Text

Description automatically generated with medium confidence

A quality-of-service comparison of the network protocols IPv4 compared to IPv6 with the usage of migration technique’s

Mr. Aslam Patel

BSc (Hons) Computer Networks Final Year project report 2022

Supervisor: Faiz Jethwa

# Abstract

With the depletion of available IPv4 address and the introduction of IPv6, coexistence between these protocols in networks is vital. I evaluate the performance of user applications and parameters over the three main transition mechanism, translation, Dual-stack, and tunnelling which are all used in industry today.

A set of experiments have been carried out by creating these 3 transition mechanisms to replicate a small organisation, but not only these three I have also created the same topology for a IPv4 only network which is the least ideal scenario and IPv6 only network which is the most ideal scenario.

The data has been compiled and compared in order to determine the best strategy to the organisation at hand. The findings reveal that the 5 networks produced various amounts of data, with dual stack outperforming the rest, receiving a 4/5 on the final analysis. This was owing to the fact that it had one of the fastest performance times, the least amount of load, and the least amount of jitter.

On certain cases, such as tunnelling, when the load was even lower than Dual stack, performance was higher in other networks. This paper compares five key factors to the five most common networks which also allow for the coexistence of IPv4 and IPv6 in use today within the industry.

# Acknowledgments

First and foremost, I want to thank God Almighty for giving me the opportunity and guidance I need to reach my goal and be successful at this point of my life's journey.

Then I'd want to express my gratitude to my family and friends for their support and unwavering efforts in helping me reach my objectives and aspirations.

I would like to give special thanks to the University of Salford for giving me the opportunity to study and hopefully granting me a bachelor’s degree in the field of Computer Networks.

Many thanks are ought to be for my professors who have assisted in this journey, notably, Dr. Omar Alani who has played major role in guiding me through my whole study programme, alongside all the other professors during my time at this university.

The completion of this report would not have been accomplished without Mr. Faiz Jethwa, my supervisor, who has been a constant source of encouragement and reference throughout my university career, but especially for this project.

And finally, the faculty of computer science and engineering for being instrumental in providing me with the resources I need to accomplish not only this paper but my whole degree programme.

Aslam Adam

April 2022

Contents

[Abstract 2](#_Toc101480632)

[Acknowledgments 3](#_Toc101480633)

[List of Figures 7](#_Toc101480634)

[List of tables 8](#_Toc101480635)

[List of keywords 9](#_Toc101480636)

[1.0 Introduction 12](#_Toc101480637)

[1.1 Motivation for project 12](#_Toc101480638)

[1.2 Aims and Objectives 12](#_Toc101480639)

[1.2 Methodology 13](#_Toc101480640)

[1.3 Project plan 14](#_Toc101480641)

[1.4 Structure of paper 14](#_Toc101480642)

[2.0 Background 15](#_Toc101480643)

[2.1 Internet protocol (IPv4) 15](#_Toc101480644)

[2.1.1 Internet protocol (IPv6) 16](#_Toc101480645)

[2.2 characteristics and benefits of IPv6 16](#_Toc101480646)

[2.3 Comparison of IPv4 vs IPv6 header 17](#_Toc101480647)

[2.3.1 Key difference of headers 18](#_Toc101480648)

[2.4 Transition from IPv4 to IPv6 19](#_Toc101480649)

[2.4.1 Advantages of Migration 19](#_Toc101480650)

[2.5 Dual Stack 20](#_Toc101480651)

[2.5.1 NAT64 21](#_Toc101480652)

[2.5.2 Tunnelling 22](#_Toc101480653)

[2.6 Comparison of 3 chosen Migration Mechanisms 23](#_Toc101480654)

[3.0 Specification and Design 24](#_Toc101480655)

[3.1 Hardware Devices 24](#_Toc101480656)

[3.2 Software and tools 24](#_Toc101480657)

[3.2.1 Cisco Packet Tracer 24](#_Toc101480658)

[3.2.2 GNS3 24](#_Toc101480659)

[3.2.3 GNS3 Virtual Machine 25](#_Toc101480660)

[3.2.4 Wireshark 25](#_Toc101480661)

[3.2.4 IPERF3 25](#_Toc101480662)

[3.2.5 SolarWinds 25](#_Toc101480663)

[3.2.6 Images for GNS3 25](#_Toc101480664)

[3.2.7 Layer 3 Switch 25](#_Toc101480665)

[3.2.8 Layer 2 Switch 26](#_Toc101480666)

[3.2.9 Cisco Router 26](#_Toc101480667)

[3.2.10 Windows PC’s 26](#_Toc101480668)

[3.2.11 Virtual PC’s 26](#_Toc101480669)

[3.2.12 Microsoft Excel 26](#_Toc101480670)

[4.0 Development and Implementation 27](#_Toc101480671)

[4.1 Choice of design 27](#_Toc101480672)

[4.2 IPv4 and IPv6 Networks 27](#_Toc101480673)

[4.4 NAT64 Network 29](#_Toc101480674)

[4.5 Dual Stack Network 31](#_Toc101480675)

[4.6 Tunelling Network 32](#_Toc101480676)

[5.0 Testing and Results 34](#_Toc101480677)

[5.1 IPv4 only network 34](#_Toc101480678)

[5.1.1 Speed and load 34](#_Toc101480679)

[5.1.2 Jitter and packet size 35](#_Toc101480680)

[5.1.3 Delay 36](#_Toc101480681)

[5.2 IPv6 only network 37](#_Toc101480682)

[5.2.1 Speed and load 37](#_Toc101480683)

[5.2.2 Jitter and packet size 37](#_Toc101480684)

[5.2.3 Delay 38](#_Toc101480685)

[5.3 NAT64 network 39](#_Toc101480686)

[5.3.1 Speed and Load 39](#_Toc101480687)

[5.3.2 Jitter and packet Size 41](#_Toc101480688)

[5.3.3 Delay 41](#_Toc101480689)

[5.4 Dual stack network 42](#_Toc101480690)

[5.4.1 Speed and Load 42](#_Toc101480691)

[5.4.2 Delay and Packet loss 43](#_Toc101480692)

[5.4.3 Jitter and Packet size 43](#_Toc101480693)

[5.5 Tunneling network 44](#_Toc101480694)

[5.5.1 Speed and load 44](#_Toc101480695)

[5.5.2 Jitter and packet size 47](#_Toc101480696)

[5.5.3 Delay and Packet loss 48](#_Toc101480697)

[6.0 Comparative Analysis 49](#_Toc101480698)

[6.1 Load 49](#_Toc101480699)

[6.2 Delay 51](#_Toc101480700)

[6.3 Jitter 53](#_Toc101480701)

[6.4 Additional Test/Analysis 55](#_Toc101480702)

[6.4.1 FTP Download Time 55](#_Toc101480703)

[6.4.2 Email download time 56](#_Toc101480704)

[6.5 Final results 57](#_Toc101480705)

[7.0 Conclusion 59](#_Toc101480706)

[References 61](#_Toc101480707)

[Appendices 62](#_Toc101480708)

# List of Figures

[Figure 1 (Number of internet users worldwide 2021 | Statista, 2022) 14](#_Toc101479504)

[Figure 2 (IPv6 & IPv4 Headers - 2022) 15](#_Toc101479505)

[Figure 3 Dual stack Design (Transition Techniques, 2017) 19](#_Toc101479506)

[Figure 4 NAT64 Design (SCHLUTING, 2022) 20](#_Toc101479507)

[Figure 5 Tunnelling Design (SCHLUTING, 2022) 21](#_Toc101479508)

[Figure 6 IPv4 Network created in GNS3 26](#_Toc101479509)

[Figure 7 IPv4 Connection Test 26](#_Toc101479510)

[Figure 8 IPv6 Network created in GNS3 27](#_Toc101479511)

[Figure 9 IPv6 Connection Test 27](#_Toc101479512)

[Figure 10 NAT-64 Network created in GNS3 28](#_Toc101479513)

[Figure 11 NAT-64 Connection Test 1 28](#_Toc101479514)

[Figure 12 NAT-64 Connection Test 2 28](#_Toc101479515)

[Figure 13 NAT-64 Connection Test 3 29](#_Toc101479516)

[Figure 14 NAT-64 Wireshark Translation Capture 29](#_Toc101479517)

[Figure 15 Dual-Stack Network created in GNS3 30](#_Toc101479518)

[Figure 16 Dual-Stack Network Connection Test 30](#_Toc101479519)

[Figure 17 Tunneling Network created in GNS3 31](#_Toc101479520)

[Figure 18 Tunneling Network Connection Test 31](#_Toc101479521)

[Figure 19 CPU load through SolarWinds 38](#_Toc101479522)

[Figure 20 Speed of the link through SolarWinds 39](#_Toc101479523)

[Figure 21 CPU load through SolarWinds 44](#_Toc101479524)

[Figure 22 Speed of the link through SolarWinds 44](#_Toc101479525)

[Figure 23 Total bytes transferred graph through SolarWinds 45](#_Toc101479526)

[Figure 24 Amount of packet transferred and received graph through SolarWinds 45](#_Toc101479527)

[Figure 25 Bar chart of Load comparison 48](#_Toc101479528)

[Figure 26 Line Graph of Load comparison 49](#_Toc101479529)

[Figure 27 Bar chart of Delay comparison 50](#_Toc101479530)

[Figure 28 Line Graph of Delay comparison 51](#_Toc101479531)

[Figure 29 Bar chart of Jitter comparison 52](#_Toc101479532)

[Figure 30 Line Graph of Jitter comparison 53](#_Toc101479533)

[Figure 31 Average FTP Download time in seconds 54](#_Toc101479534)

[Figure 32 Average email download time in seconds 55](#_Toc101479535)

# List of tables

[Table 1 Comparison of IPv4 vs IPv6 19](#_Toc101480601)

[Table 2 Migration Mechanism Comparison ( Huawei, 2021) 23](#_Toc101480602)

[Table 3 IPv4 Speed and Load 35](#_Toc101480603)

[Table 4 IPv6 Jitter and packet size 36](#_Toc101480604)

[Table 5 IPv6 Delay 36](#_Toc101480605)

[Table 6 IPv6 Speed and Load 37](#_Toc101480606)

[Table 7 IPv6 Jitter and packet size 38](#_Toc101480607)

[Table 8 IPv6 Delay 38](#_Toc101480608)

[Table 9 NAT64 Speed and Load 39](#_Toc101480609)

[Table 10 NAT64 Jitter and packet size 41](#_Toc101480610)

[Table 11 NAT64 Delay 42](#_Toc101480611)

[Table 12 Dual stack Speed and Load 42](#_Toc101480612)

[Table 13 Dual Stack Delay 43](#_Toc101480613)

[Table 14 Tunneling Speed and Load 45](#_Toc101480614)

[Table 15 Tunneling Jitter and packet size 48](#_Toc101480615)

[Table 16 Tunelling Delay 48](#_Toc101480616)

[Table 17 comparison of parameters against networks 57](#_Toc101480617)

# List of keywords

IPv4 the fourth version of the Internet Protocol

IPv6 the sixth and most recent Internet Protocol

QoS Quality of Service

EIGRP Enhanced Interior Gateway Routing Protocol

ICMP Internet Control Message Protocol

MTR My traceroute

VM Virtual Machine

VLAN virtual LAN

IPERF3 Internet Performance Working Group

# Introduction

## Motivation for project

The primary motivation roots from the current cyber data age, the increased usage of the Internet over the previous 20 years has demonstrated the Internet's capacity to modify and enhance various fields such as study, trade, and leisure, among others. Along with the rise of the tech craze of the Internet of things (IoT) nobody could have predicted that the internet could develop into a global communication route, and it was understood the quantity of addresses permitted by Internet Protocol v4 (IPv4) would suffice.

IPv4 was first introduced in 1981 and is still commonly used to link multiple networks. After three iterations, IP was given the number 4 and was dubbed IPv4 in the internet universe.

On the Internet, two versions of the Internet Protocol are currently in use. Because the two protocols are not directly compatible, there is some competition between them, forcing network providers and customers to choose between supporting one or the other for certain network services and scenarios.

This paper is a study into the quality-of-service comparison between the network protocols IPv4 compared to IPv6 alongside how this worldwide transition is taking place in industry.

The networks replicate a small organisation in which different parameters have been used to test, analyse, and evaluate the data produced. The goal of this study is to determine the optimum approach for a small organisation to transition from IPv4 to IPv6 with the use of tested parameters.

## 1.2 Aims and Objectives

The primary objectives of this project and paper are as follows.

1. Research

* To compare differences and similarities of IPv4 and IPv6
* To discuss advantages and disadvantages of IPv4 and IPv6
* To look at various migration technologies from IPv4 to IPv6
* To gather information on what sort of companies have which sort of networks.
* To understand how these networks can work together and the different ways to achieve this

1. Design

* Create two networks (one with IPv4 routing and one with IPv6 routing)
* Design 3 migration techniques within the networks i.e., tunnelling, NAT64 and dual stack.

1. Testing

* Experiment with both networks to ensure correct operation
* Usage of analysis tools to assess certain parameters which would be prominent to the scenario suggested

1. Data analysis

* Packet sizes
* Speed
* Delay
* Jitter
* Load

1. Evaluation

* Compile the following four objectives in order to identify the benefits and drawbacks of both networks.
* Comparing what I would conclude before to constructing the network (theory) to what I have after creating the network (Design/Testing).

## Methodology

In relation to the methodology I adopted, there are 3 key things to note. The initial design of the networks, the testing of the networks and the evaluation of the results.

The approach I undertook of designing my networks was split into four key phases, which I simply named, phase 1-4. This was key in allowing me to utilise my time, organisation, and research to the best of my capacity.

Phase 1 consisted of simply creating these networks as a plan within the simulation software cisco packet tracer. This allowed me to create a brief design which was configured and fully functional to then replicate these onto the main software that I used. The networks created in this Phase was IPv4 only, IPv6 only, Tunneling, Nat64 and dual stack networks.

Phase 2 was a similar approach to phase 1 but this time I had built these within the software GNS3. Again, the networks created in this Phase was IPv4 only, IPv6 only, Tunneling, Nat64 and dual stack networks. As GNS3 was a new software to learn, this took up big portion of my allotted time, alongside the usage of images which needed to be embedded within this network, which is discussed further in section 3 of this paper.

Phase 3: I had 5 completely functional networks with connection amongst all devices at this time. The goal now was to utilise an analytical tool to test for the metrics I had set out to achieve, which were speed, load, jitter, delay, in relation to packet size. This analysis was conducted using one internal and one external tool, which will be addressed in further detail in section 3 of this study.

Phase 4, the final stage, was to collect all readings/data from each network that had been built and determine what the optimal scenario would be for the small business in question.

I chose to collate parts of phases 2 and 3. This was accomplished by finishing a network and then gathering the findings for that network; I only completed the first network in this manner before moving on to the next. In the event that the entire objectives of five investigated networks could not be realised, this strategy allowed me to have reading for as many networks as possible.

## Project plan

The project plan which was created in the early period of the project was key in providing a direct path to this paper. With the utilisation of the 5 aims and objectives I set myself at the beginning, this made it key for not only writing this paper but the whole journey of this project. The proposed project plan has been included in the appendix section of this report.

## Structure of paper

This document is organised into six main sections, each of which has been assigned a number.

The first portion of the introduction explained the general purpose of the report as well as the major objectives and goals of the work at hand. The approach chosen has been given, along with a brief explanation of why it was chosen.

Section 2 compares the characteristics, benefits, and headers of the protocols IPv4 and IPv6, as well as their features, benefits, and headers. In addition, the three migration strategies will be discussed, as well as the theory behind them and how they operate. Many articles, websites, and books have been examined and appropriately referenced.

Section 3 discusses the various software and tools that were required to develop the designs and analyse any data that was generated. In its following subsections, the rationale for using certain tools is also discussed.

The created networks, how they were constructed, what devices were utilised, and an overall review of how these networks perform in terms of protocols, routing, and configuration are all covered in Section 4. The networks will be displayed as graphics, with explanations as to why this particular network was constructed.

Section 5 now presents the results of the network testing, as well as the reasons why these tests were necessary for the research. This section will provide graphs and tables illustrating the tools utilised and the data discovered.

Section 6 carries on to the evaluation of these tests and brings together all of the preceding sections to offer the critical review, which I believe is the most important section of this study. The network findings will be compiled and compared against one another in this part, with the goal of determining what would be the best-case scenario for this organisation to adopt.

With the goal of summarising the key ideas of this paper, the last section will end with a quick review of research, building, and test findings.

To establish a clear reference point for all readings, detailed captures of readings will be supplied in the Appendices portion of this work.

The papers, journals, and websites that were used were mentioned in the references section following the Harvard referencing style.

# 2.0 Background

IPV4 was originally designed to allow two computers or other network devices to interact with one another. As the network and internet mechanisms increase and adapt, so does the demand for unique addresses. As a result, technologies like NAT and DHCP have evolved to help ease the current IPV4 address shortage. While these solutions help to ease addressing shortages, they undermine IP level end-to-end protection and decrease resilience, rendering them ineffectual.

To resolve the existing IPV4 problem, a different internet protocol known as Internet Protocol v6 was created (IPv6). This protocol was built by IETF with routing addresses and protection in mind. Among a variety of choices, IPv6, also well-known as IP next generation, was chosen as the most suitable replacement to the existing Internet Protocol (IPV4). In terms of performance, scalability, and security, IPv6 is extra efficient, scalable, secure, and routable than IPV4.

The primary motivation for developing a modern IPv6 is to increase availability of IP address space. IPV4 has a maximum address space of 4.29 billion addresses, whereas IPv6 has a maximum address space of 3.4 x 10^38 addresses. (Patrizio, 2022)

## 

## 2.1 Internet protocol (IPv4)

IPV4 is the fourth version of the Internet Protocol. It was put into service in 1982 on the SATNET and in January 1983 on the ARPANET. In comparison to IPv6, IPV4 is still widely utilised and accounts for about 95% of all internet traffic.

Over 50% of the world's population now has access to the internet. In 1995, it was less than 1%, and in certain cases, it was even less. Figure 1 shows the surge of net users from its birth to 2021.

Chart, bar chart, histogram

Description automatically generated

Figure 1 (Number of internet users worldwide 2021 | Statista, 2022)

## 2.1.1 Internet protocol (IPv6)

IPv6 addresses are 128 bits long, whereas IPV4 addresses are 32 bits long. IPv6 has therefore given a solution to the problem of IP address depletion. IPv6 is taking some time to catch up with the rest of the Internet world. The fundamental reason for this is that IPV4 and IPv6 coexistence is problematic. Transmission techniques are required when an IPv6 host communicates with an IPV4 host. When IPV4 addresses run out, the Internet industry may be obliged to convert to IPv6.

## 2.2 characteristics and benefits of IPv6

1. Because IPv6 decreases the size of routing tables, routing is more effective and categorized. The source device, rather than the router, manages fragmentation in IPv6 networks, using a protocol to establish the path's maximum transmission unit.
2. Unlike IPV4, IPv6 does not contain an IP-level checksum, which eliminates the requirement to renew the checksum at each router hop.
3. IPv6 provides multicast rather than broadcast for directed data flows. Multicast permits traffic-heavy packet flows to be sent to several destinations at the same time, reducing network capacity requirements.
4. IPv6 devices that are connected to other IPv6 devices may auto-configure. Two setup activities that can be automated are IP address assignment and device numbering.
5. delivers privacy, verification, and data reliability

## 

## 2.3 Comparison of IPv4 vs IPv6 header

Even if the number of addresses in IPv6 causes its header to grow in size, the structure of the IPv6 header is simpler than that of IPV4. The IPV4 header is just 20 octets long; nevertheless, the options field variable length increases the entire IPV4 packet size.

IPv6 headers are 40 octets in length. In IPv6, 6 of the 12 IPV4 header fields are removed. By altering and adjusting names, some of the fields of the previous protocol have been replaced. As seen in Figure 2.3, certain additional fields have been introduced to inject new functionality.

Diagram, table

Description automatically generated

Figure 2 (IPv6 & IPv4 Headers - 2022)

Even though the removal allows for faster processing of simpler IPv6 headers, total speed and routing efficiency are reliant on the usage of option headers and the processes that every specific machine would use. Aside that, IPv6 has the benefit of being compatible with current-generation 64-bit CPUs, as IPv6 header fields are 64 bits long.

Some irrelevant data, such as IHL and TOS, as well as fragmentation-related information, are eliminated or transferred to optional extension headers, as shown in Figure 2. The Header Checksum is likewise deleted, leaving the link layer and the transport layer to handle the task. The IPv6 header makes router processing easier.

## 

## 2.3.1 Key difference of headers

|  |  |  |
| --- | --- | --- |
| **Features** | **IPV4** | **IPV6** |
| **Address** | 32 bits | 128 bits |
| **Checksum in header** | Integrated | Checksum not done |
| **Header options** | Essential | Transferred to IPV6 extension headers |
| **Minimum allowed MTU** | 576 bytes | 1280 bytes |
| **Fragmentation** | Carried out by routers and source node | Carried out by source node |
| **Ip configuration** | Manual or DHCP | Auto-configuration or DHCP |
| **IPsec support** | IPsec is not required | IPsec is required |
| **Network access translation (NAT)** | NAT is applied | NAT is not applied |
| **Loopback address** | It uses 127.0.0.1 as loopback address | It uses ::1 as loopback address |
| **Field name** | Use Time to Live as field name | Use Hop Limit as field name |
| **Field header** | Has 12 field inside the header | Has 8 field inside the header |
| **Unknown address** | Use 0.0.0.0.0.0 as unknown address | It uses ::1 as loopback address |
| **Broadcast/multicast** | broadcast for all the host on the network | no broadcast instead it used group of multicasts |
| **Data priority** | No data precedence | It prioritizes data |

Table 1 Comparison of IPv4 vs IPv6

## 2.4 Transition from IPv4 to IPv6

Because IPv6, the most recent edition of the internet protocol, is not backward compatible withIPV4, IPv6 networks are unable to transmit with IPV4 networks. That instance, the IPv6 network can only send IPv6 packets to other IPv6 networks, whereas the IPV4 network can only send packets to other IPV4 networks. As a result, the internet's presence of two protocols causes a huge interoperability problem. (Qaid, 2022)

To resolve the communication issue between IPV4 and IPv6 networks and to enable packet transfer 'Configured tunnelling' or 'explicit tunnelling' is when network managers manually configure the tunnel within the end devices. The IETF and the NGtrans created IPV4 to IPv6 transition methods to alleviate the lack of compatibility problem in encapsulating end devices.

Each transition mechanism must be thoroughly studied and analysed in order to fully grasp the transition mechanisms and their significance. The three primary forms of transition mechanisms are dual-stack, tunnels, and translation techniques. One by one, these strategies are detailed in detail below.

The cost of migration is something that is worth noting but won’t be discussed or tested in this report.

By 2025, it is expected that 60% of users' networks would be IPv6-enabled on average. RTI calculated that the current worth of expenditures for all stakeholder groups to transition to IPv6 will be around $250 billion based on these penetration forecasts. (Tassey, n.d.)

## 2.4.1 Advantages of Migration

IPv6 makes routing more efficient and hierarchical by decreasing the size of the routing table. With the help of ISPs, IPv6 combines the prefixes of various client networks and introduces them as a single common prefix to the IPv6 internet. The procedure is now faster and more efficient. In IPv6 networks, fragmentation is readily avoided since the source device uses a protocol to investigate different MTU paths.( Patrizio, A., 2022)

In IPv6, a packet header is used to speed up packet processing. Unlike IPv4, which uses a header checksum to detect defects in the header of IPv4 packets, IPv6 does not use a header checksum. Because the link-layer technologies and transport layers provide error-control capabilities, there is no need for multiple checksums in diverse places. This saves time and improves packet processing efficiency. (Patrizio, A., 2022)

## 2.5 Dual Stack

Specialists predict to give additional address space and service rising universal connectivity, network infrastructure will migrate from IPV4 to IPv6. Dual stack networks are one of the several IPV4 to IPv6 migration methods that have recently been presented.

A dual stack network is one in which all nodes may communicate in both IPv4 and IPv6 at the same time. Because the router is frequently the first node on a network to receive data from the outside, this is very important.Diagram

Description automatically generated

Figure 3 Dual stack Design (Transition Techniques, 2017)

Because Host is a dual stack device, if it wishes to interact with the server, it must first contact with its local DNS server and request all of example.com's accessible IP addresses. When the DNS server receives a request, it responds with both addresses.

The host choose whether to use IPV4 or IPv6, and in most circumstances, IPv6 is used by default, therefore an IPv6 session is started.

However, if IPv6 does not operate with the server, the client will be delayed. (Yusuf Maitama, 2017)

Because of the IPV6 connection field, the client would attempt to establish a connection using IPV4 again, requiring two visits to the server.

To address the issue of latency and reduce it, Google devised a concept known as 'happy eyes,' in which if an IPv6 connection is not established within 300 milliseconds, it will abort and fall back to an IPV4 connection, saving time. (Design and Comparison Migration between Ipv4 and Ipv6 Transition Techniques, 2017)

## 2.5.1 NAT64

Network address translation (NAT) is a familiar concept within IPV4. NAT is a usually and commonly used method in IPV4, to translate private IPV4 address and a public IPV4 address. NAT 64 provides a translation between ipv6 only networks and IPV4 only networks.

There are three components to NAT64.

* NAT64 prefix can be network specific prefix or a SP or a well know prefix. A NSP is assigned or determined by an organisation.
* DNS64 server is a key component in this process it is DNS server for both ipv6 and IPV4 .
* NAT64 router advertises the NAT64 for prefix into ipv6 only network and it performs the translation between the ipv6 only networks and IPV4 only networks

Diagram

Description automatically generated

Figure 4 NAT64 Design (SCHLUTING, 2022)

In the above example/figure the Host is an IPv6-only device that wishes to connect with the server example.com over an IPV4-only network, which is the most typical use case for NAT64.

To begin, Host requests an IPv6 address for the service example.com from the DNS64 server. If the DNS64 server does not have such a record, it will request one from the IPv6 DNS64 server.

If the IPV4 DNS server on the right-hand side does not respond with the IPV4 address of example.com, it will resort to querying the IPV4 DNS server on the left-hand side.

The address is then transmitted to the IPv6 network's DNS64 server. By prefixing 64:ffb::/96 to the IPV4 address in hexadecimal numerals, the dns64 server creates a NAT64 address for the example.com server. This address is given to host A by the DNS 64 server, and then Host A uses it as a target address.

The NAT64 router then translated the IPV4 and IPV6 headers. The new IPV4 packet is routed to the server after this transformation. The IPV4 packet is then sent to the host, completing the technique.

## 

## 2.5.2 Tunnelling

Tunneling is the technique of encasing one protocol inside another. A packet can be transported by an incompatible network towards the target network via this method.

Because the end points must be set, the most typical kind of tunnelling is router-to-router. These routers must be dual stack and serve as the connection between the IPv6 and IPv4 networks.

Tunnels can be built manually or automatically at end locations. 6to4, ISATAP and 6th mechanisms are examples of automatic tunnelling.

Diagram

Description automatically generated

Figure 5 Tunnelling Design (SCHLUTING, 2022)

In the above network the Host is a IPV4 only device. In order for it to retrieve data from the Server which is a IPV6 only device it must go through a sequence of events like any other migration technique.

The host first creates a packet in which it has the IP address of the server within it. It then inserts this IP packet into an ethernet frame. This ethernet frame is addressed to the CPE router. The host will then put the frame on the ethernet.

When the CPE router takes this frame, it removes the IP packet and it is then inserted into the payload packet. Once the router has done this it forwards the packet onwards towards the host server.

The node at the tunnel's end is in charge of deleting the IPv4 header and forwarding the packet to its final destination.

Regrettably, this is not a long-term solution. There will be no need for tunnelling solutions once dual-stack or native IPv6 is completely adopted. (SCHLUTING, 2022)

## 2.6 Comparison of 3 chosen Migration Mechanisms

|  |  |  |  |
| --- | --- | --- | --- |
| **Transition Mechanism** | **Suitability** | **Limitation** | **Preferability** |
| **Dual-Stack** | Easy to understand, implement, configure, scalable, inexpensive.  Lower packet error-rates.  Wide supportation of devices. | Requires IPv4 address on each network, dual routing, CPU load increased | Best for main stage of migration |
| **Translation** | Enables ISPs to change. | Vulnerable to Dos attacks.  Low scalability  Increase complexity and management costs. | Only for early stages of implementation of migration |
| **Tunelling** | Parameters need to be set but easy to deploy. No need for modifications only at end points. | Costly approach.  Difficult to overcome faults.  Vulnerable to risks in relation to security | Best for early stages of migration. |

Table 2 Migration Mechanism Comparison ( Huawei, 2021)

Out of the table, just the performance analysis will be focused on in this project. Security, cost, and industrial applications will not be examined in great length but will simply be noted.

IPv6 transition has been a matter of debate for years. This section covered three transition mechanisms: dual-stack, tunnelling, and translation.

# 3.0 Specification and Design

This section seeks to introduce the tools that were required and used throughout the project's building. The hardware, software, and virtual images have all been documented, along with the justification for each choice.

## 3.1 Hardware Devices

**PC Disk storage**

Windows 10 OS Minimum 1 GB of free disk space

**Display monitor GPU CPU**

1920x1080 Nvidia GeForce gtx 1050 ti Intel with 64-bit support

**Internet**

Internet connection needed for software download, activation, and tests

## 3.2 Software and tools

## 3.2.1 Cisco Packet Tracer

Cisco Packet Tracer is a Cisco tool, as the name implies. This programme allows you to practise simple and complicated networks using a network simulation. The usage of this application within my project was vital as it provided me with a primer and playground in designing my networks.

The reason for this is that according to research of engineers I came across, “64% of engineers utilise a simpler programme to brainstorm their ideas before they begin construction.” (Sri, 2020) And “Before implementing any protocol, engineers like to test it using Cisco Packet Tracer. Engineers that want to deploy any modification in the production network prefer to utilise Cisco Packet Tracer to test the changes first and then deploy if and only if everything works as planned.” (Sri, 2020)

All of the networks I planned and developed were initially built in Packet Tracer to ensure that all devices, protocols, and processes were recognised to some level.

## 3.2.2 GNS3

Graphical network Simulator-3 is abbreviated as GNS3. GNS3 is another simulation programme that allows you to use Cisco Packet Tracer's capabilities as well as others.

GNS3 was chosen because of its comprehensive capabilities to analyse and visualise packets. Virtual machines have also been introduced, enabling for the usage of external software (which are mentioned below) for packet collection and analysis. (GNS3.com n.d.)

Their programme includes features like connecting your virtual network to real-world networks and collecting data using Wireshark.

## 3.2.3 GNS3 Virtual Machine

The GNS3 VM is a virtual machine that runs GNS3 and connects to it using the GNS3 client installed on my PC. It's a new feature in GNS3 since version 1.4, however the virtual machine isn't really new. Many individuals created their own GNS3 virtual machines, and some of them are now publicly available through initiatives like GNS3 Workbench.

The main usage of this is to reduce load on the PC’s RAM providing a smooth-running experience without system failures.

## 3.2.4 Wireshark

Wireshark is a packet analyser and sniffer. It records network activity on the local network and saves it for examination later.

A lot of information can be deciphered from this tool however I only used this tool in a few scenarios. My analysis, which is presented below, was primarily done with the programme IPERF3. (Wireshark, 2022)

## 3.2.4 IPERF3

This is an acronym for Internet performance working group. I used the third generation (most recent). It's a cross-platform utility that can provide standardised network performance metrics.

It uses to points of connection I to measure multiple metrics which are all outputted with timestamps. (GUEANT, 2022)

## 3.2.5 SolarWinds

Another tool I used to display performance data over time was this one. The tool is essential for displaying collected data as a graph and providing an accurate visual representation of it. Engineers most typically use this tool to measure bandwidth.

Any IT company's foundation is built on its infrastructure; if something goes wrong with its infrastructure, such as servers or routers, it has a negative impact on the company's reputation. If we don't require any impact, we'll need to manually inspect all of the company's manufacturing devices, which will take extra manpower and time. (TroubleSnoop, 2020)

## 3.2.6 Images for GNS3

Multiple images were used for the networks that were created in GNS3. The images were downloaded from the GNS3 recourses page, a page where the administrators and verified users upload this for public use. The use of the images has been discussed in the subsections below.

## 3.2.7 Layer 3 Switch

Simply said, a layer 3 switch is a network switch that also serves as a router. The layer 3 switch's primary function is to accelerate data transfer inside a big LAN. This is also accomplished using the routing function. It can handle many packet forwarding operations as well as one route.

## 3.2.8 Layer 2 Switch

A layer 2 switch is a network switch or device that operates at the data link layer (OSI Layer 2) and uses the MAC address to identify the path through which packets should be sent. It connects and transmits data in a local area network using hardware-based switching methods (LAN).

## 3.2.9 Cisco Router

A router's principal job is to forward packets to their intended destination. This is performed through the employment of a switching function, which is the mechanism by which a router accepts a packet on one interface and forwards it to another.

## 3.2.10 Windows PC’s

The goal of these workstations was to be able to utilise pings to evaluate end-to-end connection inside the network.

## 3.2.11 Virtual PC’s

Within GNS3, a virtual machine PC was also utilised, which included some of the previously described applications. The primary goal of this system was to serve as a server and retrieve raw data using IPERF3 and SolarWinds.

This approach was made because the depth of the data that I wanted for my project could not have been obtained with a standard Windows end computer.

## 3.2.12 Microsoft Excel

This simple software was used to input all readings in a collative table to create both a graph and a bar chart to display the results in comparative form. This was key to see an easy visualisation of the results and how they compare with each other.

All these above listed tools played a major role in the construction and analysis of my network testing.

# 4.0 Development and Implementation

## The goal of this part is to show off the five networks that were constructed, as well as screen captures of the connection test between end devices. Each subsection includes a brief explanation of how each network was put together.

## 4.1 Choice of design

The network architecture was designed to represent two organisational branches connected by an EIGRP network. The objective for this design was to maintain the proposed scenario and focus on getting critical data for the network's performance for numerous factors, at a specific degree of complexity.

There were five separate networks established in all. The major goal was to figure out which migration method would work best for this company's network. There was one IPv4-only network, one IPv6-only network, and three migration technology networks built which are listed in the subsections to come.

The IPv4-only and IPv6 only networks were both created to use a sample to refer back for the 3 migration networks to see how all compare to the basic 2 networks. This was vital when reading needed to be analyzed and conclude which scenario is the best for this small organization.

## 4.2 IPv4 and IPv6 Networks

The topologies in figure 5.1 and in figure 5.2 depicts two organizational branches connected by the EIGRP network. The traffic is routed through the routers that connect the branches. The network's redundancy is provided through several links connecting the branches.

In the event that the primary connection fails, traffic will be routed through the backup link with the least amount of downtime possible.

Each branch is connected to the multilayer switch via two separate VLANs. In both branches, the inter-vlan routing is provided by the multilayer switch.

A layer 3 link exists between the multilayer switch and the router. In both branches, EIGRP is also in use between the switch and the routers.

Both IPv4 and IPv6 protocols have utilized the same topology.

Chart, scatter chart

Description automatically generated

Figure 6 IPv4 Network created in GNS3

EIGRP has been configured successfully and the connectivity between the branches has been tested successfully.

Table

Description automatically generated

Figure 7 IPv4 Connection Test

Chart, scatter chart

Description automatically generated

Figure 8 IPv6 Network created in GNS3

EIGRP has been configured successfully and the connectivity between the branches has been tested successfully.

Text

Description automatically generated

Figure 9 IPv6 Connection Test

## 4.4 NAT64 Network

Again, GNS3 simulation software is used to create the topology.

Different versions of IP addresses are used by the two branches. Branch1 solely uses IPv6 addresses, whereas Branch2 only uses IPv4 addresses.

The communication between the branches is provided by the routers in the middle. The NAT64 routers B1-R1 and B1-R2 are in use. They are in the process of translating the addresses.

In this configuration, B1-R2 serves as a backup route.

In the event that the primary route to Branch 1 through Router B1-R1 fails, the alternative route will be employed. Multiple connections were employed to create redundancy in the network.

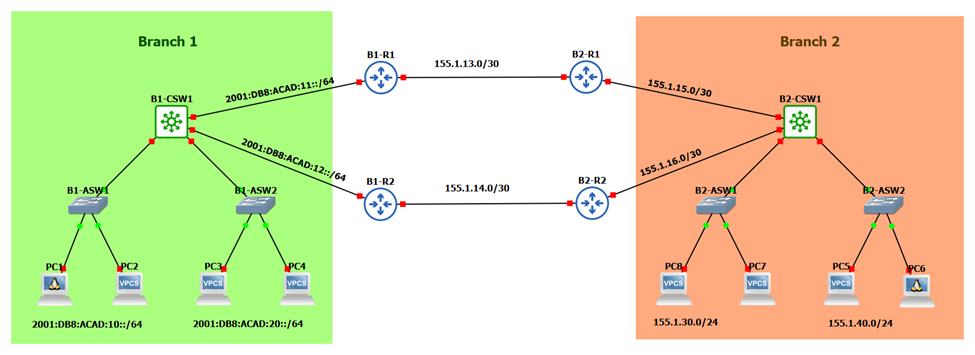


Figure 10 NAT-64 Network created in GNS3

The two branches have been successfully configured and connectivity has been tested. The Branch1 is using IPv6 addresses and Branch2 is using IPv4 addresses. In the folloiwng example the 50.50.1.2 IPv4 address is getting tanslated in to 2001:db8:café:1:1::2 and is pinging the PC3 with an IPv6 address 2001:DB8:ACAD:20::13.

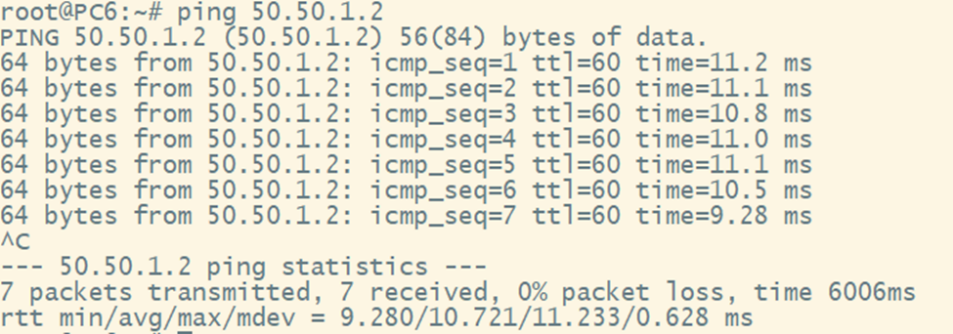


Figure 11 NAT-64 Connection Test 1

In the following exhibit we are pinging from PC6 having an IPv4 address to PC1 having IPv6 address.

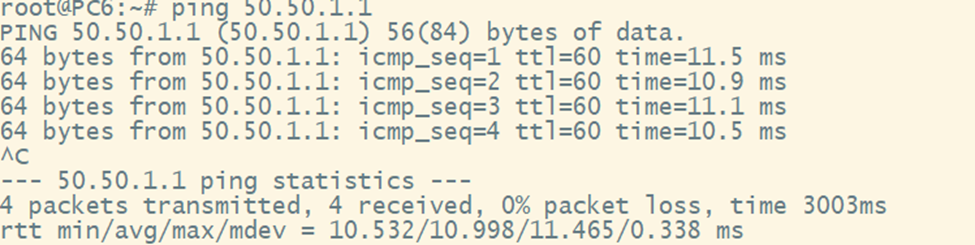


Figure 12 NAT-64 Connection Test 2

The following can be confirmed by looking at the NAT64 translations on the NAT64 Router (B1-R1).



Figure 13 NAT-64 Connection Test 3

It can also be verified by capturing the packet. In the following exhibit I have captured the packet on the router B1-R1 and analysed using the Wireshark software. It can be seen from the captured packet that the NAT64 router (B1-R1) is translating the IPv4 addresses into IPv6 address.



Figure 14 NAT-64 Wireshark Translation Capture

## 4.5 Dual Stack Network

The network shown in figure 15 again was created inside GNS3. I configured branch 1 with IPv4 addresses and branch 2 with IPv6 addresses. Again, like all other networks created there is a backup route, in this case from Router 2 in case of a link failure.

All routers used in this topology are dual stack routers, allowing for connection from both IPv4 and IPv6 addresses.

This network required more CPU power than all the of the other networks created due to the main reason being all routers had to the stacks at the same time.

A picture containing text, boat, different

Description automatically generated

Figure 15 Dual-Stack Network created in GNS3

The two branches have been successfully configured and connectivity has been tested. Both the branches are using IPv6 addresses. In the folloiwng screenshot I have performed the connectivity test between the branches using PC6 and PC1. I have used the ping utility to check the reachability of PC6 to PC1. It was successful as it can be seen from the screenshot below.

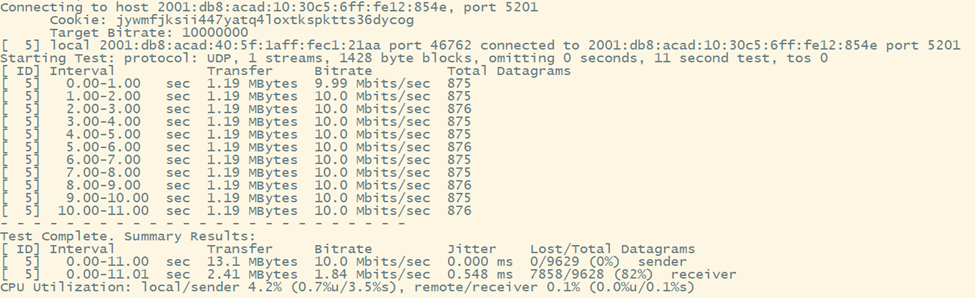


Figure 16 Dual-Stack Network Connection Test

## 4.6 Tunelling Network

Like the other networks GNS3 simulation software is used to create the topology shown in figure 17. I configured Branch 1 and Branch 2 with IPv6 addresses in this topology, however they will communicate via IPv4 infrastructure.

It signifies that the IPv4 network is used to connect these two branches. As we make the transition from version 4 to version 6, this is the most prevalent scenario.

The infrastructure of many ISPs is still based on IPv4. Between them, the OSPF protocol is used for IPv4 communication, and I use EIGRP for IPv6 connectivity. The GRE is being utilised within this tunnel.

In this situation, a tunnel is built between the two branches, allowing IPv6 packets to be sent via an IPv4 network. The tunnel's IPv6IP mode is used.

Chart

Description automatically generated

Figure 17 Tunneling Network created in GNS3

The two branches have been successfully configured and connectivity has been tested. Both the branches are using IPv6 addresses. In the folloiwng screenshot I have performed the connectivity test between the branches using PC6 and PC1. I have used the ping utility to check the reachability of PC6 to PC1. It was successful as it can be seen from the screenshot below.

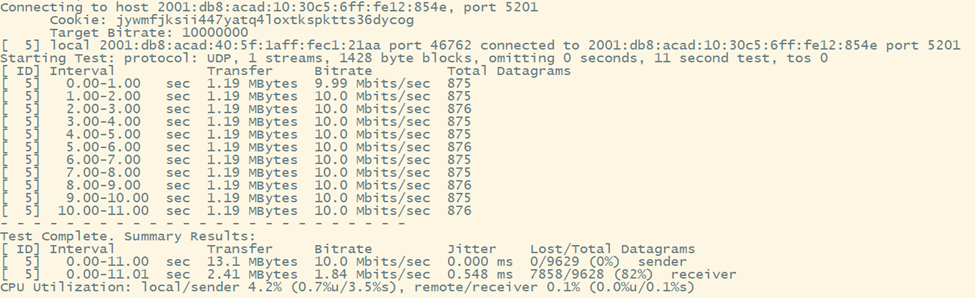


Figure 18 Tunneling Network Connection Test

The network architecture in all of the 5 networks was designed to represent two organisational branches connected by an EIGRP or OSPF network.

The objective for this design was to maintain the proposed scenario across all networks and focus on getting critical data for the network's performance for numerous factors, at a specific degree of complexity.

There were five separate networks established in all. The major goal was to figure out which migration method would work best for this company's network. There was one IPv4-only network, one IPv6-only network, and three migration technology networks built which are listed in the subsections above.

Now as all the networks were established as similar as possible this ensured the testing for the chosen parameters would be fairer across all 5 networks.

# Testing and Results

IPERF3 was the major tool I used to test my networks. IPERF3 is an open-source network performance monitoring tool, as indicated in section 3 of this study. It simply transfers data from one server to another while calculating available bandwidth. It can offer data on packet size, jitter, delay, and speed load, among other things.

IPERF had to be installed virtually within each network to allow the PCs on the network to test the network performance. One device in a network serves as a server, while another serves as a client.

Wireshark again was utilised to capture packets to see any data change about that packet. For example, in the NAT64 network the packet translation between the routers is captured.

The load of the device is also being captured through a network management software SolarWinds. This a tool like mentioned in section 3 that can provide graphical charts to the change in test for example load on a CPU over 30 minutes.

This part will not feature a full analysis; instead, it will focus on the tests that have been conducted and their outcomes; section 6 will go over this in further depth.

## IPv4 only network

### 5.1.1 Speed and load

The network speed is measured in the topology by transferring varying amounts of data between the source and destination across the EIGRP network. I began by sending a little quantity of data through the network and measuring the amount of speed consumed as well as the CPU load.

I steadily raised the amount of data while also keeping an eye on the bandwidth. For the final three tries, the data was sent at a speed of 100 Mbps. The more data was delivered, the more the load was monitored.

It has been found that as the amount of sent data grows, bandwidth utilisation grows along with it, and the CPU load grows as well. For various time periods, different amounts of data were sent. The data transfer time must be at least 10 seconds. I raised the data transfer duration to 20 seconds to see how it affected the bandwidth and CPU load.

Table 3 shows the readings of the speed and load that were observed and recorded.

|  |  |  |  |
| --- | --- | --- | --- |
| **Speed & Load** | | | |
| **Attempts** | **Speed (Mbps)** | **Data Transfer (MB)** | **Load (%)** |
| 1 | 1.05 | 1.38 | 2.4 |
| 2 | 10.00 | 13.1 | 3.90 |
| 3 | 50.00 | 71.5 | 9.40 |
| 4 | 60.00 | 93 | 8.80 |
| 5 | 70.00 | 117 | 9.80 |
| 6 | 80.00 | 134 | 9.90 |
| 7 | 90.00 | 129 | 10.30 |
| 8 | 100.00 | 179 | 12.00 |
| 9 | 100.00 | 191 | 10.60 |
| 10 | 100.00 | 203 | 11.20 |

Table 3 IPv4 Speed and Load

The measurements were acquired with the iperf3 programme and a variety of settings. The screenshots may be seen in Appendix A-1.

### 5.1.2 Jitter and packet size

The iperf3 programme was used to monitor and record the network's jitter. The jitter on the network has been observed to be constant.

Over the course of the ten trials, there isn't much variation in the jitter. Throughout the jitter experiment, several packet sizes are employed. In order to acquire a clearer understanding of the jitter variation, the data between the branches was transferred during different time periods.

The data was transmitted for a maximum of 20 seconds with a packet size of 264 KB and a minimum of 10 seconds with a packet size of 128 KB.

The UDP protocol was utilised to improve readability. The jitter was calculated using the UDP protocol since TCP has the lowest jitter on the network because it is connection orientated.

Table 4 shows the jitter and packet size readings.

|  |  |  |
| --- | --- | --- |
| **Jitter & Packet Size** | | |
| **Attempts** | **Jitter (ms)** | **Packet Size (KB)** |
| 1 | 25.762 | 128 |
| 2 | 26.277 | 128 |
| 3 | 24.751 | 128 |
| 4 | 31.349 | 128 |
| 5 | 18.575 | 128 |
| 6 | 35.801 | 256 |
| 7 | 37.12 | 264 |
| 8 | 36.059 | 128 |
| 9 | 46.424 | 128 |
| 10 | 20.964 | 128 |

Table 4 IPv6 Jitter and packet size

The measurements were acquired with the iperf3 programme and a variety of settings. The screenshots may be seen in Appendix A-2.

### 5.1.3 Delay

The ICMP protocol is used to calculate the delay in the network.

There are two ways to calculate the delay, we can use the traceroute utility to find the path between the source and the destination and then add all the times of different hops between source and destination.

The second option is to use the ping utility and find the time variance between the echo request and echo reply.

I have used the ping utility in my network to calculate the delay. Same number of attempts have been conducted to calculate the delay as well.

The number of attempts has been same throughout the project to calculate different network parameters.

It has been observed that the delay is also constant within the network. The readings of the delay recorded is provided in the below table.

|  |  |  |  |
| --- | --- | --- | --- |
| **Attempts** | **Delay** | **Attempts** | **Delay** |
| 1 | 12.29 | 6 | 12.3474 |
| 2 | 11.8588 | 7 | 11.5504 |
| 3 | 11.749 | 8 | 11.7684 |
| 4 | 11.965 | 9 | 11.857 |
| 5 | 11.7494 | 10 | 11.481 |

Table 5 IPv6 Delay

The measurements were acquired with the iperf3 programme and a variety of settings. The screenshots may be seen in Appendix A-3.

## IPv6 only network

## Speed and load

The same scenarios that were used to assess network performance and load for IPv4 have been applied to IPv6. Again, different speeds were employed to deliver varied amounts of traffic on the network between the two branches via the EIGRP network.

I started at 1 Mbps and worked my way up to 100 Mbps over the course of ten attempts. When the amount of data is increased at a faster rate, the load is somewhat higher than when using IPv4.

In the final three attempts, I transferred the data at a speed of 100 Mbps, and the load increased, but remained steady, according to the measurements.

Table 6 shows the speed and load readings that were observed and recorded using IPv6.

|  |  |  |  |
| --- | --- | --- | --- |
| **SPEED & LOAD** | | | |
| **Attempts** | **Speed (Mbps)** | **Data Transfer (MB)** | **Load (%)** |
| 1 | 1.05 | 1.25 | 2.1% |
| 2 | 10.00 | 11.9 | 6.4% |
| 3 | 50.00 | 59.6 | 9.5% |
| 4 | 60.00 | 71.5 | 7.0% |
| 5 | 70.00 | 83.4 | 10.2% |
| 6 | 80.00 | 95.4 | 10.4% |
| 7 | 90.00 | 107 | 10.9% |
| 8 | 100.00 | 119 | 10.9% |
| 9 | 100.00 | 143 | 10.9% |
| 10 | 100.00 | 155 | 11.8% |

Table 6 IPv6 Speed and Load

The measurements were acquired with the iperf3 programme and a variety of settings. The screenshots may be seen in Appendix B-1.

## Jitter and packet size

Again, the iperf3 program was used to record the jitter in the IPv6 network. The data was delivered over a longer period of time. The packet size has also been tweaked to gain a clearer understanding of the network's jitter.

The jitter is noted to be continuous throughout the experiment. The testing was carried out for a total of ten times. The average jitter across all tries is 2.203 milliseconds.

Various packet sizes are tried, with the average packet size being 264 KB throughout the 10 tries. For testing purposes, the UDP protocol is utilized.

The jitter and packet size readings are provided in the below table 7.

|  |  |  |
| --- | --- | --- |
| **Attempts** | **Jitter (ms)** | **Packet Size (KB)** |
| 1 | 1.108 | 128 |
| 2 | 2.923 | 244 |
| 3 | 2.665 | 305 |
| 4 | 2.278 | 305 |
| 5 | 1.864 | 184 |
| 6 | 2.960 | 367 |
| 7 | 0.566 | 128 |
| 8 | 2.382 | 428 |
| 9 | 2.371 | 184 |
| 10 | 2.913 | 367 |

Table 7 IPv6 Jitter and packet size

The measurements were acquired with the iperf3 programme and a variety of settings. The screenshots may be seen in Appendix B-2.

## 5.2.3 Delay

Table 8 IPv6 Delay

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Attempts** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** |
| **Delay (ms)** | 15.1 | 10.9 | 11.4 | 12.0 | 11.7 | 11.4 | 11.2 | 12.1 | 11.5 | 11.8 |
| **Packet Loss (%)** | 0% | 0% | 0% | 0% | 0% | 1.76% | 0.56% | 0% | 3.66% | 1.94% |
| **No. of Packets** | 10 | 50 | 100 | 150 | 200 | 250 | 300 | 350 | 400 | 450 |

The mtr utility has been used to record the network latency. The delay is measured across a series of attempts, with the number of packets increasing each time and the time length being maintained to a minimum to capture the network's performance during times of network congestion.

It has been discovered that if the network is appropriately set, the network's latency and congestion may be reduced. In the initial attempt, I sent 10 packets and escalated to 450 packets in the last attempt.

The average latency across all tries is 11.9 milliseconds, with an average packet loss of 1% in the event of network congestion. The recorded delay readings are listed in table 8 above.

The measurements were acquired with the iperf3 programme and a variety of settings. The screenshots may be seen in Appendix B-3.

## NAT64 network

## 5.3.1 Speed and Load

The network's speed is measured by transferring varying amounts of data between the two branches. I began by sending a little quantity of data through the network and measuring the amount of speed consumed as well as the CPU load.

I steadily raised the amount of data while also keeping an eye on the bandwidth. For the final three tries, the data was sent at a speed of 100 Mbps. The more data was delivered, the more the load was monitored.

It has been observed that as I have increased the speed and amount of data transferred between the branches the CPU load keeps on increasing. I have sent the data for the maximum of 100 Mbps bandwidth for maximum of 20 second’s period.

The reading of the speed and load monitored and captured can be seen in the table below.

|  |  |  |  |
| --- | --- | --- | --- |
| **SPEED & LOAD** | | | |
| **Attempts** | **Speed (Mbps)** | **Data Transfer (MB)** | **Load (%)** |
| 1 | 1.05 | 1.25 | 3% |
| 2 | 10.00 | 11.9 | 3.5% |
| 3 | 50.00 | 59.6 | 11.7% |
| 4 | 60.00 | 71.5 | 11.1% |
| 5 | 70.00 | 83.4 | 12.8% |
| 6 | 80.00 | 95.4 | 13.1% |
| 7 | 90.00 | 107 | 9.5% |
| 8 | 100.00 | 119 | 17.0% |
| 9 | 100.00 | 119 | 10.9% |
| 10 | 100.00 | 119 | 10.8% |

Table 9 NAT64 Speed and Load

The measurements were acquired with the iperf3 programme and a variety of settings. The screenshots may be seen in Appendix C-1.

The load of the device is also being captured through a network management software SolarWinds.

The traffic was sent for 30 minutes using different speeds and different amount of data. I started sending the data from 10 Mbps and increased it to 150 Mbps speed and also increased the amount of data sent at a specific time.

As it can be seen from the graph below the CPU load increased as the traffic on the link increases. At 10 Mbps the load of the CPU is recorded at 16% and at the peak the load is recorded at 23%.

Graphical user interface, text, application, email

Description automatically generated

Figure 19 CPU load through SolarWinds

*Chart, line chart

Description automatically generated*

Figure 20 Speed of the link through SolarWinds

## 5.3.2 Jitter and packet Size

The jitter on the network has been observed to be steady and minimal. Over the course of the ten trials, there isn't much variation in the jitter. Throughout the jitter experiment, several packet sizes are employed.

In order to acquire a clearer understanding of the jitter variation, the data between the branches was transferred during different time periods. The data was transferred for a maximum of 20 seconds, with a packet size of 128 KB and a minimum of 10 seconds.

With the following readings it can be concluded that the performance of the network is good, and network is stable.

The jitter and packet size readings are provided in table 10.

|  |  |  |
| --- | --- | --- |
| **Jitter & Packet Size** | | |
| **Attempts** | **Jitter (ms)** | **Packet Size (KB)** |
| 1 | 0.323 | 128 |
| 2 | 0.363 | 128 |
| 3 | 0.467 | 128 |
| 4 | 0.327 | 128 |
| 5 | 0.581 | 128 |
| 6 | 0.329 | 128 |
| 7 | 0.427 | 128 |
| 8 | 0.409 | 128 |
| 9 | 0.476 | 128 |
| 10 | 0.454 | 128 |

Table 10 NAT64 Jitter and packet size

The measurements were acquired with the iperf3 programme and a variety of settings. The screenshots may be seen in Appendix C-2.

## 5.3.3 Delay

I have used the ping utility to calculate the delay in the network. The ping uses the ICMP protocol. The delay for the network has been recorded for 10 attempts.

In every attempt five echo requests were sent, and equal number of echo reply were received.

It has been observed that the packet loss is 0% in this network and delay is also not very high.

Different number of packets were sent in each attempt to monitor the packet loss. The delay in the network is stable and does not increase substantially.

Table 11 below displays the results.

|  |  |  |  |
| --- | --- | --- | --- |
| **Attempts** | **Delay** | **Attempts** | **Delay** |
| **1** | 11.135 | **6** | 10.726 |
| **2** | 11.037 | **7** | 11.076 |
| **3** | 11.011 | **8** | 10.982 |
| **4** | 9.975 | **9** | 12.713 |
| **5** | 11.017 | **10** | 10.735 |

Table 11 NAT64 Delay

The measurements were acquired with the iperf3 programme and a variety of settings. The screenshots may be seen in Appendix C-3.

## Dual stack network

## Speed and Load

The speed of the network is recorded using different amount of data transfer between the two branches.

I began by sending a little quantity of data through the network and measuring the amount of speed consumed as well as the CPU load.

I steadily raised the amount of data while also keeping an eye on the bandwidth. For the final three tries, the data was sent at a speed of 100 Mbps. The more data was delivered, the more the load was monitored.

It has been observed that as I have increased the speed and amount of data transferred between the branches the CPU load keeps on increasing.

I have sent the data for the maximum of 100 Mbps bandwidth for maximum of 20 second’s period.

The reading of the speed and load monitored and captured can be seen in table 12.

|  |  |  |  |
| --- | --- | --- | --- |
| **SPEED & LOAD** | | | |
| **Attempts** | **Speed (Mbps)** | **Data Transfer (MB)** | **Load (%)** |
| 1 | 1.05 | 1.25 | 3.2% |
| 2 | 10.00 | 11.9 | 3.5% |
| 3 | 50.00 | 59.6 | 11.7% |
| 4 | 60.00 | 71.5 | 11.1% |
| 5 | 70.00 | 83.4 | 12.8% |
| 6 | 80.00 | 95.4 | 13.1% |
| 7 | 90.00 | 107 | 9.5% |
| 8 | 100.00 | 119 | 17.0% |
| 9 | 100.00 | 119 | 10.9% |
| 10 | 100.00 | 119 | 10.8% |

Table 12 Dual stack Speed and Load

## Delay and Packet loss

I have used the ping utility to calculate the delay in the network. The ping uses the ICMP protocol. The delay for the network has been recorded for 10 attempts. In every attempt five echo requests were sent, and equal number of echo reply were received.

It has been observed that the packet loss is 0% in this network and delay is also not very high.

Different number of packets were sent in each attempt to monitor the packet loss. The delay in the network is stable and does not increase substantially.

Table 13 below displays the results.

|  |  |  |  |
| --- | --- | --- | --- |
| **Attempts** | **Delay** | **Attempts** | **Delay** |
| 1 | 13.854 | 6 | 12.987 |
| 2 | 13.698 | 7 | 13.245 |
| 3 | 12.968 | 8 | 13.265 |
| 4 | 12.854 | 9 | 13.148 |
| 5 | 13.102 | 10 | 12.986 |

Table 13 Dual Stack Delay

## Jitter and Packet size

The jitter on the network has been observed to be steady and minimal. Over the course of the ten trials, there isn't much variation in the jitter. Throughout the jitter experiment, several packet sizes are employed.

In order to acquire a clearer understanding of the jitter variation, the data between the branches was transferred during different time periods. The data was transferred for a maximum of 20 seconds, with a packet size of 128 KB and a minimum of 10 seconds.

With the following readings it can be concluded that the performance of the network is good, and network is stable.

The jitter and packet size readings are provided in table 14.

|  |  |  |
| --- | --- | --- |
| **Jitter & Packet Size** | | |
| **Attempts** | **Jitter (ms)** | **Packet Size (KB)** |
| 1 | 0.456 | 128 |
| 2 | 0.485 | 128 |
| 3 | 0.489 | 128 |
| 4 | 0.512 | 128 |
| 5 | 0.497 | 128 |
| 6 | 0.397 | 128 |
| 7 | 0.468 | 128 |
| 8 | 0.512 | 128 |
| 9 | 0.478 | 128 |
| 10 | 0.489 | 128 |

Table 14 Dual stack Jitter and packet size

## 5.5 Tunneling network

## 5.5.1 Speed and load

The speed of the network is recorded using different amount of data transfer between the two branches. I began by sending a little quantity of data through the network and measuring the amount of speed consumed as well as the CPU load.

I steadily raised the amount of data while also keeping an eye on the bandwidth. For the final three tries, the data was sent at a speed of 100 Mbps. The more data was delivered, the more the load was monitored..

It has been observed that as compared to the NAT64 more amount of data has been transferred using the same parameters. As the speed of the link has increased the amount of data transferred also increased and the load on the CPU increases as well.

But the CPU performance is monitored to be stable, and the load has not increased very significantly as compared to NAT64.

The reading of the speed and load monitored and captured can be seen in table 15 below.

|  |  |  |  |
| --- | --- | --- | --- |
| **SPEED & LOAD** | | | |
| **Attempts** | **Speed (Mbps)** | **Data Transfer (MB)** | **Load (%)** |
| 1 | 10 | 23.8 | 7.8% |
| 2 | 30.00 | 71.5 | 7.4% |
| 3 | 50.00 | 119 | 8.4% |
| 4 | 60.00 | 143 | 11.2% |
| 5 | 70.00 | 167 | 11.4% |
| 6 | 80.00 | 191 | 14.3% |
| 7 | 90.00 | 215 | 11.7% |
| 8 | 100.00 | 238 | 12.1% |
| 9 | 100.00 | 238 | 16.4% |
| 10 | 100.00 | 238 | 12.9% |

Table 15 Tunneling Speed and Load

The measurements were acquired with the iperf3 programme and a variety of settings. The screenshots may be seen in Appendix D-1.

SolarWinds, a network management programme, is also used to monitor the device's load. For 30 minutes, traffic was transmitted at various speeds and with varying amounts of data. I started delivering data at 10 Mbps and gradually raised the speed to 150 Mbps while simultaneously increasing the quantity of data delivered at a given time.

The CPU load rose as the traffic on the link increased, as shown in the graph below. At 10 Mbps, the CPU load is 16 percent, and at the peak, the load is 37 percent. In addition, SolarWinds has been used to record and collect link activity.

The graphs below illustrate how much bandwidth is used and how much data is transported.

Graphical user interface, text, application, email

Description automatically generated

Figure 21 CPU load through SolarWinds

*Chart, line chart

Description automatically generated*

Figure 22 Speed of the link through SolarWinds

*Chart, line chart

Description automatically generated*

Figure 23 Total bytes transferred graph through SolarWinds

Chart, line chart

Description automatically generated

Figure 24 Amount of packet transferred and received graph through SolarWinds

## 5.5.2 Jitter and packet size

The iperf3 programme was used to monitor and record the network's jitter. When tunnelling, the jitter on the network is somewhat larger than when using NAT64, according to tests. Over the course of the ten trials, there isn't much variation in the jitter.

The tunnel contributes to the rise in jitter since traffic passes through it on its way to its destination. It has also been discovered that even with a greater number of packets sent, the loss rate remains at 0%. Throughout the jitter experiment, several packet sizes are employed. In order to acquire a clearer understanding of the jitter variation, the data between the branches was transferred during different time periods.

The data has been sent for 20 seconds maximum with the maximum packet size of 128 KB and minimum of 10 seconds. With the following readings it can be concluded that the performance of the network is good, and network is stable.

The jitter and packet size readings are provided in below table 15.

|  |  |  |
| --- | --- | --- |
| **Jitter & Packet Size** | | |
| **Attempts** | **Jitter (ms)** | **Packet Size (KB)** |
| 1 | 1.754 | 128 |
| 2 | 1.503 | 128 |
| 3 | 2.302 | 128 |
| 4 | 1.486 | 128 |
| 5 | 1.182 | 128 |
| 6 | 1.316 | 128 |
| 7 | 1.563 | 128 |
| 8 | 1.386 | 128 |
| 9 | 0.868 | 128 |
| 10 | 1.728 | 128 |

Table 15 Tunneling Jitter and packet size

The measurements were acquired with the iperf3 programme and a variety of settings. The screenshots may be seen in Appendix D-2.

## 5.5.3 Delay and Packet loss

I have used the ping utility to calculate the delay in the network. The ping uses the ICMP protocol. The delay for the network has been recorded for 10 attempts. In every attempt five echo requests were sent, and equal number of echo reply were received.

It has been observed that the packet loss is 0% in this network and delay is also not very high. The delay in the network is less compared to the delay recorded in the NAT64.

Different number of packets were sent in each attempt to monitor the packet loss. The delay in the network is stable and does not increase substantially.

The readings are shown in table 16.

|  |  |  |  |
| --- | --- | --- | --- |
| **Attempts** | **Delay** | **Attempts** | **Delay** |
| 1 | 8.845 | 6 | 6.961 |
| 2 | 8.030 | 7 | 6.373 |
| 3 | 8.335 | 8 | 6.913 |
| 4 | 6.848 | 9 | 7.662 |
| 5 | 7.465 | 10 | 6.790 |

Table 16 Tunelling Delay

The measurements were acquired with the iperf3 programme and a variety of settings. The screenshots may be seen in Appendix D-3.

# 6.0 Comparative Analysis

This section brings together all of the results from testing in section 5 and provides the results in collative graph form.

## 6.1 Load

The speed of the network is recorded using different amount of data transfer between the two branches for all networks. I started with sending small amount of data across the network and monitored the load it takes on the CPU. This is shown in figure 25.

Figure 25 Bar chart of Load comparison

Immediately we can see that as the amount of data transferred across all networks increases the load on the CPU also increases which is expected. The reason for this is simple as the network has more data to transfer there is a need for more processing power on the CPU.

All networks, on average, have the anticipated load percentage. Tunneling has the greatest average load, but it isn't much higher than the others, while the IPV4 only network has the lowest average load, albeit it isn't significantly lower.

The main reason for this occurring is the fact that the IPv6 traffic are sent within the Tunneling topology it has to run through the IPv4 network between the two routers.

The router (R1) which receives the first package has to encapsulate the package in order for it to be sent to the branch 2 (R2) router. During its encapsulation a new source and destination will be given by R1 and sent to R2.

This router would now have to decapsulate this package in order for it to be sent to the original destination address. The transport across the tunnel is reason why this load on the CPU is higher on this this network.

Because there is no need to alter the packets, whether through translation, encapsulation, or other means, IPv4 only networks traffic may simply deliver the packet directly to the target address through all devices on the networks, reducing the required CPU load. During this procedure, no changes to IPv4 traffic will be required.

Figure 26 Line Graph of Load comparison

The remaining 3 networks follow a similar trend with none of them being significantly higher or lower load in comparison to the other networks. This is shown in figure 26 above.

## 6.2 Delay

Delay is the time it takes for data to move from one endpoint on the network to another. It is a complex measurement affected by multiple factors. . I have used the ping utility in my networks to calculate the delay. This is shown in the figure below.

Figure 27 Bar chart of Delay comparison

Noticeably the delay in all of the networks stays constant on average, with Tunneling network having the lowest delay and the dual stack network having the highest delay.

It was expected that Dual-stack network would have had a higher response time than pure IPv4 and IPv6 networks since both IP stacks were processing at the same time and packets had to go through two distinct IP stacks, causing a minor delay and the results proved this.

In terms of a basic IPv4-only network vs an IPv6-only network, we can observe that on average, IPv4 had a smaller latency than IPv6, although this difference is not substantial, indicating that IPv4 performs similarly to IPv6.

Figure 28 Line Graph of Delay comparison

Before this parameter was tested it was expected that tunnelling would be one of the networks which would provide higher delay than the received results. This is due to the encapsulation and decapsulation of the packets inside the GRE tunnel.

## 6.3 Jitter

Jitter, in essence, is the difference in delay between two packets. Immediately we can see that that IPv4-only network has the highest jitter readings compared to the other 4 networks which is shown in the figure below.

Figure 29 Bar chart of Jitter comparison

As all networks have a similar topology especially the IPv6-only network, the network congestion cannot be the primary reason for the jitter to be so high in the IPv4-only Network.

The main reason for this that all the devices that were used within this network were IPv4 only hardware devices. Most of the equipment used in this topology were classed as ‘older equipment’. The delay across this traffic remained constant throughout the 10 attempts.

Any jitter less than 30ms, according to TechTarget-2021, is acceptable and optimal as long as there is no packet loss and the link between the devices is reliable.

This would be a problem if the network in question dealt with VOIP or video streaming. This is omitted in my study since neither of the five networks has any voice devices, but it will be acknowledged in the final conclusion.

Jitter buffering could be a feature that can help a device deal with jitter. When the buffer receives network packets first and then feeds them to the application at a regular rate, this is what happens.

The figure below shows the average trend line of the 4 (without IPv4 for ease of interpretation) out of the 5 created networks.

Figure 30 Line Graph of Jitter comparison

The IPv6 only network was produced more jitter time at the packet size of 128 Bytes while comparing with others.

The tunnelling network is the follows this, with being the second highest average readings. The reason for this again goes back to the concept of encapsulation and decapsulation using the IPv4 tunnel within the network. The router (R1) which receives the first package has to encapsulate the package in order for it to be sent to the branch 2 (R2) router. During its encapsulation a new source and destination will be given by R1 and sent to R2. This router would now have to decapsulate this package in order for it to be sent to the original destination address. This causes in increase in the delay between the two packets being transmitted.

The networks which have the lowest Jitter is both the NAT64 and Dual-stack Networks. For both of the networks the Jitter remains constant.

For NAT64 this is not something that was expected, as NAT must process every packet, update the header in both incoming and outgoing traffic, and as bandwidth grows, the completely inefficient NAT table will grow in size as more traffic is present.

However, after referring back to the research in a small topology this is overlooked, when the network is much larger this is when the jitter increases.

## 6.4 Additional Test/Analysis

Some additional tests were also running which were not the original plan of this report. The aim of these tests was to add to the amount of data found for these networks. The tests have been listed as subsections of this section below.

### 6.4.1 FTP Download Time

The File Transfer Protocol (FTP) is a network standard for transferring computer data from one host to another. FTP is based on a client-server design, with the client and server using separate control and data connections.

The download response time for an FTP programme is the time between issuing a request and getting the response packet.

This is the time it takes for a client programme to submit a request to the server and get a response packet. It is an essential aspect in determining the performance of FTP programmes; a faster download reaction time indicates greater performance.

This was tested over 10 attempts and the average readings of all of the 5 networks tested has been included in figure 31 below.

Figure 31 Average FTP Download time in seconds

Dual stack has the quickest response time, outperforming the other two, Tunneling and NAT64 networks.

Again, this is because of the translation procedure inside the NAT64 routers and the encapsulation and decapsulation of packets within the Tunnelling routers

### 6.4.2 Email download time

The email transmission is when messages are sent of the network. The time is measured from when the message is sent from the end device (source) to another end device (destination), or in other terms one is the server and one is the client.

This was tested over 10 attempts and the average readings of all of the 5 networks tested has been included in figure 32 below.

Figure 32 Average email download time in seconds

Out of the 3 migration networks the dual stack again performed the best in this. Tunneling had the highest response time which was expected and the NAT64 network only performed slightly better than tunnelling.

Again, this is because of the translation procedure inside the NAT64 routers and the encapsulation and decapsulation of packets within the Tunnelling routers.

## 6.5 Final results

The research aspect was crucial in not only gaining a better understanding of the internet protocols IPv4 and IPv6, but also how packets behave for example when they are encapsulated and decapsulated through the tunnel. As it was discovered that in a tunnelling network, packets must be encapsulated and decapsulated through the tunnel, which, based on the research, I expected the load to increase, which was demonstrated during the testing and analysis of this network.

In terms of coming to a conclusion of the best method of transition for the small organisation at hand, after taking the research, application, testing, evaluation, and table 17 into account the dual stack method looks to be the ideal method of transition.

This because of the simple reason that it used both IPv4 and IPv6 concurrently within the topologies routers and it is easy to handle for these routers. The load and Jitter are one of lowest from the 3 migration mechanisms which the Delay being not significantly higher than the others. This means it has the lowest pressure on the CPU’s and ensures that packets are received in the earliest time possible.

The table shown below displays the results in the form of colours. Green signifies, acceptable. Yellow signifies, should be considered but is acceptable. Red signifies, improvement needed and is not acceptable.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **Load** | **Delay** | **Jitter** | **FTP Speed** | **Email Speed** | **Total score** |
| **IPv4-only** |  |  |  |  |  | **3/5** |
| **IPv6-only** |  |  |  |  |  | **3/5** |
| **NAT-64** |  |  |  |  |  | **2/5** |
| **Dual-Stack** |  |  |  |  |  | **4/5** |
| **Tunnelling** |  |  |  |  |  | **2/5** |

Table 17 comparison of parameters against networks

From the research and creation of the migration mechanisms we also know that the translation mechanism NAT64, renders the network susceptible because if something goes wrong with the routers during the transfer, the entire network will collapse.

From the testing of this network the translation mechanism also increased the time it took for packets to be sent from the source point to the end device with the main reason being the translation of IPv4 headers into the IPv6 headers by the routers. This also meant the routers had to ensure more processing power ultimately resulting in the Load increasing.

Tunneling complicates network administration and makes security threats more difficult, which will irritate executives. The tunnel had to encapsulate the packets and the first router and then decapsulate and the second router which ultimately meant that the load and delay was both again increased more than the other 4 networks.

However, this does not negate the fact that it was only tested on one type of network. If the test were conducted for a larger company, with additional routers, tunnels, and branches, the advice for which migration technique should be utilised may easily vary.

The usage of the Voice over Internet Protocol may have also influenced the ultimate decision. Jitter is one of the most important aspects to consider while working with VOIP, as previously stated. Again, if more time had been gifted on this project, VOIP or multimedia packets would have been one of the features that would have been thoroughly examined.

In an ideal scenario it would be best to have an IPv6 only network as this had the quite easily a balance of all of the tested parameters as the download of FTP packets and the download of email packets were the lowest aside from IPv4 only network compared to the other migration networks. However, there are multiple things that are stopping this total IPv6 adoption.

Because of the lack of interoperability, operators will have to operate IPv4 and IPv6 at the same time for the near future. Also , Millions of routers and switches make up the internet and as these were created to function with IPv4 at first. It will take time and money to replace or upgrade them. Like it was mentioned in the research aspect of this study, “commercial businesses are expected to spend in excess of $26B for a total transition from IPv4 to IPv6.

The project plan, which was prepared early on in the project, was critical in paving the way for this report. The use of the five goals and objectives I set for myself at the start was crucial not only for writing this paper but throughout the entire project's journey. The suggested project plan is presented in the report's appendix section.

The plan was followed for a short time; however, it was changed as the project progressed. As stated in the opening part of this report, the building of the entire project has been divided into phases. The initial intention was for me to build all of the networks first, then test them once the construction for each network was finished, however as the project progressed, this was adjusted. Parts of stages 2 and 3 were the ones I selected to collect. This was performed by completing a network and then gathering the network's results; I only did this for the first network before going on to the next. This was done to ensure sufficient number of results were gathered in the case of any issued that arose.

# 7.0 Conclusion

The 3 core transition methods and a brief description of each methodology that aid service providers in migrating from IPv4 to IPv6 network and routing attributes were discussed. The performance of each network has been tested and the readings were discussed.

Many things were accomplished throughout the course of this project, including a better grasp of the IP protocols IPv4 and IPv6, as well as how industry is currently migrating from IPv4 to IPv6.

It was discovered that the worldwide IPv4 free pool had been depleted, and that switching to IPv6 will be necessary in the near future. Dual stack, translation, and tunnelling were the three most often used transition mechanisms, hence these three were created and tested individually. Each one has its own set of benefits and drawbacks.

The major goal of this research was to identify four main findings for each of the five built networks using varying packet sizes. All of the networks were successfully built, and all of the readings were received.

Although dual stack is straightforward to set up, network equipment must support both IPv4 and IPv6 protocols. It can also transfer packets between IPv4 and IPv6 networks. On the other hand, the tunnelling transition approach is a great option for networks with devices that do not yet support IPv6.

The SolarWinds tool is not something I planned to use, however as the project ran its course I felt it was something that could be used to display my results. Although this is not utilised extensively it provided me with an image in mind of how the packets run through the network.

Over the course of this project some problems have arisen, especially with the usage of software that was new for me, GNS3. The main problem came with setting up the virtual machines which integrate within the PCs within the topology which was the main usage for the testing tool IPERF3. After a long look at walkthroughs, forums, and articles the setup of this was complete. This did set the project timeline back around 3 weeks, however, the contingency weeks that had been arranged were utilised were used to overcome this issue.

It is now realised that for different organisation needs different networks are needed. My future research into this matter could be to test these networks with more performance parameters, in particular, how these networks would function with multimedia packets.

In organisation there will always be a need for migration from IPv4 to IPv6, however, Implementation and migration to IPv6 cannot and should not be done in a single day. To accommodate migration, major adjustments are necessary in every business. While significant progress is being made toward deployment of the new protocol, an IPv6-only Internet is still decades away.

# References

AbouSalem Z., Bin Abdul-Aziz University 2022. Compared Between Ipv6 and With Ipv4,Differences and Similarities. [PDF] Bin Abdul-Aziz University, Computer Department, KSA, pp.Range of pages from 7360-7361. Available at: <https://rajpub.com/index.php/ijct/article/view/7805> [Accessed 4 October 2021].

GNS3.com n.d.  [online] Available at: <https://docs.gns3.com/docs/> [Accessed 13 October 2021].

GUEANT, V., 2022. Iperf - The TCP, UDP and SCTP network bandwidth measurement tool. [online] Iperf.fr. Available at: <https://iperf.fr> [Accessed 14 November 2022].

Firewall.cx. 2022. IPv6 Subnetting - How and Why to Subnet IPv6. [online] Available at: <https://www.firewall.cx/networking-topics/protocols/877-ipv6-subnetting-how-to-subnet-ipv6.html> [Accessed 5 March 2022].

Patrizio, A., 2022. IPv4 vs. IPv6: What’s the Difference?. [online] IPv4 vs. IPv6: What’s the Difference?. Available at: <https://www.avast.com/c-ipv4-vs-ipv6-addresses> [Accessed 30 December 2022].

Qaid, A., 2022. Transition from IPv4 to IPv6 Mechanisms by GNS3 Emulation. [PDF] Dubai, United Arab Emirates: IEEE, p.A single Page 26. Available at: <https://ieeexplore.ieee.org/document/9615647> [Accessed 15 February 2022].

SCHLUTING, C., 2022. Networking 101: Understanding Tunneling | Enterprise Networking Planet. [online] Enterprise Networking Planet. Available at: <https://www.enterprisenetworkingplanet.com/standards-protocols/networking-101-understanding-tunneling/> [Accessed 11 January 2022].

Sri, N., 2020. What is Cisco Packet Tracer. [online] Available at: <https://www.geeksforgeeks.org/what-is-cisco-packet-tracer/> [Accessed 5 February 2022].

Statista. 2022. Number of internet users worldwide 2021 | Statista. [online] Available at: <https://www.statista.com/statistics/273018/number-of-internet-users-worldwide/> [Accessed 12 March 2022].

Tassey, G. IPv6 Economic Impact Assessment [eBook]. USA.

TroubleSnoop. 2020. What is SolarWinds Tool? and All about SolarWinds Tool. [online] Available at: <https://www.troublesnoop.com/what-is-solarwinds-tool/> [Accessed 7 November 2022].

Wireshark.org. 2022. Wireshark · Go Deep.. [online] Available at: <https://www.wireshark.org> [Accessed 19 October 2022].

Yusuf Maitama 2017. Design and Comparison Migration between Ipv4 and Ipv6 Transition Techniques. [PDF] Kano: Yusuf Maitama, p.A single page. Available at: <https://export.arxiv.org/ftp/arxiv/papers/1912/1912.11419.pdf> [Accessed 30 March 2022].

# Appendices

## Appendix 1-A

Table

Description automatically generated with medium confidenceTable

Description automatically generated with medium confidenceTable

Description automatically generated with low confidenceText

Description automatically generatedText

Description automatically generated with medium confidenceText

Description automatically generated with medium confidenceText

Description automatically generatedTable

Description automatically generatedA picture containing text

Description automatically generatedTable

Description automatically generated with medium confidence

## Appendix 1-B

Text

Description automatically generatedText

Description automatically generatedGraphical user interface, text

Description automatically generatedGraphical user interface, text

Description automatically generatedText

Description automatically generatedText

Description automatically generatedText

Description automatically generatedGraphical user interface, text

Description automatically generatedGraphical user interface, text

Description automatically generatedText

Description automatically generated

## Appendix 1-C

Table

Description automatically generatedTable

Description automatically generatedTable

Description automatically generatedTable

Description automatically generated

## Appendix 2-A

**Table

Description automatically generatedText

Description automatically generatedText

Description automatically generatedText

Description automatically generatedText

Description automatically generatedText

Description automatically generatedText

Description automatically generatedA picture containing timeline

Description automatically generatedText

Description automatically generated**

## Appendix 2-B

**Table

Description automatically generatedText

Description automatically generatedText

Description automatically generated with medium confidenceText

Description automatically generatedText

Description automatically generatedText

Description automatically generated with medium confidenceText

Description automatically generatedText

Description automatically generatedTable

Description automatically generated with medium confidence**

## Appendix 2-C

## **Text Description automatically generated with medium confidenceText Description automatically generated with medium confidenceText Description automatically generatedText Description automatically generatedText Description automatically generated with medium confidenceText Description automatically generatedText Description automatically generated with medium confidenceText Description automatically generatedText Description automatically generated with medium confidenceText, table Description automatically generated with medium confidence**

## Appendix 3-A

**Text

Description automatically generated with medium confidence**

**Table

Description automatically generated**

**Text

Description automatically generated with medium confidence**

**Text

Description automatically generated with medium confidence**

**Table

Description automatically generated**

**Table

Description automatically generated with medium confidence**

**Table

Description automatically generated with low confidence**

**Table

Description automatically generated with medium confidence**

**Text

Description automatically generated with medium confidence**

## Appendix 3-B

Table

Description automatically generated with medium confidenceText

Description automatically generatedText

Description automatically generated with medium confidenceText

Description automatically generatedA picture containing table

Description automatically generatedText

Description automatically generated with medium confidenceTable

Description automatically generated with low confidenceText

Description automatically generated with medium confidenceText

Description automatically generatedTable

Description automatically generated

## Appendix 3-C

## Table Description automatically generated

## Text Description automatically generated

## Table Description automatically generated

## Table Description automatically generated

## Appendix 4-A

**Table

Description automatically generated**

**Table

Description automatically generated**

**Text

Description automatically generated**

**Table

Description automatically generated**

**Table

Description automatically generated**

**Table

Description automatically generated**

**Table

Description automatically generated**

**A picture containing table

Description automatically generated**

**Table

Description automatically generated**

**Table

Description automatically generated**

## Appendix 4-B

## **Qr code Description automatically generated with medium confidence**

## **Text Description automatically generated**

## **Text Description automatically generated**

## **Text Description automatically generated with medium confidence**

## **Text Description automatically generatedText Description automatically generated with medium confidence**

## **Text Description automatically generated with medium confidence**

## **Scatter chart Description automatically generated with medium confidence**

## **Text Description automatically generated**

## **A picture containing text Description automatically generated**

## Appendix 4-C

## **Text Description automatically generated**

## **Text Description automatically generated with medium confidenceText Description automatically generated**

## Appendix 5 Project Plan Updated

Graphical user interface, text, application, Word

Description automatically generated

## Appendix 6 Time plan

## Chart Description automatically generated with low confidence