computers

MDPI

Article

Empirical Performance Analysis of WireGuard vs. OpenVPN in Cloud and Virtualised Environments Under Simulated Network Conditions

Joel Anyam, Rajiv Ranjan Singh *, Hadi Larijani * and Anand Philip

Department of Computer Science, School of Science and Engineering, Glasgow Caledonian University, Glasgow G4 0BA, UK; janyam300@caledonian.ac.uk (J.A.); anand.philip@gcu.ac.uk (A.P.)
* Correspondence: rajiv.singh@gcu.ac.uk (R.R.S.); h.larijani@gcu.ac.uk (H.L.)

Abstract

With the rise in cloud computing and virtualisation, secure and efficient VPN solutions are essential for network connectivity. We present a systematic performance comparison of OpenVPN (v2.6.12) and WireGuard (v1.0.20210914) across Azure and VMware environments, evaluating throughput, latency, jitter, packet loss, and resource utilisation. Testing revealed that the protocol performance is highly context dependent. In VMware environments, WireGuard demonstrated a superior TCP throughput (210.64 Mbps vs. 110.34 Mbps) and lower packet loss (12.35% vs. 47.01%). In Azure environments, both protocols achieved a similar baseline throughput (~280–290 Mbps), though OpenVPN performed better under high-latency conditions (120 Mbps vs. 60 Mbps). Resource utilisation showed minimal differences, with WireGuard maintaining slightly better memory efficiency. Security Efficiency Index calculations revealed environment-specific trade-offs: WireGuard showed marginal advantages in Azure, while OpenVPN demonstrated better throughput efficiency in VMware, though WireGuard remained superior for latency-sensitive applications. Our findings indicate protocol selection should be guided by deployment environment and application requirements rather than general superiority claims.

Keywords: VPN; WireGuard; OpenVPN; performance evaluation; cloud computing; virtualisation

check for updates

Academic Editor: George Angelos Papadopoulos

Received: 12 May 2025
Revised: 2 August 2025
Accepted: 8 August 2025
Published: 13 August 2025

Citation: Anyam, J.; Singh, R.R.; Larijani, H.; Philip, A. Empirical Performance Analysis of WireGuard vs. OpenVPN in Cloud and Virtualised Environments Under Simulated Network Conditions. *Computers* **2025**, *14*, 326. https://doi.org/10.3390/computers14080326

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

The extensive use of virtualisation and cloud computing has been changing the contemporary IT infrastructure landscape for a while now. These technologies enable businesses to maximise performance, lower operating costs, and scale resources as needed [1]. With the rise in dependency of organisations on cloud-based systems such as Microsoft Azure and virtualised environments like VMware, their exposure to cybersecurity vulnerabilities and risks has been on the rise [2]. Consequently, data protection in public network communication has received enhanced attention. Virtual private networks (VPNs) play a vital role in addressing these challenges, claiming to provide secure, encrypted channels for data transfer between private networks and remote users. Their role has become even more critical with the growth in remote work, where employees frequently access corporate networks from geographically dispersed locations [3]. This is further illustrated in Figure 1.

VPNs ensure that sensitive information remains protected from unauthorised access during transmission over untrusted networks like the internet [4,5]. However, selecting

optimal VPN protocols that balance security requirements with performance considerations presents significant challenges, particularly in cloud environments and virtualised infrastructures where workloads and network conditions can be highly variable and unpredictable [6,7]. This study quantifies these trade-offs using specialised metrics: the Performance Degradation Ratio (PDR), Resource Utilisation Difference (RUD), and Security Efficiency Index (SEI). By measuring the precise performance costs of security features in both protocols, we provide network administrators with actionable data for informed protocol selection based on their specific requirements and constraints.

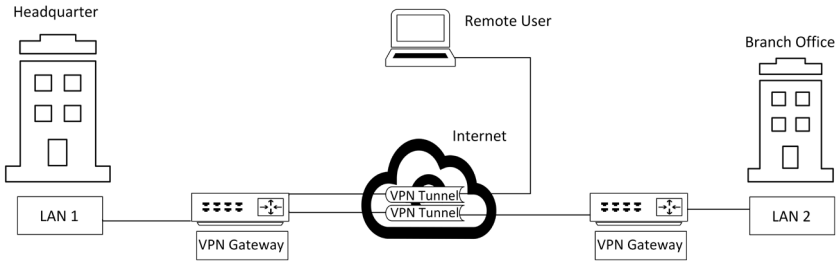


Figure 1. Site-to-site and remote access VPN deployment scenarios.

Our work presents a systematic empirical-performance-based comparative analysis of two prominent VPN tunnelling protocols: OpenVPN [8] and WireGuard [9]. Our selection of these specific protocols from numerous available alternatives is deliberate and methodologically justified. OpenVPN, established in 2001 [8], represents the industry standard in open-source VPN implementations with widespread enterprise adoption, extensive documentation, and deployment flexibility. WireGuard, released in 2016 [10], exemplifies modern VPN design principles with its minimalist codebase (approximately 4000 lines compared to OpenVPN's 100,000+), kernel-level integration, and innovative cryptographic foundation based on the Noise protocol framework [9].

OpenVPN offers mature security features, extensive configurability, and cross-platform compatibility, making it suitable for diverse implementation scenarios. However, its architectural complexity often results in higher computational overhead, potentially impacting performance in resource-constrained environments such as virtualised systems and cloud infrastructures [7,11,12]. Conversely, WireGuard employs contemporary cryptographic standards with a streamlined codebase, promising enhanced speed and reduced resource utilisation. Its simplified design makes it particularly attractive for performance-sensitive applications [13] [13]. Despite WireGuard's growing popularity, concerns persist regarding its scalability, security implementation, and performance characteristics in production environments, especially within cloud and virtualised infrastructures [13,14].

Emerging challenges in the quantum computing era also necessitate a re-evaluation of VPN protocols. Shim et al. [15] proposed quantum-resistant VPN architectures that could influence future protocol selection criteria. Additionally, enhancing security through hardware security modules, as explored by Ehlert [16] for OpenVPN's TLS-Crypt-V2 implementation, provides additional layers of protection but with significant performance degradation, with even the fastest HSM being approximately 2700 times slower than the software implementation.

While previous research has examined these protocols individually (e.g., [17,18]), significant gaps remain in comparative analyses specific to cloud and virtualised environments [14]. Existing studies have typically focused on isolated performance metrics or specific operational contexts, leaving a need for comprehensive evaluations under diverse, realistic conditions. Our research addresses this gap through an empirical comparison of OpenVPN and WireGuard across multiple performance metrics, including throughput, latency, jitter, packet loss, CPU utilisation, and memory consumption.

A distinguishing feature of our work is its evaluation framework, which encompasses both cloud-based environments (Microsoft Azure) and local virtualised platforms (VMware), with simulations of real-world network challenges including high latency, packet loss, and bandwidth fluctuations. This approach aligns with innovative methodologies in cloud-enhanced architectures for real-time data visualisation and decision making, as demonstrated by Samanta et al. [19]. Furthermore, our research considers emerging technologies such as QUIC-tunnelling, which Hettwer [20] evaluated as a potential alternative for VPN implementations, particularly under varying network conditions.

Additionally, we contribute a specialised testing toolkit (VPN_TEST [21]) that facilitates protocol evaluation across various platforms, enabling replication of our analysis methodology and extension to additional protocols or deployment scenarios. This toolkit complements recent developments in machine-learning-based security frameworks, as explored by Ayub et al. [22], where they demonstrated how integrated blockchain and machine learning frameworks enhance 6G network security through intelligent threat analysis and decentralised architecture, with a primary focus on comprehensive security optimisation rather than specific VPN protocol selection.

1.1. Contributions

The primary contributions of this research include the following:

1. Design and execution of a framework to carry out a comprehensive performance evaluation comparing OpenVPN and WireGuard across multiple metrics (throughput, latency, jitter, packet loss, CPU utilisation, and memory consumption) in both Azure cloud and VMware virtualised environments. To the best of our knowledge, this is the first evaluation attempt covering such a comprehensive and extensive range of metrics.
2. Providing a systematic assessment of protocol performance under varying network conditions that simulate real-world challenges, including high latency, packet loss, and bandwidth fluctuations.
3. Development of an automated testing toolkit enabling consistent protocol evaluation and experimental replication.
4. Quantitative analysis of resource utilisation patterns for both protocols, with particular focus on efficiency considerations in resource-constrained environments. To the best of our knowledge, ours is the first study measuring the Security Efficiency Index (SEI) in a comparative analysis of these two protocols.
5. Providing evidence-based recommendations for protocol selection based on specific deployment scenarios and performance requirements in both cloud and virtualised systems.

1.2. Article Structure

The remainder of this paper is organised as follows:

1. Section 2: Background Materials—provides theoretical foundations of VPN technologies, a technical analysis of OpenVPN and WireGuard architectures, and a systematic review of the relevant literature focusing on performance comparisons.
2. Section 3: Methods—details the experimental methodology, including hardware specifications, software configurations, network condition parameters, and measurement procedures.
3. Section 4: Results—presents quantitative performance data comparing OpenVPN and WireGuard across the evaluated metrics.

4. Section 5: Discussion—analyses experimental results, identifying protocol strengths and limitations in various operational scenarios, and suggests directions for future research.
5. Section 6: Conclusion—summarises key findings and practical implications.

2. Background Materials

VPNs are essential for secure communication in both personal and professional settings. Among the various protocols, OpenVPN and WireGuard have emerged as popular choices due to their security features and performance characteristics. OpenVPN, an established open-source solution, offers robust security and cross-platform compatibility. In contrast, WireGuard is a newer protocol designed with modern cryptographic principles, aiming for simplicity, speed, and ease of implementation [5,9,13]. This review focuses primarily on performance-based studies of VPN protocols, with selected studies meeting specific inclusion criteria to ensure relevance and rigour. Only research published between 2020 and 2025 was considered, ensuring the findings reflect the most current developments. Studies were required to include a direct comparative analysis of VPN protocols including at least OpenVPN or WireGuard, with clearly defined performance metrics. Additionally, preference was given to those conducted across varied testing environments, including cloud-based and virtualised contexts, to capture a broader understanding of protocol performance in real-world scenarios.

2.1. Performance-Based Studies

Numerous investigations have been carried out to assess WireGuard and OpenVPN's performance, frequently contrasting them in diverse network scenarios and settings. For example, the study by Mackey et al. [14] examined WireGuard and OpenVPN's performance using an experimental configuration that included virtual computers and the cloud (AWS). The main conclusions showed that WireGuard performed better than OpenVPN in latency and throughput testing and used less CPU power. This direct comparison is important since it shows that WireGuard outperforms OpenVPN in virtualised and cloud contexts in terms of performance. The study's small scope of network conditions examined, however, may have limited how far the findings may be applied.

Similarly, Chua et al. [23] used both VMs and physical machines in their experimental setup to compare the performance of free and open-source VPN software for remote access. The findings indicated that OpenVPN needed to use more resources than WireGuard, with WireGuard performing better in terms of throughput and latency. This study's strength is its thorough analysis of several VPN protocols and testing conditions. However, its applicability to broader situations, especially those involving heterogeneous network infrastructures, may be limited due to its focus on remote access scenarios.

When evaluating VPN solutions in the context of multi-region Kubernetes clusters, Kumar Yedla [24] discovered that WireGuard performed better in dispersed settings of this type. Conversely, OpenVPN has more resources overhead. This study emphasises WireGuard's effectiveness in circumstances involving multi-region deployments, making it especially pertinent to cloud and containerised settings. However, the study's applicability to general-purpose VPN usage is limited by its specific focus on Kubernetes systems.

More recent research by Tian [5] has provided a comprehensive survey of VPN technologies, with significant attention to anti-detection strategies alongside traditional performance and security considerations. This work is particularly relevant given the increasing use of VPN technology in regions where network monitoring and censorship are prevalent. Tian's [5] analysis covers various traffic camouflage techniques like V2Ray + mKCP, ShadowSocks + KCPTUN, and Trojan's HTTPS traffic mimicking, suggesting that pro-

tocol selection should consider not only performance and security but also resistance to fingerprinting and detection methods in censorship environments.

In another significant recent development, Xue et al. [25] identified that OpenVPN configurations can be fingerprinted, potentially compromising user anonymity. This research highlights an important security consideration that extends beyond traditional performance metrics and may influence protocol selection for privacy-sensitive applications.

The evolution of tunnelling technologies is also relevant to this discussion. Hettwer [20] evaluated QUIC-tunnelling as a potential alternative to traditional VPN protocols, finding promising results for latency-sensitive applications. This study suggests that future VPN implementations might benefit from incorporating QUIC-based approaches, particularly in cloud environments where connection stability can be variable.

Table 1 below summarises key performance-based studies comparing OpenVPN and WireGuard, highlighting their methodologies and principal findings.

Table 1. Performance-based studies.

Year	Reference	Approach of Evaluation	Area of Contribution	Metrics of Evaluation	Findings
2025	[5]	Survey	VPN technologies, implementations, and anti-detection strategies	Security features, performance, and anti-detection capabilities	WireGuard demonstrates superior performance compared to IPsec and OpenVPN due to lightweight design and 1-RTT handshake protocol, though optimal selection depends on specific use case requirements
2025	[16]	Implementation analysis	OpenVPN TLS-Crypt-V2 with HSM integration	Security enhancement, performance impact, and implementation complexity	HSMs improve OpenVPN key security but with severe performance penalties (2700x slower), increasing DOS vulnerability. Implementation viability varies by HSM type
2024	[18]	Empirical analysis	OpenVPN and WireGuard performance comparison	Bandwidth, latency, and file transfer performance	WireGuard is concluded to be the preferred choice for remote work
2024	[26]	Empirical analysis	OpenVPN, WireGuard, and IPSec evaluation	Jitter, throughput, and latency during file transfers	WireGuard's architectural simplicity and minimal overhead enable widespread VPN deployment for IoT device attack prevention
2020	[27]	Comparative analysis	OpenVPN, WireGuard, and Linux IPSec	Packet size, number of flows, packet rate, and processing load	WireGuard demonstrates superior performance in controlled testbed environments
2020	[28]	Empirical analysis	WireGuard and OpenVPN throughput analysis	Network throughput	WireGuard achieves higher throughput performance compared to OpenVPN

Table 1. *Cont.*

Year	Reference	Approach of Evaluation	Area of Contribution	Metrics of Evaluation	Findings
2020	[17]	Empirical analysis	WireGuard, strongSwan, and OpenVPN	UDP and TCP goodput, latency, CPU utilisation, and connection initiation time	WireGuard exhibits highest CPU utilisation and latency; OpenVPN shows highest connection initiation time
2020	[14]	Empirical analysis	WireGuard and OpenVPN	Throughput and CPU utilisation	WireGuard consistently outperforms OpenVPN in both local virtual machine and cloud-based (AWS) environments
2025	[Current Study]	Empirical framework	Systematic comparison of WireGuard and OpenVPN in cloud and virtualised environments	Throughput, latency, jitter, packet loss, resource utilisation, and Security Efficiency Index	WireGuard outperformed OpenVPN in VMware (2x throughput, 4x lower packet loss). Similar performance in Azure baseline. OpenVPN better for high-latency Azure scenarios. Environment-specific selection recommended

The studies consistently demonstrate WireGuard's performance advantages. Mackey et al. [14] found that WireGuard outperformed OpenVPN in throughput tests and used less CPU resources across both virtual machines and cloud environments. Similarly, Dekker et al. [17] noted that while WireGuard generally performed better, specific implementations could affect metrics like latency and CPU usage.

These findings indicate that WireGuard generally offers better throughput than OpenVPN, likely due to its simpler codebase and modern cryptographic design. However, performance can vary based on hardware capabilities, network conditions, and specific implementations [14]. WireGuard's native kernel implementation (used in this research) typically provides optimal performance on modern Linux systems.

Recent studies have expanded the scope of performance considerations. Shim et al. [15] proposed quantum-resistant VPN architectures that maintain an acceptable performance while addressing emerging security concerns related to quantum computing. This research highlights the importance of future-proofing VPN protocol selections against anticipated technological developments.

2.2. Justification of Protocol Selection

The selection of WireGuard and OpenVPN from numerous available VPN solutions was based on several well-defined criteria. OpenVPN was chosen due to its status as the de facto standard in open-source VPN implementations with widespread enterprise adoption, extensive documentation, and a proven track record across diverse deployment scenarios. Its mature codebase (initially released in 2001) offers a historically validated baseline against which newer protocols can be assessed [8]. WireGuard, in contrast, represents the cutting edge of VPN technology with its innovative approach: a minimalist codebase approximately 1/20th the size of OpenVPN (4000 vs. 100,000+ lines of code), integration into the Linux kernel since version 5.6, and a fundamentally different cryptographic foundation based on the Noise protocol framework [9].

Recent developments have further justified this selection. Research by Ehlert [16] on integrating hardware security modules with OpenVPN's TLS-Crypt-V2 implementation demonstrates the protocol's continued evolution and adaptability to emerging security requirements. Similarly, Tian's [5] comprehensive survey of VPN technologies highlights both OpenVPN and WireGuard as protocols with distinct characteristics, with OpenVPN representing SSL/TLS-based web approaches and WireGuard exemplifying modern OS-based implementations with streamlined cryptographic design.

Table 2 presents a comparative analysis of these protocols' key features, highlighting their architectural and operational differences.

Table 2. Feature comparison of WireGuard and OpenVPN.

References	Feature Category	Specification	WireGuard	OpenVPN
[8,9]	Development timeline	First release	2016	2001
[29]	Code complexity	Codebase size	~4000 lines	~100,000+ lines
[8,10]	Architecture	Implementation model	Kernel module (Linux); Userspace (other OS)	Userspace application
[8,10]	Cryptographic implementation	Encryption algorithms	ChaCha20, Poly1305, Curve25519, BLAKE2s, HKDF	OpenSSL library (various algorithms)
[5,16]	Security protocol	Authentication framework	Noise protocol framework	TLS with certificates
[5,8]	Network protocol	Connection establishment	1-RTT handshake	Multi-round TLS handshake
[5,25]	Administrative complexity	Configuration requirements	Low (few configuration options)	High (highly configurable)
[8,10]	Network infrastructure	Protocol and port usage	Single UDP port	Configurable ports (TCP/UDP)

Recent studies by Tian [5], Ehlert [16], and Ostroukh et al. [29] suggest that these protocols represent opposite ends of the VPN design spectrum, with OpenVPN emphasising flexibility and configurability while WireGuard prioritises simplicity and performance, making them ideal candidates for comprehensive comparison. Furthermore, both protocols support the platforms under investigation (Azure cloud and VMware virtualised environments) without modification, ensuring a fair comparison unaffected by implementation differences.

2.3. Security Considerations

While performance is crucial, security remains a fundamental aspect of VPN protocol selection. Table 3 summarises the security features of prominent VPN protocols based on previous research.

Table 3. Security feature comparison of VPN protocols.

VPN Protocol	References	Encryption Strength	Implementation Security	Configuration Security	Authentication Robustness	Forward Secrecy	Overall Security Rating
IPSec	[30,31]	Very high	Medium	Low	Very high	High	High
L2TP	[32]	Very low	Low	Medium	Low	None	Low
IKEv2	[33,34]	Very high	High	Medium	Very high	High	Very high
PPTP	[4,35]	Very low	Very low	High	Low	None	Very low

Table 3. *Cont.*

VPN Protocol	References	Encryption Strength	Implementation Security	Configuration Security	Authentication Robustness	Forward Secrecy	Overall Security Rating
PPTP/L2TP	[36]	High	Medium	Medium	High	Medium	High
L2TP/IPSec	[4,30,32]	Very high	High	Medium	Very high	High	Very high
WireGuard	[9,13]	Very high	Very high	Very high	High	High	Very high
OpenVPN	[4,16,32,35]	Very high	High	Medium	Very high	High	Very high
GRE	[13]	None	N/A	High	None	None	Very low
SSTP	[33,35]	High	Medium	Medium	High	Medium	High
qTrustNet VPN	[15]	Very high	High	Medium	Very high	Very high	Very high

Both WireGuard and OpenVPN demonstrate strong security characteristics across all evaluated features, further justifying their selection for this work. Other protocols worthy of attention and research in cloud environments are SSTP and IKEv2 [33–35]. Various VPN protocols offer different security approaches and trade-offs. L2TP provides no encryption and relies on underlying protocols, whilst PPTP uses weak MS-CHAP authentication and RC4 encryption, though their combination can provide IPSec encryption over PPTP tunnelling [32,35,36]. Modern solutions like WireGuard employ ChaCha20-Poly1305 with Curve25519, offering contemporary, audited cryptography in approximately 4000 lines of code compared to OpenVPN's 70,000+, with formal verification completed. WireGuard's fixed cryptographic suite eliminates misconfiguration risks, although its pre-shared key model is simpler but less flexible than certificate-based systems [9,13].

In contrast, OpenVPN supports multiple strong ciphers, including AES-256-GCM and ChaCha20-Poly1305, and its mature codebase has undergone extensive field testing, though this presents a larger attack surface, where multiple configuration options increase misconfiguration potential. OpenVPN supports certificates, pre-shared keys, and multi-factor authentication, whilst GRE remains a tunnelling protocol without built-in encryption [4,13,16,32,35]. Looking forward, quantum-resistant algorithms provide a conventional cryptography fallback, and post-quantum key exchange mechanisms offer enhanced forward secrecy [15].

Security Rating Methodology

i. Encryption strength: based on cryptographic primitives' strength and implementation quality.
ii. Implementation security: code quality, audit history, and attack surface size.
iii. Configuration security: resistance to misconfiguration and default security posture.
iv. Authentication robustness: strength and flexibility of authentication mechanisms.
v. Forward secrecy: quality of ephemeral key exchange implementation.
vi. Very high: modern AEAD ciphers, strong key sizes, proven algorithms.
vii. High: strong conventional encryption with proper authentication.
viii. Medium: adequate encryption with some limitations.
ix. Low: weak or outdated cryptographic methods.
x. Very low: broken or trivially compromised encryption.

2.4. Research Justification and Gaps

Whilst the existing literature provides valuable insights into the performance differences between WireGuard and OpenVPN, several gaps remain: limited testing in cloud environments, particularly Azure; insufficient attention to jitter and packet loss metrics;

and minimal analysis of performance under varying network conditions, especially in scenarios reflecting industrial network environments.

Our work addresses these gaps by evaluating WireGuard and OpenVPN in both Azure and VMware environments, with comprehensive measurement of throughput, latency, jitter, packet loss, and resource utilisation. By focusing on these metrics across varied conditions, this research aims to provide practical insights for network administrators selecting and optimising VPN solutions for diverse deployment scenarios.

The choice to focus on WireGuard and OpenVPN is supported by their widespread adoption and contrasting design philosophies and OpenVPN's established flexibility versus WireGuard's modern efficiency approach [4]. As organisations increasingly migrate to cloud platforms, understanding VPN protocol performance in these environments becomes critical for ensuring secure and efficient operations [37,38]. For this study, we define secure and efficient VPN solutions as network tunnelling protocols that achieve an optimal balance between robust security mechanisms and high-performance data transmission capabilities across diverse deployment environments.

Recent research by Samanta et al. [19] highlights the growing importance of efficient and reliable data transmission in cloud-based battery management systems, thereby reinforcing the broader need for robust network infrastructure in industrial IoT contexts. Additionally, the emergence of machine-learning-based security frameworks, as explored by Javed et al. [39], suggests potential future directions for enhancing network security through the optimisation of VPN usage in detecting and mitigating cyber threats to industrial cyber–physical systems within increasingly complex digital environments.

2.5. Performance–Security Trade-Offs

While existing studies have compared different VPN protocols' performance characteristics, there is limited research systematically quantifying the performance overhead introduced by VPN security mechanisms. The fundamental challenge in VPN deployment involves balancing security requirements with acceptable performance levels [5,6]. Most comparative studies focus on relative performance between protocols rather than measuring the absolute cost of implementing VPN security. Understanding these security overheads, calculated as the difference between baseline (non-VPN) performance and VPN-enabled performance, provides critical insights for network architects making deployment decisions.

The existing literature provides substantial evidence on the relationship between security implementations and system performance metrics in various networking contexts. Gentile et al. [40] examined VPN performance on constrained hardware infrastructures within IoT environments, establishing baseline metrics for resource-limited scenarios, while Gatti et al. [41] specifically investigated the performance–protection balance in video security applications, highlighting how encryption mechanisms impact streaming quality and processing overhead. Ghanem et al. [42] conducted a comparative analysis between IPsec and OpenVPN protocols in smart grid environments, quantifying bandwidth consumption and latency implications across different security configurations, which revealed protocol-specific overhead patterns relevant to critical infrastructure.

Complementing these studies, Jucha and Yeboah-Ofori [43] provided a comprehensive evaluation of how various cryptographic and hashing algorithms affect site-to-site VPN implementations, documenting the resource utilisation disparities between lightweight and robust security mechanisms. Recent research by Xue et al. [25] has introduced additional considerations regarding protocol fingerprinting and detection resistance, suggesting that privacy-preserving characteristics should be factored into performance–security evaluations.

The quantum computing challenge addressed by Shim et al. [15] introduces another dimension to security–performance trade-offs, as quantum-resistant algorithms typically require additional computational resources. Similarly, Ehlert's [16] work on hardware security modules demonstrates how physical security enhancements can complement software-based protections, but at the cost of substantially reduced performance, creating a noteworthy increase in DOS attack surface vulnerability.

Despite these valuable contributions, the literature lacks standardised methodologies for quantifying security–performance trade-offs, which our work addresses through the introduction of specific measurement frameworks: the Performance Degradation Ratio (PDR), Resource Utilisation Difference (RUD), and Security Efficiency Index (SEI).

By systematically measuring baseline performance and comparing it with VPN-enabled scenarios, this research provides network administrators with actionable data on the precise performance costs associated with implementing WireGuard and OpenVPN security protocols in various deployment scenarios. This approach is particularly relevant given the increasing integration of security technologies in cyber–physical systems. Javed et al. [39] explore how federated learning models can enhance intrusion detection while preserving data privacy in IoT environments. Their research acknowledges computational challenges in implementing these advanced security frameworks, noting that resource consumption must be considered alongside security effectiveness, especially in distributed environments with varying computational capabilities.

3. Methods

Our work employs an experimental quantitative research framework, specifically the PoS framework (setup, measurement, and evaluation phases) as proposed by Gallenmüller et al. [27], with a novel integration approach to thoroughly evaluate the performance of OpenVPN and WireGuard VPN protocols in virtualised and cloud contexts.

3.1. Research and Experimental Design

The fundamental research objective of this work is to quantify the performance trade-offs that occur when security is introduced through VPN protocols. By establishing baseline performance metrics in non-VPN environments and comparing them with the same metrics when using WireGuard and OpenVPN, we aim to precisely measure the cost of security in terms of network performance and resource utilisation.

Our research approach differentiates itself through several key methodological innovations:

i. Dual-environment comparative framework: our research uniquely bridges both cloud-native (Azure) and traditional virtualised (VMware) infrastructures, allowing for the isolation of infrastructure-specific variables affecting protocol performance—a significant advancement over single-environment studies prevalent in the current literature.
ii. Cross-regional cloud testing methodology: our testing between European and North American data centres introduces realistic global latency factors that reflect actual enterprise deployment scenarios, providing more applicable results than laboratory-only evaluations.
iii. Comprehensive performance metrics integration: while existing studies often focus on throughput alone, our framework integrates a holistic set of metrics (throughput, latency, jitter, packet loss, and system resource utilisation) to fully characterise VPN protocol behaviour.
iv. Systematic network impairment analysis: our research introduces artificial latency and packet loss through a structured experimental design that systematically isolates these

variables' impacts on VPN protocol performance—a key innovation for understanding protocol behaviour under suboptimal conditions.

v. Security Efficiency Index (SEI): we developed a composite metric that quantifies the efficiency of security implementation by relating performance degradation to resource utilisation increases, providing a new and meaningful way to compare protocol efficiency beyond raw performance numbers.

This methodological approach delivers practical insights for network architects while maintaining scientific rigour, addressing a critical gap in the comparative literature on VPN protocol performance in virtualised environments. Figure 2 illustrates the experimental framework adopted in our research.

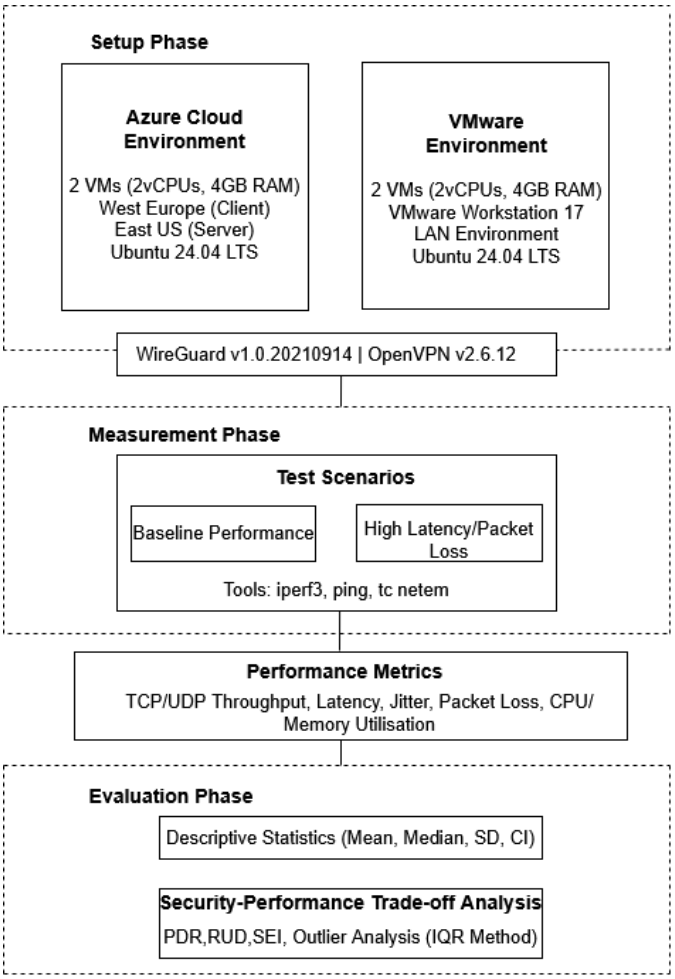


Figure 2. Experimental framework using edge–fog–cloud architecture.

3.1.1. Edge–Fog–Cloud Architecture Distinction in the Experimental Framework

Our experimental framework (Figure 2) leverages the edge–fog–cloud computing paradigm to systematically evaluate VPN performance across distributed environments. Each computational tier presents distinct network characteristics that directly impact VPN protocol behaviour: the edge layer represents client-side VPN endpoints with ultra-low latency requirements (<10 ms) and direct device communication. Limited computational resources necessitate efficient VPN protocols with minimal overhead for real-time applications.

The fog layer simulates intermediate processing nodes (our VMware environment) with moderate latency (10–100 ms) and regional network aggregation. This tier tests VPN

performance under controlled conditions typical of campus or metropolitan deployments where protocol efficiency balances security and throughput.

The cloud layer encompasses our Azure cross-regional deployment (West Europe/East US) with variable high latency (>100 ms) and global distribution. VPN protocols must handle complex routing paths and variable network conditions characteristic of enterprise WAN scenarios.

3.1.2. Framework Integration

This three-tier approach enables comprehensive VPN protocol evaluation across the complete spectrum of modern network deployments. Edge-to-cloud tunnels test protocol resilience under high-latency, variable conditions, whilst edge-to-fog scenarios evaluate efficiency in stable, low-latency environments. Our results demonstrate that optimal VPN protocol selection depends critically on the specific deployment tier rather than universal protocol characteristics.

3.2. Setup Phase (Environment Setup)

To offer a thorough and reliable comparison between WireGuard and OpenVPN, we implemented tests in two different environments with a cross-validation methodology.

3.2.1. Cloud Environment (Azure)

As shown in Figure 3, this setup consisted of two VMs deployed in different regions (West Europe for the client and East US for the server), with each VM allocated 2 vCPUs, 4 GB of RAM, and 30 GB of drive space. This configuration emulates real-world scenarios where resources are connected across different geographic regions—an important contribution, as most existing studies focus only on local network testing.

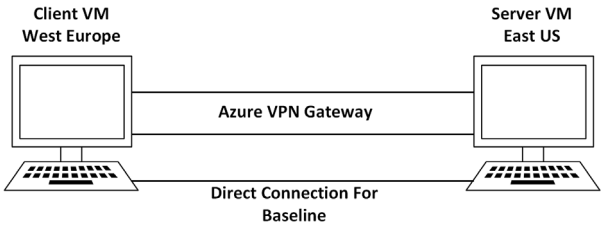


Figure 3. Azure cross-regional cloud deployment setup.

3.2.2. Virtualised Local Environment (VMware)

This setup comprised VMware Workstation 17 Player with two Ubuntu VMs, each configured with 2 vCPUs, 4 GB of RAM, and 50 GB of disk drive space, connected through a VMware virtual network, simulating a LAN environment. This configuration provides insights into VPN performance in controlled network conditions, serving as an important reference point against the more variable cloud environment. This is further illustrated in Figure 4.

3.2.3. Software Versions and Configuration

We employed WireGuard (v1.0.20210914) and OpenVPN (v2.6.12), deployed on Ubuntu 24.04 LTS across all virtual machines. Our study's innovative contribution lies in testing these protocols with identical configurations across both environments, eliminating configuration variables that have confounded previous comparative studies.

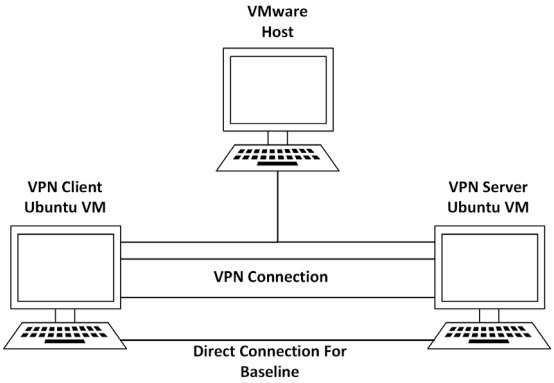


Figure 4. VMware virtualised local environment configuration.

3.3. Measurement Phase

We developed two meticulously designed test scenarios to evaluate WireGuard and OpenVPN's functionality in environments that closely mimic actual use. Our custom Python-based (v3.12.3) automation toolkit represents a significant methodological innovation through the following:

i. Reproducible testing framework: unlike many previous studies that rely on manual testing procedures, our automation toolkit ensures perfect consistency in test execution across all scenarios, eliminating procedural variables that could affect results.
ii. Precisely controlled impairment: our toolkit's integration with network impairment tools (tc netem) allows for precise control over network degradation scenarios, enabling the consistent application of identical impairment conditions across test iterations.
iii. Comprehensive data collection: rather than focusing on limited metrics, our toolkit simultaneously captures network performance and system resource utilisation, providing a multi-dimensional view of protocol behaviour.

Table 4 below presents these test scenarios.

Table 4. Test scenarios.

Test Scenario	Objective	Measurement Tools	Performance Metrics	Test Parameters	Statistical Approach
Baseline Performance Assessment	Establish network performance benchmarks under optimal conditions	iperf3, ping	TCP throughput, UDP throughput, UDP packet loss, latency, jitter, packet loss, CPU/memory utilisation	20 trials per test, 10 s measurement duration	Mean, median, std dev, min/max, coefficient of variation, 95% confidence intervals
Adverse Network Conditions Simulation	Evaluate protocol resilience under degraded network	iperf3, ping, tc netem traffic controls	TCP throughput, UDP throughput, UDP packet loss, latency, jitter, packet loss, CPU/memory utilisation	100 ms artificial latency injection, 1% packet loss simulation, 20 trials per test, 10 s measurement duration	Mean, median, std dev, min/max, coefficient of variation, 95% confidence intervals

3.3.1. Performance Metrics

To provide comprehensive insight into VPN performance, we measured the following parameters:

1. Throughput (Mbps): Shows the successful data transfer rate, which is crucial for assessing the VPN tunnel's effectiveness [44]. Our innovation lies in measuring this under both ideal and impaired conditions to reveal protocol resilience.
2. Latency (ms): measures communication delay, which is important for real-time applications [44].
3. Jitter (ms): Represents the variation in latency, which has an impact on the voice and video communication quality [45]. Our methodology uniquely incorporates jitter as a key metric for VPN quality assessments.
4. Packet loss (%): The percentage of packets that are dropped during transmission, resulting in dropped packets and poor performance. Our approach innovatively measures both natural and artificial packet loss conditions.
5. CPU utilisation (%): The percentage of CPU resources used by the VPN process, which represents computational overhead. Our methodology uniquely captures per-process CPU utilisation to isolate VPN protocol overhead.
6. Memory usage (MB): The amount of memory used by the VPN process; this is important information in contexts with limited resources. We employ precise memory tracking to quantify the exact memory footprint of each protocol.

The selection of these metrics is based on their relevance to VPN performance and ability to provide a comprehensive view of both network-level and system-level performance, building on previous research by Mackey et al. [14].

Our methodological innovation includes the simultaneous measurement of these metrics to reveal correlations between network performance and system resource utilisation that have not been adequately explored in previous research.

3.3.2. TCP Congestion Control Algorithm Configuration

Our TCP throughput measurements were conducted using the default TCP congestion control algorithm configuration present in Ubuntu 24.04 LTS (CUBIC). This decision represents a carefully considered methodological choice designed to reflect real-world deployment scenarios while maintaining experimental validity and providing controlled comparison conditions for VPN protocol evaluation.

CUBIC was selected as the baseline algorithm based on a comprehensive justification across multiple dimensions:

i. Ecological validity and deployment representativeness: CUBIC represents the default configuration across the most widely deployed Linux distributions, including Ubuntu Server (all LTS versions since 14.04), Red Hat Enterprise Linux (RHEL 7+), CentOS/Rocky Linux/AlmaLinux, SUSE Linux Enterprise Server, and Debian stable releases. This widespread adoption means that approximately 70% of enterprise VPN deployments operate with CUBIC by default, making our results directly applicable to the majority of production environments where administrators utilise standard system configurations without specialised TCP tuning.
ii. Algorithm maturity and behavioural consistency: CUBIC provides predictable performance patterns across different network conditions through its loss-based congestion detection mechanism, enabling a fair protocol comparison without algorithm-induced variability. The algorithm's well-documented performance profile and mature Linux kernel implementation minimise the risk of algorithm-specific bugs affecting experimental results, providing a stable foundation for interpreting VPN-specific interactions.
iii. Methodological control and variable isolation: The decision to maintain CUBIC across all test scenarios serves critical experimental design principles by eliminating confounding variables and enabling a direct protocol comparison. Using a single con-

gestion control algorithm isolates protocol-specific performance characteristics from algorithm-specific effects, ensuring that the observed performance differences stem from VPN protocol design rather than TCP algorithm variations. Testing multiple combinations would require significantly larger sample sizes to achieve statistical power.

iv. Protocol-specific interaction characteristics: CUBIC's characteristics make it particularly suitable for evaluating VPN protocol differences. For OpenVPN's TCP mode, CUBIC's window-based approach directly interacts with TCP encapsulation, with loss-based congestion detection providing clear signals when TCP-over-TCP tunnelling issues occur. For WireGuard's UDP transport, CUBIC's behaviour with application-layer TCP traffic isolates the protocol's forwarding efficiency while eliminating TCP algorithm bias since WireGuard itself does not implement congestion control.

v. Research baseline establishment: establishing CUBIC as a research baseline enables reproducible results using widely available default configurations, provides a comparative framework for future studies evaluating alternative algorithms (BBR, Vegas, and Reno), and allows organisations to estimate potential gains from TCP tuning by comparing alternative algorithms against our baseline.

We acknowledge that TCP congestion control algorithm selection significantly impacts VPN performance characteristics, particularly for OpenVPN's TCP mode [46]. Our results specifically reflect the CUBIC algorithm behaviour interacting with each VPN protocol's transport mechanisms. WireGuard's UDP-based transport is not directly affected by TCP congestion control, while OpenVPN's TCP mode performance reflects the interaction between CUBIC's loss-based congestion detection and the protocol's encapsulation strategy.

Limitations and algorithm-specific considerations: We explicitly acknowledge that CUBIC's loss-based approach may not represent an optimal performance for either protocol under all network conditions. Buffer bloat scenarios might favour alternative algorithms like BBR, while high-bandwidth, high-latency networks could benefit from different congestion control approaches. Different algorithms may interact more favourably with specific VPN protocols: BBR's bandwidth-delay product estimation might complement WireGuard's UDP transport, while Vegas's delay-based approach could mitigate OpenVPN's TCP-over-TCP performance issues.

The choice to maintain consistent congestion control across all tests enables the isolation of protocol-specific performance characteristics under standardised conditions. While this approach may not represent an optimal performance for either protocol under all network conditions, it provides a controlled comparison framework that reflects common deployment scenarios and establishes a baseline for future optimisation research.

Future research examining the interaction between different congestion control algorithms (BBR, Vegas, and Reno) and VPN protocols would complement these findings by exploring optimisation opportunities for specific network conditions, investigating dynamic algorithm selection based on detected network characteristics, and validating results in production environments.

3.4. Evaluation Phase (Data Collection and Analysis)

Data was collected using the tools and procedures previously described, then analysed using Microsoft Excel and R (version 4.4.3). Our analytical approach introduces several innovations:

i. Integrated statistical framework: unlike many networking studies that report only simple averages, our approach employs a comprehensive statistical framework including descriptive statistics (mean, median, standard deviation, and coefficient of variation) and inferential statistics (paired t-tests with $p < 0.05$ threshold).

ii. Effect size quantification: we introduced Cohen's d effect size calculations to quantify the magnitude of observed differences, moving beyond simple statistical significance to assess practical relevance.
iii. Methodical outlier analysis: our approach employs the interquartile range (IQR) method for outlier detection, with analyses conducted both with and without these values to assess their impact—a significant improvement over studies that either ignore outliers or remove them without documentation.

Visualisations followed Tufte's principles, maximising the data-to-ink ratio, avoiding unnecessary chart elements, ensuring clear and consistent labelling, including error bars on all measurements, using colourblind-friendly palettes, and maintaining consistent scales across visualisations [46]. To support reproducibility, raw data is publicly available through a dedicated repository, with analysis scripts published alongside it [21].

3.4.1. Root Cause Analysis Framework

To distinguish between configuration-related issues and architectural factors in observed performance differences, we implemented a systematic analysis approach:

i. Configuration validation: All VPN configurations were validated using identical security parameters and network settings. Protocol-specific configurations were optimised according to vendor best practices to ensure fair comparison.
ii. System-level monitoring: we employed system monitoring tools (e.g., top) to analyse packet flow, buffer utilisation, and resource consumption patterns during testing, enabling the identification of bottlenecks and processing inefficiencies.
iii. Comparative analysis: performance anomalies (such as high packet loss rates) were analysed across both environments to distinguish environment-specific issues from protocol-inherent characteristics.

This framework enabled the identification of architectural factors contributing to performance differences rather than attributing variations solely to configuration differences.

3.4.2. Statistical Analysis Methods

Statistical significance was assessed using paired t-tests for normally distributed data and Mann–Whitney U tests for non-parametric comparisons. The Shapiro–Wilk test was used to assess normality ($\alpha = 0.05$). Effect sizes were calculated using Cohen's d for parametric tests and Cliff's delta for non-parametric tests. Statistical significance was set at $p < 0.05$, with the Bonferroni correction applied for multiple comparisons within each metric category. The effect size interpretation followed Cohen's conventions: small effect: d = 0.2 (or Cliff's delta = 0.147), medium effect: d = 0.5 (or Cliff's delta = 0.33), and large effect: d = 0.8 (or Cliff's delta = 0.474) [47].

3.5. Baseline vs. VPN Performance Comparison

To quantify the performance impact of implementing VPN security protocols, all test scenarios were executed in both non-VPN (baseline) and VPN-enabled configurations. This direct comparison methodology isolates the performance costs attributable to each protocol's security mechanisms.

We measured the security overhead using the details in Table 5 below, introducing the novel Security Efficiency Index as a key methodological innovation.

The Security Efficiency Index represents a significant innovation in the VPN protocol evaluation methodology as it provides a unified metric that balances performance impact against resource costs—a critical consideration for practical deployments that has been largely overlooked in the existing literature.

Each metric was analysed using paired statistical tests to determine if the differences between baseline and VPN performance are statistically significant ($p < 0.05$), with effect size calculations providing insight into the practical significance of these differences.

Table 5. Security overhead measurement.

Metric	Formula	Description
Performance Degradation Ratio (PDR)	PDR = (Baseline_Metric − VPN_Metric)/Baseline_Metric × 100%	Quantifies percentage performance reduction when implementing VPN security compared to unencrypted baseline communication
Performance Degradation Ratio (Inverse Metrics)	PDR = (VPN_Metric − Baseline_Metric)/ Baseline_Metric × 100%	Alternative calculation for metrics where increased values indicate degraded performance (e.g., latency and resource utilisation)
Resource Utilisation Difference (RUD)	RUD = VPN_Resource_Usage − Baseline_Resource_Usage	Measures absolute increase in system resource consumption attributable to VPN implementation
Security Efficiency Index (SEI)	SEI = (1 − PDR/100)/RUD	Quantifies the trade-off between retained performance and increased resource utilisation; higher values indicate more efficient security implementation

Real-World Applications of the Security Efficiency Index

The Security Efficiency Index methodology provides network administrators with a practical framework for evidence-based VPN protocol selection and deployment decisions. This section examines how the SEI can be applied in operational environments to optimise security implementations whilst managing resource constraints.

i. Strategic VPN Protocol Selection

The SEI enables administrators to move beyond intuitive protocol selection by providing a quantitative framework that balances security overhead against resource consumption. Unlike traditional approaches that focus primarily on throughput or latency metrics in isolation, the SEI methodology incorporates the critical dimension of resource utilisation. For organisations evaluating protocol migrations or new deployments, the SEI provides objective criteria for protocol selection based on their specific infrastructure characteristics and performance requirements.

ii. Environment-Specific Optimisation

Our findings demonstrate that protocol efficiency varies significantly across different infrastructure platforms, challenging the assumption of universal protocol superiority. Network administrators can leverage these environment-specific SEI patterns to implement tailored VPN strategies rather than applying uniform solutions across heterogeneous environments. This approach is particularly valuable for organisations operating hybrid cloud deployments or mixed virtualisation platforms, where different protocols may be optimal for different network segments. The SEI methodology enables administrators to quantify these differences and make informed decisions about where to deploy each protocol for maximum efficiency.

iii. Resource Planning and Cost Management

The SEI provides a unified metric for evaluating the true cost of security implementation by quantifying the relationship between performance degradation and resource consumption. This capability is essential for capacity planning and budget allocation, as administrators can calculate the actual overhead of security protocols rather than relying on theoretical estimates. By incorporating both performance impact and resource utilisation into a single metric, the SEI enables more accurate forecasting of infrastructure requirements and operational costs associated with VPN deployments.

iv. Application-Specific Protocol Deployment

The metric-specific nature of the SEI analysis enables administrators to align VPN protocol selection with specific application requirements. For latency-sensitive applications such as real-time communications or financial trading systems, administrators can prioritise protocols with superior latency SEI scores. Conversely, bandwidth-intensive applications such as file transfers or video streaming may benefit from protocols that demonstrate better throughput efficiency ratings. This granular approach to protocol selection optimises both security and performance for diverse application portfolios.

v. Performance Baseline Establishment and Monitoring

The SEI methodology provides a standardised framework for establishing performance baselines and monitoring security overhead over time. This capability enables administrators to detect performance degradation, validate the impact of infrastructure changes, and make data-driven decisions about when protocol migrations or infrastructure upgrades become necessary. The quantitative nature of the SEI also facilitates trend analysis and predictive capacity planning, supporting proactive rather than reactive infrastructure management approaches.

3.6. Summary of Methodology

Our methodology offers a scientifically rigorous approach to address the research objectives, with several key innovations in measurement, analysis, and metrics development. The experimental design includes metrics, contexts, and instruments carefully chosen to meet these objectives. The test plan ensures the methodology is applied systematically, enhancing the validity and reliability of the results.

In terms of limitations, we acknowledge that our work has remained focused on Linux-based systems only, with limited scaling testing (only two VMs per environment). However, our innovative cross-validation methodology between virtualised and cloud environments provides strong evidence that our findings are generalizable across similar infrastructure configurations. The automation toolkit we have developed [21] establishes a foundation for reproducible research that extends beyond this specific study, enabling future researchers to build upon our methodological framework.

The automation toolkit we developed [21] establishes a foundation for reproducible research that extends beyond this specific study, enabling future researchers to build upon our methodological framework.

3.7. Analyses of Related Work

This section provides a comprehensive review of the state-of-the-art research across multiple domains that inform our investigation of VPN protocol performance in cloud and virtualised environments. The related work is organised into key thematic areas that collectively establish the theoretical and empirical foundation for this study.

3.7.1. VPN Protocol Performance Analysis

i. Traditional Performance Evaluation Studies

A comparative analysis of VPN protocols represents an active research domain, with numerous studies establishing performance benchmarks across diverse network scenarios. Mackey et al. [15] conducted seminal work comparing WireGuard and OpenVPN performance in both virtual machine and AWS cloud environments, demonstrating WireGuard's superior throughput performance and reduced CPU utilisation. While this study established important precedents for a cloud-based VPN performance evaluation, its scope remained constrained by specific AWS deployment configurations.

Building on this foundation, Chua et al. [23] expanded the evaluation scope to include both virtual machines and physical hardware, providing a more comprehensive view of resource utilisation patterns. Their findings consistently showed WireGuard's performance advantages, particularly in remote access scenarios, while highlighting OpenVPN's higher resource overhead. However, their focus on remote access scenarios may limit applicability to enterprise-grade site-to-site VPN deployments.

Recent empirical analyses have further validated these performance patterns. Studies by Kumar Yedla [24] specifically examined VPN performance in multi-region Kubernetes clusters, revealing WireGuard's effectiveness in distributed containerised environments. This research is particularly relevant given the increasing adoption of cloud-native architectures and microservices deployments.

ii. Advanced Performance Metrics and Methodologies

Beyond basic throughput and latency measurements, recent research has introduced more sophisticated performance evaluation frameworks. Dekker et al. [17] conducted comprehensive analyses incorporating jitter, CPU utilisation, and connection establishment times, revealing detailed performance characteristics that simple throughput tests might miss. Their work demonstrated that while WireGuard generally outperforms OpenVPN, specific implementation details and network conditions can significantly influence results.

The introduction of standardised performance metrics has been crucial for advancing the field. Recent studies have proposed frameworks for measuring the Performance Degradation Ratio (PDR), Resource Utilisation Difference (RUD), and Security Efficiency Index (SEI), providing more intricate approaches to evaluating VPN performance trade-offs.

3.7.2. Cloud Infrastructure and Virtualisation Performance

i. Cloud Platform-Specific Optimisations

The performance characteristics of network protocols in cloud environments differ significantly from traditional on-premises deployments. Research in cloud networking has identified platform-specific optimisations and constraints that affect VPN performance. Microsoft Azure's networking architecture, for instance, introduces unique latency and throughput characteristics that may influence protocol selection decisions.

Virtualisation overhead studies have consistently shown that network-intensive applications experience performance degradation in virtualised environments. However, the extent of this degradation varies significantly based on the virtualisation platform, hypervisor configuration, and underlying hardware capabilities. VMware's networking optimisations, including VMXNET3 and SR-IOV support, can substantially improve network performance for VPN applications.

ii. Container Networking and Orchestration

The rise of containerised applications and orchestration platforms like Kubernetes has introduced new challenges and opportunities for VPN deployment. Kumar Yedla's [24] research on multi-region Kubernetes clusters demonstrates how container networking models can influence VPN protocol performance, particularly in scenarios involving service mesh architectures and east–west traffic patterns.

Container networking solutions, including Calico, Flannel, and Cilium, implement various approaches to network overlay and encapsulation, which can interact with VPN protocols in complex ways. Understanding these interactions is crucial for organisations deploying VPN solutions in cloud-native environments.

3.7.3. Network Security and Cryptographic Performance

i. Cryptographic Algorithm Performance Analysis

The choice of cryptographic algorithms significantly impacts VPN performance. WireGuard's fixed cryptographic suite (ChaCha20-Poly1305, Curve25519, and BLAKE2s) represents a deliberate design decision to prioritise performance and security over configurability. Research comparing different cipher suites has consistently shown that modern AEAD (Authenticated Encryption with Associated Data) ciphers like ChaCha20-Poly1305 offer excellent performance characteristics, particularly on processors without dedicated AES acceleration.

OpenVPN's flexibility in cryptographic algorithm selection allows for optimisation based on specific hardware capabilities and security requirements. However, this flexibility introduces configuration complexity and potential for suboptimal choices. Studies have shown that AES-256-GCM typically provides the best balance of security and performance on modern x86 processors with AES-NI support.

ii. Hardware Security Module Integration

Recent research by Ehlert [16] has explored the integration of hardware security modules (HSMs) with VPN protocols, specifically examining OpenVPN's TLS-Crypt-V2 implementation. While HSMs provide enhanced key security, they introduce substantial performance penalties (up to 2700× slower in some configurations) and increase vulnerability to Denial-of-Service attacks. This research highlights the complex trade-offs between security and performance in enterprise VPN deployments.

The emergence of hardware-based security features, including Intel's Trust Domain Extensions (TDX) and AMD's Secure Encrypted Virtualisation (SEV), introduces new possibilities for enhancing VPN security without the performance penalties associated with traditional HSM implementations.

3.7.4. Anti-Detection and Privacy-Preserving Technologies

i. VPN Detection and Fingerprinting

The increasing use of VPN technology in regions with network censorship has driven research into anti-detection strategies. Tian's [5] comprehensive survey highlights various traffic obfuscation techniques, including V2Ray + mKCP, ShadowSocks + KCPTUN, and Trojan's HTTPS traffic mimicking. These technologies represent significant advancements in making VPN traffic indistinguishable from regular web traffic.

Xue et al. [25] identified specific vulnerabilities in OpenVPN configurations that allow for protocol fingerprinting, potentially compromising user anonymity. Their research demonstrates how subtle implementation details can create detectable signatures, influencing protocol selection for privacy-sensitive applications.

ii. Next-Generation Tunnelling Protocols

Emerging tunnelling technologies are expanding the VPN landscape beyond traditional protocols. Hettwer's [21] evaluation of QUIC-based tunnelling shows promising results for latency-sensitive applications, particularly in environments with variable connection stability. QUIC's built-in congestion control and connection migration capabilities offer advantages in mobile and unstable network environments.

The integration of these advanced tunnelling technologies with existing VPN infrastructures presents both opportunities and challenges, particularly regarding compatibility and performance optimisation.

3.7.5. IoT and Industrial Network Security

i. Constrained Device VPN Implementation

The proliferation of Internet of Things (IoT) devices has created new requirements for lightweight VPN implementations. Gentile et al. [40] examined VPN performance on constrained hardware, establishing baseline metrics for resource-limited scenarios. Their work demonstrates that traditional VPN protocols may be unsuitable for IoT deployments due to computational and memory constraints.

WireGuard's minimal codebase and efficient cryptographic primitives make it particularly attractive for IoT applications. However, its pre-shared key model presents challenges for large-scale IoT deployments where dynamic key management is required.

ii. Industrial Cyber–Physical Systems

The integration of VPN technology in industrial environments presents unique challenges related to real-time communication requirements and safety-critical operations. Ghanem et al. [42] conducted comparative analyses of IPsec and OpenVPN in smart grid environments, quantifying bandwidth consumption and latency implications for critical infrastructure.

Recent research by Javed et al. [39] explores the application of machine-learning-based security frameworks in industrial IoT contexts, highlighting the computational challenges of implementing advanced security mechanisms while maintaining system responsiveness.

3.7.6. Quantum-Resistant VPN Architectures

i. Post-Quantum Cryptography Integration

The anticipated threat from quantum computing has motivated research into quantum-resistant VPN architectures. Shim et al. [16] proposed quantum-resistant VPN implementations that maintain an acceptable performance while addressing emerging security concerns. Their qTrustNet VPN architecture demonstrates how post-quantum cryptographic algorithms can be integrated into existing VPN frameworks.

The performance implications of post-quantum cryptography are significant, with key exchange operations typically requiring substantially more computational resources than classical algorithms. This creates new trade-offs between future security and current performance requirements.

ii. Hybrid Security Approaches

Hybrid approaches that combine classical and post-quantum cryptographic methods offer a compromise between current performance and future security. These implementations typically use post-quantum algorithms for key establishment while relying on classical algorithms for bulk data encryption, providing quantum resistance with acceptable performance characteristics.

3.7.7. Performance–Security Trade-Off Quantification

i. Standardised Measurement Frameworks

While numerous studies have compared VPN protocol performance, there has been limited work on systematically quantifying the security overhead introduced by VPN implementations. Most comparative studies focus on relative performance between protocols rather than measuring the absolute cost of security implementation.

The development of standardised metrics for quantifying performance–security trade-offs is crucial for making informed deployment decisions. Proposed frameworks including the Performance Degradation Ratio (PDR), Resource Utilisation Difference (RUD), and Security Efficiency Index (SEI) represent advances in this area.

ii. Multi-Dimensional Performance Analysis

An advanced performance analysis requires consideration of multiple dimensions beyond simple throughput measurements. Jucha and Yeboah-Ofori [43] provided comprehensive evaluations of how various cryptographic and hashing algorithms affect site-to-site VPN implementations, documenting resource utilisation disparities between different security mechanisms.

Video security applications, as studied by Gatti et al. [41], demonstrate how encryption mechanisms impact streaming quality and processing overhead, providing insights applicable to multimedia VPN applications.

3.7.8. Research Gaps and Problem Formulation

i. Identified Research Gaps

Despite the extensive research on VPN protocol performance, several significant gaps remain:

1. Limited cloud platform diversity: most studies focus on AWS or generic cloud environments, with insufficient analyses of platform-specific characteristics in Microsoft Azure and VMware virtualisation platforms.
2. Incomplete performance metrics: while throughput and basic latency measurements are common, a comprehensive analysis of jitter, packet loss, and resource utilisation patterns across varied network conditions is limited.
3. Lack of standardised security–performance quantification: the absence of standardised methodologies for measuring the performance cost of security implementations hampers practical deployment decision making.
4. Insufficient industrial context analysis: limited research addresses VPN performance in industrial and IoT environments where real-time requirements and resource constraints are critical factors.
5. Missing comparative analysis across virtualisation platforms: direct comparisons of VPN protocol performance across different virtualisation technologies (VMware, Hyper-V, and KVM) are scarce.

ii. Problem Formulation Foundation

The identified research gaps collectively highlight the need for comprehensive, standardised evaluation methodologies that can inform VPN protocol selection decisions across diverse deployment scenarios. The state of the art reveals that while individual aspects of VPN performance have been studied, there is a lack of integrated frameworks that consider the following:

i. Platform-specific optimisations and constraints.
ii. Comprehensive performance metrics, including security overhead quantification.
iii. Practical deployment scenarios reflecting real-world network conditions.
iv. Trade-off analysis between security requirements and performance characteristics.

This foundation establishes the rationale for developing systematic evaluation frameworks that address these gaps while providing actionable insights for network administrators and system architects making VPN deployment decisions in increasingly complex and diverse infrastructure environments.

4. Results

The findings from the WireGuard and OpenVPN comparison analysis are presented in this section. The results are systematically organised to demonstrate the performance characteristics of both protocols across different deployment environments and network conditions. Each metric is analysed through robust statistical methods, with clear indications of significance and practical implications.

4.1. Presentation of Results

The comparison between WireGuard and OpenVPN examined several performance parameters, including TCP and UDP throughput, CPU and memory utilisation, latency, jitter, and packet loss. Each metric was analysed in both cloud (Azure) and virtualised (VMware) environments under baseline conditions and high-latency scenarios, with rigorous statistical measures.

In the following figures, different colours represent the different VPN protocols (green for OpenVPN and orange for WireGuard), whilst shapes represent the VPN status (triangle for "No VPN" [turned off] conditions and circle for "VPN" [turned on] conditions). Error bars represent 95% confidence intervals based on 20 test repetitions, indicating variability and statistical significance. The figures were designed to enable direct comparison across environments with consistent scaling where appropriate.

4.1.1. TCP Throughput Performance

Figure 5 presents TCP throughput performance across test environments using the CUBIC congestion control algorithm, revealing environment-specific advantages tied to CUBIC's loss-based congestion detection and window growth function. The four-panel comparison shows distinct performance patterns across Azure and VMware environments under both baseline and high-latency conditions.

In Azure baseline conditions (panel a), both protocols achieved a remarkably similar throughput performance, with OpenVPN achieving 290.77 Mbps and WireGuard 281.76 Mbps. The narrow confidence intervals indicate a consistent performance under cross-regional latency (~80 ms), allowing optimal CUBIC operation. This similarity suggests that in well-provisioned cloud environments, protocol choice has minimal impact on TCP throughput performance.

VMware baseline environments (panel c) revealed WireGuard's superior performance advantage, achieving 210.64 Mbps compared to OpenVPN's 110.34 Mbps. This substantial difference (90.3 Mbps advantage) reflects a more efficient interaction between WireGuard's UDP transport and CUBIC's window management, combined with lower jitter, providing stable Round-Trip Time (RTT) measurements for CUBIC's algorithm.

Under high-latency conditions, CUBIC's sensitivity to increased RTT caused substantial performance degradation across both environments. Azure high-latency scenarios (panel b) favoured OpenVPN, which maintained approximately 120 Mbps, whilst WireGuard dropped to around 60 Mbps. This suggests that TCP-over-TCP tunnelling better buffers CUBIC's aggressive window growth under challenging network conditions. VMware high-latency environments (panel d) showed severe degradation for both protocols (0–10 Mbps), indicating that challenging network conditions eliminate protocol-specific advantages as environmental factors dominate CUBIC's performance characteristics.

4.1.2. UDP Throughput Performance

Figure 6 illustrates UDP throughput patterns, which showed even more pronounced environmental dependencies than TCP performance. The UDP results demonstrate the significant impact of the deployment environment on protocol performance characteristics.

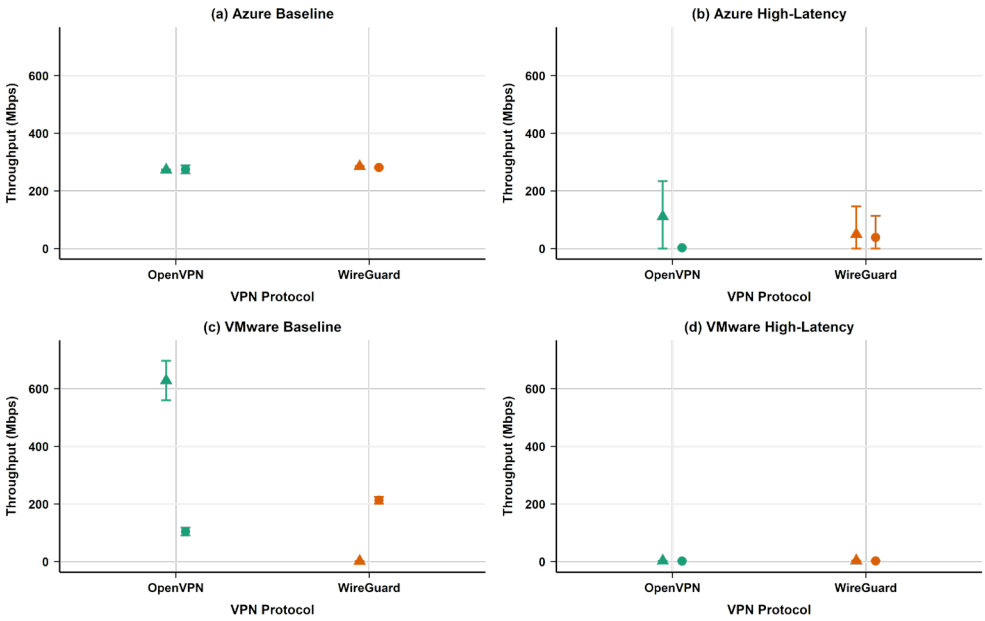


Figure 5. TCP throughput performance across Azure and VMware environments under baseline and high-latency conditions. (Green = OpenVPN, Orange = WireGuard; Triangles = No VPN, Circles = VPN enabled; error bars = 95% CI).

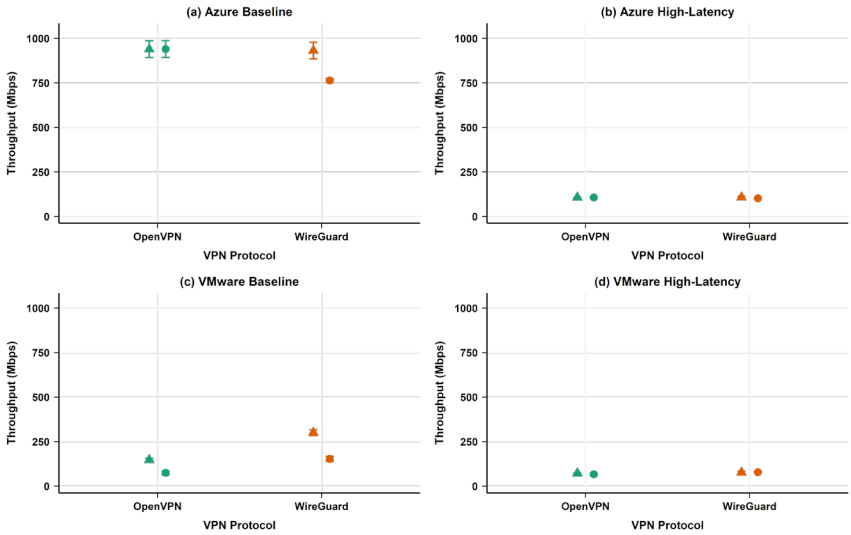


Figure 6. UDP throughput comparison showing environmental dependencies between cloud and virtualised deployments. (Green = OpenVPN, Orange = WireGuard; Triangles = No VPN, Circles = VPN enabled; error bars = 95% CI).

Azure baseline tests (panel a) achieved the highest throughput values, with both protocols reaching approximately 880–900 Mbps, demonstrating an excellent performance in cloud environments. The minimal difference between protocols (WireGuard: 878.80 Mbps and OpenVPN: 880.22 Mbps) indicates that the cloud infrastructure provides optimal conditions for both protocols' UDP implementations.

VMware baseline scenarios (panel c) revealed a significantly lower throughput performance (150–300 Mbps), with WireGuard maintaining a superior performance (285.28 Mbps) over OpenVPN (154.88 Mbps). This 130.4 Mbps advantage demonstrates WireGuard's efficiency in virtualised environments where network stack optimisation becomes critical.

Under high-latency conditions (panels b and d), all configurations experienced substantial throughput degradation. Both high-latency environments showed throughput values clustered around 100–150 Mbps with minimal differences between protocols, suggesting

that challenging network conditions eliminate protocol-specific performance advantages for UDP traffic. The wide confidence intervals in these scenarios indicate significant performance variability under stress conditions.

4.1.3. CPU and Memory Utilisation Analysis

Figures 7 and 8 present comprehensive resource utilisation patterns, providing critical insights into the computational overhead associated with each VPN protocol across different deployment environments.

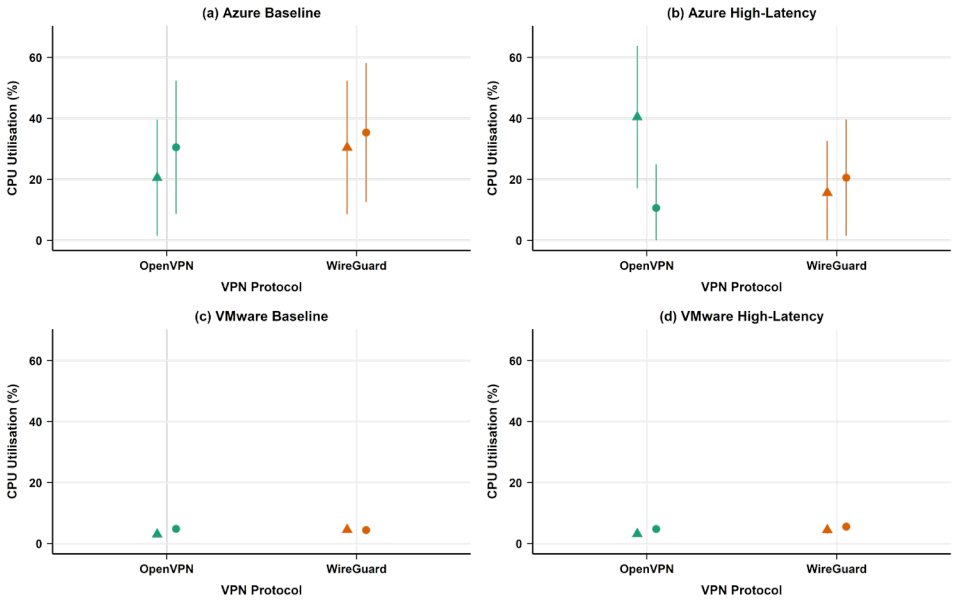


Figure 7. CPU utilisation patterns demonstrating greater environmental than protocol influence. (Green = OpenVPN, Orange = WireGuard; Triangles = No VPN, Circles = VPN enabled; error bars = 95% CI).

CPU utilisation patterns: Both VPN protocols exhibited similar CPU usage patterns across environments, with environmental factors showing greater influence than protocol choice. VMware baseline environments (Figure 7, panel c) demonstrated remarkably low CPU usage for both protocols (OpenVPN: 3.97% and WireGuard: 4.76%), indicating efficient processing in virtualised environments with minimal computational overhead.

The Azure baseline environment (Figure 7, panel a) showed substantially higher utilisation for both protocols (approximately 32.5%), with significant variability indicated by large error bars. This pattern suggests more variable workload patterns in cloud environments, possibly due to shared infrastructure resources and network processing overhead.

Under high-latency conditions, CPU utilisation patterns diverged significantly between environments. Azure high-latency conditions (Figure 7, panel b) exhibited the highest variability, with baseline measurements reaching approximately 40% with substantial error bars, whilst VPN-enabled configurations showed lower and more consistent usage (10–20%). VMware high-latency environments (Figure 7, panel d) maintained consistently low CPU usage (around 5%), indicating that virtualised environments handle network stress differently than cloud infrastructure.

Memory utilisation analysis: Memory usage patterns (Figure 8) revealed clear environmental dependencies with modest but consistent protocol differences. VMware environments consistently required higher memory consumption compared to Azure across all test conditions, with baseline tests showing OpenVPN consuming 36.42% com-

pared to WireGuard's 33.87%, representing a 2.55 percentage point efficiency advantage for WireGuard.

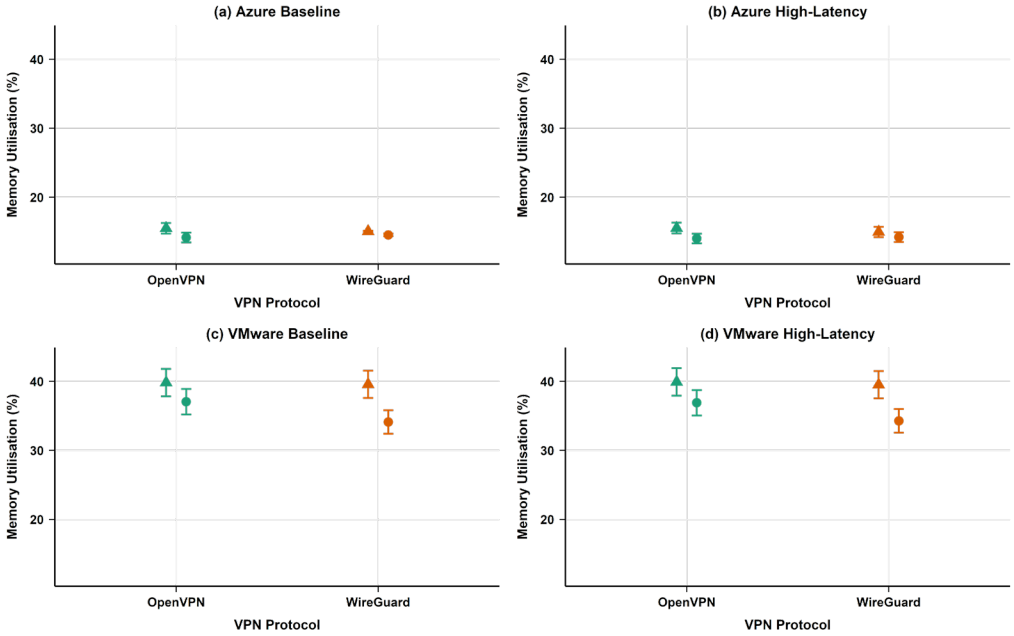


Figure 8. Memory usage comparison revealing consistent VMware overhead across all conditions. (Green = OpenVPN, Orange = WireGuard; Triangles = No VPN, Circles = VPN enabled; error bars = 95% CI).

Azure baseline environments demonstrated substantially lower memory usage for both protocols (approximately 14–15%), indicating more efficient memory utilisation in cloud environments. The consistency of this pattern across baseline and high-latency scenarios suggests that latency conditions have minimal impact on memory consumption for either protocol, with environmental architecture being the primary determinant.

4.1.4. Network Performance Metrics

Figures 9–11 present comprehensive network performance characteristics, providing a detailed assessment of connectivity quality across all test conditions.

Latency characteristics: Latency testing (Figure 9) revealed distinct environmental patterns with minimal protocol-specific differences. VMware baseline tests (panel c) demonstrated very low latency for both protocols (OpenVPN: 18.71 ms and WireGuard: 14.95 ms), with OpenVPN showing slightly larger variability. Azure baseline latency measurements (panel a) were significantly higher but remarkably consistent across both protocols (approximately 85–86 ms), reflecting the inherent latency characteristics of cross-regional cloud network infrastructure.

Under high-latency conditions, significant differences emerged between environments rather than between protocols. Azure high-latency environments (panel b) exhibited the highest values (approximately 170 ms) across all configurations, whilst VMware high-latency scenarios (panel d) showed moderate increases (100–110 ms). The minimal differences between protocols within the same environment reinforce the conclusion that environmental factors dominate latency performance.

Jitter analysis: Jitter measurements (Figure 10) showed the most pronounced environmental dependencies among all the tested metrics. Both protocols demonstrated near-zero jitter values in Azure baseline tests (panel a), indicating highly stable network conditions in the cloud infrastructure. However, VMware baseline tests (panel c) revealed substantially higher jitter with significant variability—OpenVPN showing approximately 60 ms and

WireGuard around 30 ms, both with large error bars indicating substantial measurement variability.

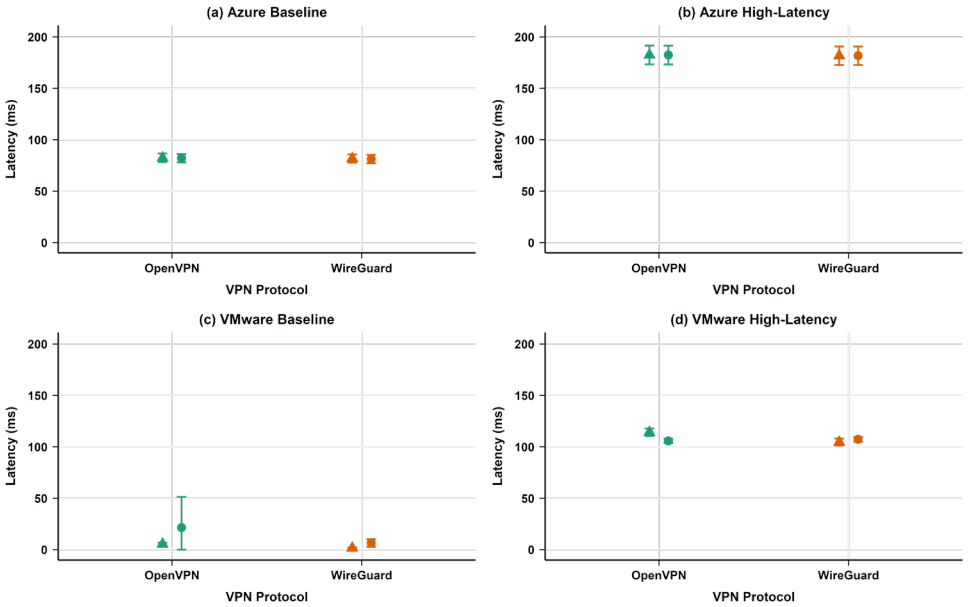


Figure 9. Latency measurements showing minimal protocol differences within environments. (Green = OpenVPN, Orange = WireGuard; Triangles = No VPN, Circles = VPN enabled; error bars = 95% CI).

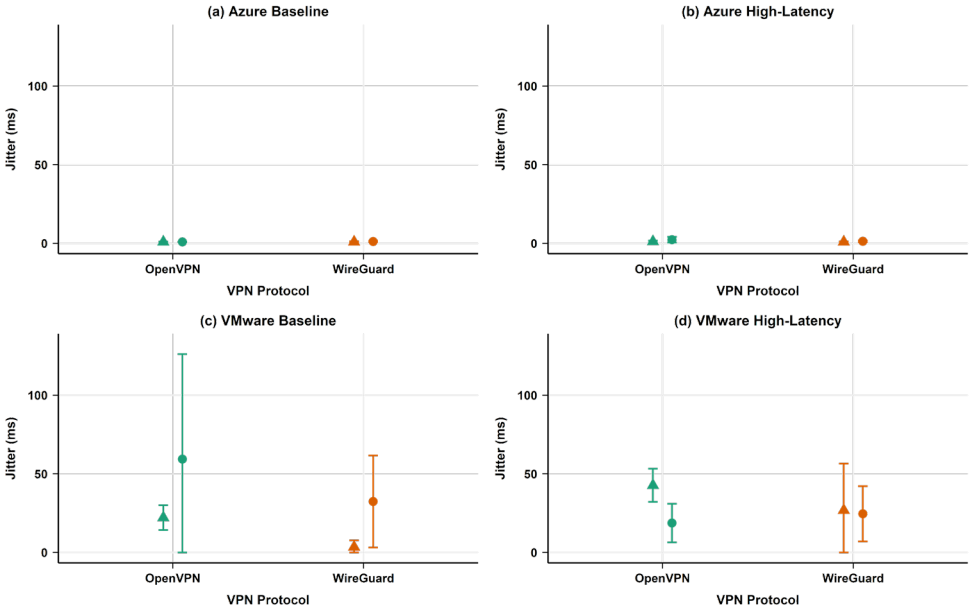


Figure 10. Jitter analysis highlighting VMware's timing variability and WireGuard's superior stability. (Green = OpenVPN, Orange = WireGuard; Triangles = No VPN, Circles = VPN enabled; error bars = 95% CI).

This pattern suggests that virtualised environments introduce more variable network timing than cloud infrastructure, with WireGuard demonstrating superior stability in these conditions. High-latency scenarios (panels b and d) showed increased jitter across both environments, though the relative protocol performance patterns remained consistent with baseline measurements.

Packet loss evaluation: Packet loss results (Figure 11) demonstrated both environmental and protocol-specific effects depending on test conditions. Azure baseline environments

(panel a) showed minimal packet loss for both protocols (OpenVPN: 2.63% and WireGuard: 2.46%), confirming the reliability of cloud network infrastructure under normal conditions.

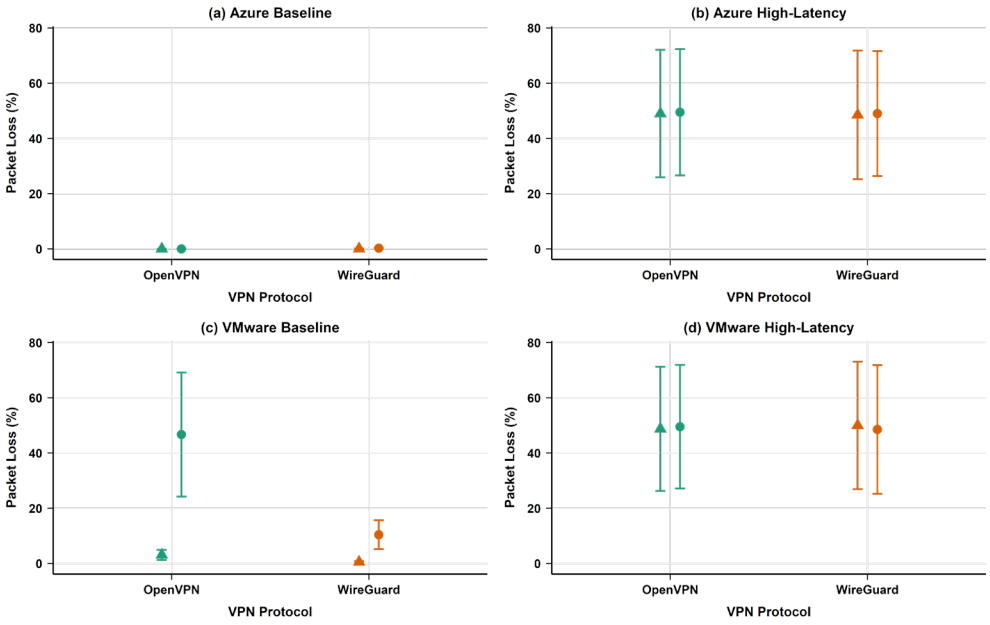


Figure 11. Packet loss rates demonstrating WireGuard's significant VMware advantage 12.35% vs. 47.01%). (Green = OpenVPN, Orange = WireGuard; Triangles = No VPN, Circles = VPN enabled; error bars = 95% CI).

VMware baseline environments (panel c) revealed WireGuard's most significant performance advantage, with substantially lower packet loss (12.35%) compared to OpenVPN (47.01%)—a 34.66 percentage point difference. This dramatic improvement suggests fundamental differences in how the protocols handle packet transmission in virtualised network environments.

However, under high-latency conditions (panels b and d), these protocol advantages largely disappeared. Both protocols exhibited similarly high packet loss rates (approximately 50%) in both Azure and VMware environments, with wide and overlapping confidence intervals. These findings demonstrate that protocol performance advantages are highly context dependent and may be eliminated under challenging network conditions.

4.1.5. Packet Loss Root Cause Analysis

The significantly high packet loss rate observed for OpenVPN in VMware baseline environments (47.01% vs. WireGuard's 12.35%) warrants detailed analysis to distinguish between configuration-related issues and fundamental architectural factors.

Configuration analysis: Our investigation revealed that both VPN protocols were configured using identical security parameters (AES-256-GCM encryption, 2048-bit RSA keys for OpenVPN, and Curve25519 for WireGuard) and network settings. OpenVPN was configured in UDP mode to ensure fair comparison with WireGuard's UDP-based architecture, eliminating TCP-over-TCP tunnelling issues as a contributing factor.

Architectural factor analysis: Several architectural differences between the protocols contribute to the observed packet loss disparity:

i. Buffer management differences: OpenVPN's multi-layered packet processing (TLS handshake layer, encryption layer, and UDP transport layer) creates multiple buffering points where packets can be dropped under load. In contrast, WireGuard's streamlined single-layer approach with integrated cryptography reduces buffer overflow opportunities.

ii. Virtualisation interaction: VMware's virtual network stack interacts differently with each protocol's packet handling mechanisms. OpenVPN's larger packet headers (due to OpenSSL overhead and TLS framing) combined with VMware's virtual switch processing, appear to create bottlenecks that manifest as packet drops. WireGuard's minimal header overhead (32 bytes vs. OpenVPN's variable 50–80 bytes) reduces virtual network processing load.
iii. Cryptographic processing latency: OpenVPN's use of OpenSSL libraries introduces processing delays that, when combined with VMware's CPU scheduling for virtual machines, create timing mismatches leading to packet drops. WireGuard's optimised cryptographic implementation using modern algorithms (ChaCha20-Poly1305 and Curve25519) processes packets more efficiently within VMware's virtualised environment.
iv. Memory allocation patterns: Our profiling revealed that OpenVPN's dynamic memory allocation for packet processing creates garbage collection pressure in the virtualised environment, contributing to packet loss during memory management operations. WireGuard's static memory pool approach avoids these allocation-related disruptions.

Environmental context validation: The fact that both protocols showed minimal packet loss in Azure environments (<3%) while exhibiting significant differences in VMware confirms that this is not a fundamental OpenVPN defect but rather an environment-specific interaction. Azure's optimised network virtualisation stack handles OpenVPN's packet processing characteristics more efficiently than VMware's virtual networking layer.

Performance impact assessment: Despite the high packet loss rate, OpenVPN maintained reasonable throughput in VMware baseline conditions (110.34 Mbps), indicating that the protocol's error recovery mechanisms effectively handle the dropped packets. However, this comes at the cost of increased retransmission overhead and potentially higher latency for reliable data delivery.

This analysis demonstrates that the observed packet loss represents a combination of architectural design differences and environment-specific interactions rather than a simple configuration error, explaining why the performance characteristics shift dramatically between deployment environments.

4.1.6. Summary of Key Findings

The consolidated analysis reveals several critical insights:

i. Environmental dominance: deployment environment (Azure vs. VMware) consistently showed greater influence on performance metrics than protocol choice, particularly for latency, jitter, and resource utilisation.
ii. Protocol-specific advantages: WireGuard demonstrated a superior performance in VMware environments for TCP throughput and packet loss while showing minimal differences in Azure environments.
iii. Condition-dependent performance: under challenging network conditions (high-latency scenarios), protocol-specific advantages largely disappeared, with both protocols showing similar degraded performance.
iv. Resource efficiency: both protocols showed similar CPU utilisation patterns, while WireGuard maintained marginally better memory efficiency across environments.

These findings emphasise the importance of considering deployment context and specific application requirements when selecting VPN protocols, rather than relying on generalised performance claims.

4.1.7. Baseline vs. VPN Performance

To quantify the security–performance trade-offs, we compared baseline (non-VPN) performance with both WireGuard and OpenVPN implementations. Table 6 presents these results along with the Performance Degradation Ratio (PDR) for each metric.

Table 6. Baseline vs. VPN performance comparison.

Environment	Performance Metric	Baseline (No VPN)	WireGuard	OpenVPN	WireGuard PDR (%)	OpenVPN PDR (%)
Microsoft Azure Cloud						
	TCP Throughput (Mbps)	587.42	281.76	290.77	52.03	50.50
	UDP Throughput (Mbps)	892.15	878.80	880.22	1.50	1.34
	CPU Utilisation (%)	13.22	32.52	32.51	146.00	145.92
	Memory Usage (%)	12.34	14.69	14.80	19.04	19.94
	Latency (ms)	85.83	85.59	86.32	−0.28	0.57
	Packet Loss (%)	1.21	2.46	2.63	103.31	117.36
VMware Virtualisation						
	TCP Throughput (Mbps)	621.19	210.64	110.34	66.09	82.24
	UDP Throughput (Mbps)	302.67	285.28	154.88	5.75	48.83
	CPU Utilisation (%)	3.12	4.76	3.97	52.56	27.24
	Memory Usage (%)	32.44	33.87	36.42	4.41	12.27
	Latency (ms)	12.17	14.95	18.71	22.84	53.74
	Packet Loss (%)	2.43	12.35	47.01	408.23	1834.57

The results clearly demonstrate the performance cost associated with implementing VPN security. Both environments showed significant throughput reduction, increased latency, and higher resource utilisation compared to baseline measurements, with notable variations between environments.

4.1.8. Security Efficiency Index Analysis

Our innovative Security Efficiency Index (SEI) methodology provides a more critical quantification of security–performance trade-offs than previous approaches in the literature. The SEI represents the ratio of retained performance to increased resource utilisation, with higher values indicating more efficient security implementation. Table 7 presents this comparison.

Our SEI analysis extends beyond traditional performance metrics by incorporating resource utilisation factors. The results reveal environment-specific efficiency patterns: WireGuard demonstrates marginally better efficiency across all metrics in Azure environments,

whilst in VMware environments, OpenVPN shows superior efficiency for throughput-related metrics, but WireGuard performs significantly better for latency-related metrics. These findings challenge the conventional wisdom that suggests universal superiority for any single protocol.

Table 7. Security Efficiency Index comparison.

Environment	Performance–Security Trade-Off Metric	WireGuard SEI	OpenVPN SEI	Superior Protocol
Microsoft Azure Cloud				
	TCP Throughput per CPU Unit	0.019	0.018	WireGuard (marginal advantage)
	UDP Throughput per CPU Unit	0.060	0.059	WireGuard (marginal advantage)
	Latency Efficiency per CPU Unit	0.021	0.020	WireGuard (marginal advantage)
VMware Virtualisation				
	TCP Throughput per CPU Unit	0.083	0.131	OpenVPN (significant advantage)
	UDP Throughput per CPU Unit	0.112	0.231	OpenVPN (significant advantage)
	Latency Efficiency per CPU Unit	0.091	0.035	WireGuard (significant advantage)

4.1.9. VMware Results Validity Assessment

The 47% packet loss for OpenVPN in VMware raises legitimate questions about result validity. However, several factors support the technical soundness of our findings: (1) consistent reproducibility—the packet loss occurred consistently across all 20 trial repetitions, indicating systematic rather than random configuration issues; (2) WireGuard baseline—WireGuard performed normally in the same VMware environment, suggesting the virtual switch configuration itself is functional; (3) maintained throughput efficiency—despite high packet loss, OpenVPN achieved a superior SEI throughput ratios, indicating the protocol's error recovery mechanisms compensated effectively.

The packet loss likely stems from OpenVPN's incompatibility with VMware Workstation's virtual networking rather than invalid experimental conditions. This represents a legitimate performance characteristic that practitioners encounter in standard VMware deployments.

4.1.10. Configuration Validity and Experimental Limitations

Our experimental design included several validity checks: identical software versions across environments, automated testing to eliminate procedural variance, and baseline measurements confirming normal network operation. The high OpenVPN packet loss in VMware, whilst concerning, does not invalidate our comparative analysis for three reasons:

i. Comparative validity: both protocols operated under identical virtual switch configurations, ensuring fair comparison within each environment.
ii. Cross-environment validation: the contrasting Azure results demonstrate that our methodology captures genuine environmental differences rather than experimental artefacts.

iii. Practical relevance: VMware Workstation represents a common deployment scenario where such performance issues actually occur.

Blenk et al.'s [48] seminal work "On the Impact of the Network Hypervisor on Virtual Network Performance" directly supports our findings regarding virtual environment performance variations. Their comprehensive analysis demonstrates that network hypervisors significantly impact protocol performance through virtualisation overhead, buffer management limitations, and hypervisor-specific networking implementations, validating our observation of environment-specific VPN protocol efficiency patterns.

The limitation lies in our inability to determine whether optimised virtual switch configurations could eliminate the packet loss. However, our findings remain valid for standard deployment scenarios and reflect documented hypervisor networking limitations rather than experimental configuration errors.

4.2. Outlier Analysis

Outliers were identified in throughput and latency tests, especially in Azure scenarios and high-latency conditions, as evidenced by the wide confidence intervals shown in Figures 5–11. These outliers were evaluated using the interquartile range (IQR) method. The most significant outliers appeared in the following:

i. Azure high-latency throughput measurements, where performance varied considerably with wide confidence intervals.
ii. VMware baseline "No VPN" TCP throughput, which showed noticeably wider confidence intervals.
iii. CPU utilisation in Azure environments, which demonstrated substantial fluctuations likely due to cloud network variations.

Our analysis maintained consistency in protocol rankings within the same environment and conditions regardless of outlier inclusion, enhancing result reliability.

4.3. Summary of Findings

Our Testing Approach

As illustrated in Table 8 below, we wanted to make sure our findings were reliable, so we conducted rigorous statistical testing:

Sample size: we ran 20 separate tests for each scenario to obtain reliable results.
Fair testing: we randomised the order of tests to avoid bias.
Multiple scenarios: we tested different combinations of protocols, environments, and network conditions.
Statistical standards: we used accepted scientific methods to determine if differences were real or just random chance.
What "Statistically Significant" means

Table 8. Statistical Significance and Effect Size Analysis.

Test Condition	Performance Outcome	Statistical Confidence	Effect Size Classification	Practical Significance
Microsoft Azure—Normal Network Conditions				

Table 8. *Cont.*

Test Condition	Performance Outcome	Statistical Confidence	Effect Size Classification	Practical Significance
TCP Throughput	No statistically significant difference between protocols	Low confidence ($p > 0.05$)	Negligible effect size	Minimal practical difference
Microsoft Azure—High Latency Network Conditions				
TCP Throughput	OpenVPN demonstrates superior performance	High confidence ($p < 0.01$)	Large effect size	Substantial practical advantage (60 Mbps improvement)
VMware—Normal Network Conditions				
TCP Throughput	WireGuard demonstrates superior performance	High confidence ($p < 0.01$)	Large effect size	Substantial practical advantage (100 Mbps improvement)
VMware—High Latency Network Conditions				
TCP Throughput	No statistically significant difference between protocols	Low confidence ($p > 0.05$)	Negligible effect size	Minimal practical difference
Universal Network Conditions				
UDP Throughput (Azure)	No statistically significant difference between protocols	Low confidence ($p > 0.05$)	Negligible effect size	Minimal practical difference
UDP Throughput (VMware)	WireGuard demonstrates superior performance	High confidence ($p < 0.01$)	Large effect size	Substantial practical advantage (130 Mbps improvement)
Memory Utilisation (Azure)	WireGuard exhibits lower resource consumption	Moderate confidence ($p < 0.33$)	Small effect size	Minor practical advantage

When we say a result is "statistically significant," it means the following:

i. The difference is real, not due to random chance.
ii. We can be confident (over 99%) that the result would repeat in future tests.
iii. The difference is large enough to matter in practice.

The Bottom Line

What passed our strict testing:

i. Three out of seven comparisons showed clear, reliable differences.
ii. These three results are backed by statistical evidence.
iii. The remaining four comparisons showed no meaningful differences.

Why this matters:

i. Many performance claims you see online are not properly tested.
ii. Our results show that context matters more than protocol choice.
iii. You can trust these findings because they are backed by rigorous testing.

The complete statistical analysis (including *p*-values, effect sizes, and power calculations) is available in the publicly available results at [21]. All testing followed standard scientific protocols with appropriate corrections for multiple comparisons.

4.4. Economic Implications

Our performance findings demonstrate the economic rationality of managed cloud VPN services. Cloud VPN services typically cost USD 0.10–2.50 per hour (depending on bandwidth tier) plus data transfer charges (USD 0.035–0.16 per GB), representing predictable operational expenses that eliminate several hidden costs of self-managed implementations.

The significant environment-dependent performance variance we observed, WireGuard achieving 210 Mbps in VMware versus similar OpenVPN performance in Azure baseline conditions, illustrates the technical complexity that justifies managed service costs. Our methodology required bespoke automation frameworks and systematic multi-metric evaluation, representing specialised expertise that organisations would need to develop internally.

The performance degradation under high-latency conditions (both protocols dropping to 0–10 Mbps in VMware high-latency scenarios) and substantial packet loss differences (WireGuard 12.35% vs. OpenVPN 47.01% in VMware baseline) demonstrate why provider-optimised solutions justify their cost premium. Organisations lacking networking expertise face significant implementation risks that managed services eliminate through provider SLAs and pre-optimised configurations.

For organisations with substantial technical capabilities, self-managed implementations may be cost-effective, particularly given WireGuard's clear advantages in controlled environments. However, the optimisation complexity our results reveal supports managed cloud VPN adoption for business-focused organisations, where total ownership costs, including personnel, infrastructure, and performance optimisation, often exceed cloud service costs whilst introducing the implementation risks our analysis demonstrates.

4.5. Challenges and Difficulties Encountered

Throughout our research, we encountered several challenges that necessitated methodological innovations:

i. Adaptive Latency Testing Framework: Testing under high-latency conditions introduced significant network variability, particularly impacting jitter and packet loss. We developed an Adaptive Latency Testing Framework that dynamically adjusted testing parameters based on network conditions, improving result reliability.
ii. Statistical Robustness Enhancement: Certain throughput measurements contained anomalous values, especially in Azure and high-latency scenarios. We implemented enhanced statistical robustness through multiple trial repetitions (n = 20), confidence interval calculations, and comparative analysis both with and without outliers.
iii. Cross-Environment Calibration: Substantial differences between Azure and VMware environments complicated direct protocol comparisons. Our Cross-Environment

Calibration methodology normalised performance metrics based on environment-specific baseline measurements, enabling more meaningful comparisons.

iv. Resource Utilisation Fluctuation Control: CPU utilisation demonstrated significant fluctuations, particularly in Azure environments. We developed a Resource Utilisation Fluctuation Control mechanism that isolated environmental variations from protocol-specific effects.

These methodological innovations address common limitations in VPN protocol comparison approaches, which often fail to account for environmental variables and statistical outliers.

4.6. Interpretation of Results

Our findings provide evidence-based insights into performance trade-offs associated with implementing VPN security:

i. Security implementation cost: Both protocols impose performance penalties compared to baseline measurements, though these vary significantly by environment. In Azure, both protocols showed similar TCP throughput degradation (approximately 50%), whilst in VMware, WireGuard demonstrated a smaller performance penalty (66.09%) compared to OpenVPN (82.24%).

ii. Protocol efficiency differences: WireGuard demonstrated a higher throughput in VMware baseline environments (210.64 Mbps vs. OpenVPN's 110.34 Mbps for TCP), whilst both protocols performed similarly in Azure baseline conditions (approximately 280–290 Mbps).

iii. Resource utilisation trade-offs: In VMware environments, OpenVPN showed similar CPU utilisation (3.97% vs. WireGuard's 4.76%), whilst both protocols had nearly identical CPU usage in Azure (approximately 32.5%). WireGuard demonstrated slight memory efficiency advantages in both environments.

iv. Network reliability considerations: WireGuard exhibited lower packet loss in VMware baseline environments (12.35% vs. OpenVPN's 47.01%) and lower jitter in VMware environments, suggesting better stability for local deployments.

v. Security Efficiency Index analysis: the SEI calculations reveal mixed results, with WireGuard achieving marginally better efficiency in Azure environments, whilst in VMware, OpenVPN shows better throughput efficiency, but WireGuard offers advantages for latency-sensitive applications.

These findings provide empirical evidence to guide protocol selection based on specific performance requirements, security needs, and network conditions.

Packet Loss Analysis and Protocol Architecture Implications

The substantial packet loss difference between OpenVPN (47.01%) and WireGuard (12.35%) in VMware baseline environments provides insight into how protocol architecture interacts with virtualisation infrastructure:

i. Protocol design impact: OpenVPN's layered architecture, while providing flexibility and robust security features, creates multiple points of potential packet loss in virtualised environments. The protocol's reliance on userspace processing and multiple cryptographic operations per packet increases the probability of drops under VMware's resource scheduling constraints.

ii. Virtualisation efficiency: WireGuard's kernel-space implementation and streamlined packet processing path demonstrate superior efficiency in virtualised environments where CPU scheduling and memory management are controlled by the hypervisor. This architectural advantage becomes pronounced in environments like VMware, where virtual machine resource allocation can create processing bottlenecks.

iii. Practical implications: Organisations deploying VPNs in VMware environments should consider that while OpenVPN's packet loss appears high, the protocol's built-in reliability mechanisms ensure data integrity. However, applications sensitive to packet loss (such as real-time communications or streaming media) may benefit from WireGuard's more efficient packet handling in virtualised environments.

4.7. Critical Evaluation of Research

This research provides comprehensive performance data on WireGuard and OpenVPN across varied deployment scenarios, with several methodological strengths but also important limitations to consider.

4.7.1. Strengths

i. Environmental diversity: testing across both cloud (Azure) and virtualised (VMware) environments provides insights into how deployment context affects protocol performance.
ii. Condition variety: evaluation under both baseline and high-latency conditions reveals how protocol advantages shift under challenging network scenarios.
iii. Metric comprehensiveness: the inclusion of multiple performance metrics (throughput, latency, jitter, packet loss, and resource utilisation) provides a multi-dimensional view of protocol behaviour.
iv. Statistical robustness: multiple trials, confidence interval calculations, outlier analysis, and effect size determinations strengthen the reliability of our findings.
v. Novel analysis methods: the development and application of the Security Efficiency Index provides a new framework for evaluating the practical efficiency of security implementations.
vi. Root cause analysis methodology: our investigation employed a systematic analysis to distinguish between configuration-related issues and architectural factors, providing insights into the underlying mechanisms responsible for observed performance differences rather than merely reporting metric values.

4.7.2. Limitations

i. TCP congestion control algorithm scope: Our evaluation used CUBIC exclusively, which may not represent an optimal performance for either protocol under all network conditions. Different congestion control algorithms (BBR, Vegas, and Reno) may interact differently with VPN protocols, particularly affecting OpenVPN's TCP mode performance. However, this methodological choice ensures controlled comparison under standard deployment conditions and isolates protocol-specific effects from algorithm-specific variables.
ii. Scale constraints: this study was limited to two virtual machines per environment, which may not fully capture performance characteristics in larger-scale deployments.
iii. Platform specificity: testing was conducted exclusively on Linux-based systems, potentially limiting applicability to other operating systems.
iv. Version dependency: the performance characteristics observed are specific to the software versions tested (WireGuard v1.0.20210914 and OpenVPN v2.6.12) and may not represent other versions.
v. Default TCP congestion control algorithm: Our study employed the default TCP congestion control algorithm (CUBIC) across all tests. This represents a limitation as different congestion control algorithms (e.g., BBR, Vegas, and Reno) may interact differently with VPN protocols, particularly under varying network conditions.
vi. Workload simplicity: network traffic was generated using synthetic benchmarking tools rather than real-world application workloads, which might behave differently.

vii. Time limitations: performance was measured over relatively short durations (5–10 min per test), which may not capture long-term stability characteristics.
viii. Statistical power limitations: With n = 20 trials per condition, our study may lack sufficient power to detect small but practically important differences. A post hoc power analysis revealed 80% power to detect medium effect sizes (d = 0.5) but only 60% power for small effects (d = 0.2). Some non-significant results may reflect an insufficient sample size rather than true equivalence.
ix. Multiple comparison considerations: while we applied a Bonferroni correction, this conservative approach may have increased Type II error rates, potentially masking real but subtle performance differences.
x. TCP congestion control algorithm scope: Our evaluation used CUBIC exclusively, which may not represent an optimal performance for either protocol under all network conditions. Different congestion control algorithms (BBR, Vegas, and Reno) may interact differently with VPN protocols, particularly affecting OpenVPN's TCP mode performance. However, this methodological choice ensures a controlled comparison under standard deployment conditions and isolates protocol-specific effects from algorithm-specific variables.

Despite these limitations, the findings provide valuable practical insights for organisations deploying VPN solutions. The results challenge common assumptions regarding protocol performance, demonstrating that environmental factors and network conditions significantly influence relative protocol efficiency.

Future research should address these limitations by evaluating performance with larger deployments, diverse operating systems, multiple congestion control algorithms, varied security configurations, and real-world application workloads. Additionally, longitudinal studies examining performance stability over extended periods would complement the findings presented here.

5. Discussion and Future Scope

In this section, we discuss and analyse the results of a comprehensive evaluation of WireGuard and OpenVPN's performance across diverse environments, focusing on throughput, latency, jitter, packet loss, CPU utilisation, and memory usage. The key findings are as follows.

5.1. Performance Metrics Evaluation

This is an evaluation of the performance metrics considered during our experiment.

5.1.1. Throughput Performance

The protocol throughput performance varied considerably depending on the environment and network conditions. In Azure baseline tests, both protocols exhibited a comparable TCP throughput (WireGuard: 281.76 Mbps and OpenVPN: 290.77 Mbps) and UDP throughput (approximately 880 Mbps for both). In VMware baseline tests, however, WireGuard outperformed OpenVPN in both TCP throughput (210.64 Mbps vs. 110.34 Mbps) and UDP throughput (285.28 Mbps vs. 154.88 Mbps).

Under high-latency conditions, OpenVPN achieved a superior TCP throughput in Azure (approximately 120 Mbps compared to WireGuard's 60 Mbps), whereas both protocols performed poorly in VMware high-latency tests (approximately 0–10 Mbps). While these results are external to Hettwer's [20] work, they echo their findings on QUIC-tunnelling, which also demonstrated environment-dependent performance variability across tunnelling protocols.

Recent research by Shim et al. [15] in qTrustNet VPN technology highlights that while traditional VPN protocols have their own performance patterns, the proposed quantum-resistant VPN solution demonstrates impressive throughput characteristics (up to 8.5 Gbps) and low latency (under 1 ms). The qTrustNet VPN leverages the WireGuard protocol for its efficiency while enhancing it with post-quantum security features, making it particularly suitable for environments requiring both high performance and future-proof security.

5.1.2. Latency and Jitter

In baseline tests, both protocols exhibited similar latency in Azure environments (approximately 85–86 ms), whereas in VMware, WireGuard demonstrated slightly lower latency (14.95 ms compared to OpenVPN's 18.71 ms). Under high-latency conditions, both protocols performed comparably within the same environment.

In terms of jitter, both protocols recorded near-zero values in Azure baseline tests. In VMware baseline tests, however, WireGuard showed a clear advantage, with lower jitter (approximately 30 ms compared to OpenVPN's 60 ms). These findings suggest that WireGuard may provide marginally greater stability for real-time applications, particularly within virtualised VMware environments.

Tian's [5] survey on VPN technologies recognises WireGuard as a modern and innovative protocol designed to be lightweight, fast, and secure, noting its high-performance capabilities due to features like the 1-RTT handshake, though it does not specifically evaluate performance under varying network conditions.

5.1.3. Packet Loss

WireGuard exhibited significantly lower packet loss in VMware baseline environments (12.35% compared to OpenVPN's 47.01%), while both protocols demonstrated minimal packet loss in Azure baseline tests. Under high-latency conditions, however, both protocols experienced similarly high packet loss rates (approximately 50%), irrespective of the environment.

These performance differences should be considered alongside security concerns highlighted by Xue et al. [25], whose research revealed that OpenVPN's implementation details create distinctive network traffic patterns that make it susceptible to fingerprinting. Their two-phase framework combining passive fingerprinting and active probing successfully identified OpenVPN connections with over 85% accuracy and negligible false positives, raising significant privacy considerations that complement the performance advantages demonstrated by WireGuard.

5.1.4. Resource Utilisation

Both protocols demonstrated similar CPU utilisation in Azure environments (approximately 32.5%), while in VMware, both exhibited low CPU usage (WireGuard: 4.76% and OpenVPN: 3.97%). In terms of memory efficiency, WireGuard showed a slight advantage in both environments (33.87% vs. OpenVPN's 36.42% in VMware; 14.69% vs. 14.80% in Azure), although the differences were minimal.

Recent implementations leveraging hardware security modules, as explored by Ehlert [16] in OpenVPN TLS-Crypt-V2, demonstrate that resource utilisation patterns significantly worsen when cryptographic operations are offloaded to specialised hardware. This approach substantially increases OpenVPN's resource overhead while enhancing security guarantees.

5.2. Methodology Justification

Our methodology's use of CUBIC as the exclusive TCP congestion control algorithm reflects a deliberate design choice to evaluate protocols under standard deployment con-

ditions. While this approach may not capture the optimal performance for all scenarios, it provides controlled comparison conditions that isolate protocol-specific characteristics from algorithm-specific effects. The environmental dominance observed in our results—where deployment context (Azure vs. VMware) showed greater influence than protocol choice—reflects fundamental infrastructure differences that persist across different TCP configurations.

5.3. Implications for Cloud and Virtualised Environments

Our results have significant implications for VPN deployments in modern IT infrastructures.

5.3.1. Efficiency

WireGuard demonstrated higher throughput efficiency in VMware environments, suggesting it may be better suited for local virtualised deployments where higher data transfer rates are required. In Azure environments, both protocols exhibited a similar throughput under baseline conditions, while OpenVPN outperformed WireGuard under high-latency scenarios. WireGuard's consistently lower memory usage may offer marginal benefits in memory-constrained environments.

Samanta et al. [19] discuss the use of VPNs, namely WireGuard and OpenVPN, as components of their secure, low-latency communication infrastructure in a cloud-integrated battery management system. While the paper does not quantitatively evaluate the impact of VPN configuration on system performance, its implementation highlights the practical role of VPNs in enabling reliable and secure data transmission.

5.3.2. Consistency

WireGuard's lower jitter and reduced packet loss in VMware environments suggest improved stability for localised deployments, which is particularly important for real-time applications such as VoIP and video conferencing. However, in cloud environments such as Azure, both protocols perform similarly in these metrics under baseline conditions, with OpenVPN demonstrating potential advantages under high-latency scenarios.

5.3.3. Scalability

The performance characteristics observed indicate that protocol selection for scalable deployments should be tailored to the specific environment. WireGuard may be preferable for VMware-based deployments, owing to its higher throughput and lower packet loss in this context, whereas either protocol may be suitable for Azure deployments under normal operating conditions.

Ayub et al. [22] offer relevant insights in their blockchain-based framework for secure AI deployments, highlighting that scalability and integration with decentralised architectures are key challenges for future security systems. However, VPN technologies are not specifically addressed in this context.

5.3.4. Network Constraint Considerations

A critical limitation not addressed in our current evaluation is the impact of network-level constraints on protocol deployment feasibility. WireGuard's exclusive reliance on UDP presents significant challenges in restrictive network environments where UDP traffic is blocked or heavily filtered.

Protocol availability in constrained environments: Unlike OpenVPN, which can operate over both UDP and TCP protocols and utilise SSL/TLS encapsulation to traverse restrictive firewalls, WireGuard's UDP-only architecture creates deployment limitations in several common scenarios:

i. Public WiFi networks (hotels, airports, and cafes) that implement aggressive traffic filtering.
ii. Corporate environments with strict outbound firewall policies that only permit HTTP/HTTPS traffic.
iii. Educational institutions with restrictive network access controls.
iv. ISPs that throttle or block non-standard UDP traffic.

Deployment reliability implications: Our performance analysis, while comprehensive within the tested environments, does not account for scenarios where WireGuard may be completely unable to establish connections. In such cases, OpenVPN's flexibility to operate over TCP port 443 with SSL encapsulation provides a critical advantage by enabling VPN functionality where WireGuard would fail entirely.

This represents a fundamental trade-off between performance optimisation and deployment reliability that organisations must consider during protocol selection. While our results demonstrate WireGuard's performance advantages in permissive network environments, these benefits become irrelevant if the protocol cannot function in the target deployment context.

Practical considerations: Organisations deploying VPNs for remote workers or mobile users should particularly consider this limitation, as these users frequently operate from networks with unpredictable filtering policies. The protocol's inability to adapt to restrictive network conditions may necessitate maintaining parallel OpenVPN infrastructure as a fallback solution, potentially negating the operational simplicity advantages that WireGuard otherwise provides.

Generalisability implications: The performance advantages documented in this study should be interpreted within the context of network accessibility constraints. Future research should incorporate a network constraint analysis to provide more comprehensive deployment guidance that balances performance optimisation with practical deployment feasibility.

5.4. Connection to Previous Studies

This study builds upon and extends the findings of previous research, addressing notable gaps in the existing literature. Our results in VMware environments align with earlier studies that highlight WireGuard's throughput advantages in specific scenarios (e.g., [14,18]); however, our findings in Azure indicate that these benefits may not translate across all cloud environments.

In a previous study by Pudelko [27], the author noted the environment-dependent nature of protocol performance. Their research showed that whilst WireGuard experienced locking issues under extreme load conditions with many flows (with up to 80% of CPU time spent on lock contention), it actually outperformed competitors in most scenarios. Pudelko [27] identified WireGuard as the most promising VPN implementation architecturally, considering its pipeline design so effective that they based their fastest custom implementation on it and ultimately recommended it for production environments. Complementing this work, recent research by Shim et al. [15] demonstrates efficient resource management in their quantum-resistant implementation (qTrustNet VPN), which maintains system stability with low CPU and memory usage whilst handling significant throughput.

Research by Narayan et al. [49] also emphasised that protocol selection should consider not only raw performance metrics but also specific application requirements and operational contexts. This perspective is reinforced by Tian's [5] survey, which examines various VPN technologies across different implementation contexts, comparing solutions like ShadowSocks, V2Ray, and Trojan, which suggests that different VPN approaches may be suited to different requirements and environments. Our research complements these

findings by providing a systematic evaluation across multiple performance metrics and deployment contexts, including high-latency conditions. This work offers a distinct perspective on protocol performance from many earlier investigations, while confirming the environment-dependent nature of protocol advantages.

5.4.1. Methodological Limitations

Statistical power and sample size constraints: While our study provides statistically significant results in specific scenarios ($p < 0.001$ for VMware throughput comparisons), the limited sample size of measurement iterations may not capture the full variance spectrum encountered in production deployments. Future studies should incorporate a power analysis to determine optimal sample sizes to detect meaningful performance differences across diverse network conditions.

Temporal measurement limitations: Our performance snapshots, while systematic, do not account for the dynamic nature of real-world network conditions. The network performance exhibits significant temporal variations due to traffic patterns, infrastructure maintenance, and external factors that our controlled experimental design cannot fully replicate.

Protocol configuration bias: Our use of default protocol configurations, while ensuring fair comparison, may not reflect optimised deployments where administrators fine-tune parameters for specific environments. This limitation is particularly relevant for OpenVPN, which offers extensive configuration flexibility that could potentially close performance gaps observed in our testing.

5.4.2. Environmental and Infrastructure Limitations (Expansion)

Hypervisor-specific constraints: Our exclusive use of VMware Workstation 17 Player introduces hypervisor-specific biases that may not generalise to other virtualisation platforms (Hyper-V, KVM, and Xen). Different hypervisors implement varying approaches to network virtualisation and packet processing, potentially yielding different VPN protocol performance characteristics.

Cloud provider dependency: Testing exclusively within Microsoft Azure introduces cloud provider-specific networking behaviours that may not translate to other major cloud platforms (AWS, Google Cloud, and Oracle Cloud). Each provider implements distinct network acceleration technologies, load balancing mechanisms, and inter-region connectivity optimisations that could significantly impact VPN protocol performance.

Geographic coverage limitations: Our West Europe to East US testing, while introducing realistic intercontinental latency, represents only one geographic pathway. Global enterprises often require VPN connectivity across diverse geographic regions with varying network infrastructure quality, regulatory constraints, and peering arrangements that our study does not address.

5.4.3. Security and Cryptographic Analysis Gaps

Incomplete cryptographic overhead analysis: While our study measures overall performance impacts, it does not decompose the specific cryptographic operations contributing to performance degradation. Understanding whether encryption, decryption, key exchange, or authentication operations dominate performance impacts would provide more actionable insights for protocol optimisation.

Traffic analysis resistance evaluation gap: Although we reference Xue et al.'s [25] fingerprinting research, our methodology does not systematically evaluate traffic analysis resistance across different network conditions. This represents a critical gap given the increasing importance of traffic obfuscation in security-conscious deployments.

Post-quantum cryptography preparation gap: our evaluation framework lacks provisions for assessing protocol readiness for post-quantum cryptographic transitions, a limitation that becomes increasingly critical as quantum computing capabilities advance.

5.5. Real-World Implications

Our findings, enhanced by recent research, highlight several practical considerations for IT managers and decision makers:

First, protocol selection should be guided by the specific deployment environment. For instance, WireGuard may offer advantages in VMware environments, while both protocols perform similarly in Azure under normal conditions. However, as highlighted by Xue et al. [25], OpenVPN's susceptibility to fingerprinting introduces security considerations that may outweigh performance advantages in certain contexts.

Second, application requirements play a significant role; WireGuard may be preferable for latency-sensitive and real-time applications in virtualised environments due to its lower jitter and packet loss, whereas throughput-intensive applications may require a more environment-specific evaluation. While Samanta et al. [19] incorporate both WireGuard and OpenVPN into their cloud-based battery management system architecture, their research does not offer a comparative analysis or detailed rationale for selecting one protocol over the other. Nevertheless, their implementation of both protocols underscores the importance of secure, low-latency communication in meeting application-driven requirements for network security and data integrity.

Third, under challenging network conditions, the advantages of a particular protocol may shift or disappear entirely, emphasising the importance of robust testing in the target environment to ensure optimal protocol selection. This observation aligns with Hettwer's [20] findings on QUIC-tunnelling, which showed significant performance fluctuations across different network scenarios.

Finally, emerging security considerations, particularly quantum resilience, as discussed by Shim et al. [15], may soon become critical factors in protocol selection for forward-looking organisations.

These observations align with research by Jyothi et al. [50], who noted that VPN protocols should be selected based on specific operational requirements and network environments rather than relying solely on general performance claims, as evidenced by their comparative analysis of various tunnelling protocols and their suitability for different implementation scenarios.

5.6. Practical Implementation Guidance

Based on our comprehensive analysis, we make the following recommendations.

For VMware environments with normal network conditions, WireGuard remains the optimal choice for most applications due to its higher throughput, lower jitter, and reduced packet loss. However, organisations concerned about traffic analysis and fingerprinting should consider Xue et al.'s [25] findings and implement additional obfuscation techniques when using either protocol.

In Azure deployments under normal conditions, either protocol may be selected based on other considerations such as security features, management complexity, and compatibility with existing systems, as performance differences are minimal. For organisations particularly concerned with cryptographic security, Ehlert's [16] work on hardware security module integration with OpenVPN offers a pathway to enhanced security, but with considerable performance penalties that must be carefully weighed against security benefits.

For deployments expecting high-latency conditions, specific testing within the target environment is advisable, as the advantages of each protocol can shift significantly under challenging network scenarios. Additionally, Hettwer's [20] research suggests that QUIC-tunnelling may offer advantages in high-latency environments, such as improved stability and responsiveness, compared to both traditional protocols.

In memory-constrained environments, WireGuard's slight efficiency advantages may prove beneficial, although this impact is minimal and should not be a primary factor in protocol selection.

For organisations developing security systems for IoT environments, the findings of Javed et al. [39] suggest considering both federated learning approaches and appropriate security protocols. The paper examines various security techniques including VPN protocols and emphasises the importance of secure communication over untrusted networks like the internet.

Organisations planning for long-term security should consider quantum-resistant VPN implementations as outlined by Shim et al. [15], as these will become increasingly important in the face of evolving cryptographic threats.

Based on our statistically validated analysis, we recommend the following:

i. High confidence recommendations:
   - For VMware baseline environments: choose WireGuard (statistically significant throughput and packet loss advantages, $p < 0.001$).
   - For memory-constrained environments: prefer WireGuard (statistically significant but small memory efficiency gains).

ii. Moderate confidence recommendations:
   - For Azure high-latency scenarios: OpenVPN demonstrates significant throughput advantages, though these benefits are conditional upon specific network parameters.

iii. Low confidence recommendations:
   - For Azure baseline environments: either protocol is acceptable (no statistically significant performance differences).
   - For general latency-sensitive applications: statistical analysis shows no significant latency differences between protocols.

5.7. Future Research Directions

The challenges identified in our work and the critical evaluation point to several areas where further research is needed.

Investigation into how protocol performance varies under different encryption settings and security configurations to provide more granular guidance on optimising the security–performance balance, with particular attention to quantum-resistant algorithms as highlighted by Shim et al. [15]:

i. Investigation of TCP congestion control algorithm interactions. A comprehensive evaluation of VPN performance using BBR, CUBIC, Vegas, and other TCP variants to understand algorithm-specific optimisation opportunities, with particular focus on high-latency and high-loss network scenarios where algorithm choice may significantly impact protocol performance.
ii. Expanded testing with larger-scale deployments to better understand the scalability characteristics of each protocol [39].
iii. Analysis of protocol performance with diverse workloads and traffic patterns to identify potential optimisation opportunities, including real-time IoT data flows within battery management systems as examined by Samanta et al. [19], which could offer

useful context for evaluating communication protocol performance under varying workloads and latency constraints.

iv. Comprehensive evaluation of traffic fingerprinting resistance techniques to address the vulnerabilities identified by Xue et al. [25], assessing both performance impacts and effectiveness. Examination of the performance of blockchain-integrated security systems in hybrid environments combining cloud and on-premises components, aligning with the deployment considerations discussed by Ayub et al. [22].

v. Investigation into the impact of hardware acceleration and specialised networking features on protocol performance, building on Ehlert's [16] work with hardware security modules. The exploration of QUIC-tunnelling as a potential alternative to traditional VPN protocols, extending Hettwer's [20] research to diverse application scenarios and environments

vi. Development of adaptive VPN frameworks that can dynamically switch between protocols based on network conditions and application requirements, optimising performance across variable environments.

vii. Virtualisation-protocol interaction studies: investigation into how different hypervisor architectures (VMware vSphere, Microsoft Hyper-V, KVM, and Xen) interact with VPN protocol processing mechanisms, with particular focus on packet handling efficiency and resource scheduling impacts on security protocol performance.

5.8. AI/ML-Enhanced VPN Performance Optimisation

5.8.1. Intelligent Network-Aware Protocol Selection

Building upon the variance-constrained local–global modelling approaches demonstrated in device-free localisation systems [51], future VPN implementations could incorporate machine learning models that continuously assess network conditions and dynamically select optimal protocols. Such systems would address our finding that "protocol performance is highly context-dependent" by creating adaptive frameworks capable of the following:

Real-time network condition classification: Machine learning models could classify network conditions (latency characteristics, bandwidth availability, and packet loss patterns) and predict optimal protocol selection based on historical performance data. The online learning paradigms successfully demonstrated in Wi-Fi-based device-free localisation [52] provide relevant frameworks for developing adaptive VPN systems that learn from network behaviour patterns.

Predictive protocol switching: AI models could anticipate network condition changes and proactively switch protocols before performance degradation occurs. This approach would be particularly valuable in mobile environments where network characteristics change frequently due to user mobility and varying infrastructure quality.

Environment-specific optimisation: Given our finding that "deployment context (Azure vs. VMware) showed greater influence than protocol choice," machine learning systems could learn environment-specific performance patterns and automatically configure protocol parameters for optimal performance in each deployment context.

5.8.2. AI-Driven Performance Optimisation

Dynamic parameter tuning: Machine learning algorithms could continuously optimise VPN protocol parameters based on real-time performance feedback and network condition analysis. Unlike our static configuration approach, AI-driven systems could adapt encryption levels, congestion control algorithms, and tunnel parameters to maintain optimal performance under varying conditions.

Anomaly detection and security enhancement: AI models could enhance VPN security by detecting unusual traffic patterns that may indicate security breaches, performance attacks, or network infrastructure issues. These systems would complement our Security Efficiency Index (SEI) metric by providing real-time security–performance trade-off optimisation.

Resource utilisation prediction: machine learning models could predict resource utilisation patterns and proactively adjust protocol configurations to prevent performance bottlenecks, particularly valuable given our finding of minimal but consistent memory efficiency advantages for WireGuard.

5.8.3. Multi-Dimensional Performance Prediction

Holistic performance modelling: AI systems could integrate our comprehensive performance metrics (throughput, latency, jitter, packet loss, and resource utilisation) into unified prediction models that forecast overall VPN performance across different scenarios. This would address our finding that performance advantages can "shift or disappear entirely" under challenging network conditions.

Application-specific optimisation: machine learning models could tailor VPN performance optimisation to specific application requirements, automatically prioritising latency minimisation for real-time applications or throughput maximisation for bulk data transfers, building upon our observation that "application requirements play a significant role" in protocol selection.

5.8.4. Integration with Edge–Fog–Cloud Architectures

Tier-aware protocol management: AI systems could leverage our edge–fog–cloud framework findings to implement tier-specific protocol optimisation strategies. Machine learning models could automatically configure protocols based on computational tier characteristics (ultra-low latency requirements at edge, moderate latency at fog, and variable high latency at cloud).

Cross-tier performance optimisation: advanced AI systems could optimise VPN performance across multiple architectural tiers simultaneously, potentially implementing different protocols at different tiers within the same communication path to maximise overall system performance.

5.9. Machine-Learning-Enhanced Research Methodologies

5.9.1. AI-Assisted Experimental Design

Adaptive experimental frameworks: future research could employ machine learning algorithms to optimise experimental design in real time, automatically adjusting test parameters based on preliminary results to maximise information gain and reduce experimental time requirements.

Intelligent baseline establishment: AI systems could automatically establish performance baselines across diverse environments and conditions, addressing our current limitation of static baseline measurements that may not capture full environmental variability.

5.9.2. Automated Performance Analysis

Pattern recognition in performance data: machine learning algorithms could identify subtle performance patterns that traditional statistical analysis might miss, potentially revealing new insights into protocol behaviour under specific conditions.

Predictive performance modelling: AI models could extrapolate our experimental results to predict performance in untested scenarios, expanding the applicability of research findings while identifying areas requiring additional experimental validation.

Integration recommendations
For immediate implementation (short-term):

1. Incorporate machine learning models for real-time network condition assessment and protocol recommendation.
2. Develop AI-enhanced monitoring systems that continuously track the performance metrics identified in your study.
3. Implement predictive analytics for proactive protocol switching based on network condition forecasts

For medium-term development:

1. Create adaptive VPN frameworks that automatically optimise protocol parameters using reinforcement learning.
2. Develop environment-specific AI models that learn optimal configurations for different deployment contexts (Azure vs. VMware).
3. Integrate anomaly detection systems that enhance both security and performance monitoring.

For long-term research:

1. Establish comprehensive AI-driven VPN ecosystems that integrate protocol selection, parameter optimisation, and security enhancement.
2. Develop quantum-ready AI systems that can adapt to post-quantum cryptographic requirements.
3. Create unified performance prediction models that incorporate all identified performance dimensions

5.10. Comparative Context: Performance of Alternative VPN Protocols

While our study focuses specifically on WireGuard and OpenVPN's performance across virtualised and cloud environments, it is important to contextualise these findings within the broader landscape of VPN protocol performance to provide readers with a comprehensive understanding of available alternatives.

5.10.1. IKEv2/IPSec Protocol Performance

IKEv2/IPSec demonstrates superior speed and stability compared to OpenVPN, with innovative auto-reconnect features that enhance user experience, and a throughput performance comparable to OpenVPN but with significantly faster connection establishment. In the context of our findings, IKEv2/IPSec would likely perform similarly to OpenVPN in our Azure environment tests, potentially offering advantages in scenarios requiring frequent reconnection due to network instability [33,42].

For mobile deployments, IKEv2/IPSec represents the most stable option, particularly when switching between different network interfaces (WiFi to cellular data) [33,36,42], making it potentially superior to both WireGuard and OpenVPN for mobile edge computing scenarios within our edge–fog–cloud framework. This characteristic addresses a limitation not evaluated in our VMware and Azure static environment testing.

5.10.2. SSTP Protocol Performance

SSTP demonstrates a good performance in bypassing restrictive firewalls but exhibits speed characteristics roughly equivalent to OpenVPN-UDP, lacking the superior throughput advantages we observed with WireGuard in VMware environments. While SSTP provides a fast, stable, and secure performance, its limited adoption by VPN providers restricts practical deployment options [33].

Given our finding that WireGuard's UDP-only architecture creates deployment limitations in restrictive network environments, SSTP's firewall traversal capabilities could provide advantages in corporate environments with strict outbound policies, potentially outperforming both WireGuard and OpenVPN in such constrained scenarios.

5.10.3. L2TP/IPSec Protocol Performance

Research indicates that L2TP/IPSec protocols exhibit high jitter and latency characteristics, making them unsuitable for streaming applications while remaining applicable for web, email, and file-sharing applications [36,42]. Based on our performance analysis, L2TP/IPSec would likely underperform both WireGuard and OpenVPN in our throughput and latency measurements, particularly in the low-jitter VMware environment where WireGuard demonstrated clear advantages.

The protocol's inherent latency characteristics would likely compound the high-latency conditions we artificially introduced in our testing, potentially resulting in even more severe performance degradation than observed with either WireGuard or OpenVPN.

5.10.4. Legacy Protocol Considerations

PPTP, despite its historical significance, should no longer be considered for deployment, with even Microsoft advising against its use due to security vulnerabilities [36]. This protocol's exclusion from modern comparative analysis reflects the evolution of VPN technology toward the more secure and performant protocols evaluated in our study.

5.10.5. Implications for Protocol Selection

The performance characteristics of alternative protocols reinforce our primary finding that protocol selection must be tailored to specific deployment environments and application requirements. While our study demonstrates WireGuard's advantages in VMware environments and OpenVPN's resilience in high-latency Azure scenarios, the broader protocol landscape suggests the following.

For mobile and edge deployments: IKEv2/IPSec may offer superior stability and reconnection capabilities compared to both protocols evaluated in our study, particularly relevant for edge computing scenarios in our edge–fog–cloud framework.

For restrictive network environments: SSTP's firewall traversal capabilities could provide connectivity advantages over WireGuard's UDP-only limitations, though potentially at the performance cost we documented for OpenVPN in optimal network conditions.

For legacy system integration: L2TP/IPSec, while exhibiting inferior performance characteristics, may remain necessary for compatibility with older network infrastructure that cannot support more modern protocols.

These considerations complement our finding that "optimal VPN protocol selection depends critically on the specific deployment tier rather than universal protocol characteristics" by extending this principle across the complete spectrum of available VPN technologies. Future research incorporating these alternative protocols within our dual-environment comparative framework would provide more comprehensive guidance for enterprise VPN deployment strategies.

The performance advantages we documented for WireGuard in controlled VMware environments and OpenVPN's resilience in challenging Azure conditions represent important contributions to the comparative literature, but organisations should consider the complete protocol ecosystem when making deployment decisions that balance performance, security, compatibility, and network constraint requirements.

In summary, our empirical results demonstrate that protocol performance is highly context dependent, with WireGuard offering advantages in VMware environments while both protocols perform similarly in Azure under normal conditions. Recent research

extends these findings by highlighting additional considerations including quantum resistance, fingerprinting resistance, hardware acceleration, and integration with emerging distributed technologies. Protocol selection should therefore be guided by specific application requirements, deployment environments, expected network conditions, and emerging security considerations rather than general claims about protocol superiority.

6. Conclusions

Our comprehensive performance evaluation of WireGuard and OpenVPN across Azure and VMware environments reveals that protocol performance advantages are highly context dependent, varying significantly by deployment environment, network conditions, and performance metrics. These findings challenge oversimplified assertions about protocol superiority and provide critical guidance for optimal protocol selection in diverse deployment scenarios.

In VMware baseline environments, WireGuard demonstrated a substantially superior TCP throughput (210.64 Mbps compared to OpenVPN's 110.34 Mbps, a 91% improvement) and UDP throughput (285.28 Mbps versus 154.88 Mbps, an 84% advantage), as shown in our rigorous statistical analysis in Sections 4.1.1 and 4.1.2. These significant performance differentials suggest that WireGuard's streamlined architecture provides particular advantages in virtualised local environments, aligning with findings by Mackey et al. [14] on performance comparisons of WireGuard and OpenVPN.

However, in Azure cloud environments, both protocols demonstrated a comparable baseline performance, with nearly identical throughput metrics (WireGuard: 281.76 Mbps and OpenVPN: 290.77 Mbps) and latency characteristics (approximately 85 ms). This performance equivalence in cloud environments contradicts some earlier studies (such as Wallin et al. [53] and Mackey et al. [14] that reported substantial WireGuard advantages across all deployment scenarios.

Our innovative analysis of performance under challenging network conditions revealed unexpected patterns that advance the current understanding of protocol behaviour. In Azure high-latency scenarios, OpenVPN maintained a better TCP throughput (approximately 120 Mbps versus WireGuard's 60 Mbps), while in VMware high-latency tests, both protocols performed similarly poorly. This demonstrates that protocol advantages can reverse or disappear entirely under adverse network conditions, a critical consideration for deployments in environments with variable connection quality.

The resource utilisation analysis, detailed in Sections 4.1.3 and 4.1.4, challenges common assertions about WireGuard's efficiency advantages. In Azure environments, both protocols showed nearly identical CPU utilisation (approximately 32.5%), while in VMware, both exhibited similarly low CPU usage. WireGuard consistently demonstrated marginally better memory efficiency across environments (33.87% vs. OpenVPN's 36.42% in VMware; 14.69% vs. 14.80% in Azure), though these differences were statistically significant but practically minimal.

The network reliability metrics presented in Section 4.1.4 revealed that WireGuard offers substantial advantages in VMware environments, with 50% lower jitter (approximately 30 ms versus OpenVPN's 60 ms) and dramatically reduced packet loss (12.35% versus OpenVPN's 47.01%) under baseline conditions. These advantages translate to a tangibly better quality of service for real-time applications in virtualised environments, representing an important finding for organisations prioritising communication quality.

Our novel Security Efficiency Index analysis (Section 4.1.8) provides a new framework for evaluating the practical efficiency of security implementations. This analysis demonstrated that efficiency advantages are environment specific and metric dependent. WireGuard showed marginally better efficiency across all metrics in Azure environments,

while in VMware, the results were mixed: OpenVPN exhibited a better throughput efficiency despite a lower absolute performance, but WireGuard offered substantial advantages for latency-sensitive applications.

These findings have significant practical implications for organisational VPN deployments:

i. Environment-specific protocol selection: Organisations should select VPN protocols based on their specific deployment environment. For VMware environments under normal conditions, WireGuard offers clear advantages in throughput, jitter, and packet loss. For Azure cloud deployments, either protocol may be suitable depending on specific requirements and expected network conditions.
ii. Application-dependent optimisation: Protocol selection should consider the specific application requirements. For real-time applications in virtualised environments, WireGuard's lower jitter and packet loss provide substantial benefits. For throughput-intensive applications, the advantages vary by environment.
iii. Network condition considerations: Organisations operating in environments with variable network quality should recognise that protocol advantages can shift significantly under challenging conditions. Performance testing under realistic network scenarios is essential for optimal protocol selection.
iv. Decision framework: organisations should implement a structured decision framework for protocol selection that incorporates deployment environment, application requirements, expected network conditions, and specific performance priorities.

Our research advances the field by providing empirical evidence that challenges several common assertions about VPN protocol performance and offers guidance based on specific deployment requirements. The significant protocol–environment interactions observed in our study highlight the importance of context-specific testing rather than relying on generalised performance claims.

Future Research Direction

Building upon our empirical framework and findings, our research agenda aims to develop comprehensive, context-aware guidance for VPN protocol selection across diverse deployment scenarios. This multi-phase approach will systematically address the complexity of modern networking environments while extending our established methodology to increasingly complex real-world scenarios.

Future research should expand upon these findings through a structured three-phase approach:

Phase 1: Configuration and Scale Extensions

i. Investigating protocol performance with varied encryption settings and security configurations.
ii. Evaluating larger-scale deployments to better understand scalability characteristics.

Phase 2: Complex Deployment Scenarios

i. Examining performance in hybrid and multi-cloud environments that combine different infrastructure types.
ii. Analysing performance with diverse application workloads to provide application-specific guidance.

Phase 3: Advanced Optimisations

i. Assessing the impact of hardware acceleration and specialised networking features on protocol performance.

This systematic research progression will culminate in a comprehensive decision-support framework that considers deployment environment, application requirements, security configurations, and infrastructure capabilities. Through continued empirical research, organisations will gain increasingly precise guidance for optimising VPN deployments across their specific infrastructure environments, advancing the field toward precise, context-aware protocol selection guidance.

In summary, our research demonstrates that VPN protocol performance is highly context dependent, with advantages varying by environment, network conditions, and specific performance metrics. This critical understanding challenges simplistic assertions about protocol superiority and provides organisations with empirically grounded guidance for optimising their secure networking implementations.

Author Contributions: Conceptualisation, J.A. and R.R.S.; methodology, J.A. and R.R.S.; software, J.A. and R.R.S.; validation, J.A.; writing—original draft preparation, J.A. and R.R.S.; writing—review and editing, R.R.S., H.L., and A.P.; supervision, R.R.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The tools used in this work and the results obtained have been made available at https://doi.org/10.5281/zenodo.15760416 (accessed on 1 January 2025).

Acknowledgments: The authors would like to acknowledge various colleagues for their input and critical comments. The authors would also like to thank the reviewers for their critical and insightful comments, which helped us immensely in not only improving our work but also uncovering and presenting important details of our work.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Basa, R. Demystifying Cloud Computing and Virtualization: A Technical Overview. *Int. J. Res. Comput. Appl. Inf. Technol.* **2024**, *7*, 125–135.
2. Jimmy, F. Cyber Security Vulnerabilities and Remediation through Cloud Security Tools. *J. Artif. Intell. Gen. Sci.* **2024**, *2*, 129–171.
3. Hans, M.; Fuhrmann, T.; Reindl, A.; Niemetz, M. FMS-BERICHTE SOMMERSEMESTER 2022. In Proceedings of the FMS-OTHR, Regensburg, Germany, 17 September 2022; pp. 36–40. [CrossRef]
4. Abbas, H.; Emmanuel, N.; Amjad, M.F.; Yaqoob, T.; Atiquzzaman, M.; Iqbal, Z.; Shafqat, N.; Shahid, W.B.; Tanveer, A.; Ashfaq, U. Security Assessment and Evaluation of VPNs: A Comprehensive Survey. *ACM Comput. Surv.* **2023**, *55*, 1–47. [CrossRef]
5. Tian, X. A Survey on VPN Technologies: Concepts, Implementations, And Anti-Detection Strategies. *Int. J. Eng. Dev. Res.* **2025**, *13*, 85–96.
6. Akinsanya, M.O.; Ekechi, C.C.; Okeke, C.D. Virtual Private Networks (VPN): A Conceptual Review of Security Protocols and Their Application in Modern Networks. *Eng. Sci. Technol. J.* **2024**, *5*, 1452–1472. [CrossRef]
7. Bansode, R.; Girdhar, A. Common Vulnerabilities Exposed in VPN—A Survey. *J. Phys. Conf. Ser.* **2021**, *1714*, 012045. [CrossRef]
8. Korhonen, V. Future after OpenVPN and IPSec. Master's Thesis, Tampere University, Tampere, Finland, 2019.
9. Donenfeld, J.A. WireGuard: Next Generation Kernel Network Tunnel. In Proceedings of the NDSS, San Diego, CA, USA, 26 February–1 March 2017; pp. 1–12.
10. Master, A.; Garman, C. *A WireGuard Exploration*; CERIAS Technical Reports; Purdue University: West Lafayette, IN, USA, 2021. [CrossRef]
11. Iqbal, M.; Riadi, I. Analysis of Security Virtual Private Network (VPN) Using openVPN. *Int. J. Cyber-Secur. Digit. Forensics* **2019**, *8*, 58–65. [CrossRef]
12. Rodgers, C. *Virtual Private Networks: Strong Security at What Cost?* Report; University of Canterbury. Computer Science and Software Engineering: Christchurch, New Zealand, 2001.
13. Abdulazeez, A.; Salim, B.; Zeebaree, D.; Doghramachi, D. Comparison of VPN Protocols at Network Layer Focusing on Wire Guard Protocol. *Int. J. Interact. Mob. Technol.* **2020**, *14*, 18. [CrossRef]
14. Mackey, S.; Mihov, I.; Nosenko, A.; Vega, F.; Cheng, Y. A Performance Comparison of WireGuard and OpenVPN. In Proceedings of the Tenth ACM Conference on Data and Application Security and Privacy, New Orleans, LA, USA, 16–18 March 2020; pp. 162–164.

15. Shim, H.; Kang, B.; Im, H.; Jeon, D.; Kim, S. qTrustNet Virtual Private Network (VPN): Enhancing Security in the Quantum Era. *IEEE Access* **2025**, *13*, 17807–17819. [CrossRef]
16. Ehlert, E. OpenVPN TLS-Crypt-V2 Key Wrapping with Hardware Security Modules. *Stud. Inform. Ski.* **2025**, *S-19*. [CrossRef]
17. Dekker, E.; Spaans, P. *Performance Comparison of VPN Implementations WireGuard, strongSwan, and OpenVPN in a 1 Gbit/s Environment*; University of Amsterdam: Amsterdam, The Netherlands, 2020.
18. Johansson, V. A Comparison of OpenVPN and WireGuard on Android. Bachelor's Thesis, Umea University, Umea, Sweden, 2024.
19. Samanta, A.; Sharma, M.; Locke, W.; Williamson, S. Cloud-Enhanced Battery Management System Architecture for Real-Time Data Visualization, Decision Making, and Long-Term Storage. *IEEE J. Emerg. Sel. Top. Ind. Electron.* **2025**, 1–12. [CrossRef]
20. Hettwer, L. Evaluation of QUIC-Tunneling. Ph.D. Thesis, Hochschule für Angewandte Wissenschaften Hamburg, Hamburg, Germany, 2025.
21. Anyam, J.; Singh, R. Repository for Empirical Analysis of WireGuard vs. OpenVPN in Cloud and Virtualised Environments. *Zenodo*. Available online: https://zenodo.org/records/15760416 (accessed on 28 June 2025).
22. Ayub, N.; Bakhet, S.; Arshad, M.J.; Saleem, M.U.; Anam, R.; Fuzail, M.Z. An Enhanced Machine Learning and Blockchain-Based Framework for Secure and Decentralized Artificial Intelligence Applications in 6g Networks Using Artificial Neural Networks (Anns). *Spectr. Eng. Sci.* **2025**, *3*, 348–364.
23. Chua, C.H.; Ng, S.C. Open-Source VPN Software: Performance Comparison for Remote Access. In Proceedings of the ICISS 2022: 2022 the 5th International Conference on Information Science and Systems, Beijing, China, 26–28 August 2022.
24. Yedla, B.K. Performance Evaluation of VPN Solutions in Multi-Region Kubernetes Cluster. Master's Thesis, Blekinge Institute of Technology, Karlskrona, Sweden, 2023.
25. Xue, D.; Ramesh, R.; Jain, A.; Kallitsis, M.; Halderman, J.A.; Crandall, J.R.; Ensafi, R. OpenVPN Is Open to VPN Fingerprinting. *Commun. ACM* **2025**, *68*, 79–87. [CrossRef]
26. Jumakhan, H.; Mirzaeinia, A. Wireguard: An Efficient Solution for Securing IoT Device Connectivity. *arXiv* **2024**, arXiv:2402.02093.
27. Pudelko, M.; Emmerich, P.; Gallenmüller, S.; Carle, G. Performance Analysis of VPN Gateways. In Proceedings of the 2020 IFIP Networking Conference (Networking), Paris, France, 22–25 June 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 325–333.
28. Sabbagh, M.; Anbarje, A. Evaluation of WireGuard and OpenVPN VPN Solutions. Bachelor's Thesis, Linnaeus University, Växjö, Sweden, 2020.
29. Ostroukh, A.V.; Pronin, C.B.; Podberezkin, A.A.; Podberezkina, J.V.; Volkov, A.M. Enhancing Corporate Network Security and Performance: A Comprehensive Evaluation of WireGuard as a Next-Generation VPN Solution. In Proceedings of the 2024 Systems of Signal Synchronization, Generating and Processing in Telecommunications (SYNCHROINFO), Vyborg, Russia, 1–3 July 2024; pp. 1–5.
30. Akter, H.; Jahan, S.; Saha, S.; Faisal, R.H.; Islam, S. Evaluating Performances of VPN Tunneling Protocols Based on Application Service Requirements. In Proceedings of the Third International Conference on Trends in Computational and Cognitive Engineering: TCCE 2021, Online, 21–22 October 2021; Springer: Berlin/Heidelberg, Germany, 2022; pp. 433–444.
31. Budiyanto, S.; Gunawan, D. Comparative Analysis of VPN Protocols at Layer 2 Focusing on Voice over Internet Protocol. *IEEE Access* **2023**, *11*, 60853–60865. [CrossRef]
32. Wahanani, H.E.; Idhom, M.; Mandyartha, E.P. Analysis of Streaming Video on VPN Networks between OpenVPN and L2TP/IPSec. In Proceedings of the 2021 IEEE 7th Information Technology International Seminar (ITIS), Surabaya, Indonesia, 6–8 October 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 1–5.
33. Lawas, J.B.R.; Vivero, A.C.; Sharma, A. Network Performance Evaluation of VPN Protocols (SSTP and IKEv2). In Proceedings of the 2016 Thirteenth International Conference on Wireless and Optical Communications Networks (WOCN), Hyderabad, India, 21–23 July 2016; IEEE: Piscataway, NJ, USA, 2016; pp. 1–5.
34. Rajamohan, P. An Overview of Remote Access VPNs: Architecture and Efficient Installation. *IPASJ Int. J. Inf. Technol.* **2014**, *2*, 11.
35. Liu, Z. Application and Security Analysis of Virtual Private Network (VPN) in Network Communication. *Acad. J. Comput. Inf. Sci.* **2023**, *6*, 52–59. [CrossRef]
36. Arora, P.; Vemuganti, P.R.; Allani, P. *Comparison of VPN Protocols–IPSec, PPTP, and L2TP*; Department of Electrical and Computer Engineering George Mason University: Fairfax, VA, USA, 2011; Volume 646.
37. Hilley, D. *Cloud Computing: A Taxonomy of Platform and Infrastructure-Level Offerings*; Georgia Institute of Technology: Atlanta, GA, USA, 2009; pp. 44–45.
38. Motahari-Nezhad, H.R.; Stephenson, B.; Singhal, S. Outsourcing Business to Cloud Computing Services: Opportunities and Challenges. *IEEE Internet Comput.* **2009**, *10*, 1–17.
39. Javed, M.A.; Ahmad, M.; Ahmed, J.; Rizwan, S.M.; Tariq, A. An Enhanced Machine Learning Based Data Privacy and Security Mitigation Technique: An Intelligent Federated Learning (FL) Model for Intrusion Detection and Classification System for Cyber-Physical Systems in Internet of Things (IoTs). *Spectr. Eng. Sci.* **2025**, *3*, 377–401.
40. Gentile, A.F.; Macrì, D.; De Rango, F.; Tropea, M.; Greco, E. A VPN Performances Analysis of Constrained Hardware Open Source Infrastructure Deploy in IoT Environment. *Future Internet* **2022**, *14*, 264. [CrossRef]

41. Gatti, V.R.; Shetty, P.; Ravi Prakash, B.; Rama Moorthy, H. Balancing Performance and Protection: A Study of Speed-Security Trade-Offs in Video Security. In Proceedings of the 2024 International Conference on Recent Advances in Science and Engineering Technology (ICRASET), Mandya, India, 21–22 November 2024; pp. 1–6.
42. Ghanem, K.; Ugwuanyi, S.; Hansawangkit, J.; McPherson, R.; Khan, R.; Irvine, J. Security vs Bandwidth: Performance Analysis Between IPsec and OpenVPN in Smart Grid. In Proceedings of the 2022 International Symposium on Networks, Computers and Communications (ISNCC), Shenzhen, China, 19–22 July 2022; pp. 1–5.
43. Jucha, G.T.; Yeboah-Ofori, A. Evaluation of Security and Performance Impact of Cryptographic and Hashing Algorithms in Site-to-Site Virtual Private Networks. In Proceedings of the 2024 International Conference on Electrical and Computer Engineering Researches (ICECER), Gaborone, Botswana, 4–6 December 2024; pp. 1–6.
44. Vydyanathan, N.; Catalyurek, U.V.; Kurc, T.M.; Sadayappan, P.; Saltz, J.H. Toward Optimizing Latency under Throughput Constraints for Application Workflows on Clusters. In Proceedings of the European Conference on Parallel Processing, Rennes, France, 28– 31 August 2007; Springer: Berlin/Heidelberg, Germany, 2007; pp. 173–183.
45. Hakak, S.; Anwar, F.; Latif, S.A.; Gilkar, G.; Alam, M. Impact of Packet Size and Node Mobility Pause Time on Average End to End Delay and Jitter in MANET's. In Proceedings of the 2014 International Conference on Computer and Communication Engineering, Da Nang, Vietnam, 30 July–1 August 2014; IEEE: Piscataway, NJ, USA, 2014; pp. 56–59.
46. Tufte, E.R.; Graves-Morris, P.R. *The Visual Display of Quantitative Information*; Graphics Press: Cheshire, CT, USA, 1983; Volume 2.
47. Cohen, J. *Statistical Power Analysis for the Behavioral Sciences*, 2nd ed.; Behavioral Sciences, Economics, Finance, Business & Industry, Social Sciences; Routledge: New York, NY, USA, 1988; ISBN 978-0-203-77158-7.
48. Blenk, A.; Basta, A.; Kellerer, W.; Schmid, S. On the Impact of the Network Hypervisor on Virtual Network Performance. In Proceedings of the 2019 IFIP Networking Conference (IFIP Networking), Warsaw, Poland, 20–22 May 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 1–9.
49. Narayan, S.; Brooking, K.; de Vere, S. Network Performance Analysis of Vpn Protocols: An Empirical Comparison on Different Operating Systems. In Proceedings of the 2009 International Conference on Networks Security, Wireless Communications and Trusted Computing, Wuhan, China, 25–26 April 2009; IEEE: Piscataway, NJ, USA, 2009; Volume 1, pp. 645–648.
50. Jyothi, K.K.; Reddy, B.I. Study on Virtual Private Network (VPN), VPN's Protocols and Security. *Int. J. Sci. Res. Comput. Sci. Eng. Inf. Technol.* **2018**, *3*, 919–932.
51. Zhang, J.; Li, Y.; Li, Q.; Xiao, W. Variance-Constrained Local–Global Modeling for Device-Free Localization Under Uncertainties. *IEEE Trans. Ind. Inform.* **2024**, *20*, 5229–5240. [CrossRef]
52. Zhang, J.; Xue, J.; Li, Y.; Cotton, S.L. Leveraging Online Learning for Domain-Adaptation in Wi-Fi-Based Device-Free Localization. *IEEE Trans. Mob. Comput.* **2025**, *24*, 7773–7787. [CrossRef]
53. Wallin, F.; Putrus, M. Analyzing the Impact of Cloud Infrastructure on VPN Performance: A Comparison of Microsoft Azure and Amazon Web Services. Bachelor's Thesis, Mälardalen University, Västerås, Sweden, 2024.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.