JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

An Introduction to Software Defined Networking

[1]Ajay Prashar, [2]Jawahar Thakur
[1]Ph.D. Student, [2]Professor
[1]Department of Computer Science,
[1]Himachal Pradesh University, Shimla, India

Abstract: Traditional networks are very complex and arduous to manage. It uses fixed and dedicated infrastructure devices such as routers and switches to control network traffic and are also vertically integrated i.e. there is a strong coupling between control and data planes. The functionality of the traditional network mostly implemented on dedicated infrastructure devices so deploying new services and protocols is a challenging tasks because all the switches need to be updated or replaced manually. The main idea behind Software Defined Networks (SDN) is to separate the control plane from the data plane and move it to a centralized server known as controller. In this paper, an introduction to Software Defined Networking is presented. In SDN (Software Defined Networking), the whole intelligence lies in the controller that programmatically control the underlying hardware devices. This paper also covers the introduction of SDN, SDN Architecture, SDN Controller, Characteristics of various SDN controller and SDN controller service.

Keywords **- Traditional Network, Control Plane, Data Plane, Software Defined Networking, SDN Controller.**

I. INTRODUCTION

In the current scenario with the advancement of computer technology, there is a rapidly increase in the mobile devices and emerging technologies such as cloud computing and virtualization that caused changes of traffic pattern, arise the need of high network capacity and network scaling. To Support these needs network devices become very complicated. Various networking devices such as switches, routers and intermediate devices are present in the traditional network in which application specific integrated circuits are installed to perform dedicated operations [1].There is a strong coupling between control plane and data plane in the conventional networking, so data controlling and forwarding policies are provided in the same layer. Therefore, in the traditional networking, deploying new services and protocols is a challenging tasks because all the switches need to be updated or replaced manually [2].

A few other short comings of traditional networks are flexibility, manageability and extensibility. To overcome such limitation the concept of Software Defined Networking has been proposed [3].

The main objective of SDN (Software Defined Networking) is to separate the control plane from the data plane and move it to centralized server named controller. In SDN, the whole intelligence lies in the controller that programmatically control the underlying hardware devices. SDN environment allow network administrator to manage and control the networking infrastructure automatically [4]. SDN has emerged as a mean to improve programmability with in the network to support the dynamic nature of future network functions [5]. This research has conducted a survey on Software Defined Networking.

1.1 SDN Architecture

SDN architecture consist of three layer:

- Infrastructure Layer
- Control Layer
- Application Layer

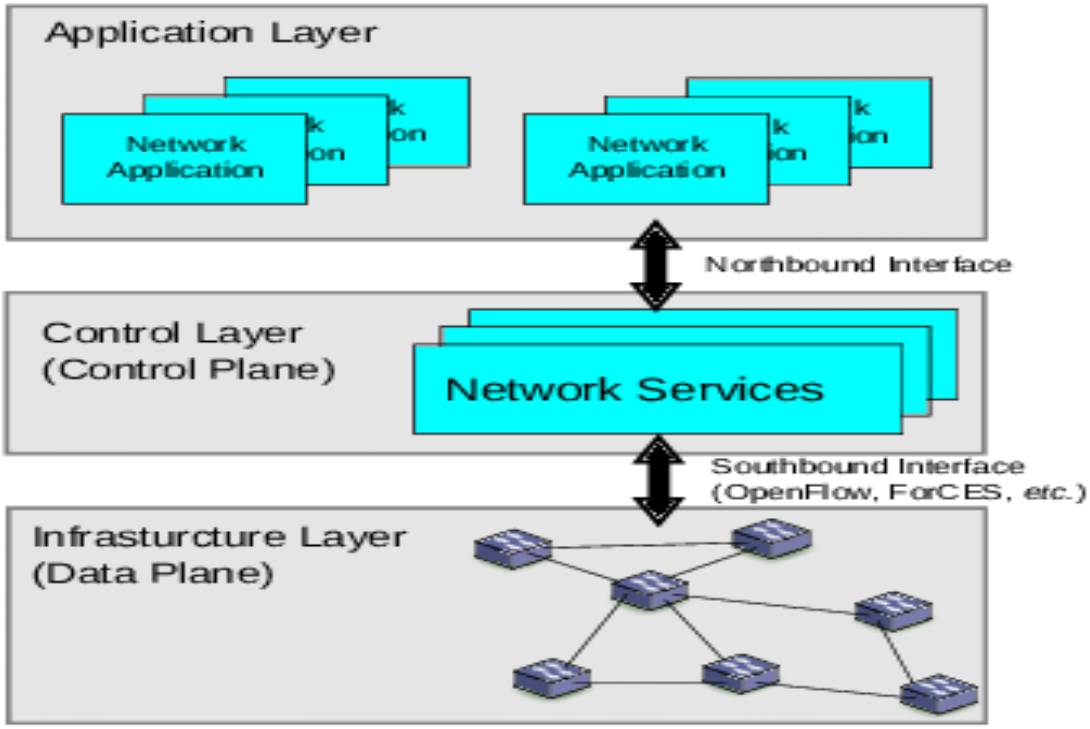

Figure 1.1 SDN Architecture [30]

1.1.1 Infrastructure Layer

Infrastructure layer or forwarding layer is the bottom layer of the SDN architecture also known as data plane. This layer consist of various network devices like routers, switches, access point and the supporting physical hardware devices.

1.1.2 Control Layer

Control Layer or Control Plane is the middle layer of the SDN architecture which contains software based SDN controller that provide control functionality in order to supervise the network forwarding behavior. The SDN controller communicate with the rest of two layer (i.e. Infrastructure Layer or Application Layer) using Southbound Interface (controller and infrastructure layer) and Northbound Interface (controller and application layer)

The SDN controller maintains the global network view and update the table of switch using southbound API. OFP (Open Flow Protocol) is used for communication between controller and infrastructure layer device using secure connection such as the TLS or SSL connection.

1.1.3 Application Layer

Application layer in SDN architecture contain one or more applications and services running on the network. Application layer interact with controller using Northbound API and convey their network requirement and desired network behavior to the SDN controller. Northbound interface facilitate the communication among the high level component. The application communicate with controller and told them what resource the application need and their current destination. Traditional network use load balance or firewall to control the data plane behavior but SDN applications communicate with the controller through its northbound interface [2, 6, 7].

1.2 SDN Controller

SDN separates control plane from the data plane and the intelligence of the entire network is moved to the controller, all computations are done there and additional features or applications can be added as needed by network. The SDN controller manages flow control for efficient network management. It typically runs on a server and use protocol i.e. some set of procedure or rule to tell switches where to send the upcoming packets.

Software defined network use two types of controllers.

- Centralized Controller
- Distributed Controller

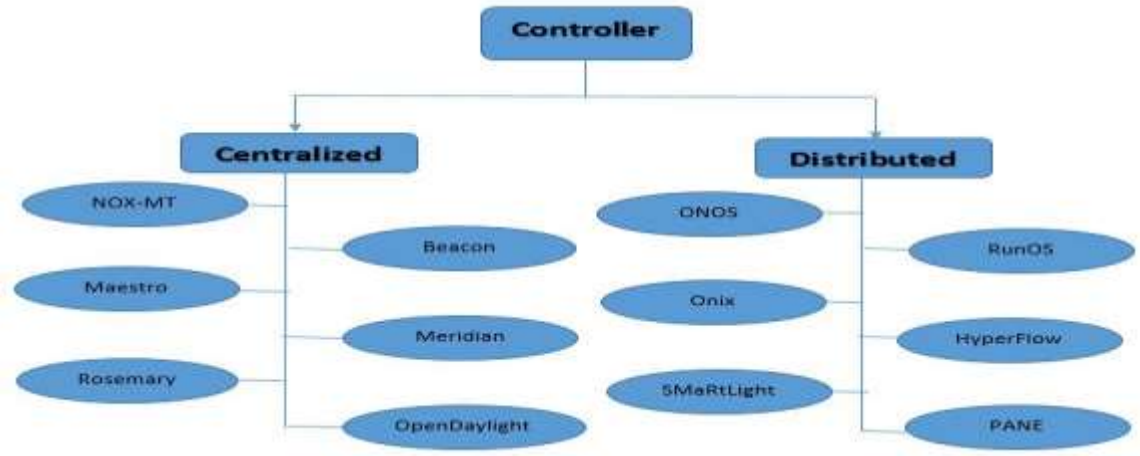

Figure 2 Types of Controller

1.2.1 Centralized Controller

Centralized controller implement all control plane logic at a single location. In centralized controller only a single server required for all the activities of control plane. It provide a single point of control so the management of this type of controller is easier as compared to the distributed controller [8]. This types of controller suffer from two principal issues:

- **Scalability Issue:** Centralized controller suffers from scalability issue because every server has limited capacity to deals with infrastructure layer devices. In Software Defined Networking, scalability means the capacity of the SDN controller to handle large amount of forwarding path requests from switches. SDN controller has limited number of resource when handling a large amount of requests.

- **Robustness Issue:** In centralized controller only a single server required for all the activities of control plane. A single SDN controller has a major problem of single point of failure. When a SDN controller fails, switches will not be able to send new packets, and the entire network will break down.

1.2.2 Distributed Controller

Distributed SDN controller take over both the issues that are exists in the centralized SDN controller. In SDN, distributed controller means multiple controllers that can share the load in the network equally. If one controller is failed it can take over another controller. In comparison to centralized controllers, distributed controllers have advantages in case of scalability and high performance during increase demand of requests [9].

Table 1: Characteristics of Different SDN Controller's

Sr. No	Controller	Developed by	Architecture	Language	Open Flow Version	Platform Support	Description
1	NOX	Nicira	Centralized	C++	OF 1.0	Most supported on Linux	Provides fast and asynchronous IO. NOX-MT has a multi-threaded architecture and it use low level input output programming [10][11].
2	Beacon	Stanford University	Centralized	Java	OF 1.0	Linux, Windows and MacOS	Stable and cross platform that support both event-based and threaded operation. It is dynamic in nature and support rapid development [12].
3	Maestro	RICE & NSF	Centralized	Java	OF 1.0	Linux, Windows and MacOS	Multi-threaded controller which use task batching [13].
4	Meridian	IBM T. J.	Centralized	Java	OF1.0, OF	Cloud-	Supports a service-level

		Watson Research Center			1.3	based	model for application networking in clouds [14].
5	Rosemary	Atto Research, KAIST, CSLSRI International	Centralized	C	OF1.0, OF1.3 & XMPP	Linux	A Robust, Secure, and High-Performance Network Operating System [15].
6	OpenDayLight	Linux Foundation With Memberships Covering Over 40 Companies, Such As Cisco, IBM, NEC	Centralized	Java	OF1.0, OF1.3	Linux, Windows and MacOS	Modular, pluggable, and flexible controller platform, supporting multiple southbound protocols [16]
7	ONOS (Open Network Operating System)	At&T, Ciena, Cisco, Ericsson, Fujitsu, Huawei, Intel, Ntt Communication	Distributed	Java	OF1.0, OF1.3	Linux, MacOS, Windows	Supports both configuration and real-time control of the network, eliminating the need to run routing and switching control protocols inside the network fabric [17].
8	RunOS	Applied Research Center for Computer Networks (ARCCN), RUSSIA	Distributed	C++	OF 1.3	Linux	Modular, high performance and scalability with the use of multithreading. Used as monitoring tool (control traffic statistics and port statistics) [18].
9	Onix	Nicira , NTT & Google	Distributed	C++	OF 1.0, OVSDB	-	Distributed control platform for large scale production network that deals with reliability and scalability Problem [19].
10	HyperFlow	University of Toronto	Distributed	C++	OF 1.0	-	Distributed event-based control plane for OpenFlow.it provides scalability and decrease response time.
11	SMaRtLight	University of Lisbon	Distributed	Java	OF 1.3	Linux	Address the fault tolerance issue in the network.
12	PANE	Brown University	Distributed	Haskell	OF 1.0	MacOS, Linux	API for SDN Control, Security [8].

1.3 SDN Controller Services

SDN controller is used to control, manage or update the flow of data for improved network management and application performance. SDN separates the control plane from the data plane and the intelligence of the whole network is transfer to the controller.

Some of the basic services provided by SDN controllers are:

- **Link discovery:** Link discovery means collecting the information of physical link. In SDN, there are two types of link discovery. The first one is the link discovery between switches and the second one is the link discovery between switch and end host. On the basis of these information neighbor database is prepared at the controller level.

- **Topology Manager:** Topology manager manage the database at the controller level and compute the shortest path between any node and host [20].
- **Flow Programmer:** For selecting the proper valid rules flow Rule manager will communicate with the southbound and then load the rules into the switches which avoid clash of rules.
- **Host Tracker:** Host tracker is used for storing the information of end host in the network, which help to find the end node by their physical address [21].

II. Related Works

This section focuses on highlighting the related work in the field of Software Defined Network (SDN).

Nunes BA et al. (2014) did a survey on software-defined network and provide an overview of programmable network. In this paper SDN architecture and OpenFlow standard are described in details [22]. **Cabaj K et al. (2014)** proposed DFSA system after analyzing the features of software defined networking in the context of security application. The proposed system can be used efficiently for the detection of various network attacks that are observed nowadays in IP networks [23]. **Blial O et al. (2016)** discussed a comprehensive overview of SDN multicontroller architecture and also discuss OpenFlow. In this research paper the difference between various types of multicontroller architectures are explained on the basis of distribution method and the communication [24]. **Rastogi A et al. (2016)** did a comparative analysis of software defined network controllers POX and RYU in terms of traffic handling capabilities. The experimental results shows that in case of layer 1 switching, POX has efficient traffic handling capabilities and efficient performance. But, in case of layer 2 switching, RYU has efficient performance as compared to POX. For testing purposes Mininet emulator and Distributed-Internet Traffic Generator traffic generator are used [25].

Ominike Akpovi A et al. (2016) did a study on Software Defined Network (SDN), its architecture and how it is different from traditional networking. OpenFlow and Mininet simulation tools are also discussed in this paper [26]. **Singh S et al. (2017)** did a survey on Software Defined Network architecture and discussed historical evolution of SDN, functional architecture of the SDN and its related technologies. An architecture on software defined heterogeneous network are also purposed, pointing towards new network technology, that will facilitate in handling of large amount of internet traffic and helps infrastructure and service providers to customize their resources dynamically[27]. **Zhang Y et al. (2018)** did a survey on SDN controllers and also discussed their benefits or challenges after giving an overview of SDN and OpenFlow [28]. **Lee S et al. (2019)** compares Software Defined Networking simulators for tactical network. In this paper OPNET and Mininet SDN simulator are compared on the basis of end-to-end delay, jitter, and RTT between controllers and switches. The experimental results shows that the Mininet environment cannot apply the network load, so it is the best and efficient choice to use OPNET for simulation of the real world environment [29].

III. Conclusion and Future Scope

In this paper, we provided an overview of software defined networks and also describe the various SDN controller used in the network. SDN is based on the idea of separating the control plane from the data plane and centralizing the overall intelligence of the network in a single controller that manages the entire network. However, throughout the years, it is realize that the future of Software Defined Networking relies on distributed controller, because centralized controller suffers from scalability and robustness issues.

REFERENCES

[1] Bera S, Misra S, Vasilakos AV. Software-defined networking for internet of things: A survey. IEEE Internet of Things Journal. 2017 Aug 29;4(6):1994-2008.

[2] Rowshanrad S, Namvarasl S, Abdi V, Hajizadeh M, Keshtgary M. A survey on SDN, the future of networking. Journal of Advanced Computer Science & Technology. 2014 Jul 1;3(2):232-248.

[3] Farhady H, Lee H, Nakao A. Software-defined networking: A survey. Computer Networks. 2015 Apr 22;81:79-95.

[4] Software Defined Networking, available online: https://opennetworking.org/sdn-resources/whitepapers/software-defined-networking-the-new-norm-for-networks/ last visit: 2021 Jul 10.

[5] Sharma S, Sood M. An Architectural Proposal for an SDN based Data Centre: A Case Study. International Journal. 2013 Oct;2(10).

[6] Karakus M, Durresi A. A survey: Control plane scalability issues and approaches in software-defined networking (SDN). Computer Networks. 2017 Jan 15;112:279-293.

[7] Chen J, Zheng X, Rong C. Survey on software-defined networking. In Second International Conference on Cloud Computing and Big Data in Asia 2015 Jun 17 (pp. 115-124). Springer, Cham.

[8] Paliwal M, Shrimankar D, Tembhurne O. Controllers in SDN: A review report. IEEE Access. 2018 Jun 11;6:36256-70.

[9] Oktian YE, Lee S, Lee H, Lam J. Distributed SDN controller system: A survey on design choice. computer networks. 2017 Jul 5;121:100-11.

[10] NOX, available online: https://www.sdxcentral.com/directory/nox-repo/nox/, last visit: 2021 Jul 10.

[11] Shah SA, Faiz J, Farooq M, Shafi A, Mehdi SA. An architectural evaluation of SDN controllers. In2013 IEEE international conference on communications (ICC) 2013 Jun 9 (pp. 3504-3508). IEEE.

[12] Beacon , available online: https://openflow.stanford.edu/display/Beacon/Home, last visit: 2021 Jul 10.
[13] E.Ng, "Maestro: A System for Scalable OpenFlow Control," available online: www.cs.rice.edu/~eugeneng/papers/TR10-11.pdf, last visit: 2021 Jul 10.
[14] Banikazemi M, Olshefski D, Shaikh A, Tracey J, Wang G. Meridian: an SDN platform for cloud network services. IEEE Communications Magazine. 2013 Feb 14;51(2):120-127.
[15] Shin S, Song Y, Lee T, Lee S, Chung J, Porras P, Yegneswaran V, Noh J, Kang BB. Rosemary: A robust, secure, and high-performance network operating system. InProceedings of the 2014 ACM SIGSAC conference on computer and communications security 2014 Nov 3 (pp. 78-89).
[16] Opendaylight. available online: http://www.opendaylight.org/, last visit: 2021 Jul 10.
[17] ONOS, available online: https://opennetworking.org/onos/https://opennetworking.org/onos/ , last visit: 2021 Jul 10..
[18] RunOS, available online: https://github.com/ARCCN/runos , last visit: 2021 Jul 10.
[19] Onix, available online: https://www.sdxcentral.com/networking/sdn/definitions/sdn-controllers/ last visit: 2021 Jul 10.
[20] Salman O, Elhajj IH, Kayssi A, Chehab A. SDN controllers: A comparative study. In2016 18th Mediterranean Electrotechnical Conference (MELECON) 2016 Apr 18 (pp. 1-6). IEEE.
[21] Research PJ, Raj P. Topology-based analysis of performance evaluation of centralized vs. distributed SDN controller. In2018 IEEE International Conference on Current Trends in Advanced Computing (ICCTAC) 2018 Feb 1 (pp. 1-8). IEEE.
[22] Nunes BA, Mendonca M, Nguyen XN, Obraczka K, Turletti T. A survey of software-defined networking: Past, present, and future of programmable networks. IEEE Communications surveys & tutorials. 2014 Feb 13;16(3):1617-34.
[23] Cabaj K, Wytrebowicz J, Kuklinski S, Radziszewski P, Dinh KT. SDN Architecture Impact on Network Security. InFedCSIS (Position Papers) 2014 Sep 7 (pp. 143-148).
[24] Blial O, Ben Mamoun M, Benaini R. An overview on SDN architectures with multiple controllers. Journal of Computer Networks and Communications. 2016 Apr 1;2016.
[25] Rastogi A, Bais A. Comparative analysis of software defined networking (SDN) controllers—In terms of traffic handling capabilities. In2016 19th International Multi-Topic Conference (INMIC) 2016 Dec 5 (pp. 1-6). IEEE.
[26] Ominike Akpovi A, Adebayo AO, Osisanwo FY. Introduction to Software Defined Networks (SDN). International Journal of Applied Information Systems. 2016 Dec; 11(7):10-14.
[27] Singh S, Jha RK. A survey on software defined networking: Architecture for next generation network. Journal of Network and Systems Management. 2017 Apr 1;25(2):321-74.
[28] Zhang Y, Cui L, Wang W, Zhang Y. A survey on software defined networking with multiple controllers. Journal of Network and Computer Applications. 2018 Feb 1;103:101-18.
[29] Lee S, Ali J, Roh BH. Performance comparison of software defined networking simulators for tactical network: Mininet vs. OPNET. In2019 International Conference on Computing, Networking and Communications (ICNC) 2019 Feb 18 (pp. 197-202). IEEE.
[30] Braun W, Menth M. Software-defined networking using OpenFlow: Protocols, applications and architectural design choices. Future Internet. 2014 Jun;6(2):302-36.