



Optimizing Domain Name System (DNS) Performance Using Quality of Service (QoS) and Load Balancing in a Campus Network

Shitu Murtala*, Yahaya Basira, Ore-ofe Ajayi, Isah Abdulrazaq Imam
Computer Engineering Department,
Ahmadu Bello University, Zaria.

ABSTRACT

This project focuses on optimizing the performance of the Domain Name System (DNS) in a campus network by utilizing Quality of Service (QoS) mechanisms and load balancing techniques to improve responsiveness, reliability, and scalability. The study aims to prioritize DNS queries and efficiently distribute requests across multiple DNS servers to minimize response time and prevent server overload. Four virtual machines (VMs) were deployed on two Dell PowerEdge R270 servers, with two Cisco switches, one as a core switch and the other as an access switch, forming the testbed environment. Wireshark was used to analyze DNS traffic in real scenario, measuring Round Trip Time (RTT) and Query Response Time (QRT) as key performance metrics. The results showed significant improvements in DNS resolution times, reduced network congestion, and enhanced system reliability, demonstrating the effectiveness of QoS and load balancing in optimizing DNS performance. While the study is limited by the testbed environment and may need further validation in larger, real-world deployments, the findings offer valuable insights for enhancing DNS performance in complex campus network infrastructures.

ARTICLE INFO

Article History

Received: August, 2024

Received in revised form: November, 2024

Accepted: December, 2024

Published online: December, 2024

KEYWORDS

Domain Name System (DNS), Dell PowerEdge R270, Quality of Service (QoS), Query Response Time (QRT), Round Trip Time (RTT), Virtual Machines.

INTRODUCTION

DNS (Domain Name System) is a hierarchical and distributed naming system that translates human-readable domain names (e.g., www.campus.edu.ng) into machine-readable IP addresses (e.g., 192.0.2.1). This system is essential for the functionality of the Internet, enabling users to access websites, send emails, and use networked services without needing to remember numeric IP addresses. DNS operates using a network of servers that store and resolve domain name records, ensuring a scalable and fault-tolerant mechanism for addressing resources on the Internet [1] [2]. Load balancing and Quality of Service (QoS) techniques play a significant role in optimizing DNS performance in such large-scale networks. Load balancing ensures that DNS queries are distributed efficiently across multiple servers, preventing individual servers from becoming overwhelmed by traffic spikes. This distribution can significantly reduce DNS query resolution time and improve network reliability [3]. The motivation behind this project stems from the increasing reliance on DNS

services in modern campus networks. As educational institutions expand their digital presence, DNS is tasked with handling a larger volume of requests for applications such as online learning platforms, email systems, and research databases. A performance bottleneck in DNS resolution can significantly affect user experience, especially during peak usage times. Slow DNS responses or DNS server failures can lead to delays in accessing critical online resources, frustrating users and hindering productivity. Furthermore, many campus networks are built on traditional DNS architectures, which might not be equipped to handle the demands of modern usage. Optimizing DNS performance using QoS and load balancing strategies can enhance the overall user experience, ensure business continuity, and improve the efficiency of network resources. This research aims to explore how these techniques can be effectively implemented in a campus network to overcome DNS performance challenges.

Campus networks often face several challenges in maintaining efficient and reliable

Corresponding author: Shitu Murtala

✉ shitulle@gmail.com

Computer Engineering Department, Ahmadu Bello University, Zaria.

© 2024. Faculty of Technology Education. ATBU Bauchi. All rights reserved

DNS services due to: High Traffic Volumes: Increased demand for online resources and services leads to a higher number of DNS queries, especially in institutions with a large number of students and staff; Network Congestion: Traffic from different applications and services can cause congestion in the network, affecting DNS query resolution times; Single DNS Server Bottleneck: Many campus networks rely on a single DNS server, which can become overwhelmed by traffic and fail to provide timely responses to DNS queries; Lack of Redundancy: Failure to implement DNS redundancy may lead to service disruptions if the primary DNS server goes down. This research seeks to address these problems by optimizing DNS performance through QoS to prioritize DNS traffic and load balancing to distribute DNS queries across multiple servers. By addressing these challenges, the research aims to enhance DNS reliability and reduce the chances of service outages or performance degradation using Round Robin Load Balancing.

LITERATURE REVIEW

DNS is often the first point of contact for users trying to access resources over the internet, making its performance paramount [3]. In campus networks, which often experience heavy traffic, DNS queries can lead to significant latency if not managed properly. Some of the main challenges include high query volumes, DNS caching issues, and the risks of single points of failure [4]. Studies have also identified security concerns, such as DNS amplification attacks, which can further hinder performance [5].

QoS techniques help prioritize network traffic to ensure that time-sensitive applications, like DNS, are given precedence over less critical traffic. According to [6], using QoS in a campus network ensures that DNS queries are processed with lower latency, even during periods of high traffic. Load balancing helps distribute DNS queries across multiple servers to prevent overload on a single server. It can either be done using round-robin methods or more advanced algorithms based on traffic volume and server health [7]. Load balancing not only reduces DNS query response times but also enhances reliability by mitigating the impact of server failures [8].

Recent advancements in DNS optimization, such as the use of anycast routing for geographically distributed DNS servers, have helped to reduce latency and improve fault tolerance [9]. Furthermore, adaptive DNS algorithms that dynamically adjust based on traffic patterns have shown promising results in improving campus network performance [10].

Round-robin DNS is a widely used technique for distributing DNS queries across multiple servers to balance the load and increase the availability of network services. This method involves mapping a single domain name to multiple IP addresses in a DNS record. When a client queries the DNS server for the domain name, the server returns a list of IP addresses, and the order of the IP addresses is rotated with each subsequent query, ensuring that requests are distributed evenly across the servers [11]. A class map is a mechanism used to classify traffic based on specific attributes such as IP address, protocol type, port number, or other packet characteristics. Class maps define the types of traffic that you want to apply QoS policies to by matching traffic with particular criteria [12]. Policy map on the other hand defines how the router should treat traffic after it has been classified. Policy maps are used to specify actions (such as prioritizing, policing, or shaping) for different classes of traffic that were defined in class maps [13].

METHODOLOGY

The methodology for this research on optimizing DNS performance using load balancing and Quality of Service (QoS) focused on collecting data from three web servers and a DNS server to evaluate the effects of Round Robin load balancing and QoS on DNS query response times and overall network performance. The domain name of the DNS server was configured as www.campus.edu.ng, and the system infrastructure was built using four Virtual Machines (VMs) created from two live Dell R270 Rack servers and hosted on Microsoft Window Server 2008. Networking was managed with a Cisco 3760 Switch and a Cisco 2960 Access Switch. Wireshark was used for packet analysis to monitor and evaluate network traffic before and after the implementation of Round Robin load balancing and QoS. In this setup, three web servers were created with the private IP addresses:

Corresponding author: Shitu Murtala

✉ shitulle@gmail.com

Computer Engineering Department, Ahmadu Bello University, Zaria.

© 2024. Faculty of Technology Education. ATBU Bauchi. All rights reserved

10.202.0.11, 10.202.0.12, and 10.202.0.13, while the DNS server was configured with the IP address 10.202.0.14. These servers were designed to simulate real-world DNS query resolution, with the web servers hosting basic applications to generate traffic and the DNS server responsible for resolving queries for the domain www.campus.edu.ng.

The analysis was conducted in two phases: Before the configuration of Round Robin load balancing and QoS and after the configuration of Round Robin load balancing and QoS. The Round Robin load balancing technique was used to distribute incoming DNS queries evenly across the three web servers. This method helped ensure that no single server became overloaded, thereby improving overall system performance and reducing the chances of bottlenecks during high traffic periods. In parallel, QoS was configured on the Cisco 3760 Switch to prioritize DNS query traffic, ensuring that critical DNS traffic was handled with higher priority compared to other types of traffic.

To measure the performance of the DNS queries under both configurations, data was collected and analyzed using two key metrics: Round Trip Time and DNS query response. The Round-Trip Time (RT) metric measures the time taken for a DNS query to travel from the client to the DNS server and back, providing insights into the efficiency of the DNS resolution process, while the DNS Query Response Time metric assesses how long it takes for the DNS server to respond to a query, a critical factor in the overall user experience for web applications. Data captured from the web servers, DNS servers, and network traffic were analyzed using Wireshark in both phases. Wireshark allowed for the detailed examination of DNS packet flow, enabling a comparison of DNS query performance before and after configuring Round Robin load balancing and QoS policies. The collected data was then evaluated to identify improvements in DNS query response times, reduced latency, and enhanced network efficiency as a result of the applied optimizations.

RESULTS AND DISCUSSION

The round-trip ping results to the DNS server (www.campus.edu.ng) before optimization are as follows. The first ping recorded a maximum round trip time (RTT) of 5ms and an average RTT of 2ms. The second ping recorded a maximum RTT of 4ms and an average RTT of 1ms. The third ping measured a maximum RTT of 2ms and an average RTT of 1ms. The fourth ping recorded a maximum RTT of 2ms and an average RTT of 1ms. The summary of the ping results before optimization is presented in Table 1.

Table 1: Summary of RTT before optimization

Ping	Maximum RTT	Average RTT
First Ping	5ms	2ms
Second Ping	4ms	1ms
Third Ping	2ms	1ms
Fourth Ping	2ms	1ms

After optimization, the round-trip ping results to the DNS server were recorded. The first ping showed a maximum RTT of 3ms and an average RTT of 1ms. The second ping measured a maximum RTT of 1ms and an average RTT of 0ms. The third ping had a maximum RTT of 1ms and an average RTT of 0ms. The fourth ping recorded a maximum RTT of 1ms and an average RTT of 0ms. The summary of the ping results after optimization is presented in Table 2.

Table 2: Summary of RTT after optimization

Ping	Maximum RTT	Average RTT
First Ping	3ms	1ms
Second Ping	1ms	0ms
Third Ping	1ms	0ms
Fourth Ping	1ms	0ms

The key findings indicate that after optimization, there was a consistent reduction in latency across all pings. The maximum and average RTTs were significantly lower, indicating an improvement in network performance. Before optimization, the network showed a gradual decrease in round trip times but still exhibited higher latencies than after optimization. The optimized network

demonstrated remarkable reductions in RTT, particularly in the second, third, and fourth pings, where the average RTT dropped to 0ms.

Two DNS query tests were conducted, one before optimization and one after optimization. The first randomly captured Wireshark packet before optimization is shown in Figure 1, while the second packet is illustrated in Figure 2. These packets provide insights into network behavior, particularly focusing on Request in and Time values. The first packet capture after optimization is presented in Figure 3, and the second is shown in Figure 4. These figures highlight improvements in DNS query response times after optimization. The input/output packet capture before optimization is depicted in Figure 4.13, where the y-axis represents the number of packets per second, and the x-axis represents the time taken for a packet to travel from source to destination. The graph highlights overall packet transmission, TCP errors, and filtered DNS packets, providing an analysis of network performance before optimization. After optimization, Figure 4.14 presents the improved input/output graph, which demonstrates a reduction in packet travel time, fewer TCP

errors, and optimized DNS queries, reflecting enhanced network efficiency.

The DNS query response time was analyzed before and after optimization. Before optimization, the first test recorded a request in time of 493ms and a round-trip time of 299.06ms, while the second test had a request in time of 3353ms and a round-trip time of 2101.28ms. After optimization, the first test had a request in time of 1135ms and a significantly reduced round trip time of 127.23ms. The second test showed even greater improvement, with a request in time of 504ms and a drastically reduced round trip time of 1.38ms. The findings indicate that before optimization, the round-trip times were significantly higher, especially in the second test, which recorded a latency of over 2 seconds. After optimization, the improvements were evident, with the second test achieving an almost instantaneous response. The significant reduction in DNS resolution time suggests that the optimization techniques were effective in reducing network latency and improving performance.

Wireshark - Packet 529 - Ethernet

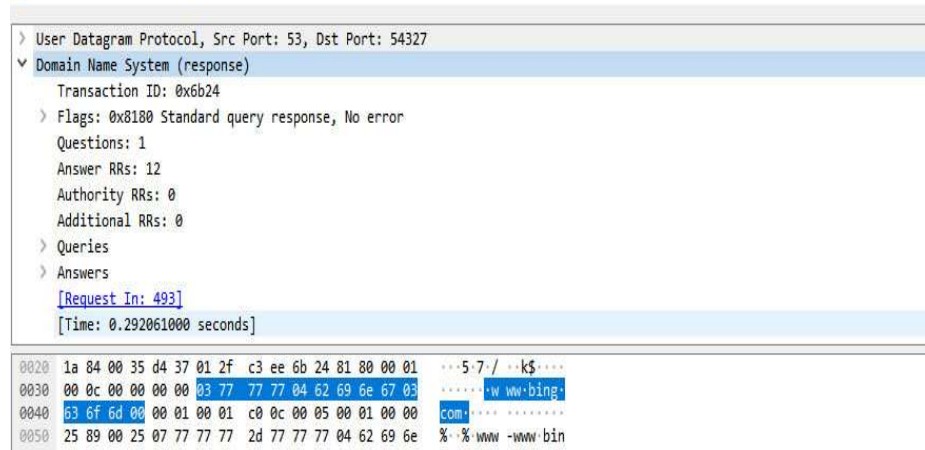
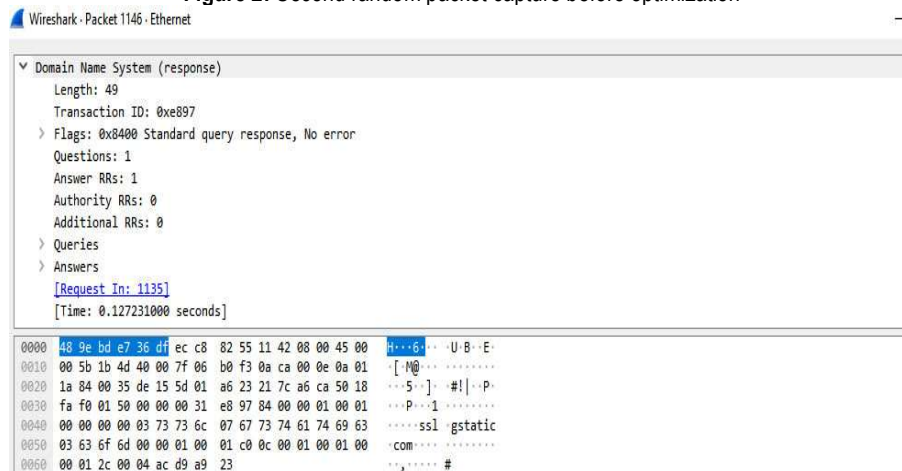


Figure 1: First random packet capture before optimization

```
> Frame 3382: 105 bytes on wire (840 bits), 105 bytes captured (840 bits) on interface \Device\NPF_{AA9FDFDE-8378-477C-AFC5-FC10C58DB047}, id 0
> Ethernet II, Src: Cisco_55:11:42 (ec:c8:82:55:11:42), Dst: HP_e7:36:df (48:9e:bd:e7:36:df)
> Internet Protocol Version 4, Src: 10.202.0.14, Dst: 10.1.26.132
> Transmission Control Protocol, Src Port: 53, Dst Port: 61914, Seq: 1, Ack: 36, Len: 51
v Domain Name System (response)
  Length: 49
  Transaction ID: 0xc08c
  Flags: 0x0400 Standard query response, No error
  Questions: 1
  Answer RRs: 1
  Authority RRs: 0
  Additional RRs: 0
  Queries
  Answers
  [Request In: 3353]
  [Time: 2.101276000 seconds]
```

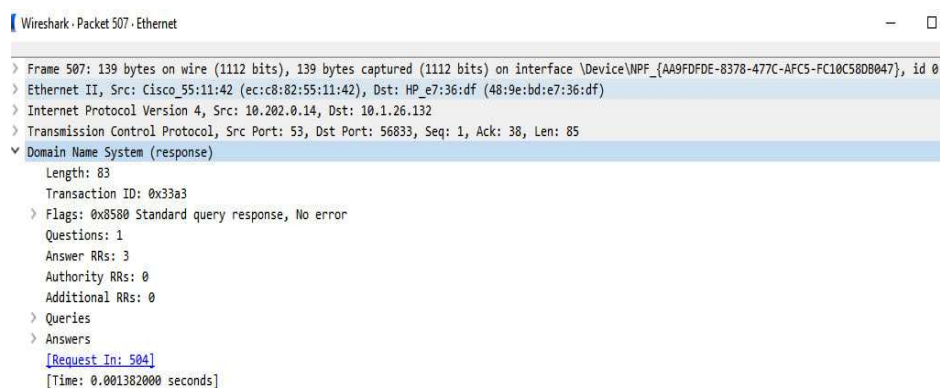
Figure 2: Second random packet capture before optimization



```
Wireshark - Packet 1146 - Ethernet
v Domain Name System (response)
  Length: 49
  Transaction ID: 0xe097
  Flags: 0x0400 Standard query response, No error
  Questions: 1
  Answer RRs: 1
  Authority RRs: 0
  Additional RRs: 0
  Queries
  Answers
  [Request In: 1135]
  [Time: 0.127231000 seconds]

0000 48 9e bd e7 36 df ec c8 82 55 11 42 08 00 45 00  H...6...U.B..E
0010 00 5b 1b 4d 40 00 7f 06 b0 f3 0a ca 00 0e 0a 01  :[.M.....
0020 1a 84 00 35 de 15 5d 01 a6 23 21 7c a6 ca 50 18  ..5...#|...P
0030 fa f0 01 50 00 00 00 31 e8 97 84 00 00 01 00 01  ...P...1.....
0040 00 00 00 00 03 73 73 6c 07 67 73 74 61 74 69 63  ....ssl.gstatic
0050 03 63 6f 6d 00 00 01 00 01 c0 0c 00 01 00 01 00  .com.....
0060 00 01 2c 00 04 ac d9 a9 23  ....#
```

Figure 3: First random packet captured after optimization



```
Wireshark - Packet 507 - Ethernet
v Frame 507: 139 bytes on wire (1112 bits), 139 bytes captured (1112 bits) on interface \Device\NPF_{AA9FDFDE-8378-477C-AFC5-FC10C58DB047}, id 0
> Ethernet II, Src: Cisco_55:11:42 (ec:c8:82:55:11:42), Dst: HP_e7:36:df (48:9e:bd:e7:36:df)
> Internet Protocol Version 4, Src: 10.202.0.14, Dst: 10.1.26.132
> Transmission Control Protocol, Src Port: 53, Dst Port: 56833, Seq: 1, Ack: 38, Len: 85
v Domain Name System (response)
  Length: 83
  Transaction ID: 0x33a3
  Flags: 0x0500 Standard query response, No error
  Questions: 1
  Answer RRs: 3
  Authority RRs: 0
  Additional RRs: 0
  Queries
  Answers
  [Request In: 504]
  [Time: 0.001382000 seconds]
```

Figure 4: Second random packet captured after optimization

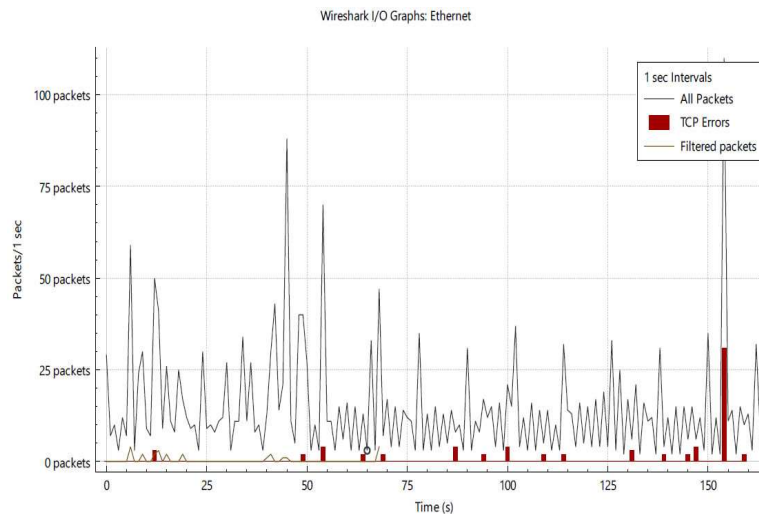


Figure 5: Input/output graph before optimization

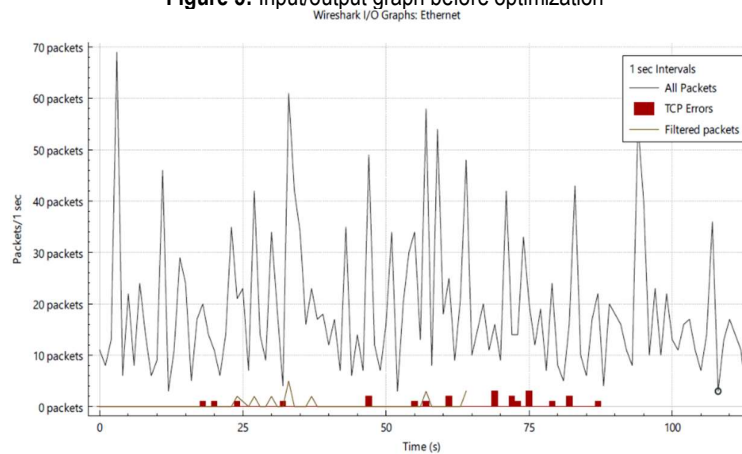


Figure 6: Input/output graph after optimization

ACKNOWLEDGMENT

I would like to sincerely thank the Ndukwe Kalu Foundation (NKF) for their generous support through their grant, which played a significant role in making this work possible. Their encouragement and investment in research have been truly invaluable.

I also extend my gratitude to everyone who contributed to this work, including mentors, colleagues, and loved ones, for their guidance and unwavering support throughout this journey.

CONCLUSION

The paper aimed to optimize network performance by utilizing QoS and load

balancing techniques, focusing on two key metrics: Round Trip Time (RTTT) and Query Response Time. This study analyzed and optimized DNS query response times by measuring Request in and RTTT both before and after implementing optimization strategies. The results demonstrated a significant improvement in DNS performance. In the first test, RTTT decreased from 299.06ms to 127.23ms, while in the second test, it plummeted from 2101.28ms to 1.38ms, reflecting a near-instantaneous DNS resolution after optimization. Similarly, the Request In time for the second test improved from 3353ms to 504ms. These results confirm that the DNS optimization techniques effectively reduced

Corresponding author: Shitu Murtala

shitulle@gmail.com

Computer Engineering Department, Ahmadu Bello University, Zaria.

© 2024. Faculty of Technology Education. ATBU Bauchi. All rights reserved

latency, accelerated query initiation, and improved resolution times, thereby enhancing overall DNS performance and user experience. Two significant findings of the research include much faster website response, as the optimization significantly improved the time it takes for the system to respond, and less delay before initiating website lookups, reducing the wait time before the system starts resolving a website's address. For further work, larger-scale testing should be conducted across different types of networks and DNS providers to assess the optimization's effectiveness in various real-world scenarios. Additionally, advanced optimization techniques such as DNS caching, dynamic DNS load balancing, and DNS prefetching should be explored to further minimize delays, particularly during high traffic periods. Incorporating security features like DNSSEC is also recommended to protect against threats such as cache poisoning while maintaining an optimal balance between security and performance.

REFERENCES

- [1] Mockapetris, P. V. (1987). RFC1034: Domain names-concepts and facilities.
- [2] Liu, C., & Albitz, P. (2006). *DNS and Bind*. "O'Reilly Media, Inc."
- [3] Huang, S., Zhang, J., & Yao, L. (2021). A review of DNS performance optimization in large-scale networks. *IEEE Access*, 9, 13225-13240. <https://doi.org/10.1109/ACCESS.2021.3052789>
- [4] Klein, M., & Liu, W. (2020). Managing DNS query load in high traffic networks: A review. *International Journal of Networking and Computing*, 10(2), 198-210. <https://doi.org/10.1559/ijnc.10.2.198>
- [5] Li, L., Zhang, Y., & Zhang, Y. (2022). Enhancing DNS security and performance under load: A survey. *Computers & Security*, 109, 102367. <https://doi.org/10.1016/j.cose.2021.102367>
- [6] Zhang, T., Hu, Z., & Liu, H. (2023). Optimizing DNS performance using QoS in a campus network. *Wireless Networks*, 29
- [7] Wang, H., Zhou, X., & Wei, J. (2020). Load balancing strategies for DNS servers: An empirical study. *International Journal of Computer Applications*, 177(7), 15-22. <https://doi.org/10.5120/ijca2020918487>
- [8] Gao, L., & Wu, P. (2021). A comprehensive survey of DNS load balancing methods. *Computer Networks*, 193, 108060. <https://doi.org/10.1016/j.comnet.2021.108060>
- [9] Zhou, Z., Guo, C., Xu, H., Zhang, X., Fan, Y., & Zhang, L. (2023, October). Be-dns: Blockchain-enabled decentralized name services and p2p communication protocol. In *2023 IEEE 9th World Forum on Internet of Things (WF-IoT)* (pp. 1-6). IEEE.
- [10] Chen, Z., Zhang, X., & Li, S. (2022). Optimizing DNS query performance using hybrid load balancing techniques. *Journal of Network and Systems Management*, 30(4), 1201-1215. <https://doi.org/10.1007/s10922-022-09516-7>
- [11] Li, D. C., & Chang, F. M. (2007). An In-Out Combined Dynamic Weighted Round-Robin Method for Network Load Balancing. *The Computer Journal*, 50(5), 555-566.
- [12] Cisco Systems (2024), "Understanding Class Maps, Policy Maps, and Service Policies," Cisco Documentation, Cisco.com.
- [13] Cisco Systems (2020), "QoS Overview," Cisco IOS Quality of Service Solutions Configuration Guide, Cisco.com.

Corresponding author: Shitu Murtala

✉ shitulle@gmail.com

Computer Engineering Department, Ahmadu Bello University, Zaria.

© 2024. Faculty of Technology Education. ATBU Bauchi. All rights reserved