International Journal of Computing and Digital Systems
ISSN (2210-142X)
Int. J. Com. Dig. Sys. 12, No.1 (Nov-2022)

https://dx.doi.org/10.12785/ijcds/1201103

Load Balancing Implementation Strategy for Various Services in Software Defined Network using ONOS Controller

Rohmat Tulloh[1], Muhammad Iqbal[2], Mohammad Rifki Baihaqi[3] and Aditya Bayu Aji Pamungkas[4]

[1,2,3,4] *School of Applied Science, Telkom University, Bandung, Indonesia*

E-mail address: rohmatth@telkomuniversity.ac.id

Received 30 Jun. 2021, Revised 18 Aug. 2022, Accepted 29 Nov. 2022, Published 30 Nov. 2022

Abstract: The popularity of Software Defined Network (SDN) is due to its ability to simplify and automate operational network processes to improve service performance. SDN is not only related to the separation of control plane and data plane, but its architecture also includes network services at other top tiers such as load balancing, security, and application performance. Load balancing is a technique to divide traffic loads equally. The correct load balancing algorithm will make the network better stable because network resources must always be operational and available to cope with the increasing demand for services and service users. Scalability and diversity of services on the network such as HTTP, FTP, and VoIP are also concerned with selecting the proper load balancing algorithms. SDN-based software makes it easy to develop and integrate with other software and hardware. This study implemented a load balancing algorithm using an L4-L7 load balancer connected with an SDN controller, ONOS, which automates traffic load sharing on a Web server, FTP server, and VoIP server. The scheduling algorithms used are round robin, least connection, dynamic ratio, and ratio. The test parameters used are Throughput, Response time, Request loss, Block call, and CPU Utilization.

Keywords: Software Defined Network, Load Balancing, Least Connection, Round Robin, Dynamic Ratio, Ratio.

1. Introduction

Computer network technology is developing very rapidly, and the impact of this rapid development makes the human need for services through internet access increase sharply. The increase in service requests and the number of users of internet services make the server often overloaded. A common way to overcome overload is to add a new server or add a hard disk to the database, but this costs a considerable amount. The solution that can solve it is the application of load balancing techniques. Load balancing is a mechanism for dividing compute loads into multiple servers. The goal is to optimize resources and increase throughput to not overload [1]–[3].

However, load balancing algorithms are very diverse. Therefore, selecting the proper load balancing algorithm will make the network have better stability because network resources must always be available to cope with the increasing demand for services and the number of service users. In addition, scalability and diversity of services on the network such as HTTP, FTP, and VoIP are also essential benchmarks in selecting the proper load balancing algorithms [4].

Software-Defined Network (SDN) is an innovation in network architecture. The concept of SDN is to separate the Control Plane and the Data Plane through the use of the OpenFlow protocol [5], [6]. Control Plane is moved out of the network device so that only the Data Plane is inside the network device [7] , and the load balancing policy can be managed centrally through a controller. SDN centralized network control makes network setup easier, flexible, and faster [8]–[10]. SDN-based software makes it easy to develop and integrate with other software and hardware. SDN can also be used to change network behavior, make such changes automatically, and maximize network devices such as load balancing [11].

2. Current State and Challenges

The load balancing application works by dividing the load in requests given by the client to the server [12], [13]. The algorithms observed are dynamic ratio, ratio, round robin, and least connection.

The dynamic ratio is an algorithm whose weight ratio is based on continuous monitoring of the server and, therefore, is constantly changing and performs connection sharing based on the monitored server performance parameters in real-time. In comparison, the ratio algorithm uses a fixed ratio parameter calculation for each server. The server with the highest ratio is given a more significant load. on the other hand, servers with smaller ratio values will be given less load [14]. The Round-Robin algorithm on load balancing works when receiving requests from clients will be directed to multiple servers in turn and sequentially from one server to another to receive load evenly [15]. While the least connection algorithm serves to do load sharing based on the number of connections being served by the server. The server with the least connection will be given the next load [16].

In research [1], [17], it has been proven that the selection of the correct load balancing algorithm provides improved performance of Web Server, FTP Server, and VoIP services on conventional networks. In SDN-based networks dedicated to high scalability networks, the use of load balancing algorithms is interesting to research in more depth.

Additionally, research [6], [18], uses Mininet to create virtual network environments for evaluation. However, it turns out that the use of Mininet does not reflect the actual overhead. Therefore, this research spread experimental environments and built experimental platforms using real hardware [19]. SDN controller implementation handles datalink layers to network layers only, while integrating with F5 can add SDN controller capabilities to handle from transport layer to application layer [14].

However, SDN implementation has a challenge: how the strategy is to provide good performance results. This study implemented load balancing in the transport layer to the application layer using the F5 application. F5 is used to enable applications and networks to communicate with each other. Then, analyze four load balancing algorithms on SDN-based networks consisting of MikroTik switch that act as data plane and ONOS as controller. ONOS is a variant of the SDN controller that uses the Java programming language that takes advantage of the Open Service Gateway (OSGi) framework initiative. With this framework, it will be more accessible when installing and updating applications [20], [21].

3. System Design And Implementation

A. System Design

The network topology design implemented in this study can be seen in figure 1. There are two MikroTik routers set as OpenFlow switches and connected to the ONOS controller. The switch only serves as a data plane, while the control plane function becomes centered in the controller. The ONOS controller feature used in study can be seen in figure 1. There are two MikroTik routers set as OpenFlow switches and connected to the ONOS controller. The switch only serves as a data plane, while the control plane function becomes centered in the controller. The ONOS controller feature used in this study uses a reactive forwarding application that serves to allow or reject data traffic on SDN networks. On servers 1 and 2, each has three virtual servers with different services. Each servers consist of web services with http protocol, FTP for file transfers, and SIP for VOIP services. For testing, the number of clients accessing the server will increase to see network performance. The addition of the client is done by a traffic generator that will provide a request to the server.

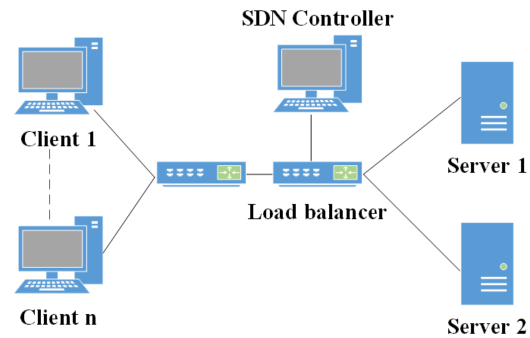

Figure 1. System Design

The software used in this study includes:

- ONOS-2.0.0 as SDN controller.
- F5-BIG-IP-12.1.4 LTM sebagai virtual load balancer.
- Ubuntu 16.04 as a server and client operating system.
- VMware is software for virtual machines that run many operating systems in one hardware.
- Nginx as Web server software.
- Proftpd as FTP server software.
- Asterisk as a VoIP server software.
- Httperf is the software used to generate requests on the webserver service.
- Jmeter Application serves to generate traffic for the FTP service.
- SIP serves as a call generator on the VoIP server service.

B. Load Balancing Algorithm

The load-feeding scenario on the server using four compared algorithms is as follows:

1) Round Robin. The round robin algorithm is used in this scenario because the round robin algorithm is the simplest and easiest to implement load balancing algorithm. This algorithm aims to distribute the traffic load on more than one connection links in a balanced way so that traffic can run optimally, increase the throughput value, reduce delay and prevent connection line overload. This algorithm can also be improved by giving weights the administrator can adjust. Put Server load on the round-robin algorithm by dividing the request load sequentially from server 1 to server 2.

2) Least Connection. Server load on the least connection algorithm is done by dividing the load based on the number of requests served by the server. The

server with the least requests will be given the next load. Otherwise, the server with many requests will be redirected to the server with a lower load. With this algorithm, it can minimize the possibility of server overload.

3) Dynamic Ratio. Server load on dynamic ratio algorithm is done dynamically by looking at the fastest node response. Dynamic ratio algorithm testing disposes of requests to servers with the same significant CPU and memory specifications. The dynamic ratio algorithm will load dynamically by looking at the CPU and memory conditions of both servers. Dynamically loading is helpful for the more balanced performance of each server.

4) Ratio. The ratio algorithm distributes requests to servers that have the same significant CPU and memory specifications. The ratio algorithm performs load sharing evenly between the two servers used.

C. Test Scenario

We are testing four load balancing algorithms on SDN using web servers, FTP servers, and VoIP servers. Testing is conducted to obtain throughput parameters, response time, and request loss on web server services and FTP servers. While the measurement of services on VOIP server using block call parameters.

As in Table I, it appears that in the webserver service is done testing by giving load to the server as many as 200 requests, 400 requests, 600 requests, 800 requests, and 1000 requests. step-by-step request generation using HTTP software.

The FTP server service is done testing by doing file transfer upload and download with a file size of 653.5 KB as much as 10 file transfers, 20 file transfers, 30 file transfers, 40 file transfers, and 50 file transfers. This scenario using Jmeter application.

The VoIP server service will be tested by creating calls as much as 10 cps, 15 cps, 20 cps, 25 cps, and 30 cps within one minute using SIPp software. In the test, 30 tests were conducted to get maximum results.

TABLE I. Test Scenario

Service Type	**Number of Testing**				
	1	**2**	**3**	**4**	**5**
Web (Request)	200	400	600	800	1000
File Transfer (File)	10	20	30	40	50
VoIP (call per second)	10	15	20	25	30

4. Performance Evaluation and Experimental Result

A. Controller Testing

In testing, the controller is done to determine if all devices can already communicate with the controller. Controllers can recognize all devices connected to the controller according to the topology used in this study, such as clients, switches, and server load balancers. For Web server, FTP server and VoIP server are integrated by load balancer server, so the controller does not monitor it. Then done test connectivity to know the data plane device and controller is running well so that the host connected with the data plane device can communicate.

B. Web Service Performance Evaluation

1) Throughput
Throughput measurement in web service aims to know the server's ability to provide services to the client. In this parameter, the more significant the throughput value, the better the load balancing performance. Figure 2 shows that the dynamic ratio algorithm provides better throughput when the number of requests is still below 600, although it is not very significant compared to the ratio algorithm. However, when the number of requests is getting bigger, it turns out that the ratio algorithm can outperform other algorithms. This condition proves that sharing the load to other servers when the service request increases has a significant effect on throughput value. In addition, the more accessible load balancing settings performed at SDN make the load balancing algorithm based on ratio settings superior to others.

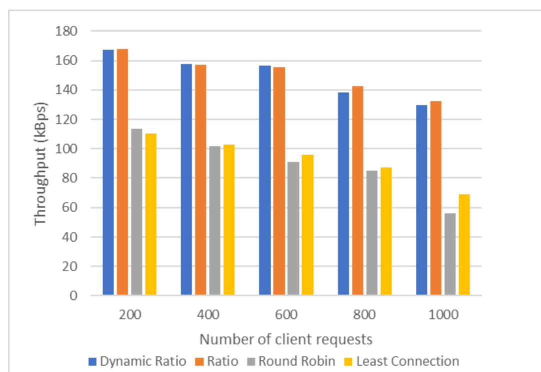

Figure 2. Throughput measurement results on web service

2) Response Time
Measurement of response time on the web service aims at how quickly a server can respond to request packets from the client. The smaller the response time number, the faster the server responds to clients. Figure 3 shows that the round robin algorithm is faster when the number of requests is still below 800, although it is not very significant compared to the least connection algorithm. However, when the

number of requests is getting bigger, it turns out that the least connection algorithm can outperform other algorithms. This condition proves that when a service request increases, it dramatically affects how it takes the server to respond to user requests.In addition, web services are not real-time, so that the response time tolerance will differ from real-time services.

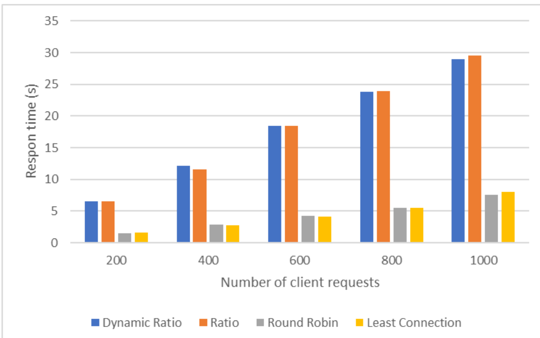

Figure 3. Measurement results of time response on web service

3) Request Loss
The measurement of request loss on the web service aims to determine the number of request failures that the server cannot serve. The smaller the number of the request loss, the more reliable the server responds to requests from the client. Figure 4 shows that the dynamic ratio and ratio algorithms are more reliable because 1000 requests did not occur until the experiment. While at the time of the request above 800, the failure rate o,46% occurs in the round-robin algorithm but is still below the threshold of request loss allowed.

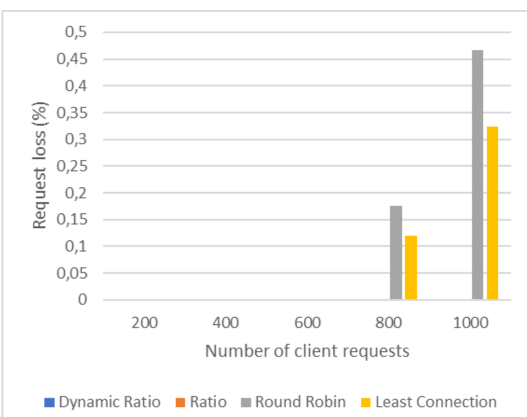

Figure 4. Measurement results of request loss on web service

4) CPU Utilization
CPU utilization measurement in web service aims to observe resources used to run the client's request service process. The smaller the number of CPU utilization, the fewer resources used. Figure 5 shows that the dynamic ratio and ratio algorithm is more wasteful although still below 8.5%, compared to algorithms round robin and least connection that looks more optimal. The least connection algorithm can reduce a load of each server more optimally than other algorithms. The smaller the CPU value obtained, it can reduce the occurrence of overload in the server.

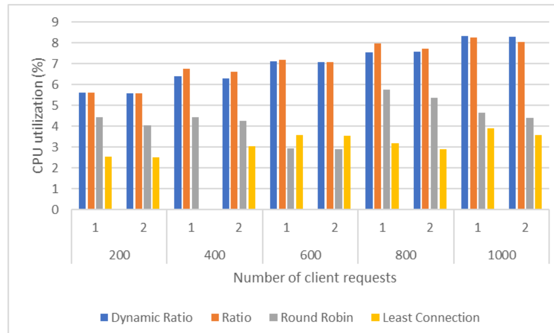

Figure 5. CPU Utilization measurement results on FTP Service

C. FTP Service Performance Evaluation

1) Throughput
Throughput measurement in FTP service aims to know the server's ability to provide services to the client. In this parameter, the more excellent the throughput value, the better the load balancing performance. Figure 6 shows that the round-robin algorithm provides better throughput than other algorithms as the number of transfer files increases. It is because the time it takes to serve all file transfers is more stable and constant on the appeal of web servers.

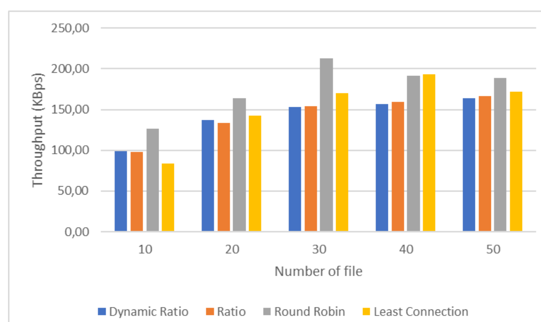

Figure 6. Throughput measurement results on FTP Service

2) Response Time
Measurement of response time in FTP service aims at how quickly a server can respond to the client. The smaller the response time level, the faster the server responds to the clients. Figure 7 shows that the value of response time increases as the number of transfer files increases. The increasing number of file transfers causes the longer it takes the server to serve requests. The fastest response time value

is found in the round robin algorithm compared to other algorithms.

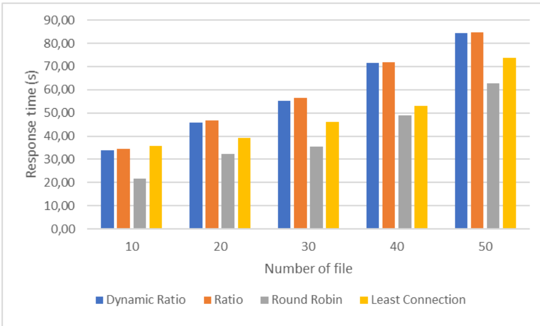

Figure 7. Response time measurement results on FTP Service

3) Request Loss
The measurement of request loss on the FTP service aims to determine the total of request failures that the server unserved. The smaller the value of the request loss, the more reliable the server responds to requests from the client. Figure 8 shows that the algorithm shows that dynamic ratio and ratio algorithms are more reliable because there are no failures. While the highest request loss when generating 50 transfer files occurred in the round-robin algorithm of 0.66%, while the algorithm at least connected 0.4% but still below the threshold of request loss allowed.

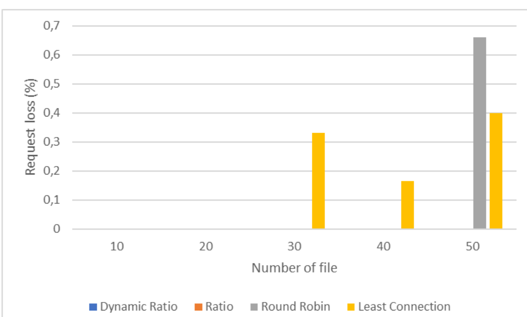

Figure 8. Request loss measurement results on FTP Service

4) CPU Utilization
CPU utilization measurement in FTP service aims to observe resources used to run the client's request service process. The smaller the number of CPU utilization, the fewer resources used. Figure 9 shows that dynamic ratio and ratio algorithms are better when compared to around robin and least connection that looks more wasteful. Dynamic ratio algorithms can reduce a load of each server more optimally than other algorithms. The smaller the CPU value obtained, it can reduce the occurrence of overload in the server.

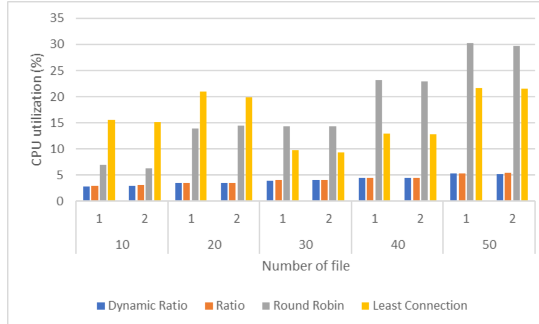

Figure 9. CPU Utilization measurement results on FTP Service

D. VoIP Service Performance Evaluation

1) *Block Call*
To generate traffic load on the VOIP server service, we used the SIPp Tester application. Traffic load in the form of calls per second is generated in stages to see the server's performance. Figure 10 shows that the increasing number of VoIP calls and the number of blocks calls also increased but still below 1% for all algorithms used. From the test results, it can be seen that at the most significant traffic load (30 cps), the Round robin algorithm has the highest block call, which is 0.96 percent, then the Dynamic Ratio is 0.66 percent, and the Algorithm Ratio is 0.53 percent. The least connection algorithm produces the smallest call block among the other four algorithms, which is 0.46 percent. It can be seen that the Round robin algorithm will experience a significant increase when the traffic load on VOIP services increases

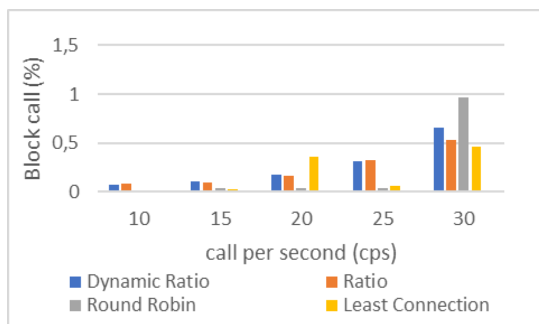

Figure 10. Block call measurement results on VOIP Service

2) CPU Utilization
CPU utilization measurement in VoIP service aims to observe resources used to run the client request service process. The smaller the number of CPU utilization, the fewer resources used. Figure 11 shows that the dynamic ratio, least connection, and ratio algorithms are better when compared to around robin algorithm that looks more wasteful. The least connection algorithm can reduce a load of each server more optimally than other algorithms. The smaller the CPU value obtained, it can reduce the occurrence of overload in the server.

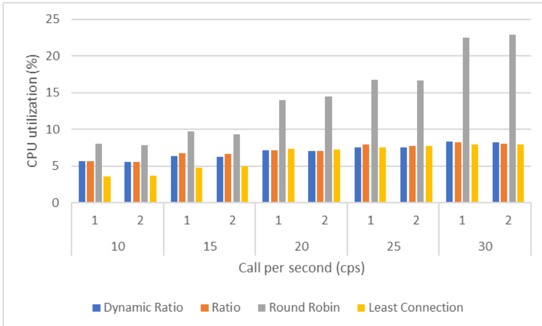

Figure 11. CPU Utilization measurement results on VOIP service

5. Conclusion

The selection of the correct load balancing algorithm will make the network have better stability because network resources must always be operational and available to cope with increasing service requests and the number of service users. Based on the test result, the service on the web server will provide better throughput when using the dynamic ratio algorithm, but if the number of requests is more than 800, it is better to use the ratio algorithm. Dynamic ratio and ratio algorithms have advantages in throughput and request loss parameters. While the dynamic ratio is not superior on the CPU utilization parameter, the measurement results show a number below 10 percent for 1000 requests. The FTP service test also consists of 4 parameters, where the round-robin algorithm looks superior in throughput and response time parameters but is very poor for request loss and CPU utilization, significantly when the number of files transferred is increasing. The Dynamic ratio algorithm gives the best results on request loss and CPU utilization parameters but still has promising results on throughput and response time. Then for VoIP services, it is shown that the least connection algorithms provide more reliable performance in controlling the increase in the number of service requests. In the Block call and CPU Utilization server measurements, it can be seen that the Least connection algorithm shows advantages over other algorithms. Based on the measurement results and the scenarios made, it can be said that the dynamic ratio and ratio algorithms are very suitable for web services. Dynamic ratio algorithm is also suitable for FTP service, and the least connection algorithm is very suitable for VOIP service.

From the measurement results, it can be concluded that the load balancing algorithm selection is strongly influenced by the characteristics of the service passed and also the parameters that are the priority. There is no superior algorithm in all measurement strategies. Currently, the implementation of SDN is still using a single controller architecture. Subsequent research will explore a larger scale of SDN implementation in the distributed data plane and control plane layers. Future studies consider scenarios such as the number of controllers, traffic variations and more advanced load balancing schemes.

References

[1] K. W. Murti, T. A. Riza, and A. Mulyana, "Comparative analysis of load balancing dynamic ratio and server ratio algorithms," in *2020 FORTEI-International Conference on Electrical Engineering (FORTEI-ICEE)*. IEEE, 2020, pp. 162–167.

[2] K. Rishabh, K. Angadi, K. Chegu, D. Harikrishna, and S. Ramya, "Analysis of load balancing algorithm in software defined networking," in *2017 2nd International Conference on Computational Systems and Information Technology for Sustainable Solution (CSITSS)*. IEEE, 2017, pp. 1–4.

[3] M. A. Shahid, N. Islam, M. M. Alam, M. M. Su'ud, and S. Musa, "A comprehensive study of load balancing approaches in the cloud computing environment and a novel fault tolerance approach," *IEEE Access*, vol. 8, pp. 130 500–130 526, 2020.

[4] R. Tulloh, J. G. A. Ginting, A. Mulyana, and M. Lutfi, "Performance comparison of file transfer protocol service between link state and distance vector routing protocol in software defined network," in *IOP Conference Series: Materials Science and Engineering*, vol. 982, no. 1. IOP Publishing, 2020, p. 012026.

[5] H. Kar and G. Rather, "Multilayer software defined networking architecture for the internet of things," *International Journal of Computing and Digital Systems*, vol. 9, no. 4, pp. 735–745, 2020.

[6] H. Tussyadiah, R. M. Negara, and D. D. Sanjoyo, "Distributed gateway-based load balancing in software defined network," *Telkomnika*, vol. 18, no. 5, pp. 2352–2361, 2020.

[7] D. B. Rawat and S. R. Reddy, "Software defined networking architecture, security and energy efficiency: A survey," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 1, pp. 325–346, 2016.

[8] O. Marzuqi, A. Virgono, and R. M. Negara, "Implementation model architecture software defined network using raspberry pi: a review paper," *Telkomnika*, vol. 17, no. 3, pp. 1136–1141, 2019.

[9] R. Dewanto, R. Munadi, and R. M. Negara, "Improved load balancing on software defined network-based equal cost multipath routing in data center network," *Jurnal Infotel*, vol. 10, no. 3, pp. 157–162, 2018.

[10] H. T. Zaw and A. Maw, "Traffic management with elephant flow detection in software defined networks (sdn)," *International Journal of Electrical and Computer Engineering*, vol. 9, no. 4, p. 3203, 2019.

[11] S. Ejaz, Z. Iqbal, P. A. Shah, B. H. Bukhari, A. Ali, and F. Aadil, "Traffic load balancing using software defined networking (sdn) controller as virtualized network function," *IEEE Access*, vol. 7, pp. 46 646–46 658, 2019.

[12] K. A. Jadhav, M. M. Mulla, and D. Narayan, "An efficient load balancing mechanism in software defined networks," in *2020 12th International Conference on Computational Intelligence and Communication Networks (CICN)*. IEEE, 2020, pp. 116–122.

[13] R. D. Devapriya and S. I. Gandhi, "Enhanced load balancing and qos provisioning algorithm for a software defined network," in *2020 International Conference on Emerging Trends in Information Technology and Engineering (ic-ETITE)*. IEEE, 2020, pp. 1–5.

[14] K. Hikichi, T. Soumiya, and A. Yamada, "Dynamic application load balancing in distributed sdn controller," in *2016 18th Asia-Pacific Network Operations and Management Symposium (APNOMS)*. IEEE, 2016, pp. 1–6.

[15] A. A. Alsulami, Q. A. Al-Haija, M. I. Thanoon, and Q. Mao, "Performance evaluation of dynamic round robin algorithms for cpu scheduling," in *2019 SoutheastCon*. IEEE, 2019, pp. 1–5.

[16] L. Zhu, J. Cui, and G. Xiong, "Improved dynamic load balancing algorithm based on least-connection scheduling," in *2018 IEEE 4th Information Technology and Mechatronics Engineering Conference (ITOEC)*. IEEE, 2018, pp. 1858–1862.

[17] A. B. Prasetijo, E. D. Widianto, and E. T. Hidayatullah, "Performance comparisons of web server load balancing algorithms on haproxy and heartbeat," in *2016 3rd International Conference on Information Technology, Computer, and Electrical Engineering (ICITACEE)*. IEEE, 2016, pp. 393–396.

[18] F. Chahlaoui and H. Dahmouni, "A taxonomy of load balancing mechanisms in centralized and distributed sdn architectures," *SN Computer Science*, vol. 1, no. 5, pp. 1–16, 2020.

[19] M.-L. Chiang, H.-S. Cheng, H.-Y. Liu, and C.-Y. Chiang, "Sdn-based server clusters with dynamic load balancing and performance improvement," *Cluster Computing*, vol. 24, pp. 537–558, 2021.

[20] R. C. Kurniawan, R. Tulloh, and I. D. Irawati, "Vpls on software defined network using onos controller based on raspberry-pi 3," in *2021 IEEE Asia Pacific Conference on Wireless and Mobile (APWiMob)*. IEEE, 2021, pp. 19–24.

[21] A. H. Eljack, A. H. M. Hassan, and H. H. Elamin, "Performance analysis of onos and floodlight sdn controllers based on tcp and udp traffic," in *2019 International Conference on Computer, Control, Electrical, and Electronics Engineering (ICCCEEE)*. IEEE, 2019, pp. 1–6.

Rohmat Tulloh received B.Eng. in Telecommunication Engineering in 2008 and M.Eng. Electrical Engineering in 2012 from Telkom University Indonesia. Currently, he works as a lecturer at Telkom University, Faculty of Applied Sciences, in the Diploma of Telecommunication Technology study program. He is an instructor at the CISCO network academy and Huawei talent, especially in the field of computer networks. His current research interests include Software Defined Networks, adaptive network and Internet of Things.

Muhammad Iqbal received B.Eng. Telecommunications Engineering and M.Eng. Electrical Engineering from Telkom University Indonesia. Currently, he works as a lecturer at Telkom University, Faculty of Applied Sciences, in the Diploma of Telecommunication Technology study program. He is an instructor at the CISCO network academy and Huawei talent, especially in the field of computer networks. His current research interests include Telecommunications Networks and Network Security.

Mohammad Rifki Baihaqi received a diploma in telecommunications engineering from School of Applied Science, Telkom University Indonesia in 2020. Now he is pursuing a bachelor's degree in telecommunications engineering at the same University and currently researching software defined networks and the Internet of Things. His current research is on load balancing analysis using a programmable data plane on a software-defined network.

Aditya Bayu Aji Pamungkas received a diploma in telecommunications engineering from School of Applied Science Telkom University Indonesia in 2020. His research interests include Software Defined Network, Internet of Things, Big Data, and cloud computing. Currently, he works in the field of telecommunication networks, which focuses on troubleshooting and monitoring networks.