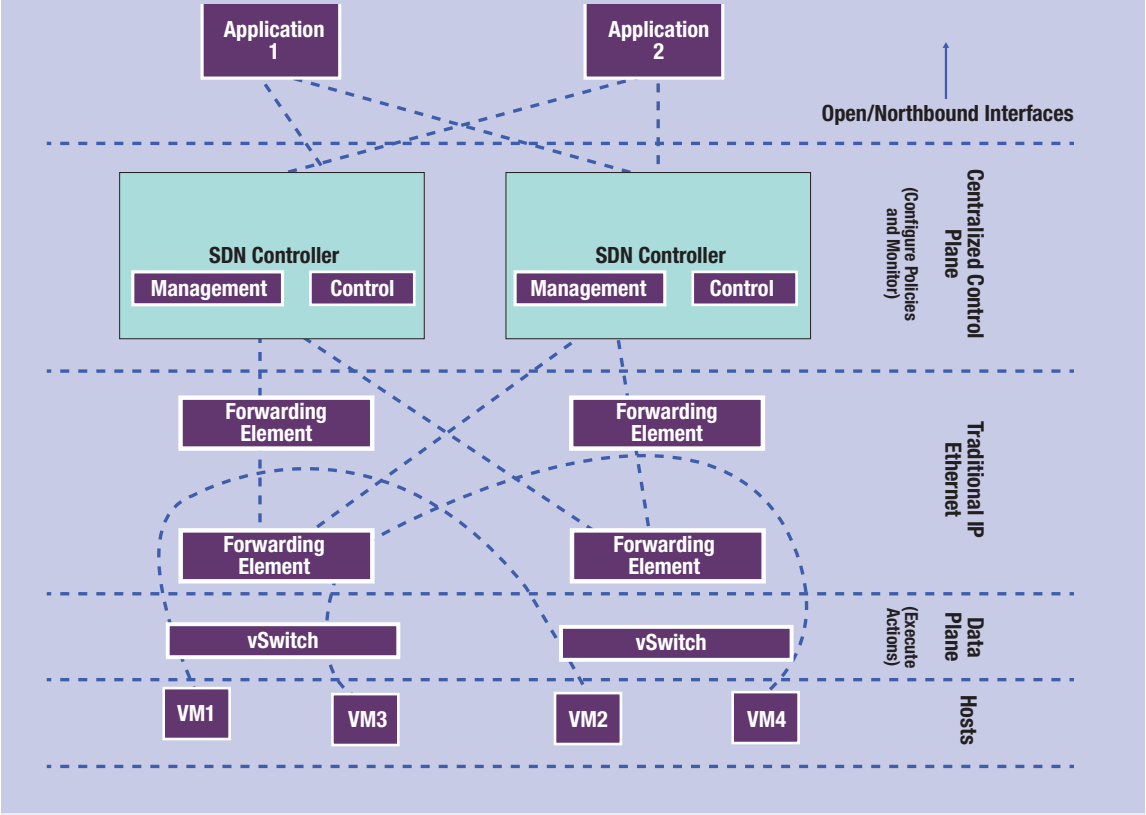
In SDN, using overlay nodes in the overlay network can be thought to be connected by virtual or logical links, each of which represents a path of its own so that there is an overlay of the virtual network and the existing physical one. This is the most popular model as it supports agility, which is key to networking solutions. In SDN overlay, the overlay implementation is built over the existing architecture to leverage a physical network that already exists. This suits organizations as they do not have to do anything other than add the new network over the existing one. The overlay is created using virtual switches inside hypervisors. A hypervisor or virtual machine monitor (VMM) is a piece of computer software, firmware or hardware that creates and runs virtual machines. A host machine is a computer on which a hypervisor is running one or more virtual machines. Each virtual machine is called a guest machine. The controller communicates with the hypervisor's virtual switches. These set up tunnels that make use of the underlying physical network, but do not need to actually configure the hardware

Figure 4—API-based SDN



Source: Nikesh Dubey. Reprinted with permission.

Figure 5—Network Overlay SDN



Source: Nikesh Dubey. Reprinted with permission.

to send traffic to its destination. If agility is the key objective for the proposed network architecture, then overlay is a good choice to implement.

Virtualization technologies, e.g., Generic Network Virtualization Encapsulation (Geneve), Virtual Extensible LAN (VXLAN), Stateless Transport Tunneling (STT) and Network Virtualization Using Generic Routing Encapsulation (NVGRE), provide this solution by using network encapsulation. Big Switch Networks' Big Virtual Switch offers SDN overlay application using OpenFlow. **Figure 5** depicts a network overlay SDN architecture.[3]

Advantages of SDN

There are numerous advantages of SDN. SDN increases network flexibility through holistic management of the network and enables rapid innovation. But why should organizations consider SDN, especially if it is still in the development stage and has not been widely adapted? The SDN model has the potential to make significant improvements to service request response times, security, reliability and scalability. It could also reduce costs by automating many processes that are currently done manually, which are resource intensive, slow and costly due to the use of restrictive commodity hardware. SDN offers a more efficient and flexible network that increases the speed of service delivery. It delivers cost savings on hardware and also offers the ability to test new protocols in hindsight.

SDN Limitations and Challenges

Before looking at the limitations of SDN, it is important to understand the principal concept that drives SDN—virtualization. Virtualization adds overhead and network latency, which is an issue

for any operations that require fast response times from time-sensitive systems (e.g., financial systems or stock-related applications). It is also important to note that networking is static and not getting faster. Moreover, dependency on the Internet to do business is expanding traffic by a huge percentage, hence the demand to maintain or reduce existing response times would be a considerable challenge.[4] The need for faster speeds and the fundamental limitations of visualization, such as overhead and latency, may place limits on what SDN can practically achieve.

The adaptation of SDN will also be slow. This is because networks are considered the backbone of any infrastructure, and changing it is not easy. Unlike the adaptation of virtualization, which was more of an end-user change, SDN requires fundamental detailed planning as it impacts everything being serviced on the network. The centralized SDN controller also makes it vulnerable to become a single point of attack and failure.

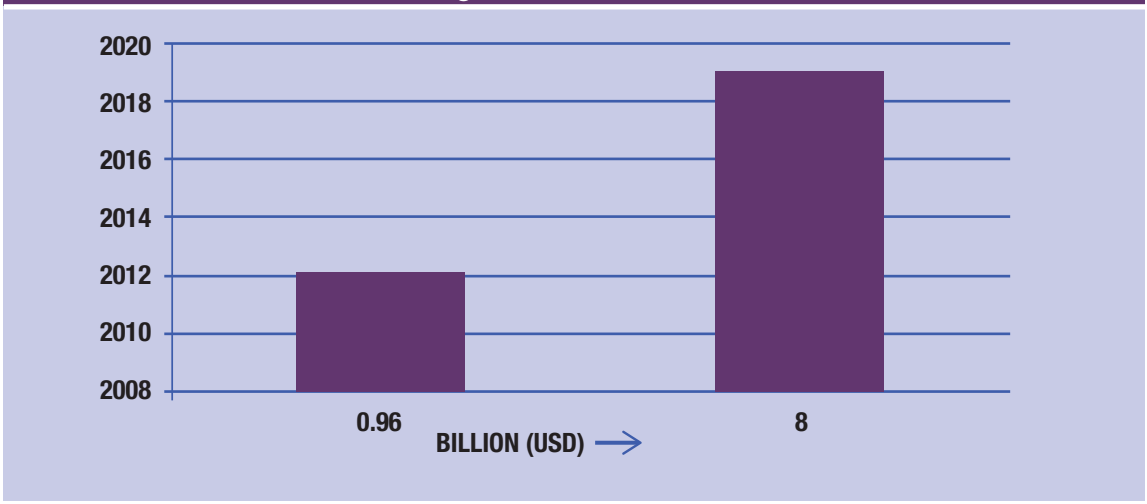
Will SDN Really Catch On?

Although SDN promises to deliver benefits for the networking industry, the big questions are if anyone is using the concept productively and whether it will be the future direction of the network industry. There is an estimated rise in the SDN market worldwide from US $1 billion in 2014 to US $8 billion in 2018 (**figure 6**).[5] The SDN market includes network infrastructure, network virtualization, professional services, and network services and applications.

Conclusion

Computers have evolved from a hardware-driven architecture to a software-defined module. In the 1970s and 1980s, the IT industry was primarily driven by hardware-centric devices that were limited in speed, size and network latency. The advancement in technology and its evolutionary process eventually guided it to a software-centric architecture, dramatically increasing speed and reducing size and cost, resulting in higher efficiency. The networking industry is undergoing the same transformations. The foundation of SDN is the concept of virtualization that has benefitted the IT industry in various ways. In principal, SDN promises to deliver a network that is enabled with network technology innovation and versatility while reducing complexity and administrative overhead and cost. It is important to identify the key pain points, drivers and use cases that SDN could address in an organization. If agility is the main priority, then organizations should deploy an SDN overlay solution. However, if there is a need to foster support for innovation in all three planes, then an

Figure 6—SDN Growth



Source: Nikesh Dubey. Reprinted with permission.

OpenFlow-based architecture takes precedence. If the focus is on programming APIs to better meet the specific needs of an organization through their applications, an API-based SDN is suitable. In general, SDN offers agility by allowing external control and automation of the network, making it directly programmable. It offers management benefits by improving operational efficiencies by making network intelligence centralized in software-based controllers that maintain a full view of the network. Besides lowering the capital and operational costs, it is also important to note that SDN represents an entirely new way to manage network connectivity—one that is defined not by the vendors and equipment makers, but by those who use the network for their own business needs. SDN is intelligent and flexible enough to prioritize traffic; direct network resources to where they are needed most; and adapt, change and evolve over time to meet the business needs of today and address the challenges of the future.

Endnotes

1 Open Networking Foundation, *Software-Defined Networking: The New Norm for Networks*, 13 April 2012, *https://www.opennetworking.org/images/stories/downloads/sdn-resources/white-papers/wp-sdn-newnorm.pdf*

2 Bombal, David; "SDN and OpenFlow Overview—Open, API and Overlay based SDN," YouTube video, 28 October 2014, *https://www.youtube.com/watch?v=I-DcbQhFAQs*

3 Marschke, D.; "Is SDN Read for Prime Time or Junk Time?," APAC CIO Outlook, *www.apacciooutlook.com/ciospeaks/is-sdn-read-for-prime-time-or-junk-time-nwid-658.html*

4 O'Reilly, J.; "SDN Limitations," *Information Week*, 17 October 2014, *www.networkcomputing.com/networking/sdn-limitations/241820465*

5 Statica, "Software-defined Networking Market Size Worldwide in 2014 and 2018 (in Billion U.S. Dollars)," *www.statista.com/statistics/468636/global-sdn-market-size/*