QALAAI ZANIST JOURNAL
A Scientific Quarterly Refereed Journal Issued by Lebanese French University – Erbil – Kurdistan – Iraq
Vol. (2), No. (2), April 2017
Speical Issue : The 1st International Conference on Information Technology (ICoIT'17)
ISSN 2518-6566 (Online) - ISSN 2518-6558 (Print)

An Overview of Software Defined Environment

Ali Dheyaa Jalil
Networks Engineering Department, College of Information Engineering, Al-Nahrain University - Iraq
ali.dheyaa@coie-nahrain.edu.iq
Dr. Abdulkareem A. Kadhim
Computer Engineering Department, College of Engineering, Al-Nahrain University -Iraq
abdulkareem.a@coie.nahrainuniv.edu.iq

ARTICLE INFO

Article History:

Received: 17 March 2017

Accepted: 1 April 2017

Published: 10 April 2017

DOI:

10.25212/lfu.qzj.2.2.14

Keywords: *Networks, Software Defined Environment, Software Defined DataCenters, Software Defined Network, OpenFlow protocol, Software Defined Storage, Software Defined Compute.*

ABSTRACT

Datacenters are a complex environment consisting of network equipment, server hardware and storage systems. The rapid evolution in Information Technology applications and the widely used services and datacenters provided by different vendors make them a big challenge. Traditional datacenters are managed by administrators manually. The growth of datacenters thus reveals problems in managing and controlling. A new technology called Software Defined Environment (SDE) is introduced to solve such problems. SDE solution separates the controlling functions of the datacenters from its physical components and performs such operations in a centralized controller. The present work introduces an overview of SDE and its features to solve common problems in datacenters such as the controlling problems and the management problems in adding an extra physical hardware. Suitable scenarios are suggested with the assistance of Software Defined Network (SDN) and its useful features. Possible tools and resources to implement SDE are also presented.

1. INTRODUCTION

Finding a standard way for controlling and managing the infrastructure of datacenters is an outstanding demand. Thus the separation between the control plane and data plane is proposed. Software Defined Environment (SDE) is a new approach where the software based management and configuration of all resources, based on the workload, can be achieved. Software Defined Network (SDN) is considered as the main component of SDE. The latter is a term used by IBM for its "software-defined everything" vision [1]. A part of SDN, the umbrella of SDE technology covers SD Compute (SDC), SD DataCenters (SDDC) and SD Storage (SDS). In SDE environment, intelligent software is responsible of managing and controlling the network, storage and/or datacenter infrastructure instead of their hardware components [1]. L. MacVittie discussed the idea of Software Defined and compares it with Object Oriented idea where the implementation of “the data layer” is abstracted from the interface “the control layer” [2]. So it is easy to make implementation changes with modifications in the interface.

The idea of software defining the datacenter components is discussed by several works. V. Törhönen has designed his own SDDC by utilizing multi-vendor hardware [3]. His proposed datacenter is based on VMware as the virtualization layer, HP server hardware as the computing component, NetApp storage, and HP and Cisco networking equipment. All the components provide an Application Programming Interface (API) that is used to control the datacenter [3]. VMware in 2015 introduced an architecture using their software components to create an SDDC as follows [4]:

- VMware vSphere: as a Server virtualization platform (instead of Software defined compute).
- VSAN and vSphere: VSAN provides distributed storage capabilities and vSphere is responsible of the Policy-based storage capabilities.
- VMware NSX: NSX provides a platform for Network virtualization (instead of SDN).

Until the present time there is no emulator designed to emulate SDE, SDS or SDDC. Mininet is one of the widely used tools for emulating SDN [5]. Mininet is an *emulation tool* which can build a virtual SDN network consists of hosts, switches, controllers, and links. Mininet switches support OpenFlow protocol for more flexible customized routing and SDN applications.

Darabseh et. al. customized Mininet to emulate several Software Defined experimental frameworks [6-9]. They customized Mininet's elements by adding storage parameters, tables, and functions to the hosts, switches, and controllers, respectively to build SDS_host, SDS_switches and SDS_controller. They proposed an SDsecurity framework in the same way. They also integrated their SDS and SDSecurity frameworks into a software defined secure storage. SDDC is built by integrating the SDN (using Mininet), SDS, SDSecurity frameworks and with another block; the Software Defined compute (SDC) that is considered as part of the virtualization and the computing resources.

The remaining parts of the present paper are; part-2 covers the concept of traditional network architecture, while part-3 deals with SDE based network with its useful components (SDC, SDS, and SDN). Part-4 gives an overview of the available tools for SDE. Finally, the main concluding remarks are given in part-5.

2. TRADITIONAL NETWORK ARCHITECTURE

In this section, the architecture of traditional networks and its main differences compared to SDN are introduced. There are some difficulties to manage the traditional networks. they consist of different types of equipment such as routers, switches and middleboxes (e.g. intrusion detection systems, network address translators (NAT), load balancers and firewalls). Routers and switches run a control software that is very complex and usually locked and proprietary. Usually, such software implemented network protocols which subject tests for standardization and interoperability for years. Administrators can configure each device with its configuration interfaces. These interfaces are different from one vendor to another and also among the same vendor products. This leads to increase the complexity, slow innovation and inflate both the capital cost and the cost running the network [10].

Datacenter is a facility used to house servers and computer hosts which are connected using dedicated switches and links. Current Datacenters may contain huge number of hosts with bandwidth and delay requirements [11]. The topology of the network has an important effect on the agility and reconfigurability of the datacenter infrastructure to respond to changing application demands and service requirements. Basic tree topology consists of

QALAAI ZANIST JOURNAL
A Scientific Quarterly Refereed Journal Issued by Lebanese French University – Erbil – Kurdistan – Iraq
Vol. (2), No. (2), April 2017
Speical Issue : The 1st International Conference on Information Technology (ICoIT'17)
ISSN 2518-6566 (Online) - ISSN 2518-6558 (Print)

either two or three levels of switches or routers, with the host as leaves. In 3-tier topology; at the root there is the core tier, at the middle there is the aggregation tier, and at the bottom there is the edge tier where the switches are connected to the servers. In a 2-tier topology, the aggregation tier does not exist [12]. A 3-tier tree topology is shown in Figure-1.

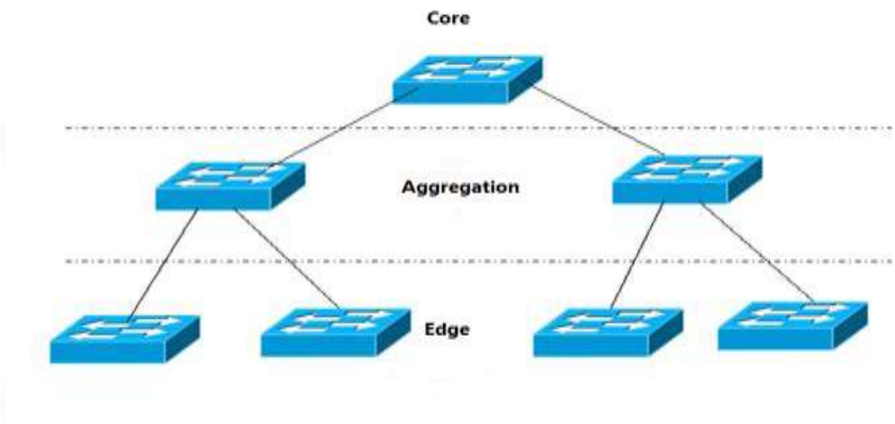

FIGURE 1. 3-Tier datacenter tree topology

A more complex type of tree topology called Fat Tree topology is used by different works [12, 13]. In Fat Tree a large number of routers and switches are used where the traffic is forwarded based on the limited view of each network device. This networking model is unable to meet the proper flexibility required with the current massive data amounts. The coupling between the software and the hardware of the routers and the switches leads to an expensive scaling and complexity in maintenance as well [14].

3. SDE BASED NETWORK

In this section the Software Defined Environment (SDE) and its main components are presented. SDE is a general term which covers multiple different products, technologies and solutions. Defining what the needs are will determine the selection of correct components. These can be put in three main areas [1]:

3.1 CLOUD WORKLOAD ORCHESTRATION AND MANAGEMENT

It is the foundation of SDE environment. IBM SDE with OpenStack provides a platform for cloud computing and enables configuration of clouds, application workloads to each deployment topology along with the other SD technologies [1].

3.2 SOFTWARE DEFINED COMPUTE

SDC term is used to describe the functions of server virtualization technology. These serve as resources to virtual machines guests [1]. Hypervisors can be used as an SDC in the SDE. Examples of these hypervisors are KVM for Linux Operating Systems [15], Hyper-V for Microsoft [16], and ESXI vSphere from VMware [17].

3.3 SOFTWARE DEFINED STORAGE

SDS is new storage architecture for wide variety of data storage requirements based on a set of loosely coupled software and hardware components. IBM admitted that the SDS architectural concepts are new and evolving technological approaches. So it is not currently

QALAAI ZANIST JOURNAL
A Scientific Quarterly Refereed Journal Issued by Lebanese French University – Erbil – Kurdistan – Iraq
Vol. (2), No. (2), April 2017
Speical Issue : The 1st International Conference on Information Technology (ICoIT'17)
ISSN 2518-6566 (Online) - ISSN 2518-6558 (Print)

fully developed into mature, straightforward implementations [18]. The main idea of SDS is that all storage functions are embedded in software so there is no need for sophisticated, dedicated storage equipment. Thus, SDS will be more flexible than disk arrays or the traditional hard disks [1]. The main characteristic of the SDS reference architecture is the separation of the storage functions into two main layers:

- SDS control layer: This layer is responsible for managing the storage resources. It is based on a software that provides flexible, scalable, optimized, and rapid provisioning storage infrastructure capacity.
- SDS data layer: This layer is responsible for processing the data. The data layer represents the interface of the data storing hardware infrastructure. It provides range of data access possibilities and spans the current access methods such as block I/O (e.g. iSCSI) [18].

There is no specific emulator to emulate and test SDS. An experimental framework emulator for SDS systems was presented but with the use of SDN Mininet emulator [6]. By following such approach the Mininet SDN elements are customized to make their framework cover the following components:

- Host: where some extra SDS parameters are added to the hosts, such as no. of directories, no. of files inside each one, and the size of each file.
- Switch: where a table is created to store the updating hosts information about their status while keeping such information available to other hosts.
- Controller: where POX controller is used with some functions being added like "Get Number of Stored Files, isFull, used space, addFile and available space" [6].

3.4 SOFTWARE DEFINED NETWORK

SDN is a new network architecture where all the networking functions and tasks are performed by software in a controller (the control plane) separated from the network equipment (the data plane) like switches and routers. The limitations of the current network architectures are; scalability problem, different vendors interfacing problem, and configuration management problem. The idea of programmable network is older than SDN. Active Network has common visions with SDN, it showed a programmable networks, but with an emphasis on programmable data planes. Active network programming interface exposed the resources (e.g., processing, storage, and packet queues) on each node of the network. The problem with Active Network is that it has a lack in the clear use case and the incremental deployment path [10].

The interaction between the control and the data planes is organized by a protocol called the OpenFlow protocol, which is an open source protocol by the Open Networking Foundation (ONF) [19]. Each OpenFlow network consists of one or more Open flow switches; each switch with one or more flow table. Each port and flow table in the switch is associated with many counters that gather various statistics describing the events that the switch must handle. The flow tables and their entries are all created by the controller.

SDN Controllers are considered as the main part of SDN. It acts as strategic point of controlling, and managing the flow control to the routers or switches. The controller communicates via southbound API with the forwarding plane, the infrastructure layer. through the OpenFlow protocol. The controller communicates via northbound API with application layer to deal with various network applications [20]. Figure-2 shows the SDN architecture.

QALAAI ZANIST JOURNAL
A Scientific Quarterly Refereed Journal Issued by Lebanese French University – Erbil – Kurdistan – Iraq
Vol. (2), No. (2), April 2017
Speical Issue : The 1st International Conference on Information Technology (ICoIT'17)
ISSN 2518-6566 (Online) - ISSN 2518-6558 (Print)

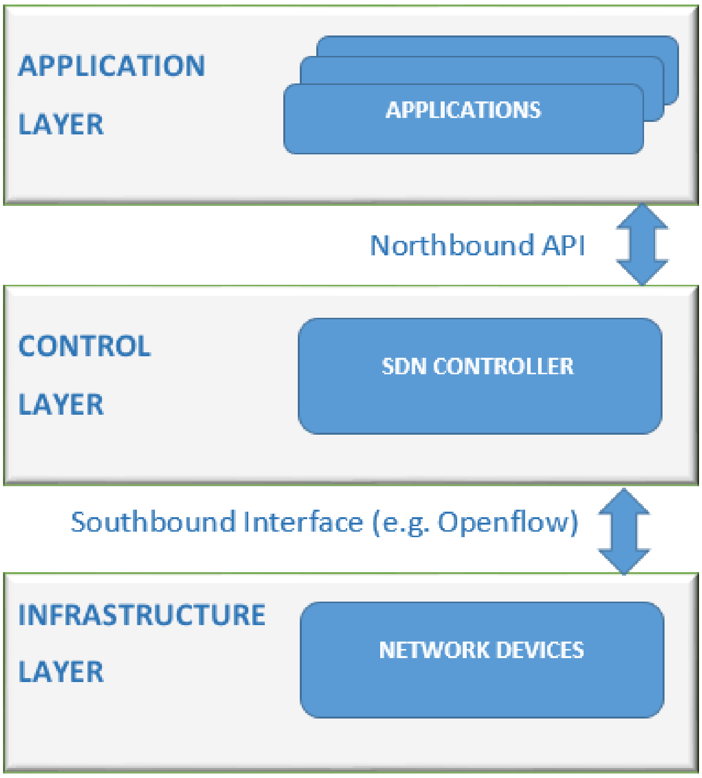

FIGURE 2. SDN Architecture

There are several emulators that can emulate the SDN networks, such as Mininet [5], ns-3 and Estinet [21]. Mininet is widely used by SDN researchers because it is simple, open source and written in Python programming language. It is based on Ubuntu Linux distribution. Mininet consists of three main components: the host, which sends and receives the packets, the switch, which stores the rules for forwarding the packets to their destinations, and the controller, which is responsible for the operations of controlling and managing the network [5]. Mininet also gives the ability to connect the SDN network to a remote controller. There are several controllers that can be connected to Mininet, such as NOX, POX, Beacon and Floodlight. NOX is the first SDN controller developed at Nicira networks [22]. The main difference between NOX and POX is that POX is based on Python programming language, while NOX is based on C^{++} programming language [23]. Beacon and Floodlight are written in Java programming language [24,25]. The Mininet emulator has a problem of its inability to handle large-scale networks. MaxiNet is created as an extended version of Mininet for large scale networks. Its main idea is to build a distributed emulation environment [26].

4. COMMON TOOLS FOR SDE

The available tools for SDE are introduced in the following;

4.1 SDE MANAGEMENT TOOLS

The OpenStack project started in 2011 as the main component in cloud management [27]. It is developed by the large number of companies and developers all around the world. The OpenStack project used Apache 2.0 license and runs on variety of Linux operating systems. OpenStack is software that controls the resources of datacenter such as the compute, storage, and networking resources. These resources are managed through the OpenStack API. OpenStack can work with well-known enterprise and open source technologies that make

QALAAI ZANIST JOURNAL
A Scientific Quarterly Refereed Journal Issued by Lebanese French University – Erbil – Kurdistan – Iraq
Vol. (2), No. (2), April 2017
Speical Issue : The 1st International Conference on Information Technology (ICoIT'17)
ISSN 2518-6566 (Online) - ISSN 2518-6558 (Print)

OpenStack as a good management tool for heterogeneous infrastructure [27]. It is worth to be noted here that an excellent programming proficiency is required when operating OpenStack API.

IBM suggested their tools to manage clouds and deploy orchestrated services for the client's cloud environment. One of these tools is IBM cloud manager, which is based on OpenStack, easy to use and to deploy cloud management software, with additional IBM enhancements and enterprise class support [1]. It also supports many different hypervisors, storage, and network solutions.

4.2 SDE COMPUTE TOOLS

Hypervisors can be used as an SDC in the SDE. The main advantage of SDC tools is that multiple servers can run on one physical server, which translates into more efficient use of resources. The main disadvantage here is that if there is a problem in this physical server, all the virtual servers will be affected. Examples of the hypervisors are:

- IBM PowerVM: This is a virtualization technology available on IBM POWER servers and can provide virtualized or physical server resources to the virtual machines. PowerVM enables dynamic, non-disruptive, manual or automatic resource relocation, such as dynamic CPU or RAM reassignment between different logical partitions [1].
- Hyper-V: This is a virtualization technology introduced by Microsoft [16]. It runs on x86 servers and supports Linux and Windows guest virtual machines. It is also supported by IBM Cloud Manager as one of the underlying cloud technologies.
- ESXI : This is a virtualization technology provided by VMware [17]. It runs on x86 servers and supports Linux and Windows guest virtual machines. It is also supported by IBM Cloud Manager as one of the underlying cloud technologies. The main drawback when using such tool is the difficulty in accessing it by the administrator directly from the working server. vSphere client software need to be installed on another host and used to access the ESXI.

4.3 SDE STORAGE TOOLS

In SDE, SDS tools run on the physical storage and provide the advantage of multiple virtual storage pools to the hosts. Different storage products and solutions can be used as in the following:

- IBM SAN Volume Controller (SVC) is a well-known and proven, robust storage solution. SVC can virtualize huge number of external disk arrays to provide common storage pool to hosts [1].
- IBM Storwize V7000 is a disk storage system which provides block storage to hosts. Storwize and SVC run the same code with the same features as SVC. It differs from SVC that it actually handles up to several hundreds of real physical disks, which can be gathered in RAID arrays and then provide common storage pool for physical or virtual servers [18].
- As mentioned, VSAN and vSphere can be used together as SDS tools, where VSAN can provide distributed storage capabilities and vSphere is responsible for the policy-based storage capabilities [4].

QALAAI ZANIST JOURNAL
A Scientific Quarterly Refereed Journal Issued by Lebanese French University – Erbil – Kurdistan – Iraq
Vol. (2), No. (2), April 2017
Speical Issue : The 1st International Conference on Information Technology (ICoIT'17)
ISSN 2518-6566 (Online) - ISSN 2518-6558 (Print)

All the above tools are based on the virtualization idea. One of the main disadvantages of such approach is that a more complex troubleshooting effort is required.

4.4 SDE NETWORK TOOLS

The IBM Software Defined Network for SDE has a set of capabilities. It provides integrated management of OpenFlow networks and Distributed Overlay of Virtual Ethernet (DOVE) networks. IBM Software Defined Network for Virtual Environments (SDN-VE) is software for networking implementation, based on the OpenDaylight open source platform. SDN VE creates virtual networks that are isolated from physical networks in the same way similar to how a virtual machine is separated from a physical host. SDN VE integrates with OpenStack Havana via the neutron network component of OpenStack [1].

Cisco Application Centric Infrastructure (ACI) is an SDN architecture proposed by Cisco [28]. It is a policy-based automation solution that supports a greater scalability through a distributed system, and a greater network visibility. These benefits are achieved through the integration of physical and virtual environments under one policy model for networking, servers, and storage. Cisco ACI is implemented only with Cisco Application Policy Infrastructure Controller (APIC) and Cisco Nexus 9000 series switches.

Most of the SDE tools, for instant the storage and server system tools, are vendor-based. The source code of these tools is usually unrevealed, so there is a difficulty to improve the system that uses such tools. The improvement is required to support the legacy compute and storage systems in datacenters. For network tools, there are many open source controllers. The users can develop their own application to manage and improve their networks. As an example for such applications is the user security policy.

5. CONCLUSION

Virtualization and Software Defined technologies can be considered as the basic elements of the next generation datacenters where all the components are either virtualized or Software Defined. In this paper, the idea of Software Defined Environment (SDE) is covered. SDE technology can be divided into three main technologies, SDN, SDC and SDS with an orchestration or management tool. Software Defined Networking (SDN) is the most well-known technology where the controlling functions are separated from switches and routers. There are different research topics linked to SDN such as traffic engineering and security. The idea of Software Defined Storage (SDS) is close to SDN idea where all storage functions are dealt with in software. Software Defined Compute (SDC) can be considered as the server virtualization which is implemented in a physical server to create multiple virtual servers. Hypervisors can be used to give the functionality of SDC.

It seems that there is no general emulation tool that covers all SDE functions and components. The advantages of SDE implementation in datacenters can be seen clearly in the management and scalability of such datacenters. In SDE, there is no need for specific-purpose devices. Thus SDE seems to be a very promising technology for implementation when there are multi-vendor devices and requirements for scalability. With SDE, great reductions in physical space and power requirements of datacenters will be achieved as a result of software defined and virtualization processes.

REFERENCES

[1] D. Quintero, W. Genovese, K. Kim, M. Li, F. Martins, A. Nainwal, D. Smolej, M. Tabinowski, A. Tiwary, *IBM Software Defined Environment*, IBM Red-Book, August 2015.

[2] L. MacVittie, "Stop conflating software-defined with software deployed", Devcentral 2013. https://devcentral.f5.com/articles/stop-conflating-software-defined-with-software-deployed (Accessed on 14 March 2017).

[3] V. Törhönen, "Designing a Software-Defined Datacenter", M.S. thesis, Tampere University of Technology, Tampere, Finland, December 2013.

[4] VMware Team, "Software-Defined Data Center", Technical White Paper, VMware Inc, 2015. (www.vmware.com)

[5] Mininet, Mininet Official Site, [online] 2017, http://mininet.org (Accessed on 14 March 2017).

[6] A. Darabseh, M. Al-Ayyoub, Y. Jararweh, E. Benkhelifa, M. Vouk, A. Rindos, "SDStorage: A software defined storage experimental framework", in *IEEE International Conference on Cloud Engineering (IC2E 2015)*, 2015.

[7] A. Darabseh, M. Al-Ayyoub, Y. Jararweh, E. Benkhelifa, M. Vouk, A. Rindos, "SDSecurity: A software defined security experimental framework", in *Third Workshop on Cloud Computing Systems, Networks, and Applications (CCSNA-2015)*, 2015.

[8] A. Darabseh, M. Al-Ayyoub, Y. Jararweh, E. Benkhelifa, M. Vouk , A. Rindos, " A novel framework for software defined based secure storage systems", *Simulation Modelling Practice and Theory* 2016.

[9] A. Darabseh, M. Al-Ayyoub, Y. Jararweh, E. Benkhelifa, M. Vouk, A. Rindos, "SDDC: A software defined datacenter experimental framework", in *Third International Conference on Future Internet of Things and Cloud (FiCloud-2015)*, Rome, 2015.

[10] N. Feamster, J. Rexford, and E. Zegura, "The Road to SDN: an intellectual history of programmable networks", *ACM SIGCOMM Computer Communication Review*, New York, April 2014.

[11] S. Seo, "Dynamic Traffic Engineering for Improving Energy Efficiency and Network Utilization of Data Center Networks", Ph.D. dissertation, Pohang University, Korea, December 2013.

[12] M. Al-Fares, A. Loukissas, and A. Vahdat, "A scalable, commodity datacenter network architecture", in *ACM SIGCOMM2008 Conference on Data Communication*, 2008.

[13] R. Niranjan Mysore, A. Pamboris, N. Farrington, N. Huang, P. Miri, S. Radhakrishnan, V. Subramanya, and A. Vahdat, "Portland: a scalable fault-tolerant layer 2 data center network fabric" *ACM SIGCOMM Computer Communication Review*, vol. 39, no. 4, 2009.

[14] S.A.Chasib ,Dr. A. A. Khadim. "Software Defined Network for Data Center Using OpenFlow Protocol ",*International Journal of Scientific Engineering and Technology* vol.5 ,no.12, December 2016.

[15] Kernel Virtual Machine, *KVM*, [online] 2008, https://www.linux- kvm.org, (Accessed on 14 March 2017).

[16] Microsoft Hyper-V, *Microsoft Official Site*, 2017. https://www.microsoft.com/en-us/cloud-platform/server-virtualization (Accessed on 14 March 2017).

[17] vSphere ESXI, *VMware Official Site*, 2017. http://www.vmware.com/ products/esxi-and-esx.html (Accessed on 14 March 2017).
[18] L. Coyne, M. Defiebre, P. Gilmer, S. Gopalakrishna, M. Lindström, B. Mann, G.Perilli, M. Portak, *IBM Software-Defined Storage Guide*, IBM RedBook, January 2016.
[19] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner,"OpenFlow: enabling innovation in campus networks" *ACM SIGCOMM Computer Communication Rev*iew, vol. 38, no. 2, 2008.
[20] What are SDN Controllers (or SDN Controllers Platforms)?, *SDx Central*, 2015. https://www.sdxcentral.com/sdn/sdn-controllers/ (Accessed on 14 March 2017).
[21] S. Wang, C. Chou, C. Yang, "EstiNet OpenFlow Network Simulator and Emulator", *IEEE Communications Magazine*, vol.51, no.9, pp.110-117, 2013.
[22] N. Gude, T. Koponen, J. Pettit, B. Pfaff, M. Casado, N. Mxkeown, "NOX: Towards an Operating System for Networks" *ACM SIGCOMM Computer Communication Review*, Vol. 38, No. 3, 2008.
[23] M. McCauley, A. Al-Shabibi, "POX Wiki". https://openflow.stanford.edu/display/ONL/POX+Wiki (Accessed on 14 March 2017).
[24] D. Erickson, "What is Beacon?", [online] 2013, https://openflow.stanford.edu /display/Beacon/Home. (Accessed on 14 March 2017).
[25] Floodlight, *Project Floodlight official Website*, 2017. http://www.projectfloodlight.org/floodlight/ (Accessed on 14 March 2017).
[26] P. Wette, M. Draxler, A. Schwabe, F. Wallaschek, M. Zahraee, and H. Karl, "MaxiNet: Distributed emulation of software-defined networks", in *2014 IFIP Networking Conference*, Norway, June 2014.
[27] OpenStack, OpenStack Official Website, [online] 2017, https://www.openstack.org/ (Accessed on 14 March 2017).
[28] Cisco Application Centric Infrastructure, Cisco Official Website. http://www.cisco.com/c/en/us/solutions/data-center-virtualization/application-centric-infrastructure/index.html (Accessed on 17 March 2017).