JOIV : Int. J. Inform. Visualization, 8(2) - May 2024 882-890

INTERNATIONAL JOURNAL ON INFORMATICS VISUALIZATION

journal homepage : www.joiv.org/index.php/joiv

Performance Comparison of Zevenet Multi Service Load Balancing with Least Connection and Round Robin Algorithm

Windiya Ma'arifah [a,*], Sarmini [a]

Information System, Universitas Amikom Purwokerto, Purwokerto Utara, Banyumas, 53127, Indonesia

Corresponding author: maarifahwindiya@gmail.com

Abstract— **Amikom Purwokerto University concentrates on Technology and Digital Business. This requires technology to be utilized optimally. The use of technology, especially internships, will make various jobs easier. KRS online is taking lecture schedules online via the AMIKOM Purwokerto Student website. There are several problems with the web server that arise due to the increasing need for information access, which causes the data traffic load to increase. Increased data traffic causes workload overload, resulting in server downtime. Experimental methods were used in this research to look for the causes of the web server's downtime. Then, implement the technology. The purpose is to evaluate the Zevenet load balancer performances by comparing the round-robin and least-connection algorithms. The decision is which algorithm will be used best to implement the Zevenet Load balancer to achieve a more efficient backend server traffic cluster distribution. The TIPHON standard Quality of Service parameters used in Zevenet Load Balancer performance testing are throughput, delay, jitter, packet loss, and CPU usage. The quality-of-service parameter test results show that the Zevenet Load Balancer with the round-robin algorithm has superior performance and shows less CPU usage. It is concluded that using the round-robin algorithm in implementing the Zevenet load balancer to overcome the problem of data traffic load sharing and minimize server downtime on the Student Amikom Purwokerto web server is more appropriate and more effective.**

Keywords— **Zevenet; quality of service; THIPON; load balancer; round robin algorithm; least-connection algorithm.**

Manuscript received 11 Jul. 2023; revised 9 Sep. 2023; accepted 16 Nov. 2023. Date of publication 31 May 2024.

I. Introduction

Work can be completed effectively and efficiently with the development of technology, especially the internet [1], [2]. Amikom Purwokerto University has taken full advantage of technological developments using the Amikom Purwokerto student website. KRS online is a pre-academic activity to take class schedules online through the student website [3], [4]. In online KRS activities, the demand for information on the internet causes the data traffic load on the student web server to increase [5]. An increase in the load (request) on the student website causes an overload of work (request), so that the server experiences a down (overload) [6], [7]. The load balancing method in implementing web server clustering can optimize the flow of traffic and requests on each web server [8], [9]. This study requires the least number of connections for load balancing, which uses the round-robin scheduling technique. The round-robin method creates a queue of tasks for scheduling [10]–[12]. Meanwhile, the Least-connection algorithm directs network connections from active servers with few connections [13], [14]. The different types of services that use the two algorithms impact the differences in turnaround time and the distribution of server workload [15].

Zevenet (Zen Load Balancer) is a multi-service load balancing based on Lx4NAT, HTTP, and HTTPS [12], [16]. Zevenet open-source project is the best solution for load balancing TCP (Transmission Control Protocol) and UDP (User Data Protocol), which can provide high availability of distributed systems [17], [18]. To measure the level of efficiency of Zevenet load balancing, quality of service (QOS) calculation is needed, especially throughput, delay, packet loss, and jitter, to provide accurate results as a determinant of the characteristics of a dark connection in a network [8], [19].

Research by [20] shows that a round-robin algorithm is more efficient. The CPU load balancer graph illustrates this. Using the round-robin algorithm, the average CPU load value is slightly lower, and the throughput is higher than when using the least-connection algorithm, which yields average CPU load values of 0.1%, 0.25%, and 1.15% for average durations of one minute, five minutes, and fifteen minutes, respectively. The average throughput value is 14.74 kbps. In another study,

[16] results show that Haproxy shows better work test results on the response time parameter.

Meanwhile, Zevenet shows that the value of performance testing results is superior to the CPU resource utilization parameter. From these two studies, information was obtained that the round-robin algorithm is more efficient than the last connection algorithm. Then, Zevenet's multi-service load balancing is superior in CPU resource utilization parameters than Haproxy. Therefore, conducting a comparative study of Zevenet Multi Service Load Balancing with Least Connection and Round Robin Algorithm is interesting.

Based on the description above, this study uses Apache Jmeter software to generate requests and measure performance on a web server [21]–[23]. The QOS parameters used in testing and assessing the performance comparison of the HTTP Zevenet load balancer service in this study are throughput, delay, jitter, packet loss, and CPU utilization [24]–[27]. This study aims to determine the round-robin or least-connection algorithm applied to the Zevenet load balancer to obtain a more efficient distribution of traffic cluster server backend student Amikom Purwokerto websites.

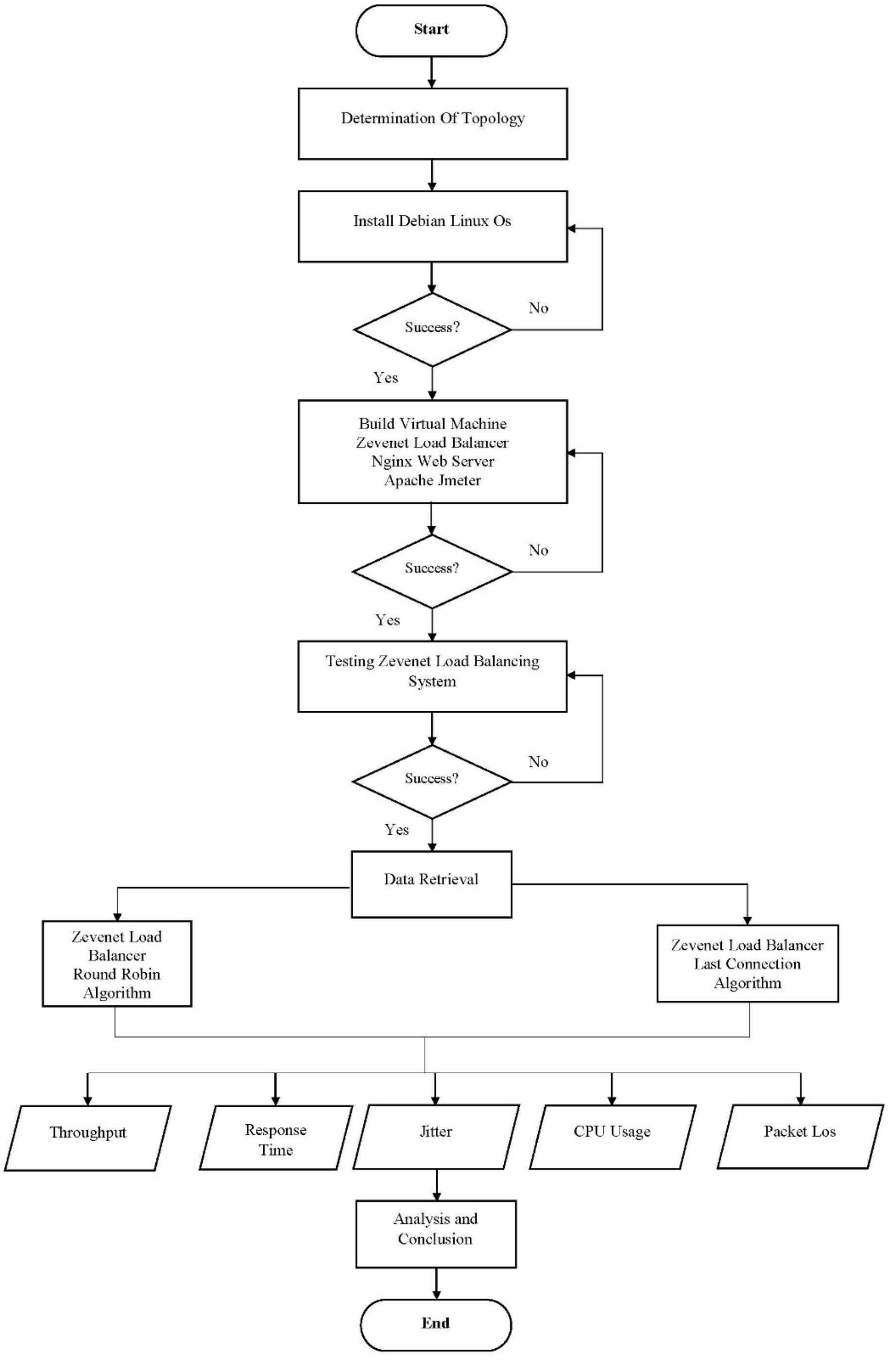

Fig. 1 Flowchart of Research Methods

II. MATERIALS AND METHOD

This study uses the open-source Zevenet (Zen Load Balancer) as a multi-service load balancer. A server cluster of 3 servers is used as a backend server. Nginx is an open-source web server software that functions as an HTTP web server used as a cluster web server [28], [29]. Debian Linux is used to configure the Zevenet load balancer and Nginx web server. Apache JMeter is used to measure the performance of each web server cluster. Each stage in this study is discussed using an algorithm in the form of a flowchart shown in Figure 1.

The research phase begins with topology design, Debian Linux configuration, Nginx web server configuration, and Zevenet load balancer configuration. Then, calculate the quality-of-service parameters on each backend server with the algorithm (round-robin and least-connection) and compare them to determine the most efficient algorithm [30]. Throughput, Delay, Jitter, Packet Loss, and CPU Utilization are the quality of service test parameters calculated [31]–[34]. The bandwidth used on each backend web server cluster is 1Gbps utilizing an internet connection.

A. Zevenet Load Balancer Topology Design Scheme

This study's scheme for designing a network topology for the Zevenet load balancer system uses a Wide Area Network (WAN) simulation. The Zevenet load balancer system network topology combines two network models: a private (local) network and a public network. On a private (local) network, there is a cluster with three web servers as backend servers and a Zevenet load balancer. On server 1, server 2, and server 3, the nginx web server is running. Zevenet load balancer acts as a recipient of requests from clients. Based on the load balancing method, the least-connection algorithm, and the round-robin algorithm, it assigns each request to each backend server load balancer system network topology Figure 2.

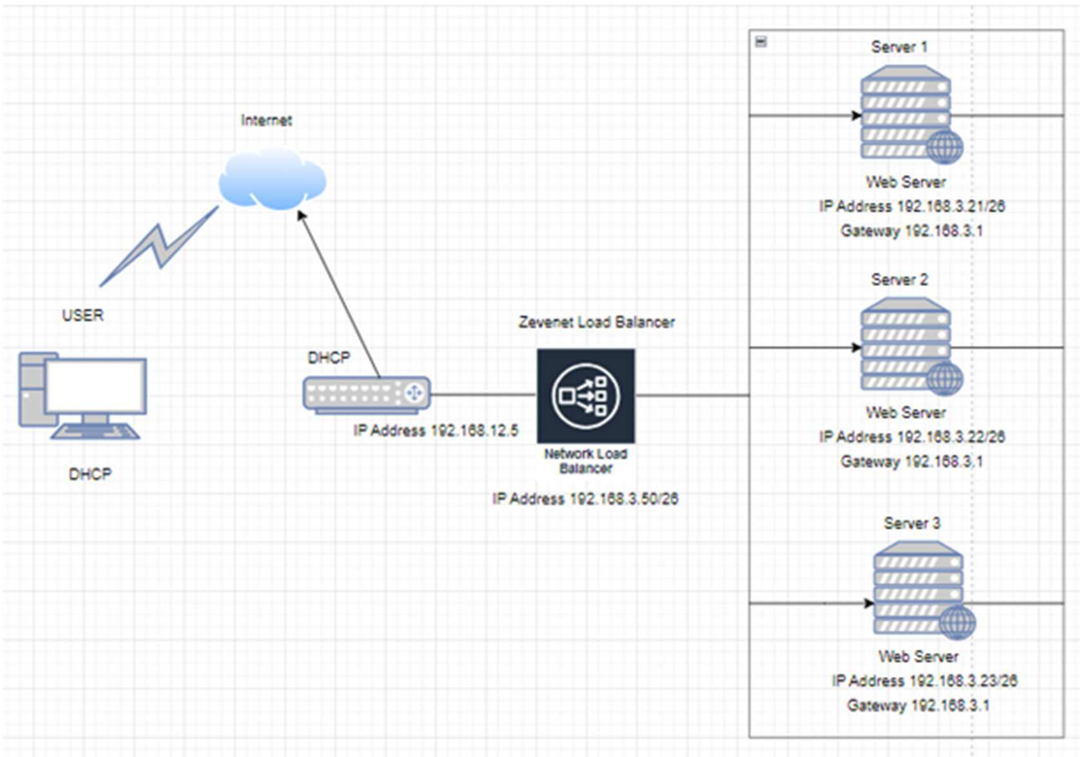

Fig. 2 Zevenet Load Balancing System Network Topology

B. Scenario Simulation and Service Testing Schemes

The characteristics examined in this study are connected to the run simulation exercises. Each trial will use distinct attribute values as a yardstick for evaluating how well Zevenet multi-service load balancing handles HTTP request quality. Simulation attributes for HTTP services Table 1. As shown by the scheme in Table 1, Differences in load variations on HTTP services are used to test the performance of Zevenet load balancing services. The simulation time is 5 minutes, and the bandwidth allocated is 1 Gbps in five trial simulations with round-robin load-sharing algorithm (RR3:2:1) (RR4:2:1) (RR5:2:1) (RR5:31) (RR 5:4:1) and least-connection algorithm (ALC 3:2:1) (ALC4:2:1) (ALC 5:2:1) (ALC5:31) (ALC 5:4:1).

TABLE I
SIMULATION ATTRIBUTES FOR HTTP SERVICES

Attribute	Service and Value		
	Low Load HTTP	**Medium Load HTTP**	**High Load HTTP**
Service Type	1000 byte	2500 byte	5000 bytes
Simulation Area	WAN (Wide Area Network)		
Simulation Time	5 Minutes		
Bandwidth	1 Gbps		
Trial Simulation	5 Trial		
Load Sharing	(RR/ALC3:2:1) (RR/ALC4:2:1) (RR/ALC 5:2:1) (RR/ALC5:31) (RR/ALC 5:4:1)		

C. Index Parameters: QOS from TIPHON

In the context of networks, quality of service is the capacity to offer customized services for various types of network traffic [35]–[37]. Better, planned network services with allocated, regulated bandwidth delay and increased characteristic loss are the ultimate goals of quality of service [38]–[41]. The ability to ensure the transmission of crucial data is another way to define quality of service; put another way, it is a set of standards that establish how satisfied a customer is with a given service [42]–[44]. The extent of quality TIPHON's tables are used to adapt the network to Quality-of-Service requirements [45]. The network quality level categories for QOS based on TIPHON are grouped into Very Good, Good, Not Good, and Bad [45], Index Parameters Quality of Service (QOS) from TIPHON Table 2.

TABLE II
INDEX PARAMETER QOS FROM TIPHON

Value	Presentation (%)	Index
3.8-4	95-100	Very Good
3-3.79	75-94.75	Good
2-2.99	50-74.75	Not Good
1-1.99	25-49.75	Bad

D. Attributes to Measure Service Quality

There are four quality of service characteristics that need to be taken into account when determining the quality of HTTP services. Next, based on the TIPHON standard, the output results of the four Qos (Quality of Service) parameters will be categorized [46], [47]. The four quality of service parameters of service quality observed and accompanied by the TIPHON standardization table can be seen below:

1) Throughput: Throughput is calculated by dividing the total number of incoming packets received over a certain period by the length of that period [48]–[50]. Equation 1 is the formula for calculating throughput. TIPHON standard values for throughput [27] as shown in Table 3.

$$Throughput = \frac{Amount\ of\ data\ sent}{Time\ of\ data\ transmission} \quad (1)$$

TABLE III
TIPHON THROUGHPUT STANDARDIZATION

Category	Throughput	Index
Excellent	>2,1 Mbps	4
Good	1200 kbps- 2,1 Mbps	3
Fair	700- 1200 kbps	2
Poor	338-1200 kbps	1
Bad	<338 kbps	0

2) Delay: There is a delay when a packet travels across a network from sender to recipient [51]. The delay calculation formula is found in Equation 2. TIPHON standard values for delay [45] are shown in Table 4.

$$Delay\ (ms) = \frac{Total\ Delay}{Total\ packets\ received} \quad (2)$$

TABLE IV
TIPHON DELAY STANDARDIZATION

Category of Latency	Delay	Index
Best	<150 ms	4
High	150 < 250 ms	3
Medium	250 ms <350 ms	2
Low	<450 ms	1

3) Jitter: Jitter, often known as the discrepancy in packet arrival times at the destination terminal, is a variation of delay [48], [52]. Equation 3 is the formula for calculating jitters. TIPHON standard values for jitter [37] as shown in Table 5.

$$Jitter\ (ms) = \frac{Total\ delay\ variations}{Total\ packets\ received} - 1 \quad (3)$$

TABLE V
TIPHON JITTER STANDARDIZATION

Category	Jitter (ms)	Index
Very Good	0 ms	4
Good	1-75 ms	3
Normal	75-125 ms	2
Bad	125-255 ms	1

4) Packet Loss: The parameter packet loss describes a condition that indicates the entire packet loss in data transfers that take place. Packet loss is a result of network congestion and collisions [53]. Equation 4 is the formula for calculating packet loss. TIPHON standard value for packet loss [45] as demonstrated in Table 6.

$$\frac{Sent\ data\ packets - Received\ data\ packets}{Sent\ data\ packets} x\ 100 \quad (4)$$

TABLE VI
STANDARDIZATION OF TIPHON PACKET LOSS

Degradation Of Packet Lost	Packet Loss	Index
Perfect	0 %	4
Good	3%	3
Medium	15%	2
Poor	25%	1

5) CPU Utilization: A PC client, or user, tests the two methods by sending simultaneous HTTP requests from the client PC to the Zevenet load balancer server. The final incident data on the CPU and network traffic graphs is the data that was retrieved later [54], [55].

III. RESULTS AND DISCUSSION

A. Results of Comparison of QoS throughput parameters

The tables and graphs of the Zevenet load balancer test results compare the quality-of-service parameter throughput on the round-robin and least-connection methods. Table 7 and Figure 3 show the outcomes of the round-robin algorithm's performance test. Meanwhile, Table 8 and Figure 4 show the results of testing the least-connection algorithm.

TABLE VII
THROUGHPUT ZEVENET ROUND-ROBIN ALGORITHM

		Throughput (mbit/s)		
Test	Scenario	1000 byte	2500 byte	5000 bytes
1	RR 3:2:1	0.9	1.7	4
2	RR 4:2:1	0.6	1.2	3.8
3	RR 5:2:1	0.5	1.2	3.5
4	RR 5:3:1	0.3	1	2.8
5	RR 5:4:1	0.3	0.9	2.1

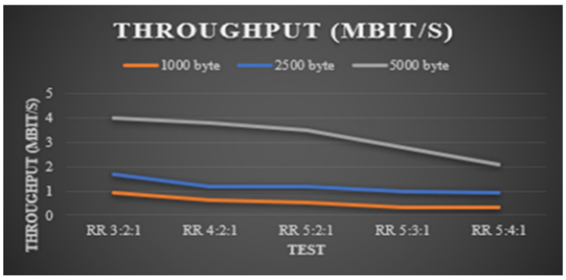

Fig. 3 Throughput Zevenet Round-Robin Algorithm

Compared to alternative weight distribution schemes, the Zevenet load balancing system with the round-robin algorithm on the RR3:2:1 weight distribution scheme has the most significant average throughput value, at 2.2 Mbps. The 3:2:1 RR weighted round-robin algorithm can balance the performance of the three backend web servers to serve user requests optimally.

TABLE VIII
THROUGHPUT ZEVENET LEAST-CONNECTION ALGORITHM

		Throughput (mbit/s)		
Test	Scenario	1000 byte	2500 byte	5000 bytes
1	ALC 3:2:1	0.1	1	1.5
2	ALC 4:2:1	0.3	1.2	1.4
3	ALC 5:2:1	0.2	1.1	1.3
4	ALC 5:3:1	0.5	1.3	1.5
5	ALC 5:4:1	0.8	1.7	3

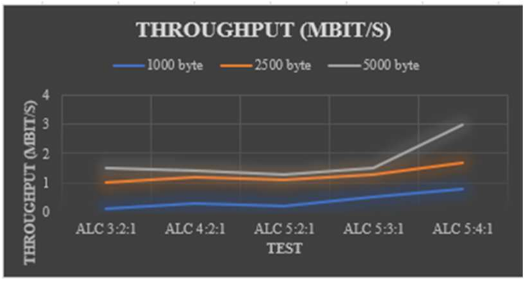

Fig. 4 Throughput Zevenet Least Connection Algorithm

Among the various weight distribution schemes, the Zevenet load balancing system with the least-connection algorithm in the ALC5:4:1 weight distribution scheme has the most significant average throughput value, 1.83 Mbps. The least-connection algorithm with the ALC5:4:1 weight can balance the performance of the three backend web servers to serve user requests optimally.

The round-robin algorithm has an average value of 2.2 Mbps, while the least-connected algorithm has an average value of 1.8 Mbps, according to the throughput computation for each algorithm. Thus, it can be said that both fall into the "Excellent" and "Good" categories for the round-robin method and the least-connection approach, respectively, according to TIPHON's standardized throughput [27].

B. Comparison Results of QoS Parameter delay

The tables and graphs show the outcomes of evaluating the Zevenet load balancer system's performance and compare the Quality-of-Service delay parameter in the round-robin and least-connection algorithms. The average throughput value and the average delay value are correlated; the more significant the resulting throughput value, the lower the average delay value [56], [57]. Table 9 and Figure 5 show the outcomes of evaluating the round-robin algorithm's performance.

TABLE IX
DELAY ZEVENET ROUND-ROBIN ALGORITHM

		Delay (ms)		
Test	**Scenario**	**1000 byte**	**2500 byte**	**5000 bytes**
1	RR 3:2:1	10	40	20
2	RR 4:2:1	40	90	50
3	RR 5:2:1	50	90	80
4	RR 5:3:1	70	80	210
5	RR 5:4:1	70	110	290

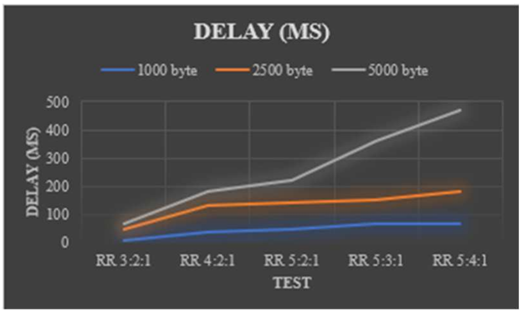

Fig. 5 Delay Zevenet Round-Robin Algorithm

In the meantime, Table 10 and Figure 6 display the outcomes for assessing the least-connection algorithm's performance. With the round-robin algorithm applied to the RR3:2:1 weight division scheme, the Zevenet load balancing system achieves a lower average delay value of 23.3 ms. The RR3:2:1 weight distribution scheme gets the highest throughput value, resulting in a smaller average delay value.

TABLE X
DELAY ZEVENET LEAST CONNECTION ALGORITHM

		Delay (ms)		
Test	Scenario	1000 byte	2500 byte	5000 byte
1	ALC 3:2:1	90	150	150
2	ALC 4:2:1	70	80	170
3	ALC 5:2:1	80	90	160
4	ALC 5:3:1	50	70	150
5	ALC 5:4:1	20	40	60

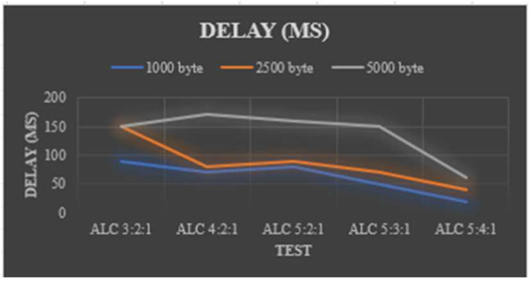

Fig. 6 Delay Zevenet Least-Connection Algorithm

With the least-connection method in the ALC5:4:1 weight division scheme, the Zevenet load balancing system achieves a lower average delay value of 40 ms. The ALC5:4:1 weight distribution scheme gets the highest throughput value, resulting in a minor average delay.

Based on the outcomes of these computations, the round-robin algorithm's delay value is 23.3 ms, while the least-connected algorithm's delay value is 40 ms. Based on the TIPHON standardization [45], the test results are included in the "Best" category. Nonetheless, the round-robin approach has the minimum delay value when comparing the delay values of the least-connection and round-robin algorithms. Therefore, the round-robin algorithm outperforms the least-connected approach in terms of efficiency.

C. Results of Comparison of QoS Jitter Parameters

The tables and graphs of the results of assessing the Zevenet load balancer system's performance show a comparison of service jitter parameters' quality in the round-robin and least-connection algorithms. The results of measuring the performance of the round-robin algorithm can be seen in Table 11 and Figure 7. Meanwhile, the results of measuring the performance of the least-connection algorithm can be seen in Table 12 and Figure 8.

TABLE XI
JITTER ZEVENET ROUND-ROBIN ALGORITHM

		Jitter (ms)		
Test	Scenario	1000 byte	2500 byte	5000 bytes
1	RR 3:2:1	5	20	10
2	RR 4:2:1	20	45	25
3	RR 5:2:1	40	60	50
4	RR 5:3:1	40	50	140
5	RR 5:4:1	40	80	90

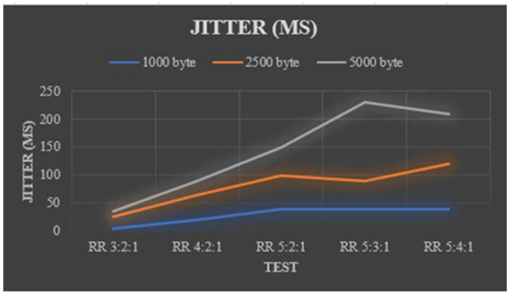

Fig. 7 Jitter Zevenet Round-Robin Algorithm

The round-robin algorithm on the RR3:2:1 weight distribution scheme yields the minimum average jitter value of 11.6 ms for the Zevenet load balancing system.

TABLE XII
JITTER ZEVENET LEAST-CONNECTION ALGORITHM

		Jitter (ms)		
Test	**Scenario**	**1000 byte**	**2500 byte**	**5000 byte**
1	ALC 3:2:1	50	130	140
2	ALC 4:2:1	30	50	120
3	ALC 5:2:1	80	90	160
4	ALC 5:3:1	40	70	150
5	ALC 5:4:1	15	35	50

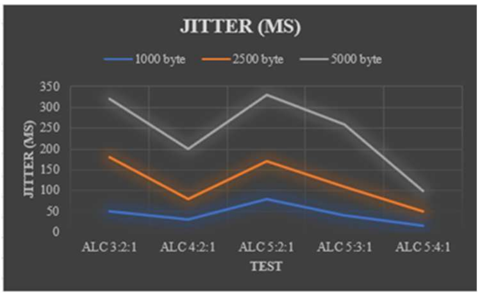

Fig. 8 Jitter Zevenet Least Connection Algorithm

The Zevenet load balancing system achieves the smallest average jitter value of 33.3 ms by employing the least-connection algorithm in the ALC5:4:1 weight distribution scheme. According to the TIPHON standards [37], the round-robin and least-connection algorithms' Jitter computation results fall into the "Good" category, as they remain within the 1-75 ms range. The round-robin algorithm combined with Zevenet load balancing is more efficient than the least-connection algorithm.

D. *Results of Comparison of Qos Parameters for Loss Packets*

The tables and graphs of the results of testing the performance of the Zevenet load balancer system show the comparison of the quality-of-service parameter Packet loss in the round-robin and least-connection algorithms. The results of measuring the performance of the round-robin algorithm can be seen in Table 13 and Figure 9. In the meantime, Table 14 and Figure 10 display the outcomes of evaluating the least-connection algorithm's performance.

TABLE XIII
ZEVENET PACKET LOSS ROUND-ROBIN ALGORITHM

		Packet Loss (%)		
Test	Scenario	1000 byte	2500 byte	5000 byte
1	RR 3:2:1	0,0	0,0	0,0
2	RR 4:2:1	0,1	0,2	0,8
3	RR 5:2:1	0,2	0,5	0,9
4	RR 5:3:1	0,3	0,7	1
5	RR 5:4:1	0,3	4	5

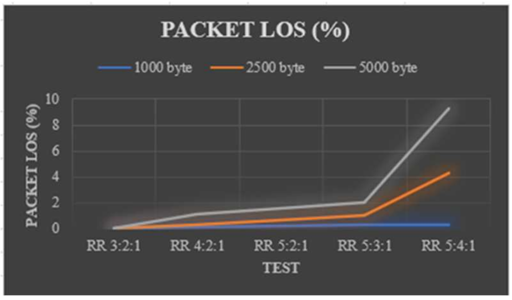

Fig. 9 Zevenet Packet Loss Round-Robin Algorithm

Using the round-robin algorithm on the RR3:2:1 weight distribution scheme, the feasibility value of the Zevenet load balancing system packet loss parameter yields the minimum average value of 0, indicating the " Perfect " category based on the TIPHON standard [45]. The data packets retrieved show this: in the RR3:2:1 scheme test, no packets were dropped, and the final throughput value was more significant than 2 Mbps.

TABLE XIV
PACKET LOSS LEAST-CONNECTION ALGORITHM

		Packet Loss (%)		
Test	Scenario	1000 byte	2500 byte	5000 bytes
1	ALC 3:2:1	0,9	8	6
2	ALC 4:2:1	0,7	7	7
3	ALC 5:2:1	0,8	6	7
4	ALC 5:3:1	0,8	7	5
5	ALC 5:4:1	0,2	2	3

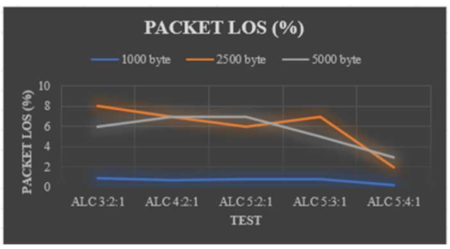

Fig. 10 Packet Loss Least Connection Algorithm

The feasibility value of the Zevenet load balancing system packet loss parameter using the least-connection algorithm in the ALC5:4:1 weight distribution scheme gets the smallest average value of 1.7, which indicates a "Good" category based on the TIPHON standard [45]. Thus, using the round-robin method rather than the least-connection algorithm is more efficient when utilizing the Zevenet load balancing system.

E. Comparison of CPU Utilization Calculation Results

The tables and graphs showing the results of the Zevenet load balancer performance test show the CPU Utilization comparison for the round-robin and least-connection methods. Table 15 and Figure 11 show the outcomes of the round-robin algorithm's performance test. In the meantime, Table 16 and Figure 12 display the outcomes of the least-connection algorithm's performance test.

TABLE XV
CPU UTILIZATION ZEVENET ROUND-ROBIN ALGORITHM

Resource Utilization (CPU%)				
Test	Scenario	1000 byte	2500 byte	5000 bytes
1	RR 3:2:1	15	22	29
2	RR 4:2:1	16	20	35
3	RR 5:2:1	15	21	33
4	RR 5:3:1	14	22	31
5	RR 5:4:1	14	23	39

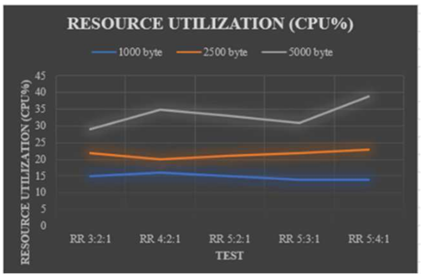

Fig. 11 CPU Utilization Zevenet Round-Robin Algorithm

By analyzing the data in Table 8 and Figure 4 above, it can be concluded that the request distribution process carried out by the Zevenet load balancer on the HTTP service with the round-robin algorithm shows a relatively small average value of request resource utilization because it does not cause overload on the backend server. The average resource utilization (CPU) value of the Zevenet load balancer system using the round-robin algorithm on the RR3:2:1 weight distribution gets a minor average resource utilization (CPU) value of 22.

TABLE XVI
CPU UTILIZATION ZEVENET LEAST-CONNECTION ALGORITHM

Resource Utilization (CPU%)				
Test	Scenario	1000 byte	2500 byte	5000 bytes
1	ALC 3:2:1	38	47	60
2	ALC 4:2:1	34	49	54
3	ALC 5:2:1	38	45	59
4	ALC 5:3:1	49	49	60
5	ALC 5:4:1	40	46	63

The distribution of requests by the Zevenet load balancer on the HTTP service with the least-connection algorithm demonstrates a minor average request utilization compared to the round-robin algorithm on a small or large number of connections, as can be inferred from the data in Table 7 and the graph in Figure 3. The average resource utilization (CPU) value of the Zevenet load balancer system using the least-connection algorithm on the ALC 5:4:1 weight distribution gets a minor average resource utilization (CPU) value of 49.6. As a result, although the CPU use in the least-connection algorithm is more than in the round-robin algorithm, it is still relatively tiny. Table 17 below compares all the estimated quality of service test parameters.

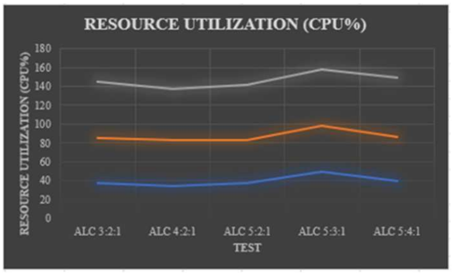

Fig. 12 CPU Utilization Zevenet Least Connection Algorithm

TABLE XVII
COMPARISON OF QOS PARAMETER TESTING RESULTS

Parameter QoS	**Round Robin**	**QoS Category**	**Least Connection**	**QoS Category**
Throughput	2.2 mbps	Excellent	1.83 Mbps	Good
Delay	23.3 ms	Best	40 ms	Best
Jitter	11.6 ms	Good	33.3 ms	Good
Packet Loss	0	Perfect	1.7	Good
CPU Utilization	0.22 %	-	0.49%	-

IV. CONCLUSION

The implementation of Zevenet Load Balancer, which divides the data traffic load of the backend data cluster on the Amikom Purwokerto student web server using the round-robin algorithm and the least-connection algorithm, was suggested as a solution to issues with the student web server. From this research, the results of testing the quality-of-service parameters on the Amikom Purwokerto student HTTP web server service show that the Zevenet Load Balancer with the round-robin algorithm has superior performance and shows less CPU usage. Using the round-robin algorithm in implementing the Zevenet load balancer to overcome the problem of load-sharing data traffic and minimize server downtime on the Student Amikom Purwokerto web server is more appropriate and effective.

The information technology leadership at Amikom University Purwokerto will greatly enhance information technology governance and information technology service management (ITSM) by putting this idea into practice. Moreover, this implementation will bring more benefits in improving information systems, information technology, and internet networks, especially the stability of the Amikom Purwokerto student web server.

In future research, the author plans to develop a network monitoring or Network Monitoring Service that is useful for informing web servers of uptime and downtime, making it easier to resolve web server network problems. Then, carry out research to improve network security on the Amikom Purwokerto student web server.

ACKNOWLEDGMENT

The authors thank Amikom Purwokerto University for supporting this research.

REFERENCES

[1] T. A. Cahyanto, R. D. Antoko, T. T. Warisaji, S. Santosa, And R. Rodianto, "Analysis of The Dynamic Source Routing Protocol on The Performance Of File Transfer Protocol And Video Conference Services In The Mobile Adhoc Network Simulation," *Ilkom Jurnal Ilmiah*, Vol. 15, No. 1, Pp. 165–174, Apr. 2023, doi:10.33096/Ilkom.V15i1.1526.165-174.

[2] N. F. Aisyah and M. A. F. Ridha, "High Availability Service Dengan Multiple Master Pada Kubernetes Cluster Menggunakan Virtualisasi Kernel Based Virtual Machine (Kvm)," In *Proceeding Applied Business And Engineering Conference*, Padang, Nov. 2022, Pp. 17–19.

[3] R. R. Rerung and Y. R. Ramadhan, "Rancang Bangun Sistem Informasi Akademik Dalam Penerapan Smart Campus Untuk Meningkatkan Pelayanan Akademik," JTERA (Jurnal Teknologi Rekayasa), vol. 3, no. 2, p. 191, Dec. 2018, doi:10.31544/jtera.v3.i2.2018.191-210.

[4] P. Sukmasetya, A. Setiawan, and E. R. Arumi, "Penggunaan Usability Testing Sebagai Metode Evaluasi Website KRS Online pada Perguruan Tinggi," JST (Jurnal Sains dan Teknologi), vol. 9, no. 1, pp. 58–67, Jun. 2020, doi: 10.23887/jstundiksha.v9i1.24691.

[5] L. Chen, J. Lingys, K. Chen, and F. Liu, "AuTO," Proceedings of the 2018 Conference of the ACM Special Interest Group on Data Communication, Aug. 2018, doi: 10.1145/3230543.3230551.

[6] J. Kong, J. Kim, And J. Bae, "Hifi-Gan: Generative Adversarial Networks For Efficient And High Fidelity Speech Synthesis," In *34th Conference On Neural Information Processing Systems (Neurips 2020), Vancouver, Canada.*, 2020, Pp. 1–12. [Online]. Available: https://Jik876.Github.Io/Hifi-Gan-Demo/

[7] S. Páll et al., "Heterogeneous parallelization and acceleration of molecular dynamics simulations in GROMACS," The Journal of Chemical Physics, vol. 153, no. 13, Oct. 2020, doi: 10.1063/5.0018516.

[8] E. Prasetyo, A. Hamzah, And E. Sutanta, "Analisa Quality Of Service (Qos) Kinerja Point To Point Protocol Over Ethernet (Pppoe) Dan Point To Point Tunneling Protocol (Pptp)," *Jurnal Jarkom*, Vol. 4, No. 1, Pp. 29–37, Dec. 2016.

[9] S. Syamsu, "Implementasi Cluster Database Berbasis Mysql Dan Haproxy Sebagai Pembagi Beban Kerja Server," *Jurnal Teknologi Informasi Dan Komunikasi*, Vol. 8, No. 1, Pp. 48–58, Jun. 2018.

[10] K. V, "A Stochastic Development of Cloud Computing Based Task Scheduling Algorithm," September 2019, vol. 2019, no. 1, pp. 41–48, Sep. 2019, doi: 10.36548/jscp.2019.1.005.

[11] A. Et al., "Developing Load Balancing for IoT - Cloud Computing Based on Advanced Firefly and Weighted Round Robin Algorithms," Baghdad Science Journal, vol. 16, no. 1, p. 0130, Mar. 2019, doi:10.21123/bsj.2019.16.1.0130.

[12] J. H, Y. Fitrisia, And M. A. F. Ridha, "Implementasi Multi Server Data Storage Pada Cloud Computing," *Jurnal Komputer Terapan*, Vol. 4, No. 2, Pp. 1–9, Nov. 2018.

[13] E. Rohadi, A. Prastyo, And M. F. Rahmat, "Implementasi Klaster Komputer Mini Raspberry Pi Metode Load Balancing Menggunakan Algoritma Round Robin," *Jurnal Informatika Polinema*, Vol. 5, No. 3, Pp. 132–135, May 2019.

[14] A. Solehudin, R. Mayasari, G. Garno, and A. Susilo Yuda Irawan, "Perbandingan Algoritma Round Robin dan Algoritma Least Connection pada Haproxy untuk Load Balancing Web Server," Systematics, vol. 2, no. 1, p. 21, Apr. 2020, doi:10.35706/sys.v2i1.3634.

[15] N. Mukhtar, "Herbs Variation at Different Canal Bank Of Tehsil Depalpur," *The International Journal Of Biological Research*, Vol. 4, No. 1, Pp. 1–5, 2021.

[16] F. M. Azmi, M. Data, And H. Nurwasito, "Perbandingan Kinerja Haproxy Dan Zevenet Dalam Pengimplementasian Multi Service Load Balancing," *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer*, Vol. 3, No. 1, Pp. 253–260, Jan. 2019.

[17] N. Samsudin and A. Ahmad, "A Study Of Internet Protocol Digital Microwave Radio (Ip Dmr) Propagation Delay Affected Downtime In Oil And Gas Company," *International Journal Of Advanced Computing Science And Engineering*, Vol. 1, No. 3, Pp. 133–141, 2019.

[18] A. A. Asril, P. Maria, Y. Antonisfia, And R. Hadi, "Design Attention System Of Single Mode Aerial Fiber Optic Cable Transmission On Connection Loss On Passive Splitter," *International Journal Of Advanced Science Computing And Engineering*, Vol. 5, No. 2, Pp. 95–107, 2023.

[19] S. S. Mohamed, A.-F. I. Abdel-Fatah, and M. A. Mohamed, "Performance evaluation of MANET routing protocols based on QoS and energy parameters," International Journal of Electrical and Computer Engineering (IJECE), vol. 10, no. 4, p. 3635, Aug. 2020, doi:10.11591/ijece.v10i4.pp3635-3642.

[20] H. Triangga, I. Faisal, and I. Lubis, "Analisis Perbandingan Algoritma Static Round-Robin dengan Least-Connection Terhadap Efisiensi Load Balancing pada Load Balancer Haproxy," InfoTekJar (Jurnal Nasional Informatika dan Teknologi Jaringan), vol. 4, no. 1, pp. 70–75, Sep. 2019, doi: 10.30743/infotekjar.v4i1.1688.

[21] B. Busran and A. Ridwan, "Analisis Perbandingan Performa Apache Web Server Dan Nginx Menggunakan Apache Jmeter," *Jurnal Teknoif Teknik Informatika Institut Teknologi Padang*, Vol. 8, No. 2, Pp. 87–92, 2020, doi: 10.21063/Jtif.2020.V8.2.87-92.

[22] D. I. Permatasari *Et Al.*, "Pengujian Aplikasi Menggunakan Metode Load Testing Dengan Apache Jmeter Pada Sistem Informasi Pertanian," *Jurnal Sistem Dan Teknologi Informasi*, Vol. 8, No. 1, Pp. 135–139, 2020.

[23] K. Kamarudin, K. Kusrini, And A. Suyoto, "Uji Kinerja Sistem Web Service Pembayaran Mahasiswa Menggunakan Apache Jmeter (Studi Kasus: Universitas Amikom Yogyakarta)," *Jurnal Teknologi Informasi*, Vol. 8, Pp. 44–52, 2018.

[24] D. Biswas and Md. Samsuddoha, "Determining Proficient Time Quantum to Improve the Performance of Round Robin Scheduling Algorithm," International Journal of Modern Education and Computer Science, vol. 11, no. 10, pp. 33–40, Oct. 2019, doi:10.5815/ijmecs.2019.10.04.

[25] Pamungkas, Sumbogo W., et al. "Analisis Quality of Service (QoS) Pada Jaringan Hotspot SMA Negeri XYZ." Jurnal Sistem Informasi dan Teknologi Informasi, vol. 7, no. 2, 2018, pp. 142-152, doi:10.36774/jusiti.v7i2.249.

[26] A. Mikola and M. Sari, "Analisis Sistem Jaringan Berbasis QoS untuk Hot-Spot Di Institut Shanti Bhuana," Journal of Information Technology, vol. 2, no. 1, pp. 31–35, Mar. 2022, doi: 10.46229/jifotech.v2i1.398.

[27] M. Purwahid and J. Triloka, "Analisis Quality Of Service (Qos) Jaringan Internet Untuk Mendukung Rencana Strategis Infrastruktur Jaringan Komputer Di Smk N I Sukadana," *Jtksi*, Vol. 2, No. 3, Pp. 100–109, 2019.

[28] R. Riska and H. Alamsyah, "Analisa Dan Perancangan Load Balancing Web Server Mengunakan Haproxy Analysis And Design Of Web Server Load Balancing Using Haproxy," *Techno.Com*, Vol. 20, No. 4, Pp. 552–565, Nov. 2021.

[29] F. M. Nurzaman, F. Chahyadi, And M. R. Rathomi, "Jurnal Sustainable: Jurnal Hasil Penelitian Dan Industri Terapan Analisis Perbandingan Performa Load Balancer Nginx Dan Haproxy Pada Docker," *Jurnal Sustainable: Jurnal Hasil Penelitian Dan Industri Terapan*, Vol. 11, No. 1, Pp. 16–25, May 2022.

[30] V. Barot, V. Kapadia, and S. Pandya, "QoS Enabled IoT Based Low Cost Air Quality Monitoring System with Power Consumption Optimization," Cybernetics and Information Technologies, vol. 20, no. 2, pp. 122–140, Jun. 2020, doi: 10.2478/cait-2020-0021.

[31] M. Ulfah and A. S. Irtawaty, "Pengukuran Dan Analisa Quality Of Service (Qos) Jaringan Internet Di Gedung Terpadu Politeknik Negeri Balikpapan," *Snitt Politeknik Negeri Balikpapan*, Vol. 2, No. 5, Pp. 351–357, 2020.

[32] M. Mahmud and Y. Aprizal, "The Penerapan QoS (Quality Of Service) Dalam Menganalisis Kualitas Kinerja Jaringan Komputer (Studi Kasus Hotel Maxone Palembang)," Journal of Information System Research (JOSH), vol. 3, no. 4, pp. 374–379, Jul. 2022, doi:10.47065/josh.v3i4.1567.

[33] K. Kurniawan and A. Prihanto, "Analisis Quality Of Service (QoS) Pada Routing Protocol Routing OSPF (Open Short Path First)," Journal of Informatics and Computer Science (JINACS), vol. 3, no. 03, pp. 358–365, Feb. 2022, doi: 10.26740/jinacs.v3n03.p358-365.

[34] M. Rusdan, "Analisis Quality Of Service Qos) Pada Jaringan Wireless (Studi Kasus: Universitas Widyatama)," *Jurnal Sistemik*, Vol. 5, No. 2, Pp. 17–20, 2017.

[35] M. C. Ramadhan, A. Widodo, D. H. Putra, And N. Noviandi, "Natural Sciences Engineering &," *Natural Sciences Engineering & Technology Journal*, Vol. 2, No. 1, Pp. 79–84, 2022.

[36] A. Charisma, A. D. Setiawan, G. M. Rahmatullah, And M. R. Hidayat, "Quality F Service (Qos) N 4g Telkomsel Networks In Soreang," In *2019 Ieee 13th International Conference On Telecommunication Systems, Services, And Applications (Tssa)*, 2019, Pp. 145–148.

[37] S. Budiyanto and D. Gunawan, "Comparative Analysis of VPN Protocols at Layer 2 Focusing on Voice Over Internet Protocol," IEEE Access, vol. 11, pp. 60853–60865, 2023, doi:10.1109/access.2023.3286032.

[38] D. Y. Setiawan, S. Naning Hertiana, and R. M. Negara, "6LoWPAN Performance Analysis of IoT Software-Defined-Network-Based Using Mininet-Io," 2020 IEEE International Conference on Internet of Things and Intelligence System (IoTaIS), Jan. 2021, doi:10.1109/iotais50849.2021.9359714.

[39] M. Yusro, N. S. Azlyn, and S. I. Purnama, "Adapting ISO 17025 to Enrich QoS as Quality Measurement on Internet of Medical Things," 2022 IEEE International Conference on Communication, Networks and Satellite (COMNETSAT), Nov. 2022, doi:10.1109/comnetsat56033.2022.9994345.

[40] E. Darmawan, S. Budiyanto, and L. M. Silalahi, "QoS Analysis on VoIP with VPN using SSL and L2TP IPSec Method," 2022 IEEE International Conference on Communication, Networks and Satellite (COMNETSAT), Nov. 2022, doi:10.1109/comnetsat56033.2022.9994572.

[41] P. Tang, Y. Dong, Y. Chen, S. Mao, and S. Halgamuge, "QoE-Aware Traffic Aggregation Using Preference Logic for Edge Intelligence," IEEE Transactions on Wireless Communications, vol. 20, no. 9, pp. 6093–6106, Sep. 2021, doi: 10.1109/twc.2021.3071745.

[42] D. Iswadi, R. Adriman, and R. Munadi, "Adaptive Switching PCQ-HTB Algorithms for Bandwidth Management in RouterOS," 2019 IEEE International Conference on Cybernetics and Computational Intelligence (CyberneticsCom), Aug. 2019, doi:10.1109/cyberneticscom.2019.8875679.

[43] B. Dwinanto and A. S. Arifin, "Integrated Strategy Framework To Improve Quality Of Network on The BMKG Communication Network System," 2021 IEEE International Conference on Communication, Networks and Satellite (COMNETSAT), Jul. 2021, doi:10.1109/comnetsat53002.2021.9530814.

[44] P. A. Fadhila, M. I. Nashiruddin, and M. A. Nugraha, "Addressing Spectrum Scarcity in Indonesia Dense Urban Market by Using 700 MHz for 4G LTE-Advanced Network Deployment," 2021 IEEE 12th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON), Oct. 2021, doi:10.1109/iemcon53756.2021.9623165.

[45] A. Hafiz, D. Afriansyah, F. K. Ikhsan, B. Suprapto, and I. W. Pratama, "Measuring Quality of Wireless Local Area Network Using Quality of Service Framework," IJISCS (International Journal of Information System and Computer Science), vol. 3, no. 3, p. 90, Dec. 2019, doi:10.56327/ijiscs.v3i3.790.

[46] R. Wulandari, "Analisis Qos (Quality Of Service) Pada Jaringan Internet (Studi Kasus : Upt Loka Uji Teknik Penambangan Jampang Kulon-Lipi)," *Jurnal Teknik Informatika Dan Sistem Informasi*, Vol. 2, No. 2, Pp. 162–172, Aug. 2016.

[47] M. A. A. Putra, I. Fitri, and A. Iskandar, "Implementasi High Availability Cluster Web Server Menggunakan Virtualisasi Container Docker," Jurnal Media Informatika Budidarma, vol. 4, no. 1, p. 9, Jan. 2020, doi: 10.30865/mib.v4i1.1729.

[48] B. Arifwidodo, V. Metayasha, and S. Ikhwan, "Analisis Kinerja Load Balancing pada Server Web Menggunakan Algoritma Weighted Round Robin pada Proxmox VE," Jurnal Telekomunikasi dan Komputer, vol. 11, no. 3, p. 210, Dec. 2021, doi: 10.22441/incomtech.v11i3.11775.

[49] S. Yusnita, L. Markis, W. Tristianti, And T. L. Wijaya, "Analysis on The Effect of Channel Bandwidth Occupying In Lte Frequency Band On Throughput," *International Journal Of Advanced Science Computing And Engineering*, Vol. 5, No. 1, Pp. 8–14, 2023.

[50] D. Chandra, F. Rahmat, S. Aulia, And A. F. Kasmar, "Effect of Modulation On Throughput Of 4g Lte Network Frequency 1800 Mhz," *International Journal Of Advanced Science Computing And Engineering*, Vol. 5, No. 1, Pp. 44–53, 2023.

[51] J. Al Amien and D. Winarso, "Analisis Peningkatan Kinerja Ftp Server Menggunakan Load Balancing Pada Container," *Jurnal Fasilkom*, Vol. 9, No. 3, Pp. 8–18, Nov. 2019.

[52] H. H. Mail, D. Indra, And R. Satra, "Buletin Sistem Informasi Dan Teknologi Islam Analisis Perbandingan Layanan Data Server Menggunakan Failover Cluster Pada Platform Nginx Dan Apache," *Jurnal Buletin Sistem Informasi Dan Teknologi Islam*, Vol. 1, No. 2, Pp. 87–91, May 2020.

[53] U. A. Ahmad, R. E. Saputra, And R. M. Harahap, "Implementasi High Availability Server Menggunakan Platform Haproxy (Studi Kasus: Aplikasi Zammad Untuk Online Help Desk)," In *E-Proceeding of Engineering*, Oct. 2021, Pp. 6237–6242.

[54] B. Prasetyo, S. R. Akbar, And W. Yahya, "Implementasi High Availability Pada Gateway Wireless Sensor Network Dengan Protokol Komunikasi Message Queuing Telemetry Transport," *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer*, Vol. 2, No. 10, Pp. 3280–3289, Oct. 2018.

[55] W. Ahmed, J. Vidal-Alaball, J. Downing, and F. López Seguí, "COVID-19 and the 5G Conspiracy Theory: Social Network Analysis of Twitter Data," Journal of Medical Internet Research, vol. 22, no. 5, p. e19458, May 2020, doi: 10.2196/19458.

[56] A. Arifin, G. P. Wardana, S. Arifin, M. F. Ridho, And H. K. Prabu, "Penerapan Cloud Load Balancing Dengan Menggunakan Haproxy Dalam Meningkatkan Server Availability Pada Studi Kasus Learning Management System (Lms) Universitas Xyz," In *Seminar Nasional Mahasiswa Ilmu Komputer Dan Aplikasinya (Senamika)*, Jakarta, Aug. 2022, Pp. 550–562.

[57] G. P. Sajati And B. T. Handoko, "Implementasi Sistem Terdistribusi Load Balancing Haproxy Dan Replikasi Mysql," *J Teknol*, Vol. 6, No. 1, Pp. 1–6, Dec. 2019.