COMPARISON OF VARIOUS TRANSITION MECHANISMS FROM IPv4 TO IPv6

A MASTER'S THESIS

IN

COMPUTER ENGINEERING

ATILIM UNIVERSITY

BY

FARIS AL-FAYYADH

JULY, 2015

COMPARISON OF VARIOUS TRANSITION MECHANISMS FROM IPv4 TO IPv6

A THESIS SUBMITTED TO

THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCE

OF

ATILIM UNIVERSITY

BY

FARIS AL-FAYYADH

IN PARTIAL FULFILLMENT OF THE REQUIREMENT FOR THE

DEGREE OF MASTER OF SCIENCES

IN THE

DEPARTMENT OF COMPUTER ENGINEERING

JULY, 2015

Approval of the Graduate School of Natural and Applied Sciences, Atılım University.

_____

Prof. Dr. K. İbrahim AKMAN

Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science.

_____

Prof. Dr. K. İbrahim AKMAN

Head of Department

This is to certify that we have read the thesis "**Comparison of Various Transition Mechanisms from IPv4 to IPv6**" submitted by "**Faris AL-FAYYADH**" and that in our opinion it is fully adequate, in scope and quality, as a thesis for degree of Master of Science.

_____

Assoc. Prof. Dr. Murat KOYUNCU

Supervisor

Examining Committee Members

Assoc. Prof. Dr. Murat KOYUNCU _____

Asst. Prof. Dr. Gökhan ŞENGÜL _____

Asst. Prof. Dr. Sibel TARIYAN ÖZYER _____

Date: July 13, 2015

I declare and guarantee that all data, knowledge and information in this document has been obtained, processed and presented in accordance with academic rules and ethical conduct. Based on these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Faris AL-FAYYADH

Signature:

ABSTRACT

COMPARISON OF VARIOUS TRANSITION MECHANISMS FROM IPv4 TO IPv6

AL-FAYYADH, Faris

M.Sc., Computer Engineering Department

Assoc. Prof. Dr. Murat KOYUNCU

July 2015, 62 pages

IPv4 which is the old version of Internet Protocol has a new successor named IP Next Generation (IPng) or IPv6 developed by IETF. This new version is developed especially to resolve some issues of IPv4 like scalability, performance and reliability. Although new version is ready for usage, it is obvious that it will take years to transit fully from IPv4 to IPv6. We have to use these two versions together for a long time. Therefore, we have to investigate and know transition mechanisms that we can use during transition period to achieve a transition with minimum problem.

This thesis analyzes present IP transition techniques. Here an attempt has also been made to make empirical evaluation of the three most generally used transition mechanisms which are Automatic 6to4 Tunneling, Manual 6in4 Tunneling and Dual-Stack. The obtained test results are also compared with the results of native IPv6 and native IPv4 environments. Empirical evaluation is based on simulations which are carried out using OPNET simulation framework. The outcomes of the thesis are important for providing an insight for choosing an appropriate transition technique, an idea about network capacity planning and migration.

Keywords: IPv6, IPng, IPv4, Transition Mechanism, OPNET

ÖZ

IPv4'den IPv6'ya FARKLI GEÇİŞ MEKANİZMALARININ KARŞILAŞTIRMASI

AL-FAYYAD, Faris

Yüksek Lisans, Bilgisayar Mühendisliği Bölümü

Doç. Dr. Murat KOYUNCU

Temmuz 2015, 62 Sayfa

Internet protokolünün eski sürümü olan IPv4, IP Gelecek Nesil (IPng) veya IPv6 adlı IETF tarafından geliştirilmiş yeni bir sürüm ile takip edilmektedir. Bu yeni sürüm özellikle IPv4'ün ölçeklenebilirlik, performans ve güvenilirlik gibi bazı sorunlarını çözmek için geliştirilmiştir. Yeni sürüm kullanıma hazır olmasına rağmen IPv4'den IPv6'ya tam geçiş yıllar alacaktır. Bu nedenle uzun bir süre bu iki versiyonu da birlikte kullanmak durumundayız ve en az sorunu olan bir geçiş dönemi sağlamak için, geçiş dönemine ait mekanizmaları araştırıp öğrenmemiz gerekmektedir.

Bu tez, mevcut IP geçiş tekniklerini analiz etmektedir. Ayrıca, Otomatik 6to4 Tünel, Manuel 6in4 Tünel ve Dual Stack olarak adlandırılan ve yaygın olarak kullanılan üç geçiş mekanizmasının ampirik değerlendirmesi yapılmaktadır. Elde edilen test sonuçları aynı zamanda saf IPv6 ve saf IPv4 ortamlarının sonuçları ile karşılaştırılmıştır. Ampirik değerlendirme OPNET simülasyon çerçevesini kullanarak yürütülen simülasyonlara dayanmaktadır. Tezin sonuçları uygun bir geçiş tekniği seçimi, ağ kapasite planlaması ve geçiş hakkında fikir vermesi açısından önemlidir.

Anahtar Kelimeler: IPv6, IPng, IPv4, Geçiş Mekanizması, OPNET

To My Family and Friends

ACKNOWLEDGEMENTS

I would like to express my deep gratitude to Assoc. Prof. Dr. Murat KOYUNCU, my research supervisor, for his patient guidance, enthusiastic encouragement and useful critiques of this research work.

I would also like to extend my thanks to the technician of the laboratory of the Information System Engineering Department for his help in offering me the resources in running the program.

I would also like to express my great thanks to all my friends and Iraqi government for their support and help.

Last but not least, I wish to thank my father Khaleel AL-FAYYADH, my mother, my wife and my son Yousef for their support and encouragement.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

NAT	Network Address Translation
DHCP	Dynamic Host Configuration Protocol
IETF	Internet Engineering Task Force
IPng	IP next generation
NATs	Network Address Translators
DoD	Department of Defense
API	Application Programming Interface
BDMS	Bi-directional Mapping System
DSTM	Dual Stack Transition Mechanism
EED	End-to-End delay
GRE	Generic Routing Encapsulation
ISATAP	Intra-Site Automatic Tunnel Addressing Protocol
OSPF	Open Shortest Path First
QoS	Quality of Services
IP	Internet Protocol
MTU	Maximum-Transmission Unit
CIDR	Classless Inter Domain Routing
MIPv6	Mobile IPv6
TOS	Type of Services
TLA	Top Level Aggregator
NLA	Next Level Aggregator
SLA	Site Level Aggregator
NGtrans	Next Generation Transition
DNS	Domain Name System
PDF	Probability Density Function

CHAPTER 1

INTRODUCTION

The increasing use of the Internet in the last two decades has shown the potential that the Internet can change and improve different areas such as education, businesses or entertainment, etc. No one could guess that World Wide Web would become a worldwide communication channel, and it was thought that the number of addresses provided by the Internet Protocol version 4 (IPv4) were more than enough. IPv4 developed in 1981 and was used to make interconnection between different networks. After three versions, IP got the name of next version number 4 and declared as IPv4 to the internet community. The first version of IPv4 was generally used to ensure that two computers or any two network devices could connect with each other. As there is an ever growing expansion and advancement in the network and internet mechanism, the requirement of unique addresses is increasing. Therefore, to solve the address limitation of current IPv4, several technologies have come out like Network Address Translation (NAT) and Dynamic Host Configuration Protocol (DHCP). However, while these technologies decrease the addressing shortage, they prevent IP level end-to-end security, reduce robustness, so they are not good solutions [1].

To fix the problem of current IPv4, a new internet protocol was developed, which is called Internet Protocol version 6 (IPv6). This protocol was developed by the Internet Engineering Task Force (IETF) with reference to routing addresses and security. IPv6, actually is known as IP next generation (IPng), is chosen from numerous suggested alternatives as the most appropriate successor of the present Internet Protocol (IPv4). IPv6 is more effective, scalable, secure and routable than IPv4.

The reason to create a new Internet Protocol (IPv6) is basically to boost the quantity of IP address space [2]. The IPv6 can give above $3.4x10^{38}$ unique addresses as compared to IPv4 which gives $4.3x10^{9}$ unique addresses (IPv6 has the capacity of 128-bit/16 bytes address scheme, whereas IPv4 has just 32 bits/4 bytes). This means, IPv6 solves the problem by eradicating the requirement of Network Address. It easily

provides all devices like MP3 player, telephone, mobile phone or automobiles their own IP addresses. Moreover, it also supports multimedia transmissions, security and scalability. This proves that IPv6 was modeled by keeping in consideration the future applications. Therefore, numerous organizations like Department of Defense (DoD) of USA, have made a timetable to implement the new IPv6 for their requirement of future deployments [3].

1.1 Motivation of Research

Transition to IPv6 is not possible that quickly as the installed general network infrastructure is IPv4 based and we need co-existence of them and integration between them for a period of time until the process of migration is complete. Changing the already present data structures needs remodeling of the inserted IP addresses as well as the Application Programming Interface (API) supporting IPv4 needs alteration [4].

The NAT, DHCP, ICMP and PPP legacy protocols which are written with IPv4 preference in mind have to go through changes as well. All these changes are really important to the Internet as not only the software like operating systems and application programs etc., but also the hardware requires enhancement at TCP/IP layers. Therefore, for IPv6 support, current protocols are modernized or re-organized. In times to come, the Internet will completely change into huge cluster of non-comparable protocols working in dual IPv6/IPv4 environment for an extended period of time. For smooth continuous coexistence of IPv6/IPv4 protocols, numerous transition mechanisms are suggested by Internet Engineering Task Force (IETF) that is largely categorized under Transition Mechanism, Tunneling and Dual-Stack.

1.2 Related Works

There are different studies addressing the transition mechanisms and the most related ones are highlighted in the following:

Mellor and Awan [2] investigate the behavior of Bi-directional Mapping System (BDMS) transition mechanism in comparison to Dual-Stack Transition Mechanism (DSTM) using three scenarios: first:V4-to-V4 and V6-to-V6 Direct-link connections via a router, second:V4-to-V6 and V6-to-V4 connections via the BDMS translator and third:V6-to-V4 connection via the DSTM gateway. End-to-End delay (EED) and Throughput are used as key performance metrics. Simulation results have shown that the EED of the v6-to-v4 in DSTM communication session is large compared with the EED of the other types of communication sessions due to the impact of the tunneling process that causes more delay. In addition, the throughput of v6-to-v4 in DSTM communication session is small as compared with the throughput of the other types of communication sessions in scenarios one and two due to the impact of the tunneling process that may lead to network congestion which causes more reduction in the throughput.

Latief and Parvez [3] evaluated three most commonly used transition mechanisms, namely Dual-Stack, Automatic 6to4 Tunneling and Manual 6in4 Tunneling using TCP delay, Link Throughput and Response Time as the key performance metrics. The result shows that using automatic tunnels (6to4) outperforms Dual-Stack and Manual 6in4 Tunneling. They compared also the result with native IPv6 environment. Results have shown that IPv6 only network produces a higher throughput and provides a minimum delay compared to transitioning networks. This also means that IPv6 has faster packet processing and forwarding capability than transitioning networks. The worst performance is shown by manual tunneling and Dual-Stack scenarios.

Savita and Monalisa [4] study different transition strategies like Dual-stack, Manual Tunnel, Generic Routing Encapsulation (GRE), Intra-Site Automatic Tunnel Addressing Protocol (ISATAP) and 6to4 tunnel using Round-trip-time, Jitter and throughput as the key performance metrics for network with address allocation and router configuration with Open Shortest Path First (OSPF) routing protocol. The result shows that automatic tunnels (6to4 and ISATAP) outperform manual tunnels. They recommend using the automatic tunnels for the applications which are sensitive to jitter, throughout and RTT.

Albkerat and Issac [5] examined the performance of five different network topologies which are IPv4, IPv6, Dual-Stack, 6to4 tunnel, and manual tunnel. To provide appropriate results, the network performance is analyzed across various mechanisms. More delay is observed with 6to4 and manual tunnels since the packets are not transferred directly in comparison to Dual-Stack which shows less delay. Four different network simulations are examined with three dissimilar data rates: 1, 2 and 3 Mbps. The obtained results show that pure IPv6 throughput is higher than the others. Manual tunneling gives better throughput than 6to4 up to 5 Mbps.

1.3 Objective of Research

The main idea of this thesis is to focus on the comparison between protocol stacks of IPv4 and IPv6 as well as transition mechanisms from IPv4 to IPv6. To achieve the main idea, the following objectives are considered in this thesis:

1. To study the transition mechanisms from IPv4 to IPv6
2. To analyze the performance of networks where different transition mechanisms are used
3. To give recommendations considering obtained results

1.4 Methodology of Research

This thesis used the simulation approach as evaluation technique. The simulation has been widely accepted as a research methodology for computer engineering research areas. Since there are a number of simulation tools available, we have investigated different tools to determine most suitable tool for this thesis. Finally, we had chosen OPNET Simulator to evaluate the performance of the transition mechanisms from IPv4 to IPv6.

1.5 Contribution of Research

The main contribution of this thesis is to provide comprehensive comparison between transition mechanisms (Dual-stack, Automatic 6to4 and manual 6in4) from IPv4 to IPv6 for different traffic applications like Voice, Video conferencing, web browser (HTTP), Database, E-mail and FTP. These applications were chosen, due to their popularity among the Internet's users today. This research deals with routing performance of IPv6 and IPv4 transition mechanism. The Evaluated parameters are jitter, traffic send; traffic received, throughput, response time and delays of the end users. In this thesis it is presumed that all routers are connected with point-to-point links and have local links for clients as well as all have default configurations.

1.6 Organization of Thesis

The thesis is organized as follows: Literature review and theoretical developments on IPv4 as well as IPv6 are given along with crucial differences in Chapter 2. Chapter 3 explains different transition mechanisms that can be used while upgrading from IPv4 to IPv6. Chapter 4 is basically about OPNET and also deals with the Dual-Stack transition mechanism. It presents obtained test results with various network metrics and discusses those outcomes. Chapter 5 deals with Automatic 6to4 and Manual 6in4 transition mechanisms by presenting obtained test results with various network metrics and discusses those outcomes. Chapter 6 gives conclusions and future work.

CHAPTER 2

LITERATURE REVIEW AND THEORETICAL DEVELOPMENTS

This chapter's major target is to give the theoretical background for IPv4 and IPv6. Here we have also examined some other studies pertaining to the same subjects. Since we have started to use IPv4 and IPv6 protocols together in recent years and probably we are going to use them together in the near future, we have to know the architectures of them, the differences between them, advantages and disadvantages of each one. That is why we are stimulated to work on these two protocols.

2.1 Internet Protocol (IPv4)

The network-layer protocol, i.e. Internet Protocol (IP), possesses not only control information but also addressing information which makes packets routed. IP, as the major network layer protocol in the internet protocol suite, is documented in RFC 791.There are two main duties of IP; the first one is to provide connectionless datagrams best effort delivery, and the second one is to support maximum-transmission unit (MTU) sized fragmentation and reassembly of datagrams [6].

Today almost 40% of the people in the world use internet connection. Whereas, it was not more than 1% in 1995, in fact it was even less. As seen in Figure 2.1, there is an enormous increase of internet users from the year 1999 to 2013 which is ten times more. The number of Internet users reached to 1 billion in 2005, 2 billion in 2010, and is around 3 billion when we come to 2014 [7].

It was soon evident that continuing with IPv4 is not possible with the rapid growth of the Internet [8]. Quantity of IP addresses in IPv4 was not sufficient to keep up with the proliferation of devices on the Internet. IPv4 has 32-bit address length which gives us about 4.2 billion IP addresses. When IPv4 was developed, it appeared to

have a sufficient amount of IP addresses. However, as time progressed, the Internet grew with the advent of new networking devices such as phones, televisions and gaming consoles, which were IP-capable. Temporary solutions were found to overcome the exhaustion of IPv4 address spaces [9].

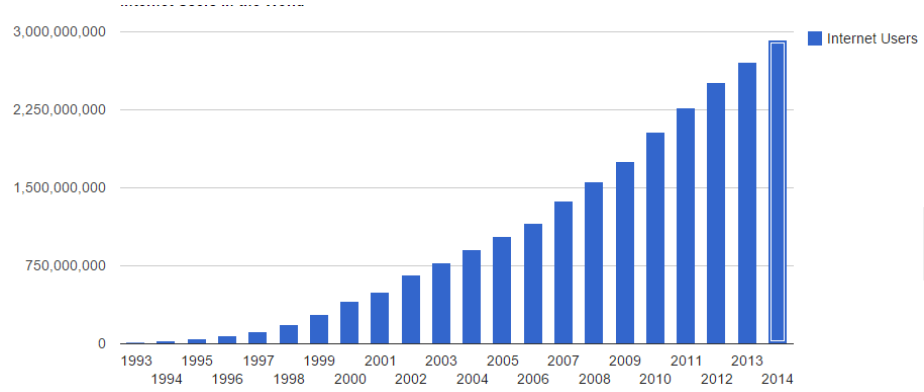

Figure 2.1: Internet Users in the World [7]

The first solution was Classless Inter Domain Routing (CIDR) which is a method for allocating IP addresses and routing IP packets in a more flexible way. The goal here was to decrease the growth of routing tables within the Internet in order to restrict the rapid exhaustion of IPv4 addresses.

The second solution was a technique termed Network Address Translation (NAT) in which one IP address could be translated to multiple hosts within the NAT network.

The third solution is termed Dynamic Host Configuration Protocol (DHCP) which is used on IP networks as the automatic configuration protocol. However, in CIDR, the need for larger routing tables in routers became evident, which resulted in routing difficulties. The NAT solution breaks the principle of the Internet and is not supported by some applications [10]. These three technologies were designed as solutions but made the networks much slower and complex [9]. In addition, these three technologies did not overcome the problem of IPv4 address exhaustion, but only delayed it.

2.2 New Version of IP (IPv6)

The Internet Engineering Task Force (IETF) came up with a solution called IP next generation (IPng) to solve the IP address exhaustion problem. IPng was a result of the proposals reviewed by IETF. IPng was not a complete protocol, but was a product which had to be reviewed considering the features and limitations. After multiple reviews and changes to IPng, IPv6 was developed [11] [12].

The IPv6 address size is 128 bits compared to the 32-bit address in IPv4. The 128-bit size gives approximately 1500 addresses per square foot of the earth's surface. Even if every device around you is IP capable, there are 1500 addresses per square foot which is sufficient for any kind of requirements. Thus, IPv6 has provided a solution to IP address exhaustion. The Internet community is taking time to adapt to IPv6. The main reason would be that it is difficult for IPv4 and IPv6 to coexist. Whenever an IPv6 host wants to communicate with an IPv4 host, it has to use transmission mechanisms. It may be predicted that when IPv4 addresses are exhausted the Internet community will be forced to adopt IPv6 conversion at a faster rate.

2.3 Benefits and Characteristics of IPv6 Usage

Apart from fulfilling the expected demands for the future address requirements, the advantages of using IPv6 for the skilled IT people are given below:

Scalability: IPv6 possesses 128-bit addresses compared to 32-bit IPv4 addresses. IPv6 provides 2^{128} theoretical addresses versus 2^{32} addresses of IPv4 [13].

Autoconfiguration and "Plug-and-Play": Plug and Play technology enables IPv6 devices to configure them independently. The device determines its address which will probably be unique based on the network prefix and its Ethernet MAC address. With this support, it is possible to plug a node into an IPv6 network without requiring any human intervention. This support is very important for the new mobile systems and its various services. As a result, network devices could connect to the network without manual configuration and without any servers such as Dynamic Host Configuration Protocol (DHCP) servers.

Mobility: Contrary to the Mobile IPv4 protocol, the Mobile IPv6 (MIPv6) helps avoid triangular routing experienced earlier, and makes it possible for mobile (WiFi) clients to select a new router without renumbering, which results in a more reliable and faster connection with less network interruption.

Security: IPv6 protocol has been developed so that the security features become an integral part of the protocol to ensure maximum security. Added security features do not affect the performance, speed and efficiency. Security features that have been added to the IPv4 protocol later as something optional are now found as a part of protocol implicitly in IPv6. IPv6 encrypts data and examines the integrity of the packets similar to those offered by the VPN data transmitted over the Internet [14].

Quality of Service: The quality of service in IPv6 can be handled in the same way as IPv4. Traffic Class support of IPv6 works well with the Differentiated Service model of Internet Engineering Task Force (IETF). In addition, the header of IPv6 has a new field called flow label, which can contain a label specifying a particular flow such as video or video stream. The source node generates this flow label. The existence of flow label enables devices on the way to take appropriate action based on this marker for quality of service.

2.4 IPv4 and IPv6 Header Comparison

The format of IPv6 header is simpler in comparison to IPv4, even though the quantity of address of IPv6 makes its header increase in size. The header size of IPv4 is basically just 20 octets; however, the options field variable length further builds the total IPv4 packet size. The header of IPv6 has a size of 40 octets. 6 IPv4 header fields are extracted out of 12 in IPv6. Some of the fields of the former protocol have been taken over by changing as well as adapting names. Some new fields are added to infuse new features as can be seen in Figure 2.3 [6].

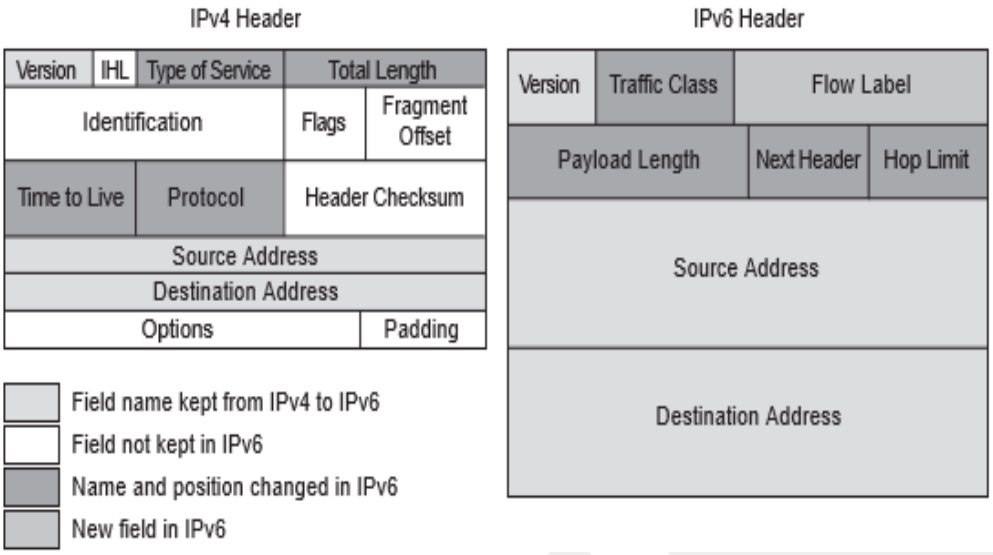

Figure 2.3: IPv4 and IPv6 Header Comparison [6]

Even though the removal helps simpler IPv6 header to process fast, whereas, overall performance as well as the routing efficiency is dependent on the treatment of option headers as well as the algorithms that any given device should run. Apart from this, IPv6 has the advantage of having 64-bit processors of the present generation, because of which IPv6 header fields comprise of 64 bits.

In IPv6 header, fragmentation does not require fields in it and is handled in a different manner. IPv6 routers do not fragment datagrams, which further eradicates issues relating to processing done by IPv4 routers when dealing with fragmentation. In IPv6 networks, the source device determines the maximum transmission unit (MTU) size using a discovery protocol which eliminates fragmentation requirement on the routers [13][14].

As checksum of IP header is perceived neither necessary nor practical as most of the link-layer technologies do error control, it is removed from IP layer. In addition to the link layer technologies, the transport layer which tackles end-to-end connection possesses checksum which makes it able to detect errors. Here it is important to note that this cancelation enforces upper layer optional checksum become compulsory.

IPv6 changes IPv4 options field and handles the same functionalities using extension headers. Most of the other fields were a bit modified or were not replaced at all [13]. Moreover, the header is 64 bits lined to a fewer number of fields to make faster

processing with the help of present processors. The header of IPv6 possesses the fields specified below:

Version Number: It is a 4-bit field as in IPv4. The field contains the number 4 for IPv4 and number 6 for IPv6.

Traffic Class: This is just the same as the IPv4 TOS service field which is an 8-bit field.

Flow Label: It is an IPv6 new 20-bit field. It is utilized for tagging packets of a certain flow to distinguish network layer packets.

Payload Length: Just like IPv4 total length, it points out complete length of the packet data portion.

Next Header: It shows which kind of information follows the basic IPv6 header.

Hop Limit: This is similar to the Time to Live field of IPv4 packet header. The aim of the Hop Limit field is to determine the highest routers' hops that IPv6 packet may go through before it is discarded.

Source Address: Source address of IPv6 is just like the one of IPv4, only difference is its field size having 128-bit rather than 32-bit source address of IPv4.

Destination Address: The destination address field of IPv6 is just like the IPv4 packet header Destination Address field, the only difference is that the field has 128-bit destination address rather than 32-bit destination address of IPv4.

2.5 IPv6 Address Types

There is a big difference between the addressing requirement of an IPv4 node and an IPv6 node. An IPv4 node usually uses one IP address; but an IPv6 node needs more than one IP address. IPv6 addresses are classified into three categories as follows:

Unicast: As the name implies it is a single interface addressing that when any packet is dispatched to its address it is sent to a recognized interface by the same unicast address.

Anycast: Belonging to various nodes providing the same service, a packet is sent to the nearest interface.

Multicast: Usually this is associated to various nodes or for a set of interfaces. A packet is sent to every interface recognized by (in a specific scope) the multicast address. Table 2.1 compares IPv4 versus IPv6 addressing from different perspectives.

Table 2.1: Comparison of IPv4 and IPv6 Addressing [15]

Address Space Element	IPv4 Address	IPv6 Address
Unspecified address	0.0.0.0	0:0:0:0:0:0:0:0: or ::
Loopback address	127.0.0.1	0:0:0:0:0:0:0:1: or ::1
Address type	Public IPv4 addresses	Global addresses (aggregatable global unicast addresses)
Address type	Private IPv4 addresses; 10.0.0.0/8 172.16.0.0/12 192.168.0.0/16	Site-local addresses, Always begin with FEC0::/48
Address type	Automatic Private IP Addressing (APIPA), which uses the 169.254.0.0/16 prefix	Link-local addresses, Always begin with FE80::/64
Text representation	Dotted-decimal format	Colon-hexadecimal format with suppression of leading zeros and zero compression. IPv4-compatible addresses are expressed in dotted-decimal notation.
Network bits representation	Subnet mask in dotted-decimal format or prefix-length notation	Prefix-length notation only
DNS name resolution	IPv4 host address (A) resource record	IPv6 host address (AAAA) resource record

A single interface address is named as a unicast address. Unicast delivers packets to the identified interface by the same address. The details of different unicast addresses are given below:

Aggregatable Global Unicast: It is the effective hierarchical addressing and routing supported by IPv6 design. In the internet, the aggregatable global unicast addresses are recognized by the 001 formal prefix as shown in Figure 2.4. The usage of fields of the address structure is given below:

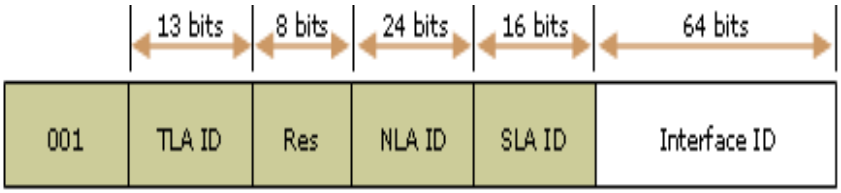

Figure 2.4: Structure of an IPv6 Global Address [16]

TLA ID: Top Level Aggregator (TLA) is an address recognized in the routing hierarchy at the highest level.

Res field: It is the expansion size for Next Level Aggregator (NLA) ID or TLA ID for the future use.

NLA ID: It specifies the address of a special customer site/organization.

SLA ID: It points out an address as the Site Level Aggregator (SLA), and is utilized for labeling subnets within organizations.

Interface ID field: It shows a special node interface in a subnet.

Local-Use Unicast: There are two kinds of local-unicast addresses:

Link-local is designed to utilize on a single link for address auto-configuration when a host cannot obtain an IP address, and it has the format given in Figure 2.5.

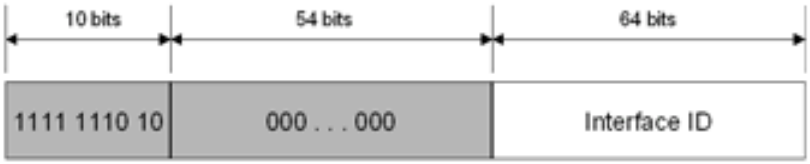

Figure 2.5: Link-Local Addresses [17]

Site-Local is equivalent to the IPv4 private address space. Site-Local addresses were originally designed to be used for addressing inside of a site without the need for a global prefix, and it has the format given in Figure 2.6.

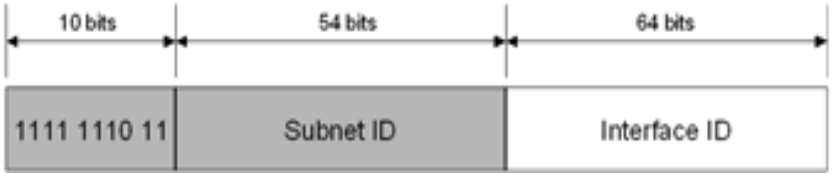

Figure 2.6: Site-Local Addresses [17]

Unspecified Address: The unspecified address indicates the absence of an IPv6 address and it has the form 0:0:0:0:0:0:0:0 or ::0. For example, an initializing node on an IPv6 network may use the unspecified address as the source address in its packets until it receives its IPv6 address.

Loopback Address: It is represented by 0:0:0:0:0:0:0:1 or ::1. The loopback address can be used by a node to send an IPv6 packet to itself. The loopback address in IPv6 has the same functionality with the loopback address 127.0.0.1 in IPv4 [18].

IPv4 Compatible IPv6 Addresses: There are two types of IPv6 addresses which can contain IPv4 addresses:

The first type is the "IPv4-compatible IPv6 address" and has the format 0:0:0:0:0:0:A.B.C.D or ::A.B.C.D, where A.B.C.D is an IPv4 unicast address. Figure 2.7 shows representations of the IPv4-compatibly IPv6 address. The IPv6 transition mechanisms involve a technique for hosts and routers to dynamically tunnel IPv6 packets over IPv4 routing infrastructure. IPv6 nodes that use this technique customize special IPv6 unicast addresses that involve a global IPv4 address in the low-ranking 32 bits [19].

The second type of IPv6 address which holds an embedded IPv4 address is called "IPv4-mapped IPv6 address" This type of address has the format 0:0:0:0:0:FFFF:A.B.C.D or ::FFFF:A.B.C.D, where A.B.C.D is an IPv4 unicast address. Figure 2.8 gives an IPv4-mapped IPv6 address structure. This address type is used to represent the addresses of IPv4 nodes as IPv6 addresses.

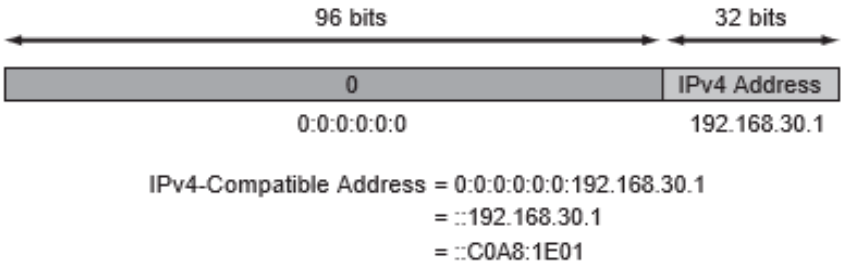

Figure 2.7: IPv4-Compatible IPv6 Address Format [6]

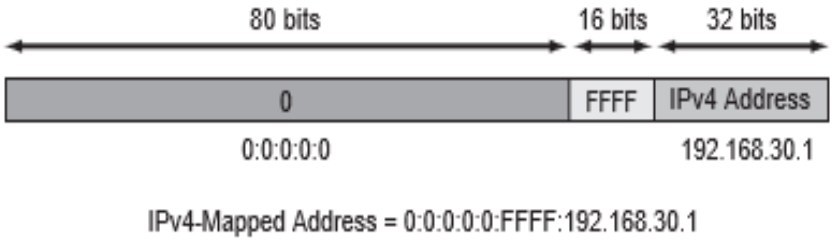

Figure 2.8: IPv4-Mapped IPv6 Address Format [6]

2.6 Performance Metrics

Users and network administrators would like to see the network performance metrics for different purposes such as optimizing their use of the network and troubleshooting performance degradations when they happen. There are a variety of metrics used to analyze data networks. Important ones are summarized in this section.

Average Packet End-to-End Delay (Latency): it is the time, which is measured in the form of seconds, taken by a packet to go from one source to a destination. The important sources of delay can be classified into groups like transmission delay, processing delay, queuing delay and propagation delay. end-to-end delay is measured

as the difference between the arrival time and sending time of a packet. Equation 2.1 gives the average packet end-to-end delay calculation [20].

$$Average\ Packet\ end\ to\ end\ Delay = \frac{\sum i\ (Packet\ Arrival_i - Packet\ Start_i)}{n} \tag{2.1}$$

Here, the 'Packet Arrival' represents the time when the packet 'i' reaches destination and the 'Packet Start' represents the time when the packet leaves the source. 'n' is the total packet number.

Average Jitter: It can be said that jitter is the variation in delay or in other words it is the packet delay variation measured in seconds. It can be calculated using packet end-to-end delays. It is a crucial parameter for determining network's performance and QoS that network provides. It is also generally utilized as an indicator of stability and consistency of network. Equation 2.2 indicates the method of calculation of average jitter [20].

$$Jitter = \frac{\sum i\ |(Packet\ Arrival_{i+1} - Packet\ Start_{i+1}) - (Packet\ Arrival_i - Packet\ Start_i)|}{n-1} \tag{2.2}$$

Download Response Time: This is the application elapsed time in seconds from the time of sending of the request to the time of response received. It measures the time of the request of a service to the moment it is allowed. When there is an increase in the response time it is an indication of a network problem. Here it needs to be noted that to measure the performance of FTP traffic, download response time is used [21].

Page Response Time: It is the time that an HTTP system or server uses to react to an input given. It indicates time needed to retrieve a whole page with every object in it and this also is utilized as a metric to estimate a website's success. Apart from this, it is used to estimate HTTP traffic performance as well [21].

Throughput: This presents the successful delivery of number of bits by network to users. Equation 2.3 indicates how the throughput is calculated [20]. Here '$packetSize_i$' is the total packet size of the i^{th} packet reaching the destination, the '$PacketStarttime_0$' is the time when the first packet leaves from the source and '$PacketArrivaltime_n$' is the last packet arrival time. The units which are used to measure throughput are packets/second, bits per second (bps), channel utilization (%) and packets/slot.

$$Throughput = \frac{\sum_i Packet\ size\ (i)}{Packet\ Arrivaltime(n) - Packet\ Starttime(0)} \tag{2.3}$$

Delay: It is measured in seconds like others. It presents end-to-end delay of both received and forwarded packets to higher layers of every network wireless and wired nodes [21].

Packet Loss: This particular term defines the failure of packet to reach the destination due to the device's overload or unacceptance of incoming data at a certain moment. These packets are dropped during network congestion period. Some applications cannot tolerate any packet loss, some others can tolerate to a certain level. For example, voice applications, can tolerate up to 3% packet loss (1% is optimum) during conversation. Equation (2.4) shows the packet loss ratio calculation which is explained as the number of ratio of packets lost to the number of total packets transmitted. In the equation, N is the total packets transmitted in a particular period of time and N_L is the packet lost number in the same period of time [22].

$$\text{Loss packets ratio} = (N_L / N) \times 100\% \tag{2.4}$$

CHAPTER 3

TRANSITION MECHANISMS

This research focuses on the transition mechanisms from IPv4 to IPv6 networks, in particular the Dual-stack, Automatic 6to4 and Manual 6in4. This chapter provides the detailed information for the transition mechanisms. Section 3.1 describes the transition mechanisms from IPv4 to IPv6, while Section 3.2 elaborates the Dual-Stack Transition Mechanism. Meanwhile, Section 3.3 explains the tunneling mechanism. Section 3.4 explains the translation mechanism.

3.1 Transition Mechanisms from IPv4 to IPv6

The IPv6 which is the latest version of internet protocol isn't backward compatible with IPv4, which means that IPv6 networks cannot communicate with IPv4 networks. That is they cannot send packets to each other, IPv6 network can only send IPv6 packets to other IPv6 networks, as well as the IPv4 network can send to other IPv4 networks. Therefore, there is an important interoperability problem with the coexistence of both protocols on the internet [23].

In order to solve the communication problem between IPv4 network and IPv6 network and make packets transmission between networks smoothly, the Internet Engineering Task Force (IETF) and Next Generation Transition (NGtrans) work group established IPv4/IPv6 transition mechanisms to eliminate lack of compatibility problem and support co-existence of protocols, which will last probably for a very long time. To comprehend the transition mechanisms and their importance, it is necessary to study and analyze each transition mechanism properly. The transition mechanisms are divided into three main groups as dual-stack, tunnels (including configured and automatic tunnels), and translation mechanisms. These mechanisms are explained one by one in detail below.

3.2 Dual Stack Transition Mechanism (DSTM)

Dual-stack transition mechanism enables IP devices to run both IP stacks (IPv4 and IPv6) in a single node. Dual-Stack Mechanism possesses double protocol stacks (IPv4 and IPv6) which work parallel and permits network nodes to operate either by IPv6 or IPv4 [24]. It can be applied in both network nodes and end systems. End systems allow IPv6 and IPv4 applications to work simultaneously as it supports the transportation of their packets.

As defined in IETF RFC 2893, a network node implements both IPv6 and IPv4 in parallel in the dual-stack mechanism as shown in Figure 3.1. IPv4 and IPv6 applications use their own stacks in parallel. Figure 3.2 shows TCP/IP model for a dual-stack node.

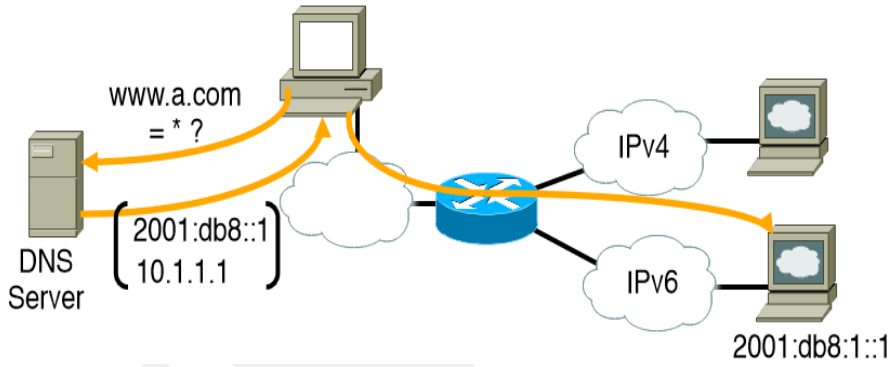

Figure 3.1: End-System with Dual-Stack Transition Mechanism [25]

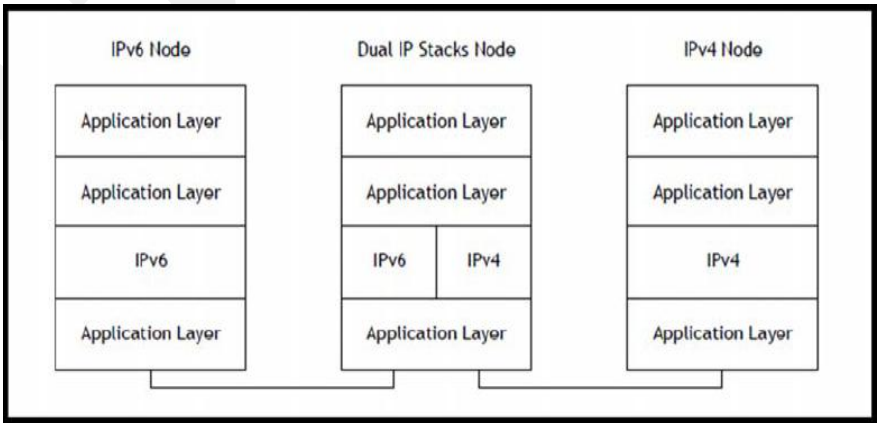

Figure 3.2: TCP/IP Model for Dual-Stack Node

The IP protocol version can be determined based on version field of IP header, accordingly a decision is made about the stack to be used. Stack selection might be done according to the returned Domain Name System (DNS) records, since the address types are generally derived from the DNS lookups. Most of the operating systems used today like Windows XP, Vista, Windows 7, Windows server 2003, Linux, Mac OS support stacks of both IP protocols. This is an answer for the question why the dual stack mechanism is used generally and mostly as a transition mechanism solution between IPv4 and IPv6 networks. Unfortunately this mechanism just allows same network nodes to communicate with each other i.e. IPv6 to IPv6 and IPv4 to IPv4. A lot of effort needs to be done to make an appropriate solution to support communications from IPv4 to IPv6 and vice versa.

The IPv6 and IPv4 functions do not depend on each other; therefore the dual-stack makes both protocols run side by side. As both IPv4 and IPv6 are supported in dual-stack case, both protocols can benefit from multicast policies, independent routing, security, high availability (HA) and quality of service (QoS). So that the dual stack also provides forwarding performance benefits as they go forward without any requirement of lookup ahead as well as additional encapsulation [26].

3.3 Tunnels

Tunneling is a process of encapsulating one protocol into other protocol. Wrapping an IPv6 packet within an IPv4 packet is shown in Figure 3.3. With the help of such tunneling mechanism, a packet can be carried by an incompatible network towards the destination network [27]. The tunneling migration strategy is that, a node encapsulates an IPv6 packet in an IPv4 packet for transmission across an IPv4 network and then the packet is de-capsulated to the original IPv6 packet by another node. The tunnel mechanism can be utilized to establish connection between IPv6 networks which are in isolation. Unfortunately this is not a permanent solution. When dual-stack or native IPv6 is fully implemented, there will be no requirement for tunneling strategies [4] [28].

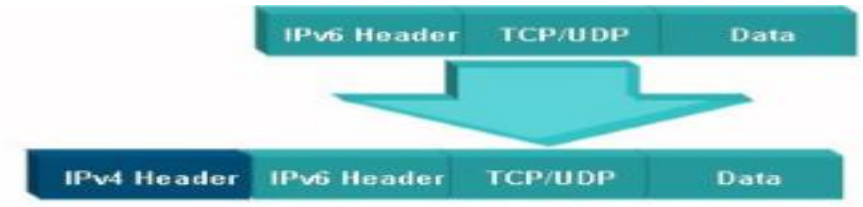

Figure 3.3: IPv6 over IPv4 Tunneling [28]

3.3.1 Configured Tunneling (Manual Tunneling)

If network administrators manually configure the tunnel within the end devices, this is called 'configured tunneling' or 'explicit tunneling'. Configuration information that is stored in encapsulating end devices is used to determine the addresses of tunnel endpoints. These tunnels might be bidirectional or unidirectional. Bidirectional configured tunnels work like a virtual point to point link. Figure 3.4 shows a manually configured tunnel that is utilized to link IPv6 hosts or networks over the IPv4 infrastructure. Generally these tunnels are utilized when exchanged traffic is regular. Main disadvantage of the technique is that it requires more administration effort when the quantity of tunnels increases [29].

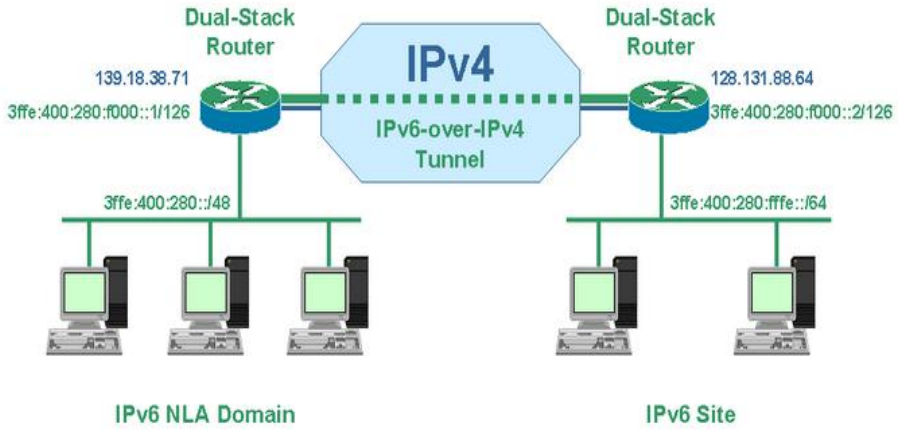

Figure 3.4: Configured Tunnel (Manually) [30]

3.3.2 Automatic Tunneling

If a device directly creates its own tunnels this called 'automatic tunneling'. In this type of tunnel, the address of an IPv4 tunnel endpoint is chosen from the inserted IPv4 address in IPv4 compatible destination address of the tunneled IPv6 packet [31]. So the packet which is being tunneled helps in determining the address of the tunnel endpoint. Figure 3.5 shows Automatic Tunneling technique.

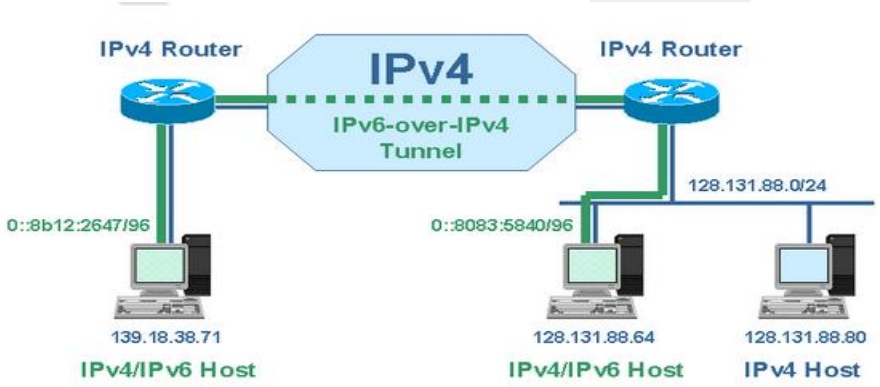

Figure 3.5: Automatic Tunneling [30]

The packet can be delivered through automatic tunneling if the IPv6 address is of type IPv4-compatible address. But when the destination address is IPv6-native, it is not possible to deliver the packet through automatic tunneling. To direct automatic tunneling, a routing table needs to be entered. A specific static routing table entry should be defined for the prefix **0:0:0:0:0:0/96**. Packets matching with this prefix are delivered to a pseudo-interface driver that makes automatic tunneling. Generally these tunnels can be utilized between individual hosts or networks in which there are incidentally needs for traffic exchanges.

3.3.3 6to4 Automatic Tunneling

In 6to4 tunneling technique, tunneling endpoints are configured automatically between devices. 6to4 mechanism means IPv6 traffic is tunneled upon IPv4 networks between separated IPv6 networks. The format of 6to4 network address includes the prefix 2002::/16 after which the universally distinct IPv4 address come [32]. A

concatenated form of a 48 prefix is given as an example in Figure 3.6 for the 192.168.99.1 (IPv4 address), 2002:c0a8:6301::/48 prefix of 6to4 address (where c0a8:6301 is the hexadecimal notation for 192.168.99.1).

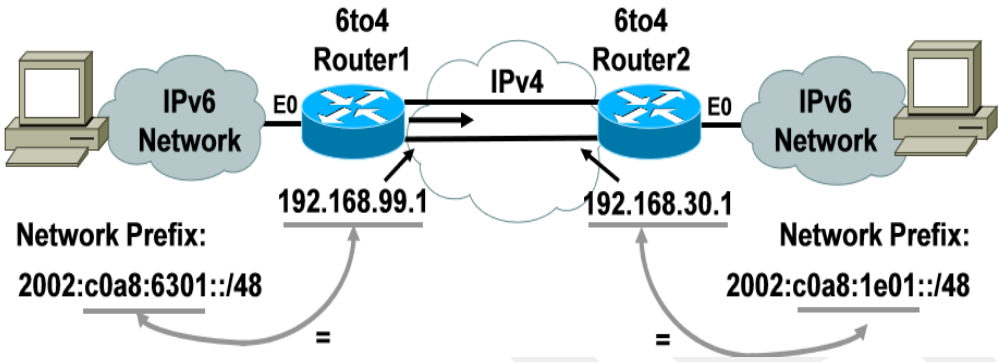

Figure 3.6: 6to4 Addresses in a Network [33]

3.3.4 Intra-Site Automatic Tunnel Addressing Protocol (ISATAP)

ISATAP is an IPv6 transition mechanism used to transmit IPv6 packets between dual-stack nodes on top of an IPv4 network. ISATAP is used to link IPv6 address with the specific prefix fe80::5efe/96 and this address is followed by the IPv4 which is 32 bit as shown in Figure 3.7. The IPv6 address will be within an IPv4 address [43]. The ISATAP tunnel is capable to supply a link between IPv6 and IPv4 routers. When the link is established the host within ISATAP receives an address called local ISATAP address and then host detects the next step of the ISATAP router. The packets are then sent by the tunnel after inserting the IPv6 address into an IPv4 address [35]. At the receiving side, the IPv4 header is deleted and the packet is sent to the IPv6 Host; there the server sends the packets to the ISATAP network and finally the ISATAP router prepares the IPv6 packets into IPv4 and sends them to the ISATAP host, which then removes the IPv4 header and extracts the IPv6 packets [36].

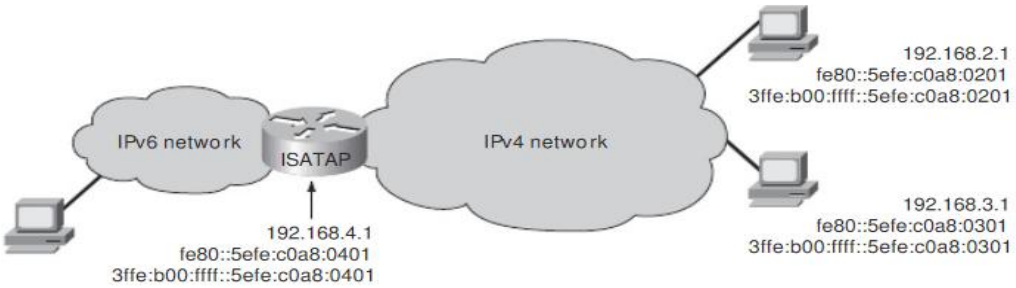

Figure 3.7: ISATAP [34]

3.4 Translation Mechanisms

Translation mechanism refers to devices capable of direct conversion from the IPv4 protocol to the IPv6 protocol. This mechanism requires translators that can convert IPv4 address to IPv6 address as shown in Figure 3.8. When translation mechanism is used, there is no need for dual-stack network and the network interoperability problem is solved since routers play as communicators. However, translation mechanism faces with limitation problem like IPv4 Network Address Translation (NAT) as well as it is hard to control on larger scale networks.

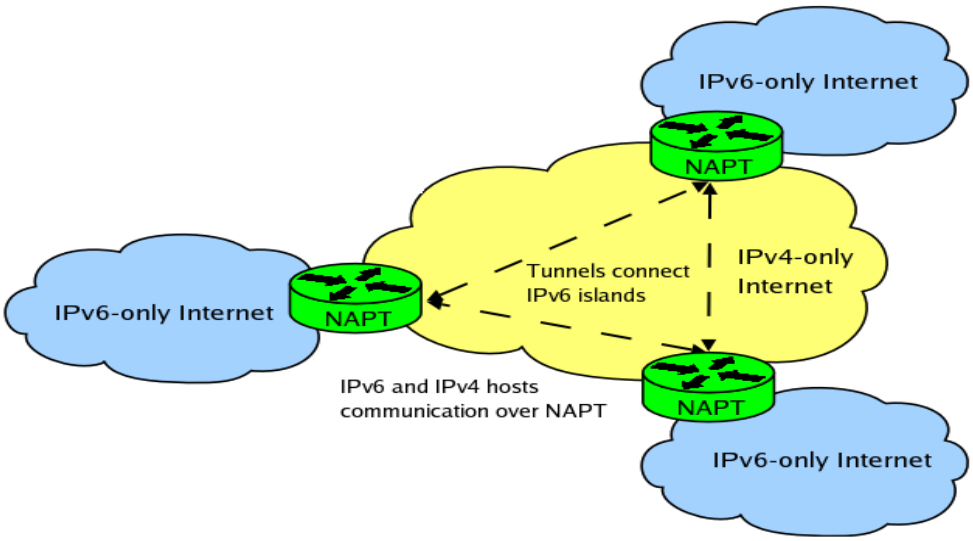

Figure 3.8: Translation Mechanisms [37]

CHAPTER 4

DESIGN AND EVALUATION OF DUAL-STACK TRANSITION MECHANISM

In this chapter, we test Dual-Stack Transition Mechanism for both Internet Protocol versions IPv6 and IPv4. We use OPNET Modeler for designing and testing of networks, and analyzing the obtained results. Therefore, we first explain OPNET Modeler. Then, we describe the simulation scenario in order to study and investigate the dual-stack transition mechanism.

4.1 OPNET

OPNET (OPtimum Network Performance) is a simulator which provides a full development environment that models and evaluates the performance of the communication networks as well as distributed systems. OPNET is preferred over other simulation software due to its user-friendly simulation platforms, high-quality documentation, and compatibility with new communication technologies and, these features make OPNET one of the best simulation tools that are used to study the performance of data communication networks by researchers [38][39]. Researchers prefer the OPNET Modeler due to its powerful tools, which allow researchers to specify models in more details.

OPNET uses a number of model-specification editors which are organized in a hierarchical way in order to simulate the structure of real network systems. These editors are Project Editor, Node Editor, Process Editor and Parameter Editor,

Parameter Editor, which is used for developing representation of a system being modeled, is used to define parameters that are more complex than simple numeric or string input. Parameter types involve functions of one or two independent

variables, which are specified graphically, and data tables, which are specified via a spreadsheet-like interface. The parameters created in the editor are: Probability Density Function (PDF), Packet Formats, Antenna Patterns, and Modulation function [40].

A network model is created inside the Project Editor and is used for defining the topology of a communication network. It may contain some communicating entities such as nodes and links.

Communication devices which are used in the created network model and interconnected at the network level are defined in the node domain by using the Node Editor. Node models consist of interconnected modules. The connections between the modules allow information to stream between the communication end devices.

Process models, created and edited using the process editor, are used for implementation of the logic flow and behavior of processor and programmable modules.

The aim of defining the model of a network communication topology is to get performance measures of that network topology or observe its behavior. The execution of the simulations and data collection are done in three stages:

Stage One: Defining Data Collection

Researchers need to specify which information should be extracted from the simulation scenario such as application-specific statistics, behavioral characterizations, and sometimes application-specific visualization.

Stage Two: Simulation Construction

OPNET simulations are obtained by executing a simulation scenario, which is an executable file in the host computer's file system.

Stage Three: Execution

Simulation execution is the last stage of a modeling experiment. In general, based on the results observed during this stage, changes are made to the model's specification or to the probes, and additional simulations are executed. OPNET provides a number of options for running simulations, including internal and external execution, and the ability to configure attributes that affect the simulation's behavior.

4.2 Network Design and Implementation for IPv4/IPv6 Dual Stack

In this experiment, we aim at testing the Internet protocol IPv4 and IPv6 dual-stack mechanism that is widely used. Though its simplicity, it represents a typical case of dual-stack layout that is the most common layout. This topology represents an abstract view for an enterprise campus having three network segments: a company headquarter supporting both IPv4 and IPv6 (dual-stack), a company branch supporting only IPv4 and another company branch supporting only IPv6.

The network is implemented using different network entities such as routers, workstations and servers that we get them from the object palette. Figure 4-1 shows the simulation topology that we use in this experiment. No special configuration is required for IPv4 and IPv6. We have configured the network just by assigning IPv4 addresses for IPv4 network interfaces and IPv6 addresses for IPv6 network interfaces.

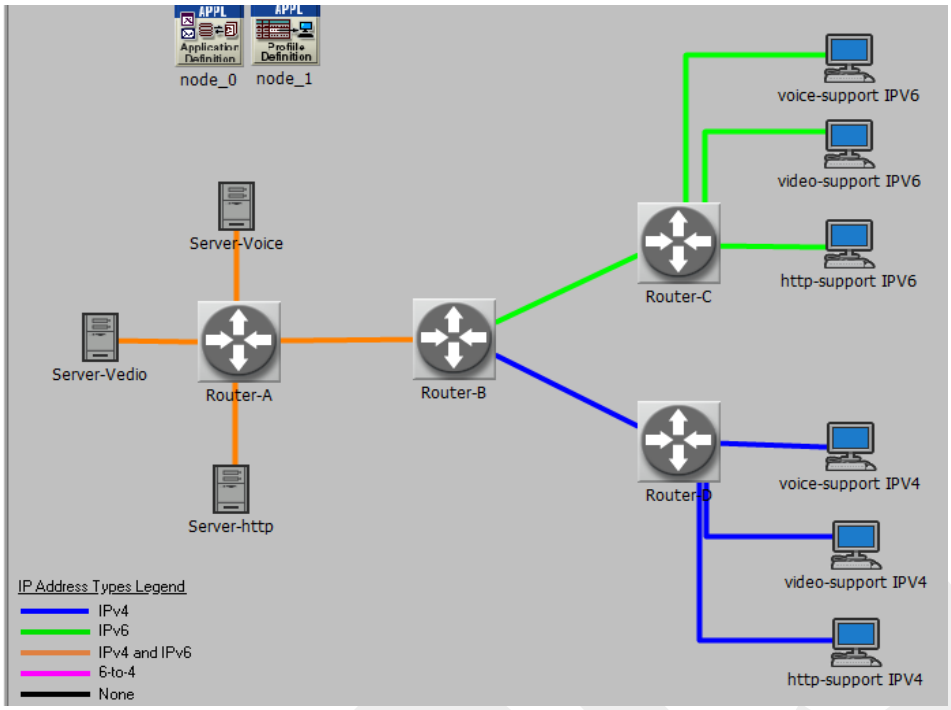

Figure 4.1: Network Dual Stack Topology

In order to create our simulation scenario we have done the following configuration:

- **Workstation Configuration**

The workstation can be configured by right clicking on the workstation icon and selecting "edit attributes." Extend "IP" and then "IP Host Parameters." IPv4 is assigned by putting its IP address in "Address" as shown in Figure 4.2 and then putting the mask in "subnet" mask. This configuration should be applied to all workstations in the topology.

- **Router Configuration**

The router settings are edited by clicking on "IP Routing Parameter" and then clicking on "Interface Information." There, IP addresses for IPv4 can be assigned to related interfaces. In a similar way, IPv6 addresses may also be added by clicking on "IPv6 Parameter". The number of ports depends on the number of connections. Change the "not active" to "Default EUI-64" and assign the IPv6 address to the

interfaces as shown in Figure 4.3. All hosts are connected to the edge routers by 100Base-T cables (100 Mb/s), routers are connected to each other by PPP_DS3 cable (45 Mb/s).

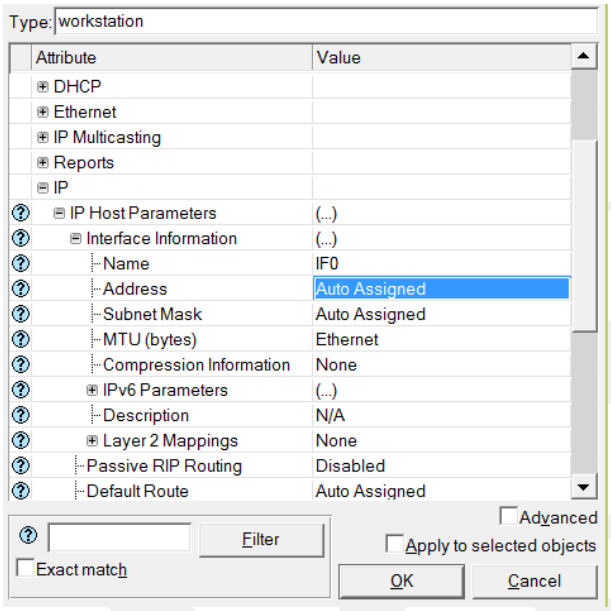

Figure 4.2: IPv4 Address Assignment for Workstations

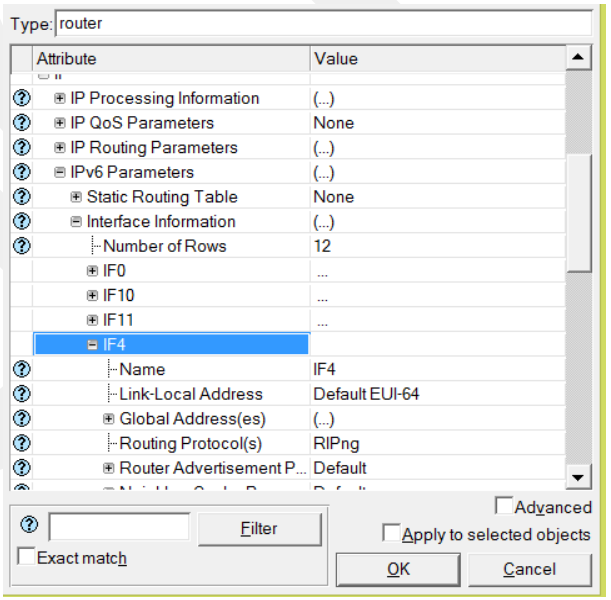

Figure 4.3: IPv6 Address Assignment for Routers

- **Application Configuration**

There are two important network elements that should be configured to establish the traffic when designing the network. The first one is the **"Application Config"** object which is used to configure profiles. Therefore, we must create application using the "Application Config" object before using it. We can specify the traffic patterns followed by the application as well as the configured profiles on the object by selecting it from the “Object Palette” and right clicking to extend the “Application Definitions”. From the “Number of Rows” select the number of applications that will be applied for the network as shown in Figure 4.4.

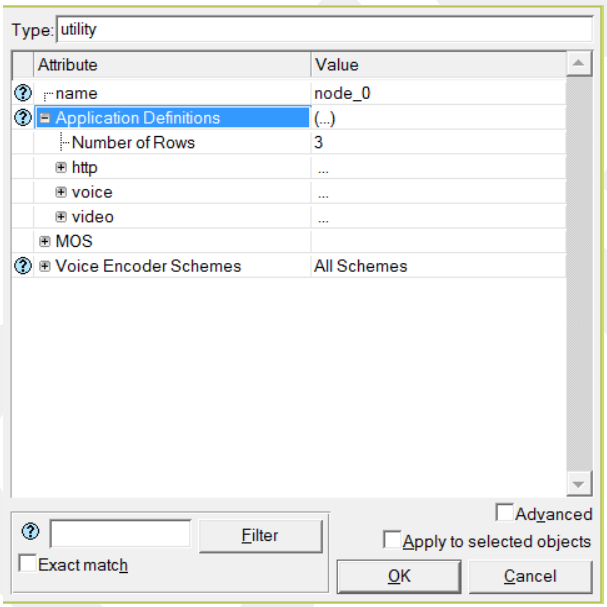

Figure 4.4: Application Configuration

The service types are assigned in “Application Definitions” and this information is moved back to the server. The server can then decide which services will be applied to the network. The “Description” field gives the choice of service type; for this experiment the choices are video conferencing, HTTP and voice application. The second network element is the "**Profile Config**" object which is used to create user profiles. These user profiles can then be specified on different nodes in the network to generate application layer traffic as shown in Figure 4.5.

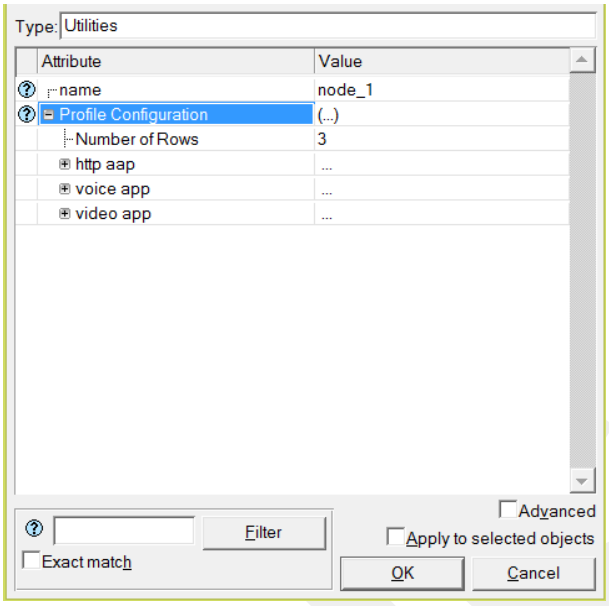

Figure 4.5: Profile Configuration

4.3 Performance Metrics Used for Analyses

The performance metrics that we use to evaluate the dual stack network in this simulation scenario are:

1. **Throughput:** It is defined as the average data transferred across the communication link per unit time. In our simulation scenario, throughput is calculated between different router pairs.
2. **Traffic Received:** It shows the average bytes per second received by a client.
3. **Traffic Send:** It is the average number of bytes sent by the server to a client.
4. **Jitter:** Jitter is the variation in time between packets arriving, caused by network congestion, timing drift, or route changes.
5. **End-to-End Delay:** It gives the end-to-end packet delay between devices.

4.4 Applications Used in the Simulation Scenario

There are many Internet applications to be tested. Due to hardness of considering all Internet applications, we have selected some applications which are popular among the users today. Selected applications are:

1. Video conferencing
2. Voice
3. HTTP

4.5 Results and Discussion

Now, let us proceed with a discussion of the simulation results, which are based on the OPNET simulation. Graphs are used to illustrate the obtained results. The total simulation runtime is adjusted as 30 minutes. Figure 4.6 shows settings for duration of the network simulation.

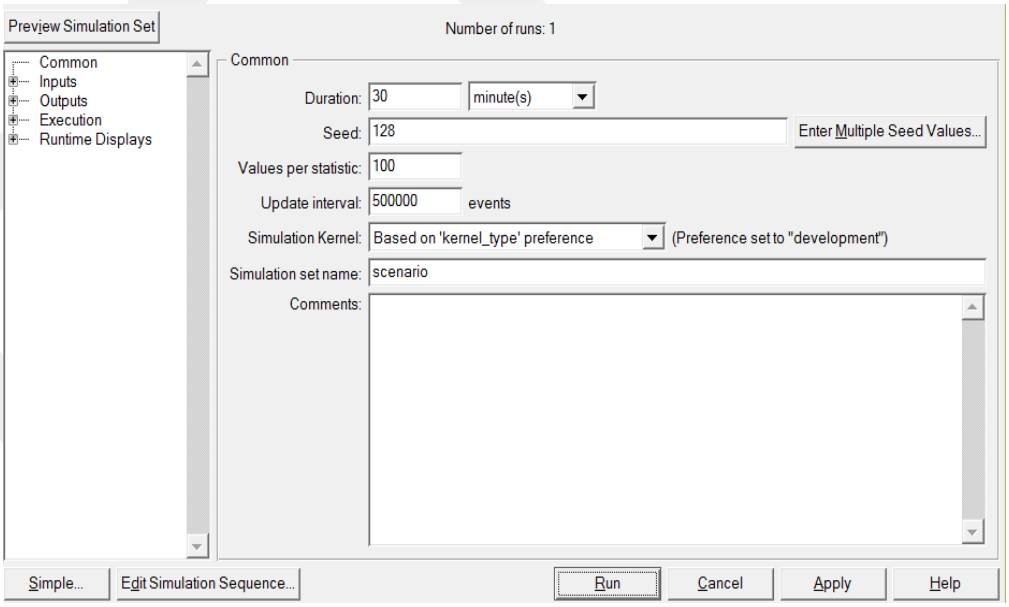

Figure 4.6: The Implementation Period of the Network Simulation

Result 1: End-to-End delay for Video Application

Figure 4.7 shows end-to-end delay for video application on a network configured for Dual Stack Transition Mechanism (DSTM) for IPv6 and IPv4.

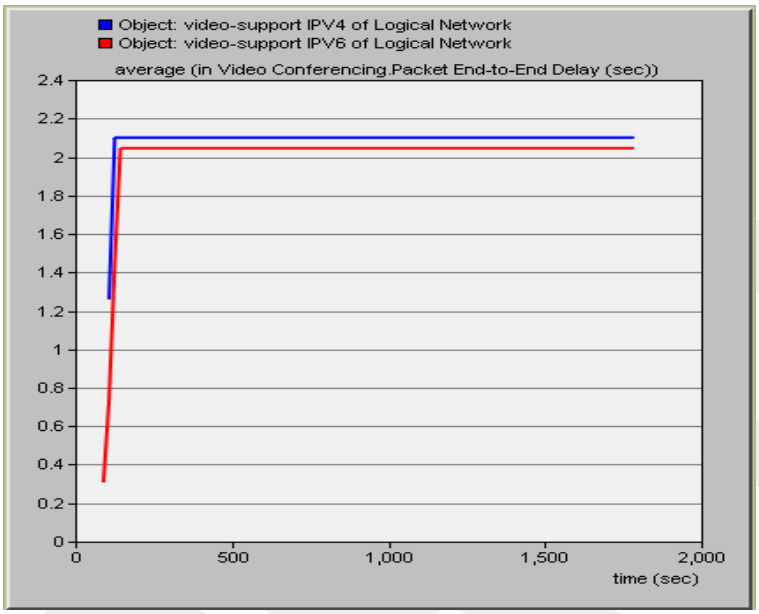

Figure 4.7: Average End-to-End Delay for Video Application

As seen in the figure, we can observe that the amount of end-to-end delay of the video application using IPv4 is higher than the one using IPv6. Although the difference is not very high, it is clear and considerable. The reason is that the IPv6 protocol header has been designed to support fast processing in the routers.

Result 2: End-to-End Delay for Voice Application

We present the end-to-end delay for voice application in Figure 4.8. From the figure, we can observe that the end-to-end delay of the voice application using IPv6 is almost equal to IPv4.

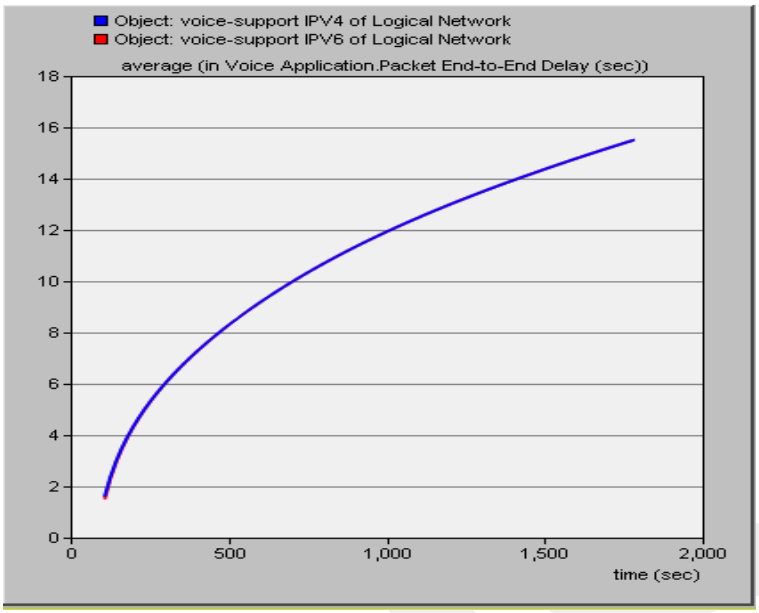

Figure 4.8: Average End-to-End Delay for Voice Application

Result 3: Averages of Video Traffic Sent and Received in IPv4 and IPv6

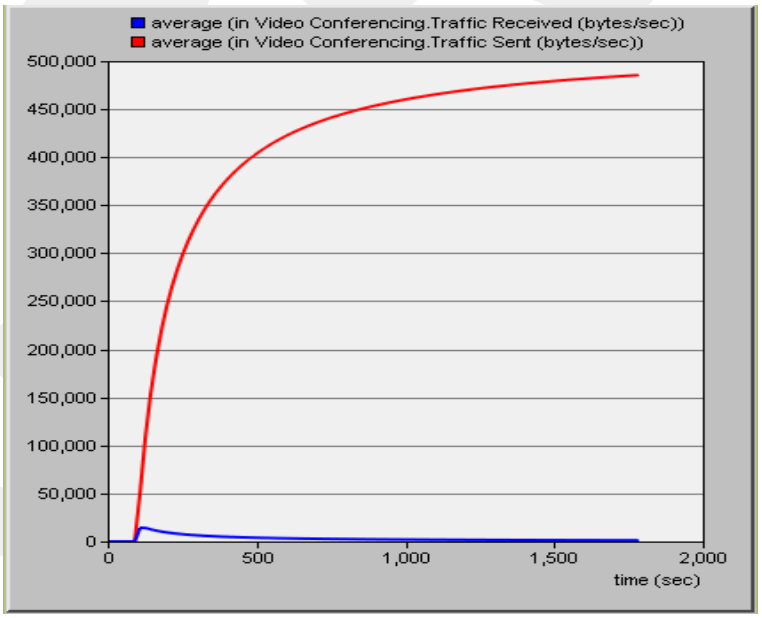

Figure 4.9: Sent and Received Video Traffic in IPv4

Figure 4.9 shows the amount of sent and received traffic for video application over IPv4 networks and Figure 4.10 shows the amount of sent and received traffic for video application over IPv6 networks. To make it more clear, obtained results are compared in Table 4.1. We observe that the amounts of sent traffic for both protocols are almost same. However, IPv6 environment has higher traffic received than IPv4 environment. In other words, we have more packet loss in the IPv4 network. The reason for this result is that, IPv6 modifications resolve the signaling redundancies and can transfer more traffic compared to IPv4.

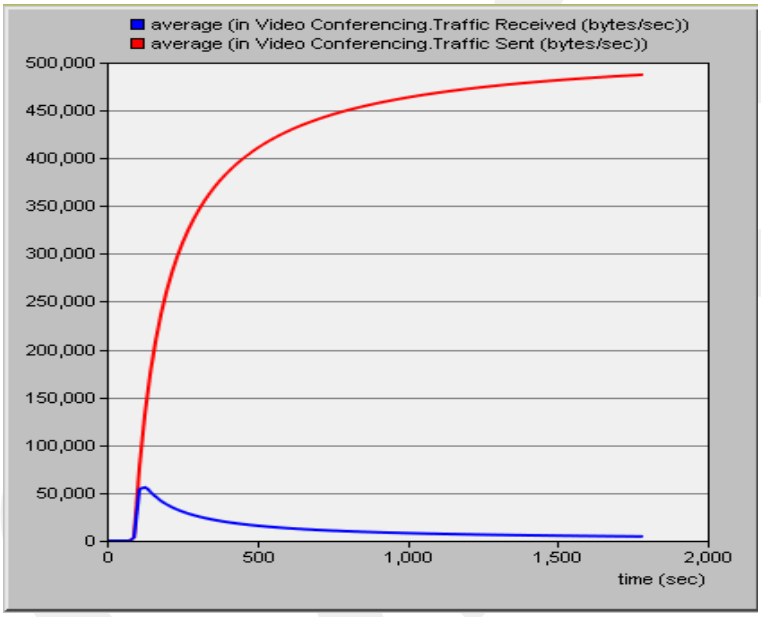

Figure 4.10: Sent and Received Video Traffic in IPv6

Table 4.1: Traffic Sent and Received in Video Application

Application	Parameter	IPv6	IPv4
Video Conferencing	Traffic Sent (bytes/sec)	487,315	485,395
	Traffic Received (bytes/sec)	4,512	1,113

Result 4: Averages of Voice Traffic Sent and Received in IPv4 and IPv6

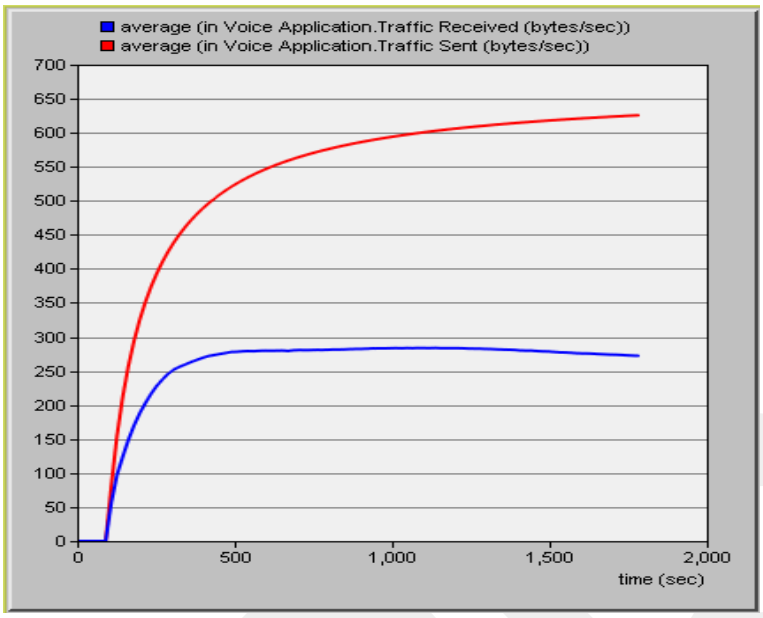

Figure 4.11: Sent and Received Voice Traffic in IPv4

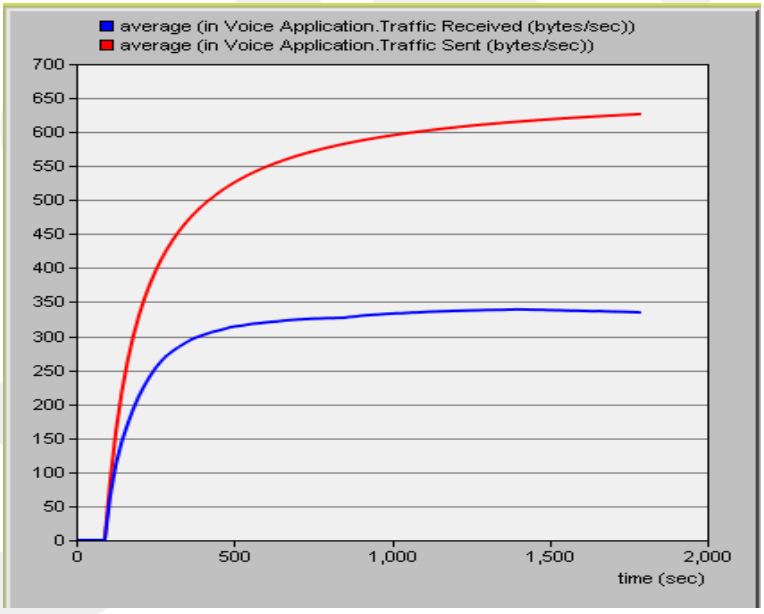

Figure 4.12: Sent and Received Voice Traffic in IPv6

Figure 4.11 and Figure 4.12 shows the results of sent and received voice traffic in IPv4 and IPv6, respectively. We can observe that the amount of received traffic of

voice application in the IPv6 network is more than the one in the IPv4 network. Here, we see a similar result with the video application given above.

.

Result 5: Jitter for Voice Application

Figure 4.13 shows the results for jitter of voice application for IPv6 and IPv4. From the figure, we can observe that the amount of jitter of voice application in the IPv4 network is higher than the one in the IPv6 network. The reason of this result is that the flow label field in IPv6 packet header allows the routers have direct access to the flow label and don't have to guess or find the flow using the transport and application data [41].

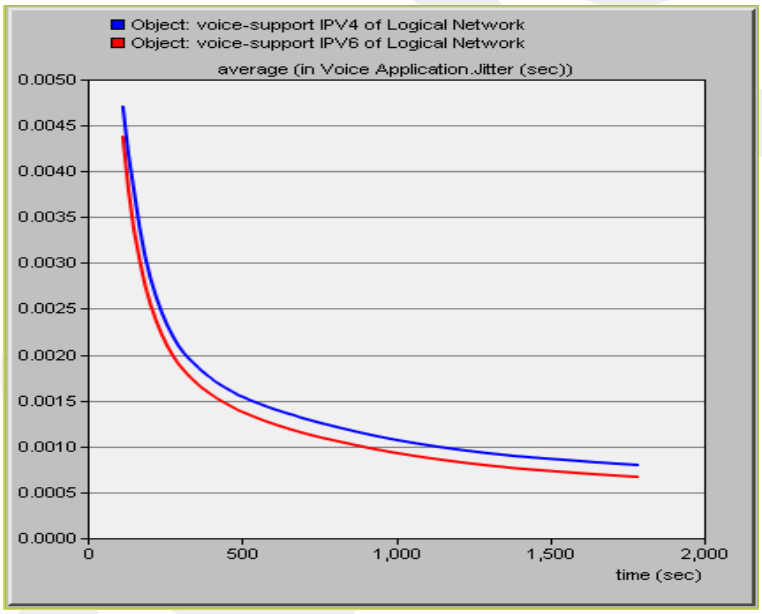

Figure 4.13: Jitter

Result 6: Average TCP Delay

Figure 4.14 shows the results of average TCP delay of the HTTP applications for IPv6 and IPv4. As seen in the figure, the TCP delay for HTTP application using IPv4 is higher than the one using IPv6. Although the difference is not very high, it is clear and considerable. The reason is that the IPv6 protocol header has been designed to support fast processing in the routers.

.

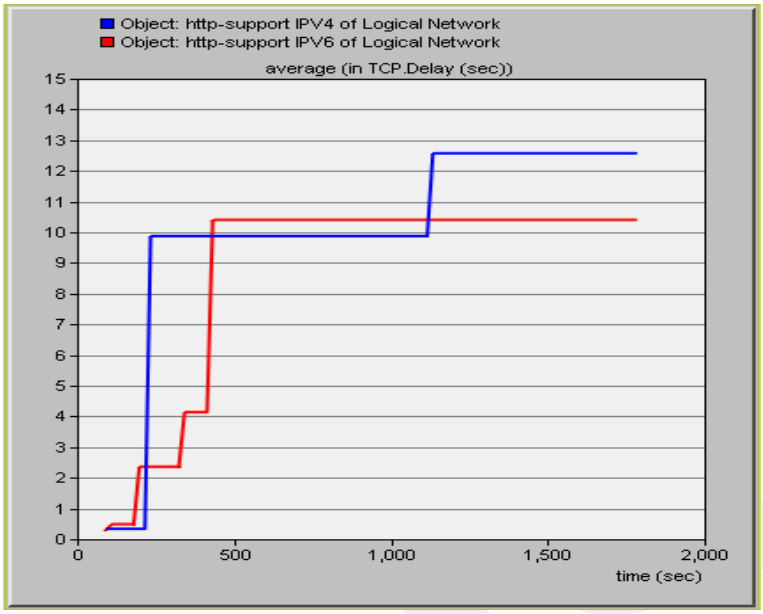

Figure 4.14: Average TCP Delay

Result 7: Averages of HTTP Traffic Sent and Received in IPv4 and IPv6

Figure 4.15 shows the amount of received traffic for the HTTP application over the IPv4 network and received traffic for the HTTP application over the IPv6 network**.** We can observe that the amount of traffic received for the HTTP application running on the IPv6 network is a little bit higher than one running on the IPv4 network. With this result, we observe that for all applications IPv6 provides more traffic received. On the other hand, it means that we have less packet loss in the IPv6 network.

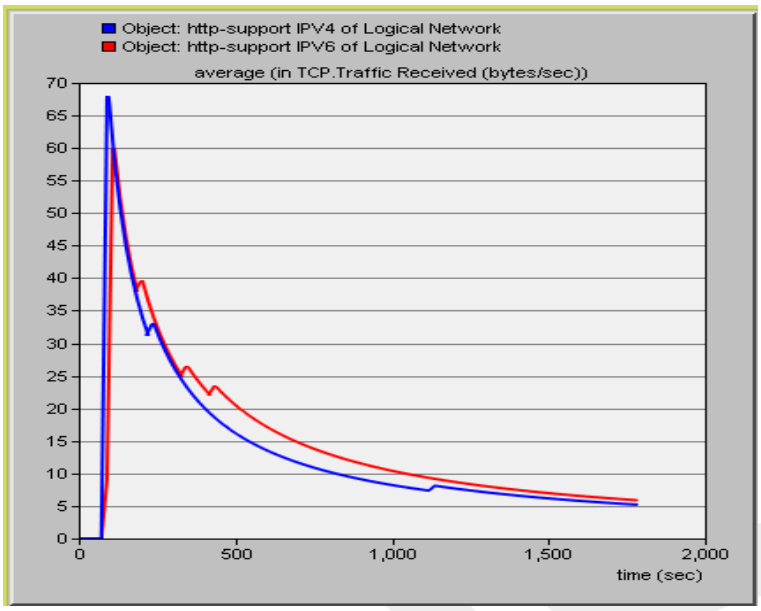

Figure 4.15: Received TCP Traffic in IPv4 and IPv6

Result 8: Averages of Throughput in IPv4 and IPv6 links

Figure 4.16 shows the throughput results for the link between Router-B and Router-C (see Figure 4.1) in both IPv4 and IPv6. From the figure, we observe that IPv6 has higher throughput than IPv4. This is obvious because IPv6 has bigger header size.

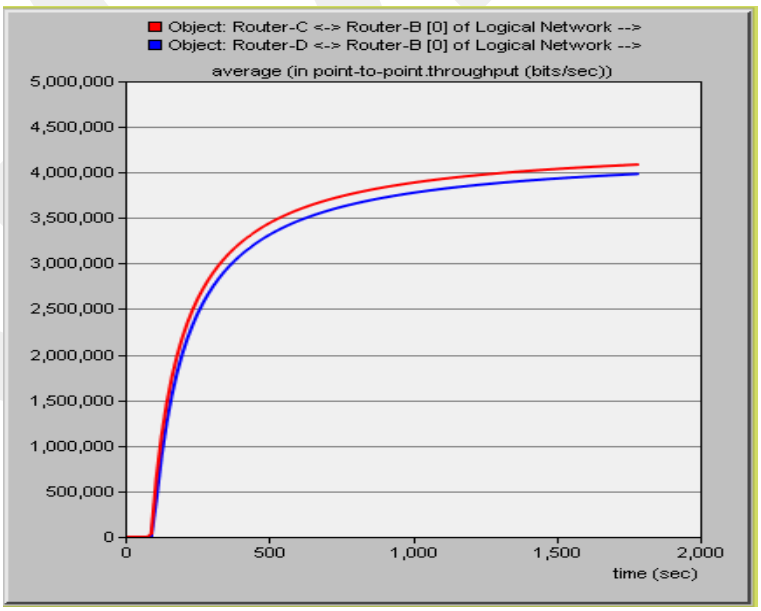

Figure 4.16: Throughput in Point-to-Point Links in IPv4 and IPv6

CHAPTER 5

DESIGN AND EVALUATION OF AUTOMATIC 6TO4 AND MANUAL 6IN4 TRANSITION MECHANISMS

In this chapter, we test two popular transition mechanisms which are used for transitioning from the Internet Protocol IPv4 to the Internet Protocol IPv6. In addition to these two transition mechanisms, we also test native IPv4 and native IPv6 networks for comparison. Therefore, we analyze the following four scenarios in this chapter:

A. Automatic 6to4 Tunneling
B. Manual 6in4 Tunneling
C. Native IPv4
D. Native IPv6

5.1 Network Design and Implementation

The devices that we use in these experiments are workstations, servers, routers, IP backbones, and 100BaseT cables (100Mb/s) and PPP_DS3 cable (45 Mb/s). The experiments are modeled using the OPNET simulator. The simulation topology used for Automatic 6to4 tunneling network and the topology used for manual 6in4 tunneling are shown in Figure 5.1 and Figure 5.2, respectively. The same topology is adapted for native IPv4 as seen in Figure 5.3 and native IPv6 as seen in Figure 5.4.

In Automatic 6to4 and Manual 6in4, workstations use the IPv6 protocol and try to communicate with servers using the same Internet protocol. The communication network in between is IPv4-based. For the first scenario, the network is configured for the 6to4 automatic tunneling. For the second scenario, the network is configured for the manual 6in4 tunneling between routers (Router-A and Router-C) which are dual-stack routers. The workstation named ‘PC-1’ is connected to Router-A using a

100baseT link. The link between Router-C and Server is also 100baseT. The routers are connected to each other by IPv4 backbone using PPP_DS3 links.

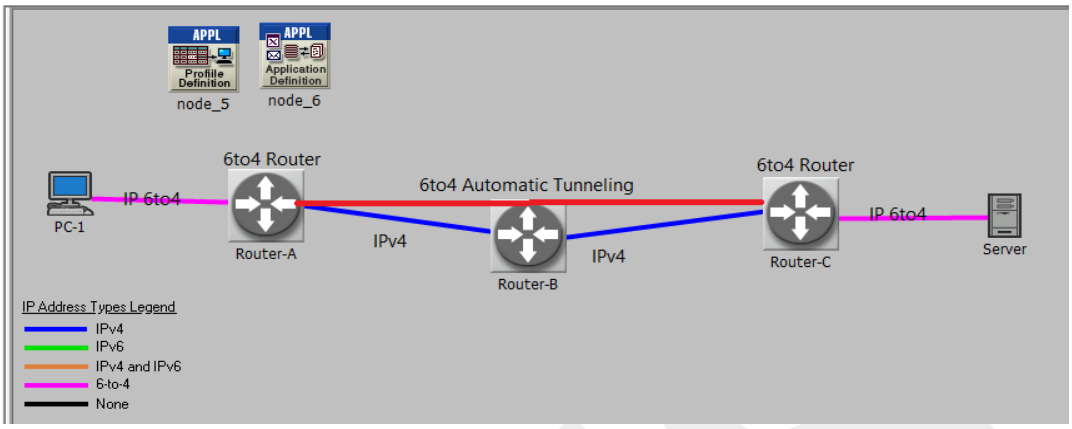

Figure 5.1: Automatic 6to4 Tunneling Topology

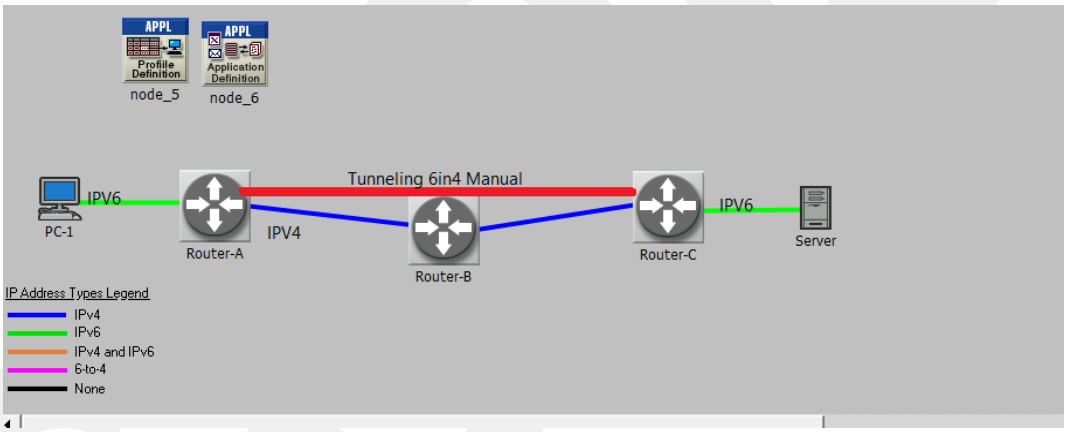

Figure 5.2: Manual 6in4 Tunneling Topology

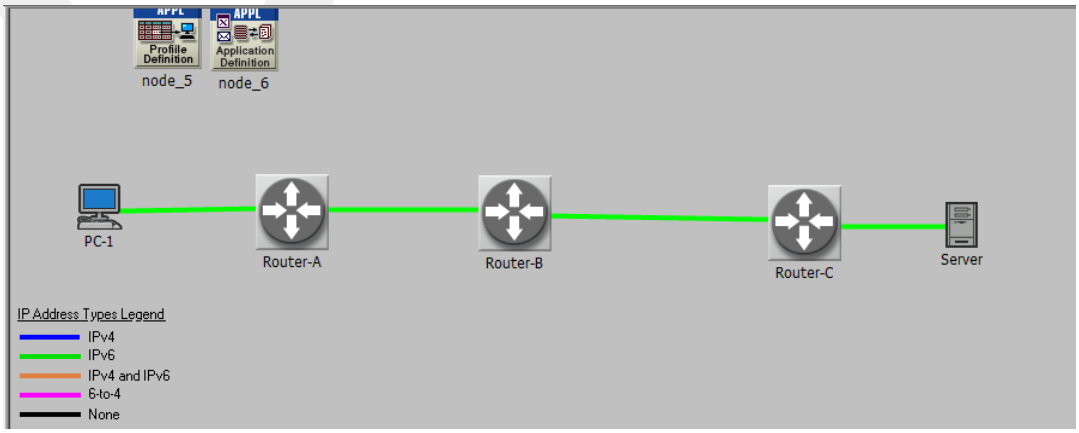

Figure 5.3: IPv6 Only Topology

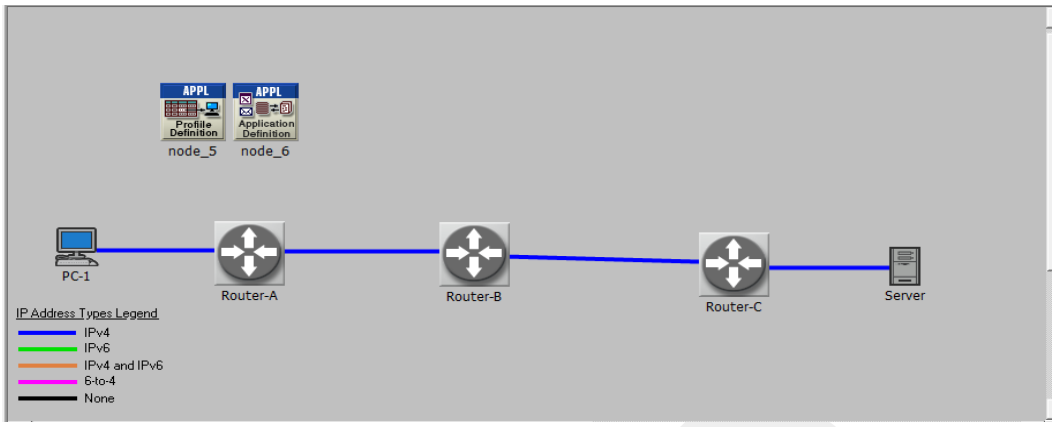

Figure 5.4: IPv4 Only Topology

5.2 OPNET Configurations of Simulation Scenarios

5.2.1 Configuration of Automatic 6to4 Tunneling

Figure 5.5 shows the list of IPv6 Hosts Interfaces. The client and server interfaces are configured as IPv6 interfaces using special address formatting. In addition to the configuration of the 6to4 tunnel, we must manually configure all IPv6 nodes behind the tunnel routers with an address that starts with 2002 and embeds the public IPv4 address inside the IPv6 address.

.

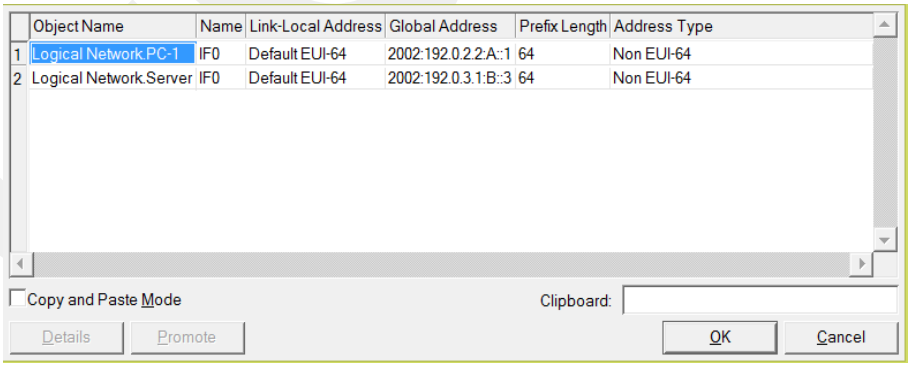

Figure 5.5: IPv6 Hosts Interfaces

Figure 5.6 shows an example for router interfaces. Router-A is configured with a 6to4 address which starts with 2002 and contain IPv4 address within the IPv6 address.

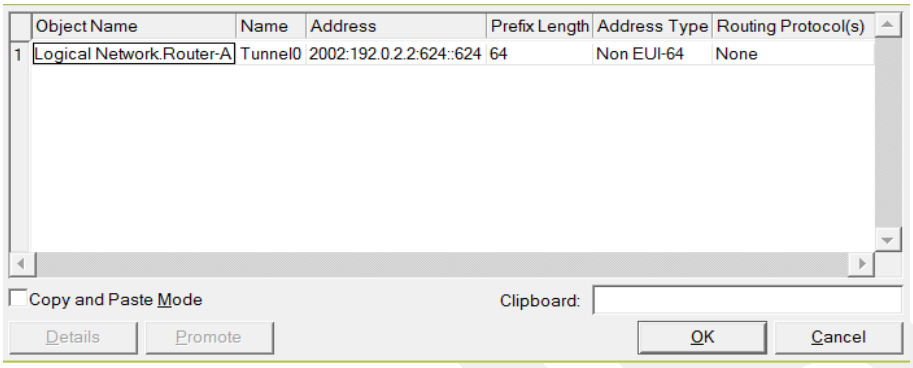

Figure 5.6: IPv6 Routers Interface

We define a tunnel between Router-A and Router-C. Figure 5.7 shows basic tunnel configuration window and Figure 5.8 shows tunnel information table configuration for routers.

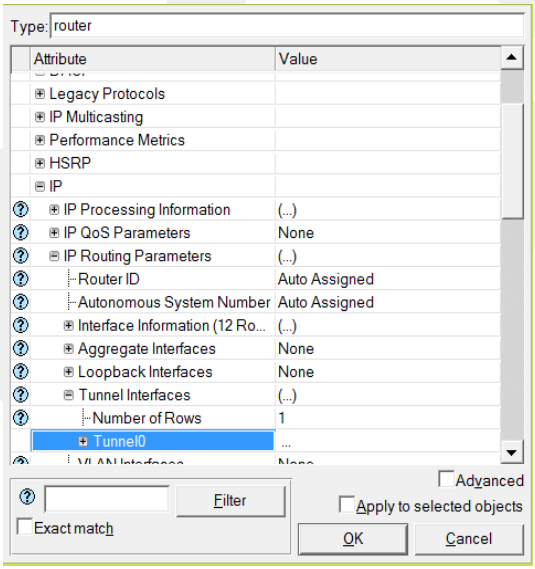

Figure 5.7: Basic Tunnel Configuration

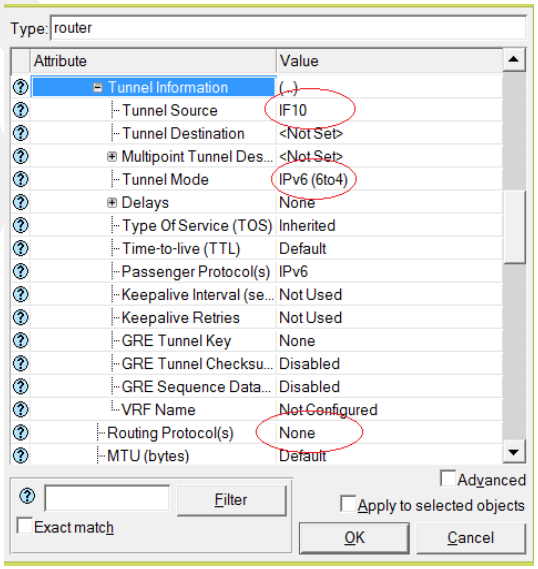

Figure 5.8: Tunnel Information Table

We need to configure IPv6 address of tunnel interface in order to get a correct topology working properly. Global address table must provide the special address which embeds IPv4 address inside the IPv6 address. Figure 5.9 and Figure 5.10 show IPv6 parameters and global address, respectively.

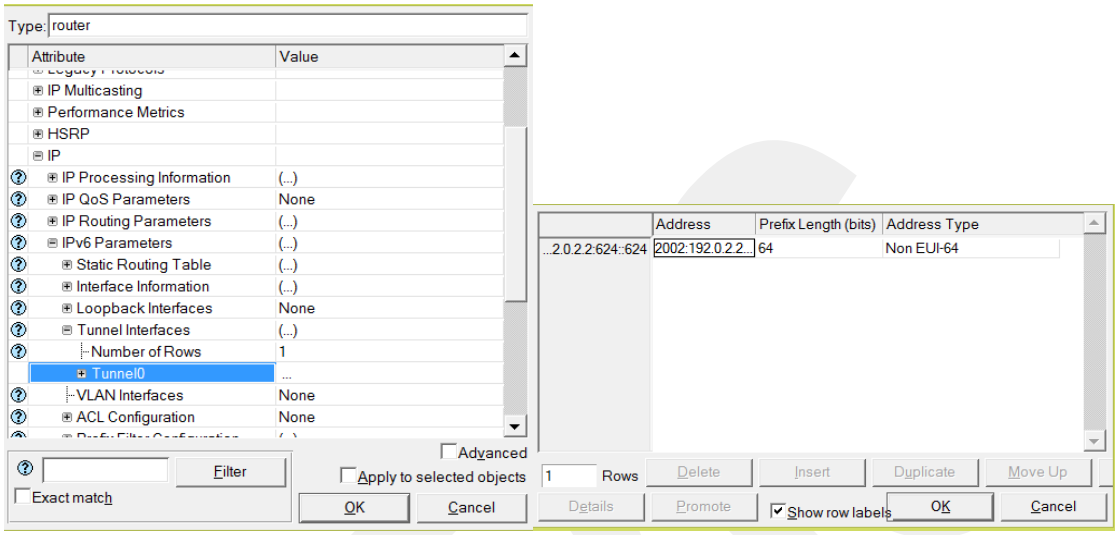

Figure 5.9: IPv6 Parameters for Tunnel Interface **Figure 5.10:** Global Address Table

Routers must process and forward packets correctly. To achieve this, Router-A and Router-C must be configured to send traffic using a default statistic route. Figure 5.11 shows configuration of IPv6 static routing and Figure 5.12 shows IPv6 routing table.

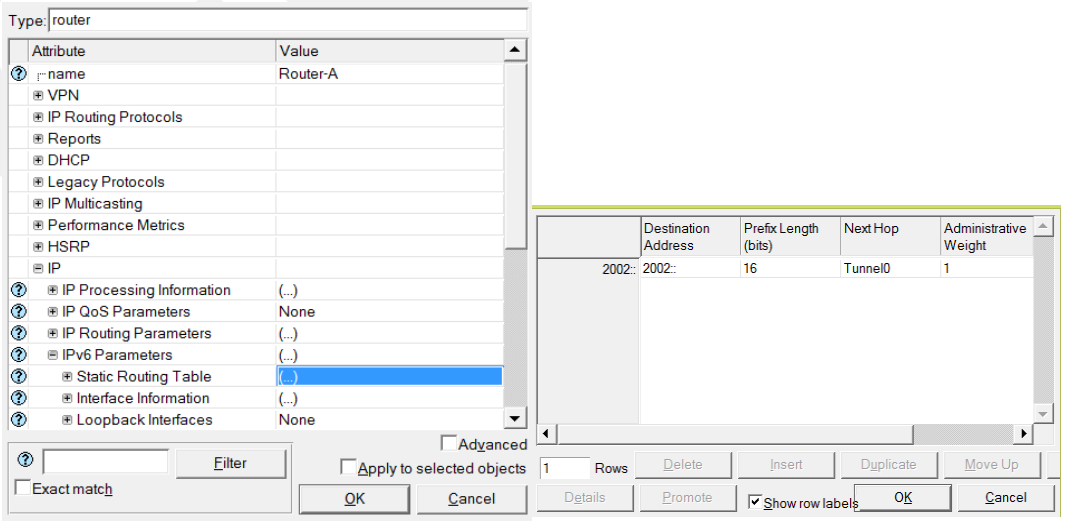

Figure 5.11: Configuration of IPv6 Route **Figure 5.12:** Routing Table of IPv6

5.2.2 Configuration of Manual 6in4 Network

Figure 5.13 shows the configuration of IPv4 which is generated manually on both tunnel end devices. Tunnel destination must be set to the other tunnel end using IPv4 address.

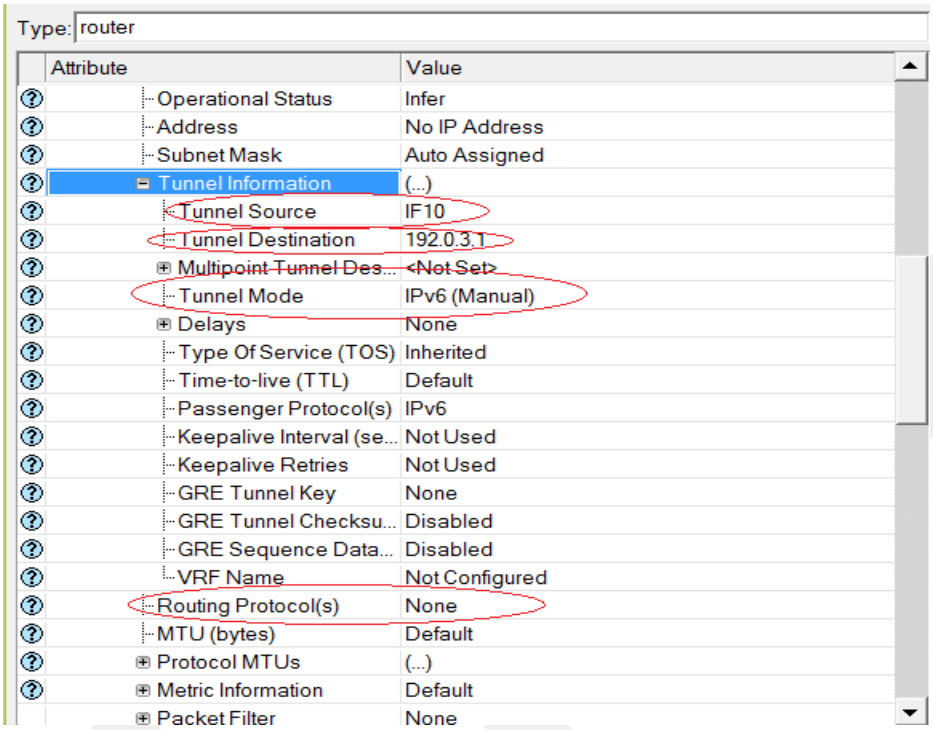

Figure 5.13: IPv4 Manuel Tunneling Configuration

IPv4 is used as the link layer of IPv6 as shown in Figure 5.14. IPv6 packets are encapsulated in IPv4 packets and transmitted to the destination as IPv4 packets.

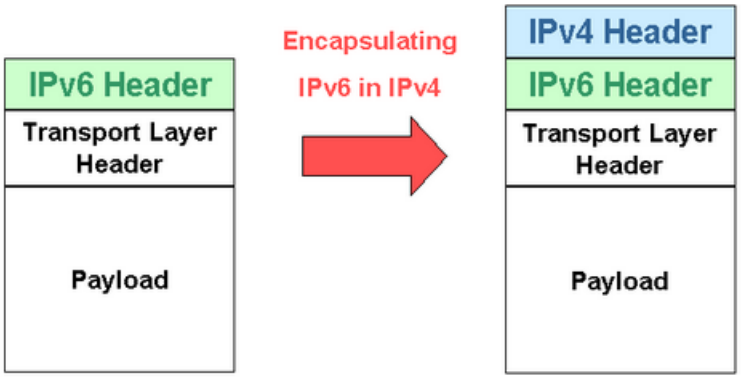

Figure 5.14: IPv6 Packet Encapsulated in IPv4 Packet

Figure 5.15 shows IPv6 tunnel configuration between ends-devices. Routing Information Protocol Next Generation (RIPng) is configured as IPv6 routing protocol. Both end devices have IPv6 addresses, but they are not assigned to the interfaces of routers. This method is called Virtual Link.

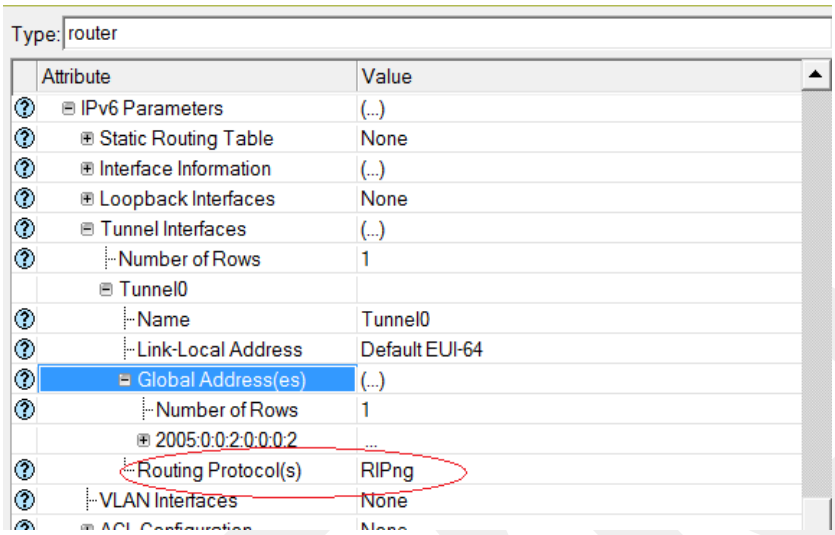

Figure 5.15: IPv6 Manuel Tunneling Configuration

5.2.3 Configurations of Native IPv4 and Native IPv6 Networks

In order to configure native IPv4 and IPv6 networks, configuration is done by assigning IP addresses to all related interfaces. IPv4 address assignment for IPv4 network is done using the options **Protocols > IP > addressing>auto assign IPv4 address.** Similarly, IPv6 address assignment for IPv6 network is done using **Protocols > IPv6 > Enable IPv6 on All Interfaces—enables IPv6.**

5.3 Performance Metrics of Simulation Scenarios

The performance metrics that we are going to evaluate in these simulation scenarios are:

1. **Response Time**: The response time represents the time elapsed between sending a request and receiving the response packet for the application.

2. **Throughput:** The throughput is defined as the average data transferred across the communication link per unit time.

5.4 Applications of Simulation Scenarios

We have selected some applications due to the popularity of these applications among the users today. These applications are:

1. HTTP
2. Database
3. E-mail
4. FTP

5.5 Results and Discussion of Simulation Scenarios

Now, let us proceed with a discussion of the simulation results, which were based on the OPNET simulation. Graphs are used to explain each scenario and discuss its results. The total simulation runtime is 60 minutes. Figure 5.16 shows implementation period of the network simulation.

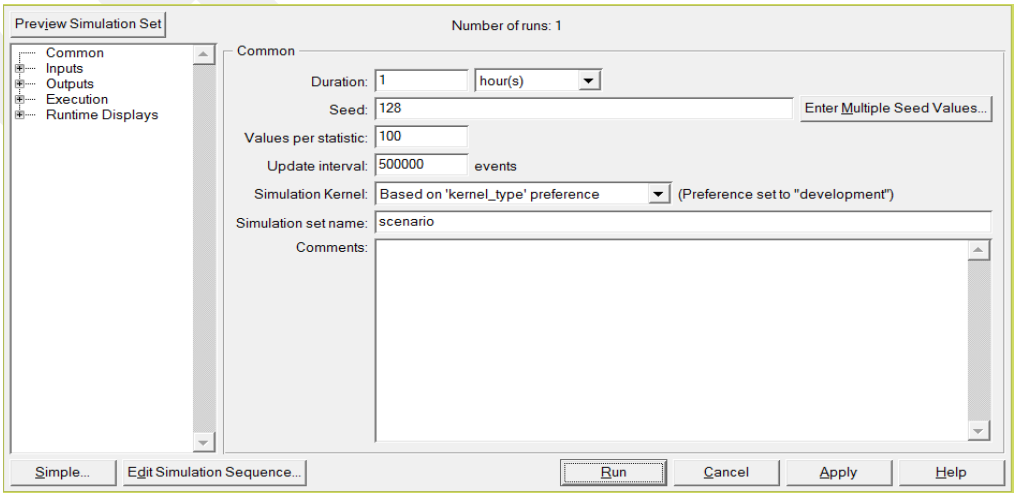

Figure 5.16: The Implementation Period of Network

Result 1: Response Time for HTTP

Figure 5.17 shows the results of response time of HTTP application for the 6to4 tunneling, 6in4 tunneling, native IPv6 and native IPv4 scenarios.

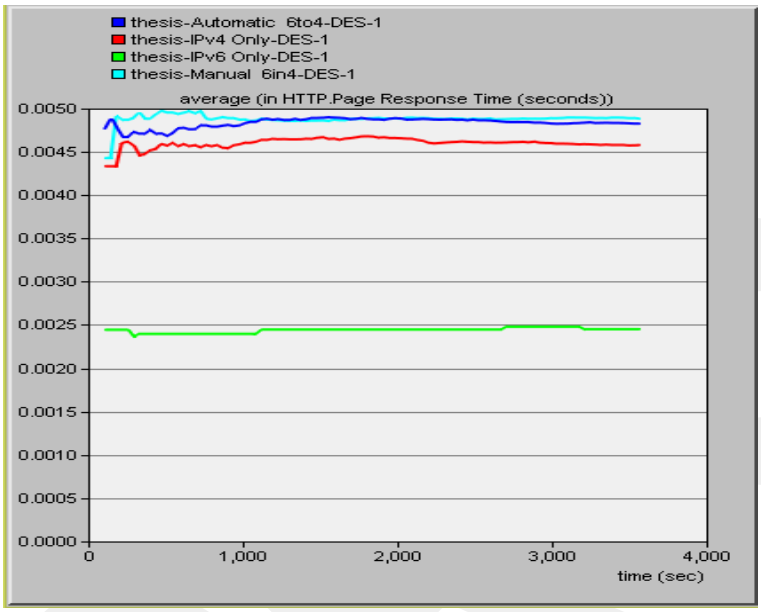

Figure 5.17: HTTP Response Time

As seen in the figure, we observe that native IPv4 and native IPv6 have less response time than 6to4 and 6in4 for HTTP application. Because we do not have encapsulation overhead in native IPv4 and IPv6, they are better than the networks with tunneling mechanism. When we compare native IPv4 and IPv6, IPv6 has better response time than IPv4. The reason is that the IPv6 protocol header has been designed to support fast processing in the routers. Although the difference is not very high, 6in4 has a higher response time than 6to4. The reason is that, 6in4 must update its configuration data regularly in addition to encapsulation and de-capsulation. This process cause some extra delay for 6in4.

The Researchers in [3] investigate three different transmission mechanisms which are Automatic 6to4, Manual 6in4 and Dual-stack and compare them with native IPv6. The results that we obtain in this study are in line with the results given in [3].

Result 2: Throughput for HTTP

Figure 5.18 shows the results of throughput of HTTP application for the 6to4 tunneling, 6in4 tunneling, native IPv6 and native IPv4 scenarios.

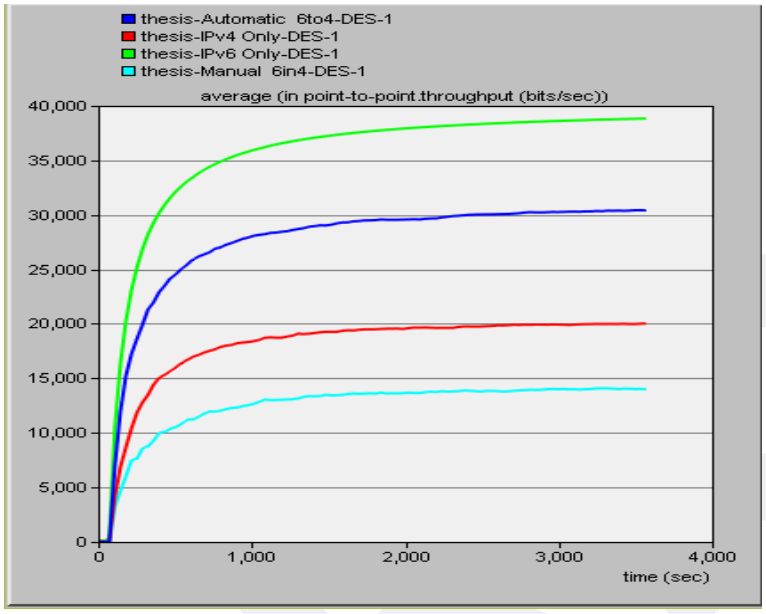

Figure 5.18: HTTP Throughput

As seen in the figure, we observe that native IPv6 network has more throughput than the others for HTTP application.

The Researchers in [3] investigate three different transmission mechanisms which are Automatic 6to4, Manual 6in4 and Dual-stack and compare them with native IPv6. The results that we obtain in this study are in line with the results given in [3].

Result 3: Response Time for Database Application

Figure 5.19 shows the results of response time of database application for the 6to4 tunneling, 6in4 tunneling, native IPv6 and native IPv4 scenarios.

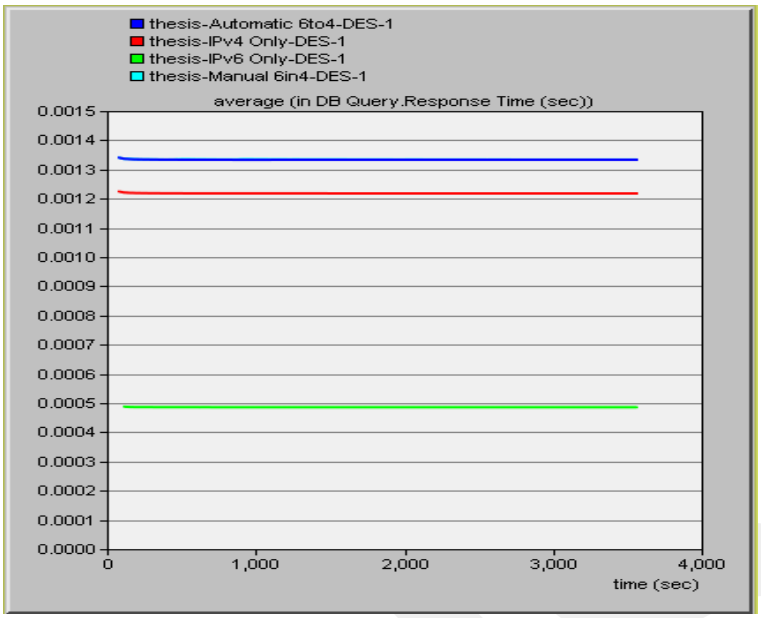

Figure 5.19: Database Application Response Time

As seen in the figure, we observe that native IPv4 and native IPv6 have less response time than 6to4 and 6in4 for database application. Because we do not have encapsulation overhead in native IPv4 and IPv6, they are better than the networks with tunneling mechanism. When we compare native IPv4 and IPv6, IPv6 has better response time than IPv4. The reason is that the IPv6 protocol header has been designed to support fast processing in the routers. Although we see 6in4 and 6to4 on the same line, 6to4 is a little bit better than 6in4 as shown in Table 5.1. The reason is that, 6in4 must update its configuration data regularly in addition to encapsulation and de-capsulation. This process cause some extra delay for 6in4.

The Researchers in [42] investigate one transmission mechanism which is Manual 6in4 and compare it with native IPv4 and native IPv6. The results that we obtain in this study are similar with the results given in [42].

Table 5.1: Average Database Application Response Time

Application	Parameter	IPv6	IPv4	6to4	6in4
Database Application	Response Time (sec)	0.0004855	0.0012179	0.0013333	0.0013338

Result 4: Throughput for Database Application

Figure 5.20 shows the results of throughput of database application for the 6to4 tunneling, 6in4 tunneling, native IPv6 and native IPv4 scenarios.

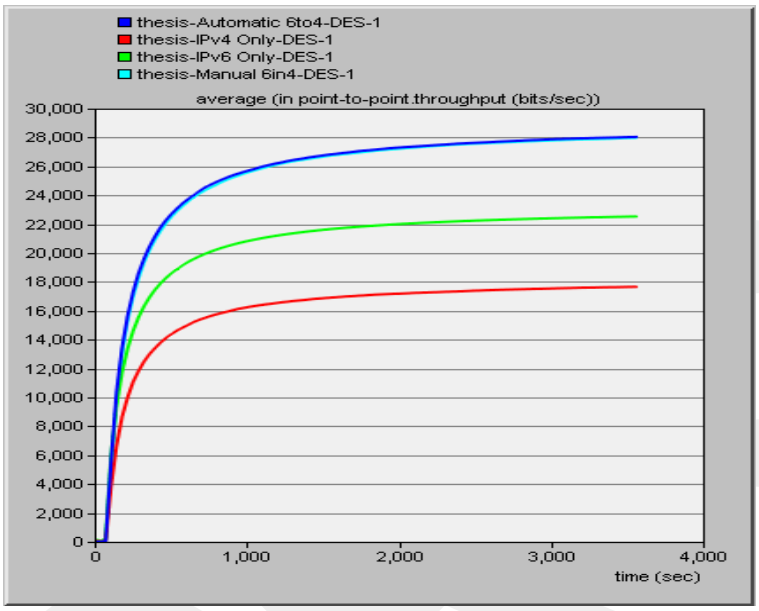

Figure 5.20: Database Application Throughputs

As seen in the figure, we observe that the native IPv6 and native IPv4 have less throughput than the tunneling mechanisms for database application. The reason is that, 6to4 and 6in4 have larger packet sizes because of encapsulation in comparison to the others. Therefore, they transfer more data. we observe that IPv6 has higher throughput than IPv4. This is obvious because IPv6 has bigger header size.

The Researchers in [42] investigate manual 6in4 transmission mechanism and compare it with native IPv4 and native IPv6. The results that we obtain in this study are in line with the results given in [42].

Result 5: Response Time for E-mail Application

Figure 5.21 shows the results of response time of e-mail application for 6to4, 6in4, native IPv6 and native IPv4.

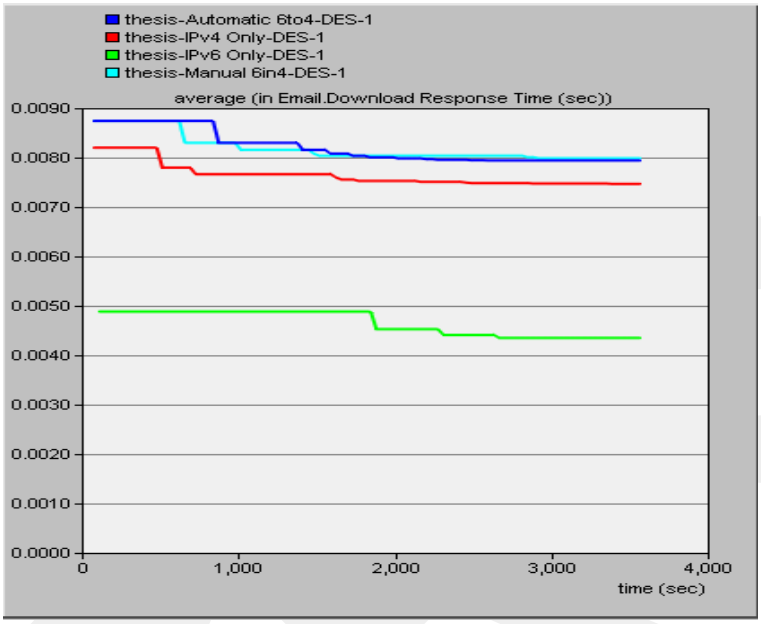

Figure 5.21: E-mail Application Response Time

As seen in the figure, native IPv4 and native IPv6 have less response time than 6to4 and 6in4 for e-mail application. Because we do not have encapsulation overhead in native IPv4 and IPv6 and threfore they are better than the networks with tunneling mechanisms. When we compare native IPv4 and IPv6, IPv6 has better response time than IPv4. The reason is that the IPv6 protocol header has been designed to support fast processing in the routers. Although the difference is not very high, 6in4 has a higher response time than 6to4. The reason is that, 6in4 must update its configuration data regularly in addition to encapsulation and de-capsulation. This process cause some extra delay for 6in4.

Result 6: Throughput for E-mail

We present the results of the simulation experiments in Figure 5.22. The graph shows the results of throughput for e-mail application for the 6to4 tunneling, 6in4 tunneling, native IPv6 and native IPv4 scenarios.

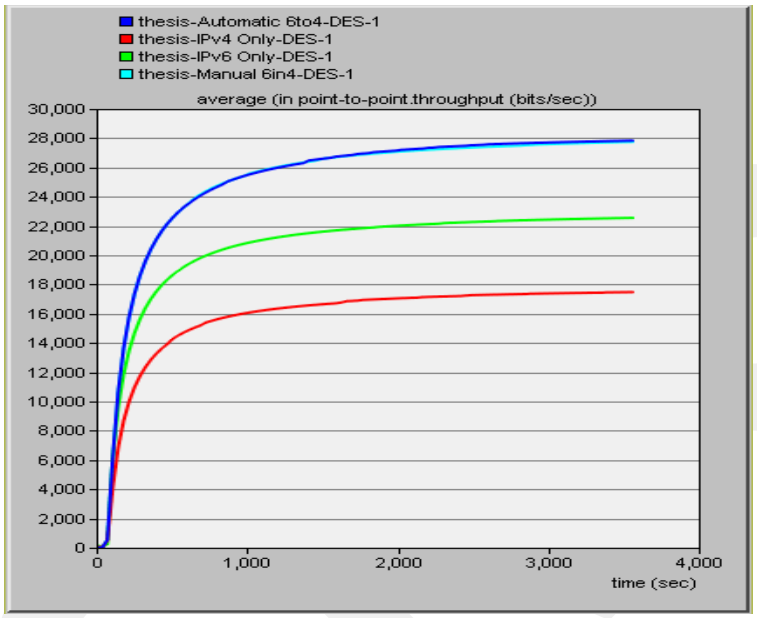

Figure 5.22: E-mail Application Throughputs

We observe that the native IPv6 and native IPv4 have less throughput than the tunneling mechanisms for email application. The reason is that, 6to4 and 6in4 have larger packet sizes because of encapsulation in comparison to the others. Therefore, they transfer more data. we observe that IPv6 has higher throughput than IPv4. This is obvious because IPv6 has bigger header size.

Result 7: Response Time for FTP Application

Figure 5.23 shows the results of response time of FTP application for 6to4, 6in4, native IPv6 and native IPv4.

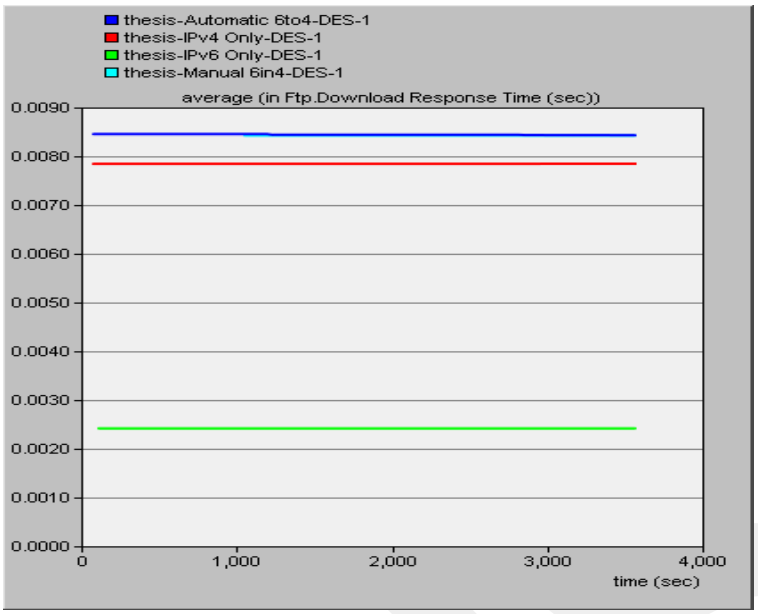

Figure 5.23: FTP Application Response Time

As seen in the figure, native IPv4 and native IPv6 have less response time than 6to4 and 6in4 for FTP application. Because we do not have encapsulation overhead in native IPv4 and IPv6, and therefore they are better than the networks with tunneling mechanism. When we compare native IPv4 and IPv6, IPv6 has better response time than IPv4. The reason is that the IPv6 protocol header has been designed to support fast processing in the routers from IPv4. 6to4 and 6in4 have almost equal response time as detailed in Table 5.2.

Table 5.2: Average FTP Application Response Time

Application	Parameter	IPv6	IPv4	6to4	6in4
FTP Application	Response Time (sec)	0.002416	0.007845	0.008430	0.008424

Result 8: Throughput for FTP Application

Figure 5.24 shows the results of throughput of FTP application for 6to4, 6in4, native IPv6 and native IPv4. As seen in the figure, native IPv6 and native IPv4 have less throughput than the tunneling mechanisms for FTP application. The reason is that, 6to4 and 6in4 have larger packet sizes because of encapsulation in comparison to the others. Therefore, they transfer more data. we observe that IPv6 has higher throughput than IPv4. This is obvious because IPv6 has bigger header size.

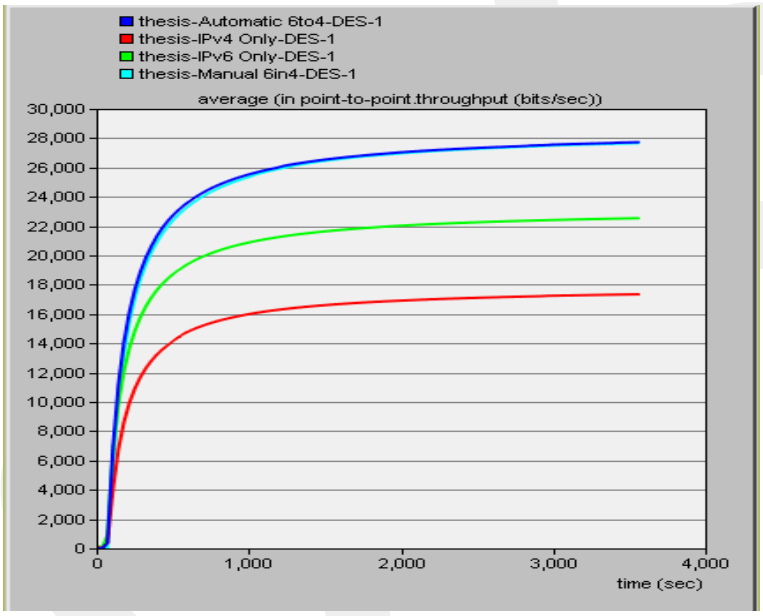

Figure 5.24: FTP Application Throughputs

5.6 Discussion

We observe that native IPv6 has shorter response time than native IPv4, 6to4 and 6in4 for all application. The reason is that the IPv6 protocol header has been designed to support fast processing in the routers.

When we compare two transition mechanisms which are 6to4 and 6in4, 6to4 has a little bit better performance than 6in4. The reason is that, 6in4 must update its

configuration data regularly in addition to encapsulation and de-capsulation. This process causes some extra delay for 6in4.

Finally, if we compare native IPv6 and native IPv4, IPv6 has considerably higher performance than IPv4 in terms of response time since it is processed more quickly by routers. However, it consumes more bandwidth since it has larger header size.

CHAPTER 6

CONCLUSIONS AND FUTURE WORK

6.1 CONCLUSIONS

In this thesis, we examine the present transitioning techniques from IPv4 to IPv6. We have also done some empirical evaluation on the generally utilized transition mechanisms like Automatic 6to4 tunneling, Dual-Stack and Manual 6in4. We have compared the performances of these transition mechanisms with each other and also with the performances of native IPv6 and native IPv4.

According to the outcomes of the thesis, native IPv6 has the minimum delay in comparison to other methods. That means IPv6 has much quicker forwarding and processing ability of the packets than the other methods. Therefore, converting IPv4 networks to IPv6 is a good solution for the future of the Internet and this conversion should be done as soon as possible.

When we consider dual-stack mechanism, in which both IPv4 and IPv6 are running, applications can use either IPv4 or IPv6. However, experimental results show that using IPv6 is more advantageous than using IPv4 considering the performance results. Therefore, applications requiring IPv4 protocol at the network layer should be modified and be able to use the IPv6 protocol.

If we compare two tunneling methods tested in this thesis, automatic 6to4 tunneling has better performance compared to manual 6in4 tunneling. If it requires connecting two separate IPv6 networks over an IPv4 network, then automatic 6to4 tunneling can be preferred to manual 6in4 tunneling.

When we compare native IPv4 and native IPv6 networks with the networks having tunneling mechanism, native ones have better performances compared to others. This is because of the tunneling encapsulation/de-capsulation delays.

6.2 FUTURE WORK

The work presented here is based upon simulation environment on a single computer. However, further work is possible in a real world network. So to get even better outcomes, utilized real world test can be realized.

There are various other transition mechanisms such as Toredo, ISATAP, etc. These transition mechanisms can be investigated as future work.

There are many different applications running on the Internet. Different applications such as video streaming can be tested on different transition mechanisms.

REFERENCES

[1] F. K. James, W. R. Keith, "Computer networking: a top-down approach featuring the Internet", Book, 6th Edition, pp. 331-400, 2012.

[2] R. E. AlJa'afreh, J. Mellor, I. Awan, "A Comparison between the Tunneling process and Mapping schemes for IPv4/IPv6 Transition", IEEE In Advanced Information Networking and Applications, pp. 601-606, 2009.

[3] J. L. Shah, J. Parvez, "An examination of next generation IP migration techniques", IEEE Constraints and evaluation. In Control, Instrumentation, Communication and Computational Technologies (ICCICCT), pp. 776-781, 2014.

[4] S. Savita, Monalisa, "Comparison analysis of various transition mechanisms from ipv4 to ipv6", International Journal of Engineering and Computer Science ISSN: 2319-7242 Vol. 2, pp. 2006-2011, 2013.

[5] A. Albkerat, B. Issac, "Analysis of IPv6 Transition Technologies", International Journal of Computer Networks & Communications (IJCNC) Vol.6, No. 5, 2014.

[6] Cisco Systems, The ABC of IP Version 6, online: "www.cisco.com/go/abc", accessed on 20.3.2015.

[7] Internet live stats, Online: "http://www.internetlivestats.com/internet-users/", accessed on 1.4.2015.

[8] W. Stallings, "IPv6: the new Internet protocol", IEEE Communications Magazine, pp. 96-108, 1996.

[9] L. Ladid, "IPv6-the next big bail-out: will IPv6 save the internet? ", In Proceedings of the International Conference on Computer Systems and Technologies and Workshop for PhD Students in Computing, pp. 21-27, 2009.

[10] A. Durand, "Deploying ipv6", IEEE Internet Computing, pp. 79-81, 2001.

[11] J. G. Jayanthi, S. A. Rabara, "Next generation internet protocol-Technical realms", IEEE in Computer Science and Information Technology (ICCSIT), Vol. 9, pp. 394-399, 2010.

[12] S. Bradner, A. Mankin, "The Recommendation for the IP Next Generation Protocol", (Internet RFCs RFC 1752), http://www.hjp.at/doc/rfc/rfc1752.html, 1995.

[13] J. J. Amoss, D. Minoli, "Handbook of IPv4 to IPv6 transition", Methodologies for institutional and corporate networks, Book, pp. 5-134, 2007.

[14] M. Degermark, "IP header compression", Online: "http://tools.ietf.org/html/rfc2507.html", 1999.

[15] D.T. Ustundg, "Comperative Routing Performance Analysis of IPv4 and IPv6", Computer Engineering Atilim University, pp.20, 2009.

[16] Microsoft, Tech Net Library, Online: "https://technet.microsoft.com/enus/library/cc759208(v=ws.10).aspx", accessed on 4.4.2015.

[17] Microsoft, Tech Net Library, Online: "https://technet.microsoft.com/en-us/library/bb726995.aspx", access on 5.4.2015.

[18] J. Hanumanthappa, D. H. Manjaiah, "A Study on Comparison and Contrast between IPv6 and IPv4 Feature Sets", 2008.

[19] Cisco, Network Group System, Online: "https://tools.ietf.org/html/rfc3513", accessed on 6.4.2015.

[20] R. A. Talwalkar, "Analysis of Quality of Service (QoS) in WiMAX networks", (M.Sc. Thesis), Florida Atlantic University, Boca Raton, Florida, USA, 2008.

[21] J. K. Byeon, "A Study of the Impact of Traffic Type and Node Mobility on the Performance of an IEEE 802.16 WiMAX", (M.Sc. Thesis), School of Computing and Mathematical Sciences, Auckland University of Technology, New Zealand, 2011.

[22] H.A. Mohammed, H.A. Adnan, J.M. Hawraa, “The Affects of Different Queuing Algorithms within the Router on QoS VoIP Application using OPNET”, International Journal of Computer Networks & Communications, vol. 5, pp. 117-124, 2013.

[23] I. Raicu, S. Zeadally, “Evaluating IPv4 to IPv6 transition mechanisms”, In Telecommunications, Vol. 2, pp. 1091, 2003.

[24] Y. Wu, X. Zhou, “Research on the IPv6 performance analysis based on dual-protocol stack and tunnel transition” International Conference on Computer Science & Education (ICCSE), 3-5 Aug. 2011, Singapore, pp. 1091-1093, 2011.

[25] Online: “http://xuanbo.blog.51cto.com/499334/203557”, accessed on 10.4.2015.

[26] Q. Zheng, T. Liu, X. Guan, Y. Qu, N. Wang, “A new worm exploiting IPv4-IPv6 dual-stack networks”, In Proceedings of the workshop on Recurring malcode, pp. 9-15, 2007.

[27] T. Guan, Y. Xia, “The Building of Campus Network Transition Scheme Based on IPv6 Automatic Tunnel Technology”, IEEE Second Pacific-Asia Conference on Web Mining and Web-based Application, 6-7, June 2009, Wuhan, pp. 109-111, 2009

[28] T. Rooney, “IPv4-to-IPv6 Transition and Co-Existence Strategies”, BT Diamond IP, pp. 5-11, 2011.

[29] CISCO, IPv6 Tunnel through an IPv4 Network, Online: “http://www.cisco.com/c/en/us/support/docs/ip/ip-version-6/25156-ipv6tunnel.html”, accessed on 12.4.2015.

[30] Institute of Telecommunications Networks Group, Basic Transition Mechanisms, Online: “http://www.ibk.tuwien.ac.at/~ipv6/transition.htm”, accessed on 16.4.2015.

[31] R. E. Gilligan, E. Nordmark, “Transition mechanisms for IPv6 hosts and routers”, Transition, online: “https://tools.ietf.org/html/rfc2893”, 2000.

[32] D. G. Waddington, F. Chang, "Realizing the transition to IPv6", Communications Magazin, pp. 143, 2002.

[33] Understading & Configuration IPv6 6to4 Tunnels, Online: "http://www.ebrahma.com/2013/12/understanding-configuring-ipv6-6to4-tunnels/", accessed on 17.4.2015.

[34] Deploying IPv6 over IPv4 Tunnels**,** online, "https://josephmlod.wordpress.com/network/ipv6/deploying-ipv6-over-ipv4-tunnels/", accessed in 18.4.2015.

[35] M. Aazam, A. M. Syed, "Evaluation of 6to4 and ISATAP on a Test LAN", In Computers & Informatics (ISCI), IEEE Symposium on, pp. 46-50, 2011.

[36] V. Visoottiviseth, N. Bureenok, "Performance comparison of ISATAP implementations on FreeBSD, RedHat, and Windows 2003", IEEE 22nd International Conference on, pp. 547-552, 2008.

[37] Tomicki.net, network adrress translation, protocol translation IPv4/IPv6, online: "http://tomicki.net/naptd.php", accessed on 20.4.2015.

[38] X. Chang, "Network simulations with OPNET", InProceedings of the 31st conference on winter simulation: Simulation---a bridge to the future-Vol1, pp. 307-314, 1999.

[39] M. Dixon, T. Koziniec, "Using OPNET to Enhance Student Learning in a Data Communications Course", PP. 350-355, 2002.

[40] Z. Lu, H. Yang, "Unloking the Power of OPNET Modeler", Book, pp. 5-100

[41] F. N. Fatah, A. Suhendra, M. A. Marwan, H. F. H. Firdaus, "Performance Measurements Analysis of Dual Stack IPv4-IPv6", pp. 100-106, 2013.

[42] J. L. Shah, J. Parvez, "Performance evaluation of applications in manual 6in4 tunneling and native IPv6/IPv4 environments", IEEE in Control, Instrumentation, Communication and Computational Technologies (ICCICCT), pp. 782-786, 2014.