International Journal of Control Theory and Applications

ISSN : 0974-5572

Volume 10 • Number 32 • 2017

"Technical Comparative Analysis between ODL and ONOS Software Defined Controllers"

Vaishnavi Moorthy[a] Vamsi Krishna Patchava[b] and Mansi Dikshit[b]

[a]*Assistant Professor, Dept of Computer Science & Engineering, SRM University, Kattankulathur (India)*

[b]*B.Tech, Final Year, Dept of Computer Science & Engineering, SRM University, Kattankulathur (India)*

Abstract: The ease of Software Defined Networking has opened several pathways to dynamically modify network routing information. Various software defined controllers have been deployed in working environment to bring the best out of SDN capabilities. The proposed paper depicts the details and the comparison between two OpenFlow controllers – open network operating system (ONOS) and OpenDayLight (ODL) and also drives home the fact as to which one provides more optimal solutions. The analysis, as a whole, is done on the basis of throughput and latency constraints which are some of the key factors in determining the stability of a network. The paper encourages the key facts as well as employs standard measurement in terms of bandwidth so as to check the maximum number of switches and nodes each of the controllers can sustain in the network and the reason for its behaviour. Sometimes the routing information is not correctly synchronized and this may cause loss of packets to be transmitted. Statistical flows are identified and displayed with the help of graphical structures. The obtained results demonstrate the need to choose ONOS controller over Open Day Light with the help of various assumptions and facts put forth and experimental analysis which proves that ONOS gives on-par strategical advantage in terms of packet delivery and user-friendliness.

Keywords: *Software Defined Networking-ONOS controller-OpenDayLight-Throughput-Latency-OpenFlow.*

1. INTRODUCTION

In traditional networking systems, the data plane and the control plane are bound together, making it difficult for the vendors to operate dynamically. Recently, the concept of software defined networking has been proposed to organize and manage networks. By decoupling data and control plane, the ability to operate in a large-scale network environment using software has been made possible by the implementation of protocols such as OpenFlow and ForCES (Forwarding and Control Element Separation) [1] [2]. To control networks with software, many SDN controllers have been proposed such as NOX, POX, RoseMary, OpenDayLight, Open Network Operating System (ONOS), and so on [3]. Also, the advancement in cloud services made it easy for the software to be reliably stored and easily retrieved. SDN provides flexibility for networks and operates in a vendor-neutral environment. The delightful fact is that most of these SDN controllers are released as open source projects.

Although many of the SDN controllers are extensively in use for experimental purposes today, it is quite reasonable to know which of these controllers give optimal solutions in terms of specific constraints like bandwidth, packet delivery, throughput and latency, which governs the operation and stability of the network [4]. The proposed paper takes into consideration two important SDN controllers – OpenDayLight(ODL) and ONOS controller. ONOS is preferred because it provides a distributed architecture for a global network view to applications, which is logically centralized even though it is physically distributed across multiple servers [5]. Adding on, the root factor in choosing OpenDayLight controller is that it strives to unite all the controllers by offering a specific standard for the network hardware providers and other software defined controllers [6].

The proposed paper aims at studying the throughput and latency parameters in both the SDN controllers and comparing the analysis results for further study.

Also, the impact of how the number of switches affects the throughput and latency curves is depicted. The paper strives to bring some key facts into light - capability, performance of controllers under dynamic environment and determining the advantage of ONOS controller over OpenDayLight(ODL) with real-time implementation.

2. SOFTWARE DEFINED CONTROLLERS

ONOS has an Web GUI which can be run on a browser to visualise the flow statistics. Many versions of ONOS controller has been released since its inception. This paper aims to build the facts related to ONOS controller with respect to the cardinal release 1.2.0. Each of the links, hosts and switches can be individually visualised with the help of network interface topology and the ONOS summary table on the right end corner of the screen [7].

On the other hand, OpenDayLight(ODL) has a similar UI platform named DLUX. ODL provides a flexible and common platform with wide variety of applications and use cases like automated service delivery, cloud, NFV and Network resource optimization [8].

In the proposed paper, linear and tree topologies are taken into consideration and the estimation of number of packets transferred versus the time taken in milliseconds is plotted, by taking into account both latency and throughput parameters.

First, a linear topology with ‘n’ nodes and ‘n’ switches is created with the help of a simulator like mininet. This simulator enables the user to create and interact with an SDN prototype and accounts for smooth functioning on hardware [9]. This is to be run on the background to support ONOS Web GUI as it captures virtual packet flow within the network.

When a topology is run on mininet and the controller is run on another terminal, the virtual network can be seen in the UI with all the switches and nodes and a default controller, termed as a cluster node [10].

Following this scenario, an observation is conducted where the hosts are communicated individually and the same procedure is repeated to communicate between all different combination of hosts, taking two at a time. This will enable in keeping tracking of network traffic. A success in tracing the path will show the host number as output and a failure will result in non occurrence of any even in the activity log [11] [12].

After the ping has been successfully completed, the desired switch can be viewed in the summary table. This will show the switch ID, serial number, protocol implemented, whether it is binded to the master or slave node, ports and flows. An OpenFlow table is generated which also includes the port ID. This information includes the port ID, number of packets sent and received, number of bytes sent and received, the number of packets which are sent and dropped, the number of packets which are received and dropped, the time duration of the network connection establishment and delay, if any.

Upon gathering the above sequence of results, it is likely to plot the graph and trace whether there is any significant change in the behaviour of the network in each of these topologies. This study aims to point the difference between two major controllers – ONOS and OpenDayLight.

Since the whole study is under the base aim of comparing the controllers in terms of latency and throughput, it is important to know what these terms signify.

Latency is a measure of the time taken for a packet to reach from one node to another in a network [13].

Throughput is the amount of data transferred from one location to another within a stipulated period of time [14].

3. COMPARISON FEATURES

As software defined networking (SDN) is an approach for building flexible networks in contrast to user's needs, it has become essential to compare the performance and stability of the software defined controllers and choose the best of them for practical purpose [15]. The proposed paper identifies two of the established controllers - OpenDayLight(ODL) and Open Network Operating System (ONOS). This comparative study focuses on which of these two controllers signify a greater level of importance in accordance to the proposed theoretical study and graphical analysis. The ODL project is led by the Linux Foundation and aims to produce an extensible, open source driven networking platform on top of existing standards such as OpenFlow and ForCES that provide a universal interface for the successful establishment of control in virtual or physical switches via software [16] .

On the other hand, ONOS being released at a later stage than ODL, provides a single interface GUI application that provides a visual interface for the operation of ONOS controller during runtime [17]. Both ONOS and ODL install the apache karaf feature to visualize the flow of switches and hosts and also to establish the links between devices pertaining to the topological needs. The GUIs have multiple panes that are used to display controls, end hosts and summary.

While ONOS is a very active community and has a good documentation in terms of guides when compared to ODL, the mode of progress is not so vast as ODL and ONOS is still in its early phases of development. The main differentiability between these controllers is that ONOS and ODL, both focused on solving different problems. ONOS mainly focuses on service provider's needs, ODL was created to be linux of networking in order to have a very long life and to enable people to have a wide range of solutions [18].

Any SDN controller when introduced is categorized based on various features like programming language, OpenFlow support, northbound interfaces, southbound interfaces, etc. When analyzed according to these features of a controller, both ONOS and ODL approximately lie on the same page [19]. This is further elaborated with the help of Table 1 which describes various controller features and the respective configurations of ONOS and ODL. Hence, a study of the performance features like latency and throughput of the controllers becomes a necessity to determine a better controller between the two.

In the proposed study of controllers, some key facts revitalize which stimulate the evidences in support with the graphical parameters. Since the main objective is to compare the flow statistics, throughput and latency parameters form the perfect analysis [20]. The graphical study brings out some valid facts about the behavior of these controllers when a certain time delay is introduced, keeping the connection uninterrupted. The connections of the host and switches can be observed with the help of the GUIs of both the controllers. Upon observing the statistics, it has been found that ONOS and ODL, both transmit a considerable number of packets when compared to other SDN controllers like Ryu, POX, NOX, etc [21] . It can also be noted that on increasing the number of switches, the end hosts become more cluttered in ODL when compared to ONOS as displayed in the GUI because of the interlacing of the end hosts with the switches.

The behavior of ONOS provides variations during the inception phase of packet transmission. It is observed that for simple topologies like linear, the average time taken by ONOS controller is low when compared to ODL controller but the time becomes almost equal with the rise in the complexity of topologies. The time taken for transmission of the first packet in ONOS is equal to time taken in transmitting a considerable number of packets in ODL. This is certainly a great loss to start [22].

After the first few packets are successfully transferred in ONOS, the average time taken by the remaining packets is quite low. On the other hand, ODL starts transferring the packets at a considerably high average time when compared to ONOS, but the rate is quite stable. The enormous delay seen in the case of ONOS might be due to disturbances in the pathway provided by the node links [23]. However, this is just an assumption and there is no specific real-time implementation scenario. In order to support this claim, a delay has been introduced in all the topologies considered.

Although the GUI of ONOS has a better ability to display the connection between hosts and switch than of ODL, yet ONOS gives away much before ODL when the number of switches are increased above 300 [24]. The system structures represent the hosts or nodes while the blue boxes represent the switches.

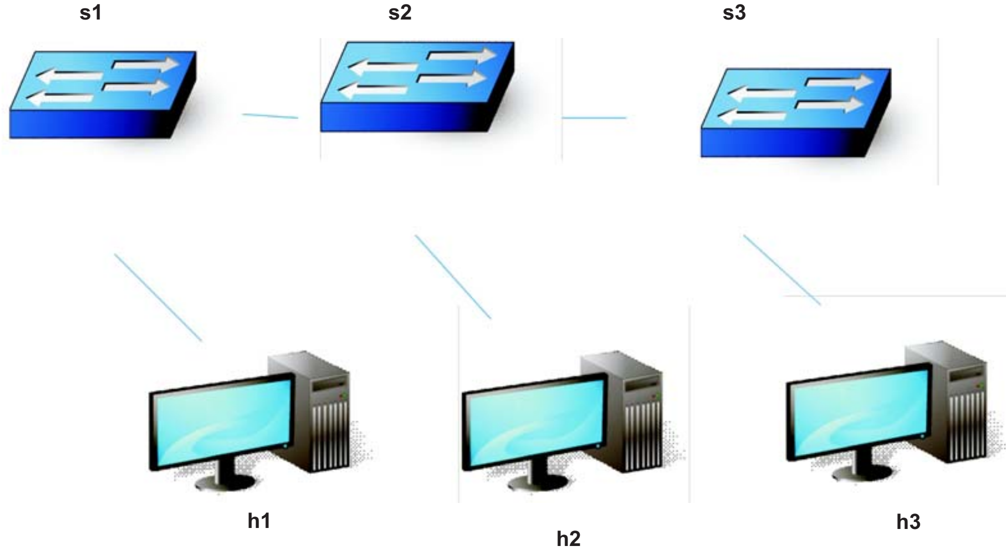

Figure 1: Linear topology

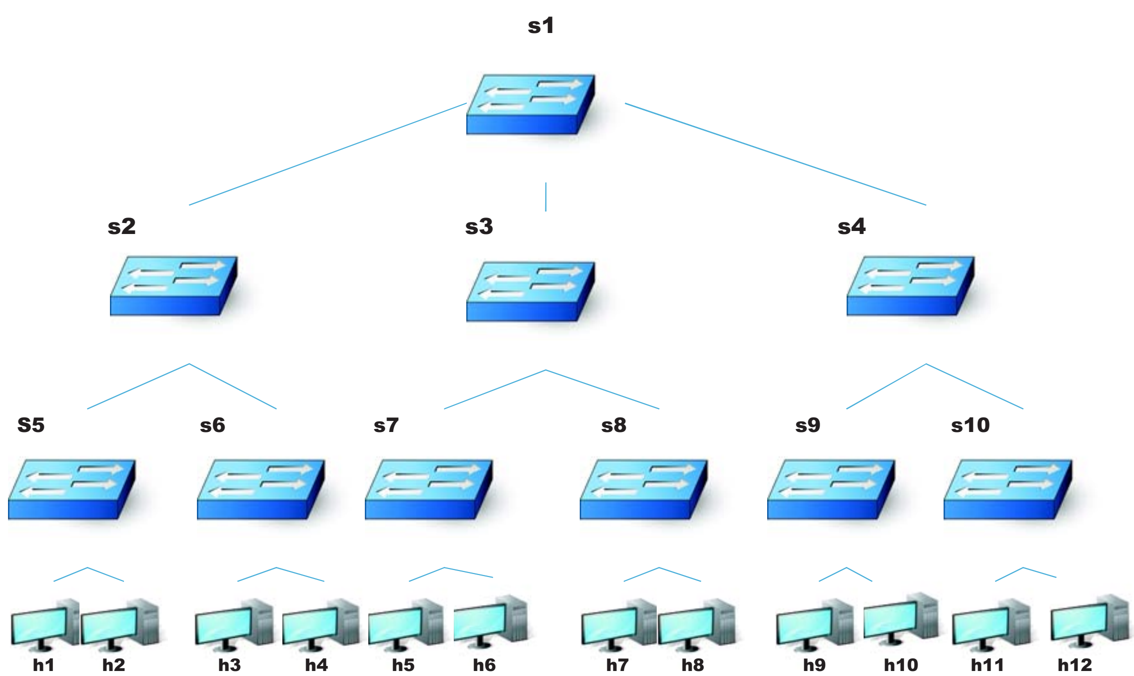

Figure 2: Tree topology

4. COMPARISON RESULTS

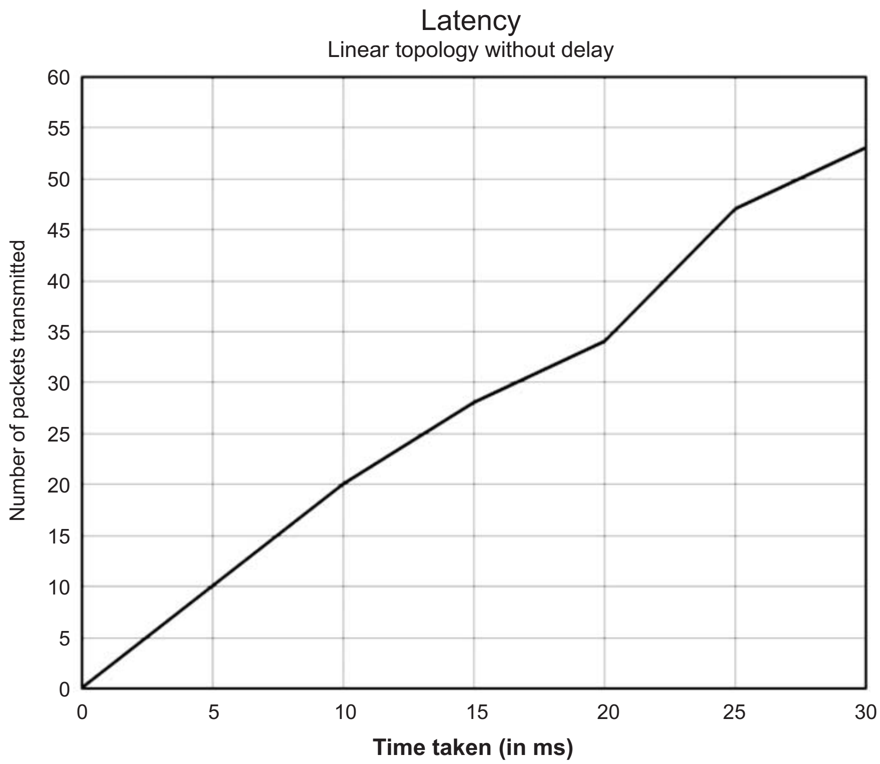

Figure 3

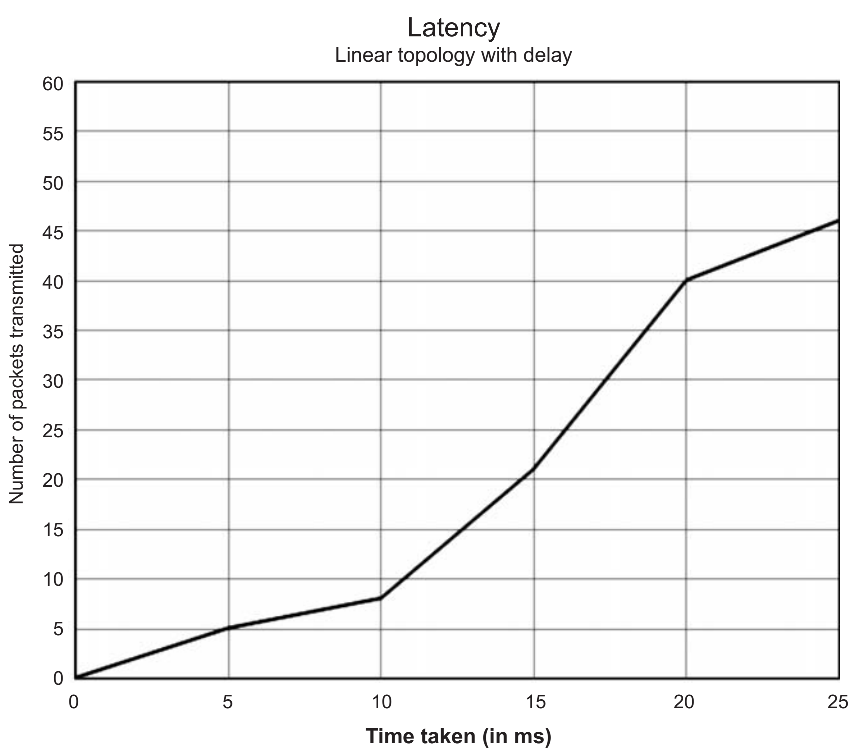

Figure 4

In figure 3, a Latency plot of a linear topology without any time delay is shown by taking time on *x*-axis and number of packets transmitted on *y*-axis [25]. The graph clearly shows a hike in the number of packets transmitted as the time (in ms) increases. The above observation shows that when there is no delay, a linear topology provides an added advantage of all the packets being transmitted without any significant loss or flow-forwarding issues. However, judging the facts only on the basis of a single topology might not produce effective and efficient circumstances. This is the reason the topology further induces to add up a time delay so that the controller's consent over the lateral entry packets can be determined and to accurately measure the actions that are to be taken in further enhancing this type of issue in the future.

In figure 4, we have introduced a minimal delay to see how effective the packet transmission rate can be if the packet flow is disturbed [26]. As shown in the figure, the number of milliseconds is greater when compared to the previous scenario. The number of packets transmitted have diminished and the time delay made the decision of flow forwarding to be delayed as well. This drives home the fact that introduction in delay can be viewed as an obstacle to the network pathway and that the flow parameter characteristics do not function as effectively as before.

The above two scenarios were based on linear topology, which is a straightforward approach in any type of network, and a safe assumption to observe what happens in the initial stages.

Next, we take into consideration the tree topology. The reason for choosing tree topology over other topologies is because it is an extension of star and bus topology. So, tree topology can be implemented where star and bus topologies cannot be individually applied due to scalability issues in the network [27].

The main aim is to change the appropriate links and determine which would be effective to use for high data transmission in minimum time.

Figure 5

The tree topology statistics without any time delay has been shown in figure 5 by taking into consideration the time taken on *x*-axis and the number of packets transferred on *y*-axis. The plot will be the same as the earlier case, but since the scenario has changed, there is going to be a difference in the flow of packets from source to destination. The fact can be observed that during the last few milliseconds, there is a significant rise in the number of packets transferred. This can be due to the extra links which tree topology provides, which is an added advantage when compared to linear topology. The flow table shows that more packets have been sent within the stipulated time, which is quite high when compared to linear topology. Also, initially it can be observed that the transmission started off at a slower pace.

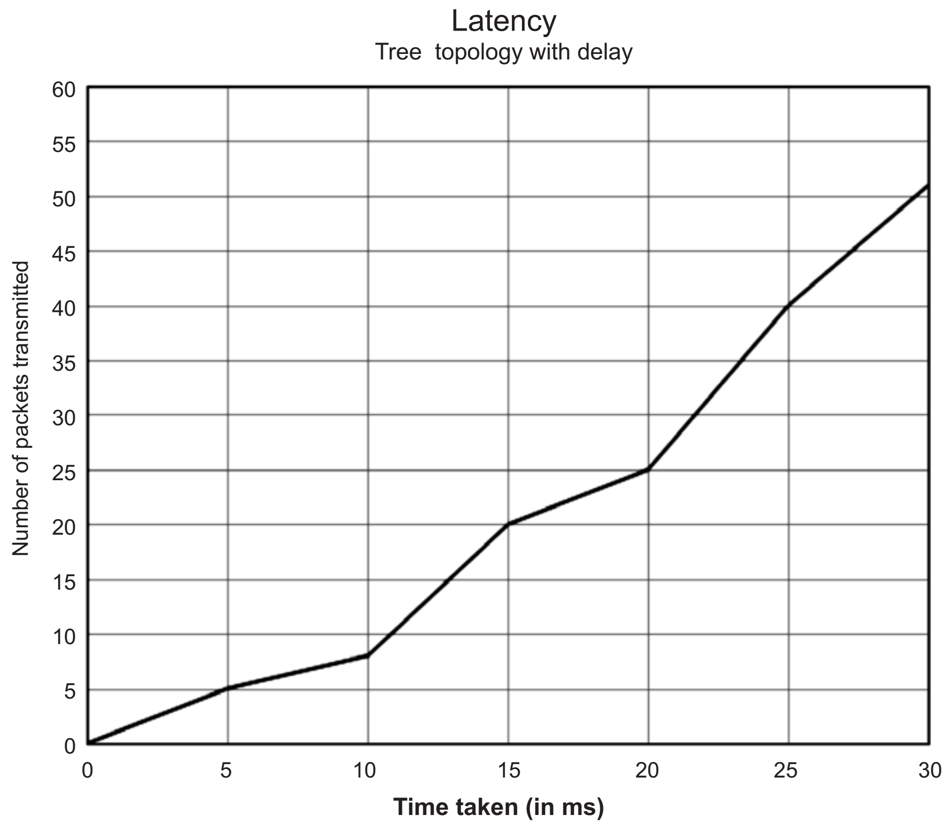

Figure 6

On the contrary, a different case has been observed when tree topology has been implemented with a constant time delay in the topology, which is shown in figure 6 [28].

It can be inferred that during the first 85% of the time interval, there is a constant increase in number of packets transmitted, but after that, there is a linear slope and packets have not transmitted in the last 15% of the time interval. This might be due to the disturbance caused to one of the hosts unable to communicate with another and hence, the overall number of packets sent and received have seen a steady fall. This also drives home the point that latency is an effective measure when we are to measure time as the liability constraint and only efficient tracking and monitoring might fetch 99% results in packet transmission [29].

On the other hand, even throughput is as much, if not, of greater importance to latency. The number of packets sent or received within an amount of time shows the accuracy of the delivery or the promising nature of the network [30]. Just how fast a network is going to tolerate the traffic conditions is an important factor which can determine various functions of an organisation effectively. That being said, figure 7 shows a graph for throughput in linear topology of ONOS.

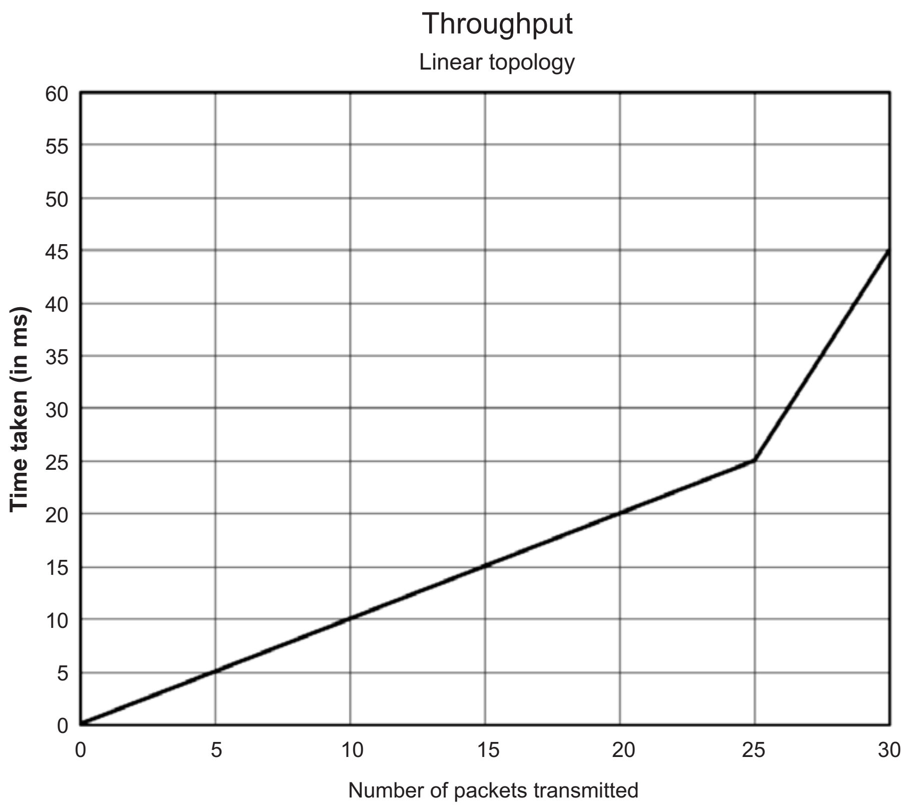

Figure 7

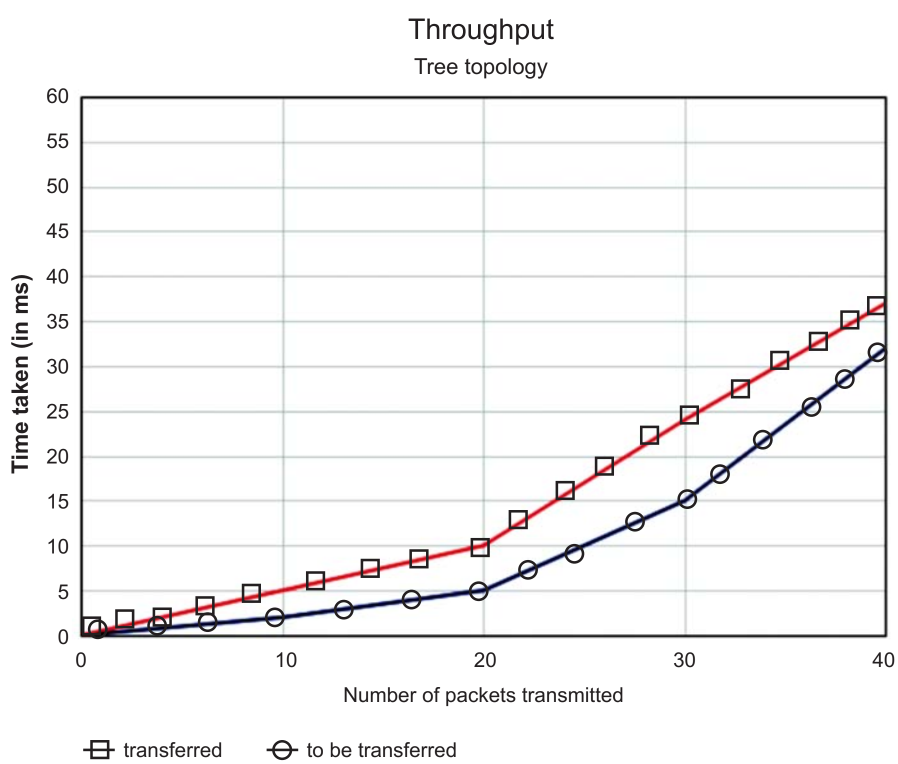

Figure 8

An observation is recorded noting the number of packets that can be transmitted within the specified time, considering that there is no delay within the network. It can be observed that the flow has a constant diagnol plot upto a certain period of time, and a steady sudden decrease just after crossing a time mark. Earlier on, on an average, one packet per millisecond was being transmitted and the flow continued to remain smooth. After a certain phase, only few packets could be transmitted in the same amount of time, which is relatively higher when compared to the previous case. The flow has slowed down by multiple times. This is due to the adverse traffic conditions the network has been facing until a period of time after which, the node could not forward any more packets at the same rate to the switch, and the switch in turn changed it's transmission rate of flow forwarding rules to the controller. Figure 8 also gives a clear idea in the form of throughput in tree topology about the change in virtual switch and host environment and how accurate the results can be over a period of time.

Table 1

		ONOS		*ODL*	
Topology	*Parameter*	*Number of packets transmitted (in thousands)*	*Time taken (in seconds)*	*Number of packets transmitted (in thousands)*	*Time taken (in seconds)*
Linear	Latency (without delay)	10	148	10	133
Linear	Latency (with delay)	10	199	10	266
Tree	Latency (without delay)	10	171	10	183
Tree	Latency (with delay)	10	288	10	347
Linear	Throughput (without delay)	10	310	10	302
Tree	Throughput (with delay)	10	453	10	554

5. CONCLUSION

In the proposed paper, a new environmental setup in the network has been considered for the comparison of two widely used controllers- ONOS and ODL. ONOS is chosen because of its flexibility to operate in a multiple cluster environment and due to its Web UI. OpenDayLight is chosen because of its common platform to provide solutions for various types of networks. The paper also stresses the mode of comparison restricted to the key parameters – throughput and latency, in determining the stability of a network. Considering the results obtained from various graphs by introduction of two main topologies – linear and tree topology, the packet transmission rate is recorded and the flow statistics of the network are observed. An analysis has been performed for both the controllers, taking into consideration the above parameters and topologies, and it has been clearly found that ONOS gives a better advantage in packet forwarding when compared to OpenDayLight. This is mainly due to the ability of the ONOS to visualize the network traffic and also corresponding to efficient rate of transmission of data packets. As depicted above, the behaviour of both the controllers differ on a larger scale when flow rate is the factor of consideration. Different controller versions are being released every year. The above facts are constrained only to the cardinal release of ONOS controller and Helium release of ODL controller. The current versions of ODL and ONOS might show a great deal of improvement in network statistics and path flow management. Considering the future aspects of ONOS and ODL controllers, it is quite certain that efficient and effective data forwarding can be done with the help of better abstraction of the control plane and implementation of already defined protocols, which guide the way for ideal software defined controllers operating in a multi-vendor environment.

6. FUTURE WORKS

Many challenges faced by SDN controllers still need assuring attention and many multinational organizations across the globe have started adapting to SDN controllers based on their requirements, size of the organisational network, ease of use and pleasant interface functionalities.

Recently, ONOS and ODL controllers have released more than three versions within an year and collaborated with few of the top companies in the world. As of now, only network protocols like BGP, DHCP are able to be deployed on a user level interface driven software, supported by VMware. ONOS is further moving towards a proposal where Inter Cluster ONOS Network Application (ICONA) can be built to support the centralization of multiple controllers and imply more fault detection and management techniques through access based control. The recent release of ODL named “Boron”, led to a strong deployment benefit to cloud and network function virtualization support with enhancement in performance parameters and increased large scale networks. Different controllers offering different functionalities in wide range of aspects are yet to be developed for the market and since, security of the network is on high demand today, the protection of data transmitted through these controllers is a vital section to be considered and implementation of more secure firewalls and packet level filters are to be developed and optimized for secured network administration and control.

REFERENCES

[1] “Software-Defined Networking Using OpenFlow: Protocols, Applications and Architectural Design Choices” by Wolfgang Braun * and Michael Menth.

[2] “Centralization of Network Using Openflow Protocol” - Ankita Vinod Mandekar [1*], Krishna Chandramouli [2]

[3] “OFMon: OpenFlow Monitoring System in ONOS Controllers” (woojoong kim, jian li, james won-ki hong, young-joo sub).

[4] Is SDN the Real Solution to Security Threats in Networks? A Security Update on Various SDN Models Naveen Bindra [*], Manu Sood

[5] “ONOS: Towards an Open, Distributed SDN OS” (Pankaj Berde†, Matteo Gerola‡, Jonathan Hart†, Yuta Higuchi§, Masayoshi Kobayashi§, Toshio Koide§, Bob Lantz†, Brian O’Connor†, Pavlin Radoslavov†, William Snow†, Guru Parulkar†).

[6] http://www.techrepublic.com/article/opendaylight-one-open-source-sdn-controller-to-rule-them-all/

[7] https://wiki.onosproject.org/display/ONOS/The+ONOS+Web+GUI

[8] https://www.opendaylight.org/

[9] http://mininet.org/sample-workflow/

[10] “Software Defined Networks (S.D.N): Experimentation with Mininet Topologies”-Deepak Kumar [*], Manu Sood.

[11] https://github.com/mininet/mininet/wiki/Introduction-to-Mininet

[12] Joint Congestion Control and Packet Scheduling in Wireless Sensor Network M. Sengaliappan [*], A. Marimuthu

[13] www.linfo.org/latency.html

[14] Searchnetworking.techtarget.com/definition/throughput

[15] "Software Defined Networking – A Comprehensive Survey" by Diego Kreutz, Member, IEEE, Fernando M. V. Ramos, Member, IEEE, Paulo Verissimo, Fellow, IEEE, Christian Esteve Rothenberg, Member, IEEE, Siamak Azodolmolky, Senior Member, IEEE, and Steve Uhlig, Member, IEEE.

[16] http://www.infoworld.com/article/2614152/sdn/opendaylight--a-big-step-toward-the-software-defined-data-center.html

[17] https://wiki.onosproject.org/display/ONOS/The+ONOS+Web+GUI

[18] http://www.slideshare.net/opennebula/clash-of-titans-in-sdn-opendaylight-vs-onos-elisa-rojas

[19] http://whatis.techtarget.com/definition/northbound-interface-southbound-interface

[20] Survey on Identifying Packet Misbehavior in Network Virtualization S. Reshmi [1*], M. Anand Kumar

[21] Leveraging the Power of Software Defined Paradigm to Control Communication in a Network X. Agnise Kala Rani [*], Deepa Balagopal

[22] Fast IP Hopping Protocol SDI Implementation Vladimir Krylov [1*], Kirill Kravtsov [2], Eleanora Sokolova

[23] Shortest and Efficient Multipath Routing in Mobile ad hoc Network (MANET) S. J. Sultanuddin [1*], MD. Ali Hussain

[24] https://www.sdntesting.com/sdn-testing/

[25] "Address Allocation Algorithm with Cooperative Communication in MANET" by Parameswaran T1 , Dr.Palanisamy C2, Logeshwari N3

[26] LUNAR: Working and Performance Evaluation in MANETs Shitalkumar A [*], Vijay T Raisinghani

[27] http://www.ianswer4u.com/2012/01/tree-topology-advantages-and.html

[28] A Survey of Factors Influencing Network Lifetime in Delay Tolerant Network D. Jyothi Preshiya [1*], C. D. Suriyakala

[29] http://www.cs.unc.edu/~tracker/media/pdf/Miller2002_latencymeter.pdf

[30] "Performance Analysis of Mobile Ad-Hoc Routing Protocols by Varying Mobility, Speed and Network Load" by Nilotpal Sarmah.