ISSN(Online) : 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering
An ISO 3297: 2007 Certified Organization Vol.5, Special Issue 2, April 2017

An International Conference on Recent Trends in IT Innovations - Tec'afe 2017

Organized by

Dept. of Computer Science, Garden City University, Bangalore-560049, India

Software Defined Network, Controller Comparison

Saleh Asadollahi, Dr. Bhargavi Goswami, Dr. Atul M Gonsai

Department of Computer Science, Saurashtra University, Rajkot, India

Department of Computer Science, Garden City University, Bangalore, India

Department of Computer Science, Saurashtra University, Rajkot, India

ABSTRACT:There is an stagnant growth observed in the domain of Networks for the last few years even though technology is taking big leap as a whole. We checked the reason for that and researchers gave answers through papers and articles that there is requirement of change of Network Architecture from root to meet the fast growing technology and wide spread technological implementation in such a short time. To meet the requirements, a set of researchers' team started working on it and came up with the concept of SDN. The solution proposed by the multiple top universities worldwide was to bring drastically notable change in the domain of networks. With this paper, authors want to throw light upon the architecture, applications, controllers, comparison and its limitations to answer three questions of the beginners i.e what is SDN, Why it was introduced and How does it work.

KEYWORDS*:* SDN, OpenFlow, Controller, API, NOX, ONOS, Ryu, OpenDayLight.

I. INTRODUCTION

In the century of technology, all the devices are connected to each other via Internet. Users connect to this global network by more than one unit device that cause increasing the data which traveling through that. Data center where store users data are expanding. Big data, cloud computing and so on are inducing computer network for high transformation rate, high bandwidth, ubiquitous accessibility, mobility and dynamic management. Data centers need to keep updating their devices such as router and switches to acquit very well.

Equipments (data plane and control plane) which have built current network devices knot into single devices cause complexity, heterogeneity, difficult configuration, and reducing the optimization[1]. SDN[2][3] new approached in network campus evident by idea of separation of control plane (brain) from data plane, makes kinky devices to just simple forwarding devices and second provide programmability instead of configuration mode.

In this paper, we first explain the comparison between traditional and SDN network architecture in section 2. Further, we introduce SDN Elements in section 3 and SDN Applications in section 4. Detailed SDN controllers and its features are discussed in section 5 followed by Conclusion and references.

II. COMPARISON OF TRADITIONAL AND SDN NETWORK

Computer network are include of various number of devices such as router and switches among different vendors are responsible for transfer data through network. Traditional network devices include of two fundamental elements 1) data plane 2) control plane that reside into a unit devices. These two planes are vertically integrated where same planes are able to communicate horizontally with peer plane residing adjacent to each other in a same segment[4]. Data plane (forwarding layer) has various port and a table that meets and handles entry packets form one port to another port. When a packet enters to the devices, data plan digs into the packet header information, then looks up to the data plan table, find match record and depends on it take decision either forward, drop or if there is no yet any information reside send it to control plane. Control plane (software layer) is known as brain of devise depend to configuration and type of device know about topology, learning and collecting information, builds control plane table and send same table to data plane so data plane in depends of top layer would be able to transfer data. Figure 1 illustrate current devices architecture.

ISSN(Online) : 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering
An ISO 3297: 2007 Certified Organization Vol.5, Special Issue 2, April 2017

An International Conference on Recent Trends in IT Innovations - Tec'afe 2017

Organized by

Dept. of Computer Science, Garden City University, Bangalore-560049, India

Routing, traffic monitoring, server balance loading, access control list are a part of current concurrent network task that administrator need to performance high policy by using low level or pre-defined commands.

The fact is each device has its own control plane where network manager needs to configure each individual device separately. Spend time, no view of entire network, cost and troubleshooting will be more knot in a scenario by various vendor of switches and routers. Another big challenge is network dynamically change and response automatically to various network events. Due to all problem mentioned above maintaining such a network is expensive and time consuming.

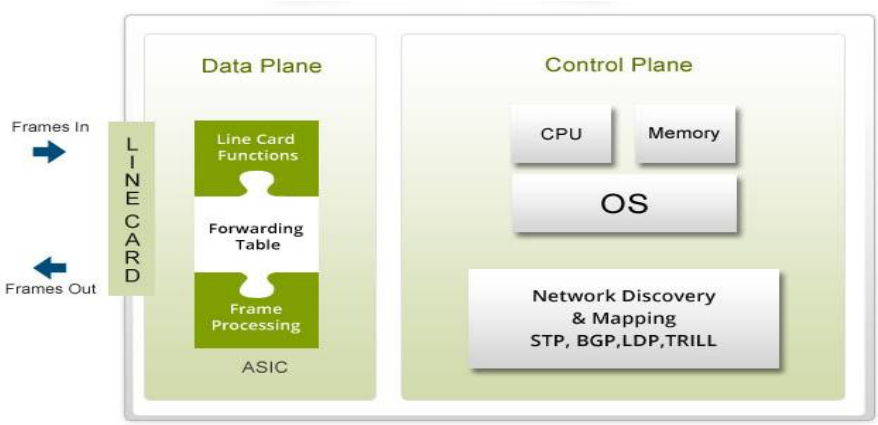

Fig. 1. Router component in traditional device

Additionally traditional networks route on hob-by-hob, its mean it is not provide overall view of network. On the other side lack of programmability devices prevent innovation, experience, and usage full potential of hardware make tickle researcher to find solution.

SDN is new approach how data packets flow through the network. In SDN architecture, data plan and control plane separated and control logic (brain) is implemented in logically centralized controller as traffic is actually controlled by intelligent controller. Figure 2 illustrate the SDN in high-level architecture.

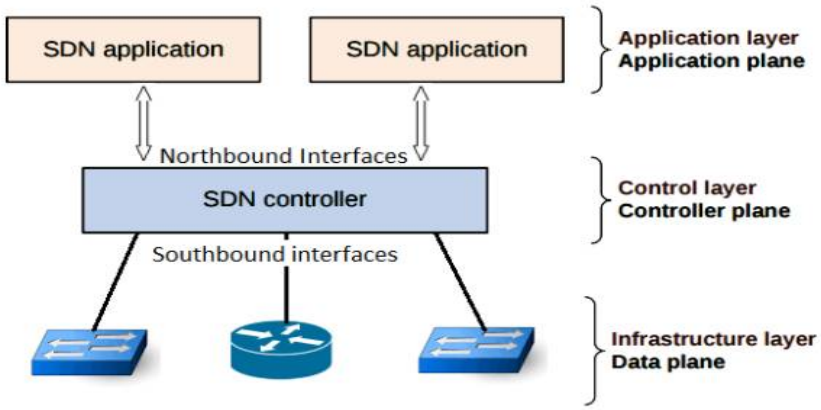

Fig. 2. Software Defined Networking high-level architecture

ISSN(Online) : 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering
An ISO 3297: 2007 Certified Organization Vol.5, Special Issue 2, April 2017

An International Conference on Recent Trends in IT Innovations - Tec'afe 2017

Organized by

Dept. of Computer Science, Garden City University, Bangalore-560049, India

This idea by decoupled data plane form control plane makes multipart current device to simple forwarding device. Figure 3 show different between traditional and SDN architecture.

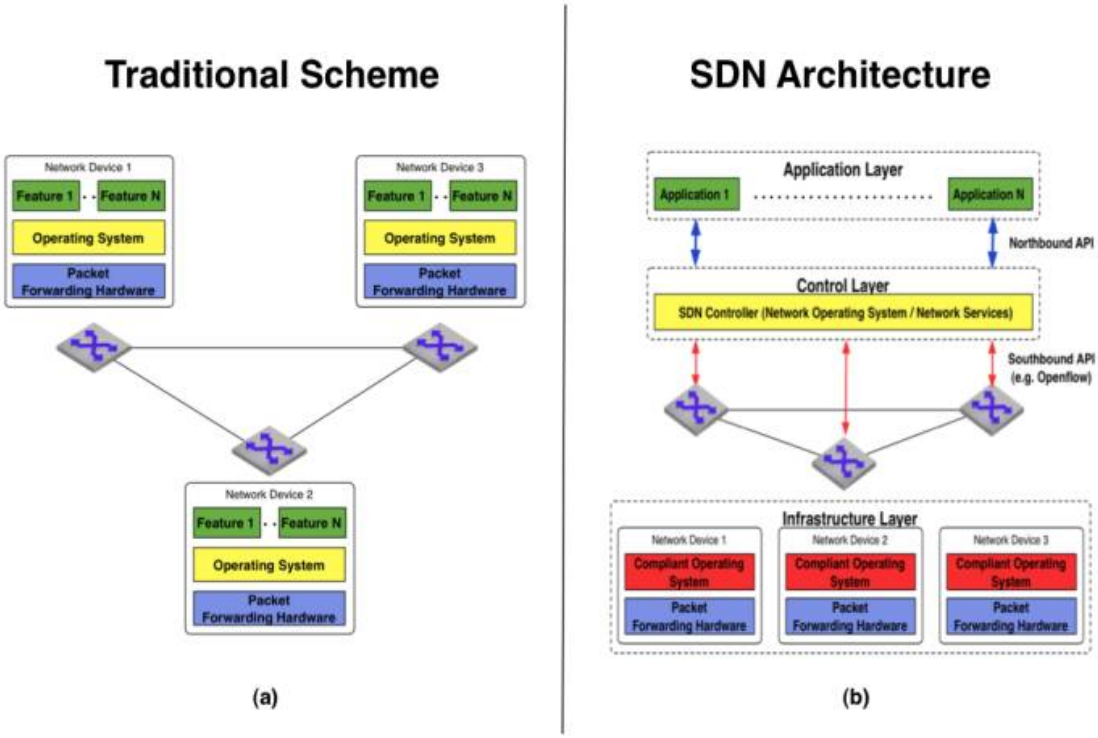

Fig. 3. Software Defined Networking Architecture v/s Traditional

III. SDN ELEMENTS

Control plan in SDN is responsible to determines optimal path for data flows. The primary different between traditional and SDN architecture is that control plan is centralized and can determined optimal routing from end to end.

Data plan in SDN is merely responsible to carrying user data across the network according to the path determined by control plan.

Management plane in SDN does not deal with routing or transfer data. It is network administrative information that play rule of network orchestration and administration duties such as HTTPS, SSH, and SNMP[5].

IV. SDN APPLICATION

An SDN application is a piece of software that is compatible with an SDN network. The application must have the ability to announce what capabilities it needs -- in terms of network resources -- to run optimally.

A. Northbound Interface

A northbound interface is an API that is used to communicate information between the SDN controller and the SDN-compatible applications running on the network. Northbound APIs essentially take the network requirements from the SDN applications and negotiate needs with the network controller that is responsible for providing applications with optimal network resources and paths.

ISSN(Online) : 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering
An ISO 3297: 2007 Certified Organization Vol.5, Special Issue 2, April 2017

An International Conference on Recent Trends in IT Innovations - Tec'afe 2017

Organized by

Dept. of Computer Science, Garden City University, Bangalore-560049, India

B. Southbound Interface

As soon as the northbound interface informs the controller of what capabilities an SDN application needs, the controller must then inform the SDN switch hardware how to treat the data flow. OpenFlow[6] is used as a southbound interface to communicate this information between the controller and the SDN-compatible switch hardware.

C. Open flow

Barring a logically architecture that show all the switches and router and other devices are managed by a controller a standard and a secure channel is desired between the SDN controller and network devices. Open flow is known as one of the first software defined networking standard. It is originally a protocol that allows a server communicates with network switches over a secure channel using SSL or Transport Layer security (TLS). In other word it is standard protocol that enables SDN controller interacts and manages the network devices such as routers and switches and tells where to send the packet. Using this protocol, controller can modifies, update, add and delete action to the flow entries in the flow table.

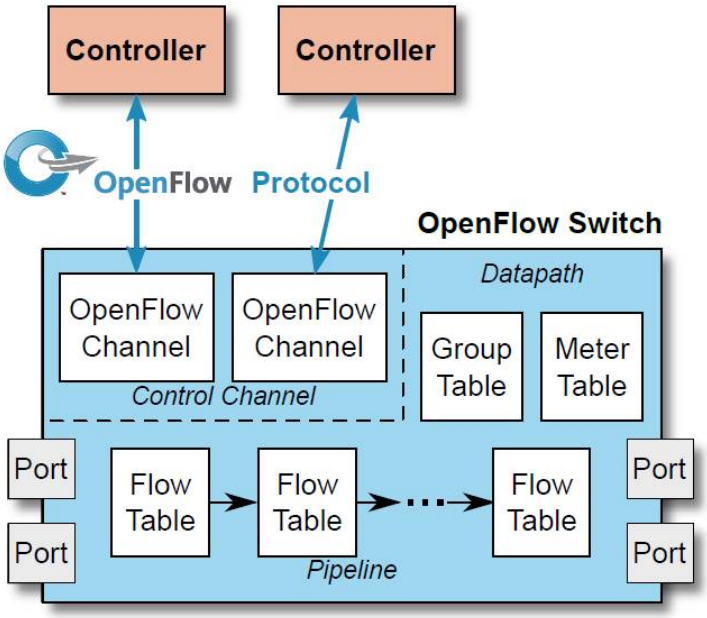

Fig. 4. OpenFlow switch atchitecture

An OpenFlow Switch consists of one or more flow tables and a group table, which perform packet lookups and forwarding, and a secure OpenFlow channel that provide communication between controller and openflow switches. It is notable that V.1.0 recall as single flow table where the most salient shift to V.1.1 is due to the addition of multiple flow table. Figure 4 describe OpenFlow switch architecture.

Flow table are the fundamental data structure in an openflow device. Each flow table in the switch contains a set of flow entries. These flow tables allow the devices to inspect incoming packet based on certain filed and take proper action according on the contents of the packet information that has been received. Matching starts at the first flow table and may continue to additional flow tables. Flow entries match packets in priority order, with the first matching entry in each table being used. If a matching entry is found, the instructions associated with the specific flow entry are executed. These actions may include forwarding the packet to a specific port, dropping the packet, or flooding the packet on all port. If no match is found in a flow table, the outcome depends on switch configuration: the packet may be forwarded to the controller over the OpenFlow channel, dropped, or may continue to the next flow table.

ISSN(Online) : 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering
An ISO 3297: 2007 Certified Organization Vol.5, Special Issue 2, April 2017

An International Conference on Recent Trends in IT Innovations - Tec'afe 2017

Organized by

Dept. of Computer Science, Garden City University, Bangalore-560049, India

V. SDN CONTROLLERS

SDN controller is an application that works as the brain of network and act as strategic control point in the SDN network.

SDN controllers is kind of operating system for network that any communications between applications and devices have to go through the controller. Controller resides between network devices once side and application layer in other side, translate the requirements from the application layer and manage flow control to the network devices (via southbound APIs) and providing the SDN application with an abstraction view of network and business logic (via northbound APIs)

The SDN Controller defines the data flows that occur in the SDN Data Plane. Each flow through the network must first get permission from the controller according to the network policy. If the controller allows a flow, it computes a route for the flow to take, and adds an entry for that flow in each of the switches along the path. With all complex functions subsumed by the controller, switches simply manage flow tables whose entries can be populated only by the controller. Communication between the controller and the switches uses a standardized protocol and API.

The SDN controller serves as a sort of operating system (OS) for the network. All communications between applications and devices have to go through the controller. The OpenFlow protocol connects controller software to network devices so that server software can tell switches where to send packets. The controller uses the OpenFlow protocol to configure network devices and choose the best path for application traffic. Because the network control plan is implemented in software, rather than the firmware of hardware devices, network traffic can be managed more dynamically and at a much more granular level.

Here we introduce number of SDN controller platform option which are used in huge company in current research and commercial project.

i. Nox/Pox:

NOX[7] was the first OpenFlow controller and a piece of the Software Defined Networking. It was developed by Nicira and becoming open source in 2008. Afterward extended and supported by ON. Lab activity at Stanford University, UC Berkeley and ICSI. The people who build the frame of SDN got to gather in 2011 and established Open Network Research Center (ONRC) and ON.Lab (Open Network Lab) to focused, develops, deploys and supports open source SDN tools and platforms. NOX version could be defined as: 1) NOX Classic: This is the version that has been available under the GPL since 2009. 2) NOX: The “new NOX.” Former is a network control platform which is based in C++ and support Python programming language, how over its line of development is suspended. In other side New NOX just supports C++ with less network application but, it is much faster and provide better codebase compared to NOX-Classic.

The below table 1 summarizes the differences between Nox classic and New Nox.

TABLE I. DIFFERENCE BETWEEN NOX & NEW NOX.

	NOX	New NOX
Core Apps	openflow , switch	messenger, snmp, switch
Network Apps	—	Discovery, Topology, , Routing Authenticator, Monitoring
Web Apps	—	Web service,web server,web service/client
Language support	c++ only	c++ and python
GUI	no	yes

ISSN(Online) : 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering
An ISO 3297: 2007 Certified Organization Vol.5, Special Issue 2, April 2017

An International Conference on Recent Trends in IT Innovations - Tec'afe 2017

Organized by

Dept. of Computer Science, Garden City University, Bangalore-560049, India

NOX suffered from some limitation, To address these problems, a new platform based on NOX was born POX[8] calls itself as the younger sibling of NOX. POX controller is a pure Python version of NOX. It is known as a general, open source, open flow controller written in python. It is renewal to improve the performance of original Python NOX.

ii. OpenDaylight [9]:

The idea of Modular application development in sense that a set of loosely coupled modules can be integrated into large application cause Open Services Gateway Initiative (OSGI) [10], a dynamic module system for Java, defines one such architecture for modular application development. SDN controller OpenDaylight is based on OSGI architecture. ODL is used by different project and community that trying to add features to the controller. It is attended as "shape the future of SDN" because it provides open platform for developers to contribute, use, and, in fact, even build commercial products. OpenDaylight (ODL) is open source controller that unannounced in 2013 and hosted by the Linux Foundation. It combine multi-vendor, multi-project ecosystem to encourage innovation and an open/transparent approach toward SDN. The OpenDayLight controller is developed based on Java and policy making was done by Technical Steering Committee. It claims extensive compatibility support from vendors such as Big Switch and Cisco. Many other SDN platforms, such as Brocade's, are based on the OpenDaylight platform. OpenDaylight not only supports the OpenFlow protocol, but also other southbound protocols such as BGP-LS and LISP for added flexibility.

iii. Open Network Operation System (ONOS) [11]

Like NOX, it was driven by ON. Lab (OpenNetworks Laboratory) and is open source too. By increasing the number of element in computer networking, thereat number of network even and corresponding handling-processes could grow exponentially. Idea of multiple and distributed controller that could handle this kind of scenario has been the hot research point new days. In another side how they interact, how the information is shared, etc are another questions which came up with building a SDN controller platform where control plan is implemented as a distributed system. Open Network Operation System is a distributed platform controller that is deputized for scalability and high availability new days. ONOS support multiple protocol at the southbound interface that provide communication with various devices and exploit right API in northbound in order to hold the needs of service provider use cases and application developers. Synchronization between multiple instances of the controller is implemented using an anti-entropy protocol (a type of gossip-based protocol) in ONOX.

1) ONOS include multiple databases with high-availability, reliable transactions, scalability and performance improvement approaches such as replication techniques, strong-consistency and partitioning. Additionally eventual consistency model is used, which informally guarantees that if no new updates are made to a given data item, eventually all accesses to that item will return to the last updated value.

2) Additionally ONOS uses Vector clocks, one of the approaches for generating a partial ordering of the events in a distributed system. The messages that are exchanged between processes contain the state of the sending process's logical clock (or an array of logical clocks for multiple processes), which helps in generating the partial ordering.

3) ONOS uses Hazelcast — an in-memory open source software data grid based on Java – for its cluster membership management.

4) ONOS is one of the design principles, and not an afterthought support.

iv. Floodlight [12]

Floodlight is a Java-based and Apache licensed open source SDN platform, used Beacon controller as its foundation. Floodlight is one of the significant contributions from Big Switch Networks (BNC) to the open source community. The platform primarily supports OpenFlow-only physical and virtual switches, it allows the capability to run Floodlight over a non-OpenFlow network. It is also Java development environment, Floodlight is both an easy to use and robust SDN controller. Modular architecture is a term to describe the Floodlight architecture. The core architecture includes various modules, such as topology management, device/end-station management, path/route computation, infrastructure for web access (management), counter store (OpenFlow counters), and a state storage system, that are well stitched a by module-management system.

ISSN(Online) : 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering
An ISO 3297: 2007 Certified Organization Vol.5, Special Issue 2, April 2017

An International Conference on Recent Trends in IT Innovations - Tec'afe 2017

Organized by

Dept. of Computer Science, Garden City University, Bangalore-560049, India

v. Ryū [13]

Ryū in Japanese stands for dragon, flow and a school of thought. It is commonly referred to as component-based, open source software defined by a networking framework. It is implemented entirely in Python, and supported by NTT's labs. One of the strengths of Ryu is that it supports multiple southbound protocols for managing devices, such as OpenFlow, Network Configuration Protocol (NETCONF), OpenFlow Management and Configuration Protocol (OF-Config), and others. The Ryu Controller provides software components, with well-defined application program interfaces (APIs), that make it easy for developers to create new network management and control applications. This component approach helps organizations customize deployments to meet their specific needs; developers can quickly and easily modify existing components or implement their own to ensure the underlying network can meet the changing demands of their applications. Ryu supports fully 1.0, 1.2, 1.3, 1.4, 1.5 and Nicira Extensions. All of the code is freely available under the Apache 2.0 license. The Ryu Controller source code is hosted on GitHub and managed and maintained by the open Ryu community. OpenStack, which runs an open collaboration focused on developing a cloud operating system that can control the compute, storage and networking resources of an organization, supports deployments of Ryu as the network Controller. Written entirely in Python, all of Ryu's code is available under the Apache 2.0 license and open for anyone to use. The Ryu Controller supports NETCONF and OF-config network management protocols, as well as OpenFlow, which is one of the first and most widely deployed SDN communications standards.

VI. CONCLUSION

Software Defined Networking (SDN) and OpenFlow have emerged as a new paradigm of networking. Software Defined Networking (SDN) is changing how we design, build, and operate networks to achieve business agility. With SDN, networks are no longer closed, proprietary, and difficult to program. They are transformed into an open and programmable component of the larger cloud infrastructure. SDN gives network owners and operators more control of their infrastructure, allowing customization and optimization, and reducing the overall capital and operational costs. SDN also allows service providers to create new revenue opportunities at an accelerated pace through the creation of software-based applications –as the PC, mobile, and Web industries have been doing successfully for years.

Comparing SDN controllers is both interesting and challenging; however, using such comparative analysis to select a controller is even more difficult. One needs to consider various aspects of the controller, and this article is one such attempt to share different aspects for comparative studies. There is still a strong need for a comprehensive comparative analysis, as mentioned in the preceding critical analysis section. Finally, applicability is an interesting aspect which should be considered when comparing SDN controllers.

REFERENCES

[1] T. Benson, A. Akella, and D. Maltz, "Unraveling the complexity of network management," in Proceedings of the 6th USENIX Symposium on Networked Systems Design and Implementation, ser. NSDI'09, Berkeley, CA, USA, 2009, pp. 335–348.
[2] N. Mckeown, "How SDN will Shape Networking," October 2011. [Online]. Available: http://www.youtube.com/watch?v=c9-K5OqYgA
[3] S. Schenker, "The Future of Networking, and the Past of Protocols,"October 2011. [Online]. Available: http://www.youtube.com/watch?v=YHeyuD89n1Y
[4] Y. Jin, Y. Wen, G. Shi, G. Wang, and A. Vasilakos, "CoDaaS: An experimental cloud-centric content delivery platform for usergenerated contents," in Proc. Int. Conf. Comput. Netw. Commun., 2012, pp. 934–938.
[5] D. Harrington, R. Presuhn, and B. Wijnen, "An Architecture for Describing Simple Network Management Protocol (SNMP) Management Frameworks," Internet Engineering Task Force, dec 2002. [Online]. Available: http://www.ietf.org/rfc/rfc3411.txt
[6] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, "OpenFlow: enabling innovation in campus networks," SIGCOMM Comput. Commun. Rev., vol. 38, no. 2, pp. 69–74, Mar. 2008.
[7] NOX detailed implementation, available online: http://www.noxrepo.org
[8] POX detailed implementation, available online: http://www.noxrepo.org
[9] Zuhran Khan Khattak , Muhammad Awais , Adnan Iqbal "Performance Evaluation of OpenDaylight SDNController," in Procceding of the 20th IEEE International Conference on Parallel and Distributed Systems (ICPADS), 16-19 Dec. 2014. DOI: 10.1109/PADSW.2014.7097868
[10] *OSGi Core Release 5*, OSGi Alliance, San Ramon, CA, USA, Mar. 2012.
[11] U. Krishnaswamy, P. Berde, J. Hart, M. Kobayashi, P. Radoslavov,T. Lindberg, R. Sverdlov, S. Zhang, W. Snow, and G. Parulkar,"ONOS: An open source distributed SDN OS,"2013.[Online]. Available: http://www.slideshare.net/umeshkrishnaswamy/open-network-operating-system
[12] "Floodlight is a Java-based OpenFlow controller," 2012. [Online].Available: http://floodlight.openflowhub.org
[13] NTT,NTT Laboratories OSRG Group.[Online].Available:http://osrg.github.com/ryu/