CHAPTER

SDN IN OTHER ENVIRONMENTS

9

The urgency associated with the growth of data centers has caused SDN to leap to the forefront of solutions considered by IT professionals and CIOs alike. But the data center is not the only environment in which SDN is relevant. This chapter looks at other domains in which SDN will play a major role. The following environments and some accompanying use cases will be examined in this chapter:

- Wide Area Networks (Section 9.1)
- Service Provider and Carrier Networks (Section 9.2)
- Campus Networks (Section 9.3)
- Hospitality Networks (Section 9.4)
- Mobile Networks (Section 9.5)
- Optical Networks (Section 9.6)

As we examine each of these domains, we will explain the most pressing issues and then we consider the ways in which SDN can play a part in helping to resolve them. We will discuss current implementations and *proofs of concept* (PoCs) that have been developed to address needs in these areas. Before we look into each environment in detail, we will review some advantages that accompany an SDN solution that are particular to these environments.

CONSISTENT POLICY CONFIGURATION

One of the major issues currently facing IT departments is the problem of scale, related to having consistent configuration across very large populations of networking devices. In the past this task has fallen to manual configuration or network management, but with SDN there is the promise of great simplification of these tasks, with the associated reduction in operational costs for the enterprise. Some specifics of SDN with respect to configuration are (1) *dealing with large numbers of devices*, (2) *consistency of configuration*, (3) *central policy storage*, (4) *common policy application*, and (5) *granularity of control*.

The sheer number of devices operating in networks today has grown to the point of becoming nearly unmanageable. In large *Wide Area Networks (WANs)* there may be thousands of physical switches deployed. IT personnel today are required to maintain a large number of devices using traditional tools such as CLI, web interfaces, SNMP and other manual, labor-intensive methods. SDN promises to remove many of these tasks. In order for SDN to work and scale, the controllers are being built from the

Software Defined Networks. http://dx.doi.org/10.1016/B978-0-12-804555-8.00009-0

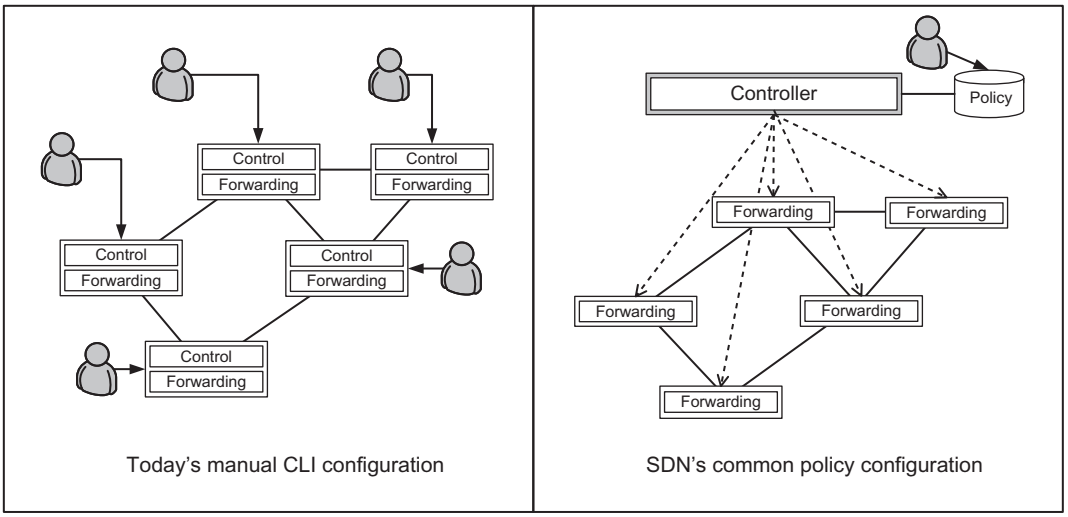

FIG. 9.1

Ensuring consistent policy configuration.

ground up to comprehend and manage a huge number of devices. In addition, this centralization drives more commonality between the devices. The task of managing large numbers of devices is simplified as a direct result of this homogeneity. This simplification of network configuration is one of the major operational cost savings brought about by SDN.

One common issue with large networks today is the difficulty in maintaining consistency across all the devices, which are part of the infrastructure. Fig. 9.1 shows a simple example of the comparison of autonomous devices individually and manually configured versus common policy distributed via flow rules from a central controller. The trivial case depicted in the diagram is compounded by orders of magnitude in real-life networks. Keeping configuration of policy consistent simply cannot be done by purely manual means. For some time now, sophisticated network management tools have existed that purport to provide centralized configuration and policy management. However, as we have described in previous chapters, these tools have only realized marginal success. They are generally based on a model of creating a *driver* that abstracts away vendor and device differences. These tools attempt to standardize at such a high level that the actual low-level vendor and device differences pose a constant challenge. This model has not been able to scale with the amount of change that is endemic in networking. SDN approaches configuration in a radically different way. Configuration is performed in a standard but entirely new and more fine-grained fashion, at the flow level. The configuration is based on the fundamental construct of the flow, which is shared by all networking devices. This permits configuration by a common protocol such as OpenFlow. Rather than trying to band-aid over a configuration morass with a system of the aforementioned drivers, this SDN approach has the ability to scale with the evolution of the network and to implement consistent network-wide configuration and policy. Central policy storage falls out naturally from the centralized controller architecture of SDN.

These attributes stem from SDN's very roots. We recall from Section 3.2.7 that Ethane, the direct precursor to OpenFlow, was specifically targeted at the distribution of complex policy on a network.

GLOBAL NETWORK VIEW

We have mentioned the importance of network-wide visibility in previous chapters, noting this as a real value of controller-based networking. This is true for any type of system that relies on a central management and control system, such as phone systems, power grids, etc. Having complete visibility allows the controlling system to make optimal decisions, as we discuss later. Such a (logically or physically) centralized controller platform can perform global optimizations rather than just per-node optimizations.

The control plane of today's network devices has evolved to the point where each device holds the network topology represented as a graph. This solution is imperfect as the individual graphs held in each device are constructed by propagating changes across the network from neighbor to neighbor. This can result in relatively long periods before the devices converge on a consistent view of the network topology. During these periods of convergence, routing loops can occur. In such networks, it is cumbersome to understand exactly how the network will behave at any point in time. This legacy technology was geared toward maintaining basic connectivity, not the use cases we presented in Chapter 8 and continue to discuss here. In addition, the traditional networks do not have real-time data about traffic loads. Even maximum bandwidth information is not typically shared. A centralized control system is able to gather as much data (traffic loads, device loads, bandwidth limits) as is necessary to make the decisions required at any moment.

Having the aforementioned information makes it possible for the controller to make optimal routing and path decisions. Running on a processor which is many times more powerful than that present on a networking device increases the probability that the controller will be able to perform faster and higher quality analysis in these decisions. Additionally, these decisions can take into account traffic patterns and baselines stored in secondary storage systems, which would be prohibitively expensive for networking devices. The distributed counterpart to this centralized control suffers the additional disadvantage of running a variety of protocol implementations on a selection of weaker processors, which creates a more unpredictable environment. Lastly, if the capabilities of the physical server on which the controller is running are exceeded, the controller can use horizontal scaling approaches, something not possible on a networking device. In this context, horizontal scaling means adding additional controllers and distributing the load across them. Horizontal scaling is not possible on the networking device because such scaling would change the network topology. Changing the network topology does not provide more power to control the same problem, but rather it alters the fundamental problem and creates new ones.

A network-wide view takes the guesswork and randomness out of path decisions and allows the controller to make decisions which are reliable and repeatable. This deterministic behavior is critical in environments which are highly sensitive to traffic patterns and bandwidth efficiency, in which suboptimal behavior based on arbitrary forwarding assignments may cause unacceptable packet delays and congestion.

These benefits of SDN-based networks are generally relevant to the network environments described later. In certain cases they are of particular importance and will be highlighted in those sections.

9.1 WIDE AREA NETWORKS

WANs have historically been used to connect remote islands of networks, such as connecting geographically dispersed LANs at different offices that are part of the same enterprise or organization. A number of non-Ethernet technologies have been used to facilitate this WAN connectivity, including:

- *Leased line*. These point-to-point connections were used in the early days of networking and were very expensive to operate.
- *Circuit switching*. Dial-up connections are an example of this type of connectivity, which had limited bandwidth, but were very inexpensive.
- *Packet switching*. Carrier networks at the center carried traffic from one endpoint to another, using technologies such as X.25 and Frame-Relay.
- *Cell relay*. Technologies such as ATM are still used today, featuring fixed cell sizes and virtual circuits.
- *Cable and digital subscriber loop*. These technologies play a key role in bringing WAN connectivity to the home.

However, recently these technologies have significantly given way to Ethernet-based WAN links, as we describe in greater detail in Section 9.2. Nowadays, companies and institutions are using Ethernet to connect geographically dispersed networks, using either private backbones, or in certain cases, the Internet. When we use the term Ethernet in this context, we are referring to the use of Ethernet framing over the WAN optical links. We do not imply that these links operate as broadcast, CSMA/CD media, like the original Ethernet.

Ethernet has proven to be an excellent solution in areas which were not envisioned in the past. However, the move toward Ethernet brings with it not only the advantages of that technology but also many of those challenges that we have described in detail in this book. The following section addresses some of the major issues that are relevant for WAN environments.

The key issues that face network designers when it comes to WANs are reliability and making the most efficient use of available bandwidth. Due to the cost of long-haul links in the WAN, bandwidth is a scarce commodity as compared to LANs. In environments such as a data center, it is possible to create redundant links with little concern for cost, as cables are relatively inexpensive. Furthermore, bandwidth needs can be met by adding more ports and links. Not so with the WAN, where bandwidth costs are orders of magnitude greater over distance than they are within a data center or campus. Adding redundant ports just exacerbates this problem. As a result, it is important to drive higher utilization and efficiency in the WAN links.

9.1.1 SDN APPLIED TO THE WAN

Overcoming the loss of connectivity on a link using redundancy and failover is common in all forms of networking, including WANs. This technology exists in devices today and has existed for many years. However, routing decisions made during a failover are not predictable during the period of convergence and often are not optimal even after convergence. The suboptimal paths are due to the lack of a central view which sees all paths as well as bandwidth capabilities and other criteria. Such global knowledge permits repeatable and optimal decisions. This aspect of reliability is keenly desired by organizations which rely on wide-area connections to operate their businesses and institutions. As we discussed in

Section 8.4, the SDN controller can have access to all of this information, and as it presumably runs on a high-performance server it can compute optimal paths more quickly and more reliably than the traditional distributed approach.

There are a number of pre-SDN technologies that attempt to enable optimal routing decisions. These include:

- Network Monitoring and Traffic Management Applications. For example, sFlow and NetFlow collect the necessary information so that a network management system can make optimal routing decisions.
- MPLS-Traffic Engineering (MPLS-TE) (see Section 9.2.2) uses traffic engineering mechanisms to choose optimal paths for MPLS *Label Switched Paths* (LSPs). An LSP is a path through an MPLS network. Packets are forwarded along this path by intermediate routers that route the packets by a preassigned mapping of ingress label-port pairs to egress label-port pairs. An LSP is sometimes referred to as an MPLS tunnel.
- The *Path Computation Element (PCE)-based Architecture* [1] determines optimal paths between nodes based on a larger set of network loading conditions than may be considered in simple best-path analysis such as IS-IS.

Even when these pre-SDN technologies attempt to make optimal path decisions based on a global view of the network, they are thwarted by devices that retain their autonomous control planes. We consider an example of this in the following section.

9.1.2 EXAMPLE: MPLS LSPs IN THE GOOGLE WAN

In [2] Google describes how their pre-SDN WAN solution handles the case of a failed link. In this particular scenario the problem revolves around rerouting MPLS LSPs in the event of a link failure.

We depict the pre-SDN situation in Fig. 9.2. In the figure the link on the right labeled *Best route* goes down. When a link goes down and the network needs to reconverge, all the LSPs try to reroute. This

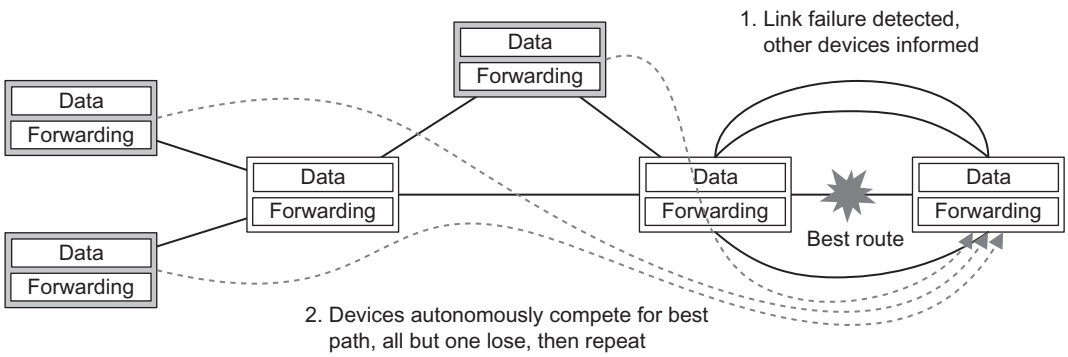

FIG. 9.2

Google without OpenFlow.

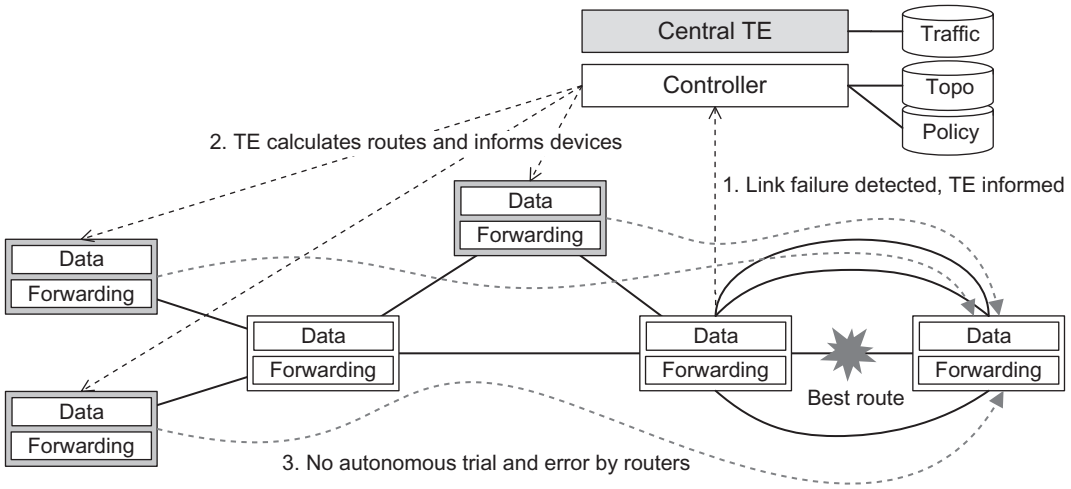

FIG. 9.3

Google with OpenFlow.

process is done autonomously by the routers along the path using the *Resource Reservation Protocol* (RSVP) to establish the path and reserve the bandwidth. When the first LSP is established and reserves the remaining bandwidth on a link, the others have to retry over the remaining paths. Because the process is performed autonomously and *ad hoc*, it can be repeated multiple times with a single winner declared each time. This can iterate for some time until the last LSP is established. It is not deterministic in that it is not known which LSP will be the last in any given scenario. This is a real-life problem and this pre-OpenFlow solution is clearly lacking.

We contrast this with Fig. 9.3, which shows how an SDN system with a centralized controller containing knowledge of all possible routes and current available bandwidth can address this problem much more efficiently. To accomplish this, the controller works in concert with the *Traffic Engineering* (TE) application shown in the figure. The computation of an optimal solution for mapping the LSPs to the remaining network can be made once and then programmed into the devices. This SDN-based solution is deterministic in that the same result will occur each time.

Google has been at the forefront of SDN technology since its outset, and they have made use of the technology to their advantage in managing their WAN networks as well as their data centers. They have connected their data centers using OpenFlow switches connected to a controller which they have designed with the features we just described in mind. The benefits that Google has realized from their conversion to OpenFlow in the WAN are:

- Lower cost of managing devices
- Predictable and deterministic routing decisions in the case of failover
- Optimal routing decisions based on bandwidth and load

DISCUSSION QUESTION

In the example illustrated in Figs. 9.2 and 9.3 is the solution depicted in Fig. 9.3 preferable because the LSPs are ultimately routed over better paths after the failure or merely that they converge on those paths more quickly? What part of this answer cannot be ascertained from the information provided here?

9.2 SERVICE PROVIDER AND CARRIER NETWORKS

Service Provider (SP) and carrier networks are wide-area in nature and are sometimes referred to as *backhaul* networks, since they aggregate edge networks, carrying huge amounts of data and voice traffic across geographies, often on behalf of telecommunications and Internet Service Providers. Examples of Service Providers and carriers are Verizon, AT&T, Sprint, Vodafone, and China Mobile.

A *Network Service Provider* sells networking bandwidth to subscribers, who are often Internet Service Providers. A *carrier* is typically associated with the telecommunications and mobility industries, who in the past have been concerned primarily with voice traffic. The carrier typically owns the physical communications media. The line between carriers and SPs is somewhat blurred. Carriers and SPs now carry a much more diverse traffic mix, including *Voice over IP* (VoIP) and video. In addition to a greater variety of traffic types, the total volume of traffic grows at an ever-increasing pace. Smartphones with data plans have played a large role in increasing both the total amount of traffic as well as its diversity.

Without taking a new approach to managing bandwidth, this growth in diversity and volume of traffic results in costs spiraling out of control. Over-provisioning links to accommodate changing traffic patterns is too costly for the WAN links used by the SPs. Traditionally, bandwidth management is provided via network management platforms which can certainly measure traffic utilization and can perhaps even recommend changes that could be made manually in order to respond to traffic patterns.

Not only does network bandwidth need to be utilized most efficiently, it is also necessary to respond immediately to changes in requirements brought about by service contract upgrades and downgrades. If a customer wishes to have their traffic travel across the network at a higher priority or at greater speeds, SPs would like to be able to implement changes to their service policies immediately, without disruption to existing flows. Being able to provide *bandwidth on demand* is a selling point for SPs. Other requirements include the ability to dynamically change paths to higher bandwidth, lower latency paths, and to re-prioritize packets so that they take precedence in queues as they pass through networking devices.

Another aspect of reducing costs involves the *simplification of devices*. In the large core networks operated by carriers, the OPEX costs of managing a complex device outweigh the increased CAPEX outlay for that device. Thus a simpler device provides cost savings in two ways. This is true as well not only for network devices but also for network appliances which today require specialized hardware, such as load balancers, firewalls, and security systems.

SPs are responsible for taking network traffic from one source, passing it throughout the SP's network, and forwarding it out the remote network edge to the destination. Thus the packets themselves must cross at least two *boundaries*. When more than one SP must be traversed, then the number of boundaries crossed increases. Traffic coming into the SP's network is typically marked with specific tags for VLAN and priority. The SP has needs regarding routing traffic as well, which may require

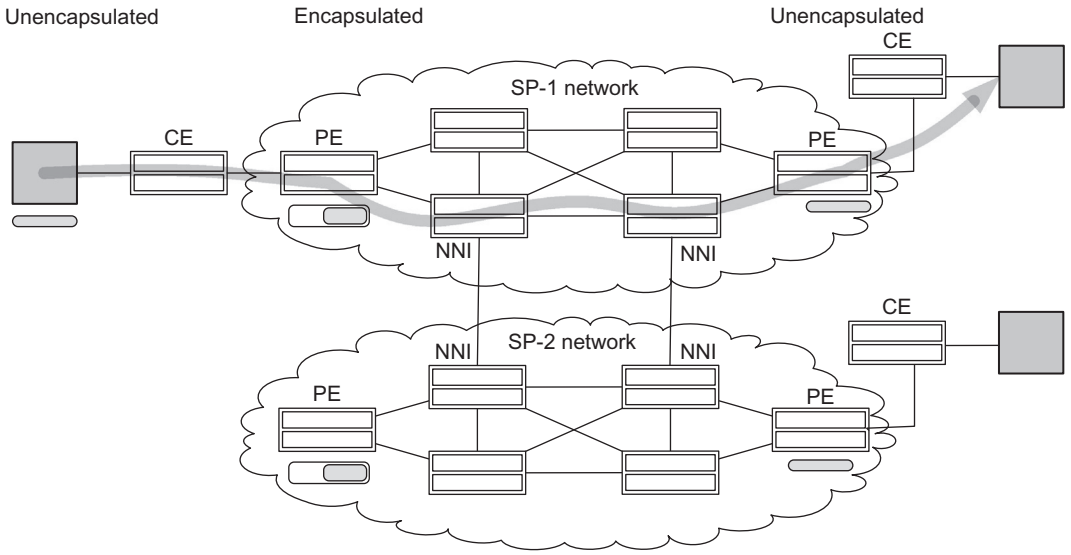

FIG. 9.4

Service provider environment.

the packet to be tagged again. Furthermore, the routing mechanism may be different within the SP network. For example, they may use MPLS or VLAN tagging for internal routing. When these additional tagging mechanisms are used, this entails another layer of encapsulation of the customer data packet. The boundaries that traffic must cross are often referred to as *Customer Edge* (CE) and *Provider Edge* (PE). The *Network to Network Interface* (NNI) is the boundary between two SPs. Since the NNI is an important point for policy enforcement, it is important that policy be easily configurable at these boundaries. Technologies supporting SPs in this way must support these boundary-crossing requirements.

Fig. 9.4 shows a simplified version of a network with customers and a pair of SPs. The two endpoint hosts are attempting to communicate by passing through the CE and then PE. After traversing SP1's network, the packet will egress that network at the PE on the other side. After passing through the CE, the destination receives the packet.

The figure shows a packet traversing the network from the endpoint on the left to the endpoint on the right (see arrow in figure). The original packet as it emanates from the source device is unencapsulated. The packet may acquire a VLAN tag as it passes through the CE (probably provided by the ISP). When the packet passes through the PE, it may be encapsulated using a technology such as PBB that we discussed in Section 5.6.7, or is tagged with another VLAN or MPLS tag. When the packet exits the provider network and passes through the other PE on the right, it is correspondingly either decapsulated or the tag is popped and it is then passed into the destination CE network.

Fig. 9.4 additionally depicts the NNI between SP1 and SP2. If the customer packets were directed to a destination on the SP2 network, they would traverse such an NNI boundary. Policies related to business agreements between the SP1 and SP2 service providers would be enforced at that interface.

9.2.1 SDN APPLIED TO SP AND CARRIER NETWORKS

A key focus of SPs when considering SDN is *monetization*. This refers to the ability to make or save money by using specific techniques and tools. SDN is promoted as a way for providers and carriers to monetize their investments in networking equipment by increasing efficiency, reducing the overhead of management, and rapidly adapting to changes in business policy and relationships.

Some of the ways in which SDN can help improve monetization for providers and carriers are (1) *Bandwidth management*, (2) *CAPEX and OPEX savings*, and (3) *Policy Enforcement at the PE and NNI boundaries*.

SDN exhibits great agility and the ability to maximize the use of existing links using TE and centralized, network-wide awareness of state. The granular and pliable nature of SDN allows changes to be made easily and with improved ability to do so with minimal service interruption. This facilitates the profitable use of bandwidth as well as the ability to adapt the network to changing requirements related to customer needs and *Service-Level Agreements* (SLAs).

SDN can reduce costs in a couple of ways. First, there are CAPEX savings. The cost of white box SDN devices is appreciably lower than the cost for comparable non-SDN equipment. This may be due to *bill of materials* (BOM) reduction as well as the simple fact that the white box vendors are accustomed to a lower-margin business model. The BOM reductions derive from savings such as reduced memory and CPU costs due to the removal from the device of so much CPU- and memory-intensive control software. Second, there are reduced OPEX costs, which come in the form of reduced administrative loads related to the management and configuration of the devices.

As described previously, packets will travel from the customer network across the PE into the provider's network, exiting at the remote PE. OpenFlow supports multiple methods of encapsulating traffic as is required in these circumstances. OpenFlow 1.1 supports the pushing and popping of MPLS and VLAN tags. OpenFlow 1.3 added support for PBB encapsulation. This makes it possible to conform to standard interfaces with Open SDN technology. Additionally, Open SDN provides a versatile mechanism for policy enforcement on the aggregated traffic which traverses NNI boundaries. Since these policies are often tied to fluid business relationships between the SPs, it is essential that the technology supporting policy enforcement be easily manipulated itself. With the additional bandwidth, cost-cutting, visibility, and policy enforcement advantages of SDN, this makes an SDN solution even more compelling.

9.2.2 EXAMPLE: MPLS-TE AND MPLS VPNs

A research project at Stanford [3] demonstrated MPLS-TE using OpenFlow. The authors stated that "while the MPLS data plane is fairly simple, the control planes associated with MPLS-TE and MPLS-VPM are complicated. For instance, in a typical traffic engineered MPLS network, one needs to run OSPF, LDP, RSVP-TE, I-BGP and MP-BGP." They argue that the control plane for a traditional MPLS network that determines routes using traffic information is unnecessarily complex. The amount and complexity of control signaling in the traditional approach is such that when there are frequent changes in the network, there are so many control packets exchanged that some can readily be lost. This results in a cascade of instability in the MPLS-TE control plane. The authors demonstrate that it is simple to build an equivalent network using OpenFlow to push and pop MPLS tags according to the topology and traffic statistics available at the NOX OpenFlow controller. In addition to providing a more stable

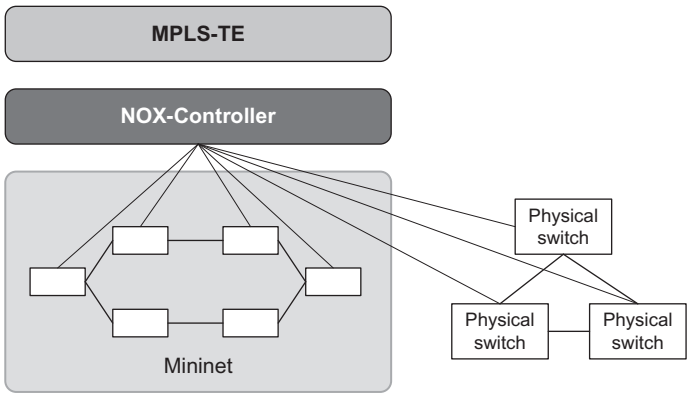

FIG. 9.5

Service provider and SDN: MPLS.

(Reproduced with permission from Sharafat A, Das S, Parulkar G, McKeown N. MPLS-TE and MPLS VPNs with OpenFlow. In: SIGCOMM'11. Toronto; 2011.)

and predictable alternative to the traditional approach, the OpenFlow-based solution was able to be implemented in less than 2000 lines of code, which is far more simple than the current systems.

Fig. 9.5 shows the architecture of the research project, replacing many routing and TE protocols on the devices with an MPLS-TE application running on the OpenFlow controller, setting flows consistent with the connectivity and QoS requirements. The project demonstrated this both on the Mininet network simulator as well as on physical switches. This OpenFlow-based MPLS solution obviates the need for a number of other protocols and functions that would otherwise have had to be implemented by the individual devices.

MPLS-TE remains an active area of research for the application of SDN. For example, Ericsson has published related research in using OpenFlow for MPLS-TE in [4].

DISCUSSION QUESTION

We indicate that the solution illustrated in Fig. 9.5 replaces many routing and TE protocols that would be needed to implement this solution in legacy networks. Name five examples.

9.2.3 EXAMPLE: CLOUD BURSTING WITH SERVICE PROVIDERS

Cloud bursting allows an SP to expand an enterprise's private cloud (data center) capacity on demand by dynamically allocating the compute and storage resources in the SP's data center to that enterprise. This additionally entails allocating the network resources to allow the free flow of data between the original, private cloud, and its dynamic extension in the SP. In [5], the author proposes to accomplish this cloud bursting via an SDN network. Controllers in the enterprise's private cloud make requests

to the SP's SDN controller for the needed facilities, and the SP SDN controller allocates the needed networking, compute and storage components to that enterprise. This model facilitates a major business opportunity for SPs. Verizon, in collaboration with HP and Intel, describes an SDN-based proof of concept project [6,7] to implement a solution for cloud bursting. In this PoC, an Intel private cloud in Oregon is dynamically augmented by the capacity in a Verizon public cloud in Massachusetts. Without manual intervention, network bandwidth is increased to handle the surge in data between the two data centers. The security that Intel requires on the Verizon VMs can be implemented without the need for physical appliances, using SDN techniques such as those that we discuss in Chapter 10. This PoC demonstrates a method where an SP can obtain improved CAPEX *Return on Investment* (ROI), lower operating expenses and improved business agility through the use of SDN.

9.3 CAMPUS NETWORKS

Campus networks are a collection of LANs in a concentrated geographical area. Usually the networking equipment and communications links belong to the owner of the campus. This may be a university, a private enterprise or a government office, among other entities. Campus end-users can connect through wireless access points (APs) or through wired links. They can connect using desktop computers, laptop computers, shared computers, or mobile devices, such as tablets and smartphones. The devices with which they connect to the network may be owned by their organization or by individuals. Furthermore, those individually owned devices may be running some form of access software from the IT department or the devices may be completely independent.

There are a number of networking requirements that pertain specifically to campus networks. These include (1) *differentiated levels of access*, (2) *bring your own device* (BYOD), (3) *access control and security*, (4) *service discovery*, and (5) *end-user firewalls*.

Different types of users in the campus will require different levels of access. Day guests should have access to a limited set of services, such as the Internet. More permanent guests may obtain access to more services. Employees should receive access based on the category into which they fall, such as executives or IT staff. These differentiated levels of access can be in the form of access control (i.e., what they can and cannot have access to) as well as their quality of service, such as traffic prioritization and bandwidth limits.

BYOD is a phenomenon that has arisen from the exponential increase in functionality available in smartphones and tablet computers. Campus network users want to access the network with the devices they are familiar with, rather than with a campus-issued device. In addition to mobile devices, some individuals may prefer Apple or Linux laptops over the corporate-issued Windows systems.

In years past, access to networks was based on physical proximity. If you were in the building and could plug your device into the network, then you were granted access. With the popularity of wireless connectivity and heightened security, it is now more common for employees and guests alike to have to overcome some security hurdle in order to be granted access. That security may be in a more secure form, such as IEEE 802.1X [8], or it may be some more limited form of security, such as authentication based on MAC address or a captive portal web login.

Campus end-users want access to services such as printers or file shares, or perhaps even TVs. Service discovery makes this possible through simple interfaces and automatic discovery of devices

and systems which provide these services. In order to achieve this level of simplicity, however, there is a cost in terms of network traffic load associated with protocols providing these services.

One of the dangers of campus networks is the possibility of infected devices introducing unwanted threats to the network. This threat is magnified by the presence of BYOD devices on the network. These may take the form of malicious applications like port scanners which probe the network looking for vulnerable devices. End-user firewalls are needed to protect against such threats.

9.3.1 SDN ON CAMPUS: APPLICATION OF POLICY

At the beginning of this chapter we described the value of centralized policy management and deployment. Campus networks are one area where this capability is readily realized. SDN's flexibility to manipulate flow rules based on policy definitions makes it well suited to the application of policy in the campus. The general idea of policy application is as follows:

- The user connects to the network and attempts to send traffic into the network.
- No policy is in place for this user, so the user's initial packets are forwarded to the controller. At this point, the user has either no access or only limited access to the network.
- The controller consults the policy database to determine the appropriate prioritization and access rights for this user.
- The controller downloads the appropriate flow rules for this user.
- The user now has access which is appropriate to the group to which he or she belongs, as well as other inputs such as the user's location, time of day, etc.

This system provides the differentiated levels of access that are needed in campus networks.

Clearly, as we explained in Section 3.2.2, this functionality is similar to what *Network Access Control* (NAC) products provide today. However, this SDN-based application can be simple, open, software-based, and flexible in that policies can be expressed in the fine-grained and standard language of the flow. We discuss programming an SDN NAC application in greater detail in Section 12.11.

Note that this type of access control using SDN should be applied at the edge of the network, where the number of users and thus, the number of flow entries will not exceed the limits of the actual hardware tables of the edge networking device. We introduced the topic of scaling the number of flow entries in Section 4.3.5. After the edge, the next layer of network concentration is called the *distribution* or *aggregation* layer. Application of policy at this layer of the campus network would take another form. At this level of the network, it is not appropriate or even practical to apply end-user rules which should be applied at the edge, as described previously. Rather, SDN policy in these devices should be related to classes of traffic or traffic that has been aggregated in some other way, such as a tunnel or MPLS LSP.

Flows can also be established for various types of traffic that set priorities appropriately for each traffic type. Examples of different traffic types are HTTP versus email. For example, HTTP traffic might be set at a lower priority during office hours, assuming that the organization's work-based traffic was not HTTP-based.

9.3.2 SDN ON CAMPUS: DEVICE AND USER SECURITY

Technological trends such as BYOD, access control and security can also be addressed by SDN technology. For example, one of the requirements for registering users' BYOD systems and guests

involves the use of a *captive portal*. This is the mechanism by which a user's browser request gets redirected to another destination website. This other website can be for the purpose of device registration or guest access. Captive portals are traditionally implemented by encoding the redirection logic into the switch firmware or by in-line appliances. To implement this in a pre-SDN environment could entail upgrading to switches that had this capability, implying more complex devices, or by installing specialized in-line appliances, adding to network complexity. Configuring which users need to be authenticated via the captive portal would entail configuring all these devices where a user could enter the network. Below we describe a simpler SDN solution for a captive portal solution:

- The user connects to the network and attempts to establish network connectivity.
- No access rules are in place for this user, so the SDN controller is notified.
- The SDN controller programs flows in the edge device which will cause the user's HTTP traffic to be redirected to a captive portal.
- The user is redirected to the captive portal and engages in the appropriate exchange to gain access to the network.
- Once the captive portal exchange is complete, the SDN controller is notified to set up the user's access rules appropriately.
- The user and/or BYOD device now has the appropriate level of access.

Fig. 9.6 shows an SDN-based captive portal-based application. The network edge device is initially programmed to route ARP, DNS, and DHCP requests to the appropriate server. In the figure, the end-user connects to the network and makes a DHCP request to obtain an IP address. When the

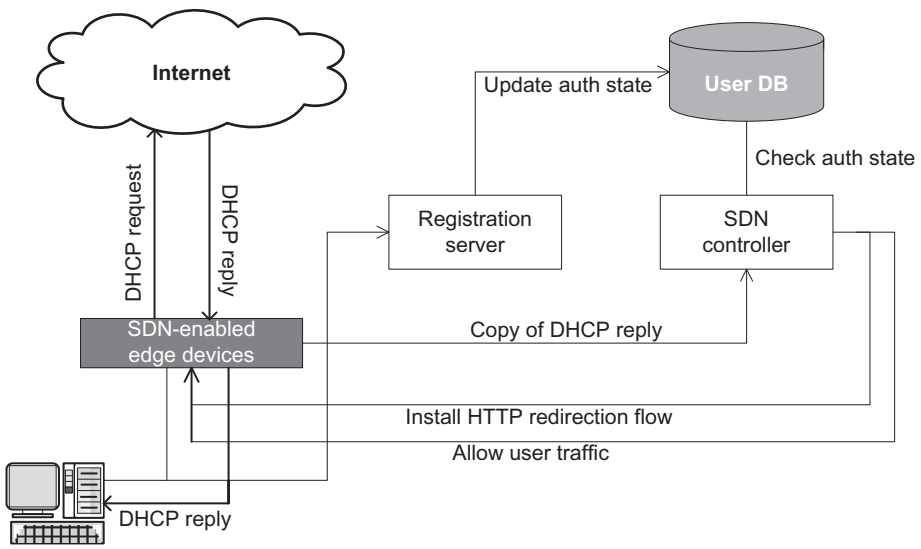

FIG. 9.6

NAC captive portal application.

DHCP reply is returned to the user, a copy is sent to the SDN controller. Using the end-user MAC address as a lookup key, the controller consults the database of users. If the user device is currently registered, it is allowed into the network. If it is not, then OpenFlow rules are programmed to forward that user's HTTP traffic to the controller. When the unauthenticated user's HTTP traffic is received at the controller, that web session is redirected to the captive portal web server. After completing the user authentication or device registration, the controller updates the user's flow(s) so that the packets will be allowed into the network. Note that this is also the appropriate time to configure rules related to the user's policy which can include levels of access and priority, among other items.

This type of functionality, like policy, is appropriately applied at the edge of the network. The further into the network this type of application is attempted, the more problematic it becomes, with flow table overflow becoming a real issue.

HP currently offers a commercial SDN application called Sentinel [9], which provides secure BYOD access. The approach of securing the network from end-user infection by installing anti-malware software on end-user devices is not viable in the BYOD case since they are not under the control of the network administration. The Sentinel SDN application provides a solution by turning the entire network infrastructure into a security-enforcement device using mechanisms that are hardened and more sophisticated versions of the basic methods for campus security we described previously. Sentinel is an SDN application that works with HP's TippingPoint threat database to secure and apply policy to the intranet as follows:

1. Identify *botnets* on the intranet and neutralize them by blocking DNS requests and IP connections to their *botnet masters* in the Internet.
2. Identify and prevent users or machines from accessing infected or prohibited sites on the Internet by monitoring and blocking DNS requests and IP connections (see Section 9.3.3).
3. Identify and quarantine infected machines on the intranet so that they can be remediated.

While traditional NAC solutions provide similar functionality, as an SDN application Sentinel can be deployed without installing additional hardware and it can provide protection at the edge of the network. Sentinel is a sophisticated example of the *blacklist* technology discussed in the next section.

9.3.3 SDN ON CAMPUS: TRAFFIC SUPPRESSION

Having flow table rules at the edge of the network carries the additional benefit of being able to suppress unwanted traffic. That unwanted traffic could be benign, such as service discovery multicasts, or it could be malicious, such as infected devices bringing viruses into the network. With flow tables at the edge of the network, service discovery traffic can be captured by the edge device and forwarded to the controller. The controller can then keep a repository of services and service requestors, which can be used to satisfy requests without having to forward all those multicasts upstream to flood the possibly overloaded network.

The ability to set policy and security for end-users with highly granular flow-based mechanisms facilitates effective per-user firewalls. Only certain types of traffic (e.g., traffic destined for certain UDP or TCP ports) will be allowed to pass the edge device. All other types of traffic will be dropped at the very edge of the network. This is an effective means of blocking certain classes of malicious traffic. Note that the implementation of per-user firewalls requires that the controller know

who the user is. If an NAC solution is running in the SDN controller, then this is not a problem. If the NAC solution, such as RADIUS, is implemented outside of SDN, then there is a challenge as the relationship between that external NAC solution and OpenFlow is neither standardized nor well-understood.

One facet of a per-user firewall is an application known as a blacklist. Blacklist technology blocks attempts by end-users to access known malicious or harmful hostnames and IP addresses. Traditional blacklist solutions rely on in-line appliances which *snoop* all traffic and attempt to trap and block attempts to reach these bad destinations. SDN is able to implement a blacklist solution without the insertion of additional appliances into the network. This provides both CAPEX savings due to less network equipment and OPEX savings due to a network that is easier to administer. An SDN application for blocking attempts to reach specific hostnames entails setting up flow table rules in edge devices to capture DNS requests and sending them to the controller. The SDN controller consults a database of undesirable hostnames, and would block the DNS request from being sent out when a blacklisted hostname is encountered.

Fig. 9.7 shows the simple manner in which a DNS blacklist could be implemented. The controller sets flows in the edge devices directing them to forward all DNS requests to the controller. When DNS requests arrive at the controller, it consults a local or remote database of known malicious sites. If the result is that the hostname is clean, then the controller returns the DNS request with instructions to the edge device to forward the packet as it normally would. If the hostname is deemed to be unsafe, the controller instructs the edge device to drop the packet, denying the user access to that host.

A savvy user can circumvent this first blacklist application if he/she knows the IP address of the destination host. By specifying the IP address directly instead of the hostname, no DNS request is sent. Fig. 9.8 shows a simple SDN implementation for this variant of the blacklist problem. In this second

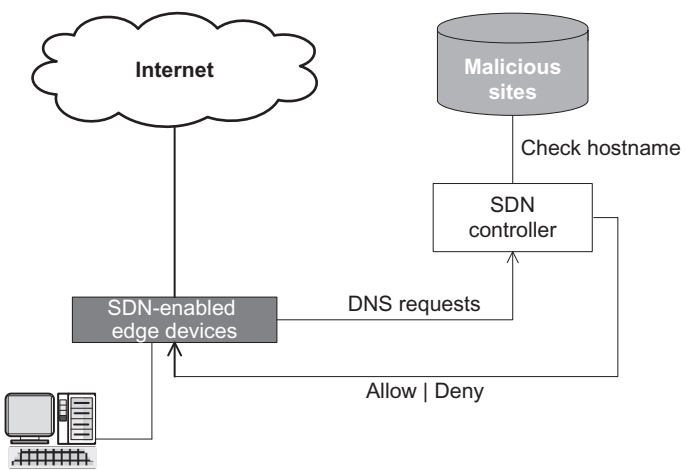

FIG. 9.7

DNS blacklist application.

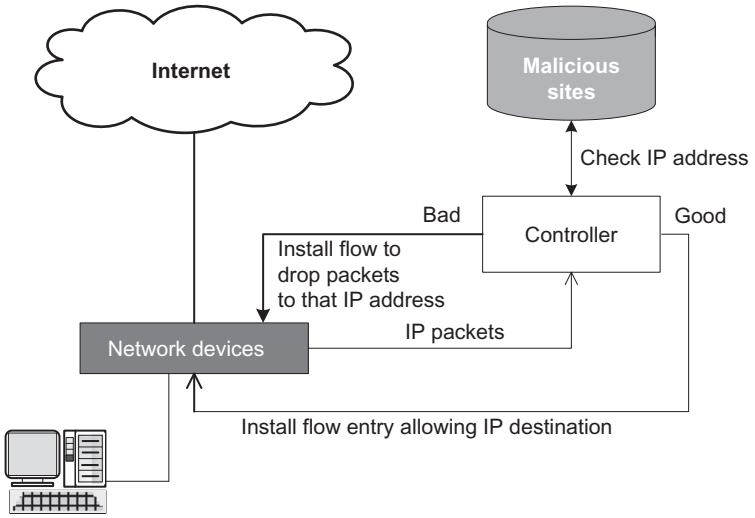

FIG. 9.8

IP address blacklist application.

solution, packets with destination IP addresses of unknown virtue are forwarded to the controller, which inspects the packets and either allows the IP address (so that packets to and from that host will now automatically be allowed) or else the device is programmed to drop this and future packets sent to that bad address. If the packet is allowed, the application will install a transient flow allowing requests to that destination IP address.

Note with this latter solution application design is critical. If destination IP address *allow* rules have aging timers which are too short, then they will age out in between requests and, consequently, every time a destination is visited, there will be delay and overhead associated with verifying and allowing the packets to that destination. If those rules have aging timers which are too long, then the device runs the risk of being overloaded with flow entries, possibly even exceeding the maximum size of the flow table.

We provide a detailed analysis with accompanying source code for an Open SDN blacklist application in Section 12.6.

Blacklist is really a simple host-level firewall. HP's Sentinel application described earlier provides a blacklist feature as a component of its commercial SDN security application.

DISCUSSION QUESTION

Is the blacklist application depicted in Figs. 9.7 and 9.8 a reactive or proactive application?

9.4 HOSPITALITY NETWORKS

Hospitality networks are found in hotels, airports, coffee shops, and fast food restaurants. There is a fair degree of overlap between the user requirements in campus networks and those of hospitality networks. The primary end-user in a hospitality network is a guest who is using the network either through the largesse of the establishment or through the purchase of a certain amount of time. One class of the for-purchase situation is end-users who have purchased connectivity for a certain duration, such as a month or a year, and who are thus able to connect to the network without a direct financial exchange each time they connect.

The application of SDN for captive portals in the campus discussed in Section 9.3.2 applies similarly to hospitality networks. The difference is that in the case of hospitality networks the user is paying directly or indirectly for the access, so a corresponding form of identification will likely be involved in authenticating with the registration server. These could include providing hotel room or credit card information.

Hospitality networks frequently offer WiFi access. We describe a more general application of SDN to WiFi access in the next section on mobile networks.

9.5 MOBILE NETWORKS

Mobile networking vendors, such as AT&T, Verizon, and Sprint, compete for customers to attach to their networks. The customers use smartphones or tablets to connect using the available cellular service, whether that is 3G, 4G, LTE, or another cellular technology.

When mobile customers use traditional WiFi hotspots to connect to the Internet, those mobile vendors effectively lose control of their customers. This is because the users' traffic enters the Internet directly from the hotspot. Since this completely circumvents the mobile vendor's network, the vendor is not even aware of the volume of traffic that the user sends and receives, and certainly cannot enforce any policy on that connection. When their customers' traffic circumvents the mobile provider's network, the provider loses a revenue-generating opportunity. Nevertheless, since their cellular capacities are continually being stretched, from that perspective it is advantageous for the mobile vendors to offload traffic to WiFi networks when possible. Thus the mobile provider is interested in a solution that would allow their customers to access their networks via public WiFi hotspots *without their losing control of and visibility to their customers' traffic*. The owner of such hotspots may wish for multiple vendors to share the WiFi resource offered by the hotspot. The multitenant hotspot we describe here is somewhat analogous to network virtualization in the data center. Just as SDN shines in that data center environment, so can it play a pivotal role in implementing such multitenant hotspots?

9.5.1 SDN APPLIED TO MOBILE NETWORKS

Mobile vendors interested in gaining access to users who are attaching to the Internet via WiFi hotspots require a mechanism to control their users' traffic. Control, in this context, may simply mean being able to measure how much traffic that user generates. It may mean the application of some policy regarding

QoS. It may mean diverting the user traffic before it enters the public Internet and redirecting that traffic through their own network. SDN technology can play a role in such a scheme in the following ways:

- Captive portals
- Tunneling back to the mobile network
- Application of policy

We discussed captive portals and access control in Section 9.3.1. This functionality can be applied to mobile networks as well. This requires allowing users to register for access based on their mobile credentials. Once valid credentials are processed, the user is granted appropriate levels of access.

One of the mechanisms for capturing and managing mobile user traffic is through the establishment of tunnels from the user's location back to the mobile vendor's network. The tunnel would be established using one of several available tunneling mechanisms. By programming SDN flows appropriately, that user's traffic would be forwarded into a tunnel and diverted to the mobile vendor's network. Usage charges could be applied by the mobile provider. In addition to charging for this traffic, other user-specific policies could be enforced. Such policies could be applied at WiFi hotspots where the user attaches to the network. SDN-enabled access points can receive policy, either from the controller of the mobile vendor or from a controller on the premises.

As an example of the utilization of SDN and OpenFlow technology as it relates to the needs of mobile networks and their SPs, consider Fig. 9.9.

The example depicted in Fig. 9.9 shows the basic means by which SDN and OpenFlow can be used to grant SP-specific access and private or public access from mobile devices to the Internet. In Fig. 9.9 the customers on the left want to access the Internet and each set has a different SP though which they

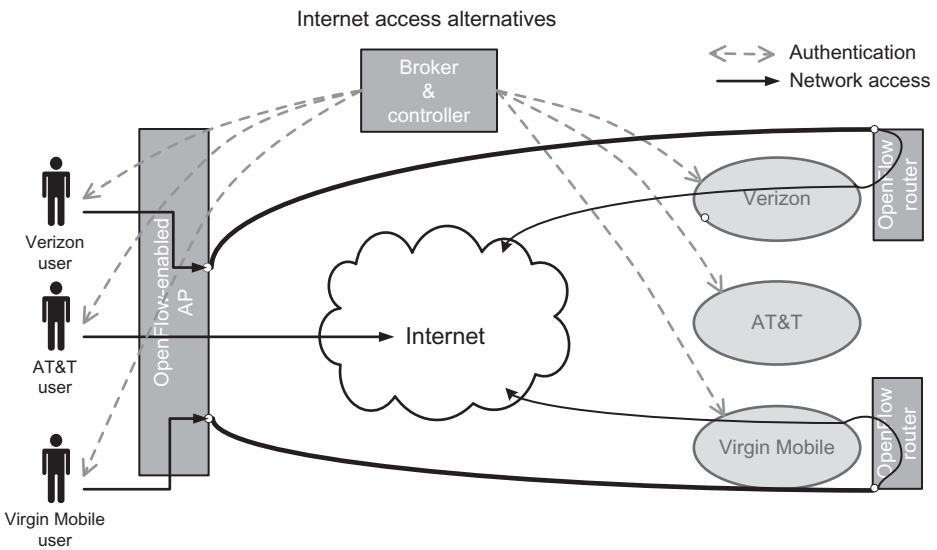

FIG. 9.9

Mobile service providers.

gain network access. Connecting through an OpenFlow-enabled wireless hotspot, they are directed through a broker who acts as an OpenFlow controller. Based on their SP, they are directed to the Internet in various ways, depending on the level of service and mechanism set up by the SP. In the example, AT&T users gain access directly to the Internet, while Verizon and Virgin Mobile users access the Internet by being directed by OpenFlow through a tunnel to the SP's network. Both tunnels start in the OpenFlow-enabled AP. One tunnel ends in an OpenFlow-enabled router belonging to Verizon, the other in an OpenFlow-enabled router belonging to Virgin Mobile. These two routers then redirect their respective customers' traffic back into the public Internet. In so doing, the customers gain the Internet access they desire and two of the mobile providers achieve the WiFi offload they need while maintaining visibility and control over their users' traffic. In order to facilitate such a system there is presumably a business relationship between the providers and the hotspot owner whereby the hotspot owner is compensated for allowing the three providers' users to access the hotspot. There would likely be a higher fee charged Verizon and Virgin Mobile for the service that allows them to retain control of their customers' traffic.

In 2013, the ONF's Wireless and Mobile Working Group published a number of OpenFlow-based use cases. These included:

- Flexible and Scalable Packet Core
- Dynamic Resource Management for Wireless Backhaul
- Mobile Traffic Management
- Management of Secured Flows in LTE
- Media-Independent Handover
- Energy Efficiency in Mobile Backhaul Networks
- Security and Backhaul Optimization
- Unified Equipment Management and Control
- Network-Based Mobility Management
- SDN-Based Mobility Management in LTE
- Unified Access Network for Enterprise and Large Campus

These use cases are very detailed and specific applications relevant to mobile operators. In a number of cases, implementing them would require extensions to OpenFlow. The mobile use cases listed above as well as others are described in [10].

DISCUSSION QUESTION

In the hypothetical example shown in Fig. 9.9, what potential benefits does AT&T forego by not tunneling its users' traffic back through the AT&T network?

9.6 OPTICAL NETWORKS

An *Optical Transport Network* (OTN) is an interconnection of optical switches and optical fiber links. The optical switches are layer one devices. They transmit bits using various encoding and multiplexing techniques. The fact that such optical networks transmit data over a lightwave-based *channel* as opposed

to treating each packet as an individually route-able entity lends itself naturally to the SDN concept of a flow. In the past, data traffic was transported over optical fiber using protocols such as *Synchronous Optical Networking* (SONET) and *Synchronous Digital Hierarchy* (SDH). More recently, however, OTN has become a replacement for those technologies. Some companies involved with both OTN and SDN are Ciena, Cyan (now acquired by Ciena), and Infinera. Some vendors are creating optical devices tailored for use in data centers. Calient, for example, uses optical technology for fast links between racks of servers.

9.6.1 SDN APPLIED TO OPTICAL NETWORKS

In multiple-use networks, there often arise certain traffic flows which make intense use of network bandwidth, sometimes to the point of starving other traffic flows. These are often called *elephant flows* due to their sizable nature. The flows are characterized by being of relatively long duration yet having a discrete beginning and end. They may occur due to bulk data transfer, such as backups that happen between the same two endpoints at regular intervals. These characteristics can make it possible to predict or schedule these flows. Once detected, the goal is to re-route that traffic onto some type of equipment, such as an all-optical network which is provisioned specifically for large data offloads such as this. OTNs are tailor-made for these huge volumes of packets traveling from one endpoint to another. Packet switches' ability to route such elephant flows at packet-level granularity is of no benefit, yet the burden an elephant flow places on the packet switching network's links is intense. Combining a packet switching network with an OTN into the kind of *hybrid* network shown in Fig. 9.10 provides an effective mechanism for handling elephant flows.

In Fig. 9.10 we depict normal endpoints (A1, A2) connected through *Top-of-Rack* (ToR) switches ToR-1 and ToR-2, communicating through the normal path, which traverses the packet-based network fabric. The other elephant devices (B1, B2) are transferring huge amounts of data from one to the other; hence, they have been shunted over to the optical circuit switch, thus protecting the bulk of the users from such a large consumer of bandwidth.

The mechanism for this shunting or *offload* entails the following steps:

1. The elephant flow is detected between endpoints in the network. Note that, depending on the flow, detecting the presence of an elephant flow is itself a difficult problem. Simply observing a sudden surge in the data flow between two endpoints in no way serves to predict the longevity of that flow. If the flow is going to end in 500 ms, then this is not an elephant flow and we would not want to incur any additional overhead to set up special processing for it. This is not trivial to know or predict. Normally, some additional contextual information is required to know that an elephant flow has begun. An obvious example is the case of a regularly scheduled backup that occurs across the network. This topic is beyond the scope of this work, and we direct the interested reader to [11,12].
2. The information regarding the endpoints' attaching network devices is noted, including the uplinks (U1, U2) which pass traffic from the endpoints up into the overloaded network core.
3. The SDN controller program flows in ToR-1 and ToR-2 to forward traffic to and from the endpoints (B1, B2) out an appropriate offload port (O1, O2), rather than the normal port (U1, U2). Those offload ports are connected to the optical offload fabric.

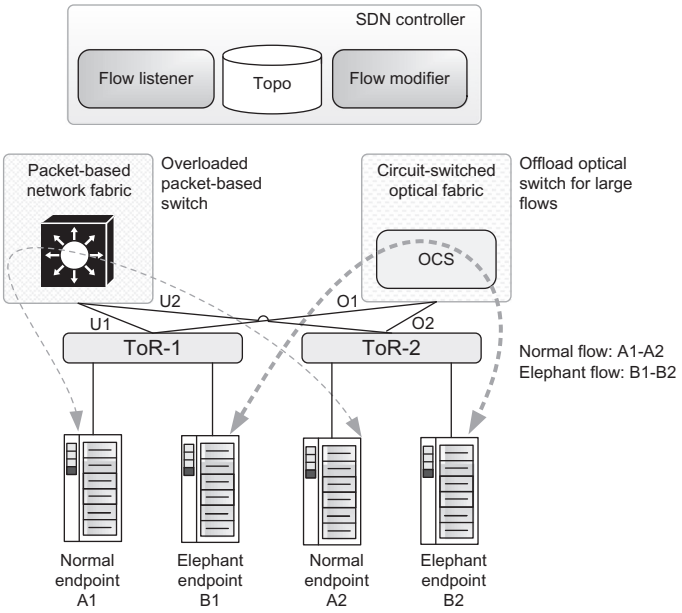

FIG. 9.10

Optical offload application overview.

4. The SDN controller programs flows on the SDN-enabled optical offload fabric to patch traffic between B1 and B2 on the two offload links (O1, O2). At this point, the re-routing is complete and the offload path has been established between the two endpoints through the OTN.
5. The elephant flow eventually returns to normal and the offload path is removed from both connecting network devices and from the optical offload device. Subsequent packets from B1 to B2 traverse the packet-based network fabric.

We discuss details of programming an offload application in Section 12.10.

9.6.2 EXAMPLE: FUJITSU'S USE OF SDN IN OPTICAL NETWORKS

Fujitsu has been an early adopter of SDN and focused on optical networks. One of their first forays into optical SDN was to leverage the technology to accelerate network storage access [13]. The SDN controller observes the storage access (storage flow) in the network on *Fiber Channel over Ethernet* (FCoE) and performs flow manipulation. Fujitsu separated the storage flow detection and storage flow manipulation from the functions needed for FCoE data relays. They then created a converged fabric switch with a software interface to the centralized controller. Fujitsu reported that this SDN implementation resulted in a twofold increase in throughput.

Other Fujitsu optical SDN efforts are targeted toward OTN. Fujitsu, a founding partner of the *Open Network Operating System* (ONOS) community, recently demonstrated a use case of packet-over-optical transport [14]. The ONOS *Cardinal* release was used to demonstrate this

packet-over-optical use case, which is central to the application of SDN to OTN. With Cardinal, Fujitsu was able to leverage new southbound plugins to develop *transaction language 1* (TL1) interfaces from the ONOS controller to the *FLASHWAVE 9500* Packet Optical Networking Platform. These interfaces allowed the ONOS controller to provide *Dense Wavelength Division Multiplexing* (DWDM) services such as on-demand bandwidth, bandwidth calendaring, and multilayer optimization.

Through these SDN efforts, Fujitsu has expanded its *Virtuora* SDN/NFV platform [15]. This platform has been built on the OpenDaylight (ODL) controller, but Fujitsu has purposefully ensured that its platform is portable to other controllers. For instance, they note that the Virtuora platform is easily portable to ONOS. The Virtuora NC 3.0 SDN framework was recently launched and it is based on ODL [16]. This framework has southbound interfaces that support TL1 and NETCONF. Based on Fujitsu's optical work with ONOS, the TL1 interface can support the DWDM services previously mentioned.

DISCUSSION QUESTION

If you were asked to map the Fujitsu FLASHWAVE 9500 onto the hypothetical example in Fig. 9.10, which functional box would it be?

9.7 SDN VS P2P/OVERLAY NETWORKS

At a conceptual level *P2P/Overlay networks* resemble the overlay networks presented in detail throughout this book. Just as the data center virtual networks are overlaid on a physical infrastructure the details of which are masked from the virtual network, so also is the P2P/Overlay network overlaid over the public Internet without concern or knowledge of the underlying network topology. Such networks are comprised of a usually ad hoc collection of host computers in diverse locations owned and operated by separate entities, with each host connected to the Internet in either permanent or temporary fashion. The peer-to-peer (and hence the name P2P) connections between these hosts are usually TCP connections. Thus all of the hosts in the network are directly connected. Napster is the earliest well-known example of a PTP/Overlay network.

We introduce P2P/Overlay networks here primarily to distinguish them from the overlay networks we describe in SDN via Hypervisor-based Overlays. Although the nature of the overlay itself is different, it is interesting to consider where there might be some overlap between the two technologies. Just as scaling SDN will ultimately require coordination of controllers across controlled environments, there is a need for coordination between P2P/Overlay devices. SDN helps move up the abstraction of network control, but there will never be a single controller for the entire universe, and thus there will still need to be coordination between controllers and controlled environments. P2P/Overlay peers also must coordinate among each other, but they do so in a topology independent way by creating an overlay network. A big distinction is that Open SDN can also control the underlay network. The only real parallel between these two technologies is that at some scaling point, they must coordinate and control in a distributed fashion. The layers at which this is applied are totally different, however.

9.8 CONCLUSION

Through examples, we have illustrated in this chapter and the preceding chapter that SDN can be applied to a very wide range of networking problems. We remind the reader of the points made in Section 4.1.3 that SDN provides a high-level abstraction for detailed and explicit programming of network behavior. It is this ease of programmability that allows SDN to address these varied use cases. One of the most exciting aspects of SDN is that the very flexibility that has rendered it capable of crisply addressing traditional network problems will also make it adaptable to solve yet-to-be conceived networking challenges. In Chapter 10 we examine a closely related technology, *Network Functions Virtualization* (NFV), and consider some of the use cases where NFV and SDN overlap.

REFERENCES

[1] Farrel A, Vasseur JP, Ash J. A path computation element (PCE)-based architecture. RFC 4655, Internet Engineering Task Force; 2006.
[2] Hoelzle U. OpenFlow@Google. Open networking summit. Santa Clara, CA: Google; 2012.
[3] Sharafat A, Das S, Parulkar G, McKeown N. MPLS-TE and MPLS VPNs with OpenFlow. In: SIGCOMM'11. Toronto; 2011.
[4] Green H, Kempf J, Thorelli S, Takacs A. MPLS OpenFlow and the split router architecture: a research approach. In: MPLS, 2010. Washington, DC: Ericsson Research; 2010.
[5] McDysan D. Cloud bursting use case. Internet draft. Internet Engineering Task Force; 2011.
[6] Verizon to demonstrate software defined networking principles with collaborative lab trials. Verizon Press Release; 2012. Retrieved from: http://newscenter2.verizon.com/press-releases/verizon/2012/verizon-to-demonstrate.html.
[7] Schooler R, Sen P. Transforming networks with NFV & SDN. Open networking summit. Santa Clara, CA: Intel/Verizon; 2013, p. 24–9.
[8] IEEE Standard for local and metropolitan area networks: port-based network access control. New York: IEEE; 2010. IEEE 802.1X-2010.
[9] Case study Ballarat grammar secures BYOD with HP Sentinel SDN. Hewlett-Packard Case Study; Retrieved from: http://h20195.www2.hp.com/v2/GetPDF.aspx/4AA4-7496ENW.pdf.
[10] Open Networking Foundation. Wireless and mobile working group charter application: use cases; 2013.
[11] Platenkamp R. Early identification of elephant flows in Internet traffic. In: 6th Twente Student Conference on IT, University of Twente. 2007. Retrieved from: http://referaat.cs.utwente.nl/conference/6/paper/6797/early-identification-of-elephant-flows-in-internet-traffic.pdf.
[12] Rivillo J, Hernandez J, Phillips I. On the efficient detection of elephant flows in aggregated network traffic. In: Networks and Control Group, Research School of Informatics. Loughborough University; Retrieved from: http://www.ee.ucl.ac.uk/lcs/previous/LCS2005/49.pdf.
[13] Fujitsu develops SDN technology to accelerate network storage access. Fujitsu Press Release; 2013. Retrieved from: http://www.fujitsu.com/global/about/resources/news/press-releases/2013/1209-01.html.
[14] Fujitsu successfully demonstrates ONOS interoperability. Fujitsu Press Release; 2015. Retrieved from: http://www.fujitsu.com/us/about/resources/news/press-releases/2015/fnc-20150615.html.
[15] Fujitsu product overview of SDN/NFV. Fujitsu Product Page; 2016. Retrieved from: http://www.fujitsu.com/us/products/network/technologies/software-defined-networking-and-network-functions-virtualization/index.html.
[16] Burt J. Fujitsu launches multilayer SDN suite for service providers. eWeek 2016; Retrieved from: http://www.eweek.com/networking/fujitsu-launches-multi-layer-sdn-suite-for-service-providers.html.