Decentralizing Software-Defined Wireless Mesh Networking (D-SDWMN) Control Plane

Hisham Elzain, Wu Yang

Abstract—Wireless Mesh network (WMN), serves as a key to enable wireless communication technology for indoor, enterprise and community networks besides the hard-wired networks. Characteristics features of WMN promise to provide a highly scalable wireless backbone. Despite this promise, the current WMN architecture is inflexible to fulfill the recent explosion of smart services and applications requirements. Moreover, network management itself is a difficult task. Software-Defined Networking (SDN) presents flexible network architecture, it creates centralized network control, offers simple and programmable forwarding devices totally abstracted from network applications. However, to improve reliability, scalability, and performance of SDN architectures, a number of logically centralized but physically distributed control plane design approaches have been proposed in order to overcome the expected bottleneck in a physical centralized approach. In this paper, firstly, we highlight the challenges of current SDN distributed control plane. Secondly, we propose Decentralized Software-Defined Wireless Mesh Network (D-SDWMN) architecture to address these issues in SDWMN. D-SDWMN is a framework that presents physical and logical distributed SDWMN control plane. D-SDWMN adopts a hierarchy distribution of controllers that can adapt to WMN distributed nature. Distributed local controllers enhance WMN's scalability and reliability. Furthermore, they boost main controller performance by decreasing control flow overhead. That positively affects the WMN performance. The initial results evaluation shows the quality of D-SDWMN properties.

Index Terms— WMN, SDWMN, SDN, Distributed Control Plane.

I. INTRODUCTION

WIRELESS Mesh Network (WMN) is considered to be one of the promising wireless technologies to create wireless multi-hop backbone network. It provides a coverage service for a variety range of applications in both urban and rural areas, like transportation, enterprise networks, mining fields, public safety and emergency response. That is because of its quick and relatively low-cost deployment. WMN characterized by a dynamic and distributed network control (protocols) and challenged by management complexity, scalability, and reliability issues [1]. Figure 1 illustrates a typical WMN architecture.

Currently growing of Internet of Things (IoT) applications and high demand for services like (cloud services, big data, video traffic, among others), across large-scale networks to the Internet, is a big deal for networks operators and Internet service providers (ISPs). In order to address the mentioned demands of networks' users efficiently, that necessitate changing in existing IP network architecture to be able to dynamic and auto-configuration network elements responding to new policies and business requirements.

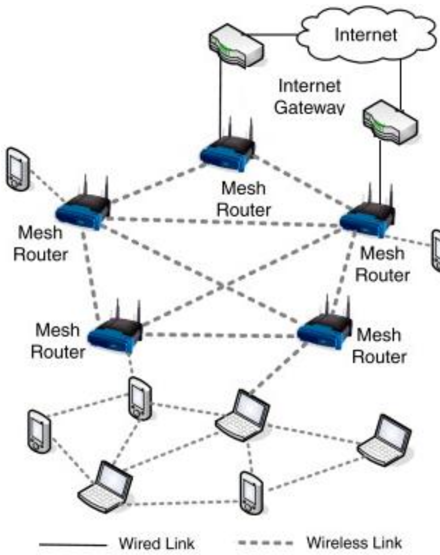


Fig. 1. A typical Wireless Mesh Network Architecture

Therefore, the evolution of current networks motivated the emergence of Software-defined networking (SDN) as a new paradigm in networking that aims to achieve manageable and flexible network architecture by defining, a higher-level of abstraction layers connected by clearly defined interfaces, which makes programming network behavior easier using high-level languages [2], [3].

In contrast to traditional networking, the essential concept of SDN is the separation between the network data and control planes. Therefore, in SDN network architecture, the forwarding devices (data plane) are isolated from their decision-making capability and centralized in control unit (control plane) called SDN controller. For more reliability and scalability issues, centralized control does not mean physical centralization, which may lead to present the SDN controller as a single point of failure. SDN distributed control plane [4], [5] is supposed to solve this problem.

Manuscript received March 24, 2018; revised April 05, 2018.
Hisham Elzain, Ph.D. Student with Harbin Engineering University, College Of Computer Science and Technology, Information Security Research Center, Harbin, China, (email:Hisham_alzain21@yahoo.com).
Wu Yang, Professor with Harbin Engineering University, College of Computer Science and Technology, Information Security Research Center, Harbin, China, (email: yangwu@hrbeu.edu.cn).

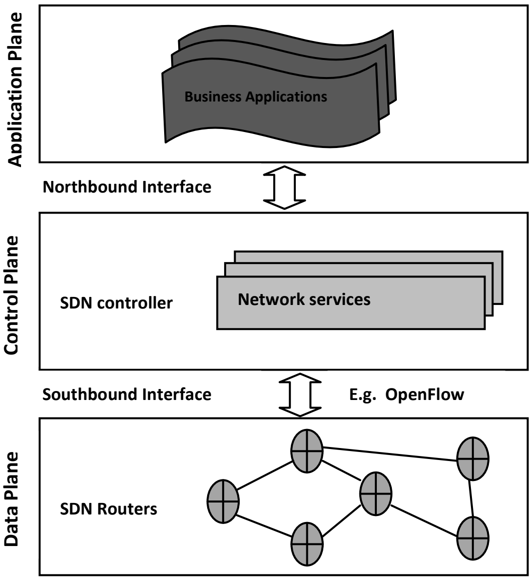


Fig. 2. SDWMN General Architecture

The combination of SDN concepts and WMN architecture is called Software-Defined Wireless Mesh Networking SDWMN [6]. This approach is challenging due to combining different control natures (i.e. Centralized nature of SDN with distributed nature of WMN). Figure 2 shows the basic logical view of the SDWMN. Where the infrastructure layer resides at the bottom, it consists of a set of SDN-enabled routers. Routers interconnect to provide network's data plane functionality.

In the middle, there is the control layer (the focus of the paper), consisting of software-based SDN controller/s. The controller provides and maintains a global view of the network to the application layer. It abstracts the complexity of managing, configuring and discovering the distributed network nodes, and maintaining connectivity among them. The interaction between the controller and the infrastructure layer takes place through the southbound interface. OpenFlow is the dominant southbound standard, founded by Open Network Foundation (ONF) [7].

At the top of the SDWMN, architecture is application layer, which is responsible for defining and implementing network policy requirements. The northbound interface is the interface between the control layer and the application layer. There is no any standard is defined for northbound till now [3].

With such flexible architecture provided by SDN, the WMN becomes more prepared for programmability and innovation at a higher rate. Therefore, new network applications, policies, and services can be simply implemented through applications (such as routing, load balancing, firewalling, etc.) running on the top of the controller. The controller then translates these needs in instructions to the data plane elements (mesh routers).

Current SDN distributed control plane is a challenge, - to various types of SDN networks - in terms of controller design complexity and managing. Hence, controller developers should take into consideration distributed control plane limitations, like controller coordination-related issues such as synchronization, replication, and placement. Moreover, decouple them from the fundamental functionality of the controller [8].

To address these issues in SDWMN, we propose Decentralize-SDWMN (D-SDWMN), a framework that distributes WMN based-SDN control plane physically and logically, forming two control layers: global and local control layer. The SDN controllers distributed hierarchically between the two layers, both of the two layers discussed in details later. The efficiency of D-SDWMN qualitatively evaluated in terms of reducing the SDN controller overhead load, increased network (scalability and reliability) and overall control plane better managing.

II. CURRENT SDN CONTROL PLANE

Initially, SDN has emerged as centralized control plane architecture as defined by ONF [9]. But, due to the adverse consequences in network scalability, reliability and performance - expected when adopting SDN centralized control plane architecture - a logically centralized, physically distributed, control plane has been proposed to address these problems[10].

Although the distributed control plane approaches incur different complexities to develop and manage SDN controllers. However, these solutions are more responsive to network status changing and handle related events, because the controllers distribute through the network and closer to resources of events than in the centralized architecture.

The authors in [11] discussed the scalability issues of controllers. They presented a comprehensive survey of state-of-the-art works in distributed SDN control plane. Their work classifies the scalability of the control plane in two approaches: "Topology-related approaches and Mechanisms-related approaches".

There are many studies [12-14], that distribute the controllers to handle the network control plane workload based on topology. They follow topological structures on controllers' distribution in network architecture, such as flat and hierarchical designs [11]. In the flat distributed design, each controller has its own local domain view, and it knows the whole network view by communicating and exchanging with other controllers its state information. Different design from the flat structure is a hierarchical structure, in which the local distributed controllers only handle local control issues. They left the global network control to the main controller, which has more power of control and maintain a global network view by coordinating the communication to local controllers [10, 15, 16].

In this work, we adopted the hierarchical control plane design in SDWMN in band - (control plan shared with data plane [17], some routers should host a controller) - with suitable modifications to fit WMN architecture.

III. D-SDWMN OVERVIEW

In the SDWMN, the centralized controller controls all the routers of the network. That means each router in the network needs to rely on the controller forwarding decision. Thus, for every new flow, the router generates a request to the controller, which responds with appropriate flow entries

ISBN: 978-988-14047-9-4
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

message(s). This scenario obviously costs the controller more load and decrease the network performance.

Alternatively, in D-SDWMN the control plane divided into two control layers: the Global Control Layer (GCL) and the Local Control Layer (LCL). The substantial difference between two layers is that GCL maintains a global view of the entire network, and the LCL defines the local control of the distributed control domains.

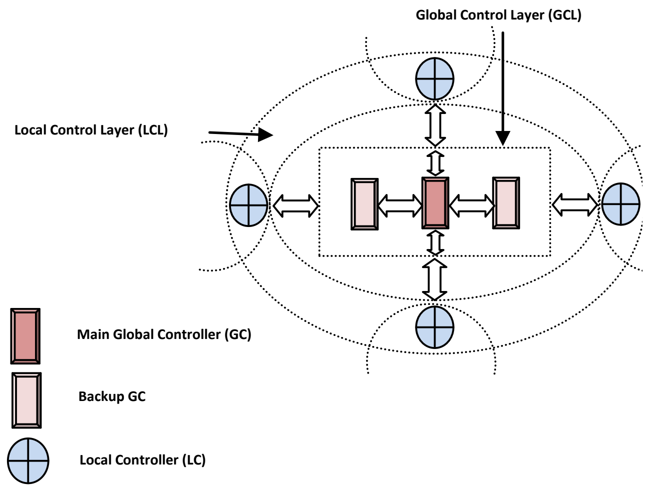


Fig. 3. D-SDWMN control Layers

The main Global Controller (GC) and GCs backup controllers represent the Global Control Layer. The main GC selects some routers to act as Local Controllers (LCs) upon its authorization. In addition, each LC builds its control domain. As a result, any new flow arriving at domain's nodes does not need to request the GC. Instead, the corresponding LC directly handles the flow. The local control domains shape the Local Control Layer. Figure 3 shows the layers of D-SDWMN.

The hierarchical architecture of D-SDWMN is built by the distribution of LCs hierarchically with the GC as in section IV-A. The initial state of the main GC is to be able to receive SDN-routers requests, and since each router has pre-entries (e.g. GC controller IP & port no. and respond to delegated LC) to reach the GC controller or respond to an LC. As initial handshake, when a router powered-on, it sends advertise message, which contains its identification-information and capabilities (e.g. transmission power PX and location), seeking for GC controller. Then The GC starts the authentication procedure, and the process going on through control-follow messages as in IV-B.

GC-LC communication: control delegation is one of GC-LC communication scenarios. Also as a fault-tolerant behavior of the GC, it informs the LCs with the GC-backup controllers.

Other communications scenarios between GC and LCs are connectivity maintaining and topology discovery. As connectivity is a critical issue of concern for wireless networks; D-SDWMN architecture maintains nodes connectivity via periodic Hello messages as keep-alive signaling sending from LCs to the GC, and the replies of these messages inform the existence of GC. The same happens in a small-scale scenario within a local domain, when the LC receives Hello messages from domain members.

In topology discovery messaging, the LC sends a list of its domain topology one time and resends updates when needed. In such case, a router goes-down (not send Hello message in time) LC updates its topology database and sends topology update message to the GC.

Fault Tolerance: the D-SDWMN enhances network reliability by programmable fault tolerant behavior. In case of the main GC failure, the LCs connects directly to one of the pre-programmed backup GCs. When one of the LCs misses the connection to the GC, which knows all the nodes of particular LC's domain, it selects a closer router to the missed LC and grants it domain control privilege. The old LC's controller automatically goes to sleep after a particular time.

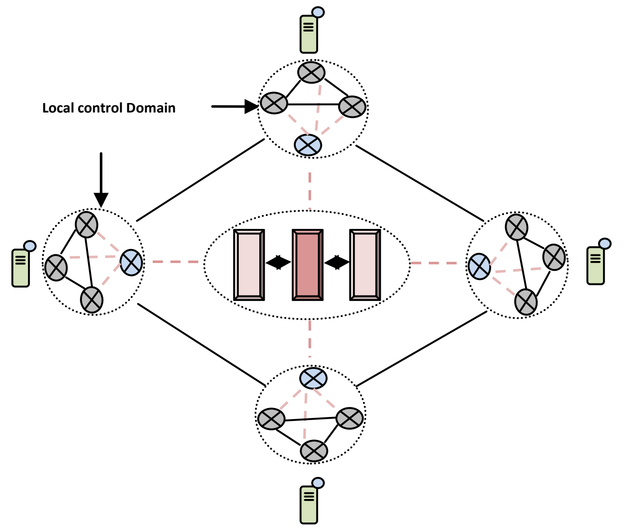
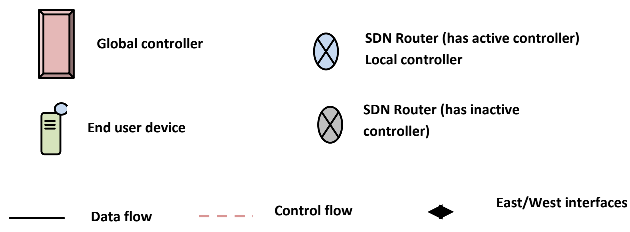


Fig. 4. D-SDWMN Architecture



IV. D-SDWMN DESIGN AND IMPLEMENTATION

A. *D-SDWMN Architecture*

In this section, we present the architecture of D-SDWMN and its implementation details. Figure 4 depicts the architecture of D-SDWMN. The GC and LC functionalities as follow:

Global Control (GC): The GC structure is made up of several modules as follows:

1) **Authentication:** this is an assistant module. It confirms the identity of the router that requests to reach the GC. Identity verification is done according to specifications set by the network operator. Then this module informs the next module with trusted routers.
2) **Control-plane topology:** it is a basic module of D-SDWMN. With it, the network operator can specify the control plane initial setting such as a number of nodes

ISBN: 978-988-14047-9-4
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

in local domains. By default, the basic control plane topology initials with one main centralized controller and two backup controllers. Then this module grants the control delegation to the selected routers, and adds them as LCs to the control plane, using information from the former module (i.e., authentication).

3) **Global-network topology discovery:** Through this module, the main controller discovers the routers, hosts, and links in the architecture. Firstly, it recognizes LCs using the information from the previous module (control-plane topology). In addition, it sends LLDP (Link Layer Discovery Protocol) to LCs to discover the links between them. Secondly, it requests a topology of each local domain, to build the network entire topology model.
4) **Control-plane synchronization:** this module specifies how GCL synchronize. It uses to synchronize the state of backup controllers with the main controller. POX controller uses Zookeeper [18] as a synchronization mechanism. In this mechanism, the main controller reports its state (e.g. control-plane topology) to a file. Then this file replicated across the backup controllers to synchronize their state to the main controller.

Local Control (LC): Each local domain of the LCL consists of a number of SDN-Routers controlled by one LC. The important module of LC is Local-topology discovery. It is a basic module of LC controller. It discovers the routers, hosts, and links in the LC's local domain. It uses LLDP packets exchange between the LC controller and routers to build the domain topology. The LC sends a list of the domain topology to the Global-network topology discovery module in the GC to form entire network topology.

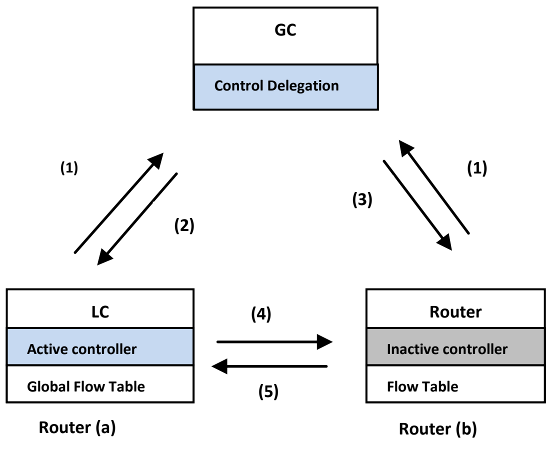


Fig. 5. A delegation of control from GC to a Router

Control Delegation Scenario

(1) Routers (a) & (b) initialize connection to GC.
(2) GC grants control delegation to Router (a), activates its controller to become a LC.
(3) Router (b) does not grant control delegation.
(4) LC advertises control service to neighbors.
(5) Router (b) responds and joins the LC's local domain.

B. D-SDWMN Operation

The following steps describe the work of D-SDWMN framework. How does it build its architecture, and how does it distribute the control plane (local controllers) through the architecture, into a dynamic, flexible, and fault-tolerant way.

1) Structure Initial setup: the scenario of building the D-SDWMN architecture initializes with one main GC and two backup GCs. The Control-plane synchronization module uses to synchronize backup GCs. At this state, the main controller is ready to receive requests from mesh SDN-Routers.
2) When GC received an advertise-request from a router. It checks the router's capabilities contained in the request. If the router is suitable to act as LC, the GC starts authentication module to authenticate the router. Then the GC sends the reply to Authenticated router with activation parameters to activate its controller and grants it a control delegation privilege as LC. Otherwise, the GC replies to the router, where it runs its second pre-set entry as in (4). Figure 5 illustrates control delegation scenario.
3) The router that acts as LC then sends requests to nearby routers to join its domain. Then the LC starts all the controlling functionalities on its domain (e.g. topology discovery).
4) The router that does not grant control delegation responds to the nearest LC's request and joins its domain.
5) The GC controller builds the control-plane topology of the D-SDWMN network. It updates control-plane topology adding any LC controller passed the check in step (2). Then, it discovers the links between LCs by using LLDP protocol. The LLDP protocol only discovers the directly connected nodes, because it is a single-hop protocol[19].
6) The GC controller starts the topology discovery module, which builds the Data-plane topology model of the network. As in section A, the topology will contain all the LCs controllers (treated here as an ordinary router) and the links between them. Then, this module receives the local domains topologies to maintain the global network view. Links properties are provided to use by network application like routing.
7) At this point, the D-SDWMN framework is working properly, which provide a favorable environment to the application (e.g., routing) on top of the control plane, to specify their requirements through the controller, to be executed efficiently by the network devices. Ultimately, WMN in D-SDWMN architecture provides a good coverage for users as expected.

C. D-SDWMN INTIAL PROTOTYPE IMPLEMENTAION DETILS

We use the mentioned below tools and technologies to realize D-SDWMN architecture and evaluate its characteristics.

1) SDWN Emulation: Mininet-wifi [20]
2) Wireless Routers: OpenVswitch [21]

ISBN: 978-988-14047-9-4
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

3) SDN Controller: POX Controller [22]
4) Distribution synchronization: Apache Zookeeper
5) Host: Virtual Box.

V. D-SDWMN QUALITATIVE EVALUATION

In this section, we provide and discuss the qualitative evaluation of D-SDWMN's capabilities. The exhaustive quantitative evaluation will be presented in our future work.

A. *Control plane management separation:*

As we showed in the D-SDWMN, the management of the control shares between GC and LC controllers. LCs can perform local tasks, like routing, topology discovery, connectivity maintaining among others within local domains. That mitigates the heavy load from the GC, which boosts its performance. However, it helps to provide separation of concern between global and local applications development.

B. *Network scalability:*

The GC can achieve dynamic control plan scaling, by LCs scale up or down. Network operators can easily programme this dynamic behavior according to the network state. Moreover, this affects network Data plane scalability.

Scale-up: GC needs to select idle router for activate its controller and add it to control plane as an LC. Therefore, any new router can connect to the nearest CL, which extends data plane and widens the coverage of WMN.

Scale-down: if the status of the network does not need a particular LC, the GC deactivates its controller and it returns as an ordinary router, and then it joins the nearest local domain. Other routers of the canceled domain also change to the nearest one. This mechanism also enhances the reliability of D-SDWMN architecture against LC failure.

VI. CONCLUSION AND FUTURE WORK

This paper, highlighted current SDN distributed control plane challenges, in terms of load overhead and complexity of the controller. To overcome these challenges in SDWMN, we proposed D-SDWMN platform that separates control load between the global centralized controller and distributed local controllers. We described the hierarchal architecture of D-SDWMN. In addition, how it enhances WMN scalability, reliability and performance. Our current qualitative evaluation shows the properties of D-SDWMN that stemming from the ease of controller behavior development to share network control load. The future work will present rigorous quantitative results evaluation of D-SDWMN.

ACKNOWLEDGMENT

The International Exchange Program of Harbin Engineering University funds this paper, for Innovation-oriented Talents Cultivation.

REFERENCES

[1] Bokhari, F. and G. Záruba, Partially overlapping channel assignments in wireless mesh networks, in Wireless Mesh Networks-Efficient Link Scheduling, Channel Assignment and Network Planning Strategies. 2012, InTech.
[2] Nunes, B.A.A., et al., A survey of software-defined networking: Past, present, and future of programmable networks. IEEE Communications Surveys & Tutorials, 2014. 16(3): p. 1617-1634.
[3] Kreutz, D., et al., Software-defined networking: A comprehensive survey. Proceedings of the IEEE, 2015. 103(1): p. 14-76.
[4] Bannour, F., S. Souihi, and A. Mellouk, Distributed SDN Control: Survey, Taxonomy and Challenges. IEEE Communications Surveys & Tutorials, 2017.
[5] Chen, Y., et al., Adaptive Distributed Software Defined Networking. Computer Communications, 2017. 102: p. 120-129.
[6] Abujoda, A., et al., Software-defined wireless mesh networks for internet access sharing. Computer Networks, 2015. 93: p. 359-372.
[7] Azevedo, F., A SCALABLE ARCHITECTURE FOR OPENFLOW CONTROLLERS.
[8] Patil, P., A. Gokhale, and A. Hakiri. Bootstrapping Software Defined Network for flexible and dynamic control plane management. in Network Softwarization (NetSoft), 2015 1st IEEE Conference on. 2015. IEEE.
[9] Sezer, S., et al., Are we ready for SDN? Implementation challenges for software-defined networks. IEEE Communications Magazine, 2013. 51(7): p. 36-43.
[10] Dixit, A., et al. ElastiCon; an elastic distributed SDN controller. in Architectures for Networking and Communications Systems (ANCS), 2014 ACM/IEEE Symposium on. 2014. IEEE.
[11] Karakus, M. and A. Durresi, A survey: Control plane scalability issues and approaches in Software-Defined Networking (SDN). Computer Networks, 2017. 112: p. 279-293.
[12] Bari, M.F., et al. Dynamic controller provisioning in software defined networks. in Network and Service Management (CNSM), 2013 9th International Conference on. 2013. IEEE.
[13] Phemius, K., M. Bouet, and J. Leguay. Disco: Distributed multi-domain sdn controllers. in Network Operations and Management Symposium (NOMS), 2014 IEEE. 2014. IEEE.
[14] Berde, P., et al. ONOS: towards an open, distributed SDN OS. in Proceedings of the third workshop on Hot topics in software defined networking. 2014. ACM.
[15] Hassas Yeganeh, S. and Y. Ganjali. Kandoo: a framework for efficient and scalable offloading of control applications. in Proceedings of the first workshop on Hot topics in software defined networks. 2012. ACM.
[16] Santos, M.A., et al. Decentralizing SDN's control plane. in Local Computer Networks (LCN), 2014 IEEE 39th Conference on. 2014. IEEE.
[17] e Silva, B.R., et al., Multi Controllers Architecture with Adaptive Monitoring in SDWMN In-Band.
[18] Hunt, P., et al. ZooKeeper: Wait-free Coordination for Internet-scale Systems. in USENIX annual technical conference. 2010. Boston, MA, USA.
[19] Pakzad, F., et al. Efficient topology discovery in software defined networks. in Signal Processing and Communication Systems (ICSPCS), 2014 8th International Conference on. 2014. IEEE.
[20] Fontes, R.R., et al. Mininet-WiFi: Emulating software-defined wireless networks. in Network and Service Management (CNSM), 2015 11th International Conference on. 2015. IEEE.
[21] Open vSwitch 2.9.0 http://www.openvswitch.org/features/
[22] Kaur, S., J. Singh, and N.S. Ghumman. Network programmability using POX controller. in ICCCS International Conference on Communication, Computing & Systems, IEEE. 2014.