Jarkko Mäntysaari

Migration to a New Internet Protocol in Operator Network

Aalto University
School of Electrical Engineering

Thesis submitted for examination for the degree of Master of Science in Technology
Espoo 14.01.2013

Supervisor: Professor Raimo Kantola

Instructor: M.Sc. (Tech.) Zahed Iqbal

Abstract

AALTO YLIOPISTO
SÄHKÖTEKNIIKAN
KORKEAKOULU

MASTER'S THESIS
ABSTRACT

Author: Jarkko Mäntysaari
Title of the Thesis: IPv4 to IPv6 Migration in Operator Network
Date: 14.01.2013
Number of pages: 56

Department of Communications and Networking
Professorship: S-38 Networking Technology

Supervisor: Professor Raimo Kantola
Instructor: M.Sc. (Tech.) Zahed Iqbal

This thesis explains the differences between IPv4 and IPv6. Another important part of the thesis is to review the current readiness of IPv6 for worldwide production use. The status (in terms of readiness, adaptability, compatibility and co-existence) of IPv6 in TeliaSonera is discussed in more detail.

The most important reason for migrating to IPv6 is the address exhaustion of IPv4. This may not be a big problem in the developed countries but in developing countries the growth of Internet is fast and lots of more addresses are needed. The need for addresses is not only from computers but from many devices connected to the Internet.

Attempts to slow down the exhaustion of free addresses have been made but current solutions are not enough. IPv6 will solve the problem by using much longer addresses. It will also add security features and simplify headers to speed up routing.

TeliaSonera has started to roll out IPv6 services. At the beginning the corporate customers will receive IPv6 connectivity and consumers will follow later. TeliaSonera International Carrier is already serving its customers with IPv6.

It seems that IPv6 is ready, standards have been ready for years and support in devices and software is prevalent. To achieve and keep up the global connectivity, IPv6 is a must and should not be avoided.

Keywords: IPv4, IPv6, Internet Protocol, migration, planning

Prologue

The writing process of this thesis has been interesting. I have learned a lot of new things, especially those regarding the business world. The time has really been flying. I have tried to continue writing all the time but the writing took more time than I expected. There was really much to read, check and learn.

First of all I want to thank my wife Heli for bearing with me for almost nine months of continuous writing which was boring to her, I think. I also want to thank my friends from High School and University for encouraging me. They told me that this thesis will be ready someday when I thought it would never be completed.

For more to this thesis related thanks go to my supervisor, Professor Raimo Kantola. I want to thank my instructor and manager M.Sc. (Tech.) Zahed Iqbal for practical help in arrangements and making possible for me to do this work in TeliaSonera. Many thanks go also to Allu Helenius for helping me with spelling and grammar. Without him most of the readers would not understand anything about this text. Last but not least I want to thank my whole team for many discussions and hints helping to write the thesis!

Table of Contents

Index of Tables

Index of Figures

Symbols and Abbreviation

6rd	IPv6 Rapid Deployment
ACL	Access Control List
AfriNIC	African Network Information Centre
AH	Authentication Header
ALG	Application Layer Gateway
APNIC	Asia-Pacific Network Information Centre
ARP	Address Resolution Protocol
AS	Autonomous System
BGP	Border Gateway Protocol
BSD	Berkeley Software Distribution
CES	Customer Edge Switching
CETP	Customer Edge Traversal Protocol
CGN	Carrier Grade NAT (Network Address Translation)
CIDR	Classless Inter-Domain Routing
DAD	Duplicate Address Detection
DHCP	Dynamic Host Configuration Protocol
DHCPv6	Dynamic Host Configuration Protocol version 6
DNS	Domain Name System
DoS	Denial of Service
EGP	Exterior Gateway Protocol
ESP	Encapsulating Security Payload
EUI	Extended Unique Identifier
FQDN	Fully Qualified Domain Name
GB	Gigabyte
HTTP	Hypertext Transfer Protocol
IANA	Internet Assigned Numbers Authority
ICMP	Internet Control Message Protocol
ICMPv6	Internet Control Message Protocol version 6
IGP	Interior Gateway Protocol
IKE	Internet Key Exchange
IP	Internet Protocol
IPsec	Internet Protocol Security
IPv4	Internet Protocol version 4
IPv6	Internet Protocol version 6
IS-IS	Intermediate System to Intermediate System
ISP	Internet Service Provider
IT	Information Technology
IXP	Internet Exchange Point
LAN	Local Area Network
LIR	Local Internet Registry
NAT	Network Address Translation
ND	Neighbor Discovery Protocol
NIC	Network Interface Controller
NIDS	Network Intrusion Detection System
NPC	Network Control Protocol
MAC	Media Access Control
MTU	Maximum Transmission Unit
OECD	Organisation for Economic Co-operation and Development

OSPF	Open Shortest Path First
P2P	Peer-to-Peer
PRGW	Private Realm Gateway
QoS	Quality of Service
RE2EE	Routing Edge to Edge and through Ethernets
RIP	Routing Information Protocol
RIPE NCC	Réseaux IP Européens Network Coordination Centre
RIR	Regional internet registry
RTT	Round Trip Time
SEND	Secure Neighbor Discovery
SIP	Session Initiation Protocol
SLAAC	Stateless Address Autoconfiguration
SOHO	Small Office and Home Office
TCP	Transmission Control Protocol
TLD	Top-Level Domain
TS	TeliaSonera
TSF	TeliaSonera Finland
TSIC	TeliaSonera International Carrier
TSP	Tunnel Setup Protocol
UDP	User Datagram Protocol
UMTS	Universal Mobile Telecommunications System
UPnP	Universal Plug and Play
VoIP	Voice over IP
WWW	World Wide Web

1 Introduction

1.1 Background

Internet Protocol version 4 (IPv4) is a communications protocol designed to connect systems of packet-switched computer networks. The protocol transfers information in the form of small datagrams, pieces of data, from a source to a destination using fixed length addresses. IPv4 does not guarantee reliability of transmissions [1]. Internet Protocol version 6 (IPv6) is intended to become the successor of IPv4. The most important differences between IPv4 and IPv6 are expanded address space, simplifications in header format to speed up packet handling, improvements in support for expansions, flow level labeling capability and Security functions provided by now required Internet Protocol security (IPsec) support. [2]

It has been long known that IPv4 will run out of addresses. There are only about 4 billion (2^{32}) IPv4 addresses [1]. Ineffective usage of the addresses and new devices entering the Internet, for example smart phones, will deplete the current address pools in a couple of years. The use of private IPv4 addresses (Network Address Translation, NAT), Classless Inter-Domain Routing (CIDR) and reclamation of already allocated addresses have postponed the exhaustion but these methods will not provide more addresses forever. Currently all /8 networks (these networks have about 16 million addresses) are allocated to Regional Internet Registries (RIR) by Internet Assigned Numbers Authority (IANA). [3]

The solution for the IPv4 address exhaustion problem has been developed years ago: IPv6 was specified in 1995 [2]. IPv6 provides 128-bit addressing meaning that there are approximately $3{,}4 \times 10^{38}$ possible addresses to be used. IPv6 changed also the packet format to reduce the amount of computation needed in routers.

Current deployment of IPv6 is still in its infancy. According to measurements conducted by an American backbone Internet Service Provider (ISP) Hurricane Electric, autonomous systems (AS) using IPv4 outnumber AS's using IPv6 almost eight to eight [4]. This does not tell the exact usage numbers of IPv6 but gives us a hint that IPv6 is not yet ready to be the only protocol on the network layer.

The migration process will take years meaning that IPv4 and IPv6 will be used side-by-side. Transition mechanisms are needed to achieve global

connectivity between IPv4 and IPv6 hosts. Currently there are multiple used and standardized solutions for interconnection between IPv4 and IPv6 hosts [5].

The most well-known and the largest application of Internet Protocols is in the Internet. The current Internet spawned from ARPANET, a research network of the Department of Defense and several universities in the USA. In 1970, the network had grown to a point where a new host-to-host protocol was needed. The protocol was called Network Control Protocol (NPC). Because the original ARPANET had expanded into Internet, capabilities of NPC were not enough. The main idea of the Internet was that it would not be only a single network but it would be a network of networks. However, NPC had limitations with addressing hosts and networks, also error correction or detection was not enough for required reliability levels. [6]

The answer to the problem was a new protocol stack to replace the aging NPC. The new protocols were Transmission Control Protocol (TCP) and Internet Protocol (IP, currently known as IPv4). Robert E. Kahn and Vinton G. Cerf were the main architects of the new protocols [6]. It was noted quite early that the network would run out of host and network addresses. A temporary solution was to use CIDR [7]. In 1995 the first specification for IPv6, the successor of IPv4, was published by the Internet Engineering Task Force (IETF). [2]

1.2 Goals and objectives of the Thesis

The purpose of this work is to give guidelines and recommendations how to migrate from IPv4 to IPv6. There will be discussion regarding other protocols that could be used instead of IPv6 and reflections if any change is needed at all. This document will present the current state of IPv6 usage in TeliaSonera and also what the situation could, and should, be in a couple of years. A technical introduction will be given to IPv6 and differences between IPv4 and IPv6. The thesis will give a review of current situation of IPv6 in Finland like the situation is seen by FICORA, Finnish Communications Regulatory Authority.

Other considered topics are the IPv6 compatibility in operating systems, with applications and some network devices used in TeliaSonera. The thesis will also address the security viewpoint: what are the most important things to note when using both IP versions in a network. Techniques used to interconnect IPv4-only and IPv6-only networks will also be checked.

1.3 Scope of the Thesis

An overview will be given on differences of IPv4 and IPv6. Some services will be considered more carefully: Dynamic Host Configuration Protocol (DCHP), Domain Name System (DNS) and load balancers will be discussed in more depth. However, the purpose is not to list changes needed in configuration files or menus of single applications. The task will be accomplished as a review of literature and by interviewing TeliaSonera specialists.

Some measurements will be conducted in a network of TeliaSonera. The purpose of the measurements is to find out possible performance differences between native IPv4, native IPv6 and different tunneling methods. Performance measurements will be active point-to-point measurements for estimating latencies, jitter and throughput achieved by different technologies. A small scale study will be made how to migrate a network into using IPv6 in practice. All actual migration work towards implementing the change of the network protocol is out of scope of this thesis.

1.4 The Structure of the Thesis

This thesis is roughly divided into two parts. The part one, consisting of chapters 1 to 3, is about overall information about IPv6 and differences between IPv4 and IPv6. Chapter 2 describes the reasons for a need to migrate from using IPv4 to using IPv6. Some general information about IPv6 is also provided. Chapter 3 deals with issues that need to be considered during the migration. These issues include transition technologies and security. The IPv6 compatibility in operating systems and applications will also be reviewed as well as the current IPv6 usage.

The second part, chapters 4 and 5, of the thesis is related to the situation in TeliaSonera. Chapter 4 presents the current situation of TeliaSonera networks and a picture of what the network should be in a couple of years regarding to IPv6. The chapter will also include performance measurements and a study about migration. Chapter 5 concludes the thesis.

2 Why to Migrate?

This chapter will provide information about the differences between IPv4 and IPv6. It will also explain why it is important for companies and individual people to start considering issues related to IPv6. The current status of IPv4 address exhaustion and reasons for exhaustion will be investigated. The possibilities to obtain new addresses will be reviewed. Finally, the status of IPv6 deployment will be checked with special attention given to Finland.

2.1 Differences between IPv4 and IPv6

This section describes the most important differences between IPv4 and IPv6. The protocols contain noticeable differences, of which the addressing may be the most important and most well-known. Although operations of the both IP versions are similar, the differences make the protocols incompatible with each other.

2.1.1 Addresses and addressing

The IPv4 header contains both 32-bit source and destination addresses. This 32-bit addressing provides

$$2^{32} \approx 4,3 \times 10^{9} \approx 4.3 \text{ billion} \quad (1)$$

possible addresses [1]. IPv6, on the other hand, has an address space of 128 bits, resulting in

$$2^{128} \approx 3.4 \times 10^{38} \quad (2)$$

addresses. From here, it is possible to calculate that there are

$$\frac{2^{128}}{2^{32}} = 2^{96} \approx 7.9 \times 10^{28} \quad (3)$$

times more IPv6 addresses than IPv4 addresses. In other words, IPv6 provides more addresses per person alive than the number of all IPv4 addresses.

It should be noted that both protocols use a sizeable amount of address for different infrastructural purposes: multicasting, broadcasting, addresses of networks and loopback addresses. Therefore, only a part of the address space is reserved for global unicasting [3] [8]. Global unicast addresses are used to connect single machines or networks using Network Address Translation (NAT) to a larger network. In other words, the number of global unicast addresses, not the size of the whole address space, is the figure limiting the number of users in the network.

Subnetting in IPv6 appears more similar to IPv4 classful addressing than the current classless CIDR-addressing of IPv4. Although the size of a subnet can

be configured to be any number of bits, IETF has precise recommendations in which way the addressing should be done. Currently, it is recommended to use 64-bit network addresses leaving another 64 bits for identifying the interface in the subnet. Such a network could contain

$$2^{64} \approx 1.8 \times 10^{19} \quad (4)$$

addresses to be used in every subnet. When compared the IPv6 address space to the total number of addresses in IPv4 (Equation 3), it exists more large subnets for IPv6 than addresses in IPv4. The sizes of the address spaces of different versions of IP are shown in Figure 1. [8]

Large companies and ISPs will receive more than one subnet. In Europe RIPE, the local Regional Internet Registry (RIR), usually allocates /32 address spaces for ISPs. The largest customers for these ISPs usually obtain /48-sized networks to enable the internal subnetting of companies while still using /64 subnets as the smallest networks.

This may eventually lead to subnet exhaustion, as the operators have only a few bits to be allocated for their large customers. If the current addressing system would fail, the change of the addressing system would be easier than with IPv4, as more than 75% of the address space is still unallocated and reserved by the IETF [9]. It is also possible to use smaller subnets than /64, however, this will render Stateless Address Autoconfiguration (SLAAC) non-operational.

The IPv4 address is presented as four 8-bit decimal numbers separated by dots. The IPv6 address consists of eight groups of four hexadecimals separated by colons. Consecutive groups consisting of zeroes can be replaced by two colons. However, only one double colon per address can be used to maintain unambiguousness. Leading zeroes of the digit groups of the IPv6 address can be omitted.

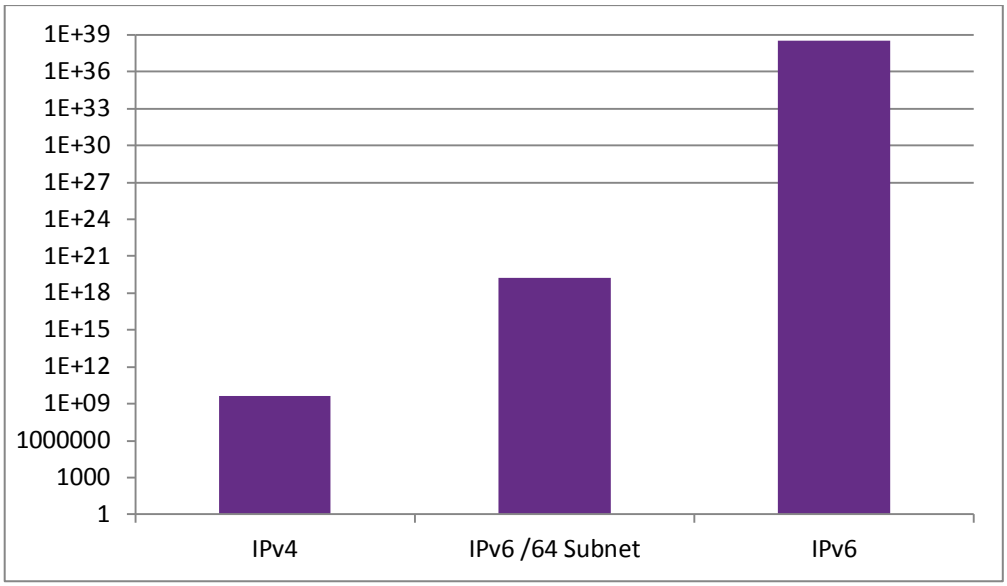

Figure 1 – Size of address space, note the logarithmic vertical axis

IPv6 has three different address classes: unicast, anycast and multicast. IPv6 has no broadcast capability similar to the one that IPv4 has. However, it is possible to use multicast for broadcast operations by using the all-nodes link-local multicast group. [8]

A unicast address identifies a single network interface. Packets sent to a unicast address are delivered to the interface identified by the destination address. Unicast addresses can be divided into groups. Link-local addresses are required for every interface in a network. Link-local addresses are used only on a single link or a broadcast area of Ethernet and routers must not forward packets with the link-local source or destination addresses. These addresses are used for configuration and discovery purposes and when no router exists in a network. Site-local unicast addresses were defined for similar use as the private addresses in IPv4, but their use is now deprecated. Global unicast addresses are used for communication between interfaces in different networks. The global unicast addresses are similar to the public IPv4 unicast addresses. Some IPv6 unicast addresses point to IPv4 addresses in order to help the interoperability of the protocols. Unspecified addresses and loopback addresses are special cases of unicast addresses. The loopback address is not assigned to any physical interface and it is used for intra-host communication. Routable loopback addresses can identify also hosts instead of interfaces. The loopback address specified for intra-host communication is not routable. [8]

The large address space of IPv6 allows stopping to use IPv4 style private addresses. However, the large demand for routable local addresses caused the IETF to allocate addresses for local use. Unique local unicast addresses are routed

in a network or networks administered by one entity. The addresses do not depend on addressing of the ISP of the network's owner. These addresses can be used, for example, in an enterprise network to keep its own addressing unchanged. Change of the ISP would not change the internal addressing. [10]

An anycast address is an address assigned to more than one interface, usually situated in different hosts. Packets destined to an anycast address are routed to the nearest interface bound to the anycast address. The "nearest" interface is measured by the routing protocols' measure of distance [8]. One possible use of anycast is to load balance traffic of an entity providing services in wide area to locations nearer to the customer, lowering latency and load of the network.

A multicast address in IPv6 identifies a group of interfaces. Multicast addresses can have different scopes, such as interface-local, link-local, site-local and global scopes. The scopes are intended to define destinations more specifically than for example "all routers". The scopes allow the transmission of packets to "all routers in this network". Currently allocated IPv6 address ranges are presented in Table 1.

Table 1 – IPv6 addresses

Address type	**Binary prefix**	**IPv6 notation**
Unspecified	000…000 (128 bits)	::/128
Loopback	000…001 (128 bits)	::1/128
IPv4-mapped	000…000111111111111111 (96 bits)	::FFFF:<IPv4-address>
Multicast	11111111	FF00::/8
Link-local unicast	1111111010	FE80::/10
Local unicast	1111110	FC00::/7
Unicast/Anycast	Everything else	

A host is required to recognize a link-local address for each network interface, additional unicast and anycast addresses configured for its interfaces, a loopback address, all-nodes multicast address and a multicast address of the groups that the host belongs to. Routers are required to recognize all the same addresses as hosts and also subnet-router anycast addresses for all interfaces, any other configured anycast addresses and the all-routers multicast address. [8]

2.1.2 Headers

The header in IPv6 (Table 3) is more simplified than the IPv4 header (Table 2). This is achieved by fixed length headers and extension headers. Next header field

points to the next extension header or the header of the payload, such as TCP or User Datagram Protocol (UDP). This mitigates the specific protocol, option and header length fields. As the fragmentation is moved from routers to end-hosts, the fields used for the fragmentation information in IPv4 (Table 2 – bits 32-63) are absent from IPv6. [1] [11]

Table 2 - IPv4 Packet

	0 1 2 3	4 5 6 7	8 9 0 1 2 3	4 5	6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
0	Version	Header Length	Diff. Serv.	ECN	Total Length
32	Identification			Flags	Fragment Offset
64	Time to Live		Protocol		Header Checksum
96	Source IP Address				
128	Destination IP Address				
160	Options if Header length is larger than 5 (measured in 4 bytes)				
160 or 192+	Data				

Table 3 - IPv6 Packet

	0 1 2 3	4 5 6 7	8 9 0 1	2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
0	Version	Traffic Class	Flow Label	
32	Payload Length		Next Header	Hop Limit
64	Source Address			
96				
128				
160				
192	Destination Address			
224				
256				
288				
320+	Data (or next header)			

Time to Live (TTL) field in IPv4 header is renamed to Hop Limit in IPv6. The word “time” gave an impression of a field that measured time, for example, in seconds, while the purpose was that “time” actually referred to hops between nodes in the network. The header checksum field is also removed from the IPv6 header. If a bit error occurs in the IPv4 header, the packet is dropped or an Internet Control Message Protocol (ICMP) packet is sent to the source address of the original packet. The source address field of the IPv4 packet without the options field consumes 20% of the bits in the header indicating a twenty percent chance that the source address is the corrupted part of the header [1] [11]. The upper layer protocols are needed to protect the data if requiring reliability as there is a high probability that the message notifying the sender about the error is sent to a wrong address. If a packet is lost or dropped because of errors in the IPv6 header, upper layers should also provide the reliability if it is needed, detect the missing packet and request retransmission.

Another reason to remove the checksum was to release some work from routers. As every router using IPv4 must decrease the TTL value, they also need to change the header of the packet. This led to a need of recalculation of the checksum and need for more computation resources in the router.

IPv6 has no options field for additional and optional information as already noted earlier. IPv6 uses extension headers to carry this information. The extension headers can be chained. The idea of header chaining is presented in Figure 2. If the IPv6 packet contains an IPsec packet that contains the data, the next header field in the IPv6 packet contains a pointer to an Encapsulating Security Payload (ESP) packet. Now, the next header of the ESP packet contains a pointer to the data which can be, for example, a TCP header or an UDP header. Extension headers can also be used to deliver jumbograms, which allow a single packet to be one byte less than 4 Gigabytes (GB) [11].

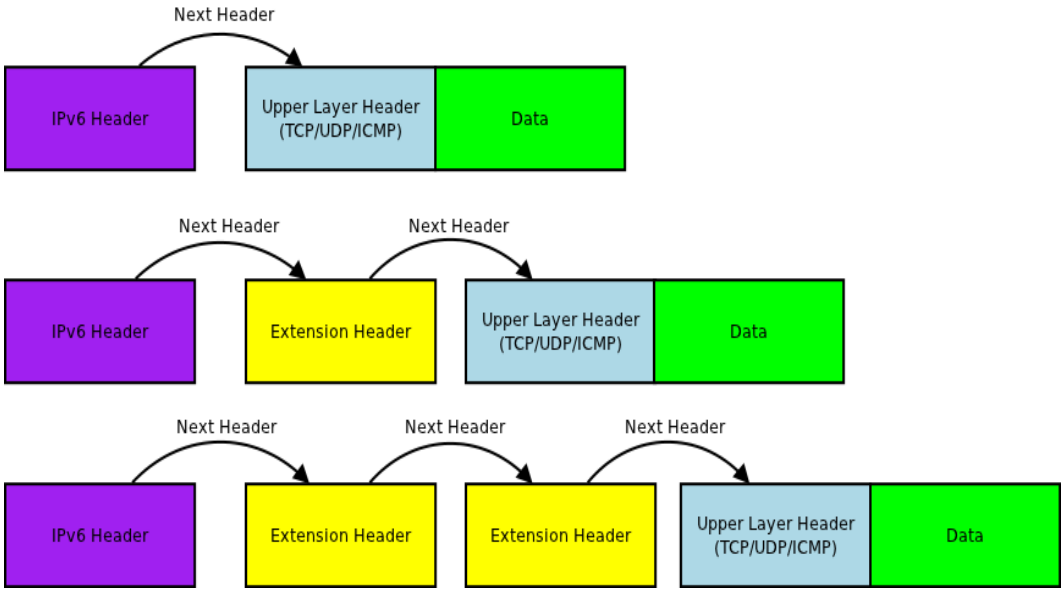

Figure 2 - Header chaining in IPv6

2.1.3 Configuring interfaces

Automatic configuration of the IP address, the DNS servers and the default gateway in IPv4 is performed by Dynamic Host Configuration Protocol (DHCP). IPv6 has a comparable protocol: Dynamic Host Configuration version 6 (DHCPv6). Using DHCPv6 is also called using stateful configuration.

Address Resolution Protocol (ARP) is the part of the IPv4 suite that is used to resolve the connection between link layer level addresses, usually Ethernet and MAC addresses and network layer level addresses: IPv4 addresses. ARP that operated on the same protocol layer as IPv4 on the IPv4 suite is removed from the IPv6 suite. The functionality of ARP is replaced by Neighbor Discovery Protocol (ND) of ICMPv6 on IPv6. ND incorporates ICMP Redirect and router discovery functions from IPv4 suite. ND uses link-local addresses to communicate with the neighbors before the interface has received a globally routable unicast address. Neighbor Discovery is also responsible for Duplicate Address Detection, Router Discovery and Prefix Discovery functionalities. [12]

Stateless Autoconfiguration (SLAAC) is a mechanism for configuring hosts without manual work on hosts and with minimal configuration needed in routers serving the configuration information for hosts. Additional servers are not required for SLAAC [13] [14]. SLAAC uses a Router Discovery functionality of ND to operate. It is also possible to use DHCPv6 and SLAAC simultaneously, to obtain the IP address with SLAAC and to use DHCPv6 to get other needed configuration information. SLAAC creates the address in two parts. The first part contains the first 64 bits of the address. This part is copied from the router

configuration. The second part, last 64 bits – called interface identifier - of the IPv6 address, identifies uniquely its interface. This part is generated from EUI-64 (Extended Unique Identifier) address, which is in most cases a filled up version of MAC-48 (Media Access Control) or EUI-48 address. EUI-48 and MAC-48 are basically synonyms. EUI or MAC address is usually used by Ethernet to identify any network interface controller (NIC) uniquely.

The method of generating addresses for hosts from MAC addresses led to concerns about the privacy of the Internet users. The interface identifier part of the IPv6 address remains the same even if the host is relocated to a different part of the network as the part is based on MAC address. It should be unique for every interface. This allows the use of the interface identifier to track hosts in a network. Server administrators could identify a user connecting to their servers on multiple WWW sites and could easily track sites visited by users. [15]

To prevent the previous problem with privacy, a new approach to generate interface identifiers was needed. As 64 bits need to be random generated, there will be lots of different possibilities (see Equation 4) for the interface identifiers. As the current need for addresses based on the number of IPv4 addresses is use – even if NAT usage is taken into account – is much smaller than what can be provided with 64 bits, address collisions are unlikely to occur. The probability of an address collision is even smaller as the used addresses are divided into multiple subnets.

Duplicate Address Detection (DAD) is a mechanism to lower the probability of address collisions even further. The detection of duplicate addresses must be performed on all unicast addresses before binding the addresses to a network interface. DAD must be used in all the possible cases of the IP address configuration: stateless and stateful (DHCPv6) autoconfiguration and manual configuration of the interface. [14]

Configuration of routers, on the other hand, is a special case. The source of information for SLAAC is often the router addresses and prefixes advertised by the router to other hosts in the network. Usually routers must be configured manually. Without the correct autoconfiguration information from a router, a host can only generate a link-local address. These addresses can be used to communicate with other hosts on the same local network. [14]

2.1.4 Mobility

The purpose of the mobility functions of both IPs is to allow mobile nodes, such as mobile phones, to continuously communicate while roaming around the

network. The problem with both Internet Protocols is that when a node changes to another base station or access point, the IP network and address of the node changes with a high probability. IPv6 simplified the mobile usage over IPv4.

Mobile IPv6 operates by using a home agent in the home network of the mobile node. The home agent tunnels the traffic destined to the mobile node into the current IP address of the mobile node. A triangular routing can be used to counter inefficient routing: the traffic going through the home agent between the mobile node and the correspondent node. Triangular routing means that traffic from the mobile node to the correspondent node will be routed straight from the source to the destination while traffic from the correspondent node to the mobile node will travel through the home agent. Routing can be optimized by binding to the mobile node and the correspondent node to route the traffic without passing it through the home agent. The bindings should be performed using IPsec to prevent man-in-the-middle attacks. [16]

The mobility support in IPv6 simplifies the mobile implementation of IPv4 by removing the need for a foreign agent and requires no support from the local router in a place where the mobile node is visiting. Mobile IPv6 is decoupled from specific link-layer protocols as IPv6 uses Neighbor Discovery instead of ARP allowing easier implementations on different networks.

2.1.5 Security changes

Internet Protocol Security (IPsec) is a protocol suite to provide secure communications for both Internet Protocols [17]. As the specification for IPsec is newer than the specification of IPv4, IPsec is an optional addition to IPv4. However, with IPv6, IPsec is a required part of the protocol [11]. IPsec provides confidentiality by encrypting the payload data using Encapsulating Security Payloads (ESP). Integrity, ensuring that the information has not been changed, is provided by Authentication Headers (AH) and ESP. Authentication of the origin of the data is also provided by ESP and AH. The third protocol, Internet Key Exchange (IKE), is responsible of initial functionality used in establishing the secured communication between the endpoints. IPsec headers in IPv6 are chained to IPv6 header as extension headers.

IPSec provides partial traffic flow level confidentiality. The protocol can be used to tunnel traffic between two collaborating networks. Between the networks where the tunnel is used, real source and destination addresses of the packets cannot be seen; only the addresses of tunnel's endpoints are visible to everyone.

Other features that contribute to the security architecture of IPv6 exist as well. However, I consider these features a side effect. There are features that have been designed primarily for some other purpose but those also affect security. More of these features can be found in section "3.3 Security".

2.2 Address shortage

IPv4 was designed as a test protocol for experimenting with packet switched networks. The plan was to test a packet switched network with IPv4, which had more than enough addresses for the test network of the Department of Defense of the USA. The network could not in any case contain millions of networked devices, it would have been too expensive. Researchers working on the IP technology thought that if the test network was successful, they would have time to implement a production version of the protocol. As it has been previously mentioned in this thesis, this was not the case and what is currently known as IPv4 somehow escaped to commercial use. [18]

The original specification of IPv4, RFC 791, (Request for Comments, the documents defining the standards related to the IP suite) divided the IP address space into 5 classes. 87.5% of the whole address space was assigned to unicast addresses (including the private addresses). Although this left more than 3 billion addresses for unicasting, the address allocation was very inefficient. As the address spaces of different classes were totally in different magnitudes, numerous addresses were left unused: most companies or entities had more than 256 hosts for their network but 65 536 addresses would have been an overkill (see Table 4). Many larger companies and entities, which were not necessarily network operators, received a full Class A network. [3]

Table 4 – IPv4 classful (original) address allocation plan

Class	Leading bits	Size of network number bit field	Number of networks	Addresses per network	Start address	End address
Class A	0	8	128	16 777 216	0.0.0.0	127.255.255.255
Class B	10	16	16 384	65 536	128.0.0.0	191.255.255.255
Class C	110	24	2 097 152	256	192.0.0.0	223.255.255.255
Class D (multicast)	1110				224.0.0.0	239.255.255.255
Class E (reserved)	1111				240.0.0.0	255.255.255.255

A large number of addresses were given to different entities before IANA started borrowing the addresses. Today IANA borrows the addresses: unused

addresses can be taken back to the address pool and reallocated after a guarantee time. The A class networks and addresses, on the other hand, were yielded to their users and getting back the unused addresses of these networks is much more difficult. Although CIDR was introduced in 1993 and the address allocation became more efficient, it only slowed down the rate of the address exhaustion. Even with CIDR the size of networks increases in powers of two, so the addresses-in-use-ratio may not be always very good. Use of Network Address Translation (NAT) slowed more the allocation need of the IPv4 addresses. However, the growth of the mobile Internet usage and especially growth in Asia has made the address exhaustion very imminent.

One thing contributing a bit to the address exhaustion of IPv4 could be the transformation of home users' connections: from dial-up connections being online from time to time to broadband connections being online practically all the time. Another step of progress in the Information Technology (IT) world speeding up the address allocation is virtualization: as setting up new computers has become extremely easy, and without practically any additional cost on top of already purchased hardware, individual servers can be set up for serving a single network service using a single IP address. With physical servers it would not be resource-wise efficient to provide a server for a little used service but multiple services could be bundled into a single server machine using a single IPv4 address.

IANA allocated the last of its free /8 networks (networks with the size of class A) in February 3rd 2011. RIRs still have a couple of /8 network blocks unallocated to Local Internet Registries (LIR) or companies except for Asia Pacific Network Information Center (APNIC) that was first to allocate all /8 networks [19]. Addresses of the last /8 networks will be allocated to users or LIRs in small blocks (1024 addresses) meant for IPv4 to IPv6 transition, not to be delivered to end users. These small blocks are allocated with the prerequisite of having already an IPv6 address allocation. Currently other RIRs than APNIC are able to serve new IPv4 addresses to their customers. However, with the current rate of the address allocation the Réseaux IP Européens Network Coordination Centre (RIPE NCC) in Europe and north-western Asia will be the second RIR to run out of addresses somewhere in the latter part of the year 2012. Last free IPv4 addresses in world will be allocated in 2015 by the African Network Information Center (AfriNIC). [19]

2.3 Other possibilities

Is IPv6 the only possibility to provide the Internet access for everyone? Is there another, better solution for exhaustion of the IPv4 addresses? This section tries to shortly answer these questions by reviewing couple of alternatives.

2.3.1 Using Network Address Translation

One solution could be to do nothing new and continue by even more intensive use of NAT. Deploying multiple layers of NAT devices would provide enough addresses for the foreseeable future. A multilayer NAT will break the end-to-end connectivity on the Internet. With a single NAT router administered by the user, it is still possible to use port forwarding to allow use of servers, such as a personal HTTP server. With multiple NATs administered by different parties, the connectivity from the public network is restricted even if the user wants the connectivity. Using multiple consecutive NATs for consumer customer networks would not help to preserve IP addresses as consumer customer networks are usually small, one NAT device is enough.

Operators NATting customers would save IP addresses, especially if the customers would use their own NAT devices too. However, running servers would become difficult for the customers as the operator should do the traffic forwarding from their NAT towards the customer. Still the scalability of the device doing the address translation could be a problem if the operator would try to use only a couple of addresses for consumer customers and allocate the public IP addresses for better paying business customers.

While the use of NAT devices is usually outsourced to the end customers by telling about the security benefits of NAT, it is also possible for ISPs to deploy NATs. By using NAT itself instead of letting the users to NAT themselves, ISPs can ensure the minimal usage of the public IPv4 addresses. I think one of the best sides of the Carrier Grade NAT (CGN) system is that it decreases the quality of service (QoS) experienced by the end users. The decrease would be possible to circumvent by using voluntarily IPv6 with globally routable unicast addresses. This may force and drive more users towards IPv6. However, currently the situation seems to that customers are waiting for operators, not vice versa.

CGN has been criticized for the same reasons as a classical NAT: Moving the intelligence towards the core of the network and limiting the end-to-end-principle. A special problem presented by the CGN is the scalability of the system. A huge amount of state information about ongoing sessions will need to be stored all the time. To provide a possibility to trace malicious Internet users, like hackers and spammers, this state information should also be logged. Deployments of CGN would also make it more difficult for the end users to keep their own servers running as there would be one extra level of NAT needing configuration if the server needs to be reachable from outside of the ISP's network. Using Universal Plug and Play (UPnP) could create a security nightmare

and using manual port forwards, on the other hand, would create an administrative chaos. [20]

One proposed solution to problems with the scalability of classical NATs is A+P NAT. A+P comes from "Address + Port" as the idea of A+P NAT is to extend the IPv4 address space by borrowing some bits of TCP or UDP port numbers for use of addressing the hosts. The proposal is to use 9 or 10 bits of the port number to expand the IPv4 address space. This would allow multiplexing 512 or 1024 users on the same IP address. However, the number of the ports that can be used by a single host would be reduced to 128 or 64 as the number of bits left for specifying the port would be 7 or 6. The multiplexing is designed to happen at ISP provided modems or routers meaning that no changes would be needed to the devices of the customers. [20]

Another place needing to be A+P NAT capable would be the provider edge routers, the routers connecting the ISP to other ISPs. The core network with correctly set-up tunneling would need no modifications to provide A+P NAT capability for the network. ICMP is more problematic as it has no port numbers. Other portless protocols should be very little used by the end customers. The A+P NAT provides some kind of solution for the problem: ICMP packets can be generally divided into two categories: error and echo (also known as "ping"). The error messages need to contain a part of the packet causing the error [21]. For echo messages the proposition is to rewrite the sequence number and identifier fields at A+P NAT devices to provide enough information for delivering the packets. Fragmented IP packets also create problems as only the first fragment has the TCP or UDP header with port numbers. This means that the A+P NAT device needs to reassemble the original packet to be able to forward all the data to the correct receiver.

The problems of the A+P NAT are also problems in classical NAT but even the authors of A+P NAT admit that their solution will probably make some of the problems even worse [20]. Multiplexing many more addresses into a couple of public IPv4 addresses compared to NAT, will make the design of the system more complex. In my opinion, the number of ports left for each user is a problem. Especially P2P (Peer-to-Peer) software creates numerous connections easily exhausting the ports. The given smaller limit of 64 ports can be easily used just by the web browser. The problem of multiplexing based on ports will also render well known port numbers, for example TCP port 80 for HTTP, unusable for most of the users; only one of the users using the particular public IP address can run a server using a specific port. The need for waiting for reassembling the fragmented packets could also be used to consume the resources of the A+P NAT device.

2.3.2 More ideas

The idea of A+P NAT could be used also inside the IPv4 header. Reassigning a little used fields, like differentiated services, would give 6 more bits for the addresses; 3 bits for the source address and 3 bits for the destination address. Adding three more bits would give 8 times more addresses which should be enough for a short time to allow transition to better technologies. Another possibility would be to use options field in the IPv4 header to increase the address space. This would give more addresses but also add more overhead to each packet.

The major problem with these approaches is that neither of them are supported currently. Adding the support all networking software would take as much work as with starting to use IPv6. This renders both approaches practically unusable.

Another possible solution to replace IPv4 would be to use of Routing Edge to Edge through Ethernets (RE2EE) protocol [22]. Private Realm Gateway (PRGW) is part of RE2EE. PRGW replaces traditional NAT devices. Outgoing (initiated from the private network to the public network) traffic flows and is translated as in NAT. Incoming (from the public to the private) traffic uses DNS and Fully Qualified Domain Names (FQDN) to identify hosts behind the PRGW device. A connection state is created after a host from a public network has made a DNS query and sent the first packet of the actual connection. In a time window between the DNS query and the first packet of the actual connection, the connection state is called “waiting state”. While in waiting state, the connection reserves one of the public IP addresses of the PRGW device. [23]

PRGW allows global connectivity and does not require additional servers. It does not also add delay for the connection establishment or the actual connection. However Application Layer Gateways (ALG) will be needed as some protocols carry IP addresses in their data. These IP addresses must be translated in PRGW to allow smooth operation. The same drawback affects also NAT. Another drawback of PRGW is the possibility for resource exhaustion attacks by sending numerous false DNS queries. It is possible to defend PRGW against these attacks by filtering the malicious traffic. The functionality of PRGW requires an operational DNS service. This requirement lowers reliability but even with the plain IP in use, the Internet would not be usable without DNS. [23]

Another concept in RE2EE is Customer Edge Switching (CES). PRGWs can be seen as independent replacements of NAT or as a part of CES. The goal of CES is to place hosts into private networks and this way achieve more efficient

use of the IPv4 addresses. The best use of CES can be achieved when both endpoints are located in private networks behind PRGWs. In these cases the traffic can be easily and effectively tunneled through the core network as the endpoints on both sides of the core network are fully aware of CES. The protocol used for tunneling is Customer Edge Traversal Protocol (CETP). Still it is possible to connect CES enabled and IP only networks. RE2EE is in development stage and no production ready implementation exists. An operational research prototype is ready. [24]

3 Issues to be considered

The purpose of this chapter is to give an insight to what should be taken into account when starting to deploy IPv6. In which way are IPv4 and IPv6 networks connected to each other? What are the mechanisms for translating addresses between these network protocols? What about security? IPv6 compatibility is reviewed in different applications and operating systems. Finally a discussion of the current IPv6 usage will be presented: how much IPv6 is used, for what it is used and how it is used.

3.1 Current IPv6 usage

The current IPv6 standard, RFC 2460 [11], was finished in December of 1998. The main reason for developing IPv6 was the upcoming shortage of the IPv4 addresses. Even though the protocol meant to solve the shortage has been ready and available more than a decade, the deployment of IPv6 has been slow and most of the traffic flowing through the Internet is still transported by IPv4.

Real usage numbers of IPv6 are hard to obtain: there is no single point for the measurements in the Internet. One more thing making it more difficult to get exact numbers is to decide what should be the measures for the IPv6 usage: the number of the hosts capable to use IPv6, the number of the IPv6 capable networks or amount of traffic? Is the amount of traffic measured by the number of flows, packets or bytes?

The Organisation for Economic Co-operation and Development (OECD) published measurements of the IPv6 usage in April of 2010. In early 2010 the usage of IPv6 was growing faster than the usage of IPv4; however as the IPv6 usage is still very small the growth is not very fast in absolute numbers. While the client side support for IPv6 is good (see section "3.4 Current IPv6 support") the server side lacks behind. In January of 2010 only 1.45% of the top 1000 WWW sites offered IPv6 service. The number grew to 8% in March of 2010 when Google started providing IPv6. When checking a situation of the top 1 000 000 sites instead of the top 1000, the situation changes a lot worse: in March of 2010 only 0.16% of the sites supported IPv6 service. Google's experiment in 2009 showed that only 0.25% of visitors in the websites of Google used IPv6 [25]. Most likely the figures have risen in the couple of last years. The rise in percentages may be high but as the start point is low, the actual IPv6 usage is still small.

3.1.1 Infrastructure readiness

The allocation of the IPv6 addresses is one of the key measurements in interest towards a potential IPv6 deployment; without IPv6 addresses the use of IPv6 is impossible [25]. In April 2012 about 0.0039% of the IPv6 address space was allocated to different RIRs [19]. While in early 2010 the allocation ratio of IPv6 was 0.003% [25]. This gives about a one third growth in about 2 years for the number of the IPv6 address allocations.

Another way of measuring the readiness of the infrastructure for IPv6 is to see the number of advertised IPv6 networks or ASes. In April 3rd of 2012 there were 5467 IPv6 capable ASes of which 139 were IPv6 only. As the total number of the autonomous systems at the same time was 40903; about 13.3% of the all ASes in the Internet were capable to transfer IPv6 traffic [4]. The number of the IPv6 capable autonomous systems has risen up quickly; in 2010 5.5% of all the autonomous systems were able to handle IPv6. This means that the number of IPv6 capable ASes was about 2500.

One important part of the infrastructure are the end hosts. In 2010 approximately 90% of the installed operating systems were IPv6 ready, although some of them may need additional configuration to use IPv6 [25]. According to a research conducted by Wikimedia of the users visiting their websites more than 95% of the used operating systems were IPv6 ready in 2011. It should be noted that the reliability of the research can be questioned, the page loading requests were counted to calculate the share of each operating system. Even though the number of individual page requests is huge it may be biased; for example it is usually easier to load multiple pages with a desktop computer than with a mobile phone. It is also possible to spoof the requests to present your operating system as another operating system [26]. The statistics are from WWW servers of a single foundation although these sites are not probably biased towards users of a specific operating system. More information can be found in section "3.4.1 Operating systems".

The IPv6 support is needed in DNS to allow IPv6 hosts to reach other IPv6 hosts. The DNS data can also be used to estimate IPv6 support in content providers. In January of 2010 7 out of 13 root DNS servers could be contacted using IPv6. At the same time 65% of top-level domains (TLD) had IPv6 records in the root DNS zone. 80% of the TLDs had a server or servers with IPv6 connectivity. There were 1.5 million domain names with IPv6 record, about 1% of all registered domain names at the time [25]. In April of 2012 total number of registered domains had risen to about 155 million while the number of the IPv6 records was about 3.2 million giving the figure of about 2%. [4]

3.1.2 Indicators of actual IPv6 usage in the Internet

According to the OECD the actual usage of IPv6 was still in its infancy in the latter part on year 2009. The considered data included IPv6 connectivity of the end users and observed IPv6 traffic. Although the usage was very low, it was growing. An experimentation of Google estimated that 0.25% of users of their sites were IPv6 capable. Universities and research institutions were the most active users of IPv6 with notable exception of a French operator Free.fr. However, as the Free.fr was using a transition mechanism to provide the IPv6 access; latencies of their users were higher than the latencies of universities using native IPv6. [25]

Free.fr reported that some 3% of global traffic of their customers used IPv6 in 2009. At the same time IPv6 traffic level at Amsterdam Internet Exchange Point (IXP); one of the largest IXPs; constituted only about 0.3% of the total traffic volume [25]. Even though the usage of IPv6 seems to be growing; the usage of IPv6 compared to IPv4 is currently (April of 2012) low and insignificant.

3.1.3 Survey Data

The European Commission conducted a survey on RIPE and APNIC service areas about what ISPs think about IPv6. The deployment levels of the European and Asian ISPs were similar although the interest to deploy or continue deploying was higher in Asia. Most of the respondents (80% in Asia and Europe) found the amount IPv6 traffic insignificant while 7% in Asia and 2% in Europe claimed that the amount of IPv4 and IPv6 traffic was approximately equal. The major barriers for the IPv6 deployment were cost of the transition and a lack of vendor support according to ISPs. [25]

3.1.4 Situation in Finland

In Finland FICORA, the Finnish Communications Regulatory Authority, conducted a survey in the spring of 2012 about the usage of IPv6 in 19 telecommunication companies active in Finland. Seven of the companies provided IPv6 services. However, IPv6 was mostly provided for the corporate customers, not for the consumer customers. The largest reasons for not providing IPv6 services were the lack of resources, lack of demand from the customers and large enough pool of free IPv4 addresses. [27]

While the situation in other parts of the world may not be similar (see section 2.2 Address shortage), the address shortage in Finland is not a major problem yet. As the population of Finland is growing very slowly and most people

and corporations already have the devices needed for operations, the usage of IP addresses is not likely going to explode.

Twelve of the 19 surveyed operators planned to start or to expand IPv6 deployment during the year 2012. One of the operators planned to be IPv6 only before 2017. The risks seen in IPv6 deployment were mostly related to IPv6 support in network nodes and the current expertise of the employees administering IPv6 networks. IPv6 support of the devices, possibilities for misconfiguration and lacking education and experience with IPv6 were mentioned. Other items that were considered were autoconfiguration and how it will operate in LAN (Local Area Network). Also, the privacy issues related to autoconfiguration were mentioned. The challenges and the difficulties that the operators had experienced included the already discussed lack of experience and software support. Generally IPv6 deployment had a low priority. [27]

3.2 Transition technologies

The transition from IPv4 to IPv6 will continue for years. IPv4 will probably coexist a long time with IPv6. The protocols are not directly compatible. Different technologies will be needed to allow connections from IPv4 hosts to IPv6 hosts. Sending traffic from an IPv6 network to an IPv4 network is easy, IPv6 has enough addresses to refer to all IPv4 addresses and many addresses are still left for other uses. The other way, from IPv4 to IPv6, is not trivial and many solutions have been proposed.

The solutions (also known as the mechanisms) can be divided into three categories: dual stack, tunneling and translation. Some solutions from each category will be reviewed next. Some operators are providing their own closed solutions. The closed solutions are usually only for their own customers or need a registration. The operator specific solutions will be out of the scope of this thesis.

3.2.1 Dual stack

Dual stack, also known as dual IP layer, is a networking protocol stack in an operating system providing complete support for IPv6 and IPv4. The dual stack allows a programmer to use both protocols transparently. In other words, the dual stack hosts can receive and send IPv6 and IPv4 packets implying that the dual stack hosts can communicate directly with IPv4 hosts using IPv4 and with IPv6 hosts using IPv6. Usually it is possible to configure the dual stack to use only one of the protocols while disabling the other. [28]

As the dual stack requires connectivity using either IPv4 or IPv6 for the whole path, it may not be enough to provide global connectivity. Dual stack hosts

can be configured to talk to each other using the transition technologies presented in the following sections. Dual stack adds more complexity and overhead resource-wise. However, other transitions techniques are adding even more complexity and overhead. Dual Stack also adds connectivity to both protocols easing the migration, so the dual stack should be used as a part of the transition. The difference of dual stack, IPv4-only and IPv6-only stacks is presented in Figure 3.

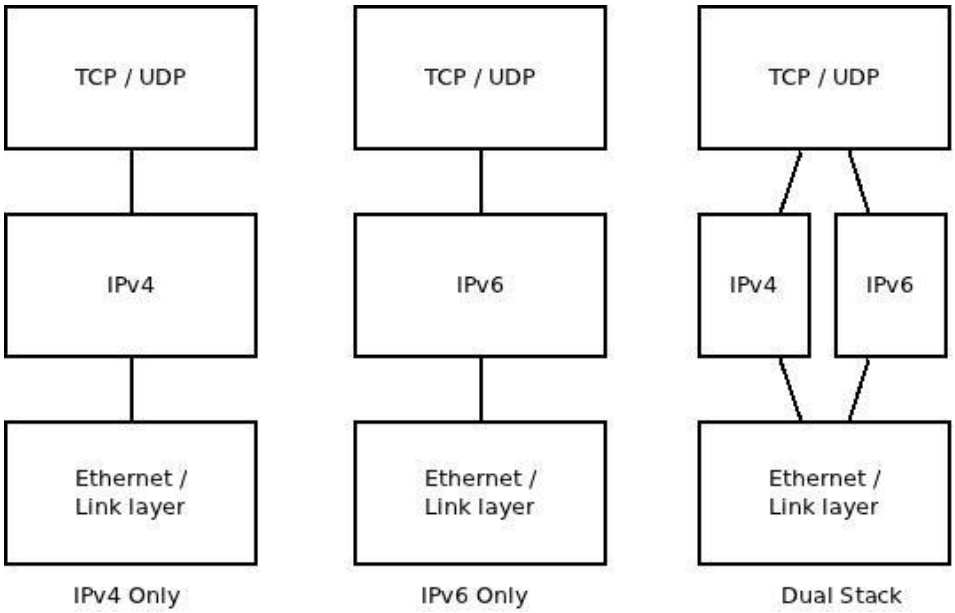

Figure 3 – Differences of IPv4 only, IPv6 only and dual stacks

3.2.2 Tunneling: 4in6 and 6in4

Protocols 4in6 and 6in4 are simple tunneling protocols. These tunnels can be configured either manually or automatically using, for example, the Tunnel Setup Protocol (TSP) [29]. In 4in6 an IPv4 packet is encapsulated into a IPv6 packet [30] and in 6in4 an IPv6 packet is encapsulated into a IPv4 packet [28]. A 6in4 tunnel is presented in Figure 4. These tunnels can be configured to transfer packets from a source host to a destination host, from a router in the middle to another router, from a source host to a router or from a router to a destination host. Both the protocols add an overhead of the header of encapsulated packet.

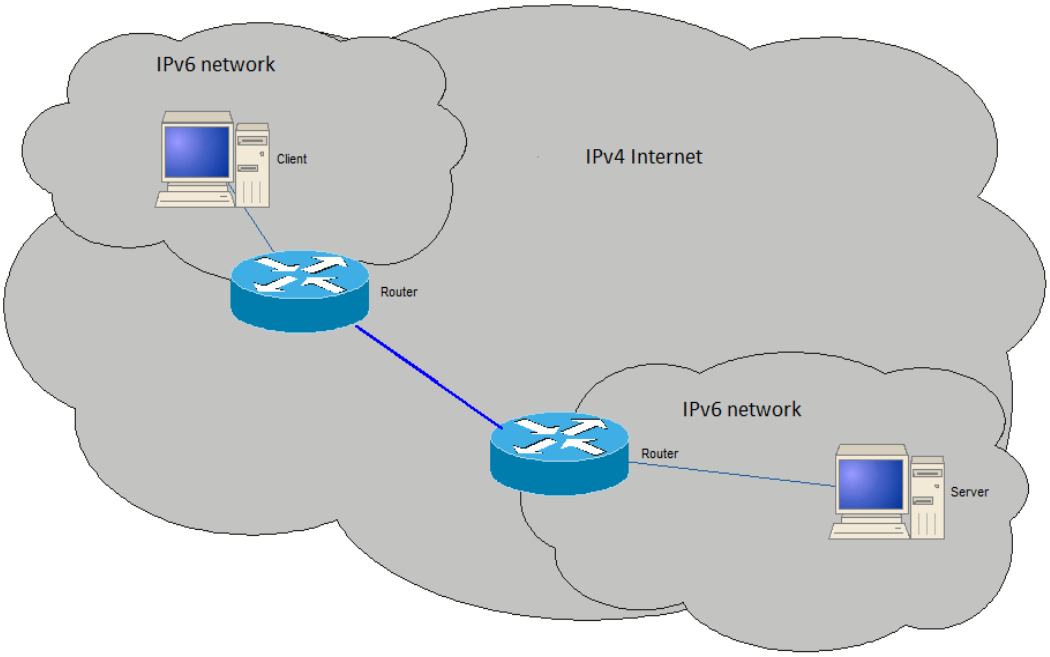

Figure 4 – IPv6 traffic tunneled over IPv4 using 6in4

NAT is usually a service-stopping feature for 4in6 and 6in4. As both protocols have another IP header as second header in the stack instead of TCP or UDP header, NAT devices will have difficulties translating the addresses. The usual solution of setting a server behind a NAT, called port forwarding, is impossible for the same reason.

As both protocols are simple tunneling methods, the protocols do not include Access Control Lists (ACL) or other security features. The previous indicates that when using IPv6 traffic encapsulated into IPv4 packets, it may be possible to get otherwise blocked IPv6 traffic through the firewall. The problem exists also with IPv4 packets encapsulated into IPv6 packets. Network administrators using these tunnels must be aware of this and make sure that the firewall blocking the unwanted traffic is not the only line-of-defense in the network: for example firewalling between different subnets and firewalling in the hosts could be used.

It is possible also for a 3rd party to slip packets into the tunnel to exploit the previously mentioned vulnerability. When the attacker is using source address spoofing, forging the source address of the sent data, the receiving end of the tunnel has no means of checking if the packet really is coming from the other end of the tunnel. To prevent these problems, IPsec should be used to transfer the tunneled protocol.

3.2.3 Tunneling: 6to4 and 6rd

A transition mechanism called 6to4 is used for from IPv4 to IPv6 migration. It allows IPv6 packets to be transmitted over an IPv4 network without a need to configure explicit tunnels. However, the protocol needs special relay servers for this function. 6to4 can be used on an individual host or by a local IPv6 network to connect to other IPv6 networks over an IPv4 only network. If 6to4 is used by a single host, the host must have a public IPv4 address. Networks using 6to4 have to have also a public IPv4 address but this is usually not a problem. [31]

The 6to4 protocol uses a specific IPv6 address format: the first 16 bits of 128 bits in the address are always hex "2002". The address format is shown in Table 5. The next 32 bits are the IPv4 address followed by 16 bits of arbitrary data. The last 64 bits are the host part of the address. These bits are needed for SLAAC (see chapter 2.1.3 Configuring interfaces) to be able to operate. It is not possible for IPv4-only and IPv6-only hosts to communicate with 6to4, it just allows communication between IPv6 nodes over an IPv4 network.

Table 5 – 6to4 address format

128 bits – IPv6 address			
16 bits	32 bits	16 bits	64 bits
2002: -prefix	IPv4-address of host	Arbitrary subnet address	Host address within the subnet

Allowing the use of 6to4 between hosts and networks requires use of relay routers. The relay router is connected to both IPv4 and IPv6 network. A relay router receiving a packet from an IPv4 interface will remove the encapsulation and forward the packet to an IPv6 network while when a relay router receives a packet with 2002 -prefix from an IPv6 interface the packet will be encapsulated and forwarded to the IPv4 network. The relay routers are used to interconnect a 6to4 network (a network using IPv6 internally and having 6to4 addresses) and a native IPv6 network. While a 6to4 border router (or just a 6to4 router) is used to connect to the 6to4 site. This architecture will lead to asymmetric routing when a 6to4 router relay is used. The asymmetric routing is a result of using anycast to locate the nearest relay. [31]

When a 6to4 host wants to communicate with a host in a native IPv6 network, it must have its IPv6 default gateway set to a 6to4 address containing the relay router's IPv4 address. To avoid manual configuration of IPv4 addresses of the gateways, IPv4 address 192.88.99.1 has been allocated as anycast address for finding the 6to4 relay routers [32].

IPv6 Rapid Deployment (6rd) is a transition mechanism derived from 6to4. Just like 6to4, 6rd is also designed to connect IPv6 sites via IPv4 networks. The most important difference between 6to4 and 6rd is that 6rd uses ISP's own IPv6 addresses instead of the 2002::/16 network block. This is why use of 6rd is limited so that all sites using 6rd and the block of unicast IPv6 addresses must be under the administrative control of one ISP or company. Another benefit of using 6rd instead of 6to4 is that 6rd removes the triangular routing. [33]

The migration mechanism 6rd uses an algorithmic mapping between the IPv4 and IPv6 addresses. This mapping allows automatic resolution of the IPv4 addresses of the tunnel endpoints from the IPv6 addresses. Because of this, 6rd can operate without any stored states. To map all IPv4 addresses to IPv6 addresses, a 32 bit IPv6 address space is needed. The address space consumption can be mitigated by omitting redundant parts of the IPv4 address space. [33]

3.2.4 Teredo

Like the previously presented protocols, Teredo is also an IPv4 to IPv6 transition mechanism. As Teredo traffic is tunneled using UDP over IPv4 (an IPv6 packet inside a UDP datagram) it is capable to function also behind a NAT unlike most of the other transition mechanisms [34]. The capability to operate even from behind a NAT router makes Teredo very suitable for a home and small office use.

Teredo client is a network node with access to the IPv4 Internet. The client may have a public IPv4 address or it may as well be situated behind a NAT device. The purpose of the client is to gain access to the IPv6 Internet. [34]

Teredo servers are used by the clients to detect automatically if they are located behind a NAT router and what kind of NAT it is. Clients keep sending UDP packets to a server regularly to maintain a possible NAT binding which allows the server to contact any of its clients at any time. A Teredo server will also be used as a middle point in communications between two Teredo clients. The server delivers an initialization message to the client. After this the client can make the required entries into the NAT table and allow bidirectional connectivity between the clients. The Teredo server transmits the ICMPv6 packets from clients to the IPv6 Internet. ICMPv6 echo messages (ping) are used by the clients to find Teredo relays with the help of Teredo server. The connection establishment procedure of Teredo is shown in Figure 5. [34]

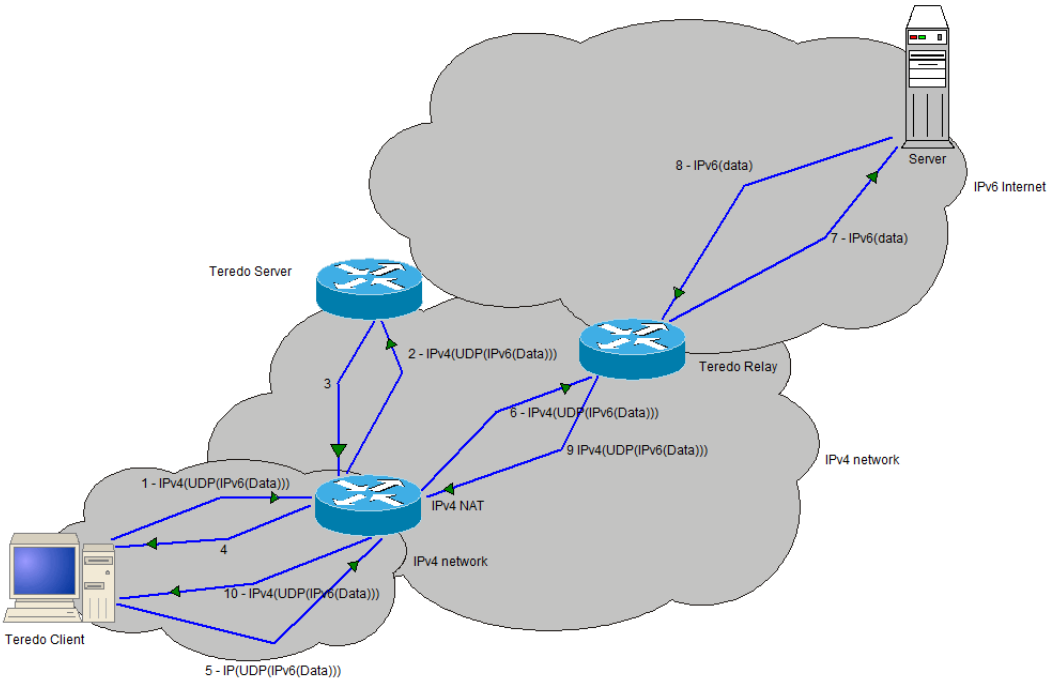

Figure 5 - Teredo connection establishment

As the Teredo servers are not actually relaying traffic, but only relay ICMPv6 traffic and messaging related to connection establishment and upkeep, the bandwidth requirements for Teredo servers are small. The amount of memory required by the Teredo server is also low, as there is no network or connection state that the server needs to maintain.

Teredo relays, the remote ends of the Teredo tunnel, are devices actually doing the translation from IPv4 to IPv6. For this reason the bandwidth requirements are high. Because Teredo relays need to advertise their prefixes to other IPv6 hosts, the administration of the relay must control the network in which the relay is used. Another option is to use Border Gateway Protocol (BGP) to advertise the relay to other networks. [34]

In Teredo the first 32 bits are used as the prefix to identify Teredo service and are “2001:0000”. The next 32 bits are the IPv4 address of the used Teredo Server. The bits from 64 to 79 are used for different flags. Currently all bits are not used: only higher order bits are in use to inform other parts of the Teredo service about the use of NAT. The next 16 bits are used to present UDP port number that is mapped by the NAT to the Teredo client. The last bits of the address are for the public IPv4 address of the client. The port and the IPv4 address bits are inverted bit by bit: 0 to 1 and 1 to 0. [34] The format of the Teredo address is presented in Table 6.

Table 6 - Teredo addressing

	128 bits				
Bits	0 - 31	32 - 63	64 - 79	80 - 95	96 - 127
Length	32 bits	32 bits	16 bits	16 bits	32 bits
Description	Prefix	Teredo server IPv4 address	Flags	Obfuscated UDP port	Obfuscated Client public IPv4

Although the NAT support with Teredo is better than with the other transition technologies, it is not perfect: Teredo is not able to operate behind every type of NAT. However some implementations have non-standard extensions to cope with the compatibility problems, but even these implementations have problems with connections from a Teredo client to another Teredo client.

Teredo tunnels can provide only one IPv6 address per tunnel endpoint. This means that connections from a client to different hosts need all their own Teredo tunnels. Combined with the fact, that the Teredo connection establishment is slow because of the multiple phases of the process the user experience when using Teredo may not be as good as possible, especially if the server and the relay are not topologically near the client. Although the cost of the connection establishment is insignificant with longer traffic flows, the shorter flows can introduce problems. In worst case scenarios, for example, DNS queries might need own Teredo connections.

3.2.5 NAT64 and DNS64

NAT64 refers to Network Address Translation from IPv6 to IPv4. NAT64 is a transition mechanism meant to allow IPv6 hosts to communicate with IPv4 servers. As the address space of IPv4 is considerably smaller than the address space of IPv6, the translation cannot be symmetric. This indicates that one-to-one mapping is impossible and connectivity from an IPv4 address to all IPv6 addresses cannot be provided. The specialty of NAT64 lies in the feature that the hosts can be IPv4-only and IPv6-only, no dual-stacking or tunneling is needed. [35]

The mappings of the NAT64 are configured in the same way as the port forwarding of the traditional NAT used also at homes. In general NAT64 is designed so that an IPv6 client is the one who initiates a new connection. However, some mechanisms exist allowing connection establishment by an IPv4 host: for example a static address mapping. Operation of NAT64 is illustrated in

Figure 6. “SYN” refers to TCP packets with a SYN bit set, “A” refers to a DNS reply with an IPv4 address and “AAAA” refers to DNS reply with an IPv6 address.

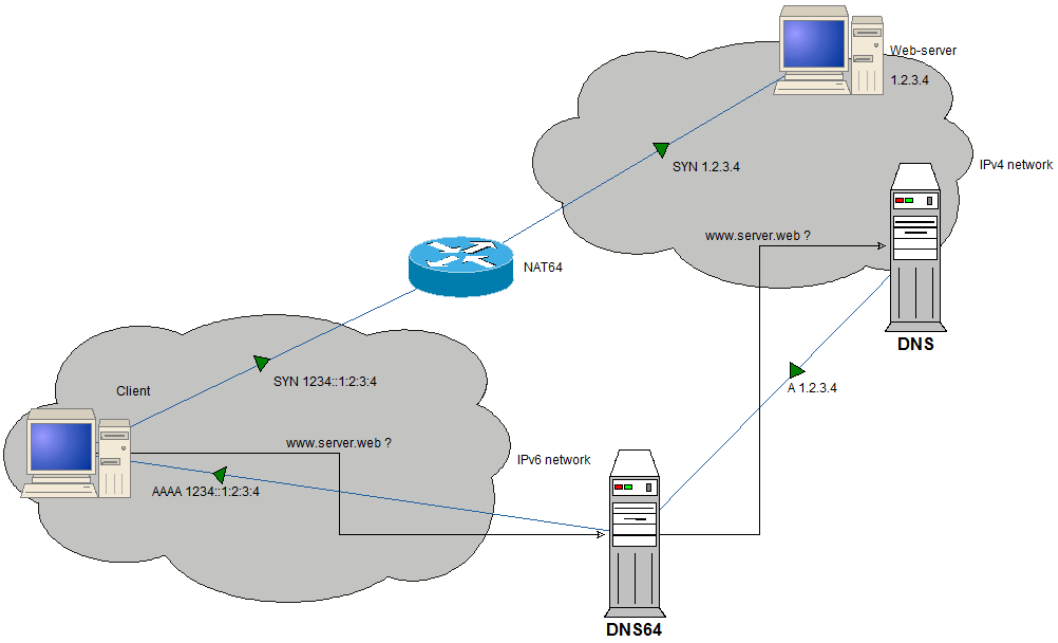

Figure 6 - NAT64 and DNS64

DNS64 describes a DNS server synthesizing an AAAA record from an A record. An AAAA record is created from an A record in case if no AAAA record is found for the requested domain name [36]. DNS64 is usually used with NAT64: when an AAAA record is requested for a server with only IPv4 connectivity; a DNS64 server answers with AAAA record pointing to NAT64 device which forwards the traffic to the desired server.

3.3 Security

This section covers issues related to the security of IPv4, IPv6 and the migration. The section mostly considers the differences between IPv4 and IPv6 as well as the migration. See also the section 2.1.5 “Security changes”.

One of the security problems in the migration from IPv4 to IPv6 will be that there are two network layer protocols coexisting. This means that the security of the both protocols needs to be handled: a security hole with one protocol provides a possibility to compromise the security of the whole network. While configurations of firewalls and other security measures may be quite similar for both of the Internet Protocols, configurations need to be made for both protocols separately and in some cases even for inter-protocol communications as well. [13]

One of the most significant differences for the consumer customers is that no need for NAT exists when using IPv6. NAT is not needed as the address space of IPv6 is larger than the address space of IPv4. However, many users have been thinking NAT as a security system. If NAT is removed because it is not needed for saving the addresses, its security functions can be replaced with something else, for example, with a firewall [37]. In general the vulnerabilities are very similar in both Internet Protocols. No new major vulnerabilities exist in IPv6 but not all of the IPv4 vulnerabilities are fixed either. The transition mechanisms (see "3.2 Transition technologies") also provide additional layers to be exploited, so administrators have to be careful to check that all the possible inbound routes to the network are protected.

Fingerprinting is a method used to identify operating systems by inspecting the packets that the hosts are sending. Fingerprinting is used as a reconnaissance before the actual attack. The RFC documents describe only the expected normal behavior. The RFC documents lack the definitions of operation in unexpected situations and do not define default values for some header fields (for example TTL field of IPv4 header [1]). The previous will make it possible to identify TCP stacks and operating systems using them. Fingerprinting in IPv6 world will not differ significantly from IPv4. As the upper layer protocols are not affected by the IPv4 to IPv6 migration, the fingerprinting possibilities remain unchanged. While some of the fields in the headers have changed when changing from IPv4 to IPv6, the basic functionalities of these fields remain very similar in both Internet Protocol versions [38]. The security of IPv6 is not in this sense getting better compared to IPv4, even though most likely the only security update is that there is no such a large fingerprint library at the beginning. This kind of pseudo-security will not help forever.

3.3.1 Vulnerability assessment

Port scanning is one way to find services on the Internet hosts. An attacker uses it to find potentially vulnerable hosts, victims or targets. Of course the security administrators try to target the same machines likewise. After locating the vulnerable machines, they can take other preventive measures like updating the vulnerable software and applications. As the port scanning happens on upper layers than the network layer, change from IPv4 to IPv6 does not affect it. While the actual port scanning is similar with both protocols, finding the hosts to be scanned is more difficult when using IPv6 [39]. Section "2.1.1 Addresses and addressing" presented discussion about differences of addressing between the Internet Protocols. Because of the large address space and a liberal address allocation used in IPv6, port scanning will be slower when using IPv6 only.

I think the slowness of scanning is a two edged sword: attackers are not able to find targets as fast but it is also harder to scan own networks for defensive purposes. Although the owners should know the used IP addresses, large networks make it hard to locate possible rogue devices. Even though a large address space may sound a good defense, it is important to notice that it will not completely prevent hacking attempts, merely slows them. This kind of security-through-obscurity provides no real security. An attacker with time and multiple computers to scan the network will eventually find the vulnerable machines. There is also a possibility to decrease the number of addresses which are needed to be scanned. When the target network of the scan is using SLAAC, the last 64 bits of the addresses are generated from the MAC addresses as discussed in section "2.1.3 Configuring interfaces".

Although IPv6 does not have ARP which can be used to find live systems on the local network, ICMPv6 messages to all active link-local addresses (ff02::1) can be used instead [8]. While restricting the host discovery with ARP would lead to a reduced connectivity, the case with IPv6 is not the same. It is possible to filter requests to multicast address ff02::1 to prevent the host discovery. While the filtering will render SLAAC non-operational, it is possible to use DHCPv6 instead. [39]

As simple brute-force scanning of every address in a subnet even with technical tricks to reduce the scanned address space is impractical, DNS will be the main source of information about the live hosts in the network. To be able to administer all the hosts in the network, administrators need a list of hosts in their network. One of the easiest ways could be the use of DNS which attackers might also be able to abuse. [40]

One more part of the vulnerability assessment is identifying i.e. fingerprinting the operating system. Fingerprinting allows identifying the operating system that is running on a remote targeted host. Identifying the operating system is an important part of the attack as with the knowledge of used OS, the attacker can try to exploit the known vulnerabilities in the operating system. As the RFCs do not tell how the network protocol stacks should operate in every situation, the designers of the operating systems have done the work. This leads a bit different implementations and allows identifying the operating system. However, the same basic techniques have also existed in IPv4 networks so also in this case the security of IPv6 is not inferior to that of IPv4. [38]

3.3.2 Transition mechanisms

As seen in the section "3.2 Transition technologies" multiple transition mechanisms are based on tunneling. The simplest firewalls are not able to investigate the whole packets but only the outermost IP header and the transport layer header (such as TCP or UDP header). This way tunneled traffic may get through the firewall uninspected. For example, when using 6in4, the traffic between the endpoints must be permitted to allow operation of 6in4. For more security related information see section "3.2.2 Tunneling: 4in6 and 6in4". If only the first IP header is checked, in this case all the traffic would be allowed though the firewall. This includes both, legitimate and malicious traffic. Of course, the same problem applies to IPv4-only networks as many types of data can be transported over HTTP [41]. The data transported over HTTP includes for example video streaming or instant messaging.

The technologies using automatic tunnel creation also introduce possibilities for DoS attacks, spoofing attacks and service thefts, in which someone is using resources without permission. Similar attacks exist also in IPv4-only networks, but IPv4 to IPv6 transition mechanisms provide additional attack surface to exploit the vulnerabilities. [41] [42]

3.3.3 Protocol vulnerabilities

In a man-in-the-middle attack the attacker usually reroutes the traffic between the victims through attacker's own systems to either eavesdrop or modify the data. In some cases attacker may be already on the route so there is no need for rerouting. Encrypted data transmissions with certificates or other means to authenticate endpoints are used to prevent man-in-the-middle attacks.

The man-in-the-middle attacks for IPv4 do not work for IPv6 but similar attacks are still possible. While a local network connection could be hijacked or eavesdropped with ARP poisoning when using IPv4, the Neighbor Discovery of IPv6 provides similar tools for an attacker. It is possible to circulate the traffic between the victims through the host of the attacker by sending spoofed Neighbor Discovery messages to both victims.

Another way to execute a man-in-the-middle attack using IPv6 is autoconfiguration spoofing. Just like with IPv4 and DHCP, the autoconfiguration with IPv6 and DHCPv6 is vulnerable for spoofed or malicious DHCP servers. SLAAC is also vulnerable to spoofed router advertisement messages. By spoofing the above-mentioned messages the attacker can configure the victim to use DNS

server or router of the attacker. This way the attacker can route desired traffic through his own hosts and alter or listen to the data. [39]

The defenses against the previous attacks are very similar to ones used with IPv4. Scanning and listening for rogue DHCP servers and checking that the router advertisement messages flowing in the network are valid together with general monitoring of the traffic in the network are the key to notice and prevent the possible attacks. It is also possible to use Secure Neighbor Discovery (SEND) protocol instead of normal Neighbor Discovery. Encryption providing the security may, however, increase needed resources for serving the network. [43]

A denial of Service (DoS) attack is an attack targeted against availability of a resource. The attack may consume resources (network bandwidth, CPU time, memory) or crash software or a device. [44]

One of the most devastating attacks with IPv4 was to send ICMP echo (Ping) packets to a broadcast address with a spoofed source address. As multiple hosts within the destination broadcast domain received an ICMP message they all sent a response amplifying the effect of the packet flood by the number of responding hosts in the broadcast domain. This way the victim of the attack, whose IP address was used as a source address of the original ICMP packet sent by the attacker, will now receive many times more traffic as the attacker has to send. Even though IPv6 does not have broadcast addresses the same techniques can be used with multicast address ff02::1 (all nodes). [39] The attack can be countered by configuring the systems not to respond to ICMP echo packets destined to “all something” multicast addresses.

Another way to perform a DoS attack is to use weaknesses in Duplicate Address Detection (DAD) (see section 2.1.3 Configuring interfaces). DAD checks the possible duplicate addresses by sending ICMPv6 neighbor solicitation messages. If the address that is being checked does not answer, the address is unused and free for use by the host making the check [14]. The attack works so that the attacker simply responds to every neighbor solicitation message sent to the network effectively blocking all the new hosts from joining the network. [39]

Most of other attacks are very similar to ones that can be used with IPv4. Packet fragmentation can be used to try to evade Network Intrusion Detection Systems (NIDS) just the same way with IPv6 as with IPv4. With IPv6 the only hosts allowed to fragment or assemble packets are the source and the destination hosts [11] making it more difficult for routers and firewalls on the route of the packet to inspect the packet by using NIDS. [40]

The mobile IPv6 (see section 2.1.4) is another potential weak point. Although it is required for a home agent and a correspondent node and to use IPsec to protect the integrity and authenticity of binding updates [16], most implementations have a possibility to operate also without the protection of IPsec [40]. Without the protection man-in-the-middle attacks are possible by sending forged binding updates so that the attacker poses as the home agent for the correspondent node and vice versa.

The problems with the upper protocol layers are not of course fixed with IPv6. For example using TCP RST packets (TCP-packets with the reset bit set to 1) or ICMP error packets to close BGP connections, which will cause a lot of damage, is not affected in any way by IPv6. Although IPv6 has not so far presented new security vulnerabilities and by making scanning more difficult has in fact made the situation better, there is still lots of problems. Van Hauser lists multiple IPv6 software related vulnerabilities and bugs, ranging from desktop to server operating systems and router software. [40]

3.4 Current IPv6 support

One of the first things needed to understand is that the IPv6 support in operating systems and applications are completely separated from each other. This means that running IPv4-only application on an operating system using the dual stack does not allow the application to use IPv6, the application and the operating system need to be designed to use IPv6. Even though IPv6 is the technology that will be used in future according to the IETF, they still recommend designing all software to be able to handle both Internet Protocols. [45]

3.4.1 Operating systems

Operating systems in this section are divided into two categories. Traditional computer category contains desktops, laptops and servers, the devices people usually call computers. The other category contains mobile devices, such as smart phones.

According to my own research on manufacturers web pages there will be no problems with the IPv6 support of the operating systems of the traditional computers. Practically all current the operating systems have fully operational and production quality IPv6 stacks.

The IPv6 support in operating systems should not be a problem for migration. According to a research made by Paul Weissmann for his thesis, all current operating systems have an operational IPv6 stack [46]. It should be noted that only some of the operating systems have IPv6 enabled as default. It also

seems that legacy or obsolete systems like Windows 9x do not have an official IPv6 support from their corresponding vendors.

Linux operating systems have had a production quality IPv6 implementation since kernel version 2.6. Microsoft Windows has had the IPv6 stack since Windows 2000. However, a production quality implementation has been shipped only since Windows XP. The support in Windows 2000 was for experimentation and development. [46] [47]

Mac OS X along with its roots in Berkeley Software Distribution (BSD) has also working IPv6 implementations. Even though BSDs and Linux usually have many common open source applications and pieces of software, the IPv6 implementations in these operating systems are not related. Other commercial UNIXes, at least the following ones: HP-UX, AIX and Solaris, have their own implementations of the IPv6 stack. [46]

The mobile operating system of Apple, IOS is based on MAC OS X and is also IPv6 compliant. According to multiple online forums Android and Symbian operating systems have also their IPv6 stacks. Windows phones do not support IPv6 in version 7 but according to multiple mobile phone news sites the mobile version of Windows 8 has added the support.

3.4.2 Application

The IPv6 support situation for the operating systems was good (see section 3.4.1 Operating systems). The situation of the IPv6 support in applications is also good. Most of the applications are IPv6 compatible and in the cases that applications have no IPv6 support there are applications that can be used to replace the ones without the support. [48]

The IPv6 application support list of Deep Space 6 is from the first half of year 2011. The situation should be now even better as developers have had more time to implement the IPv6 support for their software. The list of Deep Space 6 is mostly about applications for Linux. However, most of the applications are available for other operating systems, for example for Microsoft Windows and Apple's MAC OS X [48]. One of the most popular applications without IPv6 support is Skype, a peer-to-peer Voice-over-IP (VoIP) software. Java applications supporting the Java 1.4 standard have an IPv6 support.

The basic applications (such as ping or traceroute) seem to be able to handle IPv6 mainly without problems. However, there is no public knowledge of the custom applications made for different companies for specific purposes. There may not be publicly available software for these purposes. Companies and other

entities wanting to migrate to using IPv6 should give a special attention for these custom applications to make sure IPv6 will not cause problems in these applications.

To create connectivity across the Internet also the routing protocols need to be IPv6 compatible in addition to the network devices and the end hosts. Generally this does not create problems for non-IETF routing protocols which are designed to be independent of IP.

Interior Gateway Protocols (IGP) include widely used protocols like IS-IS, RIP (Routing Information Protocol) and OSPF (Open Shortest Path First). These protocols are used inside an autonomous system (AS). Only one entity is administering the routing making it easy to guarantee IPv6 compatibility. IS-IS is not designed by the IETF and it is working directly on top of the link-layer protocol. This means that IS-IS does not need either of IPs to operate allowing very easy adaptation to route IPv6 traffic. No changes to the protocol itself were needed [49]. RIP is the oldest of the previously mentioned routing IGPs. The earliest versions of RIP (RIPv1 and RIPv2) did not have a support for IPv6 and a support for CIDR was added in RIPv2. A new version of RIP was needed to support IPv6 and RIPng (Routing Information Protocol next generation) was developed [50]. OSPF is an IGP using the same kind of routing algorithm as IS-IS. As OSPF was designed by the IETF, it is very heavily geared towards routing in the IP networks. OSPFv3 is the first version of OSPF capable of routing IPv6 networks and the protocol had to be redesigned from OSPFv2 to cope with IPv6. [51]

Exterior Gateway Protocols (EGP) are meant to interconnect the networks created by the IGPs. While the network that IGP creates is usually controlled by a single entity, the EGPs are used to connect networks of completely different entities. Currently used EGP is Border Gateway Protocol (BGP). Even though BGP runs on top of TCP, which is an IETF designed protocol, it was quite easy to modify BGP to route the IPv6 networks as well. BGP had a possibility for extensions even before the need to route IPv6 was recognized. The Multiprotocol Extensions for BGP was created to enable IPv6 routing using the currently deployed protocol. [52]

3.4.3 Networking devices

This section does not try to provide a complete list but a short review to a situation of IPv6 support in different networking devices. Routers will have the focus. The other devices, such as VoIP devices and printers will be out of the scope of this review.

According to the homepages of the manufacturers, the IPv6 support varies in SOHO (Small Office, Home Office) routers. While most of the new models have an IPv6 support, the older models have a very different situation, even with only 3 to 5 years old models.

The firmware can be changed to a 3rd party firmware on some of the SOHO routers. This firmware, like OpenWRT, DD-WRT and Tomato projects seem to have a good situation with the IPv6 support. There projects are based on a Linux kernel so adding the support for IPv6 is only a question of computing resources, memory and disk space. ISPs should research the IPv6 capabilities of their customers before dropping the IPv4 support to make sure numerous customers do not get disconnected from the Internet.

Two of the largest companies producing enterprise class routers are Cisco and Juniper. The largest one, Cisco, had much of the IPv6 support added in 12.2 and 12.3 versions of the Cisco's routing software, IOS (originally Internetwork Operating System). Practically everything that can be needed in the enterprise networks is included in the current version 15 of IOS. [53]

The second largest router producer Juniper has routing software called JunOS. Unlike IOS, which is completely made by Cisco, JunOS is based on a FreeBSD kernel. The current JunOS versions have a considerable list of supported RFCs so the support in Juniper routers will not be a problem, although it seems that the IOS has support for more IPv6 related RFCs than JunOS. [54]

4 Status of IPv6 in TeliaSonera

This chapter reviews the situation of IPv6 in networks of TeliaSonera now and what the situation should and might be in the future. The chapter will also contain a discussion of the responsibilities of author's team in TeliaSonera: DNS, DHCP, load balancing, proxies and middleware software. The name Sonera is used to refer to the Finnish part of TeliaSonera while Telia refers to the Swedish part of the company.

4.1 Short overview of the company

TeliaSonera is the 5th largest telecommunications operator in Europe with about 172 million subscriptions. TeliaSonera has operation in Finland, Sweden, Azerbaijan, Denmark, Estonia, Georgia, Kazakhstan, Latvia, Lithuania, Moldova, Nepal, Norway, Russia, Spain, Tajikistan, Turkey and Uzbekistan. The company has currently about 27 900 employees worldwide [55] and its share is traded in the stock exchanges of Helsinki and Stockholm.

4.1.1 History of Sonera

TeliaSonera was created as a merger of Swedish and Finnish telecommunications companies Telia and Sonera in March of 2002 [56]. Both companies had a history as state owned monopolies although the predecessors of Sonera had a monopoly only in international and long distance calls. Sonera was established as a Russian bureau in 1850es. In 1885 Finnish telegraph and post services were merged into one Finnish bureau. As Finland gained its independence in 1917, the communication network that was located in Finland was handed over to Finnish Post and Telegraph bureau (Posti- ja lennätinlaitos). Between the Finnish wars a comprehensive phone network was built. The first data transfers were executed in 1964 and in 1978 a car phone network was opened. The name was changed in 1981 to Finnish Post and Telecommunications bureau.

The year 1982 was the opening year of a Nordic Mobile Telephone (NMT) network. The network was expanded in 1986 by opening a NMT 900 network. In 1992 a GSM network was opened to the customers as one of the first GSM networks in the world. In the year 1994 the bureau was reorganized into a company called "Suomen PT Oy" which had subsidiaries "Suomen Posti" (Finnish Post) and Telecom Finland. At the same time the international and the long distance calls were freed to the competition in Finland. This lead to a situation where the profitability of Telecom Finland had to be increased and about 3000 employees were dismissed.

In 1997 the Cabinet of Finland made a decision to begin a privatization of the company. In 1998 the company was renamed to Sonera and it was completely detached from the post service. In March of the same year Sonera was introduced in the Stock Exchange of Helsinki when the state sold 22,2% of the shares for 7 billion Finnish marks. State of Finland continued to lower its percentage of shares to 52,8%.

In 1999 Sonera was listed in NASDAQ and in summer of 2000 followed the 3G (UMTS, Universal Mobile Telecommunications System) license purchases in Germany and Italy. In 2002 Sonera shut down its operations in Germany and recorded huge financial losses. Because of the bad financial situation of the company after the 3G businesses in Europe, a merger with Swedish Telia seemed to be the only solution to save Sonera. The merger was officially completed in January 1st 2003. In March 21st TeliaSonera AB had bought all shares of Sonera and Sonera was renamed to TeliaSonera Finland. The merger resulted in the largest telecom operator in Nordic and Baltic countries. [56]

4.2 IPv6 status now

The network of TeliaSonera International Carrier (TSIC) has been dual-stack operational since 2009. [57] This makes it possible to interconnect more local ISPs, corporate and other end users networks. TeliaSonera has also been participating in the World IPv6 day every year.

New IPv4 allocations for Sonera, the Finnish part of TeliaSonera are almost impossible. When RIPE NCC allocates the last /8 –network LIRs will change their policies to allocate /22 –networks (1024 addresses per network). Sonera cannot get these allocations because it does not have its own IPv6 allocation as Sonera is using IPv6 allocation of whole TeliaSonera Corporation. The company has an allocation of IPv6 addresses 2001:2000::/20.

One thing that may have speeded up TeliaSonera's decision to deploy IPv6 although IPv4 addresses at least in Sonera are not totally depleted is that the company has multiple subsidiaries in Asia. Asia has higher usage of IPv6 because of the lack of IPv4 addresses (see section 2.2 Address shortage). To be able to communicate with the subsidiaries natively is easier with companywide IPv6 deployment than with some possible problems created by the translations and the tunneling protocols (section 3.2 Transition technologies).

Finland already has more mobile phone subscriptions than inhabitants and the fixed broadband connection market is saturated as well. Even if the number of fixed broadband connections is slightly increasing, this gives impression that IPv4

address shortage is not necessarily a problem in Finland. However, FICORA is expecting the mobile data usage to grow with estimations of the number of the subscriptions to double. [58] While Sonera still has free IPv4 addresses, this growth would most likely lead to IPv4 address depletion without intensive use of NAT. In case of wireless users the use of NAT may be a good idea, the NAT protects the battery of a mobile device by preventing unwanted traffic to the mobile device and mobile devices are not as likely to act as a server as hosts in fixed networks.

There is still legacy software that does not have IPv6 support at all or needs upgrading to gain the support. This software is not directly connected to the public networks and therefore it is possible to use different transition technologies on hosts running such software.

4.3 Value Added Services

This section contains a short review to areas where in which the author is working. These services are not required for a network connection to be operational but the use of these services will ease the networking.

4.3.1 DNS

The importance of the Domain Name System (DNS) will grow in IPv6 compared to IPv4. In a world using IPv4 it is still possible to memorize some addresses. Memorizing a 128-bit IPv6 address will be harder especially if the addresses are generated from the MAC addresses and are therefore more random. This is why DNS will be practically required for humans to be able to use the network services in the IPv6 networks.

The DNS servers of Sonera are capable of serving both IPv4 and IPv6 clients. DNS server software serves clients with A and AAAA records. The A records are used to translate domain names to IPv4 addresses and the AAAA records are used similarly to translate domain names to IPv6 addresses. Reverse DNS lookups, translating an IP address into a domain name, is another required function of the DNS that is operational with IPv6 as it is in the DNS servers of Sonera.

4.3.2 Address allocation for the customers

As centralized control of the customers using SLAAC would be more difficult than using DHCPv6, TeliaSonera is using DHCPv6 for IPv6 address allocation to the customers and SLAAC will not be used. An additional benefit of using DHCPv6 is that the protocol is quite similar to currently operational DHCP used

with IPv4. This gives for the DHCP system administration some experience and understanding of the system right from the beginning.

There are no major differences between DHCP and DHCPv6. Both protocols operate on top of UDP but use different ports (The used UDP ports are presented in Table 7). DHCP clients use broadcasting to find the DHCP servers before an address allocation has been made. This is impossible with DHCPv6, as IPv6 does not have a broadcasting capability. A DHCPv6 host must use multicasting to find a server. [59] [60]

Table 7 - UDP ports used by DHCP and DHCPv6

Protocol	UDP port
DHCP (client)	68
DHCP (server)	67
DHCPv6 (client)	546
DHCPv6 (server)	547

The current DHCP server software used to serve the external customers, like other companies and the consumers, is IPv6 compatible. DHCP is also used to provide 6rd configurations to the customers when IPv6 is being provided to the consumers.

4.3.3 Load balancing

TeliaSonera has found load balancers to be excellent devices to deliver traffic between the public Internet and the internal networks. As connections of users are terminated at the load balancers, IPv6 is not needed in the internal networks. On the other hand, all traffic could already be delivered with IPv6 in the internal networks.

As the load balancers operate like proxies or NATs, they are also filtering out the direct traffic to servers behind them. This increases the network server security. The difference between the TeliaSonera load balancers and traditional NATs used at home is that the load balancer changes both, the source and destination addresses. Not only the source address is changed when the packet is sent to the destination or not only the destination address is changed when the packet is coming back. As the traffic is completely terminated between the endpoints, it is also possible to do translations from IPv4 to IPv6 and vice versa. As a result of the previous architecture, the migration work can be completed flexibly and all the networks do not need to be transferred to use IPv6 at once.

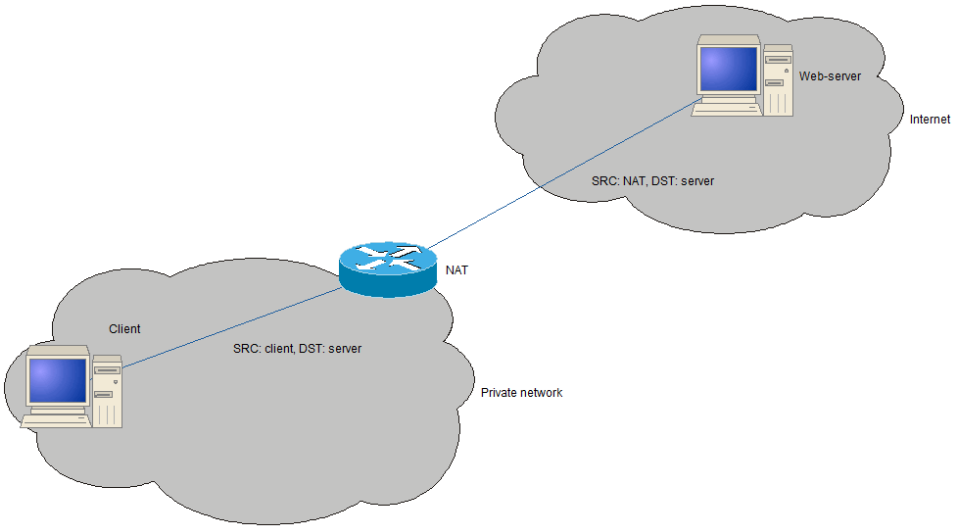

Figure 7 – NAT

A NAT device hides a local IP address by changing the source address of outgoing packets from a local address to own (public) address and by maintaining a connection state table. With the incoming packets it checks for a corresponding connection and when found, it changes the destination address to the original local source address. [61]

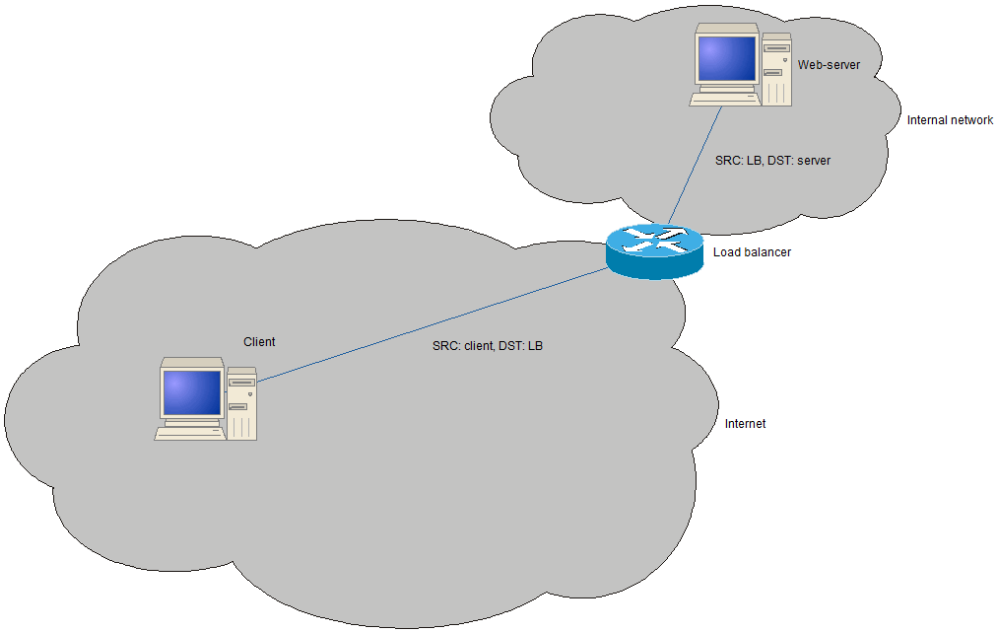

Figure 8 - Load balancer

As presented in Figure 7 and Figure 8 there is only a minor difference between traditional NAT used also in homes and the load balancing in TeliaSonera. While NAT only changes the source address of the initiator of the connection to a public and routable IP address, the load balancers in TeliaSonera operate differently. The network traffic is terminated in the load balancer that

basically proxies traffic. Both, the client and the server, see that they are talking to the load balancer, not to each other.

4.3.4 Middleware and other software

As already mentioned in section "4.2 IPv6 status now" few hosts are running legacy operating systems without an IPv6 support. Probably other software running on these servers is also not capable to handle IPv6 traffic. As these hosts are situated in private networks, load balancers or routers can be used to connect these legacy IPv4-only hosts to IPv6 networks. Security of the legacy software may be very questionable but none of these hosts are directly connected to the public networks.

The most important part of providing IPv6 services from middleware hosting servers are the load balancers and the proxies which are able to translate IPv4 traffic to IPv6 and vice versa. It should be noted that applications need to be IPv6 aware even in this case if the application data contains IP addresses. The IPv6 support of the middleware solutions like Apache Tomcat, Apache Web Server and Red Hat's JBoss used in TeliaSonera are in order.

4.4 In the Future

At the time of writing this thesis there are projects going on both in Finland and in Sweden to start providing IPv6 services to the business customers. Consumer customers are not part of these projects but they will follow afterwards.

IPv4 and IPv6 will coexist in TeliaSonera networks for years. Most legacy IT systems will never be updated to IPv6 due to diminishing use. The work needed for redesigning the software for IPv6 could be huge. In these cases it is easier and more cost efficient to use the IPv4 to IPv6 translations to provide needed connectivity for the legacy systems.

All new IT systems being deployed currently and in the future should be able to use IPv6 and preferably also IPv4. The lack of the IPv4 support is not most likely going to be a problem as practically everything still is developed for IPv4. It is still the most used IP version in the western countries by a large margin (see section "3.1 Current IPv6 usage"). As the IPv4 address shortage is already here and IPv6 is the only solution ready to be deployed, IPv6 should not be avoided.

A possibility of using F5 Integrated Architectures is under investigation in TeliaSonera. F5 Integrated Architectures would allow placing the load balancing, the DNS and the DHCP servers into one physical set of machines [62]. This

would ease the migration as there would be only one interface to migrate instead of having one interface for each of the services.

4.5 Measuring performance differences between IPv4 and IPv6

The performance differences of IPv4 and IPv6 were measured using a program called Iperf (http://sourceforge.net/projects/iperf/). One endpoint of the measurements was Ubuntu 12.04 virtual machine running on a Windows 7 host with Intel Core i5 processor and 4 GB random access memory. Another endpoint was a physical Linux server. The endpoints of the measurements were 5 network hops from each other. As I did not have access to the intermediate hops, the loads or the utilization rates of those hosts could not be checked during the measurements so this may affect the results. It should also be noted that the measurements are not telling how fast exactly both protocols are. The measured values may vary greatly depending on the environment. The purpose of the measurements was to see the relative performance differences, which of course may be different in other environments.

Iperf used for the measurements does not support raw IP packets but only TCP and UDP. The measurements were made using TCP. It was selected as the protocol itself has to make sure not to flood connection between the hosts. In addition TCP uses lots of small packets to acknowledge the transferred packets (the reliability of the connection) so the generic traffic will be more complex and demands handling of very different sized packets compared to UDP which may send full MTU sized packets all the time. [63] [64]

Delay and packet delay variation (jitter) were measured with standard Linux ping program. With the ping measurements it is important to note that first round trip may take more time as ARP or ND is used to find out to which MAC address the given IP address belongs to.

4.5.1 Theoretical values

Usually the Maximum Transmission Unit (MTU) used with Internet Protocols is 1500 bytes which is a limit set by Ethernet protocol [65]. Part of these 1500 bytes is taken for the headers of the upper layers meaning that all 1500 bytes may not be actual payload containing the user data. The minimum sizes of the IPv4 and IPv6 headers are 20 bytes and 40 bytes respectively. In addition to IP headers, also TCP, UDP, ICMP or some other upper layer header is going to take some bytes away from the payload that user sees.

Assuming that network hosts are able to handle a certain number of maximum sized Ethernet packets, it is possible to calculate a theoretical throughput difference between the Internet Protocols:

$$\frac{\textbf{IPv6 payload}}{\textbf{IPv4 payload}} = \frac{\textbf{1500 B} - \textbf{40 B}}{\textbf{1500 B} - \textbf{20 B}} = \frac{\textbf{1460}}{\textbf{1480}} \approx \textbf{0.986} \quad (5)$$

In other words IPv6 can transfer 98% of the traffic that IPv4 can because of larger header. In theory, however, the delays should not be affected: considerably longer address may take more time to be read or written and renamed TTL field still exists in the IPv6 header. The difference is that IPv6 header does not contain the header checksum that needs to be recalculated on each network hop. The packet delay variation should not be affected.

Different encapsulation techniques add more overhead by including an extra layer of headers compared to the normal IP and TCP or UDP headers. For example 6in4 protocol has headers of IPv4 and IPv6 in a single packet:

$$\frac{\textbf{6in4 payload}}{\textbf{IPv4 payload}} = \frac{\textbf{1500 B} - \textbf{40 B} - \textbf{20 B}}{\textbf{1500 B} - \textbf{20 B}} = \frac{\textbf{1440}}{\textbf{1480}} \approx \textbf{0.973} \quad (6)$$

As can be seen in Equation 6, adding another IP header does not considerably reduce the size of the payload. There may be problems if a badly written application needs to send more information in a single packet. The performance is not an issue.

Different tunneling or other transition mechanisms will increase delays especially in the connection establishment as there will be more signalling before the actual connection for user data can be opened. The values differ greatly between the transition mechanisms depending on amount of signalling.

4.5.2 Measurements in Practice

The test environment was the one described in the beginning of section 4.5. Round Trip Time (RTT) and delay variation were measured using the Ping program. Ping sends an ICMP echo request and the remote host responds to it with an ICMP echo reply. Time between sending the echo and receiving it is measured to get RTT. RTT was measured in groups on 20 echo request and replies. IPv4 and IPv6 Ping were tested in turns, both for ten times. All together there were 200 requests sent with both Internet Protocols. No packet loss or other network faults were noticed during the measurement.

After each set of 10 echo request and reply pairs Ping program was terminated and it delivered a summary of the test set. The summary includes

following information: minimum RTT, maximum RTT, average RTT and standard deviation. The standard deviation is calculated [66]:

$$\mathbf{mdev} = \sqrt{\sum \frac{\mathbf{RTT^2}}{\mathbf{N}} - \left(\sum \frac{\mathbf{RTT}}{\mathbf{N}}\right)^2} \qquad (7)$$

Where *mdev* is the standard deviation, *RTT* is round trip time and *N* is number of the echo requests sent.

Table 8 – Practical delay and delay variation measurements (in milliseconds)

Protocol	Average RTT	Minimum RTT	Maximum RTT	Standard deviation
IPv4	3.272	3.134	3.389	0.174
IPv6	3.727	3.478	3.958	0.187

Table 8 contains the measurements explained above. All the values in the table are averages of the values measured in milliseconds by Ping in the individual test sets. It can be seen that the RTT with IPv6 is higher in all the test cases as the maximum of IPv4 is slightly lower than the minimum delay of IPv6. The most likely reason for the observed behavior is that longer addresses of IPv6 may take a little longer to compute. Even though the difference should be minimal, it is multiplied with the number of hops along the route. Another possible reason is that the IPv6 stack, the program or part of the operating system that handles the IPv6 packets, is not yet as highly optimized as the IPv4 stack that has been under development for decades. The measured differences in round trip times can be seen in Equation 8: IPv6 has roughly 1.13 times higher RTT than IPv4.

$$\frac{\boldsymbol{IPv6\ RTT}}{\boldsymbol{IPv4\ RTT}} = \frac{\mathbf{3.727}}{\mathbf{3.272}} \approx \mathbf{1.1391} \qquad (8)$$

Distribution of individual measurements can be seen in Figure 9. Even though the average values of measurements with IPv6 seem to vary significantly more than with IPv4, standard deviations within a single measurement set do not have such a huge variance as can be seen in Table 8. The cause of this observed phenomenon is unknown but it could be related to the reasoning above about the CPU resource usage combined with small variations in the network usage levels.

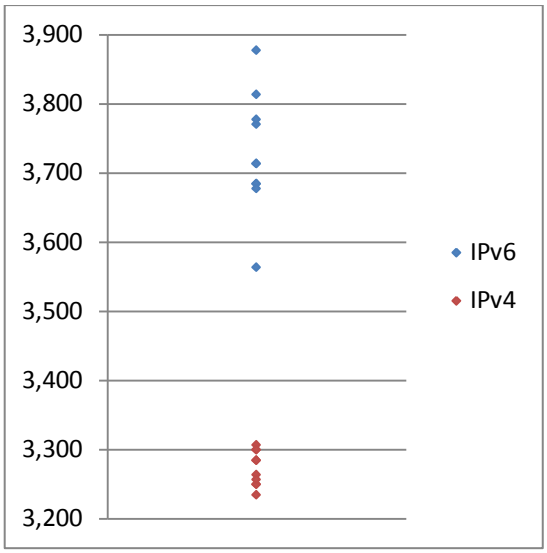

Figure 9 – Individual delay measurements. Time in milliseconds

The throughput was measured with Iperf using TCP as explained in section "4.5 Measuring performance differences between IPv4 and IPv6". The throughput tests were performed by measuring the amount of data transferred in 30 seconds. The test was done 20 times for both protocols and the average of individual tests was calculated. The results are shown in Table 9. It can be seen that IPv6 was capable to transfer only about 89.5% of the data that IPv4 is able to carry as calculated in Equation 9.

$$\frac{\textbf{IPv6 throughput}}{\textbf{IPv4 throughput}} = \frac{\textbf{16.38MBps}}{\textbf{18.30MBps}} = \textbf{0.895} \qquad (9)$$

The difference between the measured and the theoretically calculated value can be explained similarly to differences in calculated and measured delays: non-optimized protocol stacks. Even one weak link (or a hop) on a route makes the performance lower. As TCP was used, the delay also affects by hindering acknowledging which leads to less optimal use of the TCP send window. UDP would not have the same problem but without acknowledging the reliability of the transfer is not guaranteed. Unreliable transfer is in some cases completely useless. The throughput differences cannot be explained only by delayed acknowledging as the relative difference in throughput is considerably greater than in the delay measurements.

Table 9 – Throughput measurements in Megabytes per second

Protocol	Average throughput	Minimum throughput	Maximum throughput
IPv4	18.30	17.83	18.67
IPv6	16.38	16.03	16.57

Network congestion was not likely playing any part in the measurements as single tests were made in turns. In addition to this the measured values did not vary greatly between different measurements of the same protocol. This can be seen in Figure 10.

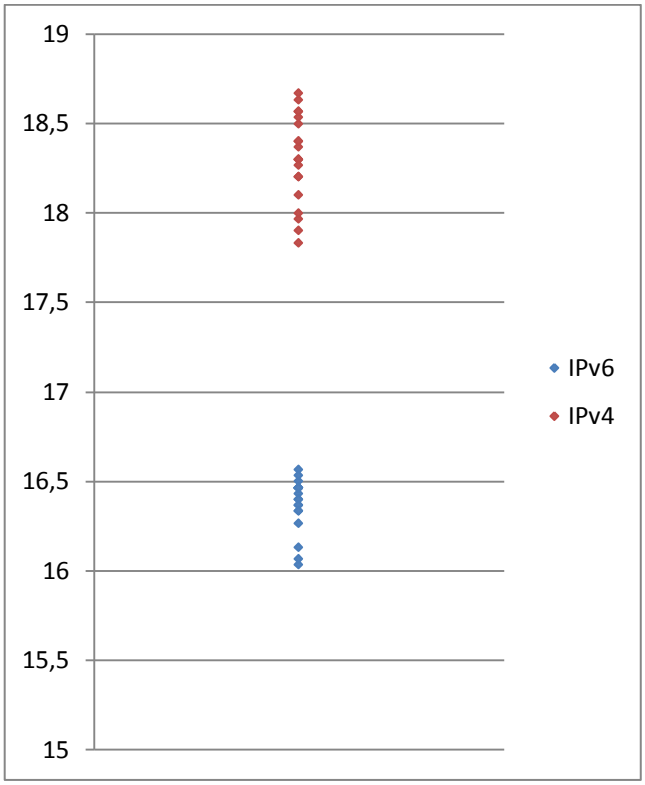

Figure 10 – Measured throughput values in Megabytes per second

The variation of the individual throughput measurements between the protocols was, unlike the delay variation, quite similar and could be addressed to measurement inaccuracies and small differences in network usage levels.

4.5.3 6rd tunneling method

Tunneling method to be measured was 6rd as it is the chosen mechanism for TeliaSonera. The test setup was the same as in previous tests. Native IPv6 connectivity was broken by unbinding the assigned IPv6 address from the network interface in the client machine.

The results of the measurements are shown in Table 10. The results of native IPv6 are copied from the previous measurements. 6rd performance was quite similar to IPv6 although the expected performance hits were seen. Additional headers affect the throughput a little as well as increase delay.

Table 10 – 6rd performance (in milliseconds and MBps)

Protocol	Average delay	Average throughput
Native IPv6	3.727	16.38
6rd	3.814	14.98

4.6 Teredo functionality investigation

Teredo is a migration mechanism that is able to operate even if the user is behind a NAT gateway. This section discusses about the ability in practice and about the performance of Teredo.

The tests about NAT and Teredo were performed by using an open source Teredo implementation called Miredo. The test environment consisted of virtual Linux machines running a Teredo client, server, relay and a NAT gateway. Desired network connectivity between the virtual hosts was created by manual routing. All traffic between the client and the server in the virtual IPv6 Internet was routed through the NAT gateway. The different NAT types were created by using iptables firewall rules. The used iptables rules are listed in Table 11. The used rules do not actually create usable NATs as static IP addresses are used in the rules. However, the functionality with one host is similar in regards to Teredo.

Three types of NAT were tested. The first tested NAT type was a one-to-one NAT. This type assigns one public IP to a host in the private network. Once the NAT mapping is made, anyone from the public network is able to contact the host in the private network. Teredo was able to operate with this type of NAT.

The second type of tested NAT was an address mapped NAT. The address mapped NAT adds a mapping for the packets to a specific destination in the public network. Traffic between the destination and the source can use any port but no new connections from other hosts in the public network to the private network are accepted. Teredo also operated with this type of NAT.

The third tested NAT type was an address and port mapped NAT. This NAT type is the most popular one according to the experience of the author as it allows the most significant savings for the used public IPv4 addresses and is the most secure by being the most restrictive. The NAT mapping is valid only for specific address and port pairs in the hosts. Connections from the public network to the private network are not possible without manual configuration. Unfortunately Teredo connections with this NAT type failed. According to traffic captures with Wireshark the reason seemed to be that the Teredo server was not able to guess or predict the ports that NAT gateway was using. Static port forward mappings

circumvented the problem but this wrecks the idea of automated IPv6 tunneling. Another possibility to fix the problem would be to place the server and relay in one host. However, this option would still require the server and relay to communicate with each other. As this communication was not supported by Miredo, the possible solutions to the problem could not be fully tested.

Table 11 – The iptables rules used to create different NAT types

NAT type	**Iptables rule**	**Teredo able to operate**
One-to-one NAT	○ iptables -t nat -A POSTROUTING -o eth0 -j SNAT --to-source <public IP> ○ iptables -t nat -A PREROUTING -i eth0 -j DNAT --to-destination <private IP>	Yes
Address mapped NAT	○ iptables -t nat POSTROUTING -o eth1 -p tcp -j SNAT --to-source <public IP> ○ iptables -t nat POSTROUTING -o eth1 -p udp -j SNAT --to-source <public IP> ○ iptables -t nat PREROUTING -i eth1 -p tcp -j DNAT --to-destination <private IP> ○ iptables -t nat PREROUTING -i eth1 -p udp -j DNAT --to-destination <private IP> ○ iptables -A INPUT -i eth1 -p tcp -m state --state ESTABLISHED,RELATED -j ACCEPT ○ iptables -A INPUT -i eth1 -p udp -m state --state ESTABLISHED,RELATED -j ACCEPT ○ iptables -A INPUT -i eth1 -p tcp -m state --state NEW -j DROP ○ iptables -A INPUT -i eth1 -p udp -m state --state NEW -j DROP	Yes
Address and port mapped NAT	○ echo "1" > /proc/sys/net/ipv4/ip_forward ○ iptables --flush ○ iptables -t nat -A POSTROUTING -o eth1 -j MASQUERADE --random ○ iptables -A FORWARD -i eth1 -o eth0 -m state --state RELATED,ESTABLISHED -j ACCEPT ○ iptables -A FORWARD -i eth0 -o eth1 -j ACCEPT	No, not without port forwarding

The performance of Teredo was measured in the same network. The purpose was to investigate the delay in connection establishment. The delay was measured by fetching a nonexistent file with HTTP. The delay was compared to native IPv6 connection. The native connection was created by removing the NAT and by routing the IPv6 traffic through the former NAT gateway. The time for getting the HTTP 404 error ("Not Found") with native IPv6 connection was on average 0.05 seconds. The variation of time was minimal (± 0.01 seconds). The results were presented with a precision of 0.01 seconds. The time used for the same task with Teredo was on average 1.14 seconds. The longest measured time with Teredo was 1.98 seconds. No reason for large deviations was found with network traffic captures.

The long delay when opening the connection is not an issue when transferring large files, for example video. However, the delays may be irritating when loading web pages as all the connections require their own Teredo tunnels. All different files are fetched with different HTTP GET commands meaning multiple TCP connections for a web page that includes multiple pictures. Also, the test network had Teredo servers and relays very close to the client making the situation ideal for Teredo.

4.7 Migrating an existing network

A migration project demands careful planning. Multiple devices and applications will make understanding all the cause-and-effect relationships difficult. This is why planning and testing is needed before a new IPv6 network can be transferred into production use. Another important matter to secure is communication. Communication is needed between possible different teams working on the migration as well as with the users of the IT systems to be migrated.

The first step in migration process should be a pre-study about the current situation of IT systems that are planned to be migrated. The study should include information about the status of infrastructure and applications. In this case the infrastructure means the network equipment, mainly the routers as switches operate on layer 2, and the operating systems of the workstations and the servers. Applications are the actual software that is used to achieve the wanted results of the IT-system. The information should include at least current IPv6 usage and versions of software. With this information it is possible to check the current status of the IPv6 support and possibilities to upgrade the software to support IPv6. Also configurations should be checked on some level: do the interfaces of the computers and the routers have IPv6 addresses, subnets, gateways and other needed information configured, and do the routers have IPv6 routing enabled. At the same time it should be checked that IPv6 connections to other networks are

available if needed, usually meaning that IPv6 support of these networks needs to be checked.

The rest of the process depends highly on the results of the pre-study phase. In the case of legacy IT systems, the migration will become complicated: no easy possibility for upgrade. If IPv6 is needed, transition mechanisms are required. In case that completely new IT systems to replace the legacy systems are possible, native IPv6 support can be achieved more easily. Even if the migration currently seems costly and difficult, the migration should be planned as the migration will not be any easier in the future, after all some kind of IPv6 support will be required to allow global connectivity.

With a possibility to use at least mostly native IPv6, or in fact IPv4 and IPv6 dual stacks at the beginning to guarantee the connectivity to anywhere, the migration process becomes a bit easier. In this case it is only needed to ensure connectivity for each of the protocols, not also the interoperability of the protocols with the selected transition mechanism.

The network topology can remain the same as putting it very simply: IPv6 is the same as IPv4 with considerably longer addresses. If a topology change for IPv4 network is needed, it may be wise to perform it already before the IPv6 migration. This way the number of variables with possible problems is smaller and the diagnostics to find the cause of the problems is easier.

The actual migration should be started by updating the router software and hardware as needed and by configuring the interfaces and the appropriate routing protocols. As a IPv4 to IPv6 migration could be categorized as a network change starting with the routers seems logical. Even though the actual applications using the network may not yet be operational, the operations of the network can be ensured. This way the whole new IPv6 can be built bottom-up avoiding unnecessary work by checking that the foundations are working as designed.

The next step of the migration should be to start verifying the IPv6 support in the end hosts, in the workstations and in the servers. To continue bottom-up building, the first thing to do should be the possible operating systems upgrades. At this point special care should be taken to ensure that no application data is lost during the possible installations of newer operating system versions. After the operating system installations it is time for configuration of the operating systems. This is not needed if DHCPv6 or SLAAC is used to provide autoconfiguration. At this stage it would be good to check also the IPv6 support of used DNS system: AAAA records and the reverse records.

After successful installations and configurations, thorough testing of the current network should be conducted. With the network devices and the operating systems in the end hosts ready, the functionality, performance and reliability of the IPv6 infrastructure can be tested. After the test results are accepted required software may be installed or updated in the end hosts.

Most of the modern software seems to support IPv6 (see section 3.4 Current IPv6 support). However, custom made or highly customized software may be more difficult to update. This should be noted early enough in the process to allow enough time for changes into networking code of such software. After the installations and the configurations for the applications are made, the final testing should follow before the production use of the applications.

Most likely problems during the migration process are economic problems and problems related to demands that the IPv4 network is needed all the time during the process. Economic problems include things that are a part of practically all the projects: too small human resources, too tight deadlines and the lack of money, for example, for new hardware or software licenses.

The need for continuing the usage of IPv4 network during the migration may hinder IPv6 installations. If breaks for rebooting hardware or restarting software are not allowed anytime, the process should be started early enough to be able to have enough service windows for the reboots. Using the dual stack makes the process a little easier because change from IPv4 to IPv6 does not have to happen at once. This gives more time for the migration work as the IPv6 deployment can be done in parts while using IPv4 at the same time.

The best scenario would be a possibility of building a new parallel network. In this way the IPv4 network in production use and the IPv6 network under construction would not hinder each other at all. In the end of the process the data in both networks could be synchronized and the shutdown of the IPv4 network could be started. A disadvantage of this approach is the cost.

5 Conclusions

When considering the technical readiness of IPv6, it is ready for a worldwide deployment. Currently the biggest problem seems to be the lack of network effect. Without a large user base others are not willing to start the deployment. The imminent lack of IPv4 addresses will most likely speed up the deployment in the near future.

There are not many significant differences between IPv4 and IPv6. IPv6 has considerably longer addresses and at the same time the header format has been simplified. IPv4 had IPsec as optional expansion. In IPv6 it is a required part. Practically the situation has not changed as the use of IPsec is not mandatory. The larger address space makes it harder for the attackers and the owners to find vulnerable hosts. Otherwise security aspects remain quite the same: some names have changed but basically nothing more. The longer addresses are almost impossible for a human to remember, so DNS is needed.

The deployment in TeliaSonera has been started and IPv6 services for the corporate customers should be available at the time this thesis is published. In situation of Sonera the load balancers ease a lot the migration work because the internal and external networks do not need to be migrated simultaneously. TeliaSonera's international connections are already using IPv6 so global connectivity can be achieved quickly.

In the future IPv6 deployment will be continued by providing connectivity also for the consumers. The schedule is not yet decided. IPv4 will coexist with IPv6 for years before all systems have been migrated to use IPv6. The migration process needs to be planned carefully. There are many factors that generate the work extremely challenging as the systems cannot be shut down for long periods of time to allow easier updates.

5.1 Measurement conclusions

As a conclusion on measurements (section "4.5 Measuring performance differences between IPv4 and IPv6") it would be a good idea to test IPv6 network performance as part of the migration process before putting the network into production us. As seen, there may be large to moderate deviations in performance. In most cases the measured differences would not be a problem, but in IT systems requiring very high network performance, for example, the core network of an ISP, the measured differences could cause trouble. In case of TeliaSonera this means mostly TSIC networks. As noted before TSIC is already using IPv6 so this is not an issue in TeliaSonera.

The measured transition mechanism, 6rd, did not add significant performance degradation over IPv6. As all the measurements were done in a fast network over 5 hops, the differences between IPv4 and IPv6 performance for a single client or a server host would be relatively insignificant. Use of IPv6 or transition mechanisms should not be a problem regarding the performance. The only problems may arise if lots of signaling is needed in the connection establishment. For example Teredo (section 3.2.4) multiplies the delay in connection establishment by several round trip times which may be inconvenient when transferring small files. In case of 6to4 it may also be necessary to measure the delays as the out-of-own-control 6to4 relays could be far away in the network topology. With 6rd the relay is handled by the ISP so the number and length of the hops should stay limited also lowering the delay.

If the transfer speeds would have been more limited, for example because of software limitations or by a narrow link between hops, the results could have been more even. With smaller loads non-optimized protocol stack would not have been such a problem because of a smaller number of packets per second that needed to be handled.

5.2 Why not to use some other protocol?

Some of the other possibilities to be used instead of IPv6 are reviewed in section "2.3 Other possibilities". While some of the ideas are basically just more intensive use of NATs, some ideas are more radically changing the current Internet.

More extensive use of NAT would allow almost infinite number of hosts to be connected to the network. However, if multiple NAT devices were used in line the end customer loses the control of the address translation process. As already noted, applications requiring server capabilities will be very difficult to use and operate in an environment with multiple NATs. Lots of users could be unsatisfied as these applications requiring server functions include some very popular ones, like P2P software, multiple communications and messaging applications as well as for example personal WWW servers.

With mobile phones the use of NAT would not be such a severe issue as at least currently P2P and VoIP usage with the mobile phones is low compared to the desktop and the laptop computers. However, at least in Finland the number of users using 3G or 4G data connections as their only Internet connection is increasing. A user using mobile broadband with a laptop requires the same services as a user with laptop and wired connection so using carrier grade NATs for the mobile users is not a solution.

RE2EE seems to have multiple good design decisions compared to the more traditional NAT solutions. RE2EE does not break completely the end-to-end principle of the Internet and does not add any delays during the connection establishment. The network changes do not need to be global.

The problem in deploying RE2EE will be that the protocol specifications are not yet finished. Without a complete solution there will most likely be no implementations of the protocol soon. IPv6 is already being slowly deployed, so the protocol implementations should already be available to compete with IPv6 in my opinion. The possible deployment would be much faster than with IPv6 as the lack of IPv4 addresses is imminent.

5.3 Future research

As the topic seems to be currently under quick development similar studies as this one could be done on regular basis. Exhausted IPv4 address space demands something to be done and progress should be fast so checking the situation again, for example in one year interval, seems to be a good idea.

Another topic could be real life performance comparison of IPv4 and IPv6 instead of this small laboratory measurement done as part of this thesis. This research could also include practical testing how different transition mechanisms effect on overall performance.

References

[1] Information Sciences Institute, University of Southern California. 1981. Internet Protocol. RFC 791.

[2] S. Deering and R. Hinden. 1995. Internet Protocol, Version 6 (IPv6). RFC 1883.

[3] IANA. IANA IPv4 Address Space Registry. Cited: February 21, 2012. http://www.iana.org/assignments/ipv4-address-space/ipv4-address-space.xml.

[4] M. Leber. Global IPv6 Deployment Progress Report. Cited: April 03, 2012. http://bgp.he.net/ipv6-progress-report.cgi.

[5] F. Baker, X. Li, C. Bao and K. Yin. 2011. Framework for IPv4/IPv6 Translation. RFC 6144.

[6] V. Cerf, et al. 1999. A Brief History of the Internet. Cited: February 10, 2012. http://arxiv.org/html/cs.NI/9901011.

[7] Y. Rekhter and T. Li. 1993. An Architecture for IP Address Allocation with CIDR. RFC 1518.

[8] R. Hinden and S. Deering. 2006. IP Version 6 Addressing Architecture. RFC 4291.

[9] IANA. Internet Protocol Version 6 Address Space. 2010. Cited: February 24, 2012. http://www.iana.org/assignments/ipv6-address-space/ipv6-address-space.xml.

[10] R. Hinden and B. Haberman. 2005. Unique Local IPv6 Unicast Addresses. RFC 4193.

[11] S. Deering and R. Hinden. 1998. Internet Protocol, Version 6 (IPv6). RFC 2460.

[12] T. Narten, E. Nordmark, W. Simpson and H. Soliman. 2007. Neighbor Discovery for IP version 6 (IPv6). RFC 4861.

[13] S. Sotillo. 2006. IPv6 Security Issues. Cited: February 28, 2012. http://www.infosecwriters.com/text_resources/pdf/IPv6_SSotillo.pdf.

[14] S. Thomson, T. Narten and T. Jinmei. 2007. IPv6 Stateless Address Autoconfiguration. RFC 4862.

[15] T. Narten, R. Draves and S. Krishnan. 2007. Privacy Extensions for Stateless Address Autoconfiguration in IPv6. RFC 4941.

[16] C. Perkins, D. Johnson and J. Arkko. 2011. Mobility Support in IPv6. RFC 6275.

[17] S. Kent and R. Atkinson. 1998. Security Architecture for the Internet Protocol. RFC 2401.

[18] V. Cerf. Google IPv6 Conference 2008: What will the IPv6 Internet look like? Youtube - Broadcast Yourself. http://www.youtube.com/watch?v=mZo69JQoLb8.

[19] G. Huston - potaroo.net. 2012. IPv4 Address Report. Cited: March 06, 2012. http://www.potaroo.net/tools/ipv4/index.html.

[20] O. Maennel, R. Bush, L. Cittadini and S. Bellovin. 2008. A Better Approach than Carrier-Grade-NAT. https://mice.cs.columbia.edu:443/getTechreport.php?techreportID=560.

[21] J. Postel. 1981. Internet Control Message Protocol. RFC 792.

[22] J. Ryynänen. 2008. Routed End-to-End Ethernet Network - Proof of Concept. http://lib.tkk.fi/Dipl/2008/urn012212.pdf.

[23] R. Kantola, J. Santos, N. Beijar and P. Leppäaho. 2012. Implementing NAT Traversal with Private Realm Gateway.

[24] R. Kantola, J. Santos, N. Beijar and P. Leppäaho. 2012. Traversal of the Customer Edge with NAT-Unfriendly Protocols.

[25] OECD. 2010. Internet Addressing: Measuring Deployment of IPv6.

[26] E. Zachte. 2012. Wikimedia Traffic Analysis Report - Operating Systems. Cited: April 04, 2012. http://stats.wikimedia.org/archive/squid_reports/2011-12/SquidReportOperatingSystems.htm.

[27] FICORA. 2012. Kysely IPv6-protokollan tilanteesta Suomessa 2012. http://www.ficora.fi/attachments/68DG30Ll9/IPv6-kysely_2012_-_yhteenveto_2.pdf.

[28] E. Nordmark and R. Gilligan. 2005. Basic Transition Mechanisms for IPv6 Hosts and Routers. RFC 4213.

[29] M. Blanchet. 2010. IPv6 Tunnel Broker with the Tunnel Setup Protocol (TSP). RFC 5572.

[30] A. Conta and S. Deering. 1998. Generic Packet Tunneling in IPv6 Specification. RFC 2473.

[31] B. Carpenter and K. Moore. 2001. Connection of IPv6 Domains via IPv4 Clouds. RFC 3056.

[32] C. Huitema. 2001. An Anycast Prefix for 6to4 Relay Routers. RFC 3068.

[33] W. Townsley and O. Troan. 2010. IPv6 Rapid Deployment on IPv4 Infrastructures (6rd). RFC 5969.

[34] C. Huitema. 2006. Teredo: Tunneling IPv6 over UDP through Network Address Translations (NATs). RFC 4380.

[35] M. Bagnulo, P. Matthews and I. van Beijnum. 2011. Stateful NAT64: Network Address and Protocol Translation from IPv6 Clients to IPv4 Servers. RFC 6146.

[36] M. Bagnulo, A. Sullivan, P. Matthews and I. van Beijnum. 2011. DNS64: DNS Extensions for Network Address Translation from IPv6 Clients to IPv4 Servers. RFC 6147.

[37] S. Szigeti and P. Risztics. Will IPv6 Bring Better Becurity? http://mycite.omikk.bme.hu/doc/37374.pdf.

[38] C. Eckstein and A. Atlasis. 2011. OS fingerprinting with IPv6. http://www.sans.org/reading_room/whitepapers/testing/os-fingerprinting-ipv6_33794.

[39] A. Pilihanto. 2011. A Complete Guide on IPv6 Attack and Defense. http://www.sans.org/reading_room/whitepapers/detection/complete-guide-ipv6-attack-defense_33904.

[40] Van Hauser. 2008. Attacking the IPv6 Protocol Suite. Cited: May 22, 2012. http://www.thc.org/papers/vh_thc-ipv6_attack.pdf.

[41] S. Convery and D. Miller. 2004. IPv6 and IPv4 Threat Comparison and Best-Practice Evaluation (v1.0). Cited: May 30, 2012. http://seanconvery.com/v6-v4-threats.pdf.

[42] B. Claise. 2004. Security Considerations for 6to4. RFC 3954.

[43] J. Arkko, J. Kempf, B. Zill and P. Nikander. 2005. SEcure Neighbor Discovery (SEND). RFC 3971.

[44] T. Ptacek and T. Newsham. 1998. Insertion, Evasion, and Denial of Service: Eluding Network Intrusion Detection. http://www.dtic.mil/cgi-bin/GetTRDoc?Location=U2&doc=GetTRDoc.pdf&AD=ADA391565.

[45] M-K. Shin, Y-G. Hong, J. Hagino, P. Savola and E. M. Castro. 2005. Application Aspects of IPv6 Transition. RFC 4038.

[46] P. Weissmann. 2012. IPv6 Operating Systems. Cited: February 14, 2012. http://ipv6int.net/systems/index.html.

[47] University of Wisconsin. 2011. IPv6 support on common operating systems. Cited: February 15, 2012. https://kb.wisc.edu/ns/page.php?id=13736.

[48] P. Bieringer, F. Baraldi, S. Piunno, M. Tortonesi, E. Toselli and D. Tumiati. 2011. Current Status of IPv6 Support for Networking Applications. Cited: February 16, 2012. http://www.deepspace6.net/docs/ipv6_status_page_apps.html.

[49] C. Hopps. 2008. Routing IPv6 with IS-IS. RFC 5308.

[50] G. Malkin and R. Minnear. 1997. RIPng for IPv6. RFC 2080.

[51] R. Coltun, D. Ferguson and J. Moy. 1999. OSPF for IPv6. RFC 2740.

[52] T. Bates, R. Chandra, D. Katz and Y. Rekhter. 2007. Multiprotocol Extensions for BGP-4. RFC 4760.

[53] Cisco. 2011. Start Here: Cisco IOS Software Release Specifics for IPv6 Features. Cited: February 22, 2012. http://www.cisco.com/en/US/docs/ios/ipv6/configuration/guide/ip6-roadmap.pdf.

[54] Juniper Networks. 2011. Supported IPv6 Standards. Juniper Networks. Cited: February 22, 2012. http://www.juniper.net/techpubs/en_US/junos11.3/topics/reference/standards/ipv6.html.

[55] TeliaSonera. 2012. TeliaSonera in brief. Cited: June 28, 2012. http://www.teliasonera.com/en/about-us/teliasonera-in-brief/.

[56] TeliaSonera. 2012. TeliaSonera History. Cited: June 29, 2012. http://www.teliasonerahistory.com/.

[57] TeliaSonera International Carrier. IP. Cited: January 13, 2012. http://www.teliasoneraic.com/Ourservices/IP/index.htm.

[58] FICORA. 2012. Two data transfer subscriptions per each Finn in coming years. http://www.ficora.fi/en/index/viestintavirasto/uutiset/2012/P_28.html.

[59] R. Droms. 1997. Dynamic Host Configuration Protocol. RFC 2131.

[60] R. Droms, J. Bound, B. Volz, T. Lemon, C. Perkins and M. Carney. 2003. Dynamic Host Configuration Protocol for IPv6 (DHCPv6). RFC 3315.

[61] P. Srisuresh and M. Holdrege. 1999. IP Network Address Translator (NAT) Terminology and Considerations. RFC 2663.

[62] N. Meyer and C. Liu. 2010. F5 and Infoblox DNS Integrated Architecture: Offering a Complete Scalable, Secure DNS Solution. Cited: September 13, 2012. http://www.f5.com/pdf/white-papers/infoblox-wp.pdf.

[63] Information Sciences Institute, University of Southern California. 1981. Transmission Control Protocol. RFC 793.

[64] J. Postel. 1980. User Datagram Protocol. RFC 768.

[65] X. Zhou, Xiaoming, M. Jacobsson, H. Uijterwaal and P. van Mieghem. 2007. IPv6 delay and loss performance evolution.

[66] Y. Hideaki. 2010. iputils - Source code. iputils. http://www.skbuff.net/iputils/iputils-s20101006.tar.bz2.