Making Everything Easier!™

Cisco Special Edition

Software Defined Networking

FOR DUMMIES®

A Wiley Brand

Learn to:

- Grasp the excitement around SDN
- See the benefits SDN offers
- Pick the right SDN solution

Brought to you by

CISCO™

Brian Underdahl
Gary Kinghorn

Software Defined Networking

FOR DUMMIES®
A Wiley Brand

Cisco Special Edition

by Brian Underdahl and Gary Kinghorn

FOR DUMMIES®
A Wiley Brand

***Software Defined Networking* For Dummies®, Cisco Special Edition**

Published by
John Wiley & Sons, Inc.
111 River St.
Hoboken, NJ 07030-5774
www.wiley.com

ISBN 978-1-119- 09855-3 (pbk); ISBN 978-1-119-09856-0 (ebk)

Manufactured in the United States of America

10 9 8 7 6 5 4 3 2 1

Publisher's Acknowledgments

Some of the people who helped bring this book to market include the following:

Project Editor: Jennifer Bingham
Acquisitions Editor: Katie Mohr
Editorial Manager: Rev Mengle
Business Development Representative: Karen Hattan

Table of Contents

Introduction

Are you trying to figure out exactly what software defined networking is and whether it should be part of your organization's technology plans? If so, this book is designed to help.

Software defined networking (SDN) is a new way of looking at how networking and cloud solutions should be automated, efficient, and scalable in a new world where application services may be provided locally, by the data center, or even the cloud. This is impossible with a rigid system that's difficult to manage, maintain, and upgrade. Going forward, you need flexibility, simplicity, and the ability to quickly grow to meet changing IT and business needs.

About This Book

Software Defined Networking For Dummies, Cisco Special Edition, shows you what SDN is, how it works, and how you can choose the right SDN solution. This book also helps you understand the terminology, jargon, and acronyms that are such a part of defining SDN.

Along the way, you'll see some examples of the current state of the art in SDN technology and see how SDN can help your organization.

Icons Used in This Book

This book uses the following icons to call your attention to information you may find helpful in particular ways.

The information marked by this icon is important and therefore repeated for emphasis. This way, you can easily spot noteworthy information when you refer to the book later.

This icon points out extra-helpful information.

This icon marks places where technical matters, such as SDN jargon and whatnot, are discussed. Sorry, it can't be helped, but it's intended to be helpful.

Paragraphs marked with the Warning icon call attention to common pitfalls that you may encounter.

Beyond the Book

You can find additional information about Cisco's take on SDN by visiting:

- http://cisco.com/go/aci
- http://cisco.com/go/sdn
- http://blogs.cisco.com/tag/sdn

Chapter 1

Introducing Software Defined Networking

In This Chapter

- Getting to know SDN
- Looking at the hype and excitement

Networks have become an absolutely essential element in today's modern business climate. Whether the network is completely on-premises, cloud-based, or a hybrid of both, networks provide the vital communication links that organizations need in order to run their applications, deliver services, and be competitive. Software defined networking (SDN) represents a whole new way of looking at how networks are configured, controlled, and operated.

This chapter provides an introduction to SDN, explains why SDN is needed, and shows you why people are so excited about the arrival of SDN solutions.

Understanding SDN

Many people have different definitions of SDN. Probably because SDN is evolving as the technology matures and solutions are introduced. In general, *SDN* most commonly means that networks are controlled by software applications and SDN controllers rather than the traditional network management consoles and commands that required a lot of administrative overhead and could be tedious to manage on a large scale.

Initially, there was a great deal of enthusiasm around SDN simply because software-based control was much more flexible

than the old, rigid management consoles and command line interfaces (CLI).

This capability to control networks through software quickly led to the realization that many complex IT tasks that had to be implemented through clunky management tools could now be automated and done much more efficiently. Speed and automation are key requirements for emerging cloud and multitenant networks that need more scale and can't be bogged down with tedious administrative tasks. In fact, cloud automation (in its many forms) quickly emerged as a primary use case for SDN technology. Today, many SDN solutions are really platforms for hosting cloud automation solutions.

When SDN first appeared on the technology landscape, there were more rigid ideas of how SDN architectures should be designed and what defined an SDN solution. Today, customers take a broader view of what kind of SDN solution is right for them. As the primary use case for SDN has evolved toward cloud automation, customers consider what they're looking for in a policy-based automation solution instead of just the specifics of the underlying SDN technology.

SDN is also truly an open technology. This leads to greater interoperability, more innovation, and more flexible, cost-effective solutions. If a network is compliant with the right SDN standards, it could be controlled by multiple SDN controller applications. This is better than each network platform having its own management console and commands that increase vendor lock-in and make network management even more complex. Today, multiple SDN standards are evolving in different areas, and successful SDN strategies will always be based on open, interoperable multivendor ecosystems with key open source technologies or standardized protocols.

Along with the evolution to SDN, there are a number of technology trends that are affecting the architecture and design of modern data center and enterprise networks that have to be factored into SDN technology requirements. In most organizations, the data center is shifting away from traditional client-server architectures to models in which significantly more data is being transferred between servers within the data center (frequently called east-west traffic). This requires more network scalability and more sophisticated policies for

resource allocation. In addition, many IT departments are showing great interest in moving to public, private, or hybrid cloud environments.

Public cloud services from companies such as Amazon, Microsoft, and Google have given corporate IT departments a glimpse of self-service IT and demonstrate how agile applications and services can be. Organizations are now demanding the same service levels from their own IT departments. SDN, in fact, is being looked at as a key contributor to increasing IT agility and improving self-service IT offerings.

Enterprises are also investing in big data applications to facilitate better business decision making. These types of applications require massive parallel processing across hundreds or thousands of servers. The demand to handle huge data sets is placing greater stress on the network and driving the need for greater capacity and automation.

All of these elements play a significant role in the demand for more efficient, agile, and higher performing corporate network environments. SDN is intended to meet those demands.

The top level benefits of an SDN strategy accrue to all areas of the organization. You will receive a competitive advantage because your infrastructure will do more. The speed of your business will increase, the total cost of ownership will go down and your risks will be decreased because of the greater security.

Where were we before SDN?

To better understand why SDN has become so important, you need to look at what existed before SDN. Traditional networking architectures have significant limitations that must be overcome to meet modern IT requirements. Today's network must scale to accommodate increased workloads with greater agility, while also keeping costs at a minimum. But the traditional approach has substantial limitations:

- **Complexity:** The abundance of networking protocols and features for specific use cases has greatly increased network complexity. Old technologies were often recycled as quick fixes to address new business requirements. Features tended to be vendor specific or were implemented through proprietary commands.

- **Inconsistent policies:** Security and quality-of-service (QoS) policies in current networks need to be manually configured or scripted across hundreds or thousands of network devices. This requirement makes policy changes extremely complicated for organizations to implement without significant investment in scripting language skills or tools that can automate configuration changes. Manual configuration is prone to error and can lead to many hours of troubleshooting to discover which line of a security policy or access control list (ACL) was entered incorrectly on a given device. In addition, when applications were removed, it was almost impossible to remove all the associated policies from all the devices, further increasing complexity.
- **Inability to scale:** As application workloads change and demand for network bandwidth increases, the IT department either needs to be satisfied with an oversubscribed static network or needs to grow with the demands of the organization. Unfortunately, the majority of traditional networks are statically provisioned in such a way that increasing the number of endpoints, services, or bandwidth requires substantial planning and redesign of the network.

Where is SDN taking us?

Initially, there was a lot of hype around SDN prior to understanding real customer use cases. But what is slowly emerging and driving all the investment, pilots, and product designs is a much better way to manage the enterprise WAN, data centers and cloud networks — and to automate IT tasks so that the infrastructure can respond dynamically to rapidly changing business conditions and requirements. The intelligence to make all that happen is moving from the network devices and device management consoles to SDN-based centralized policy management controllers.

What's caused the biggest evolution in SDN is the realization that very few organizations really have the desire, skills, and incentives to write a new class of applications to program the network. The vast majority of organizations are just looking to automate IT tasks, accelerate application deployment, make their cloud networks more flexible, and better align their IT infrastructure with business requirements. The focus has shifted to SDN being a platform capable of hosting a myriad of

orchestration and IT workflow automation solutions that drive customers to their end goal. So network administrators won't necessarily be writing new applications, but they will be buying new turn-key automation solutions built on SDN platforms and technology.

Seeing Why There's So Much Excitement

SDN has generated a lot of interest and excitement in the IT community already. The following sections take a quick look at some of the reasons this is happening

SDN facilitates server virtualization and cloud networks

Traditionally, organizations have had a simple answer to growing demand for data capacity and increasing bandwidth needs — additional (and expensive) hardware capacity. Unfortunately, most corporations can no longer afford such a costly approach, especially in the face of exponential growth in demand. Extremely competitive markets mean that you can't afford to spend your way out of the problem. As one benefit, SDN is helping to leverage server virtualization to increase resource efficiency, reduce the complexity of the network, and simplify the manual IT processes to deploy, manage, and tune applications and networks.

One reason the traditional approach to solving the need for additional computing resources was so expensive is that the traditional model assigned a complete computing unit to a single task. For example, each user had her own, dedicated PC and each network server consisted of a physical computer located somewhere in a rack in a data center. The more modern approach uses server virtualization to make a single physical server function as if it's actually multiple servers, each fulfilling different tasks and using the full capacity of the server. SDN makes use of virtualization to greatly expand network efficiency and thus provide solutions to the need for increased capacity without breaking the bank, and simplifying the management of those consolidated resources.

SDN is primarily about policy-based automation

As networks grow in size, they become much more complex to manage and maintain. In the traditional model, this complexity means that ever-more IT resources are needed to handle processes such as provisioning, configuration, and remediation. Quite simply, these have typically been manual processes, so growing your network from ten to one hundred nodes meant that someone had to manually touch and configure ten times as many devices.

SDN changes that equation in a very fundamental way: It automates processes like provisioning, configuration, and remediation via software. Rather than requiring an IT worker to physically configure each piece of hardware, SDN enables you to roll out network changes by sending out software updates.

All SDN implementations aren't equal. For example, Cisco's Application Centric Infrastructure (ACI) fabric encompasses smart network devices that make automating network processes much easier because you don't have to specifically address each device in order to properly propagate those processes where they need to be applied. You'll want to make sure that the SDN solution you choose fully supports the automation features you need in order to reap the full benefits of moving to SDN.

ACI, Cisco's most comprehensive SDN solution, accelerates application delivery times substantially because of simplified and automated IT processes (see Figure 1-1).

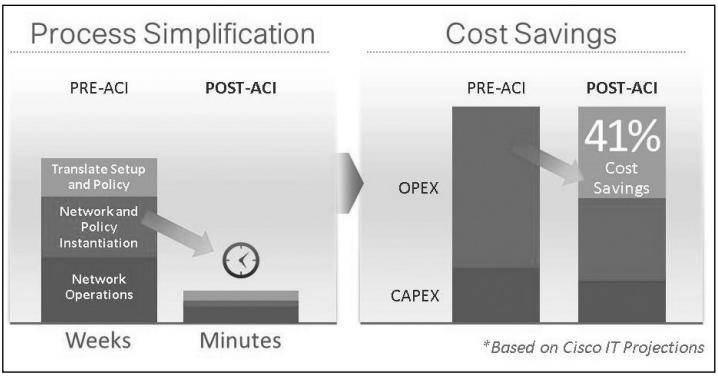


Figure 1-1: How ACI can help.

Chapter 2

Looking at the Benefits and Use Cases of SDN

In This Chapter

- Applying some new ideas
- Seeing where SDN applies

Corporations typically don't follow trends just because they're popular. Rather, decisions are made based upon return on investment (ROI). Something like SDN has to offer real, quantifiable benefits in order to justify its implementation.

This chapter discusses the types of benefits organizations can realize from SDN. It also shows you how SDN's benefits reach well beyond the corporate data center.

SDN Automation Leads to Business Agility

The more agile IT becomes, the more responsive it can be to changing business and application demands, and the more IT will be aligned with business objectives, resulting in greater revenue opportunities and competitive advantage through IT. The agility and the alignment of IT and business objectives are largely accomplished through vastly greater degrees of IT process and infrastructure automation.

Automation takes the serial, box-by-box, component-by-component manual management steps and moves them into a software-based automation platform that accelerates infrastructure processes by orders of magnitude. For example, the

Cisco ACI-enabled data center is built around an application-centric, policy-based automation platform that makes it dramatically easier to represent business and application requirements.

A new approach to network policy

Cisco has taken a very different approach to policy than many SDN solution providers. Traditional SDN policies specified how the network should behave in networking terms. A better approach is to build a policy language that specifies what the business applications need from the network. Policies are defined for groups or classes of applications and the network adjusts to the application requirements.

How the network is implemented isn't directly relevant to the overall business policies of the organization. But how applications work and perform is very closely related to high-level business policies and objectives. This immediately aligns IT and business policies and makes IT more responsive to business requirements.

Cisco calls this new approach *group-based policy* (GBP) — an application-centric policy model that separates information about application connectivity requirements from information about the underlying details of the network infrastructure. GBP makes managing the network much easier, and is one of the benefits of Cisco's SDN solution.

This approach offers a number of advantages, including:

- **An easier, application-focused way of expressing policy:** By creating policies that mirror application semantics, this framework provides a simpler, self-documenting mechanism for capturing policy requirements.
- **Improved automation:** Grouping policies together allows higher level automation tools to easily manipulate groups of network endpoints and applications simultaneously.
- **Consistency:** By grouping endpoints and applying policy to application groups, the framework offers a consistent and concise way to handle policy changes, as well as consistency across both physical and virtual workloads, and physical and virtual application services and security devices.

What Cisco means by policy

Organizations have (at least) two types of policies, a business policy and an IT policy. Business policies are often expressed in real, nontechnical business terms, such as "We authorize this set of applications to run in the cloud when it is cost-effective to do so, or at a particular activity threshold." IT policies are usually reflected in the capabilities of devices (for example, servers, switches, or firewalls), such as "Allow traffic on port:80 to this server," or "Drop traffic from this domain to some subnet."

It is very challenging to translate business policies to IT policies when the IT policies are box-focused. They are in completely different languages built around different concepts. An application-centric IT policy allows for commands with the focus being on the application rather than a box. For example, "Application X can connect to Application Y through a firewall and a load balancer with a guaranteed quality of service."

Most SDN solutions focus on configuring only the network, in network terms, such as "allow traffic on port:80," "assign VM to VLAN 100," etc. Because applications are the lifeblood of a business, it is *much* easier to align business policies with IT policies when the IT policies are application-centric, meaning they can be expressed and managed in terms of the application requirements (as a reflection of business activity).

So, unlike other SDN solutions that historically have focused on the language of the network, ACI focuses at a higher level, expressing policies in the language of application requirements. This accelerates IT's responsiveness to business demands, increases business agility, and aligns business with IT capabilities.

- **Extensible policy model:** Because the policy model is open and can be extended to other vendors and other device types, it can easily incorporate switches, routers, security, Layer 4 through 7 services, and so on.

Centralized policy management with distributed control points

Compared to traditional inflexible network management systems, modern SDN solutions provide an entirely different approach to managing networks that's far more powerful, efficient, and flexible. Rather than requiring an admin to touch every device, this new approach provides programmatic

What Cisco means by application centric

In many traditional or early SDN solutions, the software policy was specific to the infrastructure component or device type. The policy rules were in the language of traditional networking concepts such as network flows, IP addresses, VLANs, and so on. The Cisco Application Centric Data Center is built around a policy model that articulates the requirements of the applications in terms of the connectivity and security policies, and not the infrastructure (network devices and servers).

The ACI automation platform translates these application-specific rules into managing the whole infrastructure. The great advantage is that an application policy model is vastly more relevant to business activity than network or other infrastructure-based policies, making it easier for IT to implement business policies and objectives. It also allows the policy model to extend beyond any single infrastructure component, for example, not just the network as in SDN, but to include the entire infrastructure (network, servers, storage, security, application services) under a single unified policy language. Through this Application Centric policy model, Cisco unifies data center infrastructure and operations under a single system.

software control from a central point that's no longer restricted to the limits of the network management system and doesn't require knowledge of the CLI (command line interface) of the network switches and other devices. Centralized policy repositories make the policies much easier to change and audit. Distributing those policies out to the entire network or cloud infrastructure from a central spot also saves a lot of time compared to updating network nodes one at a time.

Cisco ACI uses a controller called the Application Policy Infrastructure Controller (APIC). APIC is implemented as a cluster of three boxes for fault-tolerance and scalability benefits. The primary function of the controller cluster is to provide policy authority and policy resolution mechanisms. The system isn't directly involved in data plane forwarding, or enforcing policies on individual network flows, so a complete failure or disconnection of all elements in a cluster will not result in any loss of existing data center functionality.

Effectively, this new approach uses software to tell the network nodes what you want them to do — not how you want them to do it. The various devices are intelligent enough to understand how to apply your policies so you no longer need to individually configure each device to make your network function the way you want.

Some Popular Use Cases for Deploying SDN

Some common use cases for SDN beyond general cloud, data center, and IT automation include:

- **DevOps:** DevOps, the synergistic integration of *development* and *operations,* is a rapidly emerging method of developing applications for IT organizations, with the goal of accelerating IT innovation and service delivery (coincidently, some of the same objectives as SDN). Including an SDN technology-based approach can facilitate DevOps through the automation of application updates and deployments and the automation of IT infrastructure components as the DevOps applications and platforms are deployed. DevOps gives developers more control over the IT environment, which not only implies an underlying SDN capability, but also is facilitated by an application-centric approach to policies because DevOps is very software-application oriented.
- **Big Data and Everything-as-a-Service:** Data is routinely collected and traded in the new economy. Big Data brings with it a new class of data and server-intensive applications that present an enormous opportunity to make organizations more efficient and more competitive. But systems must be in place to efficiently and effectively harness, manage, and access it.

  Clouds are taking many forms – private, public, and hybrid (and many are combined into a *multicloud* environment). Business processes are changing; service industries are exploring as-a-service, online, and virtual models to accelerate business and their engagement with customers, partners, and suppliers. SDN can help automate and deploy these new application architectures and allow the infrastructure to adapt quickly to the changing application requirements.

- **Mobility apps and the Internet of Things (IoT):** Consumer technology is reshaping technology expectations at work. A demographic shift to mobile, and more social customers and employees will require IT to integrate with social networks and provide expanded options for flexible workspaces and collaboration technologies. Also, more and more *things* are being linked (via an Internet of Things or IoT) with a view to further expansion into linking people, process, data, and devices (the Internet of Everything or IoE). The changing nature of these application types and the demands for flexibility that they place on the entire network infrastructure can be addressed by SDN solutions more easily than prior approaches.

Expanding Beyond the Network and the Data Center

As organizations realize how IT benefits from network automation, they're now looking to extend the benefits beyond just the network and beyond the data center. SDN can be applied across the entire computing infrastructure, not just to the network, to extend the programmatic control to all the IT resources that applications require. For example, security, various application services, and servers can all be a part of a larger software-defined infrastructure.

There's also a consumerization of IT as users demand more bring-your-own-device (BYOD) flexibility, so that personal laptops, tablets, and smartphones can be used to access corporate information. A result of this trend is a need for greater emphasis on protection of corporate data with security policies and enforcement. Here, too, SDN can play a vital role in distributing policy across the entire infrastructure.

Cisco ACI includes the APIC EM (enterprise module), which extends ACI policies from the data center to the campus network and WAN and access edge. Based on the OpenDaylight Controller architecture, the APIC EM controller is highly programmable through open APIs (representational state transfer, or RESTful, and OSGi APIs). It can enable independent and third-party software developers to create innovative network services and applications to fuel enterprise business growth.

Chapter 3

Understanding the Technology

In This Chapter

- Understanding controllers
- Seeing the role of policies
- Getting to know overlay networks
- Automating with SDN

The benefits of an SDN solution result from applying technology in new and innovative ways. In some cases, the technology itself is new and in others, SDN uses existing technologies but with a different approach than how they were used in the past.

This chapter discusses the important things you need to know about SDN technology, how it is used and how it fits into IT environments.

Getting to Know SDN Controllers

Complexity has become a roadblock to on-time enterprise network service delivery and quality. You need an open and programmable approach to networking through open Application Programming Interfaces (APIs) for policy-based management and security. Essentially, you need a way to automate what has traditionally been tedious manual configuration.

An SDN controller is the centralized repository of policy and control instructions for the network or application infrastructure. An ideal solution provisions network services

consistently and provides network information and analytics across all network resources: LAN and WAN, wired and wireless, and physical and virtual infrastructures. When you write your control software to implement your desired policies, it is typically the case that you're programming the controller and using APIs on the controller.

The controller bridges the gap between open, programmable network elements and the applications that communicate with them, automating the provisioning (the setup and management) of the entire infrastructure, including the network, services, and applications.

The controller gives you a programmatic interface (sometimes called a *northbound* interface) for setting policies and provisioning services across your network. You use it to eliminate network complexity and simply run your physical and virtual network automatically.

To reduce the burden of managing your network, the controller translates business policy directly into network device-level policy. It automates the deployment, compliance checking, and enforcement of network policies across your environment. A controller also provides:

- Consistency across the enterprise network that keeps downtime to a minimum and lowers operational complexity and associated cost
- Automated end-to-end provisioning and configuration to enable rapid deployment of applications and services
- Support for both existing and new deployments that lets you implement programmability and automation with the infrastructure you already have

In Cisco ACI, the APIC Controller manages an application-centric policy that unifies the network, security, server, storage, application, and cloud teams around the infrastructure requirements of the business-critical applications. See Figure 3-1.

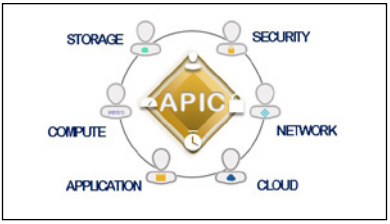


Figure 3-1: APIC provides a common SDN controller across IT teams.

Understanding Policies

Businesses need to deploy, scale, and optimize applications on-demand to increase business agility and lower operating costs. SDN accomplishes this through programmatic extensions to the network, so that policies can be implemented in software and the software can automate network administration tasks, reducing the need for manual operations. This helps achieve cloud-level scale, supporting tens of thousands of servers and millions of workloads. Essentially, Cisco's SDN policies are the instructions that tell the controller how the various network elements are supposed to deal with network and application traffic.

At the heart of ACI is a concept called an application network profile (ANP) (see Figure 3-2 for a visual). The ANP is a policy that defines the requirements of the application in terms of network resources and how application components are connected. When new instances of an application are needed (for capacity expansion or in a new location), the ANP is used as the template. The ANP shows how to deploy the application and all the resources and services it needs wherever it can be optimally placed in the cloud fabric. ACI translates this replicable template into a long series of commands to the individual devices/boxes to properly configure all the services and infrastructure components automatically. These tedious steps are no longer left to the administrators (or multiple teams of administrators, more likely!). This automation accelerates IT processes, aligns them with business needs, and reduces errors in large scale deployments.

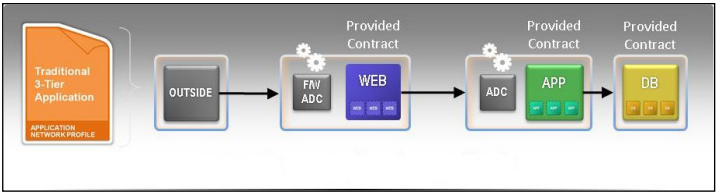


Figure 3-2: An application network policy defining workload connectivity and security and service policies.

To implement policies, Cisco uses an open architecture that includes the SDN controller and policy repository, as well as infrastructure nodes such as network devices, virtual switches, network services, and so on that are controlled by the controller. There is also an open communication protocol between the controller and the infrastructure nodes (also called a *southbound* protocol or interface).

OpFlex is the name of the communication protocol between the controller and the managed devices in the Cisco ACI system. OpFlex was designed with an application-centric policy model in mind. With OpFlex, the devices like the switches and the firewalls still enforce the policies in Cisco ACI. This approach provides the centralization of policy management with distributed control, allowing automation of the entire infrastructure without limiting scalability through a centralized control point or creating a single point of catastrophic failure.

Cisco ACI and OpFlex implement what is called a *declarative* management model. This means that the policy is centralized, but the enforcement of the policy isn't, it's distributed (and performed by the network nodes which decide the best way to enforce the policy). This is compared with the *imperative* model, where the controller issues all the control instructions when they're needed, which requires that the controller understand all the device states in real time, which may be impractical. The imperative model essentially centralizes both the policy and the enforcement, which can be a bottleneck or create a single point of failure.

This declarative design makes sense when you consider that the ways that devices are configured and policies are enforced in network nodes vary considerably across vendors and models. These details would introduce complexity into the policy model or result in a policy that doesn't take full

advantage of each device's full capabilities if everything was enforced directly within the controller. The likely result would be a proprietary system with controlled devices modeled on a small number of device types or a single vendor's products.

OpFlex uses high-level application-oriented policy statements sent to network nodes, and it relies on those devices to determine how to implement the policies. After device configurations are completed, such as configuration of an application delivery controller to support a particular service, communication between the controller and the infrastructure may be required only in the event of policy updates.

OpFlex is a bidirectional communication protocol. In addition to communication in the form of commands from the controller to the managed devices, useful state, performance, and health metrics can be sent from network nodes to either the controller or an external analytics engine (also called an observer).

Considering Overlay Networks in SDN Environments

With the widespread adoption of server virtualization, virtual networks are increasingly important to the overall data center and cloud infrastructure. As a result, SDN's primary use cases involve cloud networks and virtual applications. Virtual networks provide a flexible, programmable infrastructure that SDN can take advantage of. A virtual network can be called an overlay network because it defines a new logical topology on top of an underlying physical network.

Virtual network overlays are logically separated virtual networks of interconnected nodes that share an underlying physical network, allowing deployment of applications that require specific network topologies without the need to modify the underlying or physical network. They're frequently used to isolate tenants and applications that share the same physical network such as a cloud provider. A *tenant* is a user of shared cloud resources. A tenant might be an organization with all its applications or an application comprised of a bunch of workloads that are all logically connected but isolated from all other application networks.

Network overlays offer a number of benefits that can help meet some of the challenges of the modern data center:

- **Fabric scalability and flexibility:** Overlay technologies allow the network to scale by focusing on the network overlay edge devices. With overlays used at the fabric edge, the spine and core devices are freed from the need to add end-host information to their forwarding tables. Additionally, the placement of end hosts is more flexible because the overlay virtual network no longer needs to be constrained to a single physical location.
- **Arbitrary Layer 2 connectivity over Layer 3 underlay:** Another benefit of network overlay technologies is that they can decouple the network services provided to end hosts from the topology used in the physical network. For example, by using certain network overlay technologies, services that require a Layer 2 network can be extended across a routed Layer 3 topology.
- **Overlapping addressing:** Most overlay technologies used in the data center allow virtual network IDs to uniquely scope and identify individual private networks. This scoping allows potential overlap in MAC and IP addresses between tenants. The overlay encapsulation also allows the underlying infrastructure address space to be administered separately from the tenant address space.
- **Separation of roles and responsibilities:** With the encapsulation used in overlay technologies, separation of administration domains can also be achieved. The administrator of the cloud fabric (infrastructure) can be responsible for fabric addressing, availability, and load balancing, while the individual tenants can be responsible for their own addressing policies and services without affecting infrastructure policies.

VXLAN (Virtual Extensible Local Area Network) is an overlay protocol initially developed by Cisco and promoted as a standard by VMware, Red Hat, Citrix, and other virtualization infrastructure vendors. It is used in many SDN installations, particularly those that rely on VMware or KVM hypervisors. Another common overlay network technology is NVGRE

(network virtualization using generic routing encapsulation), common in Microsoft data center environments.

VXLAN and NVGRE are both examples of overlay networks that span not only physical Layer 2 networks, but Layer 3 routed networks. This means that you can have one VXLAN network that runs across Layer 3 connections to other data centers and to cloud providers. It can go wherever you might need it.

These overlay networks are all implemented by encapsulating network packets within a new overlay packet header that may or may not be recognized by network devices. Only the edge switches that represent the beginning and end of the overlay tunnel need to encapsulate or decapsulate the packet. The intermediary network devices can pass the packets unimpeded.

Any cloud automation platform running on an SDN architecture will necessarily need to understand how to set up and configure these virtual network overlays in support of application deployments and to establish new tenants in cloud environments. Virtual switches, virtual firewalls, and so on will also need to integrate with the SDN infrastructure (speaking the right protocols, for example, and interfacing with the SDN controller).

Automating Your Cloud via SDN

Data center infrastructure is transitioning from an environment that supports relatively static workloads confined to specific infrastructure silos to a highly dynamic cloud environment in which any workload can be provisioned anywhere and can scale on demand according to application needs. This transition empowers developers to build the next generation of IT applications, but at the same time it also places new requirements on the computing, storage, and network infrastructure.

SDN solutions can provide a powerful framework for hosting and adding value to cloud automation solutions, which provide higher level workflow and process automation services, and rely on the underlying SDN framework to program, provision, and configure infrastructure nodes. The ideal SDN framework will be capable of hosting a range of cloud automation and workflow automation tools through the open northbound

interfaces on the SDN controller. This flexibility will give customers a choice in what automation tools they can choose for their needs.

OpenStack is a leading open source cloud orchestration tool available today and is backed by more than 125 companies. OpenStack was designed as a building block for public and private clouds, allowing automated management of computing, storage, and networking resources in a very flexible manner.

Cisco's ACI and OpenStack were both designed to help IT administrators navigate the transition to cloud architectures. Cisco is extending the ACI policy framework to OpenStack environments to enable customers to build rich application-driven network policies in their cloud environments. On top of ACI, organizations can now have a certified, supported, turn-key OpenStack solution.

The ACI fabric offers a native service-chaining capability that allows a user to transparently insert or remove services (like an application delivery controller or a firewall) between two endpoints. These services are part of the contract between two endpoint groups (EPGs) in the ACI architecture. The fabric can be configured in real time using the API of the service appliance, for example the firewall, allowing easy deployment of complex applications and security policies in a fully automated manner.

Chapter 4

Key SDN Considerations and Requirements

In This Chapter

- Understanding why you should be application centric
- Seeing how to keep your options open
- Maintaining what you already have

The emergence of software defined networking (SDN) promises a new era of centrally managed, policy-based automation tools that could accelerate network management, optimization, and remediation. To reach those goals you need to be aware of certain SDN-related considerations and requirements, so this chapter discusses several important topics that will help you understand what you need.

Focusing on Applications

Corporations are looking at cloud automation tools and SDN architectures to accelerate application delivery, reduce operating costs, and increase business agility. The success of an IT or cloud automation solution depends largely on the business policies that can be carried out by the infrastructure through the SDN architecture. You need a more business-relevant application policy language, greater scalability through a distributed enforcement system rather than centralized control, and greater network visibility.

Historically, IT departments have sought out greater automation to overcome the challenges of applying manual processes for critical tasks. About 20 years ago, the

automation of desktop and PC management was an imperative, and about 10 years ago, server automation became important as applications migrated to larger numbers of systems. Today, with the consolidation of data centers, IT must address not only application and data proliferation, but also the emergence of large scale application virtualization and cloud deployments.

What's needed is a network fabric designed as an application-centric intelligent network where the policy model is defined from the top down as a policy enforcement engine focused on the application itself and abstracting the networking functions underneath.

This policy model is built on a series of one or more tenants, which allow the network infrastructure management and data flows to be segregated. Tenants can be customers, business units, or groups, depending on organization needs. Below tenants, the model provides a series of objects that define the application itself. In the Cisco ACI system, these objects are endpoint groups (EPGs) and the policies that define their relationships. The relationship between two endpoints, which might be two virtual machines connected in a multitier web application, can be implemented by routing traffic between the endpoints to firewalls and application delivery controllers (ADC) that enforce the appropriate policies for that application.

Prior SDN solutions that focused on network protocols rather than application and business requirements can't match the scalability, flexibility, or integration of more comprehensive frameworks with an application-centric model.

Cisco's ACI infrastructure supports an application-centric approach and the requirements of a group-based policy model through the:

- Declarative control model inherent via the OpFlex protocol
- Policy language based on the requirements of the application
- Flexibility of incorporating a wide-range of network devices as policy control points to implement policy directives

Keeping Things Open

In recent years many people in the computer industry have adopted the concept of *open* — or *nonproprietary* — ways of doing things. This open approach reduces the ability of a single vendor to lock you into buying everything from it, thus increasing market competitiveness. Rather than being locked in to a single source, the market is able to choose the best products from multiple vendors. Most open source proponents feel that this openness leads to better products, lower costs, more innovation, and higher return on investment.

SDN is an open and transparent approach to the evolving needs in cloud networking and data center management. In fact, some of the biggest names in computing including Arista Networks, Big Switch Networks, Brocade, Cisco, Citrix, Ericsson, HP, IBM, Juniper Networks, Microsoft, NEC, Nuage Networks, PLUMgrid, Red Hat, and VMware founded the OpenDaylight Project in 2013 to collaborate on an open source SDN controller architecture. Cisco has also contributed key components of its SDN technologies to standards groups like IETF, the OpenStack consortium, and the OpenDaylight Project itself.

Even though SDN started as an open approach, that doesn't mean that all SDN solutions are equal. Some SDN vendors may lock you in to certain components if you adopt their SDN offerings, such as being forced to use their hypervisor, or their cloud automation platform, even if some of the components conform to specific SDN standards.

SDN: Open Protocols, Open Architectures, Open Ecosystems, and Open Source

An SDN data center automation framework, like Cisco ACI, can't be a monolithic, proprietary architecture. Organizations require an open, extensible, multivendor architecture that incorporates a wide range of infrastructure products and

open source solutions. ACI was designed as an open system, including:

- **Large ACI Ecosystem:** Cisco ACI has developed a large, compatible multivendor ecosystem of data center solutions, with major partners as of early 2015, including:
  - **Layers 4 through 7 Application Services:** A10 Networks, Citrix, Embrane, F5, and Radware
  - **Security Solutions:** Catbird, Check Point, Cisco ASA, SourceFire, and Symantec
  - **Server and Virtualization Infrastructure:** Canonical Ubuntu, Microsoft, Red Hat, and VMware
  - **Cloud Orchestration:** Cisco UCS Director, Cloudstack, Microsoft System Center and Azure, OpenStack, and VMware vCenter and vCloud Automation Center
  - **DevOps Solutions:** Puppet, Chef, and CFEngine
  - **Monitoring, Management and Analysis Tools:** CA Technologies, Cisco Prime NAM, Emulex, NetScout, NetQoS, and Splunk
  - **Server, Storage, Applications and Converged Infrastructure:** BMC, Cloudera, EMC, MapR, NetApp, Nutanix, Panduit, VCE, and SAP
- **Open API:** Open northbound interfaces on the APIC controller support a number of commercial orchestration and automation tools on top of the ACI policy model.
- **Open communication protocols:** The communication protocol (OpFlex) from the APIC controller (southbound interface) to managed devices is a published standard, along with the design of the device-side agent (available as open source and reference design), so that compatible devices (new switches and application services, and so on) can become ACI-compliant.
- **Open source efforts:** Key architectural components of the ACI architecture have been offered to the open source community to increase innovation, facilitate the development of compatible products, and further foster a compatible multivendor ecosystem, namely:

- ACI policy model specification contributed to the Open Daylight project for inclusion in the open source controller.
- Contributions of interface specifications and group policy model specification to the OpenStack consortium.
- Distribution of an OpFlex agent for the Open Virtual Switch (OVS) through major Linux open source vendors Canonical and Red Hat, as well as Microsoft for their Windows Server–based virtual switch.

Keeping What You Have by Using Cisco ACI

For the vast majority of organizations, it's simply not reasonable to throw out your existing investment in network infrastructure and start fresh. Yes, you want the benefits of implementing an SDN solution, but you also need to make use of what you already have. The challenge is finding a way to do both.

Cisco's ACI introduced an application-centric language and programmability model that aligns with business and application objectives, simplifying policy implementations and aligning IT with business needs. ACI also extends the network-specific approaches of SDN to the data center infrastructure, incorporating servers, storage, application services, and security under one policy model.

To achieve this new fabric architecture, critical features were built into the Cisco Nexus 9000 Series Switches, including hardware-optimized overlays that simplify the network topology and segment tenant and application networks. The ACI infrastructure controller, APIC, maintains the application policies and device templates that configure, provision, and manage fabric nodes, including application services, security devices, and the virtual infrastructure to support automated application deployment and optimization.

This ACI architecture is designed to apply policy rules to a fundamental application element, an endpoint, or a class of similar endpoints, called an endpoint group (EPG). *Endpoints*

are essentially workloads, either physical or virtual, with multiple EPGs making up the various tiers of a multitier application.

As you adopt and migrate to the ACI architecture, a critical requirement is that you're able to use your existing data center switching investments in Cisco Nexus hardware that predate ACI. You want to manage your data center applications as EPGs managed under APIC, but connected through networks other than Cisco Nexus 9000 Series networks. Additionally, as you build out ACI fabrics, you want to preserve your investment in existing models by using them in your new ACI network, under a common orchestration platform and a common policy wherever possible.

Most customers deploying Cisco ACI fabrics today have existing Cisco Nexus networks, usually three-tier (edge, aggregation, and core), based on some combination of Cisco Nexus 2000, 3000, 5000, 6000, and 7000 Series devices. These existing networks support a combination of physical and virtual workloads, with the virtual workloads attached to fabric extenders (FEX) or virtual switches. In addition to building out new pods based on ACI, you may want to use the existing architecture to apply ACI policies to EPGs attached to the fabric.

To meet this requirement, Cisco extends ACI policy support to application workloads across the entire existing network through a Nexus 1000V virtual switch that is enabled for ACI, called an Application Virtual Switch (AVS). In this scenario, all data traffic flows through the infrastructure that existed before ACI was added; the traffic flows to a Nexus 9000 Series network switch managed by APIC. The network can be any number of layers (such as a two- or three-tier network), with AVS switches running on the servers to connect to the EPG virtual workloads. Physical workloads can be connected directly to a Nexus 9000 Top of Rack (ToR) switch.

EPG-to-EPG workload connectivity within this network on separate servers could be handled normally, for example, through a VXLAN tunnel between servers, without reaching the gateway to communicate with workloads on the ACI network. All workloads can then be managed and provisioned as an EPG, with consistent policies across all servers and segments of the network, under a single policy controller and

orchestration model. APIC can also support cloud orchestration software, such as OpenStack and Cisco UCS Director, through northbound interfaces on the controller.

The ACI policy model provides the capability to group EPG objects based on the service they deliver and the policy and connectivity required. This feature alleviates constraints inflicted by today's networks, which require forwarding constructs such as VLANs and subnets to be used for this purpose. In this model, groups of objects are constructed, and connectivity and policy enforcement is built in.

Because policy configuration is the most complex and tedious change in a network as applications are added, removed, expanded, or moved, automation of policy changes is critical. Existing VLANs, subnets, and so on have already been configured and don't require frequent modification. It is the individual device policies applied to each application that change frequently. Thus, device packages and policy definitions within APIC are capable of fully automating policy changes, without the need to reconfigure the existing network topology or equipment.

The design of the ACI fabric allows incorporation of existing Nexus 2000, 3000, 5000, 6000, and 7000 Series networks under a common cloud orchestration platform. It also allows management of application workloads attached to the network that existed before ACI was added; these workloads are managed as APIC endpoint groups. This capability preserves existing investment in switches, which can be used in the new ACI fabric model.

Chapter 5

Ten Important Facts about Evaluating SDN Solutions

In This Chapter

- Ten SDN facts

We've covered a lot of ground in the earlier chapters bringing you up to speed on SDN. This chapter brings all of that information together to present ten important facts you'll want to consider as you evaluate SDN solutions.

Programming Rather than Managing Manually

A modern and useful definition of SDN tells you that your networking infrastructure is now programmable via software. This programmability is a very important change compared to the traditional method of managing the network via command line interface or management GUI/console. SDN programmability allows for greater flexibility, more sophisticated policies, and more agile networks overall.

SDN doesn't necessarily require separation of the network control plane from the data plane as many originally defined it. Today there are a number of interrelated technologies, protocols, and architectures that lead to policy-based automation and achieve the original vision of SDN.

Making for Easier Policy Implementation

The SDN programs that manage and control your infrastructure are also referred to as your *policies.* They can reflect your sophisticated application, IT, or business policy objectives; whatever you can implement in software within the SDN framework. By implementing your policies in software, rather than the arcane language and management tools of network devices, your policies can more easily align with business requirements, and be much easier to maintain and deploy. This allows SDN to be the key to policy-based automation, which drives down complexity and operational overhead by automating many of your IT tasks while also allowing for much greater scalability for cloud networks.

Supporting Multivendor Ecosystems

Another important consideration to remember is that for the SDN policy model to automate the entire application infrastructure, a large multivendor ecosystem across a large number of infrastructure device types must be supported by the SDN platform. You should consider what firewalls, application delivery controllers, and other security devices are supported by the proposed SDN architecture and standards.

This consideration also includes whether you can incorporate your existing network infrastructure into the SDN environment when it's deployed, or to what extent you will have to upgrade your hardware to SDN-compliant devices.

In the case of Cisco ACI, a large ecosystem of partners has become ACI-compliant, including F5, Citrix, Check Point, Cisco ASA with FirePOWER security services, and many others. This abundance of vendors and solutions provides you with far greater flexibility.

Understanding the Importance of the Controller

The software policy repository that is the brains of your IT automation is generally referred to in most SDN platforms as the *controller.* The controller architecture is the key to many features of the SDN platform, how it is programmed, and what devices can be controlled. There may be a variety of controllers and platforms to consider when selecting an SDN strategy, so make sure you choose a controller that does everything you need. A proprietary controller or one that supports a limited SDN policy model will limit your capabilities down the road.

Speaking the Right Language

The language (or policy model) of the SDN policy/controller greatly determines how the infrastructure can be automated and the flexibility you have in programming it. For example, Cisco ACI presents an application-centric policy model that represents the SDN policy in terms of application requirements and capabilities. This model allows a single policy to span the entire IT infrastructure, including network, security, application services, and so on.

With the right policy model in place, you can extend the advantages and methodology of SDN beyond the network to the entire application infrastructure and beyond the data center to the campus WAN and remote sites.

Using the Northbound Lane

Customized software programs written for an SDN environment usually run on top of the SDN controller through a *northbound interface* — an API running on the controller that hosts software applications. The specifications of this northbound API can also greatly influence the flexibility of how and what you can program the infrastructure to do, and what automation tasks your SDN platform can perform.

The northbound API should be open, flexible, and well-integrated with the capabilities of the controller. It should also support the applications such as cloud orchestration tools that you will need to run.

Making Use of Cloud Automation Tools

Many of the new generation of SDN applications that reside on the northbound API of the controller are themselves cloud automation tools, and they help to manage workflows and tasks for cloud application deployment and infrastructure management. These tools rely on the SDN controller to communicate and control the individual infrastructure components (switches, servers, firewalls, and so on) according to the SDN policy and the higher level automation tools.

Some examples of cloud automation platforms that can sit on top of SDN controller northbound interfaces are OpenStack and Cisco UCS Director.

Choose an SDN platform that gives you the greatest flexibility in cloud automation tools and platforms rather than being locked into one vendor's cloud stack. In addition to cloud management tools, any SDN platform should support a number of monitoring and performance tools that can help analyze, troubleshoot, and automate the tuning of the applications and the network.

Considering an Extensible Architecture

Because the SDN controller is the operational hub of all your integration and automation efforts, it must be an open, extensible architecture in a number of important ways (such as an open, flexible northbound API to host multiple applications or cloud automation tools). It must have an open way of communicating to a multivendor IT infrastructure of switches, routers, firewalls, application delivery controllers, and more. Make sure your SDN architecture meets all the

open requirements to successfully navigate and simplify your deployments.

OpenFlow was an early example of an open way to integrate switches from multiple vendors to a single SDN controller, but more sophisticated protocols and devices have now been developed for modern SDN platforms. For example, the Cisco ACI platform uses OpFlex as an open, generic way to communicate to a wide range of vendor platforms and device types.

Choosing an Open Source SDN Platform Carefully

Openness and interoperability are very important to any SDN strategy, so you don't want to get locked into a particular controller platform, or you may want to consider an open source controller. This choice may provide a less expensive architecture initially, but may restrict integration to a more narrow policy that runs across multiple infrastructure components, or may require a lot of customization and integration for any particular cloud environment and policy requirements.

An open SDN platform should support all major server operating systems and hypervisor platforms including VMware, Microsoft, and Linux KVM to provide maximum coverage and flexibility.

An important requirement when considering an open source SDN controller, such as OpenDaylight, is the size of the vendor ecosystem that supports it, key vendors that can provide turn-key solutions for particular environments, and an open policy model that allows for the greatest flexibility in terms of automation and programmability. Cisco ACI has, for example, contributed its application-centric policy model to the OpenDaylight controller project. Cisco also supports a commercial implementation of the OpenDaylight controller for enterprise networks as well as service providers.

Keeping It Seamless

Because cloud environments are composed of both virtual and bare-metal applications, as well as virtual and physical infrastructures (networks, firewalls, service nodes), the SDN platform, policies, and automation tools must operate seamlessly across this physical and virtual spectrum. Too often the automation only applies to the virtual infrastructure, or a particular class of physical devices (a particular vendor's switches, for example). A successful SDN strategy has to span everything, with a common set of policies, tools, and infrastructure if the IT automation strategy is to be successful.

Appendix

SDN Acronyms

SDN is a pretty recent development, but you'd never know that from all of the specialized terminology, jargon, and acronyms that have evolved to describe the subject. It can sometimes seem like an SDN expert is talking in some strange, exotic foreign language.

This appendix brings together as much of the SDN and related networking and security terminology as we could fit into the available space so you don't have to keep wondering just what a discussion of SDN is all about. We've arranged the entries in alphabetical order to make it just a bit easier for you to quickly look up definitions, but we recommend that you take the time to read through the entire list—you'll probably find some unexpected gems!

ACI: Application Centric Infrastructure, Cisco's most comprehensive SDN architecture.

ACLs: Access control lists.

ADC: Application delivery controller.

APIC: Application Policy Infrastructure Controller, Cisco ACI's controller component for single-point policy management and automation.

ARP: Address Resolution Protocol.

ASA: Cisco Adaptive Security Appliance.

ASICs: Application-specific integrated circuits.

AVS: Cisco's application-centric virtual switch, a component of its ACI architecture, a software switch that resides in the hypervisor and connects virtual workloads to the network.

BCB: Backbone core bridge.

BEB: Backbone edge bridge.

BGP: Border gateway protocol.

Bridge domain: A container for subnets.

BYOD: Bring-your-own-device.

CLI: Command-line interface.

CMIP: Common Management Information Protocol.

concrete devices: Actual devices connected to the network fabric.

CONGA: Congestion-aware load balancing.

context: A representation of a private Layer 3 namespace or Layer 3 network.

contract: An ACI policy between two EPGs, specifying the services between them.

control plane: The channel used by the SDN fabric nodes for management commands and to distribute policies for communication between nodes (separate from the underlying data).

data plane: The section of an SDN fabric containing the actual information being passed between network nodes.

DCI: Cisco data center interconnect, for networking between data center sites.

declarative (model): An orchestration model in which control is distributed to and enforced by intelligent devices based on centralized policies, rather than a central controller providing specific control commands.

DevOps: Development and operations, a new model of integrated application development and deployment that includes both organizations.

DVS: VMware's Distributed Virtual Switch.

east-west traffic: Network traffic between application components, usually within a tenant's application network, and flowing within the data center.

EIDs: Endpoint identifiers.

endpoint registry (ER): A registry of endpoints currently known to the Cisco ACI and APIC controller.

EPGs: Endpoint groups, classes of workloads, such as the web tier of an application, that share the same set of policies.

event manager: A repository for all events and faults initiated from the APIC or the fabric nodes.

EVPN: Ethernet virtual private network.

FEX: Fabric extender.

GBP: Group-based policy, a standard policy model based on application-oriented end-point groups.

GUI: Graphical user interface.

imperative (model): A top-down style of device management where the central manager/controller must be aware of the configuration commands of the underlying objects and their current state so the controller can issue all of the SDN control commands.

IS-IS: Intermediate system to intermediate system, a common Layer 3 routing protocol.

ITRs: Ingress tunnel routers.

JSON: Javascript Object Notation.

KVM: Linux Kernel-based Virtual Machine, the standard hypervisor for Linux-based applications.

LISP: Location/Identifier Separation Protocol.

LLDP: Link Layer Discovery Protocol.

logical devices: An abstraction of clusters of active and standby pairs of devices such as firewalls and load balancers.

MDT: Multicast distribution tree.

MPLS: Multiprotocol Label Switching.

MSS: Maximum segment size.

MST: Multiple Spanning Tree.

MTU: Maximum transmission unit.

multitenancy: Network slicing into logical network topologies that are assigned to separate tenants in a cloud environment.

NFS: Network file system.

NMS: Network management system.

northbound APIs: Methods that enable applications to program the network and request services.

NSH: Network Services Header, an open service chaining protocol.

NVGRE: Network Virtualization using Generic Routing Encapsulation, a tunneling or overlay protocol alternative to VXLAN, usually in a Microsoft-enabled data center.

observer: the monitoring subsystem of the Cisco APIC that serves as a data repository of the Cisco ACI's operational state, health, and performance.

ONF: Open Networking Forum, an industry standards group from several SDN technologies.

OpenFlow: A key early SDN protocol, that specified how SDN controllers and switches communicated.

OpenStack: An open source cloud orchestration platform.

OpFlex: Cisco's southbound API that controls communication from the SDN/ACI controller to individual devices.

OTV: Cisco Overlay Transport Virtualization.

overlays: Virtual networks of interconnected nodes that share an underlying physical network allowing deployment of applications that require specific topologies without the need to modify the network.

OVS: Apen virtual switch.

OVSDB: Open vSwitch Database.

PBB: Provider backbone bridge.

PEs: Policy elements.

pNIC: Physical network interface card.

policy manager: A distributed policy repository responsible for the definition and deployment of policy-based configurations.

policy repository: A collection of policies and rules applied to existing or hypothetical endpoints.

promise theory: A system of device management that relies on underlying devices, like network switches, to handle state changes and report exceptions or faults (synonymous with declarative model).

Python: A scripting language, frequently used for programming SDN applications and controllers.

QoS: Quality-of-service.

RBridges: Routing bridges.

RESTful: Representational state transfer, typically referring to a type of northbound application programming interface (API) that is common to SDN and web applications.

RLOCs: Routing locators.

RPC: Remote procedure call.

RSTP: Rapid Spanning Tree Protocol.

SCVMM: Microsoft System Center Virtual Machine Manager.

SDK: Software development kit.

service graph: A variation of the concept of a contract, specifying how services are connected and deployed in a network.

service insertion: The capability to add Layer 4 through Layer 7 devices in the path between endpoints by specifying and controlling the routing path to those service nodes.

sharding: A method of database partitioning that increases redundancy and performance.

SNMP: Simple Network Management Protocol.

southbound APIs: Methods used to communicate between the SDN controller and the network switches, routers and other devices being controlled.

SPB: Shortest-Path Bridging.

SSL: Secure Sockets Layer.

tenant: A logical container for the virtual network, resources, policies and applications representing a cloud user/organization.

Topology Manager: Maintains up-to-date Cisco ACI topology and inventory information.

VLAN: Virtual local area network.

VM: Virtual machine.

VNI: VXLAN network identifier.

vNIC: Virtual network interface card.

VNID: Virtual network identifier.

VRF: Virtual routing and forwarding.

VTEP: VXLAN tunnel endpoint.

VXLAN: Virtual Extensible LAN, an overlay or tunneling protocol for creating network slices or tenant networks.

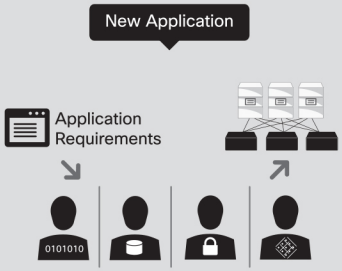
WAN: Wide-area network.

XML: Extensible markup language.

Why Choose Cisco Application Centric Infrastructure (ACI)?

IT Automation at the Speed of Business

Without ACI

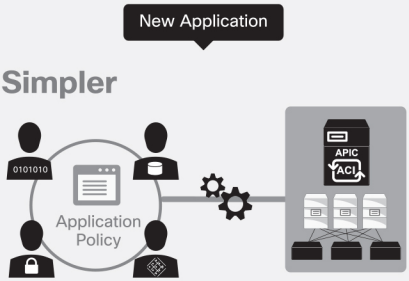


IT administrators work in silos

Manual tasks limit agility

Weeks

With ACI



Simpler

Faster

Shared model for policy automation reduces overhead

More Efficient

Policy-based automation accelerates IT agility

Minutes

CISCO

Redefine the Power of IT with ACI

Learn more at: www.cisco.com/go/aci

Automate business policies, streamline IT operations, and reduce operating expenses!

Software defined networking (SDN) provides a whole new way to approach networking that enables you to programmatically control network operations from a central point with far greater efficiency than ever before.

- ***SDN policies simplify control* — simple, easy-to-understand policies take the place of complicated, device-specific commands**
- ***Open architectures* — reduce vendor lock-in while increasing flexibility and investment protection**
- ***Focus on applications* — Use an application-centric policy approach to improve network automation**

Open the book and find:

- **Why SDN makes sense for your organization**
- **How virtualization and overlay networks fit into SDN**
- **Why open standards are important**
- **What policies mean to SDN**
- **The meaning of all those SDN acronyms and special jargon**

Go to Dummies.com®
for videos, step-by-step examples, how-to articles, or to shop!

Brian Underdahl is the author of over 130 books on modern technology and he feels that explaining that technology helps make it more accessible and useful. **Gary Kinghorn** has worked in product management and product marketing in the IT industry for more than 20 years, primarily in networking and security. He is currently a product marketing manager for data center networking solutions at Cisco.

FOR DUMMIES®
A Wiley Brand

ISBN: 978-1-119-09855-3
Not for resale