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. 4, Issue 4, April 2016

# A Comparative Study on Software Defined Networking Controller Features

Hitha Shiva, George Philip C

M.Tech Student, Dept. of ISE, M S Ramaiah Institute of Technology, Bangalore, India

Associate Professor, Dept. of ISE, M S Ramaiah Institute of Technology, Bangalore, India

**ABSTRACT:** Software Defined Networking (SDN) is an evolving networking technique. In traditional networking, the control structure and data structure are bundled together where as in SDN, they are separated from one another. The control of the network is now a centralized unit called Software Defined Network Controller and it is the single point of control for the whole network. The network implementation becomes less complex with this controller. SDN uses protocols like OpenFlow for the network implementation. There are a number of SDN controllers available with their own specialized features. In this paper, a study of the features of some of the SDN popular controllers is discussed.

**KEYWORDS**: Software Defined Networking; SDN Controllers; features;OpenFlow; Network control; Architecture

## I. INTRODUCTION

In today's digital world, network is growing at a faster rate and in large scale and has become a basic requirement in most of the fields. Scalability, efficiency, and performance have become some of the important factors to be considered in networking. As the network grows, the issues involved with the traditional networking techniques also increases. There have been many research initiatives to find a solution to overcome these issues. One of the best outcomes is Software Defined Networking.

Software Defined Networking (SDN) helps to overcome the issues present in the current network infrastructure. The network's control logic is removed from the underlying network devices that handle the traffic. The control logic is implemented within a logically centralized unit called the SDN Controller. With the separation of the control plane and data plane, network switches and routers are now just forwarding devices and this simplifies the policy enforcement and the network (re)configuration [1].

In the discussion that follows, SDN and its architecture is introduced; this is followed by an overview of SDN Controllers and a comparison of the features of popular SDN controllers.

## II. SOFTWARE DEFINED NETWORKING (SDN)

According to open network foundation (ONF) [3], Software Defined Networking (SDN) is an emerging network architecture that is adaptable, dynamic, cost-effective, and manageable. Hence, it is ideal for the high-bandwidth and dynamic nature of today's applications. According to ONF, SDN decouples the control of the network from the forwarding functions. This enables the network control to be easily programmable. The underlying infrastructure consisting of all the networking devices is abstracted for the applications and the network services. The SDN architecture is:

•Directly programmable: The network control can be programmed easily.
•Agile: It helps the administrators to dynamically adjust the network traffic flow.
•Centrally managed: The entire network is controlled by the Controller.
•Programmatically configured: Automated SDN programs help in easy configuration of the network resources.
•Open standards-based and vendor-neutral: They help in easy design and operation.

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. 4, Issue 4, April 2016

## III. SDN ARCHITECTURE

SDN can be shown as a 3-tier architecture [1], [2] consisting of three layers as in Fig. 1:

• Infrastructure Layer (Tier-1): The infrastructure of SDN is almost same as the traditional network; it consists of all the networking devices such as routers, switches and middle box appliances. The difference is that the physical devices in the traditional network are just forwarding elements in SDN without any control functions to take forwarding decisions.

• Control Layer (Tier-2): The Control logic of the network is moved to an external element called the SDN controller or Network Operating System (NOS). The control layer handles all the forwarding devices i.e. Tier-1. SDN controller is a software application that controls the flow to enable intelligent networking. The controller is the core element of a software defined network. SDN controller serves like an operating system (OS) for the network.

• Application Layer (Tier-3): Conceptually, application layer is above the control layer. Several applications and services use the layers below (the control and infrastructure layers). This provides easy development of applications for the network. The applications communicate with the forwarding devices through the control layer only. Most of these applications perform network related tasks.

SDN has introduced two APIs to connect these layers –
The northbound API and the southbound API. The Northbound API is the interface between the control layer and the application layer. Most commonly used northbound APIs are:

- NOSIX – it defines low level portable application interface.
- SFNet – it is a high level API. All application requirements are converted into low level service requests.

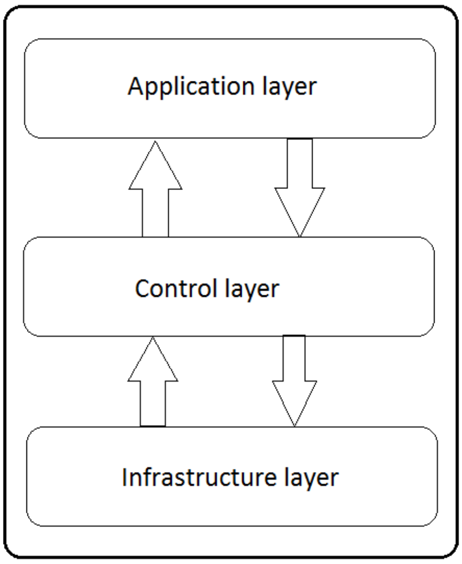


*Fig. 1 SDN Architecture*

The Southbound API is the interface between the control layer and the infrastructure layer. Two of the highly used protocols by SDN Controllers to communicate with the forwarding devices is OpenFlow protocol and Open vSwitch Database (OVSDB) protocol. OpenFlow is the first standard interface. OpenFlow® allows direct access and easy configuration of all networking devices such as switches and routers.

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. 4, Issue 4, April 2016**

## IV. SDN CONTROLLERS

A SDN controller is the core component of SDN network. It lies between the forwarding devices and the applications. All the applications communicate with the devices through the SDN controller. An SDN controller supports many protocols to configure the network devices. The controller chooses suitable network paths for different application traffics. The SDN controller serves as an operating system (OS) for the entire network. Since the network control is moved from the devices, the controller helps to manage the network automatically and makes it easier for integration of the business applications [4].

An SDN Controller is the strategic point of control in SDN network. It relays information about the network to the switches/routers and applications. An SDN Controller consists of many collections of "pluggable" modules which can be used to perform different network activities. The activities include deciding what devices are within the network, their capabilities and collecting network statistics. There are extensions available that can be included to enhance the functionality, and support advanced functionalities [5].

Some of the SDN controllers are Beacon, NOX, POX, OpenDaylight, FloodLight etc. An overview of these controllers is given below:

- Beacon [6] [15]: is a SDN Controller was introduced in the year 2010. It has been used in several research projects. This is a Java based controller. It can run on many platforms including high end multi-core Linux servers and Android phones.
- DISCO [7]: is a Distributed controller. It is mostly used for WAN and overlay networks. Each controller is responsible for one network domain. The controllers communicate through inter-controller channel. DISCO can dynamically adapt to different heterogeneous network topologies.
- IRIS [8]: is an SDN controller platform to control OpenFlow-based network entities. It can handle a large network. IRIS supports architectures that are horizontally-scalable. Hence, servers can be dynamically added to the controller cluster. This increases the performance factor of the control plane.
- Maestro [9]: is the first OpenFlow controller system that exploits parallelism. In Maestro, programmers can change the control plane functionality by writing simple single-threaded programs. Maestro has its own set of designs and techniques that helps the OpenFlow protocol. It is Java based controller and highly portable to various operating systems and architectures.
- OpenDaylight [10]: is inspired by Beacon which uses Open Service Gateway Interface (OSGi). It is a Java-based controller derived from Beacon. It supports OpenFlow and other southbound APIs. The OpenDaylight Controller is present inside its own Java Virtual Machine (JVM).
- NOX [11]: is the first SDN OpenFlow Controller platform for building network control applications. It was initially developed by Nicira Networks, alongside OpenFlow. Later, NOX was donated to the SDN community. Applications can be in Python or in C++ and it can be loaded dynamically.

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. 4, Issue 4, April 2016**

| **Properties** | **SDN Controllers** | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | **Beacon** | **Disco** | **Floodlight** | **IRIS** | **Maestro** | **NOX** | **OpenDaylight** | **POX** | **RYU** |
| Architecture | Centralized | Distributed | Centralized | Centralized | Centralized | Centralized | Distributed | Centralized | Centralized |
| Support Parallelism | Yes | Yes | Yes | Yes | Yes | No | Yes | No | Yes |
| Protocols | OpenFlow | OpenFlow | OpenFlow | OpenFlow, OVSDB | OpenFlow | OpenFlow | OpenFlow | OpenFlow | OpenFlow, OVSDB |
| Programming languages | Java | Java | Java | Java | Java | C++ | Java | Python | Python |
| Version | v1.0 | v1.1 | v1.1 | V2.0 | v1.0 | v1.0 | v1.3 | v1.0 | v1.3 |
| Graphical User Interface | Yes | Yes | Yes | Yes | Yes | Yes | Yes | Yes | Yes |
| Supported platforms | Linux, Windows | Linux, Windows | Linux, Windows, Mac OS | Linux, Windows | Linux, Windows, Mac OS | Linux, Windows, Mac OS | Linux | Linux, Windows, Mac OS | Linux |
| Supported API | ad-hoc API | REST API | REST API | REST API | ad-hoc API | ad-hoc API | REST API | ad-hoc API | ad-hoc API |
| License | GPLv2, FOSS License | - | Apache | Apache2.0 | LGPLv2.1 | GPLv3 | EPLv1.0 | GPLv3 | Apache |

*Table 1 Features of SDN Controller*

- POX [12]: is similar to NOX Controller. POX is an SDN Controller that provides rapid development and prototyping for the network. It follows OpenFlow protocol that acts as a framework between the OpenFlow switches. It is developed using Python language. POX can be easily deployed with install-free PyPy runtime.

- Floodlight [13]: is an Open source SDN Controller. It is an enterprise-class, Java-based OpenFlow Controller. It works with both the physical switches and the virtualswitches that uses the OpenFlow protocol. It also has support for OpenStack cloud orchestration platform. It can handle networks with both OpenFlow and non OpenFlow devices.

- Ryu [14]: is a component-based SDN Controller. Ryu means "flow" in Japanese. Ryu provides many extended software components and APIs to create and manage the network applications. Ryu supports protocols like OpenFlow, Netconf, OF-config, etc. Ryu is completely implemented in Python language. It is licensed under Apache 2.0.

A comparison table for the above discussed Software Defined Network Controllers is presented in Table 1.

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. 4, Issue 4, April 2016

## V. CONCLUSION

Software Defined Networking is a solution to many problems faced in traditional networking. Many organizations have taken initiative investing in its research and development of controllers that are meant for Software Defined Networking. These controllers are meant for both large scale and small scale networks. There are a number of controllers which are implemented in different languages and this gives a flexibility to choose a controller depending on the application. Most of the controllers have good user Interfaces. Along with OpenFlow protocol, some controllers use OVSDB protocol for the switches. There is a lot of scope for research on Software Defined Networking Controllers.

## REFERENCES

1. S. M. Metev and V. P. Veiko, Laser Assisted Microtechnology, 2nd ed., R. M. Osgood, Jr., Ed. Berlin, Germany: Springer-Verlag, 1998.
2. J. Breckling, Ed., The Analysis of Directional Time Series: Applications to Wind Speed and Direction, ser. Lecture Notes in Statistics. Berlin, Germany: Springer, 1989, vol. 61.
3. S. Zhang, C. Zhu, J. K. O. Sin, and P. K. T. Mok, “A novel ultrathin elevated channel low-temperature poly-Si TFT,” IEEE Electron Device Lett., vol. 20, pp. 569–571, Nov. 1999.
4. M. Wegmuller, J. P. von der Weid, P. Oberson, and N. Gisin, “High resolution fiber distributed measurements with coherent OFDR,” in Proc. ECOC’00, 2000, paper 11.3.4, p. 109.
5. R. E. Sorace, V. S. Reinhardt, and S. A. Vaughn, “High-speed digital-to-RF converter,” U.S. Patent 5 668 842, Sept. 16, 1997.
6. (2002) The IEEE website. [Online]. Available: http://www.ieee.org/
7. M. Shell. (2002) IEEEtran homepage on CTAN. [Online]. Available: http://www.ctan.org/tex-archive/macros/latex/ contrib. /supported/IEEEtran/
8. FLEXChip Signal Processor (MC68175/D), Motorola, 1996.
9. “PDCA12-70 data sheet,” Opto Speed SA, Mezzovico, Switzerland.
10. A. Karnik, “Performance of TCP congestion control with rate feedback: TCP/ABR and rate adaptive TCP/IP,” M. Eng. thesis, Indian Institute of Science, Bangalore, India, Jan. 1999.
11. J. Padhye, V. Firoiu, and D. Towsley, “A stochastic model of TCP Reno congestion avoidance and control,” Univ. of Massachusetts, Amherst, MA, CMPSCI Tech. Rep. 99-02, 1999.
12. Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specification, IEEE Std. 802.11, 1997.