www.ijecs.in
International Journal Of Engineering And Computer Science
Volume 9 Issue 02 February 2020, Page No. 24953-24961
ISSN: 2319-7242 DOI: 10.18535/ijecs/v9i02.4439

A Review on Software-Defined Networking Distributed Controllers

Worku Muluye Wubet
Wolkite University, College of Computing and Informatics, Wolkite, Ethiopia

Abstract:
A computer network is a critical issue in our day to day activity; however, today it works under various problems. Since in the current network architecture the control plane and data plane are vertically bundled on the same device. To solve this problem programmable Software-Defined Network is released. Most SDN controllers use the OpenFlow protocol to vertically separate the control plane and data plane of the network devices. In SDN the controllers are the brains of the network that controls the network devices. Today's network required successful integration of distributed controllers to make the network more consistent, available and high performance. SDN distributed controller is a controller that we can add or remove the controllers according to the number of devices change. Distributed controller architecture has investigated and compared the 6 recent distributed controllers by using 26 criteria. Orion is the first best controller and ONOS is the second-best controller**.**

Keywords: SDN, distributed controller, logically distributed controller, logically centralized controller

1. Introduction

The computer network is the basis for our day to day activity so that it becomes the critical infrastructure of our homes, businesses, and schools [1]. The current network architecture has a vertically integrated control plane and data plane and it is completely vendor dependent. The network devices such as the routers and switches are made path decision to forward the data to its final destination. In another word each of the network elements such as router and switch perform both the forwarding plane and the control plane operations. The current network has several disadvantages some of which are limited innovation, not scalable, use low-level commands, prone to error, difficult to manage, expensive equipment, consumes a lot of time to configure each individual device and vendor locked [2], [3].
SDN is a new network architecture that controls the entire network devices centrally which solves the current network limitations [4]. According to [5] SDN is an emerging network architecture that is the dynamic, manageable, cost-effective, adaptable, and scalable network. Open Networking Foundation (ONF) is a non-profit organization responsible for standardizing and promoting SDN and network devices [5]. SDN has three planes such as management plane, control plane, and data plane. As [5] report ONF has more than 502 members that produce OpenFlow network devices [5].
In the SDN network, we can add or remove devices very easily. The network administrator does not need to configure each individual device in the network. Instead, configuring the changes in the controller would deploy the modifications on the entire network. Because of SDN vertically separate control plane and data plane [6].
SDN facilitates communication between the applications and the network. This results in a dynamic network for a dynamic application. SDN provides various features than the vertically integrated current network [7]. In general, the SDN network is scalable, flexible, reliable, secure, programmable, has high performance and availability than the existing current network [8], [9].

Today networks need scalable, flexible, available, secure, and fast devices. Companies try to find solutions that allow them to expand their network easily and cheaply and to choose their equipment without linked to their previous vendors [4], [10]. To address these problems, the researchers, network companies, datacenter, Internet service providers and enterprise networks are turning to the SDN network.
However, this SDN centralized network control has performance, scalability, availability, and single point failure problems of the controllers. The solution for these problems in the large and growing network is to distribute the control plane architecture.
According to this researchers proposed different control plane architectures, such as the multi-core controller [7], the logically centralized controller [11] and the logically distributed controller [12], [13]. Since to solve the centralized SDN controllers limitation distributed controller is the first option [14], [15]. The researcher was motivated to do a comparison of the distributed controller, because SDN is a new network architecture, interesting, and it is a current hot research area.

2. SDN Architecture

In [3] SDN makes computer networks more programmable, scalable, agile, innovation in network management possible and lowers the barrier to deploying new services. As [8] SDN can be described from up to bottom as the application layer, control layer, and infrastructure layer. Also, SDN architecture has three application interfaces such as northbound, east/westbound and southbound Application Interfaces (APIs)as illustrated in Fig.1 [8].

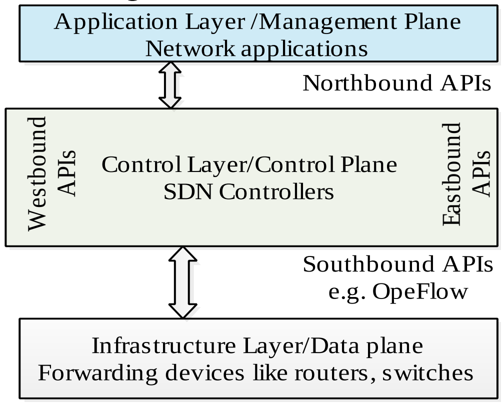
Application Layer /Management Plane
Network applications
Northbound APIs
Westbound APIs
Control Layer/Control Plane
SDN Controllers
Eastbound APIs
Southbound APIs
e.g. OpeFlow
Infrastructure Layer/Data plane
Forwarding devices like routers, switches

Figure 1: SDN architecture [16].

2.1.Application Layer

The application layer contains network applications that can introduce new network features, such as security, manageability and forwarding schemes. The northbound application interface is an interface between the application and the control layer [7].

2.2.Control Layer

The control layer contains one or more controllers that provide programmatic control of all forwarding operations, capabilities advertisement, make a decision, statistics reporting and event notification.
Eastbound/Westbound APIs are special case interfaces required by distributed controllers [17]. The functions of these interfaces are import or export data between controllers. Eastbound APIs are interconnecting current IP networks with SDN networks. Westbound APIs are delivered information between multiple controllers in diverse areas and enable management of distributed SDN architecture [8], [18].

Southbound APIs offer the software interfaces between the control layer and the infrastructure layer. OpenFlow is the most common and popular southbound API. OpenFlow is a protocol that separates the control plane and forwarding plane [7], [8]. OpenFlow was proposed by McKeown and currently used in most SDN practices [19]. OpenFlow switch is a software program or hardware device that forwards packets in an SDN environment based on the controller [20]. In an OpenFlow switch, there are one or more flow tables and each flow table has a lot of flow entries. A flow entry within a flow table used to match and process packets. The OpenFlow switch uses an OpenFlow channel to secure communication between the switch and the controller [21].

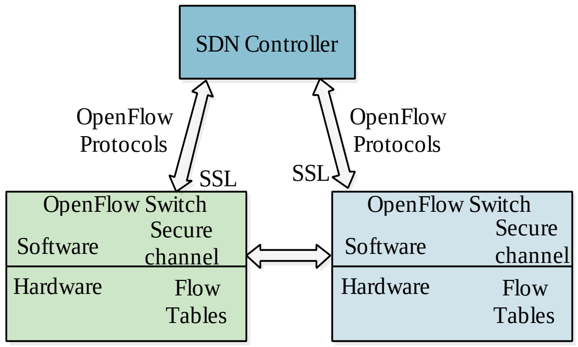

Figure 2: OpenFlow Switch architecture [20].

2.3.Infrastructure Layer

It contains a lot of forwarding devices like switch and router that performs packet forwarding, based on the forwarding rules imposed by the controller [21].

3. Related Works

In [22]compared four OpenFlow controllers such as Floodlight, NOX, Beacon, and Maestro. They considered four key performance bottlenecks including multi-core support, switch partitioning, packet and task batching. Based on these results, a Floodlight controller is the best controller, whose performance was better than the other three controllers.

In this paper [18] present a comprehensive overview of SDN multi-controller architectures by explaining their features. In this paper, the communication system and distribution method of the multi-controller were investigated.

In [23] compared five controllers such as POX, Ryu, Terma, Floodlight, and OpenDaylight. The authors used the Multi-Criteria Decision Making (MCDM) method to select the best controller. They compare based on GUI, available interfaces, supporting of virtual switching, being open-source, supporting of REST API, productivity, having documentation, age, modularity, supporting language, platform, OpenFlow and OpenStack networking. The results show that Ryu is the first best controller. Floodlight, OpenDaylight, Terma and Pox controller has the next best controllers respectively.

The authors in [14] perform performance evaluation on centralized OpenDaylight and Floodlight SDN controllers and OpenDaylight is better than Floodlight.

The authors of [24] explained a qualitative comparison of ONOS and OpenDaylight controllers. This paper compared the setup, discovery, change, and removal of northbound interfaces of ONOS and OpenDaylight SDN controllers. The result shows that the ONOS intent framework and REST APIs are better than the OpenDaylight yang interface and GBP interfaces.

The authors of [25] have compared 11 centralized and distributed controllers such as ONOS, OpenDaylight, Floodlight, Ryu, Maestro, Iris, Mul, Runos, Nox, Pox, Beacon, and LibFluid. Also, measure the throughput and latency of those controllers. The authors concluded that OpenDaylight is a good choice as a full-featured SDN controller.

Finally, all previous studies haven’t considered the comparison of the SDN distributed controller and the work presented in this paper is different from all previous works.

4. SDN Distributed Controllers

A distributed controller architecture is a set of controllers working together to achieve some level of performance, security, availability, scalability by avoiding a single point of failure [8], [18]. The distributed controller handles complex tasks such as quality of service, managing virtual private networks in large networks and traffic engineering [26].

The authors in [27] design Distributed Hopping Algorithms (DHA) and scalable control mechanisms to solve Switch Migration Problem (SMP). Also, indicate two distributed controller architectures such as hierarchical architecture are vertically layered from root controller to leaf and flat architecture is horizontally equalizing all the controllers.

This paper [6] reports, a single controller is insufficient to control the entire network; so a distributed controller is needed. They also try to determine how many controllers to use and where to place them.

The paper [8] and [18] stated three types of SDN architectures, centralized, logically centralized and physically distributed and logically distributed architectures.

4.1.Logically Centralized but Physically Distributed Control Planes

They are considered as a single controller [18]. All of the controllers have the same responsibilities and split the charge equally, but one is a master and the remaining are slaves [8]. The controllers aware of every change in the network and share the same information instantly to the network synchronization [18]. Each controller is clustered to exchange information with each other using east/west APIs protocols. The most common logically centralized but physically distributed controllers are ONOS, HyperFlow, Onix, and OpenDaylight [28].

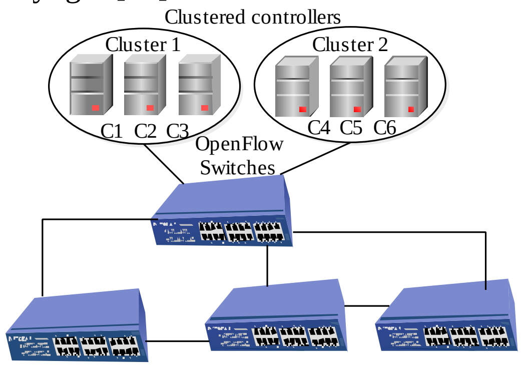

Figure 3: Logically centralized and physically distributed clustered controller architecture.

1) HyperFlow: HyperFlow [29] is developed on the top of the Nox controller, to enable logically centralized multi-controller architectures.

2) ONOS: ONOS [30] is a physically distributed and logically centralized controller and it is distributed in the form of a cluster. ONOS is designed for service providers and mission-critical networks. It contains a cluster of controllers that work together to manage the applications and the network devices [16]. As the network devices are an increase ONOS can scale by adding additional controllers into the cluster [11].

ONOS controllers in the cluster interact with all the other controllers by using a specific TCP port 9876 [31]. The controllers send and accept keepalive messages to or from other controllers to monitor the cluster members [11], [17].

3) OpenDaylight: OpenDaylight [22] was designed for datacenter networks in 2013. It can be deployed on any hardware that supports Java. Some of the features of this controller are Java-based (OSGI), modular, and supporting multiple southbound protocols. OpenDaylight supports the program bidirectional REST that supports applications running in the same address space [32].

4) Onix: Onix [33] is a distributed control plane that contains a cluster of one or more physical servers; each one may run multiple Onix instances. It reduces workload by adding controllers without merely replicating it. It allows aggregation in which the network managed by a cluster of Onix controllers appears as a single controller in a separate cluster's network information base. Onix [17] can handle forwarding elements, link, controller and connectivity between network devices and Onix controllers network failures. It doesn't support GUI, management interface, not flexible and not support all OpenFlow versions.

4.2.Logically and Physically Distributed Control Planes

The controllers are physically and logically distributed and may have flat or hierarchical control plane architecture [18]. Additionally, every controller has just a view of the domain it is responsible for, and it can take decisions for it, unlike a logically centralized controller, where each controller makes a decision based on the global network view [21]. Kandoo and Orion are the most common fully distributed controllers.

1) *Kandoo*: Kandoo [12] is a logically distributed controller with a hierarchical design of two layers. The local controller layer that contains local controllers, where each controller controls its subdomain. The local controllers only forward to events that were subscribed by the root controllers. The root controller layer contains the root controller that controls all local controllers [18].

2) *Orion*: Orion [13] is a hybrid control plane, which is a mix of horizontal and hierarchical architectures and it has three layers. The physical layer that contains a lot of connected OpenFlow switches. The middle layer of the control plane that includes the area controllers, which handle collecting physical device, link information and dealing with intra-area requests and updates. The middle layer also abstracting the network view and sending it to the management layer of the control plane. And the upper layer, which contains the domain controllers [13].

4.3.Distributed Controllers Communication

Distributed controller communication is a method of exchanging information among multiple controllers [34]. A set of switches subscribe to a particular controller and each controller does the same. Next, to this, distributed controllers publish information between each other to form a global network view. Also, each controller will send information to its neighbor's controller about its local state to build a global network view [18].

Clustering is an SDN controllers distributing method, which makes a set of connected SDN controllers work together as a single controller [16]. When one of the controllers fails in a cluster, resources are redirected and the workload is redistributed to another working controller.

In the cluster, OpenFlow switches are connected to more than one controller [11]. Then the switch determines which controller should be the master and which should be the standby or slave [31]. When running multiple controllers in a cluster, each controller has three roles, with respect to forwarding devices. The master role has knowledge of the devices and has full control read or write access to all of the connected devices. The standby role has a knowledge of the device and can read the device's state, but not manage or write to the device. None role that may or may not have knowledge of the device and cannot interact with it [30], [35].

5. Comparison of Logically Centralized and Logically Distributed Controllers

The selected recent logically centralized controllers are ONOS [11], OpenDaylight [32], Onix [33], HyperFlow [29] and logically distributed controllers are Kandoo [12] and Orion [13]. The question is, which controllers are to be selected and used? Table I below shows the comparison of SDN distributed controllers

Table 1: Comparison of distributed controllers [8, 11, 12, 13, 16, 17, 22, 25, 29, 36]

Criteria	***ONOS***	***OpenDaylight***	***Onix***	***HyperFlow***	***Kandoo***	***Orion***
Architecture	Distributed	Distributed	Distributed	Distributed	Distributed	Distributed
Physically Distributed	Yes	Yes	No	Yes	Yes	Yes
Logically Centralized	Yes	Yes	Yes	Yes	No	No
Logically Distributed	No	No	No	No	Yes	Yes
Flat architecture	No	No	No	No	No	Yes
Hierarchical architecture	Yes	Yes	Yes	Yes	Yes	Yes
Platform Support	Linux, Mac, Wins	Linux, Wins,	Linux	Linux	Linux	Linux
OpenFlow support	Yes	Yes	Yes	Yes	Yes	Yes
Language	Java	Java	Python, C++, C	C, C++	C, C++, Python	Java
Northbound APIs	RESTful API	REST, Java APIs, RESTCONF	NVP, NBAPI	NO	RPC API	Java APIs
Southbound APIs	OpenFlow, OVSDB, NetConf	OpenFlow, OVSDB, NetConf	OpenFlow, OVSDB	OpenFlow, OVSDB	OpenFlow	OpenFlow
East/Westbound APIs	Yes	No	Yes	Yes	Yes	Yes
Availability	High	High	Medium	Medium	Medium	Very high
Scalability	High	Medium	High	High	High	Very high
Performance	High	Medium	Medium	High	High	High
Security	High	High	Medium	Medium	Medium	High
Reliability	High	Medium	Medium	Medium	Medium	Very high
GUI support	Yes	Yes	No	Yes	No	Yes
Management Interfaces	GUI/CLI, REST API	GUI/CLI, REST API	No	No	CLI	CLI
Open Source	Yes	Yes	No	Yes	No	No
Consistency	Strong	Weak	Strong	Weak	Weak	Strong

Modularity	High	High	Medium	Medium	High	Very High
Flexibility	Yes	Yes	No	Yes	Yes	Yes
Developer or Partner	ON.LAB, Cisco, At&T, Ciena, Ericsson, Fujitsu, Huawei, Intel,	Linux Foundation, IBM, Cisco, NEC..	Nicira,	Amin Tootoonchian	-	-
Documentation	Weak	Medium	Weak	Weak	Weak	Weak
First Released Year	2014	2013	2010	2010	2012	2014

As we can see in Table 1 vertically designed distributed controllers are work with the global view and local view methods in two or three layers. A network with two layers has multiple local controllers and a logically centralized root controller. These controllers collectively form a logically distributed control plane. Each switch is controlled by only one local controller, and each local controller can control multiple switches. If the root controller needs to install flow-entries on switches of a local controller, it delegates the requests to the respective local controller.

As illustrated in Table I Orion is logically distributed and has a flat or horizontal architecture the controllers are distributed horizontally on one single level. This leads to super-linear computational complexity growth of the control plane that limits the scalability of SDN networks when SDN network scales to large size. So, to solve this problem Orion designed as hybrid architecture. In a flat architecture, each controller has a partial view of the network and has the same responsibilities at the same time.

In Table 1 all of the controllers have hierarchical or vertical architecture. The controllers are placed vertically and they are distributed among multiple levels, currently two or three layers. Unlike flat architecture, controllers have different responsibilities and can make decisions based on a partial view of their network.

In the hierarchical architecture if the root controller failed the global view or the communication between the local controller and the global controller is failed. In addition, communication among the local controller and switch also failed. Both flat and hierarchical architecture have improved the performance and scalability of the controller than a centralized controller.

ONOS and OpenDaylight have much better northbound interfaces, management interfaces, and Graphical User Interface (GUI).

East/westbound interfaces are an interface that imports and export data among controllers. As shown in Table I except OpenDaylight all of the controllers support these interfaces. Eastbound interface interconnecting current IP networks with SDN networks and westbound interface delivered information among distributed controllers in diverse areas and enable them to manage distributed controllers.

In addition, all of the controllers support different versions of the OpenFlow protocol to vertically separate the control plane and data plane. Furthermore, as we can see all of the controllers have weak documentation.

6. Conclusion and Future Work

Software-defined networking is an emerging network architecture that separates the control plane and the data plane. Today SDN network is a hot research area due to this researcher and network organizations highly contributed to this network. In this paper, SDN distributed controller's architecture, and way of communication was investigated. Following compared logically centralized and logically distributed SDN distributed controllers. Logically distributed controllers are better than logically centralized controllers in performance, availability, and scalability. Using hybrid (flat and hierarchical architecture) can solve root controller failure and super-linear computational complexity problems. Finally, Orion is the first best controller and ONOS, OpenDaylight, Kandoo HyperFlow and Onix are the next best controllers respectively.

Future work will focus completely distributed controller architecture to minimize discover time a which is occurred when a local controller fails the device to discover another local controller this takes time. Distributing SDN controllers leads distributed controller placement problem this directly affects the controller's performance will focus on to determine how many controllers to use and where to place them. Also, developing and integrating new applications into the northbound interface, east/west interface that supports distributed controllers. Furthermore, it will focus on analyzing and improving the SDN distributed controller's performance.

References

[1] K. H. Matthew, WHERE WIZARDS STAY UP LATE THE ORIGINS OF THE INTERNET, New York: Simon & Schuster, 1998.

[2] T. S. A. S. R. A. Y. L. Manar Jammal, "Software-defined networking: State of the art and research challenges," *Computer Networks,* pp. 75- 79, 25 July 2014.

[3] J. R. E. Z. N.Feamster, "The Road to SDN:An intellectual history of programmable networks," *ACM SIGCOMM Computer Communication Review,* vol. 44, no. 2, pp. 1-11, 14 April 2014.

[4] D. Building, "Enterprise Networking vs Data Center Networking," YourDailyTech, 11 January 2018. [Online]. Available: https://yourdailytech.com/networking/enterprise-networking-vs-data-center-networking/. [Accessed 17 April 2018].

[5] ONF, "Listing mambers (Ouer Members)," ONF, 03 February 2018. [Online]. Available: https://www.opennetworking.org/our-members. [Accessed 15 April 2018].

[6] D. G. Z. H. T. Q. P. L. Junjie Xie, "Control plane of software defined networks: A survey," *Computer Communication,* pp. 1-10, June 2015.

[7] M. M. X. N. N. K. O. T. Bruno Nunes Astuto, "A Survey of Software-Defined Networking: Past, Present, and Future of Programmable Networks," *Communications Surveys and Tutorials,* vol. 16, no. 3, pp. 1617-1634, 19 January 2014.

[8] F. V. P. C. R. S. S. Diego Kreutz, "Software-Defined Networking: A Comprehensive Survey," *Software defined networking,* vol. 103, no. 1, pp. 1-61, 31 January 2015.

[9] M. B. M. R. B. F.Benamrane, "Performances of OpenFlow-Based Software Defined Networks: An overview," *JOURNAL OF NETWORKS,* vol. 10, no. 6, pp. 2-4, 24 JUNE 2015.

[10] Farmingiham, "Datacenter Networks," IDc, 05 January 2017. [Online]. Available: http://www.idc.com/getdoc.jsp?containerId=IDC_P13. [Accessed 05 January 2017].

[11] ON.LAB, "Introducing ONOS a SDN network operating system for Service Providers," ON.LAb, England, 2016.

[12] S. H. Y. a. Y. Ganjali, "Kandoo: A Framework for Efficient and Scalable Offloading of Control Applications," *Computer-communication networks,* pp. 19-24, 13 August 2012.

[13] J. B. K. G. Z. C. J. W. B. H. Yonghong Fu, "Orion: A Hybrid Hierarchical Control Plane of Software-Defined Networking for Large-Scale Networks," *2014 IEEE 22nd International Conference on Network Protocols,* pp. 569-576, 2014.

[14] M. A. A. I. Zuhran Khan Khattak, "Performance Evaluation of OpenDaylight SDN Controller," *Software Defined Networking Benchmarking,* pp. 1-6, 23 December 2014

[15] F. L. J. G. H. X. Tao Wang, "Dynamic SDN Controller Assignment in Data Center Networks: Stable Matching with Transfers," in *IEEE INFOCOM 2016 - The 35th Annual IEEE International Conference on Computer Communications,* Hong Kong, 2016.

[16] B. P. G. G. M. Abubakar Siddique Muqaddas, "Inter-controller Traffic in ONOS Clusters for SDN Networks," *Next-Generation Networking and Internet*

Symposium, vol. 9, no. 05, pp. 1-6, 2016.

[17] G. Rahim Masoudi, "Software defined networks: A survey," *Journal of Network and Computer Application*, pp. 1-25, 26 March 2016.

[18] M. B. M. a. R. B. Othmane Blial, "An Overview on SDN Architectures with Multiple Controllers," *Journal of Computer Networks and Communications*, pp. 1-8, 14 April 2016.

[19] N. McKeown, "OpenFlow: Enabling innovation in campus networks," *SIGCOMM Comput. Commun. Rev.*, p. 69–74, March 2008.

[20] Y. W. C. H. F. Wenfeng Xia, "A Survey on Software-Defined Networking," *IEEE COMMUNICATION SURVEYS AND TUTORIALS*, vol. 17, no. 1, pp. 1-25, 24 September 2015.

[21] A. L. B. H. e. a. B. Pfaff, OpenFlow Switch Specification, Open Networking Foundation, 2014.

[22] V. A. M. K. SHIVA ROWSHANRAD, "PERFORMANCE EVALUATION OF SDN CONTROLLERS: FLOODLIGHT AND OPENDAYLIGHT," *IIUM Engineering*, vol. 17, no. 2, pp. 47-56, 30 November 2016.

[23] Z. R. M. K. B. Rahamatullah Khondoker, "Feature-based Comparison and Selection of Software Defined Networking (SDN) Controllers," *Controller Comparsion*, vol. 10, no. 03, pp. 1-7, 2015.

[24] J. K. S. v. d. M. S. W. Andrei Bondkovskii, "Qualitative Comparison of Open-Source SDN Controllers," *Network Communication*, pp. 1-6, 2015.

[25] O. S. I. E. A. K. A. Chehab, "SDN Controllers: A Comparative Study," pp. 1-6, 05 October 2017.

[26] F. F. P. C. G. S. V. L.Zuccaro, "Distributed control in virtualized networks," *The 10th International Conference on Future Networks and Communications (FNC 2015)*, p. 276 –283, 2015

[27] H. C. Z. W. S. C. Guozhen Cheng, "DHA: Distributed Decisions on the Switch Migration Toward a Scalable SDN Control Plane," *National Digital Switching System Engineering & Technological R&D Center*, pp. 1-9, 03 December 2015.

[28] P. Z. L. X. X. L. C. H. Huanzhao WANG, "Asecure and high-performancemulti-controller architecture for software-defined networking," *Security, Multi-controller, Distributed rule store*, pp. 634-646, 8 August 2016.

[29] Tootoonchian and Y. Ganjali, "HyperFlow: a distributed control plane for OpenFlow," *In Proceedings of the Internet Network Management Conference on Research on Enterprise Networking (INM/WREN '10*, pp. 1-7, 2010.

[30] M. G. J. H. Y. H. Pankaj Berde, "ONOS: Towards an Open, Distributed SDN OS," *Computer-Communication Networks*, pp. 1-6, 22 August 2014.

[31] K. Simon Hunt, "The ONOS GUI," onosproject, 12 January 2018. [Online]. Available: https://wiki.onosproject.org/display/ONOS/The ONOS web GUI.html. [Accessed 21 January 2018].

[32] L. Foundation, "OpenDaylight Home page," Linux foundation, 12 March 2018. [Online]. Available: https://www.opendaylight.org/home. [Accessed 24 April 2018].

[33] M. N. J. L. M. R. Y. H. T. T.Koponen, "Onix: A distributed control platform for large-scale production networks.," *International*, vol. 10, p. 1–6, 2010.

[34] W. is.com, "Distributed Control Plane Architecture (DCPA)," Techtarget, 18 January 2018. [Online]. Available: http://searchsdn.techtarget.com/definition/Distributed-Control-Plane-Architecture-DCPA.

[35] O. Group, "Cluster Coordination," wiki, 10 January 2018. [Online].

[36] Available: https://wiki.onosproject.org/display/ONOS/Cluster+Coordination. [Accessed 15 March 2018].

[37] F. A. G. A. G. U. K. S. A. A. e. a. Abdelaziz A, "Distributed controller clustering in software," pp. 1-19, 6 April 2017.

Worku Muluye received his first degree in Information Systems from Wollo University at Kombolcha Institute of Technology, Ethiopia in 2013 and his Master's Degree from the Adama Science and Technology University, Ethiopia in 2017. From August 2013 to October 2014 he was working as a graduate assistant and assistant lecturer at Wolkite University, Wolkite Ethiopia, to which he returned in 2017 as a full lecturer and researcher. His fields of interest cover Software-Defined Networking, Li-Fi, Artificial Intelligence, Natural Language Processing, and Image Processing.