Demystifying SDN for the Network Engineer

Out on the IT horizon, there's a change gathering force that is difficult for network engineers to ignore. Software-defined networking (SDN) promises to be a big change, and those engineers surveying the waters recognize that it is no simple blip on the radar.

Some network engineers have already sailed out to meet it; others are contemplating it back on shore and wondering at what point it will enter their day-to-day lives. Many are trying to figure out what SDN is about and exactly how it stands to impact their work as network engineers.

This white paper aims to demystify SDN for the network engineer and, in so doing, help chart the course for the journey ahead. The white paper will seek to accomplish the following:

- Explain the technological forces that have given rise to SDN
- Provide an easy-to-understand look at a new network architecture made possible with SDN, and show how the new model compares with a traditional network architecture
- Discuss why SDN is such a monumental change and enumerate the benefits it can bring to the network and the network engineer job role
- Share a couple examples to illustrate how SDN can dramatically ease the daily function of the network engineer
- Offer guidance for understanding SDN and taking your first steps

Change Is in Your Blood

First, in approaching SDN, realize that you as a network engineer are no stranger to change. It has always been a part of your growth with the network. If you've been in the field for 20 years or more, then you made the big shift from circuit switching used in telephone technology to packet switching used in IP.

And, if you've been practicing for 10 or more years, you've forged ahead with the progression from ATM and Frame Relay to Ethernet as the preferred WAN technology and all the network speed benefits that have come with it. You've also moved from wired access to wireless access, and then to converged wired and wireless networks. In addition, you've been part of the adoption of Voice over IP and all types of video, requiring a new focus on quality of service and how to support real-time applications.

Finally, look back over just the past decade and realize that you as network engineers have responded successfully to widespread adoption of mobile devices and BYOD policies by figuring out the security and VPN technologies needed to connect the network to many different end devices. You've dealt with IPv4 address exhaustion due to mobile device proliferation, and you've begun to take responsibility for managing Internet of Things (IoT) devices, exploring deeper analytics, and implementing virtualized services such as cloud.

To be a network engineer, therefore, is to have welcomed and embraced change repeatedly over the years. The SDN change is not minor, but as this white paper seeks to demonstrate, it is a shift that offers a great deal of promise for those with continued adaptability and receptivity to learning in their careers.

Why Is This Happening Now?

SDN, also known as network programmability or network automation, offers us a way to make the network nimbler by introducing a greater level of automation. Before we discuss more precisely how that gets accomplished, let's take a look at three key catalysts in the industry driving the need for network change:

- Dependence on **mobility** for digital business
- Virtualization and the flexibility of **cloud** services
- Scaling requirements for the **Internet of Things (IoT)**

Mobility is not a brand new phenomenon, of course, but mobile devices have proliferated in this age of "bring your own device" (BYOD). BYOD has become the expectation for connecting quickly, seamlessly, and reliably within an organization. Mobility is the opposite of static, and therefore network policies must be dynamic and flexible to accommodate mobile devices and allow users to operate free of artificial constraints.

As for cloud, it wasn't all that long ago that cloud services didn't exist. Every organization had its own data center, and no one thought in terms of shifting important business applications to the cloud for use only when needed. Nowadays, organizations have decisions to make about public, private, and hybrid cloud options, and along with this has come a need for agility in network policy enforcement and decision making. The network also requires further reach with cloud so that the control points of an enterprise network can extend right to where the applications and storage of information is taking place.

And then there has been the rise of IoT with its explosion of new sensors and devices connected to the network. Cisco predicts that by 2030, the number of devices connected to the Internet will be 500 billion. And all those networked devices will require talent to manage.

In a bygone time, if an IT networking team had to, let's say, double the number of network endpoints it supported, it would scale the number of network engineers on staff accordingly. But with an exponential rise of IoT endpoints, simply increasing staff to keep up is no longer realistic. There needs to be a more efficient way to scale the network to meet modern demands.

Mobility, cloud, and IoT are three very real ways in which business is being digitally transformed. And

the network—the nerve center of it all—has not been untouched. While you have no doubt been hearing about "digital business transformation" a lot these days, they are not just words. Digital transformation has spurred the need for a digitally ready network. And the way to get there—the way to scale effectively—is to implement a network that is automated and software-driven.

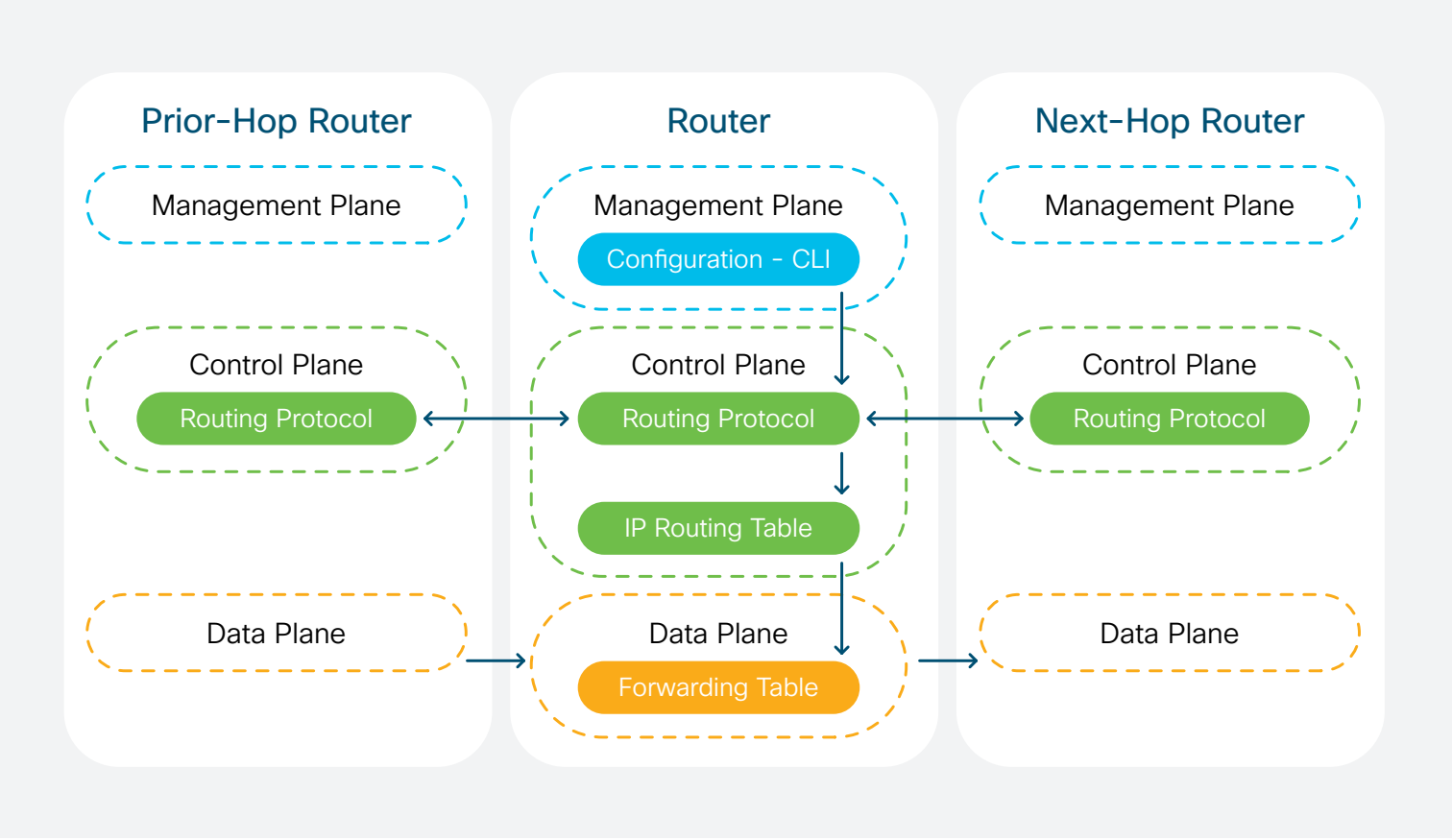
When Connectivity Was the Be All and End All

Traditionally, the network has been manual, complex, and static. This was fine for decades, where the primary goal of the network engineer was this above all else: maintain connectivity! Keep the network up and running, and you were doing your job well.

To meet this goal, network engineers have mostly worked at the command-line interface, or CLI, manually configuring a network infrastructure of dedicated hardware devices one by one. Different platforms and products in the network require memorizing different sets of commands or looking them up. This scenario represents the case for the vast installed base of routers, switches, and other devices used in today's networks.

Traditional Network Device Architecture

Network engineers are, of course, familiar with the architecture of a traditional network device, such as a router, which has three operational planes, as shown in the image below: the data plane, control plane, and management plane. The data plane is responsible for the forwarding of packets, the control plane uses routing protocols to decide where packets should be sent, and the management plane provides the ability to configure the router using policies that are implemented by the network administrator. If the data plane is the hands of the network, the control plane and management plane are the brains.



The Programmable Network

Going forward, networks need to be more sophisticated in their incorporation of today's new services and technologies, and agile and flexible in their ability to turn these services off and on as needed. The way to do this, increasingly, has been to move away from manual, device-by-device configuration to software-driven, programmatic interfaces that enable automation.

It's not a brand new concept. There has been some sort of programmatic, software aspect to networking for a while. For many years, the data center has benefited from SDN technologies. Now these same benefits are being applied to all segments of enterprise and service provider networks.

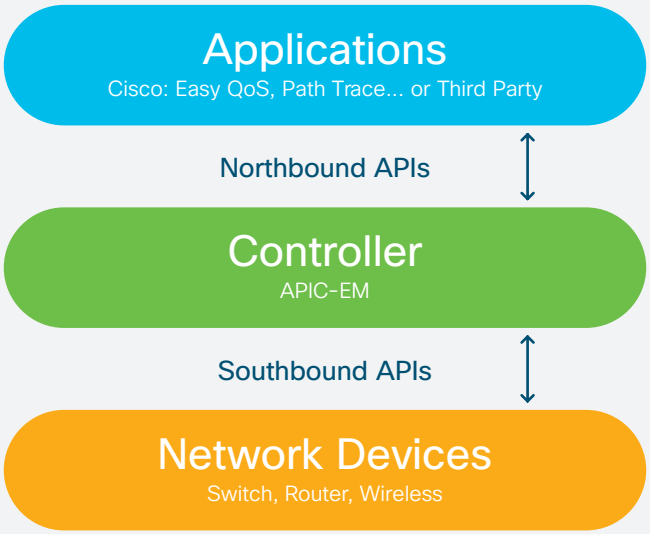
Cisco recognizes that the transition toward network automation must allow for legacy networks to migrate over time. In Cisco's model of SDN, as a first step, the management plane has been greatly enhanced and centralized within a software-based network controller developed by Cisco for enterprise networks. This is called the Application Policy Infrastructure Controller – Enterprise Module, or APIC-EM.

Leave It to the Controller

The APIC-EM controller sits in the center of the programmable network architecture. The controller serves as the network information database and control point. Even before learning programming skills, the network engineer can immediately take advantage of the application programming interfaces, or APIs, that allow the controller to implement policies in the southbound direction to the network devices.

Instead of using the typical CLI to accomplish this task, the network engineer now uses a GUI provided by the controller. This is where the automation magic happens, because instead of requiring the same configuration to be repeated manually on each device, the controller is able to apply the desired policies across all devices. The controller also provides APIs in the northbound direction in order to interact with network applications and business applications that help to manage, secure, and extract useful data from the network.

The image below shows Cisco's network automation model with APIC-EM at its core. Data plane and control plane functionality continue to exist largely at the device level. The controller takes over much of the activity of the management plane.



In this new world, the network engineer is no longer interacting through the CLI and monitoring activity and troubleshooting on a terminal. Instead, utilizing the new programmatic interfaces and prebuilt network applications, the controller-based architecture is able to greatly automate and streamline these tasks. The interaction is no longer human-to-machine but software application-to-network device.

You can now rely on the controller to determine the platform-specific configuration commands and syntax information to send to each network device. You use an intuitive GUI to tell the controller what business policy you want to implement, and the controller is able to convert the intended policy via APIs and translate these into the myriad rules and applicable configuration for the different devices in your network.

You are freed from much of the need to memorize command syntax because you can simply provide the controller with a policy and let the controller manage how it is going to interact with each device so that they can be configured, modified, or set up to achieve specific business goals.

The APIC-EM controller has a network information database that allows you to scan the network and obtain an inventory of all network devices. With this improved visibility into the network, APIC-EM then automatically configures every device that has been discovered as part of the inventory.

Big Change, Big Payoff

In the changes that network engineers have greeted over the years, SDN is monumental in that the automation of network processes through software allows the necessary agility, flexibility, and scalability in the network that mobility, cloud, IoT, and other aspects of digital business transformation demand. During the coming years, organizations in every industry will be unable to claim digital readiness unless they have acquired some proficiency with network automation.

For organizations and network engineering teams willing and able to evolve to a controller-based architecture, the benefits will be substantial. Here are some of the specific rewards to be reaped:

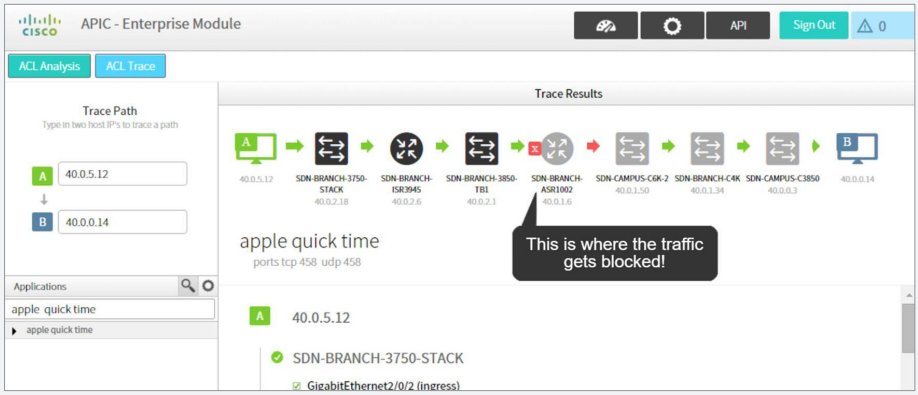
- **Greater speed and thus faster rollout of services:** If you've got a network change to make, you can now take advantage of a network application that automates the process. Traditionally, if you have to make the change to hundreds of routers, you need to do it manually at the CLI, one router at a time over the course of hours or days. With automation, your network application can configure all routers in a matter of minutes.
- **Accuracy and reliability:** If you use a validated network application, then you know it is always going to be done consistently. Conversely, if you have to do something manually across a hundred devices, chances are good you're going to make an error on at least one of them.
- **Simplicity:** The controller uses a process called abstraction to translate complex rules and policies behind the scenes to make things transparent to you the network engineer.
- **Ability to optimize the network:** With programmability, you can get the network to respond much more fluidly to constantly changing conditions. You can optimize the use of resources and adjust to changes automatically, which further increases network efficiency and speed. You can scale network services up or down per business demands.
- **Better analytics:** A truly digital network allows you to get deeper data and faster insights, which allows you to be more agile and improve security visibility by learning from and adapting to changes and needs in the network.
- **Greater OpEx value:** This white paper spoke earlier about how IT teams will continue to confront staffing demands in the face of IoT's massive scale. Automation can relieve the situation by enabling network engineers to accomplish more with the same effort and time expenditure. This benefit is particularly motivating when one considers that operating expenses (OpEx) can account for twice as much as capital expenditures (CapEx) in enterprise network deployments.

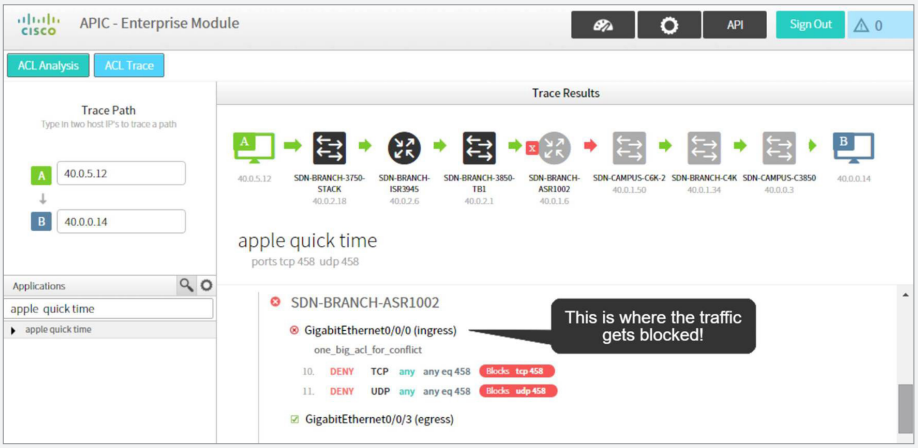
Aha, I See Why This Matters!

Let's look at a couple compelling examples of how SDN can ease the daily work of a network engineer. Working with access control lists, or ACLs, requires exact syntax and sequencing to achieve the required security controls. Configuring and troubleshooting ACLs can be a complicated and cumbersome task for network engineers. These rules about how traffic can flow through the network and who can have access can get unwieldy as the list of permissions and constraints confronting the user at the CLI builds up over time. Eventually, the ACLs can get so long that they begin to develop inherent contradictions, as earlier permissions and constraints get forgotten with the addition of new ones. ACLs can sometimes contain hundreds of entries, and sorting through them can feel like trying to follow a very complex logic diagram.

One big advantage of the Cisco APIC-EM controller is that it vastly simplifies and accelerates ACL management with an application called the Path Trace ACL analysis tool. The Path Trace ACL application gathers ACL information from each network device and shows the hop-by-hop path taken (or blocked) by traffic between endpoints. Path Trace ACL is a particular boon to the engineer, because the APIC-EM controller automatically determines all of the permissions and constraints automatically. Rather than confronting you with a tangle of logic at the CLI, a visually friendly GUI greets the engineer at the computer, revealing in short order, by means of easy-to-interpret graphics, exactly which device is blocking traffic.

The two screen shots below show the Path Trace ACL GUI within the APIC-EM controller, and its ability to pinpoint in a highly visual, straightforward manner where network traffic is getting blocked.





Hard QoS vs. Easy QoS

The simplicity and user-friendliness of SDN also becomes dramatically evident when one looks at the Easy QoS feature that comes with the APIC-EM controller. Typically, network engineers configure QoS in the network on each device. The ability to manage and prioritize network traffic and real-time applications requires QoS to be configured across all network devices in a path. If there are 10 hops in the path between the sender and the recipient, each one of those devices must be configured so that it can operate according to which packets merit the higher priority.

The network engineer might spend 10 or more minutes configuring each hop. And then, if a new business-critical application needing priority appears on the scene the next day, the network engineer must reconfigure the entire network so that this new addition to network traffic gets high-priority treatment.

Because it can be so burdensome to implement, many users do not fully utilize QoS. Instead, they accept the expense of overprovisioning the network to avoid congestion. This is a costly and inefficient way to operate networks.

But now, with the burgeoning of IoT and its myriad devices, you can't, for example, have a smart vending machine getting the same priority as a quarterly sales reporting database—there's just too much dependency on timely access to information. Gone are the days when you can simply provide a "best effort" traffic management to all users and applications.

Easy QoS solves this problem. With this feature, the network engineer can simply take advantage of the prebuilt capability within the APIC-EM controller and, using the GUI, select the person, group, device, or application and assign the appropriate priority. Once the engineer has assigned that business priority, the controller communicates that information via the southbound APIs to each device, and implements the policy by automatically configuring each device. What you used to have to do hop by hop, you can now do at one central location in the controller.

Click here for a video that illustrates the features of Easy QoS. The video demonstrates how a user can set a QoS policy via GUI and then let the controller translate that request, within minutes, into an end-to-end, validated, device-level configuration based on best practices. The video also shows how you can implement real-time prioritization of application flows.

What About My Needs?

While the organizational benefits of SDN are abundant, they also hit home on a personal level for the network engineer. For one, there is no denying that existence can sometimes get tedious at the CLI. "Did I put that command in?" "Did I add the right suffix to make it behave differently?" Network automation can remove much of the mundane and robotic from daily life.

In addition, release from some of the configuration tedium will free up avenues for applying more creativity on the job when you are able to programmatically control the network. This responsiveness greatly elevates the value of IT to the enterprise and can extend your visibility into other parts of the business, and kindle the fire of new ways that you can contribute to digital business success.

Lastly, as you get ahead of the curve trying to understand the benefits of a controller-based architecture and learning about some of the network's automation capabilities, you are potentially improving the efficiency of the organization and reducing your employer's costs. Your increased efficiency and expanded skills will make you more valuable to your employer at all levels of engineering responsibility.

Charged Up for the Change

We hope that our efforts in this white paper to demystify SDN also serve to reduce some of the anxiety that can come with change. But, we realize that network automation requires a pivot at the architectural level and so is not a small change. A "Cisco Certified Community Research Survey on Network Automation and Programmability," conducted in 2016, revealed that 58 percent of Cisco certified individuals polled

Demystifying SDN for the Network Engineer

work entirely at the CLI on a day-to-day basis. Thus, we know that many network engineers are just now investigating the benefits of automation and SDN for the first time.

We'll be addressing this specific aspect more thoroughly in a later white paper, but don't be afraid of SDN. SDN provides new tools to make your job easier and more efficient. The need for your hard-earned foundational networking skills and knowledge is not going away. Network automation doesn't mean an immediate shift to everyone becoming a programmer and tossing out the underlying networking skills and knowledge. The knowledge you currently have will remain critical to running the network.

A future white paper will introduce you to exciting possibilities inherent in being able to harness prebuilt network applications, virtualize network functions, or start to develop basic programming skills to customize applications to meet your specific network needs.

As we indicated at the beginning of this white paper, change is something you have always needed to tackle as a network engineer. Programmable networks allow you to apply what you already know and build upon that with new capabilities. If you're familiar with a wireless LAN controller, then you've already used a controller to manage network devices, in this case, wireless access points. SDN is a continuing development in technology evolution–like many changes that are always happening within the network, the IT nervous system.

The Network Needs You Now More Than Ever

As SDN picks up steam out on the horizon, it is important for network engineers not to retreat into silos of the past. SDN is hitting some industries sooner than others, but for many markets, its arrival might be only two or three years away, perhaps less. A 2016 IDC "Digital Network Readiness Survey" of more than 2000 large and midsize organizations worldwide revealed that 45 percent had a plan to achieve advanced readiness–involving network automation–within two years.

In a 2016 end-of-year forecast, Gartner's research vice president Andrew Lerner predicted "the death of the CLI," indicating that by 2020, only 30 percent of network operations teams will be relying on the CLI as their primary interface, down from 85 percent at the end of 2016.

Because of the onrush of mobile, cloud, and IoT, there are very few industries that are going to be able to isolate themselves from the forces of digitalization, least of all the network that underpins them all. The big network automation wave might not arrive tomorrow, but it is coming.

When the wave hits shore, you want to make sure that it doesn't crash over your head. You will be at the center of the excitement–that's because the network cannot be automated to the point of being completely hands-off and free of your expertise. In new and different ways, it is always going to require you, the network engineer. Your function stands to change, but your importance does not.

Gary Pfitzer
Greg Coté
Cisco

Start Now

Begin an initial exploration of network automation through a series of short "SDN and Network Programmability Basics" training videos on the Cisco Learning Network: click here.