Journal of Industrial Electronics Technology and Application Vol.2 No.1
ISSN(Online) : 2635-635X

A Survey: Architecture, Security Threats and Application of SDN.

Kashif Nisar [1,*)], Emilia Rosa Jimson [1)], Mohd. Hanafi bin Ahmad Hijazi [1)], Shuaib K. Memon [2)]

[1)] Universiti Malaysia Sabah, Faculty of Computing and Informatics, Jalan UMS, 88400 Kota Kinabalu, Sabah, Malaysia.
[2)] Auckland Institute of Studies, New Zealand

Abstract. Software Defined Networking (SDN) is a new technology that makes computer networks more programmable. This technology enables the user to control the network easily by allowing the user to control the applications and operating system. SDN not only introduces new ways of interaction within network devices, but it also gives more flexibility for the existing and future networking designs and operations. The main difference between SDN and Traditional Networking is SDN removes the decision-making part from the routers and it provides logically a centralized Control-Plane that creates a network view for the control and management applications. The SDN divides the network up in three planes: The Application-Plane, The Control-Plane and The Data-Plane Layers. Through the establishment of SDN many new network capabilities and services are enabled, such as Traffic Engineering, Network Virtualization and Automation and Orchestration for Cloud Applications. In this paper, I would like to make a comparison between SDN and traditional networking. The architecture of SDN will be explained based on the three layers: Application, Control-Plane and Data-Plane Layers. Besides that, the Controller, the OpenFlow Protocol, the SDN Security Threats and Corresponding Countermeasures will be also be discussed in this paper. In addition to that, I will also discuss the benefits, limitations and SDN Application.

Keywords; Software Defined Networking, OpenFlow, Controller, Control-Plane, Data-Plane

---

[*] Corresponding author: net4kashif@gmail.com
Received:2018.12.12; Accepted:1.25; Published: 2019.3.31

This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (http://creativecommons.org/licenses/by-nc/3.0/) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

1. Introduction

The change of traffic patterns, the increase of personal devices like notebooks and smartphones to access campus network and the increase of cloud services [2] are some of the reasons why our network needs a new network architecture.

The traffic patterns have obviously changed within the enterprise information center. Today's applications access different databases and servers, creating a flurry of "east-west" machine-to-machine traffic before returning data to the end user device in the classic "north-south" traffic pattern. Besides that, users are changing network traffic patterns as they push for access to corporate content and applications from any type of device. The increase of personal devices puts the Information Technology under pressure in order to protect the corporate data and intellectual property in a delicate manner.

The increase of cloud services resulting in unprecedented growth of both public and private cloud services add to the complexity. IT's planning for cloud services must be done in an environment of increased security, compliance and auditing requirements, along with business reorganizations, consolidations, and mergers that can change assumptions overnight. Providing self-service provisioning, whether in a private or public cloud, requires elastic scaling of computing, storage, and network resources, ideally from a common viewpoint and with a common suite of tools.

Handling mega datasets require massive parallel processing on thousands of servers, all of which need direct connections to each other. The rise of mega datasets is fueling a constant demand for additional network capacity in the data center. Operators of hyper-scale data center networks face the daunting task of scaling the network to previous unimaginable size, maintaining any-to-any connectivity without going broke.

One of the new technologies that has been proposed to overcome the stated problems above is Software Defined Networking (SDN) Technology. SDN is a Technology that introduces new network architecture, where the Control and Data Planes are decoupled. The SDN architecture illustrated in Figure 1 shows clearly that each of the switches in the network is controlled by a single controller, this means through SDN the programmers are able to configure the packet-forwarding rules installed on switches in order to have direct control of the behavior of the network [3].

Open Networking Foundation (ONF) states SDN as an "Emerging architecture that is dynamic, manageable, cost-effective and adaptable, thus making it ideal for the high-bandwidth, dynamic nature of today's applications" [1]. SDN architecture is divided into three (3) layers: Application, Control, and Data Planes layers.

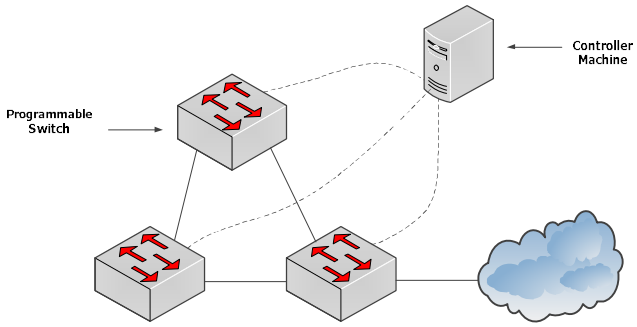

Fig. 1. Software Defined Network (SDN) Architecture

A. SDN Background

In 2007, SDN was invented at Standford University by Martin Casado [4] It was a collaboration between Martin Casado a PhD student of Standford University, Professor Nick McKeown who was Martin Casado's academic supervisor in the Electrical Engineering and Computer Science department [5], [6] and Scott Shenker, the professor of Computer Science at University of California, Berkeley [7]. The idea of SDN invention came from Martin Casado's PhD thesis entitled "Architectural Support for Security Management in Enterprise Networks". Through his research a flow based Ethernet switch controlled centrally from the outside was introduced, which is now known as "Ethane" [8]. He discovered the Ethane model to develop OpenFlow Protocol and developed program software to accomplish a range of control applications. All these activities contributed to the basic concept of SDN development [9].

Before SDN was invented, the intention to create a programmable network had long been conducted, for example the researchers in [9] supporting high speed programmable packet processing. This obviously shows that the researchers tried to create a programmable network. The team of these three doctors worked hard to make the network more programmable in order to solve technological problems. As an example, because the internet is a distributed network where individual communication devices exchange control information according to certain protocol, a new protocol needs to be created and all the communication nodes should be redesigned to make sure each communication node is compatible with the new protocol. To solve the problem, they introduced SDN which made the communication nodes externally controllable by implementing abstraction based on flow tables for each communication node and introduced right abstraction in network control. Besides that, they endorsed the concept of flexible software-based control of a whole network and clarified the layers at which to abstract external software control [10].

2. SDN Past, Present and Future

The SDN term was first coined by the [9] article which discussed the OpenFlow project at Stanford University. This article was published by MIT Technology Review [10].

3. Early Programmable Network Efforts

The effort to create programmable networks has been carried out over the years. The main reason for programmable networking is to make the implementation of new network services easier, which leads to the process of service formation and deployment. Table 1 shows early programmable network efforts that have become the SDN foundation. SDN started when its concepts were first explored in the Ethane project.

TABLE 1 EARLY PROGRAMMABLE NETWORK EFFORTS

Programmable Network Efforts	Year	Overview
Active Networking [11],[12]	In the mid-1990s	• David Tennenhouse called programmable network infrastructure as "Active Networks" [13]. • Considered Programmable Switch and Capsule approach. The Programmable Switch approach will provide a mechanism that will support the downloading of programs and retains the existing packet format. The capsules contain program fragments that can be interpreted and processed by routers. • Active networking never brought widespread industry usage due to security and performance concerns [14].
Devolved Control of ATM Networks (DCAN) [7]	In the mid-1990s	• The fundamental purpose of DCAN was to design and create infrastructure for ATM networks management. • SDN concept: DCAN removes control functions from the network device. Besides that, DCAN minimizes the protocol between the network and the manager, such as OpenFlow Protocol.
Forwarding and Control Element Separation (ForCES) [16]	2003	• The ForCES Network Element consists of Forwarding Elements and Control Elements. Both of these elements use ForCES protocol to communicate with each other. • The ForCES Network Element still presents the Forwarding Elements and Control Elements as a single network element to the industry.
4D Project [9]	2004	• 4D Project supported the separation between the routing decision logic protocols that govern the communication from network devices. • The Decision Plane has a global view of the network that used to control a Data Plane to forward traffic. • Works like NOX was inspired from the 4D Project.
Network Configuration Protocol (NETCONF) [17]	2006	• NETCONF is management protocols that modify the configuration of network devices. • NETCONF is a beneficial tool that can be used in parallel on hybrid switches to support solutions that enable programmable networking.
Ethane [18]	2006	• Ethane proposed a centralized controller to control the security and policy in a network. • Similar to SDN, Ethane has two components: • A Central Controller and Ethane Switches. • The Central Controller contains the global network policy that decides the forwarding process, while the Ethane switch provides a flow table and a secure channel to the controller. • The basic of original OpenFlow comes from switch design in Ethane.

4. SDN Challenges

Although SDN has just been introduced, many IT experts expect that deployments of SDN will increase rapidly over the next few years and at the same time will face many challenges that come from different aspects. In this section, several challenges are discussed.

5. Security

In terms of security, the SDN faces challenges to develop SDN applications that improve network security and to secure the SDN infrastructure itself. The OpenFlow specifications do not define the certificate format to ensure data integrity; this is because only minimum security is specified in SDN. This makes the SDN security need a two factor authentication (ini sy check d intrnt: nama lain bagi advanced authentication) and mechanism to encrypt for recovery of packets from failure and to avoid hackers. The researchers in divided these security challenges into 3 main problems, which are to secure the SDN infrastructure, to integrate security appliances with network control and to create languages and control methods that can enforce specified security policies.

6. Controller Design

This controller is unable to single-handedly control the whole traffic. To increase scalability, reliability, and integrity the centralized controller must be physically distributed. The [19] proposed Kandoo to preserve scalability without changing switches, introduces Bottom Layer and Top Layer controls. The Bottom Layer consists of controllers without interconnections and does not have knowledge about the network-wide state while the Top Layer maintains the network-wide state and it logically centralizes the controller.

7. Conclusion

The introduction of SDN created an opportunity for solving the Traditional Networks problems. For example, in Traditional Networks the Control and Data Planes are vertically integrated. This caused each of the elements in the network to have their own specific configuration and management interface. This makes the management of the network become complex. Through SDN the network management becomes simpler because SDN allows dynamic programmability in forwarding devices (Control-Plane elements) since the Control and Data Planes are decoupled. Besides that, SDN provides a global view of the network by logical centralization of the Control-Plane elements.

References

[1] Citrix, “SDN 101 : An Introduction to Software Defined Networking.” Open Networking Foundation, “Software-Defined Networking: The New Norm for Networks [white paper],” 2012.

[2] X. Foukas, M. K. Marina, and K. Kontovasilis, “Software Defined Networking Concepts,” 2014.

[3] N. Mckeown, T. Anderson, L. Peterson, J. Rexford, S. Shenker, G. Parulkar, J. Turner, and H. Balakrishnan, “OpenFlow : Enabling Innovation in Campus Networks,” pp. 1–6, 2008.

[4] K. Yap, M. Kobayashi, R. Sherwood, T. Huang, M. Chan, N. Handigol, and N. Mckeown, “OpenRoads : Empowering Research in Mobile Networks,” ACM SIGCOMM Comput. Commun. Rev., vol. 40, no. 1, pp. 125–126, 2010.

[5] D. Haeffner, "Software Defined Networks - Hype or Hope?", University of Applied Sciences, Osnabruck, 2013.

[6] "Nick McKeown", Yuba.stanford.edu, 2016. [Online]. Available: http://yuba.stanford.edu/~nickm/. [Accessed: 18- Aug- 2016].

[7] "Scott Shenker | EECS at UC Berkeley", Www2.eecs.berkeley.edu, 2016. [Online]. Available: https://www2.eecs.berkeley.edu/Faculty/Homepages/shenker.html. [Accessed: 18- Aug- 2016].

[8] S. Shenker, "Software-Defined Networking (SDN)", Unversity of California, Berkeley, 2014.

[9] "Martin Casado - The NAU Legacy: - People Making a Difference - Northern Arizona University", Nau.edu. [Online]. Available: https://nau.edu/legacy/profiles/martin-casado/. [Accessed: 18- Aug- 2016].

[10] D. E. Taylor, J. S. Turner, and J. W. Lockwood, “Dynamic Hardware Plugins (DHP): Exploiting Reconfigurable Hardware for High-Performance Programmable Routers,” Comput. Networks, vol. 38, no. 3, pp. 1–10, 2002.

[11] D. L. Tennenhouse, J. M. Smith, W. D. Sincoskie, W. David J, and G. J. Minden, “A Survey of Active Network Research,” IEEE Journals Mag., vol. 35, no. 1, pp. 80–86, 1997.

[12] D. L. Tennenhouse and D. J. Wetherall, “Towards an Active Network Architecture,” IEEE Conf. Publ., no. 5, pp. 2–15, 2002.

[13] J. M. Smith and S. M. Nettles, “Active Networking : One View of the Past , Present , and Future,” IEEE Trans. Syst. Man. Cybern., vol. 34, no. 1, pp. 4–18, 2004.

[14] J. T. Moore and S. M. Nettles, “Towards Practical Programmable Packets,” Conf. IEEE Comput. Commun. Soc., pp. 1–8, 2001.

[15] "Devolved Control of ATM Networks", 2000. [Online]. Available: https://www.cl.cam.ac.uk/research/srg/netos/projects/archive/dcan/. [Accessed: 18- Aug- 2016].

[16] "RFC 5810 - Forwarding and Control Element Separation (ForCES) Protocol Specification", Tools.ietf.org, 2010. [Online]. Available: https://tools.ietf.org/html/rfc5810. [Accessed: 18- Aug- 2016].

[17] "RFC 4741 - NETCONF Configuration Protocol", Tools.ietf.org, 2006. [Online]. Available: https://tools.ietf.org/html/rfc4741. [Accessed: 18- Aug- 2016].

[18] M. Casado, M. J. Freedman, J. Pettit, J. Luo, N. McKeown, and S. Shenker, “Ethane: Taking Control of the Enterprise,” Sigcomm ’07, pp. 1–12, 2007.