International Journal of Scientific Engineering and Research (IJSER)
ISSN (Online): 2347-3878
Impact Factor (2018): 5.426

Evaluation of Quality of Service in FatTree Topology Based on SDN

Sarah H. Mohammed[1], Dr. Ammar D. Jasim[2]

[1, 2] Department of ICE, College of Information Engineering, Al-Nahrain University, Iraq, Baghdad, Iraq

Abstract: ***The fast progress in the network technologies required the network environment to be fast and reconfigurable flexibly which make the network more dynamic, scalable, programmable and manageable. Software Defined Network (SDN) is suggested to find solution for all problems of previous network forms, by separating the control plane from data plane and provided centralized controlling to manage the whole network. This paper makes study about the implementation of FatTree network topology. Then shows the main idea of OpenFlow protocol that can be applied on FatTree network topology by using SDN technology and how that can be effect on the performance of overall network and make it more flexible to enable the SDN module application, like Quality of Service (QoS) for improving the SDN network. In this paper the physical switches (network equipment) are replaced by software switches in a virtual network environment and display the SDN structure in GUI, also the Floodlight controller is chosen to perform this works and to use as the Network Operating System (NOS) for SDN network.***

Keywords: Software Defined Networks, OpenFlow, FatTree, Floodlight, Quality of Service, Mininet

1. Introduction

Network infrastructure has become vulnerable to attack in the enterprises network and internet. Recently, with quickly development in both cloud services and mobile devices the network architecture has become more complex, because with limited bandwidth the network can't satisfy the user requirements.

Networking technologies set limitation on the network architecture, like complication, conflicting policies, disability to scale and vendor dependence, so that the network can't to provide the needed services in homes, schools and enterprises. Also the limited bandwidth, increase needed to add new cloud services and modifying the traffic pattern all that made necessary to find new network architecture. Software Defined Networking (SDN) is founded as a solution; it's created by Open Networking Foundation (ONF) [1].

In SDN based networks, the essential idea is to transfer the control plane to an application. The network equipment are become straightforward to configure and programed by the application layer in order to forwarding the packets according to the instructions of controller [2]

The Floodlight controller was chosen because it is very widespread today for researches and academic aims [3]. The module system that was presented by the Floodlight makes it easy to extend and enhanced. It backing both OpenFlow switches the virtual and physical, also it's can deal with mixed of OpenFlow and non-OpenFlow switches [4].

2. Openflow Protocol

OpenFlow technology is standard protocol, originated at Stanford University by the Clean Slate program in United States, the SDN core notion is OpenFlow technology [5].
The important components of the OpenFlow-enabled-switch are (a) A *Flow Table*, for each flow entry there is specific action that orders the switch how to deal and process with this flow entry. (b) A *Secure Channel*, which responsible to provide a secure connection between the switches and remote control process (controller like POX, NOX, Floodlight… etc.) to allow the commands and transmitted packets to be sent between the forwarding devices and controller by using (c) The *OpenFlow Protocol*, which allows a server to tell forwarding devices where to send packets [6].

The OpenFlow protocol is considering a key enabler for SDN. It's considering the first standardized southbound interface [7]. The basic role of OpenFlow protocol is to defining the communication protocol that manage the interaction between the SDN controller and the network forwarding devices like switches and routers, so it becomes easier to changing the configuration according the business requirements. Technically, this is based on the main idea of SDN that described as the separation of the control from the data forwarding devices [8].

3. SDN Controllers

SDN controllers represent as a brain of this type of network. The controller is a centralized device that is used to run the SDN application and to implement different functionalities that are used to provide different networking services by interacting with switches by using the OpenFlow protocol (Southband interface) [9].

3.1 Floodlight controller

Floodlight controller is popular controller in the SDN project, its open source and implemented using Java Programming language [2]. Floodlight controller supported by both SDN community and Big Switch Networks. Floodlight is considered as an Apache licensed OpenFlow controller. This controller was support both the hypervisor-based virtual switches like OpenvSwitch, and the physical OpenFlow switches which that provide connection between OpenFlow and non-OpenFlow networks [10]. The figure 1 shows the floodlight controller architecture.

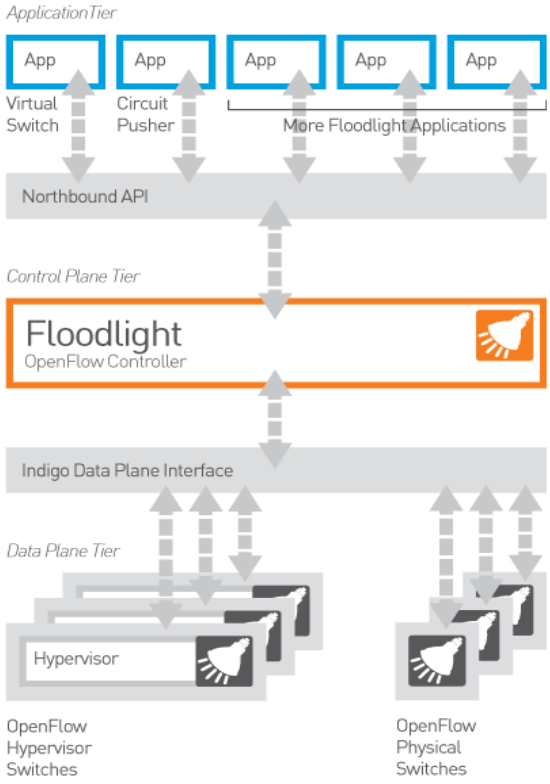

Figure 1: Floodlight Architecture [10]

4. Fattree Topology

A Data Center Network (DCN) has applications in a wide variety of networked environments, such as enterprise networks, infrastructure-based Wireless Access Networks, Optical networks, and datacenter networks [2].

DCN is making the hosts and servers easy to use, which are connected by using switches and dedicated links. Actually DCN can include a large number of hosts with particular delay and bandwidth requirements. Today this is a rising interest in academia and industry of to integrate a large scale DCN with SDN technology. We realize at recently period that most works used fixed network topologies like Tree and FatTree. It is relatively simple to apply routing algorithms on these topologies [11].

5. Proposed Design and Implementation

The topology has insignificant impact in the network performance, in which the network can respond to any needed change through the operation. The most popular topology consists from two or three layers. The FatTree topology was chosen to perform this work, which includes three layers, the core layer, aggregation layer and edge layer [12].

FatTree topology can be implemented by using multiple cheap links to load balancer of network and act as back-up link when other link was failed [13]. FatTree is considering one of the most favorable topologies of DCNs for future according to recently proposed among DCN topologies [12]. It is not considering as a static-network topology, but it can be expand according to the value of k, which in the edge tier the k port switches is connected to $k/2$ servers, and the others ports are connected to the $k/2$ switches that allocated at the aggregation level [13].

The FatTree topology that has been shown in figure 2 was implemented with k = 4. The ping commend using to check the network connectivity and to measuring Round Trip Time (RTT) between all hosts in the network as shown in figure 3.

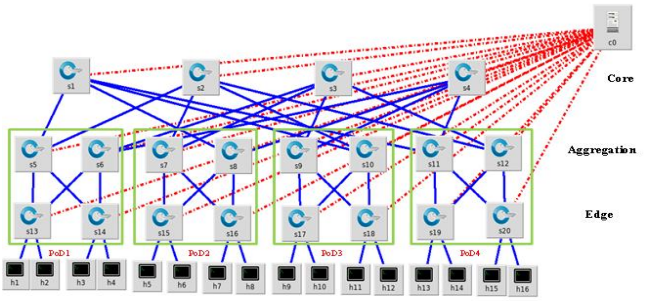

Figure 2: FatTree topology with k=4

```
mininet> pingall
*** Ping: testing ping reachability
h1 -> h2 h3 h4 h5 h6 h7 h8 h9 h10 h11 h12 h13 h14 h15 h16
h2 -> h1 h3 h4 h5 h6 h7 h8 h9 h10 h11 h12 h13 h14 h15 h16
h3 -> h1 h2 h4 h5 h6 h7 h8 h9 h10 h11 h12 h13 h14 h15 h16
h4 -> h1 h2 h3 h5 h6 h7 h8 h9 h10 h11 h12 h13 h14 h15 h16
h5 -> h1 h2 h3 h4 h6 h7 h8 h9 h10 h11 h12 h13 h14 h15 h16
h6 -> h1 h2 h3 h4 h5 h7 h8 h9 h10 h11 h12 h13 h14 h15 h16
h7 -> h1 h2 h3 h4 h5 h6 h8 h9 h10 h11 h12 h13 h14 h15 h16
h8 -> h1 h2 h3 h4 h5 h6 h7 h9 h10 h11 h12 h13 h14 h15 h16
h9 -> h1 h2 h3 h4 h5 h6 h7 h8 h10 h11 h12 h13 h14 h15 h16
h10 -> h1 h2 h3 h4 h5 h6 h7 h8 h9 h11 h12 h13 h14 h15 h16
h11 -> h1 h2 h3 h4 h5 h6 h7 h8 h9 h10 h12 h13 h14 h15 h16
h12 -> h1 h2 h3 h4 h5 h6 h7 h8 h9 h10 h11 h13 h14 h15 h16
h13 -> h1 h2 h3 h4 h5 h6 h7 h8 h9 h10 h11 h12 h14 h15 h16
h14 -> h1 h2 h3 h4 h5 h6 h7 h8 h9 h10 h11 h12 h13 h15 h16
h15 -> h1 h2 h3 h4 h5 h6 h7 h8 h9 h10 h11 h12 h13 h14 h16
h16 -> h1 h2 h3 h4 h5 h6 h7 h8 h9 h10 h11 h12 h13 h14 h15
*** Results: 0% dropped (240/240 received)
mininet>
```

Figure 3: Check the network connectivity.

Various numbers of packets 100, 500 and 1000 packets had been used to check the performance of the SDN FatTree network. The ping command was selected to executing on the hosts (h4, h8, h12, h16) with the IP address of destination (10.0.0.1), which is the IP address of the (h1). In the terminal of these hosts the following command is executed and the results show in chart 1:

#ping –c 100 10.0.0.1
#ping –c 500 10.0.0.1
#ping –c 1000 10.0.0.1

The figure 4 shown that delay of SDN based on FatTree network topology will be increased simultaneously with number of transmitted packets over the same network, so the network performance improved by using SDN technology.

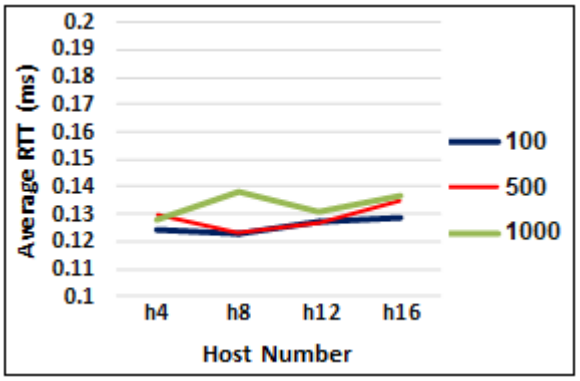

Figure 4: The average RTT.

Figure 5 display the FatTree network topology by GUI of Floodlight controller.

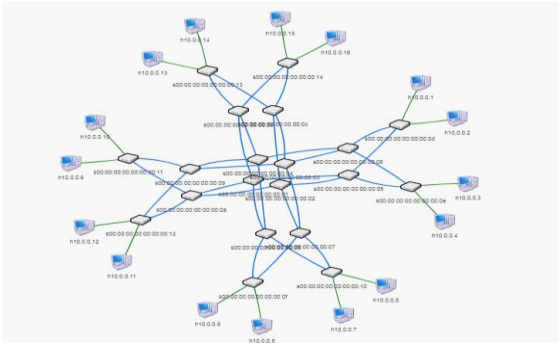

Figure 5: GUI display FatTree network topology

6. QOS Implementation

The QoS application will be implementing through FatTree network topology based Floodlight controller. This application is implemented through OpenFlow protocol in order to permits the administrator to select the path in which the data will be flow through a network, for example the administrator can to select the bandwidth for specific link in network as the optimal communication path for a specific data flow. In this part of work the FatTree network topology was connected with the Floodlight controller through its IP address and port number then enable the QoS by using some specific commands in order to adding the new QoS rules in to Flowtable.

The QoS is enabled on the controller by using the following command:

$sudo ./qosmanager2.py –e

The rules will be added by this command:

$./qospath2.py -a -S 10.0.0.3 -D 10.0.0.7 -N 2Mbit_3-7 -J '{"ethtype":"0x0800"," queue":"2"}'

Through this command the QoS is enabled on the links that connected the h3 and h7, and set 2 Mbit the bandwidth between this hosts as shown in figure 6, also figure 6 can display the process of configure the QoS over SDN network topology.

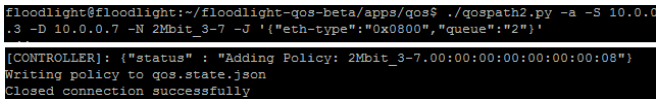

Figure 6: Configure QoS based on Floodlight controller.

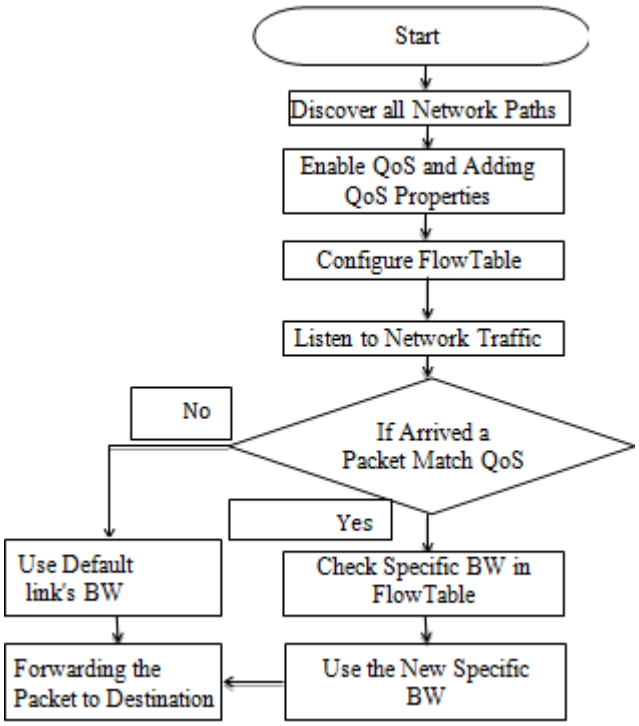

Figure 7: Flowchart of QoS implementation based SDN topology

In this scenario the QoS is applied between two hosts (h3 and h7) and the following figures will display the affecting of enabling the QoS application on determination the bandwidth over the link between these hosts. The two following figures (7 and 8) can show the difference (before/after) of configuring the QoS application over the network and it's affecting on link's bandwidth that connected these hosts.

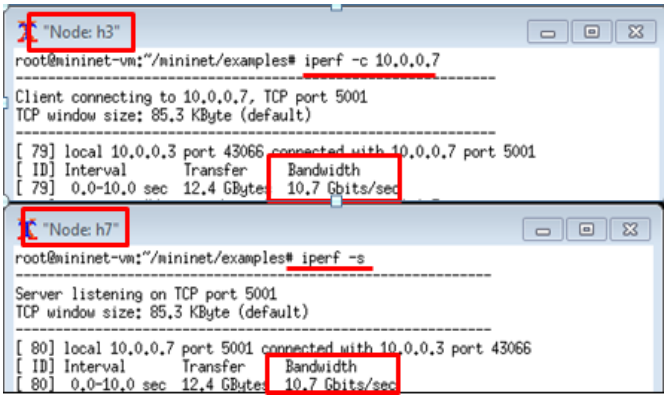

Figure 8: Before enable QoS application

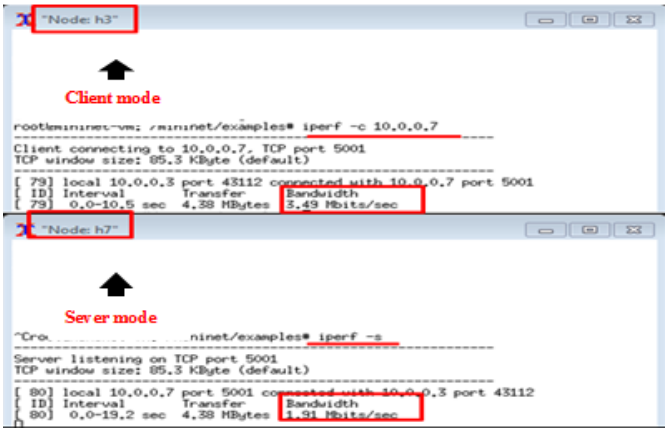

Figure 9: After enabling QoS application.

7. Conclusions

The Mininet is virtualization tool that used to implement and configure the expensive network; it's containing the basic interfaces that used to create needed network topology.

The basic idea of paper is to display that Software Defined Networks (SDN) technology makes the network configuration simpler to manage and control through OpenFlow protocol and to improve network performance by enable central controller in order to manage the network. This central controller can enable the module applications in the network without needs to configure each switch in the network individually. The FatTree network topology is chosen to implement this work, because it has complex structure, but it is also possible to implementing this work in another network structure. This work is concerned specially with the SDN. SDN technology is the most familiar where the control plane is separated from data plane.

choosing the controller in SDN have important role on the network performance, and the Floodlight controller was chosen to implementing this work, because this controller permits the developers to easily adjust software and develop applications. Also it is include a representational state transfer application program interfaces (REST APIs) which make a programming interface with the network easier.

Finally the QoS application is implemented and configured through using the Floodlight controller over FatTree network topology, in which the QoS is responsible to maintain the performance of critical application and ensure enough bandwidth for their proper operation.

The QoS application is implemented through using Floodlight controller. This application is so important to locate the needed properties over specific link. In this work the bandwidth limitation (2Mbit) controlled by using QoS in particular link.

Lastly, the Floodlight controller is considering as so flexible controller, because it is easy to apply and mange in the real network environment, moreover it can configure to achieve needed features.

References

[1] Open Networking Foundation, *"Software-Defined Networking: The New Norm for Networks"*, ONF White Paper, April 2012.
[2] H. Akcay and D. Yiltas-Kaplan, *"Web-Based User Interface for the Floodlight SDN Controller"*, Int. J. Advanced Networking and Applications, vo. 08, No. 05, pp. 3175-3180, 2017.
[3] L. Chen, M. Qiu and J. Xiong, *"An SDN-Based Fabric For Flexible Data-Center Networks "*, Computer Society, IEEE, pp. 121-126, 2015.
[4] S. Morzhov, I. Alekseev and Mikhail Nikitinskiy, *"Firewall application for Floodlight SDN controller"*, SIBCON, IEEE, 2016.
[5] Rong Peng, and Lei Ding," *Research on Virtual Network Load Balancing based on OpenFlow "*, AIP, pp. 020014-1–020014, 2017.
[6] D. Kreutz, F. Ramos, P. Verissimo, C. Esteve, S. Azodolmolky and S. Uhlig, *"Software-Defined Networking: A Comprehensive Survey"*, IEEE, Vol. 103, No. 1, pp. 14-76, 2015.
[7] O. Salman, I. Elhajj, A. Kayssi and A. Chehab, *"SDN controllers: A comparative study"*, IEEE, 18th Mediterranean Electrotechnical Conference (MELECON), Lemesos, Cyprus, DOI. 10.1109/MELCON.2016.7495430, pp. 1-6, April 2016.
[8] Alshnta, M. Abdollah and A. Al-Haiqi, *"SDN in the home: A survey of home network solutions using Software Defined Networking"*, Journal Cogent Engineering, Vol. 5, Issue 1, pp. 1-40, May 2018.
[9] L. Alexandru. S. Stancu, S. Halunga, A. Vulpe1, G. Suciu, O. Fratu and E. Popovici, *"A Comparison between Several Software Defined Networking Controllers"*, IEEE, 12th International Conference on Telecommunication in Modern Satellite, Cable and Broadcasting Services (TELSIKS), Nis, Serbia, DOI. 10.1109/TELSKS.2015.7357774, pp. 223-226, October 2015.
[10] U. Mihaela, *"Flexible and Programmable Evolved Packet Core: A New SDN-based Model"*, M.S. Thesis, Delft University of Technology, Netherlands, July 2014.
[11] T. Wang, H. Chen, G. Cheng, and Y. Lu, *"SDNManager: A Safeguard Architecture for SDN DoS Attacks Based on Bandwidth Prediction"*, Hindawi, Article ID 7545079, pp. 16, 2018.
[12] H. Qi, K. Li, *"Software Defined Networking Applications in Distributed Datacenters"*, Springer, AG Switzerland, 2016.
[13] M. Al-Fares, A. Loukissas and A. Vahdat, *"A Scalable, Commodity Data Center Network Architecture"*, conference of the Special Interest Group on Data Communications (SIGCOMM), SEATTLE, USA, Vol. 38 Issue 4, pp. 63-74, January 2008.