

Nathan Munkley - Major Project

by Nathan Munkley

FILE	17637_NATHAN_MUNKLEY_NATHAN_MUNKLEY_-_MAJOR_PROJECT_78196_236404319.DOCX (3.08M)		
TIME SUBMITTED	20-APR-2016 12:36PM	WORD COUNT	17701
SUBMISSION ID	56117454	CHARACTER COUNT	103749



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

Performance Comparison of IPv4 and IPv6.

By
Nathan Munkley
P133996

Supervisor: Dr Kapilan Radhakrishnan
Second Marker: Dr Carlene Campbell

Project submitted as part of the requirements for
the award of BSc (Hons) Computer Networks

April 2016

DECLARATION

I, ...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.

Signed:

Date:

FORM OF CONSENT

I hereby consent that my Project, submitted in candidature for the B.Sc.(Hons) Computer Network 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.

Signed:

Date:

Abstract

IPv6 is the latest version of Internet Protocol formally introduced in 1998 [1], its main purpose to replace IPv4 and introduce new and improved features such as new address structure, simplified header and quality of service mechanisms. To provide a smooth transition from IPv4 to IPv6, various transition mechanisms were introduced [2] but since then very little has happened. Almost 20 years later only 8.72% of internet traffic is utilising IPv6 [3].

The purpose of this research study was to establish whether there is a direct connection between the performance results in native and tunnel IPv4/v6 networks and the poor migration statistics. In order to determine a conclusion this exploratory research project utilised Primary, Quantitative data achieved via software simulation scenarios. These scenarios have been carefully designed in order to provide consistent, valid data. Comparing this data and that of similar research, it was possible to view various trends that detail various performance issues IPv6 has over IPv4.

In multiple scenarios analysing round trip time, throughput, delay and packet loss there was a distinct difference in performance between both native protocols as well as one of the transition mechanisms. These results reflected those found by other researchers and it was determined that this may very well be a factor of the slow transition between protocols.

Table of Contents

Abstract.....	iii
Table of Figures.....	v
ACKNOWLEDGEMENTS.....	vi
Chapter 1 - Introduction	1
1.1 - Aim	2
1.2 - Objectives.....	2
1.3 - Report Structure.....	3
Chapter 2 - Research & Review of Literature	4
2.1 - History of Internet Protocol (IP).....	4
2.2 - Internet Protocol version 4 (IPv4).....	6
2.3 - Internet Protocol version 6 (IPv6).....	11
2.4 - Major Differences between IPv4 and IPv6.....	16
2.5 - Other Versions	17
2.6 - Transition Mechanisms	17
2.6.1 - Hybrid - Dual Stack	17
2.6.2 - Tunnelling	17
2.7 - Performance Considerations.....	19
Chapter 3 - Research Methodology.....	25
3.1 - Research Philosophy & Strategies.....	25
3.2 Software Simulation Package.....	27
3.2.1 Riverbed Modeler Academic Edition 18.5	27
3.2.2 QualNet.....	29
3.2.3 Why Riverbed Modeler?	30
3.2.4 Why Software over Lab Experiment?	30
Chapter 4 - Design & Implementation	31
4.2 Network Design	31
4.3 Scenarios, Applications and Statistics	32
Chapter 5 – Evaluation.....	34
5.1 ICMP Traffic	34
5.2 TCP.....	38
5.2.1 RTT	39
5.2.2 Throughput	40
5.3 Peer-to-Peer	42
5.3.1 File Transfer	42
5.3.2 Video Conferencing	45

5.4 Client/Server FTP	47
5.5 Conclusion	49
Chapter 6 - Conclusion and Recommendations	50
6.1 Conclusion	50
6.2 Recommendations for Future Work	51
Chapter 7 – Reflection	52
7.1 Main Aim	52
7.1.1 Was the aim achieved?.....	52
7.2 Objectives	52
7.2.1 How each objective was met.....	53
7.3 Issues faced during the project	54
7.4 Personal Reflection	54
References	55
Appendix	56

Table of Figures

Figure 1 - OSI and TCP/IP models	4
Figure 2 - IPv4 Header	6
Figure 3 - IPv4 Address Example	8
Figure 4 - Example of IPv4 Address classes	8
Figure 5 - Private Address Ranges	9
Figure 6 - Broadcast Address Example	10
Figure 7 - IPv6 Header	12
Figure 8 - IPv6 Address Example	14
Figure 9 - IPv4 Header added to IPv6 Packet	18
Figure 10 - IPv4 Header removed from packet.	18
Figure 11 Research 'Onion' [48]	25
Figure 12 Network Diagram in Riverbed Modeler	32
Figure 13 Ping (64B) Average Processing Delay	35
Figure 14 Ping (20000B) Average Processing Delay	36
Figure 15 Ping (65535B) Average Processing Delay	37
Figure 16 Ping Average Traffic Dropped.....	38
Figure 17 TCP Round Trip Time	39
Figure 18 RTT Comparison.....	40
Figure 19 TCP Throughput	41
Figure 20 P2P File Transfer Traffic Dropped.....	42
Figure 21 P2P File Transfer Traffic Received	44
Figure 22 P2P File Transfer Average Traffic Sent	45
Figure 23 Low/High Res Video E2E Delay	46
Figure 24 FTP Client/Server End-to-End Delay	47
Figure 25 IPv4-IPv6 Dual-Stack Remote Server Jitter [36].....	48

ACKNOWLEDGEMENTS

Dr Kapilan Radhakrishnan – For his support and supervision throughout this project.

Dr Carlene Campbell & Mabrouka Abuhmida – For their support, time and knowledge in assisting with software package decisions and help.

Family & Friends – I would like to thank my family and friends for their support throughout my degree and this project. More notably my partner Sian and Mum and Dad, without them I would not have managed.

Chapter 1 - Introduction

Internet Protocol (IP), introduced in September 1981 [4], is a major protocol found in network infrastructures and has been a fundamental factor to establishing user connections to the internet. The first main implementation of this protocol, IPv4, was developed to facilitate communication between government researchers in the United States [5], but has since been broadened to providing all users a means of network connectivity for over 30 years. This has been made possible with the use of IPv4 Addresses, but due to its structure, only 4,294,967,296 unique addresses [6] are available. At the time of deployment, this was considered more than enough but, 'it has become apparent that the number of available addresses is being exhausted [7]'. This is a major issue and one of the main reasons for the development and deployment of IPv6 in the 1990s [5, p. 4]. IPv6 provides the possibility of more than 340 undecillion (340,282,366,920,938,000,000,000,000,000,000,000,000,000) unique addresses [8].

It was known from the start of IPv6 design that the internet would not convert from IPv4 to IPv6 overnight. Therefore, suitable transition mechanisms were developed including Dual-Stack and Tunnelling. These allowed networks using IPv4 to change over in a staggered method and not dependant on other hardware or services. The transition mechanisms were described in 1996, RFC 1933 [2] "Transition Mechanisms for IPv6 Hosts and Routers" which has given organisations and network managers 20 years to change but according to Google Statistics [9] , still a very small percentage of networks are using native IPv6 or tunnel methods. This lack of migration may be caused by several factors; one being the cost of new equipment such as servers and routers, also the re-configuration headache that this new equipment brings. Many network administrators may be in the frame of mind that the network is running fine, why change it? Another reason why the migration could be slow may be down to performance issues faced during the transition or native IPv6 may have adverse effects on the functionality of networks.

Although IPv6 is a very broad subject, this project focuses on an evaluation of performance between IPv4 and IPv6 with a range of parameters using a traditional tunnelling method. More on this has been discussed further in this document.

1.1 - Aim

This experimentation project was conducted in order to establish:

Is there a direct connection between performance results in native and tunnel IPv4/v6 networks and the poor migration statistics?

As this project progressed it was possible to analyse the results obtained in the experiment against previous results gathered in the literature review, this comparison then aided to a better understanding whether the poor migration is due to performance issues.

1.2 - Objectives

For the main aim of this project to be achieved, a detailed set of objectives were required. The objectives stated exactly what would be completed in this project. The objectives set for this project were as follows:

1. Research studies relating to IPv4/v6 (with more detailed research focusing on performance) to gain an understanding on the current research position.
2. Analyse and discuss previous research statistics to show (if any) the current patterns in data.
3. Using a network simulation package analyse the Throughput, Round Trip Time(RTT), Delay and Packet Loss in the following network setups:
 - Native IPv4 and IPv6
 - IPv6 to v4 tunnel (6to4)
 - IPv4 to v6 tunnel (4to6)
4. Using the results from the experiment, evaluate the statistics and discuss the findings compared to previous research.

1.3 - Report Structure

In order to produce the most efficient results, the project including this report was given a structured approach. This report has been separated into various chapters, each including its own introduction and the content is then split into further sub-sections. Below is a brief overview of the structure for this report.

Chapter 1 – Introduction

Introduction to the report containing main aim and objectives.

Chapter 2 – Research & Review of Literature

Contains a detailed background research and analysis of previous papers on this subject.

Chapter 3 – Research and Development Methods

Contains the methods used to perform this project including the choice of simulation software.

Chapter 4 – Design & Implementation

Explains the reasoning behind the network design and the implementation of the experiment.

Chapter 5 – Evaluation

This section will refer back to previous work's results and compare between the results obtained in the experiments.

Chapter 6 – Conclusions and Recommendations

A conclusion on the project as a whole and any recommendations for future research.

Chapter 7 – Reflection

A reflection on the project and discussion on the outcome.

Chapter 8 – References

A list of all citations used to aid understanding in the report.

Chapter 2 - Research & Review of Literature

In order to complete this project and provide the most detailed results in the experiment, a background research on the subjects and previous work needed to be undertaken. In this section of the report will be a detailed explanation on all of the relevant topics from within the project and previous work that aids understanding.

2.1 - History of Internet Protocol (IP)

The current version of Internet Protocol (IP) widely used in networks is IP version 4 (IPv4). At the time of its introduction in 1983, the internet had previously been running on Network Control Protocol (NCP), which was developed by Advanced Research Projects Agency (ARPA) in the 1960s to allow communication between hosts [10]. It was realised that this method of communication was not sustainable with the future requirements of the internet, therefore Vint Cerf and Robert Kahn wrote an article in 1974 titled, "A Protocol for Packet Network Intercommunication" [11]. It was from within this article where Transmission Control Protocol (TCP) originated, "a protocol that transmitted a message" [6]. Forouzan [10] explained, after further development, TCP was separated into two protocols, Transmission Control Protocol (TCP) and Internet Protocol (IP). These two protocols operate in a hierarchical structure with TCP at the Transport layer and IP at the Network layer [12]. These protocols were utilised as a means of connecting multiple research institutes from within the National Science Foundation, which then began to build what has become known as the Internet [13].

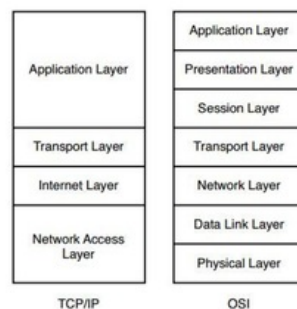


Figure 1 - OSI and TCP/IP models

As mentioned previously, IP operates at the Internet and Network layers of the TCP/IP and OSI models, which are shown in figure 1 taken from Casad's, Sams Teach Yourself TCP/IP in 24 Hours [13, p. 27]. TCP is beyond the scope of this project therefore the focus is on the IP part of the TCP/IP protocol suite. The main purpose of Internet Protocol as described by Kessler [14] is to provide "a datagram (connectionless) transport service across the network". This means that the services provided by IP are determined unreliable, as the source does not receive any acknowledgement that the data has arrived at the destination successfully. IP also has another important function, which is providing logical addressing to NIC's (Network Interface Card) that are used when routing data between two devices across a network. These logical addresses are more commonly known as IP addresses.

IPv4 has been a fundamental aspect of everyday communications for more than three decades running on hundreds of millions of devices all over the globe. Over that time, it has matured into a well-developed protocol but in the 1990's it was realised that the amount of IPv4 addresses were not enough to sustain the growth of internet users. NAT (Network Address Translation) and CIDR (Classless InterDomain Routing) have both supported the rapid expansion for quite some time but inevitably, the address space was not enough to fulfil the Internet's needs. The answer to this? IPv6 with its new 128bit address space.

Although IPv6 was introduced to solve addressing issues, the new protocol also provided additional features over IPv4 such as address type changes, auto-configuration and header simplification (more of these are discussed in detail in the IPv6 section of this chapter). It was always known that the internet would not switch from IPv4 to v6 overnight, and it would need to be a staggered change over as it was impossible to do it in one hit. To assist with the transition, Gilligan and Nordmark [2] devised a set of mechanisms as described in RFC 1933 'Transition Mechanisms for IPv6 Hosts and Routers', which was then updated in August 2000 with RFC 2893. It is from within these documents that tunnels were described. A tunnel is a method of encapsulating an IP packet inside another packet of a different protocol to be transported across a certain segment of a network (this will be explained further later in the chapter).

2.2 - Internet Protocol version 4 (IPv4)

Internet Protocol version 4, was first introduced by Cerf and Kahn [11] in 1974 as a more suitable method of communication across the internet. With it came various features including a dedicated addressing scheme, address classes and addressing types all of which are described in this section of the chapter. IP operates at the Internet layer of the TCP/IP suite and acts as the backbone of the layer, working alongside other protocols such as Internet Control Message Protocol (ICMP) and Address Resolution Protocol (ARP) [15]. The Internet layer provides similar services as the Network layer of OSI model, the main purpose being to provide addressing and route packets from the host across the network [16].

2.2.1 - Features

There are many features embedded in IPv4 that enable it to operate as it does. Some of the most notable features are; IP header, IPv4 Address format, types and classes and Fragmentation.

2.2.1.1 IP Header

The Internet layer receives the data from the upper layers in the form of a packet. IP then encapsulates the data adding an IP header to the packet containing various fields and then passes the frame onto the Network Access layer [17]. The IP Header was introduced in RFC 791 [4] and due to its variable options, the minimum header size is 160-bits. The format of an IPv4 header, shown below in figure 2 is taken from Shiranzai and Khan's, 'Internet Protocol Versions – A Review' [6].

Version	Header Length	Type of service	Total length			
Identification			Flag	Fragment offset		
Time To live		Protocol	Header checksum			
source address						
destination address						
Option				Padding		

Figure 2 - IPv4 Header

As the figure shows, the header consists of multiple fields that are described below;

Version The version of IP used.

Header Length Total length of the header in 32-bit words [15].

Type of Service Provides an indication of Quality of Service (QoS) required [4].

Total Length Length of entire IP packet.

Identification Used for packet reassembly for fragmented packets.

Flag Identifies if packet should be fragmented [6].

Fragment Offset Provides fragmentation and reassembly if packet is too large [15].

Time to Live (TTL) A measure added to a packet to prevent it from looping in the internet, each hop decreases the value by 1.

Protocol The type of upper-layer protocol used.

Header Checksum A value used to verify the header is not corrupted [17].

Source/Destination Address The addresses of the sender and receiver.

Option An optional field that can be used for various optional parameters.

2.2.1.2 Fragmentation

Another important feature provided by IP is the fragmentation of packets. Fragmentation is controlled in the fragment-offset field of the IP header. It is necessary to fragment the packet if it is too large to traverse the network in one frame. Fragmentation was described in the original IP RFC [4]. The receiving node uses the identification field to determine the order the fragmented frames are to be reconstructed.

2.2.1.3 IP Addressing Introduction

The final major feature of IPv4 is what the protocol is most commonly known for, IP addressing. IPv4 provides unique logical addressing for interfaces, these addresses are separated into classes depending on the total host and network bits required. IP also provides several addressing types depending on the circumstances. All of these features are described in detail in the following sections.

2.2.2 - IPv4 Addressing

An IPv4 address is a 32-bit dotted-decimal address formed by four 8-bit sections. The address is a logical address given to a single interface of a device. There is, in theory, a maximum of 2^{32} addresses or approximately 4.29 billion unique addresses [6]. An example of an IPv4 address is shown below in both standard and binary formats.

```
11000000.10101000.00000000.00000001
192 . 168 . 0 . 1
```

Figure 3 - IPv4 Address Example

IPv4 addresses are provided by IANA (Internet Assigned Numbers Authority) who distribute address spaces to Regional Internet Registry (RIR) organisations in a hierarchical structure as described in RFC 7020 [18]. IANA's address pool officially ran out in early 2011 [5, p. 3] although address allocation may become possible due to the release of IPv4 addresses as organisations slowly change to IPv6. BBC News states in an article written in 2015 [19], "North America has officially run out of...old net addresses", this proves that addresses may still be available but stock is certainly limited. When IANA first started assigning IP addresses to organisations they allocated based on classful boundaries.

2.2.2.1 - Address Classes

IPv4 addresses are distinguished into classes, class A through to D. This is done to separate networks down depending on the total host devices that require IP addresses. Below is an example of Class A, B and C addresses in decimal and binary form.

```
Class A  00001100.00001100.00001100.00001100
          10 . 10 . 10 . 10

Class B  10100000.10000000.00000000.00000001
          160 . 128 . 0 . 1

Class C  11000000.10101000.00000000.00000001
          192 . 168 . 0 . 1
```

Figure 4 - Example of IPv4 Address classes

A network administrator would acquire an IP address range from the ISP depending on the total amount of hosts on the network. Looking at figure 4 notice that each classful

address is highlighted in two colours. Each class is represented with a different portion of network bits (blue)/host address (green) and a subnet mask. Class A has 8-bits for network and 24-bits for host with a default subnet mask of 255.0.0.0, Class B with 16-bits equal for network and host with 255.255.0.0 being the classful mask and Class C has 24-bits dedicated to network and 8-bits for host with a mask of 255.255.255.0.

When looking at the addresses in a classful sense, meaning that they are not subnetted using VLSM (Variable Length Subnet Masks), Lammel [15, pp. 119-121] explains that; Class A provides 16,777,214 unique host addresses per network. Class B has 65,534 node addresses and class C networks contain a maximum of 254 nodes for each network. This format of addressing was being used up until 1993 when RFC 1518 [20] introduced Classless Inter-Domain Routing and Variable Length Subnet Masks, that provided an architecture for allocating IP addresses responsibly.

Before this was introduced, a network with 20 hosts for example would be assigned a class C address that can have 254 hosts, but as only 20 of these IP addresses were used, 234 addresses were wasted. This problem is amplified for class B and A, resulting in a fast depletion of available addresses. So VLSM and CIDR were introduced resulting in only the required amount of addresses being used up.

Class D and E are reserved address ranges, Class D runs through 224.0.0.0 to 239.255.255.255 and is used for multicast addresses. Class E address range is for scientific uses and runs from 240.0.0.0 to 255.255.255.255 [15].

There are also private address ranges that can be used on any device in a private network that is not routable to the internet. This is a convenient method to save IP addresses in conjunction with the use of Network Address Translation (NAT). NAT is the method of translating multiple private IP addresses into one or more public IP addresses depending on the size of the network. There is a private IP address range for each major class, which are displayed in the figure 5.

Class A: 10.0.0.0 - 10.255.255.255
Class B: 172.16.0.0 - 172.31.255.255
Class C: 192.168.0.0 - 192.168.255.255

Figure 5 - Private Address Ranges

2.2.2.2 - Address Types

Within an IPv4 network, three main address types are used by a source node to assist the network device in distinguishing the destination. Each type has its own purpose on the network

The address types are:

- Unicast
- Broadcast
- Multicast

Unicast addresses are a method of direct communication between the source and destination nodes. This is also known as one-to-one communication. A unicast address is a unique IP address given to a single interface.

Multicast addressing is a point-to-multipoint communication whereby an IP packet is sent to a multicast group address within the class D group as described in the previous section. All nodes with an IP address with this multicast address will receive the packet and respond accordingly. In a way, multicast works similar to broadcast packets; being multiple devices receive the packet and either send it up to higher layers or drop it if the IP address is incorrect, although routers do not forward broadcasts but do send multicast packets to the interfaces that contain routes to all devices within the multicast address range.

The final address type is Broadcast. There are two types of broadcast address; layer 2 broadcast and layer 3 broadcast [15]. Layer 2 broadcasts are sent to all nodes on the LAN using a hardware (MAC) address of FF.FF.FF.FF.FF.FF. These packets are not forwarded by a router past the LAN boundary but are received by all nodes within the LAN.

On the other hand, all nodes in a network receive Layer-3 broadcasts. There are two types of layer-3 broadcasts, the first is any host on a network. Figure 6 below provides an example.

Network Address – 192.100.0.0/24
First Usable Host – 192.100.0.1
Last Usable Host – 192.100.0.254
Broadcast Address – 192.100.0.255

Figure 6 - Broadcast Address Example

Any host in the 192.100.0.0/24 network would receive a broadcast packet sent to 192.100.0.255, an example of a broadcast use would be an Address Resolution Protocol

(ARP). Another layer-3 broadcast address that can be used is an 'any network, any host' address of 255.255.255.255.

2.3 - Internet Protocol version 6 (IPv6)

IPv6 has been in the making since the early 1990's and has retained many of the strengths obtained in IPv4. Version 6 also provides additional features and functionality that are missing from IPv4 some of which are discussed in the proceeding sections. IPv6's main improvement over v4 is the increase in address space by using 128-bit hexadecimal addresses over the 32-bit dotted decimal used in the current version. Due to the size of the internet, IPv6 has had a staggered deployment using transition mechanisms such as dual stack and tunnelling (discussed later in this paper) allowing the current v4 infrastructure to coexist with v6 until ISP's change to IPv6 allowing their clients to follow. Using a combination of Google Statistics [9] and Cisco Statistics [21] as of December 11th 2015 there was only 7.91% IPv6 traffic on the internet with just over 6500 alive v6 prefixes in use out of a total of 41500 allocated IPv6 prefixes. These statistics show the very little progression onto IPv6 at this stage, comparing from late 2010 to late 2015 the total allocated IPv6 prefixes have doubled from 20k to 40k [21]. Considering that IANA's IPv4 address pool was depleted in 2011 [5] this progression over a 5 year period is slow.

2.3.1 - Features

IPv6 was built upon the progress made over the years with IPv4. Many features remain the same with some improvements, but IPv6 also comes with some new features, the most notable are; Larger Address Space, Simplified Header, Auto-Configuration, Security Implementation and QoS Capabilities. In this section of the report, the above features are discussed in detail.

2.3.1.1 - Larger Address Space

Arguably the most crucial addition to IP is the increase in address space size. As mentioned previously in this document, IPv4 addresses have become very scarce even with the use of NAT and private addresses. An implementation within IPv6 is the change of address type to 128-bit hexadecimal. This drastically increases the amount of available IP addresses. A more detailed description on the IPv6 address follows in section 2.4.2.

2.3.1.2 - Simplified Header

The header has received a scale down with many fields found in the IPv4 header being removed or made optional [22]. The much needed header simplification is required to decrease routing times and packet handling by network devices thus increasing performance. Below in figure 7 is a diagram of the IPv6 header fields taken from Hanumanthappa's [8] paper comparing IPv4 and IPv6 feature sets.

Version	Class	Flow Label	
Payload Length		Next Header	Hop Limit
Source Address (128 bit)			
Destination Address (128 bit)			

Figure 7 - IPv6 Header

As the figure shows, the IPv6 header is much simpler than the example shown earlier for IPv4. This has been made possible with the addition of extension headers within IPv6. The header has a fixed length of 40 bytes unlike IPv4's varying size, consisting of two 16-byte addresses and 8-bytes of general information [5]. Any additional information is added to extension headers rather than the options field found in the base IPv4 header. There are currently six cases of extension header that must be supported by all IPv6 nodes [5], [1], [23], [24] these are;

- Hop-by-Hop Options Header
- Routing Header
- Fragment Header
- Destination Options Header
- Authentication Header
- Encapsulating Security Payload Header

The use of extension headers still proves to cause some teething trouble on the IPv6 internet. Fernando Gont [25] claims in an article regarding extension header filtering, that IETF have found when extension headers are used, public servers drop between 10% and 50% more packets. This is because such headers can provide additional security risks specific

to each header type therefore packet filters have been set to drop packets regardless of the reliability of packets. More research is required on this topic but it is beyond the scope of this project.

Taking a closer look at the fields within the header the descriptions of which are:

Version The version of IP used in this case 6.

Class Represents priority of packets.

Flow Label Indicates if the packet is part of a sequence.

Payload Length Specifies length of data that follows the header.

Next Header Indicates the protocol or extension header used.

Hop Limit Similar to TTL field in IPv4 header.

Source/Destination Address 128-bit address of source and destination nodes.

2.3.1.3 - Auto-Configuration

Auto-Configuration is an extremely useful tool built directly into IPv6 that allows devices to automatically assign both link-local and global unicast addresses (both are explained later in this chapter) to themselves. This is a great addition as it removes strain on DHCP servers that are used within IPv4 networks to provide interfaces with addresses from a given pool. DHCP servers will still be found within IPv6 networks as they can provide other information such as DNS and gateway addresses.

Address auto-configuration is performed in two stages [15, pp. 722-723];

- First, the device receives the network prefix from the router on the network.
- The device then uses this prefix along with its unique mac address and a 16-bit padding (FFFE) to generate a 128-bit IPv6 address.

2.3.1.4 - Security Implementation and QoS

During the progression of IPv4, IPsec was developed to provide security at the IP layer [26] although it was not utilized as effectively as it could have been. IPsec has three main functions; Authentication Header (AH), Encapsulation Security Payload (ESP) and Internet Key Exchange (IKE) [27].

In the early stages of IPv6, IPsec was required in all stacks with the use of IKE recommended [28] but in RFC 6434 this was downgraded to a recommendation [29] but

should still be considered when developing IPv6 networks. IPsec is an additional security measure that is built into the operation of IPv6.

With the increase in VoIP and Video traffic across the internet, QoS (Quality of Service) is a strong feature required within IPv6. With the incorporation of the Class and Flow Label fields within the IPv6 header, the QoS of IPv6 traffic is enhanced. Parra *et al* [30] states, "The flow label field reduces the average processing load...therefore reduces delays". One aspect of this project was to determine whether this case made by Parra does indeed become visible in the experiments.

2.3.2 - IPv6 Addressing

IPv6 addresses differ from IPv4 addresses. The format of an IPv6 address is 128-bit hexadecimal divided into eight 16-bit blocks separated by colons, an example of an IPv6 address can be found below in figure 8. Because of its 2^{128} size there are 340,282,366,920,938,463,463,374,607,431,768,211,456 unique addresses, which is enough to give every grain of sand on the planet multiple IP addresses [5, p. 18].



Figure 8 - IPv6 Address Example

Notice from the example that the full 128-bit address looks very complicated and difficult to remember, luckily the address can be shortened in some instances. Leading zero bits in a block can be dropped (:0012: will become :12:) and blocks of four 0's can be dropped and replaced either by a ':0:' or '::', but changing to '::' can only be done once in an address in order for the router to be able to determine how many bits have been shortened. For example 2001:0db8:0000:0000:0056:abcd:0000:1234 can be represented in the following ways [5, p. 20]:

2001:0db8:0000:0000:0056:abcd:0000:1234
2001:db8:0:0:56:abcd:0:1234
2001:db8::56:abcd:0:1234
2001:db8:0:0:56:abcd::1234

Although this abbreviation is useful, it can also cause some issues because of the vast amount of ways it can be done. All addresses from within a network should be abbreviated in the same manner and a record stored suitably in order to restrict any potential errors. RFC 5952 [31] 'A Recommendation for IPv6 Address Text Representation' suggests using the following rules when demonstrating IPv6 addresses:

- Drop leading zeros.
- Shorten address as much as possible using '::'.
- Must not use double colons to replace a single '0000' block.
- The longest run of '0000' blocks must be shortened, if there are 2 or more equal blocks the first must be shortened.
- Use lowercase letters.

2.3.3 - Address Types

Similar to IPv4, IPv6 also uses different address types depending on the purpose, but differ slightly. They are:

- Unicast
- Multicast
- Anycast
- Special Addresses

IPv6 Unicast and Multicast addresses are similar to IPv4, unicast identifies unique interface of a node and multicast is a group of interfaces where all addresses process the packet. Although there are multiple variations of IPv6 Unicast addresses as explained below [15, pp. 720-722].

Global Unicast Address – Similar to IPv4 unicast address, this address is a publicly routable address that can be used on the internet.

Link-Local Address – Link Local addresses are similar to private addresses that cannot be routed on the internet. The purpose of this address is to be assigned to devices on the same link, serial connection between routers for example.

Unique Local Address – These addresses are not intended for routing over the internet and are used in a LAN in the same sense as private IPv4 addresses within a network. Unique local addresses were designed to replace site-local addresses that were no longer used as of 2004 [15].

The new address that concerns IPv6 is an Anycast address. A packet sent to an anycast address will be delivered to the nearest interface within the address range [5]. This address type is referred to as One-to-Nearest.

There are also a range of special addresses that can be found within an IPv6 network. IPv6 has a loopback address with the same purpose as the 127.0.0.1 IPv4 loopback address, which is to test and troubleshoot the NIC. This address is represented as '::1' [5]. There are also a range of IPv6 addresses allocated for the transition period. RFC 4291 [32] outlines the majority of the IPv6 addresses including IPv4 addresses embedded inside an IPv6 address. There are also other RFC's that introduce addresses such as 6to4, 6rd, ISATAP and Teredo addresses all of which can be found in various tunnel configurations.

2.4 - Major Differences between IPv4 and IPv6

The major differences between v4 and v6 have been discussed throughout the previous chapter, to summarise these variances:

- **Address Space**
 - A larger address space increasing from 4 billion to 340 Undecillion possible addresses, removing the need of Network Address Translation.
- **Header Simplification**
 - The streamline of header fields and the inclusion of extension headers increases routing times and simplifies the routing process.
- **Auto Configuration**
 - The need of DHCP assigning IP addresses is gone, an IPv6 node can automatically assign a unique routable address to itself.

2.5 - Other Versions

Why IPv4 and IPv6? What happened to the rest? Although the first major implementation of Internet Protocol was version 4, versions 0 to 3 did exist but never left the experimental stage, same goes for IPv5. However, the development of these experimental protocols did aid the construction of both major implementation versions.

2.6 - Transition Mechanisms

As mentioned previously, it was always known that networks would not be able to change from IPv4 to v6 in one hit, therefore transition mechanisms were deployed to enable both protocols to coexist in the interim. There are various mechanisms but the most common methods used are; Dual Stack (Hybrid), Tunnelling and Translation. In a paper titled 'An Examination of IPv4 and IPv6 Networks: Constraints and Various Transition Mechanisms' J. Govil *et al* [22] describe a vast range of transition methods detailing key facts on each, within the conclusion Govil *et al* state "[There is] no single best solution for transition...there are a large set of IPv6 transition tools available".

2.6.1 - Hybrid - Dual Stack

Dual Stack networks, also known as hybrid, consists of both IPv4 and IPv6 operating together in the same network. A node can decide whether it is IPv4-only, IPv6-only or both IPv4/IPv6 [5]. The routers within the network must also be configured for both versions. Kalwar *et al* [33] states, "[Dual Stack is] the most widely used transition mechanism across the world". This is because the ease of implementation allowing both versions to run simultaneously within an internetwork during the transition.

2.6.2 - Tunnelling

Another popular mechanism is tunnelling, there are various versions of tunnels as described below but the main purpose is to encapsulate an IP packet into the header of protocol for forwarding between tunnel points. Tunnels can be configured by either the administrator or automatically using dynamic tunnel methods such as 6rd, Teredo and ISATAP.

2.6.2.1 - 6to4

An IPv6 to IPv4 (6to4) tunnel will most commonly be found during the initial transition period. This method uses an IPv4 network to connect two IPv6 end networks. At the tunnel entry point (usually going from the source network to ISP) an IPv4 header is added to the IPv6 packet as shown in figure 9

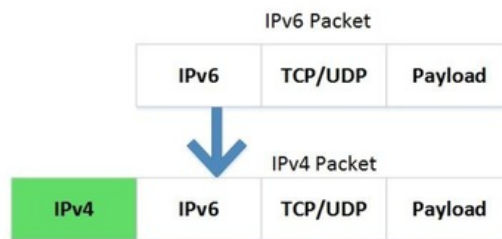


Figure 9 - IPv4 Header added to IPv6 Packet

The packet then traverses the internet until it reaches the destination IPv6 network at which point, the tunnel end point removes the IPv4 Header from the packet as shown in figure 10.

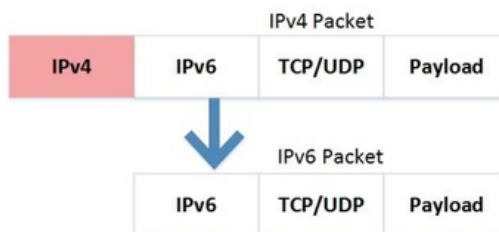


Figure 10 - IPv4 Header removed from packet.

2.6.2.2 - 4to6

IPv4 to IPv6 (4to6) tunnels are the inverse of 6to4 whereby IPv4 packets are encapsulated into IPv6 headers. This will become popular when the backbone network and

ISPs have changed over to native IPv6 and there are a few IPv4 end networks requiring a tunnel.

2.6.2.3 – Teredo

Teredo is designed to make IPv6 available through layers of NAT by tunnelling packets over UDP [5, p. 240]. Because many end networks utilise network address translation, certain issues have become apparent with other tunnel methods because of the use of private IP addresses. More detailed information on Teredo can be found in RFC 4380 [34]

2.6.2.4 – ISATAP

Intra-Site Automatic Tunnel Addressing Protocol (ISATAP) was designed to connect dual-stack hosts over an IPv4 network. Defined in RFC 5214 [35], an IPv6 node can communicate directly with an IPv4 node within an intranet. Govil [22] explains ISATAP is an ideal transition mechanism for campus networks as it allows IPv6 connectivity using the existing infrastructure as it is gradually migrated.

2.7 - Performance Considerations

As mentioned in the introduction of this document, the main aim was to determine whether there are any performance issues related to the transition stages of deployment. In order to gain a wider understanding, this section will focus on analysing results from previous researcher's findings. These results have been examined along with the quantitative data retrieved during the experiments and discussed in the evaluation section of this paper.

'Performance Measurements Analysis of Dual Stack IPv4-IPv6' [36] written by Febby Nur Fatah *et al* as part of the Advance in Information Technology 2013 conference discusses the different types of migration and the strategies used to combat these. They state, "[There are two main ways IPv6 can be implemented] Aggressive migration and Passive migration." Aggressive migration is the changing of infrastructure from IPv4 to IPv6 in one go, this is an expensive approach and as a result has been avoided by the majority of the networks.

Instead, a passive approach has been taken involving migrating the network bit by bit allowing both IPv4 and IPv6 to operate together. This was possible by using tunnelling methods such as 6to4, Teredo and ISATAP. These are some of the more common techniques used along with manual tunnel configuration [37].

In this paper, they tested Dual Stack devices comparing the IPv4 results against the same parameters for IPv6. In the RTT (Round Trip Time) test they found “IPv6 [took up to] 13.75% longer than IPv4 [to connect]” they explained this is due to the increased header size of IPv6 packets. One of the changes for IPv6 was the headers used for packets. IPv6 headers have become simpler than those found on IPv4 packets by removing five fields [38, p. 11], these are: Header Length, Identification, Flags, Fragment Offset and Header Checksum. Although the IPv6 header has lost this information, it has still doubled in size from the 20 byte IPv4 header [39] to 40 bytes [1] found in this new protocol. This increase is due to the size of IPv6 addresses, which are now 128-bits in length.

Another finding presented in this paper was that throughput was 1.97% better in IPv4 connections, another result that goes in IPv4s direction. Another addition in the IPv6 header is a flow label [40], the flow label is in place to allow routers access to the flow without having to dismantle the packet and then re-encapsulate it, this in theory should speed up the throughput experienced but in these tests has not. In a paper by Parra *et al* [30] looking at QoS mechanisms they state “Flow label field reduces... [the] processing load of the routers...reducing end-to-end delays” So in theory with less delay the throughput should be better, but in this case it does not seem that way so more research on this is required.

'IPv4/IPv6 Transition Technologies and Univer6 Architecture' [41] written by Jun Bi, Jianping Wu and Xiaoxiang Leng, goes into deep detail on the transition methods available including a new network architecture known as Univer6 [42]. In the paper it stated, “The commonly used transition mechanisms can be divided into three categories: Dual stack, Translation and Tunnelling” these have already been discussed in this paper, the main part of this publication is the tunnelling detail. The paper mentions there are four categories of tunnelling; “IPv6 over IPv4, IPv4 over IPv6, Tunnel traversing NATs and Other Tunnels” the important types for this project are IPv6 over IPv4 and IPv4 over IPv6. Bi *et al* [41] explained that IPv6 over IPv4 tunnels are necessary for IPv6 hosts to communicate with each other when there is no IPv6 link between them. “The general idea is to make the IPv6 packet as

the payload of IPv4 packet". This type of tunnel is important in the early stages of transition as there are little IPv6 backbones to support native networks. IPv4 over IPv6 networks are still not very common and will be more visible at the later stages of IPv6 transition just before a native IPv6 network is achieved. Later in the paper it goes onto list some of the guidelines for transition mechanisms; Scalability, Security, Performance, Functionality, Requirement, Ease of Use and Ease of Management. The important ones for this project are; Scalability, Performance and Functionality. Although IPv6 was designed to improve some of the security weaknesses from within IPv4, the implementation of both IPv4 and IPv6 has created some security threats itself. The authors describe how using an IPv6 over IPv4 tunnel allows an attacker to bypass a firewall into a network because the firewall does not check the packet that had been encapsulated within the IPv4 header. Within the conclusion of this article, the authors made several statements that make this project worth doing. They explained [41, p. 242]; "Current research...[is] focused on...IPv6 over IPv4...more research on IPv4 over IPv6 transition methods is necessary." This project will focus on both tunnelling methods as well as native use of each protocol. On the same page, it then goes onto say, "Some wireless consideration should be introduced into the discussion". With more and more devices becoming wireless compatible the need for wireless IPv6 research is required and with the incorporation of Mobile IPv6 [43] wireless operation for IPv6 is improving and now that Home and Care-of addresses [38, p. 132] are in place, IPv6 wireless QoS will continue to grow.

'Empirical Performance of IPv6 vs. IPv4 under a Dual-Stack Environment' [44] written by Law, Lai, Tan and Lau details analysing results of performance tests carried out in a dual stack environment.

Within their research in this paper, they used a testbed of over 2000 dual-stack hosts worldwide. For the results collection they used seven machines, the results for this paper were collected between 2006/2007 so still early on in the IPv6 development world if we take the estimate as described by Hunumanthappa [8]. The performance measures gathered were; Connectivity, Hop Count, RTT, Throughput, OS Dependence and IPv6 Address Provisioning. In comparison with some of the other research papers in this field, the same parameters seem to appear in each, meaning that these are the most relevant to the impact of the changeover.

Looking at the results from this paper, the first test was on the connectivity from both the weekday and weekend. IPv6 sites had a great result of over 95% on both tests beating IPv4, this shows that IPv6 networks are on the increase. In the hop count tests IPv6 networks resulted in fewer hop counts, this may be because there are less native IPv6 networks for the traffic to be routed through. The Round Trip Time results were not so good for IPv6 compared to IPv4 at 403.36ms and 272.78ms respectively. Again, the reduced routers that were supporting IPv6 at the time have more than likely determined this result. Making the trip from point to point longer and results from the paper discussed earlier [36] show the same. In the throughput test, IPv6 came out on top also with a 30KB/s higher average at 107.75KB/s. An interesting find in this also states that in the results from this test IPv6 achieved a higher throughput the bigger the file size sent was, so this is an important factor to use in the tests for this project. The test on OS Dependence is irrelevant for this project although it did find out that, of three OS's (Fedora Core 5, FreeBSD 6.1 and Windows XP) Windows performed the worst.

The final sections of the paper compare IPv6 tunnel networks against native IPv6. Within these results, there are not many findings that can be regarded as extremely important, the tests also found that native IPv6 networks performed better which is understandable.

'Evaluation of 6to4 and ISATAP on a Test LAN' [45] written by Aazam *et al*, all respected members of universities in Pakistan and France. This paper similar to the others mentioned above, evaluates performance between two tunnel methods; 6to4 and ISATAP (both discussed in section 2.6.2). The authors of this paper consider out of the three transition mechanisms, "Tunnelling stands out as the best possible solution" [45, p. 46]. This statement is purely personal preference and a detailed project comparing the three methods would need to be carried out in order to conclude on the 'best' solution.

The test bed used in this paper contained five devices consisting of: two hosts, two routers and a DNS server. The performance statistics analysed were; Throughput, End-to-End Delay, Jitter and RTT. Again, these have appeared in many other experiments and therefore appear to be the most valuable data to compare the given networks. In order to achieve a mixture of results, three traffic types were simulated; "UDP...audio streaming, video streaming and...ICMP-ping traffic".

The paper begins with descriptions of each tunnel type, which have been covered in this document. In the setup description, the authors describe that, "The most widely followed way to evaluate performance of protocols is to simulate them using [common simulations tools like]... OPNET. OPNET (now known as Riverbed Modeler) is discussed in section 3.2.1 of this paper and is the choice of software for this project. They state the reason behind a physical test bed for their experiment is due to, "no simulation patches available for these techniques" although this has since been updated. Now looking at the results, while they are not directly comparable to this project (as it compares two tunnel methods) the data is still valid to reference with this project. In the End-to-End Delay (E2ED), Throughput and RTT tests, 6to4 performed worse off than ISATAP, with 6to4 only better in the Jitter test. The average throughput came out as 33Kbps faster with ISATAP, E2ED was 0.07 milliseconds worse with 6to4 and RTT is 0.16 milliseconds less with ISATAP. The Jitter test resulted in 6to4 being 0.007 milliseconds better on average. Aazam *et al* determine this to be due to ISATAP traffic required to traverse more devices in the networks.

In the conclusion to this paper, the authors determine, "ISATAP would be the most efficient and economical solution to tunnel IPv6 packets over IPv4 networks". They then state that 6to4 would be better for real-time traffic as it is less jittery. This may be due to the IPv6 QoS mechanisms being preserved in the traffic whilst using a 6to4 tunnel.

The final paper to discuss titled 'Quantifying the Performance Degradation of IPv6 for TCP in Windows and Linux Networking' [46] was written by Burjiz Soorty and Nurul Sarkar.

In this paper, they compare IPv4 with IPv6 in a directly connected client to server test bed. In their related work section they state a previous study by Kolahi and Soorty [47] determined "IPv4 performed better than IPv6 by 2.7% in a peer-to-peer setup and by 4.6% on a client-server network". This provides more focus for this project in order to test both peer-to-peer and client/server scenarios to evaluate any differences.

This experiment tested a client/server network using two common server operating systems, Windows Server 2008 and Red Hat Server 5.5. Again, the parameters are TCP Throughput and TCP RTT. In the throughput test, it was apparent that IPv4 performed slightly better than IPv6 in both cases. IPv6 achieved 5.2% lower throughput than IPv4. They determine, "IPv6 deteriorates throughput as a result of its high transmission overheads

(larger header)” [46, p. 27]. This was mentioned previously that, although the IPv6 header is simpler than IPv4’s, due to the increased address size the header itself is larger in size.

Taking a look at the Round Trip Time statistics, again IPv4 performed better in both tests and was better with Windows Server 2008. In this analysis, they claim that the results differ depending on packet size therefore a varied packet size will need to be carried out in this project in order to provide a detailed set of results.

In conclusion to the previous research, it is apparent that the most important parameters are the ones that appear repeatedly; RTT, Throughput, Delay and Packet Loss. Although the research papers discussed do not compare IPv4, IPv6, 6to4 and 4to6 together therefore this project will provide new and valid information. The consensus is that IPv4 performs better, the experiments will be carried out in this project before coming to a conclusion and have been discussed in detail in chapters 5 and 6.

Chapter 3 - Research Methodology

Various research methodologies can be utilised for projects such as this. The research method section describes how the research will be conducted [48]. There are many elements that come together to finish the research method for a project, these elements are built in stages within the research 'onion' [48, p. 128]. In this section of the report, a detailed research method approach used for this project can be found.

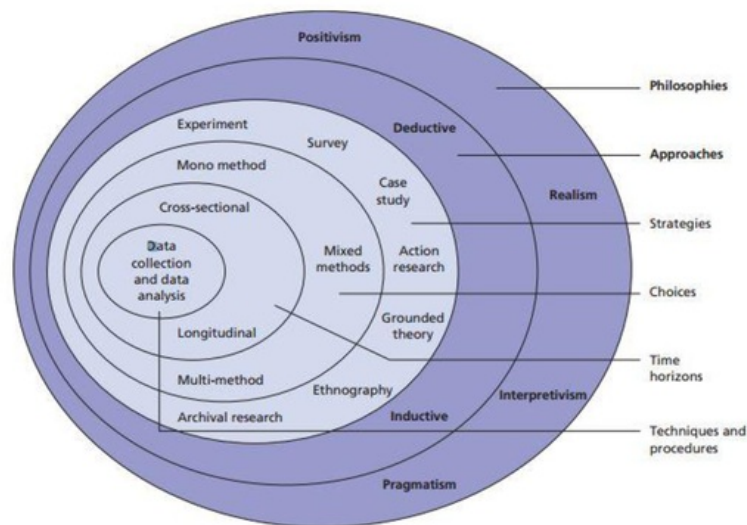


Figure 11 Research 'Onion' [48]

3.1 - Research Philosophy & Strategies

The research methods can be split down into various stages, the first step was to define the research purpose. There are three kinds of research purpose, Exploratory, Descriptive and Explanatory [49]. The purpose chosen for this project was Exploratory, this was chosen because exploratory research is designed to clarify or identify a problem as does this project, the aim is to determine whether the slow transition of IPv4 to IPv6 is caused by performance issues.

The researcher collected the data used for this project via network simulation therefore; it is classed as Primary (field) research. The data collected was quantitative meaning statistics (hard data) will be available for analysis. These methods have been chosen to insure that the quality and reliability of the data is as high as possible.

The next step was to decide on the Epistemology or research philosophy [48, pp. 132-136], there were two to choose from; Positivism and Phenomenology. The philosophy chosen for this research project was Positivism. A positivist approach is “based on observation...of a real phenomena...using experiments to observe these” [50, p. 11]. This method was chosen because in comparison with the Phenomenological approach it is best suited for a project of this manner as it includes the use of quantitative data, tests theory and the data produced is specific.

As with the philosophy, there were also two approaches to choose from, these are Deductive and Inductive, the method chosen for this project was Deductive. Burney [51] explains “[the deductive approach] is known as a 'top-down' approach”, it works off a waterfall model going from a Theory > Hypothesis > Observation > Confirmation. The theory and hypothesis have already been discussed for this project, the next step will be to run the simulation in order to gain data to observe, with this data the project will then be able to confirm whether the aim is indeed correct or whether there are other reasons behind it.

The final step was to define the strategies that were to be used, it was decided that the two main strategies would be Experimental and a Document Review. The literature review has been completed within this document. The aim of the review was to identify what studies have already been taken in this field, this work could then be used to compare either whether the project outcomes agree or disagree with other research. In order to gain creditable data for this project, experimentation is going to be used in the form of simulation. The data obtained from the tests have been compared against each other to provide sufficient results to be able to determine an answer for the project aim.

In summary to the above, the research methodology for this project are as follows:

- Exploratory research with the use of Primary, Quantitative data.
- Positivist Philosophy.
- Deductive Approach.
- Strategies to be used are Experiment and Document Review.

If we look back at the research onion, we can see that a relationship is formed from the choices used for this project. All of the methods are in the same 'section' of the onion therefore the flow should be smooth.

3.2 Software Simulation Package

As mentioned previously this project utilised primary, quantitative data therefore a means of collecting this data was required. There were two main ways of accomplishing this; Software simulation or lab testing. In this section is a discussion of some suitable packages followed by the reasoning behind the choice of software rather than lab experiment.

There are many software simulation packages available to network enthusiasts and professionals. The choice of such depends on the requirements; examples include Cisco's Packet Tracer, GNS3, Riverbed Modeler, QualNet plus many more. All of these packages offer various tools for network design, configuration, troubleshooting and analysing. Choosing the appropriate package will ensure the optimum results can be achieving based on requirements.

The project was conducted in order to analyse performance between IPv4 and IPv6 therefore software that allows result collection was required. After careful research, the software chosen to simulate the network was Riverbed Modeller Academic Edition 17.5 [52]. The choice was between Riverbed Modeler and Qualnet. This was because these packages allowed detailed statistical analysis rather than a focus on simulating a network with detail on configurations. The following sections contain a comparison between Riverbed Modeler and QualNet and the reasons behind the choice of Riverbed Modeler.

3.2.1 Riverbed Modeler Academic Edition 18.5

Riverbed Modeler, formally known as OPNET Modeler (as well as IT Guru Academic Edition), is a package offered by Riverbed Technology Inc that includes; "tools for...model design, simulation, data collection and data analysis" [53]. The version tested and used for simulation was version: '17.5 Academic Edition'. This version is free to all users and is based on the full version, Riverbed Modeler 17.5 PL6.

3.2.1.1 Riverbed Modeler Advantages

One of the major features contained in this package is the vast amount of models available. This package contains both node and link models as well as the ability to simulate wired and wireless scenarios. It is possible to simulate devices from a wide range of the most common vendors including; Cisco, HP, 3Com, Dell and many more. This is perfect to simulate a real world network using the devices found mounted in racks within a functioning network. It is also possible to simulate a wide variety of links depending on the interface type and speed.

Another positive point is the selection of applications available ranging from simple ICMP traffic to simulated video conferencing. Linked with each application is the choice of detailed statistics, which can be selected, based on the protocols. For Example, this provides the ability to analyse results at the IP layer but alter the application to see the effect different traffic types have on the protocol.

Additionally Riverbed offers both a free and premium package that offers flexibility depending on budget and requirements. The Academic Edition (Free version) contains many of the features found in the premium version.

3.2.1.2 Riverbed Modeler Disadvantages

One of the main cons for the Academic Edition of this package is the restricted capabilities compared to the full version [54]. Although compared to other free/premium packages, Riverbed are generous with the features contained in the free version. One of these restrictions resulted in a network design change during the course of this project (discussed later in this chapter). This restriction being a limit to the total nodes and events permitted per simulation. Another restriction is the lack of support for specialised models such as LTE and WiMAX although this does not have a direct impact on this project.

Because Riverbed Modeler is a port over from OPNET (an older simulation package), the interface is a bit dated and can seem overwhelming on first sight. The toolbar buttons are not obvious of their purpose compared to those found in other packages.

During research, it was difficult to find help/tutorial materials from a range of sources on the recent versions of the software. This could cause potential issues with the configuration

using this software although one recommended text that became of great use during this project is, 'The Practical OPNET User Guide for Computer Network Simulation – Adarshpal S. Sethi and Vasil Y. Hnatyshin [55]

3.2.2 QualNet

QualNet is a “communications simulation platform [for] planning and testing...the behaviour of a real communications network” [56] provided by Scalable Network Technologies. QualNet offers various components including; Network architect, analyser and packet tracer.

3.2.2.1 QualNet Advantages

One of the elements provided in QualNet that makes it easier to use is the user-friendly interface and the incorporation of animations during simulation. QualNet offers a range of scenarios similar to Riverbed Modeler but Qualnet provides the option of 3D views with simulation options such as buildings and object to simulate the effect this may have on the results.

Another pro when looking at QualNet is the ability to create large scenarios with many nodes whether it be fixed or mobile. QualNet provides the ability to simulate LAN, MAN, WAN just to name a few.

There are also notable differences when comparing the results analysis stages between QualNet and Riverbed Modeler. With Riverbed, statistics need to be selected prior to simulation whereas with QualNet all statistics are provided after simulation allowing the user to select 'on-the-fly' which data is relevant.

3.2.2.2 QualNet Disadvantages

A major drawback with QualNet is the lack of a free/trial version. This can be restricting to users due to issues with licensing in multiple locations and one of the main reasons against using QualNet for this project. Another issue discovered during testing is the frequency of bugs, some only minor errors but one that halted network testing and became a major factor in the choice of software for this project.

This next disadvantage may not be relevant to everyone depending on requirements but with relation to this project, QualNet is structured for wireless and mobile nodes for simulation purposes and does not offer a large range of vendors unlike Riverbed Modeler.

A disadvantage with QualNet is also related to an advantage mentioned previously, all statistics can be selected after the simulation has run. This can cause the simulations to take longer than they do in Riverbed because of the inclusion of unnecessary data.

Again, similar to a disadvantage with Riverbed, the amount of support material is limited for QualNet making configurations more complicated for users with little/no experience.

3.2.3 Why Riverbed Modeler?

After reviewing both packages, the most appropriate choice of software for this project was the Riverbed Modeler Academic Edition 17.5. This was determined by evaluating the PROs and CONs of both packages. In summary the main reasons for choosing Riverbed Modeler were:

- Wide Range of Vendor equipment as well as ability to create custom nodes.
- Wireless and Wired Links.
- Statistics can be selected based on protocol and application type.
- Freely available.
- Scenario duplication allows easy change of protocol version (IPv4/v6) whilst all other variables remain unaffected.

3.2.4 Why Software over Lab Experiment?

It was decided early in the project to use software packages to simulate a network in order to gather results although another option was possible. Setting up a test-bed within a lab would provide a set of results achieved in a real network. This would result in 'realistic' effects each protocol experienced on a real network.

Although this would have been ideal, it was not feasible to experiment on a test-bed over software. Funding for the correct devices was not available and analysing results achieved on a real network is harder than that found in a software package and often requires specialist software on the devices in order to capture the data required for this experiment.

Chapter 4 - Design & Implementation

Before any statistics could be obtained and analysed, a suitable simulation package needed to be chosen and a base network design completed. There were various choices for software as will be discussed in this section, following this is the reasoning behind the chosen network design and the applications used for statistical analysis.

4.2 Network Design

In order to assure the results collected were appropriate for this project, a suitable network needed to be designed. Original plans were to design a simulated, world network plan consisting of many routers connected throughout various countries and have traffic simulated between nodes located in different countries. The network also contained a simulated representation of some Transatlantic Cable networking that connects various continents over the world. This would provide a realistic view of how traffic is affected over a long distance, as it would travel in the real networks. The plan and addressing scheme for this can be found in appendices 1 and 2. After implementing this design in the software package chosen it became apparent that it was not possible to have such a large design with many nodes due to software restrictions.

Although this set back was not ideal, the project could still go ahead using a smaller, office type scenario (Figure 12). The size of the network was not a major factor in obtaining results, the most important factor was the consistency in network in order to provide valid data to analyse.

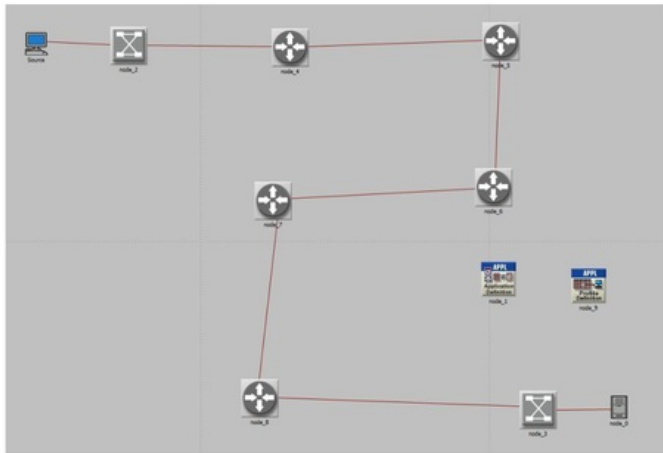


Figure 12 Network Diagram in Riverbed Modeler

The mainframe of the network remains the same for each scenario. Five generic routers connected to neighbour routers via 1GBps Ethernet links, the gateway routers (nodes 4 & 8) are then connected to each LAN via a 100MBps link to a generic Ethernet switch. It was decided to use generic devices (although Riverbed provides many vendor equipment) in order to provide a stable environment with as little as possible influence on results based on different vendor processing.

Depending on the scenario (discussed in the following section) the end nodes are as follows; two advanced Ethernet workstations, one source and other destination connected to appropriate LAN switches. OR, one advanced Ethernet workstation as source and an Ethernet server as destination as demonstrated in figure 12 above.

The software has automatically allocated IP addresses for this network for convenience purposes, although in the original design a suitable address scheme (appendix 2) was designed as if it were a real network.

4.3 Scenarios, Applications and Statistics

The main aim as defined in section 1.1 is to determine whether any performance issues are linked to the slow migration from IPv4 to IPv6. This was to be done using a variety of parameters (RTT, Throughput, Delay and Packet Loss) in four network configurations (Native IPv4, IPv6, 6to4 Tunnel, 4to6 Tunnel). Using the results from each of these tests would then provide enough material to evaluate the case. Whilst implementing these scenarios it

became apparent that due to software restrictions certain aspects had to be altered. The incorporation of 4to6 tunnel data has been removed from the project at this stage. It was not possible in neither QualNet nor Riverbed to configure 4to6 tunnels, which was unfortunate. This aspect would have provided an overview of any issues that may be apparent towards the end of the transition period. This has been added to the recommendations section towards the end of this document. Another minor change was with the statistics provided. RTT, Throughput, Packet Drop rate and Delay have all been collected with slight variations, more information of these changes can be found in the evaluation.

In order to simulate any potential effects caused by different applications, multiple scenarios have been implemented. The main scenarios that have been used to collect data are as follows; ICMP traffic (Ping) from one workstation to another, using a variety of packet sizes including 64 bytes, 20000 bytes and 65535 bytes (maximum possible IPv4 MTU size [39]). This is to simulate how each version of IP handles varied size traffic. It was also recommended to analyse how the protocols manage with both high and low priority traffic, therefore several scenarios have been setup that contain; Video Conferencing (Low and High Resolution) between two clients, FTP requests (Low and High Load) from a server and Peer-to-Peer File Sharing between two clients. This allows us to see the results from transporting different protocols (TCP, UDP etc) and whether QoS mechanisms have any effects on results.

Chapter 5 – Evaluation

After the network had been designed and implemented, it was finally time to gather and analyse the results from the scenarios. In this chapter, a detailed analysis of the data will be able to determine whether there are performance issues comparing IPv4, IPv6 and 6to4 tunnels that has delayed the transition between major protocols. The following sections have been divided into sub-sections with more in-depth discussion on each scenario/application and the results achieved. Each graph contains three data sets: Native IPv4, Native IPv6 and 6to4 Tunnel. The experiment run times alter depending on the scenario and will be discussed in each sub-section.

5.1 ICMP Traffic

“ICMP (Internet Control Message Protocol) provides query and error messages for effective communication between host and gateways in IP networks” [57]. ICMP has versions in both IPv4 (ICMPv4) and IPv6 (ICMPv6) and provide similar services in each existence. Ping is a common network tool that utilises ICMP echo queries. In this experiment, three ping packet sizes were configured in order to analyse any effects larger traffic causes. The three MTU sizes were 64 bytes (minimum size for MTU), 20000 bytes and 65535 bytes (largest size for v4 MTU). The parameters analysed with this data were Drop Rate and Process Delay.

The first chart (figure 13) shows the average processing delay calculated on node 4 (figure 12) whilst sending ping traffic of 64 bytes. This test was carried out over a 10-minute period (maximum allowed for ICMP traffic in Riverbed) as indicated on X-Axis in seconds. The Y-Axis contains the average processing delay in seconds (0.000025 equates to 0.025 milliseconds for reference). The data represented is as follows; IPv4 = Blue, IPv6 = Yellow and 6to4 = Green.

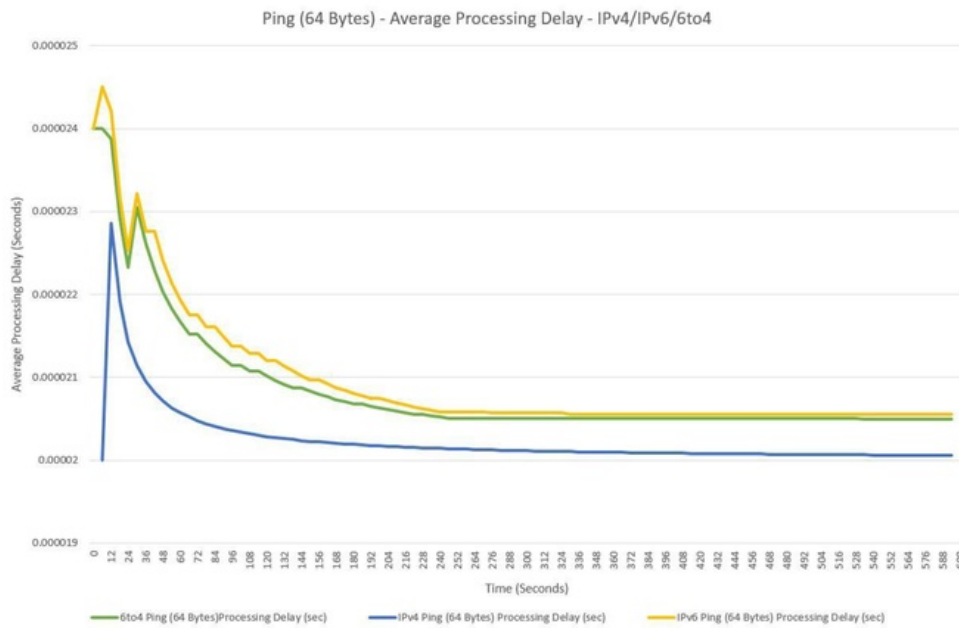


Figure 13 Ping (64B) Average Processing Delay

On first inspection of the results, it is clear that IPv6 has performed the worst, marginally followed by 6to4 tunnel and IPv4 with the least delay. As the graph shows, at the initial stages of the test each protocol has a large spike before settling down at the later stages. This original increase in delay happens within the first minute of the scenario. It is believed that this is due to extra convergence traffic traversing the network upon start up. At its largest, the gap between IPv4 and IPv6 is a difference of 7.45% or 0.0017 milliseconds, which in real terms would not be noticeable. This difference maybe because of the increased header size of IPv6 packets as discussed in section 2.3.1.2 and referenced in previous research as a factor for performance issues.

Another factor to notice from the results is the relationship between IPv6 and 6to4 tunnel data. Both protocol versions seemed to mirror each other, whereby when a spike occurred with one it is also apparent in the other. 6to4 tunnel provided a slightly lower delay rate than IPv6, which was not expected due to the extra workload the routers are required to undertake with the encapsulation of packets during tunnelling. The mirror relationship between the two is suspected to be as a result of the native protocol being IPv6 and therefore the traffic would be processed similarly by the routers prior to encapsulation.

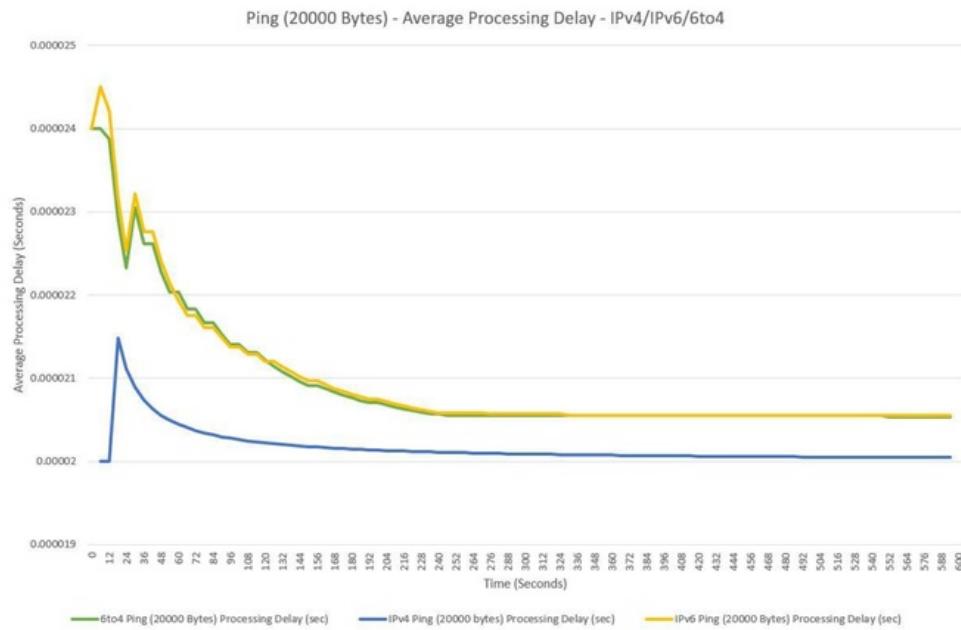


Figure 14 Ping (20000B) Average Processing Delay

The second chart for ICMP traffic shown above (figure 14) is the results of a similar scenario to the first experiment with the only difference being the packet size. In this test, the ICMP traffic is 20000 bytes sent over a 10-minute period. Straightaway it is obvious that again, IPv4 performs the best throughout the test. Taking a closer look at the IPv6/6to4 statistics, it appears that within the first 60 seconds there is very little difference between the results in the first test. It is only after this where it is evident of minor changes. Again, as in figure 13, IPv6 and 6to4 seem to continue the mirror relationship but in this case, IPv6 seems to perform slightly better than 6to4. This may relate back to some of the analysis found in previous research (section 2.6.3) where IPv6 seemed to perform better with a larger file size. Although in this scenario, IPv4 is the clear winner.

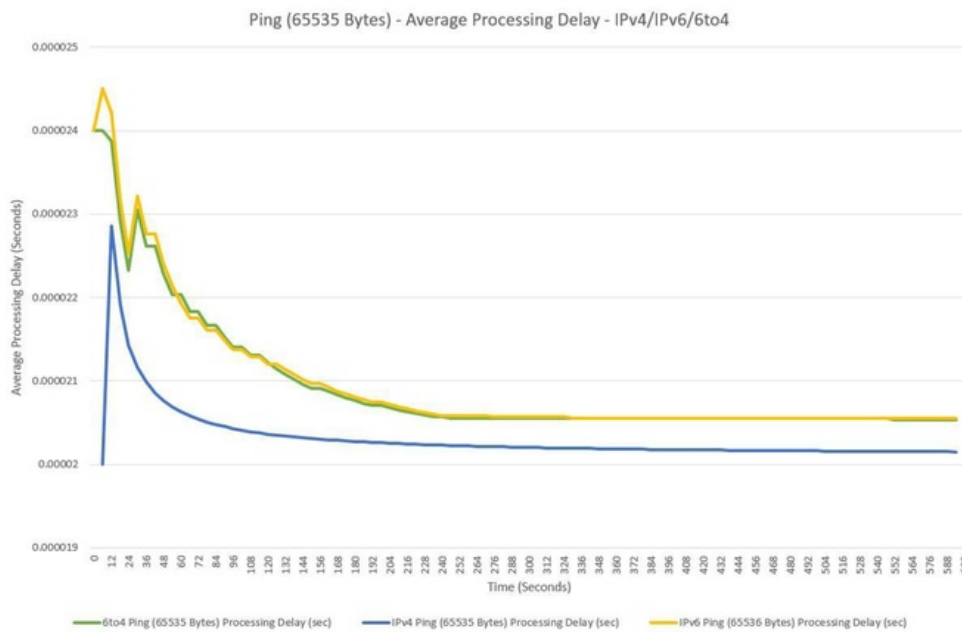


Figure 15 Ping (65535B) Average Processing Delay

The final Ping Delay scenario (Figure 15) results in a combination of results from the first two. IPv6/6to4 results appear to be the same as experiment 2 and IPv4's have crept up to those similar to experiment 1. In this experiment, the traffic size was set to 65535 bytes, which is the largest available MTU size for IPv4 traffic. The only new data that can be analysed from these results is the deterioration of IPv4 performance although out of the three it is still the best.

Another ICMP scenario implemented was to see the difference between IPv4/IPv6/6to4 with packets dropped. This scenario uses ping traffic at 20000 bytes in size. As can be seen in the results within figure 16, a similar outcome to the previous three statistics where IPv4 performed the best followed by a close 6to4/IPv6. An initial spike is most likely caused by router convergence using RIP and RIPng as routing protocols. IPv6 and 6to4 provide the same outcome throughout the test apart from three moments. This data may provide a clearer view as to why there was a longer delay for IPv6 and 6to4 in the previous scenarios. Each packet that is dropped will need to be re-sent by the source, causing the router to

receive higher loads and creating delay. Which may be directly connected to the delay results achieved.

Several distinct things can be witnessed from these results. First off, all three carrier protocols experience a spike in the initial stages as mentioned before. However, as IPv6 and 6to4 slowly begins to descend, IPv4 does not show any signs of decrease for nearly 4 minutes. This shows that IPv6 can recover faster than IPv4 in such instances.

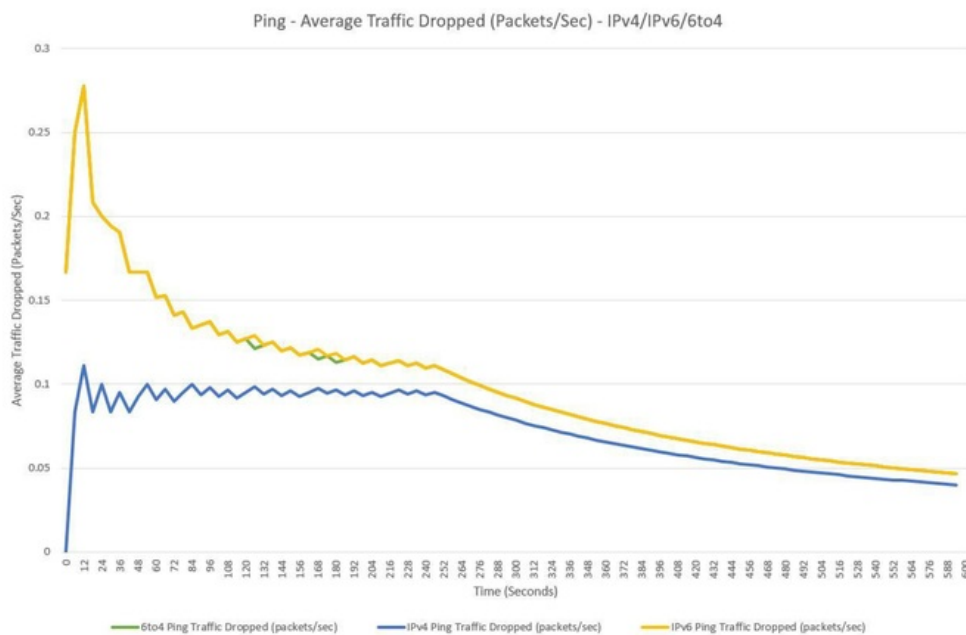


Figure 16 Ping Average Traffic Dropped

5.2 TCP

The next set of tests is to analyse how IP works with Host-to-Host layer protocols such as TCP [15, p. 96]. TCP is commonly classed as 'connection-oriented' traffic, therefore each time two hosts want to communicate a dedicated connection needs to be established with the use of handshakes and acknowledgements. It is with these methods that the next set of data becomes available; RTT provides the time (seconds) it takes for the source to send a packet and then receive an acknowledgement. Throughput is a measurement (Mbps) of the rate of traffic that can be successfully delivered across a network.

5.2.1 RTT

The data from the Round Trip Time test shown below in figure 17 comprises of three aspects: IPv4 (Green), IPv6 (Blue) and 6to4 (Yellow). The axis are as follows: X-axis = Protocol version, Y-axis = RTT in Seconds.

IPv4's RTT came in at 0.83 seconds, IPv6's at 1.27 and 6to4's at 1.15. Firstly paying attention to IPv4 and v6, there is an increase of 0.44 seconds = 53%. In terms of network speed this is drastic and the more congested the network because the more obvious these statistics will have an impact on user experience. Secondly taking a closer look at IPv6 and 6to4, it is apparent that 6to4 performed 10% faster than Native IPv6. Evaluating the general trend of this project so far it is clear that a pattern is beginning to show whereby IPv4 is out performing IPv6 and 6to4 in each test.

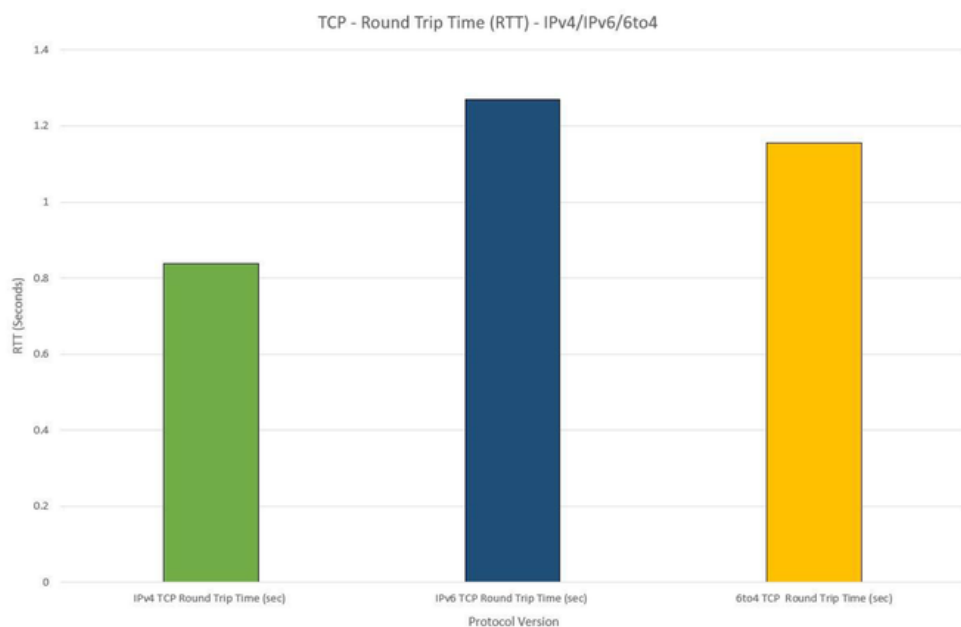


Figure 17 TCP Round Trip Time

If we take a closer look at these results in comparison to the RTT results analysed by Law *et al* [44] as shown in figure 18 it is visually clear to see the same pattern of data forming. On the left side of the graph are the two data sets obtained in this project along with the two sets reported by Law. The actual difference in speed between both sets is irrelevant as

different network sizes and configurations would have a factor on results. What is important to note is the resemblances between IPv4/v6 in each set. As mentioned previously there was a 53% difference in this project, similarly Law *et al* recoded a 48% difference between the two protocols. In Law's reasoning, he states, "The longer propagation delay experienced by IPv6 translates to higher IPv6 RTTs". This may also explain the same results witnessed in this project if looking back at the processing delay discussed earlier.

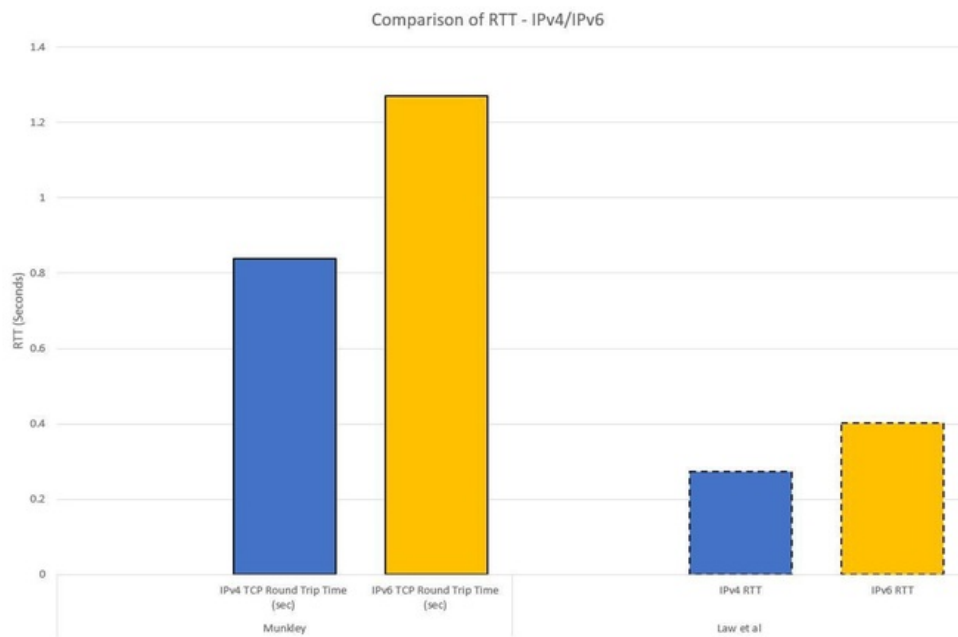


Figure 18 RTT Comparison

5.2.2 Throughput

Moving on to throughput data from IPv4, v6 and 6to4 again the data trends continue to show (figure 19). IPv4 achieved an average throughput of 6.55 Mbps, followed by 6to4's 4.36 Mbps and IPv6 coming in last at 4.03 Mbps. IPv4 maintained a 62% speed increase over IPv6 and a 50% increase over 6to4 tunnels.

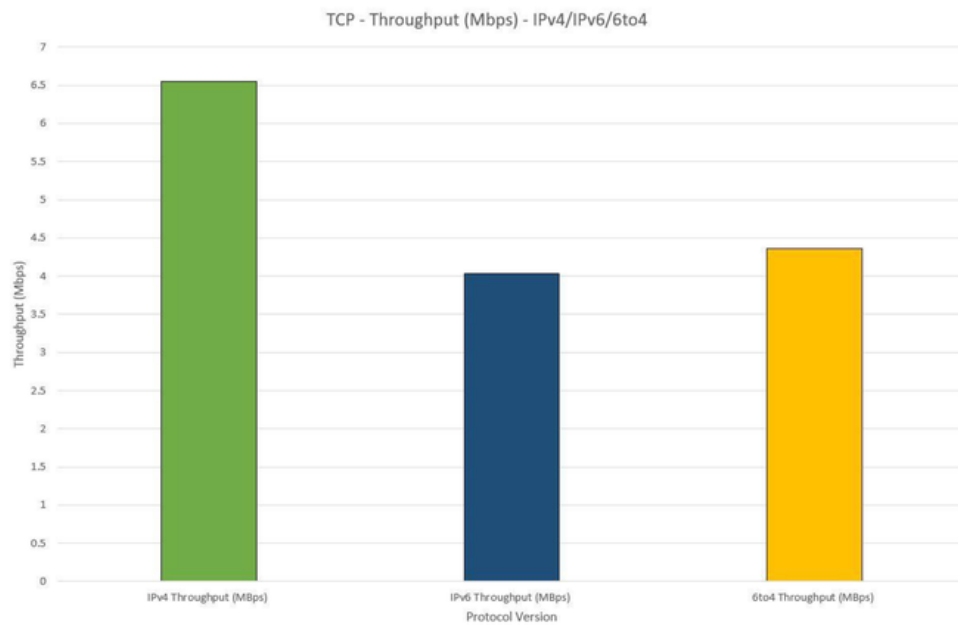


Figure 19 TCP Throughput

When comparing these results with those found in previous research, it appears to provide a mixed outcome. In the experiment by Fatah *et al* [36] IPv4 performed 1.97% better in a dual stack environment as well as a 99.08% superiority for IPv4 in a dual stack remote network scenario. The paper written by Soorty and Sarkar [46] showed a victory for IPv4 in both server OS situations with IPv6 performing 5.2% slower, they claim this to be due to higher transmission overheads. In the final comparison originating from Law *et al* paper they observed a higher throughput in IPv6 with 107.75 KB/s over 77.88KB/s with IPv4 in a dual stack environment. This is a surprising find, as it is the only statistics so far that go in IPv6's favour, and by a 38% margin. Later in the paper they also compare tunnel networks and found that compared to tunnel broker statistics, Native IPv6 performed better which is not the case in this experiment although it may prove that 6to4 tunnels are more reliable than tunnel broker services.

In conclusion, it appears that dual-stack cases provide a more suitable and stable situation, which achieve closer, if not better results for IPv6 over IPv4. Although this only gives an indication on the transmission period, and as results show in the graph above, IPv4 still outperforms native IPv6, which needs further investigation.

5.3 Peer-to-Peer

In this next section, an analysis of results relating to peer-to-peer traffic (host-to-host) will be discussed. Firstly, a collection of file sharing traffic at a high traffic rate with a detailed view of traffic received and traffic dropped over the three transport mechanisms. Following this is a look at how each protocol reacts to video conferencing with a combination of both low and high-resolution traffic.

5.3.1 File Transfer

In this section, we look at high rate file transfer in a peer-to-peer scenario. This test was run over a 30-minute period with both average traffic dropped and average traffic received being analysed. The first set of data shown in figure 20 displays the results of traffic dropped for IPv4 (Green), IPv6 (Blue) and 6to4 (Yellow).

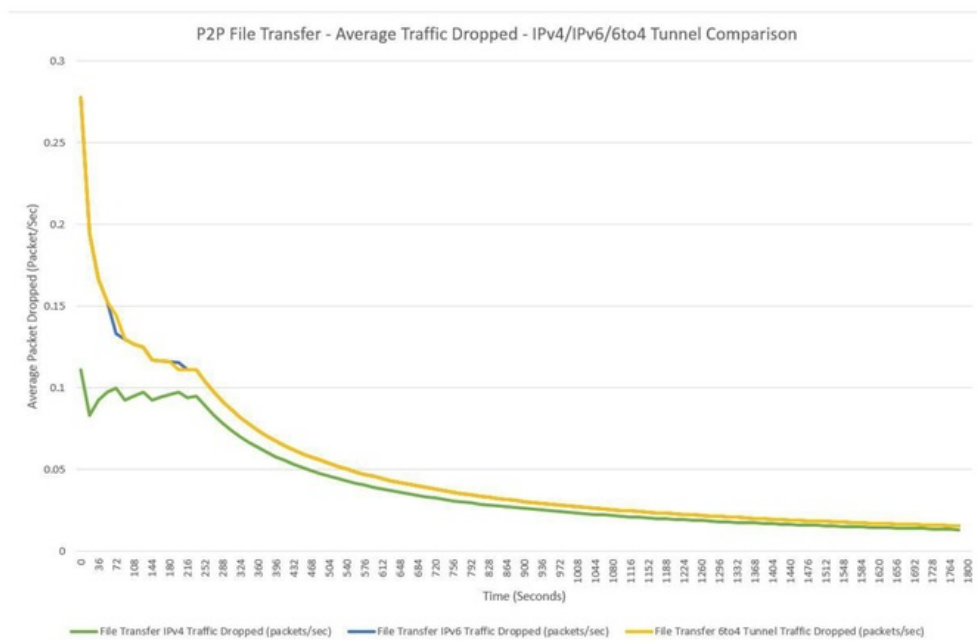


Figure 20 P2P File Transfer Traffic Dropped

On first inspection, it is apparent that all three methods face some degree of packet dropped. One thing to note is the statistics show packet dropped at the destination not packet loss en-route. Packets may be dropped for various reasons including duplicates, congestion, discards due to ACLs etc. The data in the graph shows average packets dropped in packets per second. IPv6 and 6to4 share the highest at 0.27 packets per second (PPS) at the start of the experiment. IPv4 also shows its highest drop rate at initial stage of test. This is similar to as discussed in previous scenarios where the routers and devices are sending higher amounts of traffic at initial configuration.

There are two main points that can be noted from this data set. The first being the initial differences between IPv4 and IPv6/6to4. With IPv4, the primary data is a little over 0.1 PPS, which then drops to 0.083 after 30 seconds. IPv6/6to4 on the other hand begins at 0.27 and then drops 51% to around 0.13 after 2 minutes of testing. The rate of decrease slows until the 4-minute marker and then all three methods begin to drop at an equal rate until the end of the scenario.

The second point is the difference in stabilisation between the methods, IPv4 flutters up and down for 4 minutes before beginning to stabilise where as IPv6/6to4 decreases shortly after start up. This shows that IPv6 QoS mechanisms may help the packets reach the destination more effectively preventing less packets from being dropped in a short period (as shown by the average decrease).

The next graph (figure 21) shows the average traffic received at the destination. The data set colours are the same as previous. Upon first inspection, it is obvious an initial large spike dominates all three protocols. IPv6 achieves the highest with just over 50 PPS received before falling. This initial spike may explain the higher rate of packets dropped as shown in figure 20. Because more packets are being received, the interface may be congested and therefore drops more packets than the other methods. One notable aspect is after the first spike all three methods decrease at an equal rate. Strangely, IPv6 then obtains another short spike of packets received. This is unusual as there is no similar spike in the packet dropped graph that would explain the reasoning for these extra packets received. Later in the results, 6to4 tunnel also receives a spike, which also is not apparent in the dropped chart.

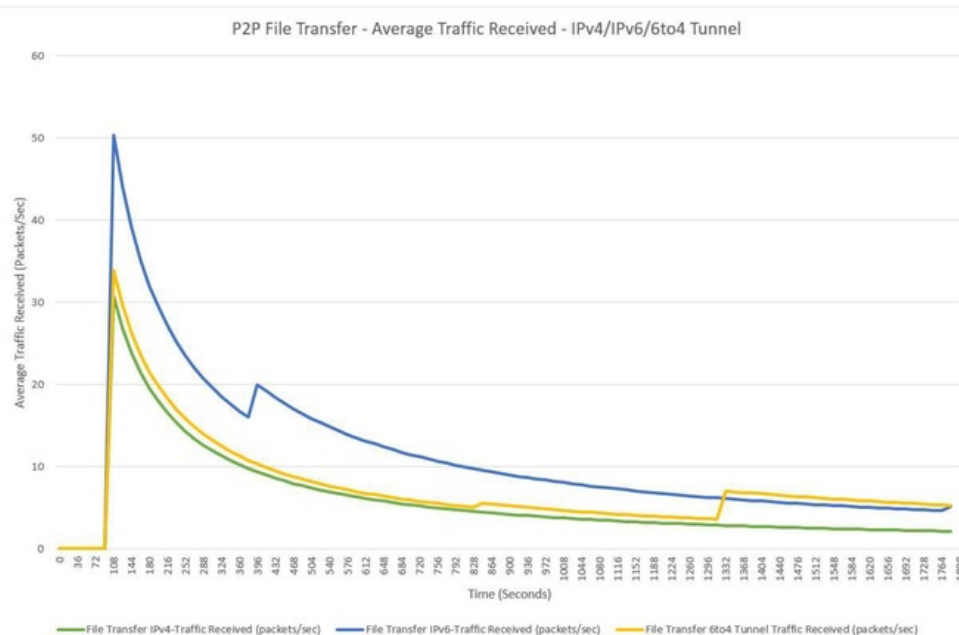


Figure 21 P2P File Transfer Traffic Received

For reference, figure 22 displays the traffic sent from the source. As it shows, the traffic sent is the same as that received at the destination. This eliminates the theory of a potential increase of packets sent/received due to packets being lost or dropped en-route.

The previous research papers did not analyse packet dropped therefore a direct comparison cannot be formed, although in Fatah *et al* experiment they did note an extremely large difference in packet loss within a dual-stack environment with IPv6, 655% compared to IPv4. They also noticed a fluctuation for packets lost throughout the experiment which is similar (although not as extreme) in the IPv4 packet dropped statistics achieved in this scenario.

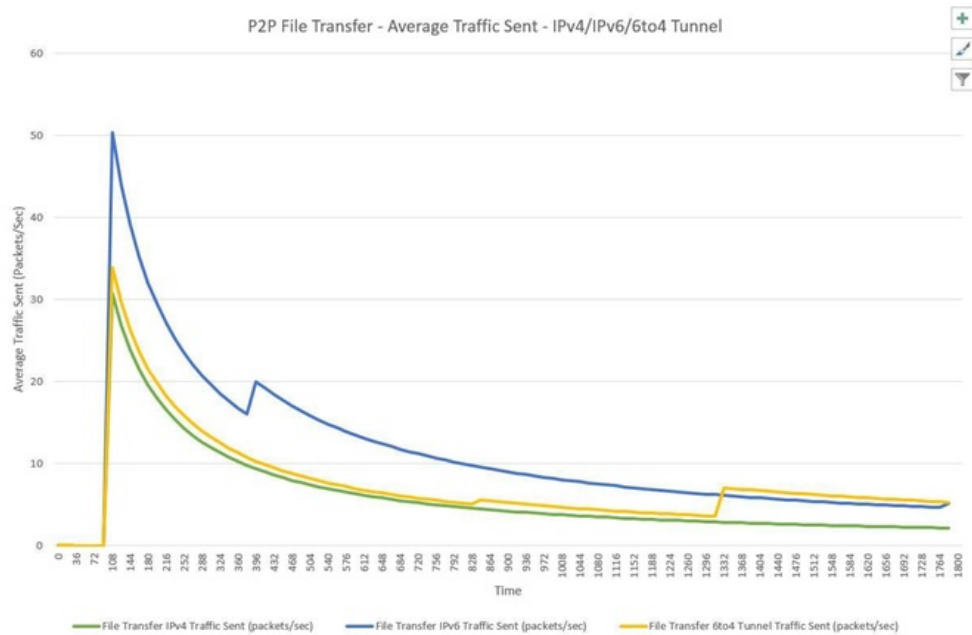


Figure 22 P2P File Transfer Average Traffic Sent

5.3.2 Video Conferencing

The next scenario was set up to determine the effect different traffic loads (Low/High resolution video conferencing) has on each transport mechanism. The results (shown in figure 23) were not as expected when referring to the same statistics obtained in previous research and will be explained below.

The data in this graph is as follows; two sets of measurements separated into low-resolution results (dotted lines) and high-resolution (solid lines). The line colours are the same as previous scenarios.

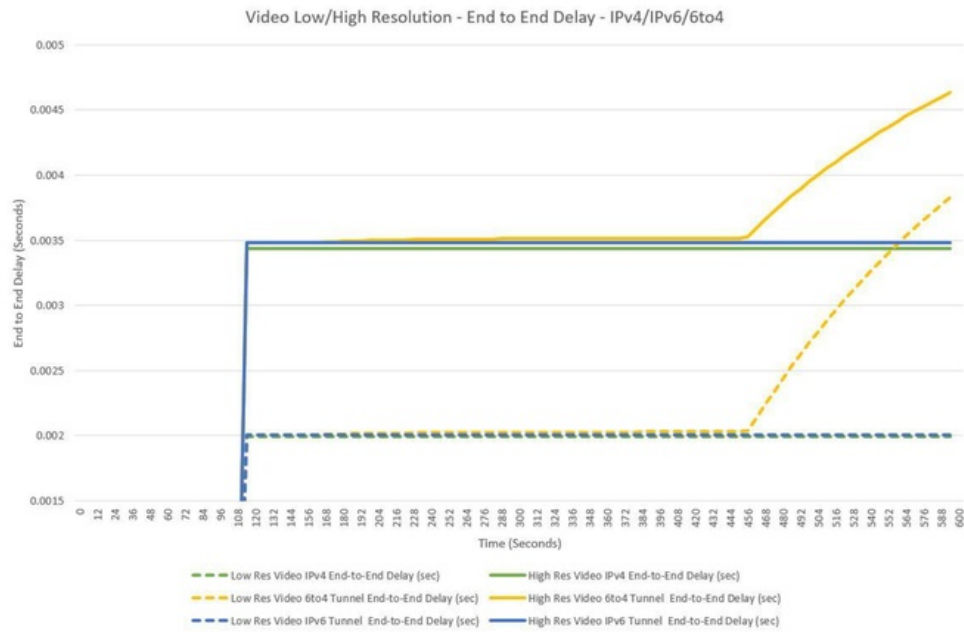


Figure 23 Low/High Res Video E2E Delay

From looking at the data above a surprising find has shown in the experiment. Unexpectedly all three protocols achieve very similar results for each independent test apart from a spike for 6to4 in the second half of the experiment.

The main difference that is visible is the extra delay high resolution has on each protocol version. Comparing low resolution to high resolution there is an average increase in End-to-End delay by 75%, which is apparent in IPv4, IPv6 and 6to4.

When comparing these results with those from other research, the data differs completely. In both Soorty [46] and Aazam *et al* [45] papers Delay was higher for IPv6. With video traffic and due to QoS mechanisms built into IPv6, the assumption was that IPv6 would perform better than IPv4. This is evidently not the case. The spike with 6to4 also goes unexplained, it may be caused by sudden network congestion but this would also have shown in both IPv4 and IPv6.

5.4 Client/Server FTP

The final scenario looks at how each protocol is effected within a client/server network. Fatah *et al* [36] also compared a dual-stack network against a client/server network so it will be interesting to see whether the results show a resemblance. This scenario analyses the average end-to-end delay in a client/server network with the traffic type set to both high load and low load FTP. File Transfer Protocol is a service used to transfer files and resources between devices over a network. Two main commands utilised with FTP are 'get' and 'put', these either send or retrieve files from the remote server and therefore can create a lot of traffic on the network depending on the file sizes. The less delay, the faster the transfer.

The data shown in figure 24 consists of similar to that found in the previous scenario, for each protocol, low load results appear as a dotted line and high load is represented with a solid line. This experiment took place over a 10-minute period.

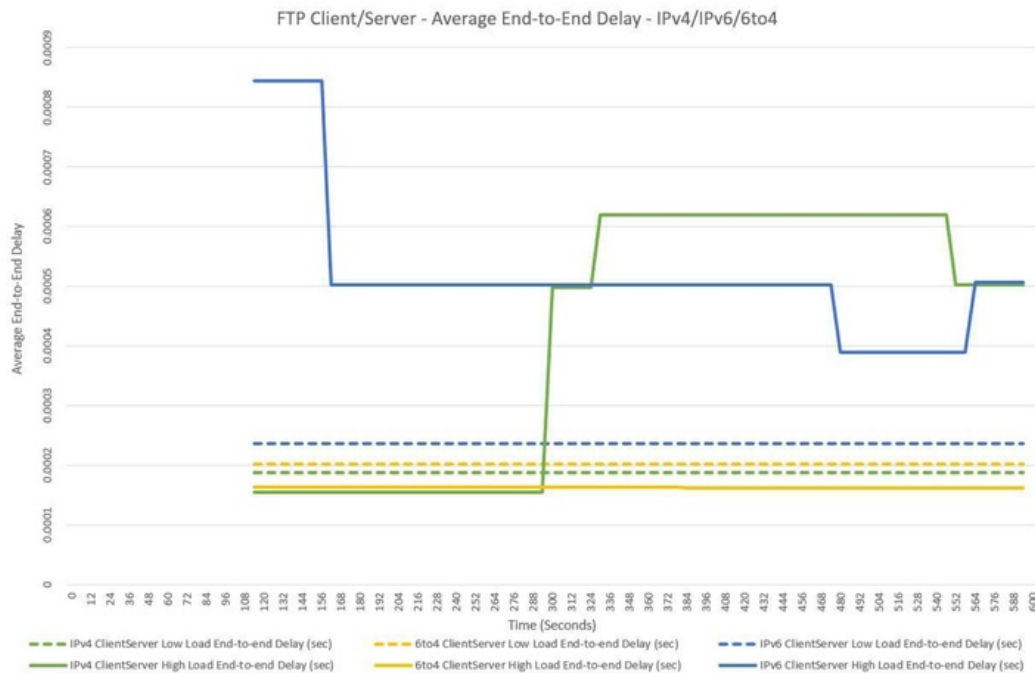


Figure 24 FTP Client/Server End-to-End Delay

The above figure shows the results collected. Firstly looking at low load results it appears that the end-to-end delay is consistent throughout the test. IPv4 has the lowest delay rate,

closely followed by 6to4 and IPv6 again coming in third place. The major differences appear in the high load traffic test where the network is under extra strain. Surprisingly 6to4 performs the best in this scenario even better than lower load. This needs extra investigation to determine why the tunnel handles the high load traffic better.

Another obvious sign from the graph is the fluctuations experienced by IPv4 and IPv6 with both versions then straightening out with similar results at the end of the 10-minute testing period. IPv6 begins with a high delay rate and then continues to fall, IPv4 operates in the opposite fashion whereas it starts low and then climbs. This shows that under extra strain IPv4 begins to decrease in performance unlike IPv6 where the performance increases and then holds a steady pattern. Unlike the low load results (and many other scenarios), IPv6 outperforms IPv4 in this example. It is under the assumption that this is the case due to IPv6's simplified header and flow control allowing the routers in the network to forward the packets faster due to the lower overheads.

Whilst reviewing Febby Nur Fatah *et al* [36] a similar scenario was found in the form of a jitter comparison between IPv4 and IPv6 in a client/server network. Their graph is displayed in figure 25 as a screenshot directly from the paper. "Jitter is the delay variation caused by long queues..." [36] although this is not the exact replicate of the End-to-End delay statistic analysed in this project, jitter is still a form of delay.

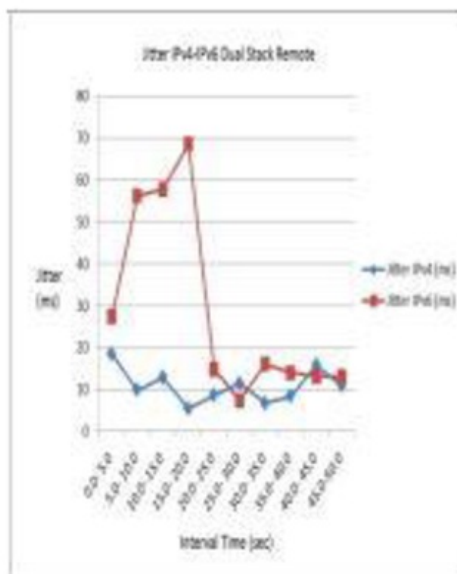


Figure 25 IPv4-IPv6 Dual-Stack Remote Server Jitter [36]

Although the graphic is of low quality, there is a huge resemblance of results compared to those achieved in the high load examples shown in figure 24. In figure 25 the results displayed are, IPv4 – Blue and IPv6 – Red. The trends appear to be alike, IPv4 starts with a low delay with IPv6 a high. IPv6 then falls sharply and has a point where it surpasses IPv4. Towards the end of the experiment both IPv4 and IPv6, seem to level off. This similarity justifies the use of IPv6 in a client/server network as it shows that in multiple occasions IPv6 can handle extra pressure although more research is required in explaining why these results are not apparent across multiple scenarios.

5.5 Conclusion

In conclusion to the results gathered in each scenario and in reflection to the original aim of this project, it appears overall, there are performance issues with IPv6 and transition mechanisms. It appears that in previous research papers and the results analysed in this project that IPv4 performance is superior on the majority of occasions. Upon analysis, it appears that dual-stack may perform better than tunnelling but further research on this is required in order to come to a conclusion.

These results will not prevent the transition as mentioned in chapter 2, the current network infrastructures cannot continue with IPv4, but it is likely to be the cause that so many networks are restraining from changing until the very last chance. In an article written on the BBC news website [58] titled, 'BT sets date for IPv6 network conversion', they state that BT plan to convert their entire network to IPv6 by the end of 2016. "BT currently has about 7.8 million customers", this means both businesses and home users will be automatically converted to IPv6, which is a big step in the advancements of the transition and will most likely result in many other networks following.

Unfortunately, because 4to6 was not available it is not possible to establish how the networks that continue to operate on IPv4 will perform at the end of the transition period, this has been noted in the recommendations section later in this report.

Chapter 6 - Conclusion and Recommendations

In this section of the report will be a conclusion on the results and analysis obtained during the experiment and a discussion on the main findings. Following this is the recommendations for future project work.

6.1 Conclusion

At the beginning of this research project, a detailed review of literature provided an understanding of the relevant topics considered within this subject. This information aided the design and implementation stages upon following the research methodology set out in chapter 3. Although the subject related to this project is vast, the main aim set out would provide crucial information that aimed at explaining any performance issues relating to IPv6 and the transition mechanisms in place. After implementation using Riverbed Modeler, various results could be analysed. In order to gain a detailed view at any effects, various traffic types were simulated to provide a more realistic view at how each protocol version operates. It is in this analysis that a comparison with previous research found that there are performance issues related to native IPv6 and 6to4 tunnel methods, which may indeed be a factor in the prolonged transition period.

These results presented some key findings, these are:

- IPv4 generally performed better than both IPv6 and 6to4 Tunnels.
- Dual Stack results from previous research shows better performance than tunnels/native.
- Traffic size had mixed results on performance of IPv6, sometimes it performed better other times it had no effect.

In reflection to this project and the results achieved, it is possible that the poor performance has an impact on the transition. As mentioned previously, the transition will go ahead regardless but these results may determine why it has taken close to 20 years since protocol design and still only 8.72% of web traffic is utilising IPv6 [3] which equates to less than 1% increase during the course of this project.

6.2 Recommendations for Future Work

During the process of this project, ideas for potential further study became apparent. These may be in connection with the main aim for this research project or a similar project with alternative aims and objectives.

The first recommendation is the experimentation of 4to6 tunnels. Originally, this project was due to analyse both native protocols along with 6to4 and 4to6 tunnels. Unfortunately, due to software restraints, 4to6 tunnel data was not accessible to capture. A further research comparing 6to4 and 4to6 to analyse any potential trends that will aid this project outcome.

A similar recommendation is the comparison of the three main transition techniques; Dual-Stack, Tunnel and Translation. During the literature stage of this project it became apparent that some research has been conducted on dual-stack networks and the results from this aided explanations during the evaluation. An in-depth comparison on all three methods would aid the transition to become as smooth as possible.

A third recommendation is to conduct a similarly structure project but with the incorporation of wireless and mobile devices. As mentioned in section 2.7, Bi *et al* [41] describes the lack of wireless testing on IPv6.

A final recommendation would be to analyse similar scenarios that have been tested in this project but within a real-world network, utilising the internet if possible. Using test beds is useful for analysing statistics and data but with the use of a real-world scenario, it will allow the testers to see effects on the users rather than theoretical data.

Chapter 7 – Reflection

In this final section of the report will be a reflection on the project and the work carried out. A reflection is required to provide an overview on whether the main aim was achieved and how the objectives were/were not met. This section also provides a summary of the issues faced during the course of the project and how they were overcome followed by a personal reflection.

7.1 Main Aim

The main aim set at the start of this project was:

Is there a direct connection between performance results in native and tunnel IPv4/v6 networks and the poor migration statistics?

7.1.1 Was the aim achieved?

Although a conclusive result cannot be achieved solely from the evidence gained, it is certainly possible that the poor performance monitored in chapter 5 of this report has some effect on the migration between protocols. The consensus of results was that the poor performance was mainly caused by the larger overheads within IPv6 such as larger header size. This resulted in network delay, slower RTT and a higher rate of packets dropped all of which are crucial elements in successful network operation.

7.2 Objectives

As discussed in the introduction of this report, in order to meet the aim detailed above certain objectives were required. The objectives were as follows:

1. Research studies relating to IPv4/v6 (with more detailed research focusing on performance) to gain an understanding on the current research position.
2. Analyse and discuss previous research statistics to show (if any) the current patterns in data.
3. Using a network simulation package analyse the Throughput, Round Trip Time(RTT), Delay and Packet Loss in the following network setups:

- I. Native IPv4 and IPv6
 - II. IPv6 to v4 tunnel (6to4)
 - III. IPv4 to v6 tunnel (4to6)
4. Using the results from the experiment; evaluate the statistics and discuss the findings compared to previous research.

7.2.1 How each objective was met

A strategic approach was taken in order to assure each object had been achieved to its full potential given the allocated time constraints. In this section a discussion on how and where each objective outcome can be found within this report.

Objective 1 was achieved in the early stages of this project and consists of the entirety of chapter 2 within this report. The information from within this chapter, along with knowledge gained from other modules during study aided with the network design and implementation found in chapter 4.

Objective 2 contained a more in-depth research gained during the first objective. This task carried out throughout the entire project with regular searches for new information and papers that provided research data relating to those necessary for this project. The information gathered has been discussed in detail within chapter 2.7 and then referenced in comparison with primary data in chapter 5.

Objective 3 was completed during the design and implementation stages of this project found in chapter 4. After researching individual software packages suitable for this task, a decision was made upon the adequate package and then implementation began in order to achieve results. During this stage, it became apparent that objective 3.iii would not be accomplishable in this project, which is discussed further in the next section.

Objective 4 was completed during the evaluation section within chapter 5. The main purpose of this chapter was to analyse the results achieved during the implementation and then compare these results to spot any trends with previous research data. The overall theory was then concluded at the end of chapter 5 and throughout chapter 6.

7.3 Issues faced during the project

Whilst undertaking a project of this size it is inevitable that a few issues will arise throughout various stages. In this final section of this report, is a brief description of some of the issues faced during this project.

The first major issue faced was the discovery of the lack of 4to6 tunnels in the chosen software. This resulted in not being able to fulfil part of an objective and has had a small impact on overall result analysis. This issue resulted in a small delay to the schedule set out at the start of the project. In order to prevent such issue from reoccurring, further research on software packages is required prior to objective design.

Another issue relating to software was the restriction in network size in the free version of Riverbed Modeler. This resulted in a change of network design at a late stage of the project.

The final issue faced was time management. Although a careful schedule was designed, certain aspects of this project did overrun which had a knock-on effect to the rest of the timeline. In order to prevent this in the future, allocating time for part-time work, social and other module work needs to be considered, as does any time lost due to issues.

7.4 Personal Reflection

Due to this being my first major project, I faced challenges and learned new skills in order to complete it on time and to a satisfying standard. During the course of this project, I also learned how I would have liked to have done things differently without the time constraints faced. All of this is discussed in this final section of the report.

Completing this project has taught me how to manage a project including everything from research gathering, analysis, time management and writing large reports collating all information from the findings. These skills will be vital on moving into employment after completing this degree.

If I were to complete this project again I would do several things differently. Time management is a crucial aspect and allocating time in the schedule for part-time work and other modules would have resulted in a smoother flowing timeline. Although the majority of tasks were completed to plan, extra time would have been available for 'fine-tuning' the report and data analysis.

References

- [1] S. E. Deering, "Internet Protocol, Version 6 (IPv6) Specification," December 1998. [Online]. Available: <https://tools.ietf.org/html/rfc2460#page-4>. [Accessed 5 May 2015].
- [2] R. E. Gilligan and E. Nordmark, "Transition Mechanisms for IPv6 Hosts and Routers," April 1996. [Online]. Available: <https://tools.ietf.org/html/rfc1933>. [Accessed 22 October 2015].
- [3] Google, "IPv6 Statistics April 2016," 7 April 2016. [Online]. Available: <https://www.google.com/intl/en/ipv6/statistics.html>. [Accessed 7 April 2016].
- [4] Information Sciences Institute, "Internet Protocol," September 1981. [Online]. Available: <https://tools.ietf.org/html/rfc791>. [Accessed 21 October 2015].
- [5] S. Hagen, "Why IPv6?," in *IPv6 Essentials*, Sebastopol, CA, O'Reilly, 2014, p. 1.
- [6] A. Shiranzaei and R. Z. Khan, "Internet Protocol Versions - A Review," in *2015 2nd International Conference on Computing for Sustainable Global Development (INDIACom)*, New Delhi, 2015.
- [7] A. N. A. Ali, "Comparison study between IPV4 & IPV6," *IJCSI International Journal of Computer Science Issues*, vol. 9, no. 3, pp. 314-317, 2012.
- [8] J. Hunumanthappa and D. Manjaiah, "A Study on Comparison and Contrast between IPv6 and IPv4 Feature Sets," in *International Conference on Computer Networks and Security (ICCNS-2008)*, 2008.
- [9] Google.com, "Google IPv6 Statistics," Google, 22 October 2015. [Online]. Available : <http://www.google.com/intl/en/ipv6/statistics.html>. [Accessed 22 October 2015].
- [10] B. A. Forouzan, "A Brief History," in *TCP/IP Protocol Suite*, Fourth ed., New York, McGraw-Hill, 2010, pp. 3-6.
- [11] V. Cerf and R. Kahn, "A Protocol for Packet Network Intercommunication," *IEEE Transactions on Communications*, vol. 22, no. 5, pp. 637-648, 1974.
- [12] W. Stallings, "The TCP/IP Protocol Architecture," in *Data and Computer Communications*, New Jersey, Pearson Education, Inc, 2007, pp. 34-40.
- [13] J. Casad, "The Development of TCP/IP," in *Sams Teach Yourself TCP/IP in 24 Hours* , Indianapolis , Pearson Education Inc, 2012, pp. 10-11.
- [14] G. C. Kessler, "An Overview of TCP/IP Protocols and the Internet," 13 November 2014. [Online]. Available: <http://www.garykessler.net/library/tcpip.html#info>. [Accessed 30 October 2015].
- [15] T. Lammle, "Introduction to TCP/IP," in *CCNA: Cisco Certified Network Associate Study Guide*, Indianapolis, Wiley Publishing Inc., 2011, p. 118.
- [16] Y. Li, D. Li , W. Cui and R. Zhang, "Research based on OSI model," in *2011 IEEE 3rd International*

- Conference on Communication Software and Networks (ICCSN)*, Xi'an, 2011.
- [17] T. Rooney, "1.1.1 The IP Header," in *IP Address Management Principles and Practice*, New Jersey, John Wiley & Sons, 2011, pp. 5-7.
 - [18] R. Housley, J. Curran, G. Huston and D. Conrad, "The Internet Numbers Registry System," August 2013. [Online]. Available: <https://tools.ietf.org/html/rfc7020>. [Accessed 13 November 2015].
 - [19] BBC News, "Old net addresses run out in US and Canada," 25 September 2015. [Online]. Available: <http://www.bbc.co.uk/news/technology-34356770>. [Accessed 17 December 2015].
 - [20] Y. Rekhter and T. Li, "An Architecture for IP Address Allocation with CIDR," September 1993. [Online]. Available: <https://tools.ietf.org/html/rfc1518>. [Accessed 19 November 2015].
 - [21] Cisco, "6Lab Cisco," Cisco, December 2015. [Online]. Available: <http://6lab.cisco.com/stats/>. [Accessed 11 December 2015].
 - [22] J. Govil, J. Govil, N. Kaur and H. Kaur, "An Examination of IPv4 and IPv6 Networks: Constraints and Various Transition Mechanisms," in *IEEE South East Con 2008*, Huntsville, AL, 2008.
 - [23] S. Kent and R. Atkinson, "RFC 2402," November 1998. [Online]. Available: <https://tools.ietf.org/html/rfc2402>. [Accessed 11 December 2015].
 - [24] S. Kent and R. Atkinson, "RFC 2406," November 1998. [Online]. Available: <https://tools.ietf.org/html/rfc2406>. [Accessed 11 December 2015].
 - [25] F. Gont, "Filtering IPv6 extension headers is sometimes necessary," November 2015. [Online]. Available: [http://searchnetworking.techtarget.com/tip/Filtering-IPv6-extension-headers-is-sometimes-necessary?utm_medium=EM&asrc=EM_ERU_49619333&utm_campaign=20151109_ERU%20Transmission%20for%2011/09/2015%20\(UserUniverse:%201842501\)_myka-reports@techtarget.com&utm_s](http://searchnetworking.techtarget.com/tip/Filtering-IPv6-extension-headers-is-sometimes-necessary?utm_medium=EM&asrc=EM_ERU_49619333&utm_campaign=20151109_ERU%20Transmission%20for%2011/09/2015%20(UserUniverse:%201842501)_myka-reports@techtarget.com&utm_s). [Accessed 17 December 2015].
 - [26] S. Kent and R. Atkinson, "Security Architecture for the Internet Protocol," November 1998. [Online]. Available: <https://tools.ietf.org/html/rfc2401#section-3.1>. [Accessed 26 November 2015].
 - [27] D. Yang, X. Song and Q. Guo, "Security on IPv6," in *Advanced Computer Control (ICACC), 2010 2nd International Conference on*, Shenyang, 2010.
 - [28] S. Hagen, "Security with IPv6," in *IPv6 Essentials*, Sebastopol, O'Reilly Media, 2014, pp. 187-217.
 - [29] E. Jankiewicz, J. Loughney and T. Narten, "IPv6 Node Requirements," December 2011. [Online]. Available: <https://tools.ietf.org/html/rfc6434>. [Accessed 26 November 2015].
 - [30] O. J. S. Parra, A. P. Rios and G. L. Rubio, "IPv6 and IPv4 QoS Mechanisms," in *Information Integration and Web-based Applications and Services*, New York, 2011.
 - [31] S. Kawamura and M. Kawashima, "A Recommendation for IPv6 Address Text

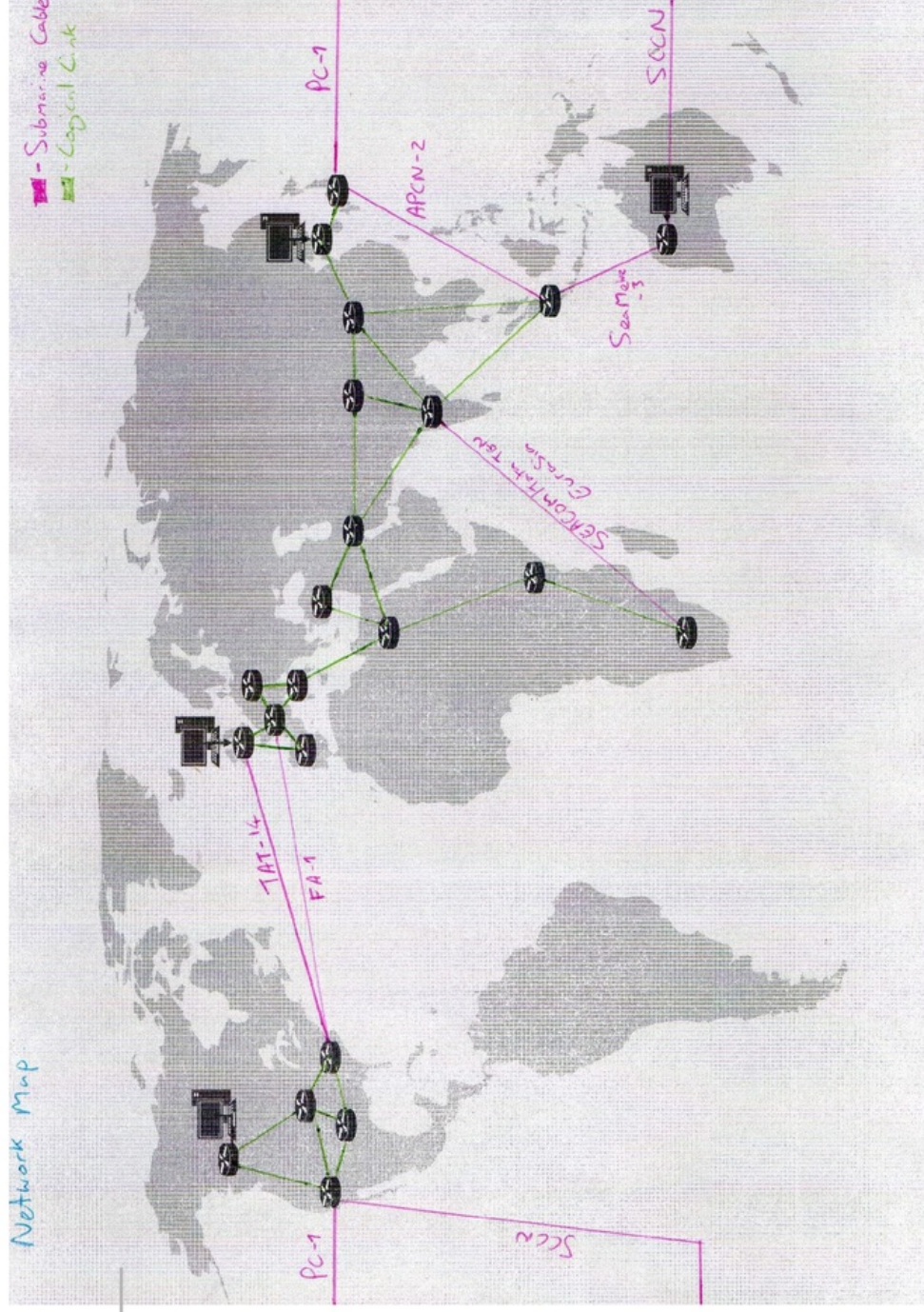
Representation,”

August 2010. [Online]. Available: <https://tools.ietf.org/html/rfc5952>. [Accessed 13 November 2015].

- [32] R. Hinden and S. Deering, “IP Version 6 Addressing Architecture,” February 2006. [Online]. Available: <https://tools.ietf.org/html/rfc4291>. [Accessed 17 December 2015].
- [33] S. Kalwar, N. Bohra and A. Memon, “A survey of transition mechanisms from IPv4 to IPv6-Simulated test bed and analysis,” in *2015 Third International Conference on Digital Information, Networking, and Wireless Communications (DINWC)*, Moscow, 2015.
- [34] C. Huitema, “Teredo: Tunneling IPv6 over UDP through Network Address Translations (NATs),” February 2008. [Online]. Available: <https://www.ietf.org/rfc/rfc4380.txt>. [Accessed 3 January 2016].
- [35] F. Templin, T. Gleeson and D. Thaler, “Intra-Site Automatic Tunnel Addressing Protocol (ISATAP),” March 2008. [Online]. Available: <https://tools.ietf.org/html/rfc5214>. [Accessed 3 January 2016].
- [36] F. N. Fatah, M. A. Marwan, A. Suhendra and H. Firdaus, “Performance Measurements Analysis of Dual Stack IPv4-IPv6,” in *Advances in Information Technology (AIT 2013)*, 2013.
- [37] R. Graziani, “Dual-Stack and Tunneling,” in *IPv6 Fundamentals*, Indianapolis, Cisco Press, 2014, pp. 344-374.
- [38] S. Hagen, in *IPv6 Essentials*, Sebastopol, O'Reilly & Associates Inc, 2002.
- [39] J. Postel, “Internet Protocol,” September 1981. [Online]. Available: <https://tools.ietf.org/html/rfc791#page-11>. [Accessed 5 May 2015].
- [40] J. Rajahalme, S. Amante, S. Jiang and B. Carpenter, “IPv6 Flow Label Specification,” November 2011. [Online]. Available: <https://tools.ietf.org/html/rfc6437#page-3>. [Accessed 5 May 2015].
- [41] J. Bi, J. Wu and X. Leng, “IPv4/IPv6 Transition Technologies and Univer6 Architecture,” *International Journal of Computer Science and Network Security*, vol. 7, no. 1, pp. 232-243, 2007.
- [42] X. Cheng, J. Bi and X. Li, “Univer6 - A Universal Network Architecture for IPv6,” in *10th International Conference on Advanced Communication Technology, 2008. ICACT 2008*, 2008.
- [43] J. Arkko, C. Perkins and D. Johnson, “Mobility Support in IPv6,” July 2011. [Online]. Available: <https://tools.ietf.org/html/rfc6275#page-7>. [Accessed 5 May 2015].
- [44] Y.-N. Law, M.-C. Lai, W. L. Tan and W. C. Lau, “Empirical Performance of IPv6 vs. IPv4 under a Dual-Stack Environment,” in *IEEE International Conference on Communications, 2008. ICC '08*, 2008.
- [45] M. Aazam, A. M. Syed, S. Atif, H. Shah, I. Khan and M. Alam, “Evaluation of 6to4 and ISATAP on a test LAN,” in *Computers & Informatics (ISCI), 2011 IEEE Symposium*, Kuala Lumpur, 2011.
- [46] B. Soorty and N. Sarkar, “Quantifying the Performance Degradation of IPv6 for TCP in Windows

and Linux Networking', in *Telecommunication Networks and Applications Conference (ATNAC), 2013 Australasian*, Christchurch, 2013.

- [47] S. Kolahi and B. Soorty, "Evaluation of Gigabit Ethernet Local Area Networks in Windows Vista-Server 2008 Environment," in *Advanced Information Networking and Applications (WAINA), 2011 IEEE Workshops of International Conference*, Biopolis, 2011.
- [48] M. Saunders, P. Lewis and A. Thornhill, "Writing you research proposal," in *Research Methods For Business Students*, Essex, Pearson, 2012, p. 54.
- [49] W. D. Richards, "The Zen of Empirical Research," 2002.
- [50] D. N. Whitehead, "Research Methodologies," 2013.
- [51] D. S. A. Burney, "Inductive & Deductive Research Approach," 6 March 2008. [Online]. Available: <http://www.drburney.net/INDUCTIVE%20&%20DEDUCTIVE%20RESEARCH%20APPROACH%2006032008.pdf>. [Accessed 3 May 2015].
- [52] Riverbed Technology Inc, "Riverbed Modeler," Riverbed Technology Inc, 2016. [Online]. Available: <https://support.riverbed.com/content/support/software/steelcentral-npm/modeler-index.html>. [Accessed 25 March 2016].
- [53] I. S. Hammoodi, B. G. Stewart, A. Kocian and S. G. McMeekin, "A Comprehensive Performance Study of OPNET Modeler For ZigBee Wireless Sensor Networks," in *2009 Third International Conference on Next Generation Mobile Applications, Services and Technologies*, Cardiff, 2009.
- [54] Riverbed Support, "What are the restrictions for Riverbed Modeler Academic Edition 17.5?," 06 Novemeber 2015. [Online]. Available: <https://supportkb.riverbed.com/support/index?page=content&id=S24443>. [Accessed 25 March 2016].
- [55] A. S. Sethi and V. Y. Hnatyshin, *The Practical OPNET User Guide for Computer Network Simulation*, Boca Raton, FL: CRC Press, 2013.
- [56] Scalable Network Technologies, "QualNet," [Online]. Available: <http://web.scalable-networks.com/content/qualnet>. [Accessed 25 March 2016].
- [57] N. C. Arjuman and S. Manickam, "A Review on ICMPv6 Vulnerabilities and its Mitigation Techniques: Classification and Art," in *IEEE 2015 International Conference on Computer, Communication, and Control Technology (I4CT 2015)*, Malaysia, 2015.
- [58] BBC Technology, "BT sets date for IPv6 network conversion," 24 September 2015. [Online]. Available: <http://www.bbc.co.uk/news/technology-34346803>. [Accessed 6 April 2016].
- [59] J. Bi, "Jun Bi," [Online]. Available: <http://netarchlab.tsinghua.edu.cn/~junbi/>. [Accessed 5 May 2015].



Appendix

Device Name	Interface (Router Links)	IP Address
UK Client	NIC 1	192.168.1.10 /24
Canada Client	NIC 1	192.168.2.10 /24
Australia Client	NIC 1	192.168.3.10 /24
North Korea Client	NIC 1	192.168.4.10 /24
UK Router	Private (Internal) Int1 (UK-France) Int2 (UK-Spain) Int3 (UK-New Jersey) <i>[TAT-14]</i>	192.168.1.1 /24 200.1.1.1/30 200.2.2.1/30 10.5.5.1/30
France Router	Int1 (UK-France) Int2 (France-Spain) Int3 (France-Germany) Int4 (France-Italy) Int5 (France-New Jersey) <i>[FA-1]</i>	200.1.1.2/30 200.3.3.1/30 200.4.4.1/30 200.5.5.1/30 10.6.6.1/30
Spain Router	Int1 (UK-Spain) Int2 (France-Spain)	200.2.2.2/30 200.3.3.2/30
Germany Router	Int1 (France-Germany) Int2 (Germany-Italy)	200.4.4.2/30 200.6.6.1/30
Italy Router	Int1 (France-Italy) Int2 (Germany-Italy) Int3 (Italy-Egypt)	200.5.5.2/30 200.6.6.2/30 200.7.7.1/30
Egypt Router	Int1 (Italy-Egypt) Int2 (Egypt-Kenya) Int3 (Egypt-Turkey) Int4 (Egypt-Iran)	200.7.7.2/30 200.8.8.1/30 200.9.9.1/30 200.10.10.1/30
Kenya Router	Int1 (Egypt-Kenya) Int2 (Kenya-South Africa)	200.8.8.2/30 200.11.11.1/30
Turkey Router	Int1 (Egypt-Turkey) Int2 (Turkey-Iran)	200.9.9.2/30 200.12.12.1/30
Iran Router	Int1 (Egypt-Iran) Int2 (Turkey-Iran) Int3 (Iran-India) Int4 (Iran-China1)	200.10.10.2/30 200.12.12.2/30 200.13.13.1/30 200.14.14.1/30
South Africa Router	Int1 (Kenya-South Africa) Int2 (South Africa-India) <i>[SEACOM/Tata TGN-Eurasia]</i>	200.11.11.2/30 10.7.7.1/30
India Router	Int1 (Iran-India) Int2 (India-China1) Int3 (India-China2) Int4 (India-Indonesia) Int5 (South Africa-India) <i>[SEACOM/Tata TGN-Eurasia]</i>	200.13.13.2/30 200.15.15.1/30 200.16.16.1/30 200.17.17.1/30 10.7.7.2/30
China1 Router	Int1 (Iran-China1) Int2 (India-China1)	200.14.14.2/30 200.15.15.2/30

	Int3 (China1-China2)	200.18.18.1/30
China2 Router	Int1 (India-China2)	200.16.16.2/30
	Int2 (China1-China2)	200.18.18.2/30
	Int3 (China2-Indonesia)	200.19.19.1/30
	Int4 (China2-North Korea)	200.20.20.1/30
Indonesia Router	Int1 (India-Indonesia)	200.17.17.2/30
	Int2 (China2-Indonesia)	200.19.19.2/30
	Int3 (Indonesia-Australia)	10.1.1.1/30
	[SeaMeWe-3]	
	Int4 (Indonesia-Japan) [APCN-2]	10.2.2.1/30
North Korea Router	Private (Internal)	192.168.4.1 /24
	Int1 (China2-North Korea)	200.20.20.2/30
	Int2 (North Korea-Japan)	20.21.21.1/30
Australia Router	Private (Internal)	192.168.3.1/24
	Int1 (Indonesia-Australia)	10.1.1.2/30
	[SeaMeWe-3]	
	Int2 (Australia-California)[SCCN]	10.4.4.1/30
Japan Router	Int1 (North Korea-Japan)	200.21.21.2/30
	Int2 (Indonesia-Japan) [APCN-2]	10.2.2.2/30
	Int3 (Japan-California) [PC-1]	10.3.3.1/30
California Router	Int1 (Japan-California) [PC-1]	10.3.3.2/30
	Int2 (Australia-California)[SCCN]	10.4.4.2/30
	Int3 (California-Texas)	200.22.22.1/30
	Int4 (California-Kansas)	200.23.23.1/30
	Int5 (California-Canada)	200.24.24.1/30
Texas Router	Int1 (California-Texas)	200.22.22.2/30
	Int2 (Texas-Kansas)	200.26.26.1/30
	Int3 (Texas-New Jersey)	200.27.27.1/30
Kansas Router	Int1 (California-Kansas)	200.23.23.2/30
	Int2 (Canada-Kansas)	200.25.25.2/30
	Int3 (Texas-Kansas)	200.26.26.2/30
	Int4 (Kansas-New Jersey)	200.28.28.1/30
New Jersey Router	Int1 (Texas-New Jersey)	200.27.27.2/30
	Int2 (Kansas-New Jersey)	200.28.28.2/30
	Int3 (UK-New Jersey) [TAT-14]	10.5.5.2/30
	Int4 (France-New Jersey) [FA-1]	10.6.6.2/30
Canada Router	Private (Internal)	192.168.2.1 /24
	Int1 (California-Canada)	200.24.24.2/30
	Int2 (Canada-Kansas)	200.25.25.1/30

IPv6

Device Name	Interface (Router Links)	IP Address
UK Client	NIC 1	FC00::1:10
Canada Client	NIC 1	FC00::2:10
Australia Client	NIC 1	FC00::3:10
North Korea Client	NIC 1	FC00::4:10
UK Router	Private (Internal) Int1 (UK-France) Int2 (UK-Spain) Int3 (UK-New Jersey) [TAT-14]	FC00::1:1 2001:db8:acad::1:1 FE80::1:1 2001:db8:acad::2:1 FE80::2:1 2001:db8:acad::100:1 FE80::100:1
France Router	Int1 (UK-France) Int2 (France-Spain) Int3 (France-Germany) Int4 (France-Italy) Int5 (France-New Jersey) [FA-1]	2001:db8:acad::1:2 FE80::1:2 2001:db8:acad::3:1 FE80::3:1 2001:db8:acad::4:1 FE80::4:1 2001:db8:acad::5:1 FE80::5:1 2001:db8:acad::101:1 FE80::101:1
Spain Router	Int1 (UK-Spain) Int2 (France-Spain)	2001:db8:acad::2:2 FE80::2:2 2001:db8:acad::3:2 FE80::3:2
Germany Router	Int1 (France-Germany) Int2 (Germany-Italy)	2001:db8:acad::4:2 FE80::4:2 2001:db8:acad::6:1 FE80::6:1
Italy Router	Int1 (France-Italy) Int2 (Germany-Italy) Int3 (Italy-Egypt)	2001:db8:acad::5:2 FE80::5:2 2001:db8:acad::6:2 FE80::6:2 2001:db8:acad::7:1 FE80::7:1
Egypt Router	Int1 (Italy-Egypt) Int2 (Egypt-Kenya) Int3 (Egypt-Turkey)	2001:db8:acad::7:2 FE80::7:2 2001:db8:acad::8:1 FE80::8:1 2001:db8:acad::9:1 FE80::9:1

	Int4 (Egypt-Iran)	2001:db8:acad::10:1 FE80::10:1
Kenya Router	Int1 (Egypt-Kenya)	2001:db8:acad::8:2 FE80::8:2
	Int2 (Kenya-South Africa)	2001:db8:acad::11:1 FE80::11:1
Turkey Router	Int1 (Egypt-Turkey)	2001:db8:acad::9:2 FE80::9:2
	Int2 (Turkey-Iran)	2001:db8:acad::12:1 FE80::12:1
Iran Router	Int1 (Egypt-Iran)	2001:db8:acad::10:2 FE80::10:2
	Int2 (Turkey-Iran)	2001:db8:acad::12:2 FE80::12:2
	Int3 (Iran-India)	2001:db8:acad::13:1 FE80::13:2
	Int4 (Iran-China1)	2001:db8:acad::14:1 FE80::14:1
South Africa Router	Int1 (Kenya-South Africa)	2001:db8:acad::11:2 FE80::11:2
	Int2 (South Africa-India)[SEACOM/Tata TGN-Eurasia]	2001:db8:acad::102:1 FE80::102:1
India Router	Int1 (Iran-India)	2001:db8:acad::13:2 FE80::13:2
	Int2 (India-China1)	2001:db8:acad::15:1 FE80::15:1
	Int3 (India-China2)	2001:db8:acad::16:1 FE80::16:1
	Int4 (India-Indonesia)	2001:db8:acad::17:1 FE80::17:1
	Int5 (South Africa-India)[SEACOM/Tata TGN-Eurasia]	2001:db8:acad::102:2 FE80::102:2
China1 Router	Int1 (Iran-China1)	2001:db8:acad::14:2 FE80::14:2
	Int2 (India-China1)	2001:db8:acad::15:2 FE80::15:2
	Int3 (China1-China2)	2001:db8:acad::18:1 FE80::18:1
China2 Router	Int1 (India-China2)	2001:db8:acad::16:2 FE80::16:2
	Int2 (China1-China2)	2001:db8:acad::18:2 FE80::18:2
	Int3 (China2-Indonesia)	2001:db8:acad::19:1 FE80::19:1
	Int4 (China2-North Korea)	2001:db8:acad::20:1 FE80::20:1
Indonesia Router	Int1 (India-Indonesia)	2001:db8:acad::17:2

	Int2 (China2-Indonesia) Int3 (Indonesia-Australia) <i>[SeaMeWe-3]</i> Int4 (Indonesia-Japan) <i>[APCN-2]</i>	FE80::17:2 2001:db8:acad::19:2 FE80::19:2 2001:db8:acad::103:1 FE80::103:1 2001:db8:acad::104:1 FE80::104:1
North Korea Router	Private (Internal) Int1 (China2-North Korea) Int2 (North Korea-Japan)	FC00::4:1 2001:db8:acad::20:2 FE80::20:2 2001:db8:acad::21:1 FE80::21:1
Australia Router	Private (Internal) Int1 (Indonesia-Australia) <i>[SeaMeWe-3]</i> Int2 (Australia-California) <i>[SCCN]</i>	FC00::3:1 2001:db8:acad::103:2 FE80::103:2 2001:db8:acad::105:1 FE80::105:1
Japan Router	Int1 (North Korea-Japan) Int2 (Indonesia-Japan) <i>[APCN-2]</i> Int3 (Japan-California) <i>[PC-1]</i>	2001:db8:acad::21:2 FE80::21:2 2001:db8:acad::104:2 FE80::104:2 2001:db8:acad::106:1 FE80::106:1
California Router	Int1 (Japan-California) <i>[PC-1]</i> Int2 (Australia-California) <i>[SCCN]</i> Int3 (California-Texas) Int4 (California-Kansas) Int5 (California-Canada)	2001:db8:acad::106:2 FE80::106:2 2001:db8:acad::105:2 FE80::105:2 2001:db8:acad::22:1 FE80::22:1 2001:db8:acad::23:1 FE80::23:1 2001:db8:acad::24:1 FE80::24:1
Texas Router	Int1 (California-Texas) Int2 (Texas-Kansas) Int3 (Texas-New Jersey)	2001:db8:acad::22:2 FE80::22:2 2001:db8:acad::26:1 FE80::26:1 2001:db8:acad::27:1 FE80::27:1
Kansas Router	Int1 (California-Kansas) Int2 (Canada-Kansas) Int3 (Texas-Kansas) Int4 (Kansas-New Jersey)	2001:db8:acad::23:2 FE80::23:2 2001:db8:acad::25:2 FE80::25:2 2001:db8:acad::26:2 FE80::26:2 2001:db8:acad::28:1 FE80::28:1

New Jersey Router	Int1 (Texas-New Jersey)	2001:db8:acad::27:2 FE80::27:2
	Int2 (Kansas-New Jersey)	2001:db8:acad::28:2 FE80::28:2
	Int3 (UK-New Jersey) <i>[TAT-14]</i>	2001:db8:acad::100:2 FE80::100:2
	Int4 (France-New Jersey) <i>[FA-1]</i>	2001:db8:acad::101:2 FE80::101:2
Canada Router	Private (Internal)	FC00::2:1
	Int1 (California-Canada)	2001:db8:acad::24:2 FE80::24:2
	Int2 (Canada-Kansas)	2001:db8:acad::25:1 FE80::25:1