- **Vendor Protection**: Another argument is that using existing protocols and devices helps established vendors continue to dominate markets in which they are already the leader.

The emerging trends toward legacy protocols such as *Network Configuration Protocol* (NETCONF) and *Border Gateway Protocol* (BGP), and toward the SDN controllers that support these protocols are discussed in the sections that follow.

DISCUSSION QUESTION:

Discuss whether you believe that the influence on SDN by dominant networking vendors is a positive or negative influence on the advancement of networking technology.

7.1.2 NETWORK MANAGEMENT VERSUS SDN

One of the protocols that is being used for the application of SDN-based policies is NETCONF, which we examine in detail in the sections that follow. But the use of such a protocol, which was developed specifically as a means of improving the effectiveness of network management, raises the issue of where network management ends and SDN begins. Is this type of solution just an improved network management or is it really software defined networking?

In Chapter 6 we compared and contrasted three classes of SDN solutions: *Open SDN*, *SDN via APIs*, and *SDN via Overlays*. As that chapter illustrated, it is difficult to precisely circumscribe SDN. For the purposes of this discussion, since network management in general shares many of the same attributes as SDN (i.e., centralized control, network-wide views, network-wide policies), we consider such network management-based solutions to also fall under the larger SDN umbrella.

Fig. 7.1 shows the spectrum of solutions being promoted as SDN. On the right hand side of the picture is *reactive* OpenFlow, which involves packets getting forwarded to the controller via PACKET_IN messages. This type of SDN solution is the most dramatically different from traditional networking technologies, as highlighted in the figure. At the other end of the spectrum is network management, which is the most similar to what we see in traditional networks today. Between those extremes reside NETCONF, *Border Gateway Protocol Link State* (BGP-LS) and *Path Computation Element Protocol* (PCE-P), which we describe later in this chapter. Note that the term PCEP is frequently used as a substitute for PCE-P.

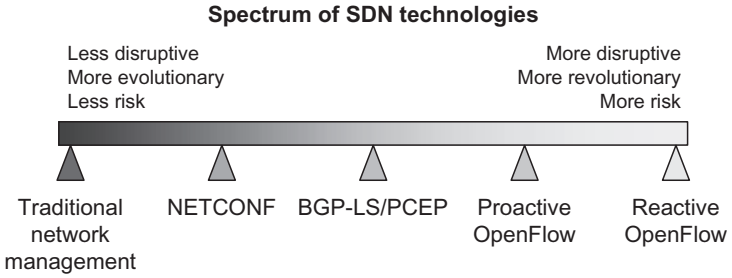

FIG. 7.1

SDN spectrum.

These SDN solutions all are considered valid approaches to SDN, based on the current and common use of the term, although they differ in their approach, their implementation and their suitability for various customer and domain needs. Each has benefits and limitations, which we examine next. The reader may wish to occasionally refer to Fig. 7.1 as our discussion below will generally follow the figure moving from left to right.

7.1.3 BENEFITS OF NETWORK MANAGEMENT-BASED SDN

We list here some benefits of SDN based on an evolved version of network management:

- **Least Disruptive**: This type of SDN is least disruptive to the current operation of the network because it operates with the current network infrastructure and capabilities of the networking staff.
- **Least Costly**: This type of SDN does not require new equipment, nor does it require a great deal of new training of IT personnel.
- **Least Risky**: This type of SDN introduces the least amount of risk into the network, since it runs on the same hardware and device software, with changes to networking behavior limited to what the SDN application is able to do through more traditional network management channels.

These advantages make network management-based SDN attractive to customers who want to eventually reach an SDN-based future through a gradual, more evolutionary path. Indeed, such solutions do move networking technology toward an SDN-based future.

7.1.4 LIMITATIONS OF NETWORK MANAGEMENT-BASED SDN

While there are benefits of network management-based SDN, there are limitations as well:

- **Limited Improvement**: Because it is restricted to using current devices, often without the capabilities for controlling forwarding and shaping of traffic, this type of SDN is limited in how much improvement it can provide in a network.
- **Limited Innovation**: Because this type of SDN is restricted to using currently configurable functionality in devices, it limits opportunities for truly innovative networking.
- **Limited Business Opportunity**: From an entrepreneurial standpoint, this type of SDN may not provide the opportunity to create disruptive and revolutionary new players in the networking device industry.

Customers with established networking environments may decide that the benefits of network management-based SDN outweigh the limitations, while others may opt for the more radical change offered by protocols such as OpenFlow.

DISCUSSION QUESTION:

Network management has been around for some time, and was never considered any type of "software defined networking." Now it is. Discuss whether you think that network management-based SDN should be considered "real" SDN or not, and defend your point of view.

7.2 ADDITIONAL SDN PROTOCOL MODELS

This book has predominantly focused on OpenFlow as the original and the most prominent SDN protocol being used in research, entrepreneurial efforts, and even in some established commercial environments (e.g., Google). However, current trends indicate that much SDN development today focuses on other protocols. We examine these protocols, controllers, and application development trends in the following sections.

7.2.1 USING EXISTING PROTOCOLS TO CREATE SDN SOLUTIONS

Assuming that one of the goals of emerging SDN solutions is to reduce risk and to make use of existing customer and vendor equipment, it is consistent that these solutions utilize existing protocols when feasible. Established protocols such as NETCONF, BGP, and *Multiprotocol Label Switching* (MPLS) are all potentially relevant here. These are mature protocols that are used in massively scaled production networks. Utilizing these protocols to implement SDN solutions makes sense for those interested in directing their SDN efforts along an evolutionary path utilizing existing technologies.

Fig. 7.2 shows a real-life example of existing protocols being used to create an SDN solution. The figure depicts some of the main components involved in the *OpenDaylight* (ODL) controller's BGP-LS/PCE-P plugin. Starting at the top of the figure, we see that there are three distinct protocols involved:

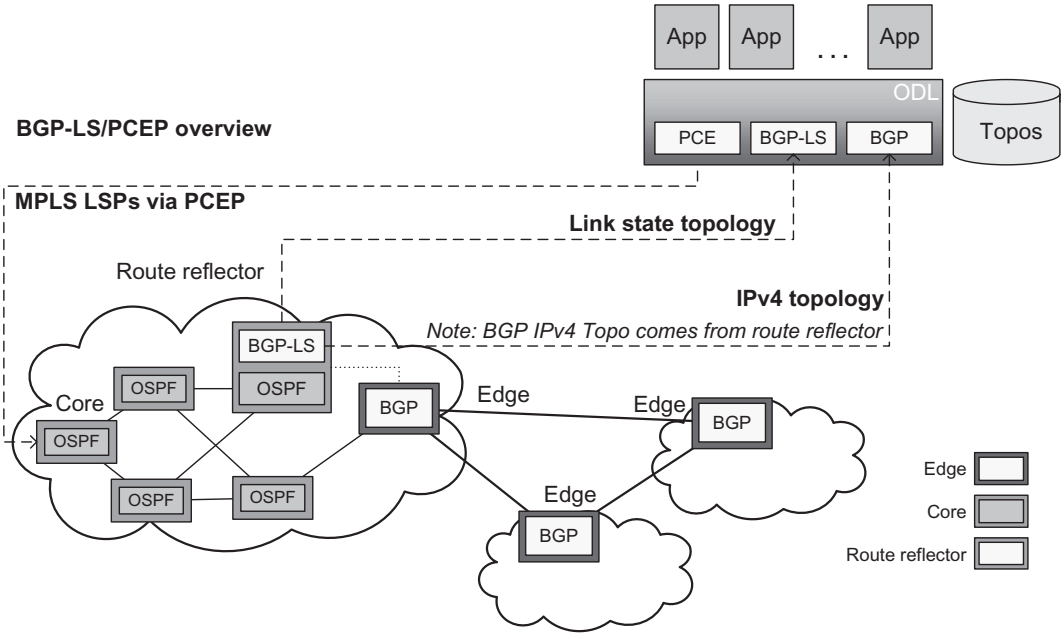

FIG. 7.2

BGP-LS/PCE-P overview.

- **BGP-LS**: The BGP-LS protocol is used by ODL to gather link state topology information from the routing protocols running in the clouds in the figure. This topology reflects routers and interconnecting links within the *Open Shortest Path First* (OSPF) or *Intermediate System to Intermediate System* (IS-IS) domains.
- **BGP**: The BGP protocol is used by ODL to gather IP *Exterior Gateway Protocol* (EGP) topology from the BGP routers connecting the clouds (domains) in the picture.
- **PCE-P**: The PCE-P protocol is used by ODL to configure MPLS *Label Switched Paths* (LSPs) for forwarding traffic across those networks.

The SDN solution in Fig. 7.2 will be discussed further in the following sections. The interested reader can find detailed information in [1]. In the next sections we will examine these protocols as well as NETCONF in order to understand their roles in SDN.

As we consider this use of existing protocols, a helpful perspective may be to look at the different control points shown in Fig. 7.3 that are managed and configured by the SDN application. These control points are *Config*, where general configuration is done, *Routing Information Base* (RIB), where routes (e.g., prefixes and next-hops) are set, and *Forwarding Information Base* (FIB), which is lower level and can be considered *flows*, where packet headers are matched and actions are taken.

The association between these control points and existing protocols is shown in Table 7.1. The table shows control points in general terms. NETCONF's role is for setting configuration parameters. BGP is

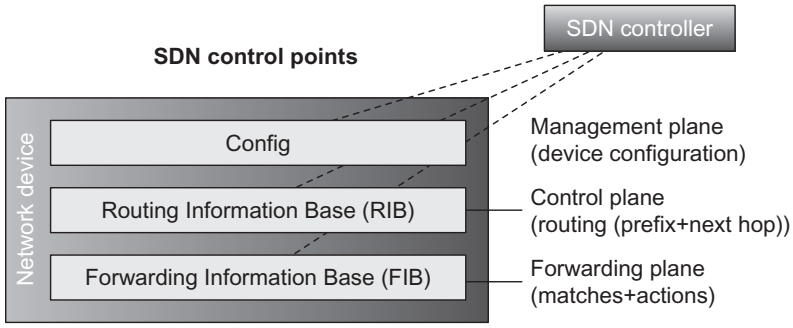

FIG. 7.3

SDN control points.

Table 7.1 Comparison of Existing Protocols for SDN

Protocol	Control Point	Details
NETCONF	Config	Interfaces, ACLs, Static routes
BGP-LS	-	Topology discovery is used to pass link-state IGP information about topology to ODL.
BGP	RIB	Topology discovery and setting RIB
PCE-P	MPLS	PCE to set MPLS LSPs. Used to transmit routing information from the PCE Server to the PCE Clients in the network.
BGP-FS	Flows	BGP-FlowSpec to set matches and actions

involved in setting RIB entries, and PCE-P is used for setting MPLS paths through the network. BGP-LS is used to gather topology information from the RIB. *BGP-FlowSpec* (BGP-FS) is employed to set matches and actions, similar to what is done with OpenFlow, using instead the BGP-FS functionality of the router. BGP-FS leverages the BGP *Route Reflection* infrastructure and can use BGP *Route Targets* to define which routers get which routes. Unlike OpenFlow, BGP-FS does not support layer 2 matches but only layer 3 and above.

7.2.2 USING THE NETCONF PROTOCOL FOR SDN

NETCONF is a protocol developed in an *Internet Engineering Task Force* (IETF) working group and became a standard in 2006, published in *Request for Comments* (RFC) 4741 [2] and later revised in 2011 and published in RFC 6241 [3]. The protocol was developed as a successor to the *Simple Network Management Protocol* (SNMP) and attempted to address some of SNMP's shortcomings. Some key attributes of NETCONF are:

- **Separation of configuration and state (operational) data.** Configuration data is set on the device to cause it to operate in a particular way. State (operational) data is set by the device as a result of dynamic changes on the device due to network events and activities.
- **Support for *Remote Procedure Call* (RPC)-like functionality.** Such functionality was not available in SNMP. With NETCONF, it is possible to invoke an operation on a device, passing parameters and receiving returned results, much like RPC calls in the programming paradigm.
- **Support for Notifications.** This capability is a general event mechanism, whereby the managed device can notify the management station of significant events. Within SNMP this concept is called a trap.
- **Support for transaction-based configurations.** This allows for the configuration of multiple devices to be initiated, but then rolled back in case of a failure at some point in the process.

NETCONF is a management protocol and as such it has the ability to configure only those capabilities which are exposed by the device. Fig. 7.4 illustrates the difference between a

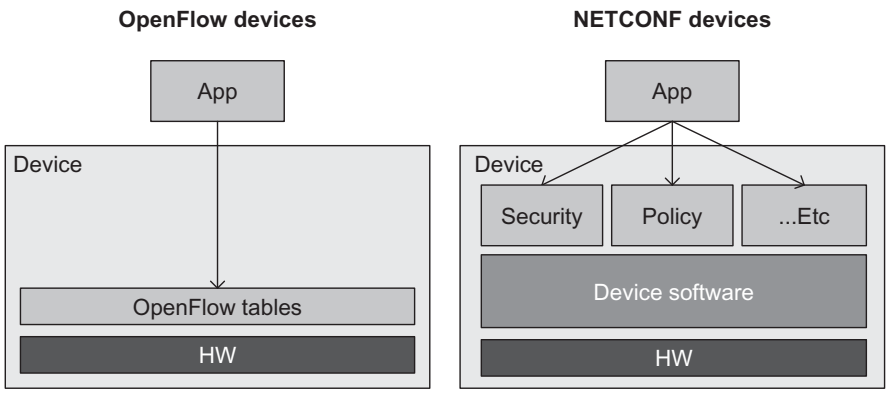

FIG. 7.4

NETCONF versus OpenFlow.

NETCONF-controlled and an OpenFlow-controlled device. We see in the figure that OpenFlow configures the lower levels of the networking device, that is, the ASIC containing the TCAM. This entails setting the matches and actions of the FIB.

NETCONF, on the other hand, performs traditional configuration of the device but via the NETCONF protocol rather than via the CLI or SNMP. The SDN application developer is limited by what the device exposes as configurable. If the device exposes NETCONF data models that allow for the configuration of *Access Control Lists* (ACLs), then the application can configure those. Similarly, if the device has data models for configuring *Quality of Service* (QoS) or static routes, then those will be possible as well.

The NETCONF programmer can learn which capabilities are exposed via the *Yet Another Next Generation data modeling language* (YANG) data models supported by the device.

NETCONF and YANG

NETCONF itself is a network management protocol, just as SNMP is a network management protocol. Such protocols become useful via the data models which convey information and requests to and from the device. With SNMP, the data models took the form of a *Management Information Base* (MIB), which we defined using *Structure of Management Information* (SMI). Contrasting the SNMP and NETCONF paradigms, SMI is analogous to YANG and the MIB is analogous to the YANG data model.

YANG provides a standardized way for devices to support and advertise their capabilities. One of the first operations that takes place between a NETCONF client on the controller and a NETCONF server running on the device is for the device to inform the client which data models are supported. This allows the SDN application running on the controller to know which operations are possible on each device. This granularity is key, since different devices will often vary in their capabilities. One of the current drawbacks of NETCONF and YANG is that different vendors often support different YANG models. This sometimes even occurs within different product families from the same vendor. Unlike the case of SNMP and standard MIBs (e.g., MIB-II, the Interfaces MIB, and the RMON MIB), there is currently no consistent set of YANG data models supported across the industry. It is currently necessary for applications to request and set data on different devices.

YANG models are still relatively new, and it is likely that standardized models will be defined in the near future. This will be facilitated by the fact that modern networking devices have better internal configuration schemas than in the early days of SNMP, so for most vendors it is relatively easy to auto-generate YANG data models that map onto those schemas. Note that in our discussions about NETCONF throughout this book we assume that YANG is used as its data modeling language.

NETCONF and RESTCONF

NETCONF uses the *Extensible Markup Language* (XML) to communicate with devices, which can be cumbersome. Fortunately, SDN controllers often support *REST-based NETCONF* (RESTCONF) as a mechanism for communicating between the controller and devices. RESTCONF works like a general REST API in that the application will use HTTP methods such as GET, POST, PUT, and DELETE in order to make NETCONF requests to the device.

As with normal REST communication, the URL specifies the specific resource that is being referenced in the request. The payload used for the request can be carried in either XML or *JavaScript Object Notation* (JSON).

Software developers with web programming backgrounds often find RESTCONF easier to work with than traditional use of NETCONF. This is due to the fact that REST APIs and JSON are generally more familiar to web developers than XML RPCs. Hence, using RESTCONF makes communication between the SDN controller and devices much simpler than it might be otherwise.

7.2.3 USING THE BGP PROTOCOL FOR SDN

Another protocol being promoted as a mechanism for SDN solutions is BGP. As we explained in Section 1.5.2, BGP is the EGP routing protocol used in the Internet. In addition to this traditional role, it is also used internally in some data centers. Consequently, the prospect of configuring BGP routes dynamically in a software defined manner is appealing. There are two major aspects of the BGP functionality currently used in ODL. These are:

- **IPv4 Topology**: The BGP plugin running inside ODL is implementing an actual BGP node, and as such it has access to topological information via the *Route Reflector* (RR). This information provides the topology between devices implementing the EGP, often referred to as the IPv4 topology. Fig. 7.5 shows an EGP network with routers supporting BGP, and an RR communicating topology information to the BGP node running inside the ODL controller. This information helps to provide the network-wide views characteristic of SDN solutions, and it can be used to dynamically configure intelligent routing paths throughout the network, via RIB configuration. This network-wide view is seen in Fig. 7.5 in the network topology to the right of the ODL controller. Note that while we specifically cite the IPv4 topology here, other topologies, such as the IPv6 topology, can be reported by the BGP plugin.

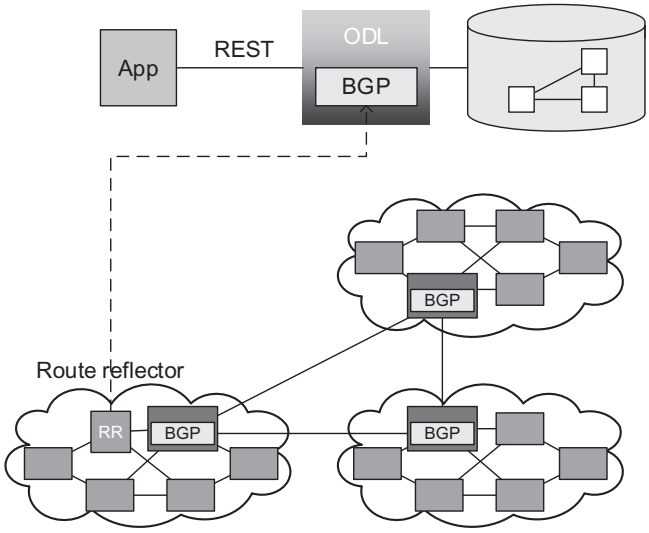

FIG. 7.5

SDN BGP topology.

- **RIB Configuration**: Within ODL there are APIs for creating a RIB application which can be used to inject routes into the network, based on the topology information, traffic statistics, congestion points to be avoided, as well as other possible relevant data.

A key aspect of this technique is that the ODL's controller's BGP plugin appears to the network as a normal BGP node. Note that the plugin does not advertise itself as a next hop for any of the routes. It will, however, effect changes on the adjacent nodes that will in turn propagate routing information throughout the network. In this way, an SDN application running on ODL can force RIB changes throughout the network, thereby achieving SDN-like agility and automatic network reconfiguration.

7.2.4 USING THE BGP-LS PROTOCOL FOR SDN

Figs. 7.6 and 7.7 depict the operation of the BGP-LS/PCE-P plugin on ODL.

- **BGP-LS** is used to pass link-state (OSPF or IS-IS) *Interior Gateway Protocol* (IGP) information about topology to ODL.
- **PCE-P** is used to transmit routing information from the PCE Server to the PCE clients in the network. A PCE client is also more simply known as a *Path Computation Client* (PCC).
- **MPLS** will be used to forward packets throughout the network, using the Label Switched Paths (LSPs) transmitted to head-end nodes via PCE-P.

Fig. 7.6 illustrates an IGP network of OSPF-supporting routers, sharing topology information with ODL. At a high level, BGP-LS is running on one of the OSPF (or another IGP) nodes in the network, and the IGP shares topology information with BGP-LS running on that node. That BGP-LS node in turn shares the topology information with the BGP-LS plugin running in ODL. That topology information is made available to the SDN application running on ODL, which can combine that knowledge with

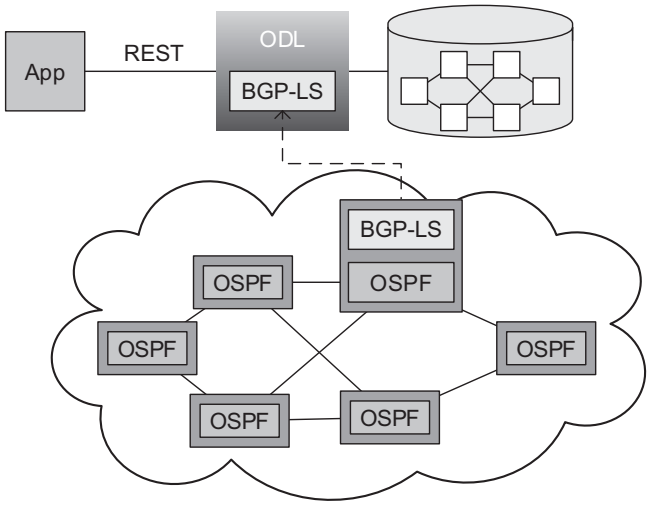

FIG. 7.6

SDN BGP-LS topology.

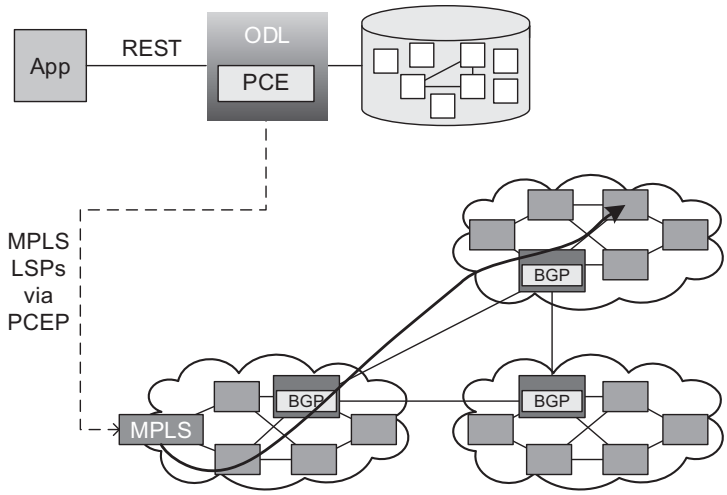

FIG. 7.7

SDN PCE-P and MPLS.

other knowledge about congestion, traffic, bandwidth, prioritization policies, and the like. This can be combined by the SDN application, which will determine optimal routing paths, and will communicate those MPLS paths to the PCCs in the network.

7.2.5 USING THE PCE-P PROTOCOL FOR SDN

PCE and its associated protocol PCE-P, have been in existence since roughly 2006 [4] and address the need to compute complex paths through IGP networks, as well as across *Autonomous System* (AS) boundaries via BGP. These paths are used in networks that support *MPLS Traffic Engineering* (MPLS-TE). The computation done by the PCE can be located in any compute node—in an MPLS head-end router, in the cloud, or on a dedicated server.

In Fig. 7.7, ODL (driven by an SDN application) sets MPLS *Label Switched Paths* (LSPs) using PCE-P. Communication is between the PCE server in ODL and the PCC on the MPLS router.

The PCC runs on the *head-end* of each LSP. Using these LSPs, the router is able to route traffic using MPLS through the network in an optimal manner. Using PCE-P in this fashion has advantages over a pre-SDN counterpart called *Constrained Shortest Path First* (CSPF). Like our PCE-P model described above, CSPF computes the LSPs but is limited to the topology of the IGP domains to which it belongs. Conversely, PCE-P can run across multiple IGP domains. Another advantage of PCE-P is that it can perform global optimization contrary to the CSPF model where each head-end router performs local optimization only.

7.2.6 USING THE MPLS PROTOCOL FOR SDN

MPLS will be used to forward packets throughout the network using the LSPs transmitted to head-end nodes via PCE-P. In the SDN solution described in the previous section the role of MPLS is to forward traffic according to the paths configured by the SDN application running on ODL. Configuration takes

place on the MPLS head-end router shown in Fig. 7.7. This router will receive the matching packets at the edge of the MPLS network, and packets will then be forwarded on the LSP that has been configured by PCE-P.

This solution does not require a new protocol such as OpenFlow in order to control the network's forwarding behavior, but rather uses these existing protocols (i.e., BGP, BGP-LS, PCE-P, and MPLS) to achieve intelligent routing of packets based on paths that have been configured by the centralized controller. This emerging solution holds promise for customers looking to utilize their existing infrastructure in a new, SDN-based manner.

DISCUSSION QUESTION:

Both the NETCONF and the BGP-LS/PCE-P southbound protocol plugins attempt to provide SDN capabilities using existing protocols. Which of these two seems to be more "SDN" and why? And what are some potential dangers in using BGP to set RIBs in the SDN controller?

7.3 ADDITIONAL SDN CONTROLLER MODELS

In this section we turn our attention to SDN controllers and controller technologies that have gained recent prominence in SDN. Hot topics in recent SDN controller innovation have included southbound protocol plugins, internal architectures, service provider solutions, scalability, and northbound interfaces to SDN applications.

As the dust begins to settle on SDN, two commercially viable controllers stand out: ODL and *Open Network Operating System* (ONOS). In the discussion that follows we are interested in the technologies used by these controllers. There are also business ramifications of the dominance that these controllers have asserted, but we defer our treatment of those until Chapter 14. Table 7.2 lists the primary areas of emphasis of each of these two controllers.

We caution the reader that the emphases denoted in Table 7.2 are just that—*emphases*. One should not infer that the other controller ignores these issues. For example, ONOS has projects for alternative southbound protocols like NETCONF and PCE-P, and ODL has projects and functionality related to service providers, scalability, and *intents*. We will see in the following sections that both controllers implement functionality in all of these areas.

7.3.1 CONTROLLERS WITH MULTIPLE SOUTHBOUND PLUGINS

Prior to the advent of ODL, almost every general-purpose SDN controller used the OpenFlow protocol as the sole southbound protocol. Other controllers were either not targeted at the open application development community or did not garner a sizeable community of developers. To gain traction,

Table 7.2 ODL and ONOS

Controller	Organization	Emphases
ODL	Linux Foundation	Multiple southbound, MD-SAL
ONOS	ON.Lab	Service providers, scalability, *intents*