Journal of Advances in Science and Technology
Vol. 12, Issue No. 25, (Special Issue) December-2016, ISSN 2230-9659

Software Defined Networking

Mohan A. Gholap[1]*, Aashita A. Mahajan[2], Y. M. Naik[3]

[1]P.G. Student, GIT, Belgaum

[2]Ph.D. Scholar, Kalinga University, Raipur

[3]Asst. Prof. HIT, Nidasoshi

Abstract – Over the last decade, the hottest topics in networking have been software defined networking (SDN). However, there is considerable confusion amongst enterprise IT organizations about SDN. There are many sources of that confusion, including the sheer number of vendors who have solutions that solve different problems using different solution architectures and technologies, all of whom claim to be offering SDN solutions. The primary goal of this our paper is to eliminate that confusion. In order to accomplish that goal, this paper will put SDN into the context of a broad movement to have more of a focus on software based solutions and it will identify the key opportunities that SDN can address.

-------------------------♦-----------------------------

I. INTRODUCTION

Communication networks are growing in size and complexity at an ever-increasing rate, with the conventional infrastructure, network systems, and protocol stack, which hardly provide adequate solutions to the contemporary networking demands. This triggered the emergence of a different approach to network systems architecture, called Software-Defined Networking (SDN) [1]. SDN, has been present for the last 20 years, [2]. Recently, OpenFlow [3] succeeded in establishing itself as an SDN industry standard. OpenFlow validated the SDN approach, and many network architectures, network systems, and data centers adopted SDN and turned it into a mainstream approach in network design.

SDN offers many advantages, such as centralized and decentralized control of multiple cross-vendor network elements, mainly data plane platforms with a common API abstraction layer for all SDN-enabled equipment. It also reduces the complexity of network configuration and operation that is achieved by automation high level configuration is translated into specific forwarding behavior of network elements. SDN allows easy deployment of new protocols and network-services as a result of high operation abstraction. Increased control granularity in SDN allows a per flow definition with a high granularity policy level. SDN infrastructure can adjust to the specific user application running on it via the control plane, which greatly improves the user experience.

Software Defined Networking, however, has its disadvantages: the added flexibility and functionality require additional overhead on the equipment, and as a result there are performance penalties in terms of processing speed and throughput. This is not to say that the overall performance is necessarily decreasing; many network services and tasks that were executed by the end-nodes or by the control or management layers of the network systems can be executed by the SDN-enabled equipment in a simpler and quicker way, thereby improving the overall performance of the networking tasks.

Despite SDN's continuing growth in popularity, there have been relatively few studies that deal with performance evaluation of SDN architectures. Tootoonchian et al. [4] focused mainly on performance evaluation of the control plane of SDN. Rotsos et al. [5] proposed a tool for evaluating performance of one specific SDN architecture, i.e., several OpenFlow implementations, and measured raw performance of OpenFlow without comparison to other SDN solutions, or to non-SDN network systems. Their work indicated that performance is affected by the number and type of actions applied to the data-frame, as well as the specific implementation of the SDN. Another study of data plane performance evaluation of OpenFlow soft switch was carried out by Bianco et al. [6], using different frame sizes and rules.

II. TRADITIONAL DATA NETWORK

In the traditional approach to networking, most network functionality is implemented in a dedicated appliance such as switch, router, application delivery controller, etc. In addition, within the dedicated appliance, most of the functionality is implemented in

www.ignited.in

dedicated hardware such as an ASIC (Application Specific Integrated Circuit).

Disadvantages of this approach to developing network appliances are:

- The ASICs that provide the network functionality evolve slowly
- The evolution of ASIC functionality is under the control of the provider of the appliance
- The appliances are proprietary
- Each appliance is configured individually
- Tasks such as provisioning, change management and de-provisioning are very time consuming and error prone

Networking organizations are under increasing pressure to be more efficient and agile than is possible with the traditional approach to networking. One source of that pressure results from the widespread adoption of server virtualization [7]. The bottom line is that a traditional network evolves slowly; is limited in functionality by what is provided by the vendors of the ASICs and the vendors of the network appliances; has a relatively high level of OPEX and is relatively static in nature. SDN holds the promise of overcoming those limitations.

III. A FLOW TOWARDS SOFTWARE BASED APPROACHES

As noted, the traditional data network has been largely hardware-centric. However, over the last few years the adoption of virtualized network appliances and the burgeoning interest in software defined data centers (SDDCs) have lead a movement towards an increased reliance on software-based network functionality. For example, in the mid to late 2000s, network appliances such as WAN Optimization Controllers (WOCs) and Application Delivery Controllers (ADCs) were purpose-built, hardware appliances[8]. That means that functions such as encryption/decryption and the processing of TCP flows were performed in hardware that was designed specifically for those functions. Driven largely by the need for increased agility, it is now common to have WOC or ADC functionality provided by software running on a general purpose server or on a VM.

A SDDC can be looked at as the complete opposite of the traditional data center network that was previously described. For example, one of the key characteristics of a software-defined data center is that all of the data center infrastructure is virtualized and delivered as a service. Another key characteristic is that the automated control of data center applications and services is provided by a policy-based management system.

A. Opportunities

One of the characteristics that is often associated with any fundamentally new approach to technology is that there is confusion about the opportunities that can be addressed by that new approach. In order to successfully evaluate and adopt a new approach to technology such as SDN [8], IT organizations need to identify which opportunity or opportunities that are important to the organization are best addressed by that new approach.

After all of the SDN-related discussions that have occurred over the last couple of years, the following have emerged as the most likely set of opportunities that SDN can address.

- Support the dynamic movement, replication and allocation of virtual resources.
- Ease the administrative burden of the configuration and provisioning of functionality such as QoS and security.
- More easily deploy and scale network functionality.
- Perform traffic engineering with an end-to-end view of the network.
- Better utilize network resources.
- Reduce OPEX.
- Have network functionality evolve more rapidly based on a software development lifecycle.
- Enable applications to dynamically request services from the network.
- Implement more effective security functionality.
- Reduce complexity.

IV. SOFTWARE DEFINED NETWORKING (SDN)

The Open Networking Foundation (ONF) is the group that is most associated with the development and standardization of SDN [7]. According to the ONF1, “Software-Defined Networking (SDN) is an emerging architecture that is dynamic, manageable, cost-effective, and adaptable, making it ideal for the high-bandwidth, dynamic nature of today’s applications. This architecture decouples the network control and forwarding functions enabling the network control to become directly programmable and the underlying infrastructure to be abstracted for applications and network services. The OpenFlow protocol is a foundational element for building SDN solutions."

A. ADVANTAGES

1. **Directly programmable:** Network control is directly programmable because it is decoupled from forwarding functions.

2. **Agile:** Abstracting control from forwarding lets administrators dynamically adjust network-wide traffic flow to meet changing needs.

3. **Centrally managed:** Network intelligence is (logically) centralized in software- based SDN controllers that maintain a global view of the network, which appears to applications and policy engines as a single, logical switch.

4. **Programmatically configured:** SDN lets network managers configure, manage, secure and optimize network resources very quickly via dynamic, automated SDN programs, which they can write themselves because the programs do not depend on proprietary software.

5. **Open standards-based and vendor-neutral:** When implemented through open standards, SDN simplifies network design and operation because instructions are provided by SDN controllers instead of multiple, vendor-specific devices and protocols.

B. ARCHITECTURE

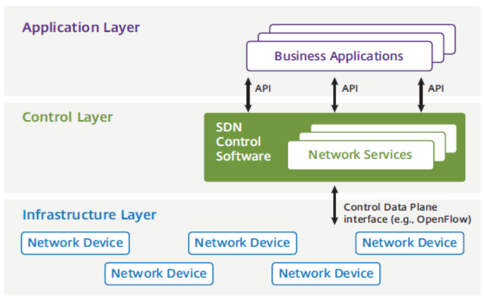

Figure 1: The SDN System Architecture

Below is the description of some of the key concepts that are part of the SDN system architecture that is shown in Figure 1.

a. Business Applications

This refers to applications that are directly consumable by end users. Possibilities include video conferencing, supply chain management and customer relationship management.

b. Network & Security Services

This refers to functionality that enables business applications to perform efficiently and securely. Possibilities include a wide range of L4 – L7 functionality including ADCs, WOCs and security capabilities such as firewalls, IDS/IPS and DDoS protection.

c. Pure SDN Switch

In a pure SDN switch, all of the control functions of a traditional switch (i.e., routing protocols that are used to build forwarding information bases) are run in the central controller. The functionality in the switch is restricted entirely to the data plane.

d. Hybrid Switch

In a hybrid switch, SDN technologies and traditional switching protocols run simultaneously. A network manager can configure the SDN controller to discover and control certain traffic flows while traditional, distributed networking protocols continue to direct the rest of the traffic on the network.5

e. Hybrid Network

A hybrid network is a network in which traditional switches and SDN switches, whether they are pure SDN switches or hybrid switches, operate in the same environment.

f. Northbound API

Relative to Figure 1, the northbound API is the API that enables communications between the control layer and the business application layer. There is currently not a standards-based northbound API.

g. Southbound API

Relative to Figure 1, the southbound API is the API that enables communications between the control layer and the infrastructure layer. Protocols that can enable this communications include OpenFlow, the extensible messaging and presence protocol (XMPP) and the network configuration protocol. Part of the confusion that surrounds SDN is that many vendors don't buy in totally to the ONF definition of SDN. For example, while some vendors are viewing OpenFlow as a foundational element of their SDN solutions, other vendors are taking a wait and see approach to OpenFlow. Another source of confusion is disagreement relative to what constitutes the infrastructure layer. To the ONF, the infrastructure layer is a broad range of physical and virtual switches and routers. As described below, one of the current approaches to implementing network virtualization relies on an architecture that looks similar to the one shown in Figure 1, but which only includes virtual switches and routers.

C. NETWORK SECURITY

SDN architecture may enable, facilitate or enhance network-related security applications due to the controller's central view of the network, and its capacity to reprogram the data plane at any time. While security of SDN architecture itself remains an open question that has already been studied, the idea consists of periodically collecting network statistics from the forwarding plane of the network in a standardized manner (e.g. using Openflow), and then apply classification algorithms on those statistics in order to detect any network anomalies. If an anomaly is detected, the application instructs the controller how to reprogram the data plane in order to mitigate it.

Another kind of security applications leverages the SDN controller by implementing some moving target defense (MTD) algorithms. MTD algorithms are typically used to make any attack on a given system or network more difficult than usual by periodically hiding or changing key properties of that system or network. In traditional networks, implementing MTD algorithms is not a trivial task since it is difficult to build a central authority able of determining - for each part of the system to be protected - which key properties are hid or changed. In an SDN network, such tasks become more straightforward tasks to the centrality of the controller. One application can for example periodically assign virtual IPs to hosts within the network, and the mapping virtual IP/real IP is then performed by the controller. Another application can simulate some fake opened/closed/filtered ports on random hosts in the network in order to add significant noise during reconnaissance phase (e.g. scanning) performed by an attacker.

Developing applications for software defined networks requires comprehensive checks of possible programming errors. Since SDN controller applications are mostly deployed in large scale scenarios a programming model checking solution requires scalability.

V. SUMMARY

While a SDN is comprised of many enabling technologies, SDN is not a technology, but an architecture. Whether it is fabric or overlay-based, network virtualization can be viewed as a SDN application. The primary benefit of a network virtualization solution is that it provides support for virtual machine mobility independent of the physical network. SDN, however, has other potential benefits including easing the administrative burden of provisioning functionality such as QoS and security.

While some of the characteristics of a SDN, such as the increased reliance on software, are already widely adopted in the marketplace, vendors have only recently begun to ship SDN solutions and SDN adoption is just beginning. Given all of the potential benefits that SDN is likely to provide, IT organizations need to develop a plan for how they will evolve their networks to incorporate SDN.

REFERENCES

G. Goth, "Software-Defined Networking Could Shake Up More than Packets," IEEE Internet Computing, vol. 15, no. 4, pp. 6–9, 2011.

D. L. Tennenhouse and D. J. Wetherall, "Towards an active network architecture," SIGCOMM Comput. Commun. Rev., vol. 26, no. 2, pp. 5–17, April 1996.

N. McKeown, T. Anderson, H. Balakrishnan, G. M. Parulkar, L. L. Peterson, J. Rexford, S. Shenker, and J. S. Turner, "OpenFlow: enabling innovation in campus networks," Computer Communication Review, vol. 38, no. 2, pp. 69–74, 2008.

A. Tootoonchian, S. Gorbunov, Y. Ganjali, M. Casado, and R. Sherwood, "On controller performance in software-defined networks," in Proc. USENIX Hot-ICE, 2012.

C. Rotsos, N. Sarrar, S. Uhlig, R. Sherwood, and A. W. Moore, "OFLOPS: An Open Framework for OpenFlow Switch Evaluation," in Proc. PAM, 2012, pp. 85–95.

A. Bianco, R. Birke, L. Giraudo, and M. Palacin, "OpenFlow Switching: Data Plane Performance," in Proc. ICC, 2010, pp.1–5.

http://www.webtorials.com/content/2014 /01/2013-guide-to-network-virtualization-sdn-3.html

citrix.com/sdn

Corresponding Author

Mohan A. Gholap*

P.G. Student, GIT, Belgaum

E-Mail – ***mohangholap04@gmail.com***