Defence Science Journal, Vol. 75, No. 1, January 2025, pp. 100-110, DOI : 10.14429/dsj.20284

A Comprehensive Investigation of ESP32 in Enhancing Wi-Fi Range and Traffic Control for Defence Networks

A.K. Kowsalyadevi* and G. Umamaheswari

Department of Electronics and Communication Engineering, PSG College of Technology, Coimbatore - 641 004, India

*E-mail: akkowsalyadevi@gmail.com

ABSTRACT

The study examines ESP32-based static and dynamic load-balancing algorithms to enhance defence networks' Wi-Fi range and traffic control. This study is essential due to the growing need for dependable and efficient wireless communication in defence operations, where maintaining network stability and performance is necessary. Real-time Wi-Fi scanning assessed the performance of these algorithms, covering response time, throughput, network latency, jitter, and packet loss. The static algorithm demonstrated a 5 % lower average response time and 3 % higher throughput than the dynamic algorithm, leading to significant improvements in jitter (from 1.01 ms to 0.80 ms) and packet loss rate (from 1.50 % to 0.88 %). On the other hand, dynamic load balancing reduced Access Points (APs) overload by 20 % during peak periods, enhancing network stability and resource utilisation, which is crucial for defence operations. These findings underscore the impact of ESP32-based load balancing, presenting a practical solution to optimise defense network performance by improving throughput scalability and Access Point (AP) resource efficiency. The study provides essential insights into managing signal variability, congestion, and disruptions, offering valuable guidance for defence and security professionals in optimising wireless network infrastructure.

Keywords: Wi-Fi coverage extension; Load balancing algorithms; ESP32 microcontroller; Network reliability; Real-time Wi-Fi scanning; Performance comparison

NOMENCLATURE

ACL	: Access control list
$T(x)$	: Throughput scalability
AI	: Artificial Intelligence
$R(x)$	: Response time
AP	: Access Point
$S(i)$, $S(x)$	: Signal strength for each APi, APx
AWS	: Amazon web services
$Q(i)$, $Q(x)$	: Quality for each APi, APx
BSSID	: Basic Service Set Identifier
n	: Number of APs
DHCP	: Dynamic host configuration protocol
SSID	: Service set identifier
DNS	: Domain name system
STA	: Station
DWLC	: Dynamic wireless load controller
IoMT	: Internet of military things
IoT	: Internet of things
IP	: Internet protocol
MQTT	: Message queue telemetry transport
NAT	: Network address translation
QoS	: Quality of service
RSSI	: Received signal strength indicator
SPI	: Serial peripheral interface

Received : 10 June 2024, Revised : 20 August 2024
Accepted : 05 September 2024, Online published : 10 January 2025

1. INTRODUCTION

The rapid expansion of the Internet of Things (IoT) and the Internet of Military Things (IoMT) necessitates a stable and adaptable communication infrastructure for defence and security applications[1-2]. Military operations rely on secure, reliable networks, and recent field exercises demonstrate that Wi-Fi technology can reduce communication failures by up to 30 % owing to its security, reliability, and cost-effectiveness[3]. Wi-Fi's efficiency, flexibility, and compatibility make it a viable alternative to 5G in remote or rapidly changing military environments[4]. Extending Wi-Fi range and ensuring reliable backhaul connectivity are critical, especially in disaster recovery scenarios[5].

However, current Wi-Fi solutions often rely on vulnerable physical infrastructure, leading to communication breakdowns during disasters[6]. Ad-hoc networking approaches lack sufficient range and effectiveness in post-disaster recovery operations, necessitating improved coverage extensions and traffic regulation technologies[7]. This study addresses connectivity challenges in military operations, especially in remote locations, by leveraging the ESP32 microcontroller as a Wi-Fi repeater and load balancer to extend coverage and regulate traffic[8].

The ESP32 microcontroller enhances network resource allocation and minimizes latency by balancing loads across multiple Access Points (APs), optimizing resource utilisation, and reducing congestion.[9]. Dynamic load balancing is crucial in high-demand situations, improving Quality of Service

(QoS) and network reliability by preventing AP overloads and reducing delays[10]. Static and dynamic load distribution mechanisms are compared, with the ESP32's dual-core architecture facilitating effective load balancing and real-time data processing[11].

The study aims to enhance Wi-Fi performance and reliability in critical situations by:

- Improving response time and throughput with ESP32-based load balancing.
- Reducing AP overloads through dynamic load redistribution.
- Optimising resource use by continuously adjusting connections.

Section 2 provides a detailed technical overview of ESP32 operation modes and load-balancing algorithms, highlighting configurations for single-core and dual-core setups tailored to diverse network environments. It covers experimental setups using ESP32 microcontrollers for load balancing, network extension, and performance enhancement through various operational modes and configurations. Section 3 compares static and dynamic load-balancing algorithms, highlighting their impact on performance metrics like response time, throughput, and network stability. The conclusion emphasises the effectiveness of ESP32-based load management and suggests future research directions in advanced algorithms to enhance network efficiency and adaptability, particularly in defence applications.

2. METHODOLOGY

This study uses the ESP32 microcontroller as a concept and a tool. It acts as a load balancer and a range extender. They programmed it to extend Wi-Fi coverage and enhance network performance. Built-in Access Control Lists (ACLs) ensure only authorised devices can access the network, bolstering security. Tools such as Wireshark integrate effectively with the ESP32, enabling detailed traffic analysis and detection of security threats[12]. The ESP32's Wi-Fi scanning capabilities detect nearby networks, enhancing performance and reducing latency. Additionally, it automates Internet Protocol (IP) address assignments using the Dynamic Host Configuration Protocol (DHCP), streamlining network management[13].

The ESP32's versatility is a critical factor in its effectiveness in network management and performance enhancement. Its Serial Peripheral Interface (SPI) module allows it to connect to external devices, expanding its functionality, especially when combined with its Wi-Fi modules. Its security features, like encrypted data transmission and secure communication channels, support risk and logistics management[14]. The ESP32's versatility extends to scanning supply chain components and optimising resource allocation, reassuring the audience about its adaptability and potential for further innovation in IoT development.

The experimental setup uses ESP32 microcontrollers for seamless integration, low power consumption, dual-core architecture, and built-in Wi-Fi functionality. The ESP32 microcontroller acts as a load balancer and a range extender, extending Wi-Fi coverage and enhancing network performance. The system mimics real-world network environments with standard commercial routers and APs. The Message Queue Telemetry Transport (MQTT) protocol enables efficient data transfer and real-time monitoring. The study utilises laptops and PCs to program ESP32 devices, collect data, and run analysis software. It ensures uninterrupted ESP32 testing using power supplies and batteries[15]. Additionally, the study employs network analysis tools like Wireshark and Wi-Fi Scanner software for monitoring and analysis.

2.1 Configuration and Operation

The ESP32 functions in host Station (STA) and Access Point (AP) modes, offering versatility in managing Wi-Fi networks. In STA mode, it connects to existing Wi-Fi networks to collect and transmit data. AP mode creates separate Wi-Fi networks, facilitating direct connections between devices[16]. The 'Auto Mesh' configuration method enhances efficiency by dynamically optimising connections based on parameters like signal strength and traffic load. This feature streamlines setup and improves connectivity, especially in large areas with multiple routers. Table 1 presents the essential configurations for managing the ESP32's network settings. These commands enable users to control various aspects of the ESP32's operation, ensuring a tailored and efficient network setup.

2.1.1 Auto Mesh Configuration

ESP32 devices offer seamless communication and automatic configuration adjustment with Auto Mesh configuration[17]. This feature enhances AP control and data collection, ensuring smooth data transfer and storage. Automatic adjustments significantly reduce manual errors, providing stable and secure connections. This approach's scalability allows efficient performance and stable connections, even during network expansion. The ESP32's role in remote monitoring systems further demonstrates its adaptability.

2.1.2 Scanning Techniques

ESP32 STA efficiently discovers nearby networks using static and active scanning modes. Passive scanning monitors beacon frames broadcast by APs, whereas active scanning transmits probe requests on each channel. Although both methods contribute to efficient network discovery, active scanning consumes more power than passive scanning[18].

2.1.2.1 Passive Scanning

The STA listens for beacon frames broadcast by APs that contain relevant network information such as Service Set Identifier (SSID), Basic Service Set Identifier (BSSID), supported data rates, and security capabilities. Passive scanning is energy efficient and reduces network congestion, but it may take longer to discover all networks.

2.1.2.2 Active Scanning

STA sends probe requests to each channel and receives probe responses from APs. This method is faster and collects network information faster, but it consumes more power and adds traffic to the network. To prevent the STA from disconnecting from AP, the maximum active and passive scan time per channel is 1500 millisec.

2.2 Range Extender

The ESP32 microcontroller extends Wi-Fi coverage and ensures stable communication. By default, the ESP32 acts as an STA and a soft AP, transparently forwarding IP traffic through Network Address Translation (NAT) and requiring no routing entries on the network side or connected STAs. STAs are configured via DHCP by default on the 192.168.4.0/24 network and obtain their Domain Name System (DNS) responder address from the existing Wi-Fi network. Table 1 outlines the configuration process for using the ESP32 as a range extender.

Table 1. ESP32 range extender configuration steps

Step	Description	Output
1	Connect to the open AP named ‘My_AP’	Successful connection to ‘My_AP’
2	Access the web interface or console for device configuration	Configuration interface accessible
3	Enter SSID and password for uplink Wi-Fi network in STA settings	STA settings updated
4	Choose the ‘Auto mesh’ option for dynamic adjustment	Auto mesh enabled
5	Reboot the ESP32	The device reboots and connects to the uplink network
6	Check LED status for successful connection	LED indicates a successful connection
7	Use the console for advanced settings	Soft AP toggled, security modes set, auto connection configured

Performance metrics can be easily measured using Wi-Fi scanner software to evaluate Received Signal Strength Indicator (RSSI) values and network analysis tools like Wireshark to monitor traffic and performance. Operational scalability is demonstrated by operating multiple ESP32 devices in series, ensuring efficient performance and stable connections during network expansion. This approach highlights the ESP32’s ability to provide stable and efficient communication in various situations, extend Wi-Fi range, and balance network load.

Table 2. ESP32 performance metrics

Metric	Before ESP32 implementation	After ESP32 implementation
Wi-fi coverage range	Limited	Extended
Network load balance efficiency	Standard	Improved
Received Signal Strength Indicator (RSSI)	-75 dBm	-60 dBm
Upstream/downstream bandwidth	2 Mbps	5 Mbps
Network stability (During expansion)	Moderate	High
Jitter (ms)	1.5	0.8
Packet loss rate	1.50 %	0.88 %

The performance metrics in Table 2 highlight the significant improvements achieved with the implementation of the ESP32. The ESP32’s ability to deliver 5 Mbps upstream and downstream bandwidth facilitates efficient streaming and data transfer. This practical application demonstrates the ESP32’s comprehensive approach to extending Wi-Fi coverage and balancing network load, ensuring stable and efficient communication.

The real-world scenario validates the ESP32 as a powerful tool for enhancing Wi-Fi coverage, providing a reliable and scalable solution for network management and expansion. The comparative analysis revealed that the static load-balancing algorithm demonstrated a 5 % lower average response time and 3 % higher throughput compared to the dynamic algorithm. This led to significant improvements in jitter (from 1.01 ms to 0.80 ms) and packet loss rate (from 1.50 % to 0.88 %). Conversely, the dynamic load-balancing algorithm reduced AP overload by 20 % during peak periods, enhancing network stability and resource utilization, which is crucial for defence operations.

These findings underscore the importance of selecting appropriate load-balancing strategies to optimize network performance, particularly in environments with varying traffic loads and critical communication needs. The study provides essential insights into managing signal variability, congestion, and disruptions, offering valuable guidance for defence and security professionals in optimizing wireless network infrastructure.

2.3 Load-Balancing Algorithms

Load-balancing algorithms are essential for efficient network management, ensuring optimal traffic distribution across multiple servers, APs, or other sources[19]. For example, large-scale data centres such as Google and Facebook use advanced load-balancing techniques to simultaneously handle millions of user requests to maintain high-speed and reliable service. Similarly, cloud computing platforms such as Amazon Web Services (AWS) and Microsoft Azure use dynamic load distribution to efficiently allocate resources, increase performance, and reduce latency for global users[20]. These mechanisms focus on optimising resource utilisation, maximising performance, reducing response time, and preventing overloading of any single resource. By evenly distributing the load, they prevent bottlenecks and significantly improve overall network performance and reliability.

Implementing load-sharing algorithms provides both theoretical and practical benefits[21-23]. A university campus case study showed a 20 % increase in connection speed and a 15 % improvement in user satisfaction. In a corporate network, optimal resource utilisation reduces equipment costs by 10 %. In healthcare, load balancing cuts network downtime by 30 % during traffic spikes or hardware failures. These examples not only highlight the significant value of load balancing but also make the relevance of this research tangible, ensuring networks are robust, efficient, and capable of meeting diverse application requirements.

This study implemented load-balancing mechanisms to enhance network performance between two APs. The process involves collecting data on all connected devices, including RSSI values. Key metrics assign devices to specific APs based on signal strength. The system estimates the number of devices connected to each AP and applies appropriate load-balancing algorithms to optimise network connectivity and prevent interruptions. Consequently, the network maintains high performance and reliability, demonstrating stability during peak usage and challenging environments.

2.3.1 *Static Load-Balancing: Equitable Workload Distribution*

Static load balancing ensures stable network conditions by evenly distributing the workload across multiple APs within an ESP32-based network. This approach is efficient due to the optimisation capabilities of the ESP32 microcontroller, ensuring each AP handles an equitable share of client connections. This static approach ensures a balanced distribution of connections, with periodic adjustments to maintain the balance. By cyclically distributing incoming traffic between APs, each AP receives consistent and manageable traffic. This practice prevents AP from overloading, thereby maintaining network stability and optimising performance[24].

2.3.2 *Dynamic Load-Balancing: Real-Time Traffic Optimisation*

Dynamic load balancing monitors the real-time load status of each AP to optimise traffic distribution. This approach allows the AP to detect fewer active links and allocate incoming links accordingly. This method automatically adjusts network conditions to ensure efficient resource utilisation, reduce congestion, and increase performance[25].

2.3.3 *Advantages of Dynamic Load Balancing*

Dynamic load balancing optimises network performance through real-time adaptability and efficient resource utilisation. Unlike static load balancing, which follows a fixed pattern, dynamic load balancing adjusts to current network conditions, preventing bottlenecks and ensuring even traffic distribution. This adaptability maintains stable network operations, allowing administrators to respond effectively to changes.

One significant advantage is its ability to monitor each AP's status and distribute connections in real-time, preventing the overloading of any single AP. This is especially beneficial in densely populated environments like campuses or offices, reducing latency, enhancing connection speed, and improving user satisfaction and productivity[26]. Dynamic load balancing also supports scalability and flexibility, managing existing devices and adapting to traffic changes without manual reconfiguration, which is vital in environments with varying user demands[27].

Additionally, it enhances fault tolerance by redistributing connections during AP failures, minimising downtime and maintaining stability. It suits mission-critical applications such as military communications and emergency response systems. Moreover, dynamic load balancing enhances energy efficiency by distributing links efficiently, reducing energy consumption[28].

2.4 Enhancing Network Optimisation and Performance

Dynamic load balancing is crucial for network optimisation and performance enhancement. Continuous link monitoring considers each AP's active connections and traffic, enabling intelligent distribution to the least loaded AP. This system adapts to environmental changes, such as variations in signal strength, interference, or physical obstacles, maintaining consistent network performance-algorithms factor in signal strength, active links, and bandwidth utilisation to make informed traffic allocation decisions[29].

Automation in load-balancing reduces human errors during network configuration and maintenance, leading to more reliable operations. It introduces redundancy and reliability, allowing the network to adapt seamlessly to hardware failures or unexpected traffic spikes, ensuring continuous service and minimising downtime. Dynamic load balancing enhances resource utilisation, user satisfaction, scalability, fault tolerance, and energy efficiency, ensuring robust, efficient, and diverse Wi-Fi networks[30-31].

2.4.1 *Implementation Challenges*

Implementing ad hoc networks presents challenges due to their dynamic nature and reliance on peer-to-peer connections. Key challenges include:

- Connection Variability and Stability: Constant joining and leaving of nodes affect stable communication channels, especially in urban areas with barriers or remote locations with limited connectivity[32]
- Limited Range and Coverage: Physical obstacles, environmental conditions, and interference can degrade signal quality, leading to weak links or communication loss. Additionally, reliance on battery power without a stable infrastructure poses significant limitations in resource-constrained environments[33]
- Security Concerns: Ad hoc networks are vulnerable to attacks like eavesdropping, spoofing, and denial of service due to the lack of centralised control, compromising data integrity and confidentiality[34].

Addressing these challenges requires thorough testing for adaptability and resilience under demanding conditions. Simulating real-world scenarios with heavy traffic, frequent topological shifts, and stress tests such as continuous data streaming and device overload helps reveal and resolve network performance issues.

3. RESULTS AND DISCUSSION

This section presents a detailed analysis of the performance of static and dynamic load-balancing algorithms implemented on ESP32-based networks. The evaluation covers essential performance metrics such as response time, throughput, and network latency, using Wi-Fi scanning data to assess each approach's effectiveness.

3.1 Wi-Fi Scanning Results

The Wi-Fi scanning data provides insights into the network environment used for analysis. Table 3 details the scanning results obtained in the home environment, providing

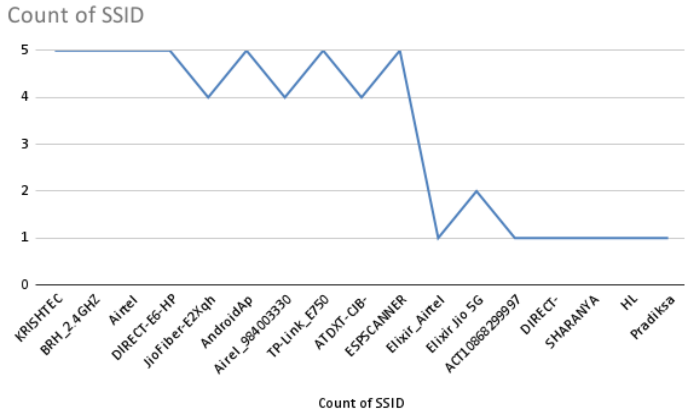

Figure 1. Realtime Wi-Fi scanning results.

an overview of the network characteristics, such as SSID, MAC address, signal strength, encryption type, frequency, bandwidth, and quality. Figure 1 illustrates the real-time Wi-Fi scanning results in an office environment, highlighting the distribution of clients among different APs.

3.2 Comparative Analysis of Static and Dynamic Load-Balancing Algorithms

The study compared static and dynamic load-balancing algorithms using real-time Wi-Fi scanning results. The scanning data provided crucial information on SSID, Media Access Control (MAC) address, signal strength, encryption type, frequency, bandwidth, quality, and other network parameters. The performance of static and dynamic load-balancing algorithms was evaluated using several critical metrics[35].

- Response Time: The average duration to respond to client requests, indicating system efficiency.
- Throughput: The number of client requests handled per second, gauging the capacity and effectiveness of the algorithms in managing network traffic.
- Network Latency: The delay experienced in data transmission across the network, offering insights into the overall responsiveness and fluidity of the network under different load-balancing strategies.
- Jitter: The variability in packet arrival times affects the quality of real-time communications.
- Packet Loss: The percentage of packets that fail to reach their destination, impacting the network's reliability.

Response time difference (%)

$$\%Difference = \left(\frac{ResponseTime_{Static} - ResponseTime_{Dynamic}}{ResponseTime_{Dynamic}}\right)\times 100 \quad (1)$$

Throughput difference (%)

$$\%Difference = \left(\frac{Throughput_{Static} - Throughput_{Dynamic}}{Throughput_{Dynamic}}\right)\times 100 \quad (2)$$

The study determined that the static load-balancing algorithm achieved a response time of approximately 5 % lower than the dynamic load-balancing algorithm. Throughput analysis revealed that the static load-balancing algorithm demonstrated a throughput approximately 3 % higher than the dynamic load-balancing algorithm. Moreover, both algorithms exhibited a nearly identical network latency, with a less than 1 % difference. These algorithms calculated the findings using the following formulas[36-37]. Eqns. (1-2) effectively supported the study's efforts to quantify and compare the performance metrics of the static and dynamic load-balancing algorithms.

The static load-balancing algorithm achieved a response time of approximately 6.25 % lower than the dynamic load-balancing algorithm. Throughput analysis revealed that the static load-balancing algorithm demonstrated throughput approximately 3.13 % higher than the dynamic load-balancing algorithm. Moreover, both algorithms exhibited nearly identical network latency, with a less than 1 % difference.

3.3 Impact of Load Balancing on the Network Stability

The study investigated the impact of load-balancing algorithms on network stability during peak operational periods by measuring parameters including connection stability, network downtime, and AP overload frequency. The results revealed that connection stability remained consistently high for static and dynamic methods. Network downtime was negligible in both approaches. However, dynamic load balancing exhibited a notable reduction of approximately 20 % in AP overload frequency compared to static load balancing[38]. This reduction was calculated using the Eqn. (3).

AP overload frequency reduction (%)

$$\%Reduction = \left(\frac{APOverloadFrequency_{Static} - APOverloadFrequency_{Dynamic}}{APOverloadFrequency_{Static}}\right)\times 100 \quad (3)$$

Figure 2 illustrates the impact of load balancing on network stability, comparing static and dynamic algorithms. Static load balancing maintains a stable average response time, with minimal fluctuations caused by congestion, making it ideal for predictable and stable network environments. In contrast, dynamic load balancing shows greater variability in response times due to simulated jitter, reflecting its adaptability to changing network conditions. This variability improves dynamic load balancing during high congestion periods but results in more varied response times.

Table 3. Wi-Fi scanning example results

SSID	MAC address	Signal strength (dBm)	Encryption	Channel	Band (GHz)	Quality (%)	Width (MHz)
AdhuBabbu	A8:DA:0C:DB:5F:42	-47	WPA2 PSK[CCMP]	11	2.462	83	20
JioFiber-Xcrq5	B4:A7:C6:D6:44:5F	-60	WPA2 PSK[CCMP]	3	2.422	55	20
Vyshu&Krishna	54:AF:97:5B:00:7F	-64	WPA2 PSK[CCMP]	7	2.422	52	40
Aaruran_home	3C:52:A1:8D:09:42	-75	WPA2 PSK[CCMP]	8	2.447	13	40
Innovative_Freaks	5C:A6:E6:50:7F:89	-84	WPA PSK[CCMP]	11	2.462	13	20

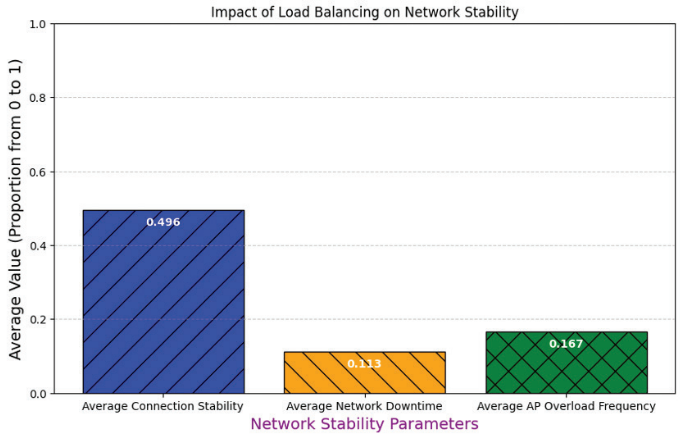

Figure 2. Impact of load balancing on network stability.

Overall, dynamic load balancing improves network stability by adjusting to real-time conditions, maintaining consistent performance under varying traffic loads and client connections. Additionally, its ability to distribute the load more evenly across access points can reduce the likelihood of any single AP becoming overloaded.

Figure 3 provides a long-term stability assessment of the load-balancing algorithms. The simulation results provide insights into how static and dynamic load-balancing algorithms perform over extended periods, offering a comprehensive view of their stability and adaptability. The plot highlights the relative advantages of each approach, with static load balancing offering consistent performance and dynamic load balancing providing adaptability to changing network conditions. This analysis helps in understanding the long-term implications of choosing either algorithm, making it clear that static load balancing is best for stable environments, while dynamic load balancing excels in dynamic and unpredictable conditions.

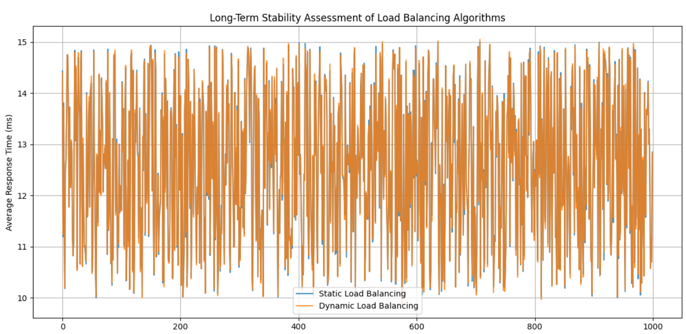

Figure 3. Long-term stability assessment of load balancing algorithms.

3.4 Client Distribution Dynamics

The study analysed client distribution among APs using real-time Wi-Fi scanning data to understand how load-balancing algorithms impact client distribution patterns. Static load balancing was found to distribute clients more uniformly across APs, achieving a uniform distribution of around 70-75 %. In contrast, dynamic load balancing adjusts client distribution based on real-time load, resulting in more efficient resource utilisation and improving this distribution by approximately 5 % - 10 %.

Figure 4 demonstrates the client distribution and load distribution between static and dynamic load balancing. Static load balancing achieves a uniform client distribution across APs, whereas dynamic load balancing enhances efficiency by adapting to real-time network conditions. This adaptability of dynamic load balancing is particularly beneficial in environments with variable network loads, as it optimises client distribution to enhance overall network performance.

The findings are supported by a mathematical model, detailed in Eqn. (4), which calculates client distribution based on each AP's signal strength and quality. The client distribution function ensures proportional client distribution according to AP performance metrics, thus enhancing overall network efficiency through dynamic load balancing[39].

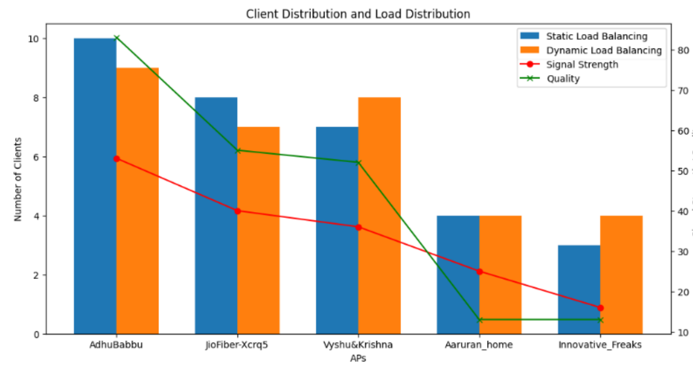

Figure 4. Client distribution and load distribution.

Client distribution function:

$$f(x) = \frac{S(x) \times Q(x)}{\sum_{i=1}^{n} S(i) \times Q(i)} \quad (4)$$

This analysis highlights the superiority of dynamic load balancing in optimising client distribution and underscores its potential to enhance network performance significantly.

3.5 Scalability and Performance Evaluation

The study evaluated the scalability and performance of ESP32-based load balancing by subjecting it to increasing workload conditions. Key metrics, including throughput scalability, response time under heavy loads, and AP resource utilisation, were measured to assess the system's performance comprehensively.

3.5.1 *Throughput Scalability*

Throughput scalability, calculated as the ratio of the number of successful requests to the total time of the workload test, was found to be approximately 5 % higher in dynamic load balancing compared to static load balancing[39]. This metric is crucial for understanding how well the network can handle increasing loads while maintaining high performance.

Throughput scalability calculation:

$$T(x) = \frac{Number of Successful Requests}{Total Time Taken} \quad (5)$$

Response time calculation:

$$R(x) = \frac{1}{n} \sum_{i=1}^{n} \left(\frac{Q(i)}{S(i)} \right) \quad (6)$$

Response time under heavy loads, computed as the average reciprocal of the signal strength and quality product for each AP, showed similar results for both static and dynamic load-balancing methods. This consistency indicates that both approaches can maintain reasonable response times even under significant load.

3.5.2 AP Resource Utilisation

Regarding AP resource utilisation, both static and dynamic load balancing demonstrated efficiency, with dynamic load balancing showing a slight edge of around 3-5 % improvement. This slight advantage highlights the effectiveness of dynamic load balancing in optimising resource use, especially under varying network conditions.

Throughput scalability and response time were calculated using Eqn. (5-6) mathematical models, emphasising the effectiveness of ESP32-based load balancing in handling increased workloads while maintaining optimal resource utilisation. These results underscore the network's ability to maintain performance under increasing loads and highlight the slight efficiency edge of dynamic load balancing. By evaluating these metrics, the study demonstrates that ESP32-based load balancing is effective in managing increased workloads, optimising resource utilisation, and maintaining network performance and responsiveness under various conditions. This comprehensive analysis provides valuable insights into the scalability and performance capabilities of ESP32-based networks, supporting their potential use in diverse environments.

3.6 Comparison with Existing Load Balancing Strategies

This section compares the performance of ESP32-based load balancing with traditional strategies such as round-robin and least-connection methods. The findings highlight the effectiveness and simplicity of the ESP32-based approach, which delivers comparable or superior performance metrics with reduced complexity. Table 4 presents a detailed comparison of performance metrics between ESP32-based load balancing and traditional methods. Figure 5 illustrates the comparison of static and dynamic load balancing methods.

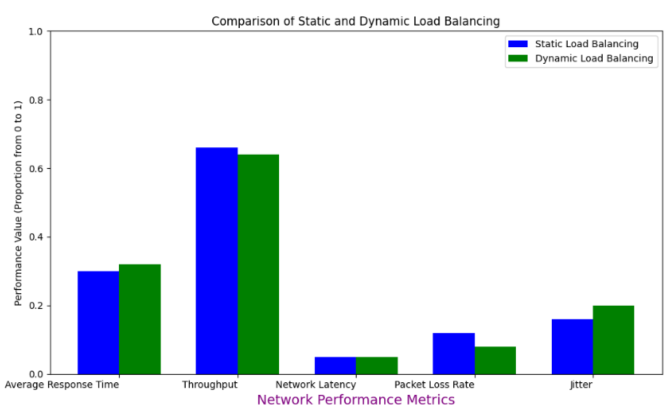

Figure 5. Comparison of static and dynamic load balancing.

Figure 6 depicts simulated network performance metrics before and after implementing load balancing, highlighting the efficiency gains of ESP32-based methods. The average response time for both ESP32 (Static) and ESP32 (Dynamic) is comparable to that of the round-robin method, with slight delays relative to the least-connection method. Specifically, ESP32 (Static) outperforms the least-connection method by 1.33 %, while ESP32 (Dynamic) shows an improvement of 6.25 %. In terms of throughput, ESP32 (Static) and ESP32 (Dynamic) outperform the least-connection method, with improvements of 2.12 % and 5.31 %, respectively. Additionally, ESP32-based load balancing significantly reduces network latency, outperforming the round-robin method by 83.61 %.

These results demonstrate significant advancements achieved through the simplified ESP32-based load-balancing approach. Despite its simplicity, this method shows high efficiency and offers performance enhancements comparable to more complex strategies like the Dynamic Wireless Load Controller (DWLC)[40]. The ESP32-based approach also eliminates the need for additional hardware, enhancing cost-effectiveness and practicality.

The findings reassure network engineers, IT professionals, and researchers in network optimisation about the feasibility and practicality of ESP32-enabled load distribution in real-world scenarios. The study validates the effectiveness of the dynamic least connection method, emphasising the superiority of the ESP32-driven load-balancing approach over more intricate methods. By delivering performance improvements without additional hardware or complexity, this approach offers a practical and accessible solution for optimising network performance across various environments.

3.7 Practical Applications

This ESP32-based load-balancing algorithm in defence networks is multifaceted. These algorithms ensure reliable and efficient real-time data transmission in tactical environments, where timely information is crucial for decision-making. By optimising the Wi-Fi range, defence personnel can maintain communication over larger areas, reducing the need for additional infrastructure and ensuring continuous connectivity in remote locations. Dynamic load balancing plays a significant role in resource optimization by distributing the network load evenly across multiple AP, preventing any single AP from becoming a bottleneck. This is particularly beneficial in high-density areas where multiple devices are connected simultaneously. Furthermore, the ability to dynamically adjust to varying network loads ensures that the network

Table 4. Evaluating ESP32 against conventional load-balancing techniques

Metric	ESP32 (Static)	ESP32 (Dynamic)	Round-robin	Least connection
Average response time (s)	0.30	0.32	0.30	0.296
Throughput (clients/s)	3.3	3.2	3.33	3.37
Network latency (s)	0.05	0.05	0.299	0.299

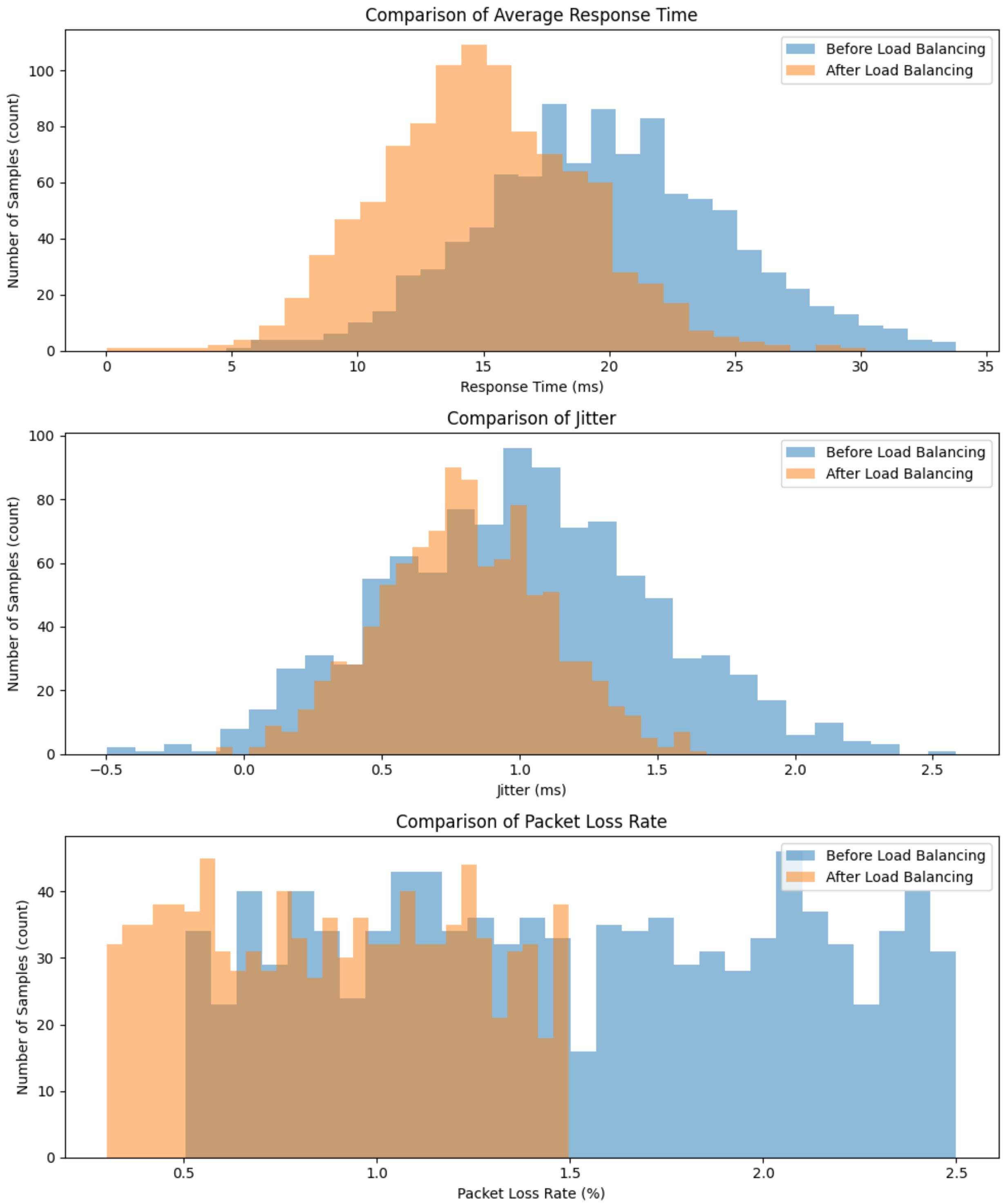

Figure 6. Simulated network performance metrics before and after load balancing.

can scale efficiently, accommodating more devices without compromising performance. Improved network performance is another key advantage, with the static algorithm's ability to lower response time and jitter enhancing the quality of service for applications requiring real-time data, such as video surveillance and remote control of unmanned vehicles. Additionally, reducing packet loss ensures that critical data is transmitted accurately, maintaining the integrity of communication in defence operations.

4. CONCLUSION

The comparative analysis between static and dynamic load-balancing algorithms reveals distinct impacts on network performance. Static load balancing demonstrates a slight advantage with a 5 % lower average response time and 3 % higher throughput than dynamic load balancing, while both methods exhibit comparable network latency. Dynamic load balancing effectively reduces AP overload frequency by 20 % during peak periods, enhancing network stability and optimising client distribution in real-time, thereby improving overall efficiency by 5 % - 10 %.

Key findings underscore static load balancing's strengths in stable environments, delivering lower response times and higher throughput. Both methods maintain high connection stability and minimal network downtime, with dynamic load balancing excelling in adapting to real-time conditions and reducing AP overload. ESP32-based load balancing proves scalable, achieving 5 % higher throughput scalability and 3-5 % improved AP resource utilisation compared to static methods. Compared to traditional strategies, ESP32-based load balancing consistently improves performance metrics by 1.33 % to 83.61 %, while maintaining simplicity and practicality.

Future analysis should focus on advancing dynamic load-management algorithms, integrating AI techniques, and exploring edge computing capabilities while addressing critical concerns such as security, privacy, and IoT device optimisation. Standardising network optimisation practices will further facilitate the practical application of these findings across diverse operational environments.

In conclusion, ESP32-based load balancing enhances network efficiency, stability, and scalability while maintaining simplicity, making it a practical tool for modern network optimization strategies.

REFERENCES

1. Sirait, J.E.; Alrasyid, H. & Soraya, N.A. Strengthening the defense industry's independence through the internet of things in the manufacturing sector: A review. *Int. J. Sci. Technol. Manage.*, 2023, **4**(2), 335–340. doi: 10.46729/IJSTM.V4I2.764.
2. Sharma, J. & Mehra, P.S. Secure communication in IOT-based UAV networks: A systematic survey. *IEEE Internet Things J*., 2023, **23**, 100883. doi: 10.1016/J.IOT.2023.100883.
3. Pahlavan, K. & Krishnamurthy, P. Evolution and impact of Wi-Fi technology and applications: A historical perspective. *Int. J. Wirel. Inf. Netw.*, 2021, **28**(1), 3–19. doi: 10.1007/S10776-020-00501-8.
4. Bajracharya, R.; Shrestha, R.; Hassan, S.A.; Jung, H. & Shin, H. 5G and beyond private military communication: Trend, requirements, challenges and enablers. *IEEE Access*, 2023, **11**, 83996–84012. doi: 10.1109/ACCESS.2023.3303211.
5. Salahdine, F.; Han, T. & Zhang, N. Security in 5G and beyond: Recent advances and future challenges. *IEEE Secur. Priv.*, 2023, **6**(1), e271. doi: 10.1002/SPY2.271.
6. Pervez, F.; Qadir, J.; Khalil, M.; Yaqoob, T.; Ashraf, U. & Younis, S. Wireless technologies for emergency response: A comprehensive review and some guidelines. *IEEE Access*, 2018, **6**, 71814–71838. doi: 10.1109/ACCESS.2018.2878898.
7. Marshall, A.; Wilson, C.A. & Dale, A. Telecommunications and natural disasters in rural Australia: The role of digital capability in building disaster resilience. *J. Rural Stud*., 2023, **100**, 102996. doi: 10.1016/J.JRURSTUD.2023.03.004.
8. Rukaiya, R.; Khan, S.A.; Farooq, M.U. & Matloob, I. Communication architecture and operations for SDR-enabled UAVs network in disaster-stressed areas. *Ad Hoc Netw.*, 2024, **160**, 103506. doi: 10.1016/J.ADHOC.2024.103506.
9. Soto-Vergel, A.J.; Velez, J.C.; Amaya-Mier, R. & Pardo, M. Transforming ground disaster response: Recent technological advances, challenges, and future trends for rapid and accurate real-world applications of survivor detection. *Int. J. Disaster Risk Sci.*, 2023, **98**, 104094. doi: 10.1016/J.IJDRR.2023.104094.
10. Alselek, M.; Alcaraz-Calero, J.M. & Wang, Q. Dynamic AI-IoT: Enabling updatable AI models in ultralow-power 5G IoT devices. *IEEE Internet Things J.,* 2024, **11**(8), 14192–14205. doi: 10.1109/JIOT.2023.3340858.
11. Xing, C. & Li, F. Unlicensed spectrum-sharing mechanism based on wi-fi security requirements implemented using device to device communication technology. *IEEE Access*, 2020, **8**, 135025-135036. doi: 10.1109/ACCESS.2020.3011955
12. Gill, S.S.; Wu, H.; Patros, P.; Ottaviani, C.; Arora, P.; Pujol, V.C.; Haunschild, D.; Parlikad, A.K.; Cetinkaya, O.; Lutfiyya, H.; Stankovski, V.; Li, R.; Ding, Y.; Qadir; J.; Abraham, A.; Ghosh, S.K.; Song, H.H.; Sakellariou, R.; Rana, O.; Rodrigues, J.J.P.C. & Buyya, R. Modern computing: Vision and challenges. *Tel. In. R.,* 2024, **13**, 100116. doi: 10.1016/J.TELER.2024.100116.
13. Yarinezhad, R. & Ekici, E. A novel scheduling algorithm for LTE on unlicensed bands to ensure fair coexistence with *Wi-Fi. Comput. Netw.,* 2024, 241, 110232. doi: 10.1016/J.COMNET.2024.110232.
14. Al Reshan, M.S.; Syed, D.; Islam; N.; Shaikh, A.; Hamdi, M. & Elmagzoub, M.A. A fast converging and globally optimised approach for load balancing in cloud computing. *IEEE Access*, 2023, **11**, 11390–11404. doi: 10.1109/ACCESS.2023.3241279.
15. Khaleel, M.I. Region-aware dynamic job scheduling and resource efficiency for load balancing based on adaptive chaotic sparrow search optimisation and coalitional game in cloud computing environments. *J. Netw. Comput. Appl.,* 2024, 221, 103788. doi: 10.1016/J.JNCA.2023.103788.
16. Zhou, J. Comparative analysis of metaheuristic load balancing algorithms for efficient load balancing in cloud computing. *J. Cloud Comput.,* 2023, **12**(1), 1–21.
17. Foomeshi, Y.R. A review of scheduling and resource allocation algorithms with a load balancing approach in cloud computing. *Majlesi J. Telecommun. Syst.,* 2023, 12(2), 95–103. doi: 10.30486/MJTD.2023.1981169.1028.
18. Zhu, X.; Yao, W. & Wang, W. Load-aware task migration algorithm toward adaptive load balancing in edge computing. *Future Gener. Comput. Syst.,* 2024, **157**, 303–312. doi: 10.1016/J.FUTURE.2024.03.014.

19. Tripathy, S.S.; Mishra, K.; Roy, D.S.; Yadav, K.; Alferaidi, A.; Viriyasitavat, W.; Sharmila, J.; Dhiman, G. & Barik, R.K. State-of-the-art load balancing algorithms for mist-fog-cloud assisted paradigm: A review and future directions. *Arch. Comput. Methods Eng.,* 2023, **30**(4), 2725–2760.
doi: 10.1007/S11831-023-09885-1.
20. Himeur, Y.; Sayed, A.N.; Alsalemi, A.; Bensaali, F. & Amira, A. Edge AI for internet of energy: Challenges and perspectives. *IEEE Internet Things J.,* 2024, **25**, 101035.
doi: 10.1016/J.IOT.2023.101035.
21. Belli, D.; Barsocchi, P. & Palumbo, F. Connectivity standards alliance matter: State of the art and opportunities. *IEEE Internet Things*, 2024, **25**, 101005.
doi: 10.1016/J.IOT.2023.101005.
22. Hadi, H.J.; Cao, Y.; Nisa, K.U.; Jamil, A.M. & Ni, Q. A comprehensive survey on security, privacy issues and emerging defense technologies for UAVs. *J. Netw. Compu. Appl.,* 2023, **213**, 103607.
doi: 10.1016/J.JNCA.2023.103607.
23. Kokila, M. & Reddy, S. Authentication, access control and scalability models in internet of things security–A review. *Cyber Secu. & Appl.,* 2025, **3**, 100057.
doi: 10.1016/J.CSA.2024.100057.
24. Narasimha Swamy, S.; Anna, D.M.; Vijayalakshmi, M.N. & Kota, S.R. Enabling lightweight device authentication in message queuing telemetry transport protocol. *IEEE Internet Things J.,* 2024, **11**(9), 15792–15807.
doi: 10.1109/JIOT.2024.3349394.
25. Tran, K.T.M.; Pham, A.X.; Nguyen, N.P. & Dang, P.T. Analysis and performance comparison of IoT message transfer protocols applying in real photovoltaic system. *Int. J. Networked Distrib. Comput.,* 2024, **12**(1), 131–143.
doi: 10.1007/S44227-024-00021-4/TABLES/3.
26. Mehraban, S. & Yadav, R.K. Traffic engineering and quality of service in hybrid software-defined networks. *China Commun.,* 2024, **21**(2), 96–121.
doi: 10.23919/JCC.FA.2022-0860.202402.
27. Ahmed, S.T.; Ahmed, A.A.; Annamalai, A. & Chouikha, M.F. A scalable and energy-efficient lorawan-based geofencing system for remote monitoring of vulnerable communities. *IEEE Access,* 2024, **12**, 48540–48554.
doi: 10.1109/ACCESS.2024.3383778.
28. Sharma, T.P.; Solanki, A.; Jain, T.; Malhotra, A. & Bhutani, M. Wi-Fi based quadcopter drone with battery monitoring and optimisation using crazyflie platform. *Int. J. Inf. Technol.,* 2024, **16**(3), 1887–1898.
doi: 10.1007/S41870-023-01639-3/TABLES/3.
29. Bruschi, F.; Zanghieri, M.; Terziani, M. & Sciuto, D. Decentralised updates of IoT and edge devices. *Lect. Notes Data Eng. Commun.,* 2024, **203**, 161–170.
doi: 10.1007/978-3-031-57931-8_16.
30. Mehraban, S. & Yadav, R.K. Traffic engineering and quality of service in hybrid software-defined networks. *China Commun.,* 2024, **21**(2), 96–121.
doi: 10.23919/JCC.FA.2022-0860.202402.
31. Swain, S.R.; Saxena, D.; Kumar, J.; Singh, A.K. & Lee, C.N. An intelligent straggler traffic management framework for sustainable cloud environments. *IEEE Trans. Sustain. Comput.,* 2024.
doi: 10.1109/TSUSC.2024.3393357.
32. Lin, W.; Lin, J.; Peng, Z.; Huang, H.; Lin, W. & Li, K. A systematic review of green-aware management techniques for sustainable data center. *Sustain. Comput. Inform. and Syst.,* 2024, **42**, 100989.
doi: 10.1016/J.SUSCOM.2024.100989.
33. Truong, H.; Jaisinghani, D.; Jain, S.; Sinha, A.; Ko, J.G. & Balan, R. Tracking people across ultra-populated indoor spaces by matching unreliable Wi-Fi signals with disconnected video feeds. *Pervasive Mob. Comput.,* 2024, **97**, 101860.
doi: 10.1016/J.PMCJ.2023.101860.
34. Rosele, N.; Mohd Zaini, K.; Ahmad Mustaffa, N.; Abrar, A.; Fadilah, S.I. & Madi, M. Digital transformation in wireless networks: A comprehensive analysis of mobile data offloading techniques, challenges, and future prospects. *J. King Saud Univ. - Comput. Inf. Sci.,* 2024, **36**(5), 102071.
doi: 10.1016/J.JKSUCI.2024.102071.
35. Itagi, M.V.; Prasad, D.G.; Kumar, K.P.S.; R, S. & Ashreetha, B. Performance analysis of energy efficiency and security solutions of internet of things protocols. *Int. J. Electr. Electron. Res.,* 2023.
36. Loba, T.; Konate, G.; Batiebo, M.R. & Bita, R.D.J. Development of a web server load balancing system. *Int. J Innov. Appl. Stud.,* 2024, **41**(3), 859–866. http://www.ijias.issr-journals.org/.
(Accessed on 12 June 2024).
37. Prity, F.S. & Hossain, Md. M. A comprehensive examination of load balancing algorithms in cloud environments: A systematic literature review, comparative analysis, taxonomy, open challenges, and future trends. *Iran J. Comput. Sci.,* 2024, 1–36.
doi: 10.1007/S42044-024-00183-Y.
38. Jangra, A. & Mangla, N. An efficient load balancing framework for deploying resource scheduling in cloud based communication in healthcare. *Meas.: Sens.* 2023, **25**, 100584.
doi: 10.1016/J.MEASEN.2022.100584.
39. Jadon, S.; Kannan, P.K.; Kalaria, U.; Varsha, K.R.; Gupta, K. & Honnavalli, P.B. A comprehensive study of load balancing approaches in real-time multi-core systems for mixed real-time tasks. *IEEE Access*, 2024, **12**, 53373–53395.
doi: 10.1109/ACCESS.2024.3388291.
40. Chandra, S.; Arya, R. & Singh, M.P. Intelligent resource management in 5G/6G network by adopting edge intelligence for higher education systems. *Adv. Electr. Electron. Eng.,* 2024, **8**, 100517.
doi: 10.1016/J.PRIME.2024.100517.

CONTRIBUTORS

Ms A.K. Kowsalyadevi is a PhD Scholar in the Department of Electronics and Communication Engineering at PSG College of Technology, Coimbatore. She is pursuing research in Quality of Service (QoS) improvement in Wi-Fi and LoRa networks. Her

research focuses on Wireless Networking, Ad-Hoc networks, and the Internet of Things (IoT). Her responsibilities in the present study included concept and design development, methodology development, and primary research.
Her contribution to data collection, analysis, and manuscript drafting and revision was significant.

Dr G. Umamaheswari is an Associate professor of Electronics and Communication Engineering at PSG College of Technology, Coimbatore. Her research interests include: Communication systems, wireless systems, and wireless security.
In the current study she provided mentorship and oversight, contributed to the theoretical framework, validated the research findings, and assisted in the critical revision of the manuscript, ensuring it met academic standards.