

COMPUTER NETWORKS VIRTUALIZATION COST EFFECTIVE AND DISTANCE LEARNING ACHIEVEMENT

FARIS KETI

Dept .of Electrical and Computer Engineering, University of Engineering, Kurdistan Region-Iraq

ABSTRACT

in the curricula of higher education faculties, computer networking is a main topic in computer Science courses. Previous methods of teaching a subject like Computer Networks has consisted of a face to face proposal, high cost labs and static and complex configuration of real networking devices restricted by in location attendance for experiment purposes and in lab work.

With computer networks virtualization, Students and researchers are allowed to do tests on real world networking configuration scenarios and to configure various and complex network scenarios by configuring virtualized equipments, such as routers and switches, through virtual consoles and Network virtualization and simulation tools .

In this paper an evaluation of the concept of computer networks virtualization and Software Defined Networks is conducted which can be used by the students to improve the learning process of computer networks subject in engineering studies. Emulating the same physical networks as in the laboratory, at home or remotely from anywhere with more flexibility and lower cost, all that through emulation tools installed on students computers.

KEYWORDS- Emulation, Virtualization, Software Defined Networking, Remote Labs.

I. INTRODUCTION

The community of computer networks researchers has been searching for ideas that enable the use of networks with more programming resources and less need for hardware elements replacement, so that new technologies designed to solve new problems can be inserted gradually into the network and without significant costs [1, 2].

Virtualization will soon allow decoupling network functions from network hardware. This will altogether reduce the network cost [3].

Software Defined Networking (SDN) is an emerging network architecture that virtualizes network infrastructure by separating the control and data plane logic of traditional network devices, creating a flexible, dynamic, automated and manageable architecture [1].

Few years ago, educational resources in engineering education become available on-line and downloadable which allow students to modify their learning process [4]. Although, the resources (Lectures, Handouts, and Lab sheets) are accessible from the web and almost free, but due to the traditional networks vertical integration, they are de-

signed, developed and prepared by lecturers and only used by their students. They are rarely developed by students in order to be used by other students. To achieve that, a new norm of computer networking called computer networks virtualization is required.

With computer networks virtualization the students have the advantages of (1) accessing the networks remotely and confirm the policies and necessary configuration. (2) Working in a virtualized environment through an open interface and not been restricted by real physical devices. And finally (3) the students can choose the controller based on their skills in any programming language that they are familiar with.

Section II of this paper discusses computer networks virtualization benefits and SDN paradigm as an example of networks virtualization by describing its motivation, network elements, and the operation among the elements. Section III describes the performance of Simulation and emulation tools in reducing the cost of computer networking laboratories, and the role of students in development of educational resources in addition to the achievement of distance learning. Section IV concludes the paper.

II. COMPUTER NETWORKS VIRTUALIZATION

One of the major issues currently facing IT departments is the problem of scale related to having consistent configuration across very large populations of networking devices [5].

The computer networks virtualization provides to students and end users some advantages such as: 1) without using real physical devices, the real traffic can be analyzed by the students. Therefore; the problems of configuration and connectivity are eliminated. 2) The routing techniques can be modified by the students due to the horizontal integration of virtualized networks. Special user authorization to

avoid machine damages or changes on the configurations is not required. 3) They can work remotely and from anywhere (home, job, etc.) outside the laboratory or the classroom. The remote lab and distance learning concepts can be performed [6].

In the real laboratories, not all changes are allowed because multiple students interact on the same devices and the changes can affect all of them, but with Software Defined Networking (SDN) there is the promise of great simplification, with the associated reduction in operational costs for the enterprise [7].

A comparison between traditional and SDN networks in term of configuration is in Figure 1.

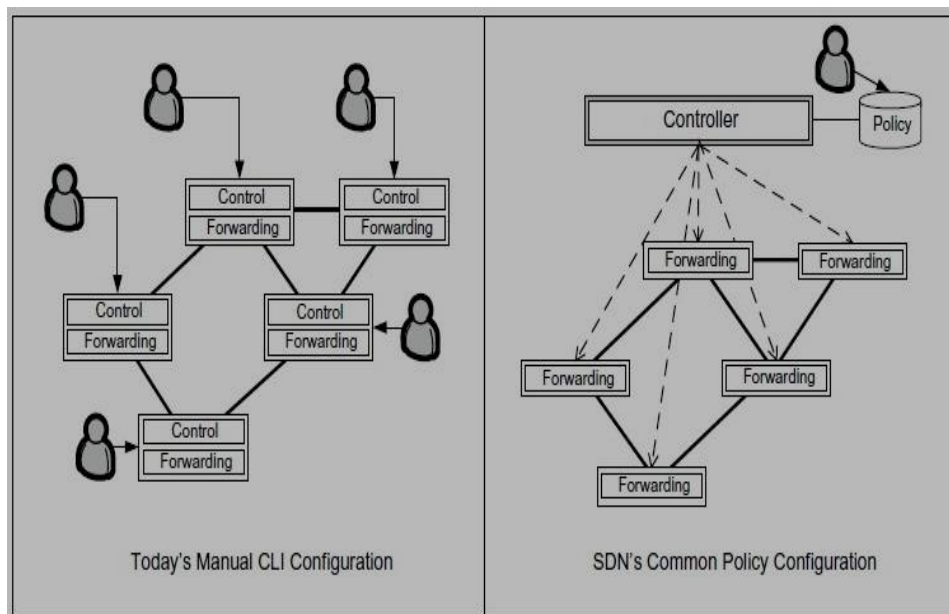


Fig.(1):- Comparison between Traditional and Software Defined Networks in term of configuration [5]

Software Defined Networking (SDN) is a new network paradigm that decouples the infrastructure (control and data planes) logic of traditional network devices [8].

SDN is built on the principle of providing an Application Programming Interface (API) for data plane (packet forwarding) devices such as switches

and routers, which allows the programmability of network elements and entire network, creating a dynamic, flexible, automated and manageable architecture [9].

Figure 2 shows the architecture of an SDN network.

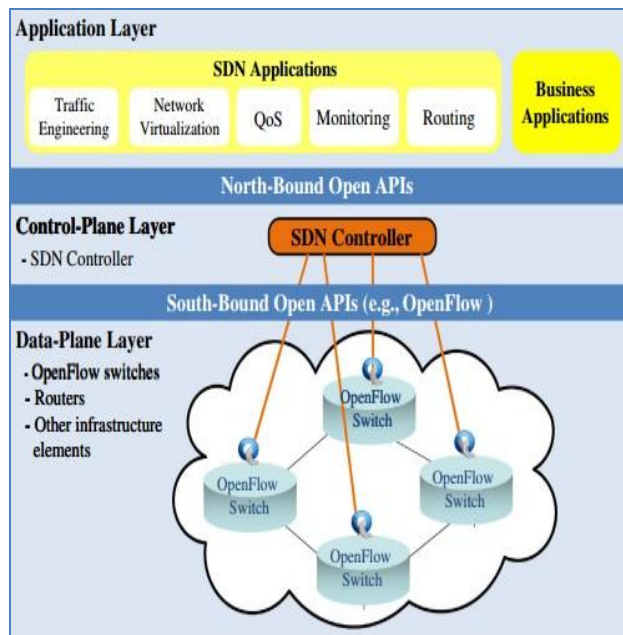


Fig.(2):- Software Defined Network Architecture [10]

Any software defined network consists of three main planes, namely; data plane, control plane, and management plane. Figure 3 shows the main soft-

ware defined network planes, their roles, and the interactions among them.

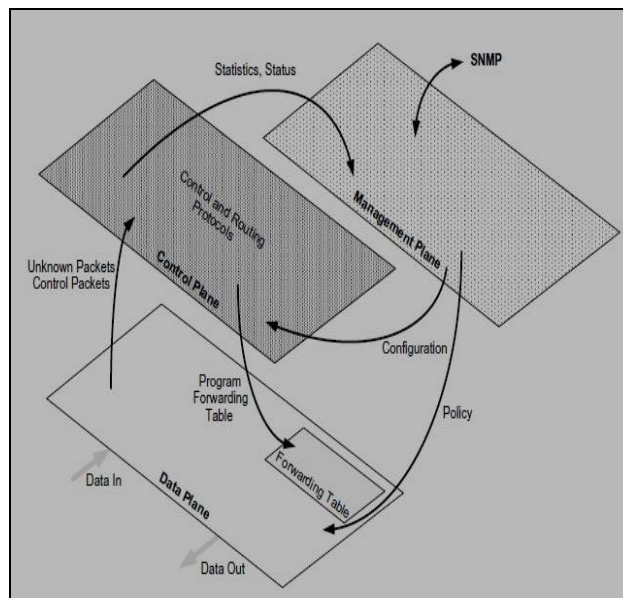


Fig.(3):- Role of SDN Planes [5]

Internet2 [11] will operate the U.S UCAN (United States Unified Community Anchor Network) program [12]. Their aim is to use software defined network principals to provide a platform for re-

search, educational and health care institutions to interconnect them together. Figure 4 shows a draft of the expected deployment.



Fig.(4):- Draft of the planned U.S. UCAN network using the Internet2 100G deployment [13].

So, for decades to come, SDN is on a way to be an important and permanent part of networking technology. Simply this is correct because the term SDN can cover any networking technology that includes a little bit of software control. Since all networking technologies contain software to some degree; therefore, they are all SDN [5].

III. COST EFFECTIVE AND DISTANCE LEARNING ACHIEVMENT

In this section many tests are presented which explain some features and advantages of computer networks virtualization for students and end users to analyze the real traffic without using real physical devices. Also, how they can work from other places (home, job, library, etc.) outside the classroom/laboratory.

The use of emulation tools to overcome the need of costly real networking devices and to eliminate the problems of configuration and connectivity, the possibility of movement from vertical integration of traditional networks to the horizontal integration of virtualized networks due to the students ability to participate in the process of education resources preparation, and the opportunity of working remotely (home, job, etc.) outside the laboratory or the classroom, So the remote lab and

distance learning concepts can be performed are covered in this section.

Simulation and emulation software is very important for quick prototyping and testing without the need for expensive physical devices. Mininet is the earliest emulation system that provides a fast, simple, and easy way to prototype and evaluates SDN protocols and applications [14].

The major key property of Mininet emulator is that controllers or applications created, developed, and tested in the emulated environment can be easily deployed in a real software defined network without any modification. Users (students) can easily emulate an SDN network with hundreds of nodes and dozens of switches by using a single personal computer [15].

The capabilities of Mininet enable students, researchers, and network programmers to prototype software defined networks in a simple manner [14]. But the characteristics and qualifications of simulation environment should be taken into consideration so Mininet can be utilized as one of the powerful tools in emulating the SDN and virtual networks [16].

At this part, some tests are done to explain some features and capabilities of Mininet emulator [17] along with POX [18] and HP Virtual Application Networks (VAN) [19] SDN controllers soft-

ware which can be used by the students to analyze the real traffic without using real physical devices (The configuration and connectivity problems are reduced or eliminated). Also, they can work from other places (home, job, library, etc.) outside the classroom/laboratory.

Oracle VM Virtual Box (Virtualization Software), Linux (Ubuntu) as Host Operating System, Mininet (Network Emulator), Python (Programming Language), and POX and HP VAN (SDN Controller Platform) are the software used in implementation and tests of this paper.

There are many SDN controllers exist, however, POX and HP Virtual Application Networks

(VAN) SDN controllers are the selected controllers for this test. POX is a software platform developed in Python [20] and HP Virtual Application Networks (VAN) SDN Controller Software provides a unified control point in an SDN network, simplifying management and provisioning.

Mininet tool has the ability to create different network topologies. The SDN linear topology is one of those many topologies. It consists of specified number of switches each with a single host connected to it, the programmer or the user (student) is the one who determines the number of switches. A linear topology of 4 switches and similar number of hosts with a single host connected to each switch is shown in Figure 5.

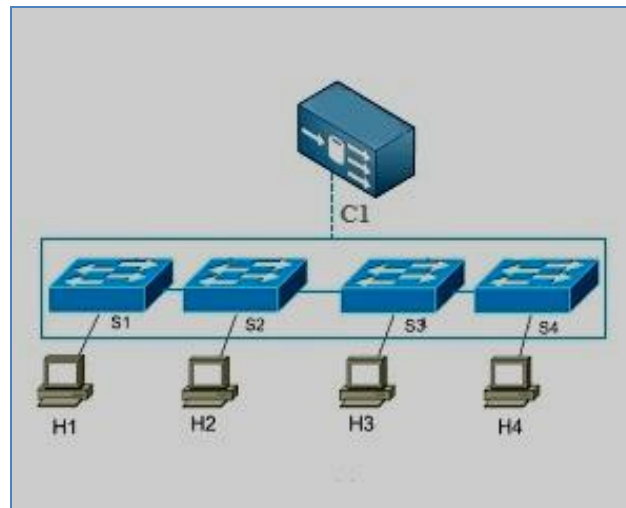


Fig. (5):- Linear topology

The script in Figure 6 bellow is the script of building the linear topology.

```
mininet@mininet-vm:~$ sudo mn --topo linear,4 --controller remote
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2 h3 h4
*** Adding switches:
s1 s2 s3 s4
*** Adding links:
(h1, s1) (h2, s2) (h3, s3) (h4, s4) (s1, s2) (s2, s3) (s3, s4)
*** Configuring hosts
h1 h2 h3 h4
*** Starting controller
*** Starting 4 switches
s1 s2 s3 s4
*** Starting CLI:
mininet>
```

Fig.(6):- The script of creating linear topology

After the network topology has been built by using MININET emulator and POX controller. To view the created network, the hosts and switches IP and MAC addresses, the flow table entries and all the specification and configuration policies of SDN networks. The direction of controller can be simply changed from POX controller to HP VAN

SDN controller by restarting the emulator with VAN controller and adding the IP address of the last to the instruction as below:

```
$ sudo mn --topo Linear,4 --controller=remote,ip=192.168.56.93 --mac
```

The created topology captured from HP VAN controller window is shown in Figure 7 below.

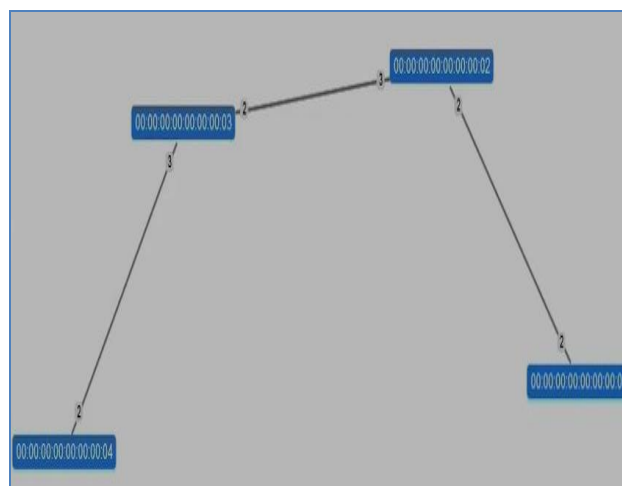
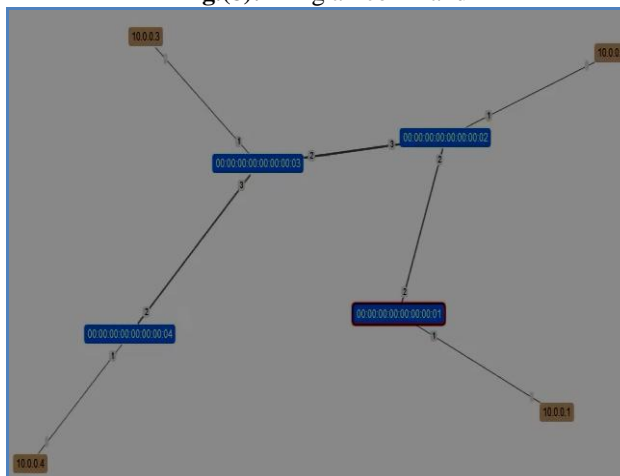


Fig.(7):- Linear topology switches

To view the hosts connected to the switches a (pingall) command is executed as in Figure 8 and the network of Figure 9 is displayed.

```
mininet> pingall
*** Ping: testing ping reachability
h1 -> h2 h3 h4
h2 -> h1 h3 h4
h3 -> h1 h2 h4
h4 -> h1 h2 h3
*** Results: 0% dropped (12/12 received)
```

Fig.(8):- Ping all command



Fi.(9):- Linear topology switches and hosts

Different SDN network topologies can be created with Mininet. Figure 10 display some of those topologies.

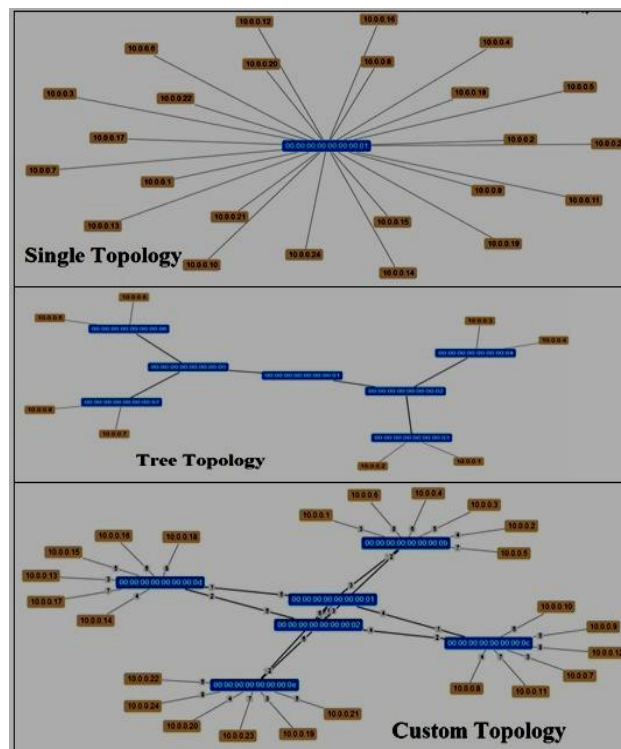


Fig.(10):- Different SDN Topologies.

From VAN SDN controller main window all the information about SDN topology, OpenFlow [21] protocol messages, connectivity, IP addresses, MAC addresses, port numbers, shortest paths available in the network topology, and many other

capabilities regarding the status of flow tables and flow matches of OpenFlow switches can be tested and analyzed simply by the students.

A view of VAN SDN controller main window is shown in Figure 11 below.

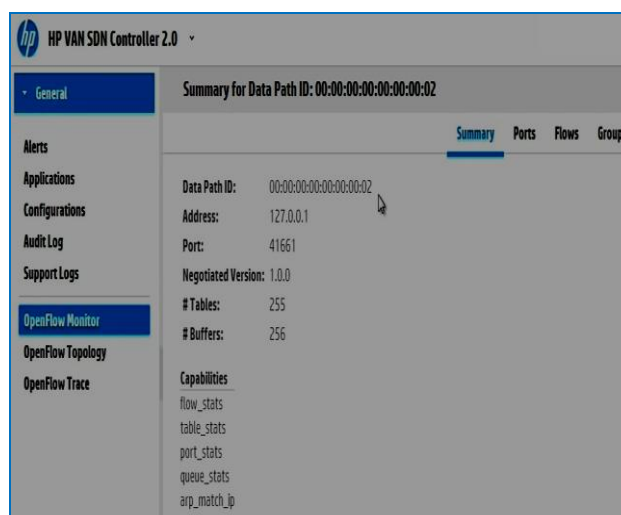


Fig.(1):- VAN SDN controller main window

Regarding the role of students in the process of their educational resources preparation, in addition to their ability of developing new applications or defining new network management policies, the student has the opportunity to do his lab job in his favorite programming language.

For example if it is considered that there are a class of three groups of students (group A), (group B), and (group C). Each group having the skills of a different programming language (C++), (JAVA), and (Python) respectively. With traditional educational resources and computer networks labs, the prementioned groups were to work on the same physical devices exist in the lab and using the same programming language comfortable to the devices command lines. With this new norm of networking, then each group of students can work on the same lab job or homework with individual controller that support the programming language of his skill.

The students of (group A) which have a (C++) programming language skills can work on NOX SDN controller, students of (group B) on Floodlight and Beacon SDN controllers which support (JAVA) programming language, while (group C) students have the opportunity to work on POX SDN controller using their (Python) programming language skills.

IV. CONCLUSIONS

This paper declared how the emulation tools have an impact on the cost of computer networking laboratories and how it eliminates the configuration problems of traditional physical devices. Also, how every student can have a role in his educational resources preparation process. Different controllers, wide range of topologies, multiple choices of programming languages, and the possibility of applications development all through open source emulation tools and remote virtualized and centralized network controllers are available for the students.

From the previous tests, it is concluded that in addition to the flexibility of accessing remote controllers or taking advantage from the open interface to develop new applications remotely. The students can overcome the vertical integration of traditional education resources prepared by the lecturers. The students were enforced to be re-

stricted by what is in the resources without any chance of innovation. So, with computer networks virtualization the students can have a role in the improvement of the learning process of computer networks subject in engineering studies.

REFERENCES

- [1] Guohui Wang, T. S. Eugene Ng, Anees Shaikh, "Programming your network at run-time for big data applications", HotSDN'12, Helsinki, Finland, August 13, 2012.
- [2] Rogério Leão Santos de Oliveira, Ailton Akira Shinoda, Christiane Marie Schweitzer, Ligia Rodrigues Prete, "Using Mininet for Emulation and Prototyping Software-Defined Networks", Brazil, ©2014 IEEE.
- [3] Nathalie Omnes, Marc Bouillon, Olivier Le Grand, "A Programmable and Virtualized Network & IT Infrastructure for the Internet of Things: How Can NFV & SDN Help For Facing The Upcoming Challenges", Lannion, France, ©2015 IEEE.
- [4] Gil, P., Candelas, F.A., Jara, C.A, Garcia, G.J, "Web- Based OERs In Computer Networks." Int. J. of Engineering Education, Spain, November 2013.
- [5] Paul Goransson, and Chuck Black, "Software Defined Networks: A Comprehensive Approach", (1st ed.), Copyright © 2014 Elsevier Inc. All rights reserved.
- [6] Pablo Gil, Gabriel J. Garcia, Angel Delgado, "Automatic network configuration in virtualized environment using GNS3: Evaluating a solution to optimize resources and achieve a distance learning", San Vicente del Raspeig (Alicante), Spain, ©2014 IEEE.
- [7] Sreekanth Sasidharan, and Saurav Kanti Chandra, "Defining future SDN based network management systems: Characterization and approach", IEEE 5th ICCCN conference, Hefei, China, July 11-13, 2014.
- [8] O. W. Paper, "Software-Defined Networking: The New Norm for Networks," Open Networking Foundation, Tech. Rep., April 2012.
- [9] Markus Vahlenkamp, Fabian Schneider, and Dirk Kutscher, "Enabling Information Centric Networking in IP Networks Using SDN", NEC Laboratories Europe, Heidelberg, Germany, 2014.
- [10] Ian F. Akyildiz, Ahyoung Lee, and Pu Wang, "A roadmap for traffic engineering in software defined networks", computer networks journal, 2014 Elsevier B.V. All rights reserved ©2014 Elsevier.
- [11] Internet2. Nation's First 100G Open, Nationwide, Software-Defined Network Launches for Education, Research, Industry and Innovators. [Online]. Available: <http://internet2.edu/news/>

- [12] United States Unified Community Anchor Network. United States Unified Community Anchor Network. [Online]. Available: <http://www.usucan.org/about>
- [13] Adrian Lara, Anisha Kolasani, and Byrav Ramamurthy, "Network Innovation using OpenFlow: A Survey", IEEE Communications Surveys & Tutorials, VOL. 16, NO. 1, First Quarter 2014, ©2014 IEEE.
- [14] B. Lantz, B. Heller, and N. McKeown, "A network in a laptop: rapid prototyping for software-defined networks," in Proceedings of the 9thACMSIGCOMM Workshop on Hot Topics in Networks. ACM, 2010.
- [15] Diego Kreutz, and Fernando M. V. Ramos, et al., " Software Defined Networking: A Comprehensive Survey", ©2014 IEEE..
- [16] Faris Ketil and Shavan Askar, "A New Investigation of Mininet Emulator for Evaluating Software Defined Networks Performance", Journal of University of Duhok, Vol. 18, No.1 (Pure and Eng. Sciences), Pp 176-185, 2015.
- [17] Mininet: An Instant Virtual Network on your Laptop. [Online] Available: <http://mininet.org>.
- [18] POX, "Pox SDN controller", [Online]. Available: <http://www.noxrepo.org/pox/about-pox>.
- [19] HP Virtual Application Networks (VAN) SDN Controller Software, [Online]. Available: <https://marketplace.saas.hpe.com/sdn/content/sdn-controller-free-trial>.
- [20] Python Software Foundation, "Python language reference, version 2.7 " , [Online]. Available: <http://www.python.org>.
- [21] OpenFlow Protocol, The Openflow Switch, [Online]. Available: <http://www.openflowswitch.org>.