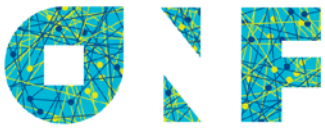

OPEN NETWORKING FOUNDATION

SDN Migration Considerations and Use Cases

ONF Solution Brief

November 21, 2014

ONF TR - 506

Disclaimer

Open Networking Foundation
2275 E. Bayshore Road, Suite 103, Palo Alto, CA 94303
www.opennetworking.org

Table of Contents

Introduction

Open Software-Defined Networking (SDN) has been well accepted by the networking industry as the way to transform enterprise, data center, service provider, carrier, and campus networks. The objective of this SDN transformation is to enable differentiated new services faster than ever before, simplify the network, and lower the total cost of ownership (TCO). The key attributes for a network that has been migrated to SDN are programmability, openness, heterogeneity, and maintainability. SDN will also facilitate the re-architecture required to address the increasing demand on the network due to dynamic connectivity.

As more network operators adopt open SDN, there is a need for best practices to facilitate the migration of existing networks and services to SDN. This solution brief provides a summary of key findings and recommendations on SDN migration. It offers a fresh perspective on SDN migration best practices through real-world use cases shared by leading SDN pioneers. It discusses various migration scenarios—including campus, edge, and inter-data-center wide-area networks—including their challenges and recommendations on migration methods, tools, and systems. For a more in-depth discussion of forward-thinking operators who have shared their migration use cases, please refer to the Open Networking Foundation (ONF) *Migration Use Cases and Methods* document.

SDN Overview

Software-Defined Networking is a new architecture that has been designed to enable more agile and cost-effective networks. The Open Networking Foundation (ONF) is taking the lead in SDN standardization, and has defined an SDN architecture model as depicted in Figure 1.

The ONF/SDN Architecture consists of three distinct layers that are accessible through open APIs:

- The **Application Layer** consists of the end-user SDN applications that consume the SDN communications services. The boundary between the Application Layer and Control Layer is traversed by the northbound API.
- The **Control Layer** provides the consolidated control functionality that supervises the network forwarding behavior through an open, programmatic interface such as OpenFlow.
- The **Infrastructure Layer** consists of the physical or virtual network elements and devices that provide packet switching and forwarding.

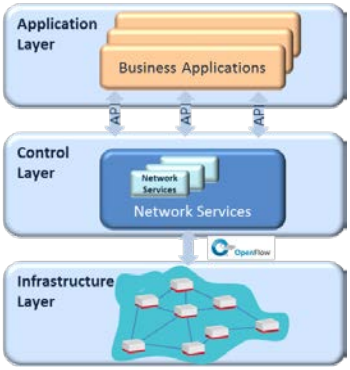

Figure 1 ONF/SDN architecture

According to this model, SDN architectures are characterized by following three key attributes:

- **Logically centralized intelligence**. In an SDN architecture, network control is decoupled from forwarding using a standardized southbound interface such as OpenFlow™. By centralizing network intelligence, decision-making is facilitated based on a global (or domain) view of the network. Today's networks are instead built on an autonomous system view, where nodes are unaware of the overall state of the network, limiting the flexibility for network control.
- **Programmability**. SDN networks are inherently controlled by software functionality, which may be provided by vendors or the end users themselves. Network control capabilities, embedded into the Control Layer are referred to as "network services." Network programmability enables conventional network management to be automated. This automation has also been influenced by rapid adoption of the cloud. By providing open APIs for applications to interact with the network, SDN networks can achieve unprecedented innovation and differentiation.
- **Abstraction**. SDN applications that consume SDN services are abstracted from the underlying network technologies. This enables a common control layer to support a diverse range of applications, and simultaneously support equipment and technologies in the Infrastructure layer from multiple vendors.

In addition to supporting the desired level of flexibility and scalability (via segmentation of the network), Open SDN also provides tremendous opportunity for automation of configuration and service management in a vendor-independent fashion. Automation ensures tight coupling of the network operations with business objectives and expected quality of experience, ultimately serving to accelerate service velocity while streamlining network operations.

Please see the white paper *Software-Defined Networking: The New Norm for Networks* for more information on ONF and SDN.

SDN Migration Considerations

While SDN deployment in a new data center is relatively straightforward, most operators do not have the luxury of a green-field environment. Consequently, migration planning is essential to paving the way towards SDN.

A number of challenges must be confronted along the way, including cost, performance, service availability, management, and security. Addressing security vulnerabilities is among the highest priorities for network operators. The Open Networking Foundation recognizes these concerns and established a Security Discussion Group to focus on SDN security considerations, which are also being addressed by several other ONF working groups. The implications of SDN security are addressed in the ONF solution brief *SDN Security Considerations in the Data Center*.

To begin the voyage towards SDN, there are questions that may naturally come up, including:

- What are my goals for migrating to open SDN?
- What are the initial steps I should take to achieve my goals for SDN?
- What are my migration options?

- How have others performed the migration, and how different from their strategies is my current SDN migration plan?

The migration to SDN can be a daunting and challenging task. However, the more time spent answering the above questions, the greater the mitigation of fundamental concerns. The key steps involved in an SDN migration are:

- **Identify and prioritize core requirements of the target network**. Not all requirements of the traditional starting network may be met, at least initially, by the target software-defined network.
- **Prepare the starting network for migration**. The starting network might need to be moved to a clean intermediate standard state from which the rest of the migration can proceed.
- **Implement a phased network migration**. Migrating individual devices will necessitate device-specific drivers and methods.
- **Validate the results**. Once migration is completed, the target network must be validated against a documented set of requirements or expectations.

Figure 2 Migration steps

Open SDN Migration Use Cases

SDN migration use cases fall into three main categories: legacy-to-greenfield, legacy-to-mixed, and legacy-to-hybrid. Greenfield scenarios are the least complex because there is no need to support integration or interoperation with an existing non-OpenFlow—based network infrastructure, unlike legacy-to-mixed or legacy-to-hybrid scenarios. With legacy-to-mixed (or ships-in-the-night), new OpenFlow devices are deployed and co-exist with traditional switches/routers and interface with legacy control planes. OpenFlow controllers and traditional devices need to exchange routing information via the legacy control plane. With a hybrid network deployment, hybrid devices interface with both OpenFlow controllers and legacy control plane.

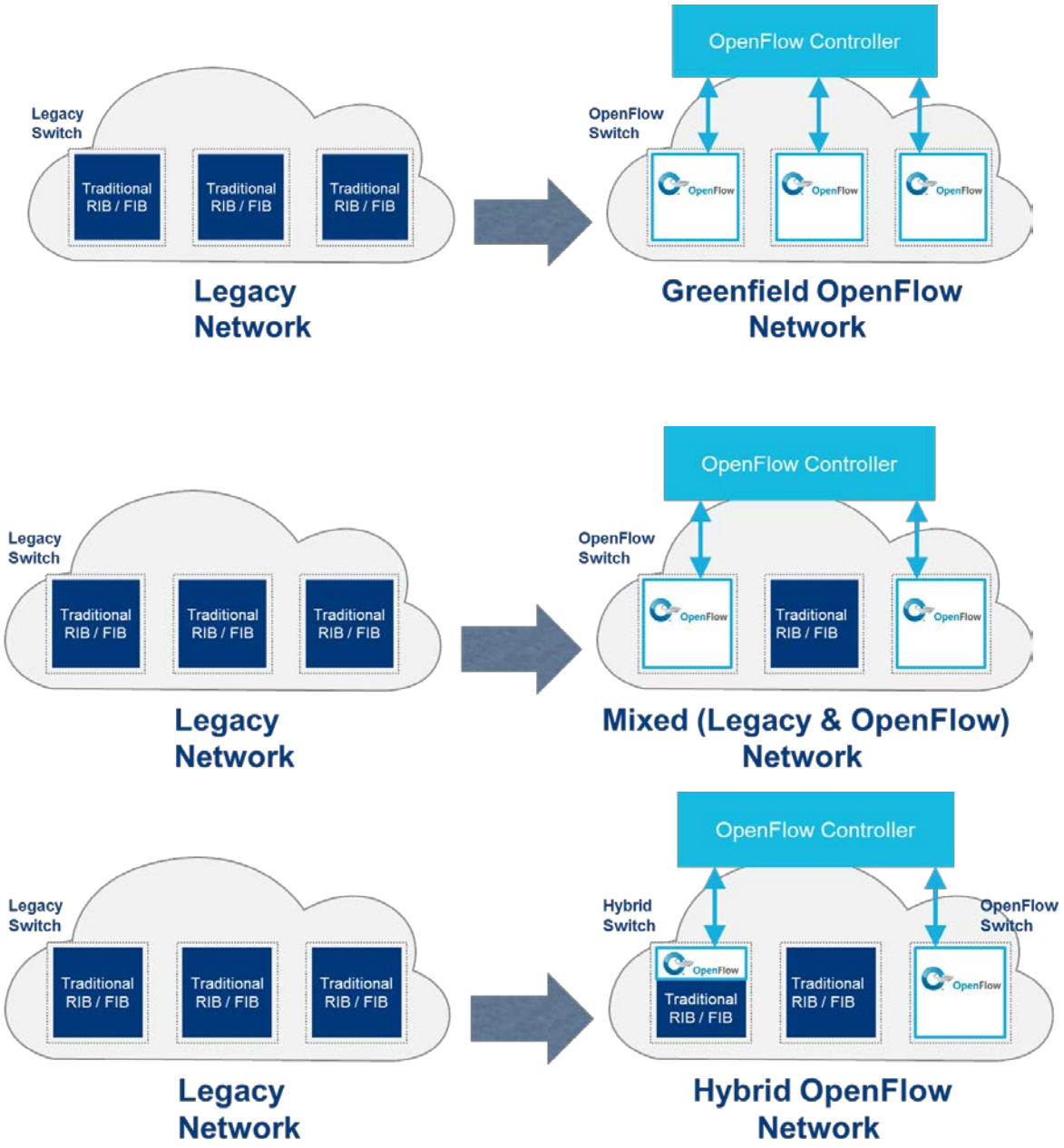

Figure 3 Migration approaches

The SDN migration categories shown in Figure 3 are relevant and applicable to multiple network segments and tiers.

The following sections summarize two well-publicized SDN migrations, one for a campus Wireless LAN (WLAN), the other for a cloud operator multi-service Wide Area Network (WAN). The two use cases were selected for their diverse breadth and scope, different customer base, and business verticals. Both of these legacy-to-hybrid migration scenarios are more challenging to design and deploy than greenfield deployments.

Stanford University legacy-to-hybrid migration

One of Stanford University's primary motivations for SDN migration was to gain better insight into and verification of OpenFlow as a viable technology. Stanford deployed a fully functional SDN network using OpenFlow controllers over part of its campus. This migration was focused initially on wireless users, followed by select wired users spanning two independent buildings. This migration encompassed several IEEE 802.1q VLANs, which were interconnected—as is commonly done—with a Layer 3 router.

In one of the buildings, Stanford deployed six 48-port 1GE OpenFlow-enabled switches from various vendors. The second building deployed one 48-port OpenFlow-enabled switch. One Layer 2 domain was used at each building to support about 34 Wi-Fi access points. These access points supported 802.11g interfaces running Linux-based switching software and one WiMAX base station in one of the buildings.

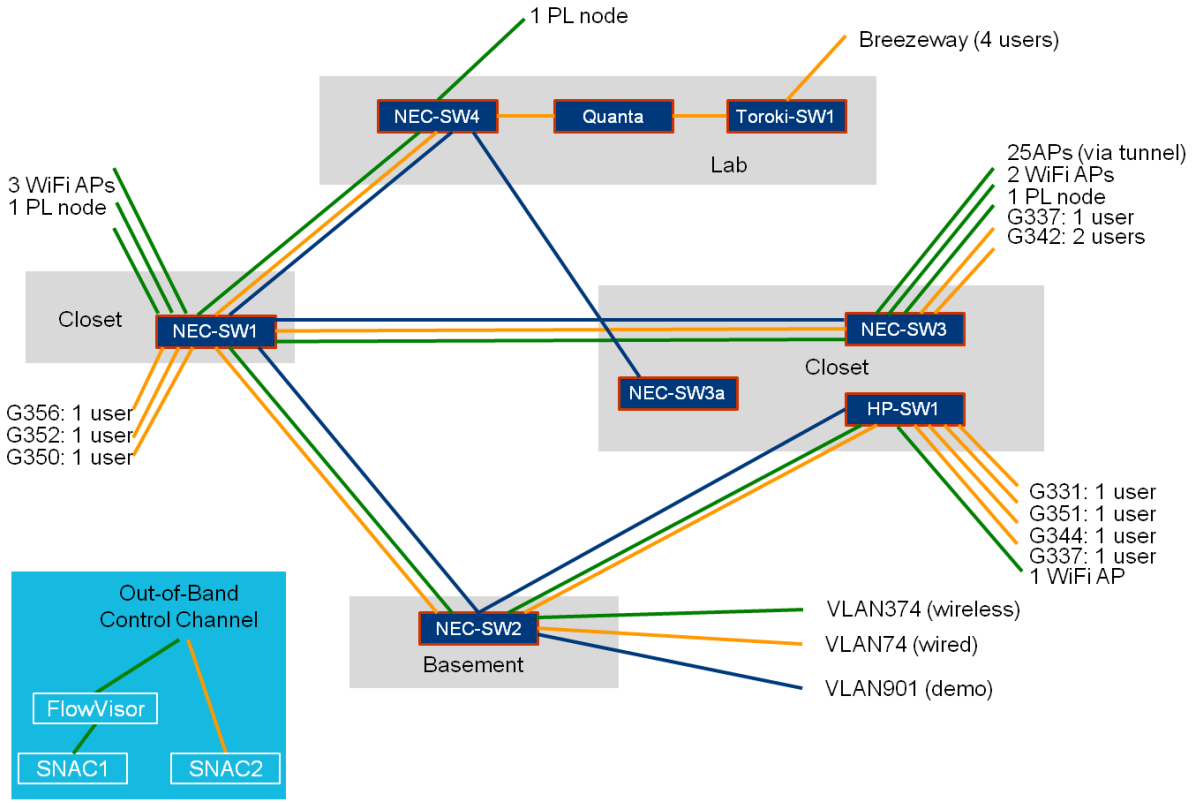

Figure 4 OpenFlow-enabled wing of Stanford's William Gates Building

The target campus deployment specified network availability to exceed 99.9% with a fail-safe scheme to revert back to the legacy network in case of significant outages. In addition, network performance was specified to be comparable to the legacy network benchmarks without any impact to user experience.

The migration was accomplished in five phases, as more fully detailed in the *Use Cases and Migration Methods* document. The first phase exposed traffic visibility to facilitate network configuration changes

for selected users. In phase 2, VLANs and users were migrated to OpenFlow-based SDN through the following steps:

1. Deployed OpenFlow support on relevant hardware via a software update.
2. Verified OpenFlow support on switches for the configured VLANs and to test endpoint reachability.
3. Migrated users to the OpenFlow-enabled network.
4. Enabled OpenFlow for the new subnet by configuring the OpenFlow controller.
5. Validated migration objectives for reachability, performance, and stability using network management tools.

Google legacy-to-hybrid migration

Google's OpenFlow WAN is organized as two distinct backbones, one carrying Internet-facing user traffic and another carrying internal traffic between Google's global data centers. The breadth of user requirements and the broad scale of the project made Google's SDN OpenFlow migration a unique, challenging use case demonstrating OpenFlow's flexibility.

The objective of Google's WAN migration was to improve scalability, flexibility, and agility in managing the Internet-facing WAN fabric to enhance Google's user-based services, including Google+, Gmail, YouTube, Google Maps, and others.

Both of Google's wide-area networks support thousands of individual applications, tremendous traffic volumes, and latency sensitivities—all governed by different overall priorities. Google's internal network that connects multiple data centers is an OpenFlow-based network today and a well-known SDN use case. This inter-data-center network was built in a three-layer architecture: switch hardware layer, site controller layer, and global control layer.

Google's SDN migration path moved in stages from a fully distributed monolithic control and data plane hardware architecture to a physically decentralized (though logically centralized) control plane architecture. The hybrid migration for the Google B4 network proceeded in three general stages:

1. **Starting network (Figure 5).** In the initial stage, the network connected data centers through legacy nodes using E/IBGP and ISIS routing. Cluster border routers interfaced the data centers to the network.

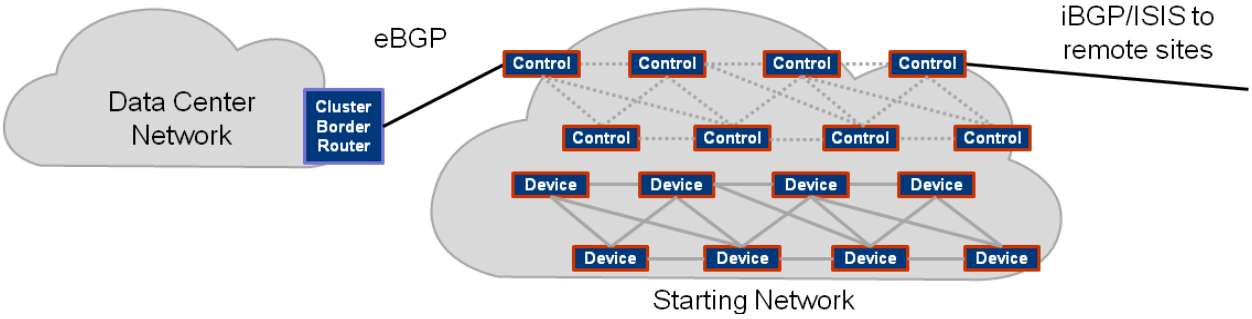

Figure 5 B4 starting network

2. **Phased deployment (Figure 6).** In this mixed-network phase, a subset of the nodes in the network were OpenFlow-enabled and controlled by the logically centralized controller utilizing Paxos, an OpenFlow controller, and Quagga open source routing stack that Google adapted to its requirements.

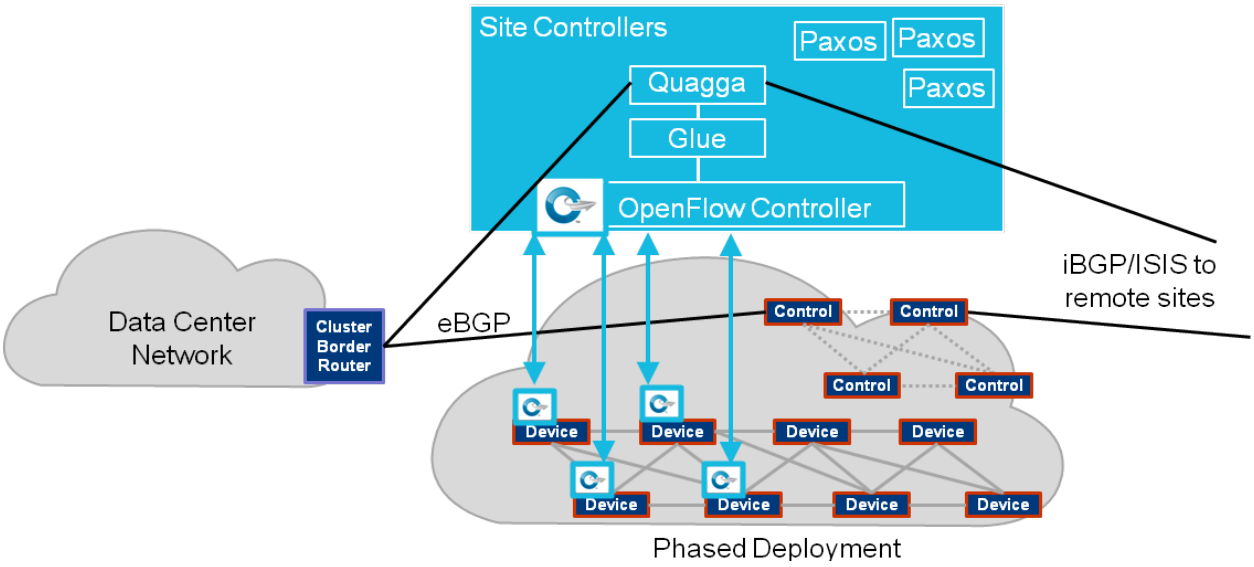

Figure6 B4 phased deployment mixed network

Target network (Figure 7). In this final phase, all nodes were OpenFlow-enabled. In the target network, the controller controls the entire network. There is no direct correspondence between the data center and the network. The controller also has a TE server that guides the traffic engineering in the network.

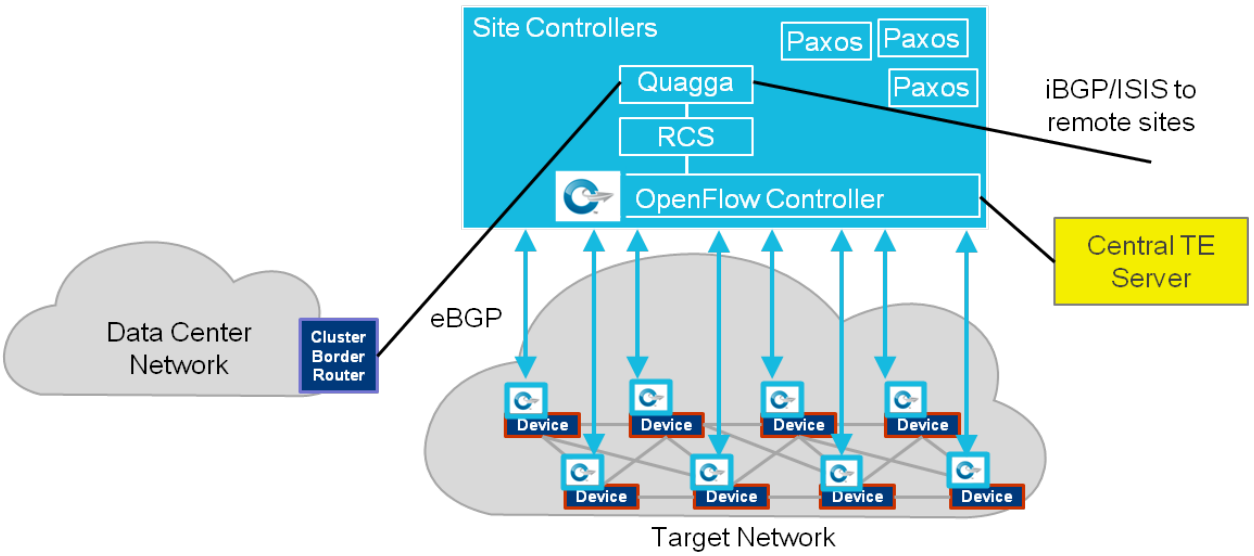

Figure 7 B4 target network

Google initially carried out a migration assessment that revealed:

- The majority of Google's WAN traffic involves synchronizing large data sets across remote data centers that can tolerate periodic failures and bandwidth disruptions, but benefit from as much bandwidth as available.
- The total number of WAN data center sites is a few dozen.
- Google enforced application priorities and traffic engineering at the WAN edge instead of resorting to inefficient over-provisioning.
- Achieving scalability similar to the legacy network would be prohibitively expensive.

The network objectives assessment influenced Google's network migration approach, which was:

- Deploy legacy routing to create a gradual path for enabling OpenFlow in both of Google's WAN networks.
- Adopt BGP as a step towards more customized protocols.
- Pursue three phases to migrate from a fully distributed monolithic control and data plane hardware architecture, to a physically decentralized control plane architecture.

In the first phase, the data centers were interconnected using legacy border routers with E/I-BGP and ISIS routing. The second phase involved upgrading a subset of nodes to OpenFlow using the Paxos distributed control algorithms, an OpenFlow controller, and a Quagga-based routing stack. In the final phase, all nodes were migrated to an OpenFlow-based network in which the OpenFlow controller is capable of traffic engineering functionality using a TE server.

Google's successful internal OpenFlow-based network might be readily misconstrued as carrying less traffic than its Internet-facing network. Actually, the opposite is true: Google's internal network not only carries more traffic than the public-facing WAN, it is also growing at a much higher rate. Google's OpenFlow-based internal network supports rapid deployment and interaction of novel control and functionality, such as traffic engineering, which achieved over 95% utilization. It is also integrated closely with user applications that allow the network to adapt to failures or changing communication patterns.

For more details on Google's OpenFlow-based network, please see Google's white paper, *Inter-Datacenter WAN with centralized TE using SDN and OpenFlow*, available on the ONF website.

Other use cases

There are multiple other examples of SDN OpenFlow-based migration use cases that have been publicly disclosed, spanning diverse network tiers and customer segments. For instance, NTT DOCOMO implemented a Mobile (Evolved) Packet Core (EPC) using OpenFlow-based SDN. This migration was motivated by the 2011 earthquake and tsunami in Japan, and it enabled NTT DOCOMO to improve the resiliency of the network against future disasters. In addition, the migration supported rapidly evolving data- and transaction-intensive mobile user applications, and enabled NTT DOCOMO to automate policy and network resources management while keeping costs under control.

SDN Migration Best Practices

Migration strategies vary widely for enterprise/campus, WAN, and carrier/service provider networks. However, there are common best practices and recommendations based on the initial experiences of leading ONF member network operators addressed in this solution brief. These recommendations are documented in significant detail in the ONF Migration Working Group's *Migration Use Cases and Methods* document.

Best practices are divided into two phases: pre-migration planning, and migration. In the pre-migration planning phase, anticipating the impact of the planned migration on existing services is crucial. For any foreseen disruptive impact, one needs to make sure that alternative options are available. Pre- and post-migration check lists should be created for assuring service continuity. Preparing for unexpected problems requires instituting and documenting procedures for reverting back to the starting network in case migration issues are encountered or in case of service degradation.

During the migration phase, we recommend operators provision all network management and troubleshooting tools (ping, trace-route, etc.) for the migrated network, and to perform timely upgrades and version control for all protocols in use (e.g., OpenFlow, etc.). Creating a dummy service to continuously check service availability during the migration can help prevent or reduce any service disruption during the process.

Migration tools

Operational tools used during the migration ideally can exploit the benefits of SDN-based deployments. There are three categories of tools to consider: monitoring tools, management and configuration tools, and testing and verification tools. Monitoring tools are used to detect, quantify, and report on network service disruptions, and to measure network service quality and network performance. Management and configuration tools help with migration-related configurations and provide support for rollback if needed. Testing and verification tools validate the OpenFlow controller and OpenFlow switch implementations.

There are many open source and commercial tools available for SDN migrations. Tools should be evaluated based on their ability to meet SDN migration requirements of the network. These requirements include multiple vendor support, technology independence, scalability, and proper protocol version support.

For a more detailed look at the tools and metrics available for use during a network migration to SDN, see the *Migration Tools and Metrics* document.

Conclusion

Enabling new services is an important motivation for SDN migration. These services achieve end-to-end connectivity, overlay on top of virtual networks, span several network segments, and/or cross several layers of networking technologies. Some or all of those situations could possibly be addressed by migrating to OpenFlow. OpenFlow is still evolving as new use cases and deployment models are defined. There is tremendous value in examining use cases across different network types to better understand unique migration strategies, tools, and methods that could be specific to each service or network type given the large diversity of emerging SDN use cases.

Many operators have identified maintaining service continuity during the migration as a top priority. It is critical to focus on service continuity with minimal disruption. And it is important to maintain the consistency of OpenFlow protocol versions between SDN controllers and switches. Pre- and post-migration checklists that address specific applications, along with the tools and guidelines mentioned in this paper, will also help ensure a smooth migration.

Given that the Google WAN and Stanford campus software-defined network use cases have been in production for a number of years, we conclude that traditional networks can be successfully migrated to OpenFlow-based SDN by leveraging the framework and approaches described in this document. Clearly, more work is still needed to cover the diverse range of possible SDN deployments. The ONF Migration Working Group will continue to revise and refine its recommendations for SDN migration methods, tools, and systems, including additional SDN migration use cases as they are defined and deployed.

References

"The Evolved Packet Core" (3rd Generation Partnership Project article)
http://www.3gpp.org/technologies/keywords-acronyms/100-the-evolved-packet-core

Inter-Datacenter WAN with centralized TE using SDN and OpenFlow (Google white paper)
https://www.opennetworking.org/images/stories/downloads/sdn-resources/customer-case-studies/cs-googlesdn.pdf

Migration Use Cases and Methods (ONF Migration Working Group document)
https://www.opennetworking.org/images/stories/downloads/sdn-resources/use-cases/Migration-WG-Use-Cases.pdf

Migration Tools and Metrics (ONF Migration Working Group document)
https://www.opennetworking.org/images/stories/downloads/sdn-resources/technical-reports/migration-tools-and-metrics.pdf

"NTT DATA's Advance in SDN Business Provides Highly-Flexible Control of Network by Software" (NTT DATA press release)
http://www.nttdata.com/global/en/news-center/pressrelease/2012/060801.html

Paxos (Yale wiki)
http://www.cs.yale.edu/homes/aspnes/pinewiki/Paxos.html

Quagga (Network Device Education Foundation website)
http://www.opensourcerouting.org/

SDN Security Considerations in the Data Center (ONF solution brief)
https://www.opennetworking.org/images/stories/downloads/sdn-resources/solution-briefs/sb-security-data-center.pdf

Software-Defined Networking: The New Norm for Networks (ONF white paper)
https://www.opennetworking.org/images/stories/downloads/white-papers/wp-sdn-newnorm.pdf

Contributors

Ash Bhalgat
Hakki Cankaya
Marc Cohn
Justin Dustzadeh
Yardiel Fuentes
Bhumip Khasnabish
Shweta Latawa
Mike McBride
Evelyne Roch