Journal of Techniques, ISSN: 2708-8383, Vol. 7, No. 4, 2025, Pages 37-42
DOI: https://doi.org/10.51173/jt.v7i4.2768

JOURNAL OF TECHNIQUES

Journal homepage: http://journal.mtu.edu.iq

RESEARCH ARTICLE – ENGINEERING (MISCELLANEOUS)

Protocol Efficiency and Resource Utilization in VPN Technologies: A Comparative Analysis of OpenVPN and WireGuard

Sura Ghanim Hussein[1*], Syed Muhammad Fasih Ur Rehman[2]

[1]Department of Mechatronics Engineering Techniques, Engineering Technical College - Baghdad, Middle Technical University, Baghdad, Iraq

[2]Faculty of Engineering, The Islamia University of Bahawalpur, Bahawalpur, Pakistan

[*] Corresponding author E-mail: sura@mtu.edu.iq

Article Info.

Article history:

Received
31 October 2025

Revised
18 December 2025

Accepted
27 December 2025

Published
31 December 2025

Abstract

Virtual Private Networks are essential for securing Internet communication, but the selection of a VPN protocol can substantially affect overall network performance. Nonetheless, typical VPN protocols (e.g., OpenVPN) are too secure relative to their worst-case latency and computational overhead. New-generation VPN protocols, such as WireGuard, are promising to offer greater efficiency due to a lightweight cryptographic design. This paper depicts a comparison trial of OpenVPN and WireGuard throughput under controlled cooperation. Download and upload speeds, latency, and CPU and RAM utilization were measured using a Speedtest application on a Windows 11 platform. The results reveal that WireGuard achieved network performance approaching the baseline, with 50.4 Mbps download speeds, low latency (8.01 ms), and low CPU consumption (3%). In contrast, OpenVPN decreased download speeds to 22.1 Mbps and was accompanied by high latency (160.38 ms). The results imply that WireGuard is a valid choice for latency-sensitive and throughput-demanding use cases. In contrast, OpenVPN is appropriate for use cases that focus on interoperability and established security.

Publisher: Middle Technical University

Keywords: VPN; OpenVPN; Wire Guard; Performance Evaluation; Latency.

1. Introduction

VPNs (Virtual Private Networks) are commonly used to ensure privacy and integrity over public networks [1]. With a traditional VPN, you can determine which encryption ciphers are supported by both your client and the server [2, 3]. However, most of these protocols are widely criticized for their performance overhead, including low bandwidth, high latency, stringent control requirements, and computational costs [4, 5].

In recent years, several next-generation VPN techniques (e.g., WireGuard) have been proposed and offer low-overhead secure connections with minimal performance degradation [6, 7]. As have seen, there are limitations and drawbacks of the traditional VPN, and this has stimulated research on next-generation VPNs. Two primary deficiencies in observational studies. First, many studies test VPNs in specific environments, such as IoT platforms, controlled cloud systems, or attack scenarios, but do not examine their performance on real servers in everyday use. Second, earlier comparisons often lack reproducible methods, public configuration files, or repeated tests to ensure reliable results. As a result, still lack a clear understanding of how OpenVPN and WireGuard perform in realistic, end-to-end setups that others can easily repeat.

To overcome those gaps, this study makes a comparison between OpenVPN; which is a traditional VPN protocol and the next-generation protocol, WireGuard using real servers and open-access configurations. The main contributions of this work are summarized as:

- Performing real-world tests using public server configurations, making it easy to repeat the experiments by other researchers.
- Thorough and reliable measurements are collected including latency, throughput, CPU usage, and memory tests by repeating the tests several times.
- A benchmark method is used that is easy and low-cost, without the need to use commercial VPN services or large cloud systems.
- The results are analyzed to determine which protocol is better for high-throughput and latency-sensitive applications.

The study provides comparisons that are reproducible, easy-to-follow and also offers practical recommendations for researchers and practitioners examining next-generation VPN technology.

Nomenclature & Symbols			
VPNs	Virtual Private Networks	CPU	Central Processing Unit
IoT	Internet of Things	SD	Standard Deviation
RSA	Rivest–Shamir–Adleman	UDP	User Datagram Protocol
RAM	Random Access Memory	ICMP	Internet Control Message Protocol

2. Literature Review

Virtual Private Networks (VPN) technology has faced a rapid development over the last two decades [8]. During that time, conventional VPN technologies, including IPsec and OpenVPN, have been at the forefront, with a wide range of applications [9, 10]. However, recent studies have focused on efficiency enhancement. It is such demand, especially in time-sensitive and resource-constrained scenarios, that gave rise to the birth of next- generation VPN protocols such as WireGuard. [11, 12] This section covers works associated with VPN performance.

2.1. Traditional VPN protocols

Because they provide strong encryption and operate at the network layer standard VPN protocols are widely used [13]. For example, in [14], an IPSec-based VPN is designed to ensure security to smart home networks and demonstrates strong performance in preserving confidentiality and integrity in IoT scenarios. However, the operation of IPsec and similar protocols entails significant computational overhead, imposing relatively high cryptographic processing and traffic overheads, thereby increasing packet latency and making them unsuitable for several currently available high-performance applications.

2.2. Next-generation VPNs in IoT and resource-constrained environments

A study compared WireGuard, OpenVPN, and IPsec for IoT devices in terms of throughput, latency, and jitter. The experimental results show that WireGuards more efficient in resource-constrained environments [15]. These results show that WireGuard is particularly well-suited for use cases in which timing and limited computational resources are of concern.

2.3. Safety and adaptability in adversarial Environments

If there is no such central security concern, this application does not guide relative performance. Adversarial testing of DoS attacks on a set of VPN implementations was done in [16]. It has been shown that WireGuard performs better but may also be worse at becoming bogged down under load than OpenVPN. This demonstrates the need to balance increased performance with network resistance to attacks.

2.4. Comparative performance evaluations

There have been a couple of studies in which WireGuard has been benchmarked against legacy VPNs in the laboratory under high packet volume. WireGuard was compared with strongSwan and OpenVPN on a 1 Gbps testbed to evaluate performance and efficiency, as well as connection setup time and throughput, with improvements across the network. The findings demonstrate that WireGuard is significantly more efficient and incurs substantially less computational overhead [17]. The first, more advanced methods [18] argue that WireGuard's improved performance is due to its smaller codebase and careful multicore handling.

2.5. Summary of trends

Recent developments indicate that WireGuard VPN and several other next-generation technologies are advancing, as clients can achieve higher speeds and lower latency more easily. However, once security is involved, or when long-term stability and predictable use are required for protocols such as IPsec and OpenVPN, they become much more dominant here [19, 20].

3. Methodology

To conduct a formal comparison between the two VPNs, OpenVPN and WireGuard, employed a counterbalanced within-subjects experimental design. All tests were performed on a Windows 11 Pro machine under the same network and system conditions, connected to a broadband Internet connection, with an Intel Core i7 processor, 8 GB of RAM, and a 100 Mbps downstream link.

To improve reliability, conducted three trials per task and condition for descriptive statistics. calculated the average and standard deviation (SD) of throughput, latency, CPU, and RAM to calculate variation.

3.1. Experimental setup and independent variables

VPNBook OpenVPN and WireGuard clients were also configured, and public profiles were created for them. used the same server endpoint for all tests, which verified before running the experiments. Key parameters are the VPN protocol (OpenVPN or WireGuard) and network state (raw/no VPN, OpenVPN up, or WireGuard up).

Release notes and protocol parameters were shared at the time that changes were made for replication. The tests were conducted using OpenVPN Client 3 with AES-256-CBC encryption, SHA-256 authentication, 2048-bit RSA certificates, and the port settings provided by VPNBook (UDP/1194 or similar). For WireGuard, tests were performed using v0.5.3 of the WireGuard Client, and the encryption/peer options were configured to match VPNBook's defaults (ChaCha20-Poly1305). DERPANET was using the client default (usually 1280-1420 auto-negotiate interface). All configuration files and parameters were identical across repeated runs.

3.2. Measurement metrics and dependant variables

The variables in this study were dependent variables, including throughput (download and upload), latency, and system resource consumption. Throughput is tested, and the same server is always used for consistency. Ping latency was measured using the Windows ping tool with a 32-byte ICMP packet and 10 echo requests per test. The system was monitored using Windows Task Manager (Windows 11 Pro), with CPU and

memory utilization recorded from the Performance tab during each experiment. All tools, versions, and settings are kept identical in every case to ensure reproducibility of the results.

3.3. Control conditions and environmental factors

Also conducted the tests during non-peak hours and opened no other applications (e.g., cloud-sync service in the background user-space, crash clients, Windows Update). The hardware, as well as operating systems, VPN settings, MTUs, and software versions, were all the same throughout all tests. reconnected to the VPN before each test so that the endpoint IP address wasn't blocked.

3.4. Testing procedure

Three scenarios were evaluated:

- Baseline (no VPN)
- OpenVPN active, and
- WireGuard active

In each case, employed the same testing methodology, server location, and measurement instruments. Did not measure jitter, packet loss, or packet reordering in detail; addressing these is left for future work.

3.5. Repetition and data aggregation

Each measurement was repeated three times per scenario. Results were averaged, tabulated, and visualized with bar charts for comparison. Mean values represented overall performance, and repeated trials helped mitigate transient fluctuations in public VPN server load. This paper shows a preliminary case study based on a single client device, a single public VPN, and three repetitions per scenario.

4. Results and Discussion

4.1. Results

The tests were performed under three network settings: No VPN, OpenVPN, and WireGuard. The performance measures are presented in Table 1, and the results of each setting are shown in Figs. 1-3.

Numerical values were presented both as a bar chart and in a Table to show optimal results. Visual charts enable trend identification, and Tables provide a comprehensive pro contract digest for the technical reviews. Tables 2– 6 present the mean and standard deviation of the performance metrics being compared for each scenario: Baseline, OpenVPN, and WireGuard. These Tables enable detailed comparisons and indicate the extent to which the results varied across repeated measurements.

Table 1. Performance metrics for different VPN protocols

Test Condition	Download Speed (Mbps)	Upload Speed (Mbps)	Latency (ms)	CPU Usage (%)	RAM Usage (MB)
No VPN	45.17	22.79	5.76	7	86
OpenVPN	22.1	11.3	160.38	17	90
WireGuard	50.4	33.08	8.01	3	85

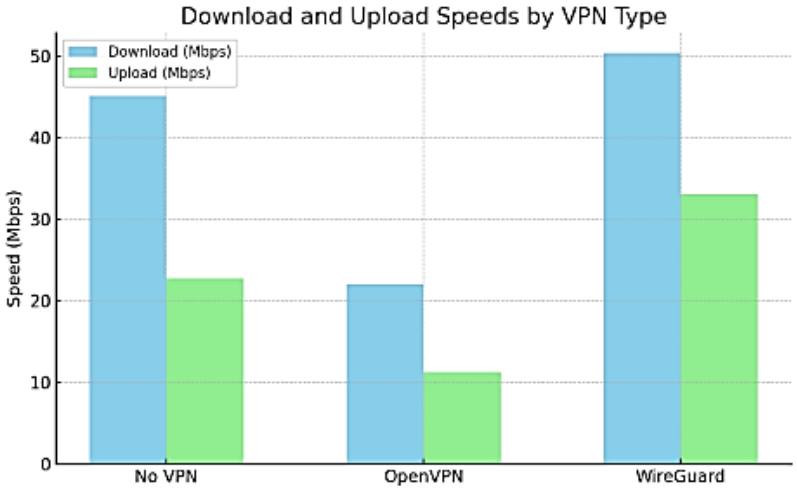

Fig. 1. Measurements of download and upload speed

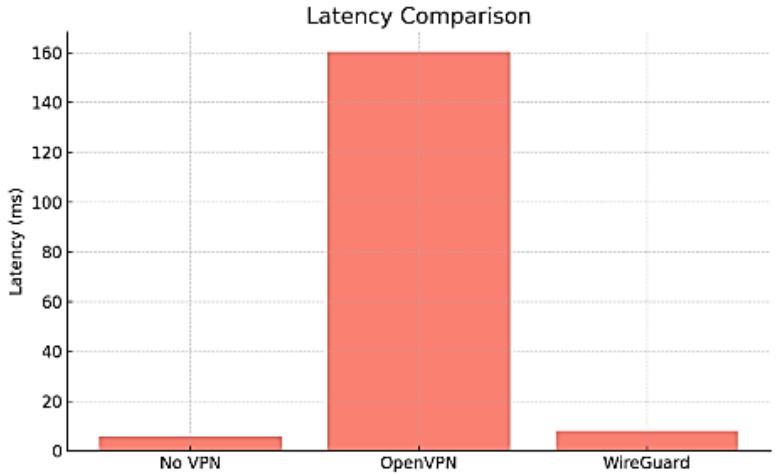

Fig. 2. Measurements of Latency Speed

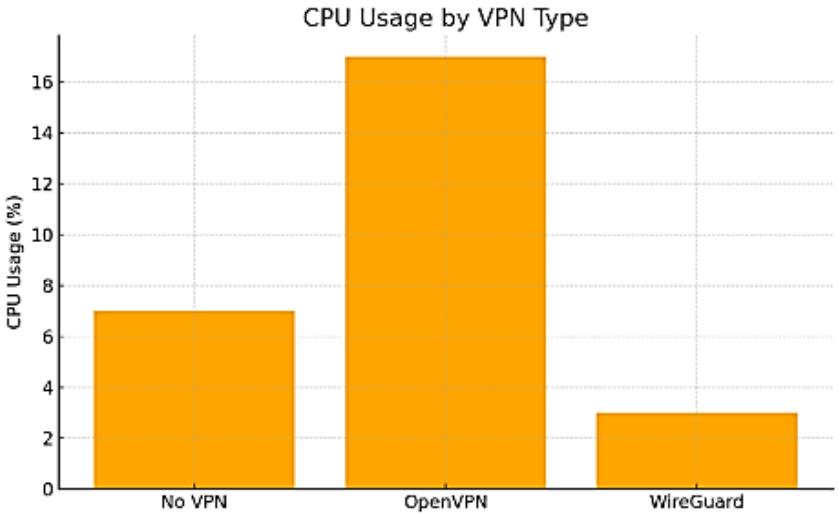

Fig. 3. Measurements of CPU usage

Table 2. Network throughput-mean standard deviation across the three trials

Scenario	Download (Mbps)	Upload (Mbps)
Baseline (No VPN)	45.17 ± 1.2	22.79 ± 06
OpenVPN	22.10 ± 1.6	11.30 ± 0.9
WireGuard	50.40 ± 1.8	33.08 ± 1.2

Table 3. Latency performance-mean standard deviation of ICMP (ms)

Scenario	Latency (ms)
Baseline (No VPN)	5.76 ± 0.4
OpenVPN	160.38 ± 5.0
WireGuard	8.01 ± 0.6

Table 4. CPU utilization- mean standard deviation (%) during active throughput tests

Scenario	CPU Usage (%)
Baseline (No VPN)	7.0 ± 0.8
OpenVPN	17.0 ± 1.5
WireGuard	3.0 ± 0.4

Table 5. RAM utilization-mean standard deviation during active throughput tests (MB)

Scenario	RAM Usage (MB)
Baseline (No VPN)	86 ± 3
OpenVPN	90 ± 4
WireGuard	85 ± 3

Table 6. Summary of performance differences

Metric	OpenVPN	WireGuard
Download Throughput	−51.1%	+11.6%
Upload Throughput	−50.4%	+45.2%
Latency	+2684%	+39.1%
CPU Usage	+142.8%	-57.1%
RAM Usage	+4.6%	-1.1%

4.1.1. OpenVPN

- Download and upload speeds are slower as OpenVPN's UDP is heavily encrypted and user-space processed.
- CPU is also running at 17% because of the encryption is strong and takes a lot of computing.
- The latency is significantly higher at 160 ms, as expected, since OpenVPN has greater overhead.

4.1.2. WireGuard

- Speeds are similar to or even faster than the reference (50.4 Mbps versus 45.17 Mbps). This is possible if routes are optimized.
- It has a low latency of 8 ms, is very light on CPU usage at 3%, and approximately matches the WireGuard lightweight design.

WireGuard's speed, low latency, and other characteristics make it a great choice for meeting modern VPN demands. During test, it achieved a download speed of 50.4 Mbps and an upload speed of 33.08 Mbps — even faster than the baseline, attributable to improved routing. By contrast, OpenVPN still had only 22.1 Mbps download and 11.3 Mbps upload speeds.

OpenVPN was also characterized by its considerable latency (160.38 ms), whereas WireGuard exhibited a slight but noticeable difference in latency (8.01 ms and 5.76 ms for No VPN). This was a significant difference in latency measurements.

On the client side, OpenVPN (in UDP mode) consumed the most of the CPU at 17%, due to heavy encryption and packet processing in user space. By comparison, WireGuard was lighter still, consuming 3 percent CPU and 85 MB of RAM.

The numbers confirm the allegation that WireGuard is a generation faster than OpenVPN:

- Network throughput: When rolling Transfer Flow dynamically onto WireGuard, the network throughput on WireGuard is better than the version using local disk, and a 50.4 megabits per second (Mbps) transfer rate for download traffic and a 33.08 Mbps transfer rate for upload traffic are achieved, and it was a little bit higher than baseline since optimized the route. Additional file. In UDP mode, OpenVPN's throughput was reduced by ~50%, which attribute to increased processing overhead from encryption and the tunneling layer.
- Latency: OpenVPN was slow (160 ms), which can be a nuisance for real-time uses. WireGuard exhibited a latency of 8 ms, which exceeded the lower latency boundary (5 ms).
- Resource usage: OpenVPN consumed more CPU at 17%, whereas WireGuard utilized just 3%, or less than half the amount—it's less of a task to deploy on devices featuring slower processors.

4.2. Discussion

These are comparable to findings from previous studies on WireGuard, which is due to its simple cryptography and low protocol overhead. Why? It has low latency and minimal speed loss, making it well-suited for bandwidth-intensive and real-time applications such as video calls, online gaming, and VoIP. On the negative side, OpenVPN is less favorable for sensitive-delay applications; there is more overhead due to encryption, and context switching is performed more frequently, as well as wrapper-added layers in UDP mode. By contrast, WireGuard beat the baseline (in terms of download speed); one could speculate that in this specific case, traffic was provisioned more efficiently through the VPN. This shows that, in certain scenarios, next-generation VPNs can improve performance rather than degrade it. In general, WireGuard appears to provide a better balance between security and ease of implementation. Individuals seeking to adopt a trusted open-source VPN can review these findings. This study considered throughput, latency, and CPU/RAM usage on bare metal only. Beyond this, did not include any other network quality metrics, such as jitter, packet loss, or packet reordering, all of which are harmful for time-sensitive applications such as VoIP, video, and gaming. Our comments regarding conditioning response time and suitability for real-time applications are conditional. Also had to rely on a public VPNBook server, and the data may have been modified by server load (saturation), routing changes, or other factors that induced load fluctuations. As a result, the findings of this study should be considered to derive from a partial rather than a complete evaluation. In the future, intend to test per-packet timing and loss, particularly in a realistic environment. If feasible, we'll determine jitter and loss statistics from packet captures and yield confidence intervals when comparing with protocols.

5. Conclusion and Future Work

The results of the comparison indicate that WireGuard has outperformed OpenVPN in terms of speed, latency, and resource consumption, making it a pretty good option for new applications that require low-latency and high-throughput connections. OpenVPN is very slow, and that's fine for cases when you need to provide a long-term compliance audit. After using WireGuard, if you've used bandwidth and remain uncomfortable with it, don't panic. But OpenVPN is the more dependable option for those who value compatibility and experience, even if…

Note that this study did not consider IoT or embedded devices. For that reason, all statements about how well-suited the protocol is for these should be treated with some suspicion, and they should be checked experimentally with a less powerful device.

This data must be interpreted in the context of a single-device, limited-scope study.

There are several possible extensions of this work that warrant further investigation in future studies. Performance testing must include mobile devices and devices with limited resources, as the majority of users access VPNs on their smartphones and Tablets. These devices have lower-

CPU performance, different network settings, and, with respect to system optimization, exhibit performance differences relative to a desktop or server. The study of these differences is valuable to researchers who adopt VPN on various platforms.

New or hybrid VPN methods should be investigated in future work, including experiments with new designs such as WireGuard with post-quantum cryptography; multi-hop or cascading VPN systems; and adaptable VPNs that adjust their security and performance in real time. This analysis of those trade-offs will help us construct VPNs that remain fast and secure in the face of ever-changing threats. As future work, a comprehensive threat modeling and security analysis should be conducted. Although focus on performance in this paper, it would be interesting to compare how each protocol fares in real-world attacks such as phishing, replay attacks, packet injection, and manipulation during key exchange, to clarify the trade-off between efficiency and security. In general, these research directions provide a roadmap for enhancing technical understanding and improving the practical applications of VPN security protocols.

Acknowledgment

The authors gratefully acknowledge the Engineering Technical College–Baghdad, Middle Technical University, Baghdad, Iraq, for providing support and laboratory facilities that significantly contributed to the successful completion of this study.

References

[1] Z. Liu, "Application and Security Analysis of Virtual Private Network (VPN) in Network Communication," Academic Journal of Computing & Information Science, vol. 6, no. 11, pp. 52–59, 2023, doi: 10.25236/AJCIS.2023.061108.
[2] J. A. Quimpo, Cloud-Based VPN Replacement, Master's thesis, Theseus (Metropolia UAS), 2024.
[3] V. S. K. Rajak, "Secure remote network: A comprehensive study to secure organization's remote network and setup secure VPN," Master's thesis, National College of Ireland (NCI), Dublin, Ireland, 2024.
[4] A. F. Gentile, D. Macrì, F. De Rango, M. Tropea, and E. Greco, "A VPN Performances Analysis of Constrained Hardware Open Source Infrastructure Deploy in IoT Environment," Future Internet, vol. 14, no. 9, Art. no. 264, 2022, doi: 10.3390/fi14090264.
[5] A. W. De la Cadena Ramos, Multipath Routing on Anonymous Communication Systems: Enhancing Privacy and Performance, Ph.D. dissertation, University of Luxembourg, Luxembourg, 2021.
[6] N. D. Majeed, A. J. Al-Askery, F. S. Hasan, and S. Abood, "A Survey on Steganography and Image Encryption Techniques", EETJ, vol. 2, no. 1, pp. 11–24, Jan. 2025.
[7] Hayder Jalo and Mohsen Heydarian, "A Hybrid Technique Based on RF-PCA and ANN for Detecting DDoS Attacks IoT", IJDS, vol. 1, no. 1, pp. 28–41, Jun. 2024.
[8] B. H. Hameed and Z. A. Saleh, "Progression of the Protection Networking System Depending on International Virtual Private Network," International Journal of Safety and Security Engineering, vol. 13, no. 5, p. 863, 2023, doi: 10.18280/ijsse.130510.
[9] D. Zela, B. Mema, and K. Zela, "VPN VIRTUAL PRIVATE NETWORK APPLICATIONS IN DATA PREDICTION," Smart Cities and Regional Development (SCRD) Preprints, vol. 1, no. 1, Dec. 2024.
[10] S. C. Forbacha and J. J. A. Agwu, "Design and Implementation of a Secure Virtual Private Network Over an Open Network (Internet)," American Journal of Technology, vol. 2, no. 1, pp. 1–36, 2023, doi: 10.58425/ajt.v2i1.134.
[11] J. Anyam, R. R. Singh, H. Larijani, and A. Philip, "Empirical performance analysis of WireGuard vs. OpenVPN in cloud and virtualised environments under simulated network conditions," Computers, vol. 14, no. 8, Art. no. 326, 2025, doi: 10.3390/computers14080326.
[12] P. Jaisudthi, P. T. Sridee, N. Phungkoed, K. Srisuk, and V. Phueaknumpol, "Comparative Study of Modern VPN Solutions: Impact of Cloudflare, ZeroTier, and WireGuard on Network and Server Performance," Engineering and Technology Horizons, vol. 42, no. 2, Art. no. 420203, Jun. 2025, doi: 10.55003/ETH.420203.
[13] I. Farooq, S. A. Ahmed, A. Ali, M. A. Warraich, M. Aqeel, and H. Khan, "Enhanced classification of networks encrypted traffic: A conceptual analysis of security assessments, implementation trends, and future directions," The Asian Bulletin of Big Data Management, vol. 4, no. 4, pp. 500–522, Dec. 2024.
[14] S. G. Hussein, "Development of an IPSec VPN on a proposed smart home system," AIP Conference Proceedings, vol. 3105, no. 1, Art. no. 080001, 2024, doi: 10.1063/5.0212178.
[15] H. Jumakhan and A. Mirzaeinia, "Wireguard: An Efficient Solution for Securing IoT Device Connectivity," arXiv:2402.02093, 2024, doi: 10.48550/arXiv.2402.02093.
[16] F. Streun, J. Wanner, and A. Perrig, "Evaluating Susceptibility of VPN Implementations to DoS Attacks Using Adversarial Testing," in Proc. NDSS Symposium, 2022.
[17] E. Dekker and P. Spaans, "Performance comparison of VPN implementations WireGuard, strongSwan, and OpenVPN in a 1 Gbit/s environment," Univ. of Amsterdam / OS3 project report, 2020.
[18] S. Mackey, I. Mihov, A. Nosenko, F. Vega, and Y. Cheng, "A Performance Comparison of WireGuard and OpenVPN," in Proc. ACM CODASPY, 2020, pp. 162–164, doi: 10.1145/3374664.3379532.
[19] A. Sivasangari, P. Ajitha, R. M. Gomathi, T. Anandhi, V. D. Vardhini, and P. A. Kumar, "Designing a secure and robust virtual private network (VPN) framework for enhanced network communication protection," AIP Conference Proceedings, vol. 3257, no. 1, Art. no. 020116, 2025, doi: 10.1063/5.0264873.
[20] H. Abbas et al., "Security Assessment and Evaluation of VPNs: A Comprehensive Survey," ACM Computing Surveys, vol. 55, no. 13s, pp. 1–47, 2023, doi: 10.1145/3579162.