Comparative Study of Modern VPN Solutions: Impact of Cloudflare, ZeroTier, and WireGuard on Network and Server Performance

Pratchaya Jaisudthi*, Pachara Threerapat Sridee, Natthakran Phungkoed, Kanyaphak Srisuk and Vasupon Phueaknumpol
Computer Engineering, Rambhai Barni Rajabhat University, Tachang, Mueang, Chanthaburi, 22000, Thailand
***Corresponding Author E-mail:** pratchaya.j@rbru.ac.th
Received: Feb 18, 2025; **Revised:** Apr 17, 2025; **Accepted:** Apr 22, 2025

Abstract

This research investigates of the performance of three popular VPN solutions namely Cloudflare, ZeroTier and WireGuard, by measuring their effect on network performance and server resource usage across multiple metrics such as file upload/download speeds, round-trip time (RTT), web latency, and server CPU usage. The aim is to find the best solution for certain workloads by benchmarking these solutions in a controlled manner. The results of these experiments showed large performance differences. The results were consistent for all tests: WireGuard provided the fastest upload and download speed (19 seconds and 52 seconds, for 1000 MB files, respectively), the lowest web latency (50 milliseconds for 1000 connections), and the most efficient CPU utilization (24% at 1000 connections). For small size of packets (less than 700 bytes), Cloudflare provided competitive RTTs around 10 milliseconds and balanced performance for light workloads. However, it was not scalable indicated by web latency about 200 milliseconds and CPU utilization higher than 32% in high-concurrency scenarios. Conversely with lower workloads, ZeroTier struggled with download of heavy file sizes and lots of connections such as downloading with 1000 MB in size took 84 seconds and up to 62% of CPU utilization. WireGuard emerges as the best-suited high-performance solution for scalable applications. Cloudflare and ZeroTier offer trade-offs helpful to particular use cases, providing perspective on which VPN solution to choose depending on workload requirements and resource constraints.

Keywords: VPN Solutions, Cloudflare, ZeroTier, WireGuard, Network Performance, Server CPU Utilization

1. Introduction

The exponential growth of distributed systems, cloud-based services, and remote working environments, the need for Virtual Private Network (VPN) solutions that deliver strong security, high performance, and large-scale delivery has exploded. As a result, VPNs have especially become a pillar in secure and efficient communication. For example, employees of an organization use VPNs to securely access internal networks, servers, and cloud-based applications. Additionally, businesses employ VPNs to protect data and maintain privacy during communication between branches over public networks. These practical applications illustrate the advantages of modern solution technologies, such as Cloudflare [1], ZeroTier [2], and WireGuard [3], which are gradually becoming the most popular VPN technologies.

They all have individual designs and architectural strengths that aim to solve different networking issues. Cloudflare is explicitly built to speed up your web traffic and lower your latency through their worldwide web network to better the experience of the end consumer, which is visible through their global outreach. ZeroTier uses peer-to-peer network virtualization to provide secure access between diverse environments while enabling high throughput. At the same time, WireGuard, which has a tiny code base and modern cryptographic building blocks, seeks to be the gold standard for VPN performance and security.

Despite the rapid acceleration in their adoption, a rigorous performance analysis of these underexplored solutions in realistic environments remains a challenge. Key performance metrics, namely upload and download speeds, round-trip time (RTT), web latency, and server CPU utilization, are crucial to determining their impact on both network efficiency and managing server resources. For enterprises and individuals, the decision of what is the relevant VPN solution to implement is an understanding of the trade-offs here in a range of network conditions.

The Key Performance Indicator (KPI) comparison among Cloudflare, ZeroTier, and WireGuard are analyzed together in this paper with a special focus on its performance under real-world network conditions. This research addresses these challenges by plotting and analyzing five key performance metrics and then translating the information into actionable insights regarding their performance characteristics. Thus, through this comparative evaluation, we intend to help guide the practitioners and researchers in selecting the best-suited VPN solution that provides the required security, performance, and computational overhead for a given application scenario.

These modern VPN solutions are in widespread use today, yet their comparative impact on the performance of networks, servers, and applications

with diverse workloads has been under-researched. Prior work primarily emphasizes either security or isolated performance metrics thus offering only a limited view of scalability, efficiency, and resource consumption. This gap also makes it difficult for practitioners and researchers to determine which VPN solution is most appropriate under different scenarios.

This study aims to fill this gap by thoroughly evaluating these solutions in order to determine the most appropriate VPN solution for different situations. The goal is to assess performance metrics including upload/download speeds, RTT, web latency, and server CPU utilization, and to offer actionable insight for solution selection according to workload requirements.

This study makes three notable contributions as follows:

1) Detailed and complete testing of Cloudflare, ZeroTier, and WireGuard VPN solutions in realistic conditions.
2) In-depth discussion of KPIs to expose trade-offs and benefits.
3) Recommendations for the selection of VPN solutions, concerning both scalability, resource utilization, and workload requirements.

The rest of this paper is organized as follows: section 2 presents related work and gives background about Cloudflare, ZeroTier, and WireGuard. Next, section 3 describes the experimental setup and methodology. Then the results are presented in section 4, and the analysis and implications are discussed in section 5. Lastly, the insights and future research directions are concluded in section 6.

2. Related Works

Cloudflare is widely used to enhance to website performance [4] and security [5–7]. Estri et al. [8] showed that the approach enables bandwidth optimization for maximizing speeds up to 95% and significant reduction in server load. Similarly, Dykstra et al. have also integrated with CloudFlare CDN, which has, as reported by [9], led to 2.2 times of increasing in throughput, under conditions of high demand. Moreover, Tarahomi et al. [10] pointed out the importance of Cloudflare to DNS resilience, but it improves latency and connection stability. This makes these types of studies critical to assessing the extent to which Cloudflare is effective in optimizing upload/download speed and RTT for VPNs. This, similar to our work, allows to evaluate Cloudflare Zero Trust effects in real world on VPN throughput and latency. For modern networks, Cloudflare Zero Trust offers an effective solution for securing networks, particularly in cloud-based and remote work environments [11]. Therefore, this research is interested in studying various performance aspects of VPNs.

ZeroTier has been proven to perform well in dynamic and secure environments. Hrișcan et al. [12] proved its scalability in IoT systems, offering centralized management and end-to-end encryption. Similarly, Mo et al. [13] showed that ZeroTier is responsible for private peer-to-peer file transfers, providing enhanced privacy through greater control of data without third-party servers. Kornel et al. [14] and Burke et al. [15] illustrated ZeroTier works very well in scenarios where devices have very dynamic demands regarding latencies and throughput, like Unmanned Aerial vehicle (UAV) telemetry. These studies provide benchmarks for understanding how ZeroTier performs across different scenarios, matching our focus on resource efficiency and RTT analysis.

WireGuard is well known for its simplicity and performance. The research of Mackey et al. [16] showed that it significantly outperforms OpenVPN in terms of throughput on multi-core systems, while Donenfeld [17] pointed out its minimal cryptographic design leads to secure connections with minimal latency. Goethals et al. [18] tested WireGuard's scalability under edge computing, demonstrating its resilience in resource-constrained environments. This aligns with our goal to investigate throughput, latency, and server utilization for a variety of workloads.

Some studies compare VPN solutions in real-world environments. Nemčik et al. [19] mainly researched the reliability and security features of a VPN in a private network and compared them to WireGuard in terms of simplicity and performance. Sio et al. [20] demonstrated ZeroTier's low-latency performance in real-time applications, while Kornel et al. [14] highlighted it role in the seamless connectivity of 4G-based telemetry systems. These works consider the trade-offs between scalability, security, and resource efficiency and inform our comparative evaluation of the three protocols through a practical setting.

Previously, other works have focused on studying individual VPN solutions or analyzing their security features, but none provide a full comparative analysis of these relevant performance metrics across several configurations and usage scenarios. Building on prior research, this paper provides an evaluation of industrial VPN solutions (in this case, Cloudflare, ZeroTier, and WireGuard) in controlled and reproducible conditions using a consistent methodology to benchmark against relevant key performance indicators (KPIs; upload/download speed, RTT, web latency and server CPU utilization) for workload-aware decisions on the appropriate VPN to deploy in practice.

3. Experimental Setup and Methodology

In this section, the experimental framework and methodology employed to assess the performance evaluation of three VPN solutions: Cloudflare VPN, ZeroTier, and WireGuard are presented for focusing on their impact on network and server performance. These solutions were chosen because they are freeware technologies and most popular VPNs. The key performance metrics like upload speed, download speed, RTT, web latency, and server CPU utilization were analyzed under controlled conditions to ensure reproducibility and comparability.

The consistent solutions in a common configuration and controlled testing environment taught the

experiment reproducibility and enabled the comparison of different approaches. This rigorous method also fills a gap in existing research that often does not adhere to a standardized testing infrastructure.

In order to obtain stable and reproducible results, the experiments were carried out in a controlled laboratory environment with the following specifications

3.1 Server and Client Setup

An Ubuntu Server 22.04 LTS server with 1-core CPU, 2 GB of RAM, and 1 Gbps network interface is a host machine. We used Proxmox VE (Virtualized Environment) as our virtualization platform to instantiate the VPN servers. For VPN Instances, Cloudflare VPN, ZeroTier, and WireGuard were deployed on separate virtual machine instances to avoid contention of resources between them. Meanwhile, A high-performance client machine with an 8-Core CPU and 16 GB of RAM is connected to VPN servers via private WiFi Network (LAN). Additionally, WireGuard utilizes WireGuard Server with 1 CPU cores, 2 GB of RAM, and a public IP address. The client and server were about 64 km apart, roughly the distance between Chanthaburi City and Trat City, Thailand, mimicking a typical medium-range VPN setup for enterprise or cloud access, while keeping the network controlled for consistent results.

3.2 Network Configuration

The VPN solutions were all configured using the default settings recommended by their respective developers to mimic real-world deployment scenarios.

3.2.1 Cloudflare

Cloudflare, an example of the above-mentioned approach is shown in **Figure 1**, uses a tunnel to allow a client device on an external network to communicate gracefully with a web server on an internal network. This configuration utilizes Cloudflare's edge network, adding security, performance, and scalability.

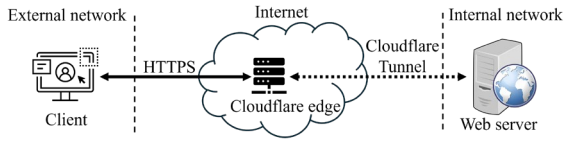

Figure 1 The Communication flow of Cloudflare solution

We deployed Cloudflare Zero Trust to strengthen secure access, providing rigorous identity verification and consistent policy enforcement across all connections. Here is a breakdown of the components

1) Client (External Network) means the end-user connecting to the web server from the internet. The client makes HTTPS requests to communicate with the web server securely.
2) Cloudflare Edge (Internet) is a gateway of a globally distributed network of servers that act as an intermediary between the client and the internal network. And it takes care of encryption, caching, load balancing, DDoS mitigation, and more.
3) Cloudflare Tunnel is a secured, encrypted tunnel established between Cloudflare edge and the internal network, which connects the internal services (say web server) without exposing those to the public internet.
4) Web Server (Internal Network) is the target resource that provide web services; they are protected within the internal network and are accessed via the Cloudflare Tunnel.

A detailed explanation of the process is given below

1) Client Request is when the client makes an HTTPS request in order to gain access to the web server resources. The request is then encrypted for transport security.
2) The HTTPS request is first routed through the Cloudflare edge closest to the client. This edge server decrypts and analyzes the request. Check whether the request follows configured security policies like firewalls, access control lists, or WAF (Web Application Firewall) rules. Also, Caching and optimization with mechanisms is applied.
3) After validating the request, the Cloudflare edge server routes it through a pre-established secure Cloudflare Tunnel. This step establishes a tunnel from the Cloudflare edge to the internal network that hosts the web server.
4) The Cloudflare Tunnel forwards the request to the web server inside the protected internal network The tunnel guarantees that nothing in the internal network can be reached from the public internet.
5) The web server handles the request from the client and responds accordingly (for instance, serving a webpage or providing requested data).
6) The response from the web server is sent back via the Cloudflare Tunnel to Cloudflare Edge. If required, the edge server then re-applies the caching or optimization rules. Encrypts response to send back to client.
7) The final response is encrypted with a key that the client had shared initially using the same procedure it had decrypted its own requests, and the encrypted response is sent to the client over HTTPS as the cycle of request-response continues.

3.2.2 ZeroTier

The Operational working architecture of a ZeroTier Tunnel is illustrated in **Figure 2**. It is used to securely access a web server located on an internal network by a client residing on an external network using software-defined networking (SDN). ZeroTier provides a virtual network overlay for easy and secure communication.

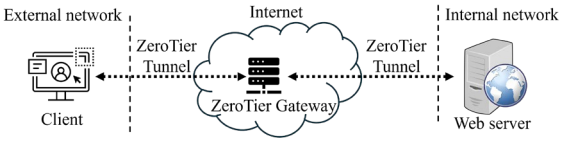

Figure 2 The Communication flow of ZeroTier solution

There are some other components apart from client and web server which are discussed below.

1) A central node within the ZeroTier virtual network, called ZeroTier Gateway, serves as a gateway and allows users to connect different devices across the internet. It allows a secure path for the traffic without exposing sensitive information.
2) A secure encrypted point-to-point tunnel, called the ZeroTier Tunnel, is managed by ZeroTier to transmit data among the client, the ZeroTier gateway, and the web server

Here is a step-by-step account of how this works.

1) The client connects to the ZeroTier virtual network after validating itself with pre-provisioned credentials to the network (e.g., network ID). Once the client is authenticated, an encrypted tunnel with the ZeroTier gateway is created over the internet.
2) The ZeroTier gateway serves as a bridge to traverse computing traffic between the external client and the internal computing network. In ZeroTier, all traffic between the client and the internal network is encrypted and encapsulated within the tunnel.
3) The encrypted traffic from the ZeroTier gateway will be routed into another secure ZeroTier Tunnel that leads to the web server in the internal network.
4) The web server then handles the request from the client and creates a response (e.g., sending web content or data). The response is then encrypted and sent back through the ZeroTier Tunnel to the gateway.
5) The ZeroTier gateway then routes the server's encrypted response to the client through the established ZeroTier Tunnel over the internet. The response is then decrypted by the client to finish the request-response cycle.

3.2.3 WireGuard

In the diagram as shown in **Figure 3**, we can see the architectural working of a WireGuard VPN where a client is connecting to a web server in an internal network through a secure channel. WireGuard is fast, simple, lightweight, and utilizes modern cryptography algorithms which makes it suitable for low-latency and high-performance networking environments.

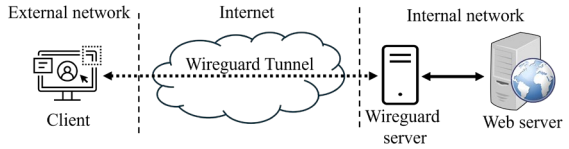

Figure 3 The Communication flow of Wireguard solution

There are other components detailed below in addition to the client and web server.

1) A WireGuard Tunnel is a secure and encrypted communication channel created by the WireGuard VPN solution. This tunnel encrypts all information passed back and forth between the client and server which no one else can intercept the communication.
2) A WireGuard Server is located at the edge of the internal network and responsible for encrypted traffic. It also authenticates clients, decrypted traffic, and forwards it to the web server on the internal network. WireGuard Server can be reachable from outside networks via a public IP address.

Here is how the process works in detail.

1) The client device uses the WireGuard configuration (consisting, for example, of public/private keys, endpoint information) to connect to the WireGuard server and establish a secure connection over the internet.
2) The WireGuard server verifies what the client's public key is then they establish an encrypted tunnel. The handshake is low overhead and ensures that the tunnel is secure and authenticated.
3) The encrypted traffic of client is forwarded to the WireGuard server via the WireGuard tunnel. The traffic becomes decrypted at the server and is then sent to the web server hosted on the internal network.
4) The web server processes the request (serving a webpage or returning the requested data) and sends the response back to the WireGuard server.
5) The response from the web server is encrypted by the WireGuard server and sent back to the client over the WireGuard tunnel. The client processes the response (by decrypting it) to complete the request-response cycle.

3.3 Metrics Collection

By measuring multiple KPIs such as file transfer speeds, RTT, web latency, and server CPU utilization, this study provides a comprehensive performance overview, enabling practical decision-making for solution selection. The architecture for collecting metrics is illustrated in **Figure 4**, based on a client-server model that exploits a set of tools/solutions to assess the upload speed, download speed, RTT, and web latency.

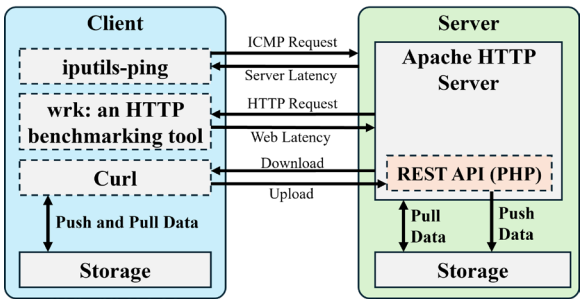

Figure 4 The architecture for collecting metrics

All metrics are collected as a combination of network requests and benchmarking tools to provide accurate and reproducible measured metrics.

3.3.1 Upload Speed

This metric shows the performance of a network in transporting data from the client to the server. The upload speed is calculated by sending data from the client side to the server side using a REST API with PHP. The client uses "curl" [21], a command line tool for sending data to the server. Upload time is recorded and the speed in MB/s is determined as the file size divided by the time taken for the upload. It is a push process, from the client's storage, data goes to the server's REST API, then written into the server's storage.

3.3.2 Download Speed

This metric measures the performance of the network in getting the data to the client. Download speed is measured by pulling data from the server's storage using "curl". The client makes a request through the REST API at the server, retrieving and sending the data back to the client. Like upload speed, download speed is calculated as a ratio of data size to the duration of time taken to download the data. We then have the server that retrieves this data from its storage, sends it over through REST API, and delivers it to the client to measure. Both upload and download tests are to perform using the standard MTU size, usually 1500 bytes for Ethernet, to mimic real-world network conditions effectively.

3.3.3 Round-Trip Time (RTT)

RTT represents the time taken for a packet to go on a whole roundtrip to the network and back, so it is useful for representing the latency that the network adds at the layers. "iputils-ping" [22] is used to capture RTT by sending Internet Control Message Protocol (ICMP) requests from the client to the server. The time it took for the ICMP request to reach the server and for the response to come back to the client is measured. ICMP requests are initiated from the client and replies are received from the server. Round-trip time is the time calculated from the time the request was generated, and the response was received.

3.3.4 Web Latency

This is a means of assessing the server's responsiveness and the duration of completing operations on the web, aside from application-level latency. Web latency is measured using "wrk" [23], an HTTP load testing tool used to simulate HTTP requests to the server. It records the time it takes for the client to send an HTTP request and receive a response from the server. HTTP requests are sent from the client to the server's Apache HTTP Server, which handles requests and responds to the client.

3.3.5 Server CPU Utilization

The server CPU measurement is one of the most important aspects of determining how much resources the server uses under various workloads. The "top" command [24] is also used to monitor the CPU in real-time, to understand how the server is processing operations.

3.4 Testing Methodology

To comprehensively evaluate the performance of Cloudflare, ZeroTier, and WireGuard VPN solutions, detailed testing was performed using a systematic test methodology to derive key performance metrics: upload/download speeds, RTT, web latency, and server CPU utilization. The methodology provides a consistent, reproducible, and comparable process across all solutions and diverse workload conditions.

3.4.1 Baseline Measurements

The tests were repeated a total of 30 times to obtain average values and mitigate the effects of anomalies. To show variation, 95% confidence intervals (CIs) were included in the figures, computed from standard deviation of the sample means, assuming a normal distribution.

3.4.2 VPN Performance Evaluation

All VPN solutions were tested individually with the same network conditions and traffic. To maintain symmetry, the upload and download tests employed the peak 1000 MB (1 GB) file transfer.

3.4.3 Server Utilization

Simulated user traffic for real-world usage was applied on each VPN server instance with multiple concurrent connections from 20 to 1000. CPU usage, measured on the web server, was logged every second on the tests.

3.4.4 Data Analysis

This aggregated data is visualized in five performance graphs: upload/ download speed, RTT, web latency, and server CPU utilization. A confident interval has been added to the figures to indicate variation.

4. Results

We investigate the performance of three different VPN solutions, namely Cloudflare, ZeroTier, and WireGuard by running a set of experiments in order to assess their efficiency, scalability, and resource utilization across different setups. These experiments measure four important metrics: file upload and download speeds, RTT, web latency, and server CPU utilization. Through a methodical assessment of how each solution responds to escalating workloads and a variety of situations, this section offers an in-depth examination of their respective strengths, weaknesses, and overall suitability for particular applications.

4.1 File Download Performance Analysis Across Varied File Sizes

In the experiment, we measured the file download performance of the three VPN solutions: Cloudflare, ZeroTier, and WireGuard, with a file size of 10 MB to 1000 MB. The results, as shown in **Figure 5** indicate substantial differences in the solutions' efficiency and scalability.

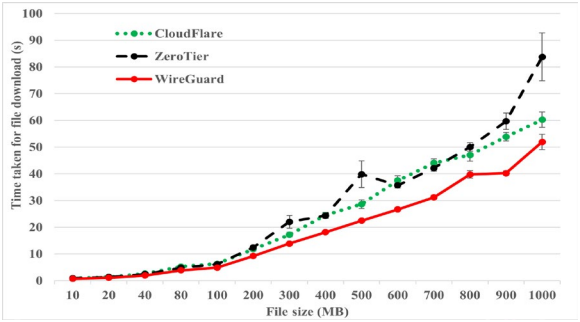

Figure 5 download times across varying file sizes with 95% confident interval.

WireGuard had the fastest download times, with a 1000 MB file downloading in about 52 seconds. Its performance was scaleable with file size, with much small variability indicated by narrow confidence intervals. This shows the efficiency and consistency of WireGuard, even for large file sizes.

Cloudflare did moderately well; download times slowly grew with file size. Cloudflare had a download time of 60 seconds for a 1000 MB file, this number came in slightly slower than WireGuard.

ZeroTier demonstrated the most significant variations in download times, with 1000 MB of files downloaded in about 84 seconds. It has value similar to Cloudflare in smaller file sizes, but when it came to files larger than 500 MB, the download times were considerably worse. Because larger file sizes show wider confidence intervals, the performance does not seem consistent, which suggests an overhead or inefficiencies when handling a large volume of data.

4.2 File Upload Performance Analysis

The experiment measured how long it took to upload files 10 MB, 100 MB, 300 MB, and 1000 MB via three VPN solutions — Cloudflare, ZeroTier, and WireGuard. As shown in **Figure 6**, there are significant differences between these solutions from the perspective of upload efficiency.

The quickest upload times across the board belonged to WireGuard, which completed 1000 MB upload in around 19 seconds. The solution showed linear and monotonous growth in upload time with respect to file size, along with relatively low variance in upload time shown by narrow confidence intervals. This performance demonstrates the efficiency and reliability of WireGuard, especially for high throughput use cases.

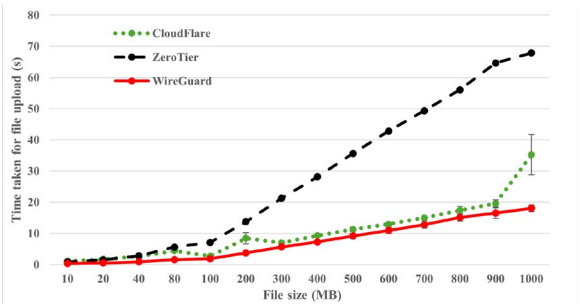

Figure 6 upload times across varying file sizes with 95% confident interval.

Cloudflare did fairly well, with upload times for smaller files holding steady but a gradual rise in upload times for files under 800 MB. Cloudflare took 36 seconds to upload 1000 MB, consistent but slower than the performance we saw with WireGuard.

ZeroTier was less able to scale when it came to file sizes. Although it competed well with Cloudflare in regard to files up to 200 MB in size, its upload times significantly increased for larger files, topping out at around 70 seconds for files of 1000 MB.

4.3 Round Trip Time (RTT) Analysis Across Varied Packet Sizes

RTT of three VPN solutions (Cloudflare, ZeroTier, and WireGuard) were measured with packet sizes from 28 bytes to 1300 bytes. The resulting figure, **Figure 7**, illustrates the major difference in network slatency handling between the various solutions over a range of packet sizes.

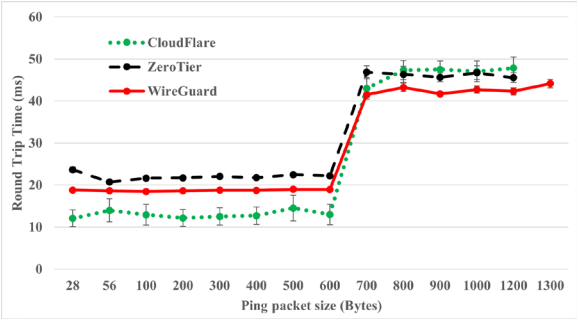

Figure 7 Round trip times across varying Ping packet sizes

In fact, for smaller packet sizes (less than 700 bytes), Cloudflare had the lowest RTT with fairly constant values around 10–15 milliseconds. This indicates its efficiency in processing smaller packets. Subsequently, the RTTs reported by WireGuard and ZeroTier were slightly higher (approximately 19 milliseconds), however the results were stable, showing low variability (narrow confidence intervals).

All solutions showed increased RTT for larger packet sizes (700 bytes and above). WireGuard stabilized at approximately 43 milliseconds, which surprisingly did not vary much with the increase in packet size. But the RTT for Cloudflare and ZeroTier surged far above 700 bytes, hitting about 47 milliseconds for the larger packets. The broad confidence intervals for larger packet sizes indicate variance in Cloudflare's performance under heavier workloads.

4.4 Web Latency Analysis Under Increasing Connections

This experiment evaluates the web latency of three different VPN solutions (Cloudflare, ZeroTier, and WireGuard) by measuring the time to complete web requests per number of concurrent connections, ranging from 20 to 1000. The results in **Figure 8** exhibit significant differences in latency performance as the number of connections increases.

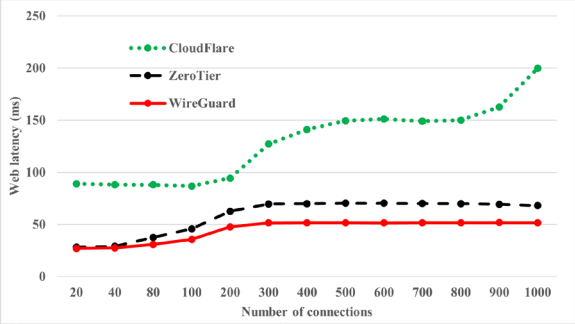

Figure 8 Web latency across varying number of connections

WireGuard had the lowest web latency for every number of connections. For 20 connections it showed a latency of about 30 milliseconds which is quite stable, even after increasing the number of connections to 1000, showing how well it scales in terms of concurrency.

Cloudflare initially performed well with a latency of around 90 milliseconds for 20 connections. As the number of connections increased, however, its latency started growing gradually. Once it hit 1000 connections, Cloudflare's latency was approximately 200 milliseconds, the highest of the three solutions.

ZeroTier displayed intermediate performance, with web latency values falling between 30 milliseconds at 20 connections to around 70 milliseconds at 1000 connections. Even though it had better increasing connections than Cloudflare, its latency was slightly higher and more volatile than WireGuard.

4.5 Server CPU Utilization Analysis Under Increasing Connections

The experiment evaluates the server CPU utilization of the three VPN solutions for Cloudflare, ZeroTier, and WireGuard, under an increasing number of concurrent connections, between 20 concurrent connections up to 1000. This experiment uses the Linux "top" command to consistently measure the server CPU utilization across all VPN solutions. The results are logged every second until the CPU utilization values reach a saturation point for each test. This value is measured using a system-wide approach, which reflects the actual resource consumption, not just the VPN process. These results are shown in **Figure 9** and demonstrate important differences in how the solutions use a server resource at different loads.

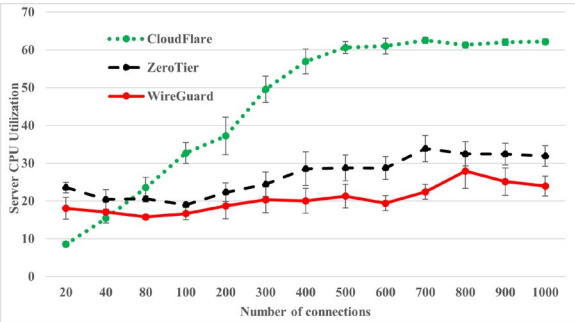

Figure 9 CPU utilization across varying number of connections

Overall, WireGuard shows the least CPU load in the entire experiment. CPU usage is around 19% at 20 connections, increasing slightly to about 24% at 1000 connections.

ZeroTier showed moderate CPU load, and it began at about 24% for 20 connections and 32% for 1000 connections. Although its resource usage was greater than that of WireGuard, it was fairly constant regardless of numbers of connection.

Cloudflare started with about 10% CPU utilization (at 20 connections). But with a growing number of connections, its CPU usage significantly increased, reaching over 60% with 1000 connections. The sharp increase indicates potential scalability problems in processing more concurrency.

5. Analysis and Discussion

Based on the capabilities of VPN solutions, we also conducted a comparative study of them and their impact on the performance of our network servers. We performed five experiments to measure and analyze file download time, file upload time, RTT, web latency, and server CPU utilization for different conditions. Below, we summarize and discuss the findings from these experiments.

5.1 File Download Performance Across Varied File Sizes

As can consistently be seen from the file download performance analysis, WireGuard is the most efficient solution. Because of its linear scaling with file size, it is well-suited for high-throughput tasks and the kind of workloads that need stable performance. Its low overhead and optimized traffic management means it can maintain superior download speeds, even for the largest files tested.

Cloudflare provides a balanced performance profile, yielding steady results at each file size. Although it is slower than WireGuard, its reliability and consistent performance make it a good fit for applications where medium workloads and predictability matter. But its slower downloads for larger files suggest some limitations in scalability relative to WireGuard.

On the other hand, ZeroTier has some issues with large file downloads. Its sharply rising download times for files over 700 MB in size indicate poor resource management or solution overhead. And wider confidence intervals of larger files capture variability, making it less likely to rely on tasks requiring predictable performance.

Overall, WireGuard is the best-performing solution when it comes to file downloads, due to its balance of speed, scalability, and general reliability. Another variation is Cloudflare, which offers a middle ground with more consistent but slower speeds, and ZeroTier, which has issues with scalability and stability on high-volume downloads.

5.2 File Upload Performance Across Varied File Sizes

As we can see, comparing file upload performance between the three VPN solutions, WireGuard comes

out on top as the clearly superior option. By being able to reach the lowest upload times with very low variability, it is also the most efficient solution when transferring such large files in the fastest time possible. Its low overhead, paired with the linear scalability of WireGuard, makes it especially well-suited for high-performance use cases where reliability and speed are paramount.

Cloudflare, as it is slower than WireGuard, provides average stability with performance. The same goes for keeping it consistent, especially for the smaller file sizes, making it a good option for moderate workloads. However, the slower response time for larger files reveals a balance between dependability and effectiveness, making it less suitable for large volumes of uploading.

ZeroTier had the worst performance for upload speed and scalability. The dramatic increase in the upload times for larger files between files of a similar size that should have taken a similar amount of time to upload suggests a lack of ability to handle a larger workload. Although ZeroTier can work well for smaller uploads or distributed systems, it is ultimately not as appealing for resource-intensive uploads due to its limited scalability and consistency.

In summary, WireGuard is the most effective and reliable solution for file uploads, while Cloudflare provides a stable middle-ground solution for moderate workloads. ZeroTier, although versatile for specific applications, is not well-suited for large-scale data transfers due to its scalability challenges and inconsistent performance.

5.3 Round-Trip Time (RTT) Analysis

The RTT analysis points to differences between how the three VPN solutions deal with latency for different packet sizes. WireGuard shows the most predictable performance in terms of RTT and latency across packet sizes with confidence intervals that are tightly packed, indicating little variation. This calculates to RTTs of 19–42 milliseconds and around across the packet size range, indicating that it is able to successfully manage latency, and is great for applications requiring a low-latency connection.

Cloudflare is much better at small packets, achieving the lowest RTT for packet sizes below 700 bytes. It therefore makes sense in use cases where small packet transmission is common, like real-time communication or a lightweight data transfer. Performance of Cloudflare degrades with larger packets, however, with RTTs increasing substantially. This makes it inapplicable for scenarios with big data payloads.

Moving on to larger packets, we can see that ZeroTier matches the performance of WireGuard, with the RTT settling at 40–50 milliseconds. It is ultimately middle-of-the-road, performing reasonably well across the full range of packet sizes; however, it is not nearly as optimized as Cloudflare for small packets nor is it as scalable as WireGuard for larger packets. Wider confidence intervals also imply some degree of performance variability, which can negatively affect latency-sensitive applications.

In summary, WireGuard offers the lowest latency along with very good performance across packet sizes, making it the most balanced and scalable solution. Cloudflare delivers the performance at a smaller packet size but has scaling issues and ZeroTier gives reasonable performance but is non-union and they are not effective and balanced enough to be directly communicative to help with extreme latency-sensitive scenarios.

5.4 Web Latency Under Increasing Connections

Analyzing web latency, WireGuard has shown to be the most scalable and efficient solution when it comes to growing the number of outflows. It is able to sustain a steady latency of about 50 milliseconds no matter how many connections are employed, which is a testament to its ability to service high-traffic volumes. The lack of variance across connection counts reinforces WireGuard as the ideal selection for latency-sensitive applications that need consistent performance in the face of concurrent user loads.

Cloudflare showed significant limitations in scalability. While it performed reasonably well for smaller connection counts, its latency increased sharply beyond 300 connections, reaching over 200 milliseconds at 1000 connections. This significant surge reveals inefficiencies in handling increased concurrency, rendering it less desirable for high-user traffic use cases. The increased variability at the higher connection counts indicates difficulty in sustaining predictable performance under stress. Though we did not dig into possible reasons like load balancing or encryption overhead, as our study focused on comparing performance metrics, future research could investigate these behaviors of Cloudflare with heavy connection loads.

ZeroTier had a middle-of-the-road performance, being slightly higher latency than WireGuard but much more consistent than Cloudflare. It managed more connections pretty well but had slightly varying performance and a higher latency than WireGuard, making it less suitable for latency-critical applications. However, ZeroTier may still be suitable for scenarios where moderate scalability and acceptable latency are required.

Ultimately it turned out to be WireGuard, which has low and stable performance among all numbers of connections which is what we need to deal with web latency. Cloudflare has restrictions on concurrency that limit its ability to run some workloads and is only useful for lighter workloads. ZeroTier offers satisfactory performance for moderate workloads but lags behind the efficiency and stability of WireGuard for high traffic.

5.5 Server CPU Utilization Under Increasing Connections

The server CPU utilization analysis shows how WireGuard is the least resource-hungry solution across different connection scenarios. Unlike other solutions,

which continue to face problems at high connection counts, it is able to keep CPU usage low and predictable even at the limit of 1000 connections or more. The minimal variability further reinforces its reliability in managing server resources under heavy loads.

ZeroTier was CPU-bound at all connection counts but struggled to maintain consistent performance with increasing connection counts. With a CPU usage of 20% to 35%, it is more suitable for moderate workload use cases where resource consumption is not the top priority. However, its slightly higher variability than WireGuard does introduce a measure of unpredictability under peak conditions.

Cloudflare had the most consumption in terms of CPU utilization, consuming more resources as the connection count increased. Its CPU usage even reached 60% for about 1000 connections, showing significant inefficiency for high concurrency. The wider confidence intervals at higher loads further highlight variability, making Cloudflare less suitable for applications requiring scalability and consistent resource management.

In summary, WireGuard is the most efficient and most scalable server CPU usage implementation, dropping down to memory and dodging computation if it hits the limits. ZeroTier provides a kind of "acceptable middle ground" solution, but is slightly less consistent than WireGuard, and will result in an increase in CPU usage. Under heavy load, Cloudflare becomes CPU-bound and consumes a lot of resources, making it less suitable for high-concurrency applications.

5.6 Summary of Performance

Table 1 summarizes the main evaluation metrics from the experiments. It compares the three VPN solutions along various dimensions, such as download time, upload time, RTT, web latency, and server CPU utilization.

Table 1 Performance Comparison of VPN Technologies

Metric	WireGuard	Cloudflare	ZeroTier
Download Time (1000 MB)	52 seconds	60 seconds	84 seconds
Upload Time (1000 MB)	19 seconds	36 seconds	70 seconds
RTT (packet size of 28-600 bytes)	19 ms	10 ms	22 ms
RTT (packet size of 700-1300 bytes)	43 ms	48 ms	46 ms
Web Latency (1000 connections)	50 ms	200 ms	54 ms
CPU Utilization (1000 connections)	24%	32%	62%

5.7 Choosing the Best Solution

By reviewing the analysis and the attached **Table 1**, WireGuard is the big winner. It provides the best download and upload times, the lowest RTT and web latency, and the least CPU usage on the server. It also makes this the preferred choice for the majority of applications, especially ones with high traffic, large file transfers, or need to be scalable.

However, Cloudflare offers a reasonable trade-off for scenarios where smaller packet transmission or moderate workloads dominate. It merges low RTT for small packets with a stable performance in file transfers but fails on high load and large concurrency. Thus, ZeroTier is somewhere in the middle that serves well for distributed applications, but less than optimally at higher loads or larger files.

Ultimately, I would conclude that WireGuard is the best overall VPN solution and that Cloudflare and ZeroTier are potentially acceptable trade-offs depending on the specific application and workload requirements. These findings contribute to a deeper understanding of VPN solution behavior, guiding practitioners in selecting the right solution for their operational needs, whether optimizing for speed, scalability, or resource constraints.

6. Conclusion

This study evaluates the performance of three modern VPN solutions—Cloudflare, ZeroTier, and WireGuard—analyzing their effects on network performance and server resource utilization using a diverse repertoire of measures, such as file upload/download speeds, RTT, web latency, and server CPU utilization for different workloads. The findings highlight significant differences in the efficiency, scalability, and consistency of the solutions, identifying their suitability for different use cases.

Across all metrics, WireGuard outperformed both Cloudflare and ZeroTier. It came first in the fastest upload and download speeds (35 seconds and 52 seconds, respectively, for files of 1000 MB), the lowest web latency (50 milliseconds for 1000 connections), and the most effective CPU utilization (25% with 1000 connections). The linear scaling and low variability make it the best option in high-performance scenarios that need both scalability and reliability.

The performance of Cloudflare was relatively moderate; however, it showed good performance for RTT around 10–15 milliseconds for small packet sizes and also had relatively steady performance for moderate workloads. The results revealed some scalability limitations with respect to larger file sizes and connection counts where latencies above 200 milliseconds and CPU usages above 60%. ZeroTier had more mixed results, proving adequate with lighter workloads but struggling to scale consistently on heavier workloads, with CPU utilization hitting 35%.

We recommend in future work that additional VPN solutions, diverse network conditions, and security resilience be explored as well. Exploring energy efficiency in resource-constrained environments and utilizing machine learning for real-time optimization could further improve the performance and responsiveness of VPNs. These directions will support informed decisions for diverse networking scenarios.

7. References

[1] Cloudflare. "Cloudflare." cloudflare.com. https://www.cloudflare.com/. (accessed Jun. 25, 2024).

[2] ZeroTier. "Zerotier." zerotier.com. https://www.zerotier.com (accessed Jun. 25, 2024).

[3] J. A. Donenfeld, "Wireguard: Fast, modern, secure VPN tunnel," wireguard.com. https://www.wireguard.com (accessed Jun. 25, 2024).

[4] E. Bajrami, F. Idrizi, and A. RUSHITI, "Enhancing Website Speed and Security with Cloudflare CDN: A Case Study on WordPress Websites," *JNSM Journal of Natural Sciences and Mathematics of UT*, vol. 9, no. 17–18, pp. 294–304, 2024, doi: 10.62792/ut.jnsm.v9.i17-18.p2824.

[5] J. John, O. Obikwelu, G. Akawuku and C. L. Onyagu, "Detecting and preventing of DDoS Attack in Cloud Computing Environment Based on Hybrid Technique (Cloudflare and WAF)," *Newport International Journal of Engineering and Physical Sciences (NIJEP)*, vol. 3, no. 3, pp. 28–40, 2023, doi: 10.59298/NIJEP/2023/10.4.1100.

[6] K. Bhargavan, I. Boureanu, P. -A. Fouque, C. Onete and B. Richard, "Content delivery over TLS: A cryptographic analysis of keyless SSL," in *2017 IEEE European Symposium on Security and Privacy (EuroS&P)*, Paris, France, 2017, pp. 1–16, doi: 10.1109/EuroSP.2017.52.

[7] J. Zirngibl, P. Sattler and G. Carle, "A first look at SVCB and HTTPS DNS resource records in the wild," in *2023 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*, Delft, Netherlands, 2023, pp. 470-474, doi: 10.1109/EuroSPW59978.2023.00058.

[8] E. J. Dewi, U. Rusydi, and R. Imam, "Implementation of Cloudflare hosting for speeds and protection on the website," M.S. thesis, Dept. of Technology, Universitas Ahmad Dahlan, Yogyakarta, Indonesia, 2019. [Online]. Available: https://core.ac.uk/download/pdf/231724704.pdf.

[9] D. Dykstra, B. Bockelman, J. Blomer and L. Field, "The open high throughput computing content delivery network," presented at EPJ Web of Conferences, Sofia, Bulgaria, Jul. 9–13, 2018, Paper 04023.

[10] S. Tarahomi, R. Holz, and A. Sperotto, "Quantifying security risks in cloud infrastructures: A data-driven approach," in *2023 IEEE 9th International Conference on Network Softwarization (NetSoft)*, Madrid, Spain, 2023, pp. 346–349, doi: 10.1109/NetSoft57336.2023.10175501.

[11] M. Hossein, "Enhancing the security and privacy of home storage servers in private clouds using zero trust principles," M.S. thesis, Centria University of Applied Sciences, Kokkola, Finland, 2024. [Online]. Available: https://www.theseus.fi/ handle/10024/876384.

[12] D. -F. Hrițcan, A. Graur and D. G. Balan, "Securing IoT environments using ZeroTier and OPNsense," in *2024 23rd RoEduNet Conference: Networking in Education and Research (RoEduNet)*, Bucharest, Romania, 2024, pp. 1–4, doi: 10.1109/RoEduNet64292.2024.10722755.

[13] Q. Mo, S. Duan, W. Huang, Y. Lai and Y. Xu, "Research and design of a personal data management system," in *2022 IEEE 5th International Conference on Information Systems and Computer Aided Education (ICISCAE)*, Dalian, China, 2022, pp. 574–579, doi: 10.1109/ICISCAE55891.2022.9927545.

[14] F. A. P. Kornel, F. A. Fernanda, R. Rahardi, W. D. Lukito, P. Herdian, R. Virginio and E. Mulyana, "A low cost, compact, and easy to set up 4G telemetry module for UAV application," in *2021 15th International Conference on Telecommunication Systems, Services, and Applications (TSSA)*, Bali, Indonesia, 2021, pp. 1–5, doi: 10.1109/TSSA52866.2021.9768230.

[15] P. J. Burke, "A 4G-connected micro-rover with infinite range," *IEEE Journal on Miniaturization for Air and Space Systems*, vol. 1, no. 3, pp. 154–162, 2020, doi: 10.1109/JMASS.2020.3018660.

[16] S. Mackey, I. Mihov, A. Nosenko, F. Vega and Y. Cheng, "A performance comparison of WireGuard and OpenVPN," in *Proc. 10th ACM Conf. Data and Application Security and Privacy*, New York, NY, USA, 2020, pp. 162–164, doi: 10.1145/3374664.3379532.

[17] J. A. Donenfeld, "WireGuard: Next generation kernel network tunnel," in *Proc. 2017 Network and Distributed System Security Symposium*, San Diego, CA, USA, 2017, pp. 1–12, doi: 10.14722/ndss.2017.23160.

[18] T. Goethals, D. Kerkhove, B. Volckaert and F. D. Turck, "Scalability evaluation of VPN technologies for secure container networking," in *2019 15th International Conference on Network and Service Management (CNSM)*, Halifax, NS, Canada, 2019, pp. 1–7, doi: 10.23919/CNSM46954.2019.9012673.

[19] J. Nemčik, P. Kănuch and I. Kotuliak, "Content distribution in private networks," in *2022 International Symposium ELMAR*, Zadar, Croatia, 2022, pp. 67–70, doi: 10.1109/ELMAR55880.2022.9899785.

[20] Y. X. Sio, R. Abdulla, N. A. Ramasenderan, and M. E. Rana, "Harnessing the Power of Drones for Precision Agriculture: A Real-Time Monitoring System Utilizing Sensor Data and RF Communication," in *2023 IEEE 21st Student Conference on Research and Development (SCOReD)*, Kuala Lumpur, Malaysia, 2023, pp. 301–306, doi: 10.1109/SCOReD60679.2023.10563268.

[21] Curl. "curl: Command line tool and library for transferring data with URLs." curl.se. https://curl.se (accessed: Sep. 17, 2024).

[22] iputils. "iputils/iputils." github.com. https://github.com/iputils/iputils (accessed: Sep. 18, 2024).

[23] W. Glozer. "wrk: A modern HTTP benchmarking tool." github.com. https://github.com/wg/wrk (accessed Sep. 18, 2024).

[24] M. Kerrisk. "top(1) Linux manual page." man7.org. https://man7.org/linux/man-pages/ man1/top.1. html (accessed: Sep. 18, 2024)