Journal of Industrial Information Technology and Application Vol.2 No.1

ISSN(Online) : 2586-0852

Future Internet: A Survey on Software Defined Networking

Emilia Rosa Jimson, Kashif Nisar*, Mohd Hanafi Ahmad Hijazi
Faculty of Computing and Informatics University Malaysia Sabah,
Jalan UMS, 88400 Kota Kinabalu Sabah, Malaysia

Abstract. The Software Defined Networking (SDN) is deliberated simplifying networks management and enabling Research & Development (R&D) innovations based on the decomposition of the control and data planes. SDN has kept a lot of consideration in very recent years, because it addresses the shortage of programmability in current network management designs and enables easier and faster network revolutionThe 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.

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.

* Corresponding author : kashif@ums.edu.my

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. Details are in Figure 1.

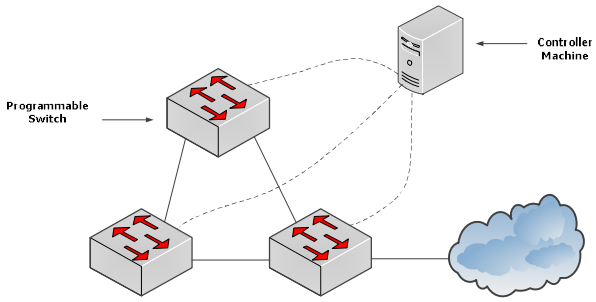

Figure 1: Software Defined Network (SDN) Architecture [15].

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.

In this paper, Section 1 introduces the needs of new network architecture and describes SDN in general. Section 2 will discuss the SDN security threats and countermeasures. Section 3 discusses the architecture of SDN based on the three layers of SDN: Application, Control-Plane, and Data-Plane Layers. Section 4 surveys the benefits of SDN. Section 5 discusses the limitation of SDN. Section 6surveys the Simulation/Emulation Tools for SDN. Section 6 provides concluding remarks.

Traditional Networking Versus Software Defined Networking (SDN)

In this section, the different concepts between Traditional Networking and Software Defined Networking are discussed. Based on [4], Traditional Networking is characterized by two main factors: (1) Most network functionality implemented in a dedicated appliance, and (2) the dedicated appliance is implemented in dedicated hardware. Dedicated appliance refers to one or multiple switches, routers, and/or application delivery controllers. In Traditional Networking, each switch has its own control and data planes, which are known as closed systems. The administrator of Traditional Networking needs to update each switch inside the network in case he or she wants to deploy new services or protocols in the network.

Nowadays, organizations are using devices from different vendors. In traditional networks, all these devices are placed in the same 'Zone' which contributes to the increase of the risks of external parties' access to the entire network. Besides that, the organization faces difficulties in incorporating all these devices within the network in a safe and structured manner. To improve this traditional networking limitation, the SDN concept should be applied to the network.

As illustrated in Figure 2 SDN removed the Control-Planes from the switches while the Data-Planes remain in the switches. Decoupling between these two planes involves leaving the Data-Plane with network hardware and moving the Control-Plane into a software layer. This makes policies which no longer have to be executed on the hardware itself because the use of centralized software application functioning as the Control-Plane makes network virtualization possible.

In contrast with Traditional Networking, in SDN, the network administrator can simply create different 'Zones' for a different device. These isolated zones provide additional layers of protection to the whole network. This means if a device gets hacked, the hacker will not get direct access to the complete network. Instead, the "leak" is restricted to one zone. In SDN the switches also become simpler since any activities such as deploying or updating new protocols are all done through the controller

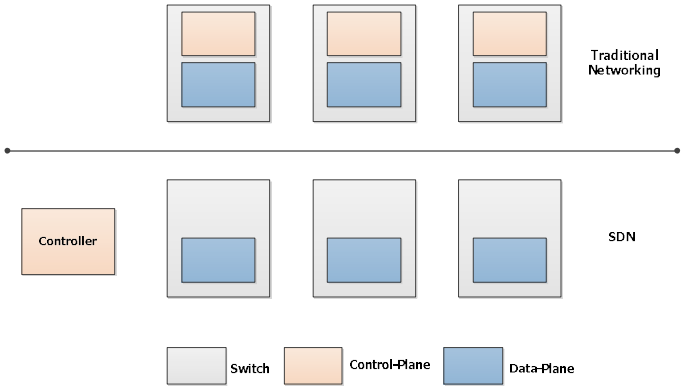

Figure 2: Illustration of Traditional Networking and SDN.

SDN Architecture

As explained in previous section, SDN introduces an abstraction, where the Control-Plane and Data-Plane are decoupled. In SDN, the forwarding state (in Data-Plane) is controlled by the SDN controller. This means the control functionality is removed from network devices. The forwarding decisions are Flow-Based, in the SDN/OpenFlow concept, a flow is a sequence of packets between a source and a destination [6]. The flow is defined by a set of packet field values acting as a match (filter) criterion and a set of actions (instructions). Each packets of flow receives identical service policies at the forwarding devices [6], [8], [9].

OpenFlow Protocol [10] is a mechanism that allows the Control-Plane Layer to communicate with Data-Plane Layer. This protocol is a standard protocol for communication over North-Bound and South-Bound APIs. The SDN controller configures the forwarding devices with the help of OpenFlow Configuration and Management Protocol (OF-Config) and the Open vSwitch Database Management Protocol (OVSDB). Besides that, OF-Config and OVSDB also act as specific extensions to OpenFlow. Send Packet Out, Packet Received and Modify Fowarding Table are examples of messages that are exchanged between the switches and the SDN controller, that defines by the OpenFlow Protocol.

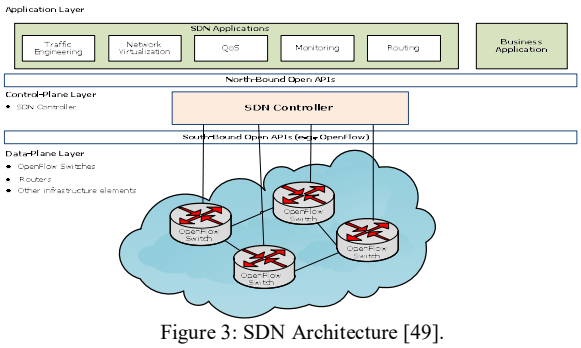

Figure 3: SDN Architecture [49].

The SDN architecture is divided into three (3) layers: Application Layer/ Management-Plane, Control-Plane Layer, and Data-Plane Layer, as illustrated in Figure 3.

2. SDN Security Threats and Countermeasures

The The researchers in [11] classify the threats and its countermeasures into three groups according to the SDN layer (Application, Control-Plane, and Data-Plane Layers) at which the corresponding attacks occur.

A. Principles for Securing SDN

The Open Networking Foundation in its article entitled Principles and Practices for Securing Software-Defined Networks has proposed 8 principles for securing SDN. The proposed security principles are for all protocol, components and interface of SDN architecture.

B. Possible Attack Points in SDN Architecture

The attack in SDN can occur through the central location for management, which is the Controller. Besides, it can also occur through the Switches' flow tables that consist of information related to switching, routing, and access control. North-Bound Interface, South-Bound Interface, and East-West Interface also can be attacked by tricking the controller to allow malicious applications to join the network and communicate with the controller, the network, and its traffic. Besides that, the channel between the controller and the switches can be also attacked.

3. SDN Applications

Rural Connections [7] is one of the SDN application. The main problem faced by the network administrator for deploying network technology in the rural area are sparse population and resource constraints. The separation of the construction of the network and the configuration of the network in SDN enables the rural infrastructure deployment business to be done in rural environments and the Internet Service Provider (ISP) business to be done remotely in cities. Through SDN the management of rural networks are not necessary to be done in the rural areas.

Date Centers Upgrading is another application of SDN. The researchers in [19] have discussed the use of SDN concepts for solving the problems faced in cloud computing services, specifically in Data Center Network. Data centers are an integral part of many companies [59]. Google using SDN technology to interconnect its large number of data center is located at a different geographical area to ensure the data can be provided quickly when requested. Through OpenFlow, the switches can be managed from a central location [30]. It helps Google to improve operational efficiency [29] and reduce

the management costs.

The researchers in proposed a network infrastructure based on OpenFlow that can be used to interconnect data center networks. The proposed network infrastructure improves the latency by moving the workload to underutilized networks.

VMware NSX is a network virtualization platform, VMware NSX is SDN-based. It simplifies the network management since through NSX the network administrator does not need to deal with VLANs and complex sets of firewall rules. Besides that, NSX also provides network segmentation where each virtual network uses their own address space and this network is also isolated from the other virtual network.

SDN provides a solution for the challenges faced by Data-Center Networks (DCNs). According to the research done by the Enterprise Strategy Group (ESG), the Data-Center Networks system has undergone rapid change due to aggressive alliances in data centers, progressive use of virtualization technology and wide deployment of web applications [16].

Vello system is SDN-based. It uses OpenFlow standard to provide a basic set of management constructs in order to enable value-added capabilities for data center local and wide area networks [17]. Besides, Vello systems also allows a unified control of the global cloud for WAN resource optimization. The SDN-based Vello systems have proposed open and scalable network virtualization solutions in order to connect the storage and compute resources in data centers within public and private cloud platforms.

Switching with In-Packet Bloom Filters (SiBF) is another example of DCN architecture which has been proposed by the researchers in [18]. SiBF introduces Rack Managers that acts as OpenFlow Controllers that provides scalability and maintain the globally required state to provide fault-tolerance in the DCN. It proposes scalable forwarding services that is self-configurable which does not require endpoint modifications. It also can be seen as another choice of forwarding service in parallel to other Ethernet flavours. This proposed data-center architecture also uses encoding technique to provide load – balancing services.

The researchers in [20] introduce Energy-Aware Data-Center Architecture based on an OpenFlow platform. The main purpose of introducing the architecture is to achieve the concept of "Green Data Center" in DCN. It provides guidelines to learn about the energy consumption in DCN elements such as switches, links and ports. The DC power and energy consumption is measured in three important places, which are: at the utility meter, at the plug and at the hardware compute load inside the box of the IT equipment itself [22]. Through the proposed architecture, the traffic load can be captured and monitored. This makes the process of analyzing the connection between the traffic load and energy consumption to be done. Through the proposed architecture also, the different energy-aware topology optimization and routing algorithms can be deployed and analyzed. The minimum power required by a network topology can be estimated through this proposed architecture.

The researchers in [21] proposed OpenFlow Switch Controller (OSC) in DCN to minimize the power consumption of switches in DCN by minimizing the influence of carbon emissions in the DCs. This proposed OSC able is to work at different power saving modes since it receives control messages from: OpenFlow controller and controls switches and links which is based on the programmable controller. OSC together with a NetFPGA based OpenFlow switch [78] can be used for power-aware networking research. The proposed OSC also helps reduce the configuration time of

network elements.

Next, the SDN Technology is also used to improve the DCN Metrics. The researchers in [23] introduced an integrated way to monitor path load metric to administrate link layer multipathing and congestion control. The integrated congestion control uses the link load information in edge switches that directly inform sources to control traffic admission. The proposed method is integrating Dynamic Load Balancing MultiPath (DLBMP) scheme with congestion control, where the routing intelligence is decoupled from data transmission (SDN techniques) to minimize overhead and to speed up the update process. Through the proposed methods, the DCN will experience loss-less delivery and data sources which can respond rapidly to congestions. Besides, the network throughput is also improved with fine flow differentiation mechanism.

SDN technique is also used to deal the header redundancy problem in DCs, the researchers in [24] introduced "Scissors" that has capability to make changes on packet header in order to decrease DC traffic and network power consumption. From their investigation, header redundancies add up to 30-40% of DC traffic which effect on latencies and complexity of processing which can increase power consumption in DC. Through Scissors the redundant header information is replaced with a shorter tag called Flow ID and packets that have the same flow. It is part of a group in the same ID. Hence, it improves network delay and power gains.

The researchers in [25] introduced SDN-based network solution to improve DCN and deployed it to a multitenant experiment. Through the proposed prototype, the multiple OpenFlow switches are managed by the central controller and the responds to network updates is based on APIs. This makes the configuration update process become simpler. Through SDN-based network solution, the DCN strategies have fulfilled the cloud service provider needs which are: multi-tenant, low-cost, flexible, easy to operate and configurable.

Multiple data centers are placed in different geographical locations. CrossRoads [26] is a network fabric that facilitates live and offline virtual machine migration across multiple data centers. CrossRoads is OpenFlow-based. It provides support for East-West and North-South traffics for virtual machine migration. East-West traffic is used for virtual machine migration within data centers while, North-South traffic is used for virtual machine migration of the external clients.

Next, the researchers in [27] introduced software middleware solution which is known as Network Infrastructure as a Services (IaaS). It is OpenFlow-based. Through this proposed solution the connectivity interruption of virtual machines during the migration process is minimized. It also supports live virtual machines migration between different DCNs.

Networking as a Service is a Cloud-based network architecture which is implemented by using OpenFlow Protocol [28]. The proposed architecture evaluates the provision, delivery and consumption of Networking as a Service. It is composed of the NRP-network resource pool, the NOI-network operation interface, the NRE-network run-time environment (Responsible for billing, resource allocation and reliability guarantee in case of network failure), and NPS-the network protocol service. The cloud-based network architecture consists of Control-Plane layer where switching and routing process occur and Network Data-Plane layer where packet forwarding activities are conducted.

Software Defined Internet Exchange (SDX) [40], is another application of SDN. The

SDX capabilities allows two networks peer only for streaming video traffic which known as application- specific peering. Besides that, SDX has also created new programming abstractions which allow participating networks to develop/run the application that is able to behave correctly when it interacts with the border gateway protocol and does not interfere with each other . By deploying SDN at Internet Exchange Points (IXPs), it can perform many different actions on packets based on multiple header fields that enable inbound traffic engineering and wide-area server load balancing.

OpenRoads (OpenFlow Wireless) is a program for innovation implementation of services for the wireless network. It creates open program to explore different mobility solutions, routing protocol and network controllers. Through OpenRoads the researcher is able to control the data path using OpenFlow and handle the configuration of the device using Simple Network Management Protocol (SNMP). The control abilities from OpenFlow and SNMP makes the OpenRoads easily manage the different wireless technologies, for example, WiFi and WiMAX. The researchers in were truly inspired by OpenRoads. They try to resolve specific needs and challenges to deploy software defined cellular network.

The researchers in proposed Software Defined Optical Network (SDON) architecture and QoS-Aware Unified Control Protocol for optical burst switching in OpenFlow-Based Software-Defined Optical Network. The main function of this architecture is to improve QoS for a different type of traffic. The effectiveness of the proposed protocol was evaluated by using the conventional GMPLS-Based distributed protocol. This proposed protocol successfully improves the QoS for a different type of traffic.

The researchers in developed OpenFlow-Based update mechanisms to support high-level abstractions. The main point creating a set of high-level abstractions is to enable the administrator to update the whole network and to make sure each packet which crosses the network is processed by single fix global network configuration. This is because changeable configuration can cause security flaws and performance disruptions.

OpenRadio [41] provides declarative programming interfaces through programmable wireless Data-Plane which gives flexibility at the Physical Layer and MAC layers. Besides that, OpenRadio provides a modular interface that has the capability to execute traffic subsets using different protocols like WiFi and 3GPP LTE-Advanced.

Odin [42] introduces programmability in enterprise wireless local area networks (WLANs) through SDN concepts. WLANs must support authentication, mobility, load balancing and interference management. Through Odin the admin can implement enterprise WLAN services as a network application. It builds access point abstraction which simplifies client managements. Table 6 summarized the SDN applications.

4. Benefits of SDN

Rural Via the centralization of the network controller, the SDN forwarding devices (switches) becomes simpler and cheaper compared with the traditional network devices. The network management and configuration is also simplified [5]. Compared with the

current network architectures SDN can be reconfigured faster to respond the new business requirements [43]. Through SDN the network performance is improved globally [44].

Besides that, any new application, protocols and policies can be easily implemented through an application running on the controller which controls the forwarding devices via well-defined APIs such as OpenFlow Protocol [5], [12]. The researchers in [45] introduces an OpenFlow controller handling IP multicast that deployed in Control-Plane, without making any changes to the forwarding devices the control software installs the forwarding entries in the switches based on the multicast application, since the OpenFlow switches supports the forwarding operations need. Should the need arise, as in the case if the protocol needs other operations that not provided by OpenFlow specifications, the OpenFlow Data-Plane needs to be upgraded. FLARE [46] is the solution for programmable Data-Plane.

SDN also has the ability to provide network virtualization via tools such as FlowVisor or OpenVirteX [5], [13]. Network virtualization is the process to combine hardware and software network resources and functionality into single virtual network, where SDN allows the network provider to integrate virtual and physical environments [47], [48]. Through network virtualization and by installing appropriate rules, a controller application can specify the SDN switches functionality widely, for example, firewalling, network address translation, and load balancing [5].

Besides that, all applications can take actions from any part of the network. All applications in the network have global network view. This means all application are able to access the same network information. The integration between different application also becomes simpler (for example load balancing and routing applications can be combined sequentially) [6], [14].

SDN enables innovation, it allows organizations to rapidly deploy new types services and applications that can provide new income streams and more value from the network because SDN introduces orchestration that enables a large number of devices to manage automatically with higher network resource utilization rates and lower capital costs. SDN also reduces the need to buy ASIC-Based networking hardware and purpose-built.

5. Limitation of SDN

Logical centralization of control does not necessarily imply physical centralization because this can result in scalability and reliability problems. In terms of network design, if there is only one centralized controller, it can cause a single point of failure. To address this issue, researchers have proposed physically distributed SDN controllers, such as the Onix system [5].

Since many controllers can manage the same flow tables, the consistency of flow table is an issue. Besides that, the flow table capacity is limited. This can cause poor network performances because the switches, and the controller depend on each other. The channels between the switches and the controller are vulnerable to attack for example DoS Attack, which has been discussed in section 6.

6. Simulation/Emulation Tools

Mininet emulator [31] is used to emulate the OpenFlow networks. It is an open source that has permits to deal with SDN networks. Mininet allows the whole OpenFlow network to be emulated on a single machine by creating a realistic virtual network, running real kernel, switch and application code using single command [33]. Mininet is able to create SDN elements, customize them and share with other network and perform interactions [38] (E.g: Hosts, Switches, Controllers, and Links). Through Mininet, it is possible to create a customized network by using Python APIs or directly build some simple network topologies through the Command-Line Interface (CLI). Besides that, Mininet also can work with several different SDN controllers, for example, Floodlight controllers. It allows researchers to rapidly test new algorithms and protocols in a built-in environment since the performance of the emulator depends on the available resources supplied by the host. Mininet CE and SDN Cloud-DataCenter are extensions to Mininet to enable wide-scale simulations. The main goal of Mininet CE is to create upper-level software over Mininet which can combine separate instances of Mininet into one Cluster.

NS-2 with OpenFlow Software Implementation Distribution (OF-SID) [39] and NS-3 [32] network simulator supports OpenFlow switches within its environment. NS-3 enables customization of the tracing output without rebuilding the simulation core since it uses a callback-based design which separates trace sources from trace sinks.

EstiNet [34], [35] is an OpenFlow network simulator and emulator. One thing that makes EstiNet different from other network simulator/emulator is that it has the capability to enable the unmodified real application to run on simulated hosts since it uses kernel re-entering methodology. This makes the simulation results of EstiNet simulator accurate and equal with the result obtained from an emulator. Besides that, EstiNet uses its own simulation clock to manage the simulation events execution order. Because of the kernel re-entering simulation methodology is used in EstiNet, the real-life OpenFlow controller programs such as NOX/POX [16], Floodlight [36] and Ryu [37] can directly run on a simulated host to control simulated OpenFlow switches without to make any changes. EstiNet also supports multiple hosts through a single kernel. It is also able to simulate multiple OpenFlow switches.

Mininet HiFi is a Container-Based Emulation (CBE). It modifies the original Mininet architecture by adding a process for performance segregation, provisioning, and monitoring for performance fidelity. The original Mininet uses lightweight, OS-Level Virtualization to emulate network links and switches that follow the Imunes system approach. Mininet-HiFi is suitable for experiments that benefit from flexible routing and topology configuration

6. 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."

[2] Open Networking Foundation, "Software-Defined Networking: The New Norm for Networks [white paper]," 2012.

[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] Traditional vs Software Defined Networking. <http://www.mavenspire.com/blog/traditional-vs.-software-defined-whats-the-difference>.

[5] F. Pakzad, M. Portmann, W. L. Tan, and J. Indulska, "Efficient topology discovery in OpenFlow-based Software Defined Networks," Comput. Commun., vol. 77, pp. 52–61, 2016.

[6] D. Kreutz, F. M. V Ramos, P. Verissimo, C. E. Rothenberg, S. Azodolmolky, S. Member, and S. Uhlig, "Software-Defined Networking : A Comprehensive Survey," Proc. IEEE, vol. 103, no. 1, pp. 14 – 76, 2015.

[7] F. Hu, Q. Hao, and K. Bao, "A Survey on Software Defined Networking (SDN) and OpenFlow: From Concept to Implementation," IEEE Commun. Surv. Tutorials, vol. 16, no. c, pp. 1–1, 2014.

[8] P. Newman, G. Minshall, and T. L. Lyon, "IP switching-ATM under IP," IEEE/ACM Trans. Netw., vol. 6, no. 2, pp. 117–129, 1998.

[9] N. Gude, T. Koponen, J. Pettit, B. Pfaff, M. Casado, N. McKeown, and S. Shenker, "NOX: towards an operating system for networks," SIGCOMM Comput. Commun. Rev., vol. 38, no. 3, pp. 105–110, 2008.

[10] "OpenFlow - Open Networking Foundation", Opennetworking.org. [Online]. Available: https://www.opennetworking.org/sdn-resources/openflow. [Accessed: 18- Aug- 2016].

[11] Z. Shu, J. Wan, D. Li, J. Lin, A. V. Vasilakos, and M. Imran, "Security in Software-Defined Networking: Threats and Countermeasures," Mob. Networks Appl., no. JANUARY, pp. 1–13, 2016.

[12] R. Sherwood, G. Gibb, K. Yap, G. Appenzeller, M. Casado, N. Mckeown, and G. Parulkar, "FlowVisor: A Network Virtualization Layer," Network, p. 15, 2009.

[13] A. Al-Shabibi, M. De Leenheer, M. Gerola, A. Koshibe, W. Snow, and G. Parilkar, "OpenVirteX: A Network Hypervisor," Open Netw. Summit, pp. 1–2, 2014.

[14] M. Casado, N. Foster, and A. Guha, "Abstractions for Software-Defined Networks," Stanford Univ., vol. 1, no. 1, p. 8, 2013.

[15] N. Foster, M. J. Freedman, R. Harrison, C. Monsanto, M. Reitblatt, J. Rexford, A. Story, and D. Walker, "Language abstractions for software-defined networks," Work. Lang. Distrib. Algorithms, 2012.

[16] J. Oltsik, B. Laliberte, Senior Principal Analyst, and Senior Analysts, "ESG Brief IBM and NEC Bring SDN / OpenFlow to Enterprise Data Center Networks," no. January, pp. 1–5, 2012.

[17] OPTIMIZING CLOUD INFRASTRUCTURE WITH SOFTWARE-DEFINED NETWORKING.<http://www.margallacomm.com/downloads/VSI_11Q4_OPN_GA_WP_01_101 2_Booklet.pdf>.

[18] [In-packet Bloom filter based data center networking with distributed OpenFlow controllers. <http://www.dca.fee.unicamp.br/~chesteve/pubs/SiBF_IEEE_Globecom_MENS_2010.pdf>.

[19] M. Jammal, T. Singh, A. Shami, R. Asal, and Y. Li, "Software defined networking : State of the art and research challenges," Comput. NETWORKS, vol. 72, pp. 74–98, 2014.

[20] N. H. Thanh, P. N. Nam, T.-H. Truong, N. T. Hung, L. K. Doanh, and R. Pries, "Enabling Experiments for Energy-Efficient Data Center Networks on OpenFlow-based Platform," IEEE Conf. Publ., no. 2, pp. 239–244, 2012.

[21] T. H. Vu, P. N. Nam, T. Thanh, N. H. Thanh, L. T. Hung, L. A. Van, N. D. Linh, and T. D. Thien, "Power Aware OpenFlow Switch Extension for Energy Saving in Data Centers," IEEE Conf. Publ., no. Atc, pp. 309–313, 2012.

[22] J. Koomey, J. R. Stanley, and K. G. Brill, "Four Metrics Define Data Center ' Greenness ,'" 2007.

[23] S. Fang, Y. Yu, C. Heng Foh, and K. Mi Mi Aung, "A Loss-free Multipathing Solution for Data Center Network : using Software-defined Networking Approach," IEEE Journals Mag., vol. 49, no. 6, pp. 2723 – 2730, 2013.

[24] K. Kannan and S. Banerjee, "Scissors : Dealing with Header Redundancies in Data Centers through SDN," IEEE Conf. Publ., pp. 295–301, 2012.

[25] L. Sun, K. Suzuki, C. Yasunobu, Y. Hatano, and H. Shimonishi, "A network management solution based on OpenFlow towards new challenges of multitenant data center," IEEE Conf. Publ., pp. 1–6, 2012.

[26] V. Mann, A. Vishnoi, K. Kalapriya, and S. Kalyanaraman, "CrossRoads : Seamless VM Mobility Across Data Centers through Software Defined Networking," IEEE Conf. Publ., pp. 88 – 96, 2012.

[27] B. Boughzala, R. Ben Ali, M. Lemay, Y. Lemieux, and O. Cherkaoui, "OpenFlow Supporting Inter-Domain Virtual Machine Migration," IEEE Conf. Publ., pp. 1–7, 2011.

[28] T. Feng, J. Bi, H. Hu, and H. Cao, "Networking as a Service : a Cloud-based Network Architecture," vol. 6, no. 7, pp. 1084–1090, 2011.

[29] S. Jain, A. Kumar, S. Mandal, J. Ong, L. Poutievski, A. Singh, S. Venkata, J. Wanderer, J. Zhou, M. Zhu, J. Zolla, U. Hölzle, S. Stuart, and A. Vahdat, "B4: Experience with a Globally-Deployed Software Defined WAN," Proc. ACM SIGCOMM 2013 Conf. SIGCOMM - SIGCOMM '13, p. 3, 2013.

[30] C. Baker, A. Anjum, R. Hill, N. Bessis, and S. L. Kiani, "Improving Cloud Datacentre Scalability , Agility and Performance using OpenFlow," Intell. Netw. Collab. Syst. (INCoS), 2012 4th Int. Conf., pp. 20 – 27, 2012.

[31] B. Lantz, B. Heller, and N. McKeown, "A network in a laptop: rapid prototyping for software-defined networks," … Work. Hot Top. Networks, pp. 1–6, 2010.

[32] OpenFlow switch support-ns-3 vns-3-dev documentation. <https://www.nsnam.org/docs/release/3.13/models/html/openflow- switch.html/>.

[33] Mininet. <http://mininet.org/>.

[34] S. Wang, "Comparison of SDN OpenFlow Network Simulator and Emulators : EstiNet vs . Mininet," IEEE Symp. Comput. Commun., pp. 1–6, 2014.

[35] S.-Y. Wang, C.-L. Chou, and C.-M. Yang, "EstiNet OpenFlow Network Simulator and Emulator," IEEE Journals Mag., vol. 51, no. 9, pp. 110–117, 2013.

[36] Project Floodlight. <http://www.projectfloodlight.org/floodlight/>.

[37] Component-Based Software Defined Networking Framework. <https://osrg.github.io/ryu/index.html>.

[38] R. L. S. de Oliveira, C. M. Schweitzer, A. A. Shinoda, and L. R. Prete, "Using Mininet for Emulation and Prototyping Software-Defined Networks," IEEE Conf. Publ., pp. 1–6, 2014.

[39] J. Sommers, P. Barford, and M. Gupta, "Fast , Accurate Simulation for SDN Prototyping," pp. 1–6, 2013.

[40] M. Caesar, D. Caldwell, N. Feamster, J. Rexford, A. Shaikh, and J. van der Merwe, "Design and Implementation of a Routing Control Platform," Second Conf. Symp. Networked Syst. Des. Implement., pp. 15–28, 2005.

[41] M. Bansal, J. Mehlman, S. Katti, and P. Levis, "OpenRadio: A Programmable Wireless Dataplane," HotSDN, pp. 109–114, 2012.

[42] L. Suresh, J. Schulz-Zander, R. Merz, A. Feldmann, and T. Vazao, "Towards Programmable Enterprise WLANs with Odin," Proc. first …, pp. 1–5, 2012.

[43] M. Algarni, V. Nair, D. Martin, and S. Shirgaonkar, "Software-Defined Networking Overview and Implementation," pp. 1–11.

[44] W. Xia, Y. Wen, C. Foh, and D. Niyato, "A Survey on Software-Defined Networking," Surv. Tutorials, vol. 17, no. 1, pp. 27–51, 2015.

[45] D. Kotani, K. Suzuki, and H. Shimonishi, "A Design and Implementation of Openflow Controller Handling IP Multicast with Fast Tree Switching," Proc. - 2012 IEEE/IPSJ 12th Int. Symp. Appl. Internet, SAINT 2012, pp. 60–67, 2012.

[46] A. Nakao, "FLARE Open Deeply Programmable Network Node Architecture," 2012.

[47] "Network Virtualization - Gartner IT Glossary", Gartner IT Glossary, 2012. [Online]. Available: http://www.gartner.com/it-glossary/network-virtualization. [Accessed: 20- Aug- 2016].

[48] Hewlett-Packard Development Company, L.P., "Software-Defined Networking and Network Virtualization[Technical white paper]," 2014.

[49] I. F. Akyildiz, A. Lee, P. Wang, M. Luo, and W. Chou, "A Roadmap for Traffic Engineering In Software Defined Networks," Comput. Networks, vol. 71, pp. 1–30, 2014.

[50] R. Masoudi and A. Ghaffari, "Software defined networks: A survey," J. Netw. Comput. Appl., vol. 67, pp. 1–25, 2016.