International Journal of Computing and Network Technology
ISSN (2210-1519)
Int. J. Com. Net. Tech. 6, No. 2 (May-2018)
http://dx-doi.org/10.12785/ijcnt/060201

Performance Comparison and Evaluation of Different Software Defined Networks Controllers

Mahmood Z. Abdullah, Nasir A. Al-awad and Fatima W. Hussein

Computer Engineering Department, Al-Mustansiriyah University, Baghdad, Iraq

Received: 7 Jan. 2018, Revised: 12 March 2018, Accepted: 19 March. 2018, Published: (1 May 2018)

Abstract: In Software Defined Networking (SDN) which is a new network architecture, the controller represents the main and intelligent part of its components. Today, there exists many SDN controllers both open source and commercial including Cisco APIC, VMware NSX Controller, NEC PF6800, Beacon, Floodlight, Iris, Maestro, RunOS and others. The question is which controller can perform better in which situations. Several works were done to compare these controllers with respect to efficiency, controllers' features, and architecture. In this paper, a Performance evaluation test of five open source controllers (libfluid, ONOS, OpenDaylight, POX and Ryu) is done, this test uses a linear topology that is built in Mininet emulator, with different number of switches. The Performance evaluation is done using Iperf and Ping commands. This paper introduces a new contribution in evaluating and comparing the End to End delay and End to End throughput responses of the five controllers while increasing the load on the linear topology and at what point of network load (number of switches) the controllers stop responding. Even though the results show that libfluid gives the best throughput performance and POX gives the best delay performance; however, the selection of the best performing controller should be based on several criteria, per the user requirements.

Keywords: Software Defined Networking (SDN), libfluid, ONOS, OpenDaylight, POX, Ryu, Mininet.

1. INTRODUCTION

Software Defined Network (SDN) refers to a way of organizing computer network functionality. It allows the network to be virtualized and programmable [1]. The approach proposed by the SDN paradigm is to move network's intelligence out from the forwarding element (switches and routers) and to put it into the logically centralized controller [2]. The architecture of SDN as shown in Fig. 1, consist of Three layers (Application, Control, and Forwarding or infrastructure), these layers communicate with each other through northbound and southbound Application Programming Interface (API). northbound API is between Applications and control layer while southbound API is between Control and Forwarding layer [3]. OpenFlow Protocol which was standardized by the Open Networking Foundation (ONF), is the first and most well-known southbound interface. The controller is the fundamental part of SDN, it provides a programmatic interface for user-written network management applications. When a packet arrives at a switch, it looks for a match in the flow table and specifies what functions are to be performed on the packets. If there is no match in the flow table the packet is send to the controller to set up rules or make decisions. Then, these decisions move down to the overseen switches which simply execute them based on the rules coming from the controller. This gives us a lot of benefits like global controlling and viewing whole network at a time [2]. Different number of OpenFlow controllers exist at present, the usability of such controllers differs from conceptual prototype to production quality. Due to the importance of the controller, there is a need to assess and compare the different existing controllers [4]. For researchers and network administrators, choosing the best controller can be problematic [5], number of works have carried out a partial performance evaluation in the past few years [4]. In this paper, we will try to compare (libfluid, ONOS, OpenDaylight, POX and Ryu) controllers and test their performance.

E-mail address: drmzaali@uomustansiriyah.edu.iq, nasir.awad@uomustansiriyah.edu.iq, fatima.wadaa@uomustansiriyah.edu.iq

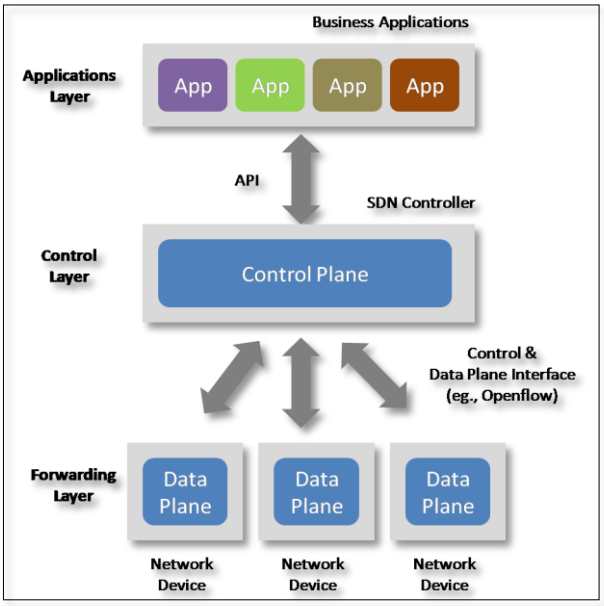


Figure 1. Software-Defined Network Architecture

The rest of this paper is organized as follows: Section 2 discusses the previous work related to SDN controller's comparison, Section 3 shortly review (libfluid, ONOS, OpenDaylight, POX and Ryu) SDN controllers, and Mininet, Section 4 shows the methodology and analyzes of results of the performance test. At last, in Section 5 conclusion is presented.

2. Related Works

In recent years, several studies have been done in the aim of comparing SDN controllers, a review of some of which is presented in this section.

In [4], five centralized controllers have been selected for performance benchmark evaluation using Cbench tool, the controllers are Ryu, POX, NOX, Floodlight and Beacon. Making a reality check on the current performance achieved by mainstream open source controllers. the performance (throughput and latency) of the controllers were measured in cases of single-thread and multi-thread.

In [5], an evaluation based on some network QoS parameters was done. Two of the most popular controllers, Floodlight and OpenDaylight were compared in terms of delay and loss, in different topologies and network loads.

In [6], A. L. Stancu, et al. measured the performances of the four SDN controllers, POX, Ryu, ONOS and OpenDaylight, using Mininet emulator. the controllers were instructed to act as a simple hub and as a simple L2 learning switch. A tree topology was used for comparison, in every phase, two tests were conducted: Ping command between two end hosts, and iperf command, also between the two hosts.

O. Salman, et al., In [7] conducted a comparison of several Controllers namely (Libfluid, NOX, POX, Maestro, Beacon, MuL, Iris, OpenDaylight, Floodlight, Ryu, Runos) based on multiple criteria. In addition, a performance test using Cbench was done, the tests were performed in the two (throughput and latency) modes. In the latency mode, varying the number of switches, and in the throughput mode, varying the number of threads binding to the controller instance.

In [8], the performance of four types of SDN controllers (Floodlight, Beacon, Open-MUL and Open-IRIS) was evaluated. This evaluation has been done by using three types of traffics: ICMP, TCP and UDP using Ping and Iperf commands. Then a method to enhance the performance of the network by using QoS technique with Floodlight controller was proposed.

3. SDN Controllers and Emulator

In this section, short review of libfluid, ONOS, OpenDaylight, POX, Ryu, and Mininet is presented. Also, a summary of the main characteristics (such as written language, Graphical User Interface (GUI) and etc.) of the selected controllers is presented in Table I.

A. Libfluid

libfluid is a library bundle that provides the basic features to implement an OpenFlow controller. libfluid was chosen to be the winner of the OpenFlow Driver Competition which was sponsored by Open Networking Foundation (ONF). the sample controller of libfluid controller, that listens to OpenFlow TCP port 6653 is used in this paper to perform the required test [9].

B. ONOS

ONOS (Open Network Operating System) project is an open source community. Creating an SDN operating system is the purpose of this project, [10] ONOS project is written in java as bundles and it is loaded to Karaf OSGi container

C. OpenDayLight

The OpenDaylight (ODl) Project is a collaborative open source project hosted by The Linux Foundation written in the Java [11]. OpenDaylight supports the programming of a bidirectional REST and OSGi framework and supports different non-OpenFlow southbound protocols [5]. For developers and others, there is a dedicated wiki, and several mailing lists and a source code repository for releases of the controller.

D. POX

POX (Pythonic Network Operating System) is a networking software platform, it is NOX's younger

sibling. POX is developed using python programming language [12]. It can be helpful in writing networking software. POX runs at different operating systems like Windows, Mac OS, and Linux. It can work with Python 2.7 and version below. POX specially support Open vSwitch/Nicira extensions and communicates with OpenFlow 1.0 switches [13].

E. Ryu

Ryu Controller is an open SDN Controller designed to increase the agility of the network. it is a component-based software defined networking framework that is fully written in Python. The word (Ryu) means "flow" in Japanese. The Ryu Controller is supported by Nippon Telegraph and Telephone Corporation (NTT). various protocols like NETCONF, OF-config, OpenFlow, and others are supported by Ryu controller [14].

TABLE I. FEATURE BASED COMPARISION OF THE CONTROLLERS

Controller Name	Written language	GUI	OpenFlow version Support	Developed by
Libfluid	C++	No	OF 1.0, 1.3	Open Networking Foundation
ONOS	Java	Yes	OF 1.0, 1.3	ON.LAB, At&T, Ciena, Cisco, Ericsson, Fujitsu, Huawei, Intel, Nec, Nsf.Ntt Comunnication, Sk Telecom
ODl	Java	Yes	OF 1.0, 1.3, 1.4	Linux Foundation With Memberships Covering Over 40 Companies, Such as Cisco, IBM, NEC
POX	Python	Yes	OF 1.0	Nicira
Ryu	Python	Yes	OF 1.0, 1.2, 1.3, 1.4,	Nippon Telegraph And Telephone Corporation

F. Mininet

Mininet is a network emulator, used to create a network of virtual switches, hosts, links, and controllers on one Linux kernel. Mininet enables the testing of topology, without wiring up the network physically and has a (CLI) for debugging network tests. Different topologies can be created, and virtual networks can be tested by sending packets to each other. There are three predefined topologies which are: single, linear (used in this paper), and tree. Remote controllers also can be used, virtual network can be connecting to any remote controller in VM, local machine, or anywhere else [5].

4. PERFORMANCE EVALUATION

To implement the performance test, linear topology will be used to measure the load on the network. The reason to choose linear topology in this paper is that it is already predefined and does not need to be created manually as a custom topology and for the other predefined topologies (single and tree), The SDN controller in single topology only connects to a single switch so it handles the flow table of only one switch and does not get affected with increasing the number of hosts so as a result the throughput and delay values will be nearly the same. Also, for tree topology it requires to specify a depth and a fan-out of the tree topology therefore, it will not give the desired direct number of switches to connect with the controller and the number of switches will not match with the number of hosts so the ping and iperf test between hosts will be affected. Fig. 2 shows the setup of the designed network which is used to evaluate the (libfluid, ONOS, OpenDaylight, POX and Ryu) Remote controller's performance. the linear topology consists of different number of switches which is connected to the underlying hosts. This scenario includes testing a different number of switches: 8, 16, 32, 64, …, etc. to see how the performance of the controller will be affected when increasing the workload on the network.

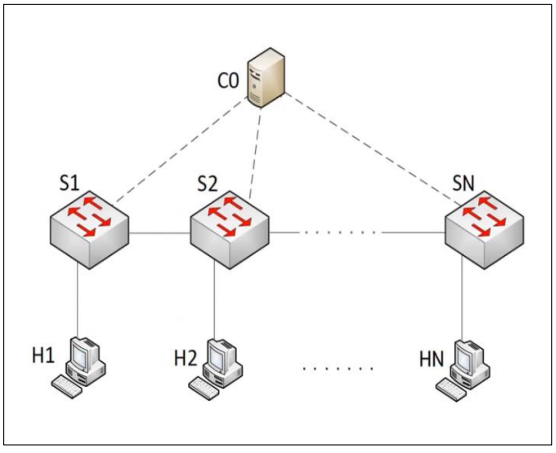


Figure 2. Virtual Linear Topology with Different Number of Switches

A. Methodology

1) End To End throughput

The first performance parameter is the Throughput, Throughput defines how much useful data can be transmitted per unit time. the throughput value in most practical cases is less than the bandwidth and if there is no protocol it is equal to the bandwidth [15]. Iperf was used to measure the throughput of the network by generating client-server TCP connection [16]. The way of measuring here is between the two end hosts of the linear topology. One of the end hosts (here the host h1) is running the iperf command in server mode and the other End host of the network is running the iperf command in client mode. The throughput test is repeated for each different number of the switches that is connected to each one of the controllers.

2) End To End delay

Delay in data networks is generally the round-trip delay (also called Round Trip Time - RTT) for a packet within the network [15]. For Delay measurement, Ping can be used for troubleshooting to test connectivity and evaluate the average RTT in ms. Ping is the reaction time of the connection in the network that shows how fast a host gets a response after sending a request to another host or server. When the ping response is fast, it means a more responsive connection is achieved. Network delays can range from a few milliseconds to several hundred milliseconds [16]. The two end hosts of the linear topology are chosen to perform the delay measurement by sending Internet Control Message Protocol (ICMP) request packets to the end host and receiving ICMP reply packets from that host. The time at which ping sends an ICMP packet and the time at which it receives the reply packet are recorded [12].

B. Analysis of Results

The results obtained through the measurements are shown, the tests were conducted for the listed controllers (libfluid, ONOS, OpenDaylight, POX and Ryu).

1) Results of throughput measurement

Fig. 3-7 shows the ETE-Throughput values in Megabits per second for each one of the controllers respectively. From the throughput measurements of these controllers, these values gradually decreased as the workload of the network increased by increasing the number of switches and hosts. This is because when increasing the number of switches, more processes are needed, and all those processes consume from the bandwidth. this decrease continues until the controller stop responding, libfluid and POX stopped responding at the test of 1024 switches connected, while the other controllers stopped at 512 switches connected. From the comparison of the throughput of these controllers in Fig. 8, libfluid and POX show the highest throughput value among the other controllers at all different switch number tests and when network overload POX gives better results than libfluid. While OpenDaylight controller shows the lowest throughput value.

2) Results of delay measurement

From the ETE-Delay measurements, Fig. 9-13, shows the average RTT values in milliseconds for each one of the controllers respectively. From these results, it can be seen that the average RTT of the controllers starts increasing as the number of switches increase due to the overload on the controller and the large number of the switches that is connected. ONOS, POX, Ryu and OpenDaylight controllers approximately have the same delay values and their maximum results at network overload is less than 4 ms. While for libfluid controller the results start increase significantly until it reaches 715.73ms at 512 switches connected. From the comparison of the delay of these controllers in Fig. 14, ONOS controller has the lowest delay values among the other controllers. While libfluid controller has the highest delay values. Finally, Fig. 15 shows the delay results without libfluid controller to clarify the results of the other controllers.

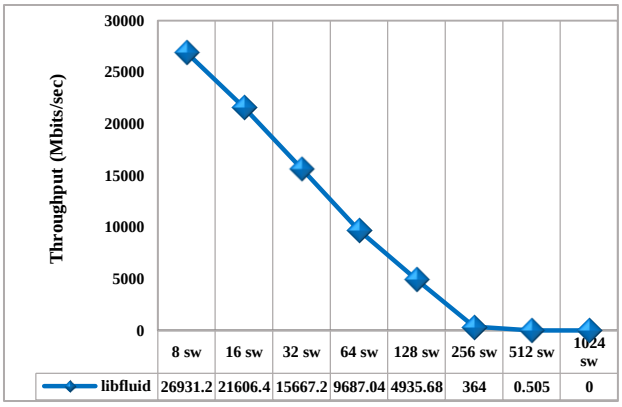


Figure 3. ETE throughput of libfluid

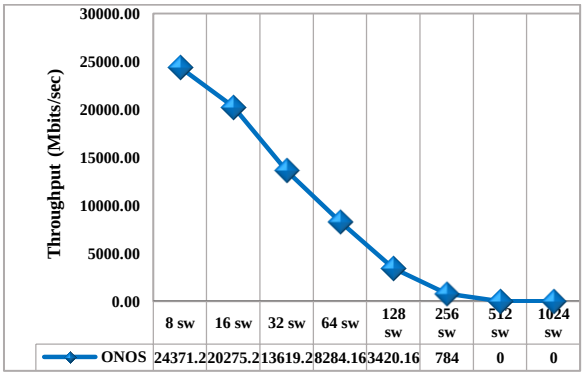


Figure 4. ETE throughput of ONOS

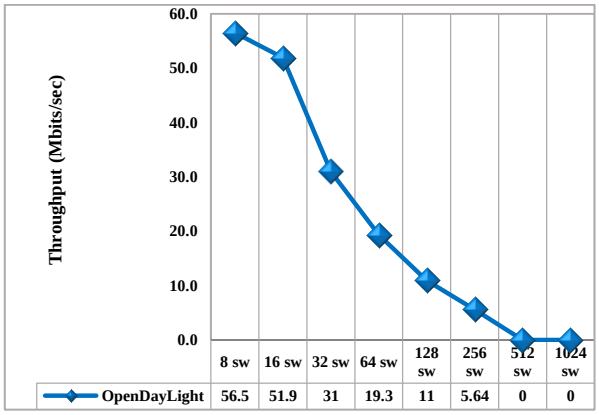


Figure 5. ETE throughput of OpenDaylight

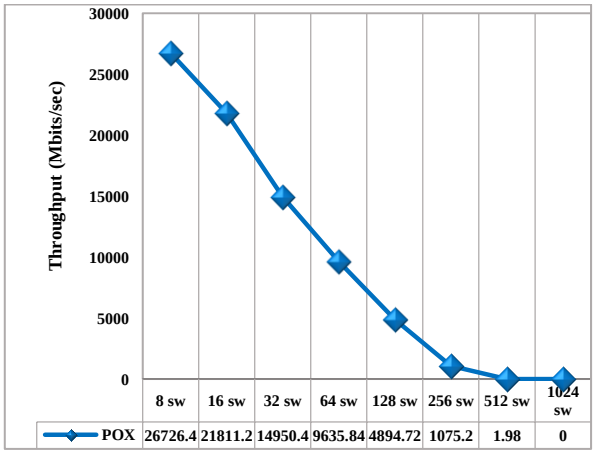


Figure 6. ETE throughput of POX

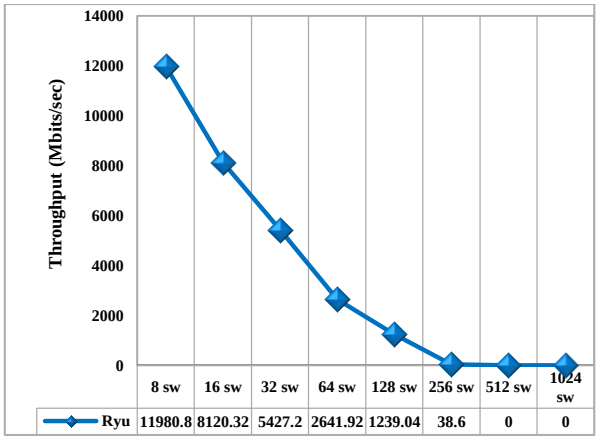


Figure 7. ETE throughput of Ryu

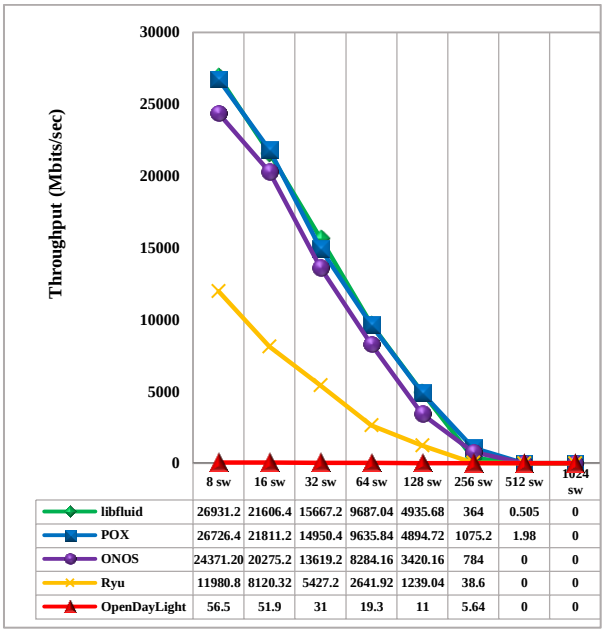


Figure 8. Total ETE throughput of the controllers

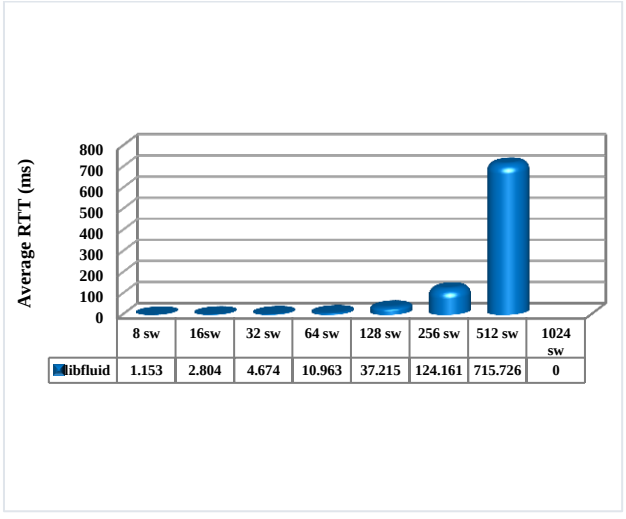


Figure 9. ETE delay of libfluid

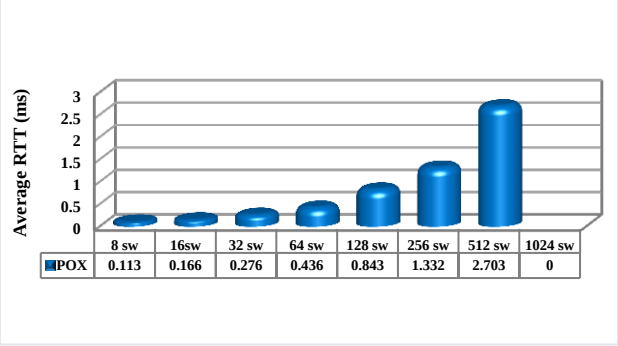


Figure 10. ETE delay of POX

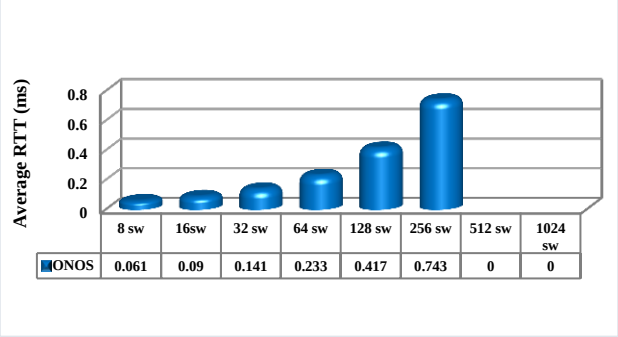


Figure 11. ETE delay of ONOS

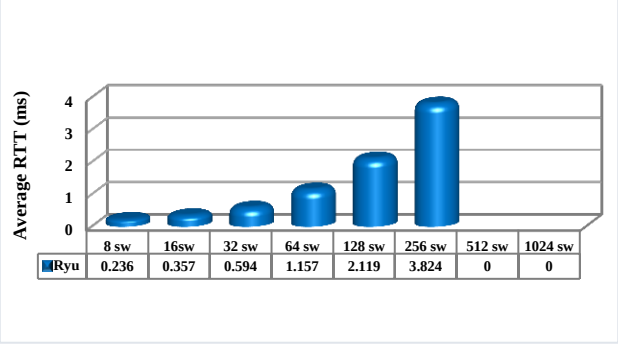


Figure 12. ETE delay of Ryu

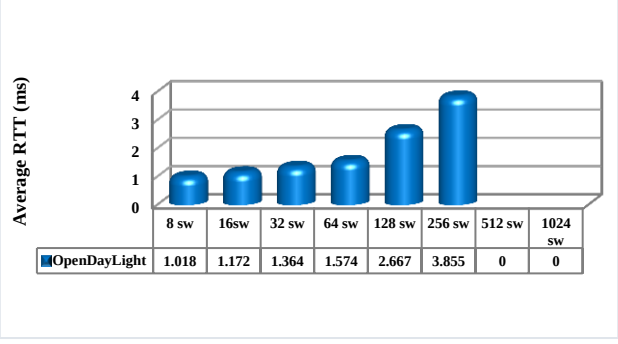


Figure 13. ETE delay of OpenDaylight

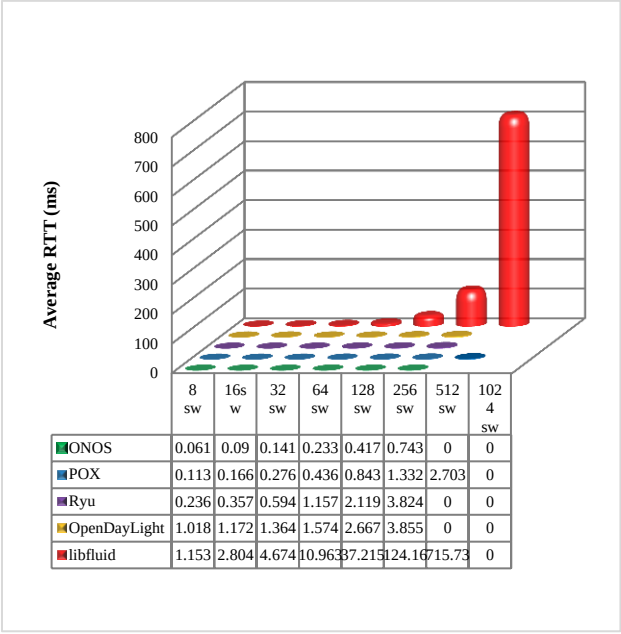


Figure 14. Total ETE delay of the controllers

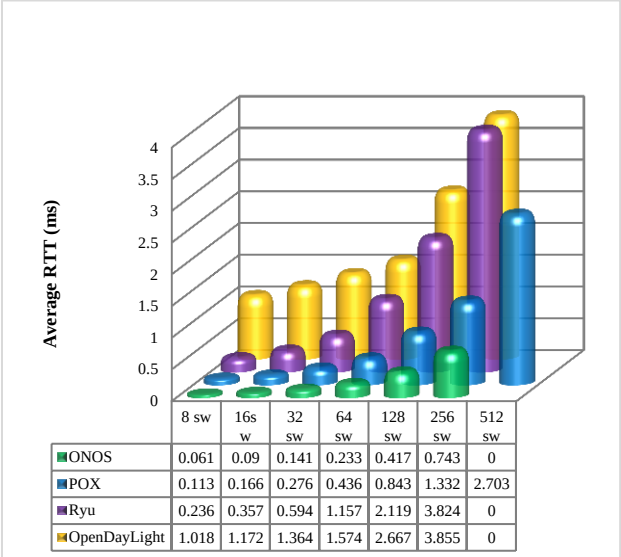


Figure 15. Delay results without libfluid controller

5. CONCLUSION

In this paper, the performance of five controllers was compered based on ETE-Throughput and ETE-Delay for linear topology with different number of switches in Mininet emulator. From the throughput measurements of the controllers, the throughput values decreased as the number of switches and hosts are increased in the network (increasing the workload on the controller). The throughput values of libfluid and POX controllers overcome the values of the other controllers. For delay measurements, the delay values increased as the number of switches and hosts are increased in the network. libfluid controller had the highest delay compared to other controllers and ONOS controller had the lowest delay values.

REFERENCES

[1] [1] P. A. Morreale and J. M. Anderson, "Software Defined Networking: Design and Deployment," CRC Press, Taylor & Francis Group, LLC, New York, USA, 2015.

[2] A. Shalimov, D. Zuikov, D. Zimarina, V. Pashkov and R. Smeliansky, "Advanced study of SDN/OpenFlow controllers," Proceedings of the 9th Central & Eastern European Software Engineering Conference in Russia, ACM, pp. 1-6, 2013.

[3] Open Networking Foundation, "Software-Defined Networking: The New Norm for Networks," ONF White Paper, 2012.

[4] Y. Zhao, L. Iannone and M. Riguidel, "On the Performance of SDN Controllers: A Reality Check," IEEE Conference on Network Function Virtualization and Software Defined Network (NFV-SDN), 2015.

[5] S. Rowshanrad, V.Abdi and M. Keshtgari, "Performance Evaluation of SDN Controllers: Floodlight and OpenDaylight," IIUM Engineering Journal, Vol. 17, No. 2, pp. 47-57, 2016.

[6] A. Stancu, S. Halunga, A. Vulpe, 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), 2015.

[7] O. Salman, I. Elhajj, A. Kayssi and A. Chehab, "SDN Controllers: A Comparative Study," IEEE 18th Mediterranean Electrotechnical Conference (MELECON), 2016.

[8] A. D. Jasim and D. A. Hamid, "Enhancing the Performance of OpenFlow Network by Using QoS," International Journal of Scientific & Engineering Research (IJSER), Vol. 7, Issue 5, 2016.

[9] A. Vidal, "libfluid: a lightweight OpenFlow framework," M.Sc thesis, Federal University of São Carlos, Sorocaba SP, Brazil, 2015.

[10] P. Berde, M. Gerola, J. Hart, Y. Higuchi, M. Kobayashi, T. Koide, B. Lantz, B. O'Connor, P. Radoslavov, W. Snow and G. Parulkar, "ONOS: Towards an Open, Distributed SDN OS," HotSDN '14 Proceedings of the third workshop on Hot topics in software defined networking, Chicago, Illinois, USA, Pages 1-6, 2014.

[11] S. N. A. Braojos "The OpenDaylight Open Source Project," M.Sc thesis, Rey Juan Carlos University, Espania, 2014.

[12] A. D. Jalil, "Performance Evaluation of Software Defined Networks," M.Sc thesis, Al-Nahrain University College of Information Engineering, Iraq, 2017.

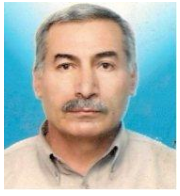
[13] A. Carranza, J. Tax, J. M. R. Álamo. "Building a Future in SDN with one Controller," Enterprise Computing Community Conference, 2014.

[14] ryu development team, "ryu Documentation Release 4.28," 2018.

[15] "Bandwidth, Throughput and Delay", [Online], Available at: http://networking.layer-x.com/p040300-1.html [Accessed 2018].

[16] D. A. Hamid, " Performance Evaluation and Enhancement of OpenFlow Controller," M.Sc thesis, Al-Nahrain University College of Information Engineering, Iraq, 2016.

Mahmood Zaki Abdullah is an associate professor Dr. in the Computer Engineering Department at the College of Engineering of Al-Mustansiriyah University. He got the Ph.D. and M.Sc. degrees from the University of Technology at 2007, and 2000 and a B.Sc. degree from the University of Baghdad at 1991. His research interests include Information Technology, Software Engineering, and Computer Networks. He has served as a Technical Program Committee member for many international conferences; he published many books and papers in these fields.



Nasir Ahmed Al-awad was born in Iraq, 1957. He received B.Sc. degree in control and system engineering from Technological University, Iraq, in 1981. M.Sc. degree in control and instrumentation engineering from Technological University, Iraq, in 1984. He is currently Assist Prof. and the head of Computer Engineering Department, Al-Mustansiriyah University, Iraq. His research interests include control theory, computer control and computer aided design of control system.

Fatima W. Hussein, M.Sc. Student at Computer Engineering Department, AL Mustansirya University, 2017. B. Sc. Degree from Computer Engineering Department, AL Mustansirya University, 2016.