White Paper

Making the Case for Open Source Controllers

Prepared by

Roz Roseboro
Senior Analyst, Heavy Reading
www.heavyreading.com

on behalf of

BROCADE
www.brocade.com

September 2014
Introduction

Telcos face a competitive landscape that is rapidly changing, forcing them to create and deliver new, innovative services. They are looking to leverage their existing infrastructure, but are challenged by the limitations of that infrastructure. This has led to increasing consideration of software-defined networking (SDN) architectures that will allow them to maximize their current network assets while also providing an environment that supports increased service velocity. SDN can bring benefits to both data center and telco wide-area network (WAN) environments.

The intelligence in SDN architectures will be provided by a new element: an SDN controller. The controller provides a logically centralized view of all network elements and plays a role similar to that of middleware. It directs policies southbound to network elements and connects to applications that are to the north.

Vendors are taking a variety of approaches to developing SDN controllers, ranging from proprietary/single vendor to open source/multi-vendor. The latter approach has numerous benefits in terms of time to market and robustness of code. An open source approach also eliminates vendor lock-in – a key criteria for telcos.

This white paper is structured as follows:

- **Section II** discusses the current state of the service provider networking environment. We look at the main drivers for deploying SDN, including enhanced service creation and delivery capabilities and better utilization of network assets.

- **Section III** discusses the role of SDN controllers and how they interact with the other elements of an open network platform, including the network OS to the south and virtualized infrastructure management and orchestration to the north. Particular attention will be paid to the importance of open application programming interfaces (APIs) for ensuring interoperability and manageability across multiple vendors’ equipment.

- **Section IV** looks at evaluation criteria for controllers, including the business benefits of a solution based on open source components versus procuring a solution based on proprietary components.

Competitive Pressure Driving New Approaches

**Need for Service Innovation & Velocity Is Forcing Change**

The pace of innovation in the telecom market has been increasing rapidly in recent years, spurred on by new mobile and content services, as well as new business models from over-the-top (OTT) and cloud players. Traffic – from video, mobile and cloud services – continues to grow much more quickly than revenues. Telcos are increasingly focused on applications as a way to generate revenues from more than bandwidth. However, much of the value from applications today is being siphoned off by non-operator companies, including device manufacturers and application developers.

As a result, telcos’ need to innovate and rapidly launch new services has become even more critical than in the past – and they need to do so while leveraging their considerable investments in their existing infrastructure. In many cases, however, they are limited by the capabilities of infrastructure that was designed for an era
where telcos were beholden to the development schedule of their vendors. The telcos are, therefore, challenged to migrate to a new architecture that will support more rapid innovation and greater agility and flexibility, and to do so without impacting service delivery.

SDN is being proposed as a way to address many of the challenges described above. The term SDN is used to describe several scenarios, but at its most basic level, in SDN architectures, the data plane is separated from the control plane. The administrator defines policies that manage flows across the network, which are executed by an SDN controller. The vision is that the interfaces between the controller and network elements, and between the controller and applications, will be standardized, and SDN will support multi-vendor environments, eliminating the vendor lock-in that has constrained operators in the past.

**Consider an Open Network Platform to Implement SDN**

In an earlier paper, *The Network as an Open Platform: Driving New Market Value for Telcos*, Heavy Reading made the case for an open network platform to implement SDN in the data center. The argument is just as valid when addressing the WAN environment. Telcos have been quite vocal in their quest to avoid vendor lock-in. Indeed, in a Heavy Reading survey, 46 percent of telco respondents indicated that SDN software must be based on open standards. Telcos don’t want to be beholden to any one vendor – unable to mix and match components as they see fit – or be dependent on vendor’s development cycle to get new features and functions.

*Figure 1* below shows the proposed architecture for an open network platform. The remainder of this report will focus on the SDN controller layer of this diagram.

---

**Figure 1: Open Network Platform – Proposed Architecture**

![Diagram of Open Network Platform: Proposed Architecture](source: Brocade and Heavy Reading)
The Role of SDN Controllers

In the report referenced above that introduced the concept of the open network platform, Heavy Reading defined the SDN controller as follows:

It provides a single point of attachment for applications that want to use the networking functionality embedded in the network OS. The controller also coordinates the delivery of the appropriate network functionality to those applications. In order to carry out this task, the network controller understands how and where networking functions are physically distributed across the platform, how to configure (“program”) them and how to connect data flows across them so that applications are supported by the right function(s) on demand. The network controller replaces manual provisioning of network hardware, enabling network functionality to be turned up faster at minimal cost.

In short, the SDN controller extends and scales the functionality of element management systems (EMS), network management systems (NMS) and network control plane through standard interfaces.

Provides Abstracted Network View to Applications to the North

The SDN controller provides the mechanism by which network details are abstracted and presented to applications in a way that makes it easier for them to govern network resources. Applications can tell the network what they need and those resources can be allocated dynamically—with no human intervention required. The interfaces between the controller and the applications have not yet been standardized, although most controllers support REST APIs.

While applications can be written by multiple parties, including independent software vendors, telcos are keen to be able to create their own innovative applications that will allow them to capture the value of those applications and differentiate themselves from the competition.

One of the major areas of emphasis in SDN controller development is the integration with orchestration tools, such as OpenStack. Indeed, cloud and infrastructure automation are two major drivers for SDN. OpenStack was created to enable virtual machine provisioning, storage and the most basic network functions. The controller layer handles the more complex network functions.

Manages the Network to the South

The SDN controller provides a mechanism for network applications to specify specialized policy and processes that are then delivered by network elements (routers, switches, L4-7 appliances, such as firewalls and load balancers—both physical and virtual versions on hypervisors). It generally supports protocols like SNMP to provide fault and performance management. OpenFlow is a standardized protocol to communicate between the controller and network elements, such as top-of-rack switches. NETCONF/YANG can also be used to manage network devices.

How SDN Controllers Help Telcos in the WAN

SDN enables better visibility and control across the data center and WAN environments, which helps optimize network resource utilization by supporting better traffic
engineering and bandwidth management. It can take into consideration the state of all the required network elements and provide the performance requested by the applications running above. The end result is that network resource utilization is optimized without any direct human intervention and service performance is maximized.

By using an SDN controller, telcos can automate provisioning across the WAN [in addition to the data center]. Being able to define configurations once and populate across multiple elements at the same time dramatically reduces the time it takes to configure the network, leading to lower opex. The value of the SDN controller to intermediate across different network elements is even more pronounced in a telco environment due to the variety of physical and virtual, older and newer, assets that need to be managed. SDN also provides better agility as the processes can be automated, which has the added benefit of reducing the risk of human errors that can occur when using a CLI to program network elements one at a time.

Importantly, SDN gives telcos the ability to develop innovative new services in software, breaking their dependence on their hardware vendors for new features. Because they can more easily and directly manage traffic flows, telcos can stitch together various services and create differentiated offerings. In addition, the ability to define and apply policies dynamically and granularly can help enhance the security of the services.

Choosing an SDN Controller Strategy

Multiple Approaches to SDN Controllers

In developing an SDN controller, vendors have numerous options. However, there are some common requirements that must be met in order to provide the most valuable functionality and usability. They include:

- Provide open APIs to which application developers can write
- Automate management of network infrastructure
- Support for multi-vendor environments
- Support for brownfield (i.e., non-virtualized environments) and greenfield deployments

Some vendors have chosen a proprietary/single-vendor approach. This is the most limiting option, as not all networking environments are single vendor. Still, this approach could be useful for certain use cases, including those that are dedicated to providing a limited set of end-user services. The vendors here will provide better mechanisms to more efficiently manage their own network elements, so telcos will see benefit from these solutions.

Others have taken a proprietary/multi-vendor approach. This is a more flexible option, but still limits flexibility as it is still controlled by a single vendor. This can lead to integration challenges, and ensuring interoperability on an on-going basis is an open issue. While the vendor may provide a plug in for open source elements like OpenDaylight (ODL), it may not necessarily have ODL code at its core, or may have a subset of ODL code and forked onto a different release path.

The approach that offers telcos the most flexibility and control is an open source/multi-vendor approach. The next section details the benefits of this approach.
Open Source

The open source development process provides two main benefits that can be leveraged by SDN controller vendors:

- **Faster time to market:** Open source development does not wait for a standards body to define a specification, which can take many years to complete, but rather begins writing code from the outset. Working code is available sooner in the process, so things like interoperability can be tested for early on. Also, the process results in an actual product in the end, not just a document that is likely to be interpreted differently by each participant before it is turned into code.

- **More robust code:** Because there are developers contributing from a variety of companies and institutions – each with different perspectives – there is often significant debate throughout an iterative process that leads to a better product in the end. Also, once released, contributors provide real world insight that makes it back into the product in enhancements and bug fixes so everyone can benefit.

Evaluating Open Source SDN Controllers

In addition to the benefits mentioned above, open source development provides others that are especially relevant to SDN controllers. Even so, there are some concerns about using an open source approach.

**Benefits of Using Open Source Solutions**

Because contributions come from representatives of many companies, using open source solutions increases the chances that the code will support devices from multiple vendors, eliminating the vendor lock-in that telcos are hoping to avoid. This will make it easier for telcos to integrate into their existing environments, which are generally multi-vendor.

Telcos will also appreciate the flexibility that comes with application portability between open source-compliant controllers. Lower software license and maintenance costs are yet another welcome effect.

When APIs are easy to use and have robust capabilities and apps have been pre-defined, this approach can enable more rapid service component creation. The most current architectures and technology options can be leveraged, as the code is being updated continuously. Also, as changes occur on the northbound and southbound side, the open source code gets updated accordingly.

**User Concerns About “Open” Solutions**

Despite these benefits, telcos and others have some concerns about using open source solutions. They worry that having the code openly available could make it more vulnerable to hackers. Also, while advocates say the open approach leads to more stable code and reduces risk, there remain concerns due to the fact that the code can be modified by anyone at any time. No longer having software written specifically for a particular device could make it more difficult to integrate into an existing network. Similarly, since no one “owns” the code, no one is responsible for supporting it.

While these are legitimate questions, the benefits look to vastly outweigh these arguments. They should not dissuade anyone from embracing an open source controller. A careful evaluation, as described below, should alleviate any misgivings.
**Additional Considerations**

As they consider different "open" solutions, telcos should be advised to watch for characteristics of the solutions that will make a significant difference in their usability. The solution should provide access to the API code so that telcos (and other application developers) can manipulate the code, not merely expose the API for others to leverage. It should also be noted that support for open standards does not necessarily guarantee that the code has not been modified or forked, so telcos should ask this question directly of their potential controller suppliers.

Another consideration is whether to directly acquire open source code. A telco could certainly download open source and compile itself and not use a distribution from a commercial supplier. However, in this scenario, the operator would then be responsible for things like testing, integration, third-party testing, software lifecycle management and troubleshooting. These are functions that the average telco is not in a strong position to support. Plus, it would not get the assurance of a warranty. With a commercially-supported solution, the customer could then focus its internal IT resources on operations/business support system (OSS/BSS) integration, APIs to other internal systems and, most importantly, new applications to support new business opportunities.

**Conclusion**

Telcos are facing challenges that are leading them to evaluate new approaches to their data center and WAN environments. They must become more agile in order to better compete against Internet players, as well as better leverage their existing WAN assets.

The architecture proposed with SDN provides a solution to many of these challenges. The SDN controller plays a critical role in achieving the benefits of more efficient network control and better application performance.

SDN controllers based on open source code benefit greatly from the power of the development community. The underlying code is made more robust by the iterative development process and the contributions from multiple industry players help alleviate concerns about vendor lock-in. Vendors that embrace open source development in their SDN controller solutions will be viewed highly favorably by telcos as they transition to new SDN architectures.