COMPERATIVE ROUTING PERFORMANCE ANALYSIS OF

IPv4 AND IPv6

A MASTER'S THESIS

in

Computer Engineering

Atilim University

by

D.TURAN USTUNDAG

JANUARY 2009

COMPERATIVE ROUTING PERFORMANCE ANALYSIS OF

IPv4 AND IPv6

A THESIS SUBMITTED TO

THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

OF

ATILIM UNIVERSITY

BY

D.TURAN USTUNDAG

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF SCIENCE

IN

THE DEPARTMENT OF COMPUTER ENGINEERING

JANUARY 2009

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

____

Prof. Dr. Abdurrahim ÖZGENOĞLU

Director

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

____

Prof. Dr. İbrahim Akman

Head of Department

This is to certify that we have read the thesis "Comperative Routing Performance Analysis of IPv4 and IPv6" submitted by "D.Turan ÜSTÜNDAĞ" and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science.

_______________________	_______________________
Dr.Kasım ÖZTOPRAK	Asst. Prof. Dr. Murat Koyuncu
Co-Supervisor	Supervisor

Examining Committee Members

Asst. Prof. Dr. Murat Koyuncu	_______________________
Asst. Prof. Dr. Nergiz Ercil Çağıltay	_______________________
Asst. Prof. Dr. Atila Bostan	_______________________
Dr.Sedat Nazlıbilek	_______________________
Instructor Ziya Karakaya	_______________________

Date: 27.01.2009

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.

D.Turan ÜSTÜNDAĞ

ABSTRACT

COMPERATIVE ROUTING PERFORMANCE ANALYSIS OF IPv4 AND IPv6

Üstündağ, D.Turan

M.S., Computer Engineering Department

Supervisor: Asst.Prof.Dr.Murat Koyuncu

Co-Supervisor: Dr. Kasım Öztoprak

January 2009, 98 pages

A basic network protocol IP, as described in RFC 791, was developed in 1981. Upon three studies (versions) IP got the name of next version number 4 and was announced as IPv4. Several changes and additions have been made over time to make IP more usable, but the core protocol is basically what it was in 1981 [10]. In 1992 some specialist warned the IETF about the shortages of IPv4 and limitation that makes continued growth of the Internet for decades to come problematic [2]. In order to fix the problem IPv6 proposed by numerous internet groups in 1993. Upon the increment of proposals on IP new generation, the IETF has produced a comprehensive set of specifications that define the next-generation IP protocol known as "IPng" or "IPv6". IPv6 claims more effective, scalable, secure, built-in, routable opportunities over IPv4. As almost every network authority agrees on that a seamless transfer from IPv4 to IPv6 is hard to imagine. On the way of native IPv6 journey, transaction mechanisms might be a temporary choice.

This thesis evaluates these claims from the aspect of routing performance analysis in current, transition period and finally IPv6 native networks. This thesis major focuses include: 1. Significant differences, proposed superiorities and advantages of IPv6 to IPv4. 2. Measurement-based comparison, especially transaction mechanism, of IPv6 – IPv4 routing and IPv6 gains on a simulated test-bed environment (OPNET).

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

ÖZ

IPv4 VE IPv6 PROTOKOLLERİNİN

KARŞILAŞTIRMALI YÖNLENDİRME PERFORMANS ANALİZİ

Üstündağ, D.Turan

Yüksek Lisans, Bilgisayar Mühendisliği Bölümü

Tez Yöneticisi: Yrd.Doç.Dr.Murat KOYUNCU

Yardımcı Tez Yöneticisi: Dr. Kasım ÖZTOPRAK

Ocak 2009, 98 sayfa

Bir ağ iletişim protokolü olan IP, RFC 791'de belirtildiği gibi 1981 yılında geliştirildi. O zamana kadar yapılmış 3 çalışmanın üzerine ortaya konan IP, sürüm olarak 4 rakamını aldı ve IPv4 olarak tanıtıldı. Birçok değişiklik ve ilavelerle daha kullanılabilir hale getirilen IPv4'ün ana yapısı 1980'lerin başındaki halini muhafaza etti. 1992 yılında birtakım uzmanlar IPv4'ün yetersizlikleri ve onyıllar boyu sürekli gelişen internetin ilerlemesinde problemler olacağı konularında IETF'i uyardılar. Çok çeşitli çalışma grupları tarafından IPv6, problemin çözümü olarak önerildi. Taleplerin artması üzerine IETF, IPng veya IPv6 adıyla geniş kapsamlı özellikler içeren, gelecek nesil internet protokolünü geliştirdi. IPv6'nın, IPv4'e nazaran daha etkin, ölçeklenebilir, güvenilir, bütünleşik yapıda barındırma ve yönlendirilebilir imkanlar sunacağı iddia edildi. Bütün ağ otoritelerinin de kabul ettiği gibi IPv4'ten IPv6'ya kusursuz ve kesintisiz bir geçişi düşünmek oldukça zordur. Özgün IPv6 yolunda geçiş mekanizmaları geçici bir tercih olabilir.

Bu tez sözkonusu iddiaları günümüz, geçiş süreci ve nihayetinde özgün IPv6 ağlarında yönlendirme performansı açısından değerlendirmektedir. Tezin odaklandığı 2 ana konu; 1. IPv6'nın IPv4'ten belirgin farklılıkları, öne sürdüğü üstünlükleri ve avantajları, 2. IPv6'nın yönlendirme performansı kazanımlarını, özellikle geçiş sürecinde, deneysel ortamda (OPNET) ortaya koymak olarak belirlenmiştir.

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

To My Parents

&

Family

ACKNOWLEDGMENTS

I would like to thank, my parents for their endless support, guidance and kind words,

To my lovely wife, daughter and son for their patience, love, and continuous support,

To Kazım Çay, for his very critical and remarkable guidance and support for constructing the testbed environment,

To my brother and my friend Nezaket Tezcan,

To my supervisor, co-supervisor and instructors.

Thank you all!

TABLE OF CONTENTS

LIST OF TABLES

TABLE

LIST OF FIGURES

FIGURE

LIST OF ABBREVIATIONS

ACE	Application Characterization Environment
AFRINIC	African Region Internet Registry
APNIC	Asia-Pacific Network Information Centre
ARIN	American Registry for Internet Numbers
BB	Backbone
BGP	Border Gateway Protocol
BSD	Berkeley Software Distribution
CIDR	Classless Inter-Domain Routing
DES	Discrete Event Simulation
DHCP	Dynamic Host Configuration Protocol
DSTM	Dual Stack Transition Mechanism
EUI	Equipment Identifier
FP	Format Prefix
GUI	Graphical User Interface
IAB	Internet Architecture Board
IANA	Internet Assigned Numbers Authority
ICMP	Internet Control Message Protocol
IETF	Internet Engineering Task Force
IGP	Interior Gateway Protocol

ISATAP	Intra-Site Automatic Tunnel Addressing Protocol
LACNIC	Latin American and Caribbean Internet Addresses Registry
MPLS	Multiprotocol Label Switching
MTU	Maximum Transmission Unit
PMTUD	Path MTU Discovery
NAT	Network Address Translator
NLA	Next Level Aggregation
OSI	Open Systems Interconnection
OSPF	Open Shortest Path First
PIM-DM	Protocol Independent Multicast - Dense Mode
QoS	Quality Of Service
RFC	Request for Comments
RIP	Router Information Protocol
RIPE NCC	The Réseaux IP Européens Network Coordination Centre
RIPNG	RIP for IPv6 (RIP Next Generation)
RIR	Regional Internet Registries
SLA	Site-Level Aggregation
STD	State Transition Diagram
TLA	Top Level Aggregation
TOS	Type of Service
VOIP	Voice over IP

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION

Internet Protocol (IP) was developed in 1981 for interconnection of different network technologies. After three versions IP got the name of next version number 4 and declared as IPv4 to the internet community. Several changes and add-ons have been made over time to make IP more usable, but the core protocol is basically what it was in 1981 [10]. The king of miraculous invisible world became very popular and unforeseenable developments in the network technologies began to push IPv4 to the corner. Here, critical point was that if the 32 bits address space be enough for the future world? In 1992 some specialist warned the IETF about the shortages of IPv4 and limitation that makes continued growth of the Internet for decades to come problematic [2].

Some temporary solutions came forth; use of private networks, CIDR, dial-up connections, DHCP, NAT, tighter control by RIRs on the allocation of addresses to local Internet registries, name-based virtual hosting, reclaiming unused IPv4 space. Especially NAT slowed the enermous consumption of the IP blocks, but as every drug that recovers from illness but harms some other parts, NAT prevent IP level end-to-end security, reduce robustness, break a number of application level protocols, complicate the network and inhibit some novel uses of the Internet (e.g., such as peer-to-peer networking) [8].

Besides those ongoing temporary activities, in order to fix the problem and to find a permanent solution, IPv6 was proposed by numerous internet groups in 1993. Upon the increment of proposals on IP new generation, the IETF has produced a comprehensive set of specifications that define the next-generation IP protocol known as "IPng", or "IPv6". IPv6 claims more effective, scalable, secure, built-in, routable opportunities over IPv4.

Basically IPv6 has 2^{96} times more addrress space in a raw manner. Configuration became much simpler. Hierarcical addressing capability, simplified header format and improvements of protocols increase the routing capability. Multicast technology, a not so well used feature of IPv4, becomes one of the basic actors of IPv6 that facilitates bandwith utilization as well. Besides tons of RFCs many more studies are still on the way.

IPv6 get its deserved place day by day. One day in the future probably all internet will be native IPv6. Till that time IPv4 and IPv6 will coexist and transition mechanisms will be in charge. Of course transition to IPv6 needs well planning, small but additive adaptations, hardware and software upgrades, rapidly and effectively coded softwares, a large amount of learning, training, and restructuring. Organisations having budgeting problems or suspicious about IPv6 proposals may try the new world behind the scene by using transition technologies. Those well designed adaptations (tunneling, dual stack and translations mechanisms), facilitate transition.

Much work has been done on IPv6 standards, networks and testbeds deployment. However, little is known about the performance of the IPv6 Internet, especially from the perspective of end users. Projects that interested in performance were mainly for IPv6 native networks and testbed of them were limited with a few network components.

This thesis is intended to give the contribution to the network world by answering “In order to have IPv6 communication do we sacrifice current performance?”, “What happens if we use transition mechanisms?”, “Will it be better or worse than today?” questions in an enterprise-wide network testbed. Parameters of the results will be throughput and end-to end delay of the end users.

For the time being it's really hard to find a native IPv6 network or an IPv4 network using tunnelling technologies. For this reason a simulated environment (OPNET) with several small networks, routers, servers and individual users in an enterprise network is chosen as a testbed.

1.2 SCOPE OF THE THESIS

The primary focus in this study is to compare routing performance analysis of IPv4 and IPv6. In order to achieve this aim specific objectives are:

1. To put forward the main differences of the two protocols from the aspect of performance,
2. To design and develop a network model depicting current, near future and future internet topologies by merging IPv4, IPv6 transition mechanisms and native IPv6,
3. To examine the throuhput and end-to-end traffic delay,
4. To present statistical tables and explanations on those performance analysis.

1.3 LIMITATIONS AND ASSUMPTIONS OF THIS THESIS

In this thesis only wired IPv6 routing performance is tested. Other network issues such as QoS and security are not examined. Wireless, satellite or multicast communication performance in IPv6 are excluded. Performance comparisons of multimedia packets are not examined as well.

Routing itself has many parameters to measure. In order to concentrate on transition mechanisms and IPv6 we limited the parameters as throughput and end-to-end delay.

It is assumed in this thesis that hardware of all routers, PCs are identical, but the configurations. All router links are point-to-point links and PCs have local links to routers. All routers are in default configuration.

1.4 ORGANIZATION OF THE THESIS

Chapter 2 covers some background and theoretical developments and literature review about IPv4 and IPv6 in general, and some of the fundamental differences between the two network protocols stack; it also describes various transition mechanisms that are available when upgrading from IPv4 to IPv6. Chapter 3 is about simulation and the OPNET. Chapter 4 covers our network model and the methodology to achieve the goals of the study. In chapter 5 we are discussing test results of IPv4 vs. IPv6 with different network metrices and focusing on transition mechanisms test results and comparisons. Chapter 6 covers the conclusions and future work. At the end there are appendices.

CHAPTER 2

BACKGROUND INFORMATION AND LITERATURE SURVEY

2.1 IPv4 SHORTAGES

By the late 1980s too much organizations, especially research-based ones, connected to the Internet. With this unforseenable growth, more hosts [Figure 1] required IP addresses.

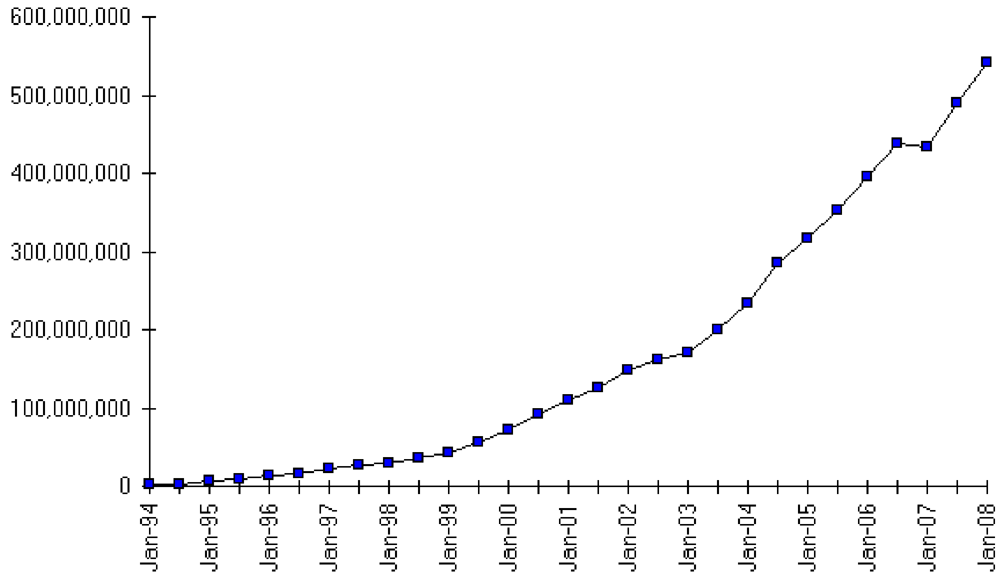


Figure 1 - Internet domain survey, host count [15]

By 1989 there were approximately 100,000 hosts connected to the Internet. That figure rose to approximately 1,000,000 in the year of 1992. Numbers increased tremendously in 1993 with the release of GUI browsers for HTML instead of text based ones. The pioneer browsers like Netscape made the World Wide Web a commercial tool. With the WWW and email becoming more focused on the consumer, internet connectivity moved from being focused at universities and research institutes to businesses and homes [21]. For today internet users reached almost 1.5 billion people [Table 1, Figure 2].

World Regions	Population (2007 Est.)	Internet Users Dec. 31, 2000	Internet Users Latest Data	Penetration (% Pop.)	Usage % of World	Usage Growth 2000-2007
Africa	955,206,348	4,514,400	51,065,630	5.3 %	3.5 %	1,031.2 %
Asia	3,776,181,949	114,304,000	578,538,257	15.3 %	39.5 %	406.1 %
Europe	800,401,065	105,096,093	384,633,765	48.1 %	26.3 %	266.0 %
Middle East	197,090,443	3,284,800	41,939,200	21.3 %	2.9 %	1,176.8 %
North America	337,167,248	108,096,800	248,241,969	73.6 %	17.0 %	129.6 %
Latin America/ Caribbean	576,091,673	18,068,919	139,009,209	24.1 %	9.5 %	669.3 %
Oceania / Australia	33,981,562	7,620,480	20,204,331	59.5 %	1.4 %	165.1 %
WORLD TOTAL	6,676,120,288	360,985,492	1,463,632,361	21.9 %	100 %	305.5 %

Table 1 - Internet usage and world population statistics, as of June 30, 2008 [9]

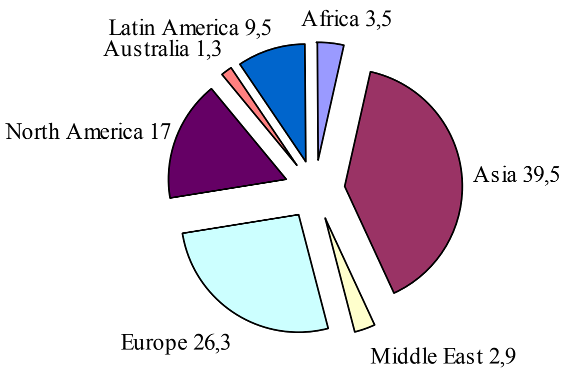


Figure 2 - World internet users, as of June 30, 2008 [9]

While the scaling properties of the Internet was discussed at the beginning of 1990s two critical factors came forth in the study were routing and addressing [RFC 1287]. The IPv4 address space is 32 bit field. So 4,294,967,296 unique addresses, considered as a sequence of 256 "/8s", where each "/8" corresponds to 16,777,216 unique address values.

As described in RFC 3330 a number of address blocks are reserved for uses outside “conventional” use in the public Internet. In adding up these special purposes use address reservations there are the equivalent of 36.086 /8 address blocks in this category. This is composed of 16 /8 blocks reserved for use in multicast scenarios, 16 /8 blocks reserved for some unspecified future use, 1 /8 block (0.0.0.0/8) for local identification, a single /8 block reserved for loopback (127.0.0.0/8), a /8 block reserved for private use (10.0.0.0/8), and a single /8 address block intended for some specialized use in so-called "public data networks" (14.0.0.0/8). Smaller address blocks are also reserved for other special uses (172.16.0.0/12 - This block is set aside for use in private networks [RFC1918]. 192.88.99.0/24 - This block is allocated for use as 6to4 relay anycast addresses, according to [RFC3068]. 192.168.0.0/16 - This block is set aside for use in private networks [RFC1918]). The remaining 219.914 /8 address blocks are available for use in the public IPv4 Internet. IANA holds a pool of unallocated addresses, while the remainders have already been allocated by IANA for further downstream assignment by the RIRs [Figure 3].

The current status of the total IPv4 address space is indicated in [Table 2].

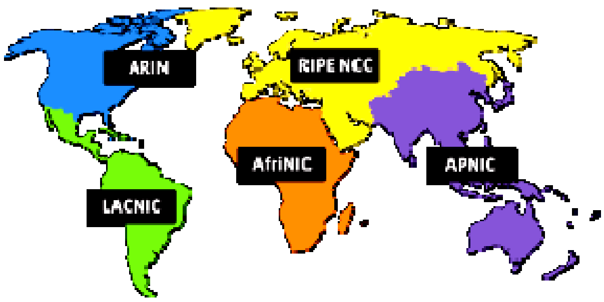


Figure 3 - IANA RIRs

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79
80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95
96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111
112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127
128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143
144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159
160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175
176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191
192	193	194	195	196	197	198	199	200	201	202	203	204	205	206	207
208	209	210	211	212	213	214	215	216	217	218	219	220	221	222	223
224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239
240	241	242	243	244	245	246	247	248	249	250	251	252	253	254	255

Reserved Allocated Unallocated

Table 2 - IPv4 addresses in 2008 [6]

According to the simulation and predictions, IPv4 unallocated address pool [Figure 4, 5] will be exhausted in the late of 2011.

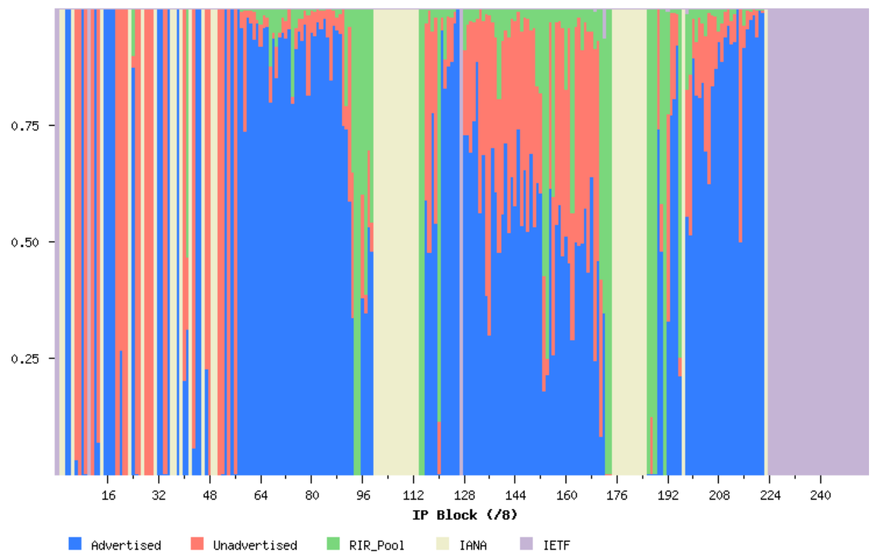


Figure 4 - IPv4 /8 address status [11]

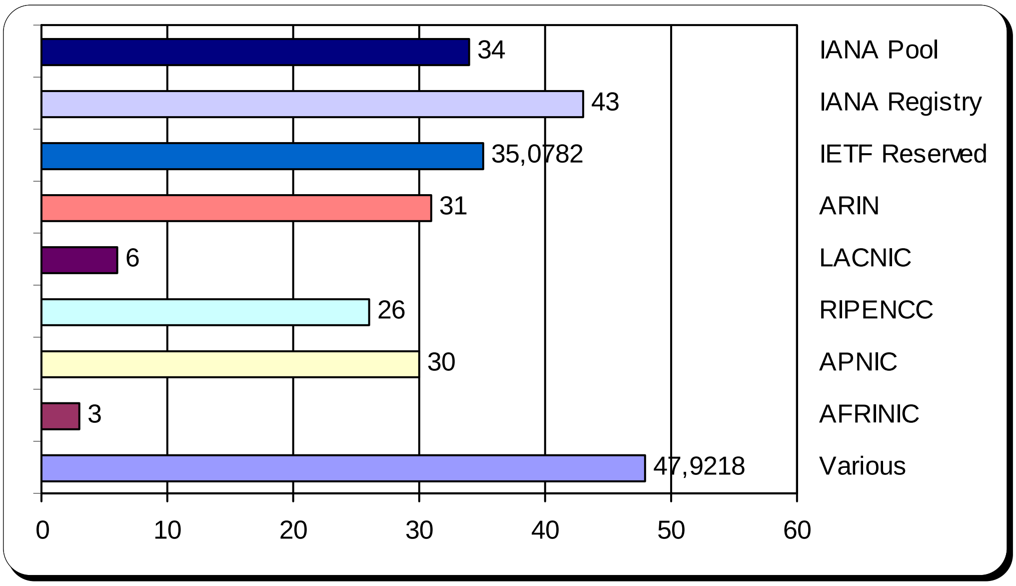


Figure 5 - IPv4 address pool status /8

A related prediction is the exhaustion [Figure 6] of the IANA IPv4 unallocated address pool on 12-Feb-2011 [22].

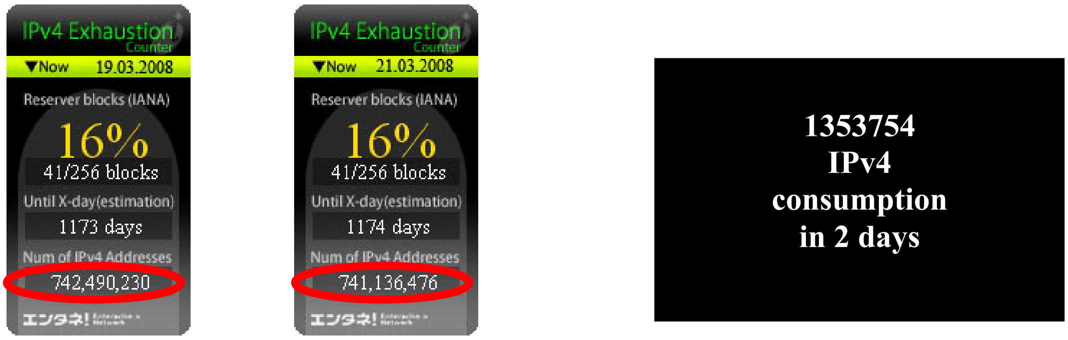


Figure 6 - IPv4 exhaustion time estimation

As more devices connect to the Internet they consume addresses, and the associated function of maintaining reachability information for these addresses implies ever larger routing tables. The work in studying the limitations of the 32 bit IPv4 address space produced a number of outcomes, including the specification of IPv6, as well as the refinement of techniques of network address translation (NAT) intended to allow some degree of transparent interaction between two networks using different address realms. Growth in the routing system is not directly addressed by these approaches, as the routing space is the cross product of the complexity of the topology of the network, multiplied by the number of autonomous domains of connectivity policy multiplied by the base size of a routing table entry. When a network advertises a block of addresses into the exterior routing space this entry is generally carried across the entire exterior routing domain of the Internet. To measure the characteristics of the global routing table it is necessary to establish a point in the default-free part of the exterior routing domain and examine the BGP routing table that is visible at that point.

One of the measurement techniques of the size of the routing table was measuring the size of the BGP table at hourly intervals. A detailed view of the 15-year dynamics of the Internet routing table growth is shown in [Figure 7] [11].

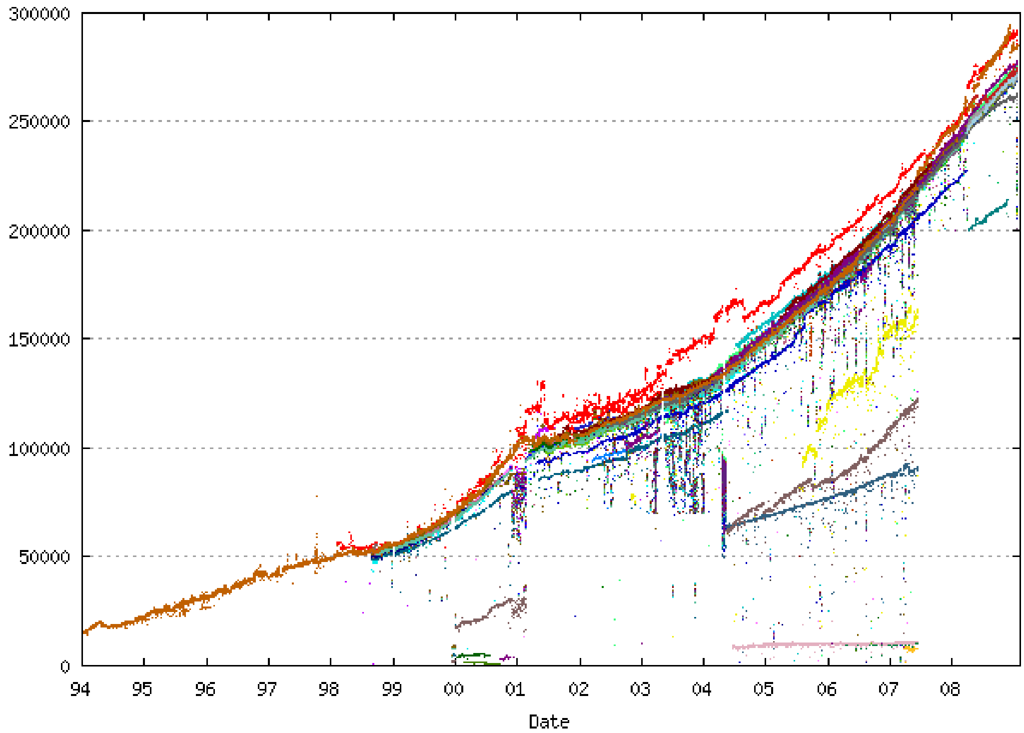


Figure 7 - Growth of the BGP tables 1994 – 2009 [22]

A technology called NAT is widely implemented to temporarily solve the IP address shortage and is commonly used to provide additional network security. A NAT firewall is placed between the real Internet and local area networks. NAT allows computers on the local network to connect with the outside world, but because the machines on the local network do not have their own Internet IPs they can not be directly accessed from the outside [22]. NAT certainly reduced the demand of IP but caused several problems as:

- Breaks end-to-end model, inhibit some novel uses of the Internet (peer-to-peer networking & applications such as instant messaging, interactive games, VOIP), enables end-to-end address conflicts, prevent IP level end-to-end security,
- Increases local support burden and complexity,
- Certain applications do not work properly in conjunction (Makes difficult to use IPsec, Mobile IP etc.),
- Increases the probability of mis-addressing,
- Reduce robustness,
- Limited control on initial connections.
- Increase network management and software development costs.
- Will become more expensive solution after the date of the IPv4 address running out [13].

2.2 THE NEED OF IPv6

It is expected that some time in the years of 2011/2012 we will definitely run out of IPv4 address space. In Asia the available IPv4 address space is already exhausted. This is why many Asian ISPs have already begun to roll out IPv6 commercially. IPv4 offers less than one IP address per person living on this planet and therefore we need a new version with a larger address space. With the new types of services that we will have in the future we will not only need IP addresses for personal computers and servers, but for all sorts of devices, like mobile phones, cars, refrigerators, TV-sets, sensor systems, home games and many more. The answer to that challenge is IPv6.

IPv6 offers a new, clean, well designed protocol stack which implements all the features of security (IPsec), quality of service and configuration (auto-configuration). All applications that are known on IPv4 can be ported to IPv6, with additional features if required. IPv6 is also designed taking into account the mobile networks, which are expected to be ubiquitous networks of the future providing always on-line, anytime and anywhere. IPv6 is considered to be the backbone of the future information society [13].

The design of IPv6 is intentionally targeted for minimal impact on upper and lower layer protocols by avoiding the random addition of new features.

Some major benefits of IPv6 are:

- Routing in IPv6 is eased by the use of aggregate addressing scheme, and hardware processing of the protocol is made easier.
- Stateless auto-configuration enables easily deployable unmanaged networks (e.g. home applications) or fast evolving networks (e.g. government applications, crisis management etc.)
- The requirement for security at the IP level. The support of IPsec is mandatory in IPv6 implementations
- The need for better support for real-time delivery of data, called as well quality of service (QoS). IPv6 keeps the existing IPv4 TOS (Traffic class in IPv6) but it can be enriched with the use of the flow label field in the IPv6 header.
- IPv6 has been designed with efficient built -in mobility mechanisms. The arrival of multimedia applications which would be far more efficient with an effective multicast support in IP. The support of multicast is mandatory in IPv6.

2.3 CORE AND INITIAL PROJECTS OF IPv6

The projects that are mentioned below are the initial and most famous projects on IPv6. Nowadays world there are so many special or government supported IPv6 networks. Most countries have a task force about IPv6. There is a great amount of discovering, putting into practice, transaction, identifying vulnerabilities projects all around the world.

2.3.1 WIDE v6 Project

WIDE Project got underway with a research group, researching UNIX based operating systems with a dream to construct a distributed processing environment with digital communication functions. The terms “Large-scale” and “Wide area” were aimed at a large-scale and global field. In hindsight, this theme has been pursued with little variation for almost 20 years. The growth and development of a globally distributed processing environment have been significant, and it seems as if the responsibilities, roles, dreams and ideals continue to spiral out remarkably. Sub projects of WIDE are KAME, USAGI and TAHI projects.

2.3.2 KAME Project

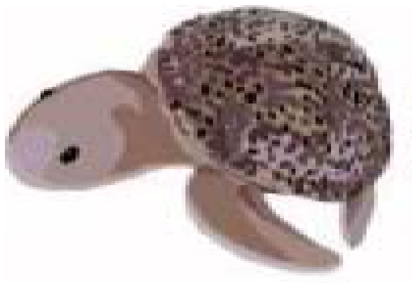


Figure 8 - Kame icon

The KAME (a Japanish word meaning “turtle” as in [Figure 8]) Project is a joint effort to create a single solid software set, especially targeting IPv6/IPsec. The KAME Project aims to provide free reference implementations of:

- IPv6,
- IPsec (for both IPv4 and IPv6),
- Advanced internetworking such as advanced packet queuing, mobility, and whatever interesting on BSD variants.

Currently several BSD variants are being developed including FreeBSD (4.0 and beyond), NetBSD (1.5 and beyond), OpenBSD (2.7 and beyond), and BSD/OS (4.2 and beyond). They are developing/improving network code (sys/netinet tree) separately. The most important task is to implement the best networking code possible, under the BSD copyright.

2.3.3 USAGI Project

Figure 9 - USAGI icon

The USAGI (UniverSAl playGround for IPv6 Project) Project depicted with a "rabbit" icon as in [Figure 9] aims to improve IPv6 environment on Linux and deploy the IPv6 Internet on the world. Many improvements on the kernel, libraries and applications have achieved. Many efforts have been merged into the mainline Linux kernel tree (2.6, especially). Some study areas of USAGI Project are:

- IPv6 stack in the Linux kernel,
- IPv6 APIs in the glibc library,
- IPv6 capable applications,
- IPSec for IPv4 and IPv6,
- Implementing general tunnel device,
- Implement extension header functions,
- Router renumbering protocol.

2.3.4 TAHI Project

Figure 10 - TAHI icon

TAHI (depicted with a fish named "snapper" as in [Figure 10]) Project is the joint effort formed with the objective of developing and providing the verification technology for IPv6.

The growth process of IPv4 was the history of encountering various kinds of obstacles and conquering such obstacles. However, once the position as infrastructure was established, it is not allowed to repeat the same history. This is a reason why the verification technology is essential for IPv6 deployment. Study areas of TAHI project are:

- Developing conformance tests and interoperability tests for IPv6.
- To help the KAME and USAGI project activities in the quality side by offering the verification.

2.3.5 The 6Bone (1996-2006):

The 6Bone started out as a network of IPv6 islands working over the existing IPv4 infrastructure of the Internet by tunneling IPv6 packets in IPv4 packets. The tunnels were mainly statically configured point-to-point links. The 6Bone became a reality in early 1996 as a result of an initiative of several research institutes. The first tunnels were established between the IPv6 laboratories of G6 in France, UNI-C in Denmark, and WIDE in Japan. It has the special prefix of 3FFE:/16 and ended in 2006.

2.3.6 Projects on Performance Comparison

There are some projects varying from directly connected two PCs to data packets traffic analysis of ISPs on performance comparison of the two protocols. These comparisons are made both in real networks and simulation environments on several operating systems as well.

In the M.S. thesis [26] there are two developed networks. In the first network router-to router performance with two PCs and two routers in a network is tested. For point-to-point performance test network of two PCs directly connected to each other is developed. Bandwith utilization and round trip time performance tests made in IPv6, IPv4 and DSTM environments both in Windows and Linux operating systems.

Authors of [27] have developed a simulated environment network in OPNET in order to measure the throughput and delay of native IPv6 and IPv4. In this network an end user downloads files from the FTP server on the other edge of simulated network.

As a real time performance test [28] Cisco made a performance comparison of IPv6, IPv4 and dual stack configured networks, with the metrices of throughput, frame loss and latency variances. Tests made with various frame sizes in bytes on different routers. There are interesting real results of the two protocols that we will cover in the conclusion section.

Another real measurement of ongoing networks made by the authors [29] on CERNET. They analyzed over 585680 packet level traces with 133,340 million packets on 936 dual-stack web servers located in 44 countries. For the performance metrices of IPv4 and IPv6, connectivity, packet loss rate, round-trip time etc. are tested.

There are many more performance related studies in the globe but one real fact of current internet is that there are so much network links that cannot handle the proposed IPv6 traffic. IPv6 was declared to operate on links with minimum MTU of 1280. In 6Bone testbed environment there were more than 30% of tunnelled links with less MTU than 1280. It's hoped that until the exhaustion date of the IPv4 all network equipments and interfaces that cannot handle native IPv6 (path MTU 1500 bytes) will be upgraded to IPv6 capable ones. It is a good hope but what happens if that expected upgrade won't take place? As of today the solution passes through tunnel operations.

In this thesis we studied on this critical point and tried to figure out the performance of current IPv4 networks, transition time tunnelled networks, and eventually native IPv6 networks.

2.4 MAJOR DIFFERENCES OF IPv4 AND IPv6

2.4.1 Header Comparison

Implementing a hierarchical addressing structure is just one way that IPv6 simplifies routing. Unlike IPv4, which uses variable-length packet headers, IPv6 uses fixed-length packet headers of 40 bytes. These fixed-length headers allow routers to parse packets more efficiently.

IPv6 further simplifies routing requirements by using a smaller number of header fields. IPv4 packet headers contain 14 fields; IPv6 packet headers contain only eight fields. [Figure 11]

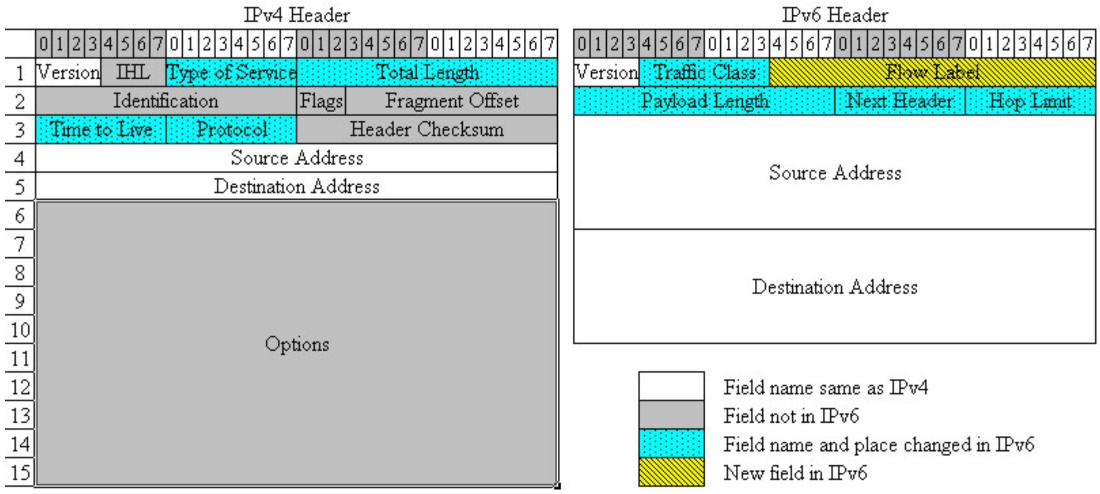


Figure 11 - IPv6 significantly reduces and standardizes the number of fields in packet headers from the IPv4 format

Since IPv6 packets have fixed-length headers, the old IPv4 header length field is obviously no longer needed. Other eliminated IPv4 fields include the fragment offset, identification, flags, and header checksum fields.

The most significant deletion is the IPv4 header checksum field, which contains a computation based on the total number of bits in each particular IPv4 header. Each time a router receives an IPv4 packet, the router recomputes the number of bits the header contains.

The router then checks its computation against the computation contained in the IPv4 header checksum field. If these two computations are identical, the data contained in the IPv4 header is most likely uncorrupted. In this case, the router forwards the packet. If the two computations are not identical, the router assumes the IPv4 packet is corrupted and discards it.

According to the IAB's Internet draft titled "The Case for IPv6," the IPv4 header checksum field is an unnecessary field that "has caused reduced performance in today's Internet". Because corrupted packets can be detected at both the data-link layer of the OSI model and the transport layer of the OSI model, routers do not need to check for bad packet headers [14]. The transport layer will catch any bad packets the data-link layer misses [16].

The main reasons for these removals are:

Header Length: The basic IPv4 header is only 20 bytes long. However, the basic IPv6 header has a fixed length of 40 octets. The IPv4 header length indicates the packet’s total length, including the Options field. When present, the Options field increases the length of the IPv4 header. Instead of the Options field, IPv6 uses the Extension field. The Extension field is handled differently from how IPv4 handles the Options field.

Identification, Flags, and Fragment Offset: Fragmentation is handled differently in IPv6. It is no longer done by intermediate routers in the networks, but by the source node that originates the packet. Removing the Fragmentation field removes costly CPU processing at intermediate routers. The path MTU discovery (PMTUD) mechanism is recommended for every IPv6 node to avoid fragmentation.

Header Checksum: Link-layer technologies (Layer 2) perform their own checksum and error control. The reliability of link-layer is now good and upper-layer protocols such TCP and UDP (Layer 4) have their own checksums. UDP checksum, which was optional in IPv4, is mandatory in IPv6. Therefore, the checksum at Layer 3 is redundant, so the Header Checksum field is unnecessary in IPv6 and suppresses the recomputation process each time a packet passes through a router.

Options and Padding: The Options field is radically changed in IPv6. The options are now handled by Extension Headers (See Appendix A). The Padding field is also removed. The removal of Options and Padding headers simplifies the IP header. Thus, the basic IPv6 header has a fixed length of 40 octets, allowing less processing by routers along the delivery path compared to IPv4.

The other fields in the IPv4 header Version, Type of Service, Total Length, Time to Live, Protocol Number, Source IPv4 Address, and Destination IPv4 Address, either were not changed or were modified only slightly.

For a complete list of header comparison of IPv4 vs. IPv6 and extension headers in IPv6 see Appendix A.

2.4.2 Comparing Address Types

If you install IPv6 on a computer that is not connected to a network in which an IPv6 router is present, the computer automatically configures a link-local IPv6 address, which is a type of address that allows you to communicate with computers on your subnet. If you connect your computer to a subnet in which an IPv6 router is present, the router assigns your computer an IPv6 global or site-local address. The site-local address allows your computer to communicate within your intranet. The global address allows your computer to communicate with computers on the IPv6 Internet.

The left-most bits of an IP address are called the format prefix (FP), which indicates the specific type of IPv6 address. IPv6 accommodates many address types, including the following:

- **Unicast addresses**. Provide point-to-point, directed communication between two hosts on a network.
- **Multicast addresses**. Provide a method for sending a single IP packet to multiple hosts in a group. A multicast address is used for one-to-many communication.
- **Anycast addresses**. Provide a method of delivering a packet to the nearest member of a group. Currently, anycast addresses are used only as destination addresses and are assigned only to routers. An anycast address is used for one-to-one-of-many communication.

In Table 3 there are comparisons of some basic elements of IPv4 and IPv6 addressing.

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

Table 3 - Comparison of IPv4 and IPv6 addressing

Other addressing and related comparisons from different aspects can be find in Appendix B.

2.4.2.1 Unicast Addresses

Unicast addresses [Figure 12] identify a single interface within the scope of a particular type of unicast address. The scope of an address is the region of the IPv6 network over which the address is unique. With the appropriate unicast routing topology, packets addressed to a unicast address are delivered only to a single interface.

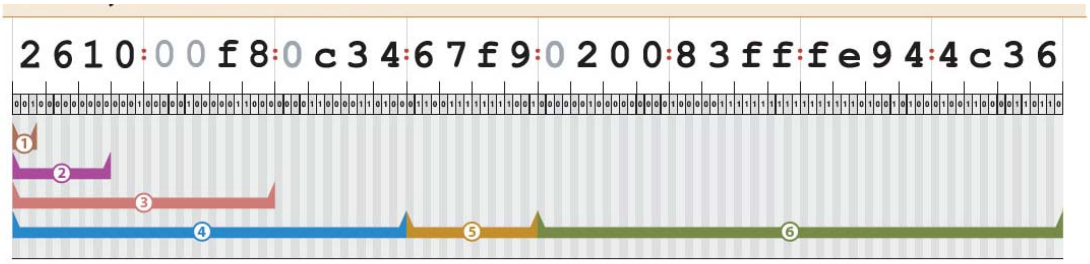


Figure 12 - Anatomy of an IPv6 unicast address [32]

- **2000::/3:** The current IPv6 address space for unicast allocations is 1/8 of the total address space.
- **IANA Allocation to Registries**: (Varies) IANA makes assignments to regional registries. New allocations are /12 bits, previous assignment have varied.
- **ISP Allocations :**Regional registries make assignments to local ISPs. A typical assignment is /32 bits, but more space may be assigned.
- **End-Site Allocations:** ISPs make assignments to their customers. The amount of address space varies, but a /48 bit allocation is common. Organizations can get larger assignments, based on need (Command Information has a /32 allocation), smaller organizations may get less space (for small companies a /56 is common).
- **Subnet Assignments:** Organizations make assignment to individual subnets, where the most common size is /64. With 16 bits subnetting bits available, an organization can deploy as many as 65,536 subnets.
- **Interface ID:** Interfaces must have a unique identifier on the subnet, often created by embedding the underlying 48-bit (Layer 2) MAC address. Theoretically then, a single subnet could support 2^64 active hosts, clearly far beyond the practical limit.

The following are types of unicast IPv6 addresses:

2.4.2.1.1 Aggregatable Global Unicast Addresses

Identified by the format prefix (FP) of 001 [Figure 13], these addresses are equivalent to public IPv4 addresses.

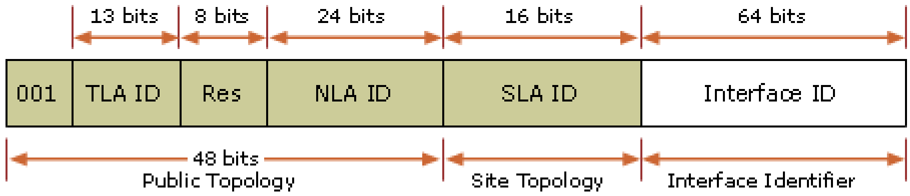


Figure 13 - Structure of an IPv6 global address [31]

- **TLA ID field:** The thirteen bits that follow the format prefix of unicast addresses contain the TLA ID. Representing 8,192 available addresses, these thirteen bits are assigned by IANA designated registries. The TLA ID is used to identify the relatively small number of large, long-haul backbone providers that exist worldwide.
- **Res field:** These 8 bits were reserved for future use and set to zero. By leaving these 8 bits between the TLA ID and NLA ID unused, they could be later used to expand either the TLA ID or NLA ID fields as needed.
- **NLA ID field:** An organization allocated a TLA ID creates address hierarchy and identifies sites using the NLA ID. Organizations that receive address allocation from the NLA space are called NLA registries. This is similar to a CIDR block of IPv4 address space given to a top-level ISP, which then gives blocks of address space with longer prefixes to its customer.
- **SLA ID field:** The NLA providers, in turn, assign single addresses or blocks of addresses to individuals and companies from the 16-bit SLA ID address space that follows the NLA ID space. NLA providers can also subdivide their own 24-bit address allocation. They can then assign blocks of address space to smaller service providers. In addition, NLA providers can allocate part of their 24-bit address space to large organizations, such as government organizations, that require more than the 65,535 addresses available within the 16-bit SLA ID space.

Finally, companies that are assigned blocks of SLA ID addresses are responsible for assigning those addresses to networks and subnetworks within their organizations. IPv6 subnet prefixes are allocated out of the SLA address space. As with IPv4 subnet prefixes, IPv6 subnet prefixes are associated with one link. Unlike IPv4, however, IPv6 allows you to assign multiple subnet prefixes to any given link. For example, the subnet prefix in a link's anycast address may be just one of the subnet prefixes that identifies that link.

- **Interface ID field:** The last 64 bits of the IPv6 128-bit address space are called the Interface ID. In IPv4, each host is assigned a unique number out of the total number of host addresses available within a given class-based address. In IPv6, on the other hand, interface addresses are assigned according to either the new Institute of Electrical and Electronics Engineers (IEEE) Equipment Identifier (EUI) 64 identifier or the old IEEE EUI-48 identifier. (EUI identifiers are also known as Medium Access Control [MAC] addresses.)

The new IEEE EUI-64 format is a 64-bit series of numbers. The first 24 bits in an EUI-64 number identify the manufacturer of a particular interface, and the last 40 bits identify the device itself.

For example, the first 24 bits of a router EUI-64 identifier identify the company that manufactured the router (such as Cisco). The IEEE assigns 24-bit manufacturer numbers, and the manufacturer then assigns the following 40 bits of the router EUI-64 identifier [30, 31].

2.4.2.1.2 Local Use Unicast Addresses

Provide two types of addresses:

- **Link-local addresses**. Identified by the FP of 1111 1110 10 [Figure 14], these addresses are used by nodes when they are communicating with neighboring nodes on the same link.

Figure 14 - Link-local addresses [31]

- **Site-local addresses**. Identified by the FP of 1111 1110 11 [Figure 15], these addresses are equivalent to the IPv4 private address space. Use these addresses between nodes that communicate with other nodes in the same site.

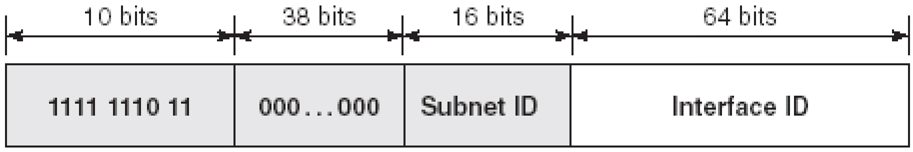


Figure 15 - Site-local addresses [31]

2.4.2.2 Unspecified Addresses

Each of the 128 bits in the IPv6 unspecified address have a value of zero. Because this reserved address is actually the absence of an address, the unspecified address should neither be assigned to a node nor be used as a packet's destination. However, hosts can use this address as a source address to initialize themselves before they have configured their own unicast addresses [17]. The IPv6 unspecified address, 0:0:0:0:0:0:0:0 or ::, is equivalent to the IPv4 unspecified address of 0.0.0.0.

2.4.2.3 Loopback Addresses

The loop back address is another reserved address that should not be assigned to a node. Packets that contain the loop back address as a destination code come back to the node from which the packet was sent. For example, you can use the loop back address to test network connections [17]. The IPv6 loopback address, 0:0:0:0:0:0:0:1 or ::1, is equivalent to the IPv4 loopback address of 127.0.0.1.

All address types are shown intogether in [Table 4].

Defining First Digits	Address Type
00	Unspecified, loopback, IPv4-compatible
2	Aggregatable global unicast
3	Aggregatable global unicast
FE8	Link-local
FEC	Site-local
FF	Multicast

Table 4 - Digits that define the address type [2]

2.4.2.4 Transition or Compatibility Addresses

Provided to help you migrate from IPv4 to IPv6; these addresses allow both types of hosts to coexist on your network. IPv6 defines the following transition IPv6addresses.

2.4.2.4.1 IPv4 Compatible Addresses

IPv4 compatible addresses are used by IPv6/IPv4 nodes that communicate with IPv6 over an IPv4 infrastructure. IPv6/IPv4 nodes are nodes that run both the IPv4 and IPv6 protocols. The format for an IPv4 compatible address is 0:0:0:0:0:0:w.x.y.z or ::w.x.y.z [Figure 16] (where w.x.y.z is the dotted-decimal representation of a public IPv4 address). Most of the IPv6 protocol implementations support for IPv4 compatible addresses, but support is not enabled by default.

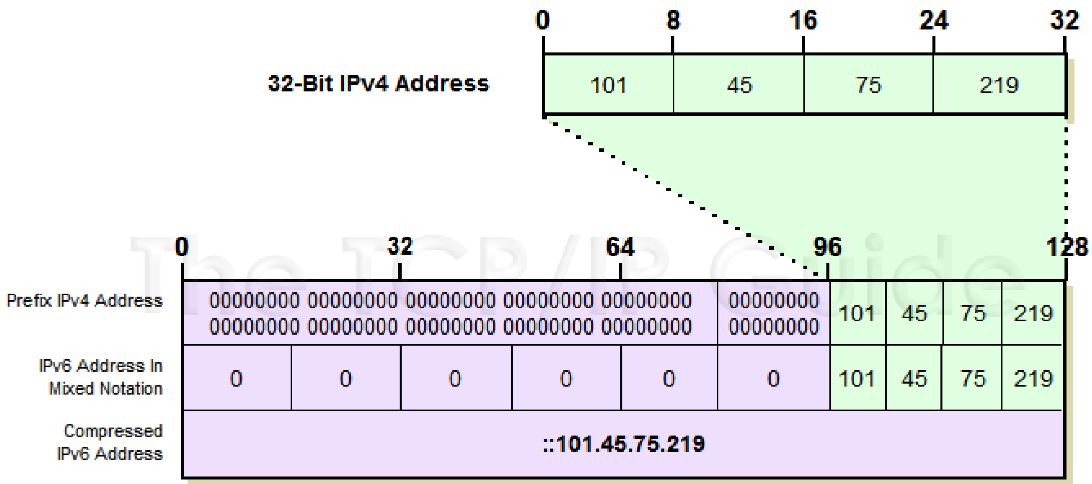


Figure 16 - IPv4 compatible addresses [10]

2.4.2.4.2 IPv4 Mapped Addresses

IPv4 mapped addresses are used to represent an IPv4 only node to an IPv6 node. The IPv4 mapped address is never used as a source or destination address for an IPv6 packet. It is used only for internal representation. The format for an IPv4 mapped address is 0:0:0:0:0:FFFF:w.x.y.z or ::FFFF:w.x.y.z [Figure 17]. Most IPv6 protocol implementation do not support IPv4 mapped addresses.

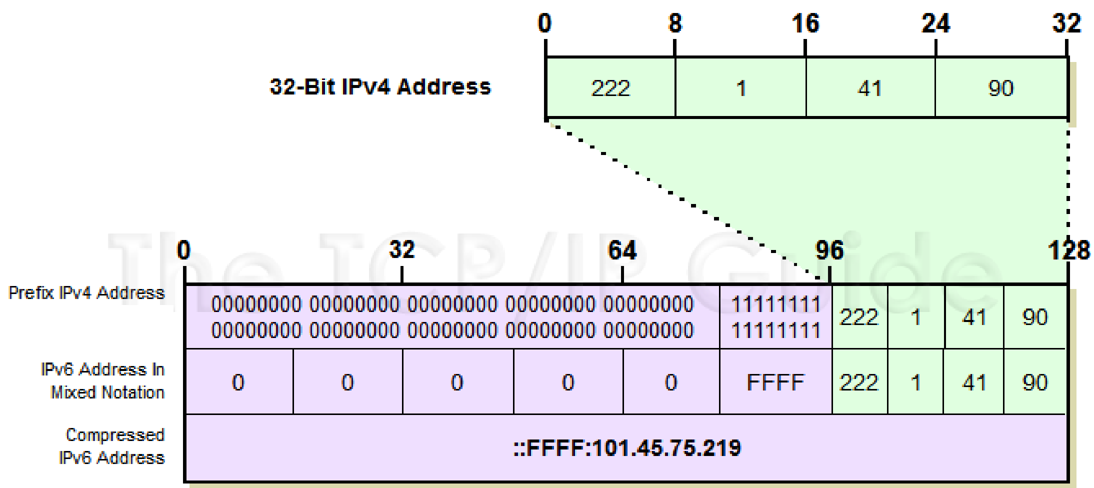


Figure 17 - IPv4 mapped address representation [10]

2.4.2.4.3 6to4 Addresses

6to4 addresses are used for communicating between two nodes that are running both IPv4 and IPv6 over the Internet. The 6to4 address is formed by combining the prefix 2002::/16 with the 32 bits of the public IPv4 address of the node or the site of the node, thus forming a 48-bit prefix. For example, for the IPv4 address of 131.107.0.1, the 6to4 address prefix is 2002:836B:1::/48 (where 836B:1 is the colon hexadecimal notation for 131.107.0.1). See [Figure 18].

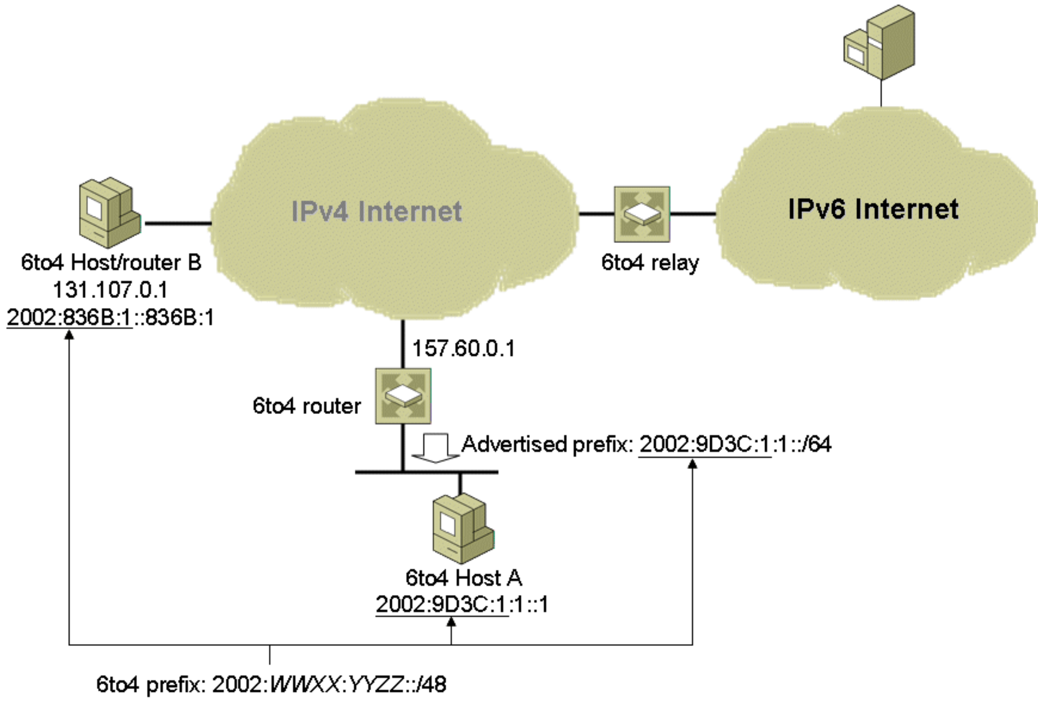


Figure 18 - 6to4 addresses in a network [31]

The 6to4 router is directly connected to the Internet and has been assigned the public IPv4 address of 157.60.0.1. The 6to4 router creates the 48-bit prefix 2002:9D3C:1::/48, in which 9D3C:1 is the colon hexadecimal notation for 157.60.0.1. The 6to4 router advertises the 2002:9D3C:1:1::/64 prefix on the LAN interface connected to the private intranet. The SubnetID portion of the 64-bit prefix can be manually configured or automatically determined by the 6to4 router. IPv6 hosts on the private intranet subnet configure an IPv6 address based on the 2002:9D3C:1:1::/64 prefix using standard IPv6 stateless address autoconfiguration. In this example, 6to4 Host A automatically configures the IPv6 address 2002:9D3C:1:1::1.

6to4 Host/router B is directly connected to the Internet and has been assigned the public IPv4 address of 131.107.0.1. The IPv6 protocol for automatically derives an address of the form 2002:*WWXX*:*YYZZ*::*WWXX*:*YYZZ*. Therefore, 6to4 Host/router B assigns itself the IPv6 address 2002:836B:1::836B:1 [2].

2.4.2.4.4 ISATAP Addresses

An address of the type [64-bit prefix]:0:5EFE:w.x.y.z [Figure 19], where w.x.y.z is a public or private IPv4 address, is used to represent a node for the address assignment mechanism known as Intra-Site Automatic Tunnel Addressing Protocol (ISATAP).

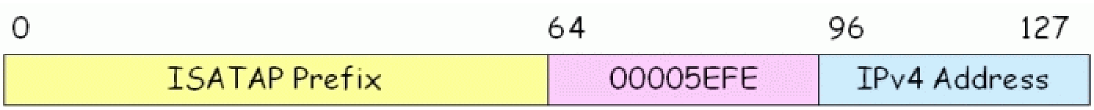


Figure 19 - ISATAP address

2.4.3 IPv6 Addresses that Must Be Supported by Nodes and Routers

2.4.3.1 IPv6 Addresses that a Node Must Support

An IPv6 node requires the following IPv6 addresses for proper operation:

- Link-local addresses for each interface,
- Assigned unicast address(es),
- Loopback address,
- All-nodes multicast address,
- Solicited-node multicast address for each of its assigned unicast or anycast addresses,
- Multicast addresses of all other groups to which the host belongs.

2.4.3.2 IPv6 Addresses that a Router Must Support

An IPv6 router requires the following IPv6 addresses for proper operation:

- All the required node addresses (See 2.4.3.1),
- Subnet router anycast addresses for the interfaces configured to act as forwarding interfaces,
- Anycast addresses configured on all the other routers,
- All-routers multicast address,
- Multicast addresses of all other groups to which the router belongs.

2.5 IPv6 TRANSITION MECHANISMS

There are different transition mechanism options available to an agency to ensure IPv4 and IPv6 interoperability. These mechanisms are categorized in the following three broad classes: dual-stack, tunnels (includes configured and automatic tunnels), and translation mechanisms.

2.5.1 Dual-Stacks

The term “dual-stack” refers to TCP/IP capable devices providing support for both IPv4 and IPv6 [Figure 20]. It is important to understand that having a device being able to communicate over both IPv4 or IPv6 does not necessarily means that all applications operating within this device are capable of utilizing both IPv4 and IPv6. The term “Dual-stack routing” refers to a network that is dual IP, that is to say all routers must be able to route both IPv4 and IPv6.

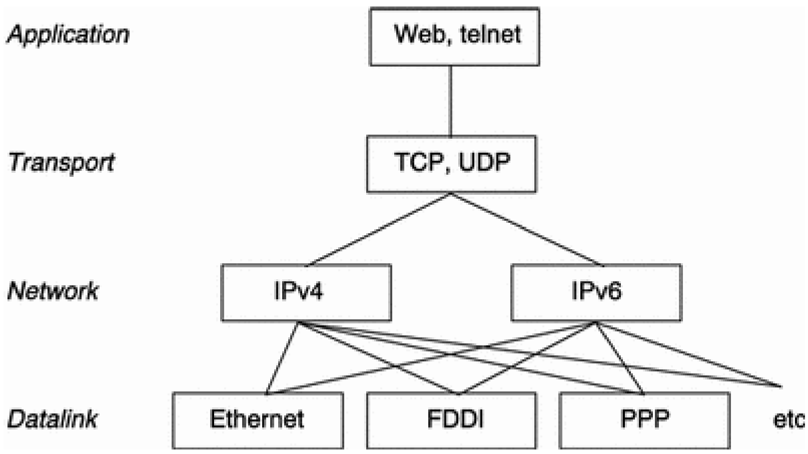


Figure 20 - Dual stack protocols [20]

Requiring all new devices be both IPv4 and IPv6 capable permits these devices to have the ability to use either IP protocol version, depending on the services available, the network availability, service, and the administrative policy. A transition scenario which calls for “dual-stack everywhere” provides the most flexible operational environment. Dual-stacked hosts running on a dual-stack network allow applications to migrate one at a time from IPv4 transport to IPv6 transport. Legacy applications and devices that are not yet upgraded to support access to the IPv6 stack can coexist with upgraded IPv6 applications on the same network system.

2.5.2 Tunnels

The term "tunneling" refers to a means to encapsulate one version of IP in another so the packets can be sent over a backbone that does not support the encapsulated IP version. For example, when two isolated IPv6 networks need to communicate over an IPv4 network, dual-stack routers at the network edges can be used to set up a tunnel which encapsulates the IPv6 packets within IPv4, allowing the IPv6 systems to communicate without having to upgrade the IPv4 network infrastructure that exists between the networks.

2.5.2.1 Configured Tunnels

The term "configured tunnels" is used when network administrators manually configure the tunnel within the endpoint routers at each end of the tunnel. Any changes to the network like renumbering must be manually reflected on the tunnel endpoint. Tunnels result in additional IP header overhead since they encapsulate IPv6 packets within IPv4 (or vice versa). A manual tunnel network is in [Figure 21].

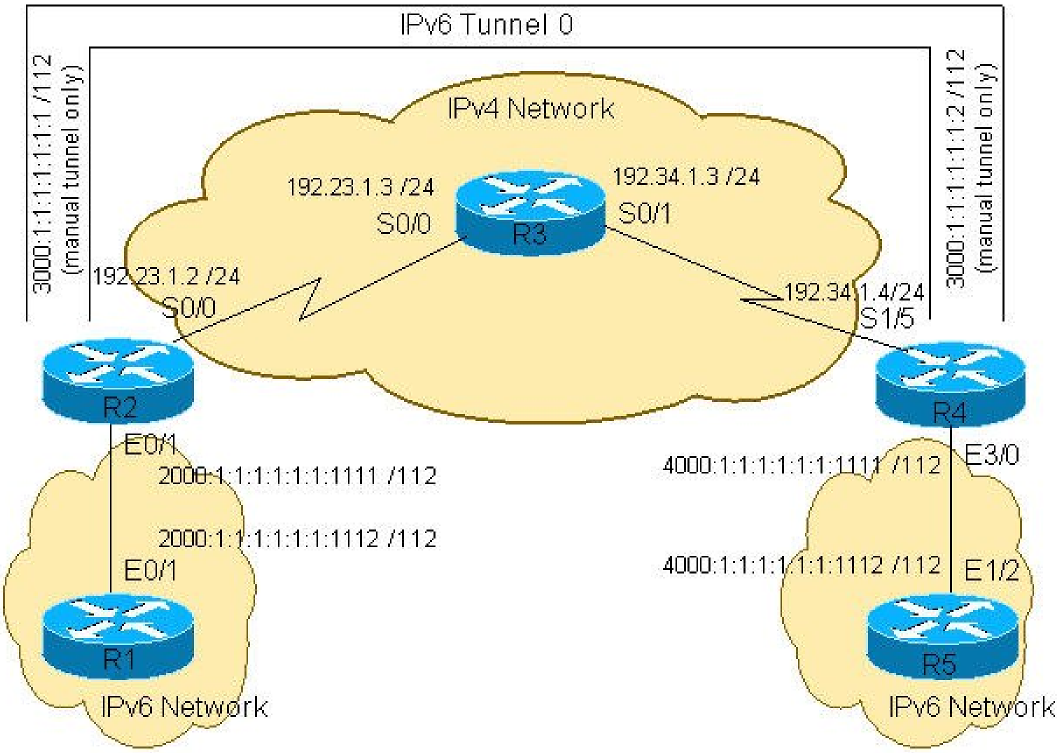


Figure 21 - Manually configured tunnel [2]

2.5.2.2 Automatic Tunnels

The term "automatic tunnels" is used when a device directly create their own tunnels to dual-stacked routers for shipping IP packets within IP. The IPv6 tunnel broker [Figure 22] (RFC 3053), 6to4 (RFC 3056), Teredo (Tunnelling IPv6 over UDP through NATs, [Figure 23]) and ISATAP (Intra-Site Automatic Tunnel Addressing Protocol) ship IPv6 packets within IPv4 and can be referenced as IPv6-over-IPv4

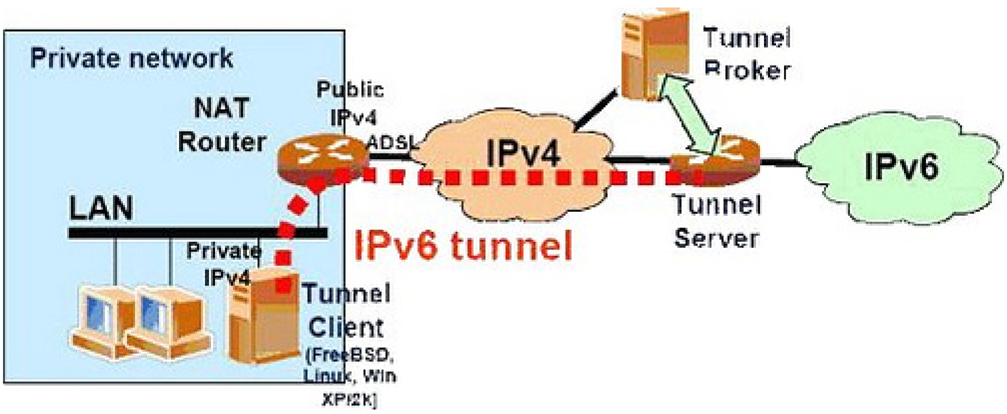


Figure 22 - Tunnel broker

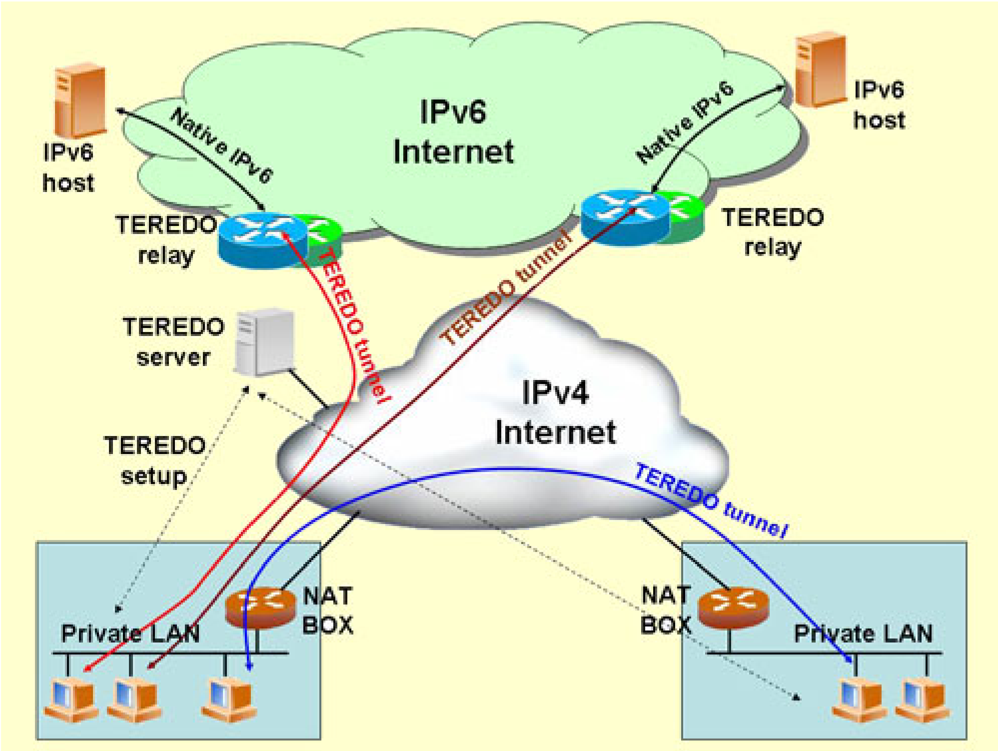


Figure 23 - Teredo tunnelling [33]

mechanisms while DSTM (Dual Stack Transition Mechanism) ships IPv4 packets within IPv6 and can be reference as IPv4-over-IPv6 mechanism.

The IPv6 tunnel broker mechanism uses dual-stacked servers sitting between IPv6 and IPv4 networks to assist in the set up of a configured tunnel to a host.

6to4, Teredo and ISATAP allow end host systems to create their own automatic tunnels to dual-stacked routers for shipping IPv6 packets within IPv4.

There is a 6to4 network diagram in [Figure 24].

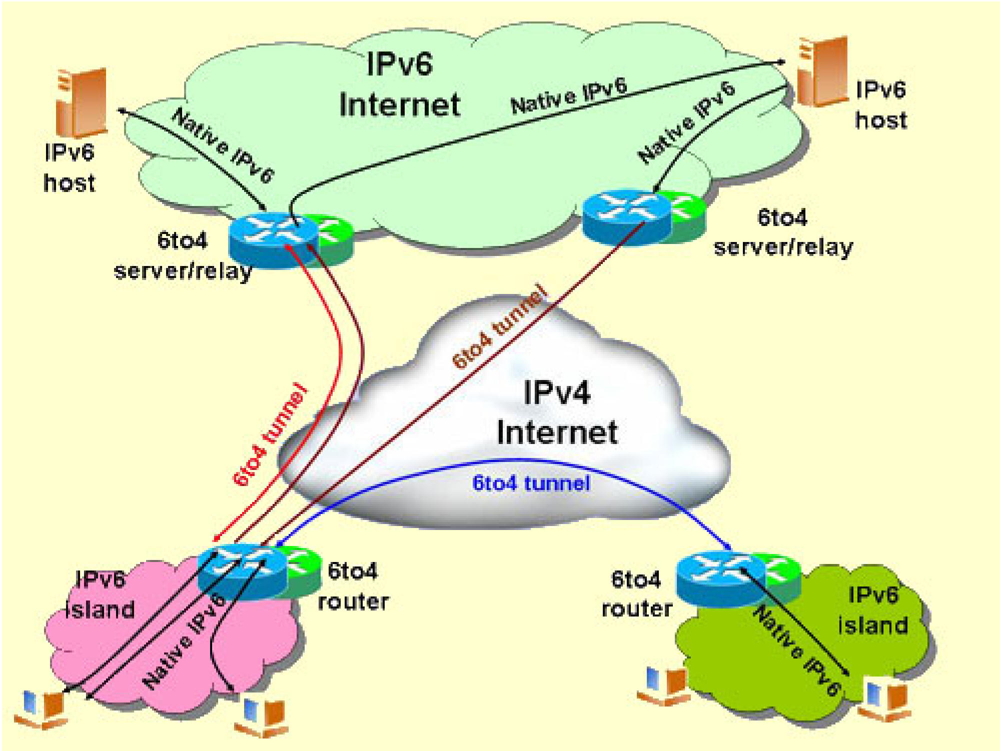


Figure 24 - 6to4 network [33]

While ISATAP is mainly for IPv6-over-IPv4 tunneling within a domain, all of the other IPv6-over-IPv4 mechanisms are designed to tunnel IPv6 packets out of an IPv4 only administrative domain.

Like configured tunnels, automatic tunneling has double IP header overhead, since tunnels encapsulate IPv6 packets within IPv4 (or vice versa).

DSTM technique provides a unique solution to the IPv4-IPv6 transition problem. This mechanism is designed to rapidly reduce the reliance on IPv4 routing and is intended for IPv6 only networks in which hosts still occasionally need to exchange information directly with other IPv4 hosts or applications. Network administration is simplified and the need of IPv4 global addresses is reduced. DSTM can be integrated with an IPv6 tunnel broker for tighter security integration. DSTM routers can be coupled with IPv4 firewalls and intrusion detection systems to secure IPv4 tunnel endpoints from IPv4 based attacks.

Special consideration must be given to the security risk associated with automatic tunneling as it allows user-nodes to establish tunnels that may bypass a site's security checkpoints such as firewalls and intrusion detection systems. In general, a full dual-stack along with IPv6-capable firewalls, guards, intrusion detection, and end-host security may provide a more secure and interoperable IPv6 transition solution than tunneling. However, for network infrastructures that contain IPv4 only or IPv6 only routing coupled with dual-stack end-nodes, automatic tunneling provides a flexible transition strategy. Again the risks associated with all potential solutions must be carefully considered.

2.5.3 Protocols Translators

The term "translators" refers to devices capable of translating traffic from IPv4 to IPv6 [Figure 25] or vice and versa. This mechanism is intended to eliminate the need for dual-stack network operation by translating traffic from IPv4-only devices to operate within an IPv6 infrastructure. This option is recommended only as a last resort because translation interferes with objective of end-to-end transparency in network communications. Use of protocol translators cause problems with NAT and highly constrain the use of IP-addressing.

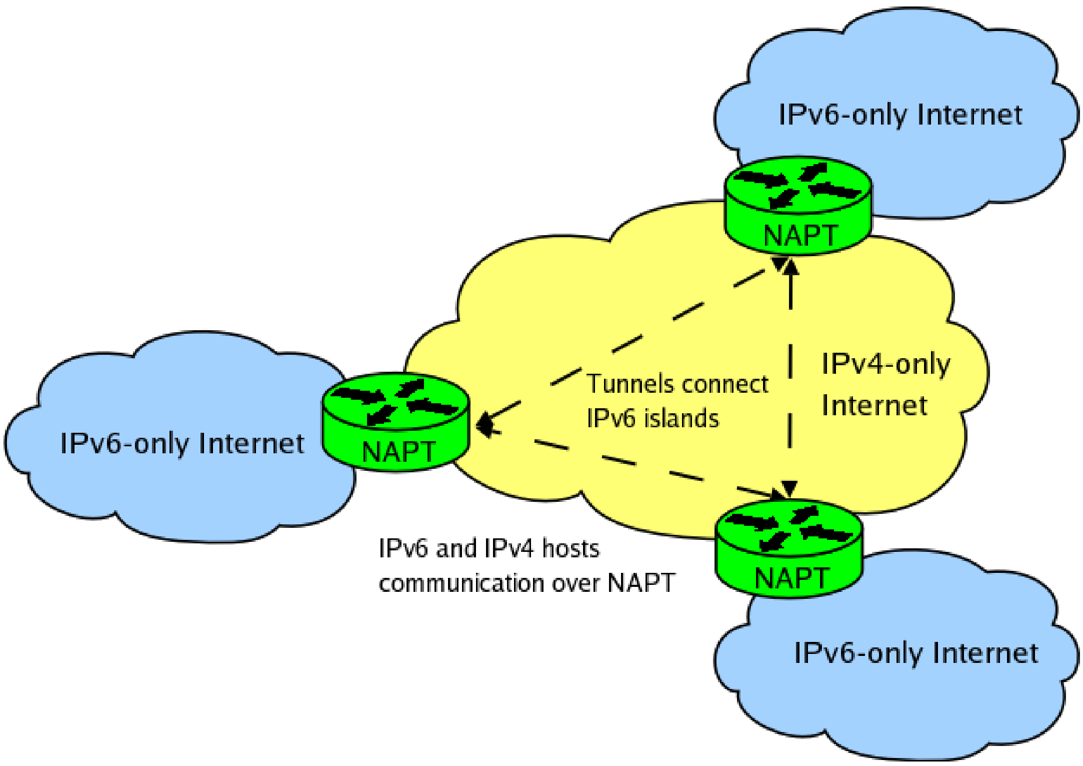


Figure 25 - Network address and protocol translation [34]

CHAPTER 3

SIMULATION AND OPNET

3.1 THE NEED OF SIMULATED ENVIRONMENT AND OPNET

Because of the costs and limitations of analysis in ongoing networks, simulation modeling is getting increasingly popular method for network performance analysis. Mainly two types of network simulation occur: Analytical network modeling and computerized network simulation. Analytical network modeling is done by mathematical analysis in order to characterize a network as a set of mathematical formulas. In this model complex overview of network cannot be put into consideration and natural network analysis cannot be achieved. For a more realistic and complex simulation as in real Discrete Event Simulation (DES) packages are needed. With DES; complex implementation of topologies, running simulation on specific portions and collecting desired results can be done. The advantages of using simulation program are obvious : less time to develop working hardware/software, ability to try out vast different scenarios of hardware and software attempts without the down side of the cost and time, and predict potential problems of hardware software before the actual use.

In our study to find out a network of various routers, switches, servers, tunnels and PCs are really hard. So we decided to use a network simulator. There are various simulators in the market. For DES specific simulation and rich result capabilities I chose OPNET simulated environment. Some studies have been made for the networking accuracy of the simulated networks. In a study of comparing the accuracy of real and simulation networks packet level analysis, with fine tuning the parameters of OPNET, close results to the real networks can be obtained [24].

OPNET is used in Ph.D. or M.S. thesis projects in almost every branch of computer network technologies [25]. As a network simulation industry leader, OPNET supports a number of techniques for building network topologies and representing protocol configurations accurately. The scope and dimension of the simulation analysis or study can be significantly reduced by building a topology appropriate to the study[18]. OPNET supports four simulation technologies: DES, Flow Analysis, ACE QuickPredict, and Hybrid Simulation. Hybrid Simulation is supported within the DES environment [Figure 26].

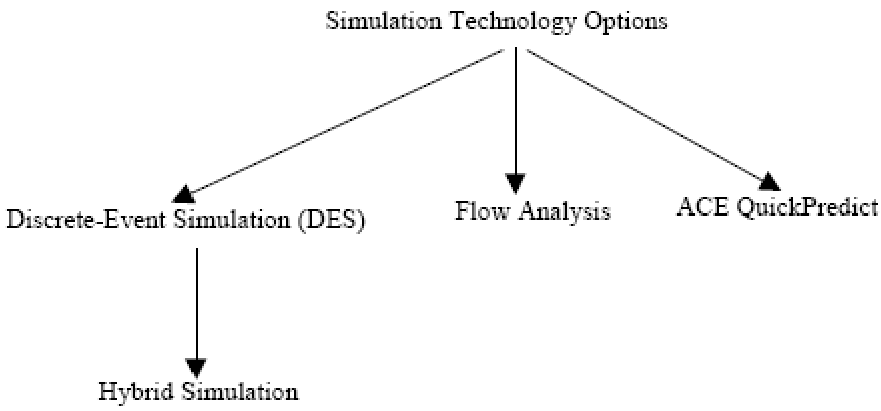


Figure 26 - OPNET simulation technology options

3.1.1 Discrete-Event Simulation

DES provides highly detailed models that explicitly simulate packets and protocol messages. The models in DES execute the protocol in much the same way as a production environment. Protocol behavior can be studied under different network conditions, convergence time, route oscillations, the impact of timers, and application performance in detail using DES as these studies require packet-by-packet simulation. Although DES provides very hi-fidelity results, simulation runtimes are longer than with other methods.

3.1.2 Hybrid Simulation

Hybrid Simulation is a different simulation technique supported within DES. Hybrid Simulation combines two distinct modeling techniques (analytical and discrete) to provide accurate, detailed results for targeted flows. It also provides greater control over simulation runtimes. The hybrid-simulation architecture lets you “fine-tune” the balance of analytical and discrete modeling to generate accurate results with reasonable runtimes.

3.1.3 Flow Analysis

Flow Analysis uses analytical techniques and algorithms to model steady state network behavior. Flow Analysis recreates the device routing tables from detailed configuration information to ensure high fidelity routing. Individual traffic flows possess detailed attributes such as source, destination, packet rate and delay. Flow Analysis does not model individual protocol messages or packets, therefore it does not generate results for transient network conditions. Flow Analysis can be used to study routing and reachability across the network in steady state, and in scenarios with one or more failed devices. Execution runtimes can be significantly faster when compared with DES.

3.1.4 ACE QuickPredict

ACE QuickPredict uses an analytical technique for studying the impact on application response time of changing network parameters such as bandwidth, latency, utilization, packet loss, and TCP window size. This technique is supported within the OPNET Application Characterization Environment (ACE).

In OPNET simulated environment the recommended simulation technologies for main study areas are as in [Table 5].

Problem	Recommended Technology
Application performance prediction	DES / ACE QuickPredict
Application impact on the network	Flow Analysis
Application impact on other applications	Hybrid Simulation
Application networkability	ACE QuickPredict

Server performance, consolidation and capacity planning	DES
Wireless, DOCSIS, MANET, SIP	DES
QoS	Hybrid Simulation
Routing protocol convergence time	DES
SLA analysis	DES
Security policies	Flow Analysis
Security – data traffic filtering	DES
IP/MPLS/ATM failure impact analysis	Flow Analysis
IP/MPLS/ATM routing and reachability (using topology and configs)	Flow Analysis
IP/MPLS/ATM capacity planning (using topology, baseline link utilization)	Flow Analysis
IP/MPLS/ATM app. response time (using topology, baseline link utilization)	Hybrid Simulation
ATM capacity planning (using topology and PVC loads)	Flow Analysis
IP/MPLS/ATM capacity planning (using topology, end-to-end flow traffic)	Flow Analysis

Table 5 - Recommended simulation technologies according to the key areas [19]

3.2 POWERFUL TOOLS IN OPNET

There are powerful tools which allow developers specify models in great detail, identify the elements of the model of interest, execute the simulation and analyze the generated output data:

- Hierarchical Model Building
    - Network Editor,
    - Node Editor,
    - Process Editor.
- Running Simulations
    - Simulation Tool,
    - Debugging Tool.

- Analyzing Results
  - Probe Editor,
  - Analysis Tool,
  - Filter Tool,
  - Animation Viewer.

3.2.1 Hierarchical Modeling

OPNET provides four tools called editors to develop a representation of a system being modeled. These editors (Network, Node, Process and Parameter Editors) are organized in a hierarchical fashion, which supports the concept of model level reuse. Models developed at one layer can be used by another model at a higher layer.

3.2.2 Network Model

Network Editor is used to specify the physical topology of a communications network, which define the position and interconnection of communicating entities, i.e., node and link.

3.2.3 Node Model

Communication devices created and interconnected at the network level need to be specified in the node domain using the Node Editor [Figure 27]. Node models are expressed as interconnected modules. These modules can be grouped into two distinct categories. The first set is modules that have predefined characteristics and a set of built-in parameters. Examples are packet generators, point-to-point transmitters and radio receivers. The second group contains highly programmable modules. These modules referred to as processors and queues, rely on process model specifications [19].

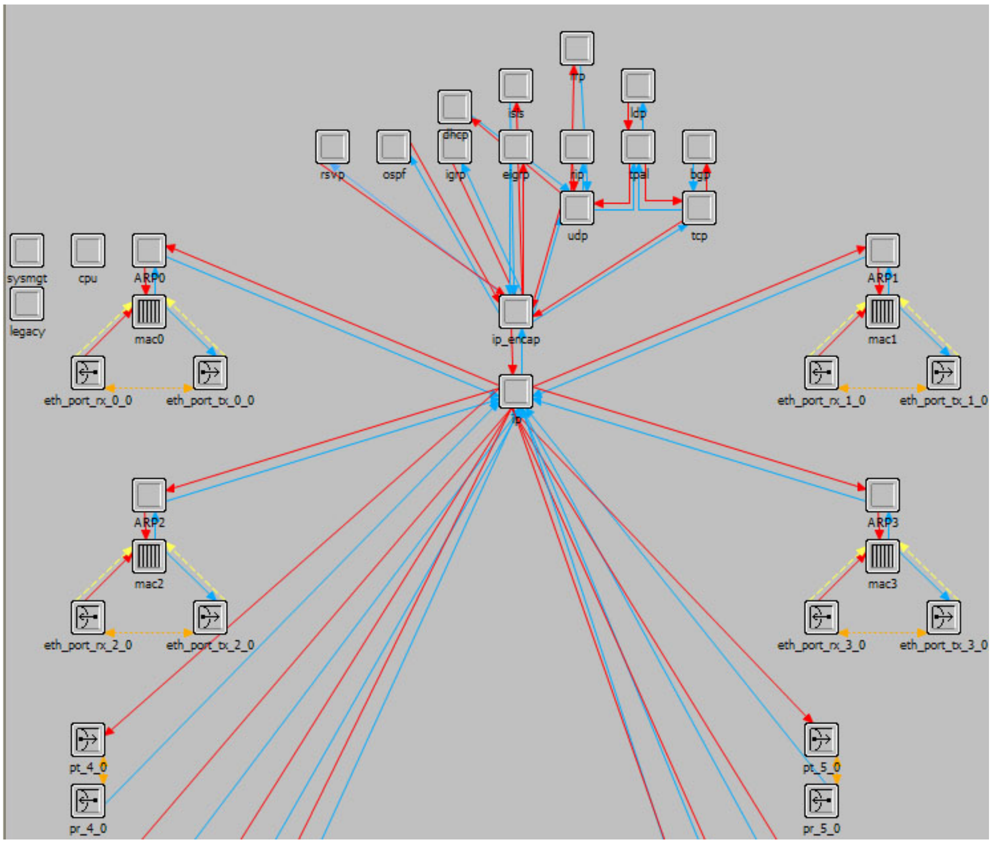


Figure 27 - Node level view of a router in OPNET model

3.2.4 Process Model

Process models [Figure 28] (created using the process editor) are used to describe the logic flow and behavior of processor and queue modules. Communication between processes is supported by interrupts.

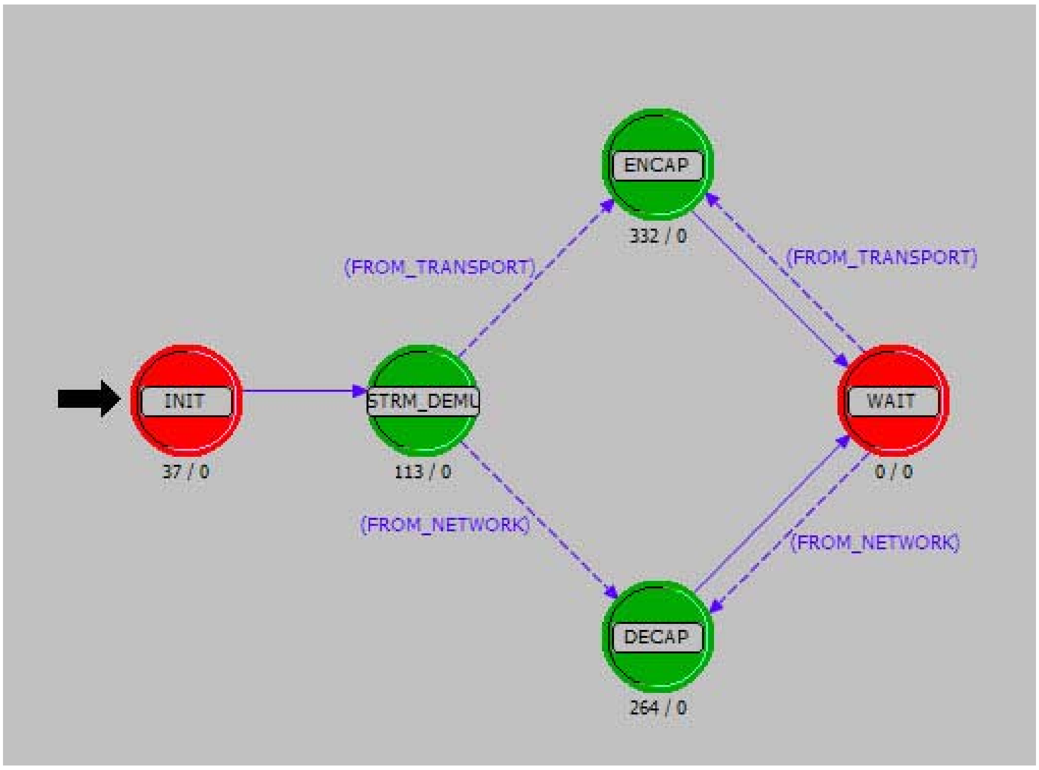


Figure 28 - Process state diagram in OPNET

Process models are expressed in a language called Proto-C [Figure 29], which consists of state transition diagrams (STDs), a library of kernel procedures, and the standard C programming language.

```
/* Declarations must always be present at the top of the block. */

Objid               own_objid, own_node_objid, neighbor_mod_objid;
List                proc_record_handle_list;
int                 record_handle_list_size, i;
OmsT_Pr_Handle      process_record_handle;
char                protocol_name [64];
int                 input_strm, output_strm;
Boolean             inet_address_supported;
char                gateway_str [128];

/** This state is used to find out which modules are        **/
/** directly connected to ip_encap. The stream indices are  **/
/** stored together with the protocol attributes for        **/
/** demultiplexing different higher-layer protocols.        **/

/* Find the object IDs for use in process discovery.        */
own_objid = op_id_self ();
own_node_objid = op_topo_parent (own_objid);
```

Figure 29 - Logic definition with C

The OPNET Process Editor uses a powerful state-transition diagram approach to support specification of any type of protocol, resource, application, algorithm, or queueing policy. States and transitions graphically define the progression of a process in response to events. Within each state, general logic can be specified using a library of predefined functions and even the full flexibility of the C language. Process may create new processes (child process) to perform sub-tasks and thus is called the parent process.

3.3 RUNNING SIMULATION

After defining all the models of the network system, we can exercise them in a dynamic simulation in order to study system performance and behavior. Generally, there are three steps for simulations executation and information collection:

- **Specifying Data Collection:** Model developers always need to decide which information should be extracted from the simulation, such as application-specific statistics, behavioral characterizations, and sometimes application-specific visualization. These can take on several different forms including visual animations, time-dependent series of values (vector), and parametric relationships (scalar).

- **Simulation Construction:** OPNET simulations are obtained by executing a simulation program, which is an executable file in the host computer's filesystem.

- **Simulation Execution:** Simulation execution is the final step in an "iteration" of a modeling experiment. In general, based on the results observed during this step, 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. This section introduces concepts, techniques, and features that support simulation execution.

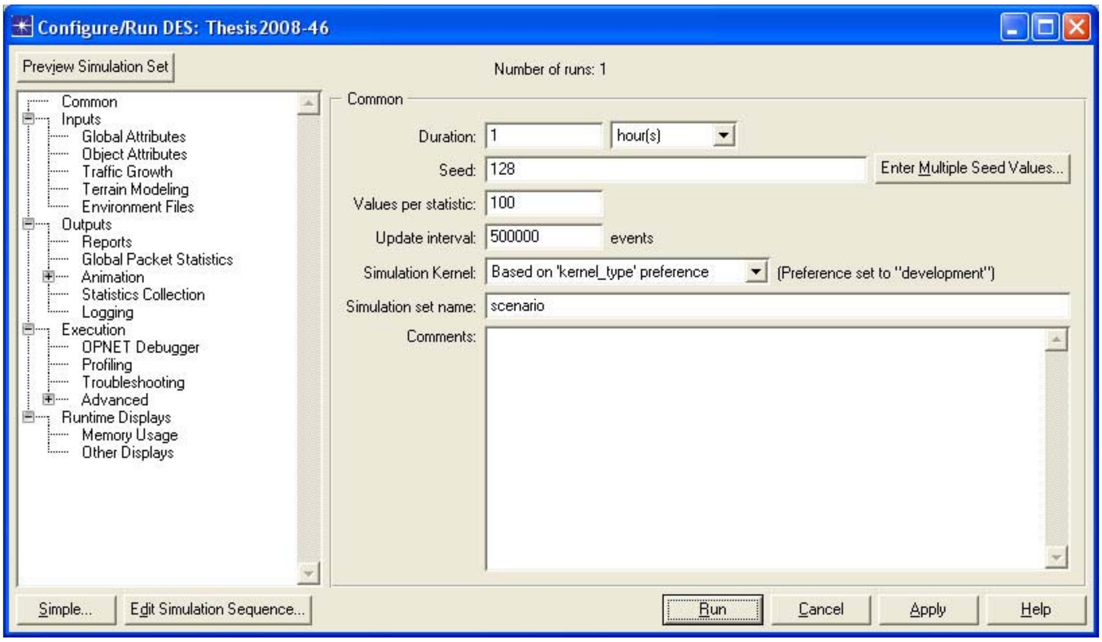


Figure 30 - Obtainable simulation resuls in OPNET

OPNET simulations can be run independently from the OPNET graphical tool by using the op_runsim utility program. However, simulations can be run from the Simulation Tool [Figure 30] within OPNET, which offers the convenience of a graphical interface. The Simulation Tool provides the following services:

- Specification of simulation sequences consisting of an ordered list of simulations and associated attribute values,
- Execution of simulation sequences,
- Storage of simulation sequences in files for later use [1].

CHAPTER 4

MODEL DESIGN

4.1 SCENARIO

In this thesis we have designed an enterprise network scenario consisting of 6 LANs connected to each other as shown in [Figure 31]. A Main Site, located in the center, with the server inside and 5 peripheral sites that have almost identical network model and in the same distance with the main site. All site directors and administrators are aware of IPv6 but they have different future plans for various reasons. Main site is Site Z and peripherals are Site A through E respectively. The ideas of site manegers for the future are as follows:

Site A administrators think that transition to IPv6 will take much longer than todays networkers think and making extra payments to the network infrastructure is not urgent. They say “Don’t tuch anything until it malfunctions”. All routers and hosts are IPv4 only and for Interior Gateway Protocol (IGP), Router Information Protocol (RIP) is used.

As many network authorities accept that a seamless transform from IPv4 to IPv6 is hard to imagine. So administrators in Site B think that all infrastructure within their site must be upgraded to IPv6 but because of budgeting problems no further payments (such as upgrading their portion of core internet) is available. Therefore some mechanisms called IPv6 transitions will occur between their site and main site. For a seamless IPv6 traffic the easiest mechanism is manuel tunnelling. Manuel tunneling allows IPv6 traverse within Site B and Site Z. It is cheap and easy to configure. Manuel tunnels are well on small traffic volume networks and they do not have multipoint connections.

Both network administrators and directors in Site C are well agreed on updating all infrastructure to IPv6 but directors insist on conserving the current situation as well. Their all hardware is IPv6 enabled and no further payment is needed. They want to apply one of the easiest way of transition that is dual stack mechanism. All routers support IPv4/v6 and some clients are IPv6, some others are IPv4.

Directors in Site D have significant doubts on upgrading to IPv6 but they also want to try the new generation protocol. Most of the network infrastructure will still remain IPv4 but 2 routers will be IPv6 enabled. Their internet backbone will remain IPv4 as well. Some routers will be dual stack enabled and automatic tunnelling mechanisms will be in trial period. For tunnelling mechanism 6to4 will be used. They also want to try IPv6 intra-tunnelling within their site. Clients will be IPv6 enabled.

The prospective people of Site E are looking forward to the benefits of IPv6 and they are sure to overcome all problems of it. So they will purely transfer all their hardware to next generation IP. Furthermore they want to upgrade their internet to IPv6 as well. Applications can handle the new protocol and are ready to use. They are still working on killer application (A software substantially increases sales of the platform that supports [23]) of IPv6.

In the main Site E there aren’t any problems in communication with the current and future networks. Routers are in dual stack mode, tunnelling mechanisms are supported on every kind of them. Server is also IPv6 enabled and all tests have been passed seamlessly. Not a significant complaint is reported.

All sites will have FTP, HTTP and e-mail communication among the edge sites and main site. For a comparison of IPv4 and IPv6, different sizes of data are generated in each scenario.

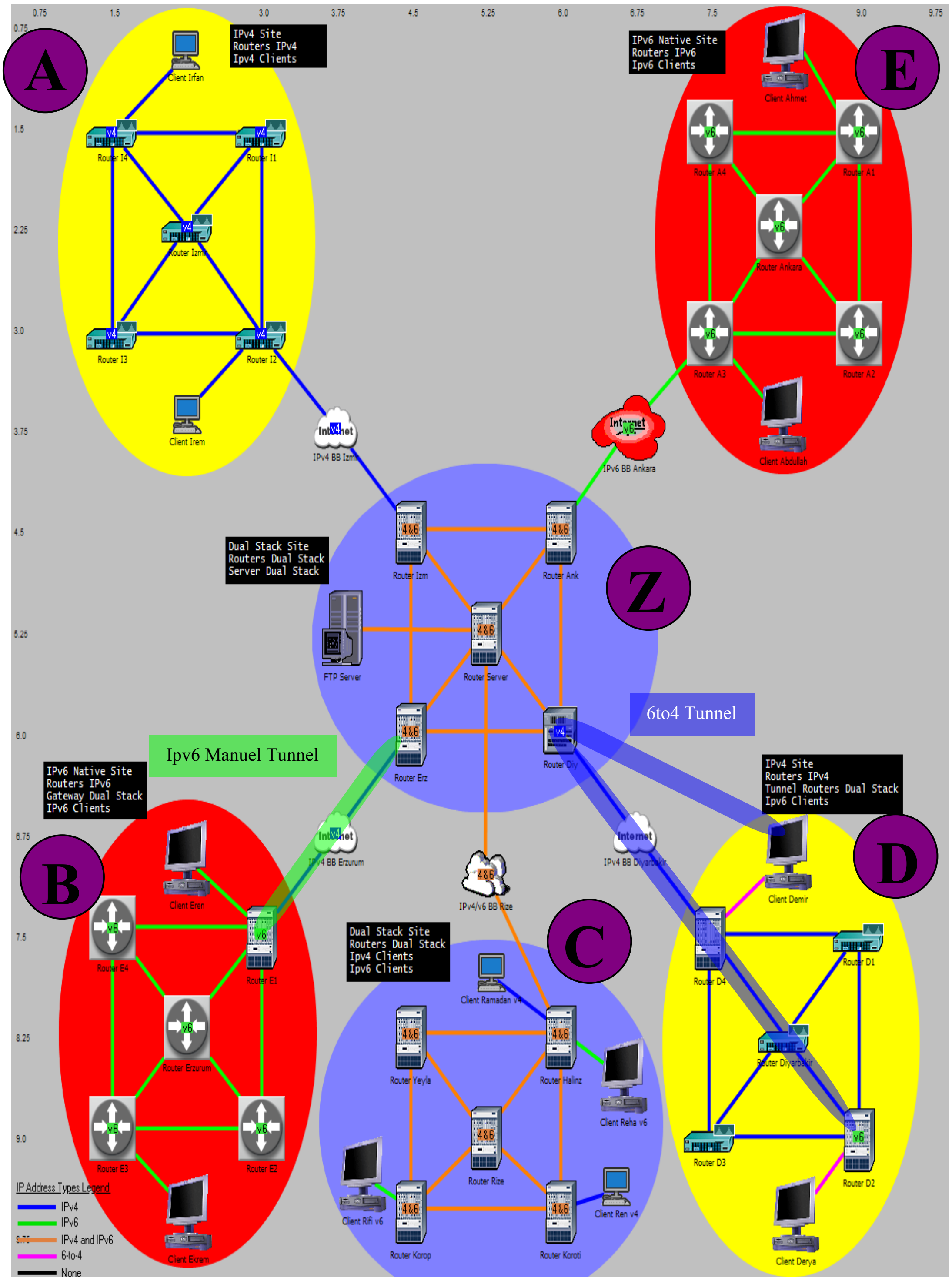


Figure 31 - Test network diagram

4.2 METHODOLOGY

The computer used for this thesis is a Toshiba A200-1BP Laptop computer with Intel Centrino Duo 2.0 CPU, 2 GB of Ram, 160 GB of HDD, 15.4" of screen. OPNET 12.0.A PL5 (Build 4522) is used as the simulation environment. The sites communicating with main site are using identical network elements and deployment of the elements are identical as well. Configuration of elements are different in order to achieve scenario objectives. In this network FTP, HTTP and e-mail traffic takes place simultaneously. Interarrival time of the requests are in normal distrubution. In this study we focused on 4 phases. These are:

- Build up the network test-bed diagram according to the needs. 6 sites from A through E and the main site Z are established.

Site A is the usual IPv4 site. Sites B, C, D are the sites that are using the transition mechanisms. Site B uses Manual IPv6 tunneling, Site C is using dual-stack and Site D is using 6to4 automatic tunneling mechanism. Site E is the next generation network that is IPv6. For the central site we have dual stack routers and FTP server.

- Using ICMPv6/v4 pings to check the correct functionality of the network. In this phase every client pings the FTP Server [Figure 32] according to its configured protocol (IPv6 or IPv4). Trace route tool of OPNET is used.

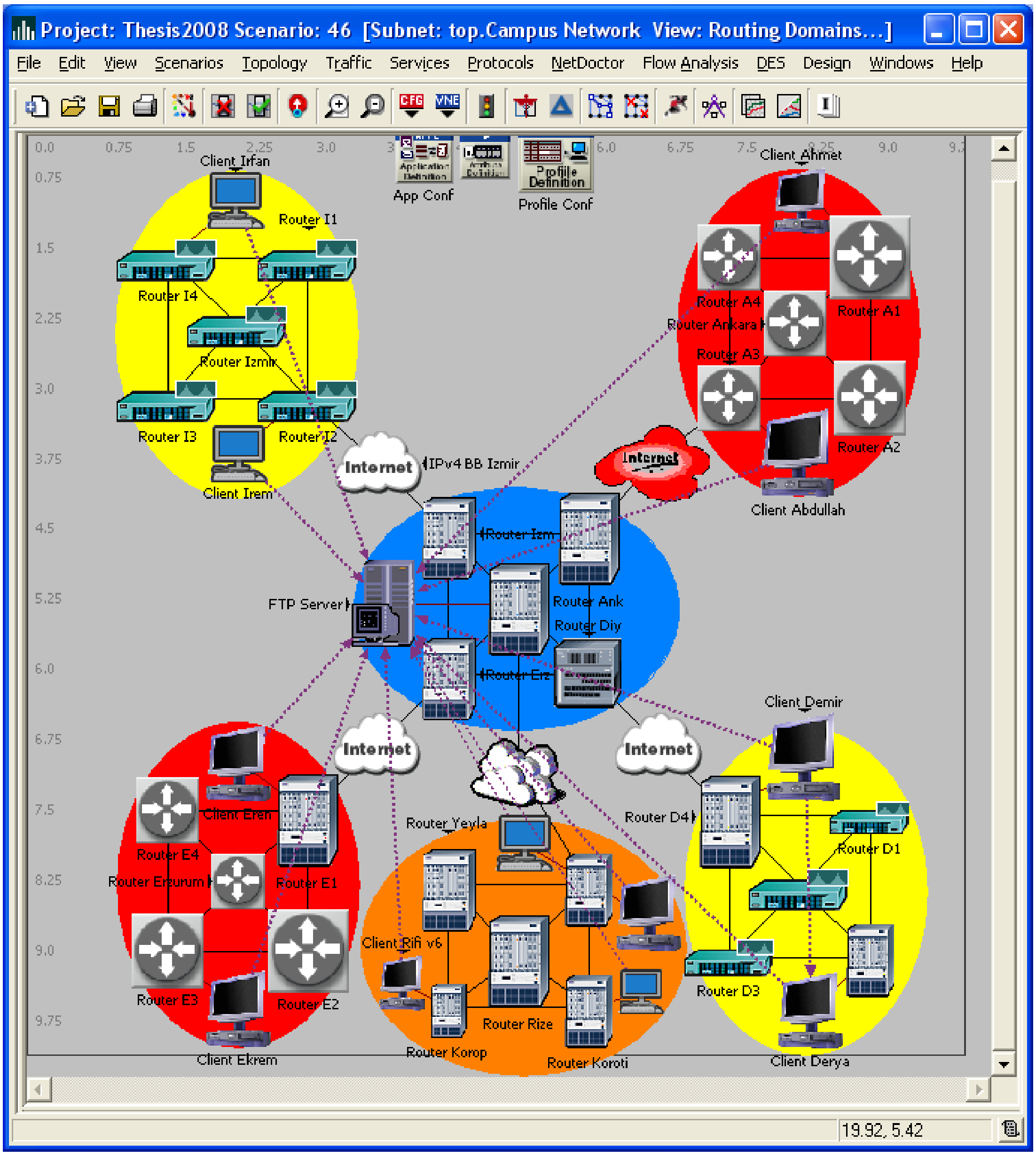


Figure 32 - Ping test of whole network

- As the main purpose of this thesis, the comparison of the IP protocols takes place in the third phase. All clients throughputs and end-to-end delays are tested in 12 scenarios (Each scenario has the same network model. The requested file size from FTP server varies in each scenario) with varying data sizes (46, 64, 256, 1024, 1280, 1460, 1480, 1500 bytes). We also tested the protocols in bigger data sizes as 10kBytes, 100kBytes, 1MBytes and 1GBytes respectively.

- In the last phase transition mechanisms performances are tested. Dual stacked, manual IPv6 tunnelled and automatic 6to4 tunnelled IP traffic is tested and compared with native IPv6 and IPv4.

4.2.1 Build up the network test-bed diagram

4.2.1.1 Site A

This site is an IPv4 only Site and has the usual configuration of nowadays routers and clients. No extra effort or configuration upgrade is needed. Network diagram is in [Figure 33]. Configurations of this network are in [Table 6].

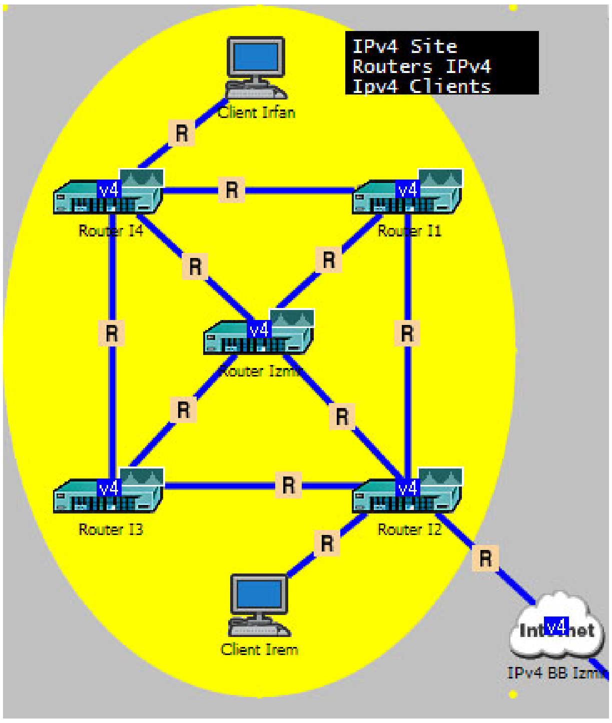


Figure 33 - Model Site A

Unit	IPv4	IPv6	Transition	Router Protocol
Client Irfan	+	-	-	-
Client Irem	+	-	-	-
Router I1	+	-	-	RIP
Router I2	+	-	-	RIP
Router I3	+	-	-	RIP
Router I4	+	-	-	RIP
Router Izmir	+	-	-	RIP
BB Izmir	+	-	-	RIP

Table 6 - Site A configurations

4.2.1.2 Site B

This site is a native IPv6 site with manuel IPv6 tunnelling. The backbone connecting to main site is IPv4. For an IPv6 conversation with main site, tunnelling is needed. Network diagram is in [Figure 34]. Configurations of this network are in [Table 7].

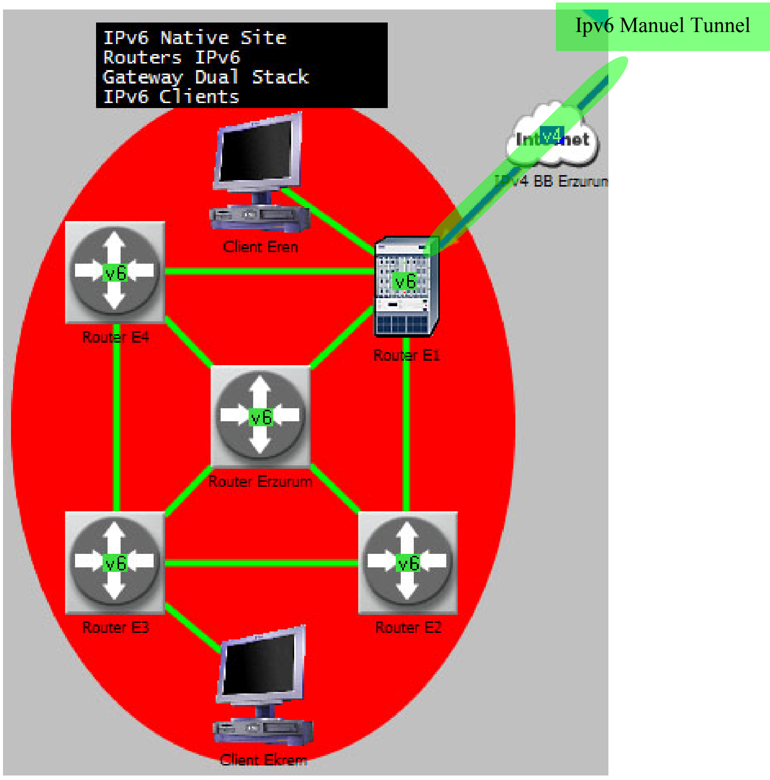


Figure 34 - Model Site B

Unit	IPv4	IPv6	Transition	Router Protocol
Client Eren	-	+	-	-
Client Ekrem	-	+	-	-
Router E1	+	+	Manuel Tunnel	RIP / RIPv6
Router E2	-	+	6to4	RIPv6
Router E3	-	+	-	RIPv6
Router E4	-	+	6to4	RIPv6
Router Erzurum	-	+	-	RIPv6
BB Erzurum	+	-	-	RIP

Table 7 - Site B configurations

The manual tunnelling configuration is in [Figure 35]. In IPv4 configuration made manually on both the tunnel-ends. Tunnel destination must be set to the other tunnel end IPv4 address. The configuration of routing protocol of the tunnel is not set. IPv4 is used as the link layer [Figure 36] of IPv6 through the tunnel.

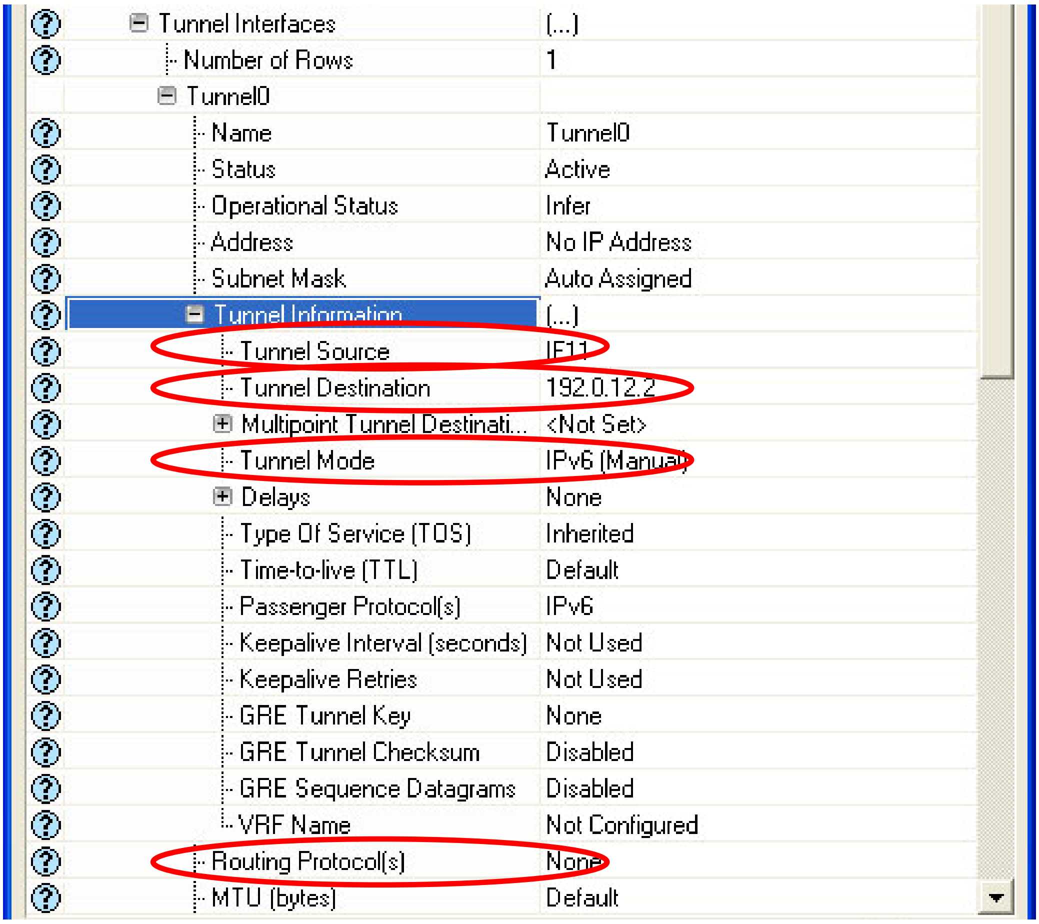


Figure 35 - IPv4 configuration of manuel tunneling

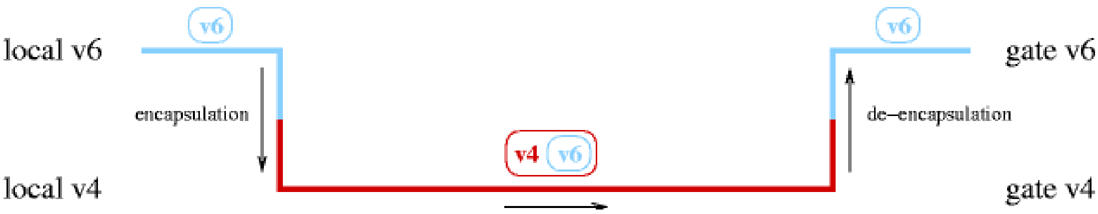


Figure 36 - IPv6 packet encapsulated in IPv4 packet

In IPv6 configuration [Figure 37] of tunnel ends, it is important to configure the carrying IPv6 protocol. In our model RIPng is configured for IPv6 passenger protocol. Both ends have IPv6 addresses but they are not configured on the interfaces of router. It’s a virtual link.

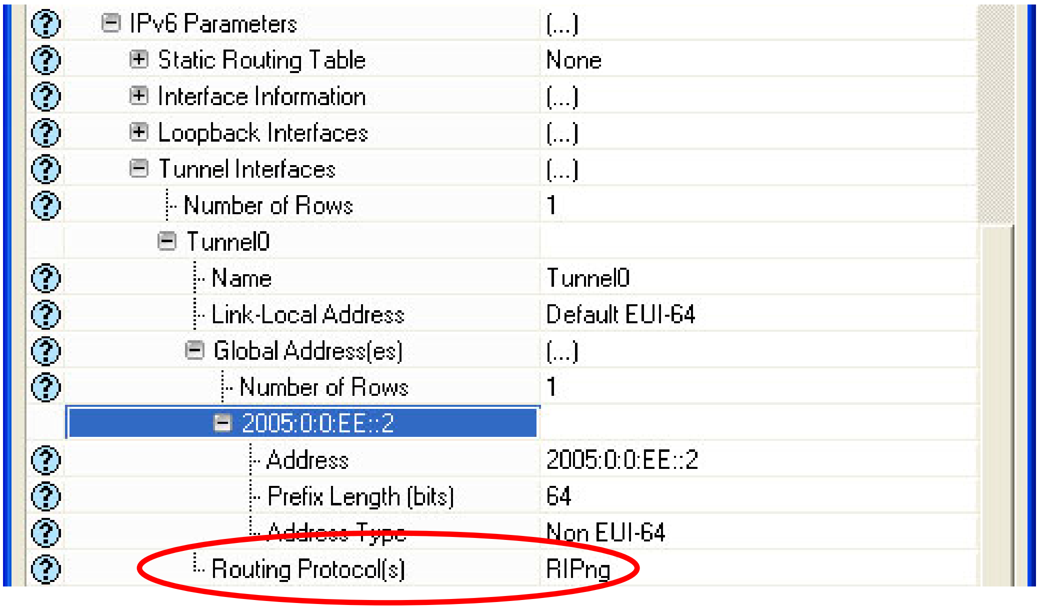


Figure 37 - IPv6 configuration of manuel tunneling

4.2.1.3 Site C

This site is a dual stack site. The backbone connecting to main site is dual stack as well. Network diagram is in [Figure 38]. Configurations of this network are in [Table 8].

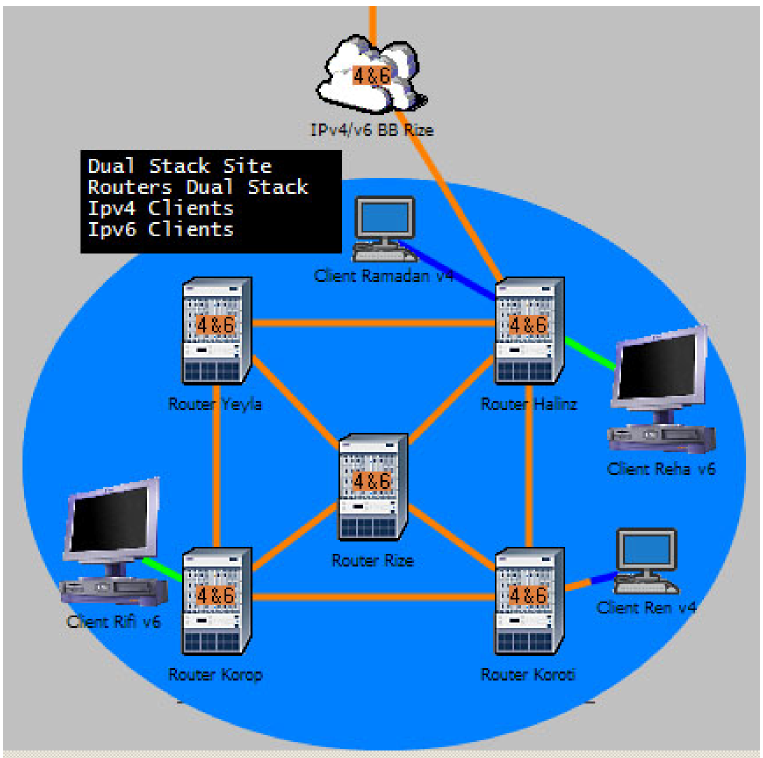


Figure 38 - Model Site C

Unit	IPv4	IPv6	Transition	Router Protocol
Client Ramadan	+	-	-	-
Client Reha	-	+	-	-
Client Ren	+	-	-	-
Client Rifi	-	+	-	-
Router Halinz	+	+	Dual Stack	RIP / RIPv6
Router Koroti	+	+	Dual Stack	RIP / RIPv6
Router Korop	+	+	Dual Stack	RIP / RIPv6
Router Yeyla	+	+	Dual Stack	RIP / RIPv6
Router Rize	+	+	Dual Stack	RIP / RIPv6
BB Rize	+	+	Dual Stack	RIP / RIPv6

Table 8 - Site C configurations

4.2.1.4 Site D

This site is a 6to4 Site. Actually it is IPv4 site but with dual stack routers on tunnel ends. Network diagram is in [Figure 39]. Configurations of this network are in [Table 9]. Tunnel end routers have double stack configurations, other routers are IPv4 only.

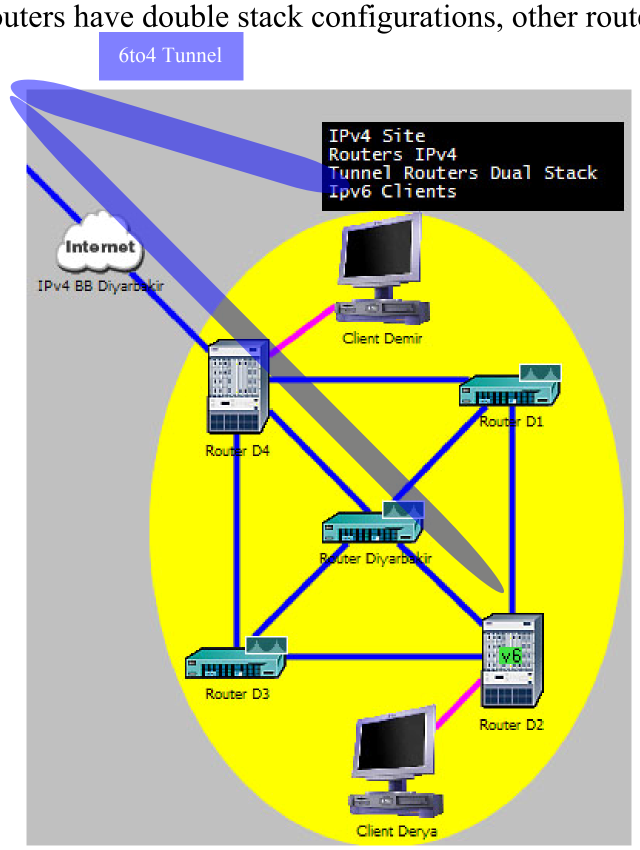


Figure 39 - Model Site D

Unit	IPv4	IPv6	Transition	Router Protocol
Client Demir	+	-	-	-
Client Derya	+	-	-	-
Router D1	+	-	-	RIP
Router D2	+	+	6to4	RIP / RIPv6
Router D3	+	-	-	RIP
Router D4	+	+	6to4	RIP / RIPv6
Router Diyarbakir	+	-	-	RIP
BB Diyarbakir	+	-	-	RIP

Table 9 - Site D configurations

In IPv4 configuration of 6to4 tunneling [Figure 40], no tunnel destination is set. For pessenger protocol, IPv4 is used.

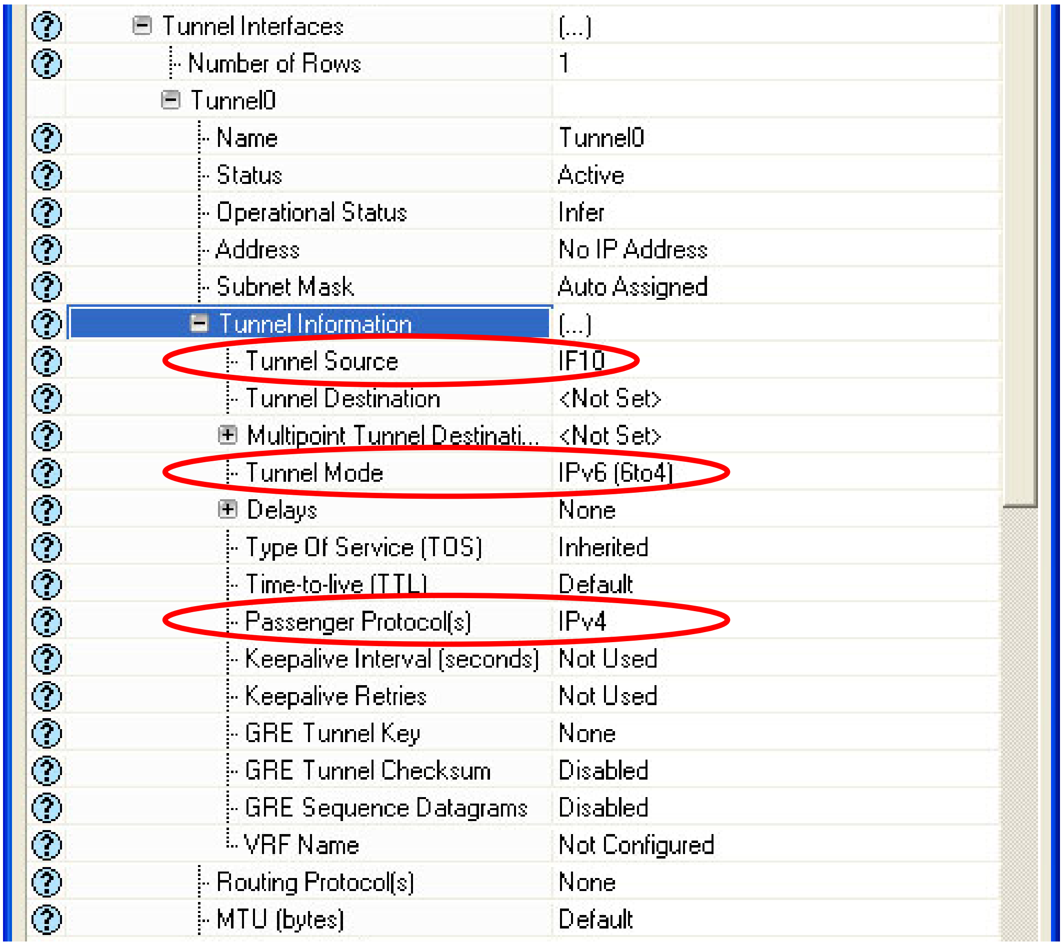


Figure 40 - IPv4 configuration of 6to4 tunneling

For IPv6 configuration of a 6to4 tunnel [Figure 41], static routing is needed. For a virtual IPv6 link each end must have a real IPv6, but this IP doesn't appear in the router interfaces.

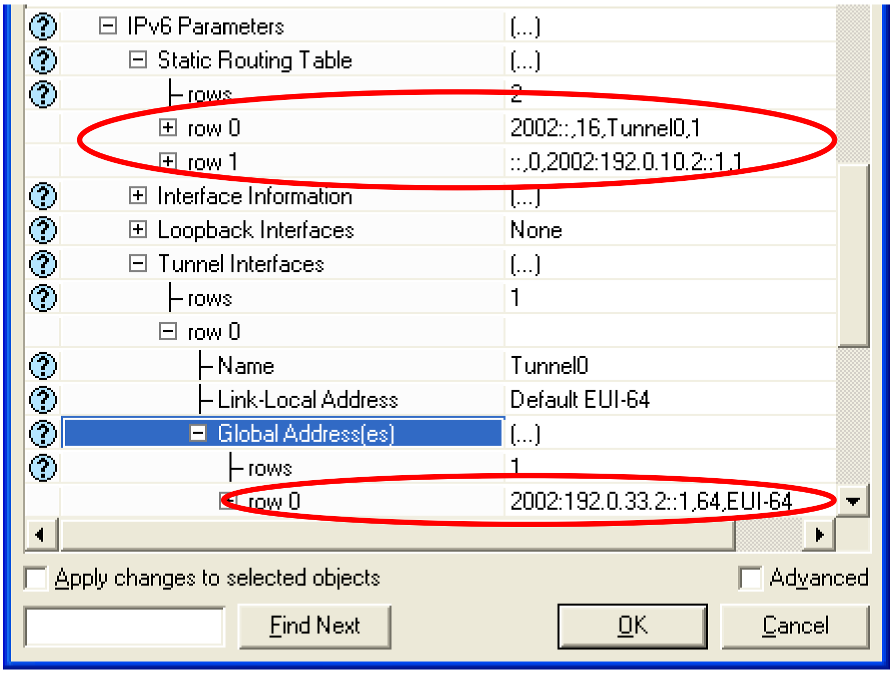


Figure 41 - IPv6 static routing and 6to4 tunnel configuration

4.2.1.5 Site E

This is a native IPv6 Site. This site and the backbone connecting to main site are native IPv6. Network diagram is in [Figure 42]. Configurations of this network are in [Table 10]. All nodes are native IPv6.

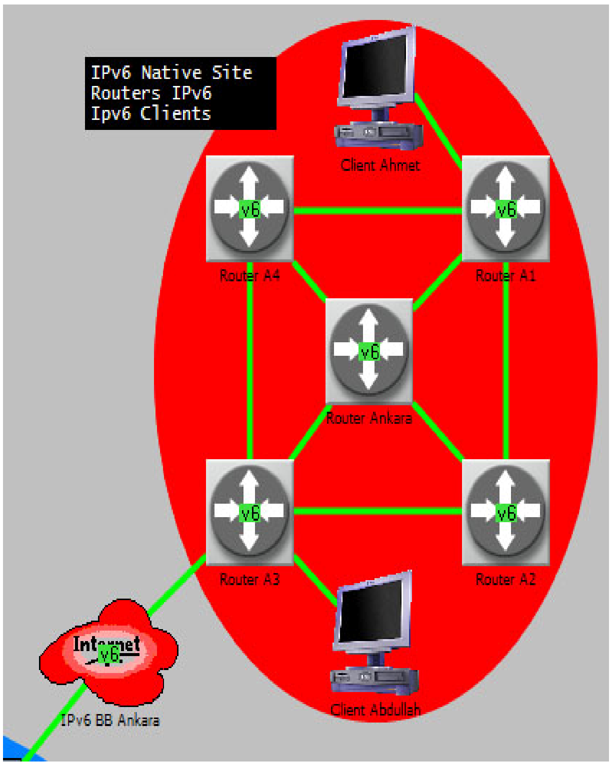


Figure 42 - Model Site E

Unit	IPv4	IPv6	Transition	Router Protocol
Client Ahmet	-	+	-	RIPv6
Client Abdullah	-	+	-	RIPv6
Router A1	-	+	-	RIPv6
Router A2	-	+	-	RIPv6
Router A3	-	+	-	RIPv6
Router A4	-	+	-	RIPv6
Router Ankara	-	+	-	RIPv6
BB Ankara	-	+	-	RIPv6

Table 10 - Site E configurations

4.2.1.6 Site Z

This site is the main site. Dual stack is used in order to communicate with whole sites. Some static routes to the 6to4 site are used. Network diagram is in [Figure 43]. Configurations of this network are in [Table 11]. All routers are dual stack.

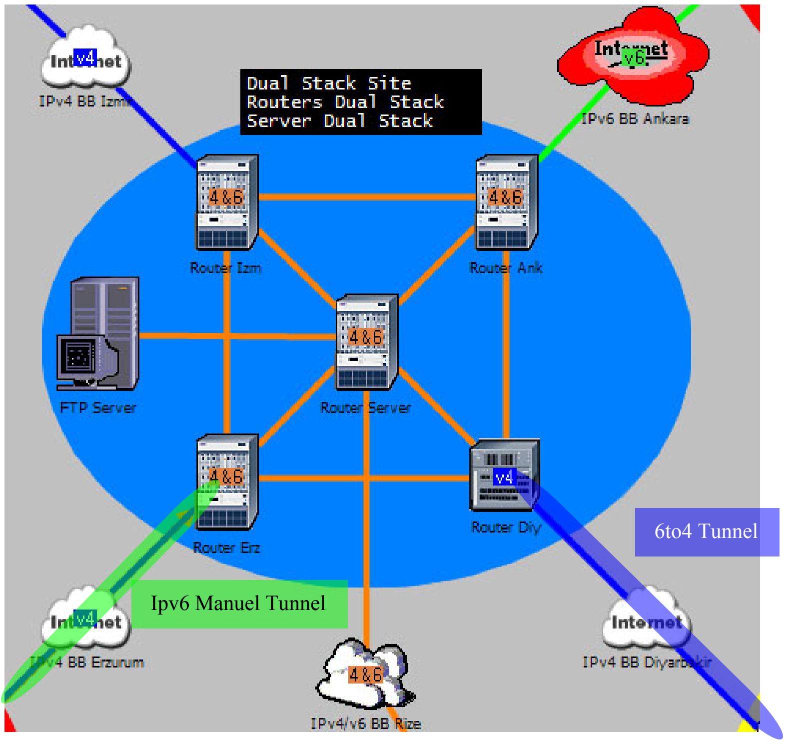


Figure 43 - Model Site Z

Unit	IPv4	IPv6	Transition	Router Protocol
Router Izm	+	+	Dual Stack	RIP / RIPv6
Router Ank	+	+	Dual Stack	RIP / RIPv6
Router Diy(Relay)	+	+	6to4	RIP / RIPv6
Router Erz	+	+	Manuel	RIP / RIPv6
Router Server	+	+	Dual Stack	RIP / RIPv6
FTP Server	+	+	-	-

Table 11 - Site Z configurations

CHAPTER 5

TESTS & RESULTS

5.1 CHECKING NETWORK FUNCTIONALITY

Before starting to obtain statistical analysis of IP protocols we checked the connections of the sites by pinging the server in the Main Site Z. Site A used IPv4 ping, sites B, D, E used IPv6 ping, Site C used both IPv4 and IPv6 ping to test the connection. In OPNET having run the scenario, ping results can be obtained from the DES Run Tables tab in the results pane.

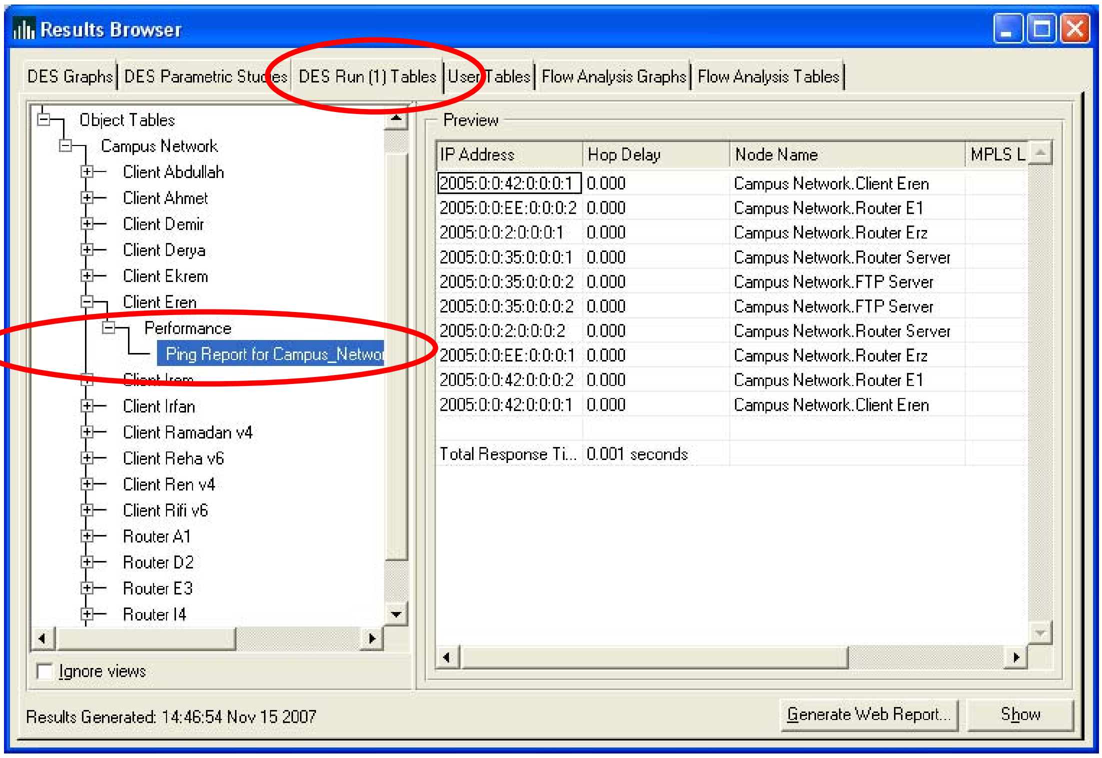


Figure 44 - Obtaining ping report as trace route analysis

5.1.1 Validating the Connectivity of Site A

Because Site A is only IPv4 we used IPv4 ping to test the connection. In trace route analysis of Client Irfan is able to reach FTP Server [Figure 45].

Performance.Ping Report for Campus_Network_FTP_Serv...

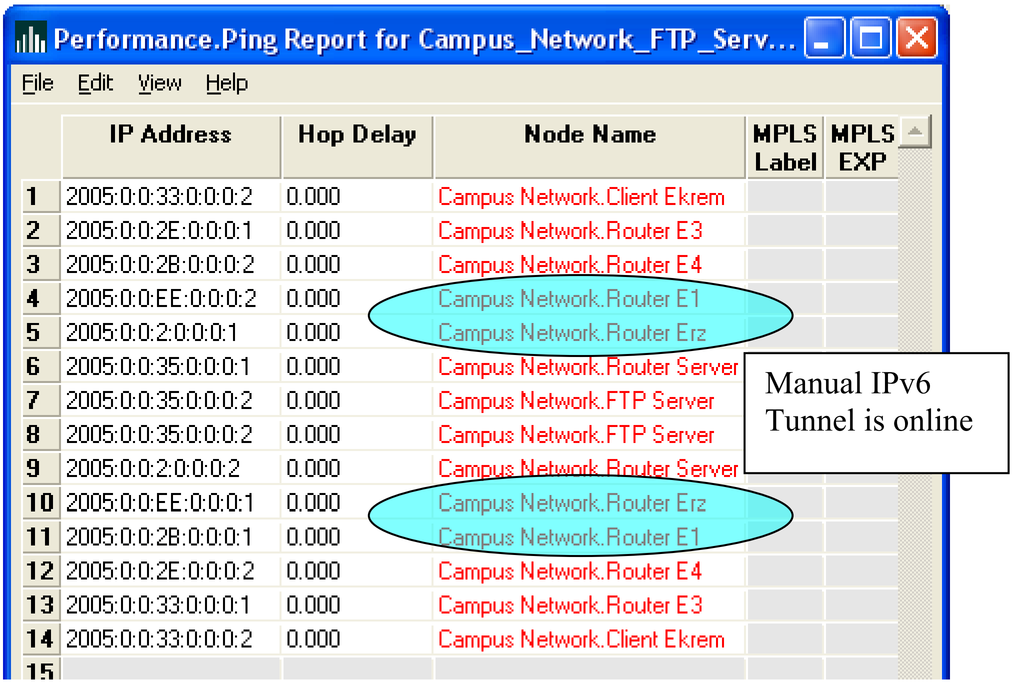
File Edit View Help

	IP Address	Hop Delay	Node Name	MPLS Label	MPLS EXP
1	192.0.32.2	0.000	Campus Network.Client Irfan		
2	192.0.29.1	0.000	Campus Network.Router I4		
3	192.0.31.2	0.000	Campus Network.Router I1		
4	192.0.14.1	0.000	Campus Network.Router I2		
5	192.0.11.1	0.000	Campus Network.IPv4 BB Izmir		
6	192.0.1.1	0.000	Campus Network.Router Izm		
7	192.0.53.1	0.000	Campus Network.Router Server		
8	192.0.53.2	0.000	Campus Network.FTP Server		
9	192.0.53.2	0.000	Campus Network.FTP Server		
10	192.0.1.2	0.000	Campus Network.Router Server		
11	192.0.11.2	0.000	Campus Network.Router Izm		
12	192.0.14.2	0.000	Campus Network.IPv4 BB Izmir		
13	192.0.26.2	0.000	Campus Network.Router I2		
14	192.0.27.1	0.000	Campus Network.Router Izmir		
15	192.0.32.1	0.000	Campus Network.Router I4		
16	192.0.32.2	0.000	Campus Network.Client Irfan		
17					

Figure 45 – IPv4 ping and trace route of Site-A

5.1.2 Validating the Connectivity of Site B

In site B we have IPv6 hosts only. The backbone connecting to main site is IPv4. The ping used to verify the connection is an ICMPv6 ping. The trace route analysis for Client Ekrem (IPv6) is in [Figure 46].



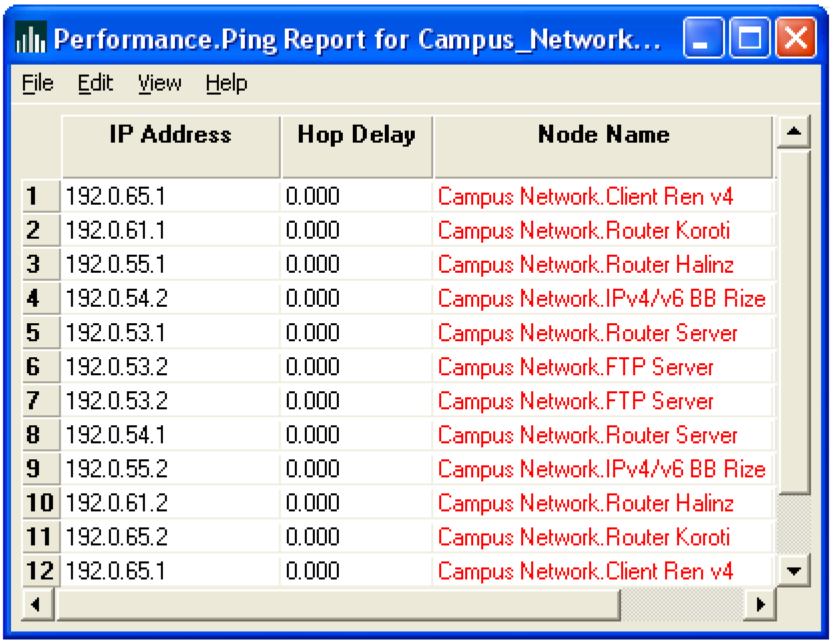
	IP Address	Hop Delay	Node Name	MPLS Label	MPLS EXP
1	2005:0:0:33:0:0:0:2	0.000	Campus Network.Client Ekrem		
2	2005:0:0:2E:0:0:0:1	0.000	Campus Network.Router E3		
3	2005:0:0:2B:0:0:0:2	0.000	Campus Network.Router E4		
4	2005:0:0:EE:0:0:0:2	0.000	Campus Network.Router E1		
5	2005:0:0:2:0:0:0:1	0.000	Campus Network.Router Erz		
6	2005:0:0:35:0:0:0:1	0.000	Campus Network.Router Server		
7	2005:0:0:35:0:0:0:2	0.000	Campus Network.FTP Server		
8	2005:0:0:35:0:0:0:2	0.000	Campus Network.FTP Server		
9	2005:0:0:2:0:0:0:2	0.000	Campus Network.Router Server		
10	2005:0:0:EE:0:0:0:1	0.000	Campus Network.Router Erz		
11	2005:0:0:2B:0:0:0:1	0.000	Campus Network.Router E1		
12	2005:0:0:2E:0:0:0:2	0.000	Campus Network.Router E4		
13	2005:0:0:33:0:0:0:1	0.000	Campus Network.Router E3		
14	2005:0:0:33:0:0:0:2	0.000	Campus Network.Client Ekrem		
15					

Figure 46 – IPv6 ping and trace route of Site B

As shown in [Figure 46] the next hop of Router E1 is Router Erz. So pinging FTP Server is successful and manual tunnelling is working.

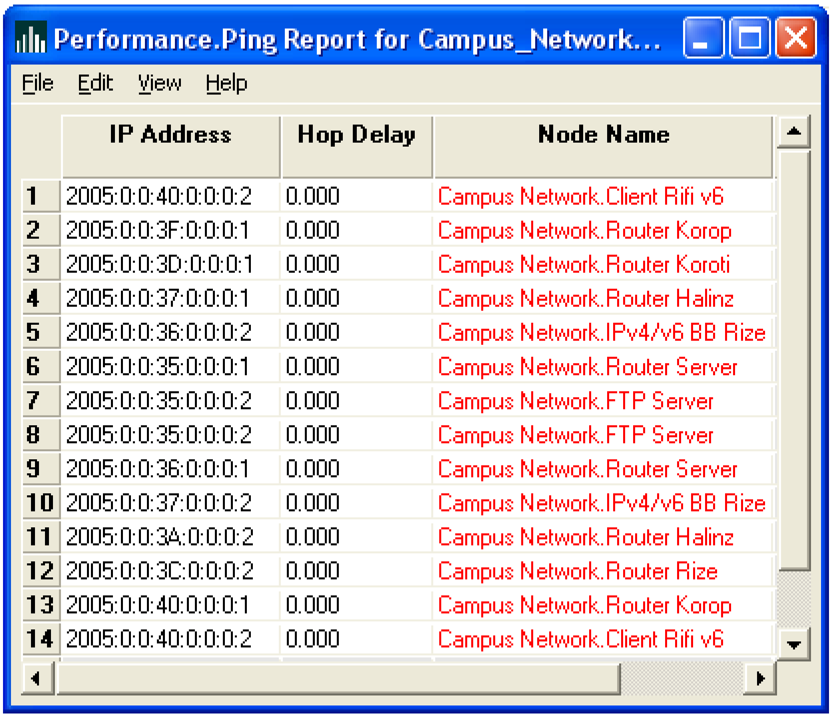
5.1.3 Validating the Connectivity of Site C

Due to the dual stack status of the site C we used both IPv4 and IPv6 ping to verify dual stackness. Pings are successful and connections are established [Figure 47, 48].



	IP Address	Hop Delay	Node Name
1	192.0.65.1	0.000	Campus Network.Client Ren v4
2	192.0.61.1	0.000	Campus Network.Router Koroti
3	192.0.55.1	0.000	Campus Network.Router Halinz
4	192.0.54.2	0.000	Campus Network.IPv4/v6 BB Rize
5	192.0.53.1	0.000	Campus Network.Router Server
6	192.0.53.2	0.000	Campus Network.FTP Server
7	192.0.53.2	0.000	Campus Network.FTP Server
8	192.0.54.1	0.000	Campus Network.Router Server
9	192.0.55.2	0.000	Campus Network.IPv4/v6 BB Rize
10	192.0.61.2	0.000	Campus Network.Router Halinz
11	192.0.65.2	0.000	Campus Network.Router Koroti
12	192.0.65.1	0.000	Campus Network.Client Ren v4

Figure 47 - IPv4 ping of dual stack Site C

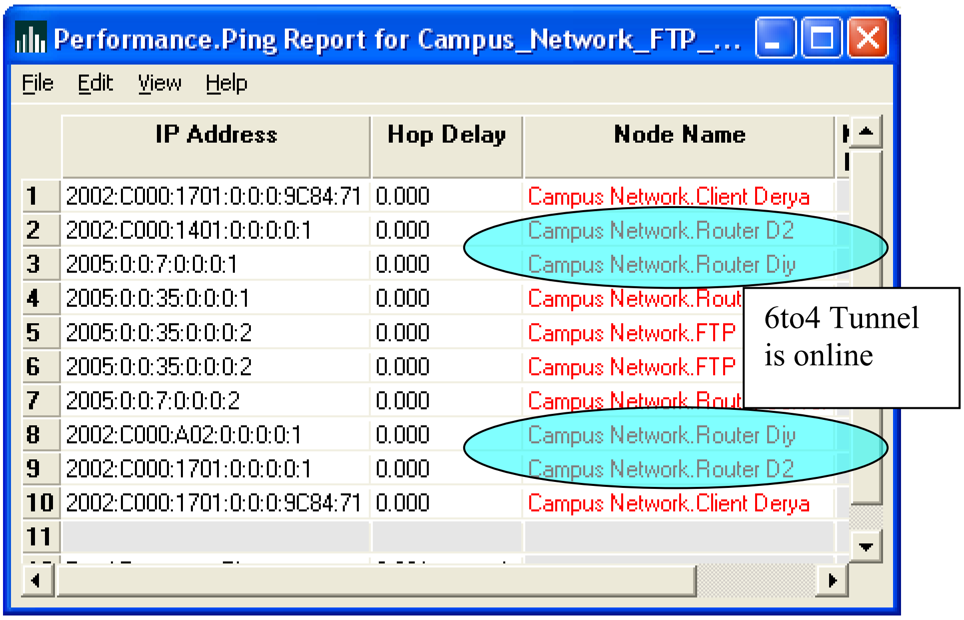


	IP Address	Hop Delay	Node Name
1	2005:0:0:40:0:0:0:2	0.000	Campus Network.Client Rifi v6
2	2005:0:0:3F:0:0:0:1	0.000	Campus Network.Router Korop
3	2005:0:0:3D:0:0:0:1	0.000	Campus Network.Router Koroti
4	2005:0:0:37:0:0:0:1	0.000	Campus Network.Router Halinz
5	2005:0:0:36:0:0:0:2	0.000	Campus Network.IPv4/v6 BB Rize
6	2005:0:0:35:0:0:0:1	0.000	Campus Network.Router Server
7	2005:0:0:35:0:0:0:2	0.000	Campus Network.FTP Server
8	2005:0:0:35:0:0:0:2	0.000	Campus Network.FTP Server
9	2005:0:0:36:0:0:0:1	0.000	Campus Network.Router Server
10	2005:0:0:37:0:0:0:2	0.000	Campus Network.IPv4/v6 BB Rize
11	2005:0:0:3A:0:0:0:2	0.000	Campus Network.Router Halinz
12	2005:0:0:3C:0:0:0:2	0.000	Campus Network.Router Rize
13	2005:0:0:40:0:0:0:1	0.000	Campus Network.Router Korop
14	2005:0:0:40:0:0:0:2	0.000	Campus Network.Client Rifi v6

Figure 48 - IPv6 ping of dual stack Site C

5.1.4 Validating the Connectivity of Site D

In this site in order to be able to ping the FTP Server we need IPv6 ping and 6to4 tunnel must work. The result is in [Figure 49].



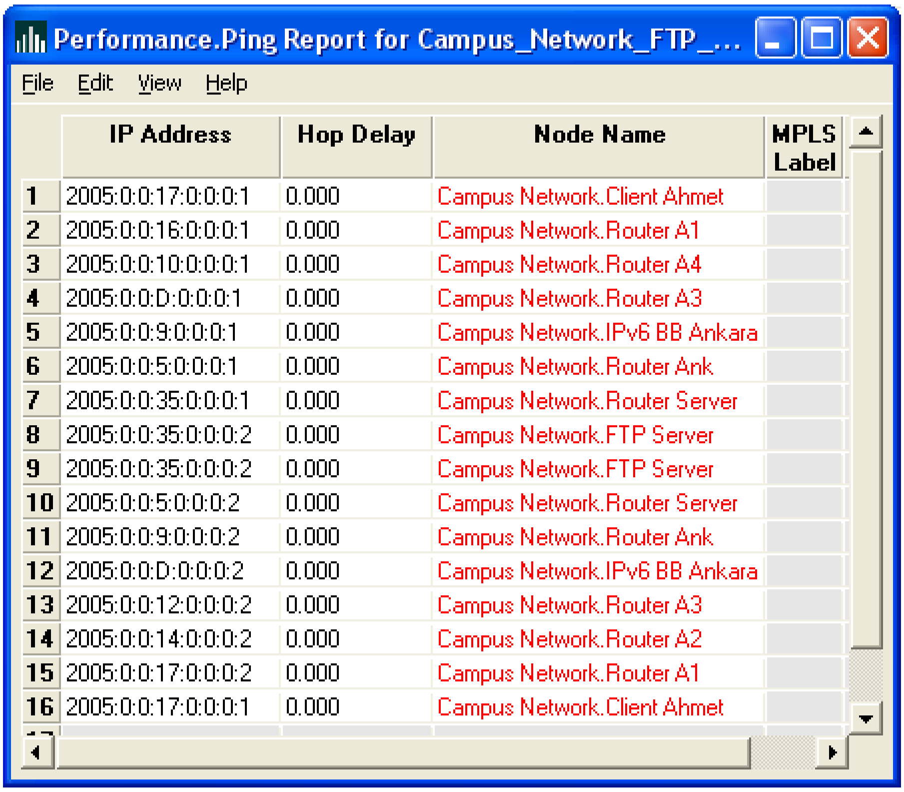
	IP Address	Hop Delay	Node Name
1	2002:C000:1701:0:0:0:9C84:71	0.000	Campus Network.Client Derya
2	2002:C000:1401:0:0:0:0:1	0.000	Campus Network.Router D2
3	2005:0:0:7:0:0:0:1	0.000	Campus Network.Router Diy
4	2005:0:0:35:0:0:0:1	0.000	Campus Network.Rout
5	2005:0:0:35:0:0:0:2	0.000	Campus Network.FTP
6	2005:0:0:35:0:0:0:2	0.000	Campus Network.FTP
7	2005:0:0:7:0:0:0:2	0.000	Campus Network.Rout
8	2002:C000:A02:0:0:0:0:1	0.000	Campus Network.Router Diy
9	2002:C000:1701:0:0:0:0:1	0.000	Campus Network.Router D2
10	2002:C000:1701:0:0:0:9C84:71	0.000	Campus Network.Client Derya
11			

Figure 49 - IPv6 ping over 6to4 tunnel of Site D

As we track back the trace route of Client Derya we can see the tunnel ends. There are various routers [Figure 49] between Router D2 (line2, 3; line 8, 9) and Router Diy but because IPv6 is used in 6to4 tunnel mechanism and IPv6 doesn't fragment in the intermediate routers, they look like adjacent routers.

5.1.5 Validating the Connectivity of Site E

Site E is IPv6 native site. Therefore ICMPv6 ping is used. Ping to FTP Server is successful [Figure 50].



	IP Address	Hop Delay	Node Name	MPLS Label
1	2005:0:0:17:0:0:0:1	0.000	Campus Network.Client Ahmet	
2	2005:0:0:16:0:0:0:1	0.000	Campus Network.Router A1	
3	2005:0:0:10:0:0:0:1	0.000	Campus Network.Router A4	
4	2005:0:0:D:0:0:0:1	0.000	Campus Network.Router A3	
5	2005:0:0:9:0:0:0:1	0.000	Campus Network.IPv6 BB Ankara	
6	2005:0:0:5:0:0:0:1	0.000	Campus Network.Router Ank	
7	2005:0:0:35:0:0:0:1	0.000	Campus Network.Router Server	
8	2005:0:0:35:0:0:0:2	0.000	Campus Network.FTP Server	
9	2005:0:0:35:0:0:0:2	0.000	Campus Network.FTP Server	
10	2005:0:0:5:0:0:0:2	0.000	Campus Network.Router Server	
11	2005:0:0:9:0:0:0:2	0.000	Campus Network.Router Ank	
12	2005:0:0:D:0:0:0:2	0.000	Campus Network.IPv6 BB Ankara	
13	2005:0:0:12:0:0:0:2	0.000	Campus Network.Router A3	
14	2005:0:0:14:0:0:0:2	0.000	Campus Network.Router A2	
15	2005:0:0:17:0:0:0:2	0.000	Campus Network.Router A1	
16	2005:0:0:17:0:0:0:1	0.000	Campus Network.Client Ahmet	

Figure 50 - IPv6 ping of native IPv6 Site E

5.1.6 Discussion

In this validation of the link tests we didn't get any destination unreachable echo request. All clients are able to communicate with the FTP Server in the Main Site Z. All tunnels are online and second phase of the objective is accomplished.

5.2 COMPARISON OF IPv4 AND IPv6 TRAFFIC METRICES

5.2.1 Throughput Comparison

The successfully delivered packets over a physical or logical link is called throughput and it is used for the effectiveness analysis in network world. Here in this model we are not concerned about the theoratical throughput of a channel, but the actual or measured one on the links. We got the peak throughput of the network edges.

For the comparison of current networks and the future networks we will examine the Site A and Site E test results. There are 12 different scenarios [Figure 51], (here we are talking about OPNET scenario, not the model scenario. In an OPNET scenario whole network is running with different given parameters.) with various data sizes.

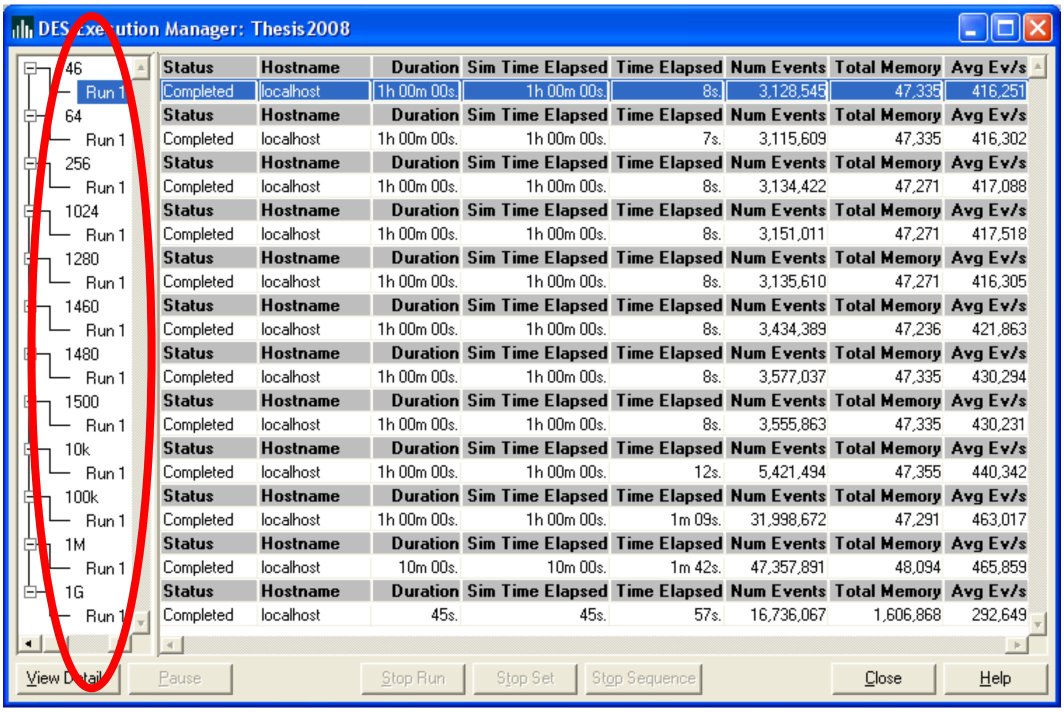


Figure 51 - DES project parameters

All test except the last two are simulated in 1 hour period. Because of enormous RAM consumption, last scenario time held for 45 seconds.

5.2.1.1 Throughput Analysis for 64-Byte Data Size

The results of total throughput in a time average chart for 64-byte data size is in [Figure 52, Table 12].

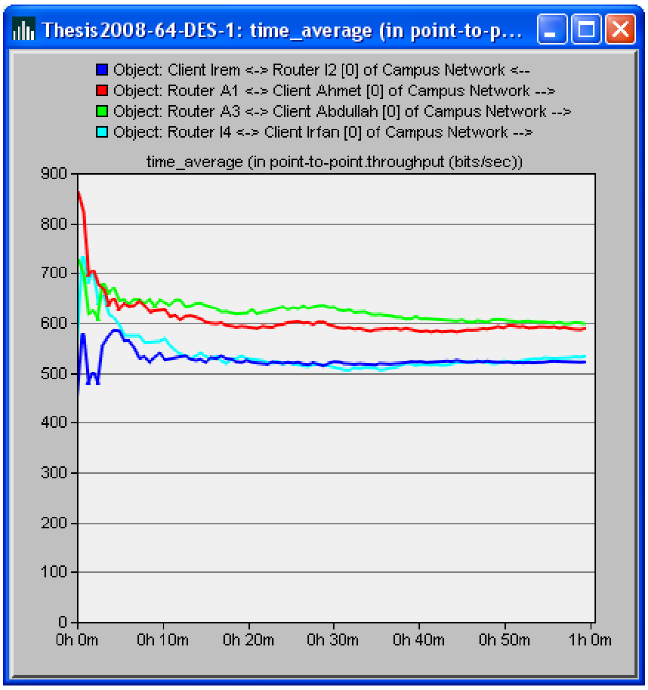


Figure 52 - Total throughput for the 64- byte data size (bits/sec.)

In the 64-byte data size scenario throughputs of IPv6 Clients (Ahmet and Abdullah) [Table 12] performed better performance than IPv4 clients. Peek gain of IPv6 is measured about 18.7 % (622 / 524).

Client	Min	Max	Expected	Confidence Intervals	
				%80	%99
Ahmet (v6)	581	863	606	600<µ<611	595<µ<616
Abdullah (v6)	597	728	622	619<µ<625	616<µ<628
Irem (v4)	451	585	524	521<µ<526	519<µ<528
Irfan (v4)	505	732	536	531<µ<541	526<µ<547

Table 12 - Total throughput for 64-byte data size (bits/sec.)

5.2.1.2 Throughput Analysis for 1280-Byte Data Size

The results of total throughput in a time average chart for 1280-byte data size is in [Figure 53, Table 13].

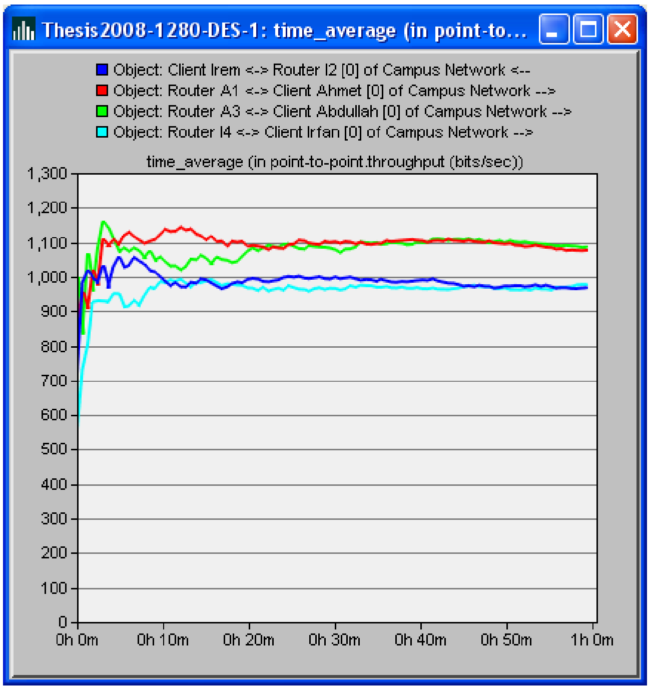


Figure 53 - Total throughput for 1280-byte data size (bits/sec.)

In the 1280-byte data size scenario throughputs of IPv6 Clients (Ahmet and Abdullah) [Table 13] performed better performance than IPv4 clients. Peek gain of IPv6 is measured about 14.1 % (1092 / 957).

Client	Min	Max	Expected	Confidence Intervals	
				%80	%99
Ahmet (v6)	863	1143	1092	1087<μ<1097	1082<μ<1102
Abdullah (v6)	835	1161	1079	1075<μ<1084	1070<μ<1089
Irem (v4)	721	1058	986	981<μ<990	977<μ<994
Irfan (v4)	559	992	957	951<μ<964	944<μ<971

Table 13 - Total throughput for 1280-byte data size (bits/sec.)

5.2.1.3 Throughput Analysis for 1500-Byte Data Size

The results of total throughput in a time average chart for 1500-byte data size is in [Figure 54, Table 14].

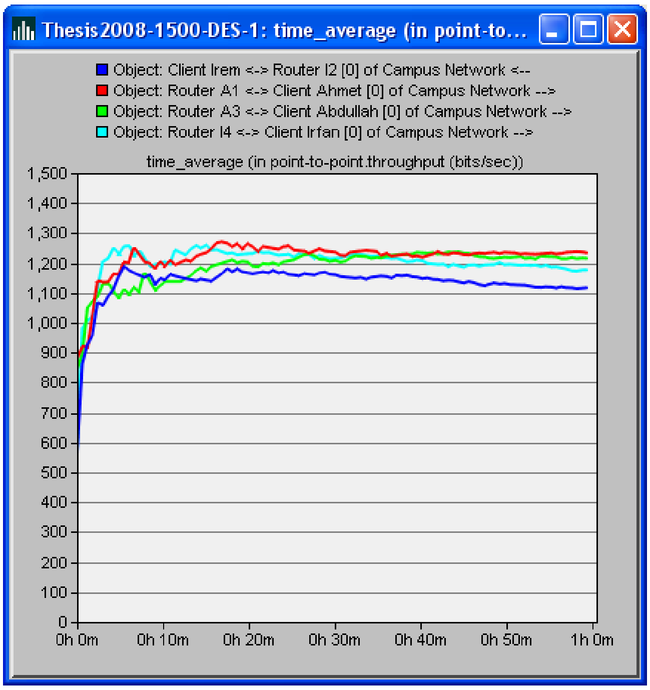


Figure 54 - Total throughput for 1500-byte data size (bits/sec.)

In the 1500-byte data size scenario throughputs of IPv6 Clients (Ahmet and Abdullah) [Table 14] performed slightly than IPv4 clients. Gain of IPv6 is measured about 7.4 % (1216 / 1132).

Client	Min	Max	Expected	Confidence Intervals	
				%80	%99
Ahmet (v6)	882	1269	1216	1208<μ<1224	1200<μ<1232
Abdullah (v6)	847	1237	1188	1180<μ<1196	1171<μ<1204
Irem (v4)	566	1188	1132	1123<μ<1141	1113<μ<1151
Irfan (v4)	575	1259	1200	1190<μ<1209	1180<μ<1219

Table 14 - Total throughput for 1500-byte data size (bits/sec.)

5.2.1.4 Throughput Analysis for 100k-Byte Data Size

The results of total throughput in a time average chart for 100k-byte data size is in [Figure 55, Table 15].

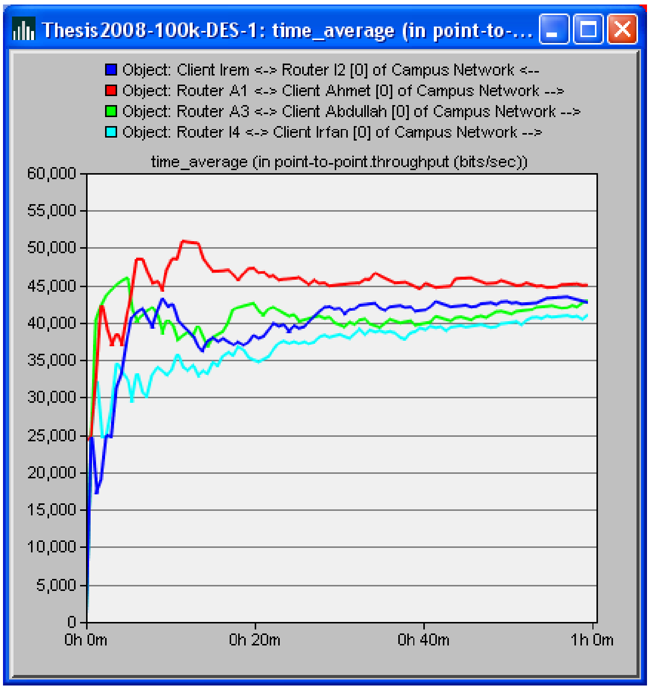


Figure 55 - Total throughput for 100k-byte data size (bits/sec.)

In the 100k-byte data size scenario throughputs of IPv6 Clients (Ahmet and Abdullah) [Table 15] performed better than IPv4 clients. Gain of IPv6 is measured about 23.1 % (45008 / 36555).

Client	Min	Max	Expected	Confidence Intervals	
				%80	%99
Ahmet (v6)	24290	50898	45008	44512<μ<45504	44011<μ<46005
Abdullah (v6)	1497	45999	40344	39766<μ<40922	39183<μ<41505
Irem (v4)	1094	43483	39571	38784<μ<40357	37990<μ<41152
Irfan (v4)	1103	40998	36555	35904<μ<37207	35246<μ<37874

Table 15 - Total throughput for 100k-byte data size (bits/sec.)

5.2.1.5 Discussion

As shown in [Table 12] and [Figure 52] 64-byte data size scenario, throughput gain rate of IPv6 over IPv4 for Client Abdullah over Client Irem is about (622 / 524= 18.7%).

In [Table 13, Figure 53] this ratio (Client Ahmet over Cliet Irem) is about 14,1 % in 1280 byte data size scenario.

In 1500 byte data size scenario [Table14, Figure 54] throughput gain (Client Ahmet over Client Irem) measured about 7,4 %.

In 100k byte data size scenario as shown in [Table15] and [Figure 55] the ratio of Client Ahmet over Client Irfan is about 23,1 % more than IPv4.

This results are parallel with the results in [27]. They measured IPv4 throughput 39.85 kbps and IPv6 throughput of 54.5 kbps. IPv6 throughput gain is about 36,8 %.

For point-to-point test it's measured [26] 1 to 3 % more in Windows and 2 to 5 % more in Linux operating systems. In router-to-router test the gain of IPv6 over IPv4 is 7 to 19 % and in Linux 9 to 12 % more than IPv4. These results are in parallel with our study as well.

In the real network performance results [28] it is measured that in 86, 128, 256 byte frame sizes IPv4 performed slightly better than IPv6. But in bigger sized frames both protocols performance are quite close.

5.2.2 End-to-End Delay Comparison

The sum of all delays that a packet experiences in its journey from source to destination is called end-to-end delay.

In a packet switching network, the time to push all bits to the wire channel is transmission (store and forward) delay. This delay has no relation with the distance of the wire. It's only related with packet length. The calculation of transmission delay is,

$D_T = N / R$

Where,

D_T is the transmission delay,

N is the number of bits, and

R is the rate of transmission (bits/second).

Propagation delay (gate delay), occurs on a medium and defines the amount of time it takes for a certain number of bytes to be transferred.

$D_P = d/s$

Where,

d is the distance,

s is the speed.

Process delay (D_{Pr}), as a key component of network delay, is the time it takes routers to process the packet header,

Queuing delay ($D_{Q)}$ is mainly used for the routers in a network and describes the time a job waits in a queue until it can be executed.

After all end-to-end delay is calculated with formula of,

$D_E = D_T + D_P + D_{Pr} + D_Q$

In this thesis we measured end-to-end delay in all scenarios.

5.2.2.1 End-to-End Delay Analysis for 256-Byte Data Size

The results of end-to-end delay in a time average chart for 256-byte data size is in [Figure 56, Table 16].

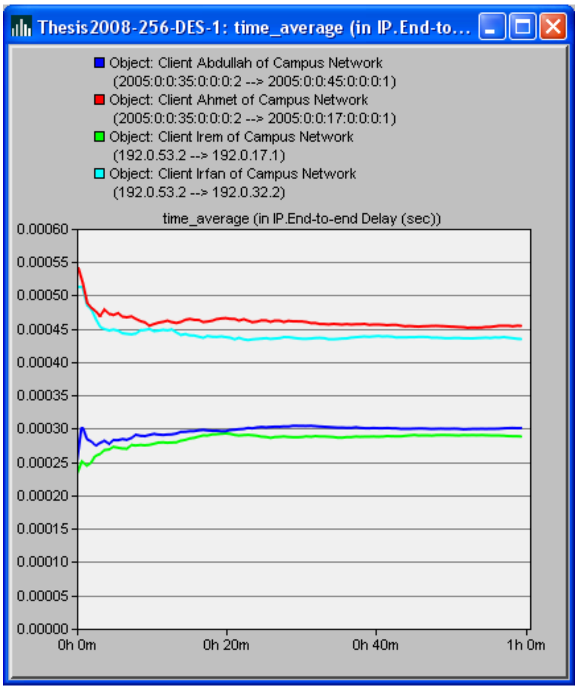


Figure 56 - End-to-end delay in 256-byte data size (sec.)

In the 256-byte data size scenario end-to-end delays of IPv4 Clients (Irem and Irfan) [Table 16] are less (0.012 [0.289 – 0.301] to 0.019 [0.435 – 0.454] miliseconds) than IPv6 clients.

Client	Min	Max	Final	Confidence Intervals	
				%80	%99
Ahmet (v6)	0.451	0.542	0.454	0.460<µ<0.463	0.458<µ<0.464
Abdullah (v6)	0.252	0.305	0.301	0.296<µ<0.298	0.295<µ<0.299
Irem (v4)	0.233	0.293	0.289	0.283<µ<0.286	0.282<µ<0.287
Irfan (v4)	0.433	0.513	0.435	0.440<µ<0.443	0.438<µ<0.445

Table 16 - End-to-end delay for 256-byte data size (milisecond)

5.2.2.2 End-to-End Delay Analysis for 1M-Byte Data Size

The results of end-to-end delay in a time average chart for 1M-byte data size is in [Figure 57, Table 17].

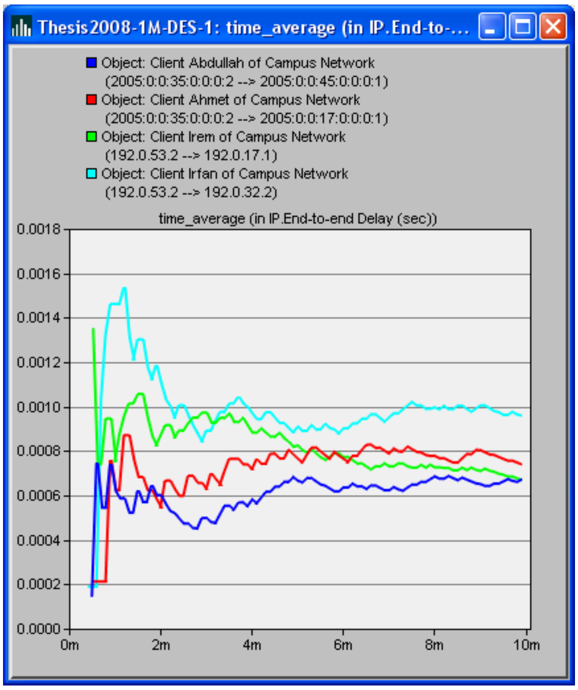


Figure 57 - End-to-end delay in 1M-byte data size (sec.)

Client	Min	Max	Final	Confidence Intervals	
				%80	%99
Ahmet (v6)	0.212	0.871	0.740	0.709<μ<0.742	0.693<μ<0.759
Abdullah (v6)	0.148	0.748	0.672	0.598<μ<0.620	0.588<μ<0.630
Irem (v4)	0.669	1.357	0.669	0.808<μ<0.838	0.793<μ<0.854
Irfan (v4)	0.189	1.541	0.959	0.956<μ<1.008	0.930<μ<1.034

Table 17 - End-to-end delay for 1M-byte data size (milisecond)

In the 1M-byte data size scenario end-to-end delay of IPv6 client (Ahmet) [Table 17] is less (0.219 [0.740 – 0.959] miliseconds) than IPv4 client Irfan.

5.2.2.3 Discussion

As shown in [Table 16, Figure 56] in small data sizes IPv6 end-to-end delay is more (0.012 to 0.019 miliseconds less) than IPv4 but as shown in [Table 17, Figure 57] IPv6 end-to-end delay becomes less (0.219 miliseconds less) than IPv4 while the data size increases.

The results [26] of delay measurement are parallel in some aspects. In point-to-point scenario it is measured that IPv6 delay is 1 to 5 % more than IPv4 in Windows operating system. In router-to router scenario IPv6 delay overhead becomes 9 to 20 % more than IPv4.

In the native IPv4 and IPv6 performance comparison [27] IPv6 delay is measured about 4.22 milisec. Where IPv4 delay is 4.59 milisec.

Project results [28] shows that IPv6 delay is more than IPv4. The delay difference is less in smaller frame sizes. There is an interesting result that the delay of IPv6 and IPv4 in older routers varies about 50 % in some frame sizes but with new technology routers the delay difference becomes almost negligable in all frame sizes.

In the CERNET study [29] it is measured that IPv6 delay varies from 320 to 420 miliseconds where its counterpart IPv4 delay varies from 230 to 600 miliseconds.

5.3.3 Comparison of Transition Technologies Throughput

In this phase we examined 3 types of transition mechanism. Site B uses manual IPv6 tunnel (Router-to-Router tunnelling), Site C has dual stack mechanism, and Site D has two 6to4 automatic tunnels (Host-to-Router & Router-to-Host Tunnelling).

5.3.3.1 Transition Mechanism Throughput in 46-Byte Data Size

The results of total throughput in a time average chart for 46-byte data size is in [Figure 58, Table 18].

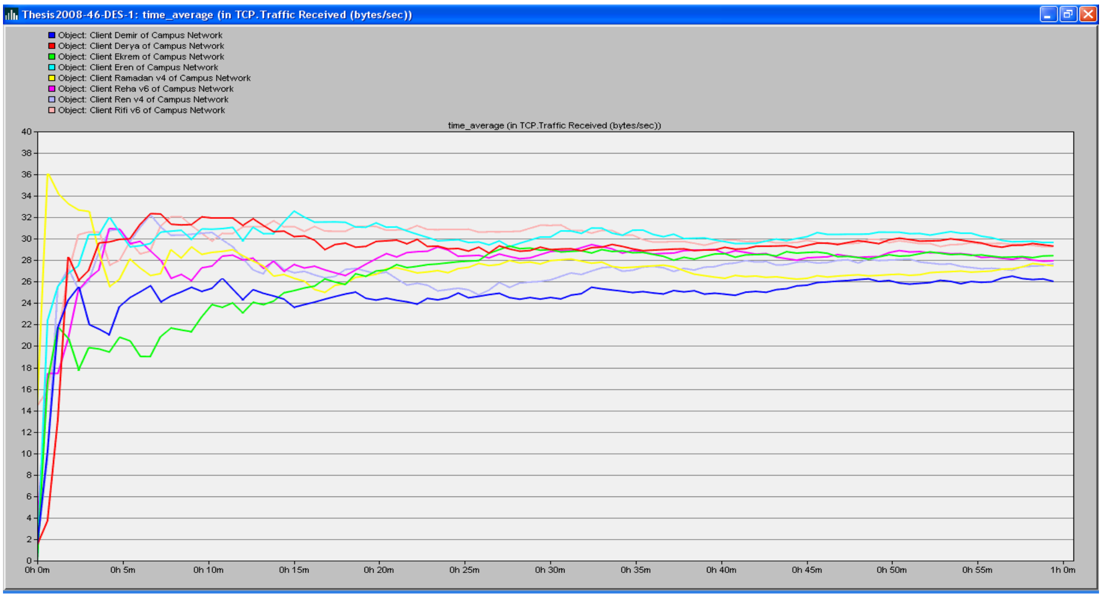


Figure 58 - Transition mechanism throughput for 46-byte data size (bytes/sec.)

In the 46-byte data size scenario throughputs of IPv6 clients [Table 18] are more than IPv4 clients.

IPv6 (Client Ekrem & Eren) gain over IPv4 (Client Ramadan & Ren) in manual tunnelling is measured approximately 7.9 % (29.66 / 27.48).

IPv6 (Client Demir & Derya) gain over IPv4 (Client Ramadan & Ren) in 6to4 automatic tunnelling is measured 6.7 % (29.32 / 27.48).

Ipv6 (Client Reha & Rifi)gain over IPv4 (Client Ramadan & Ren) in dual-stack mechanism is measured about 6.6 % (29.29 / 27.48).

Client / IP Protocol / Trans. Mechanism	Min	Max	Final	Confidence Intervals	
				%80	%99
Demir (v6 - 6to4)	1.5	26.5	26.04	24.22<μ<24.97	23.84<μ<25.35
Derya (v6 - 6to4)	1.5	32.34	29.32	28.41<μ<29.50	27.86<μ<30.05
Ekrem (v6 - Man)	0	29.32	28.42	25.73<μ<26.78	25.19<μ<27.32
Eren (v6 - Man)	0	32.57	29.66	29.49<μ<30.32	29.07<μ<30.74
Reha(v6 - DSTM)	3	30.95	27.95	27.36<μ<28.16	26.95<μ<28.57
Rifi (v6 - DSTM)	14.44	32.06	29.29	29.48<μ<30.11	29.16<μ<30.43
Ramadan (v4-DSTM)	14.44	36.11	27.48	27.04<μ<27.59	26.77<μ<27.86
Ren (v4 - DSTM)	1.5	32.22	27.66	26.59<μ<27.48	26.14<μ<27.93

Table 18 - Transition mechanisms throughput for 46-byte data size (bytes/sec.)

5.3.3.2 Transition Mechanism Throughput in 1480-Byte Data Size.

The results of total throughput in a time average chart for 1480-byte data size is in [Figure 59, Table 19].

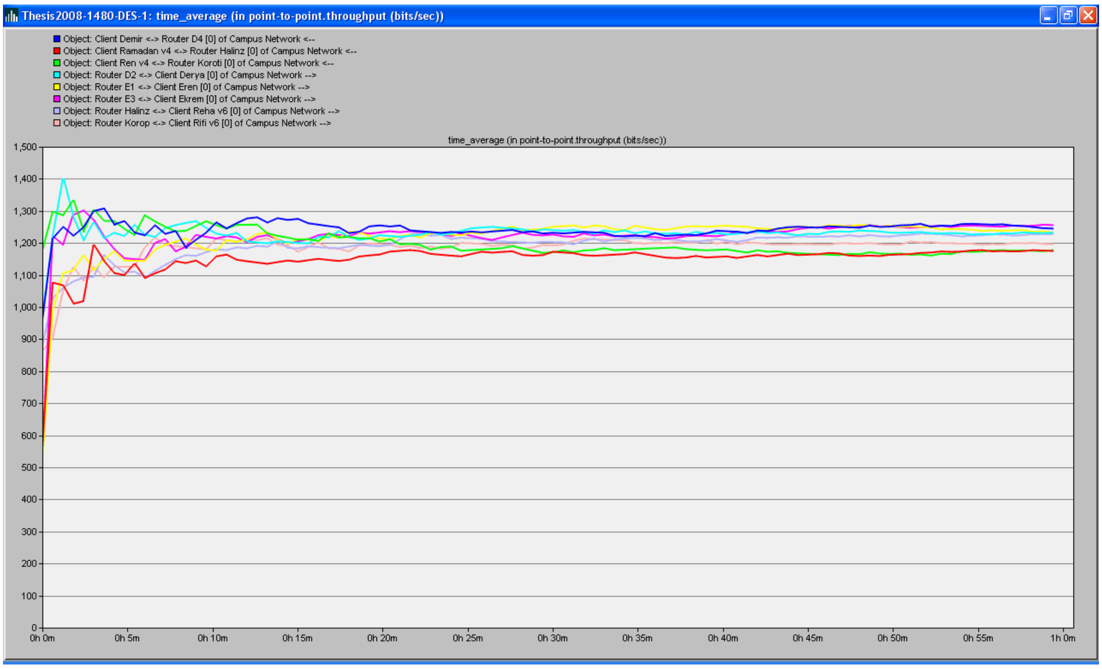


Figure 59 - Transition mechanism throughput for 1480-byte data size (bytes/sec.)

In the 1480-byte data size scenario throughputs of IPv6 clients [Table 19] are more than IPv4 clients.

IPv6 (Client Ekrem & Eren) gain over IPv4 (Client Ramadan & Ren) in manual tunnelling is measured approximately 6.9% (1256 / 1175).

IPv6 (Client Demir & Derya) gain over IPv4 (Client Ramadan & Ren) in 6to4 automatic tunnelling is measured 6 % (1245 / 1175).

Ipv6 (Client Reha & Rifi)gain over IPv4 (Client Ramadan & Ren) in dual-stack mechanism is measured about 4.5 % (1228 / 1175).

Client / IP Protocol / Trans. Mechanism	Min	Max	Final	Confidence Intervals	
				%80	%99
Demir (v6 - 6to4)	963	1307	1245	$1237<\mu<1246$	$1233<\mu<1250$
Derya (v6 - 6to4)	972	1402	1232	$1226<\mu<1234$	$1221<\mu<1239$
Ekrem (v6 - Man)	540	1301	1256	$1212<\mu<1231$	$1203<\mu<1240$
Eren (v6 - Man)	523	1256	1233	$1206<\mu<1227$	$1195<\mu<1237$
Reha(v6 - DSTM)	882	1231	1228	$1185<\mu<1198$	$1178<\mu<1205$
Rifi (v6 - DSTM)	865	1218	1197	$1177<\mu<1190$	$1170<\mu<1196$
Ramadan (v4- DSTM)	562	1194	1175	$1139<\mu<1156$	$1131<\mu<1165$
Ren (v4 - DSTM)	1161	1334	1178	$1195<\mu<1204$	$1190<\mu<1209$

Table 19 - Transition mechanism throughput in 1480-byte data size (bits/sec.)

5.3.3.3 Discussion.

In all tunnels as shown in 5.3.3.1 and 5.3.3.2 sections IPv6 throughput is slightly better than IPv4. But the IPv6 gains of dual stack, manual tunnel and 6to4 tunnel throughputs ares less than native IPv6 network throughput. Throughput gain of tunnel mechanisms varies from 4.5 to 7.9 % more than IPv4. It was 7.4 to 23.1 % more in native IPv6 scenarios. This may happen because of the overhead of tunnel mechanism.

Cisco dual stack mechanism test results [28] show that almost same results with IPv4 throughput is obtained. In small frame sizes IPv4 and DSTM throughputs are more than IPv6 but in bigger frame sizes throughput difference is too small.

In the M.S. thesis [26] IPv6 over IPv4 tunnelling mechanism is tested. In Windows environment both router-to-router tunnelling and host-to-host tunnelling mechanism throughputs are more than IPv6 native routing throughput 12 % and 5% respectively. In Linux environment test the results are at the same direction with the results of 15 % more in router-to-router tunnelling and 4% more in host-to-host tunnelling. IPv4 has performe better than IPv6 in those tests.

In our test results it is shown that there are slight differences with IPv4 throughput results.

CHAPTER 6

CONCLUSIONS AND FUTURE WORK

6.1 CONCLUSIONS

In this thesis we tried to build up and examine a number of experiments to compare the performance analysis of IPv6 and IPv4 protocols in IPv4, IPv6 native, IPv6 manuelly tunnelled, dual stack and 6to4 tunnelled networks. We examined the throughput and end-to-end delay parameters of both protocols.

Performance analysis for point-to-point architecture is carried out to see only the normal working characteristics of both protocols. With this study, traffic characteristics of transition mechanisms are also put into consideration.

In this thesis it is found that IPv6 native traffic throughput is superior to IPv4 in all scenarios. Throughput gain measured up to 23.1 % more than IPv4. This result is because of routing enhancements of IPv6.

It is mentioned [28] that networks with older routers (less CPU capacity) IPv4 performes better than IPv6 in small size frames. In bigger packets or with the new routers (more CPU capacity) the throughput difference is negligible. This result shows that networks designed to operate IPv4 may not operate IPv6 with the same performance. IPv6 is still in development phase and with new network equipment releases IPv6 performance will go further.

As in native IPv6, throughput of the networks that use tunnelling mechanism results measured better than IPv4 in all the experiments. The IPv6 throughput gain is measured up to 7.9% and it is not as significant as native IPv6 network. It can be said that overhead of tunnelling mechanisms is the main reason of less performance.

An interesting observation for this study is that in some bigger data size scenarios IPv4 performed as good as IPv6.

In these experiments we also carried out that IPv6 end-to-end delay is mostly more than IPv4. This may be because of the bigger overhead constraints of IPv6.

Final achievement of this thesis is that, for a new protocol, IPv6 deserve the higher rank from IPv4. It will be a good choice for at least the trial of IPv6 transition mechanisms, but superior one is to make some investments for IPv6 infrastructure.

6.2 FUTURE WORK

This study is made on a single computer simulation environment. More realistic work can be done in a real world configured network. If uptodate network elements will be used in that real environment better results can be taken.

We also focused on a few parameters of routing. Another parameter of high packet loss ratio (a well known problem in IPv6) and the underlying reasons could be revealed with a study.

Mobile communication is the famoust networking study in the globe. It offers many facilities and much more are on the way. As an emerging technology, IPv6 will gain its desired acceptance with mobility and mobile IPv6. Studies on this subject will be a good choice.

For a further study in routing aspect the following researches will be beneficial:

- This work can be extended to other IPv6 implementation simulation on other platforms.
- Different routers, routing protocols, links and networks such as ATM, Frame Relay etc. platforms can be injected for more realistic environment.
- This work can also be extended for multicast traffic observation in multimedia-based traffic.
- Quality of Service (QoS) testing in IPv6 implementation.

REFERENCES

[1] Chang Xinjie, Network Simulations With OPNET, 1999

[2] Cisco, http://www.ubookcase.com/book/Cisco/Routing.TCP.IP.Volume.II.CCIE.Professional.Development, accessed 03.01.2009.

[3] Cisco, IPv6 Extension Headers Review and Considerations, October 2006

[4] Cisco, IPv6 Headers At A Glance, 2005

[5] Hagen Silvia, IPv6 Essentials, O'Reilly, ISBN: 0-596-00125-8

[6] IANA, IPv4 Global Unicast Address Assignments, http://www.iana.org/assignments/ ipv4-address-space, accessed 20.03.2008

[7] IBM, Comparison of IPv4 and IPv6, https://publib.boulder.ibm.com/iseries/v5r2/ic2924/index.htm?info/rzai2/rzai2compipv4ipv6.htm, accessed 19.03.2008

[8] IEEE, IPv4+4, http://ieeexplore.ieee.org/iel5/8412/26518/01181416.pdf, accessed 20.03.2008.

[9] Internet World Stats, http://www.internetworldstats.com/stats.htm, accessed 20.03.2008

[10] IP History, Standards, Versions and Closely-Related Protocols, http://www.tcpipguide.com/, accessed 19.03.2008

[11] IPv4 Address Status, http://www.potaroo.net/tools/ipv4/index.html#r4, accessed 20.12.2008

[12] IPv4 and IPv6: A Comparison http://www.myitforum.com, accessed 20.03.2008.

[13] IPv6 Configuration Know-How 2003, http://www.nic.ad.jp/, accessed 20.06.2008

[14] IPv6: At the Starting Line, http://support.novell.com/, accessed 20.12.2008

[15] ISC, Internet Domain Survey Host Count, http://www.isc.org/index.pl?/ops/ds/, accessed 20.03.2008.

[16] King Steve, Fax Ruth, Haskin Dimitry, Ling Wenken, Meehan Tom, Fink Robert, Perkins Charles E., The Case for IPv6, http://www.6bone.net/case-for-ipv6.txt

[17] Novell, http://support.novell.com/techcenter/articles, accessed 11.01.2009

[18] OPNET Technologies Inc., Building Network Topologies, 2004

[19] OPNET Technologies Inc., Simulation Technology Options, 2004

[20] Sun, http://docs.sun.com, accessed 05.01.2009.

[21] Why IPv6 ?, http://www.ipv6tf.org, accessed 21.03.2008

[22] Why you want IPv6, http://en.linuxreviews.org/, accessed 21.03.2008

[23] Wikipedia, http://en.wikipedia.org, accessed 30.12.2008

[24] Lucio Gilberto Flores, Marcos Paredes-Farrera, Emmanuel Jammeh, Martin Fleury, Martin J. Reed, OPNET Modeler and Ns-2: Comparing the Accuracy of Network Simulators for Packet-Level Analysis using a Network Testbed, University of Essex, October 2003.

[25] Universities using OPNET, http://www.opnet.com/university_program/participant_spotlight/universities.html

[26] Mohammad Mokbul Ahmed, A Comperative Study on the Performance of IPv4 and IPv6, Independent University, Bangladesh, 2006

[27] Dicle Zafer, Bora Mocan, Mahir Kutay, IPv6 ve IPv4 İçin Karşılştırmalı Bilgisayar Benzetimi, Dokuz Eylül University.

[28] Cisco, Performance-Comparison Testing of IPv4 and IPv6 Throughput and Latency on Key Cisco Router Platforms, 2007.

[29] Yi Wang, Shaozhi Ye, Xing Li, Understanding Current IPv6 Performance: A Case Study from CERNET, Department of Electronic Engineering, Tsinghua University, 2004.

[30] Davies Joseph, Understanding IPv6, Microsoft Press, 2003.

[31] Microsoft Technet, www.microsoft.com, accessed 30.12.2008

[32] http:// www.commandinformation.com, accessed 28.12.2008

[33] http://www.ipv6tf.org, accessed 20.12.2008

[34] http://www.lucastomicki.net, accessed 20.12.2008

APPENDICES

APPENDIX A: IPv6 AND IPv4 HEADER COMPARISON

Fields of the IPv4 Header	Fields of the IPv6 Header	Comparison of IPv4 and IPv6 Headers
Version (4-bit)	Version (4-bit)	Same function but the IPv6 header contains a new value.
Header length (4-bit)	—	Removed in IPv6. The basic IPv6 header always has 40 octets.
Type of service (8-bit)	Traffic class (8-bit)	Designates the IPv6 traffic class, similarly to IPv4. Uses different codes.
—	Flow label (20-bit)	New field added to tag a flow for IPv6 packets.
Total length (16-bit)	Payload length (16-bit)	IPv6 normal packet size is up to 65536 octets. "**jumbogram**" - up to 4 billion octets for high-performance computing networks
Identification (16-bit)	—	Removed in IPv6 because fragmentation is handled differently in IPv6.
Flags (3-bit)	—	Removed in IPv6 because fragmentation is handled differently in IPv6.
Fragment offset (13-bit)	—	Removed in IPv6 because fragmenta-tion is handled differently in IPv6.

Time to live (8-bit)	Hop limit (8-bit)	Same function for both headers.
Protocol number (8-bit)	Next header (8-bit)	The type of header immediately following the IPv6 header. Uses the same values as the IPv4 protocol field. But the architectural effect is to allow a currently defined range of next headers, and is easily extended. The next header will be a transport header, an extension header, or ICMPv6.
Header checksum (16-bit)	—	Removed in IPv6. Link-layer technol-ogies and upper-layer protocols handle checksum and error control.
Source address (32bit)	Source address (128-bit)	Source address is expanded in IPv6.
Destination adr. (32bit)	Destination adr. (128-bit)	Destination address is expanded in IPv6.
Options (variable)	—	The IPv6 header has no options. Instead, IPv6 adds additional (optional) extension headers.
Padding (variable)	—	Removed in IPv6. The way to handle this option is different in IPv4.
—	Extension headers	New way in IPv6 to handle Options fields, fragmentation, security, mobility, Loose Source Routing, Record Route, and so on. The following section presents IPv6's extension headers.

Table 20 - Comparison of IP headers [3]

Extension Headers in IPv6:

In IPv4 options are placed at the end of IP header. In IPv6 they are hold separately as extension headers. Options in IPv4 makes the header size variable and isn't proccessed by every vendor because of the not clearly standarts.

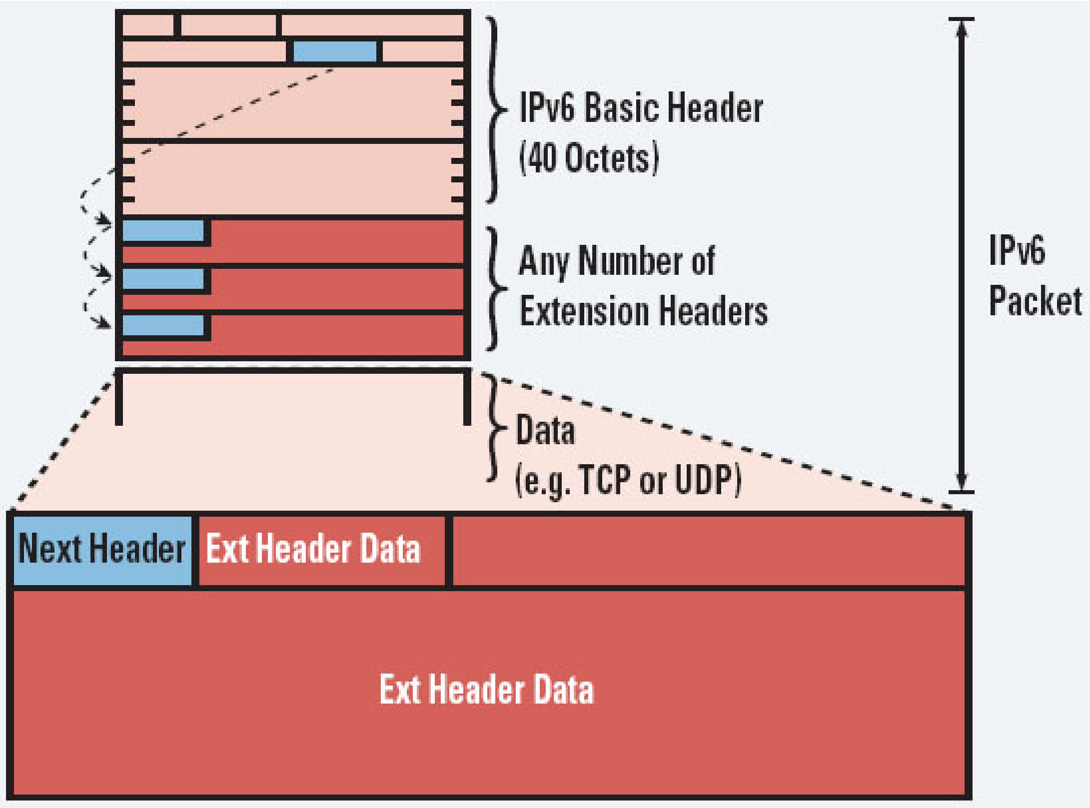


Figure 60 - Extension headers [4]

IPv6 extension headers are processed as necessary (e.g. routers only process hop-by-hop options header) and easier to define new extensions and options. Basic extension headers in IPv6 are:

Next Header Value	Extension Header Type
0	In an IPv4 header: reserved and not used In an IPv6 header: Hop-by-Hop Option Header following
1	Internet Control Message Protocol (ICMPv4)IPv4 support
2	Internet Group Management Protocol (IGMPv4)IPv4 support
4	IPv4

6	TCP
8	Exterior Gateway Protocol (EGP)
9	IGPany private interior gateway (used by Cisco for their IGRP)
17	UDP
41	IPv6
43	Routing header
44	Fragmentation header
45	Interdomain Routing Protocol (IDRP)
46	Resource Reservation Protocol (RSVP)
47	General Routing Encapsulation (GRE)
50	Encrypted Security Payload header
51	Authentication header
58	ICMPv6
59	No Next Header for IPv6
60	Destination Options header
88	EIGRP
89	OSPF
108	IP Payload Compression Protocol
115	Layer 2 Tunneling Protocol (L2TP)
132	Stream Control Transmission Protocol (SCTP)
135	Mobility Header (Mobile IPv6)
136-254	Unassigned
255	Reserved

Table 21 - Extension headers in IPv6

The process of extension header is as follows:

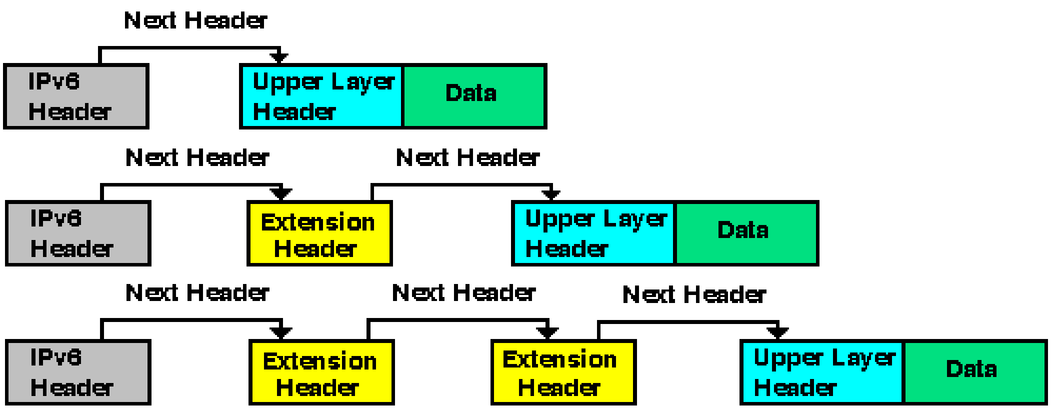


Figure 61 - IPv6 header with 0, 1 and 2 extension headers [4]

Each extension header is a multiple of 8 octets long. That way, subsequent headers can always be aligned. If a node is required to process the Next Header but cannot identify the value in the Next Header field, it is required to discard the packet and send an ICMPv6 Parameter Problem message back to the source of the packet. If more than one extension header is used in a single packet, the following header order should be used (RFC 2460):

Header Order	Header Name
0	IPv6 header
1	Hop-by-hop options header
2	Destination Options header
3	Routing header
4	Fragment header
5	Authentication header
6	Encapsulating security payload header
7	Destination options header
8	Upper-layer header

Table 22 - Extension headers order

In cases when IPv6 is encapsulated in IPv4, the upper-layer header can be another IPv6 header and can contain extension headers that have to follow the same rules [5].

APPENDIX B: DIFFERENCES BETWEEN IPv6 AND IPv4

Subject	IPv4	IPv6
address	32 bits long (4 bytes). Address is composed of a network and a host portion, which depend on address class. Various address classes are defined: A, B, C, D, or E depending on initial few bits. The total number of IPv4 addresses is 4,294,967,296.	128 bits long (16 bytes). Basic architecture is 64 bits for the network number and 64 bits for the host number. Often, the host portion of an IPv6 address (or part of it) will be a MAC address or other interface identifier.
	The text form of the IPv4 address is nnn.nnn.nnn.nnn, where 0<=nnn<=255, and each n is a decimal digit. Leading zeros may be omitted. Maximum number of print characters is 15, not counting a mask.	Depending on the subnet prefix, IPv6 has a more complicated architecture than IPv4.
		The number of IPv6 addresses is 10^{28} (79,228,162,514,264,337,593,543,950,336) times larger than the number of IPv4 addresses.
		The text form of the IPv6 address is xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx, where each x is a hexadecimal digit, representing 4 bits. Leading zeros may be omitted. The double colon (::) may be used once in the text form of an address, to designate any number of 0 bits. For example, ::ffff:10.120.78.40 is an IPv6 IPv4-mapped

		address. (See RFC 2373 for details.
address allocation	Originally, addresses were allocated by network class. As address space is depleted, smaller allocations using Classless Inter-Domain Routing (CIDR) are made. Allocation has not been balanced among institutions and nations.	Allocation is in the earliest stages. The Internet Engineering Task Force (IETF) and Internet Architecture Board (IAB) have recommended that essentially every organization, home, or entity be allocated a /48 subnet prefix length. This would leave 16 bits for the organization to do subnetting. The address space is large enough to give every person in the world their own /48 subnet prefix length.
address lifetime	Generally, not an applicable concept, except for addresses assigned using DHCP.	IPv6 addresses have two lifetimes: preferred and valid, with the preferred lifetime always <= valid. After the preferred lifetime expires, the address is not to be used as a source IP address. After the valid lifetime expires, the address is not used (recognized) as a valid destination IP address for incoming packets. Some IPv6 addresses have, by definition, infinite preferred and valid lifetimes; for example link-local (see address scope).
address mask	Used to designate network from host portion.	Not used (see address prefix).
address prefix	Sometimes used to designate network from host portion. Sometimes written as /nn suffix on presentation form of address.	Used to designate the subnet prefix of an address. Written as /nnn (up to 3 decimal digits, 0 <= nnn <= 128) suffix after the print form. An example is fe80::982:2a5c/10, where the first 10 bits comprise the subnet prefix.
Address Resolution Protocol (ARP)	Address Resolution Protocol is used by IPv4 to find a physical address, such as the MAC or link address, associated with an IPv4 address.	IPv6 embeds these functions within IP itself as part of the algorithms for stateless autoconfiguration and neighbor discovery using Internet Control Message Protocol version 6 (ICMPv6). Hence, there is no such thing as ARP6.

address scope	For unicast addresses, the concept does not apply. There are designated private address ranges and loopback. Outside of that, addresses are assumed to be global.	In IPv6, address scope is part of the architecture. Unicast addresses have 3 defined scopes, including link-local, site-local and global; and multicast addresses have 14 scopes. Default address selection for both source and destination takes scope into account.
		A scope zone is an instance of a scope in a particular network. As a consequence, IPv6 addresses sometimes have to be entered or associated with a zone ID. The syntax is %zid where zid is a number (usually small) or a name. The zone ID is written after the address and before the prefix. For example, 2ba::1:2:14e:9a9b:c3/48.
address types	Unicast, multicast, and broadcast.	Unicast, multicast, and anycast.
communications trace	A tool to collect a detailed trace of TCP/IP (and other) packets.	Same for IPv6 and IPv6 is supported, including ICMPv6 and IPv6 packets tunneled in IPv4.
configuration	Configuration must be done on a newly installed system before it can communicate; that is, IP addresses and routes must be assigned.	Configuration is optional, depending on functions required. An appropriate Ethernet or tunnel interface must be designated as an IPv6 interface. Once that is done, IPv6 interfaces are self-configuring. So, the system will be able to communicate with other IPv6 systems that are local and remote, depending on the type of network and whether an IPv6 router exists.
Domain Name System (DNS)	Applications accept host names and then use DNS to get an IP address.	Same for IPv6. Support for IPv6 exists using AAAA (quad A) record type and reverse lookup (IP-to-name). An application may elect to accept IPv6 addresses from DNS (or not) and then use IPv6 to communicate (or not).
fragments	When a packet is too big for the next link over which it is to travel, it can be fragmented by the sender (host or router).	For IPv6, fragmentation can only occur at the source node, and reassembly is only done at the destination node. Currently, the fragmentation extension header is not supported.
interface	The conceptual or logical entity used by TCP/IP to send and receive packets and always closely associated with an	Same concept as IPv4.

	IPv4 address, if not named with an IPv4 address. Sometimes referred to as a logical interface.	
	Can be started and stopped independently of each other and independently of TCP/IP.	Can be started and stopped independently of each other and independently of TCP/.
Internet Control Message Protocol (ICMP)	ICMP is used by IPv4 to communicate network information.	Used similarly for IPv6; however, Internet Control Message Protocol version 6 (ICMPv6) provides some new attributes.
		Basic error types remain, such as destination unreachable, echo request and reply. New types and codes are added to support neighbor discovery and related functions.
Internet Group Management Protocol (IGMP)	IGMP is used by IPv4 routers to find hosts that want traffic for a particular multicast group, and used by IPv4 hosts to inform IPv4 routers of existing multicast group listeners (on the host).	Replaced by MLD (multicast listener discovery) protocol for IPv6. Does essentially what IGMP does for IPv4, but uses ICMPv6 by adding a few MLD-specific ICMPv6 type values.
LAN connection	Used by an IP interface to get to the physical network. Many types exist; for example, token ring, Ethernet, and PPP. Sometimes referred to as the physical interface, link, or line.	IPv6 has the same concept.
loopback address	An interface with address of 127.*.*.* (typically 127.0.0.1) that can only be used by a node to send packets to itself.	The concept is the same as in IPv4, and the single loopback address is. 0000:0000:0000:0000:0000:0000:0000:0001 or ::1 (shortened version). The virtual physical interface is named.

	The physical interface (line description) is named	
Maximum Transmission Unit (MTU)	Maximum transmission unit of a link is the maximum number of bytes that a particular link type, such as Ethernet or modem, supports. For IPv4, 576 is the typical minimum.	IPv6 has an architected lower bound on MTU of 1280 bytes. That is, IPv6 will not fragment packets below this limit. To send IPv6 over a link with less than 1280 MTU, the link-layer must transparently fragment and defragment the IPv6 packets.
Network Address Translation (NAT)	Basic firewall functions integrated into TCP/IP, configured using iSeries Navigator.	Currently, NAT does not support IPv6. More generally, IPv6 does not require NAT. The expanded address space of IPv6 eliminates the address shortage problem and enables easier renumbering.
node info query	Does not exist.	A simple and convenient network tool that should work like ping, except with content: an IPv6 node may query another IPv6 node for the target's DNS name, IPv6 unicast address, or IPv4 address. Currently, not supported.
packet filtering	Basic firewall functions integrated into TCP/IP	Currently, packet filtering does not support IPv6. However, IPv4 filtering can be applied to tunneled IPv6 traffic.
packet tunneling	In IPv4, tunneling occurs in VPN for tunnel-mode VPN connections (IPv4 tunneled in IPv4) and in L2TP.	For IPv6, tunneling in IPv4 packets is expected to be a major part of its evolution.
		A basic and flexible type of IPv6-in-IPv4 tunneling is supported to allow IPv6 nodes to communicate across the existing IPv4 Internet. Called configured tunneling, it provides a virtual point-to-point link between two IPv6 nodes and uses a new type of tunnel line.
ports	TCP and UDP have separate port spaces, each identified by port numbers in the range 1-65535.	For IPv6, ports work the same as IPv4.
private and public addresses	All IPv4 addresses are public, except	IPv6 has an analogous concept, but with important differences.

	for three address ranges that have been designated as private by IETF RFC 1918: 10.*.*.* (10/8), 172.16.0.0 through 172.31.255.255 (172.16/12), and 192.168.*.* (192.168/16). Private address domains are commonly used within organizations. Private addresses cannot be routed across the Internet.	Addresses are public or temporary, previously termed anonymous. See RFC 3041. Unlike IPv4 private addresses, temporary addresses can be globally routed. The motivation is also different; IPv6 temporary addresses are meant to shield the identity of a client when it initiates communication (a privacy concern). Temporary addresses have a limited lifetime, and do not contain an interface identifier that is a link (MAC) address. They are generally indistinguishable from public addresses. IPv6 has the notion of limited address scope using its architected scope designations (see address scope).
renumbering	Done by manual reconfiguration, with the possible exception of DHCP. Generally, for a site or organization, a difficult and troublesome process to avoid if possible.	Is an important architectural element of IPv6, and is supposed to be largely automatic especially within the /48 prefix.
route	Logically, a mapping of a set of IP addresses (may contain only 1) to a physical interface and a single next hop IP address. IP packets whose destination address is defined as part of the set are forwarded to the next hop using the line. IPv4 routes are associated with an IPv4 interface, hence, an IPv4 address.	Conceptually, the same as IPv4. One important difference: IPv6 routes are associated (bound) to a physical interface (a link, rather than an interface. There are various reasons for this. One reason is that source address selection functions differently for IPv6 than for IPv4. See source address selection.

		Duplicate routes are allowed to improve robustness, but they are ignored during route lookup.
sockets API	These APIs are the way applications use TCP/IP. Applications that do not need IPv6 are not affected by sockets changes to support IPv6.	IPv6 enhances sockets so that applications can now use IPv6.
		The enhancements have been designed so that existing IPv4 applications are completely unaffected by IPv6 and API changes. Applications that want to support concurrent IPv4 and IPv6 traffic, or IPv6-only traffic, are easily accommodated using IPv4-mapped IPv6 addresses of the form ::ffff:a.b.c.d, where a.b.c.d is the IPv4 address of the client.
		The new APIs also include support for converting IPv6 addresses from text to binary and from binary to text.
Telnet	Telnet allows you to log on and use a remote computer as though you were connected to it directly.	Same for IPv6.
trace route	Basic TCP/IP tool to do path determination.	Same for IPv6.
transport layers	TCP, UDP, RAW. A new transport, Stream Control Transmission Protocol (SCTP), aims to offer the best features of TCP and UDP, that is, guaranteed connectionless communication. SCTP is in the earliest stage of use.	Same three transports exist and are functionally unchanged for IPv6.
unspecified address	Apparently, not defined, as such. Socket programming uses 0.0.0.0.	Defined as::/128 (128 0 bits). It is used as the source IP in some neighbor discovery packets, and various other contexts, like sockets.
IP header	Variable length of 20-60 bytes,	Fixed length of 40 bytes. There are no IP header options. Generally, the IPv6 header is

	depending on IP options present.	simpler than the IPv4 header.
Packet Header	variable size - time-consuming to handle	fixed size (40 octets) - more efficient
Special Fields in Header	many types, often not supported by vendors due to impact on performance	eliminated for efficiency or replaced by other features
virtual private networking (VPN)	Virtual private networking (using IPsec) allows you to extend a secure, private network over an existing public network.	Currently, VPN does not support IPv6. However, when IPv6 is tunneled in IPv4, existing VPN facilities can be applied to the IPv4 traffic, which then transparently handles the IPv6 payloads.

Table 23 - Other comparisons of IPv6 & IPv4 [7]

APPENDIX C: IPv6 RELATED RFCs

- **IPv6 SPECIFICATION:**

R. Hinden, S. Deering, Internet Protocol, Version 6 (IPv6) Specification, December 1998.

- **ADDRESSING:**

R. Hinden, S. Deering, IP Version 6 Addressing Architecture, RFC 2373, July 1998.

R. Hinden, S. Deering, IP Version 6 Addressing Architecture, Internet Draft, July 2001.

IAB, IESG, IPv6 Address Allocation Management, RFC 1881, December 1995.

Y. Rekhter, T. Li, An Architecture for IPv6 Unicast Address Allocation, RFC 1887, December 1995.

R. Hinden, M. O'Dell, S. Deering, An IPv6 Aggregatable Global Unicast Address Format, RFC 2374, July 1998.

R. Hinden, Proposed TLA and NLA Assignment Rules, RFC 2450, December 1998.

R. Hinden, S. Deering, R. Fink, T. Hain, Initial IPv6 Sub-TLA ID Assignments, RFC2928, September 2000.

R. Hinden, R. Fink, J. Postel, IPv6 Testing Address Allocation, RFC2471, December 1998.

D. Harrington, Link Local Addressing and Name Resolution in IPv6, Internet Draft, draft-ietf-ipngwg-linkname-01.txt, January 1997.

R. Hinden, S. Deering, IPv6 Multicast Address Assignments, RFC 2375, July 1998.

D. Johnson, S. Deering, Reserved IPv6 Subnet Anycast Addresses, RFC2526, March 1999.

M. Blanchet, A flexible method for managing the assignments of bits of an IPv6 address block, Internet Draft, draft-ietf-ipngwg-ipaddressassign-02.txt, March 2001.

R. Hinden, B. Carpenter, L. Masinter, Format for Literal IPv6 Addresses in URL's, RFC 2732, December 1999.

S. Deering, B. Haberman, B. Zill, IP Version 6 Scoped Address Architecture, Internet Draft, draft-ietf-ipngwg-scoping-arch-02.txt, March 2001.

B. Haberman, D. Thaler, Unicast-Prefix-based IPv6 Multicast Addresses, Internet Draft, draft-ietf-ipngwg-uni-based-mcast-02.txt, June 2001.

- **MULTIHOMING:**

R. Draves, Default Address Selection for IPv6, Internet Draft, draft-ietf-ipngwg-default-addr-select-05.txt, June 2001.

J. Yu, IPv6 Multihoming with Route Aggregation, Internet Draft, draft-ietf-ipngwg-ipv6multihome-with-aggr-01.txt, August 2000.

F. Dupont, Multihomed routing domain issues for IPv6 aggregatable scheme, Internet Draft, draft-ietf-ipngwg-multi-isp-00.txt, September 1999.

- **INTERNET CONTROL MESSAGE PROTOCOL:**

A. Conta, S. Deering, Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6), RFC 2463, December 1998.

A. Conta, S. Deering, Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6), Internet Draft, draft-ietf-ipngwg-icmp-v3-00.txt"> Internet Control, June 1999.

- **HOP BY HOP OPTIONS:**

C. Partridge, A. Jackson, IPv6 Router Alert Option, RFC2711, October 1999.

D. Borman, S. Deering, R. Hinden, IPv6 Jumbograms, RFC2675, August 1999.

- **MULTICAST:**

S. Deering, W. Fenner, B. Haberman, Multicast Listener Discovery (MLD) for IPv6, RFC2710, October 1999.

- **PATH MTU DISCOVERY:**

J. McCann, S. Deering, J. Mogul, Path MTU Discovery for IP version 6, RFC1981, August 1996.

- **HEADER COMPRESSION:**

M. Degermark, B. Nordgren, S. Pink, IP Header Compression, RFC2507, February 1999.

M. Engan, S. Casner, C. Bormann, IP Header Compression over PPP, RFC2509, February 1999.

S. Casner, V. Jacobson, Compressing IP/UDP/RTP Headers for Low-Speed Serial Links, RFC2508, February 1999.

- **PACKET TUNNELING:**

A. Conta, S. Deering, Generic Packet Tunneling in IPv6 Specification, RFC2473, December 1998.

- **DOMAIN NAME SYSTEM:**

S. Thomson, C. Huitema, DNS Extensions to support IP version 6, RFC 1886, December 1995.

M. Crawford, C. Huitema, S. Thomson, DNS Extensions to Support IPv6 Address Aggregation and Renumbering, RFC2874, July 2000.

M. Crawford, IPv6 Node Information Queries, Internet Draft, draft-ietf-ipngwg-icmp-name-lookups-07.txt, August 2000.

M. Crawford, Discovery of Resource Records Designating IPv6 Address prefixes, Internet Draft, draft-ietf-ipngwg-prefix-rr-disc-00.txt, November 2000.

- **TRANSITION MECHANISMS:**

R. Gilligan, E. Nordmark, Transition Mechanisms for IPv6 Hosts and Routers, RFC2893, August 2000.

D. Haskin, R. Callon, Routing Aspects Of IPv6 Transition, RFC 2185, September 1997.

B. Carpenter, K. Moore, Connection of IPv6 Domains via IPv4 Clouds without Explicit Tunnels, RFC3056, February 2001.

- **ROUTING:**

G. Malkin, R. Minnear, RIPng for IPv6, RFC 2080, January 1997.

R. Coltun, D. Ferguson, J. Moy, OSPF for IPv6, RFC 2740, December 1999.

T. Bates, R. Chandra, D. Katz, Y. Rekhter, RFC2283, Multiprotocol Extensions for BGP-4, February 1998.

R. Minnear, R. Hinden, IGRPng for IPv6, Internet Draft, November 1996.

- **RENUMBERING:**

M. Crawford, Router Renumbering for IPv6, RFC2894, August 2000.

- **SECURITY:**

IPng w.g. Co-Chairs, Statement on IPv6 Address Privacy, November 6, 1999.

S. Kent, R. Atkinson, Security Architecture for the Internet Protocol, RFC 2401, November 1998.

S. Kent, R. Atkinson, IP Authentication Header, RFC 2402, November 1998.

S. Kent, R. Atkinson, IP Encapsulating Security Payload (ESP), RFC 2406, November 1998.

P. Metzger, W. Simpson, IP Authentication using Keyed MD5, RFC 1828, August 1995.

P. Karn, P. Metzger, W. Simpson, The ESP DES-CBC Transform, RFC 1829, August 1995.

- **NEIGHBOR DISCOVERY:**

T. Narten, E. Nordmark, W. Simpson, Neighbor Discovery for IP Version 6 (IPv6), RFC2461, December 1998.

A. Conta, Extensions to IPv6 Neighbor Discovery for Inverse Discovery, RFC3122, June 2001.

- **AUTO CONFIGURATION:**

S. Thompson, T. Narten, IPv6 Stateless Address Autoconfiguration, RFC2462, December 1998.

T. Narten, R. Draves, Privacy Extensions for Stateless Address Autoconfiguration in IPv6, RFC3041 January 2001.

J. Bound, Dynamic Host Configuration Protocol for IPv6 (DHCPv6), Internet Draft, draft-ietf-dhc-dhcpv6-19.txt, June 2001.

C. Perkins, Extensions for DHCPv6, Internet Draft, draft-ietf-dhc-dhcpv6exts-13.txt, May 2000.

D. Thaler, Analysis of DNS Server Discovery Mechanisms for IPv6, Internet Draft, draft-ietf-ipngwg-dns-discovery-02.txt, July 2001.

R. Draves, Default Router Preferences and More-Specific Routes, Internet Draft, draft-ietf-ipngwg-router-selection-00.txt, May 2001.

- **PROGRAM INTERFACES:**

R. Gilligan, S. Thomson, J. Bound, W. Stevens, Basic Socket Interface Extensions for IPv6, RFC2553, March 1999.

R. Gilligan, S. Thomson, J. Bound, W. Stevens, Basic Socket Interface Extensions for IPv6, Internet Draft, draft-ietf-ipngwg-rfc2553bis-04.txt, January 2001.

W. Stevens, M. Thomas, Advanced Sockets API for IPv6 RFC 2292, February 1998.

W. Stevens, M. Thomas, Advanced Sockets API for IPv6 Internet Draft, draft-ietf-ipngwg-rfc2292bis-02.txt, November 2000.

- **OSI NSAP MAPPING:**

J. Bound, B. Carpenter, D. Harrington, J. Houldsworth, A. Lloyd, OSI NSAPs and IPv6, RFC1888, August 1996.

- **MOBILITY:**

D. Johnson, C. Perkins, Mobility Support in IPv6, Internet Draft, draft-ietf-mobileip-ipv6-13.txt, November 2000.

- **IPv6 OVER DIFFERENT MEDIA:**

M. Crawford, A Method for the Tranmission of IPv6 Packets over Ethernet Networks, RFC2464, December 1998.

M. Crawford, A Method for the Tranmission of IPv6 Packets over FDDI Networks, RFC2467, December 1998.

M. Crawford, T. Narten, S. Thomas, A Method for the Tranmission of IPv6 Packets over Token Ring Networks, RFC2470, December 1998.

B. Carpenter, C. Jung, Transmission of IPv6 Packets over IPv4 Domains without Explicit Tunnels, RFC2529, March 1999.

I. Souvatzis, A Method for the Transmission of IPv6 Packets over ARCnet Networks, RFC2497, January 1999.

D. Haskin, E. Allen, IP Version 6 over PPP, RFC2472, December 1998.

G. Armitage, P. Schulter, M. Jork, G. Harter, IPv6 over Non-Broadcast Multiple Access (NBMA) networks, RFC2491, January 1999.

G. Armitage, M. Jork, P. Schulter, G. Harter, IPv6 over ATM Networks, RFC2492, January 1999.

A. Conta, A. Malis, M. Mueller, Transmision of IPv6 Packets over Frame Relay Networks Specification, RFC 2590, May 1999.

D. Thaler, Transmission of IPv6 Packets over IEEE 1394 Networks, Internet Draft, draft-ietf-ipngwg-ipngwg-1394-02.txt, July 2001.

- **NETWORK MANAGEMENT:**

B. Haberman, R. Worzella, IP Version 6 Management Information Base for the Multicast Listener Discovery Protocol RFC3019, January 2001.

M. Daniele, B. Haberman, S. Routhier, J. Schoenwaelder, Textual Conventions for Internet Network Addresses, RFC2851, June 2000.

D. Haskin, S. Onishi, Management Information Base for IP Version 6: Textual Conventions and General Group, RFC2465, December 1998.

D. Haskin, S. Onishi, Management Information Base for IP Version 6: ICMPv6 Group, RFC2466, December 1998.

M. Daniele, IPv6 Management Information Base for the Transmission Control Protocol, RFC2452, December 1998.

M. Daniele, IPv6 Management Information Base for the User Datagram Protocol, RFC2454, December 1998.

- **IPng AREA DIRECTORS RECOMMENDATION:**

S. Bradner, A. Mankin, The Recommendation for the IP Next Generation Protocol, RFC 1752, January 1995.

Internet Engineering Steering Group, Protocol Action: The Recommendation for the IP Next Generation Protocol to Proposed Standard November 19, 1994.