Mälardalen University
School of Innovation Design and Engineering
Västerås, Sweden

Thesis for the Degree of Bachelor of Science in Engineering –
Computer Network Engineering, 15 credits

ANALYZING THE IMPACT OF CLOUD INFRASTRUCTURE ON VPN PERFORMANCE: A COMPARISON OF MICROSOFT AZURE AND AMAZON WEB SERVICES

Marwin Putrus
mps21002@student.mdu.se

Filip Wallin
fhn21008@student.mdu.se

Examiner: Elisabeth Uhlemann
Mälardalens University, Västerås, Sweden

Supervisor: Maryam Vahabi
Mälardalens University, Västerås, Sweden

10th June 2024

Abstract

In recent years, the interest and utilization of VPNs and cloud computing have surged, which has led to the development of Cloud VPNs. Cloud VPNs are often used to give employers access to company resources over a secured and trusted network, but individuals can also use them. Cloud computing offers numerous advantages, including cost efficiency, scalability, security, and reliability. Companies and individuals widely use it to streamline operations, reduce expenditures, and leverage technologies without the need to maintain on-premise infrastructure. Cloud-based solutions impose specific requirements on technologies designed to operate in the cloud, which include security, efficient resource management, and high network availability and performance. In this thesis, we will analyze and evaluate the network performance impact that Amazon Web Services (AWS) and Microsoft Azure, two of the market-leading cloud platforms, have on the VPN services OpenVPN and WireGuard while considering associated operational costs and user complexity.

The network performance impact is evaluated through measuring the throughput, latency, jitter, and packet loss. We have performed experiments divided into three setups, each simulating different traffic patterns, and VPN usage scenarios. Throughout these experiments, we observed and documented the user complexity related to setup, installation, and configuration processes. Our findings indicate that Azure has the best overall network throughput across all setups, fewer retransmissions, and fewer packet losses. Conversely, AWS exhibits lower latency and jitter. Additionally, our assessment of operational costs and user complexity reveals that Azure offers lower associated costs but a higher user complexity. Furthermore, our experiments identified that WireGuard, when paired with Azure, offers the best VPN solution.

Contents

List of Figures

List of Tables

1. Introduction

A Virtual Private Network, or VPN, is a technique used to establish an encrypted virtual network connection between two or more devices, facilitating remote access for telecommuters and providing increased security and privacy for personal usage [1]. In recent years, the utilization of VPNs has surged, driven by the growing security concerns and the growing need for remote access [2]. In 2022, the global market for VPNs was estimated at 44.6 billion USD and is projected to reach 137.7 billion USD by 2030 [3].

Likewise, cloud computing has emerged as a transformative technology, offering numerous applications and services [4]. Among these advancements is the integration of VPN into cloud environments, which has led to the development of a new technology known as cloud VPN [5]. Cloud VPNs leverage the scalability, and flexibility of cloud computing infrastructure to provide secure and reliable VPN services to users accessing cloud resources. This enables employees to access the corporate network and resources while maintaining high security and availability. In this context, the network performance of cloud VPNs becomes essential.

The objective of this thesis is to compare and analyze how the infrastructure provided by major public cloud providers, particularly Microsoft Azure and Amazon Web Services, impacts the network performance of the VPN services OpenVPN and WireGuard. We will focus on the most recent versions of these cloud platforms available. In our assessment, we consider the following network performance metrics: network throughput, jitter, latency, and packet loss. Throughput, latency, jitter, and packet loss are key metrics that impact the user experience and overall effectiveness of VPN servers in cloud environments. High throughput ensures efficient data transfer, low latency decreases communication delays, minimal jitter ensures smooth communication and low packet loss maintains data integrity. These metrics will be measured through experiments involving client-to-client communication, client-to-VPN server communication, and when utilizing the VPN as a gateway for internet access. Furthermore, we will assess user complexity during the setup, installation, and configuration process of the cloud platforms. Operational costs imposed by the different cloud providers will also be evaluated. Through this analysis, we aim to provide insights into the comparative performance, usability, and cost-effectiveness of OpenVPN and WireGuard across Azure and Amazon Web Services. These insights can help organizations and individuals decide which public cloud provider to select for their VPN infrastructure.

2. Background

In this section, we will introduce the essential technologies and concepts needed for this thesis.

2.1 Virtual Private Network

A Virtual Private Network is a technique that is designed to establish an encrypted and protected network connection between two or more network devices over an untrusted channel [6]. VPN provides privacy by routing data traffic through a VPN host, which can be configured at remote servers. VPN has grown in popularity for personal use and within enterprises because of its ability to encrypt data traffic, enhancing user privacy and security.

The key to achieving a secure communication tunnel is using the right methods to encrypt data traffic [7]. With the help of encryption, a VPN can counter many types of attack vectors such as data-fishing, interceptions or eavesdropping, and Man-in-the-Middle (MitM) attacks. The encryption process involves transferring keys between the two connected devices. The keys encrypt and decrypt data traffic upon an established connection. Two encryption methods are used by VPNs, Symmetric Encryption and Asymmetric Encryption [8]. Symmetric encryption uses a single key for both encryption and decryption. Asymmetric encryption involves two types of keys, a public, and a private key. The public key is transferred between the devices trying to establish a connection and is used for data encryption. The private key is used to decrypt data, that the public key has encrypted.

2.2 VPN Services

In the two following sections, we introduce the VPN services used in this thesis.

2.2.1 OpenVPN

OpenVPN is a software-based VPN solution [9]. It provides enterprises and private users with a VPN solution that facilitates secure remote access to network resources and applications over the Internet. OpenVPN is designed to enable businesses to implement OpenVPN easily and quickly and supports a wide range of configurations such as Remote Access, Site-to-Site VPNs, and Wi-Fi Security.

The key features of OpenVPN are secure connectivity, ease of use, cross-platform support, and scalability. OpenVPN utilizes the OpenVPN Protocol, known for its security and reliability, and encrypts data traffic between devices and networks over public internet connections.

2.2.2 WireGuard

Wireguard is a VPN protocol released in 2020 [10]. While very similar to other VPN protocols such as IPsec and OpenVPN, the remarkable difference is that the software operates in kernel mode in the operating system. While running in the kernel, the software works closer to the hardware, giving it a huge advantage in access times to necessary resources. This leads to greater performance, and higher security and minimizes the risks of data leaks. The software was first implemented in Linux but is supported across multiple operating systems such as Windows, MacOS, and Android.

2.3 Cloud Computing

Cloud computing is a service that provides computing resources, such as servers, databases, storage, and software over the internet [11]. This solution offers faster innovation, flexible resources, and economies of scale. It eliminates the need for physical hardware and managing your data centers, making it very flexible to work from anywhere.

The idea of cloud computing came across in 1960 in conjunction with network-based computing known as “time-sharing” at that time [12]. The term later evolved into cloud computing and emerged in the 21st century simultaneously with ARPANET, which laid the groundwork for the internet we have today. There are benefits to cloud computing, such as cost efficiency, scalability, performance, security, and reliability [13]. Physical hardware is no longer needed, leading to less financial expenses of purchasing such hardware or software. Electricity costs for power consumption and cooling are significantly reduced, due to the cloud provider maintaining the facility and hardware. Since cloud computing offers a wide range of computing power and storage, scalability is not an issue and companies can scale depending on their need. Cloud providers also provide a set of policies, and technologies, and control the overall security that protects data and applications from potential threats. Reliability is not an issue either, since the cloud provider provides data backup, and data can be mirrored to a different location.

Many businesses are adopting cloud computing as it aligns with their strategic goals and provides substantial advantages. The convenience, cost savings, and simplified management offered by cloud computing make it an attractive choice for infrastructure development.

2.3.1 Amazon Web Services

Amazon Web Services is a cloud-computing platform developed and released by Amazon in the early 2000s [14]. Today, AWS is the leading cloud-computing platform with over 3 million companies that rely on their services and holds a 32% market share [15]. In 2023, reports show the generated revenue for the first quarter was around 23 billion USD and the total revenue for the whole year was around 90 billion USD. AWS provides services, such as compute, storage, databases, and network services. Their most used service is the Elastic Compute Cloud (EC2) service, which offers virtual machines that can be used for different purposes.

2.3.2 Microsoft Azure

Azure is a cloud-computing platform developed by Microsoft which was first presented in 2008 and launched in 2010 [16]. It was originally named "Project Red Dog", but later renamed to Microsoft Azure in 2014. Azure is the second leading cloud platform after AWS and holds a 23% market share [15]. Furthermore, Azure provides services similar to AWS and focuses on computing services.

2.4 iPerf

One commonly used tool for network performance analysis is iPerf [17], a command-line tool that measures network performance by generating TCP and UDP data streams. It allows users to test the throughput, latency, and packet loss between two endpoints over an IP network.

iPerf has two major versions, iPerf2 and iPerf3 [17]. We employed iPerf3 as one of the tools for conducting network performance evaluations in our research. iPerf3, the newer version, offers enhanced features and improvements over its predecessor, iPerf2, and is a completely new tool without backward compatibility. iPerf3 can be used to measure the maximum available throughput, packet loss, and jitter, supporting Internet Protocol version 4 (IPv4), Internet Protocol version 6 (IPv6), User Datagram Protocol (UDP), and Transmission Control Protocol (TCP).

2.5 Speedtest

Speedtest, created by Ookla, is a tool that measures the speed and performance of the internet connection [18]. Speedtest can be used in a web browser and supports an app for iOS, Android, macOS, and Windows devices, and a command-line tool for developers. Speedtest measures the maximum internet upload and download capacity of an internet-connected device. It floods the connection with data to gather accurate assessments of network performance and Quality of Service (QoS) metrics. These metrics involve download speed, upload speed, latency, packet loss, and jitter. Ookla operates 15,000 servers worldwide and lets Speedtest users connect to the most preferable server to deliver a best-case scenario network performance result.

Speedtest has many more features, not only internet connection performance measurements [18]. It also collects mobile network samples, including radio network coverage, signal conditions, and spectrum usage. They support multiple connection types including 5G and fiber. Speedtest Android and iOS application supports a video streaming tool that captures metrics from adaptive playback. This test provides insight into how and if the user network supports the expected video playback quality.

2.6 Windows Subsystem for Linux

Windows Subsystem for Linux (WSL) is a feature in Windows that runs an isolated Linux environment [19]. This feature allows users to run and utilize a Linux operating system without a separate virtual machine. WSL supports multiple Linux distributions, such as Ubuntu and Kali Linux, and many more. WSL allows for a software separation between the Windows and Linux systems, running command-line tools and *Bash* scripts, and many more features.

3. Related Work

Persico et al. [20] compares and identifies the internal network throughput between Amazon EC2 instances, called intra-cloud networking. They compared throughput between different sizes of EC2 instances: *m3.medium*, *m3.large*, and *m3.xlarge*. The result shows that different virtual machine sizes have different throughput capabilities and that *m3.xlarge* had the highest intra-cloud throughput. This study concludes that intra-cloud network throughput is higher when utilizing the private channel. And that there is not a significant difference across regions, but also that the optimal performance is achieved when instances of the same size are paired together. This study focuses on intra-cloud performance between VMs on Amazon Web Services. Our thesis will evaluate how cloud platforms influence the performance of VPN services. Persico et al., however, show that the throughput of the EC2 instances will be an order of magnitude higher than what our computers will achieve because of bandwidth limitations.

The Osswald et al. [21] study compares the throughput of different VPN solutions with different configurations. In the study, they utilized three different VPN services: WireGuard, IPsec, and OpenVPN. Their test environment consists of two virtual hosts running Ubuntu 18.04, connected via Intel X710 10 Gb/s network cards. They experimented using different encryption algorithms, with hardware acceleration through AES-NI and varying CPU pinning settings. After executing each test, they present the throughput for each VPN system using different configurations. Their results show that with hardware acceleration turned on and CPU pinning off, IPsec outperformed both WireGuard and OpenVPN. With CPU pinning turned on and hardware acceleration turned off, the throughput of WireGuard increased by 40%, outperforming both IPsec and OpenVPN. This study shows how different configurations impact the throughput of different VPN services. Our thesis is focused on investigating how various cloud platforms and their underlying infrastructure impact the performance of distinct VPN services. While Osswlad et al., in their study, delved into the effects of different configurations on VPN throughput, our thesis encompasses a broader range of performance indicators, shedding light on the intricacies of VPN performance in cloud environments.

In their study, Mackey et al. [22] compare the performance of OpenVPN and WireGuard on both local virtual machines and Amazon Web Services instances. The tests on AWS used *t2.micro* instances with 1GB of RAM and one virtual CPU core each. In their test executed locally, they used two Ubuntu virtual machines with 1GB of RAM and one virtual CPU core each, located on the same computer. They also tested the local virtual machines with two virtual CPUs to test how multi-threading affects performance. As a result of their experiment, they found that WireGuard outperforms OpenVPN in throughput, latency, and CPU utilization. The results also show WireGuard being more effective and scales with more CPU cores, compared with OpenVPN which is not more effective with more CPU cores. They concluded that this is a consequence of OpenVPN being single-threaded, and WireGuard being multi-threaded.

In the Chua et al. [23] study, a comparison of VPN services OpenVPN, WireGuard, and OpenConnect was made. They used four different setups to conduct their benchmark and assessment. These were office networks, home networks, cloud instances, and "road warriors". They used iPerf3 to perform the performance measurements. They also measured the time to upload and download a 1GB file from and to a file server. The measurements were done on different devices using different client apps to connect to the VPNs. Their study shows that WireGuard outperforms the other VPN services when the client app can be run in kernel space and use its own TUN drivers. Results show a difference in performance with different operating systems and VPN client apps. This study shows that different devices will be impacted by which VPN is chosen. In our thesis, we will use Windows machines running the client apps. However, using a Mac or Linux machine may produce different results.

Almost all of these studies compare VPN services against each other, utilizing hardware acceleration, CPU-pinning, different cryptographic keys, and different environments. None of these have tested how cloud providers' infrastructure impacts the VPN performance. There is an interesting case to be made to evaluate how the infrastructure impacts the deployment of VPNs. Many small companies cannot afford the infrastructure cost of deploying an on-site VPN solution. Cloud

VPNs can also help individuals affected by carrier-grade Network Address Translation (CG-NAT or CGN) [24], get remote access to resources in their home network while using an unsecured network. Our thesis aims to help companies and individuals decide which cloud platform, AWS or Azure, to utilize when deploying a VPN server, by analyzing and comparing the performance, cost, and user complexity of different VPN services.

4. Problem formulation

This thesis will compare the performance in terms of throughput, latency, jitter, packet loss, cost, and user complexity of implementing OpenVPN and WireGuard in two major cloud platforms, Amazon Web Services and Microsoft Azure. The main objective is to analyze how the cloud provider's infrastructure affects the VPN service performance and operational costs. Due to the increasing popularity of both VPNs and cloud platforms, many companies and individuals stand to gain from understanding which platform best fits their requirements. Answering the following research questions we aim to find which cloud platform is more suitable for running a VPN server.

RQ1: What are the differences in throughput and retransmissions between OpenVPN and WireGuard when running on AWS and Azure for data transfer between devices connected to the same VPN server?

RQ2: What are the differences in throughput, latency, jitter, and packet loss for data sent to the VPN server using OpenVPN and WireGuard on AWS and Azure?

RQ3: What upload and download throughput differences are observed when data is sent to the internet through the VPN gateway from both AWS and Azure?

RQ4: How do different services and data transfer impact the operational costs of the VPN server on both AWS and Azure?

RQ5: What is the difference in user complexity between AWS and Azure, regarding installation, configuration, and utilization?

AWS and Azure both have a range of many different use cases. It ranges from self-hosted virtual machines to artificial intelligence and machine learning capabilities. In this thesis, we will be focusing on the computer networking perspective. The network performance metric will include throughput, latency, jitter, and packet loss. User experience will entail the ease of using the cloud platforms and how complicated the installation process is, and operational costs will be the total costs during this thesis. Although our research aims to provide valuable insights into VPN service performance on AWS and Azure from a networking perspective, we acknowledge certain limitations of our work. Firstly, the study assesses performance metrics such as throughput, latency, jitter, and packet loss, which may not fully capture all aspects of VPN functionality or user experience. Secondly, evaluation is conducted under specific configurations and scenarios, which may not represent all possible use cases or deployment scenarios in real-world environments. Our study does not account for the potential variability in performance due to factors such as network congestion or geographic location, although they are discussed.

We have further limitations on our work because of a strict budget. As sponsors of this thesis, we are operating within a strict budget, we will only use one instance type in each cloud platform for our evaluation, we will however try to choose instances that are comparable concerning CPU count and RAM. This can however be a positive note. This aspect could be transferable to individuals who are on a budget and want to set up and use a VPN server in the cloud. It could also be transferred to small companies interested in utilizing a VPN server and want to compare the costs of a cloud-based VPN server versus an on-site one.

5. Method

In this thesis, we employ two different methods. The first method involves conducting a literature overview to acquire knowledge about the field and gain an understanding of the software we will be utilizing. The literature overview will predominantly consist of reading and analyzing scientific reports and official documentation from the software and cloud developers. We used Primo and Google Scholar to research literature on our subject. We will also use websites as a source of information in our work. This method will help us to answer our research questions about user experience and costs. Additionally, the literature overview will help us to identify previous work in the field of our thesis.

In addition, we will employ a second method, which is an experiment. We started planning the experiment while conducting the literature overview. Our goal is to make the experiment as fair as possible to be able to make a fair assessment of the network performance in the different cloud platforms. We will achieve this by performing measurements in the cloud platforms following the same experiment procedure. We will only use data centers in Sweden to minimize variance in the test results because geographical differences can impact the results. The experiment aims to collect data and analyze results to address our first, second, and third research questions regarding network performance differences. Network performance encompasses throughput, jitter, latency, and packet loss. The experiment will also help us to answer our fourth and fifth research questions about the user complexity of the different cloud platforms and operational costs.

6. Ethical and Societal Considerations

Our thesis does not encompass research ethical considerations, as the study will be conducted within a test environment devoid of any private information. The environment is designated for this thesis and includes the usage of unique passwords and private IP addresses. Furthermore, the data utilized in our throughput test is entirely synthetic and specifically for assessing network performance. It does not involve any real-world observations or sensitive information. The artificial data obviates the need for ethical review or consideration within the scope of our research.

Cloud computing presents ethical and societal considerations. When discussing cloud computing, many aspects are considered, such as environmental impacts, security, and economics. Cloud computing provides a more energy-efficient way of deploying and utilizing computational power than self-hosted servers because of its scalability and power efficiency. If cloud providers use renewable energy to power their data centers, positive environmental impacts could be achieved, reducing their carbon footprint. Cloud computing can offer better economic benefits than purchasing and maintaining hardware, thanks to features such as pay-as-you-use and hardware rental options. These features can reduce the costs associated with infrastructure maintenance and upgrades, enhancing financial efficiency and user experience.

However, not owning the infrastructure can have downsides. Security considerations have a critical role in the utilization of cloud computing services. The centralized nature of cloud computing and the reliance on third-party providers introduce inherent security risks. These risks could include data breaches, unauthorized access, and potential loss of sensitive information. Organizations and individuals must carefully evaluate the security measures implemented by cloud providers. These security measures are encryption, access control, and data privacy regulations. These measures must be taken to manage risks and ensure data confidentiality, integrity, and availability.

Although cloud providers are responsible for securing underlying infrastructure organizations and individuals are accountable for implementing security measures at the application and data levels. Together, organizations, individuals, and cloud providers can work toward a secure and resilient cloud computing ecosystem.

While VPNs are commonly employed to create a secure and encrypted virtual tunnel for communication between devices, they should not be viewed as the ultimate security solution. While they do provide an additional layer of security that can be used to protect against cyber threats, it's important to recognize that VPNs have limitations and may not address all security concerns comprehensively. Individuals should adhere to best practices to enhance online safety, especially when accessing or transmitting sensitive information. This includes using strong and unique passwords, enabling two-factor authentication whenever possible, keeping software and systems updated, and staying informed about emerging cyber threats. By integrating these practices alongside VPNs, individuals and companies can strengthen their security and better protect their online activities and data.

7. Experiment

This section provides detailed information on setting up various experiments for evaluation purposes. We also describe the installation process for WireGuard and OpenVPN and explain the different measurement tools used in the experiment. Our work will primarily concentrate on evaluating the network performance of the VPN services when running on AWS and Azure regarding throughput, jitter, latency, and packet loss. To do so, we set up three experiments to evaluate these metrics: client-to-client communication, client-to-VPN server communication, and when the VPN acts as a gateway to the internet.

7.1 Hardware

This section details the hardware used in the experiment. We employed two computers running Windows 11. Hardware for Computer A is seen in Table 1, and for Computer B in Table 2. Computer A has an internet connection speed of 250 Mbits/s, and Computer B has 1000 Mbits/s.

Table 1: Hardware specification of Computer A

Component	Specification
Name	Computer A
Operating System	Windows 11
CPU	AMD Ryzen 5 5600X 6-core 12-threads
RAM	32 GB DDR4 3200MHz
Storage	2000G Kingston KC3000 4.0 NVMe M.2 SSD
Network interface card	Realtek 1Gb/s Family Controller

Table 2: Hardware specification of Computer B

Component	Specification
Name	Computer B
Operating System	Windows 11
CPU	AMD Ryzen 7 5800X 8-core 16-threads
RAM	32 GB DDR4 3200 MHz
Storage	1000G Kingston FURY Renegade PCIe 4.0 NVMe M.2 SSD
Network interface card	Realtek Gaming 2.5GbE Family Controller

The hardware for the instances used in AWS and Azure are detailed in Tables 3 and 4.

Table 3: Hardware specification of AWS instance

Component	Specification
Name	t3.micro
Type	General purpose
Operating System	Ubuntu Server 22.04
VM Architect	x64
vCPU	2
RAM	1
Storage	8 GiB gp2 (SSD)

Table 4: Hardware specification of Azure instance

Component	Specification
Name	B2ts_v2
Type	General purpose
Operating System	Ubuntu Server 22.04
VM Architect	x64
vCPU	2
RAM	1
Storage	30 GiB Standard SSD

7.2 Topology

Our experiment consists of three topologies, as illustrated in Figures 1, 2, and 3. Each topology represents a unique setup, tailored for our specific tests. These configurations are designed to simulate various common VPN use cases, providing us with valuable network performance insights to address our research questions.

The first setup involves client-to-client communication and is simulated using the topology in Figure 1. Both computers are connected to the VPN server, with Computer B functioning as an iPerf server and Computer A as an iPerf client. In this setup, the VPN facilitates secure and encrypted communication between the two endpoints, mimicking real-world applications where data is transferred between connected devices within a private network. This setup is used to answer research question one.

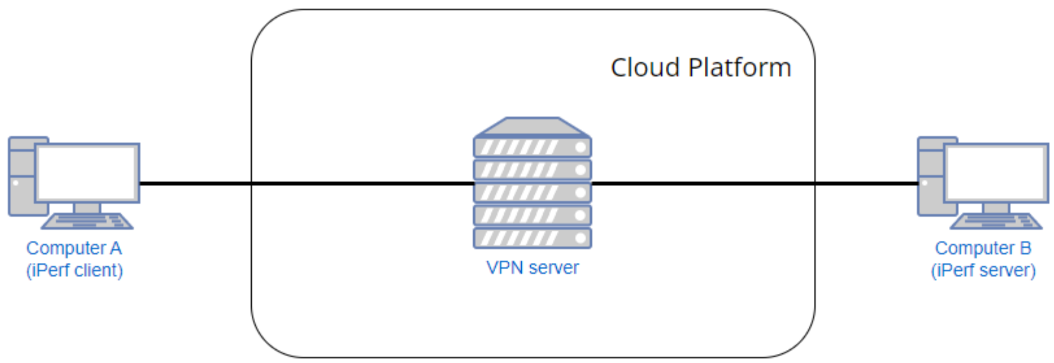

Figure 1: Setup 1 - Client-to-client communication via VPN.

The second setup involves client-to-VPN server communication and is simulated using the topology in Figure 2. Each computer, A and B, sends traffic to the VPN server. This scenario is used to test the throughput of the VPN server but also the responsiveness using ping, which mimics real-world applications where data is transferred to the VPN server in the cloud. This use case is most common if a company or individual also has other cloud-based services, that need secure access by a VPN. This setup gives insight into the performance of the client-to-VPN server throughput, packet loss, and jitter, but also the round trip time (RTT). This setup is used to answer research question two.

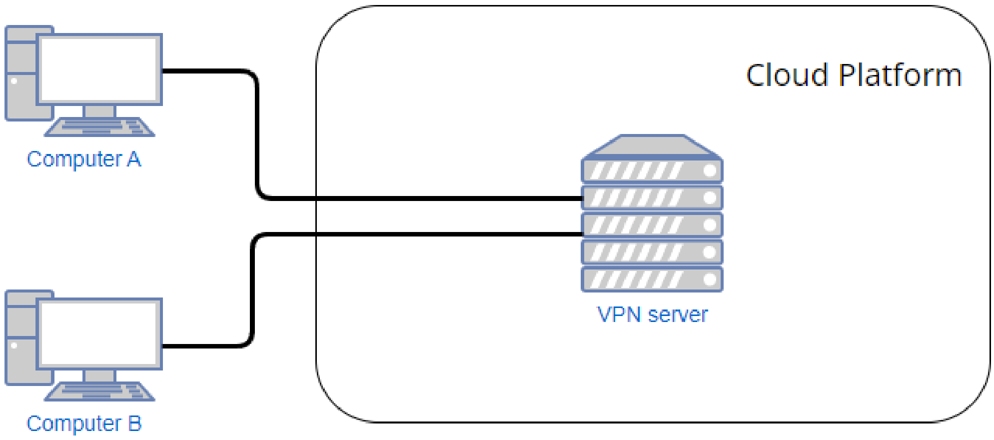

Figure 2: Setup 2 - Client-to-VPN server communication with and without VPN.

The third setup uses the topology seen in Figure 3. In this setup, the VPN server acts as a network gateway for clients to access the internet securely. Both Computer A and Computer B send traffic via VPN to the VPN server which then forwards the traffic to the internet (Speedtest server). This scenario simulates a situation where the VPN provides users security when accessing the internet from an insecure network. By routing all internet traffic through the VPN server, users can browse the web, and communicate online with enhanced security and privacy. This setup could be implemented by companies or private individuals, who would want to add an extra level of security while using an untrusted network, e.g. a public coffee shop network. This setup is used to answer research question three.

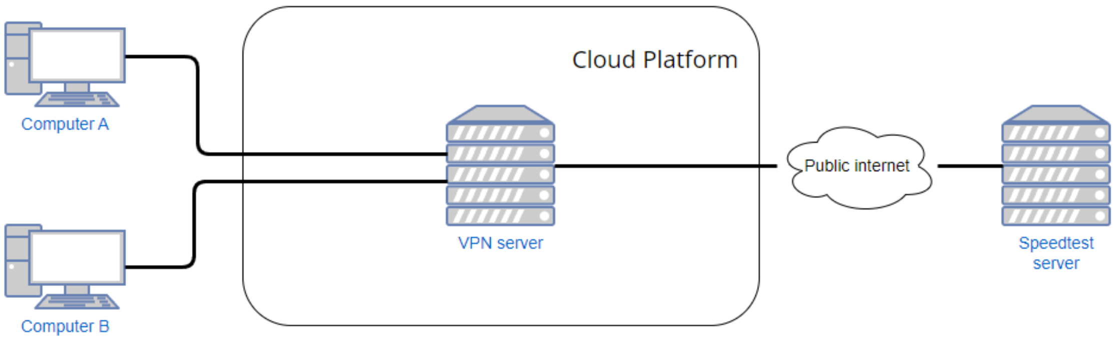

Figure 3: Setup 3 - VPN server as a network gateway for secure internet access.

These setups were chosen in our experiment to replicate the most common VPN use cases. This allows us to evaluate the network performance of the VPN services to see if there are any differences in the cloud providers' infrastructure. Combined with the cloud platforms' operational costs and user complexity, we will try to suggest which platform to use.

7.3 Performance metrics

In our thesis, we will evaluate the network performance of VPN services on AWS and Azure using a set of key performance metrics. These metrics provide valuable insight into the efficiency, reliability, and effectiveness of VPN services on the different cloud platforms. Below, we will introduce the different metrics and their importance in our evaluation.

The first performance metric is throughput or network throughput. Throughput is the amount of data, often in bits per second (bps), moved from one place to another in a given time [25]. The maximum throughput is the maximum data that can be transferred during a given period of time. The throughput of the VPN network is an important metric, it indicates how much data can be transferred and received between two devices. Higher throughput is, therefore, an indication of better efficiency of data transfer. This metric will be evaluated using iPerf and Speedtest. Speedtest measures the throughput when the VPN server acts as a gateway to the internet. iPerf, however, will be used for the client-to-client setup and the client-to-VPN server setup. iPerf is used to generate both UDP and TCP traffic.

The second metric is latency. Latency is the delay in packets traveling from source to destination. Latency is a way of measuring the responsiveness of the network. The latency in our evaluation will be the Round-Trip-Time (RTT) which is the time it takes for a small packet to travel from the source to the destination and back again [26, p.130]. This metric is important because it shows how responsive the VPN service is, and a lower latency indicates a better responsiveness of the VPN network. To measure the RTT we will utilize Ping. Ping is a CLI tool that sends Internet Control Message Protocol packets (ICMP) from a source to a destination, and measures the time it takes to get a reply.

The third metric we evaluate is jitter. Jitter measures the variation in latency between packets in a single flow [27, p.365]. It indicates the consistency and stability of the network connection, with lower jitter values reflecting a more stable network performance. In a VPN network, the added overhead of encryption and decryption can increase the jitter, lower jitter is, therefore, more advantageous and indicates a better way of handling the added overhead. The jitter measurements are collected during the iPerf UDP throughput performance test.

The fourth, and last metric is packet loss. Packet loss refers to data packets lost or discarded during transmission [28]. It can occur due to network congestion, hardware issues, or other factors, leading to degraded performance and potential data loss. Packet loss acts differently depending on what transmission protocol is used. The TCP protocol is stateful and retransmits packets that the receiver classifies as lost or out of order, while the UDP protocol does not handle any loss of packets. A lower retransmission rate or lower packet loss percentage is, therefore, an indication of a more robust and stable network. Retransmissions and packet loss metrics are gathered during the iPerf throughput performance tests, as these are some of the metrics recorded by the tool.

These four performance metrics will be utilized to evaluate the effectiveness of VPN services across AWS and Azure. Subsequently, the results from each platform will be compared to determine the most suitable cloud platform for hosting the VPN server.

7.4 VPN Installation

This section details how the VPNs were installed on the instances. We used two different installation tools for OpenVPN and WireGuard. We used these tools to automate the installation process, instead of manually installing each service and creating the necessary keys and certificates, to save time. This improved our efficiency and standardized the installations so that the WireGuard and OpenVPN installations were the same on AWS and Azure.

7.4.1 WireGuard

To install WireGuard on the instances, we opted for PiVPN [29]. PiVPN is a set of shell scripts that can install WireGuard and transfer to create a VPN server. It was chosen due to its simplicity and flexibility, despite its name it does not only work on Raspberry Pi [30], it can also run on x86_64 systems. The installer offers an intuitive graphical user interface (GUI) that simplifies the setup process. Through this interface, we could specify crucial settings such as the port number, network interfaces, and client connection configurations. PiVPN is also used to create the client's configuration. We used the *pivpn add* command to make the client configuration and then copied the files to the WireGuard Client APP to computers A and B.

To install the WireGuard Client on computers A and B, we used the WireGuard MSI installer from [31]. To connect each client to the VPN server we added a VPN tunnel that used the client configuration created by the *pivpn add* command.

7.4.2 OpenVPN

To install OpenVPN we followed this installation guide [32], which specifies an OpenVPN "road warrior" installer for Ubuntu [33]. The installer is simple and has an interactive command line interface (CLI). Like PiVPN, the installer for OpenVPN also includes the option to specify interface, port number, and client connection configuration settings.

To connect the client to the VPN server we used OpenVPN Connect [34]. We copied the configuration file to the OpenVPN Connect Client Software, that the "road warrior" installer created, to connect to the VPN.

7.5 WSL - Installation

We utilized Ubuntu through WSL 2 on our computers to conduct network performance tests using iPerf3. This decision was made due to the limitations of the iPerf3 software for Windows. In the first iteration of our test, we employed the iPerf3 tool on Windows. Interestingly, we found that the iPerf3 tool for Windows yielded lower test results when compared to Ubuntu. This might be a result of several different things. To avoid the problem, we used Ubuntu through WSL 2 during our iPerf3 experiments.

To install WSL on Windows, we ran the *wsl –install* command in Windows Command Prompt as an administrator. This enables all the necessary features and installs the Ubuntu distribution [35].

7.6 iPerf3 and Speedtest

This section provides details on how iPerf3 and Speedtest were installed and utilized. These tools were used to help us measure the performance metrics in question, giving us very detailed information about the tests we performed.

7.6.1 iPerf3 - Installation and Usage

The installation of iPerf3 was done through the Ubuntu package manager, "APT". Installing the software is simple, all we had to do was to run the command "sudo apt-get install iperf3". After the installation, the software can be executed by typing "iperf3 ..." followed by some desired flags, see Table 5 for the flags we used in this thesis. Different flags determine the behavior of the application, such as running in TCP or UDP mode or limiting the bandwidth. We also had to specify whether our application should be a server or a client. In the case where we experiment with UDP, the command can look like this: "iperf3 -c 172.0.0.1 -u -b 200M". This will initiate a UDP connection with the server host and limit the bandwidth to 200 Mbit/s.

Table 5: Description of iPerf3 commands

Command	Description
iperf3 -s	Starts the iperf server on the device.
iperf3 -c x.x.x.x	Establishes the device as a client and sends data to the specified IP address.
-t	Set time in n seconds to transmit data for.
-b	Sets the targeted bitrate to n bits/second. E.g. 10M is 10 Mbits/s or 0 for no bitrate limitation.
-O	Omits the first n seconds of the test
-P	Set the number of multiple client streams

7.6.2 Speedtest - Installation and Usage

The installation process of Speedtest is similar to iPerf. To install Speedtest we followed the directions found in the developers section on the Speedtest website [36]. After installation, we could initiate a speed test against a desired server simply by executing the program followed by a "-s" flag and a server ID which could be found on the internet, or by using the "-L´´ flag. We also had to ensure we were using the correct network interface, which we provided with the "-i" flag followed by our local IP address for the VPN interface. The information returned by the program describes the latency, packet loss, download speed (Mbit/s), and upload speed (Mbit/s).

8. Result - Performance evaluation

In this section, we evaluate the performance of the system by measuring the metrics introduced earlier together with an analysis of the data of our experiments. These experiments were conducted to evaluate the performance impact that AWS and Azure infrastructure have on OpenVPN and WireGuard. We also provide an analysis of the data, that connects the results to our research questions. The commands used in Setup 1, 2, and 3 are shown in Appendix A.

8.1 Performance Evaluation - Setup 1

In this section, we provide the results from setup 1, considering the topology illustrated in Figure 1. We utilized both single TCP and multiple TCP streams to measure the throughput and the number of retransmissions. We repeated each test 5 times for 20 seconds each. The traffic is sent from Computer A (iPerf client) to Computer B (iPerf server), both computers are connected to the VPN server running WireGuard and OpenVPN on AWS and Azure.

Figure 4 illustrates the average throughput of the system with a single TCP stream. Figure 5 shows the average number of retransmissions of the system on both Azure and AWS platforms, using either WireGuard or OpenVPN.

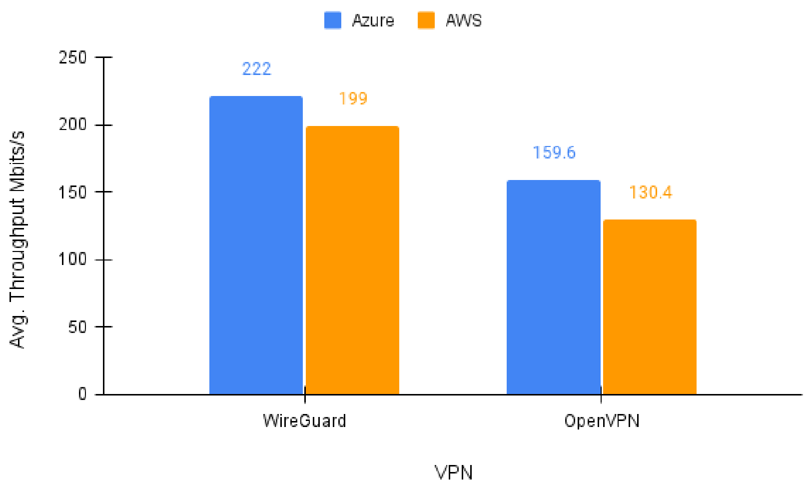

Figure 4: Average throughput for a single TCP stream from Computer A (iPerf client) to Computer B (iPerf server).

In Figure 4, Azure achieved a higher average throughput when using WireGuard and OpenVPN. These results indicate that WireGuard can achieve a higher throughput for single TCP stream traffic.

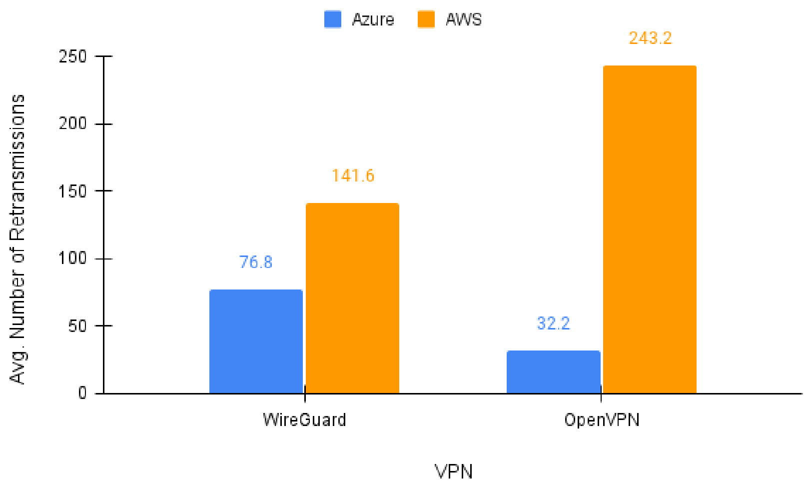

Figure 5: Average number of retransmissions for a single TCP stream from Computer A (iPerf client) to Computer B (iPerf server).

Figure 5 illustrates the average number of retransmissions. The results indicate that AWS has a higher number of retransmissions. OpenVPN achieved a lower throughput and a higher number of retransmissions on AWS. And this is what we expect. When higher retransmissions occur, it often indicates more instances of packets being corrupted or lost during transmissions. This can lead to inefficiencies in the network resulting in a lower throughput. On Azure, however, OpenVPN achieved a lower throughput and lower retransmissions. We believe these results may be because of how the different VPN services or iPerf3 handle traffic at the application layer.

To assess the throughput and retransmissions experienced by multiple users simultaneously using a VPN service, we conducted TCP traffic tests using five streams. This evaluation is used to see how the system handles traffic sent simultaneously. The average throughput for five streams can be seen in Figure 6 and the average number of retransmissions is shown in Figure 7.

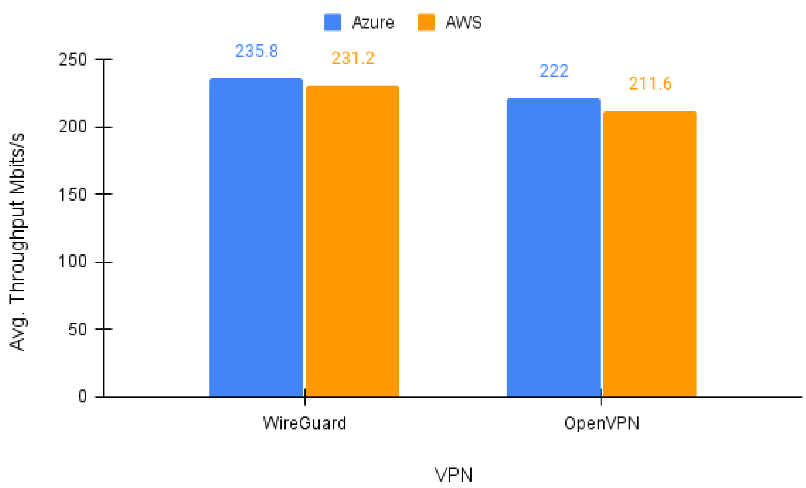

Figure 6: Average throughput for multiple TCP streams from Computer A (iPerf client) to Computer B (iPerf server).

As seen in Figure 6, when sending data in multiple TCP streams Azure achieves a higher throughput with WireGuard and OpenVPN. The results also show that WireGuard outperforms OpenVPN in terms of throughput.

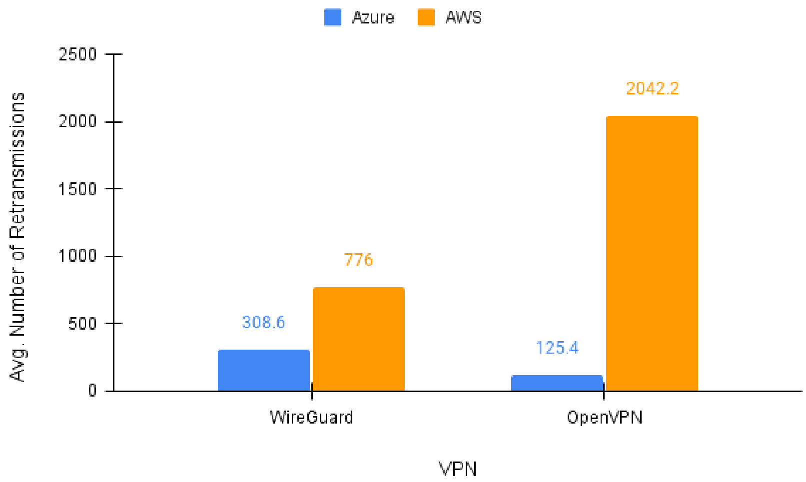

Figure 7: Average number of retransmissions for multiple TCP streams from Computer A (iPerf client) to Computer B (iPerf server).

Looking at Figure 7, we see a huge difference in the number of retransmissions. Azure has a much lower number of retransmissions compared to AWS. With WireGuard AWS has approximately 2.5 times higher number of retransmissions and with OpenVPN 16.3 times higher compared to Azure.

These findings show that, when data is sent in multiple TCP streams a slightly higher throughput is achieved but also the total number of retransmissions. WireGuard achieved a higher throughput and lower number of retransmissions on both platforms. Azure outperformed AWS regarding the highest average throughput and lowest number of retransmissions. There are contradictory results, on Azure a higher throughput yielded higher retransmission but on AWS a lower throughput yielded higher retransmissions. These results could probably be attributed to the VPN services, how the different platforms handle data traffic, or how iPerf3 handles the test.

8.2 Performance Evaluation - Setup 2

In this section, we present the results from setup 2. To evaluate the performance of this system, we used the topology seen in Figure 2. In this experiment, we performed tests sending traffic from Computer A and Computer B. Computer A has a 250 Mbits/s internet connection and Computer B has a 1000 Mbits/s connection. In this experiment, we utilized a single TCP stream and multiple TCP streams to measure throughput and retransmissions. We repeated each test 5 times for 20 seconds each. From Computer B we also utilized a single UDP stream to measure UDP throughput, packet loss, and jitter. This test was done only once targeting different bandwidth limits. From Computer B we also measured the latency (RTT) for each VPN service and cloud platform. Both computers are connected to the VPN server running WireGuard and OpenVPN on Azure and AWS. We also provide baseline results for all the tests when no VPN is used. We believe the "No VPN" results are important to show how the different VPN services impact the throughput, latency, jitter, and packet loss. However, we will not provide an in-depth analysis of these results as our thesis focuses on the impact that the cloud platforms have on WireGuard and AWS.

The results of the single TCP stream from Computer A are illustrated in Figure 8 and 9. These results show the average throughput of the system with a single TCP stream and the average number of retransmissions on both Azure and AWS, using WireGuard, OpenVPN, or no VPN.

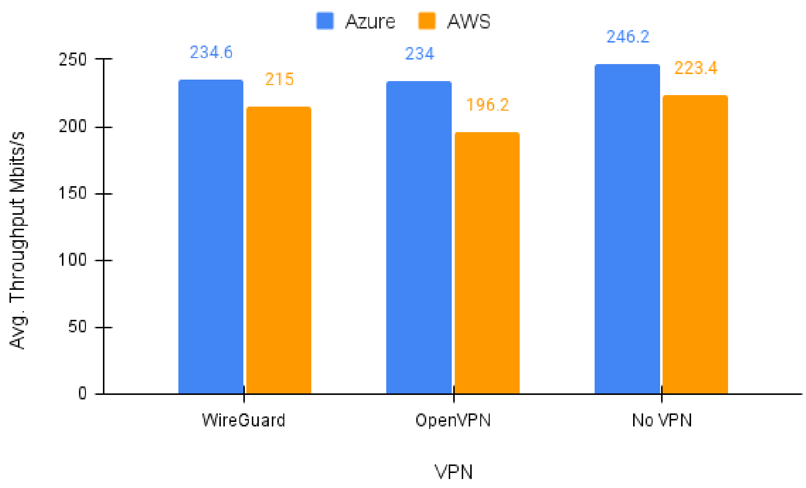

Figure 8: Average throughput for a single TCP stream from Computer A to VPN server.

Looking at Figure 8 it is seen that Azure achieved a higher average throughput when using WireGuard and OpenVPN. It is also seen that the difference between WireGuard and OpenVPN is negligible on Azure. On AWS there is approximately a 10% performance difference between WireGuard and OpenVPN. We can also see that Azure has a higher throughput without any VPN. These results indicate higher throughput for single TCP stream traffic can be achieved on AWS.

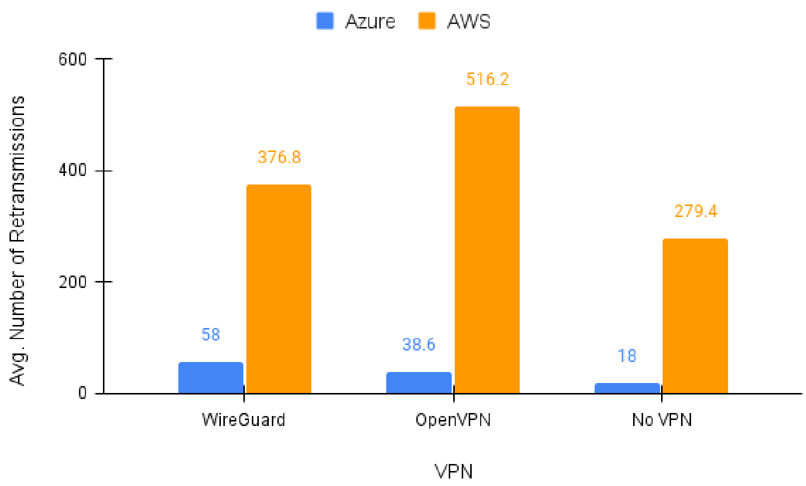

Figure 9: Average number of retransmissions for a single TCP stream from Computer A to VPN server.

Figure 9 illustrates the average number of retransmissions for the single TCP stream from Computer A. We can see a huge difference in the number of retransmissions. The results show that WireGuard and OpenVPN achieved fewer retransmissions on Azure. The number of retransmissions on AWS was approximately 6.5 times higher with WireGuard and 13.4 times higher with OpenVPN, compared to the results on Azure. The average number of retransmissions is also lower on Azure without VPN.

These findings show that when data is sent directly to the VPN server using either WireGuard or OpenVPN, higher throughput and lower numbers of retransmissions are achieved on Azure. The results also show that with WireGuard on AWS, Computer A can saturate 93.6% of its internet connection with VPN traffic, this is probably due to the added overhead of the VPN protocols.

Figure 10 and 11 show the average throughput achieved on Computer B running a single TCP stream. These figures show the average throughput and the average number of retransmissions of the system on both Azure and AWS, using WireGuard, OpenVPN, or no VPN running a single TCP stream.

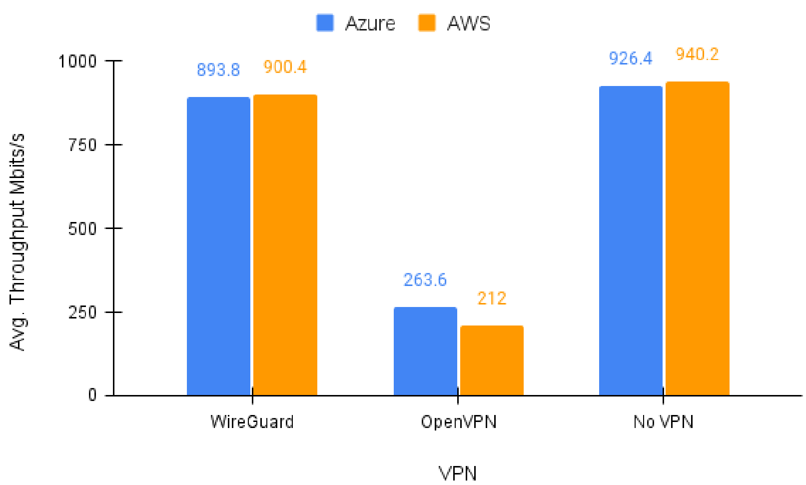

Figure 10: Average throughput for a single TCP stream from Computer B to VPN server.

Looking at Figure 10 it is evident that WireGuard achieves a higher throughput compared to OpenVPN. This we, however, knew from related work [22]. We can also see that the average throughput on Azure is higher with OpenVPN but slightly lower with WireGuard. The average throughput without a VPN is higher on AWS.

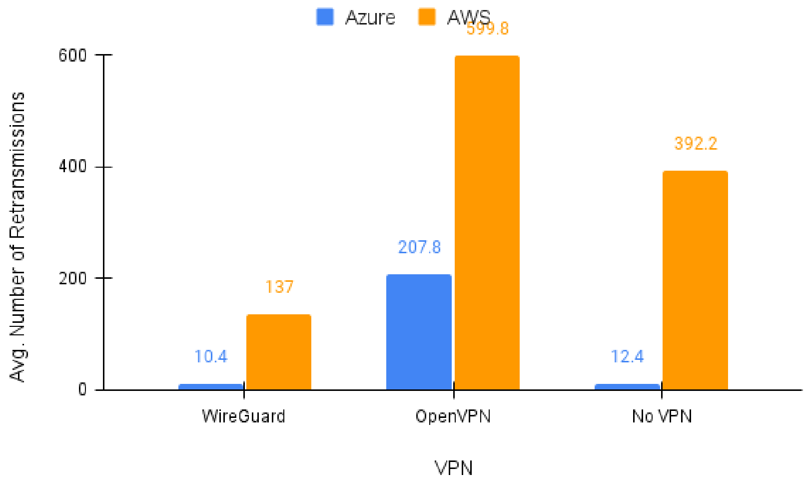

Figure 11: Average number of retransmissions for a single TCP stream from Computer B to VPN server.

In Figure 11 it is seen that the average number of retransmissions is lower on Azure for both VPN services. The results also indicate that the number of retransmissions is higher on OpenVPN than on WireGuard. Without a VPN, the average number of retransmissions is lower on Azure, however the No VPN results are higher than those when using WireGuard.

These findings indicate that for a computer or network with 1000 Mbits/s (Computer B) internet connection, WireGuard is preferable as it can reach a higher throughput compared to OpenVPN. WireGuard paired with Azure achieved a higher throughput and lower number of retransmissions. For a network with a 250 Mbits/s (Computer A) internet connection, the results indicate that both VPN services are preferable in terms of throughput, for the number of retransmissions there are fewer retransmissions with OpenVPN. Azure, however, achieved far fewer retransmissions compared to AWS.

To assess the throughput and retransmissions experienced by multiple users simultaneously using a VPN service, we conducted a TCP traffic test using five streams. This evaluation is used to see how the system handles traffic sent simultaneously with different internet connections. These tests show the average throughput for five streams and the average number of retransmissions for an internet-connected device with 250 Mbits/s (Computer A) and 1000 Mbits/s (Computer B). The average throughput for five streams and the average number of retransmissions for Computer A can be seen in Figure 12 and 13 respectively.

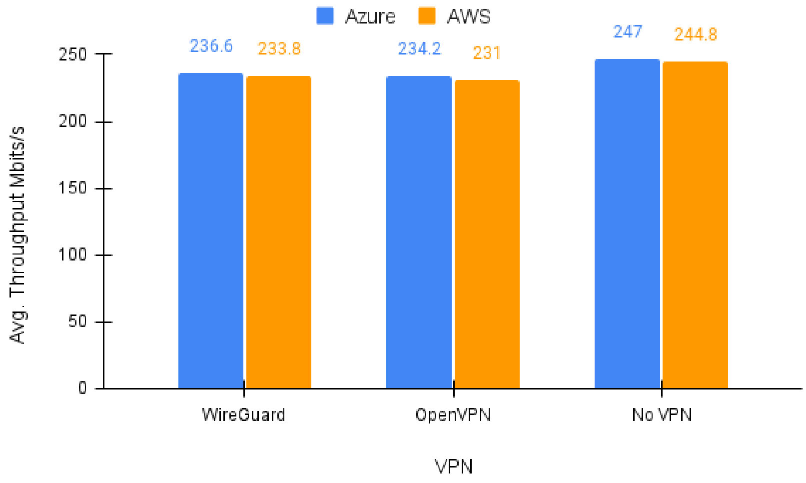

Figure 12: Average throughput for multiple TCP streams from Computer A to VPN server.

Looking at Figure 12 it is evident that Azure and AWS achieved very similar throughput results. Azure achieved a slightly higher throughput for both VPN services. A higher throughput without VPN is also achieved on Azure.

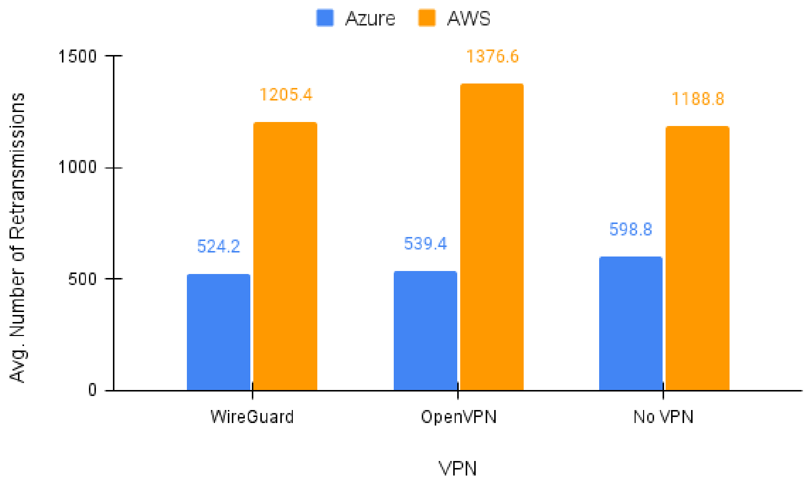

Figure 13: Average number of retransmissions for multiple TCP streams from Computer A to VPN server.

As seen in Figure 13 the average number of retransmissions for five streams was higher on AWS by approximately 2.5 times, compared to the results gathered on Azure. The results also show that the average number of retransmissions on Azure with WireGuard or OpenVPN has no significant difference. These findings suggest that for a device with a low-bandwidth internet connection, Azure paired with either OpenVPN or WireGuard will provide similar performance. There was no significant difference in the average number of retransmissions between WireGuard, OpenVPN, and no VPN.

The results for the average throughput for five streams and the average number of retransmissions for Computer B (1000Mbits/s) are seen in Figures 14 and 15 respectively.

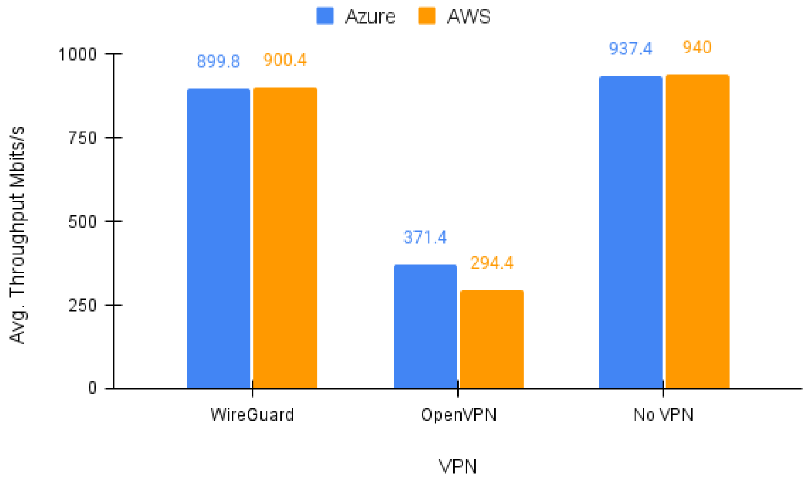

Figure 14: Average throughput for multiple TCP streams from Computer B to VPN server.

Looking at Figure 14 it is seen that WireGuard achieved a higher throughput compared to OpenVPN. We can also see that with WireGuard there is no significant difference in throughput between Azure and AWS. With OpenVPN, however, the average throughput is higher on Azure by about 77 Mbits/s. A higher throughput is achieved when not using a VPN service and AWS had a higher throughput compared to Azure.

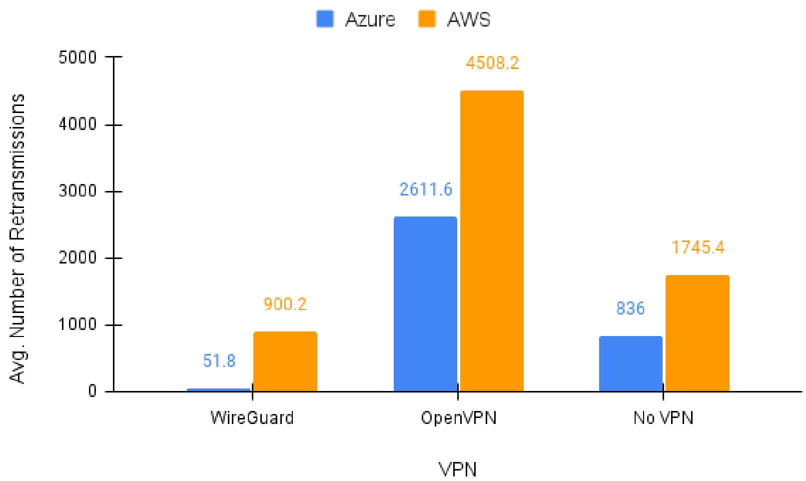

Figure 15: Average number of retransmissions for multiple TCP streams from Computer B to VPN server.

In Figure 15 the results show that AWS has far more number of retransmission compared to Azure. We can also see that WireGuard had a lower number of retransmissions. When not using a VPN, the retransmissions were higher compared to those on WireGuard, however, Azure has fewer retransmissions compared to AWS.

These findings show that when data is sent in multiple TCP streams a slightly higher throughput is achieved. It is also evident that the total number of retransmissions also increased. For devices with a high-bandwidth internet connection, Azure paired with WireGuard is the only option as it achieves a higher throughput and lower number of retransmissions. OpenVPN only achieved a third of the bandwidth with a higher average number of retransmissions. For devices with a low-bandwidth internet connection, Azure paired with either WireGuard or OpenVPN achieved the highest throughput and lowest number of retransmissions.

To assess the throughput, packet loss, and jitter associated with UDP traffic, we conducted a UDP traffic test using one stream. These tests are done to evaluate and see how the system handles UDP traffic at different bandwidth limits. For each VPN service and cloud platform, we targeted 5 different bandwidth limits and collected the throughput, packet loss, and jitter. These tests were conducted using Computer B which has a 1000 Mbit/s internet connection sending traffic to the VPN server. Computer B was connected to the VPN server running WireGuard and OpenVPN on Azure and AWS. We also provide the results without a VPN. The throughput, packet loss, and jitter for the single UDP stream using WireGuard on both platforms are seen in Figure 16, 17, and 18 respectively.

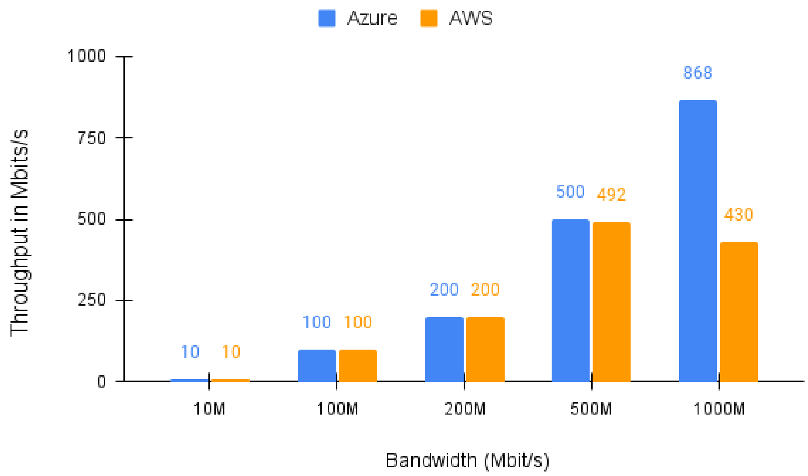

Figure 16: Throughput for a single UDP stream from Computer B to VPN Server at different bandwidth limits using WireGuard.

In Figure 16 we can see that at lower bandwidth limits (10Mbit-200 Mbit) Azure and AWS both hit the targeted throughput. At the 500Mbit/s and 1000Mbit/s target Azure achieved a higher throughput.

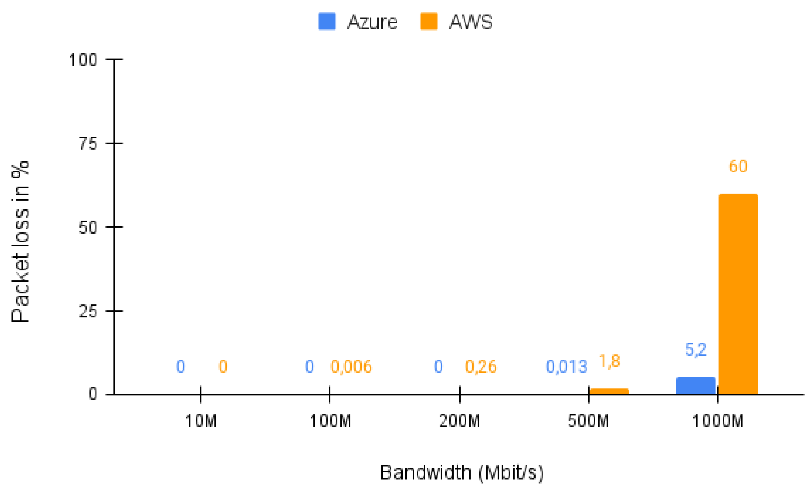

Figure 17: Packetloss for a single UDP stream from Computer B to VPN Server at different bandwidth limits using WireGuard.

Looking at Figure 17, we can see that at these higher limits, AWS has a higher number of packet losses. AWS has 1,8% packet loss at 500Mbit/s and 60% packet loss at 1000Mbit/s limit. Azure only lost 5,2% of the packets at the 1000Mbit/s limit.

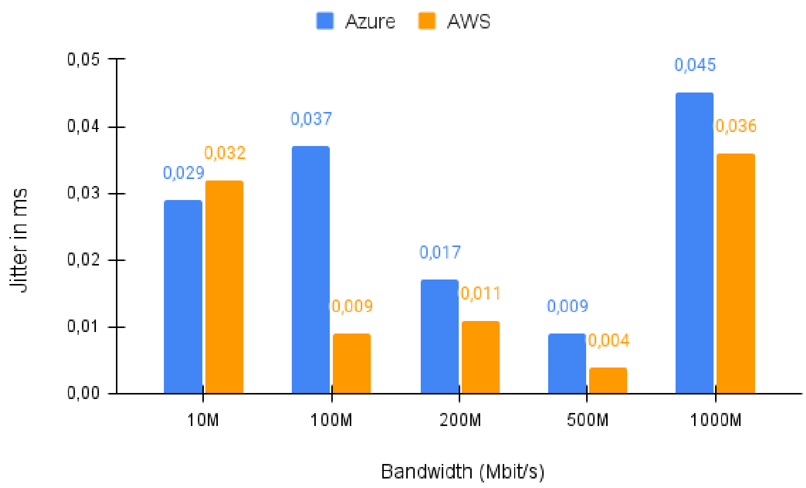

Figure 18: Jitter for a single UDP stream from Computer B to VPN Server at different bandwidth limits using WireGuard.

Figure 18 illustrates the jitter at the different bandwidth targets. The results show a high variation between AWS and Azure. We can however see that Azure had a higher jitter for most tests.

These findings indicate that at a higher single UDP stream throughput, higher packet loss is expected from both AWS and Azure, but the amount of packet loss was higher on AWS. The jitter varies for both platforms, we can see that most of the tests show that Azure has a higher jitter. These tests could however be severely impacted by Computer B's network, as many connected devices in the network use the same internet connection.

The results for the throughput, packet loss, and jitter using OpenVPN on AWS and Azure are shown in Figures 19, 20, and 21 respectively.

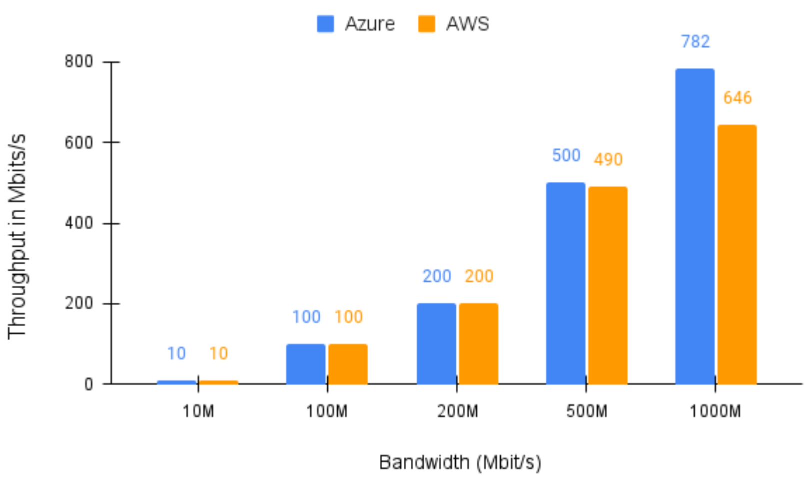

Figure 19: Throughput for a single UDP stream from Computer B to VPN Server at different bandwidth limits using OpenVPN.

As seen in Figure 19, it is seen that Azure achieved a higher throughput at 500Mbit/s and 1000Mbit/s bandwidth limits. These results share similar results to the ones on WireGuard, which also shows that Azure could achieve higher throughput at higher bandwidth limits.

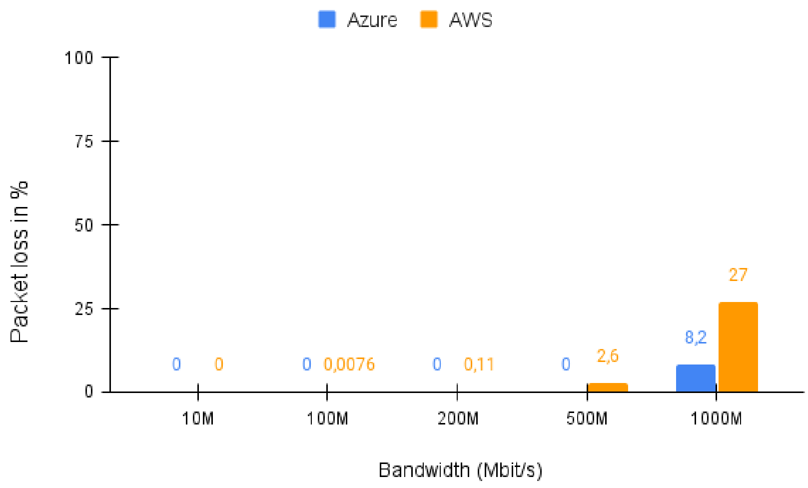

Figure 20: Packet loss for a single UDP stream from Computer B to VPN Server at different bandwidth limits using OpenVPN.

The packet loss results seen in Figure 20, also depict similar results to the ones with WireGuard. At the higher bandwidth limits, packet loss was higher on AWS compared to Azure.

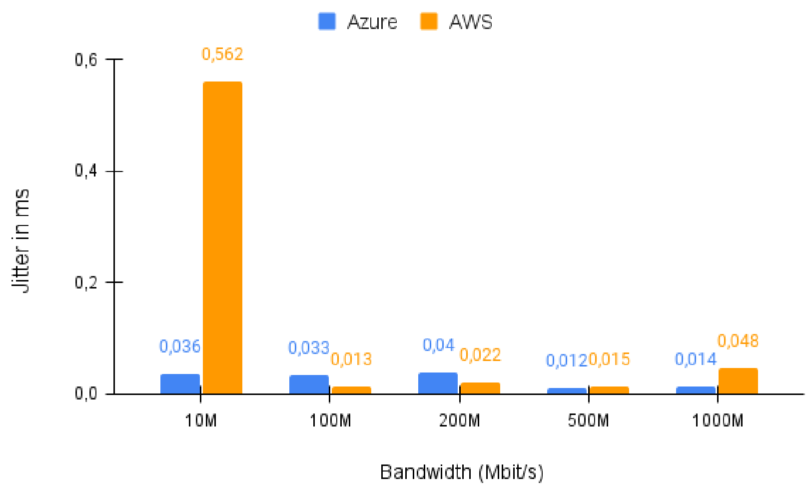

Figure 21: Jitter for a single UDP stream from Computer B to VPN Server at different bandwidth limits using OpenVPN.

Figure 21, illustrates the jitter during the bandwidth limits tests. We can see that these tests show a high variation on AWS.

These findings show that for single UDP traffic, Azure achieves a higher throughput with both OpenVPN and WireGuard. We can also see that WireGuard achieved a slightly higher throughput on Azure. The jitter was lower on WireGuard compared to OpenVPN for both AWS and Azure. The packet loss was the lowest with WireGuard on Azure. These findings suggest that WireGuard paired with Azure, achieves a higher throughput and lower packet loss. The jitter was however higher on Azure, although the differences were between 1-10 microseconds.

The results for the throughput, packet loss, and jitter without VPN, sending traffic to both AWS and Azure are shown in Figure 22, 23, and 24 respectively.

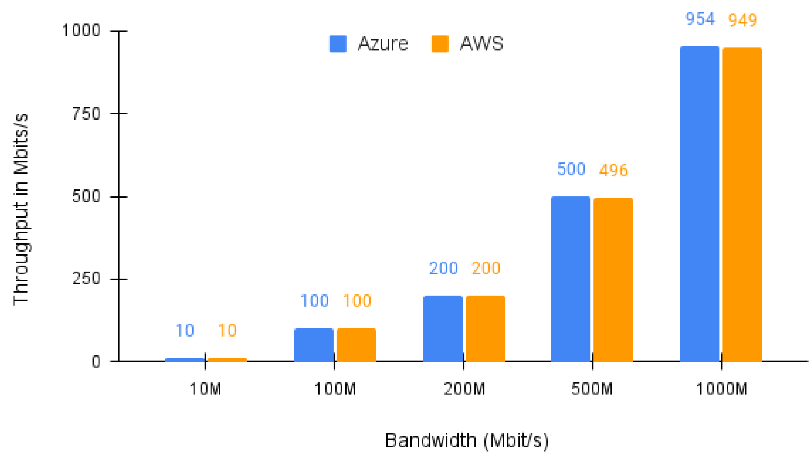

Figure 22: Throughput for a single UDP stream from Computer B to VPN Server at different bandwidth limits without VPN.

In Figure 22 we see Azure achieved a slightly higher throughput at the 500M and 1000M bandwidth targets.

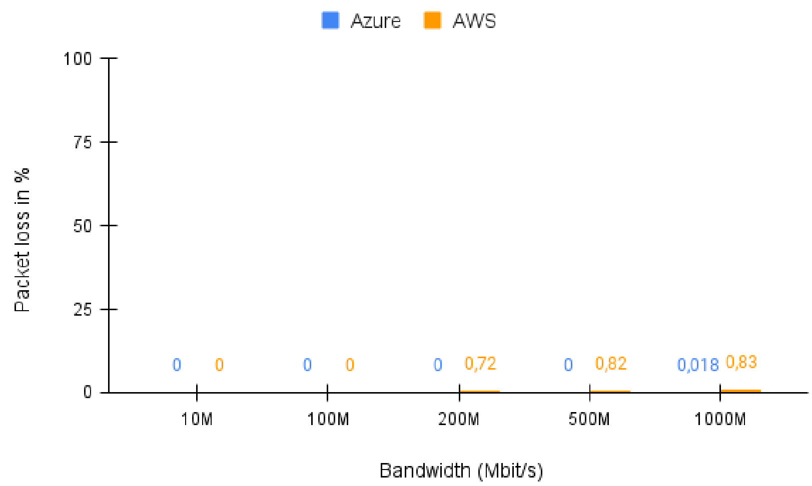

Figure 23: Packet loss for a single UDP stream from Computer B to VPN Server at different bandwidth limits without VPN.

The packet loss for each bandwidth target test can be seen in Figure 23. The graph shows that while packet loss is generally low, Azure experiences the least packet loss.

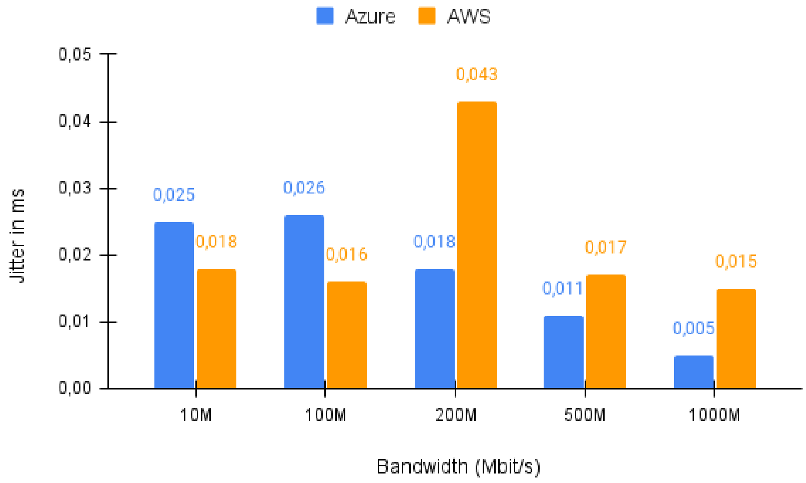

Figure 24: Jitter for a single UDP stream from Computer B to VPN Server at different bandwidth limits without VPN.

In Figure 24 we see the results for the jitter of each bandwidth target. The results indicate that Azure achieved higher jitter for lower bandwidth targets (10M and 100M). For higher bandwidth targets (200M-1000M) Azure has achieved lower jitter. These findings show that Azure outperforms AWS in client-to-VPN server communication for UDP traffic without VPN, as it achieved a higher overall throughput, lower packet loss, and a lower amount of jitter.

To assess the latency we sent data packets from Computer B to the VPN server and measured the average RTT. The results are seen in Figure 25. These tests are used to evaluate the latency of the system.

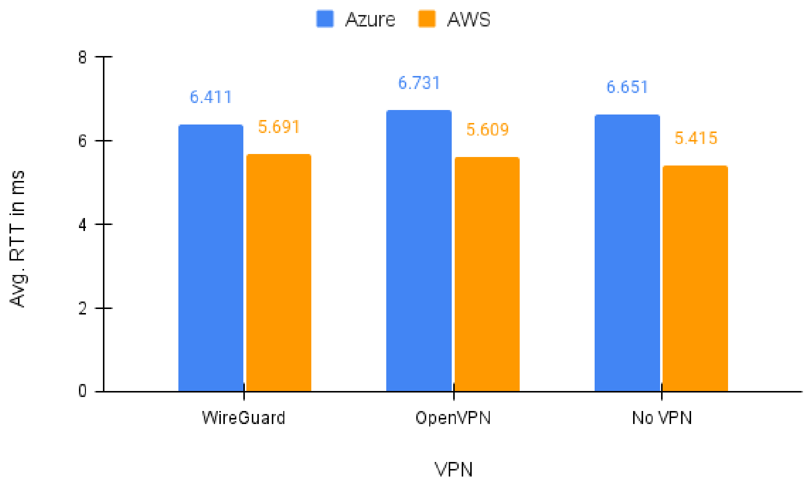

Figure 25: Average RTT latency from Computer B to VPN server.

Looking at Figure 25 we can see that the latency was higher on Azure compared to AWS. On Azure, WireGuard had a lower latency compared to OpenVPN. On AWS, the opposite is observed, OpenVPN had a lower latency. It is important to note that latency is affected by many different variables, but one of them is the length the data packet has to travel. Computer B is located closer to the AWS data center, which could impact the results. The No VPN results show that a higher latency is achieved on Azure. This result could be caused by processes running in the background on Computer B.

8.3 Performance Evaluation - Setup 3

In this section, we present the results of Setup 3, utilizing Topology 3. These tests aimed to measure the maximum download and upload throughput to the internet. We evaluated the performance when Computer A and Computer B utilized either AWS or Azure as the infrastructure for the VPN server, which also acted as their gateway for internet access. To conduct this assessment, we employed Speedtest, with the clients targeting the same Speedtest server. Additionally, we conducted a baseline test to demonstrate the maximum throughput achievable by each instance, when the instance interface targeted the same Speedtest server. The results for the baseline test are depicted in Figure 26.

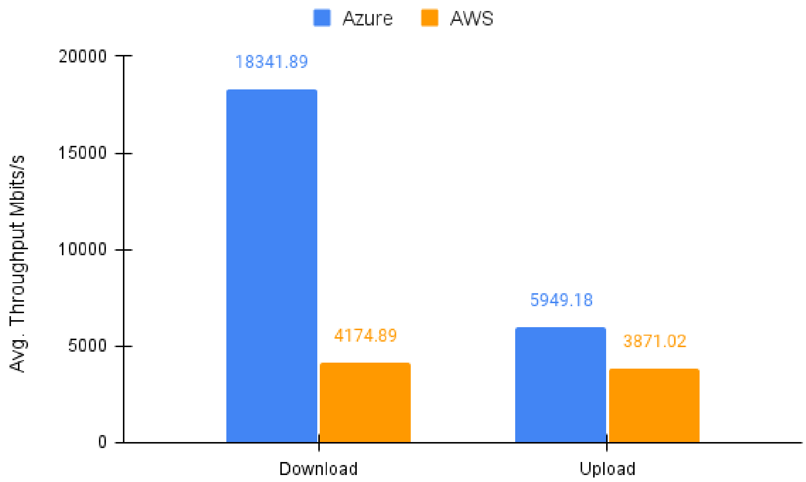

Figure 26: Baseline throughput measurements for instances hosted on AWS and Azure.

In the baseline throughput test result, we can see that Azure has a higher download and upload throughput. The download throughput of the Azure instance is approximately 4 times higher and the upload speed is approximately 1.5 times higher than the AWS instance. These findings indicate that the interface of the Azure instance is better equipped to handle higher throughput, or that the instance model has a better-equipped network interface card (NIC) than the AWS instance model. This indicates that Azure can achieve a higher upload and throughput, which is preferable for a VPN server.

8.3.1 WireGuard

This section provides data for our throughput test using Speedtest with WireGuard VPN. The data for Computer A can be found in Figure 27 and for Computer B in Figure 28. Each computer was connected to the VPN server via WireGuard, and both upload and download speeds were measured.

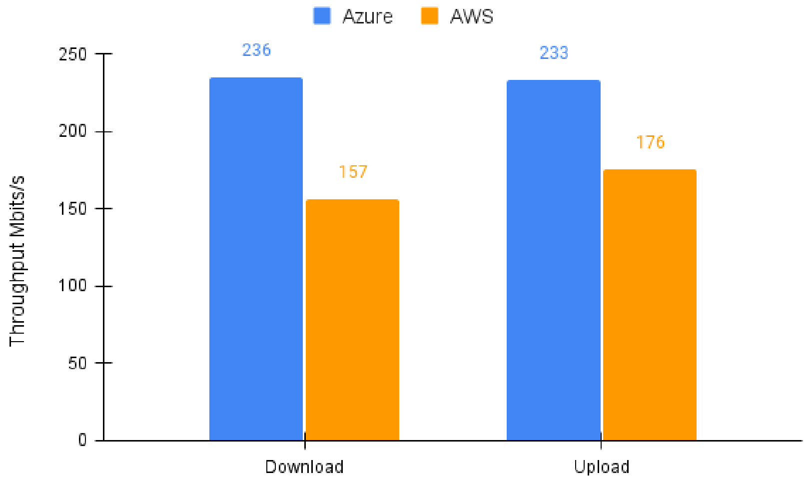

Figure 27: Comparison of throughput to the internet via VPN gateway from Computer A using WireGuard on AWS and Azure.

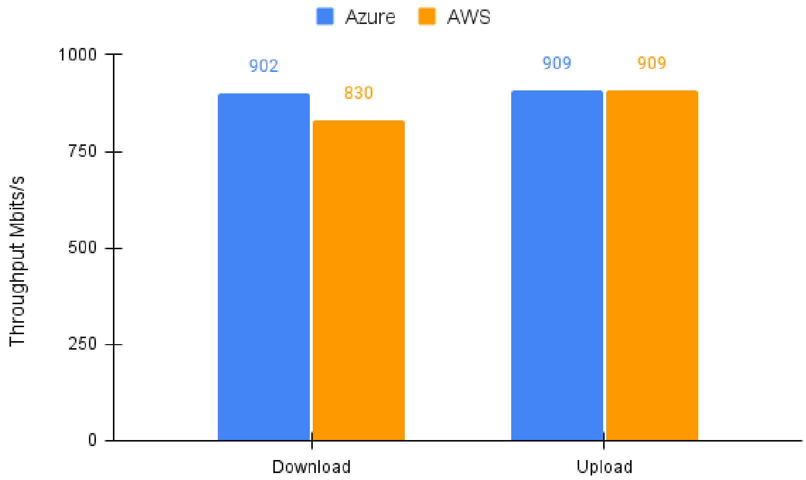

Figure 28: Comparison of throughput to the internet via VPN gateway from Computer B using WireGuard on AWS and Azure.

As seen in Figure 27, Computer A had a 50% higher download throughput and 32% upload throughput when connected to the Azure WireGuard VPN server. As seen in Figure 28, Computer B had only an 8.6% higher download throughput (equivalent to 73 Mbits/s) and no significant variation in the upload throughput when connected to the Azure WireGuard VPN server. These findings indicate, however, that Azure infrastructure outperforms AWS in terms of download throughput when using WireGuard, the upload throughput is harder to access. However, the significant difference between AWS and Azure for Computer A could be because the Speedtest server was overwhelmed with other tests, resulting in a lower result for AWS.

8.3.2 OpenVPN

This section provides data for our throughput test using Speedtest with OpenVPN protocol. The data for Computer A can be found in Figure 29 and for Computer B in Figure 30. Each computer was connected to the VPN server via OpenVPN, and both upload and download speeds were measured.

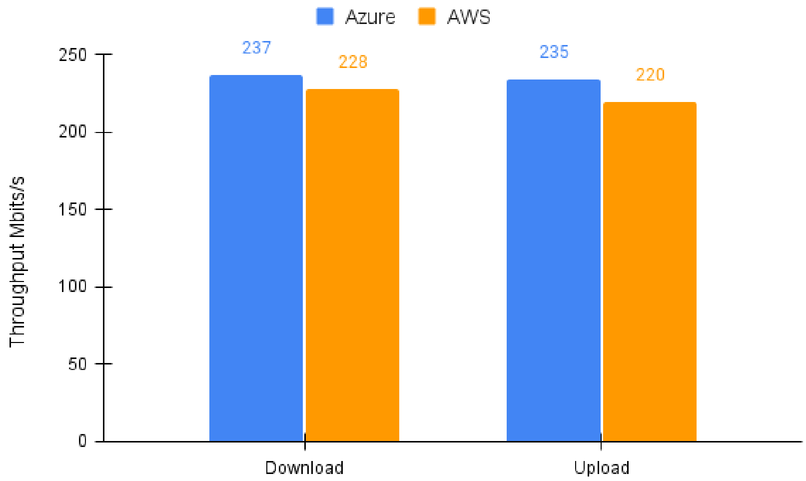

Figure 29: Comparison of throughput to the internet via VPN gateway from Computer A using OpenVPN on AWS and Azure.

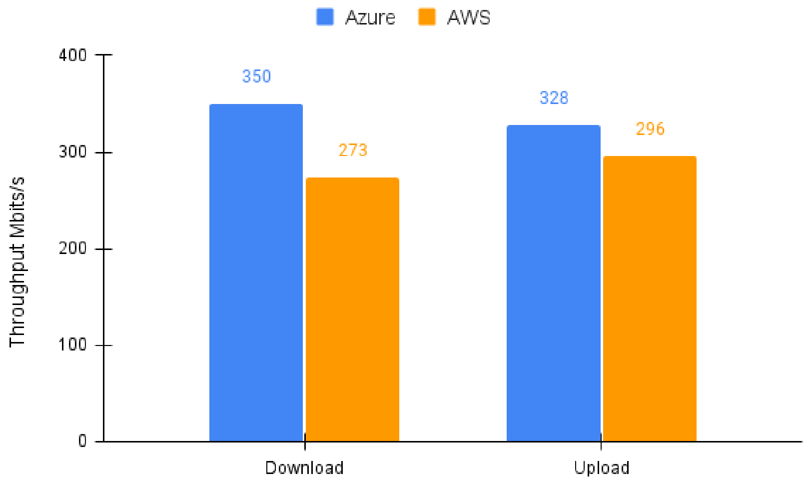

Figure 30: Comparison of throughput to the internet via VPN gateway from Computer B using OpenVPN on AWS and Azure.

As seen in Figure 29 and 30 demonstrate that both Computer A and Computer B achieved a higher download and upload throughput when connected to the Azure OpenVPN VPN server. Computer A had a 4% higher download throughput and a 6.8% higher upload throughput when connected to the Azure OpenVPN VPN server. Computer B had a 28% higher download throughput (equivalent to 77 Mbits/s) and an 8.1% higher upload throughput when connected to the Azure OpenVPN VPN server. These findings indicate that Azure infrastructure outperforms AWS in download and upload throughput when using OpenVPN.

9. Cost and User Complexity Evaluation

This section evaluates the operational costs and user complexity associated with AWS and Azure. Operational costs are assessed based on billing information gathered during experiments, while user complexity is evaluated through our subjective first-hand experiences with platform installation, configuration, and utilization. These key metrics provide valuable insights for readers considering AWS or Azure as their VPN server infrastructure provider.

9.1 Costs

The accumulated operational cost of each cloud platform reflects the duration they were utilized during our experiments, which involved various instance types and experimental variables throughout our thesis research. AWS and Azure utilize a pay-as-you-use system, incorporating factors such as instance types, static public IP addresses, virtual private cloud (VPC), and more into the total monthly cost. In our thesis, the total cost amounted to $5.02 for AWS and $4.57 for Azure, as seen in tables 6 and 7 respectively. However, it is worth noting that we used AWS's twelve-month free tier account, which offers 750 hours/month of free usage for *t3.micro* Linux deployments and 30GB of free storage, each month, a service we used in our thesis. Our Azure account did not have free tier eligibility, therefore the instance and storage costs were included in the total amount stated above. Conversely, the free tier eligibility for AWS impacted the total cost calculation by reducing the expenses. As a result, despite AWS initially appearing to have a slightly higher total cost than Azure, including the free tier account indicates that AWS expenses would likely have been even higher without it.

Table 6: Operational cost breakdown on AWS

Expenses	Amount
Total	$5.26
Virtual Private Cloud	$3.76
Tax	$1.05
EC2 - Compute	$0.45

Table 7: Operational cost breakdown on Azure

Expenses	Amount
Total	$4.57
Virtual Network	$2.94
Storage	$1.56
Virtual Machines	$0.07

The cost evaluation shown above on each platform is dependent on how we utilized each cloud platform, turning on and off the services, and how our tests were done. To understand the costs associated with each cloud platform we have summarized the costs of the cloud platforms' reporting. Tables 8 and 9 show the different costs for each service we utilized in this thesis.

Table 8: Costs for different services on AWS

Specification	Amount
t3.micro	$0.0108 per hour
Storage (gp2)	$0.10 per GB, per month
Static Public IPv4 Address	$0.005 per hour
Data transfer	100GB free each month then $0.09 - 0.05 per GB
Tax	Highly dependable on the Country and Region

Table 9: Costs for different services on Azure

Expenses	Amount
B2ts_v2	$0.0027 per hour
Storage (E4)	$2.40 per month
Static Public IPv4 Address	$0.0036 per hour
Data transfer	100GB free each month then $0.087 - 0.05 per GB
Tax	No direct indication

9.2 User Complexity

During our thesis research, we assessed the user complexity of various cloud platforms. This marked our initial interaction with each platform, offering insights applicable to new users. Firstly, AWS registration proceeded smoothly without any problems during the sign-up process. In contrast, Azure registration presented some challenges. While attempting to finalize our payment method, we encountered an error preventing us from progressing. This was attributed to stringent requirements for inputting billing information. The card and address details had to mirror our bank records precisely. We had to enter street information in uppercase, a discovery that took approximately thirty minutes. To get help with our account creation, we contacted customer support via mail and chatbot, and neither could help us resolve this issue.

Furthermore, as we initiated our first instances on each platform, we faced another problem on Azure. While configuring a new instance, we encountered an unexpectedly high price estimate. The projected cost for our instance amounted to $40 per month, whereas utilizing the Azure Pricing Calculator revealed a much lower estimate of $6 per month [37]. Determining the cause of this difference was challenging, almost leading us to consider switching to Google Cloud. However, upon investigation, we discovered the issue: the "run with Azure Spot discount" option needed to be checked, as illustrated in Figure 31. Azure Spot offers discounted unused Azure capacity, accounting for the significant difference in pricing.

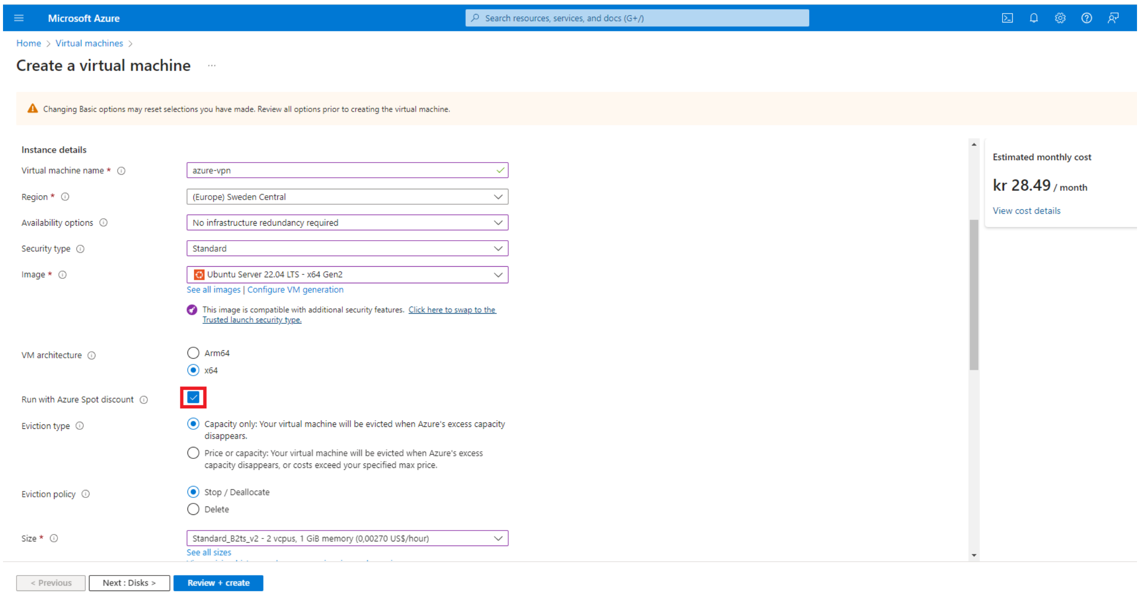

Figure 31: Azure Spot discount checkbox.

These two moments, account creation and instance initialization, originate from a flaw in user design. To assess our experience, we use Jakob Nielsen's 10 heuristics for user interface designs [38]. Nielsen's principles offer a set of rules and guidelines for effective system design, though not all of these are relevant to our evaluation. However, our interaction with Azure during this thesis suggests that principles 8 (Aesthetic and Minimalist Design) and 9 (Help Users Recognize, Diagnose, and Recover from Errors) could use improvement.

Principle 8 emphasizes the importance of minimalist design, ensuring that the interface presents only necessary information without overwhelming users [38]. Azure's interface, across various services, overwhelmed us with information, which concealed crucial elements such as the "run with Azure Spot discount" feature. In contrast, AWS's website is segmented, offering essential information in easy-to-comprehend sections, supporting instance configuration and navigation.

Principle 9 addresses error handling, advocating for clear, actionable error messages [38]. However, Azure's error messages during user account creation lacked specificity, leaving us to guess the issue.

For instance, the error message indicated a problem with billing information without specifying the exact error. In contrast, we did not encounter any account creation errors when setting up our AWS account, indicating they could handle the billing information the way we provided it.

10. Discussion

In this section, we will discuss the result from the experiment of each setup and the cost and user complexity evaluation.

Setup 1 was designed to simulate network traffic between VPN-connected devices. In our experiment, we achieved this by utilizing Computer B as an iPerf server. We designated Computer B as the server since it had a higher network bandwidth and could receive larger loads of data packets. The result we presented, indicates that Azure had a higher throughput for both single TCP streams and multiple TCP streams. The results for the number of retransmissions show that when Azure is used as the VPN server infrastructure, the number of retransmissions is lower. These results indicate that Azure has a better network infrastructure that allows VPN clients and the VPN server to achieve a slightly higher throughput and a lower number of retransmissions. Network congestion could impact the results of the throughput and retransmissions, and indicate why we see some variation between cases.

These findings suggest Azure offers fewer retransmissions and a higher throughput. In scenarios where VPNs are used for file transfer to servers or other types of encrypted communications, Azure emerges as the preferred option due to its network stability and throughput performance.

Setup 2 was designed to evaluate the throughput, jitter, latency, and packet loss, for communication between VPN clients and the VPN server. This experiment is important as it allows us to assess the effectiveness of AWS and Azure network infrastructure. Unlike setup 1 and 3, which involve more interconnected networks, setup 2 focuses on communication between fewer nodes providing an evaluation of the performance differences between the cloud platforms. By only focusing on the communication between the client and the server, we can minimize the influence of external factors and gain insight into the capabilities of AWS and Azure.

The results for setup 2, presented in Section 8.2, show a higher throughput for single and multiple TCP streams for Computer A on Azure. On Azure, Computer A also achieved a lower number of retransmissions. The results for Computer B indicate that with WireGuard there was no significant difference in throughput for single and multiple TCP streams. The results on OpenVPN however displayed a higher throughput for Computer B on Azure. The average number of retransmissions for Computer B for single and multiple TCP streams was lower on Azure.

Based on the single UDP stream traffic test results in Section 8.2, we can see that the throughput for UDP traffic is higher on Azure. However, we can see that jitter fluctuates for both AWS and Azure, the average jitter is higher on Azure. When aiming for a throughput of 10 Mbits/s on OpenVPN AWS exhibited the highest jitter, contrasting with the consistency observed in other tests. Jitter in the range of 0.03 ms is low, and depending on the traffic jitter over 30 ms is considered bad [39]. Variations in the results could be due to network congestion. The average packet loss is relatively low for both AWS and Azure at lower bandwidths, but when higher bandwidth is approached we can see that AWS had a higher packet loss percentage compared to Azure. These findings indicate that Azure's network infrastructure is capable of handling a higher throughput for UDP traffic and less packet loss than AWS but with higher jitter.

Examining Figure 25 it becomes apparent that WireGuard and OpenVPN demonstrate lower latency on the AWS platform. However, as previously mentioned during the presentation of these findings, it's worth considering that Computer B is physically closer to AWS than Azure, which may explain the higher latency observed with Azure.

The results of setup 2 show that a higher throughput for single and multiple TCP streams, and UDP traffic is achieved on Azure. Azure also offers on average a lower number of retransmissions and packet loss. Azure however had a higher jitter and a higher latency compared to AWS. These results indicate that for one-way communications (client-to-VPN server) Azure is preferable because it offers higher throughput, lower retransmissions, and packet loss.

Setup 3 was designed to evaluate the maximum internet throughput for VPN clients when utilizing the VPN server as a gateway, a common and critical scenario in VPN usage. We aimed to assess the capability of both platforms to facilitate internet access via the VPN server. The results depicted in Section 8.3, indicate that Azure's network infrastructure achieved the highest throughput. However, it is important to acknowledge the potential impact of external factors on the results. The performance of the Speedtest server, which we relied on to measure throughput, can vary based on server load and network congestion. The achievable throughput for VPN clients, Computer A and Computer B, may also be influenced by congestion within their respective internet service providers' networks. The results from Setup 3 show that, if the main usage of the VPN is for it to act as a gateway to the Internet, then Azure will provide a higher throughput.

The results from setup 1, 2, and 3 depend on the network performance of several interconnected networks. While each setup is designed to simulate VPN operations, the results are tied to the network performance of Computer A and Computer B's internet service providers. It is important to note that we did not have direct control over the network environment, including the network infrastructure of the cloud platforms. However, this approach provides insights into how real-world network performance and congestion can affect the VPN service and VPN client throughput, latency, jitter, and packet loss. Furthermore, variations in hardware configurations between AWS and Azure could introduce additional factors influencing the observed differences in network performance. For instance, differences in hardware or instruction sets may impact the efficiency of data processing and transmission. Another factor that could make a difference in the results is the VPN configurations. We used installation tools with pre-configured settings, changing different parameters could increase the overall performance for both platforms.

The geographic location of this study is in Sweden, with Computer A and Computer B located in central Sweden. We utilized AWS, located in Stockholm, and Azure, located in Gävle. The proximity of these data centers to our location is a crucial factor in our study. Closer proximity can impact network latency and performance, potentially influencing the results of our experiment. Additionally, geographical location may affect operational costs due to taxes or pricing variations. It is therefore important to consider the geographic location of the VPN server for deployments.

The cost and user complexity are important factors when selecting a cloud platform. In Section 9., we outlined the differences in operational costs incurred during this thesis and presented the advertised cost of the services utilized on each platform. We also highlighted differences in user complexity across each platform. Our evaluation showed that the operational costs were higher on AWS, while Azure had a higher user complexity. Although the complexity for a beginner was higher on Azure, and the account and instance creation took longer, the user only has to encounter these "harder" stepping points once. Once users become familiar with features like "Run with Azure Spot discount", they are less likely to experience similar challenges in the future. While Azure might not be as user-friendly for beginners, we believe that in the long run, it might be more beneficial when considering the costs. It is important to note that the costs and user complexity depend on our work and our subjective experience with each cloud platform. We, however, also know that a more in-depth dive into the cost and user complexity has to be done in the future, to acquire more knowledge.

Analyzing the results from setup 1, 2, and 3, together with the cost and user complexity, it becomes apparent that selecting the most suitable cloud platform for hosting a VPN server involves a complex evaluation. Azure paired with WireGuard would provide the most stable and performant VPN solution for individuals and small companies. Although Azure had a higher user complexity, we think lower operational costs and better network performance can in the long term be better.

11. Conclusion

To answer our first, second, and third research questions we conducted many experiments to evaluate the network performance differences between OpenVPN and WireGuard when running on AWS and Azure. Performance was measured in three different setups: client-to-client communication, client-to-VPN server communication, and utilizing the VPN as a gateway. These experiments let us gather data from each VPN service and cloud platform to see if the cloud platform's infrastructure impacts the performance according to a set of metrics. Most data show OpenVPN and WireGuard achieve higher throughput, lower retransmissions, and packet loss on Azure for TCP and UDP traffic. These results show that Azure allows VPN clients and servers to achieve a higher network performance. When evaluating the client-to-VPN server communication in setup 2 Azure outperformed AWS in almost all the tests. When the VPN server was used as an internet gateway, Azure infrastructure allowed both VPN services to achieve a higher upload and download throughput. It is important to keep in mind that performance can be affected by network congestion, hardware, and distance to the data centers.

We evaluated the operational costs thoroughly and compared the costs for each service and the required components. The total expenses for AWS were about $5.26 for the time we used their services, while for Azure the amount totaled to $4.57. This includes the virtual machine, storage, static IP addresses, and data transfer costs. The difference in operational costs was not that large, but Azure is the cheaper option while offering better network performance overall.

During the setup, installation, and configuration process, we evaluated the user complexity associated with each platform. On Azure, we encountered far more problems with the interface that overwhelmed us with information concealing important elements. Azure also had a problem with account creation and error handling that did not give clear and actionable error messages. These findings show that AWS has a lower user complexity.

In conclusion, our comprehensive evaluation leads us to recommend Azure as the preferred choice for hosting a VPN server. Our findings show Azure overall outperformed AWS in network throughput, with lower retransmissions and packet loss. When paired with WireGuard it is the most performant VPN service.

12. Future Work

In future work, several improvements could enhance the thesis findings. First, one of these improvements would include instances with higher CPU cores, RAM, and network bandwidth. We utilized servers with limited resources because of budget constraints. Incorporating higher-performance instances would enable a more comprehensive analysis of Azure and AWS performance and cost operations. These results could then be utilized by bigger companies, that need more scalability, better network performance, and have the appropriate budget for such infrastructure. Second, another improvement would be to expand the comparison to include additional cloud platforms. While AWS and Azure are industry leaders, other cloud providers may offer competitive performance and pricing for VPN deployment. Lastly, another improvement would be to perform an in-depth analysis of the user complexity according to Nielsen's 10 Usability Heuristics. This could give the cloud providers more insight into what has to be changed to increase the usability of their platform.

References

[1] P. B. Gentry, 'What is a vpn?' *Information Security Technical Report*, vol. 6, no. 1, pp. 15–22, 2001, ISSN: 1363-4127. DOI: https://doi.org/10.1016/S1363-4127(01)00103-0. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1363412701001030.

[2] B. Cruze, *VPN Consumer Usage, Adoption & Shopping Study: 2020*, Security.org, Jan. 2024. [Online]. Available: https://www.security.org/resources/vpn-consumer-report-annual/ (visited on 02/02/2024).

[3] GlobalNewswire, *Virtual Private Network (VPN) Global Market to Reach $137.7 Billion by 2030: Growing Threat of Cyberattacks and Data Breaches Spurs Adoption of VPNs*, GlobeNewswire News Room, Apr. 2023. [Online]. Available: https://www.globenewswire.com/en/news-release/2023/04/10/2643729/28124/en/%20Virtual-Private-Network-VPN-Global-Market-to-Reach-137-7-Billion-by-2030-%20Growing-Threat-of-Cyberattacks-and-Data-Breaches-Spurs-Adoption-of-VPNs.html (visited on 08/02/2024).

[4] Google, *Advantages Of Cloud Computing*, Google Cloud, 2023. [Online]. Available: https://cloud.google.com/learn/advantages-of-cloud-computing (visited on 08/02/2024).

[5] MarkWideResearch, *Cloud VPN Market 2024-2032 | Size,Share, Growth*, MarketWide Reserach. [Online]. Available: https://markwideresearch.com/cloud-vpn-market/ (visited on 08/02/2024).

[6] Norton, 'What is a VPN?,' 2022. [Online]. Available: https://us.norton.com/blog/privacy/what-is-a-vpn (visited on 18/02/2024).

[7] 'What is an eavesdropping attack?' *IPVanish*, 2023. [Online]. Available: https://www.ipvanish.com/blog/eavesdropping-attack/ (visited on 17/05/2024).

[8] M. Klimas, 'What's the difference between symmetric and asymmetric encryption?,' Nov. 2022. [Online]. Available: https://surfshark.com/blog/symmetric-asymmetric-encryption (visited on 14/03/2024).

[9] Lea Hyatt, 'What is OpenVPN and Should You Use It,' 2023. [Online]. Available: https://www.vpnmentor.com/blog/what-is-openvpn-is-it-safe-enough-to-use/ (visited on 18/02/2024).

[10] J. A. Donenfeld, 'Wireguard: Next generation kernel network tunnel.,' in *NDSS*, 2017, pp. 1–12. [Online]. Available: https://www.wireguard.com/papers/wireguard.pdf (visited on 19/03/2024).

[11] Google Clouds, *What is Cloud Computing?* Google. [Online]. Available: https://cloud.google.com/learn/what-is-cloud-computing.

[12] Wikipedia Contributors, *History of cloud computing*, Wikipedia, Jan. 2024. [Online]. Available: https://en.wikipedia.org/wiki/History_of_cloud_computing (visited on 25/02/2024).

[13] Google Clouds, *Advantages and Disadvantages of Cloud Computing*, Google. [Online]. Available: https://cloud.google.com/learn/advantages-of-cloud-computing (visited on 25/02/2024).

[14] Wikipedia Contributors, *Timeline of Amazon Web Services*, Wikipedia, Jul. 2019. [Online]. Available: https://en.wikipedia.org/wiki/Timeline_of_Amazon_Web_Services.

[15] I. A. Rayaprolu, *How Many Companies Use AWS? [AWS Usage Statistics 2024]*, Techjury. [Online]. Available: https://techjury.net/blog/how-many-websites-run-on-aws/ (visited on 18/04/2024).

[16] Wikipedia Contributors, *Microsoft Azure*, Wikipedia, Oct. 2019. [Online]. Available: https://en.wikipedia.org/wiki/Microsoft_Azure.

[17] V. GUEANT, *iPerf - The TCP, UDP and SCTP network bandwidth measurement tool*, Iperf.fr, 2013. [Online]. Available: https://iperf.fr/ (visited on 04/04/2024).

[18] Ookla, *Speedtest® Methodology | Ookla®*, Ookla. [Online]. Available: https://www.ookla.com/resources/guides/speedtest-methodology#speedtest%20-measures-the-full-throughput-capacity-of-an-internet-connection (visited on 04/04/2024).

[19] craigloewen-msft et al., *What is the Windows Subsystem for Linux?* Microsoft Learn, Nov. 2023. [Online]. Available: https://learn.microsoft.com/en-us/windows/wsl/about (visited on 04/03/2024).

[20] V. Persico, P. Marchetta, A. Botta and A. Pescapè, 'Measuring network throughput in the cloud: The case of amazon ec2,' *Computer Networks*, vol. 93, pp. 408–422, 2015, Cloud Networking and Communications II, ISSN: 1389-1286. DOI: https://doi.org/10.1016/j.comnet.2015.09.037. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S138912861500362X.

[21] Osswald, Lukas and Haeberle, Marco and Menth, Michael, 'Performance Comparison of VPN Solutions,' *tobias-lib.ub.uni-tuebingen.de*, May 2020. DOI: 10.15496/publikation-41810. [Online]. Available: https://tobias-lib.ub.uni-tuebingen.de/xmlui/handle/10900/100430 (visited on 12/02/2024).

[22] Mackey, Steven and Mihov, Ivan and Nosenko, Alex and Vega, Francisco and Cheng, Yuan, 'A Performance Comparison of WireGuard and OpenVPN,' in *Proceedings of the Tenth ACM Conference on Data and Application Security and Privacy*, ser. CODASPY '20, New Orleans, LA, USA: Association for Computing Machinery, 2020, pp. 162–164, ISBN: 9781450371070. DOI: 10.1145/3374664.3379532. [Online]. Available: https://doi.org/10.1145/3374664.3379532.

[23] C. H. Chua and S. C. Ng, 'Open-source vpn software: Performance comparison for remote access,' in *Proceedings of the 5th International Conference on Information Science and Systems*, ser. ICISS '22, Beijing, China: Association for Computing Machinery, 2022, pp. 29–34, ISBN: 9781450396837. DOI: 10.1145/3561877.3561882. [Online]. Available: https://doi.org/10.1145/3561877.3561882.

[24] I. Livadariu, K. Benson, A. Elmokashfi, A. Dhamdhere and A. Dainotti, 'Inferring carrier-grade nat deployment in the wild,' in *IEEE INFOCOM 2018-IEEE Conference on Computer Communications*, IEEE, 2018, pp. 2249–2257.

[25] S. A. Jyothi, A. Singla, P. B. Godfrey and A. Kolla, 'Measuring and understanding throughput of network topologies,' pp. 761–772, 2016. DOI: 10.1109/SC.2016.64. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/7877143 (visited on 14/05/2024).

[26] J. F. Kurose and K. W. Ross, *Computer Networking : A Top-Down Approach*, 7th ed. Pearson, 2017.

[27] B. Edgeworth, R. G. Rios, D. Hucaby and J. Gooley, *CCNP and CCIE Enterprise Core ENCOR 350-401 Official Cert Guide*. Cisco Press, Dec. 2019.

[28] A. S. Gillis, *What Is Packet Loss and How Do You Fix It?* TechTarget | Networking, Aug. 2021. [Online]. Available: https://www.techtarget.com/searchnetworking/definition/packet-loss (visited on 17/05/2024).

[29] T. P. Project, *PIVPN: Simplest way to setup a VPN*, www.pivpn.io. [Online]. Available: https://www.pivpn.io/ (visited on 05/04/2024).

[30] Ltd, Raspberry Pi, *Raspberry Pi*, Raspberry Pi. [Online]. Available: https://www.raspberrypi.com/ (visited on 05/04/2024).

[31] J. A. Donenfeld, *Installation*, WireGuard, 2022. [Online]. Available: https://www.wireguard.com/install/ (visited on 16/04/2024).

[32] Pi-hole, *Installation*, Pi-hole documentation, Nov. 2023. [Online]. Available: https://docs.pi-hole.net/guides/vpn/openvpn/installation/ (visited on 16/04/2024).

[33] Nyr, *Nyr/openvpn-install*, GitHub, Apr. 2023. [Online]. Available: https://github.com/Nyr/openvpn-install (visited on 16/04/2024).

[34] OpenVPN, *OpenVPN Connect - Client Software For Windows*, OpenVPN. [Online]. Available: https://openvpn.net/client/client-connect-vpn-for-windows/ (visited on 16/04/2024).

[35] craigloewen-msft, *How to install Linux on Windows with WSL*, Microsoft | Learn, Aug. 2023. [Online]. Available: https://learn.microsoft.com/en-us/windows/wsl/install (visited on 05/04/2024).

[36] Speedtest, *Speedtest CLI: Internet speed test for the command line*, SPEEDTEST® CLI. [Online]. Available: https://www.speedtest.net/apps/cli (visited on 17/05/2024).

[37] Microsoft, *Pricing calculator | Microsoft Azure*, Azure, 2024. [Online]. Available: https://azure.microsoft.com/en-us/pricing/calculator/ (visited on 26/04/2024).

[38] J. Nielsen, *10 Heuristics for User Interface Design*, Nielsen Norman Group, Apr. 1994. [Online]. Available: https://www.nngroup.com/articles/ten-usability-heuristics/ (visited on 24/04/2024).

[39] B. Anjum and H. Perros, 'Bandwidth estimation for video streaming under percentile delay, jitter, and packet loss rate constraints using traces,' *Computer Communications*, vol. 57, pp. 73–84, 2015, ISSN: 0140-3664. DOI: https://doi.org/10.1016/j.comcom.2014.08.018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0140366414003089 (visited on 15/05/2024).

A Appendix Experiment Commands for Each Setup

This appendix shows a table for each experiment for each of the setups. The first table lists the commands used for Setup 1. The iPerf server is initiated on Computer B with the command 'ìperf3 -s´´.

Tool	Description	Command
iPerf3	Single TCP stream from Computer A to Computer B (iPerf server)	iperf3 -c x.x.x.x -t 20 -O 3
iPerf3	Multiple (5) TCP streams from Computer A to Computer B (iPerf server)	iperf3 -c x.x.x.x -t 20 -O 3 -P 5

The following table lists the commands used in Setup 2. The iPerf server is initiated on the VPN server with the command 'ìperf3 -s´´. Change **-b** to target a different bitrate/throughput. *x.x.x.x* is exchanged to the iPerf server VPN interface IP address.

Tool	Description	Command
iPerf3	Single TCP stream from Computer A/B to VPN server	iperf3 -c x.x.x.x -t 20 -O 3
iPerf3	Multiple (5) TCP streams from Computer A/B to VPN server	iperf3 -c x.x.x.x -t 20 -O 3 -P 5
iPerf3	Single UDP stream from Computer B to VPN server	iperf3 -c x.x.x.x -t 20 -O 3 -u -b 10
Ping	Latency/RTT from Computer B to VPN server	ping x.x.x.x -c 50

The following table lists the commands used in Setup 3. *x.x.x.x* is exchanged to the client's VPN interface IP address.

Tool	Command Description	Command
Speedtest	Upload and download throughput test from cloud instance to a targeted Speedtest server	speedtest -s 34024
Speedtest	Upload and download throughput test from Computer A/B to a targeted Speedtest server	speedtest -i x.x.x.x -s 34024