



Prifysgol Cymru
Y Drindod Dewi Sant
University of Wales
Trinity Saint David

**Performance Analysis between dynamic routing protocols using IPv4
and IPv6 Addressing Scheme**

By

Shaun Holborow

Supervisor: Nitheesh Kaliyamurthy

Second Marker: Dr Kapilan Radhakrishnan

**Project submitted as part of the requirements for the award of BSc
(Hons) Computing (Computer Networks and Cybersecurity)**

July 4th, 2022

DECLARATION

I, Shaun Ross Holborow declare that I am the sole author of this Project; that all references cited have been consulted; that I have conducted all work of which this is a record, and that the finished work lies within the prescribed word limits.

This work has not previously been accepted as part of any other degree submission.

FORM OF CONSENT

I Shaun Ross Holborow hereby consent that my Project, submitted in candidature for the B.Sc. (Hons) Computing (Networking and Cybersecurity) degree, if successful, may be made available for inter-library loan or photocopying (subject to the law of copyright), and that the title and abstract may be made available to outside organisations.

Abstract:

In this project the aim of this project was to analyse the performance of dynamic routing protocols (RIP, OSPF and EIGRP) using IPv4 and IPv6 addressing scheme.

The objectives of the project were to analyse each Internet Protocol Version (IPv4 and IPv6) as well as the routing protocols chosen for the experiment (RIP, OSPF, EIGRP) with their appropriate versions included based on the literature researched, evaluate each Internet Protocol Version and Routing protocol selected based on the literature to assess whether they are valid or not to use for the experiment planned, design a suitable plan/model in handling and finding out the differences of each protocol via network topologies, Implement and test the performance of the network designs and finally based on the tests done, analyse the outcomes to ascertain a conclusion.

The methods used for this were ones that involved software running virtual networks and packet generation tools that allowed different types of packets to be sent (ICMP, TCP, UDP).

This enabled the author to receive the required data which was then extracted and compared for the results as the major findings

The major findings of this project analysis were that

- Multi hop & Multi Routing protocol Test 4 shows greater Latency & Delay. Mostly because of the serial links and the number of hops.
- Smaller Packet sizes directly reduce the throughput
- RIP and EIGRP performed well with throughput on most tests whereas OSPF was restricted by the particular networking designs used i.e., inclusion of serial links.
- Test 4 which involved all 3 dynamic routing protocols and slower serial links was particularly impacted by the network design also.
- IPv6 was found to be better performing than IPv4 especially in the case of throughput. Performing better especially with larger packet sizes.
- UDP throughput was found to be by the author to be largely unaffected by any of the following parameters such as latency, delay, packet loss and jitter.

- Standard Deviation was used to investigate distribution of throughput test results for both IPv4 and IPv6. The results indicated most test results fell within one standard deviation of the mean.
- The packet loss results were made up of packet retransmissions. This was done in an effort to portray the impact of the said retransmissions on throughput rates.

The author recommends that when planning to use this software, ensure that the requirements for running said software are met as these tools take considerable resources from the CPU and RAM of the system for them to run efficiently if at all.

Also, the author recommends if anyone wishes to use the software for their own purposes but doesn't know how to use them then be sure to look up all the various videos, websites, forums, or any other sources for tutorials on how the software works and how to use them as these tools can be rather complex for most users to use such as was the case for GNS3 and the packet generation tools used.

Table of Contents

Table of Figures	5
1. Introduction:	7
1.1 Networks and Communication	7
1.2 IP Addressing	7
1.2.1 IPv4:	8
1.2.2 IPv6:	8
1.3 Routing Protocols	8
1.3.1 Static Routing	9
1.3.2 Dynamic Routing	9
1.3.3 Distance Vector	10
1.3.4 Path Vector	10
1.3.5 RIP:	10
1.3.6 OSPF:	10
1.3.7 EIGRP:	10
1.3.8 Measurement of Routing Protocols	11
1.4 Aim and Objectives	13
1.5 Summary of Introduction	13
2. Research & Review of Literature:	14
2.1 History	14
2.1.1 IPv4	14
2.1.2 IPv6	14
2.2 Topologies	15
2.2.1 Bus Topology	15

2.2.2	Star Topology	16
2.2.3	Ring Topology.....	16
2.2.4	Mesh Topology	17
2.2.5	Hybrids	17
2.3	Tools.....	18
2.3.1	Simulation Tools	18
2.3.2	Packet Generation Tools	21
2.3.3	Packet Analyser Tools	23
2.4	Relevant Sources	24
2.4.1	<i>Network Performance Evaluation of Internet Protocols IPv4 and IPv6 on Operating Systems 2009.....</i>	24
2.4.2	<i>Performance Comparison of Mobile IPv4 and Mobile IPv6 in wireless systems 2009</i>	25
2.4.3	<i>Implementing of IPv6 Protocol Environment at University of Kuala Lumpur Measurement of IPv6 and IPv4 Performance 2009</i>	25
2.4.4	<i>Performance Comparison of IPv4 and IPv6 in Peer-Peer and Client Server Local Area Networks 2010</i>	25
2.4.5	<i>Evaluation of OSPF and EIGRP Routing Protocols for IPv6 2013.....</i>	26
2.4.6	<i>Comparison of RIP, OSPF and EIGRP Routing Protocols based on OPNET 2014</i>	26
2.4.7	<i>Performance analysis of IPv4 / IPv6 protocols over the third-generation mobile network 2014</i>	27
2.4.8	<i>Performance Evaluation of RIP and OSPF in IPv6 using OPNET 14.5 simulator 2014</i>	27
2.4.9	<i>Performance Evaluation of Different Routing Protocols in IPv4 and IPv6 Networks on the Basis of Packet Sizes 2015</i>	27
2.4.10	<i>Performance Evaluation of IPv4 and IPv6 Routing Protocols on Wired, Wireless and Hybrid Networks</i>	28
2.4.11	<i>The Performance of IPv4 and IPv6 in Terms of Routing Protocols using GNS 3 Simulator 2018.....</i>	28
2.4.12	Literature Review Conclusion and Critical Review	29
3.	Research Methodology	29
3.1	Statistical Analysis	29
3.2	Research Onion.....	30
3.2.1	Positivism	30
3.2.2	Deductive.....	31
3.2.3	Experiment	32
3.2.4	Mixed Methods	32
3.2.5	Cross-Sectional	33
3.2.6	Data Collection and Data Analysis	33

3.3	Development Model	34
3.3.1	Waterfall Model Method	34
3.4	Project Plan	35
4.	Design & Implementation	36
4.1	Design	36
4.1.1	Evidence of testing	38
4.1.2	Network Topologies	50
4.2	Implementation	58
5.	Result	60
5.1	Analysis of the result.....	60
5.1.1	IPv4 Results:.....	60
5.1.2	IPv6 Results	66
5.1.3	IPv4 & IPv6 Combined Results	67
5.1.4	Standard Deviation, Mean, Median Results	68
5.1.5	Conclusion.....	70
6.	Evaluation.....	71
6.1	Future Work.....	71
7.	Conclusions & Recommendations	72
8.	Reflection.....	73
9.	References:	75

Table of Figures

Figure 1: Static Routing.....	9
Figure 2: IPv4 Addressing Scheme	14
Figure 3: IPv6 Addressing Scheme	15
Figure 4: Bus Topology	16
Figure 5: Star Topology.....	16
Figure 6: Ring Topology	17
Figure 7: Mesh Topology	17
Figure 8: Hybrids.....	18
Figure 9: Hyper-V.....	19
Figure 10: GNS3	20
Figure 11: VMware Workstation Pro 16.....	21
Figure 12: Ostinato	22
Figure 13: Packet Sender	23
Figure 14: Wireshark.....	24
Figure 15: Research Onion[32]	30
Figure 16: Waterfall Model Method (No deployment or maintenance)	34
Figure 17: Gantt Chart	35
Figure 18: Test 1 RIP TCP MSS 1460 Iperf	39
Figure 19: Test 1 RIP UDP MSS 1460 Wireshark	40
Figure 20: Test 1 RIP UDP MSS 1460 Iperf.....	40

Figure 21: Test 1 RIP TCP MSS 960 Iperf	41
Figure 22: Test 1 RIP TCP MSS 536 Iperf	41
Figure 23: Test 1 RIP ICMP MSS 1460 Wireshark	42
Figure 24: Test 2 OSPF TCP MSS 1460 Iperf	43
Figure 25: Test 2 OSPF TCP MSS 960 Iperf	43
Figure 26: Test 2 OSPF TCP MSS 536 Iperf	44
Figure 27: Test 2 OSPF ICMP MSS 1460 Wireshark	44
Figure 28: Test 3 EIGRP TCP MSS 1460 Iperf	45
Figure 29: Test 3 EIGRP TCP MSS 960 Iperf	45
Figure 30: Test 3 EIGRP TCP MSS 536 Iperf	46
Figure 31: Test 3 EIGRP ICMP MSS 1460 Wireshark	46
Figure 32: Test 4 RIP, OSPF, EIGRP TCP MSS 1460 Iperf	47
Figure 33: Test 4 RIP, OSPF, EIGRP TCP MSS 960 Iperf	47
Figure 34: Test 4 RIP, OSPF, EIGRP TCP MSS 536 Iperf	48
Figure 35: Test 4 RIP, OSPF, EIGRP ICMP MSS 1460 Wireshark	48
Figure 36: Test 1 RIP TCP MSS 1440 Iperf	49
Figure 37: Test 1 RIP TCP MSS 1220 Iperf	50
Figure 38: Test 1 RIP UDP MSS 1440 Iperf	50
Figure 39: IPv4 Version 1 Network Logical Topology	51
Figure 40: IPv4 Version 1 Network Physical Topology	52
Figure 41: IPv4 Network v5 Logical Topology (Final IPv4 Version)	53
Figure 42: IPv4 Network v5 Physical Topology (Final IPv4 Version)	54
Figure 43: IPv6 Version 1 Network Logical Topology	55
Figure 44: IPv6 Version 1 Network Physical Topology	56
Figure 45: IPv6 Network v5 Logical Topology (Final IPv6 version)	57
Figure 46: IPv6 Network v5 Physical Topology	58
Figure 47: IPv4 Results	60
Figure 48: IPv4 TCP Throughput	60
Figure 49: IPv4 ICMP Throughput	61
Figure 50: ICMP Packet Loss IPv4	62
Figure 51: UDP Packet Loss IPv4	62
Figure 52: TCP Packet Loss IPv4	63
Figure 53: All Protocol Latency IPv4	63
Figure 54: All Protocol Delay IPv4	64
Figure 55: ICMP Jitter IPv4	64
Figure 56: UDP Jitter IPv4	65
Figure 57: TCP Jitter IPv4	65
Figure 58: IPv6 Results	66
Figure 59: ICMP Throughput IPv6	66
Figure 60: TCP Throughput IPv6	67
Figure 61: TCP Throughput IPv4 & IPV6	67
Figure 62: Standard Deviation TCP Throughput for All Tests IPv4	68
Figure 63: Standard Deviation ICMP Throughput for All Tests IPv4	69
Figure 64: Standard Deviation TCP Throughput for All Tests IPv6	69
Figure 65: Standard Deviation ICMP Throughput for All Tests IPv6	70

1. Introduction:

In this project various aspects of networking are investigated.

This starts with a look into how these networks work and communicate with each other as well as how they handle various types of files or formats when transferring back and forth through the networks as.

IP routed and routing protocols are 2 of the main components that are commonly seen in use with networks particularly for users when on the internet. It is IP protocols that allow connection between different networking systems using one or both IPv4 and IPv6. Routing protocols (RIP, OSPF, EIGRP etc.) provide the dynamic traffic management features enabling interconnectivity between discrete systems.

The problem faced with the project was learning as well as using the software chosen for the specific aim and objectives. This was particularly evident when using GNS3 and Ostinato as GNS3 tended to crash or break due to issues related to saves or access to files. One example was when an error kept being shown claiming to be a firewall issue with Windows when in fact it was to do with access to files on a USB stick.

In addition, Ostinato had issues generating packets across the network due either to mac address issues or connection permissions.

Statistical analysis was used or more specifically "Descriptive Statistical Analysis" which was chosen as the main method to the quantitative analysis of the data extracted from the tests done followed by various graphics depicting the data in a simpler view.

Aims and objectives are shown for the project to provide a clearer view as to what the purpose of the project is as well as how that purpose is met with the objectives laid out.

1.1 Networks and Communication

Technological networks are a combination of hardware, cabling, network devices and computer software used together to enable computers to communicate with each other.

How these networks communicate depends on the type of communication required from the user with some examples being the downloading of files types such as mp3's or pdfs; using a web browser to look online for information relevant to research; or using a videotelephony like Skype for making voice calls or video conferences online etc.[1]

1.2 IP Addressing

IP Addressing involves the use of a unique address to identify a device on a network and stands for "Internet Protocol" which is the set of rules that govern the format of data sent via the internet or local network.

These IP addresses consist of a string of numbers separated by periods and are expressed as a set of four numbers with one example being 192.168.1.0 with each number in the set ranging from 0 to 255 (0.0.0.0 to 255.255.255.255).[2]

1.2.1 IPv4:

This internet protocol version is currently the most dominant version of all the internet protocol versions available over others like IPv6 (Currently being used interchangeably rather than on its own) based on statistics found so far being 32% over for IPv6 adoption meaning that IPv4 is likely 68%. [3] It helped create the rules and limitations for communication over the Internet or local networks etc.

It is because of this Internet Protocol that networks across the world can connect with other with the use of Packet Forwarding thus defining the typical format of communications, IPv4 addressing and routes of data etc.

IPv4 was first described in 1981 by IETF with RFC 791 (Internet Engineering Task Force) and was later taken in for use by the DOD of the USA (Department of Defence) in 1982, SATNET (Atlantic Packet Satellite Network) in 1982 and ARPANET 1983 (Advanced Research Projects Agency Network).

IPv4 uses an addressing system consisting of 32-bit addresses which is used to identify a device or node that is connected on a network.

The benefits of using IPv4 are that it has an excellent system support (supported on all network devices), uses simple topology (easy to setup and manage) and the address length is short (easier to write and memorise). [4]

Its drawbacks however are that it lacks address space (More devices being connected than address space can accommodate), uses weak protocol extensibility (insufficient size of IPv4 header) and has issues with security of communications (No means provided to limit access to information on networks). [5, p. 4]

1.2.2 IPv6:

This internet protocol version is essentially an upgrade over IPv4 and was specified in 1998 by IETF with RFC 2460 specification.

It was essentially created to solve the address exhaustion problem seen with IPv4 by using a 128-bit addressing rather than a 32-bit addressing system thus allowing for a far larger number of addresses to be used without upsetting the system as well as other benefits.

Its arguably second purpose being to increase the security of network connection via encryption to make man-in-the-middle attacks more difficult to use.

Some of the benefits of using IPv6 are that no manual configuration or DHCP are required, supports the longer 128-bit (16 bytes) source/destination addressing system and resolves IP addresses to link layer address via multicast neighbour request message.

The drawbacks however are that IPv4 and IPv6 cannot communicate directly with each other, IPsec is required, and options cannot be labelled in IP headers like in IPv4. [6, p. 6]

1.3 Routing Protocols

The role of routing protocols is to find better or more accurate paths for the data packets to transfer from source to destination which is done using different parameters such as autonomous systems and metric to locate better paths.

These routing protocols have various pros and cons with different approaches being used to share said routing updates to locate the best path for use of transfer.[7]

1.3.1 Static Routing

The mechanism behind static routing is the manual operating of routing between systems which means that any alteration to the networks and their connections (IP addressing) will immediately affect the whole network.

These configurations are normally done via the routers with any new connections being formed not being recognised by these devices until they are entered directly into each router in use.[8]

The pro of using this scheme is that you have direct control over the connections made meaning no changes will occur without the user's knowledge or direct input being already made.

The con however of this is that if changes are made especially as a common occurrence, then each of those changes would have to be considered by the admin and fixed accordingly which can take considerable time depending on the size of the network.

It can be seen from this observation with the World Wide Web as well as the networks created for this project that this is not a viable solution which is where Dynamic routing comes into play.

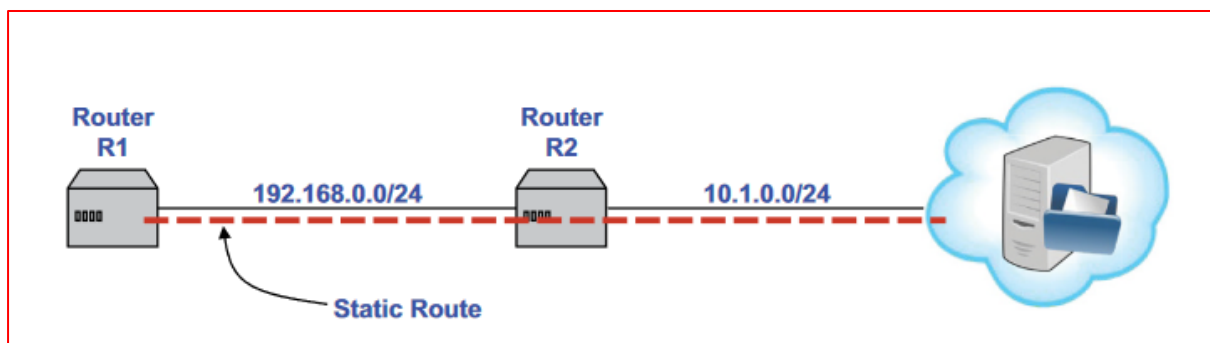


Figure 1: Static Routing

1.3.2 Dynamic Routing

A step up from the manual use of static routing, dynamic routing seeks to solve the problem with static routing whereby dynamic routing protocols are used such as RIP (Routing Information Protocol) to update all changes to routing information within networks when these changes occur.[8]

The pro of using this form of routing is that it allows networks to adapt to any changes to networks and their routing information meaning that the admin is no longer required to update or change each router manually for every network added or IP address found.

The con of this however is that the user or admin doesn't have as much control as to how it handles these changes such as if a network is added to another network, the protocol will only do as it's programmed in terms of handling routing information meaning it may not be entirely compatible with the setup planned or required by the admin thus possibly requiring

the user to either turn off the protocol or find the commands (if they exist) that allow these changes to be made without affecting the vision that the admin had in mind.

Despite the con of using dynamic routing however, it is continually used to this day for large networks and most notably the Internet as attempting to use static routing for these purposes would be deemed virtually impossible on any real scale.

1.3.3 Distance Vector

Distance vector routing involves the determination of the best path to a network destination (if multiple paths exist) based around routing metrics which show how far the destination is from the router making the decision.

The routing metrics used to reflect "Distance" is typically hop counts which is used in RIP or several variables like EIGRP.[8]

1.3.4 Path Vector

Path vector routing is another form of routing besides distance vector routing where it doesn't use "distance" or hop count to determine a loop-free path for a network.

It instead analyses the different paths available in the network using a routing policy it imposes on the network to find the most suitable path.[8]

1.3.5 RIP:

Routing Information Protocol is a dynamic routing protocol that utilises hop count as a routing metric to find the best paths between the source and destination of networks.

Hop count refers to the number of routers occurring in between the source and destination of a network with the best path being decided via the lowest hop count found.

It is a type of distance-vector routing protocols that has an AD value of 120 works on the application layer of the OSI model.

There are 3 versions of RIP currently in use today which are RIPv1, RIPv2 and RIPv6 which is only used for IPv6 Networks.[9]

1.3.6 OSPF:

Open Shortest Path First is an IP routing protocol and is an IGP (Interior Gateway Protocol) used for distributing IP routing information throughout a single AS (Autonomous System) in a network.

This routing protocol uses what is called link-state routing protocol, which means that the routers exchange topology information with their nearest neighbours.

OSPF consists of mainly 2 versions, OSPFv2 which is used with IPv4 and OSPFv3 which is used with IPv6.[10]

1.3.7 EIGRP:

Enhanced Interior Gateway Routing Protocol is a routing protocol used to automate routing decisions and configurations in computer networks with advanced distance vector routing. It is a proprietary Cisco protocol thus can only be used with Cisco routers.

It uses Minimum bandwidth to create connections from source to destination as well as using metrics from the network to measure the delay.

It is considered a hybrid protocol as it transitions well with IPv6 as well as IPv4 using classless routing technique.[11]

1.3.8 Measurement of Routing Protocols

For each routing protocol chosen, a set of testing parameters was used to obtain the data and results required.

- Throughput:

This is one of the first parameters tested, throughput represents the amount of data being sent or received by unit of time and was measured using Iperf3.

- Packet Loss:

This represents the number of packets (If any) lost when sent or received and was measured using Wireshark. Due to limitations with the testing tools in this project packet re-transmissions were considered packet losses.

- Delay

This involves the amount of time it takes for the first byte to arrive to its destination and was tested using ping.

- Latency

Like delay, it is the time it takes for all the data or packets to reach its destination and return to the source (RTT), was also tested using ping.

- Jitter

One of the more difficult ones for the author to measure, it involves the variation in latency of packets or packet flow and was measured using Iperf3 & ping.

Each of these testing parameters was used with a set of varying packet sizes and for a group of tests or scenarios involving several hops.

Test 1 – Scenario 1(RIP – Single Hop)

Using test parameters within a single hop within the RIP part of the network.

Test 2 – Scenario 2(OSPF – Multiple Hops)

Using test parameters within multiple hops within the OSPF part of the network.

Test 3 – Scenario 3(EIGRP – Multiple Hops)

Using test parameters within multiple hops within the EIGRP part of the network.

Test 4 – Scenario 4(RIP, OSPF, EIGRP – Multiple Hops)

Using test parameters within multiple hops from RIP, to OSPF and finally EIGRP.

Each of these scenarios was used for both IPv4 and IPv6 networks which were exclusive to each other with data or results obtained separately for each.

The packet sizes tested for each were as follows

IPv4:

Packet Sizes:

MTU 576 MSS 536

MTU 1000 MSS 960

MTU 1500 MSS 1460

IPv6:

Packet Sizes:

MTU 1280 MSS 1220

MTU 1500 MSS 1440

MTU = Maximum Transmission Unit

MSS = Maximum Segment Size

1.4 Aim and Objectives

The aim of this project is to analyse the performance of dynamic routing protocols (RIP, OSPF and EIGRP) using IPv4 and IPv6 addressing scheme.

To achieve the desired result of the planned experiment, 5 objectives must be followed and completed before the result can be achieved.

1. Analysis of each Internet Protocol Version (IPv4 and IPv6) as well as the routing protocols chosen for the experiment (RIP, OSPF, EIGRP) with their appropriate versions included based on the literature researched.
2. To evaluate each Internet Protocol Version and Routing protocol selected based on the literature to assess whether they are valid or not to use for the experiment planned.
3. To design a suitable plan/model in handling and finding out the differences of each protocol via network topologies.
4. Implement and test the performance of the network designs.
5. Based on the tests done, the outcomes are analysed to ascertain a conclusion.

1.5 Summary of Introduction

In summary of this introduction, various topics related to the work are discussed i.e. networks and how they communicate, the IP addressing system, routing protocols, the problem faced with the project and the aim/objectives set out for the project.

2. Research & Review of Literature:

2.1 History

In the project report *Internet Protocol Versions – A Review* they discuss the contributions to effective communication and exchange of information between people around the world using the internet.

TCP/IP (Transmission Control Protocol/Internet Protocol) is a protocol which allocates address to each device for recognition and disperses packets on the Internet. Over the years the IP has been changed because of user's requirements. The first IP has been used broadly is IPv4 (Internet Protocol Version 4) but I have encountered some problems with growth, the number of users who use the Internet. Internet Protocol version 6 is the next generation of Internet Protocol which has been used globally. IPv6 eliminates the most important problems of IPv4. This study briefly investigates the key features of IPv4 and IPv6.[12]

2.1.1 IPv4

Internet Protocol version 4 is the fourth version of Internet Protocol. This is widely developed as the first Internet Protocol. The length of address is 32 bit which permits for 4,294,967,296 unique addresses. Each address is written in dotted decimal notation, where each decimal value of the four-address byte is separated by "dots" or "periods." The range of each byte is from 0 to 255.[12]

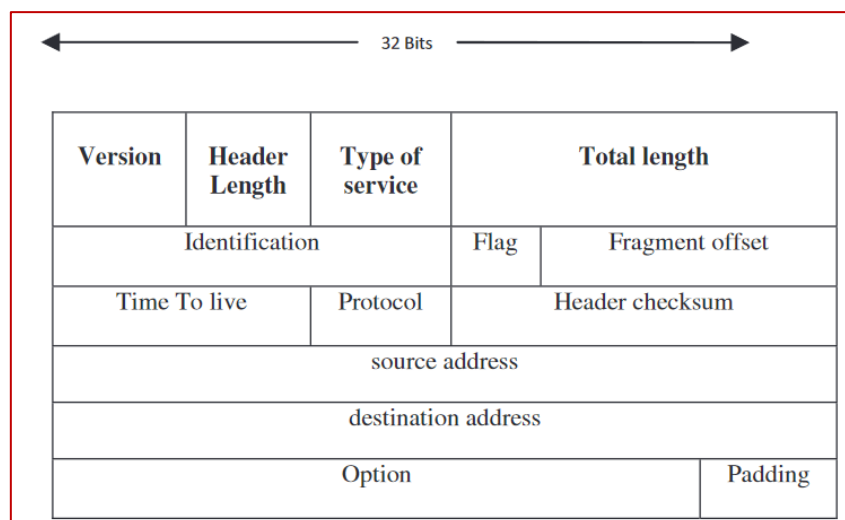


Figure 2: IPv4 Addressing Scheme

2.1.2 IPv6

The next generation of protocol which became popular is Internet Protocol Version 6 (IPv6), it is also known as Internet Protocol next generation (IPng). It has overcome some of IPv4 limitations. All the functions which worked correctly in IPv4 were integrated with IPng.[12]

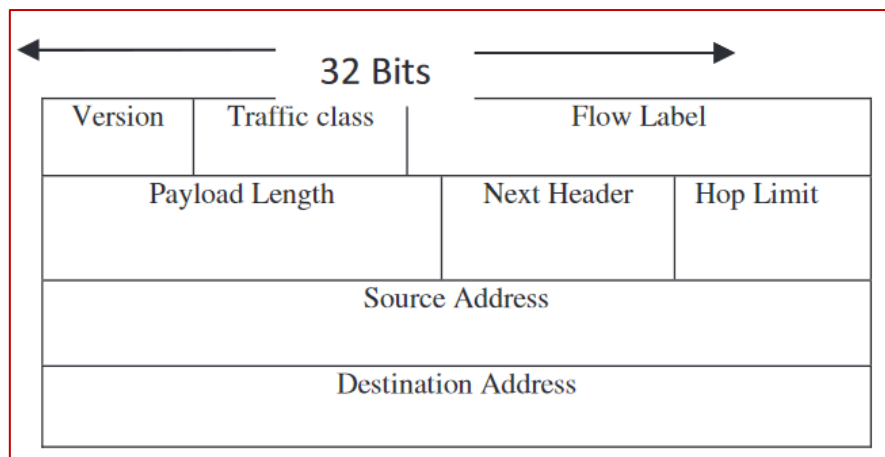


Figure 3: IPv6 Addressing Scheme

The author's opinion on this project report is that it provides useful information regarding the history and functions of each Internet Protocol Version with no visible issues regarding its content.

This project was used to find further research as to the history and functions of each Internet Protocol Version to help understand how each differs and why as my aim of the project is to test the differences between IPv4 and IPv6.

2.2 Topologies

In the book *Upgrading and repairing networks*, several unique network technologies are discussed as being developed over the past three decades. Different types of networks have different design criteria and, thus, various topologies have come into use. One Important Distinction needs to be made before we enter a serious discussion on topology: The physical topology describes the layout of a network media (such as copper and fiber-optic cables and, more recently, wireless equipment) and the devices that connect to it. The logical topology is concerned not with the actual physical connections but with the logical path through the network that data can take from one place to another. The differences will be more evident as the different topologies are discussed.[13]

2.2.1 Bus Topology

The simple bus topology structure was the first type used in Ethernet networks. The typical bus physical topology consists of a coaxial cabling common to all computer systems connected to the LAN. This coax is tapped in multiple places along its length, with each tap being used as a point of connection for a computing system. Taps can be physical cores cut into the coax (sometimes called a "vampire tap" or BNC-style "T-connectors" that join several individual pieces of coax together to form the common bus.[13]

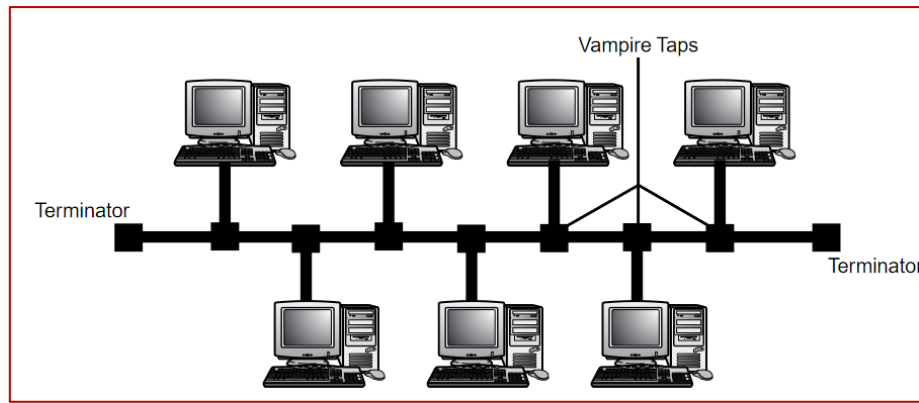


Figure 4: Bus Topology

2.2.2 Star Topology

The concept behind the star topology is simple. Every node on the LAN has a dedicated cable that is pulled back to a centralised point, typically a wiring closet. All cables are terminated in a network component within the closet, such as a hub or, more typically today, a switch, which handles the repeating or switching of traffic out to the other nodes on the network.[13]

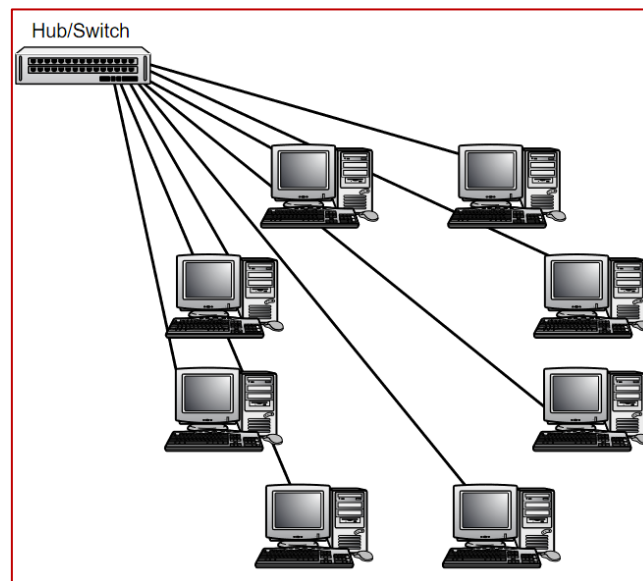


Figure 5: Star Topology

2.2.3 Ring Topology

Ring topologies are more complex than the bus and star topologies discussed in previous sections, but they offer some attractive features. Nodes logically communicate in a ring formation, with each node communicating only directly with its upstream and downstream neighbours.[13]

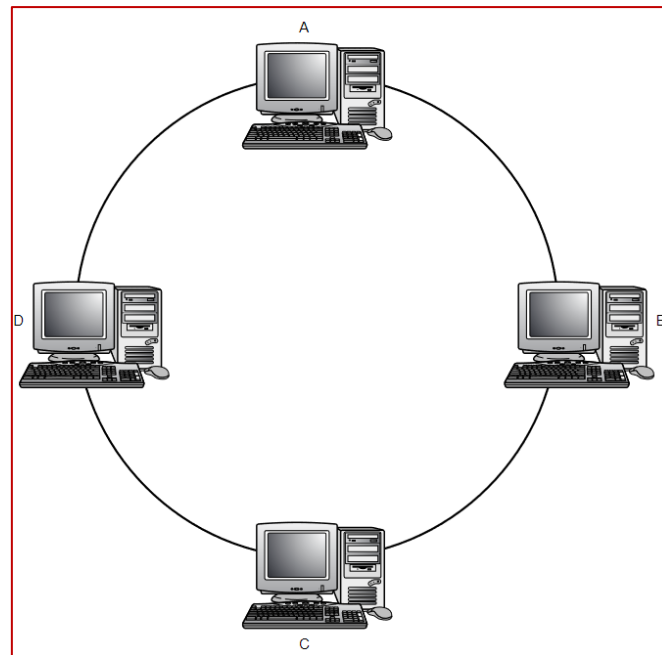


Figure 6: Ring Topology

2.2.4 Mesh Topology

Mesh topologies are a network setup where each computer and network device are interconnected with one another. This topology setup allows for most transmissions to be distributed even if one of the connections goes down. It is a topology commonly used for wireless networks. Below is a visual example of a simple computer setup on a network using a topology.[13]

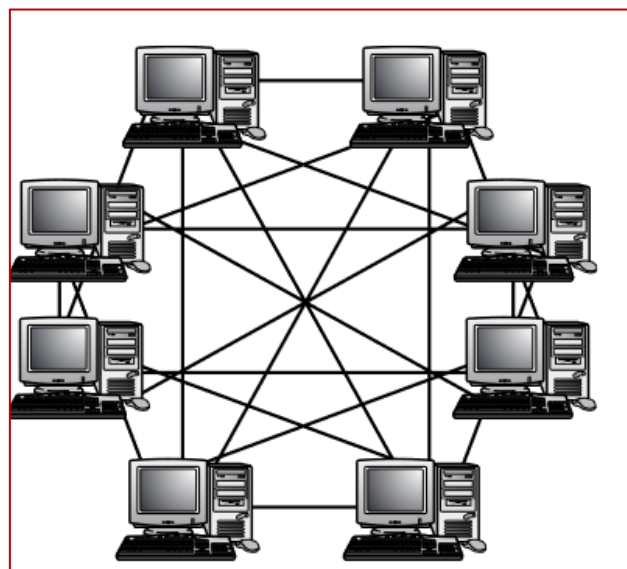


Figure 7: Mesh Topology

2.2.5 Hybrids

Hybrid topologies are a type of network topology that uses two or more differing network topologies. These topologies can include a mix of bus topology, mesh topology, ring topology, star topology and tree topology.[13]

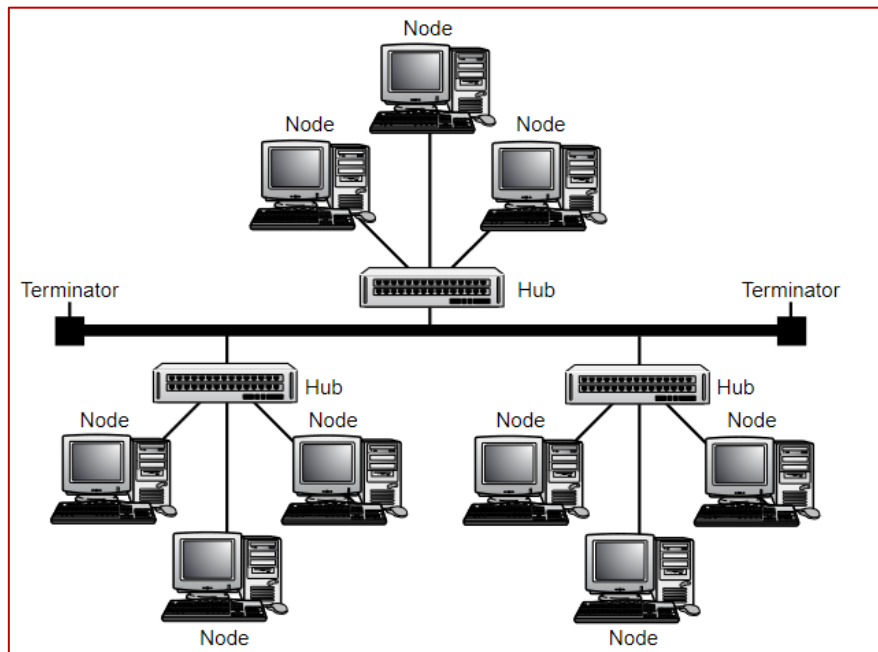


Figure 8: Hybrids

The author's opinion on this book is that provides useful information regarding things like the types of topologies available or that are currently in use as well as their benefits and no problems with found with the content either.

This book was used to research the different types of topologies that exist and that are used to then attempt to choose one and apply it to the author's networks.

2.3 Tools

2.3.1 Simulation Tools

2.3.1.1 Hyper-V

In the book *Pro Microsoft Hyper-V 2019: Practical Guidance and Hands-On Labs*, Andy Syrewicze et al covers what the program is and how it can be used for virtualisation.[14]

The author's opinion on this project report is that it provides useful information about what the program is as well as how to setup and use it for various purposes if it be to simulate networks or running another operating system etc.

This book was used to setup and use Hyper-V for the virtualisation of GNS3 to help create the networks required for testing.

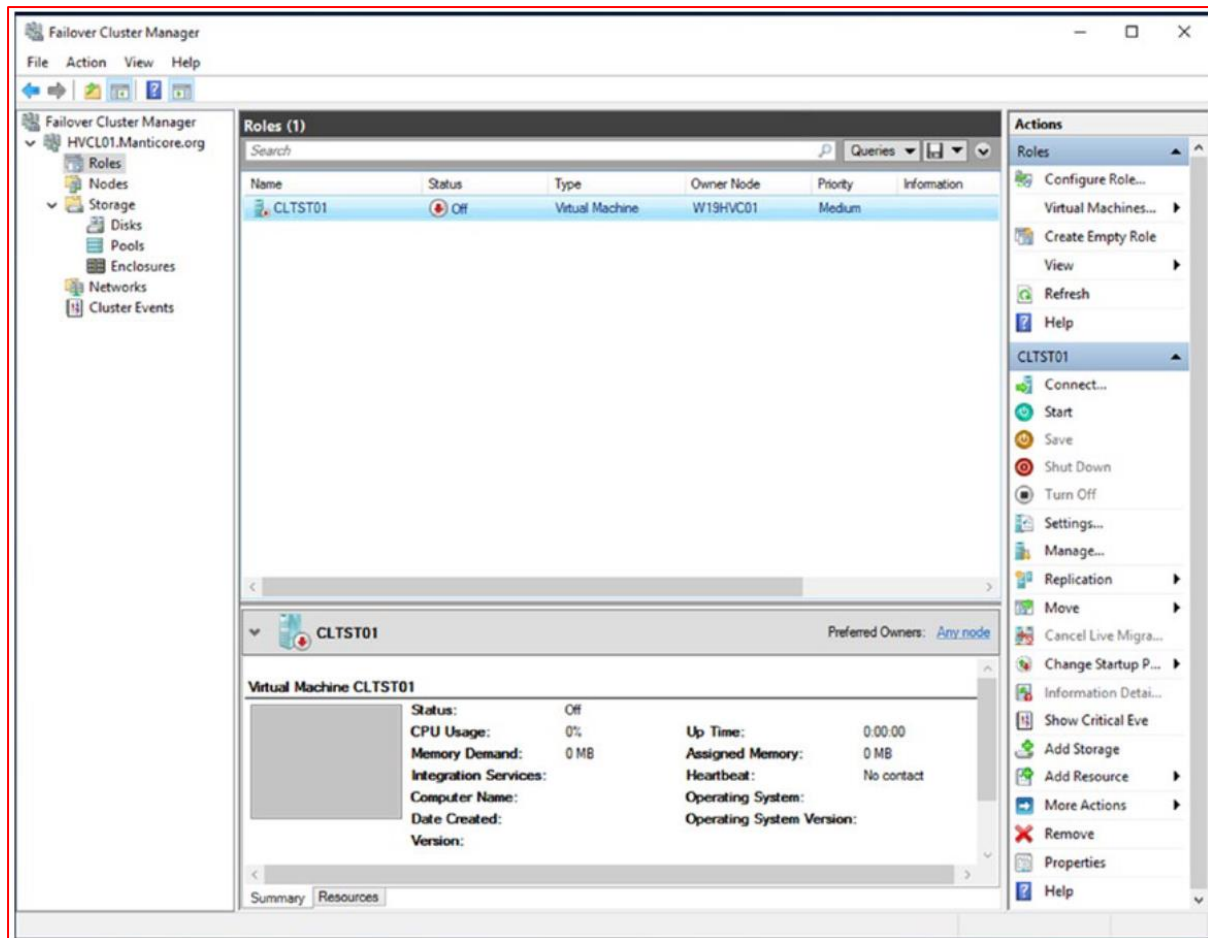


Figure 9: Hyper-V

2.3.1.2 GNS3

In *The Book of GNS3*, Jason covers everything that network engineers, administrators, and people studying for certifications need to get started, from walking you through installing and configuring GNS3 to creating and managing your projects. Jason digs deep while showcasing the true breadth of the software, covering topics like how to capture network packets, how to connect to real networks and live switches, and how to include advanced systems such as Juniper's vSRX Firefly and Cisco's IOS-XRv in your labs. He spends significant time explaining concepts and giving tips that will make you an expert user in no time.

Jason's book is ideal to harness GNS3 and make the most out of your network labs. Whether you are a beginner in the networking space or a seasoned professional, I can guarantee that you will walk away learning something new.

The author's opinion on this book is that it's a good resource for anyone to use who need more details as to how the program works and can be used with no problems in terms of

content. This book was used to install, configure, and use GNS3 for the purpose of creating the virtual networks.

It points out some pros and cons of using GNS3 of which being that it is easy to use and is portable however its requirements of RAM space and CPU speed can be costly on any system particularly low spec since they may not have the power need to run the program efficiently if at all.[14]

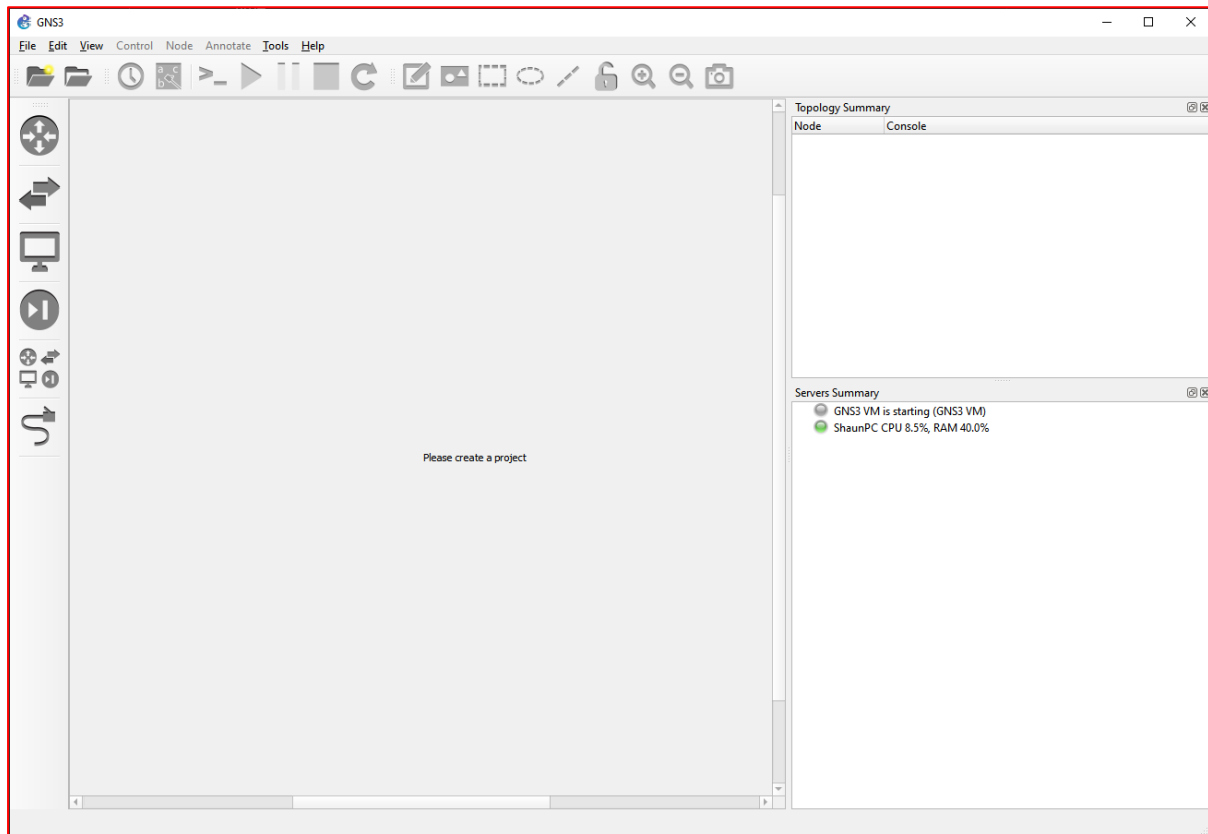


Figure 10: GNS3

2.3.1.3 VMware (Workstation Pro 16)

In *Using VMware Workstation Pro - VMware Workstation Pro 16.0*, the user guide covers everything users need to know before getting started from installation and configuration to further details pertaining to what it can do and how it can be used.

It explains the basic details of the program as well as some of the more advanced features that are available while also briefly discussing what the program is about and what the specs are that are required for using the program.

The author's opinion on this user guide is that it's a great resource for anyone to use who are new to using VMware in general and want more details as to how the program works and can be used with no problems in terms of content. This user guide was used to some extent in the installation, configuration, and use of VMware Workstation 16 Pro for the purpose of creating the virtual networks and operating systems etc. [16]

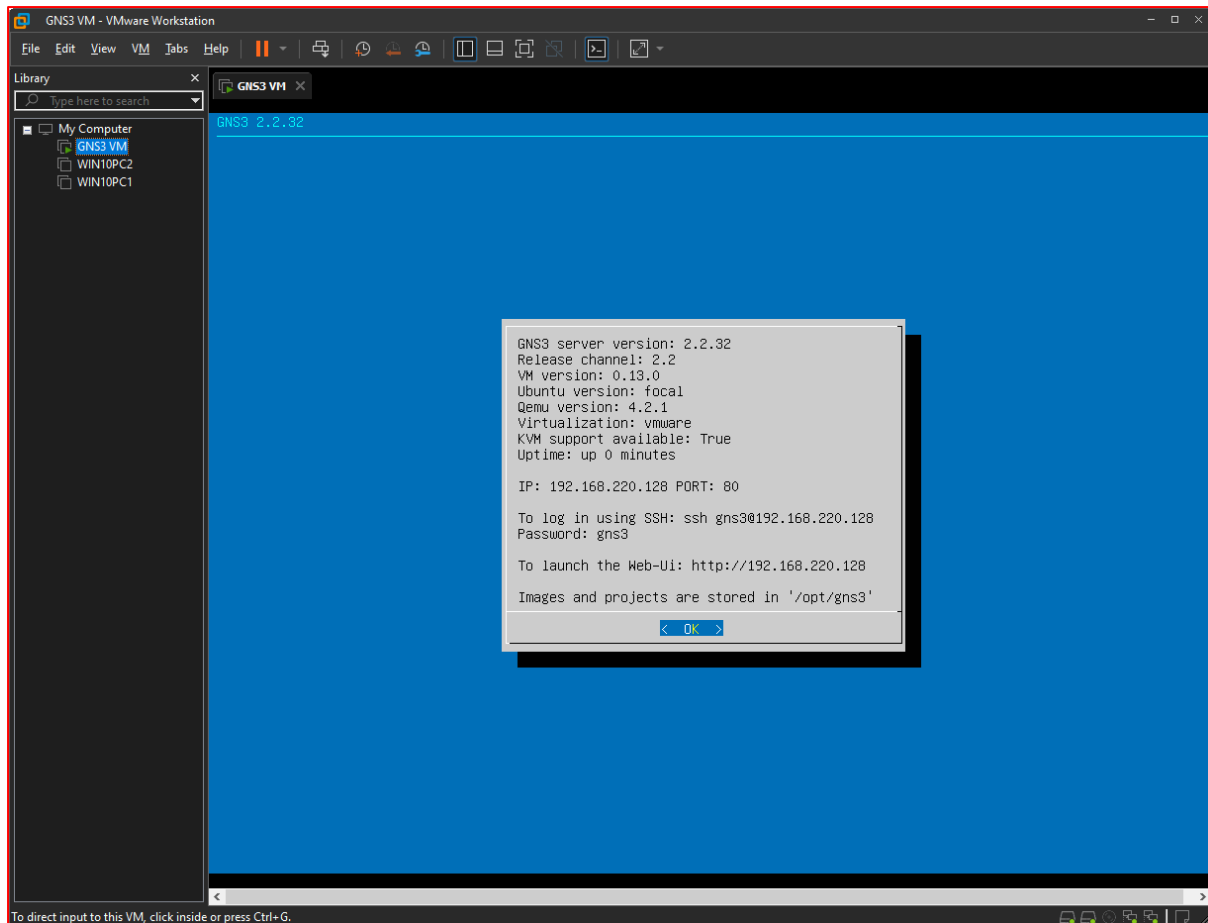


Figure 11: VMware Workstation Pro 16

2.3.2 Packet Generation Tools

2.3.2.1 Ostinato

Ostinato is a network traffic generator tool. In *Ostinato – A powerful traffic generator*, it addresses the features of Ostinato, its architecture and a basic capture and replay analysis, followed by scope of future integration with newer protocols. A packet trace of Address Resolution Protocol (ARP) is used to show how Ostinato can be used to edit and replay packets. Various snapshots of Ostinato's Graphical User Interface (GUI) are used to explain the process.

The conclusion from the project report is that Ostinato generally provides good support for various functions required by users such as build in support for Wireshark, real time statistics built to read the current packet details sent and received, and automation for large numbers of packets being used with Python support.

The author's opinion on this conference project report is that it provides useful information regarding the basics on how to use Ostinato and explains the various functions it has available to the user to utilise. This means that the user is less likely to need to search and use other tools outside Ostinato as it already provides support in key areas that network engineers would likely need for testing.

This conference project report was used to setup and use Ostinato with GNS3 in are to send and analyse the packets required for the author’s setup of networks.[17]

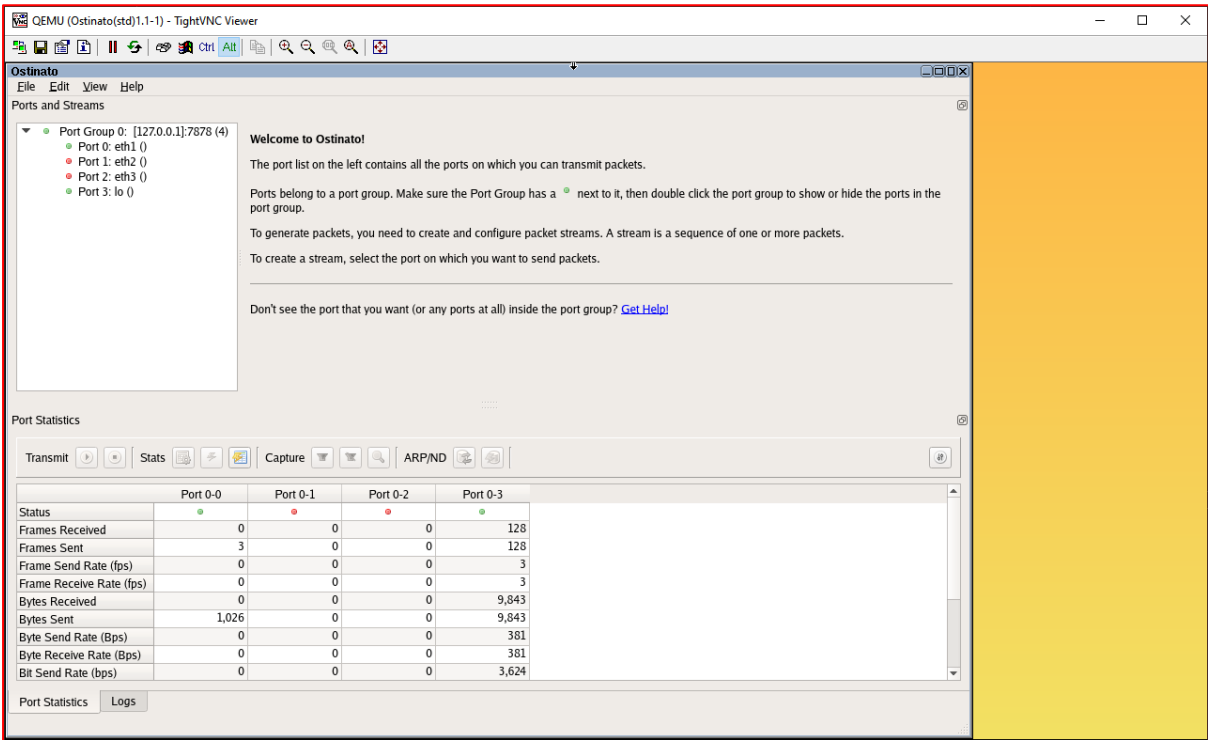


Figure 12: Ostinato

2.3.2.2 Packet Sender

On the main website, it describes Packet Sender as an open-source utility to allow sending and receiving TCP, UDP, and SSL packets as well as HTTP/HTTPS requests and panel generation. It supports Windows Mac and Linux and can be made to be supported for other operating systems if required.

The author’s opinion on this tool or source is that it provides some useful documentation regarding setup and usage, while it was initially useful for its ability to deal with sending and receiving TCP packets which was the main reason it was chosen for use over Ostinato however compared to Iperf, it proved to be more complex in usage thus it was replaced by Iperf instead.

This website or tool was used originally to find out more information about the tool as well as to send and receive TCP packets.[18]



The screenshot shows a window titled "IPv4 Subnet Calculator" with a close button (X) in the top right corner. The window has a dark background with light-colored text and input fields. The fields are arranged in a grid-like fashion. The "IP" field contains "192.168.1.23". The "Subnet" field contains "255.255.255.0". The "Start" field contains "192.168.1.1". The "End" field contains "192.168.1.254". The "Broadcast" field contains "192.168.1.255". The "Within subnet check" field contains "10.0.2.3". To the right of this field is a button labeled "No entries in subnet.". Below these fields is a large text area containing the text "Your non-loopback addresses:" followed by two lines of addresses: "192.168.254.11 / 255.255.255.0" and "fe80::49ee:35f5:288f:f355%wireless_32768 / ffff:ffff:ffff:ffff::". At the bottom of the window are two buttons: "Clear" on the left and "Close" on the right.

Figure 13: Packet Sender

2.3.2.3 iPerf

On the main website, it describes iPerf as a tool mainly for measuring the maximum achievable bandwidth on IP networks however it can measure various other parameters that were found to be useful and was developed originally by NLANR/DAST under a BSD license.

The author's opinion on this tool or source is that it provides some useful documentation regarding setup and usage, it was through this tool was that specific testing could finally be done particularly in the case of TCP and Jitter etc.

This source was used to gain measurements on TCP packets and Jitter as well as other parameters along with setup of the software.[19]

2.3.3 Packet Analyser Tools

2.3.3.1 Wireshark

In the project report *Wireshark for Security Professionals*, the author talks about Wireshark and its functions.

Wireshark is a free tool available on Unix, Windows and various other system that is used to understand data you capture from a network. The captured data is interpreted and presented in individual packet form for analysis, all within Wireshark. As you probably already know, packets are the chunks of data streaming on a network but can be called other names depending on where in the system the data is interpreted.[20]

The author's opinion on this book is that provides good information regarding the basics of using the program and its various features with no real problems found.

This book was used to setup and use Wireshark for GNS3. It was used both outside the program as a packet capturing tool and inside the program as an extra program alongside Ostinato to help with reading the packets.

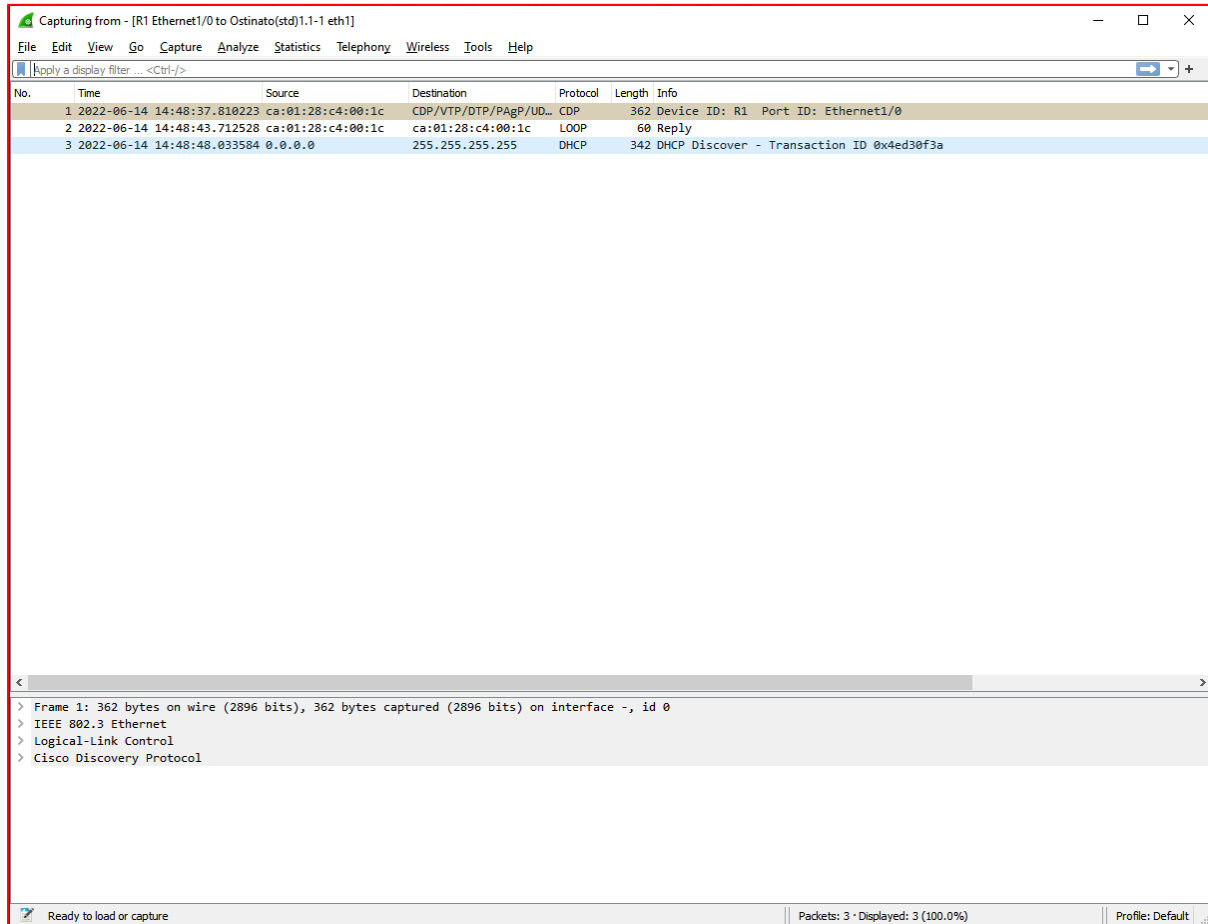


Figure 14: Wireshark

2.4 Relevant Sources

2.4.1 *Network Performance Evaluation of Internet Protocols IPv4 and IPv6 on Operating Systems 2009*

In this project report, Shaneel Narayan et al configures six operating systems namely, Windows and Linux distributions with two versions of Internet Protocol and evaluates their performance differences with the typical test parameters used.

The conclusion from this project report was that performance differences were found between IPv4 and IPv6 which was dependent on the operating system in use with one example being that Windows Vista was found to have throughput differences compared to Linux Ubuntu.

Also packet size is mentioned to have an effect on the throughput for both Internet Protocol Versions when bigger than 256 Bytes however not so much when it is smaller.[21]

The author's opinion on this project report is that it provides some interesting information regarding performance differences when using various operating system with Internet Protocol Versions with the parameters Jitter, Packet Size, Throughput etc.

This project report was used to provide a better idea as to what to expect in terms of the differences found between each of the parameters tested when applied to the project.

2.4.2 Performance Comparison of Mobile IPv4 and Mobile IPv6 in wireless systems 2009

In this project report, R. Gunasundari et al evaluates the TCP performance of MIPv4 and MIPv6 with comparisons made using simulation tools. The results from the tests done showed that both MIPs showed good performance when used with VoIP however MIPv6 was still considered the more suited option particularly if it's extended with FMIP and HMIP etc.[22]

The author's opinion on this project report is that it provides interesting differences found with MIPv4 and MIPv6 on wireless systems particularly with delay.

This project report was used to provide a better prospective as to what the differences are likely to be for my project.

2.4.3 Implementing of IPv6 Protocol Environment at University of Kuala Lumpur Measurement of IPv6 and IPv4 Performance 2009

In this project report, Mohd Nazri Ismail et al describes development of a video conferencing system with multimedia capabilities (Video – Audio) over IPv6 environment at University of Kuala Lumpur (campus environment)

They say that the benefit of the research is to present a new service of the Internet Protocol (IPv6) at the campus environment with multicasting explored.

The conclusion of the project report was that throughput between both IPv4 and IPv6 was affected after weekly statistics were conducted.[23]

The author's opinion on this project report is that it provides some interesting information relating to differences found between both Internet Protocol Versions when tested for throughput however the problem with the problem project report being that it is old which makes it not entirely suitable for application with this project due to changes over time.

This project report was used to provide a better understanding as to how these Internet Protocol Versions are used and how throughput is analysed for each as this is one of the parameters the author was required to test for.

2.4.4 Performance Comparison of IPv4 and IPv6 in Peer-Peer and Client Server Local Area Networks 2010

In this project report, Samed Salehi Kolahi et al tested the performance of IPv4 and IPv6 for Client-Server and Peer-Peer networks. For both networks. IPv4 produced higher bandwidth for TCP protocol. For UDP, IPv4 and IPv6 showed insignificant bandwidth differences except for packet size of 384 bytes where IPv4 had better performance in client-server environment.

The conclusion from this project report was that there were differences between IPv4 and IPv6 with one example from the project report that indicated that TCP throughput was higher for IPv4 in both Peer-Peer and Client-Server networks.[24]

The author's opinion on this project report is that it provides interesting information regarding the performance between both Internet Protocol Versions depending on the type of network tool used.

This project report was used to provide further information regarding the differences found between IPv4 and IPv6.

2.4.5 Evaluation of OSPF and EIGRP Routing Protocols for IPv6 2013

In this project report, Alex Hind et al made a study and comparison of two popular routing protocols; OSPF and EIGRP has been undertaken with changes between IPv4 and IPv6 discussed and compared.

The conclusion from this project report was that EIGRP had the advantage over OSPF in several areas but it is held back by its proprietary nature and costs leaving OSPF not far behind as another choice.[25]

The author's opinion on this project report is that its good information to the two routing that the other has also chosen as the routing protocols to use for the project. Particularly interesting is that they mention OSPF is best suited for large networks however EIGRP is omitted here even though it suggests it can perform similarly if not better than OSPF in the same key areas.

The only problem the author found with this project report is that it listed only IPv6 as the Internet Protocol Version of choice for testing in the title, however IPv4 is also apparently tested.

This project report was used to find more information on these two routing protocols as well as a test result done with IPv4 and IPv6.

2.4.6 Comparison of RIP, OSPF and EIGRP Routing Protocols based on OPNET 2014

In this project report, Justin Deng et al uses OPNET to simulate RIP, OSPF and EIGRP to compare their attributes and performance.

The conclusion of this project report was that EIGRP was the best for large and small networks with the fastest convergence and the most efficient use of bandwidth. If EIGRP was not chosen, then their second choice was OSPF which also performed best on large networks.

They found however that RIP was the worst in performance in large scale networks thus making it more suitable for small scale networks, this ties in with what others have said and shown saying about RIP's performance on large and small networks with most seeming to agree that RIP is not suited for large networks and better off used for simple smaller networks.[26]

The author's opinion on this project report is that it provides good information regarding the information of each routing protocol all of which were selected for experimentation with the project with interesting correlation found with regards to performance of RIP with other project reports or material found.

This project report was used to provide more detail as to the current differences known between each of the routing protocols chosen and confirm suspicions regarding performance differences depending on the size of the network.

2.4.7 Performance analysis of IPv4 / IPv6 protocols over the third-generation mobile network 2014

In this project report, Daniel Abad Camarero measures how the IPv6 protocol performs vs IPv4, analysing them each based on IP data, signalling, web load time and battery etc.

The conclusion of this project report was that varying differences were found, signalling and web loading time is less in IPv6 than in IPv4 etc.[27]

The author's opinion on this project report is that it provides interesting results pertaining to IPv6 vs IPv4 performance.

This project report was used to provide more information regarding the differences found between IPv4 and IPv6.

2.4.8 Performance Evaluation of RIP and OSPF in IPv6 using OPNET 14.5 simulator 2014

In this project report, Rajneesh Narula et al used OPNET 14.5 the simulation software to evaluate the performance of RIP and OSPF using IPv6. Three network models were used for testing with the following testing parameters being packet delay variation, delay, traffic received, traffic sent, response time, jitter etc.

The conclusion of this project report was that there was various performance between both routing protocols with one mention being that when both routing protocols under OSPFv3 and RIPng are used together for IPv6 the performance on that network increases compared to if each were used separately. Individual routing performance was found to be better with OSPFv3 than RIPng which correlates with what other project reports have said which also state RIP as being inferior when used on larger networks.[23]

The author's opinion on this project report is that it provides interesting results and statistics involving routing protocols used exclusively with IPv6.

This project report was used to find more information regarding these routing protocols and internet protocol versions to help with the project regarding the differences as evidence.

2.4.9 Performance Evaluation of Different Routing Protocols in IPv4 and IPv6 Networks on the Basis of Packet Sizes 2015

In this project report, Dipti Chauhan et al evaluated the performance of the routing protocols RIP, RIPng, OSPFv2 and OSPFv3 for IPv4 and IPv6 networks using Mobile Adhoc Networks with the simulations being done using the software Exata Cyber 1.1 Simulator.

The conclusion of this project report was that in general the packet size determines the performance of the networks with bigger packet sizes increasing the performance of said networks. Specifically, however RIPv2 on its own was considered the best performance wise.[29]

The author's opinion on this project report is that it provides interesting information and results regarding the use of different packet sizes, routing protocols and internet protocol versions.

This project report was used to implement a similar form of tests involving different packet sizes with some slight differences.

2.4.10 Performance Evaluation of IPv4 and IPv6 Routing Protocols on Wired, Wireless and Hybrid Networks

In this project report, Sonal Telang Chandel et al evaluated the performance of various routing protocols for IPv4 and IPv6 over wired, wireless and the hybrid networks. These simulations were carried out using Exata/Cyber 2.1 simulation software.

The conclusion of this project report was that AODV (IPv4) performed the best in areas like throughput, packet delivery ratio, delay, and jitter etc.[30]

The author's opinion on this project report is that it provides interesting if reliable results regarding the differences found with the testing parameters chosen.

This project report was used to provide further insight as to what to expect from the project when testing is done using these parameters.

2.4.11 The Performance of IPv4 and IPv6 in Terms of Routing Protocols using GNS 3 Simulator 2018

In this project report, Dalya Raad Al-Ani et al created a virtual scenario to test the performance of IPv4 and IPv6 with specific focus on analysing their performance in delivering the messages from the sender to the terminal, with more emphasis towards the Latency and Delay. Three different types of dynamic routing protocols, RIP, EIGRP and OSPF, had been used for this purpose.

The conclusion of this project report was that IPv6 after testing with all the parameters is better than IPv4. The size of the packets is found to indeed affect the time that the packets take to reach its destination. With IPv4 and RIP protocol, whenever the size of the packet is small this will make the packet reach slowly, compared to the other routing protocols however smaller packet sizes resulted in affected speed.[31]

The author's opinion on this project report is that it provides some useful information and data to compare for the project mainly relating to its inclusion of testing with similar if not identical prerequisites as the author in terms of protocols, parameters, and tools.

However, it is noted that the problems with this project report is its lack of detail in certain areas and evidence to show usage of BGP and RIP with testing results seeming absent.

This project report was used by the author to provide a better understanding as to how to go about the project regarding the tools, protocols, packet sizes, testing parameters etc.

2.4.12 Literature Review Conclusion and Critical Review

After reviewing the literature found and available, critical reviewing resulted in various findings compared to the author's own work.

Three authors discussed the affect packet size had on performance of networks with Shaneel Narayan saying that packets bigger than 256 bytes will become affected performance wise.

A contradiction was found between two authors' regarding performance of packets based on size as Dipti Chauhan et al said that bigger packets meant better performance all around however Dalya Raad Al-Ani et al says that the bigger packet sizes affected the time it took for packets to reach their destination which suggests that bigger packet sizes doesn't equate to better performance all of the time.

EIGRP and OSPF were discussed in detail by Alex Hind et al and Justin Deng et al where both discussed that EIGRP was found to be just as good if not better than OSPF in performance regardless of the size of networks however Alex Hind et al mentions its lack of more usage was down to its proprietary nature and cost.

3. Research Methodology

For this project to work, a certain level of research methodology had to be used to ensure that all essential areas are applied with suitable work.

To achieve this, the research methodology chosen for this task was one considered most suited which was the research onion methodology which involves a key set of areas (Philosophies, Approaches, Strategies, Choices, Time horizons, Techniques, and procedures etc.).

3.1 Statistical Analysis

The type of statistical analysis used will be Descriptive Statistical Analysis which involves quantitative analysis of the features of data by presenting it as summaries in the form of tables, charts, and graphs.

The justification behind using this form of analysis was that it was required to summarise all the relevant data obtained from the testing done into suitable form of information that could then be presented in a way that most viewers experts or not could digest such as any differences found between the datasets.

Quantitative analysis was chosen because the while the outcome is focused on the quality of networks, analysing the quality of Packets, fastness, errors etc. This quality is measured based on the quantity of data or numbers obtained through experimentation and testing which explains why it is quantitative analysis.

It is mainly because of this as well as the very nature of project of which being experimental that this form of analysis was chosen

3.2 Research Onion

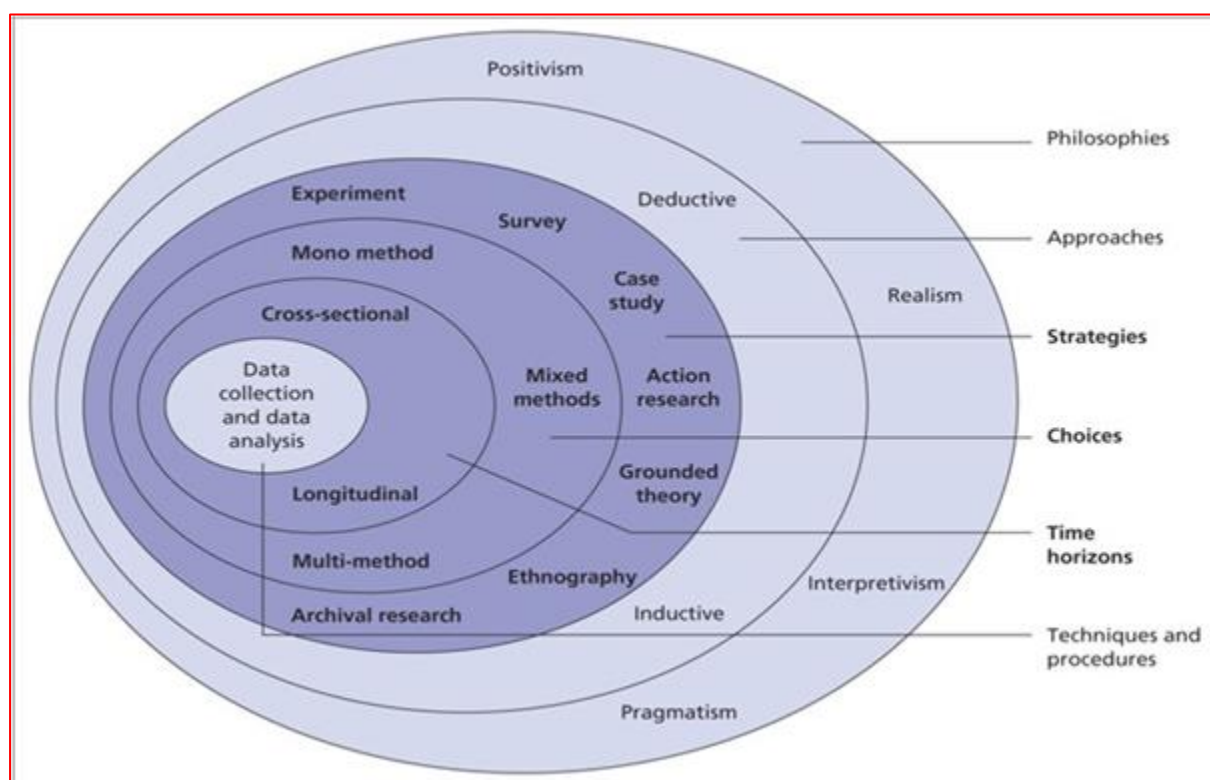


Figure 15: Research Onion[32]

3.2.1 Positivism

Positivist philosophy takes the view that knowledge exists outside of what's being studied. In other words, what is being studied can only be done so objectively, and it cannot include opinions or personal viewpoints – the researcher doesn't interpret, they only observe.

Positivism states that there is only one reality and that all meaning is consistent between subjects.

In the positivist's view, knowledge can only be acquired through empirical research, which is based on measurement and observation. In other words, all knowledge is viewed as a posteriori knowledge – knowledge that is not reliant on human reasoning but instead is gained from research.

For the positivist, knowledge can only be true, false, or meaningless. Basically, if something is not found to be true or false, it no longer holds any ground and is thus dismissed.

An example of this would be the question of whether God exists or not. Since positivism takes the stance that knowledge must be empirically vigorous, the knowledge of whether God exists or not is irrelevant. This topic cannot be proven to be true or false, and thus this knowledge is seen as meaningless.[33]

Positivism applies to this work by involving results being obtained from research and observation, in other words empirical evidence which is done through the research of the topic followed by testing of virtual network environments and then finally getting results that can be analysed and evaluated with data being found and presented graphically.

The advantages of using this layer are that it uses a quantitative approach which provides more objective information for use in scientific assumptions and a good structure for studies and discussions.

The disadvantages of using it however are that it relies on human behaviour meaning emotions can get in the way of providing accurate results and that its inflexible meaning that if any unexplained event takes place, progress is likely to be halted or ruined altogether as creativity for people in this area of thinking to help with solving these issues is not commonplace.

The author worked around these disadvantages by always keeping emotions in check (No attachment) and being prepared beforehand to be as flexible as possible for any possible unexpected events that could suddenly take hold of the project.[34]

3.2.2 Deductive

The deductive approach involves beginning with a theory and aiming to build on (or test) through research.

An example of this:

A deductive approach would be taken when investigating changes in the physical properties of animals over time, as this would likely be rooted in the theory of evolution. In other words, the starting point is a well-established pre-existing body of research.[33]

Deductive applies to this work by involving research of my topic followed by testing of virtual networks to receive results that are applicable to my work.

The advantages of the deductive approach are that it is a more focused method of testing hypotheses thus offering a possibility to show causal relationships between the concepts and the variables (quicker to conduct).

The disadvantages however are that it doesn't use divergent thinking, limits scope of creativity and makes assumptions that all disciplines in natural science function in an identical manner when they don't.

The author worked around these disadvantages by creating multiple solutions to a problem, using more creativity and not making assumptions about various disciplines that are incorrect.[35]

3.2.3 Experiment

Experimental research involves manipulating one variable (the independent variable) to observe a change in another variable (the dependent variable) – in other words, to assess the relationship between variables. The purpose of experimental research is to support, refute or validate a research hypothesis. This research strategy follows the principles of the scientific method and is conducted within a controlled environment or setting (for example, a laboratory).

Experimental research aims to test existing theories rather than create new ones, and as such, is deductive in nature. Experimental research aligns with the positivist research philosophy, as it assumes that knowledge can only be studied objectively and in isolation from external factors such as context or culture.

Let's look at an example of experimental research. If you had a hypothesis that a certain brand of dog food can raise a dogs' protein levels, you could make use of experimental research to compare the effects of the specific brand to a "regular" diet. In other words, you could test your hypothesis.

In this example, you would have two groups, where one group consists of dogs with no changes to their diet (this is called the control group) and the other group consists of dogs being fed the specific brand that you aim to investigate (this is called the experimental/treatment group). You would then test your hypothesis by comparing the protein levels in both groups.[33]

Experiment applies to this work via the creation of an experiment involving 2 or more networks that are tested for certain parameters using packet traffic generation tools.

The advantages of the experiment approach are that it provides researchers high level of control in regard to how they measure and manipulate variables in a observable setting and that the results obtained from correctly done experiments can provide valuable information for studies being done.

The disadvantages however are that can lead to artificial results due to changes to the experiment or variables in a way that does not at all relate to the real world and that it typically costs considerable money and time to setup the experiments.

The author worked around these disadvantages by ensuring that the experiment done was as close to a real scenario as possible and using virtualisation software to simulate the networks require for testing to save on time and costs from using real physical network setups.[36]

3.2.4 Mixed Methods

If you were to make use of both quantitative and qualitative data, you'd be taking a mixed-methods approach. Keeping with the previous example, you may also want to assess how many people in a community eat specific types of pizza. For this, you could make use of a survey to collect quantitative data and then analyse the results statistically, producing quantitative results in addition to your qualitative ones.[33]

Mixed methods apply to this work via the testing of networks of which to find the differences in data transfer in terms of the amount sent followed by analysis of each set of data to find the required results statistically.

The advantages of the mixed methods approach are that it provides a broadening perspective and more available data to work with.

The disadvantages however are that it is time consuming compared to quantitative or qualitative studying approach with its requirements of collecting and analysing different types of data and can be potential confusing as multiple approaches are used in the same study.

The author worked around these disadvantages by setting up the experiment to where each of the tests could be done around the same time instead of each done separately and using documentation, tables etc to keep track of the multiple approaches being taken.[37]

3.2.5 Cross-Sectional

This is where data is collected at one point in time, so you wouldn't be gathering data to see how language changes, but rather what language exists at a snapshot point in time.

For example, if you wanted to study the language used in memes at a certain point in time, you'd make use of a cross-sectional time horizon.

The type of data collected could be qualitative, quantitative or a mix of both, as the focus is on the time of collection, not the data type.[33]

Cross-Sectional applies to this work as the research and testing done is more so in the current point in time rather than over time.

The advantages of the cross-sectional are that it is a more cost-effective approach to research with it not having the requirement for follow up work and can make developing a general hypothesis easier.

The disadvantages however are that information from causal relationships is impossible and has no way to measure incidence.[38]

3.2.6 Data Collection and Data Analysis

Specifically, this is where you'll:

- Decide on what data you'll collect and what data collection methods you'll use (for example, will you use a survey? Or perhaps one-on-one interviews?)
- Decide how you'll go about sampling the population (for example, snowball sampling, random sampling, convenience sampling, etc).
- Determine the type of data analysis you'll use to answer your research questions (such as content analysis or a statistical analysis like correlation).
- Set up the materials you'll be using for your study (such as writing up questions for a survey or interview)

What's important to note here is that these techniques and procedures need to align with all the other layers of the research onion – i.e., research philosophy, research approaches, research strategy, choices, and time horizon.

For example, you if you're adopting a deductive, quantitative research approach, it's unlikely that you'll use interviews to collect your data, as you'll want high-volume, numerical data (which surveys are far better suited to). So, you need to ensure that the decisions at each layer of your onion align with the rest, and most importantly, that they align with your research aims and objectives.[33]

Data collection and data analysis applies to this work via the collection of data via testing of the virtual networks followed by the analysis of said data.

3.3 Development Model

When it comes to the creation stage of the project, a development model must be used to ensure that all the required steps have been taken or investigated to ensure no major issues along the way of development.

To ensure smooth work progression of this stage of the project, it was decided that the Waterfall Model would be used as it was considered the most relatable and appropriate for this part of the project.

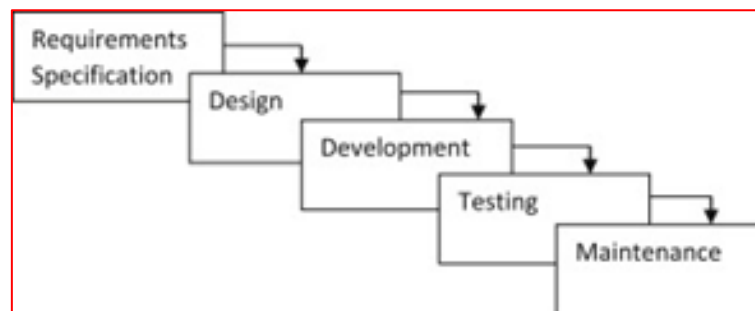


Figure 16: Waterfall Model Method (No deployment or maintenance)

3.3.1 Waterfall Model Method

The waterfall model is a model of software development in which tasks are done in a sequence manner where we start from the top with feasibility and flow down through various tasks with implementation into the live environment.

Its internal and external factors, which can be as follows:

- No ambiguous requirements in the application.
- Stability of product definition
- It is technology understood
- It is not dynamic
- Large resources with appropriate expertise are available to support the product.
- Vey short length project
- The good document, clear and fixed requirements.

The waterfall model was introduced in the year 1970 by Winston w Royce. Since then, the waterfall model state that one should switch to another phase only when the previous phases are completely tested, reviewed, and verified.

It emphasizes the logical progression of phase steps. Its functionality is like water that flows over the edge of a cliff. This approach of software development has been given the name waterfall because it develops systematically from one phase to another in a downward fashion. Some advantages of this model are that it has clearly defined stages, easy to arrange tasks and reinforces good habits etc.

While the disadvantages of this however are high amounts of risk, cannot accommodate changing requirements and is not the best model for use with more complex projects etc.[39]

3.4 Project Plan

To complete the project efficiently and in a timely fashion, a project plan was created using Project Professional as a Gantt Chart to display each part of the project as shown below.

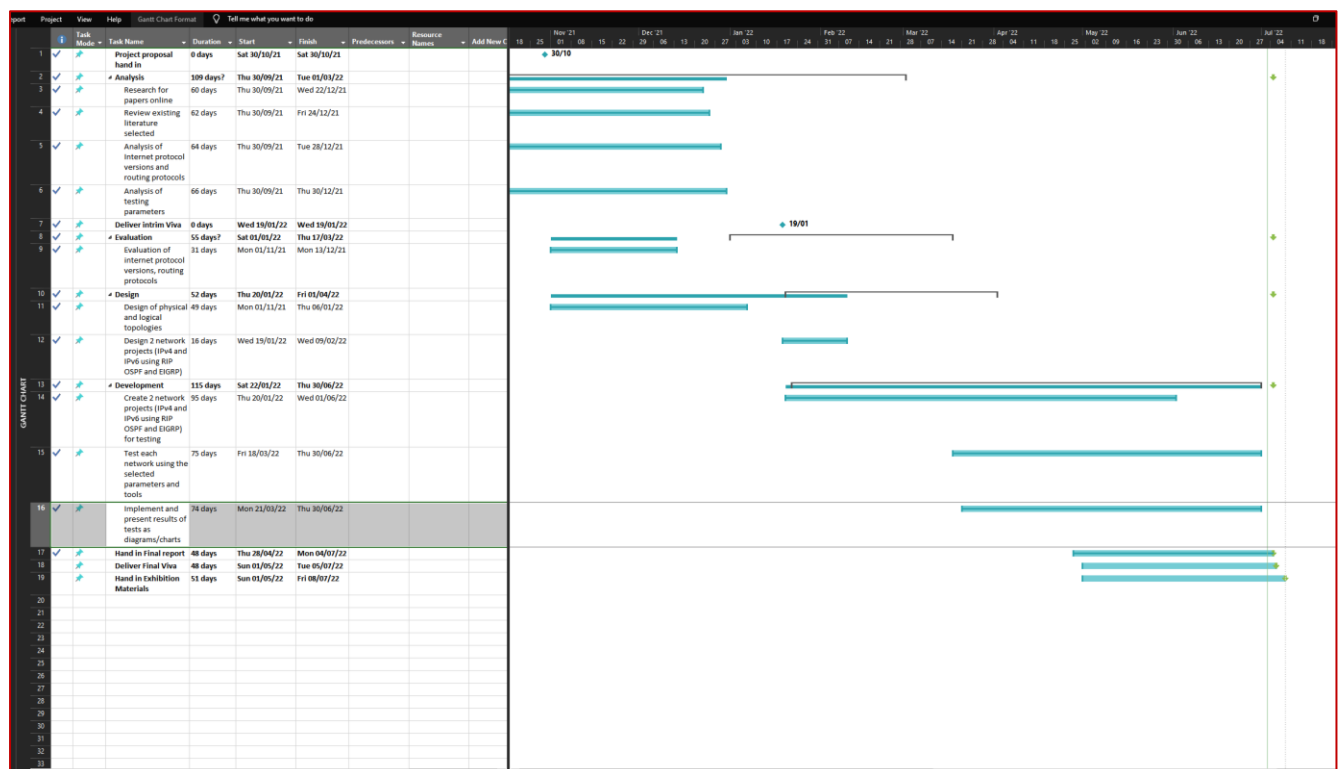


Figure 17: Gantt Chart

As seen from the Gantt Chart above, many of the objectives have already been completed except for the Final Viva and Poster as those are due after the upload point for the final report.

4. Design & Implementation

4.1 Design

Amongst the many design decisions which had to be made in relation to the experiment was the option of using Cisco Packet Tracer for the network simulation instead of GNS3 with Hyper V or VMware. This idea was scrapped early on due to its limitations with generating useful data for parameter comparison.

As for the virtualisation of the network operating systems or other systems, Virtual Box and Hyper V were considered and previously tested for their potential in this study with Hyper V being used in the beginning for the creation and testing of networks. However, due to compatibility issues with GNS3 as well as the lack of resources required in other areas these were both abandoned in favour of VMware Workstation Pro with GNS3 VM.

As far as routing protocols were concerned it was decided early on that it would be best to focus on a specific, well used selection, rather than more specialist protocols used for things like wireless networks specifically etc.

In relation to the design of the project for my network study or test, I decided to create 2 network projects (IPv4 and IPv6) in GNS3 that each contained 3 or more interconnected networks based around RIP, EIGRP or OSPF. The routing data was redistributed via the configuration on router R2. This design allowed the author more flexibility to test across one or all the routing-based protocols.

The main tools that ended up being used for the experiment were Iperf (Iperf2 for IPv6 and Iperf3 for IPv4) and Ostinato (Mostly for ICMP). Iperf uses both a client-side executable & a server-based executable. The vast majority of Iperf command switches are based at the client side of the testing.

It was originally planned to create an IPv4/6 Dual Stack setup and test it for performance differences however this was later scrapped due to time constraints and dual stack integration issues with the testing tools.

The testing parameters used for this experiment were the measurement of latency, end-to-end delay, packet loss, throughput, and jitter etc.

The design of the topology for each version of IP (IPv4 and IPv6) was changed several times throughout the experiment due to unexpected limits within the testing software.

IPv4 Table (Final Version)

Device	Interface	IP Address	Subnet Mask	Default Gateway
R1	F0/0	192.168.10.1	255.255.255.0	N/A
	F1/0	192.168.1.1	255.255.255.0	
	F1/1	192.168.7.1	255.255.255.0	
	S2/0	10.1.1.1	255.255.255.0	

R2	S2/0	10.1.1.2	255.255.255.0	N/A
	S2/1	10.2.2.2	255.255.255.0	
	S2/2	10.3.3.2	255.255.255.0	
	S2/3	10.4.4.2	255.255.255.0	
R3	F1/0	192.168.3.1	255.255.255.0	N/A
	S2/0	10.3.3.1	255.255.255.0	
R4	F1/1	192.168.2.1	255.255.255.0	N/A
	S2/0	10.2.2.1	255.255.255.0	
R5	F1/0	192.168.4.1	255.255.255.0	N/A
	F1/1	10.5.5.2	255.255.255.0	
	S2/0	10.4.4.1	255.255.255.0	
R6	E1/0 NOT IN USE!	192.168.5.1	255.255.255.0	N/A
	F1/1	10.6.6.2	255.255.255.0	
	F1/0	10.5.5.1	255.255.255.0	
R7	F1/0	10.6.6.1	255.255.255.0	N/A
	F1/1	192.168.5.1	255.255.255.0	
WIN10PCQEMU-1-C1C3	NIC	192.168.1.100	255.255.255.0	192.168.1.1
Ostinato1.1-1	NIC	192.168.1.101	255.255.255.0	192.168.1.1
WIN10PCQEMU-2-S2	NIC	192.168.2.100	255.255.255.0	192.168.2.1
WIN10PCQEMU-C2	NIC	192.168.3.100	255.255.255.0	192.168.3.1
Ostinato(std)1.1-4	NIC	192.168.3.101	255.255.255.0	192.168.3.1
WIN10PCQEMU-C3	NIC	192.168.4.100	255.255.255.0	192.168.4.1
Ostinato(std)1.1-6	NIC	192.168.4.101	255.255.255.0	192.168.4.1
WIN10PCQEMU-S3	NIC	192.168.5.100	255.255.255.0	192.168.5.1
NOT IN USE!	NIC	192.168.6.100	255.255.255.0	192.168.6.1
WIN10PCQEMU-2-S1	NIC	192.168.7.100	255.255.255.0	192.168.7.1

WIN10PC1-1	NIC	192.168.10.120	255.255.255.0	192.168.10.1
------------	-----	----------------	---------------	--------------

IPv6 Table (Final Version)

Device	Interface	IPv6 Address/Prefix	Default Gateway
R1	F0/0	2001:db8:1::1/64	N/A
	F1/0	2001:db8:2::1/64	
	F1/1	2001:db8:3::1/64	
	S2/0	2001:db8:4::1/64	
R2	S2/0	2001:db8:4::2/64	N/A
	S2/1	2001:db8:5::2/64	
	S2/2	2001:db8:6::2/64	
	S2/3	2001:db8:7::2/64	
R3	F1/0	2001:db8:d::1/64	N/A
	S2/0	2001:db8:6::1/64	
R4	F1/1	2001:db8:9::1/64	N/A
	S2/0	2001:db8:5::1/64	
R5	F1/0	2001:db8:c::1/64	N/A
	F1/1	2001:db8:8::1/64	
	S2/0	2001:db8:7::1/64	
R6	E1/0 NOT IN USE!	2001:db8:c::1/64	N/A
	F1/1	2001:db8:a::1/64	
	F1/0	2001:db8:8::2/64	
R7	F1/0	2001:db8:a::2/64	N/A
	F1/1	2001:db8:b::1/64	
WIN10PCQEMU-1-C1C3	NIC	2001:db8:2::a/64	2001:db8:2::1/64
Ostinato1.1-1	NIC	2001:db8:2::b/64	2001:db8:2::1/64
WIN10PCQEMU-S2	NIC	2001:db8:9::a/64	2001:db8:9::1/64
WIN10PCQEMU-C2	NIC	2001:db8:d::a/64	2001:db8:d::1/64
Ostinato(std)1.1-4	NIC	2001:db8:d::b/64	2001:db8:d::1/64
WIN10PCQEMU-C3	NIC	2001:db8:c::a/64	2001:db8:c::1/64
Ostinato(std)1.1-6	NIC	2001:db8:c::b/64	2001:db8:c::1/64
WIN10PCQEMU-S3	NIC	2001:db8:b::a/64	2001:db8:b::1/64
WIN10PCQEMU-2-S1	NIC	2001:db8:3::a/64	2001:db8:3::1/64
WIN10PC1-1	NIC	2001:db8:1::a/64	2001:db8:1::1/64

4.1.1 Evidence of testing

Below are some examples of actual test-based screenshots.

These screenshots provide a better idea as to how these tests were carried out and how they were used for gathering the results.

IPv4 Results

In the following tests the packet size was initially going to be varied using the IPerf3 command switch `-M` for TCP & `-l` for UDP. However, after some initial checks with Wireshark it became obvious that the command switch `-M` was not operational.

After some further research it was decided to use the Cisco router command allocated to each interface in the test path i.e. "IP TCP ADJUST-MSS <MSS VALUE>". This provided the necessary payload limit for each test but was far more time consuming to setup. It also had the limitation that it only worked for the TCP protocol.

In contrast the IPerf packet sizing command switch `-l` worked correctly for UDP testing & so the author managed to save a substantial amount of configuration time using this.

In the tests detailed below the following describes each Iperf switch;

- c executes the client side
 - u executes a UDP test (default being TCP)
 - l configures the MSS size for that specific test
- The target port is the default of 5201 – this can be changed via a command switch `-p`

The server command line is not detailed with a screen shot as the command was merely Iperf3 `-s`.

ICMP tests were performed by using Ostinato which had a friendly GUI allowing easy changes to be made to the packet size for each ICMP test. Wireshark was used to capture the test packets & used to measure the throughput.

```
C:\Apps\iperf-3.1.3-win64>iperf3 -c 192.168.7.100
Connecting to host 192.168.7.100, port 5201
[ 4] local 192.168.1.100 port 49676 connected to 192.168.7.100 port 5201
[ ID] Interval           Transfer     Bandwidth
[ 4]  0.00-1.00   sec   1.00 MBytes   8.00 Mbits/sec
[ 4]  1.00-2.00   sec   1.12 MBytes   9.81 Mbits/sec
[ 4]  2.00-3.00   sec   1.25 MBytes  10.5 Mbits/sec
[ 4]  3.00-4.00   sec   1.38 MBytes  11.5 Mbits/sec
[ 4]  4.00-5.00   sec   1.38 MBytes  11.5 Mbits/sec
[ 4]  5.00-6.00   sec   1.38 MBytes  11.5 Mbits/sec
[ 4]  6.00-7.00   sec   1.25 MBytes  10.5 Mbits/sec
[ 4]  7.00-8.00   sec   1.38 MBytes  11.5 Mbits/sec
[ 4]  8.00-9.00   sec   1.38 MBytes  11.6 Mbits/sec
[ 4]  9.00-10.00  sec   1.25 MBytes  10.4 Mbits/sec
- - - - -
[ ID] Interval           Transfer     Bandwidth
[ 4]  0.00-10.01  sec  12.8 MBytes  10.7 Mbits/sec
[ 4]  0.00-10.01  sec  12.7 MBytes  10.6 Mbits/sec
iperf Done.
```

Figure 18: Test 1 RIP TCP MSS 1460 Iperf

Includes a Wireshark screen shot below to indicate the UDP test payload size – in this case 1460 Bytes as expected.

26	2022-06-29	11:05:46.622936	192.168.1.100	192.168.7.100	UDP	1502	49195 → 5201	Len=1460
27	2022-06-29	11:05:46.622936	192.168.1.100	192.168.7.100	UDP	1502	49195 → 5201	Len=1460
28	2022-06-29	11:05:46.622936	192.168.1.100	192.168.7.100	UDP	1502	49195 → 5201	Len=1460
29	2022-06-29	11:05:46.622936	192.168.1.100	192.168.7.100	UDP	1502	49195 → 5201	Len=1460
30	2022-06-29	11:05:46.622936	192.168.1.100	192.168.7.100	UDP	1502	49195 → 5201	Len=1460
31	2022-06-29	11:05:46.622936	192.168.1.100	192.168.7.100	UDP	1502	49195 → 5201	Len=1460
32	2022-06-29	11:05:46.622936	192.168.1.100	192.168.7.100	UDP	1502	49195 → 5201	Len=1460
33	2022-06-29	11:05:46.622936	192.168.1.100	192.168.7.100	UDP	1502	49195 → 5201	Len=1460
34	2022-06-29	11:05:46.638167	192.168.1.100	192.168.7.100	UDP	1502	49195 → 5201	Len=1460
35	2022-06-29	11:05:46.717111	192.168.1.100	192.168.7.100	UDP	1502	49195 → 5201	Len=1460
36	2022-06-29	11:05:46.717111	192.168.1.100	192.168.7.100	UDP	1502	49195 → 5201	Len=1460
37	2022-06-29	11:05:46.717111	192.168.1.100	192.168.7.100	UDP	1502	49195 → 5201	Len=1460
38	2022-06-29	11:05:46.717111	192.168.1.100	192.168.7.100	UDP	1502	49195 → 5201	Len=1460
39	2022-06-29	11:05:46.717111	192.168.1.100	192.168.7.100	UDP	1502	49195 → 5201	Len=1460
40	2022-06-29	11:05:46.717111	192.168.1.100	192.168.7.100	UDP	1502	49195 → 5201	Len=1460
41	2022-06-29	11:05:46.733096	192.168.1.100	192.168.7.100	UDP	1502	49195 → 5201	Len=1460
42	2022-06-29	11:05:46.733096	192.168.1.100	192.168.7.100	UDP	1502	49195 → 5201	Len=1460
43	2022-06-29	11:05:46.734065	192.168.1.100	192.168.7.100	UDP	1502	49195 → 5201	Len=1460
44	2022-06-29	11:05:46.842579	192.168.1.100	192.168.7.100	UDP	1502	49195 → 5201	Len=1460
45	2022-06-29	11:05:46.842579	192.168.1.100	192.168.7.100	UDP	1502	49195 → 5201	Len=1460
46	2022-06-29	11:05:46.842579	192.168.1.100	192.168.7.100	UDP	1502	49195 → 5201	Len=1460
47	2022-06-29	11:05:46.842579	192.168.1.100	192.168.7.100	UDP	1502	49195 → 5201	Len=1460
48	2022-06-29	11:05:46.842579	192.168.1.100	192.168.7.100	UDP	1502	49195 → 5201	Len=1460

<

> Frame 37: 1502 bytes on wire (12016 bits), 1502 bytes captured (12016 bits) on interface -, id 0
 > Ethernet II, Src: ca:04:35:44:00:1d (ca:04:35:44:00:1d), Dst: 0c:36:55:2e:00:00 (0c:36:55:2e:00:00)
 > Internet Protocol Version 4, Src: 192.168.1.100, Dst: 192.168.7.100
 > User Datagram Protocol, Src Port: 49195, Dst Port: 5201
 ✓ Data (1460 bytes)
 Data: 62bc947a0004cdf70000000dd6c48b09f7defaad02231f336083d5b56b92057675740429...
 [Length: 1460]

Figure 19: Test 1 RIP UDP MSS 1460 Wireshark

```
C:\Apps\iperf-3.1.3-win64>iperf3 -c 192.168.7.100 -u -l 1460
Connecting to host 192.168.7.100, port 5201
[ 4] local 192.168.1.100 port 49195 connected to 192.168.7.100 port 5201
[ ID] Interval           Transfer     Bandwidth       Total Datagrams
[ 4] 0.00-1.01 sec      117 KBytes   951 Kbits/sec    82
[ 4] 1.01-2.02 sec      130 KBytes   1.05 Mbits/sec   91
[ 4] 2.02-3.02 sec      138 KBytes   1.13 Mbits/sec   97
[ 4] 3.02-4.01 sec      117 KBytes   973 Kbits/sec    82
[ 4] 4.01-5.00 sec      134 KBytes   1.10 Mbits/sec   94
[ 4] 5.00-6.01 sec      121 KBytes   987 Kbits/sec    85
[ 4] 6.01-7.01 sec      128 KBytes   1.05 Mbits/sec   90
[ 4] 7.01-8.01 sec      140 KBytes   1.14 Mbits/sec   98
[ 4] 8.01-9.01 sec      117 KBytes   955 Kbits/sec    82
[ 4] 9.01-10.01 sec     127 KBytes   1.04 Mbits/sec   89
- - - - -
[ ID] Interval           Transfer     Bandwidth       Jitter          Lost/Total Datagrams
[ 4] 0.00-10.01 sec     1.24 MBytes   1.04 Mbits/sec   0.923 ms        0/889 (0%)
[ 4] Sent 889 datagrams

iperf Done.
```

Figure 20: Test 1 RIP UDP MSS 1460 Iperf

```

C:\Apps\iperf-3.1.3-win64>iperf3 -c 192.168.7.100
Connecting to host 192.168.7.100, port 5201
[ 4] local 192.168.1.100 port 49685 connected to 192.168.7.100 port 5201
[ ID] Interval           Transfer     Bandwidth
[ 4]  0.00-1.03      sec   640 KBytes  5.09 Mbits/sec
[ 4]  1.03-2.03      sec   896 KBytes  7.34 Mbits/sec
[ 4]  2.03-3.01      sec   896 KBytes  7.46 Mbits/sec
[ 4]  3.01-4.01      sec   896 KBytes  7.34 Mbits/sec
[ 4]  4.01-5.01      sec   896 KBytes  7.34 Mbits/sec
[ 4]  5.01-6.02      sec   896 KBytes  7.31 Mbits/sec
[ 4]  6.02-7.01      sec   896 KBytes  7.37 Mbits/sec
[ 4]  7.01-8.01      sec   768 KBytes  6.29 Mbits/sec
[ 4]  8.01-9.00      sec   640 KBytes  5.32 Mbits/sec
[ 4]  9.00-10.01     sec   1.00 MBytes  8.34 Mbits/sec
- - - - -
[ ID] Interval           Transfer     Bandwidth
[ 4]  0.00-10.01     sec   8.25 MBytes  6.92 Mbits/sec
[ 4]  0.00-10.01     sec   8.09 MBytes  6.78 Mbits/sec
iperf Done.

```

Figure 21: Test 1 RIP TCP MSS 960 Iperf

```

C:\Apps\iperf-3.1.3-win64>iperf3 -c 192.168.7.100
Connecting to host 192.168.7.100, port 5201
[ 4] local 192.168.1.100 port 49687 connected to 192.168.7.100 port 5201
[ ID] Interval           Transfer     Bandwidth
[ 4]  0.00-1.01      sec   512 KBytes  4.15 Mbits/sec
[ 4]  1.01-2.01      sec   512 KBytes  4.21 Mbits/sec
[ 4]  2.01-3.01      sec   512 KBytes  4.17 Mbits/sec
[ 4]  3.01-4.02      sec   512 KBytes  4.16 Mbits/sec
[ 4]  4.02-5.02      sec   512 KBytes  4.19 Mbits/sec
[ 4]  5.02-6.01      sec   384 KBytes  3.19 Mbits/sec
[ 4]  6.01-7.02      sec   384 KBytes  3.10 Mbits/sec
[ 4]  7.02-8.01      sec   512 KBytes  4.26 Mbits/sec
[ 4]  8.01-9.02      sec   512 KBytes  4.13 Mbits/sec
[ 4]  9.02-10.01     sec   512 KBytes  4.26 Mbits/sec
- - - - -
[ ID] Interval           Transfer     Bandwidth
[ 4]  0.00-10.01     sec   4.75 MBytes  3.98 Mbits/sec
[ 4]  0.00-10.01     sec   4.62 MBytes  3.87 Mbits/sec
iperf Done.

```

Figure 22: Test 1 RIP TCP MSS 536 Iperf

Wireshark shows the throughput for all ICMP tests initiated using Ostinato as the example below shows.

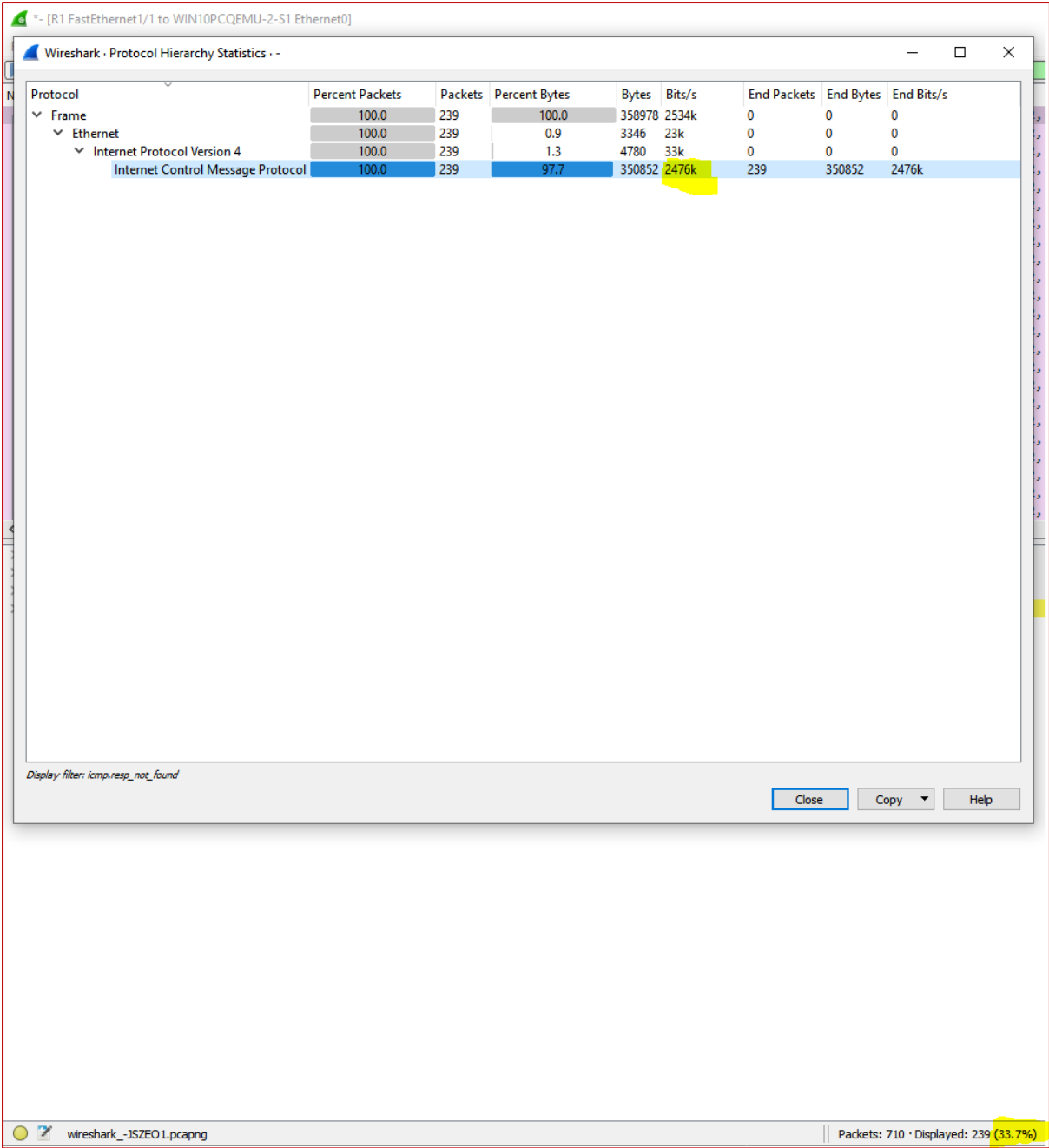


Figure 23: Test 1 RIP ICMP MSS 1460 Wireshark

```

C:\Apps\iperf-3.1.3-win64>iperf3 -c 192.168.2.100
Connecting to host 192.168.2.100, port 5201
[ 4] local 192.168.3.100 port 49676 connected to 192.168.2.100 port 5201
[ ID] Interval           Transfer     Bandwidth
[ 4] 0.00-1.01      sec    256 KBytes    2.07 Mbits/sec
[ 4] 1.01-2.01      sec    128 KBytes    1.05 Mbits/sec
[ 4] 2.01-3.03      sec     0.00 Bytes    0.00 bits/sec
[ 4] 3.03-4.03      sec     0.00 Bytes    0.00 bits/sec
[ 4] 4.03-5.01      sec     0.00 Bytes    0.00 bits/sec
[ 4] 5.01-6.01      sec    128 KBytes    1.05 Mbits/sec
[ 4] 6.01-7.03      sec    128 KBytes    1.03 Mbits/sec
[ 4] 7.03-8.01      sec    128 KBytes    1.07 Mbits/sec
[ 4] 8.01-9.01      sec     0.00 Bytes    0.00 bits/sec
[ 4] 9.01-10.01     sec     0.00 Bytes    0.00 bits/sec
- - - - -
[ ID] Interval           Transfer     Bandwidth
[ 4] 0.00-10.01     sec    768 KBytes    628 Kbits/sec
[ 4] 0.00-10.01     sec    615 KBytes    503 Kbits/sec
iperf Done.

```

Figure 24: Test 2 OSPF TCP MSS 1460 Iperf

```

C:\Apps\iperf-3.1.3-win64>iperf3 -c 192.168.2.100
Connecting to host 192.168.2.100, port 5201
[ 4] local 192.168.3.100 port 49678 connected to 192.168.2.100 port 5201
[ ID] Interval           Transfer     Bandwidth
[ 4] 0.00-1.03      sec    256 KBytes    2.04 Mbits/sec
[ 4] 1.03-2.01      sec     0.00 Bytes    0.00 bits/sec
[ 4] 2.01-3.01      sec     0.00 Bytes    0.00 bits/sec
[ 4] 3.01-4.01      sec     0.00 Bytes    0.00 bits/sec
[ 4] 4.01-5.01      sec     0.00 Bytes    0.00 bits/sec
[ 4] 5.01-6.03      sec    128 KBytes    1.03 Mbits/sec
[ 4] 6.03-7.01      sec    128 KBytes    1.06 Mbits/sec
[ 4] 7.01-8.03      sec     0.00 Bytes    0.00 bits/sec
[ 4] 8.03-9.01      sec    128 KBytes    1.07 Mbits/sec
[ 4] 9.01-10.01     sec     0.00 Bytes    0.00 bits/sec
- - - - -
[ ID] Interval           Transfer     Bandwidth
[ 4] 0.00-10.01     sec    640 KBytes    524 Kbits/sec
[ 4] 0.00-10.01     sec    452 KBytes    370 Kbits/sec

```

Figure 25: Test 2 OSPF TCP MSS 960 Iperf

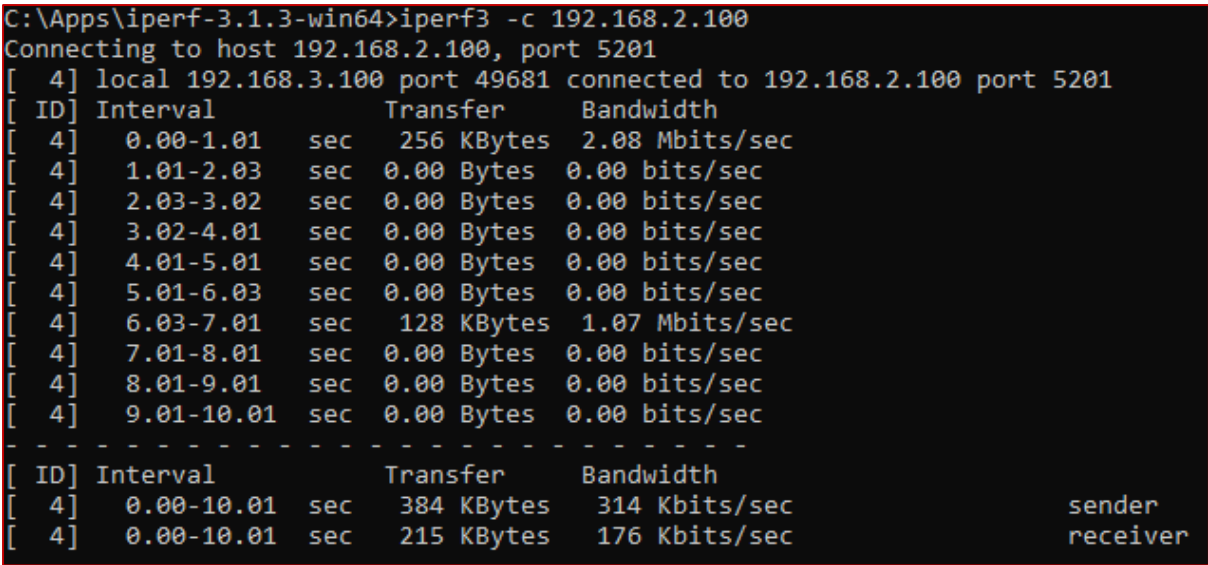


Figure 26: Test 2 OSPF TCP MSS 536 Iperf

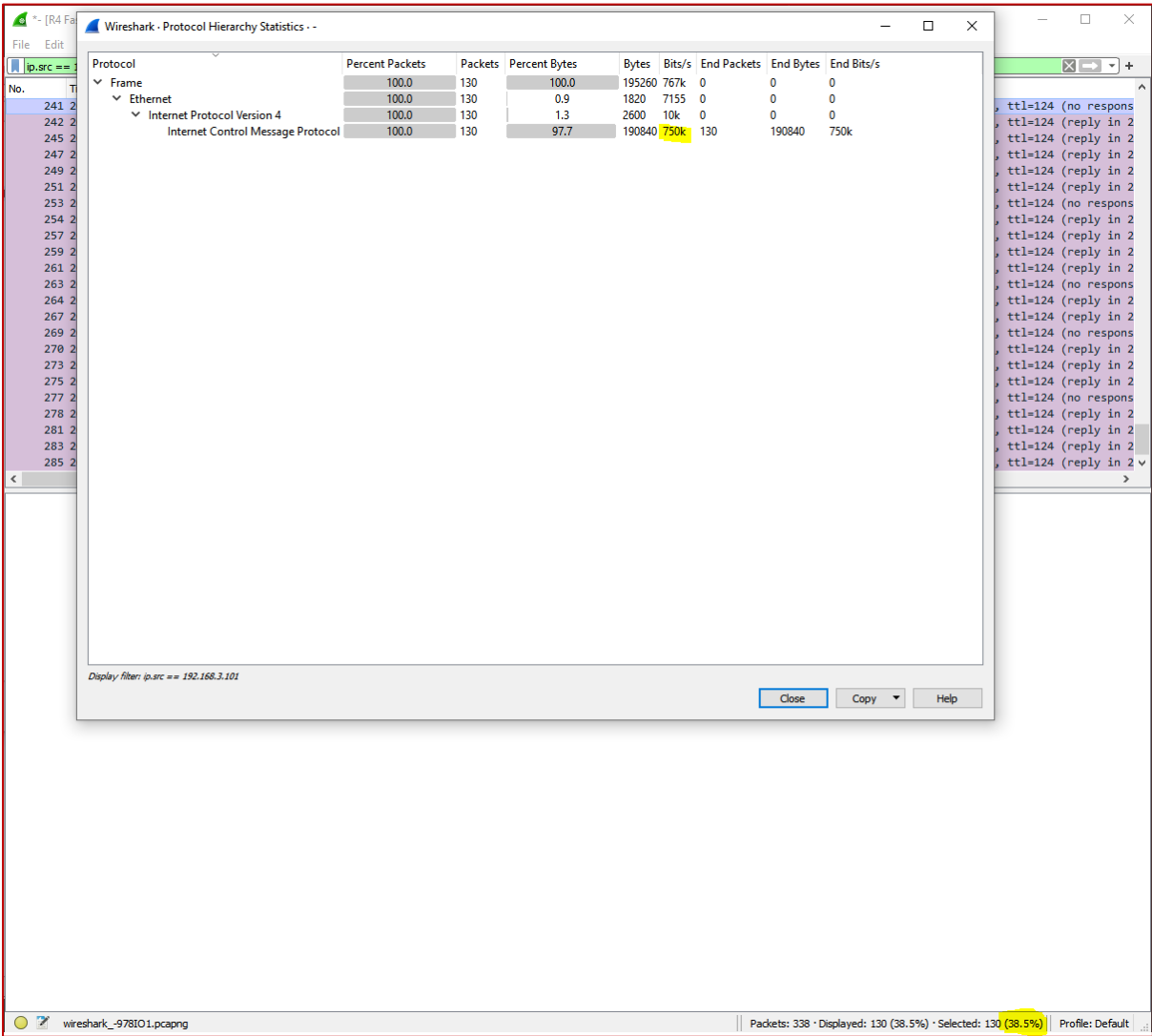


Figure 27: Test 2 OSPF ICMP MSS 1460 Wireshark

```

C:\Apps\iperf-3.1.3-win64>iperf3 -c 192.168.5.100
Connecting to host 192.168.5.100, port 5201
[ 4] local 192.168.4.100 port 49676 connected to 192.168.5.100 port 5201
[ ID] Interval           Transfer     Bandwidth
[ 4]  0.00-1.00   sec      896 KBytes    7.33 Mbits/sec
[ 4]  1.00-2.00   sec     1.25 MBytes   10.5 Mbits/sec
[ 4]  2.00-3.02   sec      384 KBytes    3.10 Mbits/sec
[ 4]  3.02-4.00   sec     1.12 MBytes    9.58 Mbits/sec
[ 4]  4.00-5.00   sec     1.00 MBytes    8.39 Mbits/sec
[ 4]  5.00-6.02   sec      896 KBytes    7.23 Mbits/sec
[ 4]  6.02-7.02   sec     1.00 MBytes    8.38 Mbits/sec
[ 4]  7.02-8.00   sec     1.00 MBytes    8.51 Mbits/sec
[ 4]  8.00-9.00   sec      896 KBytes    7.35 Mbits/sec
[ 4]  9.00-10.00  sec       768 KBytes    6.29 Mbits/sec
- - - - -
[ ID] Interval           Transfer     Bandwidth
[ 4]  0.00-10.00  sec     9.12 MBytes    7.65 Mbits/sec  sender
[ 4]  0.00-10.00  sec     9.06 MBytes    7.60 Mbits/sec  receiver

```

Figure 28: Test 3 EIGRP TCP MSS 1460 Iperf

```

C:\Apps\iperf-3.1.3-win64>iperf3 -c 192.168.5.100
Connecting to host 192.168.5.100, port 5201
[ 4] local 192.168.4.100 port 49680 connected to 192.168.5.100 port 5201
[ ID] Interval           Transfer     Bandwidth
[ 4]  0.00-1.00   sec      640 KBytes    5.22 Mbits/sec
[ 4]  1.00-2.00   sec      256 KBytes    2.10 Mbits/sec
[ 4]  2.00-3.00   sec      640 KBytes    5.24 Mbits/sec
[ 4]  3.00-4.00   sec      896 KBytes    7.34 Mbits/sec
[ 4]  4.00-5.01   sec     1.00 MBytes    8.31 Mbits/sec
[ 4]  5.01-6.02   sec      896 KBytes    7.27 Mbits/sec
[ 4]  6.02-7.01   sec      896 KBytes    7.46 Mbits/sec
[ 4]  7.01-8.00   sec      896 KBytes    7.36 Mbits/sec
[ 4]  8.00-9.00   sec      896 KBytes    7.34 Mbits/sec
[ 4]  9.00-10.00  sec      896 KBytes    7.34 Mbits/sec
- - - - -
[ ID] Interval           Transfer     Bandwidth
[ 4]  0.00-10.00  sec     7.75 MBytes    6.50 Mbits/sec  sender
[ 4]  0.00-10.00  sec     7.56 MBytes    6.34 Mbits/sec  receiver

iperf Done.

C:\Apps\iperf-3.1.3-win64>

```

Figure 29: Test 3 EIGRP TCP MSS 960 Iperf

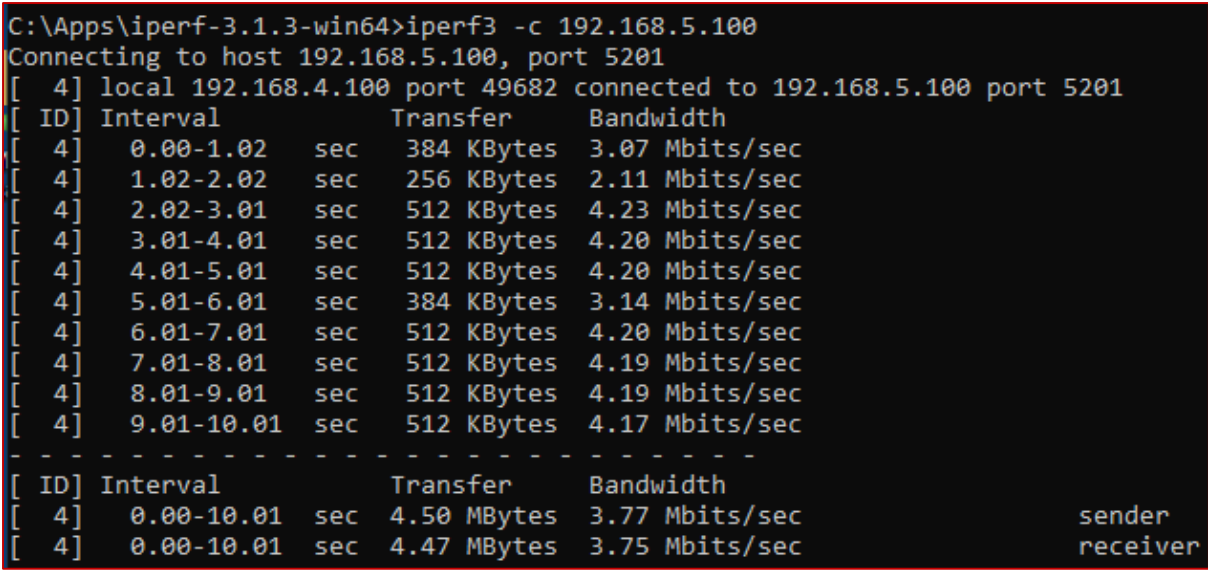


Figure 30: Test 3 EIGRP TCP MSS 536 Iperf

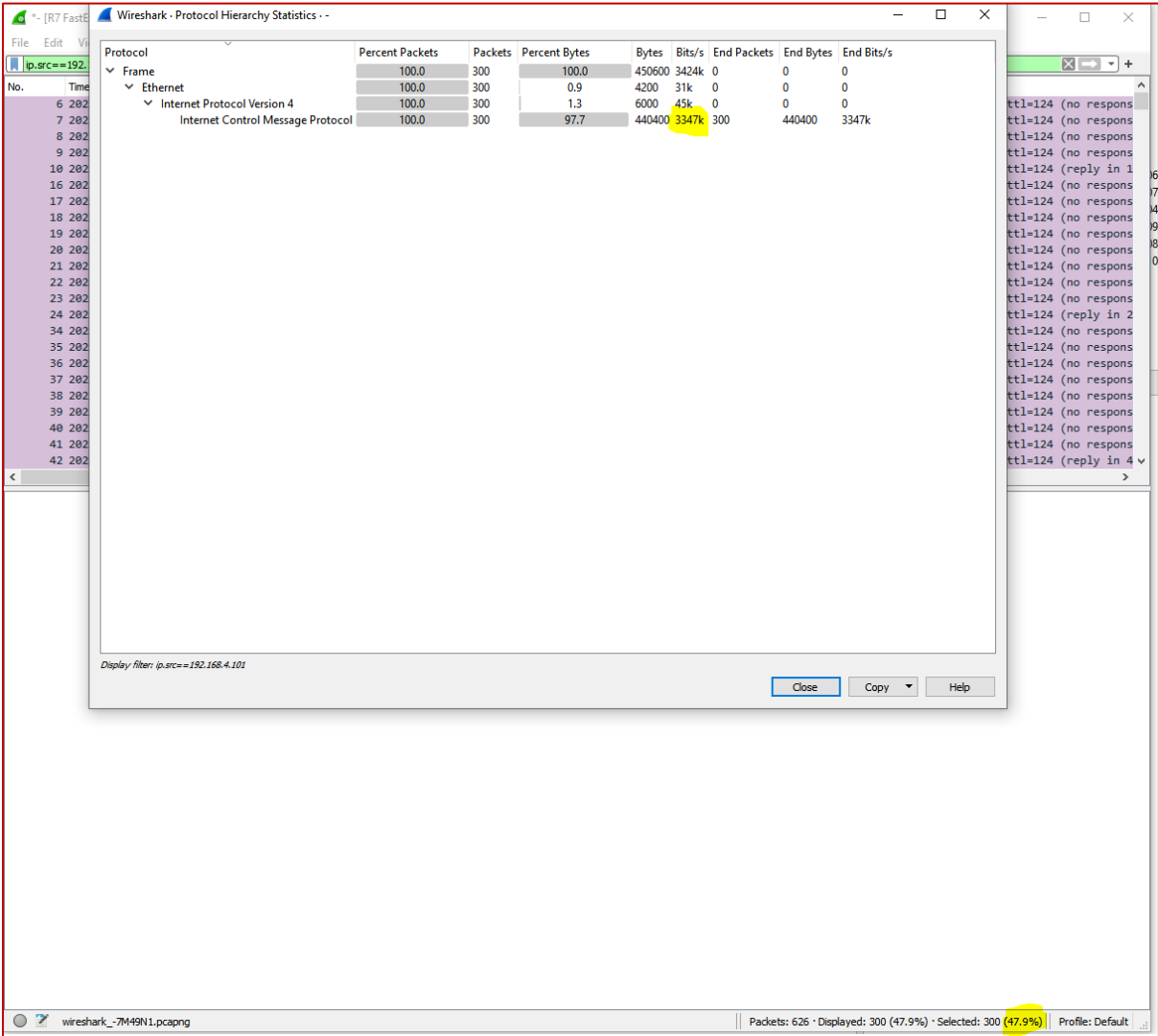


Figure 31: Test 3 EIGRP ICMP MSS 1460 Wireshark

```

C:\Apps\iperf-3.1.3-win64>iperf3 -c 192.168.5.100
Connecting to host 192.168.5.100, port 5201
[ 4] local 192.168.1.100 port 49676 connected to 192.168.5.100 port 5201
[ ID] Interval           Transfer     Bandwidth
[ 4]  0.00-1.00    sec    256 KBytes    2.09 Mbits/sec
[ 4]  1.00-2.02    sec    128 KBytes    1.03 Mbits/sec
[ 4]  2.02-3.01    sec     0.00 Bytes    0.00 bits/sec
[ 4]  3.01-4.01    sec     0.00 Bytes    0.00 bits/sec
[ 4]  4.01-5.01    sec     0.00 Bytes    0.00 bits/sec
[ 4]  5.01-6.01    sec     0.00 Bytes    0.00 bits/sec
[ 4]  6.01-7.01    sec    128 KBytes    1.05 Mbits/sec
[ 4]  7.01-8.01    sec    128 KBytes    1.05 Mbits/sec
[ 4]  8.01-9.01    sec     0.00 Bytes    0.00 bits/sec
[ 4]  9.01-10.01   sec     0.00 Bytes    0.00 bits/sec
- - - - -
[ ID] Interval           Transfer     Bandwidth
[ 4]  0.00-10.01   sec    640 KBytes    524 Kbits/sec
[ 4]  0.00-10.01   sec    493 KBytes    404 Kbits/sec
sender
receiver

```

Figure 32: Test 4 RIP, OSPF, EIGRP TCP MSS 1460 Iperf

```

C:\Apps\iperf-3.1.3-win64>iperf3 -c 192.168.5.100
Connecting to host 192.168.5.100, port 5201
[ 4] local 192.168.1.100 port 49676 connected to 192.168.5.100 port 5201
[ ID] Interval           Transfer     Bandwidth
[ 4]  0.00-1.00    sec    256 KBytes    2.09 Mbits/sec
[ 4]  1.00-2.00    sec     0.00 Bytes    0.00 bits/sec
[ 4]  2.00-3.01    sec     0.00 Bytes    0.00 bits/sec
[ 4]  3.01-4.02    sec     0.00 Bytes    0.00 bits/sec
[ 4]  4.02-5.00    sec     0.00 Bytes    0.00 bits/sec
[ 4]  5.00-6.02    sec    128 KBytes    1.03 Mbits/sec
[ 4]  6.02-7.00    sec     0.00 Bytes    0.00 bits/sec
[ 4]  7.00-8.00    sec    128 KBytes    1.05 Mbits/sec
[ 4]  8.00-9.01    sec     0.00 Bytes    0.00 bits/sec
[ 4]  9.01-10.00   sec    128 KBytes    1.05 Mbits/sec
- - - - -
[ ID] Interval           Transfer     Bandwidth
[ 4]  0.00-10.00   sec    640 KBytes    524 Kbits/sec
[ 4]  0.00-10.00   sec    406 KBytes    332 Kbits/sec
sender
receiver

```

Figure 33: Test 4 RIP, OSPF, EIGRP TCP MSS 960 Iperf

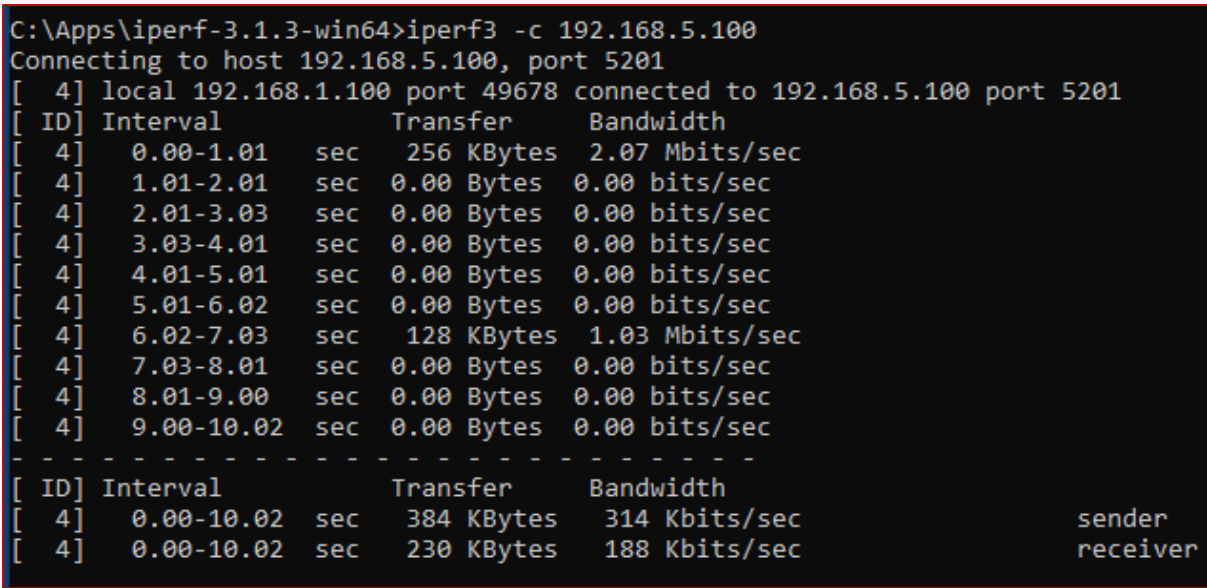


Figure 34: Test 4 RIP, OSPF, EIGRP TCP MSS 536 Iperf

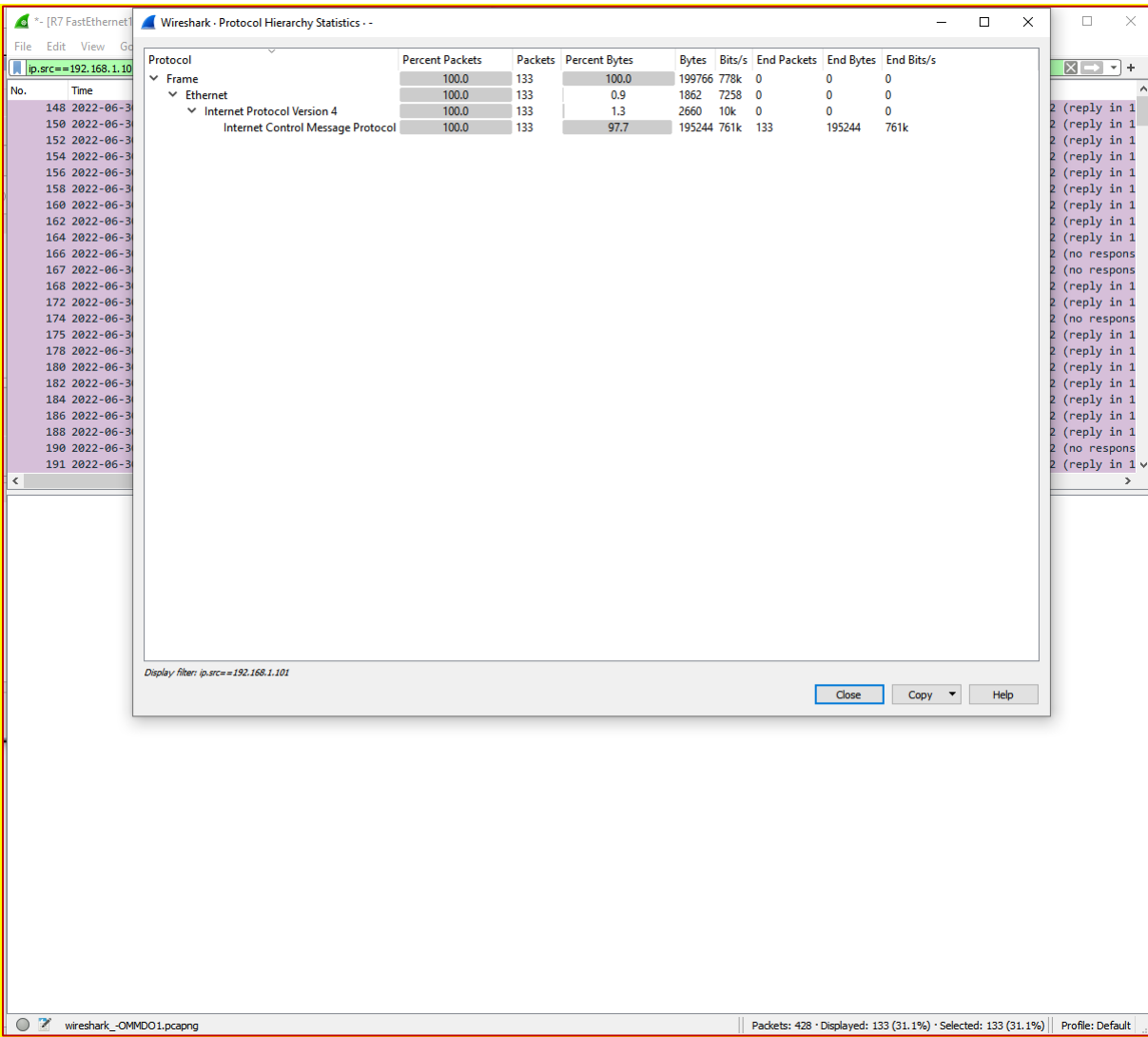


Figure 35: Test 4 RIP, OSPF, EIGRP ICMP MSS 1460 Wireshark

IPv6 Results

IPv6 testing offered many more challenges in relation to the use of the selected test tools. The Cisco interface TCP packet size command used with IPv4 does not work with IPv6 & there is no IPv6 version. This required the author to use a local netsh command line on the source test PC to configure and vary the MTU size. The range of MTU sizes was very limited i.e. 1280-1500.

Iperf command line switches with IPv6 and are as follows;

- c executes the client side
- V executes an IPv6 test.
- u executes a UDP test (default being TCP)
- l configures the MSS size for that specific test
- e shows additional pps information.

The target port is the default of 5001 – this can be changed via a command switch –p

ICMP tests were performed by using Ostinato which had a friendly GUI allowing easy changes to be made to the packet size for each ICMP test. Wireshark was used to capture the test packets & used to measure the throughput.

```
C:\Apps\iperf-2.0.9-win64>netsh interface ipv6 show subinterface
```

MTU	MediaSenseState	Bytes In	Bytes Out	Interface
4294967295		1	0	152 Loopback Pseudo-Interface 1
1500	1	115118	15736398	Ethernet

```
C:\Apps\iperf-2.0.9-win64>iperf -c 2001:db8:3::a -V
```

```
-----
Client connecting to 2001:db8:3::a, TCP port 5001
TCP window size: 208 KByte (default)
-----
[ 3] local 2001:db8:2::a port 49678 connected with 2001:db8:3::a port 5001
[ ID] Interval      Transfer      Bandwidth
[ 3] 0.0-10.0 sec  14.0 MBytes  11.7 Mbits/sec
```

Figure 36: Test 1 RIP TCP MSS 1440 Iperf

```

C:\Apps\iperf-2.0.9-win64>netsh interface ipv6 show subinterface

    MTU  MediaSenseState    Bytes In  Bytes Out  Interface
-----
4294967295      1          1          0    152 Loopback Pseudo-Interface 1
1280          1      150106    16037835    Ethernet

C:\Apps\iperf-2.0.9-win64>iperf -c 2001:db8:3::a -V
-----
Client connecting to 2001:db8:3::a, TCP port 5001
TCP window size: 208 KByte (default)
-----
[  3] local 2001:db8:2::a port 49674 connected with 2001:db8:3::a port 5001
[ ID] Interval        Transfer     Bandwidth
[  3] 0.0-10.0 sec    11.2 MBytes  9.41 Mbits/sec

C:\Apps\iperf-2.0.9-win64>pause
Press any key to continue . . .

```

Figure 37: Test 1 RIP TCP MSS 1220 Iperf

```

C:\Apps\iperf-2.0.9-win64>iperf -c 2001:db8:3::a -V -u -l 1440 -e
-----
Client connecting to 2001:db8:3::a, UDP port 5001 with pid 1716
Sending 1440 byte datagrams, IPG target: 10986.33 us (kalman adjust)
UDP buffer size: 208 KByte (default)
-----
[  3] local 2001:db8:2::a port 58882 connected with 2001:db8:3::a port 5001
[ ID] Interval        Transfer     Bandwidth      PPS
[  3] 0.00-10.02 sec    1.25 MBytes  1.05 Mbits/sec  90 pps
[  3] Sent 911 datagrams

```

Figure 38: Test 1 RIP UDP MSS 1440 Iperf

To note that not all possible tests were screenshotted and displayed here as evidence due to time constraints.

4.1.2 Network Topologies

During the project, many versions of the networks were created due to scale of changes made over time to accommodate for the testing and analysis done later.

4.1.2.1 IPv4 Network

The first network created was a IPv4 network as this was the easiest for the author to start off with and trouble shoot the designs before moving on to testing and analysis.

IPv4 Version 1 Network Logical Topology:

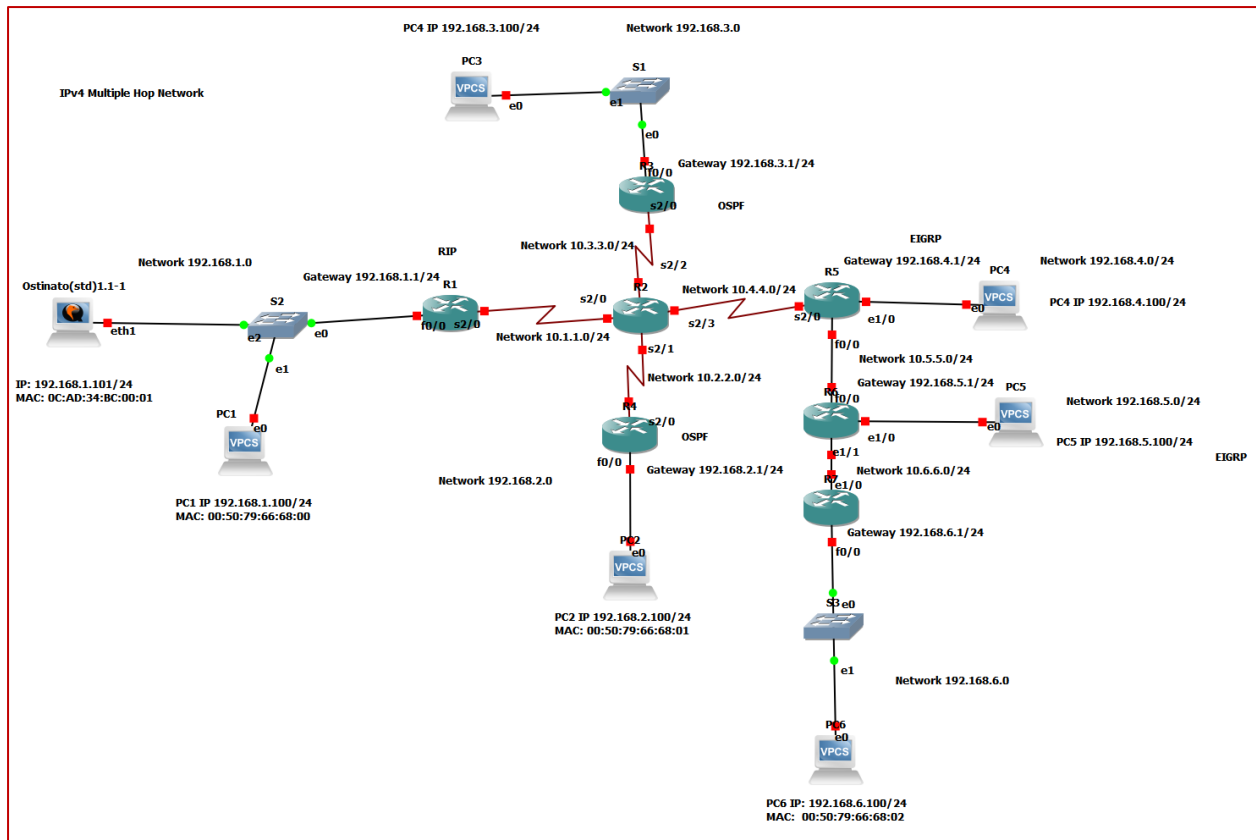


Figure 39: IPv4 Version 1 Network Logical Topology

The logical topology above consisted of not just the IP addresses for each system (except switches), but also tags displaying the “networks” and gateways of each, this insured that each part of the network was connected correctly as planned.

Also, some other tags in use were one for the title of the network and others for showing which side of the network used which dynamic routing protocols (RIP, OSPF, EIGRP).

Note the MAC addressing seen under some of the PC's was due to trouble shooting as setting them was originally postulated as being required for the tests which was later proven false (exception of Ostinato which requires MAC input for sending packets).

IPv4 Version 1 Network Physical Topology:

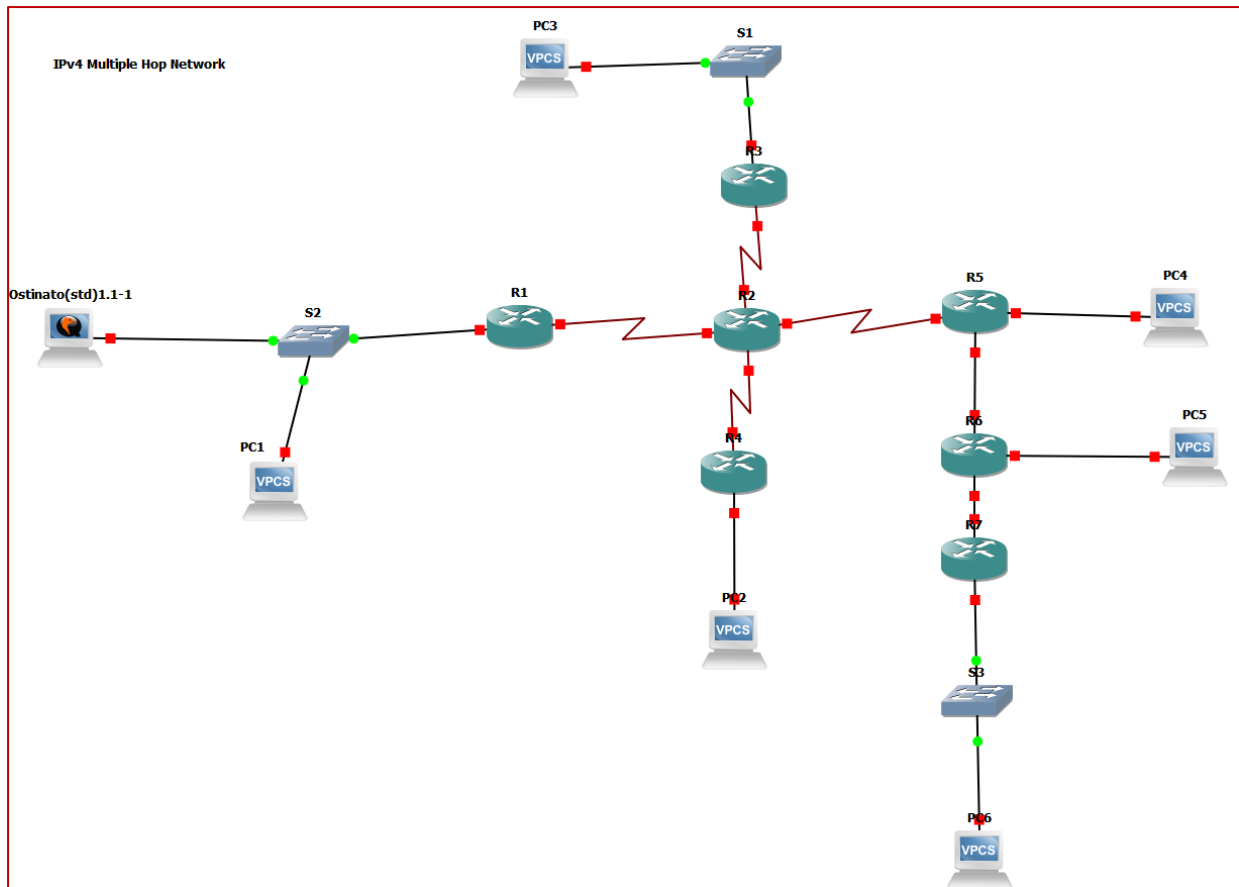


Figure 40: IPv4 Version 1 Network Physical Topology

The physical topology of the network above was carried out thanks to the recommendations provided, much of this setup remained untouched all the way to the final design as it was deemed suitable for the testing and analysis required later.

The notable aspect of this topology is that there were less PC's or systems implemented and that only one packet generation tool or node was implemented, this being Ostinato. The rest were basic virtual personal computers that were limited in what they could be used for which mainly consisted of pinging for trouble shooting.

IPv4 Network v5 Logical Topology (Final IPv4 Version):

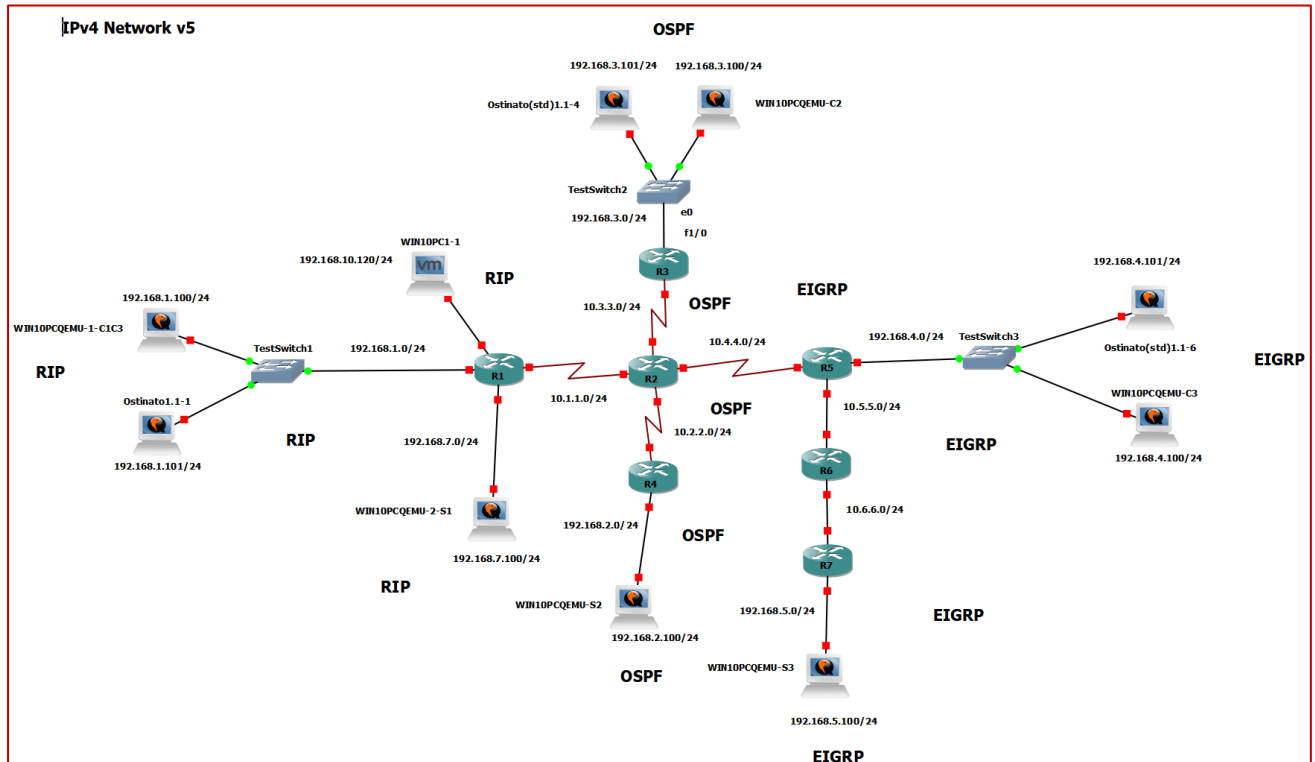


Figure 41: IPv4 Network v5 Logical Topology (Final IPv4 Version)

The logical topology above again consisted of IP addresses, “networks”, gateways, titles, dynamic routing protocols, some changes were made to the number of these to accommodate the additional systems implemented.

As can be seen here in this screenshot, the MAC addresses were taken out as they were no longer needed.

IPv4 Network v5 Physical Topology (Final IPv4 Version)

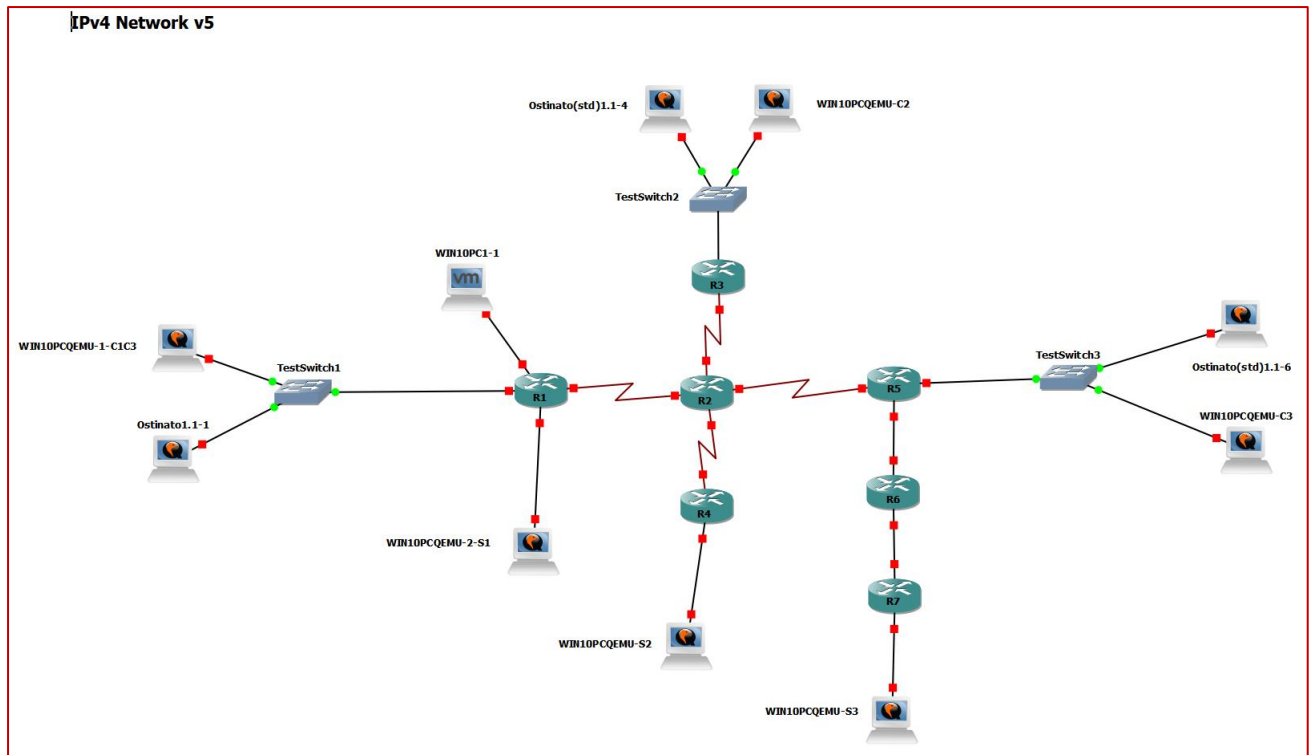


Figure 42: IPv4 Network v5 Physical Topology (Final IPv4 Version)

The physical topology of the IPv4 network above changed significantly, mainly after the author discovered that these changes were required for each of the tests to be done correctly and to retrieve the correct data for analysis later.

These changes mainly consisted of adding only Ostinato nodes and Windows 10 PC QEMU based VM's, removing basic PCs, with an additional VMWare based PC (WIN10PC1-1) added in the RIP section of the network. This additional VMWare based PC was used to move certain executable files required for testing (Iperf2 and Iperf3) to the other said Windows 10 PC QEMU based VM's in GNS3.

Each of these Windows 10 PC QEMU based VM's, except for the VMware one, was designated as either client or server depending on their position in each section of the network. This was required for Iperf to function as intended.

4.1.2.2 IPv6 Network

The second network created was for IPv6 and used a virtually identical setup to the IPv4 network shown previously. It was deemed that both networks could use the same topology for the experiment.

IPv6 Version 1 Network Logical Topology

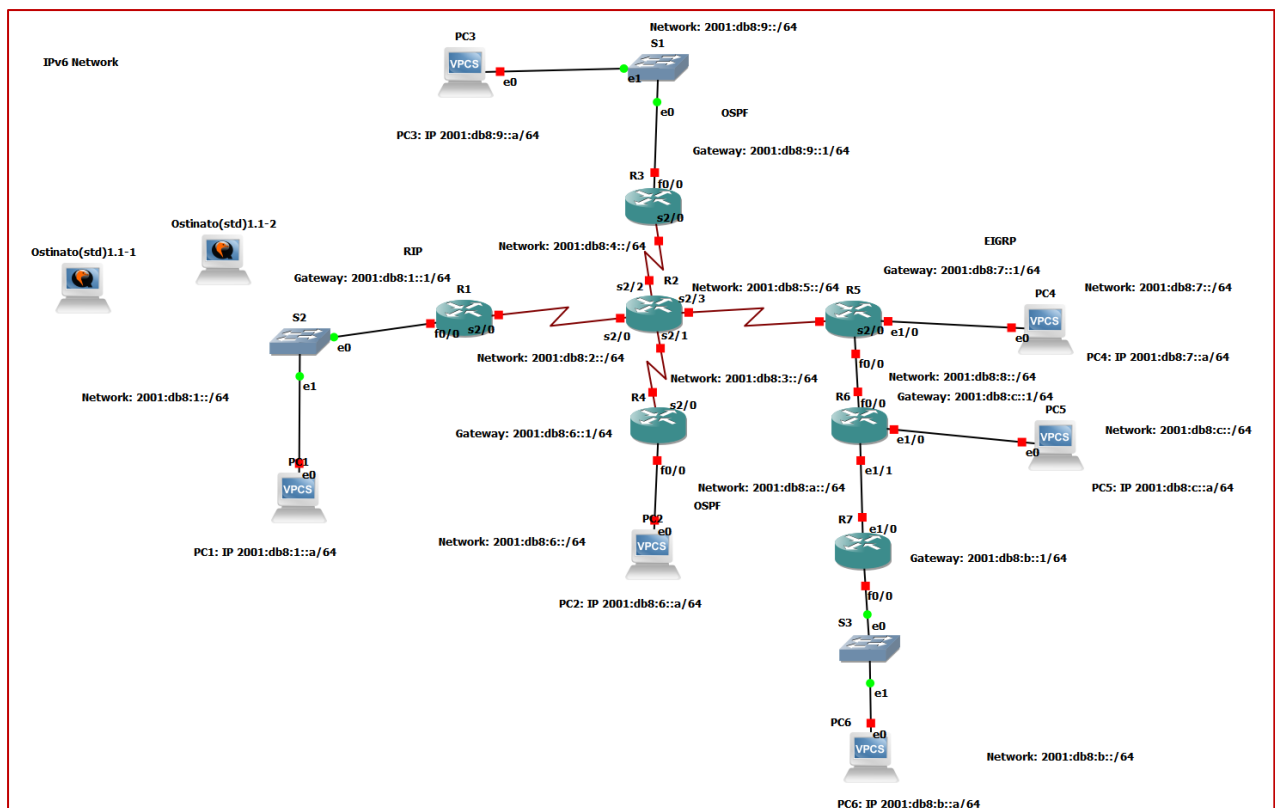


Figure 43: IPv6 Version 1 Network Logical Topology

The logical topology above as usual consisted of IP addresses (Now using IPv6 scheme), gateways, “networks”, sections of the network using RIP, OSPF and EIGRP along with other tags like the title of the project network.

Again, like the first version of the IPv4 network, this one also mostly used basic PCs with a few Ostinatos off to the side as they were used for trouble shooting.

IPv6 Version 1 Network Physical Topology:

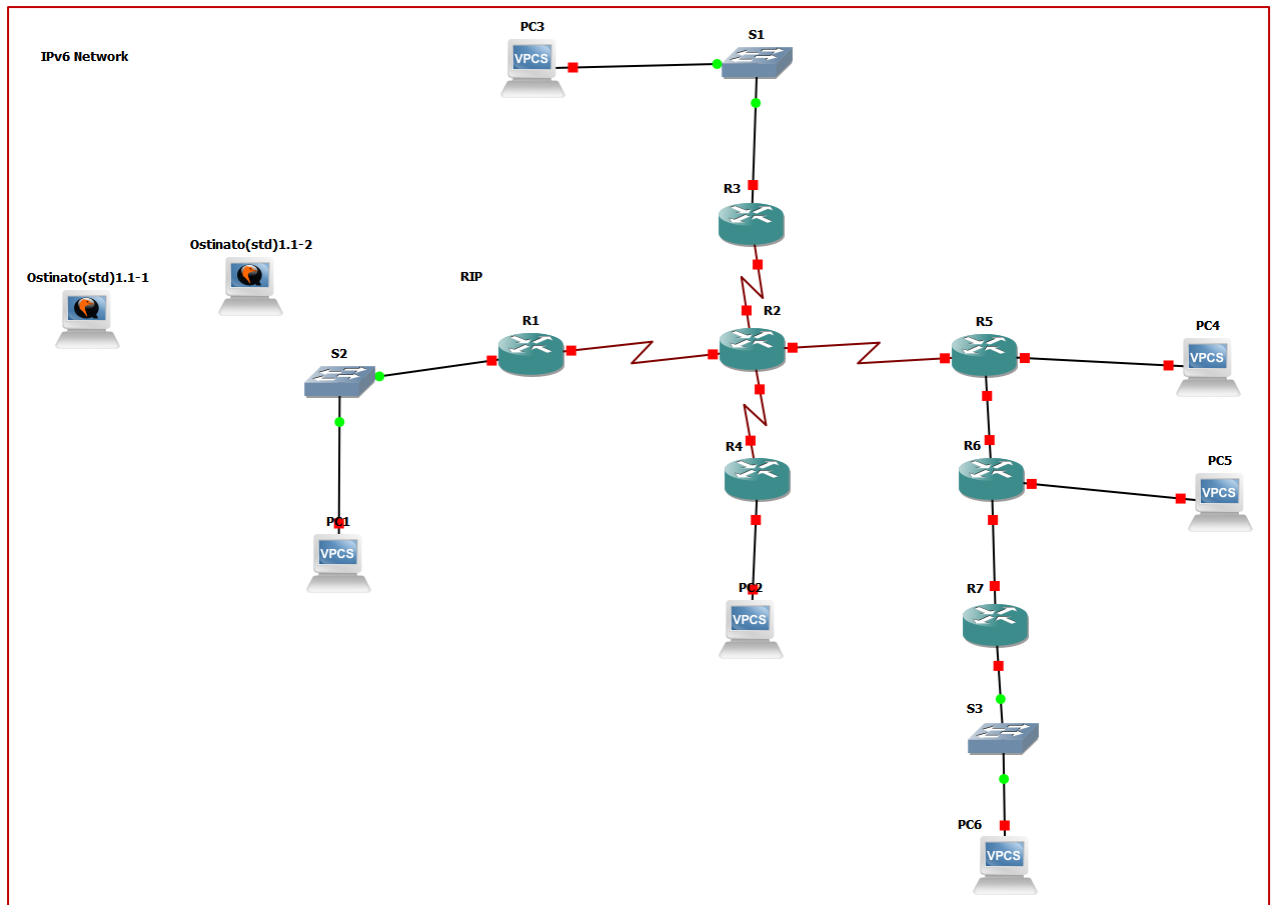


Figure 44: IPv6 Version 1 Network Physical Topology

The physical topology of the network above was virtually unchanged like the IPv4 networks. These early versions of the topologies were mainly for trouble shooting.

IPv6 Network v5 Logical Topology (Final IPv6 version):

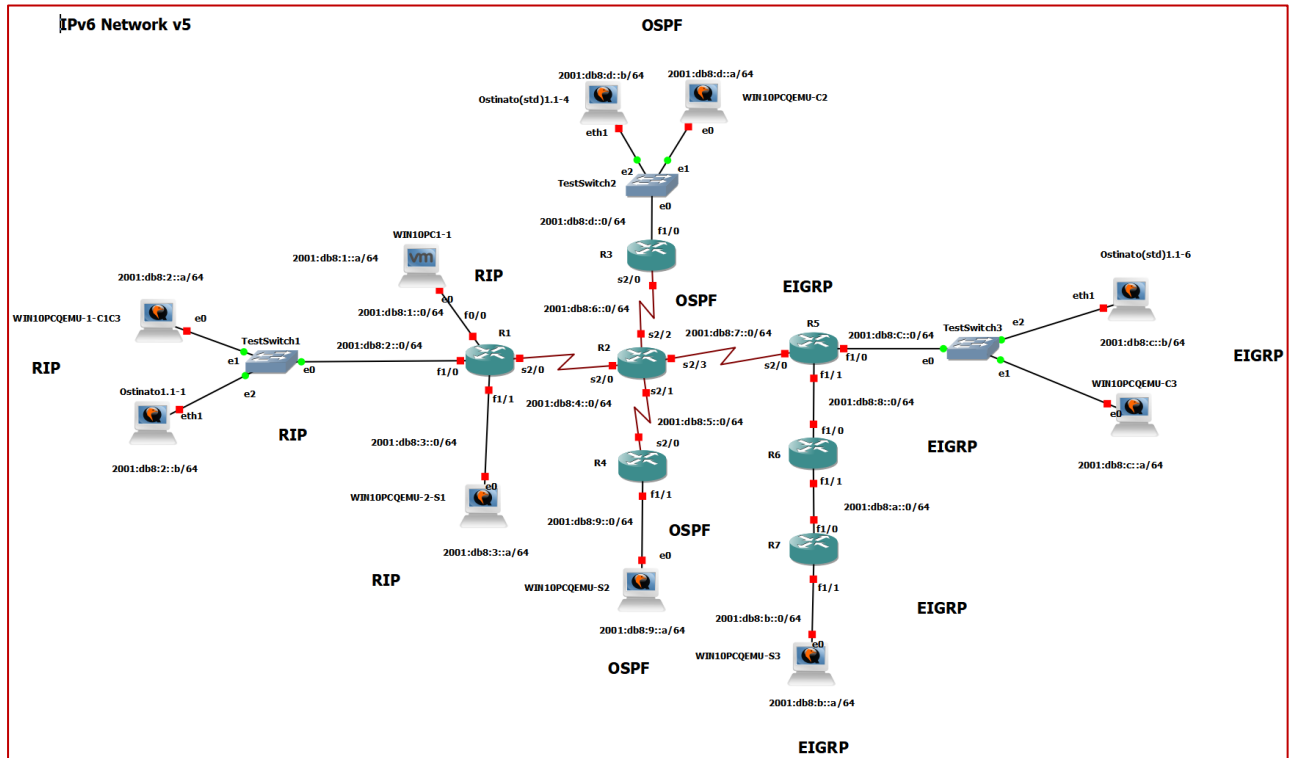


Figure 45: IPv6 Network v5 Logical Topology (Final IPv6 version)

The logical topology above again consisted of IP addresses, “networks”, gateways, titles, dynamic routing protocols, some changes were made to the number of these to accommodate the additional systems implemented.

As can be seen here in this screenshot, the MAC addresses were taken out as they were no longer needed.

IPv6 Network v5 Physical Topology:

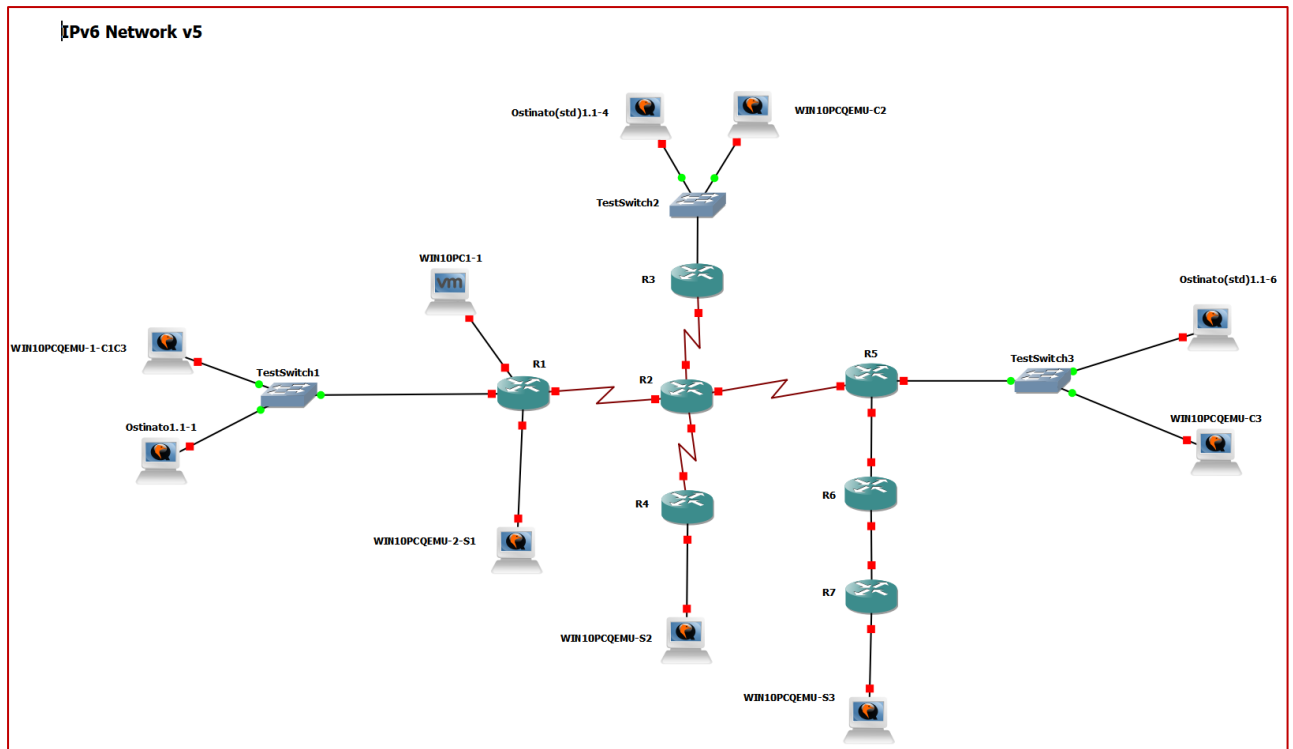


Figure 46: IPv6 Network v5 Physical Topology

The physical topology of the network above again changed significantly, mainly after the author discovered that these changes were required for each of the tests to be done correctly and to retrieve the correct data for analysis later.

The same changes as before were implemented, additional Nodes were added consisting of Ostinatos and Windows 10 PC QEMU based VM's with an additional VMWare based PC (WIN10PC-1) added in the RIP section of the network. This again was used to move certain executables required for the testing of (Iperf2 and Iperf3) to the other said Windows 10 PC QEMU based VM's in GNS3.

Each of these Windows 10 PC QEMU based VM's was designated as either client or server depending on their position in each section of the network. This was required for Iperf to function correctly.

4.2 Implementation

The implementation part of this study was done using varying methods and software to complete the goals of this project.

The tools used for the project were GNS3, Hyper V, VMware Workstation Pro, Packet Sender, Iperf and Ostinato (Add-on for GNS3), a network packet generating tool used to test varying differences in performance and other statistics. The underlying networking system used for the communication and delivery of data was Cisco based. This system provided an extremely stable foundation for all the other components to link into.

While there were many difficulties along the way, most were resolved with further investigation and troubleshooting. One example comes to mind where an error involving

GNS3 suggested a firewall problem. However, the reality was that it was due to a lack of access to the local GNS3 files or folders created when GNS3 was initially installed.

These problems were resolved by the author on his own or through some help from others mainly from videos or forums.

5. Result

5.1 Analysis of the result

The results of the experiment done using the selected software showed clear evidence of differences in performance for each IP version as well as the routing protocols used with the testing parameters chosen.

5.1.1 IPv4 Results:

IPv4	MTU Size														
	576			1000			1500			576			1000		
	Throughput(Mbits/s)			Packet Loss(%)			Delay(ms)			Latency(ms)			Jitter(ms)		
Maximum Segment Size(MSS)	536	960	1460	536	960	1460	536	960	1460	536	960	1460	536	960	1460
Test1(RIP-Single Hop)															
ICMP	0.694	1.52	2.47	28.4	32.9	33.7	8.5	8.5	8.5	17	17	17	4	3.25	6.5
UDP	1.05	1.05	1.05	0	0	0	8.5	8.5	8.5	17	17	17	0.684	0.83	1.156
TCP	3.87	6.78	10.6	1.6	0.2	0.5	8.5	8.5	8.5	17	17	17	4	3.25	6.5
Test2(OSPF-Multiple Hops)															
ICMP	0.25	0.474	0.75	0.3	0.8	38.5	30.5	30.5	30.5	61	61	61	5.25	5.25	5.25
UDP	1.05	1.05	1.05	59	26	0	26.5	26.5	26.5	53	53	53	17.933	16.29	16.418
TCP	0.176	0.37	0.503	7.6	6.7	6.5	26.5	26.5	26.5	53	53	53	4	4	4
Test3(EIGRP -Multiple Hops)															
ICMP	0.75	1.53	3.34	3.8	28.6	47.9	29.5	30.5	29	59	61	58	4.66	4.25	3.66
UDP	1.05	1.05	1.05	0	0	0	29.5	30.5	29	59	61	58	1.067	0.931	1.259
TCP	3.75	6.34	7.6	2.3	0.7	0.1	29.5	30.5	29	59	61	58	4.66	4.25	3.66
Test4(RIP, OSPF, EIGRP-Multiple Hops)															
ICMP	0.25	0.48	0.76	0.8	1.5	31.1	51	51.5	51.5	102	103	103	2.75	6.75	6.75
UDP	1.05	1.05	1.05	59	27	0	51	51.5	51.5	102	103	103	19.593	20.718	21.587
TCP	0.188	0.332	0.404	4.8	7.9	6.6	51	51.5	51.5	102	103	103	2.75	3.5	6.75

Figure 47: IPv4 Results

The image shown above is the overall data collected for each of the testing parameters, test scenarios, routing protocols, packet protocols and packet sizes.

Unlike IPv6, all tests were successful with all packet sizes done as well.

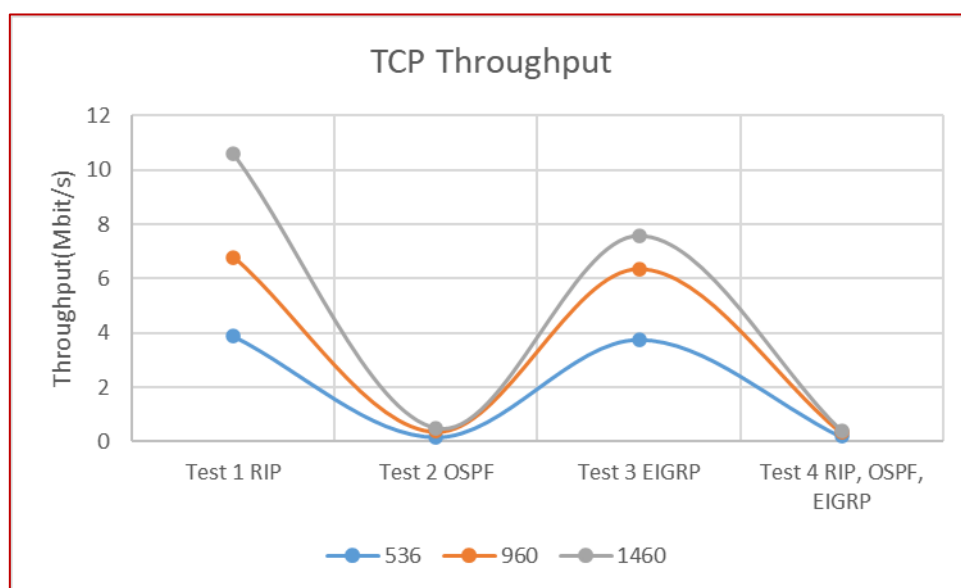


Figure 48: IPv4 TCP Throughput

As seen from this TCP graph, the throughput was shown to be the highest when tested through individual dynamic routing setups rather than through multiple dynamic routing protocols.

RIP and EIGRP were shown to have the highest throughput with RIP overtaking EIGRP by a considerable amount. It is likely that EIGRP performance suffered due to the number of hops required to passthrough (3 for EIGRP, 1 RIP).

Whereas OSPF and Test 4 (passthrough all dynamic routing protocols and multiple hops) showed a significant decrease in throughput performance. OSPF was mainly affected due to the serial connections used on R2 (which by default reduce the performance) as well as the multiple hops required for passthrough.

It is likely that a test of OSPF on their own networks or individual dynamic routing setups would result in better performance particularly if fast ethernet was used instead of serial link as fast ethernet provides better output.

Test 4 performance again was mainly affected by the serial links for R2, the number of hops passthrough.

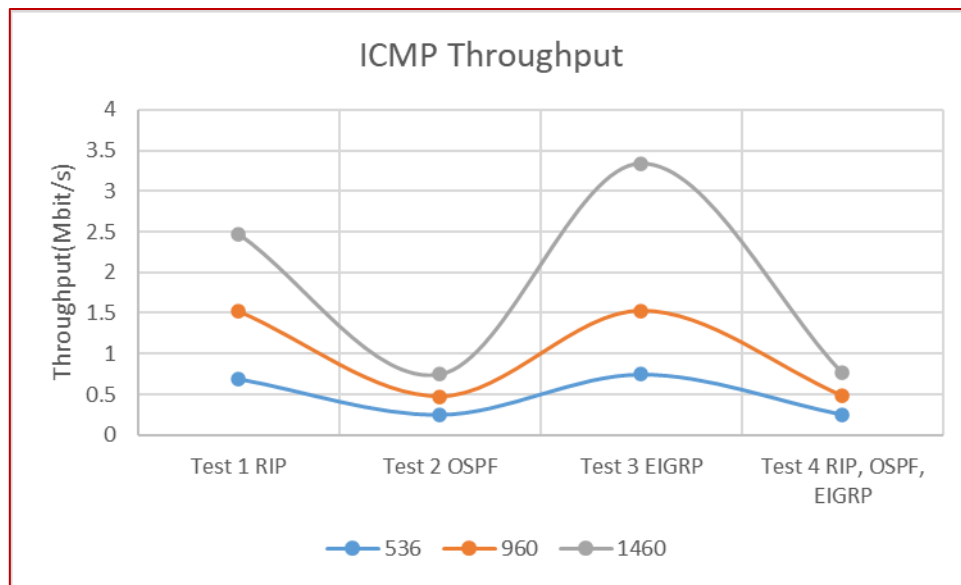


Figure 49: IPv4 ICMP Throughput

RIP and EIGRP for ICMP were shown to have the highest throughput with EIGRP overtaking RIP by a considerable amount.

OSPF and test 4 again suffered loss in performance compared to the other two.

EIGRP performed particularly well when used with a packet size of 1460 or higher which suggests that larger packet sizes increase performance.

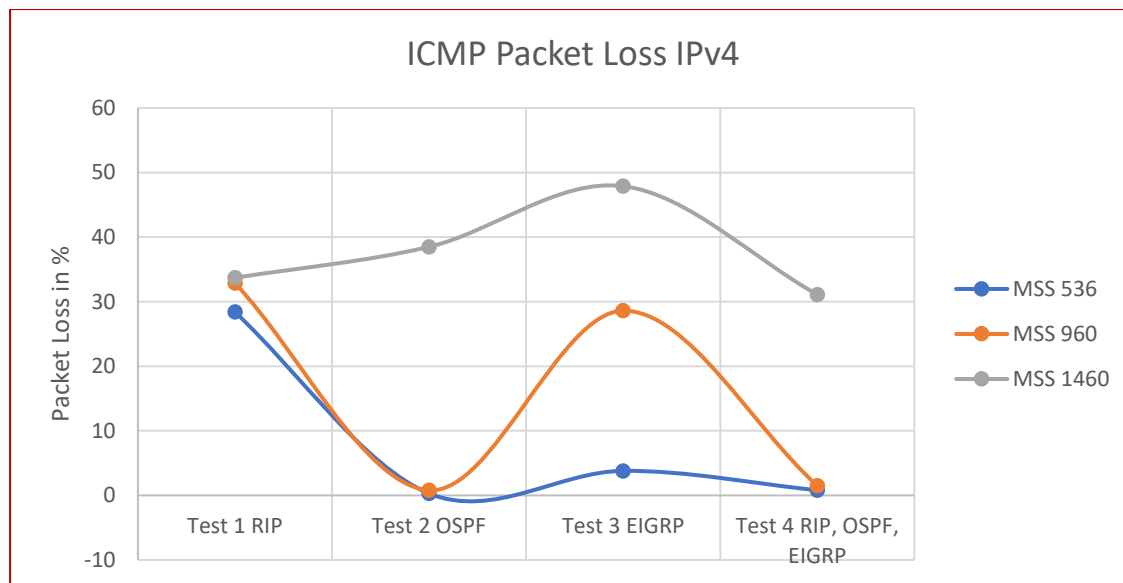


Figure 50: ICMP Packet Loss IPv4

RIP and EIGRP for ICMP were shown to have the highest amount of packet loss with EIGRP overtaking RIP.

OSPF and Test 4 were found to suffer little to no packet loss.

EIGRP seemed to perform particularly bad regarding packet loss with it having the highest amount with an MSS of 1460 and the lowest being MSS 536.

RIP seemed to have an almost consistent packet loss percentage regardless of the MSS size.

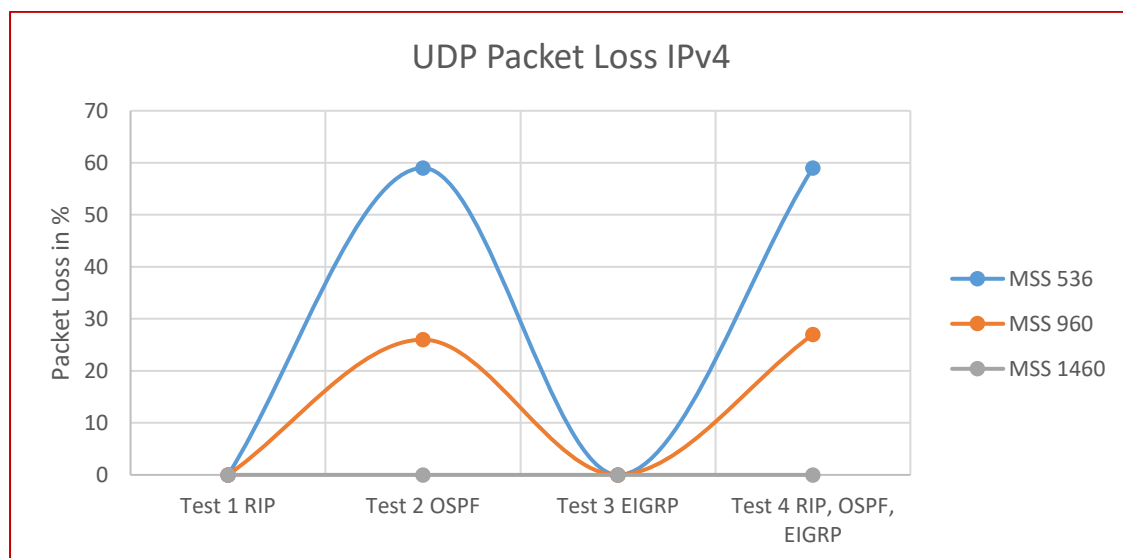


Figure 51: UDP Packet Loss IPv4

OSPF and Test 4 for UDP were shown to have the highest amount of packet loss for both MSS 536 and MSS 960 however the packet loss was 0 percent when packet sizes were higher than 960.

RIP and EIGRP were shown to be the lowest in packet loss.

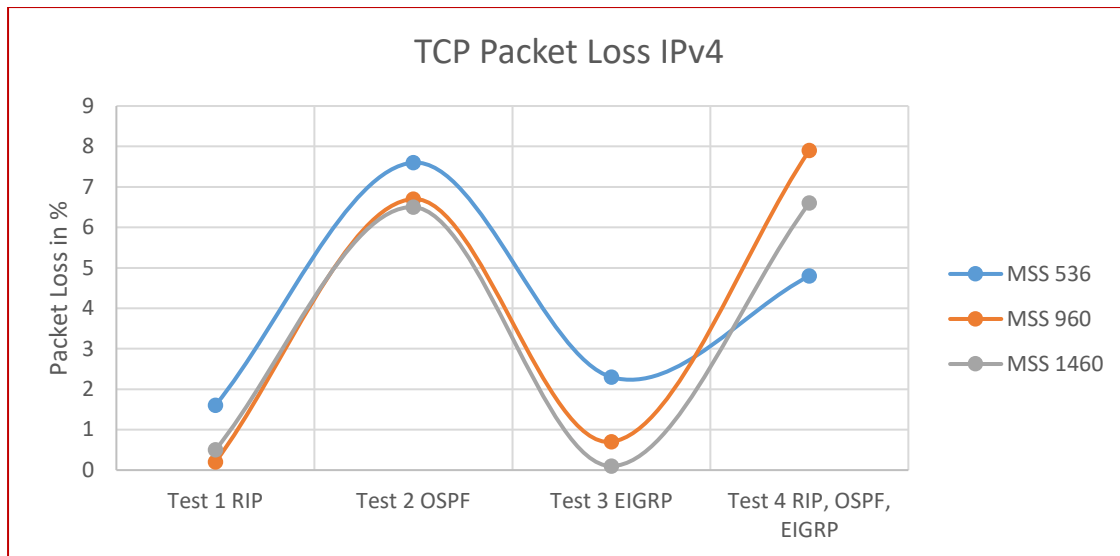


Figure 52: TCP Packet Loss IPv4

OSPF and Test 4 for TCP were shown to have the highest amount of packet loss for all packet sizes.

RIP and EIGRP were shown to be the lowest in packet loss.

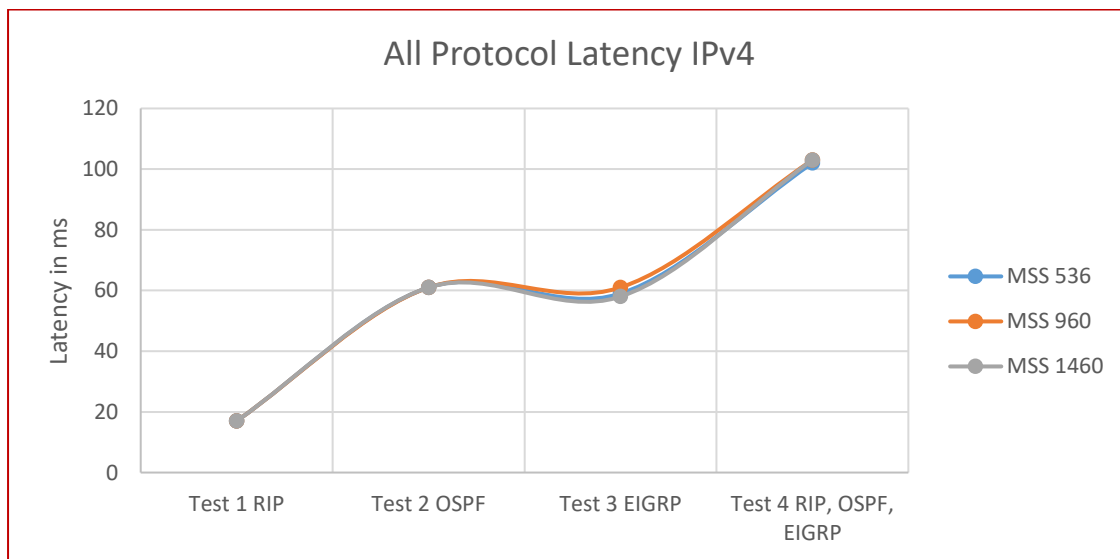


Figure 53: All Protocol Latency IPv4

In total, Test 4 with IPv4 was shown to have the highest amount of latency for all packet sizes with OSPF and EIGRP coming in at 2nd place in latency.

RIP was found to have the lowest latency out of all of them of less than 20ms.

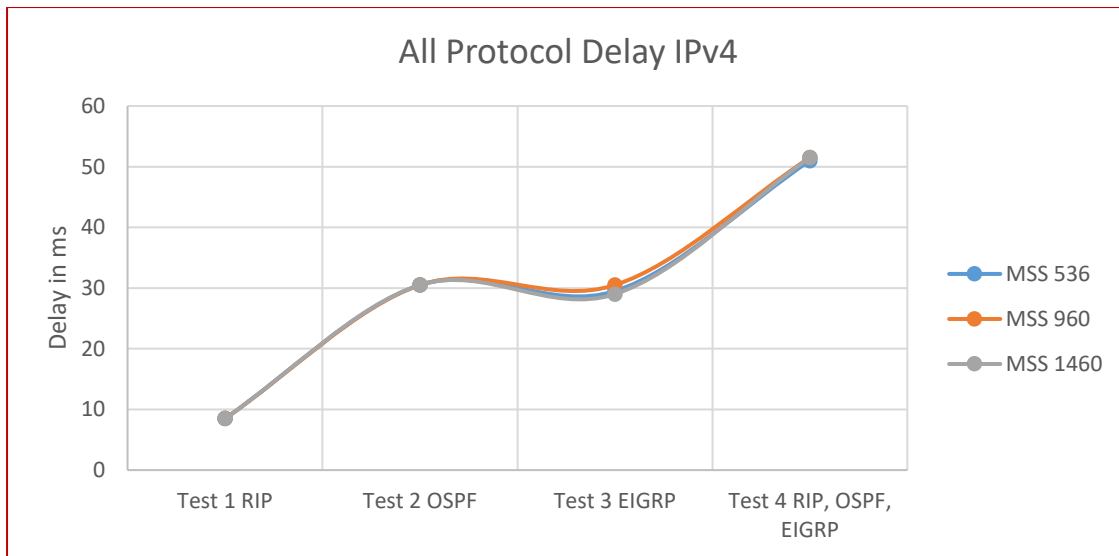


Figure 54: All Protocol Delay IPv4

In total, Test 4 with IPv4 was shown to have the highest amount of delay for all packet sizes with OSPF and EIGRP coming in at 2nd place in latency.

RIP was found to have the lowest amount of delay out of all of them of less than 10ms.

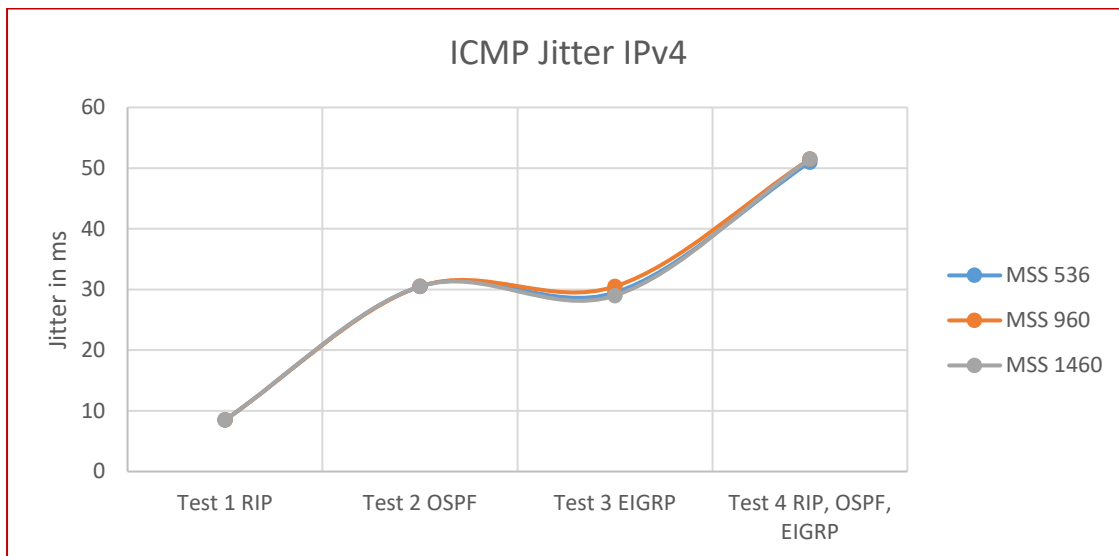


Figure 55: ICMP Jitter IPv4

In total, Test 4 with IPv4 was shown to have the highest amount of jitter for all packet sizes with OSPF and EIGRP coming in at 2nd place in latency.

RIP was found to have the lowest amount of jitter out of all of them of less than 10ms.

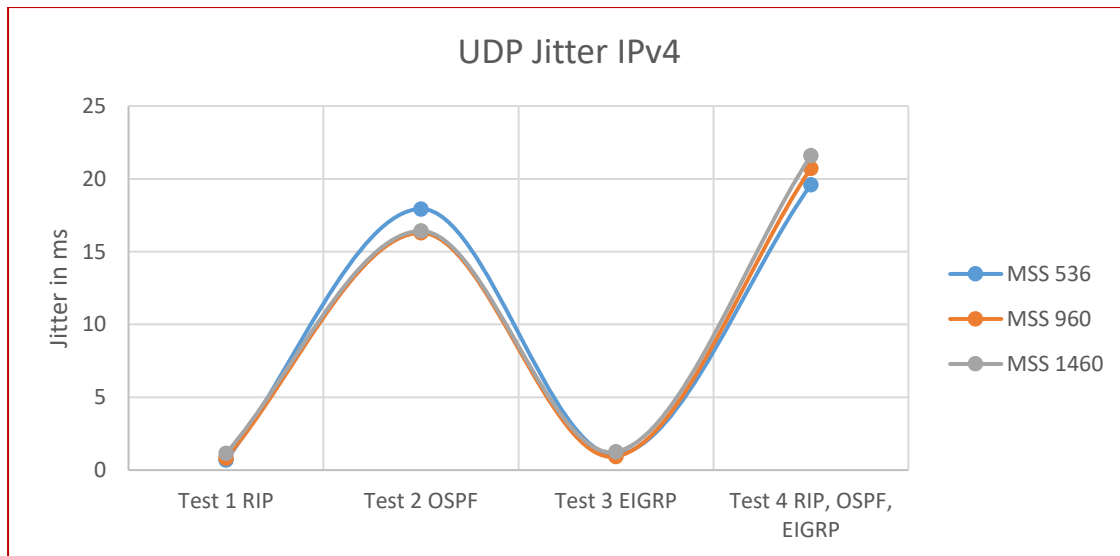


Figure 56: UDP Jitter IPv4

OSPF and Test 4 with UDP was shown to have the highest amount of jitter for all packet sizes.

RIP and EIGRP was found to have the lowest amount of jitter with less than 5ms.

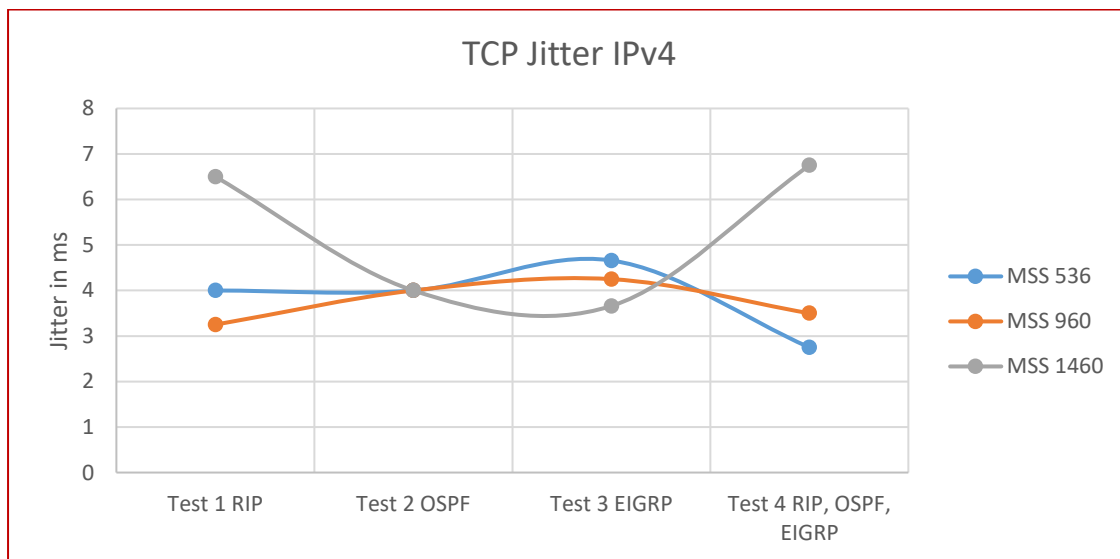


Figure 57: TCP Jitter IPv4

RIP and Test 4 with TCP was shown to have the highest amount of jitter for a packet size of 1460 however the rest were fairly similar in the amount of jitter found.

Test 4 was also the one with the lowest recorded amount of jitter at less than 3ms with a packet size of 536.

5.1.2 IPv6 Results

IPv6	MTU Size											
	N/A	1280	1500	N/A	1280	1500	N/A	1280	1500	N/A	1280	1500
	Throughput(Mbits/s)			Packet Loss(%)			Delay(ms)			Latency(ms)		
Maximum Segment Size(MSS)	N/A	1220	1440	N/A	1220	1440	N/A	1220	1440	N/A	1220	1440
Test1(RIP-Single Hop)												
ICMP		1.87	2.04		39.5	44.7		8.5	8.5		17	17
UDP		1.05	1.05		0	0		8.5	8.5		17	17
TCP		9.47	11.7		0.5	0.2		8.5	8.5		17	17
Test2(OSPF-Multiple Hops)												
ICMP		0.615	0.72		35.7	29		28.5	28.5		57	57
UDP		1.05	1.05		0	0		28.5	28.5		57	57
TCP		0.56	0.63		6.3	8.9		28.5	28.5		57	57
Test3(EIGRP-Multiple Hops)												
ICMP		1.85	2.3		37	29.3		30.5	30.5		61	61
UDP		1.05	1.05		0	0		30.5	30.5		61	61
TCP		8.6	10.02		0.3	0		29.5	29.5		59	59
Test4(RIP, OSPF, EIGRP-Multiple Hops)												
ICMP		0.614	0.72		2.5	2		50	51		100	102
UDP		1.05	1.05		0	0		50	51		100	102
TCP		0.49	0.56		6.6	6.6		50	51		100	102

Figure 58: IPv6 Results

The image shown above is the overall data collected for each of the testing parameters, test scenarios, routing protocols, packet protocols and packet sizes.

As noted in the excel sheet, the table shows missing data for one of the packet sizes (N/A) with the reasoning being the limitations of how small a packet size can be with IPv6 in use on the routers used for the experiment.

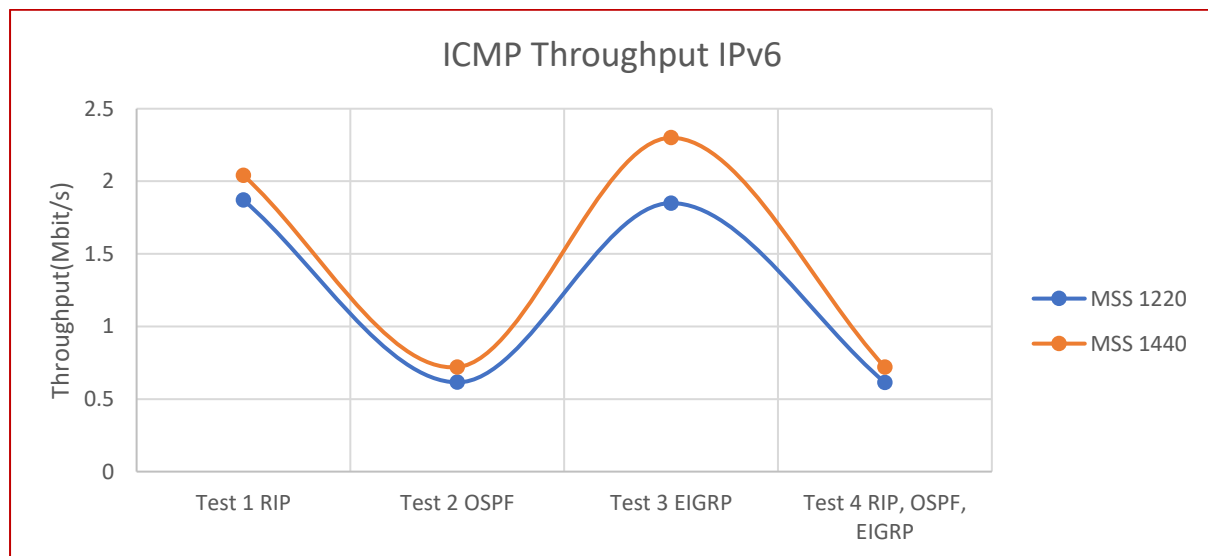


Figure 59: ICMP Throughput IPv6

RIP and EIGRP with ICMP was shown to have the highest throughput for both packet sizes. OSPF and Test 4 were recorded as having the lowest throughput for both packet sizes.

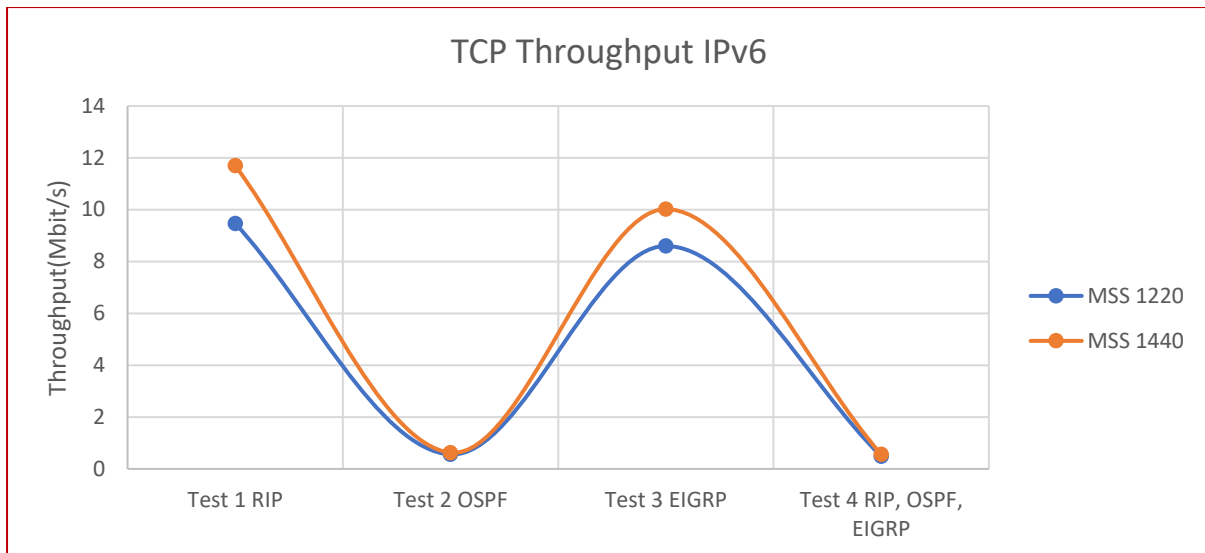


Figure 60: TCP Throughput IPv6

RIP with TCP was shown to have the highest throughput for both packet sizes with EIGRP coming in at 2nd place.

OSPF and Test 4 were both recorded as having the lowest throughput for both packet sizes.

5.1.3 IPv4 & IPv6 Combined Results

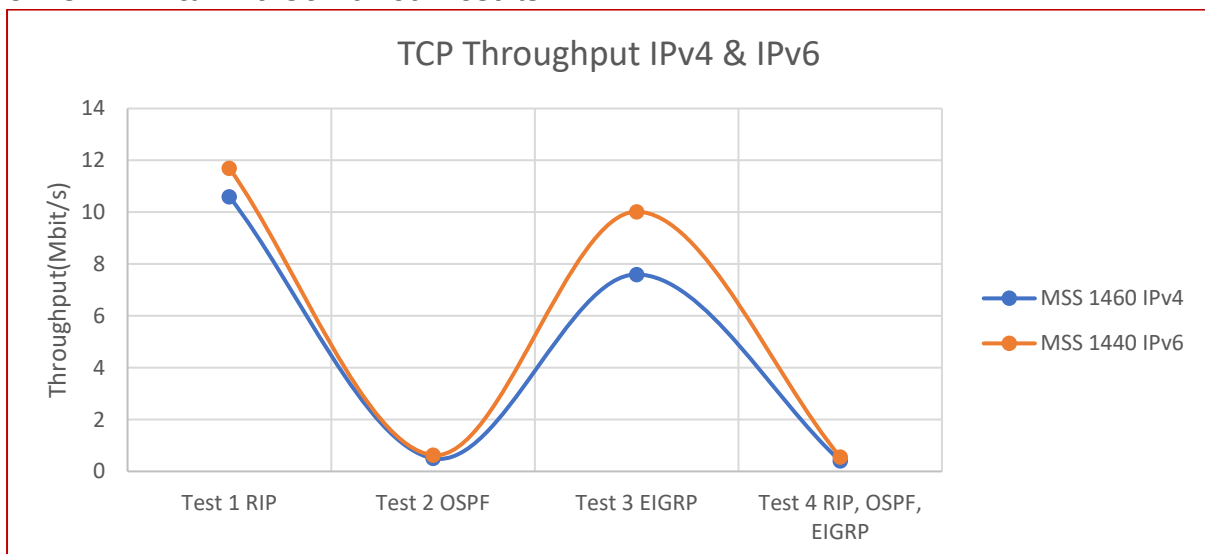


Figure 61: TCP Throughput IPv4 & IPV6

RIP for IPv6 was found to have throughput than IPv4 with EIGRP also showing a similar result

OSPF and Test 4 were both recorded as having the lowest throughput for both packet sizes of IPv4 and IPv6.

Based on this analysis, the author can be determined that RIP and EIGRP work best when used with IPv6 rather than IPv4.

OSPF and Test 4 however show no notable difference regardless of packet size or IP addressing scheme in use.

5.1.4 Standard Deviation, Mean, Median Results

Further analysis was performed on the outcomes achieved through GNS3. The analysis was specifically only performed on all the throughput results for both IPv4 and IPv6 as shown below. The data distribution was not shown as a bell curve due to the type of test and limited number of test results.

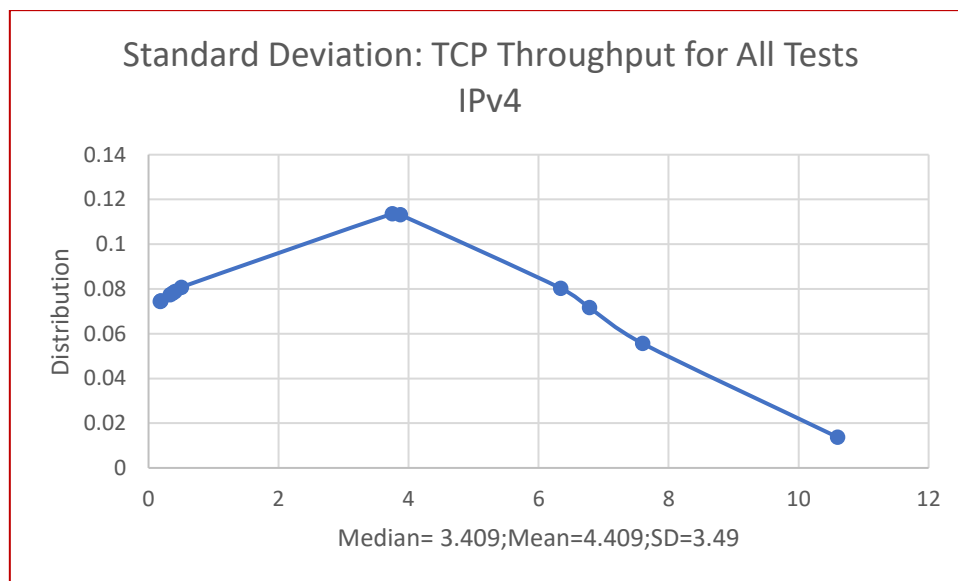


Figure 62: Standard Deviation TCP Throughput for All Tests IPv4

The mean, median and standard deviation results showed that a large portion of the test results fall within one standard deviation of the mean. This can be defined as reliable and reproduceable. However, the standard deviation result was larger than expected.

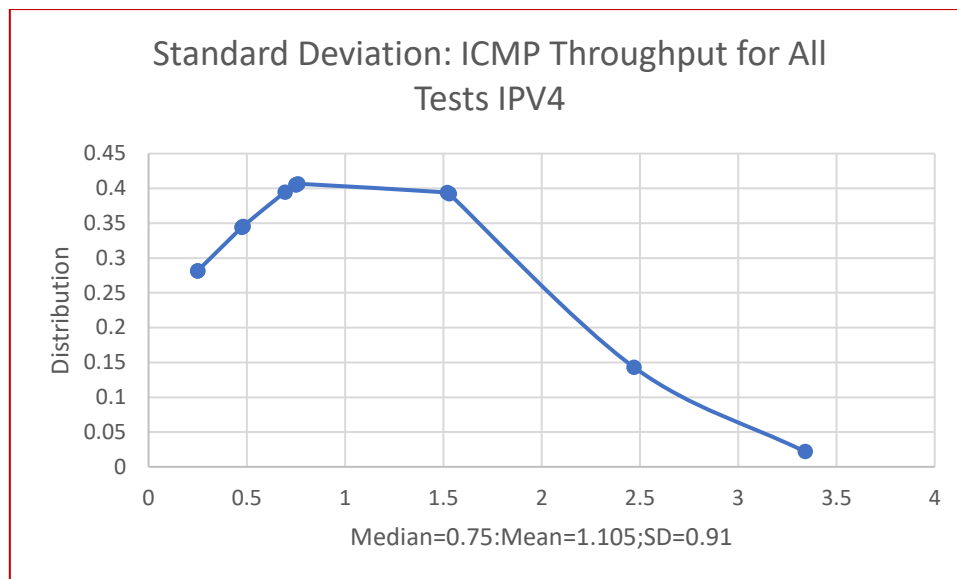


Figure 63: Standard Deviation ICMP Throughput for All Tests IPV4

The mean, median and standard deviation results showed that a large portion of the test results fall within one standard deviation of the mean. This can be defined as reliable and reproduceable. However, the standard deviation result was smaller than expected.

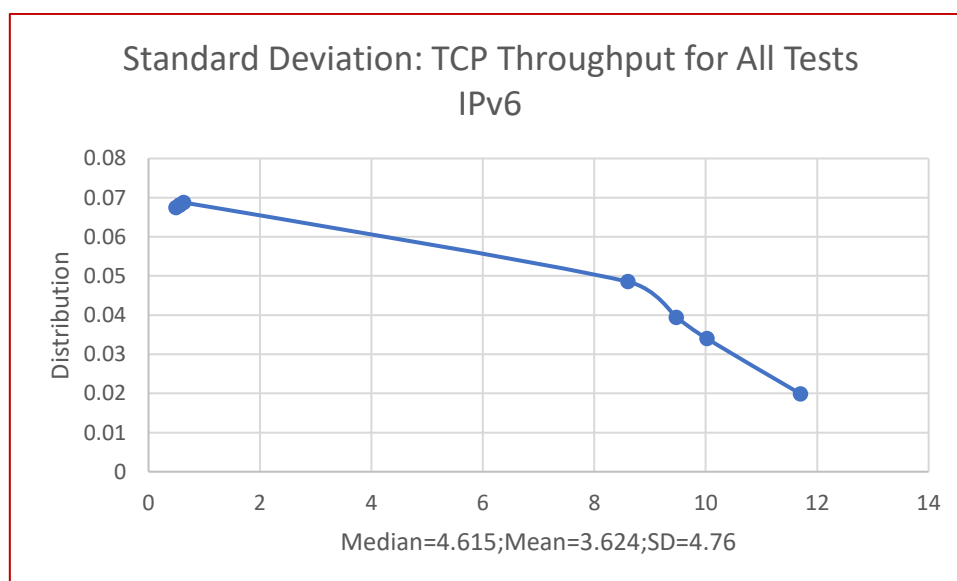


Figure 64: Standard Deviation TCP Throughput for All Tests IPv6

The mean, median and standard deviation results showed that a large portion of the test results fall within one standard deviation of the mean. This can be defined as reliable and reproduceable. However, the standard deviation result was larger than expected.

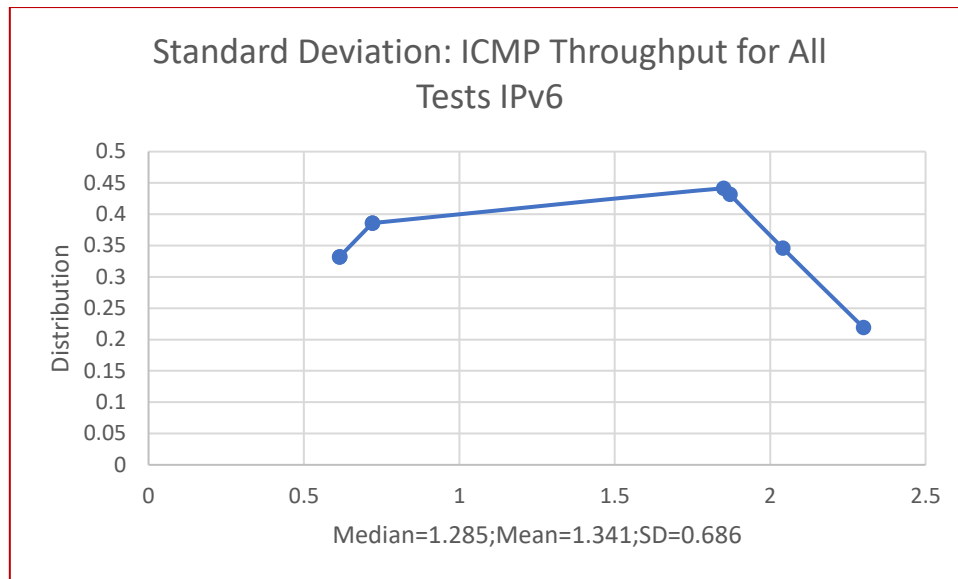


Figure 65: Standard Deviation ICMP Throughput for All Tests IPv6

The mean, median and standard deviation results showed that a large portion of the test results fall within one standard deviation of the mean. This can be defined as reliable and reproduceable. However, the standard deviation result was smaller than expected.

5.1.5 Conclusion

The conclusions from the analysis performed were found to be as follows:

Multi hop & Multi Routing protocol Test 4 shows greater Latency & Delay. Mostly because of the serial links and the number of hops.

Smaller Packet sizes directly reduce the throughput

RIP and EIGRP performed well with throughput on most tests whereas OSPF was restricted by the particular networking designs used i.e. inclusion of serial links.

Test 4 which involved all 3 dynamic routing protocols and slower serial links was particularly impacted by the network design also.

IPv6 was found to be better performing than IPv4 especially in the case of throughput. Performing better especially with larger packet sizes.

UDP throughput was found to be by the author to be largely unaffected by any of the following parameters such as latency, delay, packet loss and jitter.

Standard Deviation was used to investigate distribution of throughput test results for both IPv4 and IPv6. The results indicated most test results fell within one standard deviation of the mean.

The packet loss results were made up of packet retransmissions. This was done in an effort to portray the impact of the said retransmissions on throughput rates.

6. Evaluation

The conclusion drawn from this data is that overall IPv6 is found to perform better than IPv4 in many aspects.

The results showed that they are still used interchangeably depending on their use and the data received from the tests backed up most of the literature research found with some areas lacking in detail, likely due to the nature of the testing environments or techniques used etc.

Regarding the objectives and aim of this study, the objectives were met with little difficulty whereas the aim was at first troubling to follow due to the process of setting up the equipment mainly on the software side where various compatibility issues as well as errors had to be rectified before real progress could be made.

Overall, the answers gained from the experiment were almost entirely as expected with the few exceptions being the handling of data collection as well as the software functionality.

6.1 Future Work

If this study or experiment was to be done again, it would be handled similarly with the main differences being that testing and data collection would be done using wireless connections instead of just wired ones with different routing protocols and techniques being applied at the same time.

Perhaps even a study using real physical machines and not just through virtualisation could be used to further validate any findings made regarding this topic.

Implementing and testing the project work highlighted the pure point-to-point nature of the current network design. In essence the network has several single points of failure. This can be remedied in part by introducing some additional connections between routers 4 and 5 or routers 4 and 6. This will provide much needed redundancy in case of router or link failures.

- Do an extra test that pertains to changing the bandwidth cost from its default value to something else for EIGRP and OSPF.
- Create another project or network for testing with dual stack using IPv4 and IPv6.

7. Conclusions & Recommendations

To summarise everything done in this report, an experiment was done to analyse the performance of dynamic routing protocols (RIP, OSPF and EIGRP) using IPv4 and IPv6 addressing scheme.

Other possible areas for potential research and experimentation were considered such as wireless networks, tunnelling methods with these protocols as well as the use of other routing protocols old or new with IPv4 and IPv6.

With these areas explored, usage of the internet within the experiment was also considered. However, this idea was abandoned due to implementation and utilisation of said mechanism would not be appropriate under test-controlled conditions with many external factors potentially causing inconsistent or problematic test results.

In terms of recommendation, anyone who wants to use the software and hardware used in this study, ensure that the system chosen to run the software (VMware Workstation Pro, GNS3, Ostinato etc.) can run it correctly and efficiently as these tools can be demanding to use as it is dependent on how much resources or performance the user wishes to provide for these tools (Number of CPU cores, Amount of RAM etc.).

Based on the author's experience with this software, the main thing to consider is RAM as even with the author's setup of 16gb of RAM in use on the author's personal computer it still struggled at times to run efficiently thus its recommended to use insert more if possible.

When using GNS3, be sure to appropriately save all parts of the networks mainly the nodes (routers, pcs etc.) when working or finishing a network project as otherwise they will start without the configs or information implemented previously (IP addresses, routing settings etc.) meaning any major changes done within the nodes with a console will be lost.

To save these configs you simply enter the console of each node and enter the correct command for saving them which in the case of PC's is "save" and for routers is "copy running-config".

As for any other potential issues that the user may need help with regarding these programs, there are various sources to use online or offline that can provide most of the answers needed with one being YouTube with many tutorials done via videos or the other being documentation.

8. Reflection

In reflection of the work done, whilst many things were learned from the project, there were however many problems with setting up the experiment and obtaining the results desired by the author.

Some of the main reasons behind the troubling schedule of the project was due to the time taken to find suitable software for the experiment as well as using said software without problem as numerous issues were encountered and with some not being possible to fix or change despite assurances from outside sources found through research regarding how the issues are supposedly fixed.

If to summarise these problems into bullet points it would be as follows:

- Finding appropriate software for tasks required
- Using software or tools without issue
- Lack of Vendor support

Simply setting up the virtual machines as well as GNS3 as the interface for creating the networks was problematic. While Hyper-V was initially used as the go to virtual machine environment to house GNS3, it was replaced by VMware later due to limitations relating to Packet Generation tools used.

VMware was more difficult to setup than Hyper-V due to compatibility issues particularly when attempts are made to use it with Hyper-V in the background or still installed.

GNS3 was found to be good for making more simple or smaller networks. However, once the size and complexity of these networks was increased various problems began to pop up.

Its main problem however was its tendency to give errors without appropriate explanation with one example being where it was attempting to open GNS3 project files but failed due supposedly to a firewall issue based on its code. However, it was found by the author to be due to the original location of the files used for saving which was at first done on a USB stick and later changed to the Author's PC to fix the problem but also to help with performance.

Once these problems were delt with and the networks were setup, packet generation tools were investigated with the first tool being Ostinato and this is where most if not all the problems with project stemmed from particularly troubling due to it being at the end of the project timeline.

Various assumptions were made based on the material online which described the tools like Ostinato as good, easy to use tools however from the authors' case this was simply not the outcome desired.

Time for setup and using the tools as required were two of the biggest problems found with the additional being many errors.

The tools used for packet generation and analysis were Ostinato, Packet Sender, Iperf3 and Wireshark.

Some examples of problems found with these packets were as follows:

- Advertising features as useable when they in fact were not

Ostinato was the most notable in this regard as TCP was presented as an apparent protocol that could be tested which in fact it was not as the software itself was stateless not stateful meaning TCP could not be used as required.

- Missing helpful features or making tests more difficult and time consuming

Most if not all the tools found throughout the project were disappointing in terms of ease of use as even the ones found which did finally allows tests to be done were tedious to setup or worst of all flat out missing necessary components for testing.

Packet sender for example had a tedious way of creating packets required where the user was required to create them manually via notepad and other functions while lperf3 was odd in that it could do tests with throughput for TCP and UDP however it couldn't do this for ICMP.

Another issue as seen with this being the amount of time wasted on having to switch between each tool rather than using one tool which does all required tests. This tool at a reasonable cost does not seem to exist at the time this project was done.

- Not showing all the information necessary for better understanding

Wireshark while good for most of the tasks required of it had one problem, this problem was regarding information. The full details of what made up each packet sent from one end to the other where it would not only show a seemingly incorrect packet size it would also not say precisely what certain parts of the packets were made of.

This had to be worked out manually and even then, it wasn't 100% accurate due to the packets being used on a simulated network environment which meant that inaccuracies were likely and did occur.

Another issue that showed up towards the end was the problem with packet sizes relating to IPv6 as it was found to have limitations regarding how small the packet size could be for testing (At least on the systems used for this experiment) which resulted in slight difference in results with one packet size parameter having to be dropped due to it being below the range.

Also in general testing different packet sizes were time consuming and difficult for a time due to various features and commands being required for use with the routers used to allow these packet sizes to work and be sent for testing in the first instance.

This changed the author as a person by presenting him with a better idea as to how these networks work as well as how they are potentially implemented around the world. Things such as the complexities of these networks and how they affect our daily lives.

Regarding all this, the author is shown to have more respect for the various types of services out there that involve this technology and engineering in general as while they provide opportunities, they also expose users to many potential dangers.

With all said and done, the project could have been done better with a more powerful system, better software, and more time accolated etc.

9. References:

- [1] L. Klucznik(Adair), 'Networking Basics (Cisco)', p. 80.
- [2] 'What is an IP Address – Definition and Explanation', *www.kaspersky.com*, Jul. 05, 2021. <https://www.kaspersky.com/resource-center/definitions/what-is-an-ip-address> (accessed Mar. 29, 2022).
- [3] 'IPv6 – Google'. <https://www.google.com/intl/en/ipv6/statistics.html> (accessed Dec. 02, 2021).
- [4] 'What is IPv4? Everything you need to know - CloudDNS Blog'. <https://www.cloudns.net/blog/what-is-ipv4-everything-you-need-to-know/> (accessed Nov. 25, 2021).
- [5] 'Limitations of IPv4 and IPv4 exhaustion'. <https://nfware.com/blog-ipv4-limitations> (accessed Mar. 29, 2022).
- [6] 'What is IPv6? Why you should start using it now?', Nov. 25, 2021. <https://serverguy.com/servers/what-is-ipv6/> (accessed Nov. 25, 2021).
- [7] 'Routing Protocol | How It Works | Mechanism and Types Of Routing', *EDUCBA*, Jun. 20, 2019. <https://www.educba.com/routing-protocol/> (accessed Apr. 20, 2022).
- [8] J. Aweya, 'IP Routing Protocols; Fundamentals and Distance-Vector Routing Protocols', p. 325.
- [9] 'Routing Information Protocol (RIP) - GeeksforGeeks'. <https://www.geeksforgeeks.org/routing-information-protocol-rip/> (accessed Nov. 25, 2021).
- [10] 'What is Open Shortest Path First (OSPF)?' <https://www.metaswitch.com/knowledge-center/reference/what-is-open-shortest-path-first-ospf> (accessed Nov. 25, 2021).
- [11] 'What is EIGRP? | Fundamentals & Benefits of EIGRP Routing Protocol'. <https://www.educba.com/what-is-eigrp/> (accessed Nov. 25, 2021).
- [12] A. Shiranzai and R. Z. Khan, 'Internet Protocol Versions – A Review', p. 5, 2015.
- [13] T. W. Ogletree, 'Upgrading and Repairing Networks', p. 1247.
- [14] A. Syrewicze and R. Siddaway, *Pro Microsoft Hyper-V 2019: Practical Guidance and Hands-On Labs*. Berkeley, CA: Apress, 2018. doi: 10.1007/978-1-4842-4116-5.
- [15] J. C. Neumann, *The Book of GNS3: Build Virtual Network Labs Using Cisco, Juniper, and More*. No Starch Press, 2015.
- [16] 'Using VMware Workstation Pro - VMware Workstation Pro 16.0', p. 360.
- [17] B. R. Patil, M. Moharir, P. K. Mohanty, S. G, and S. S, 'Ostinato - A Powerful Traffic Generator', in *2017 2nd International Conference on Computational Systems and Information Technology for Sustainable Solution (CSITSS)*, Bengaluru, India, Dec. 2017, pp. 1–5. doi: 10.1109/CSITSS.2017.8447596.
- [18] D. Nagle, 'Packet Sender - Documentation'. <https://PacketSender.com/> (accessed Jun. 16, 2022).

- [19] 'iPerf - The TCP, UDP and SCTP network bandwidth measurement tool'. <https://iperf.fr/> (accessed Jun. 16, 2022).
- [20] J. Bullock and J. T. Parker, *Wireshark for security professionals: using Wireshark and the Metasploit framework*. Indianapolis, Indiana: Wiley, 2017.
- [21] S. Narayan, Peng Shang, and N. Fan, 'Network performance evaluation of Internet Protocols IPv4 and IPv6 on operating systems', in *2009 IFIP International Conference on Wireless and Optical Communications Networks*, Cairo, Egypt, Apr. 2009, pp. 1–5. doi: 10.1109/WOCN.2009.5010548.
- [22] R. Gunasundari and S. Shanmugavel, 'Performance Comparison of Mobile IPv4 and Mobile IPv6 protocols in wireless systems', in *2009 First International Communication Systems and Networks and Workshops*, Bangalore, India, Jan. 2009, pp. 1–8. doi: 10.1109/COMSNETS.2009.4808842.
- [23] M. N. Ismail and Z. Z. Abidin, 'Implementing of IPv6 Protocol Environment at University of Kuala Lumpur: Measurement of IPv6 and IPv4 Performance', in *2009 International Conference on Future Computer and Communication*, Kuala Lumpur, Malaysia, Apr. 2009, pp. 443–449. doi: 10.1109/ICFCC.2009.145.
- [24] S. S. Kolahi, N. Chand, B. K. Soorty, and Z. Qu, 'Performance Comparison of IPv4 and IPv6 in Peer-Peer and Client Server Local Area Networks', in *2010 10th IEEE International Conference on Computer and Information Technology*, Bradford, United Kingdom, Jun. 2010, pp. 41–45. doi: 10.1109/CIT.2010.51.
- [25] A. Hinds, A. Atojoko, and S. Ying Zhu, 'Evaluation of OSPF and EIGRP Routing Protocols for IPv6', *IJFCC*, pp. 287–291, 2013, doi: 10.7763/IJFCC.2013.V2.169.
- [26] J. Deng, S. Wu, and K. Sun, 'Comparison of RIP, OSPF and EIGRP Routing Protocols based on OPNET', p. 25.
- [27] D. A. Camarero, 'Performance analysis of IPv4 / IPv6 protocols over the third generation mobile network.', p. 81.
- [28] R. Narula and P. Aggarwal, 'PERFORMANCE EVALUATION OF RIP AND OSPF IN IPV6 USING OPNET 14.5 SIMULATOR', vol. 2, no. 6, p. 5, 2014.
- [29] D. Chauhan and S. Sharma, 'Performance Evaluation of Different Routing Protocols in IPv4 and IPv6 Networks on the Basis of Packet Sizes', *Procedia Computer Science*, vol. 46, pp. 1072–1078, 2015, doi: 10.1016/j.procs.2015.01.019.
- [30] S. T. Chandel and S. Sharma, 'Performance Evaluation of IPv4 and IPv6 Routing Protocols on Wired, Wireless and Hybrid Networks', *International Journal of Computer Networks and Applications*, vol. 3, no. 3, p. 7, 2016.
- [31] D. R. Al-Ani and A. R. Al-Ani, 'The Performance of IPv4 and IPv6 in Terms of Routing Protocols using GNS 3 Simulator', *Procedia Computer Science*, vol. 130, pp. 1051–1056, 2018, doi: 10.1016/j.procs.2018.04.147.
- [32] M. N. K. Saunders, P. Lewis, and A. Thornhill, *Research methods for business students*, Eighth Edition. New York: Pearson, 2019.
- [33] 'Saunders' Research Onion: Explained Simply (+ Examples)', *Grad Coach*, Jan. 26, 2021. <https://gradcoach.com/saunders-research-onion/> (accessed Mar. 22, 2022).

- [34] A. t Gardner, 'Examples of Objective Knowledge | Synonym'. <https://classroom.synonym.com/examples-objective-knowledge-23431.html> (accessed May 03, 2022).
- [35] HowandWhat.net, 'Differences between deductive and inductive approaches to research', *howandwhat*, Nov. 18, 2016. <https://howandwhat.net/differences-deductive-inductive/> (accessed May 03, 2022).
- [36] '8 Advantages and Disadvantages of Experimental Research', *FutureofWorking.com*, Mar. 10, 2016. <https://futureofworking.com/8-advantages-and-disadvantages-of-experimental-research/> (accessed May 03, 2022).
- [37] 'Advantages and Disadvantages', *Mixed Methods Research*, Feb. 18, 2016. <https://mixedmethodsresearchblog.wordpress.com/advantages-and-disadvantages/> (accessed May 03, 2022).
- [38] L. Gaille, '19 Advantages and Disadvantages of Cross Sectional Studies'. <https://vittana.org/19-advantages-and-disadvantages-of-cross-sectional-studies> (accessed May 03, 2022).
- [39] 'Waterfall Model | Different Phases With Advantages & Disadvantages', *EDUCBA*, May 20, 2019. <https://www.educba.com/waterfall-model/> (accessed Mar. 22, 2022).