RAMReS Sciences des Structures et de la Matière Vol. 8, N°2 (2024) 32 - 41
http://publication.lecames.org/index.php/mat ISSN: 2630-1180

Performance analysis of SDN controllers in various network topologies

Rolph Abraham Yao*, **Ferdinand Tonguim Guinko**

Université Joseph KI-ZERBO, Ouagadougou, Burkina Faso

Received: 16 October 2024 / Received in revised form: 10 December 2024 / Accepted: 22 December 2024

Abstract:

Software Defined Network (SDN) is a growing concept. It allows separating the control layer from the data layer, making the network programmable, and having a centralized view and management of the network. A high-performance controller is essential for efficient traffic management in SDN as it allows controlling and monitoring the entire network. This paper presents a performance study of Floodlight, ONOS, OpenDaylight (ODL), and Ryu controllers in Linear, Single, and Tree topologies. Mininet is used to create these topologies. The performance analysis criteria of the controllers are based on bandwidth utilization, jitter, packet transmission rate, round-trip time (RTT) min, and throughput in different topologies. Our analysis reveals that ONOS has the best performance in terms of bandwidth utilization, jitter, RTT min, and throughput in all topologies. ODL has the best performance in terms of packet transmission rate in all topologies.

Keywords: SDN controllers; Bandwidth; Packet transmission rate; Round-trip time; Throughput; Jitter.

1 Introduction

SDN is an innovative technology aimed at centralizing control, improving programmability and orchestration of network resources, and virtualizing these resources by separating them from the physical elements of the network [1]. SDN has many advantages such as security, network control, scalability, agility, and flexibility. Figure 1 shows the levels that make up the SDN architecture. The application layer groups together all the software that facilitates the implementation of new network functionalities, such as traffic engineering, security, quality of service, etc. The control layer, also called the "control plane", is mainly made up of one or more SDN controllers. Its role is to control and manage infrastructure equipment via the South-Bound API.

*Corresponding author:
Email address: *abraham.yao@ujkz.bf* (*R.A. Yao*)
https://doi.org/10.70974/mat08224032

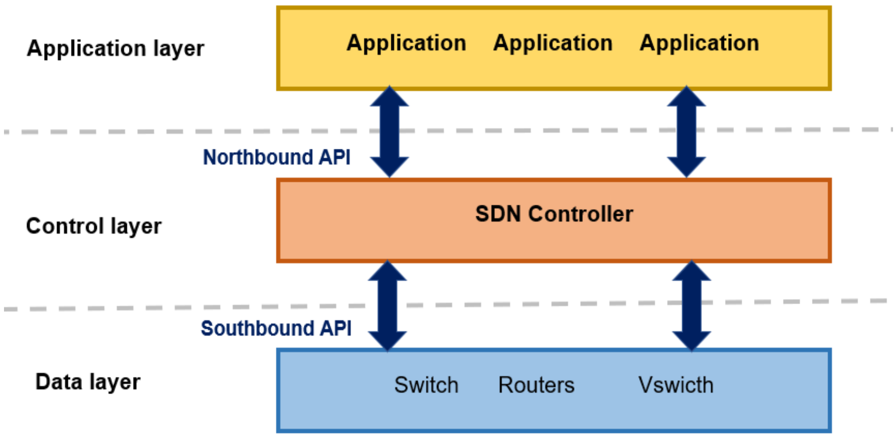

Fig. 1. Different layers of SDN architecture.

The data plane, also called the "data layer", is composed of routing equipment such as switches or routers. Its main function is to transmit information and collect statistical data.

2 Overview of SDN controllers

Controller is an important component of SDN. It provides a complete view of the network, including SDN devices in the data plane. These resources are connected to management applications and perform policy-based flow actions between devices. The ONOS, OpenDaylight, Floodlight, and Ryu controllers are presented below.

Floodlight Controller: It is an open-source controller based on the Java programming language. The Floodlight controller is licensed under the Apache License and does not support Open-Stack. It is designed to work according to Open-Flow standards and is composed of different modules and application controls [2].

ONOS Controller: Open Network Operating System controller is open source and uses the Java programming language. ONOS is particularly suitable for large-scale networks, supports real-time network configuration and control, and does not require the execution of routing control protocols [3].

OpenDaylight Controller: The ODL controller is an open source, modular, open platform controller based on the Java programming language. It allows to automate, customize, and manage networks of any scope and size [4]. It supports other protocols besides OpenFlow.

Ryu Controller: Ryu Controller is an open source SDN-based controller that increases network agility by simplifying traffic adaptation and management. Other network device management protocols such as Netconf, OF-config, OpenFlow are also supported by Ryu [5].

3 Review of literature

Several authors have conducted studies on controllers in SDN. We present some works published in the literature that are related to our work.

Altangerel et *al.* [6] analyzed the performances of Floodlight, ODL and POX controllers. Mesh and Tree typologies are created with the Mininet emulator. Criteria such as the average arrival time of the first packet of the flow as well as the throughput of TCP and UDP flows are evaluated. The analysis of the results shows that Floodlight and ODL are the best performers in terms of packet response time (RTT) and throughput than the POX controller. However, criteria such as bandwidth, throughput of transmitted packets and jitter are not addressed. Moreover, a change of topology could change the collected data.

Mamushiane *et al.* [7] analyzed the performance of ONOS and ODL controllers. Linear, Single, and Tree topologies are created with Mininet and the D-ITG traffic generator is used to evaluate the controllers based on delay, jitter, and packet loss. The results demonstrated that the ONOS controller has the best performance in terms of jitter, latency, and packet loss compared to the ODL controller in all topologies. Although all topologies are addressed, evaluation criteria such as bandwidth, packet rate, and throughput are missing.

Lunagariya and Goswami [8] analyzed the performance of Beacon, Floodlight, IRIS, ONOS, ODL, POX and Ryu controllers. The comparison criteria used are jitter, latency and stability, TCP and UDP throughput. The results reveal that Floodlight and ONOS controllers have better performance in terms of higher average throughput and lower jitter value. Although the paper provides several criteria for the analysis, it lacks performance measures such as bandwidth, packet transmission rate. Also, a change in the topology could lead to a change in results.

In the study of Sheikh *et al.* [9], the performance of Floodlight, NOX, ONOS, ODL, POX and Ryu controllers are evaluated in terms of throughput and latency. Mininet is used to create a customized topology. The analysis results show that ONOS and Ryu have better throughput as the number of tests increases. The throughput of POX and ODL controllers is good when the number of tests is lower, while NOX and Floodlight have the worst throughput regardless of the number of tests. Since the chosen topology is customized, a change in topology will lead to a change in conclusion.

Nuraeni *et al.* [10] analyzed the performance of ONOS, POX, and Ryu controllers. A linear topology composed of different numbers of switches and hosts is created with the Mininet emulator. Jitter, latency, and throughput are used as evaluation criteria. The results showed that POX outperforms Ryu and ONOS in all evaluation scenarios. Although the paper covers three evaluation criteria, bandwidth and packet throughput are not discussed.

In the study of Naim *et al.* [11], the performance of POX and Ryu controllers based on jitter, packet delivery ratio, packet loss and throughput, has been evaluated and compared. The Mininet emulator is used to create a custom topology. The results revealed that the POX controller is more efficient in terms of throughput. On the other hand, in terms of packet loss, jitter and packet delivery ratio, Ryu has better performance. Considering the chosen custom topology, these results could change following the use of other topologies. Also, performance measure such as bandwidth, RTT are missing.

Floodlight, ONOS, ODL and Ryu controllers are evaluated by Ali [12]. Custom, Linear, Single, torus and tree topologies are created with Mininet and iperf tool is used to measure jitter, latency, packet loss and throughput. The results indicated that Ryu outperforms Floodlight, ODL and ONOS in the evaluation scenarios. Although the paper covers the different evaluation criteria, bandwidth and transmitted packet rate are not discussed.

The present paper analyzes and compares the performance of Floodlight, ONOS, ODL and Ryu controllers in Single, Linear and Tree topologies. The choice of these controllers is explained by their architectures (distributed and centralized) and their programming language. Performance criteria such as bandwidth, jitter, packet transmission rate, minimum RTT and throughput are used.

The objective of this study is to determine the best combination of controllers and topology for network scenarios in order to increase SDN performance.

To achieve this objective, Mininet is used to create the different topologies. Then, the four controllers are compared on these different topologies based on bandwidth, jitter, packet rate, RTT min and throughput.

4 Materials and methods

4.1 Simulation environment

The environment for the simulation is created on an Intel® Xeon(R) E5-2620 0 @ 2.00 GHz (2 processors) 32 GB RAM computer. Ubuntu 22.04 LTS is configured on this machine. The Floodlight, ONOS, ODL and Ryu controllers are installed on Ubuntu.

4.2 Different topologies used

Mininet is an open-source emulator and allows the simple creation of nodes, links, controllers and all network components. Mininet offers ease of use, performance accuracy and scalability [13]. In our work, Mininet is used to create a Linear topology composed of 123 Switches and 123 hosts, a Single topology composed of 01 Switch and 117 hosts and a Tree topology composed of 21 Switches and 64 hosts.

The figure 2 shows an overview of the different topologies.

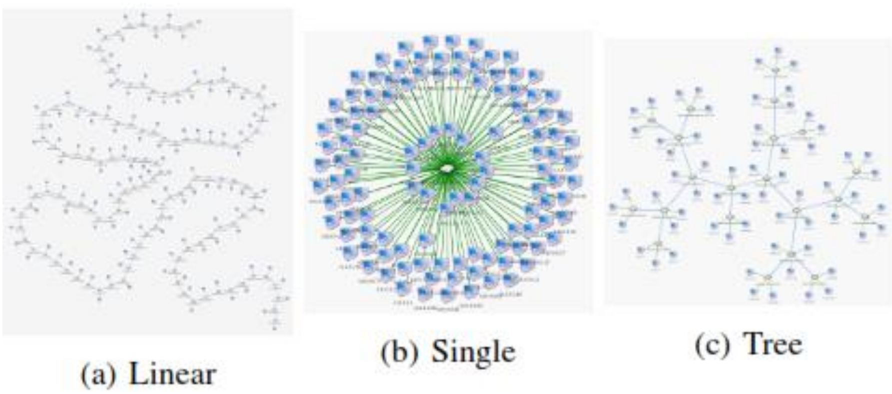

Fig. 2. Different network topologies.

5 Results and discussion

In this section, simulation results and discussion are presented. The performance of the controllers is analyzed with Linear, Single and Tree topologies. The parameters used to evaluate the performance are bandwidth utilization, jitter, packet delivery rate, minimum RTT and throughput.

5.1 Bandwidth

An analysis of bandwidth utilization for all controllers is performed in the different topologies. To run the test, these commands are used: "iperf -s and iperf -c 10.0.0.X".

The various data collected are illustrated in Figure 3. Based on the results, ONOS has the best bandwidth compared to Floodlight, ODL and Ryu in all topologies. ONOS has the highest bandwidth in the Tree topology with 52.7 Gbps and the lowest bandwidth in the Linear topology with 12.2 Gbps. Ryu has the second highest bandwidth with tree topology (52.5 Gbps), followed by Floodlight with tree topology (52.3 Gbps). ODL has the lowest bandwidth with all topologies. ONOS is ideally suited to contexts where extensive data exchange is required.

5.2 Jitter

Jitter refers to small, unpredictable variations in the arrival time of data packets on a network. When jitter occurs, the problem is usually caused by saturated bandwidth, old network infrastructure, ethernet cables, or terminals [14].

The commands: "iperf -s -u -i 1" and "iperf -c 10.0.0.X -u -t 60", are used to run the tests and measured the jitter during 60 s. Figure 4 shows the jitter values of the controllers at given time intervals. From the graph, ONOS has the best jitter compared to Floodlight, ODL and Ryu in all topologies. ONOS has the lowest jitter in the Tree topology at 0.009 ms and the highest jitter in the Linear topology at 0.014 ms. Ryu has the second best jitter with Tree topology (0.011 ms), followed by Floodlight with Tree topology (0.014 ms). ODL has the highest jitter with all topologies. ONOS has the lowest jitter, making it a preferred option for applications requiring low latency and consistent quality of service. Ryu and Floodlight follow closely with slightly higher jitter. OpenDaylight shows greater variation, which can be problematic for sensitive streams.

5.3 Packet delivery rate

In this section, we compare the packet transmission rates at the four controllers. The following command is used to run the test: "h1 ping -c50 hX".

The transmission rates of 50 packets were compared and illustrated in Figure 5. From the results obtained, ODL has the best packet transmission rate compared to Floodlight, ONOS and Ryu in all topologies. ODL has the highest packet transmission rate in the Tree topology with 1.0201 p/s and the lowest packet transmission rate in the Linear topology with 1.0192 p/s. ONOS has the second-highest packet transmission rate with Tree topology (0.9981 p/s), followed by Floodlight with Tree topology (0.9975 p/s). Ryu has the lowest packet transmission rate in all topologies. ODL has a very high packet transmission rate, making it ideal for critical environments or environments sensitive to packet loss.

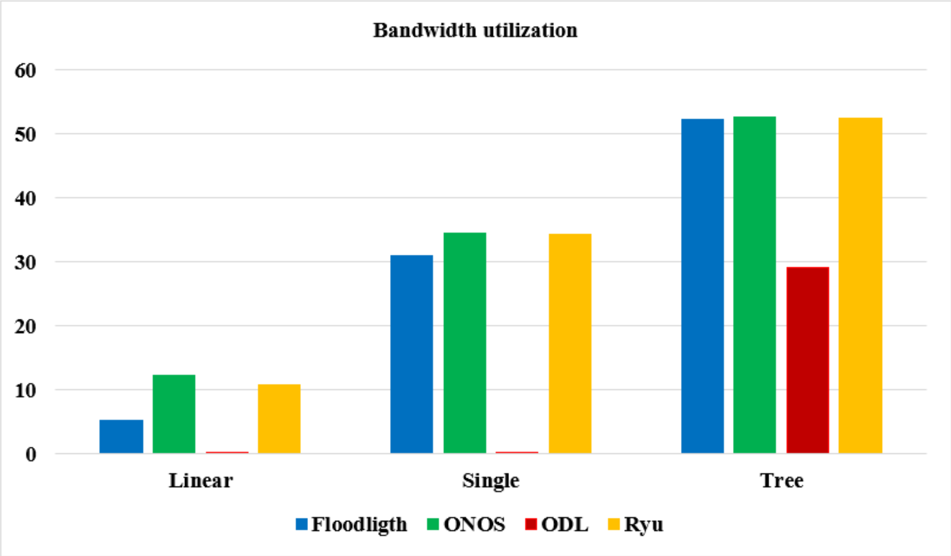

Fig. 3. Comparison of controller bandwidth.

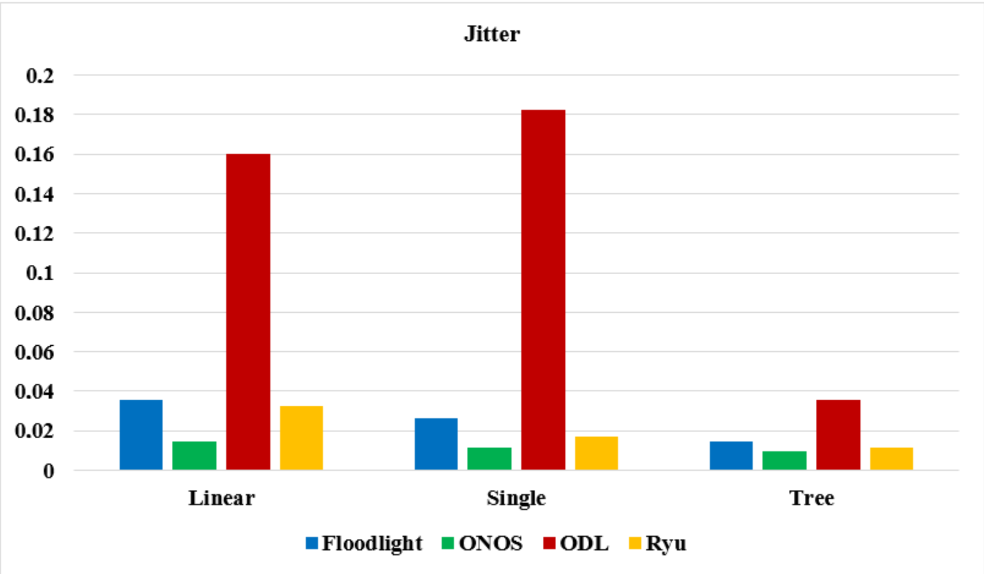

Fig. 4. Comparison of controller jitter.

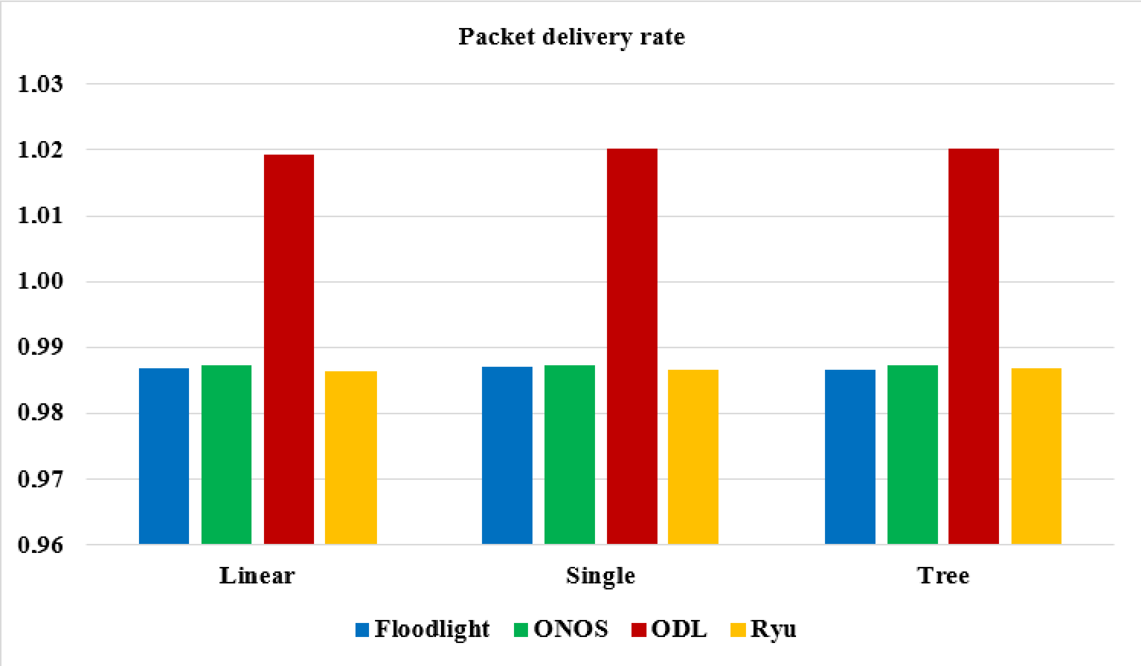

Fig. 5. Comparison of packet delivery rate of controllers.

5.4 Round-trip time

This test is performed by determining the round-trip time (RTT) min between nodes h1 and hX by running a "ping" connectivity test. Figure 6 shows min RTT measured. The following command "h1 ping -c50 hX" is executed between nodes h1 and hX.

The analysis of the graph shows that ONOS has the best RTT min compared to Floodlight, ODL and Ryu in all topologies. ONOS has the lowest RTT in the Tree topology at 0.029 ms and the highest RTT in the Linear topology at 0.151 ms. Ryu has the second best RTT with Tree topology (0.034 ms), followed by Floodlight with Tree topology (0.037 ms). ODL has the highest RTT in all topologies. ONOS has the lowest RTT min, making it an excellent choice for low-latency applications.

5.5 Throughput

Throughput is the flow of data from a source machine to a destination machine at a given time. Typically, it is expressed in bits per second (bps). The following commands: "iperf -s -i 1" and "iperf -c 10.0.0.X -t 60" measure the flow rate between source h1 and destination hX during 60 s and the values are recorded as presented in Figure 7.

From the obtained graph, ONOS has the best throughput compared to Floodlight, ODL and Ryu in all topologies. ONOS has the highest throughput in the Tree topology at 42.1 Gbps and the lowest throughput in the Linear topology at 9.24 Gbps. Ryu has the second highest throughput with Tree topology (34.9 Gbps), followed by Floodlight with tree topology (33.5 Gbps). ODL has the lowest throughput with all topologies. ONOS offers the best throughput, indicating its ability to handle network traffic under both normal and elevated conditions.

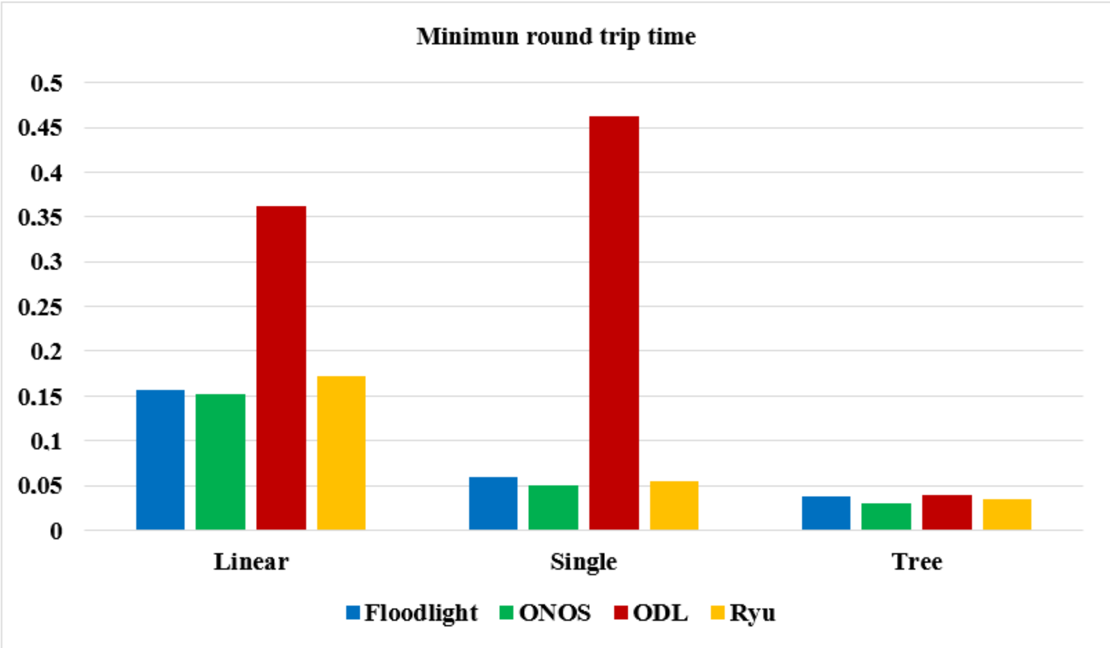

Fig. 6. Comparison of controller round-trip time min.

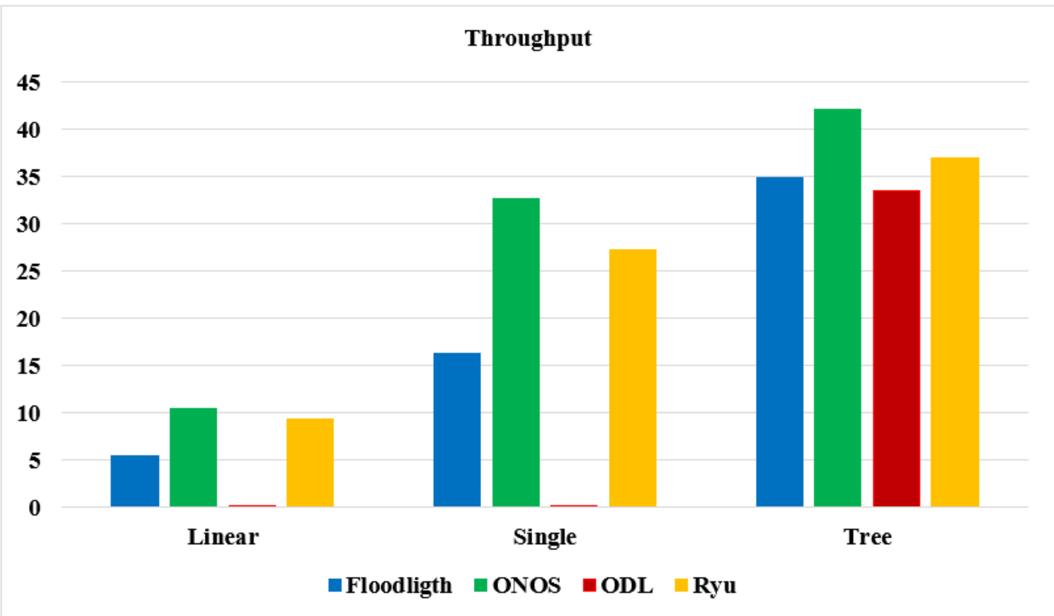

Fig. 7. Comparison of controller throughput.

In summary, ONOS has the best performance in terms of bandwidth, jitter, RTT min and throughput in all topologies compared to Floodlight, ODL and Ryu. ODL has the best performance in terms of packet transmission rate in all topologies compared to Floodlight, ONOS and Ryu.
With the results obtained, we can conclude that ONOS is the best controller for high-performance networks, followed by Ryu, Floodlight and ODL, respectively. The ONOS controller's performance is due to its distributed architecture, resilience and ability to manage large networks with high availability requirements. The best topology for high-performance networks is Tree topology. By combining Tree topology and the ONOS controller, we achieve even better results. These results will help SDN users to optimize their network performance.

6 Conclusion

In this paper, the performance of Floodlight, ONOS, ODL and Ryu controllers was analyzed in Linear, Single and Tree topologies. Evaluation criteria focused on bandwidth utilization, jitter, packet delivery rate, round trip time and throughput. The Mininet emulator was used to create these different topologies. From the evaluation, ONOS performs better in terms of bandwidth utilization, jitter, RTT min and throughput in all topologies. ODL outperforms in terms of packet transmission rate. Analysis of the results has enabled us to conclude that the ONOS controller is the best controller compared with the Floodlight, ODL and Ryu controllers. However, changing the simulation environment could change the results we obtained. In future work, we will evaluate the performance of these SDN controllers in a real environment and also in a large-scale network.

References

[1] W. Xia, Y. Wen, C. Foh, D. Niyato, H. Xie, *A survey on software-defined networking,* IEEE Communications Surveys Tutorials 17 (2014) 27-51.

[2] M. Daha, M. Zahid, K. Husain, F. Ousta, Performance evaluation of software defined networks with single and multiple link failure scenario under floodlight controller. In: “International Conference On Computing, Communication, And Intelligent Systems (ICCCIS) (2021) 959-965”.

[3] G.A. Mohammed, O.A.I. Aldabbagh, *A Comparative Evaluation of the Performance of SDN Controllers (ONOS) using DOCKER Container*, Technium 8 (2023) 25-31. https://doi.org/10.47577/technium.v8i.8585

[4] A. Eftimie, E. Borcoci, SDN controller implementation using OpenDaylight: experiments. In: “13th International Conference On Communications (COMM) (2020) 477-481”.

[5] H. Babbar, S. Rani, *Performance evaluation of qos metrics in software defined networking using ryu controller.* In: IOP Conference Series: Materials Science And Engineering, 1022, 012024 (2021).

[6] G. Altangerel, T. Chuluuntsetseg, D. Yamkhin, *Performance analysis of sdn controllers: Pox, floodlight and opendaylight, arXiv:2112.10387* (2021). https://doi.org/10.48550/arXiv.2112.10387

[7] L. Mamushiane, T. Shozi, A QoS-based evaluation of SDN controllers: ONOS and OpenDayLight. In: "IST-Africa Conference (IST-Africa) (2021) 1-10".

[8] D. Lunagariya, B. Goswami, A comparative performance analysis of stellar sdn controllers using emulators. In: "International Conference On Advances In Electrical, Computing, Communication And Sustainable Technologies (ICAECT) (2021) 1-9".

[9] M. Sheikh, I. Hwang, E. Ganesan, R. Kharga, Performance Assessment for different SDN-Based Controllers. In: "30th Wireless And Optical Communications Conference (WOCC) (2021) 24-25".

[10] M. Nuraeni, E. Wahyudi, N. Iryani, *Perbandingan Kontroler POX, Ryu dan ONOS Pada Arsitektur Software Defined Network (SDN) Menggunakan Topologi Linear*, Journal Of Telecommunication Electronics And Control Engineering (JTECE) 4(2) (2022) 66-79.

[11] N. Naim, M. Imad, M. Hassan, M. Afzal, S. Khan, A. Khan, POX and RYU Controller Performance Analysis on Software Defined Network. In: "EAI Endorsed Transactions On Internet Of Things (2023)".

[12] U. Ali, *Performance Comparison Of SDN Controllers In Different Network Topologies, Research Square* (2024). https://doi.org/10.21203/rs.3.rs-4826985/v1

[13] K. Kaur, J. Singh, N. Ghumman, Mininet as software defined networking testing platform. In: "International Conference On Communication, Computing Systems (ICCCS) (2014) 139-142".

[14] J. Sojan, K. Haribabu, Monitoring Jitter in Software Defined Networks. In: "International Conference On Advanced Information Networking And Applications (2022) 635-645".