NTNU – Trondheim
Norwegian University of
Science and Technology

IPv6 only national backbone

Ignacio Rey Gallo-Alcántara

Submission date: June 2020
Supervisor: Otto Wittner, IIK NTNU
Co-supervisor: Jørn de Jong, Uninett

NTNU – Norwegian University of Science and Technology
Department of Information Security and Communication Technology

Title: IPv6 only national backbone

Student: Ignacio Rey Gallo-Alcántara

Problem description:

With the unexpected expansion of the Internet and the upcoming IoT, with which all kind of devices will require an Internet connection, IPv4 has proven incapable of being able to provide a unique IP address to every Internet connected device in the world. To solve the issue IPv6 was developed 25 years ago, however, and because the actual size of the Internet, the transition from one technology to the other is taking more time than expected. Due to the incompatibility between IPv4 and IPv6, it has been necessary to develop a number of mechanisms, in order to provide a (sometimes limited) interoperability between the two protocols and to extend the use of IPv4 until IPv6 surpass and substitutes the previous protocol.

Uninett is the national research IP network operator in Norway and is responsible of providing access to the global internet as well as access to a range of online services to universities, university colleges and research institutions within the country. It is important to any Internet Service Provider (ISP), such as Uninett, to be aware of the global state of IPv6 and to try to update its network to it a soon as they can and other networks and services allow it, since maintaining IPv4 and IPv6 in its network is both expensive and limiting.

The main objectives of the present project are to survey the existing transition technologies between IPv4 and IPv6; to research on the actual state of Uninett's network with this regard, including the volume of IPv4 and IPv6 traffic and available transition technologies; and to suggest a novel scenario for Uninett (which could be extended to other National Research and Education Networks (NRENs)) that could be evaluated in the near future with the objective of relying only on its IPv6 core network, with all the security, usability and economic advantages that this scenario could bring to the stakeholder.

Responsible professor: Otto Wittner, IIK NTNU

Supervisor: Jørn de Jong, Uninett

Abstract

The number of free IPv4 addresses is increasingly reduced, while the number of devices that require an Internet connection has not stopped growing in recent years, and it is expected to grow faster with the upcoming Internet of Things (IoT). While the adoption of IPv6 has maintained a slow but continued growth over the years, a number of technologies have been designed to enable the interoperability of both protocols. Due to the IPv4 address shortage, it is becoming increasingly urgent for Internet Service Providers (ISPs) to reduce the presence of IPv4 on their networks and make a transition to IPv6.

The main goals of the thesis are to study the degree of adoption of IPv6 both at present and for the case of the Norwegian ISP Uninett, as well as the different technologies for transitioning between IPv4 and IPv6 that exist; with the final goal of presenting a future scenario in which Uninett can reduce the presence of IPv4 on its networks and the number of required IPv4 addresses, moving to an IPv6-only core network.

Sammendrag

Antallet tilgjengelige IPv4 adresser reduseres mer og mer, mens antall enheter som krever en internett-tilkobling har økt de siste årene, og det er forventet å øke videre grunnet Internet of Things (IoT). Bruken av IPv6 har økt jevnt, men langsomt de siste årene, og flere teknologier har blitt designet for å sikre interoperabilitet mellom begge protokollene. På grunn av mangelen på IPv4-adresser har det blitt enda viktigere for Internettleverandører å redusere bruken av IPv4 i nettverkene, og ta i bruk IPv6.

Hovedmålene med denne oppgaven er å studere i hvor stor grad IPv6 er i bruk generelt og hos den norske Internettleverandøren Uninett, samt hvilke teknologier som finnes for overgangen mellom IPv4 og IPv6; med det endelige målet om å presentere et fremtidig scenario hvor Uninett kan redusere bruken av IPv4 i deres nettverk og antallet nødvendige IPv4-adresser, for å kunne oppnå et kjernenettverk basert på kun IPv6.

Preface

This thesis serves as the final point of my Master's studies, carried out at the Norwegian University of Science and Technology (NTNU) in Trondheim, Norway.

With this paragraph I would like to say thanks, first of all, to the NTNU, for accepting me into their ranks. To my teacher, Otto Wittner, for all the help and attention he gave me, not only during the project months, but also during my Master's studies. To Jørn de Jong of Uninett, for his vast knowledge of the subject. To my family and friends, for always being a point of support for me, I do not deserve them.

Contents

List of Figures

List of Tables

List of Acronyms

4rd IPv4 Residual Deployment.

6rd IPv6 rapid development.

AFTR Address Family Transition Router.

AP Access Point.

APNIC Asia-Pacific Network Information Centre.

AYIYA Anything In Anything.

B4 Basic Bridging BroadBand.

BGP Border Gateway Protocol.

BR Border Relay.

CLAT Customer-side Translator.

CPE Customer Premises Equipment.

DCCP Datagram Congestion Control Protocol.

DHCP Dynamic Host Configuration Protocol.

DNS Domain Name System.

DNS-ALG Domain Name System Application Level Gateway.

DS-Lite DS-Lite.

DSTM Dual Stack Transition Mechanism.

GRE Generic Routing Encapsulation.

HTTP Hypertext Transfer Protocol.

IANA Internet Assigned Numbers Authority.

ICMP Internet Control Message Protocol.

IETF Internet Engineering Task Force.

IoT Internet of Things.

IP Internet Protocol.

IPng Internet Protocol next generation.

IPsec Internet Protocol Security.

IPv4 Internet Protocol version 4.

IPv6 Internet Protocol version 6.

ISATAP Intra-Site Automatic Tunnel Addressing Protocol.

ISP Internet Service Provider.

LIR Local Internet Registry.

LISP Locator/ID Separation Protocol.

LSN Large Scale NAT.

LSR Label Switch Routers.

MAC Media Access Control.

MAP Mapping of Address and Port.

MAP-T MAP - Translation.

MAP-E MAP - Encapsulation.

MPLS Multiprotocol Label Switching.

NAPT-PT Network Address Port Translation - Protocol Translator.

NAT Network Address Translation.

NAT-PT Network Address Port Translation - Protocol Translator.

NAV Network Administration Visualized.

NIR National Internet Registry.

NREN National Research and Education Network.

NTNU Norges Teknisk-Naturvitenskapelige Universitet.

PE Provider Edge.

PLAT Provider-side Translator.

QoS Quality of Service.

RIR Regional Internet Registry.

SCTP Stream Control Transmission Protocol.

SEAL Subnetwork Encapsulation and Adaptation Layer.

SIIT Stateless IP/ICMP Translation.

TCP Transmission Control Protocol.

TOS Type Of Service.

TSP Tunnel Setup Protocol.

TTL Time To Live.

UDP User Datagram Protocol.

UE User Equipment.

UiA Universitetet i Agder.

VPN Virtual Private Network.

Chapter 1
Introduction

IPv4 (Internet Protocol version 4) has been the standard network layer protocol since its creation, as well as one of the most widely used Internet protocols even today. It is the basis for the operation of communications over the Internet and is responsible for identifying each of the devices connected to it through a unique address. However, the large number of devices connected to the Internet today far exceeds the expectations of when IPv4 was developed. That is why the address space of IPv4 is almost exhausted and why a new version of the protocol, IPv6, had to be developed with, among other differences and improvements, a much larger address space. Inconveniently, the change in format of IPv6 addresses as opposed to IPv4, which is what allows for more address space, in turn causes these protocols to be incompatible to operate with each other.

Due to the large size of the Internet and how widespread IPv4 was, as well as the costly process of updating all the elements that conform the Internet, the transition to IPv6 is being much slower than expected. While this transition is taking place, the number of available IPv4 addresses is running out, and that is why a large number of technologies have been developed to allow a certain degree of interoperability between IPv4 and IPv6, as well as to extend the life of IPv4. This range of technologies is presented in this thesis as the transition technologies.

The complete transition to an IPv6 network has become one of the main objectives of service providers and Internet Service Providers (ISPs) on the current scene for all the advantages this brings. This includes Uninett, the Internet service provider for academic institutions in Norway and the main focus of this work.

1.1 Motivation

Currently, it is difficult for a large ISP to rely solely on an IPv6 network because of the limitations this would cause for end users. Not all end devices are IPv6-compatible yet, which would leave a number of them unable to connect to the Internet. In

addition, connectivity to other networks would be limited to only those using IPv6, making many services unattainable. However, getting rid of the dependence on IPv4 would bring great advantages for the stakeholders, such as greater simplicity in the network by focusing on a single technology and would reduce the number of addresses needed (which cost money) and the number of network devices needed or their complexity. In general, this translates into economic advantages for the ISP, as well as management and security advantages. However, through the joint use of an IPv6 network and these transition technologies, it should be possible to greatly reduce the problems that may initially exist.

This thesis focuses on the study of the current state of IPv6 technology both worldwide and in the Uninett network, as well as on the study of these transition technologies. The aim is to find a technology that will allow for a complete transition to IPv6, leaving IPv4 behind, and to understand the impact that this could have on the ISP and its users. Interestingly, although this work is based on data about Uninett's network, it is likely that other ISPs with similar characteristics (likely other National Research and Education Networks (NRENs)) could extrapolate the result of it for them, which, in a best case scenario, could help to accelerate the state of IPv6 transition for them.

1.2 Objectives

The main objective of this work is to discuss the possibility that a nation wide ISP will be able to make a complete transition to IPv6 today and therefore be able to cease its need for the IPv4 network. Specifically, the focus is on the Norwegian NREN Uninett, and especially its core network, although its external networks will also be analyzed in order to reach conclusions. Figure 1.1 shows in a graphical way the desired scenario for Uninett in which with the help of specific nodes it is possible to rely on an only IPv6 core network. Summarizing, the main objective of this thesis can be formulated with the following knowledge question:

- Is it feasible for a nation wide ISP to move away from IPv4 and run an IPv6 core network only?

In the event that the above appears to be feasible, the design of a scenario that could be applied by Uninett to achieve the aforementioned objective will follow. This scenario would be focus on the above mentioned specific nodes. To facilitate this task, the following knowledge questions are formulated:

- What combination of relevant IPv6 transition technologies can be used to handle WAN-traffic on a nation wide scale?

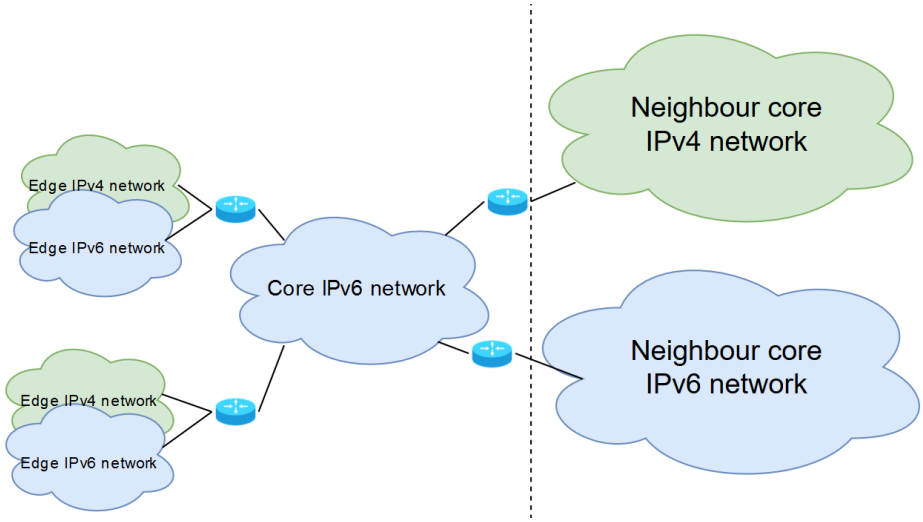


Figure 1.1: Desired scenario for Uninett

Both questions will be addressed again during Chapter 4 of the project, but in no case is an absolute answer sought for them. Many factors influence the answer to these questions, traffic and user data and their infrastructure can vary greatly between different ISP networks. Here, only the case of Uninett's network is studied and analysed. The possibility of extrapolation to other ISPs is something that should be evaluated on a case-by-case basis, and the more similar the structure and statistics of an ISP are to those of Uninett, the more useful this study may be to it.

In addition, due to the scope of the project itself, not all possibilities have been explored and those that have can be studied in greater detail. It is therefore up to each ISP to assess whether or not what is discussed here may be useful to them.

1.3 Scope

As mentioned earlier, the objective of the thesis is to study the viability of a nation wide ISP to limit its network to IPv6. This work is focused entirely on the case of Uninett and specifically on its core network. As explained, the result of the thesis may or may not be extrapolated to other ISPs.

In order to meet this objective, a preliminary study of the most relevant transition technologies that have been found has been carried out. Not all of them are explored in depth due to time constraints.

Due to the fact that the scope is limited to analyzing the viability of the proposal for the core network (i.e., that only the core network will cease to be dependent on

IPv4), it is the connection between the core network and other neighbouring networks that is the main focus. The means by which Uninett's external networks will access this supposedly IPv6 core network is explored but not analyzed in the same detail. Means will be proposed for Uninett's external networks to be able to maintain their connectivity despite the simplification of the core network. However, it is not the objective of this work to design a scenario in which these external networks also become IPv6 only.

Some of the external networks have been analyzed in greater depth, but only to give a greater context to the situation in which Uninett finds itself and the traffic load that its core network will have to weigh. Limited time and resources have prevented each of the different networks connected to Uninett from being studied in detail. Therefore the proposed solution is a general one, and knowing that at the time of applying it to the reality each case will have to be studied in detail to shape a specific solution for each one of them.

Finally, the global pandemic COVID-19 that began to show its effects in Norway in March 2020 has affected the scope of some of the studies planned for the thesis. The consequent resource limitations and the high workload that it caused in some of the collaborators of this thesis have caused detriment especially in the studies presented in sections 3.1 and 3.2. These limitations are commented with a little more detail in the corresponding chapter.

1.4 Methodology

In order to help to solve the knowledge questions formulated in 1.2, the work of this thesis is divided into several almost independent parts. Some of these parts follow the same research methods while some others do not. The different parts of the thesis are presented below, grouped by the research methods they follow, and later, in subsections 1.4.1 and 1.4.2, these methods are explained

- Research on IPv4 and IPv6 protocols. Common points and differences, degree of adoption of IPv6 at present, transition technologies between them. Research method: Literature review.

- Analysis of the selected technology. Research method: Design Science, Single-Case Mechanism Experiments.

- Analysis of Uninett network. Research method: Design Science, Observational Case Studies.

Figure 1.2 and shows in a simplified way how knowledge questions and different methods relate to each other.

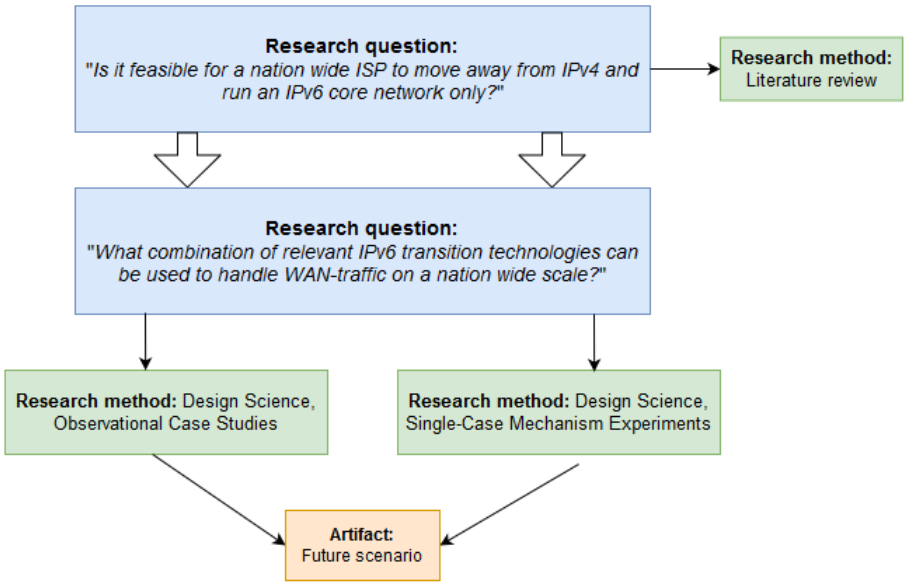


Figure 1.2: Diagram of followed methodology

1.4.1 Literature review

Literature review was the selected method to accomplish the first point of the thesis since they do not seek to create new knowledge but to obtain a better understanding of already existing knowledge in order to be able to apply it later on in the following points of the thesis. It aims to answer the first knowledge questions. The steps taken during the literature review have been:

- Select new topic of study.
- Search for relevant and reliable sources on the topic. This has mainly been limited to scientific papers and websites of official organisms or those directly related to the topic in question. The search for sources has been done mainly through the online tools Google Scholar [Goob] and ResearchGate [Res].
- Read, contrast and assimilate the knowledge of the previously selected sources.
- Synthesize this knowledge for the thesis.
- Reference the original sources in the bibliography at the end of the thesis.

1.4.2 Design Science

The second and third point mentioned above have followed a Design Science method, although due to their nature they follow different variation of the Design Science method. They can be seen as independent studies within the general method, which is Design Science, where they share the artifact and final outcome which is the model of the suggested scenario for Uninett and the context would be its network. Their goal is to answer the second knowledge question and the main stakeholder is represented by Uninett. The main points of both methods are described below and how the specific analysis were conducted is explained in their specific sections.

Single-Case Mechanism Experiments: according to Wieringa in [Wie14], this method refers to "a test of a single case in which the researcher applies stimuli to the case and explains the responses in terms of mechanisms internal to the case". This was applied to the different analysis over one of the studied transition technologies. All of them were active tests where what was analyzed was the different outcomes to specific inputs.

Observational Case Studies: according to [Wie14], the method refers to case studies in which, unlike in Single-Case Mechanism Experiments method, the tests applied were passive and no stimulus was produced. It is given in Uninett data and structure analysis. In each case, the data that were compared were done under similar conditions (same dates, parameters) and always from a position of impartiality. All of this was done with the aim of reaching conclusions about the state of the network.

The chosen validation method of the final artifact is an expert opinion. It will be presented to members of Uninett and the different details will be discussed.

1.5 Outline

- Chapter 2 presents all the background and related works for the thesis. It covers the IPv4 and IPv6 protocols, how are IPv4 addresses distributed, the current grade of adoption of IPv6, and some of the main technologies used in the transition from IPv4 to IPv6.
- Chapter 3 describes all the measures and analysis performed in order to reach the results of the thesis. It covers analysis of NAT64 usage and performance as well as an analysis of traffic and users of the Uninett network.
- Chapter 4 presents the main contributions of this work in the form of scenarios that serve as a model when redesigning the Uninett core network.
- Chapter 5 presents a conclusion to the thesis.

Chapter 2
Background and related work

This chapter presents all the background and theory considered necessary for this thesis. It also includes explanations of the functioning of a big number of technologies considered during the development of the thesis. Section 2.1 presents a description of IPv4 and IPv6 protocols; section 2.2 explains how IPv4 addresses are distributed; section 2.3 discusses the current state of adoption of IPv6 and sections 2.4 to 2.8 cover different types of transition technologies.

2.1 IPv4 and IPv6

The state of art and related work were reviewed, and an identification of the relevant background material were carried out in the project preceding this thesis [GA19]. This is amended with a discussion of a few papers that have been studied after the project.

IPv4 and the consequent IPv6 are two versions of the Internet Protocol (IP). Both are network protocols used to define the format of data packets, to provide addresses to the different interfaces of a network (e.g. Internet) and to perform routing, this is the delivery of packages from one source address to a destination address.

In 1981, the Internet Engineering Task Force (IETF) designed IPv4 and in 1983 its deployment started, today it stills the most deployed Internet layer protocol [Ali12][P+81]. IPv4 addresses follow the format A.B.C.D where each letter represents a byte and is normally expressed in decimal notation (e.g. 192.168.0.1). IPv4 addresses use an address space of 32 bits to uniquely identify all different host in the Internet. This allows up to 2^{32} different addresses, which is almost 4.3 billion addresses. Yet, due to the increasing number of Internet connected devices (mainly because of the smartphone era and the upcoming Internet of Things (IoT)) the IPv4 address pool is running out. With that in mind, in the early 1990's it started the development of a protocol to replace IPv4, being in 1994 when the IETF proposed a new version of the protocol, IPv6, also known as Internet Protocol next generation

(IPng), being [DH81] the first RFC defining it. After that, the deployment of IPv6 could begin. However, it needed time before it reached the public since a large number of protocols and technologies had to be adapted due to incompatibilities with the new protocol (more on section 2.3).

The most important difference between the protocols and most important feature of IPv6 is the address space. As said before, IPv4 has an address space of 32 bit, on the other hand, the one used by IPv6 is 128 bits long. This allows up to 2^{128} different addresses and will allow that every device can have a unique IP, even in the years to come. That big amount of addresses will also allow to stop depending on Network Address Translation (NAT) mechanisms.

NAT is a set of mechanisms used by some routers, which allows the addresses of the packets passing through these routers to be changed (in addition to those parts of the header that may be affected by this change, such as the checksum). They have different types of utilities. In relation to the scarcity of IPv4 they can be used, for example, in a private network to be able to use private addresses in communications to the Internet. When packets sent by computers within the network reach the NAT device and are routed to an external network, the private address that was being used will be changed, and one of the public addresses available for that network will be assigned to it. In this way, the distribution of IP addresses for the computers that form the private network is much more efficient, and these addresses can even be shared [SH99]. Figure 2.1 shows this operation graphically. Although NAT mechanisms have proven themselves useful over the years, they also suppose a limitation and break the end-to-end principle of the Internet.

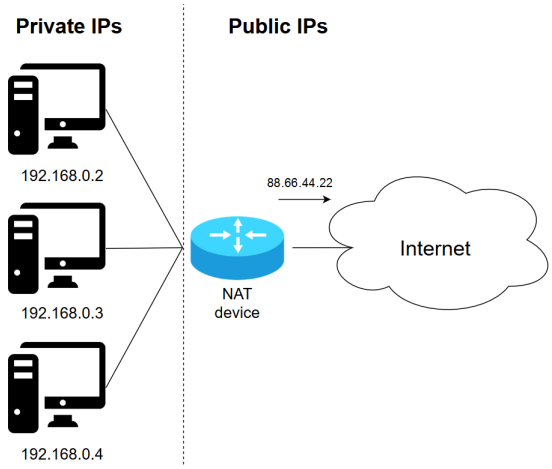


Figure 2.1: NAT environment. Packets addresses are changed when they cross the NAT

The format of the IPv6 addresses consists of 8 groups of 4 hexadecimal numbers, each group represents 16 bits and they are delimited by a colon (i.e. 20a1:12b8:19ac:45a 7:8a2e:3710:7334:6512). This new structure is less strict than the one used in IPv4 and it permits to simplify the "0" values on the addresses (i.e. 1234:0000:0000:4564 can be represented as 1234::4564[1]) [BAD14]. However, this difference between addresses provoke that the two protocols are incompatible and extra technologies are needed in order they can interoperate (this will be explored in 2.4).

Furthermore, IPv6 brings more improvements to the table. It provides labelled flows, thanks to that, routers can recognize an end-to-end flow and improve real-time communications and the overall Quality of Service (QoS) [SHP09].

Configuration is facilitated in IPv6 with a stateless auto-configuration. IPv4 needs to either be manually configured or needs the use of a DHCP server (which is also available for IPv6). This is especially relevant for big networks, in which every network device won't need to be manually configured [SHP09].

Security improvements are achieved by making the Internet Protocol Security (IPSec) mandatory [SHP09]. IPSec is a protocol suite developed by the IETF that provides authentication and encryption to the IP data packets. Unlike in IPv6, this protocol is not mandatory in IPv4.

Additionally, IPv6 allows different modes to transmit the information such as unicast, anycast or multicast (this one is defined optionally in IPv4 but not every device supports it). Broadcast addressing is no further implemented in IPv6 due to its performance related problems [SHP09].

In addition to the addresses, there are differences in the headers of the two protocols. Whereas IPv4 uses a variable header length of 20-60 bytes, IPv6 uses a fixed 40 bytes length. Also, IPv6 allows extension headers for special purposes. They are created to provide flexibility and efficiency to the packet, and they can support functions such as routing, authentication or fragmentation [SHP09].

Among many other differences, IPv6 rejects the checksum field, which simplifies the processing done by routers and can speed up connections. With IPv6 the fragmentation of packets can only happen at the sender side, unlike with IPv4 where it is possible both at sender and forwarding routers side. This is done in order to reduce time processing by the routers and improve the overall efficiency [SHP09].

Summarizing, IPv6 is a more optimized protocol, which improves the treatment of data packets, even though they are usually larger than IPv4 packets. Table 2.1 sums up some of the main differences between the IPv4 and IPv6 protocols:

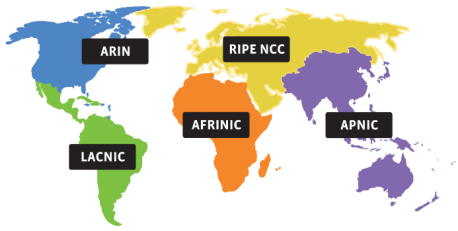
[1]Note this is not a full IPv6 address

Table 2.1: IPv4-IPv6 comparison

	IPv4	**IPv6**
Address format	32 bits (doesn't allow simplifications)	128 bits (allows simplifications)
Header size	20 - 60 bytes	40 bytes and possibility of extension headers
QoS	Differentiated services	Flow labels and traffic classes
Auto-configuration	Only through DHCP	Supports stateless auto-configuration
Security	IPSec optional	IPSec mandatory
Delivery schemes	Unicast, broadcast and optionally multicast	Unicast, anycast and multicast

2.2 Distribution and exhaustion of IPv4 addresses

Allocation of IP addresses (both IPv4 and IPv6) starts with the Internet Assigned Numbers Authority (IANA) which is the entity responsible for distributing address blocks to the five Regional Internet Registries (RIRs). Figure 2.2 shows the names and distribution of the five regions (source in [iana]).



REGISTRY	AREA COVERED
AFRINIC	Africa Region
APNIC	Asia/Pacific Region
ARIN	Canada, USA, and some Caribbean Islands
LACNIC	Latin America and some Caribbean Islands
RIPE NCC	Europe, the Middle East, and Central Asia

Figure 2.2: Distribution of the five RIR

Then the RIRs assign addresses to the different Local Internet Registries (LIRs) and National Internet Registries (NIRs) of their regions. The most common scenario is where the RIR distribute addresses directly to the LIRs since there are only a few NIRs in Asia and South America. Finally, LIRs distribute addresses to the ISPs and at the last level, users are assigned IP addresses by the ISPs.

In November 2019, RIPE NCC, the RIR responsible of Europe, west and central

Asia, announced that its remaining IPv4 pool was completely exhausted [Nik15]. Furthermore, IANA announced in February 2011 that the free pool of IPv4 addresses is depleted, and even though the remaining RIRs still have some address space, most of them are currently approximating to their limits [NCC] [ianb].

At this point the dependence on technologies, such as NAT, that can help to squeeze the current IPv4 addresses during an uncertain amount of time, is vital. However, the transition to IPv6 is becoming more demanding as the scarcity of IPv4 grows up.

2.3 IPv6 adoption in 2020

As mentioned before, all the differences between IPv4 and IPv6 have made those protocols incompatible to work together, especially the fact the address format that they use differs a lot. This has provoked an extremely slow adoption of IPv6 since the ISPs need to maintain their original IPv4 networks to keep providing service while updating its components to offer IPv6 functionality as well. This is a highly demanding process both in time and money for the stakeholders.

Nowadays, more than 20 years after IPv6 was standardized, it is present in most if not all of the biggest networks of the world. However, IPv4 still is the majority protocol in the world and it is required the use of transition technologies in order to be able to rely on IPv6 networks, since they need to maintain a connectivity with the rest of the IPv4 networks.

Besides the need to update every network device, clients and servers to be able to operate using IPv6, all the tools that they use to communicate have needed to be updated as well. This includes dozens of protocols, operating systems and technologies such as Domain Name System (DNS).

Today, the most established operating systems already support IPv6, this includes the latest versions of Windows, MacOS, Linux distributions but also mobile operating systems such as Android or IOS. The most important routing protocols (BGP, OSPF, RIP, EIGRP, IS-IS, etc) and many application layer protocols have been updated over the years and are now fully compatible with IPv6. Also, in many cases (but not always) the first Access Point in a network (especially referred to the first routers encountered in home networks) are compatible with IPv6, as well as most of commercial routers that are sell today. It is a matter of time that the old equipment gets updated.

There are several services dedicated to quantifying the presence of IPv6 in today's Internet. In [Gooa], Google creates a metric based on the total amount of access to its browser and the number of them that are over an IPv6 connection. They have

been collecting this data during more than a decade and it currently shows that the global IPv6 usage is above 30%, in addition to continued growth. They also offer individual statistics per country in which the irregularity of its distribution stands out, with countries as Belgium, Germany, Greece, Malaysia or India with values between 40% and 50% of IPv6 adoption. Then we can find average values such as USA, Mexico, Uruguay, France or Finland ranging from 30% to 40% but most of the countries of north and east Asia, Africa and Oceania don't reach a 3% or even a 1%. Generally, it is in underdeveloped countries where we find these low adoption values. Norway for instance has adopted a medium value of 11,77% IPv6 traffic.

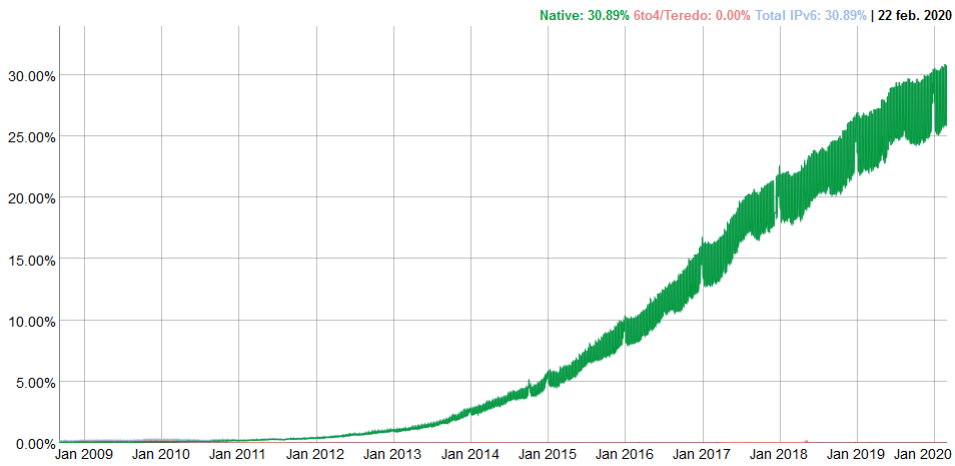


Figure 2.3: Global IPv6 adoption according to Google

There are other similar studies conducted by several private companies. Akamai in [Aka] bases its numbers on the analysis of addresses of billions of random HTTP(s) packets and offers not only a division by countries but also by networks, unfortunately they limit this information to the top 200 networks in the word, so there is no data available about Uninett. According to them the network with a higher presence of IPv6 is Comcast Cable (USA), reaching a 71%. Their division by countries shows similar numbers that the ones calculated by google.

Cisco Systems offers a very complete analysis of IPv6 utilization in [Sysd] based both in their own data as well as in data given by other of the companies that are mentioned in this section. They use the Alexa's top sties list [Aleb] per country and make a study of the percentage of them accessible by IPv6. According to them 67,79% of the top 500 of Norway are accessible with IPv6. Still referring to Norway, they calculate that the 45,98% of IPv6 addresses are already routable. Lastly, based

on old data they make a projection of IPv6 development and they expect to reach the mark of 50% of IPv6 utilization worldwide by the end of 2022 (figure 2.4).

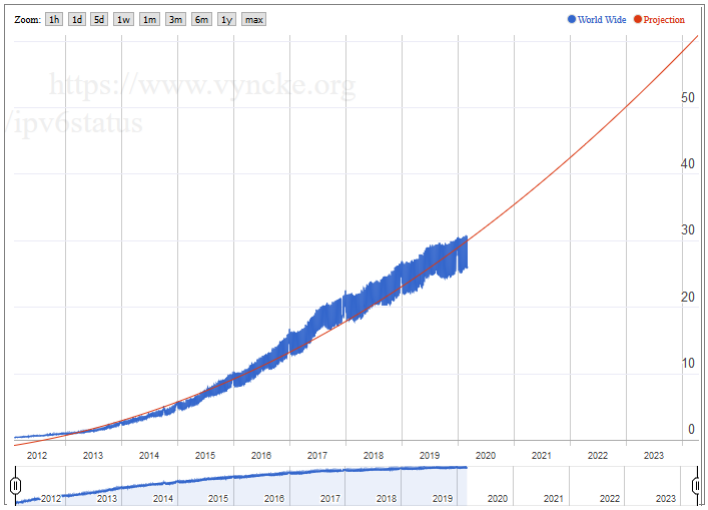


Figure 2.4: Cisco Systems' projection of IPv6 utilization

Facebook performs a very similar analysis than the one made by Google but analyzing the Facebook internet traffic. The results are showed in [Fac] and show that 26,37% of its total traffic is handled over IPv6. Their metrics divides by countries is consequent with the ones showed by the other studies.

All the RIRs offer data about the IPv6 prefixes allocated both in total and per country, about the number of networks that announce IPv6 by countries or about the number of LIRs with IPv6 resources, mainly based on the regions they operate. It is perhaps the Asia-Pacific Network Information Centre (APNIC) the one that shows the most complete data in [APN], with information about IPv6 utilization per country (this time all the countries in the world). They differentiate between "IPv6 capable" which references the percentage of users able to access online content using IPv6 and "IPv6 preference" which is the percentage of users that actually use IPv6 to establish their connections even if they have the chance to do it through IPv4. According to its data, the 24,32% of the users in Norway are IPv6 capable, while in Europe the number changes to the 20,47% and 26,84% for the Nordic countries. They also provide information about every network of each country.

All these analyses are not directly comparable among them as they base their metrics on different and excluding ways of measure the IPv6 utilization, that's why the numbers differ from one study to another. However, they are useful to have an idea of the actual state of the IPv6 implementation, both in the world and in

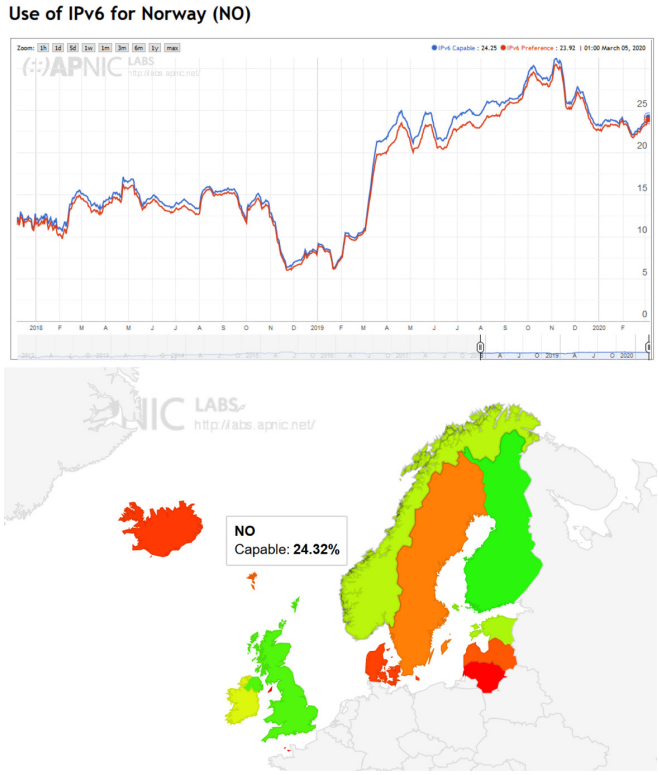


Figure 2.5: Data available for Norway in [APN]

Norway. A continued growth can be expected in the coming years and even, it could become more pronounced, as some of the initial limitations of IPv6 (compatibility with legacy protocols and systems, small user base, small availability of services) have already been overcome, at least to an extent.

Besides the networks themselves, another important factor to consider are the content providers, since many are yet not reachable through IPv6. This may be due to the state of the network they are on or the servers on which they are hosted, for example. In any case, it is important that, for IPv6 Internet to continue developing, more and more websites and content distributors support IPv6, especially important among the most visited ones.

A simple but interesting analysis consists of using Alexa's list of most visited websites [Aleb] and applying the Linux command *"host"* on them to see if they have an IPv6 address. The command *"host"* does a DNS lookup operation in order to find the IP address associated to a specific domain name (or vice versa).

In this case, it was used the list of the top 50 most visited sites in Norway and the results are shown on the table 2.2. The test was performed over the *eduroam* network at the *Gløshaugen* campus.

Table 2.2: Top 50 most visited sites in Norway and presence of IPv6 registry

Domain	IPv6 record	Domain	IPv6 record
Google.com	Yes	Yr.no	Yes
Youtube.com	Yes	Imgur.com	No
Reddit.com	No	Wowhead.com	No
Twitch.tv	No	Nettavisen.no	No
Vg.no	Yes	Ntnu.no	Yes
Facebook.com	Yes	Blackboard.com	No
Nrk.no	Yes	Fandom.com	No
Finn.no	No	Dnb.no	No
Wikipedia.org	Yes	Aftenposten.no	No
Google.no	Yes	Op.gg	No
Feide.no	No	Nav.no	No
Netflix.com	Yes	Adressa.no	No
Microsoftonline.com	No	Gamepedia.com	No
Dagbladet.no	Yes	Yahoo.com	Yes
Tv2.no	No	Uio.no	Yes
Bongacams.com	No	Vgtv.no	Yes
Livejasmin.com	No	Viaplay.no	No
Instructure.com	No	Steamcommunity.com	No
Live.com	No	Vglive.no	Yes
Imdb.com	No	Office.com	Yes
Difi.no	Yes	Discordapp.com	No
Pornhub.com	No	Ebay.com	No
Komplett.no	No	Amazon.com	No
E24.no	Yes	Redd.it	No
Itslearning.com	Yes	Tek.no	Yes

Of the top 50 sites of Norway, only for 20 of them (40%) an IPv6 record was found. An analogous procedure can be applied to Alexa's list of top 50 sites in the world, in which case we find out that for 12 of them (24%) an IPv6 record was found. Some of the sites with an IPv6 registry are within the biggest and more important networks of the world (i.e. Google, Youtube, Facebook, Netflix, etc). The companies behind them have declared several times their commitment with the IPv6 expansion,

however when it comes to smaller domains, they are still dependant of legacy IPv4 resources.

There are other means to observe the type of connection made by browsers, such as the Firefox *"SixOrNott"* or Google Chrome *"IPvFoo"* addons, as well as other online resources.

Lastly, focusing on the end users, it is possible to estimate how many of them are capable of connecting to an IPv6 network by looking at the distribution of users per operating system and how many of these operating systems support IPv6. There are several studies that seek to provide approximate data on the number of users per operating system. Each study bases its metrics on different factors, so its data are not absolute, but they are useful for obtaining an approximate idea. Here it is used the data provided by StatCounter Global Stats in [Sta], specifically from May 2020, which is based on metrics obtained directly from websites of all types based on how users access them. Then, by extrapolating data such as screen resolution or system type they can also differentiate between device types. It has been decided to rely on the data provided by StatCounter Global Stats because it was found others like [Mar] and [W3C] that offer quite similar data and give them credibility.

Its global data indicates a division of 37,81% of the devices belonging to Android systems, 35,83% to Windows, 15,28% to IOS, 8,54% to macOS and 0,79% to Linux systems, the remaining 1,75% of users belong to more specific brands such as PlayStation, XBOX or Samsung.

Subsequently, breaking down by system, it can be seen that there are a 2,39% of Android users with version below 5.0, which is where it started the support to IPv6 [Sysg]. 99,11% of Windows users use Windows Vista or later, this being the first version of the system to be fully IPv6 compatible [Mic]. In previous versions it was possible to add compatibility through external software [Bri] but for simplicity and to approximate to the less favorable scenario, it is assumed that the remaining 0,89% of Windows users are not compatible with IPv6. On the macOS side, IPv6 support has been in place since version 10.7 of the system [Wen12] and only 0,37% of macOS users worldwide use versions lower than 10.7. In the case of IOS, only 0,3% of users are below version 4.1, with which IPv6 support was started [Har]. Unfortunately, the case of Linux is not broken down between different versions of the system. However, by looking at two of the most popular Linux distributions, Ubuntu and Debian, it is found that all versions of Ubuntu that are still supported are IPv6 compatible [Ubua][Ubub], and that Debian is compatible with the technology from the version 3.0, which was released in 2002 [Con]. Therefore, and by looking at the cases of the previous operating systems, an educated approximation can be made by assuming that at least three quarters of Linux users should be IPv6 compatible, leaving that

a 0,19% of the total users without IPv6 support. With respect to the rest of the systems it is difficult to draw conclusions since no distinction is made between system versions, the most recent versions of PlayStation and XBOX support IPv6 natively as do the most modern Samsung devices while other systems mentioned represent barely 0.01% of the total. It is therefore reasonable to assume the same approach as with Linux systems, that at least three quarters of these devices are IPv6 compatible.

Correlating these numbers with the previous ones it is obtained an approximate percentage of 1,9265% of the total devices that do not support IPv6. This is always taking into account that it is based on non-absolute data and that pessimistic approximations have been made. Also, this was only focused on user-side final devices and doesn't take into account middle-devices such as routers or other end devices like servers. They should not be taken as real data but only serve to give an idea of how many devices in the hands of users are capable of connecting to an IPv6 network.

In short, the use of IPv6 on the Internet has been growing steadily over the years. An increasing number of companies and ISPs are involved with its development and encourage a faster growth. The number of compatible technologies is also increasing while the number of compatible devices it is believed to be at a very high proportion of IPv6 capable machines. Withal, due to the large size of the Internet and the high economic costs and technical difficulties of upgrading networks there is still a long way to go. It is expected, however, that the increasingly pressing shortage of IPv4 addresses will encourage stakeholders to strengthen the expansion of IPv6. On the other hand, there are (as will be seen in the following section) a number of transition technologies that facilitate the use of IPv4 and IPv6 until IPv6 reaches the desired level of development. It is plausible that with the greater number of facilities now available to interested parties, in addition to the large number of technologies already adapted to IPv6, the speed of adoption of the new protocol will increase in the coming years. This, together with the fact that IPv6 adoption can no longer be considered as insignificant, makes possible for a big ISP such as Uninett to start considering to reduce the use of IPv4 in big part of its network.

2.4 Transition technologies

In order to help to mitigate the impact of the slow transition from IPv4 to IPv6, a number of technologies have been designed with the goal of providing a sort of interoperability between these incompatible protocols. These transition technologies are often divided into three different groups: Dual Stack (which is a single method itself), tunnelling mechanisms and translation mechanisms. Table 2.3 summarizes all the technologies treated and some of their main attributes.

The following sections present some of the most relevant technologies of the three types with the aim of finding those that may be most useful in the case of Uninett, or in general of an ISP at present. They provide a small briefing of all of these technologies, divided by categories. Describing, but without deepening, the main idea of its operation, as well as its main advantages and drawbacks in comparison with others.

2.5 Dual Stack

Based on [NG+05]. It is a method that allows to have both the IPv4 and IPv6 technologies working over the same link. A dual stack node can choose to use IPv4 or IPv6 depending on the destination and the nodes available. If possible, the normal behaviour is that an IPv6 connection will be preferred over an IPv4, however this may not be always true and depends on the requirements of the connection. Also, different applications may prioritize one protocol over the other.

This method is the easiest to implement, however its biggest drawback is that it needs both an IPv4 and an IPv6 address. It helps with the transition between the two protocols but doesn't help to mitigate the IPv4 exhaustion. Besides, it doubles the configuration and maintenance efforts per interface, because of having to deal with both technologies, and also, to mitigate the incompatibilities between the two technologies this method can't stand by itself and it is normally used together with other methods in order to provide interoperability.

2.6 Tunnelling mechanisms

Tunneling mechanisms are defined as "Techniques to establish point to point tunnels by encapsulating IPv6 packets within IPv4 ones" [NG+05]. This way, packets belonging to one domain can travel through the other. There are many different implementations and subsection 2.6.1 to 2.6.11 summarizes the most relevant ones:

2.6.1 6in4

This method is usually used together with other tunnelling techniques such as 6to4 (described later on). It consists on the encapsulation of an IPv6 packet within an IPv4 one, this is achieved by adding the IPv4 headers to the IPv6 packet and treating the latter as the payload. This way, the new packet is capable of transiting within an IPv4 domain and whenever it reaches an IPv6 network it can be decapsulated to be treated as a pure IPv6 packet.

It is a simple to implement technique and it is compatible with many other transition technologies [Mau10] however it needs to be manually configured. On the other hand, the addition of an extra header increases the overall size of the packet (in 20 bytes), which decreases efficiency and increase the risk of fragmentation, which would mean transmitting a greater number of packets and therefore more redundancy with the overheads. That also leads to loss all the additional features that IPv6 brings with it, since the packet is treated as an IPv4 one [Kim17].

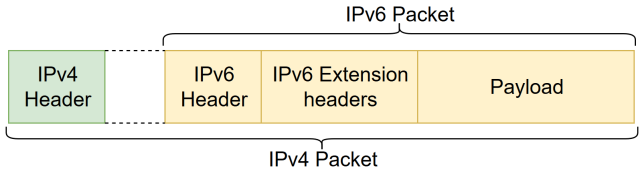


Figure 2.6: 6in4 encapsulation method

In conclusion, this method and others of the same kind were useful, and could still be, to facilitate the expansion of IPv6 networks. However, in terms of efficiency the translation methods that are mention later in this document doesn't imply the increasing of size nor the loss of capabilities and this, together with the lack of security and the existing problems with firewalls (6in4 packets uses 41 as the protocol number, and many firewalls reject packets with a number different of 6 and 17, the Transmission Control Protocol (TCP) and the User Datagram Protocol (UDP) respectively) makes it not the most preferred technology in a modern scenario.

2.6.2 Generic Routing Encapsulation (GRE)

GRE is a encapsulation protocol developed by Cisco that aims to encapsulate the payload of any kind of network-layer protocol over any other network-layer protocol [HLFT94]. It has been updated in the last years to be able to work with IPv6, both as a payload or as the delivering protocol.

GRE differentiates between the payload, which is the original packet to be encapsulated, and the delivery header, which is the IPv4 header in the case of interest of this project or IPv6 header in other cases that is appended to the payload during the encapsulation. It also adds a new header (GRE header) to the packets that encapsulates. The GRE header is placed between the delivery header and the payload and indicates, among others, the protocol of the payload and some integrity flags and fields. This new header helps in the process of decapsulation when the packets reach their destination [PB15] [HLFT94].

Similar to 6in4, GRE is another protocol that helps with the encapsulation of IPv6 packets into IPv4 ones (or otherwise). While compared with 6in4 GRE adds more redundancy and information to the encapsulated packets to facilitate to recover the original packets on the destination, it doesn't bring any new ideas that can help either to extend the IPv6 adoption or with the IPv4 address exhaustion. Also, in this scenario it is not expected to work standalone, but it should be used together with another of the transition protocols.

2.6.3 Tunnel broker

This technique is not limited to the use of IPv6 tunnels through IPv4 networks, it can be configured to work the other way around or on the same domain, being capable of offering Virtual Private Network (VPN) services. Here it is presented the basic behaviour when it is used to connect IPv6 nodes trough the IPv4 Internet (or any other IPv4 network).

The functioning principle consist on dedicated servers, called Tunnel Brokers, in charge of the creation, modification and elimination of tunnel connections. The client side (which is a dual stack node) is the entry point of the tunnel, the client starts a communication with the Tunnel Broker to set the main parameters of the tunnel. On the other endpoint there are the Tunnel Servers, dedicated dual stack nodes that configure their side of the tunnel under the Tunnel Broker commands [DFGL01].

There are several ways to implement this technique, since both the communication protocol and the encapsulation method are not specified [BP05].

This is a suitable configuration to connect isolated IPv6 sites, but, since a single tunnel needs to be configured for each of the nodes it has problems with scalability. Under a similar principle, the next method, 6to4 allows to connect several nodes under the same configuration.

2.6.4 6to4

IPv6 packets are encapsulated into IPv4 ones in order to connect isolated IPv6 "islands" (IPv6 sites connected to IPv4 networks lacking IPv6 native support) with other IPv6 sites. It doesn't need the configuration of tunnels and it uses a particular method to assign addresses within the IPv6 domain. This method requires a dual stack router that will act as a border router between the two domains and the IPv6 addresses will contain the IPv4 address that is going to be used outside the domain.

The procedure of the generation of addresses consists on a fixed prefix (2002::/16) followed by the IPv4 address of the interface that is going to initiate the tunnel in the dual stack router. After it comes the subnet ID, which is used in case that the same border router is connecting several subnets and the interface ID, which will be unique for each one of the hosts within the domain [ASS+11]. Addresses assigned with this method are known as 6to4 addresses.

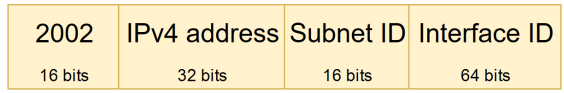


Figure 2.7: 6to4 addresses

6in4 is used later to encapsulate the IPv6 packets into IPv4 ones. As [ASS+11] mentions, "the IPv4 network is treated as the link layer". When the communication is between two 6to4 domains it happens fluently. However, when it is between a 6to4 domain and a pure IPv6 domain (one in which the addresses are not generated as shown before, generally the IPv6 Internet), the presence of 6to4 Relay Routers is needed as borders of the pure IPv6 domain. To allow the communication between 6to4 hosts and pure IPv6 hosts, there are two options, either a 6to4 address containing the IPv4 address of the 6to4 relay router is manually configured as a default gateway of the 6to4 host, or the 6to4 relay routers will use the anycast address 192.88.99.0/24 so the closest relay router can be accessed by the 6to4 routers [EVJC13]. However, this last option is actually deprecated, mainly because of problems establishing connections, principally related with the presence of firewalls [Car11].

6to4 was designed as a temporal solution. Its main drawbacks are the limitations when connecting 6to4 sites with pure IPv6 and the lack of permanent IPv6 addresses. It also lacks connectivity among IPv4 and IPv6 hosts, therefore additional methods should be implemented. 6to4 needs a public IPv4 address at the end of the tunnel, which brings some difficulties while trying to establish a connection behind a NAT since this one needs to be compatible with 6to4, this problem was solved on the later technology Teredo. It was an useful method when IPv6 was less established than today and the presence of small IPv6 islands was more frequent but it doesn't help

with today's biggest scene, however there were some other methods based on the same functional idea that came after this one.

2.6.5 IPv6 rapid development (6rd)

It was born as a variant of 6to4. The main difference with it is that in 6rd the communication is operated by the ISP. Instead of using a common prefix for the whole network of devices using the technology (2002::/16), each ISP uses its own IPv6 prefixes, the rest of the address construction is analogous to 6to4. The ISPs limit the use of the 6rd technology to its own network and therefore they have control over it, this allows to have several improvements, specially on QoS terms, where the ISP can ensure the functioning of its network, while in 6to4 the relays are often under the control of a third party [TT10]. On the contrary to 6to4, this also brings an enhanced anonymity to the connection, improving therefore the security.

In an analogous way that 6to4, 6rd is supported by two types of nodes: the 6rd Customer Edge, which is a node that operates on customer side and it is operated by the client; and the 6rd Border Relay which is managed by the ISP and is comparable with the role of the 6to4 relay routers [YPCK12].

Despite the improvements that this technology brought over 6to4, it is not efficient regarding the use of the address space since all the nodes need an IPv4 address from which the IPv6 will be extrapolated. It is a good solution to be able of maintain IPv6 communication on an IPv4 domain, and therefore to facilitate the adoption of IPv6 but the problem with the exhaustion of IPv4 addresses is still there.

2.6.6 6over4

Also called host-to-host encapsulation, 6over4 is another technology derived from 6to4. It requires double stack hosts and IPv4 multicast support, but it doesn't require any intermediate router to be IPv6 compatible [CJ99].

IPv6 addresses are derived from the IPv4 ones in a similar way than 6to4 does and then the IPv6 packets are encapsulated into IPv4 packets. Here the prefix used is fe80::/64, the last 64 bits of the address are corresponding to the IPv4 address padded to the left with 0 [CJ99].

No tunnel needs to be configured in this method but at least there must be another IPv6 host using the same mechanism within the same domain. These compatible host are discovered by multicast.

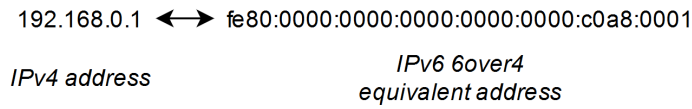


Figure 2.8: 6over4 equivalent addresses

Compared with other similar tunnelling methods, like 6to4 or 6rd, 6over4 gains in terms of simplicity since no special infrastructure is needed, however, the need of multicast support could be an important handicap in some networks since this is not guaranteed in IPv4.

2.6.7 Intra-Site Automatic Tunnel Addressing Protocol (ISATAP)

It is a similar technology to 6over4 but without the requirement of the multicast support. The process of deriving an IPv6 address from an IPv4 one is very similar to the one used by 6over4 but with some differences. ISATAP divides the address in two 64 bits parts. The first part is the link local or global IPv6 unicast prefix, while the second one is called the ISATAP interface identifier. The ISATAP interface identifier consists on the prefix 00:5EFE followed by the 32 bits of the IPv4 address (although the prefix may have small changes to indicate the type of IPv4 address, if it is globally unique or if it is an individual or a group [TGT+08][Sys98]).

ISATAP uses a basic tunnelling mechanism defined in [NG+05] at the link layer, which suppose another difference with 6over4 which works on the network layer. The operation at the link layer is quite simple since it only takes the last 4 bytes of the ISATAP address and treat them as an IPv4 address [TGT+08]. It also uses the basic neighbor discover technique for IPv6 in order to avoid the requirement of the multicast support or the presence of dedicated equipment to announce the ISATAP nodes [NNS06].

In conclusion, we can see ISATAP as the next step in the evolution of tunnelling techniques that was started by 6to4. It solves some of the incompatibilities of previous technologies by simplifying more the required infrastructure. However, it has the same problem that all the others and is the need of the IPv4 stack on the machines that want to use this technology and therefore, the need of a dedicated IPv4 address.

2.6.8 Teredo

Designed by Microsoft and later standardized by the IETF, this technology is also based on 6to4 and its main advantage among the others is the possibility of work behind a NAT.

As mentioned before, the 6to4 technology needs a public IPv4 address. When a host is trying to establish a 6to4 connection behind a NAT, it is required that the NAT machine is compatible with 6to4 in order to be able to establish that connection. Teredo solves this by changing the transport protocol and encapsulating the IPv6 packets into UDP IPv4 ones (6to4 and other similar technologies used the protocol type 41), which is permitted by the NATs in almost every situation. However, this doesn't work with every kind of NAT devices, it depends on the way these devices do the mapping of inside and outside addresses, being symmetric NATs incompatible with the technology [Hui06].

Comparable with previously tunnelling techniques, Teredo also derives the IPv6 addresses of its clients from the IPv4 ones that they use. Being more specific, the Teredo IPv6 address is conformed by the Teredo service prefix (32 bits), the IPv4 address of the Teredo server (32 bits), some flags to indicate type of address and NAT (16 bits), the UDP port used by the Teredo service in the NAT mapping (16 bits) and the NAT mapped IPv4 address of the client (32 bits) [Hui06].

Teredo defines a variety of nodes in order to establish its connections. The most important are: the Teredo clients, which are the hosts using Teredo to establish IPv6 tunnels; the Teredo servers, which are nodes with access to the IPv4 Internet and that help the Teredo clients to establish IPv6 connections; and the Teredo relays, that act as IPv6 routers between the Teredo clients. The Teredo relays can also provide interoperability with other tunnelling techniques [Hui06].

Again, Teredo offers a temporal solution designed to solve some of the existing problems with other similar technologies, but keeping other of its main problems.

2.6.9 6PE

The name of this technology comes from IPv6 Provider Edge (PE) routers and, so as Teredo did with the existing incompatibilities of establishing automatic tunnels behind NATs, this one aims to do the same with the problems originated by the Multiprotocol Label Switching (MPLS). MPLS originally didn't natively support IPv6, however there exists several studies and different approaches on how to add that support, being the one used by 6PE probably the simplest model of all.

MPLS is a routing technique based on the use of labels. Labels are used to

identify different routing paths and they are added to the packets, when routers read the labels, they forward the packets over one direction or another. This is made in order to speed up the traffic flow. The MPLS infrastructure basically consists on a set of Label Switch Routers (LSR), which read labels and forward the packets accordingly, and edge LSR, which are the routers on the edge of the MPLS network and take care of adding and removing labels from the packets[2].

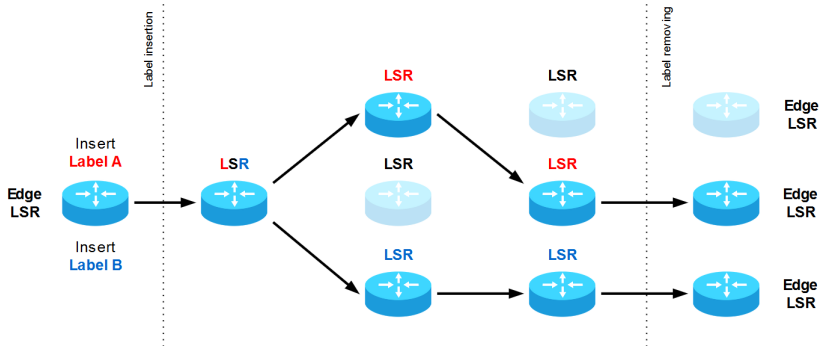


Figure 2.9: MPLS schematic

What 6PE proposes in order to be able to work with IPv6 and MPLS is to simply update the Edge LSRs, they need to be dual stack routers, while the core LSRs will keep routing and signalling using IPv4. The communication between the edge LSR and the edge router of the IPv6 island (Customer router) is done through pure IPv6. IPv6 packets don't need to be encapsulated, but instead, an IPv4 address will be mapped as part of the IPv6 address of the edge LSRs. The core LSRs would have already shared routing information through the Border Gateway Protocol (BGP) and those IPv4-mapped IPv6 addresses will appear on the routing table of all the LSRs of the network and whenever the receive an IPv6 packet directed to one of these addresses will only have to check which is the next hop on its routing table and forward the packets [UOUS03] [DCOPLF07].

Once again what 6PE proposes is a solution to interoperate the tunnelling techniques necessaries to connect different IPv6 islands among them with an already pre-existing technology, MPLS in this case.

2.6.10 4in6

Although the point of interest of the present project are the technologies that help IPv6 packets to transit an IPv4 network, since this is the most common scenario,

[2]Here the functioning and infrastructure of MPLS is highly simplified in order aim for a lower complexity and focus more on the transition technology itself.

with a pre-established IPv4 Internet and a IPv6 Internet still on development; there exist some like 4in6 that help on the opposite scenario.

In 4in6, IPv4 packets are added an IPv6 header, in a completely analogous mechanism that the one applied with 6in4. This sums at least 40 bytes of overhead. 4in6 tunnels need to be manually created, however there exists some protocols, like the Tunnel Setup Protocol (TSP) [BP05], that help to make this process automatic [CD+98].

At this point, technologies that help IPv4 packets to transit an IPv6 network are not needed in most of the situations due of the high presence that IPv4 has on the Internet. However, IPv6 is expected to keep growing and eventually it will reach a point in which it will surpass the presence of IPv4. When that time comes, it will be needed another type of transition technologies to fill the gap until everything goes under IPv6. By now, there could be stakeholders that may be interested in having a full IPv6 core, while keeping IPv4 subnetworks within its domain (such as Uninett), for them, this could be an interesting technology to consider. That's why it is important the "early" development of technologies such as 4in6.

2.6.11 Dual-Stack Lite (DS-Lite)

Despite its name, this technology has more things in common with the tunnelling techniques. It permits to transit IPv4 packets from dual stack devices and through an ISP's IPv6-only network. Its biggest feature, however, is that it permits to share IPv4 addresses among several hosts of the network with the use of a Large Scale NAT (LSN).

IPv4 packets are encapsulated into IPv6 ones to transit the ISP's IPv6 network within a tunnel and decapsulated on a special device at the end of the network. To do so DS-Lite uses 4in6. Its infrastructure consists of two key elements, the Basic Bridging BroadBand (B4) and the Address Family Transition Router (AFTR). The B4 is a dual stack device and it is the point from which customers connect to the IPv6-only ISP network. The B4 is usually a component of the Customer Premises Equipment (CPE), which is the device that connects the subscribers and the ISP's network. The connections between the CPE and the client's equipment can be established through IPv6 or IPv4, anyway the clients will use their private addresses. The B4 IPv6 address is known as the B4 tunnel endpoint. On the other hand, the AFTR is a device within the ISP core network and acts as the other endpoint of the tunnel. The AFTR decapsulates the outgoing packets and forward them to a LSN. The LSN performs a NAT translation and sends the decapsulated IPv4 packets to the IPv4 Internet. Usually the AFTR and the LSN are part of the same machine.

The main purpose of these two devices is keep a communication between them and establish the IPv6 tunnel [DDWL11].

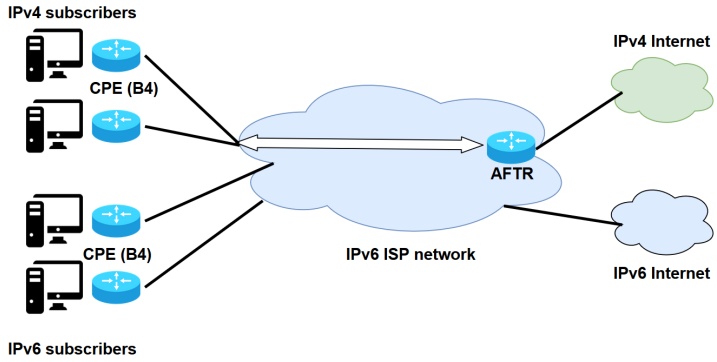


Figure 2.10: DS-Lite architecture

The main requirement of the whole technology is its architecture itself. The ISP needs to deploy enough AFTRs to provide service to all the expected tunnels and it also needs to update all the CPEs of its subscribers in order to incorporate the B4 component. The B4 is not needed for IPv6 connections, however, and with the strong presence that IPv4 has on today's Internet, it is almost mandatory for dual stack subscribers in order to be able of making IPv4 connections when needed. IPv6 customers can travel through the ISP's network and to the IPv6 Internet without any particular requirements [DDWL11].

Dual-Stack Lite can prove to be an interesting alternative for those ISPs interested in updating to IPv6 while they still need to provide connection to IPv4 machines, being its main advantage against other tunnelling technologies the presence of the LSN that can help to decrease the need of IPv4 addresses. Yet, the investment needed to have the necessary architecture is a factor to consider.

These are not the only existing tunnelling mechanisms. There are some others (Softwire, Anything In Anything (AYIYA), TSP, Locator/ID Separation Protocol (LISP), Subnetwork Encapsulation and Adaptation Layer (SEAL), Silkroad, Dual Stack Transition Mechanism (DSTM), 6bed4...) not presented in this document because of their few differences with those already mentioned. Most of the tunnelling techniques were designed as a solution to some existing lack of the previous ones (incompatibility with NAT, MPLS...) or simply to enhance the performance, security

or to reduce the pre-existing requirements. Nevertheless, the general method of tunnelling hasn't changed much over the years, and it is a method designed with the idea of be a temporary solution, not only because within time, IPv6 will catch up with IPv4, and eventually there will only be an IPv6 Internet, but mainly because none of its applications provide a solution for the IPv4 shortage; what makes it temporary even for the transition itself. At this time, the available amount of IPv4 addresses has become an important factor on networks design. The fewer amount is needed will result on less problems during the upcoming years and on a smoother transition to a future pure IPv6. With few exceptions, such as 6PE, all the studied tunnelling technologies require dual stack hosts, which make them require a unique IPv4 address together with an IPv6 and for these reasons they should not be the preferable option at this point of the IPv6 transition. However, the ones that are reversible (the ones that are able to connect IPv4 hosts through IPv6 tunnels) may prove relevant in the upcoming years, when the situation will be reversed and IPv4 islands will need to be connected through the IPv6 Internet.

2.7 Translation mechanisms

In this type of techniques, IPv4 and IPv6 packets are transformed into the other type when they leave their respective domain, e.g.: an IPv6 packet will be transform into an IPv4 packet when it is about to leave an IPv6 network to access and IPv4 one.

We can differentiate between stateless and stateful translation. In an stateless one, the state of the translation is not kept, the nodes in charge of the translation perform a 1:1 mapping between IPv4 and IPv6 addresses. On the other hand, in an stateful translation the mapping is done between the old address (let's say an IPv6 address) and the whole pool of addresses (let's say they are IPv4) within its network, it is a 1:N association. In this way, a small amount of IPv4 addresses could be used to provide connection with the Internet to a bigger amount of IPv6 addresses, which can help with the IPv4 addresses scarcity. This is achieved by keeping a dynamic state. Again, there are several approaches to these techniques:

2.7.1 Stateless IP/ICMP Translation (SIIT)

Technology able to connect IPv4 only with IPv6 only hosts. It is based on a special device responsible of changing from one type of header to the other. Contrary to other later technologies, SIIT doesn't keep the state, which means that the translation between types of addresses is algorithmic and it is needed one IPv4 address for each IPv6 one. The standard has been re-evaluated several times over the years.

The main component of this technology are the XLATs (IP/Internet Control Message Protocol (ICMP) translator), which are machines capable of translating between IPv6 and IPv4 headers and between ICMPv4 and ICMPv6. The XLATs are usually placed at the borders of IPv6 networks. There, when an only IPv6 device is trying to establish a connection with an IPv4 one, the first one sends IPv6 packets to the second. When those packets reach the border of the network they are treated by the XLAT, translating the headers from IPv6 to IPv4, assigning a new IPv4 address as source of these packets and when the response is received by the XLATs, the IPv4 packets are changed back to the IPv6 domain, reassigning the original IPv6 address to them. This translation has some limitation and for instance it doesn't translate the extension headers and the IPv4 options are also not considered. There are specific types of packets that are not translated by the XLAT [ALG16].

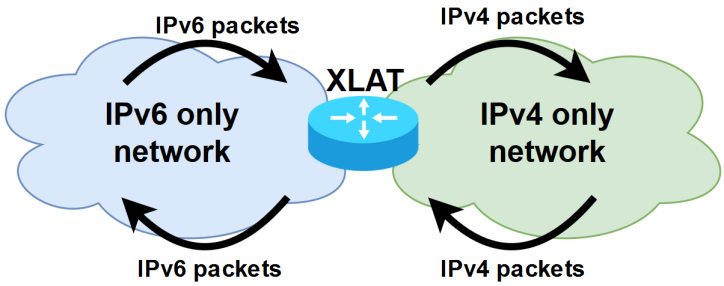


Figure 2.11: SIIT mechanism

In order to achieve the translation, SIIT defines an IPv6 address range, called IPv4-converted addresses, that represents the entirety of the IPv4 range in an IPv6 format. To do so, the XLAT (and the IPv6 sender) appends a prefix to the original IPv4 address to conform an IPv6 one, the prefix varies between different organizations.

SIIT also defines a set of IPv4 addresses, called IPv4-translatable addresses, that are part of the ISP IPv4 pool address and that are algorithmically mapped to IPv6 addresses. Basically, the IPv6 given addresses within the SIIT domain are composed of the same prefix used to translate external IPv4 addresses to IPv6, and one of the IPv4 addresses available from the ISP. Thus, when translating the source address of one of the internal computers on the IPv6 network, the only thing the XLAT does is to remove the part of the prefix [ALG16]. Figure 2.12 shows in a simplified way the transformations performed to both the source and destination address of the packets in a communication that goes from an IPv6 domain to an IPv4 one and returns.

SIIT proposes a completely different mechanism that the ones that had been utilized at the moment and allows to have a transparent routing for the packets traversing through different domains without having to update the end nodes. Its

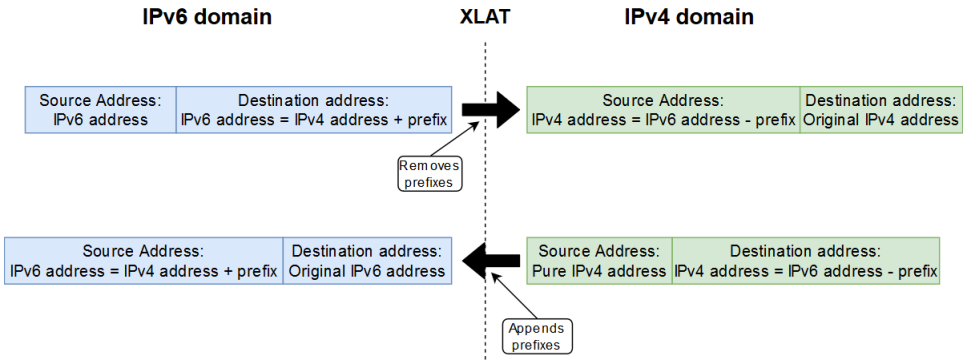


Figure 2.12: SIIT translation of addresses

biggest limitation is the lack of state during the translations, which forces the mechanism to apply an algorithmic translation that reduces the versatility of the system and the use of the address space. Nonetheless, it opens doors to new paradigms and seems the way to follow on a bigger and with more presence IPv6 Internet.

2.7.2 Network Address Translation - Protocol Translation (NAT-PT)

NAT-PT is a SIIT based technology that proposes to use a small pool of IPv4 addresses to be dynamically assigned when translating IPv6 packets. This way, the necessity of IPv4 addresses is lower than the number of devices, which is a benefit for the stakeholders.

It was born to solve a lack of SIIT, since originally it didn't define a mechanism to assign the IPv4 addresses to the IPv6 interfaces, although this was solved on later releases of the standard [ALG16].

It replaces the XLAT by a NAT-PT node which besides having to translate the packets that wants to travel from one domain to the other, it also keeps the session of the connections and ensure that the interfaces that it is keeping the session of, traverse through the same NAT-PT node. NAT-PT also distinguishes between traditional NAT-PT, which only keeps the state on one direction of the communication (from IPv6 domain to IPv4), and bi-directional NAT-PT, which is able to keep the state in both directions [TS00].

The session is kept thanks to some improvements on the translation techniques that allow to differentiate between identifiers, such as UDP and TCP ports or ICMP

query identifiers. With that additional information, the NAT-PT can assign the same IPv4 addresses with minor changes, like the source port, to several IPv6 addresses when they try to communicate with an IPv4 network [TS00].

An important requirement of the NAT-PT technology is its dependence with DNS-ALG (Domain Name System Application Level Gateway) in order to be able to send correct DNS queries to the DNS servers due to the fact that some of the packets may not be using real addresses and that NAT-PT can only modify the headers but not the payload of a packet. It is sometimes embedded as part of the NAT-PT node. DNS servers associate domain names with addresses. Domain names are usually the first information that a host knows about its destination and later, with an address, the host can send packets that will be routed until the destination. The two most common types of records in a DNS server are the A record, which maps a domain name with its corresponding IPv4 addresses, and the AAAA record which does the same but with an IPv6 address instead. On the payload of a DNS query it is stated the type of record that is being requested to the DNS server. The main function of a DNS-ALG is to intercept the DNS queries going through the boundary of a domain, and adjust the payload to match with what is needed, or if it is unknown, to request both A and AAAA records to the DNS server (figure 2.13). If both are available AAAA records are preferred over A records. It also informs the NAT-PT about the mapping of addresses so it can include it on its table [STAH99] [KTA07].

However, NAT-PT has been deprecated since the release of the [AD07]. This is due of several security problems and operative issues, being many of them related with the way DNS queries are handled. Specifically, NAT-PT raises problems with dual stack nodes, multi-addressed nodes and with multicast traffic among others.

2.7.3 Network Address Port Translation - Protocol Translator (NAPT-PT)

Its main difference in comparison with NAT-PT is that it is not only able to differentiate between transport identifiers (TCP/UDP ports, etc...) but also to translate them. This allows to assign more IPv6 addresses to each one of the IPv4 ones by being able to change the ports of it and establishing up to 63.000 TCP and 63.000 UDP sessions per IPv4 address. brings more flexibility in order to multiplex several IPv6 transport identifiers into the transport identifiers of a single IPv4 address [TS00].

This however limits the number of inbound services sharing the same IPv4 address to one of each service. This is due that each service have a specific TCP/UDP port associated. For example, there couldn't be more than one Hypertext Transfer Protocol

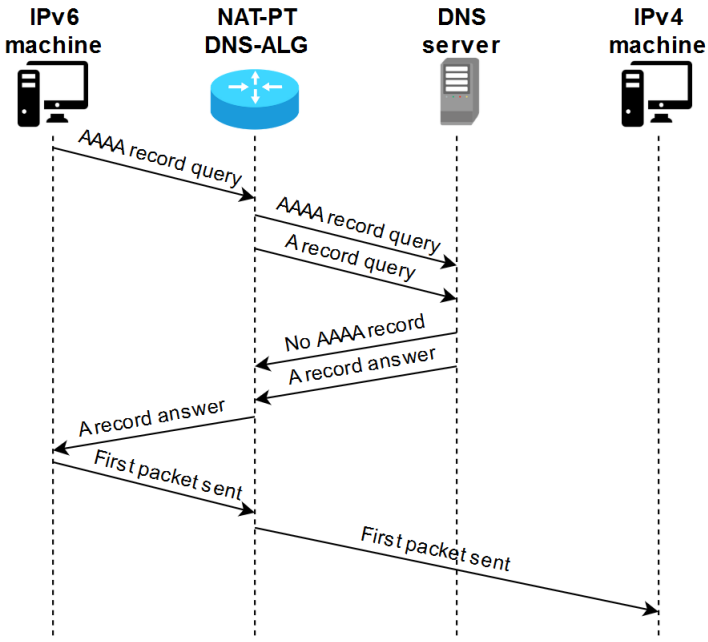


Figure 2.13: DNS procedure with DNS-ALG

(HTTP) server inside the IPv6 domain using the same port and the same address outside this domain [TS00].

Same as NAT-PT, NAPT-PT is unable of handling IP addresses embedded into the payload. This method doesn't solve many of the existing problems with NAT-PT, it only expands its functioning limit to operate with transport identifiers and, as NAT-PT it as been deprecated by the IETF.

2.7.4 NAT64

NAT64 is a translation technology originated as an evolution of NAT-PT and which solves most of its original problems. It has two modes of operation: Stateless and Stateful. Only the Stateful mode is covered here because of the few relevant differences between the two modes and also between Stateless NAT64 and other previously covered Stateless mechanisms.

Similar to NAT-PT, Stateful NAT64 is a translation mechanism that needs to be used together with DNS64. NAT64 and DNS64 together allow IPv6 only hosts to communicate with IPv4 servers via TCP, UDP or ICMP as well as peer-to-peer communications. The communication needs to either be started by the part on

the IPv6 side, or if it comes from the IPv4 domain it needs to either be manually configured first or to have a pre-established session.

There are several components needed for a NAT64 approach. The most important of them is the translator, which is placed at the limit between the IPv6 and the IPv4 networks (but is part of the IPv6 one). The packets of an IPv6 initiated communication are routed through the IPv6 network until they reach the translator and it changes them to the IPv4 type. The translation consists in two differentiated parts: the protocol translation and the address translation.

The protocol translation is performed similarly than with other comparable technologies. The headers are translated by the standard IP/ICMP translation algorithm [LBB11]. This method doesn't consider the IPv4 options neither the IPv6 extension headers (with the exception of the Fragment header, which is used to reassemble fragmented packets) due to the different size of headers. The previous header is substituted for a new one with the source and destination addresses updated to the new domain and the checksum is recomputed. Some values of the headers are derived between the two headers, like the Hop Limit in IPv6 and the Time To Live (TTL) in IPv4 while some others are directly copied from pre-existing ones, like the Traffic Class, which is converted into the IP Type Of Service (TOS), when translating IPv6 to IPv4. There are other fields, like the Fragment Offset that share name and value in the two domains.

The addresses translation is performed algorithmically. The IPv6 format of the addresses used in NAT64 is known as IPv4-embeded IPv6 addresses and it consists on a variable length prefix, an IPv4 address embedded on the IPv6 one, and a variable length suffix. The prefix can be either a standard one or one specific for the organization and the network. Prefixes can be 32, 40, 48, 56, 64 or 96 bits long [BHB+10].

There is also an octet used to ensure compatibility with other IPv6 formats. All the bits of the octet are set to 0. This octet is always placed in the 64 to 71 bits and, depending of the length of the prefix and suffix, it may split the IPv4 address in two parts, before and after the special octet. Only when a standard prefix is used neither the special octet nor the suffix is needed [BHB+10]. Figure 2.14 shows some of the variations of the IPv4-embeded IPv6 addresses.

Instead of associate IPv4 addresses to IPv6 addresses and vice versa, what NAT64 does is to associate IPv6 and IPv4 transport addresses. These addresses are fundamentally the same that normal addresses, but they are used to identify sessions. This way once a pre-established session is over, the addresses that it was using become available to be used by other communications. This is the same mechanism that NAT-PT and NAPT-PT use and it allows an efficient use of the IPv4 address

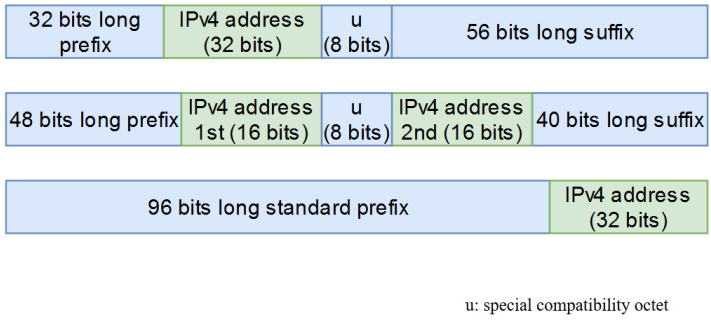


Figure 2.14: IPv4-embeded IPv6 addresses

pool, and, same that NAPT-PT, NAT64 also allows to translate the ports. The duration of the session is based on timers. This mechanism allows to the stakeholders to perform an efficient use of resources, allowing to use a small IPv4 address pool to provide service to many devices.

The biggest change of NAT64 to the previous mechanisms is the way DNS queries are handled. NAT64 is used together with DNS64 which is a "mechanism for synthesizing AAAA records from A records" [BSMVB11]. DNS64 is used to ensure that IPv6-only clients can initiate communications with IPv4-only servers by using their name.

A DNS64 acts as a regular DNS for the IPv6 initiator. Whenever an IPv6 host wants to start a communication with a server and needs its address it will request an AAAA record to its local DNS64. The IPv6 host needs an IPv6 address so it can route through the IPv6 network and therefore, the DNS64 will look for AAAA records. If the DNS64 doesn't have a registry for the requested domain name, it will look for external DNSs and request an AAAA record to them. If there is no AAAA record for the requested domain name an A record will be requested instead. If that is the case, the DNS64 will transform the received IPv4 address of the server into an IPv6 one by using the same method that NAT64 uses to translates address when exiting the network. With that IPv6 address, the client can start sending packets to the NAT64 router at the border, the NAT64 will change both the source and destination addresses into IPv4 ones and, since it uses the same method that the DNS64, the resulting destination address will be the actual address for the server [BSMVB11]. Figure 2.15 shows a simplified version of the message flow.

Usually a network may require of several DNS64 servers and NAT64 routers accessible by the IPv6 hosts. The NAT64 functionality can be installed in a handful of devices so there is no requirement of specific devices for this purpose, but it is

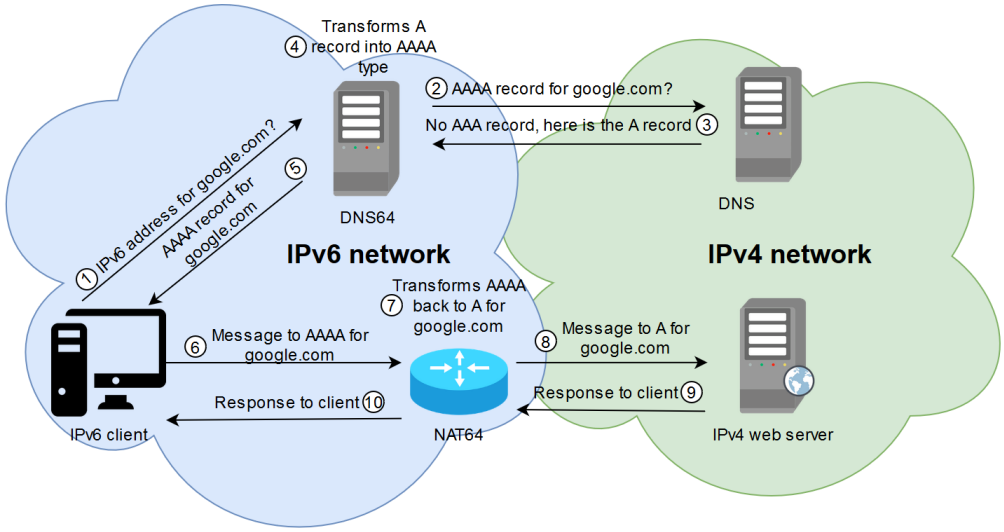


Figure 2.15: DNS64 message flow

required the presence of at least one of them for every border between the IPv6 network with an external IPv4 one. It is necessary enough DNS64 servers to provide service to all the IPv6-only hosts of the network. The DNS64 functionality can be located on specialized servers or can also be installed at the end hosts [BSMVB11]. The use of both NAT64 and DNS64 doesn't require changes for the IPv6 clients nor the IPv4 servers [BMvB+11].

Translation mechanisms usually stand out for their ability to allow a smaller dependence on IPv4 addresses, permitting to share one address with many devices and with this helping to the stakeholders and the overall limited amount of IPv4 addresses. They also bring more flexibility and they need less dedicated hardware on the network than the tunneling techniques and they allow to benefit from some of the improvements that IPv6 brings over IPv4. However, all these advantages comes with a more limited scenario. With all the different protocols and technologies in the Internet, it is expected that some issues and incompatibilities will rise with the use of translation technologies, as Škoberne and Ciglarič demonstrate in [SC11], in which they study the limitations of these technologies over specific application-layer protocols. It depends on later versions of those protocols and the presented methods

to overcome such incompatibilities.

2.8 Draft and uncertified proposals

Here are presented a few technologies that at the moment of writing this document haven't been standardized by the IETF yet, but they have been several years under evaluation. These are not treated into detail since they are still subject to change. However, it is important to have in mind that there are more technologies under development that the ones already existing and that they could be relevant in the near future. It is especially important that they exist in order to anticipate the upcoming reverse situation in which there will be needed techniques to provide connection to IPv4 equipment in an IPv6 Internet.

2.8.1 4rd (IPv4 Residual Deployment)

Stateless tunnel technology that works as the inverse for 6rd. This technology doesn't have a specification yet and is still under evaluation [PLCC15].

4rd focus on communicating isolated IPv4 islands through an IPv6 domain. It only appends a prefix to the IPv4 addresses in order to transform them into IPv6 addresses and doesn't modify the transport identifiers, relying in "TCP/UDP IPv4 packets are valid TCP/UDP IPv6 packets during domain traversal" and also to keep the IPv4 fragmentation rules [PLCC15]. It is compatible with other technologies such as NAT64.

Its main limitation is that it needs dual stack nodes to initiate communications and also a NAT64 or a Border Relay (BR) to act as endpoints of the tunnel.

This technology doesn't bring new ideas to the table, instead it tries to get ahead of the future situation in which the IPv6 networks overpass the IPv4 ones, same as 4in6 for instance.

2.8.2 MAP (Mapping of Address and Port)

Same than some of the mechanisms mention before, this technology is being planned with the moment in which the IPv6 adoption overpass IPv4 in mind. It hasn't been standardized yet and it stills under development by Cisco Systems. Currently there are two versions of the technology, MAP-E (MAP - Encapsulation) which is oriented as a tunnelling technique and MAP-T (MAP - Translation) which behaves similarly to other translation mechanisms.

MAP-E relies on the presence of NAPT nodes and MAP border relays on the edge of the domains to encapsulate IPv4 packets into IPv6 ones. It defines several ways to perform the encapsulation as well as different IPv6 prefixes to append at the beginning of the IPv4 address depending of the type of connection that wants to be established. None of those methods differs a lot from the ones exposed earlier on this document. As other tunnelling technologies, MAP-E doesn't allow to share IP addresses [TDL+13].

On the other hand, MAP-T is highly based in technologies such as NAT64 but oriented to provide connection to IPv4 islands in an IPv6 Internet. It depends of border relays with similar functions than the NAT machines. However and since there will be more than enough IPv6 addresses, they don't need to be shared and the translation mechanism is stateless, contrary than with NAT [LBD+13].

2.8.3 NAT46

The case of NAT46 is a peculiar one since, although it hasn't been certified by the IETF or other official institution, it is used by the main manufacturers, and therefore it is possible that it was developed by them. Information about the technology as well as its configuration and compatible models can be found through their websites [Syse] [Neta].

NAT46 acts precisely as the reverse of NAT64 and provides IPv6 connectivity to IPv4 hosts. Some of the drawbacks and limitations of the technology are the same than with NAT64, as NAT46 is not compatible with some of the IPv4 or IPv6 features after the conversion, however none of them is considered critical. Analogous to NAT64, NAT46 also requires the presence of a special type of DNS server, DNS-ALG, which was already mentioned during section 2.7.2. As in NAT64, a translator node is needed, the NAT46, which will perform a translation between IPv4 and IPv6 packets by means of a prefix, as in other translation technologies, and establish bindings between external IPv6 addresses and those belonging to the IPv4 address pool. As for DNS requests, if only one AAAA record is obtained from a given destination, the DNS-ALG server is able to extract the IPv6 address from the record and perform a translation to an IPv4 address, same way that DNS64 does the other way around. This way, the IPv4 host that initiated the communication is able to respond to the desired destination.

NAT46 performs a 1:1 address allocation between IPv4 and IPv6 domains, so it is not a technology suitable for reducing the use of IPv4 addresses by itself.

After carefully reviewing all the technologies presented in this chapter, it has been concluded that NAT64 may be the most suitable for Uninett's case study. The reasons for this decision are that it allows a very efficient way to reuse IPv4 addresses (and therefore it helps to mitigate the IP4 exhaustion); it is easier to implement than other methods, as it only needs some of the nodes of the network to have NAT64 capabilities and the DNS64 servers, while other methods require many compatible devices; and it solves many incompatibilities that previous technologies have. In the following sections 3.1 and 3.2 a study is carried out on the limitations of use of NAT64 and on its performance.

Table 2.3: Main aspects of the different transition technologies

Name	Main attributes	Helps with IPv4 exhaustion	Status	Subsection
Dual Stack	-Easy to implement -Needs other technologies to connect IPv4 and IPv6	No	Active	2.5
6in4	-Easy to implement -Reduces efficiency	No	Active	2.6.1
GRE	-Easy to implement -Needs other protocols	No	Active	2.6.2
Tunnel broker	-Great versatility -Needs dedicated nodes -Problems with scalability	No	Active	2.6.3
6to4	-Connects IPv6 islands -Doesn't provide connectivity between IPv6 and IPv4	No	Active	2.6.4
6rd	-Based on 6to4 -Limited to ISPs networks	No	Active	2.6.5
6over4	-Based on 6to4 -More simplicity than with similar methods -Requires multicast support	No	Active	2.6.6
ISATAP	-Improvement over 6over4	No	Active	2.6.7
Teredo	-Based on 6to4 -Can work behind a NAT -Allows interoperability with other tunneling techniques	No	Active	2.6.8
6PE	-Allows to work with MPLS -Requires to upgrade some nodes	No	Active	2.6.9
4in6	-Encapsulates IPv4 packets within IPv6 packets -Requires configuration of tunnels	No	Active	2.6.10
DS-Lite	-Used together with 4in6 -Requires specific equipment	Yes	Active	2.6.11
SIIT	-Improves efficiency compared to tunnelling techniques -Requires specific hardware	No	Active	2.7.1
NAT-PT	-Based on SIIT -Requires specific hardware	Yes	Deprecated	2.7.2
NAPT-PT	-Based on SIIT -More flexibility that NAT-PT	Yes	Deprecated	2.7.3
NAT64	-Based on SIIT -Requires specific hardware	Yes	Active	2.7.4
4rd	-Inverse of 6rd -Communicates isolated IPv4 islands	No	Draft	2.8.1
MAP	-Under development by Cisco Systems -Two versions: encapsulation and translation	No	Draft	2.8.2
NAT46	-Analogous to NAT64 but to grant IPv6 connection to IPv4 hosts -Requires the use of DNS-ALG	No	Uncertified	2.8.2

Chapter 3
Data-collection and analysis

This chapter presents the different tests and analysis of data carried out during the completion of the thesis. Section 3.1 presents an analysis of use of the NAT64 technology and section 3.2 covers an analysis of the performance of the technology. Section 3.3 analyses Uninett network with respect with its IPv6 adoption, while 3.4 and 3.5 deepen in their edge and core networks respectively.

3.1 Analysis of use

In order to get an idea of the limitations of use that a dependence on an IPv6 connection along with the use of NAT64 and DNS64 can have, the following test was performed. Over a period of time the IPv4 capabilities of a personal computer network adapter were disconnected and the use of a DNS64+NAT64 server was forced. This was done with the purpose of testing as many websites and programs that require Internet use as possible, to understand how much limiting can be a reliance on an IPv6 + NAT64-DNS64 connection. The test was always conducted with the assumption that the target users are mostly young students.

The DNS64 server was provided by Uninett and the necessary configuration was done in the network adapter configuration, the Wi-Fi adapter in this case. No complex configuration was necessary since Windows 10, the operating system used, allows to make these changes natively, as shown in the figure 3.1. As mentioned in the introductory chapter, the global situation at the time of this work affected this and other parts of it. During much of the development of the thesis, the workplace was limited to one without IPv6 connectivity. This meant that these tests had to be postponed until the last few weeks of the project, thus limiting the testing time and scope of the test.

Furthermore, this study should not be taken as a complete analysis due to its limited nature. It was conducted on a single computer, on a single operating system and by a single user, therefore the variety of services, applications or websites

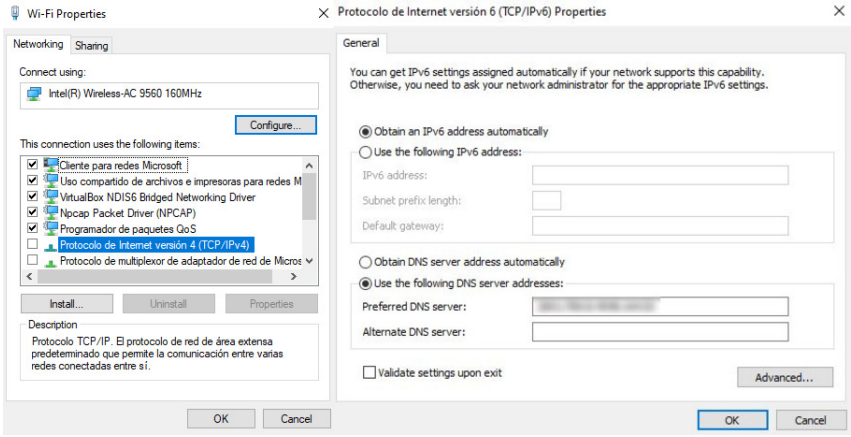


Figure 3.1: Network adapter configuration

tested were very limited to the user experience. Attempts were made to extend the range of testing, but it remains limited and should therefore only be taken as a preliminary guideline study. Also note that here are mentioned many of the websites and programs that were tested, the most relevant, but not all of them. Many are mentioned implicitly as part of a thematic set. However, all of whom were found to present problems, are explicitly mentioned. Table 3.1, at the end of the section, summarizes the results observed according to the type of service.

During the test a wide variety of websites from all kinds of fields were tested. Information, leisure, shopping, sport or academic sites, as well as educational platforms for Norwegian universities such as Blackboard, Studentweb and Søknadsweb were tested and all of them gave satisfactory results. Also, all the 50 most visited sites in Norway according to Alexa's list were tested [Alea], as well as some of the most popular social networks (Facebook, WhatsApp, Instagram, Facebook, Twitter, Snapchat, LinkedIn...) and video on demand (vod) services (Netflix, HBO, Amazon Prime, Crunchyroll, Filmin, Twitch...). It should be noted that not all the websites tested are accessible via a single IPv6 connection, but the joint use of this with NAT64 and DNS64 allowed access without any shortcomings being observed.

Websites dedicated to offering online courses such as Coursera or edX, that offer resources to students, such as Google Scholar o ResearchGate, online tools such as Github, Slack, draw.io, Scribd, Spotify or the services offered by major companies dedicated to the sector such as Microsoft (Skype, Microsoft Store, Office 365, Mixer, Microsoft Teams...) or Google (Gmail, YouTube, Maps, Photos...) were found to work perfectly. The only exception to this group is Duo, the voice and video call service over the Internet owned by Google, which was not found to work.

Because of security reasons, the treatment of packages in banking systems is usually more thorough [EvB98], it was decided to try all the banking sites that were available (this is three, one of them being Norwegian, the DNB), to make a bank transfer being one of the accounts of the transaction the Norwegian one, and to make a purchase online. All the procedures worked perfectly. No problems were found when using different email services either.

The most used software by NTNU users were also tested and in general there were problems with the use of licenses. In the case of Matlab, it was necessary to verify a user license online during its installation, part of which was not possible unless it is returned to an IPv4 connection. After the installation, it was possible to work offline since it is not necessary to re-validate the license every time the program is initiated, but it is not possible to download extra utilities or the access to other online tools. Other programs like EndNote have already licensed the installation file so it did not have the above problems. Perhaps, if this practice were possible for more types of programs, it would be advisable to use it to avoid connection problems when activating the licenses. The use of license servers was a generally favorable case. It worked as an alternative method for activating Matlab (although it was still not possible to download additional add-ons for the lack of connection with MathWorks) and Maple, but strangely enough it was not possible for IBM SPSS, even though the NTNU's dedicated server has an IPv6 address. This last case is especially serious since it requires validation of its use every time the software is started, so it is not possible to use it in any way under these circumstances. It is believed that the reason why the activation of SPSS was not possible, as well as the connection problems mentioned above, is due to the fact that the clients of these software force communication through IPv4. This makes it impossible to make any kind of connection without having an IPv4 address, even though address translation methods are used later. Probably in the case of SPSS, the NTNU license server does not connect directly to the IBM server, but rather the connection is made through the user's computer, which would explain why it is not possible to verify the license even though the license server of NTNU has an IPv6 interface.

Within the applications dedicated to making calls over the Internet, mixed results were found. A good part of them like Skype, Zoom or Meets worked correctly, while others like Duo, Facebook Messenger or Discord were only possible through an IPv4 connection. However with these it was only the voice and video calls that didn't work, other services within the applications were working. TeamViewer was tested for both remote desktop and video call functionality within the application and everything worked fine.

In the field of video games, the main PC launchers such as Steam, GOG or uPlay were incompatible with the type of connection tested, although others such as the

Epic Games Launcher or Xbox for PC were executable. The online game of the latter could not be tested because of the NTNU firewall, where it was tested, however, and although this should be evaluated by the interested parties, it is not considered relevant here that online video games are limited in an academic environment. Withal, the impact could be greater if Uninett itself provides Internet access to university residences.

Overall, the experience of using NAT64 has proven to be positive. As for web browsing, no problems of any kind were found. The address translation performed by NAT64 and DNS64 works perfectly in view of what was observed and any website, regardless of the domain it is on, is accessible and functional. As for the services, the experience has been mixed but generally positive. It seems that most of these and the most important ones are prepared to work on an IPv6 network or that they do not present problems with the use of NAT64. There are a few exceptions to this but in general they are a minority and there are functional alternatives for all of them, so this should not be a problem. Furthermore, it is expected that during the next few years many of these services will be updated to support IPv6 natively, especially considering that if more situations like the one described here were to occur in the world, they could lose market share by depending solely on IPv4, this being a limiting factor.

A more problematic case is that of limitations in the specific software. Given the academic nature of Uninett's edge networks, it is more than likely that software that presents performance problems in this scenario cannot be easily replaced, due to the few alternatives available. As seen before, there are cases for which there are solutions, albeit somewhat uncomfortable ones, and for which the use of IPv6 + NAT64 does not represent a serious impediment to use. Unfortunately, there are other cases for which it is totally impossible to work under these conditions. As already mentioned, the reason for these problems is most likely due to the fact that the client of these programs does not have IPv6 support, and therefore finds it impossible to operate in an environment without IPv4. It is uncertain at this point how many of these programs will be updated to support IPv6 natively and when this would occur. In view of this, and the probable difficulty of replacing all software that presents problems with IPv6-compatible alternatives, it is recommended that a certain IPv4 infrastructure be maintained in the edge networks, at least in specific areas (such as laboratories) where access to this type of software can be ensured.

3.2 Analysis of performance

The following study was intended to be a performance analysis of a real router with NAT64 capabilities. The original idea was to first work in a closed simulated environment to understand the basics of NAT64 configuration, then to direct real

Table 3.1: NAT64 usage summary

Type of service	Outcome	Comments
Web sites	Positive	-
Vod services	Positive	-
Video call services	Mixed	Depends on the service, correct functioning in some and impossibility of making calls in others
Social networks	Positive	-
Bank services	Positive	-
Video games services	Overall negative	Most of the clients need an IPv4 connection in order to do online functions
Academic software	Overall negative	Several clients need an IPv4 connection in order to do online functions while, some others need it to initiate the software

traffic to the simulated environment from a real traffic generator, and finally to test with a real NAT64 router and traffic generator, both provided by Uninett.

The idea behind these tests was, in addition to understanding the configuration of a NAT64 device, to test its performance by directing in all cases a large bandwidth to the router and comparing data such as the processing speeds of these packets as well as the percentage of packet loss for cases where NAT64 translation is performed on the packet flow and for cases where it is simply redirected.

Unfortunately, due to the global pandemic conditions during the realization of this thesis, only the first part of all planned tests has been possible to perform and with limitations. On the one hand, the working conditions were reduced by limiting the available resources and the working environment, which meant deviations from the originally planned times. On the other hand, the situation required a lot of work from Uninett and its workers, limiting the support and resources that were possible to allocate to this project. All of this combined to not being able to access the machines on which the tests were planned to be performed, in addition, it made the work during the first phase of the test very difficult and the part performed was limited as it did not have a user license that allowed for a higher bandwidth to be processed in the simulated environment. It is for all these reasons that what is presented here is an unfinished work, whose scope is much smaller than that initially conceived, but which should nevertheless serve to understand the operation and performance of NAT64 in a very limited way. The following is a detail of the part of the work that could be done.

The simulation environment used was the GNS3 software, which allows the creation of objects based on images of real devices. These objects can be used

to simulate the operation of these devices and then interconnect them as desired, creating with some freedom different types of interfaces for them and being able to configure them as if they were the real device. Figure 3.2 shows the environment that was created for this test.

Figure 3.2: Environment layout

Two types of devices were emulated in this environment. The objects UbuntuDockerGuest 1 and 2 are emulating a basic version of the Ubuntu operating system. There are different images of similar objects with different version of the operating system or installed functions and many of them are offered for free on official sites. Here it was used the one available in Uninett, which is based on Ubuntu 16.04.2 and had all the basic functions that were needed for the test. The csr1000v object emulates the virtual Cloud Services Router 1000v (CSR 1000v) owned by Cisco systems in its 9.16.07.1 version. This router is compatible with Stateless and Stateful NAT64 and is capable of handling up to 10Gbps of bandwidth depending on its configuration [Sysc]. It was chosen both for its availability by Uninett and for its characteristics, which matched the needs of the test.

Due to the high computer load involved in emulating a router such as the one chosen, it was not possible to simulate the environment on a personal computer and instead had to be run on an internal Uninett server. The basic configuration parameters that were given to the router were 12 GB of RAM, 8 virtual CPU cores and two Intel Gigabit Ethernet interfaces.

A list and explanation of all the needed commands and configurations for this test can be found in the Appendix A, here is only presented the logic behind it, the basic layout and the results of it.

Each of the Ubuntu containers was configured with a different IP version, giving

the UbuntuDockerGuest-1 a static IPv4 address on the interface connected to the CSR1000v router and a static IPv6 address to the UbuntuDockerGuest-2. The router was given IP addresses belonging to the same network as those interfaces it was connected to. With each container having a different type of IP, it was initially impossible to establish a connection between them. Subsequently the router was configured with Stateless NAT64 to perform a static translation of an arbitrary IPv4 address from the same network to the IPv6 address of the UbuntuDockerGuest-2 and another one from an IPv6 address of the same network to the IPv4 address of the UbuntuDockerGuest-1. Figure 3.3 shows in a graphical way the distribution of IP addresses plus the translation table within the router.

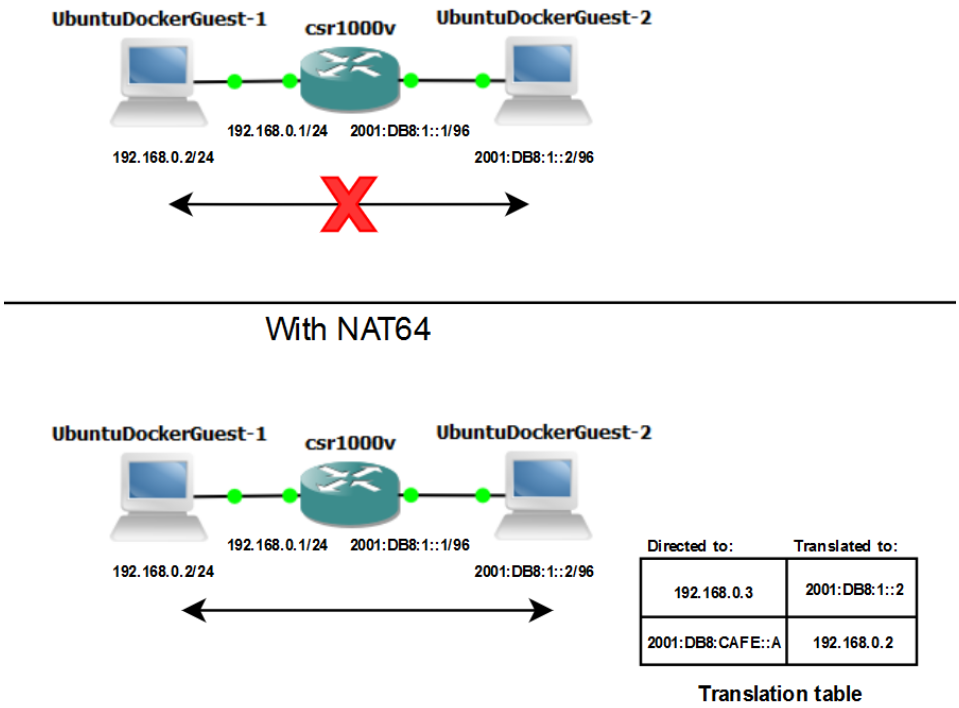


Figure 3.3: IP mapping and translation table

In a real scenario it would be advisable to perform Stateful NAT64 with dynamic translations and have a DNS64 server, because as explained earlier in this chapter, this would allow among other things a better use of IPv4 addresses and a more scalable configuration for large networks.

If, for example, UbuntuDockerGuest-1 would like to send data to UbuntuDockerGuest-2 it should use the destination address 192.168.0.3. The packets with this destination

address would be directed to the CSR 1000v router when it is configured as a gateway and there, they would be translated with the new destination address 2001:DB8:1::2 with which they would reach their destination, and 2001:DB8:CAFE::A as the new source address.

To test the connectivity in both directions the iPerf3 tool was used, leaving in each case one of the containers as a server listening to messages and the other as a client sending them to the first one. Figure 3.4 shows the results of connectivity in both directions, with very little difference between them and without any losses.

Figure 3.4: iPerf3 results with NAT64 enabled, from IPv4 to IPv6 domain on the left and from IPv6 to IPv4 on the right

Unfortunately, as mentioned at the beginning of the section, the conditions at the time of the work prevented the use of a user license that would allow a higher capacity for the CSR 1000v router. The version that could be used was capped at a maximum throughput of 1000 kbps. This was not contemplated at the time of designing and starting the test but due to the time limit it was decided to go ahead with these limitations even though they would allow a very limited view of the actual performance of the equipment.

Figure 3.5 shows an analogous test with the same objects but without the use of NAT64. There was a test first in which all the interfaces had an IPv4 address and another one in which all had an IPv6 address. The objective of it was to compare the throughput in all three different tests, specially to see how much delay can bring the use of NAT64 in a communication or, if it causes any noticeable variation on the packet loss rate. The interesting datum here is the throughput of the receiver, since it is the one limited by the router. Several tests of each kind were executed under the same circumstances but all with minimal (1 kbps at most) or any differences with the ones showed here.

The results seen here are a little surprising. It is found that, as expected, the

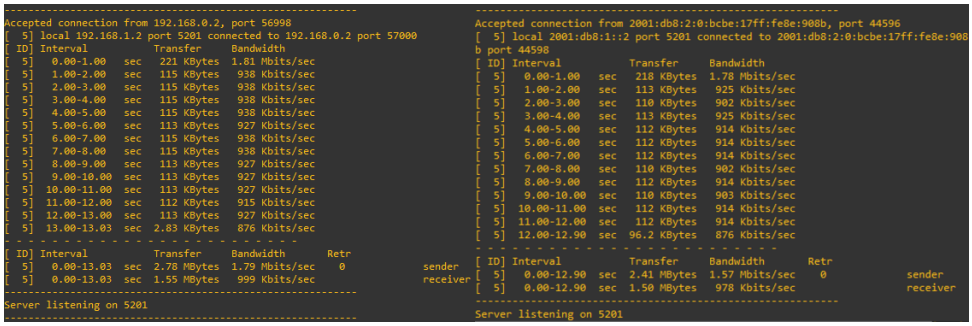


Figure 3.5: iPerf3 results with only IPv4 hosts (left) and only IPv6 host (right)

connection between IPv4 only devices reaches 999 kbps of throughput, basically staying right in the limit imposed by the license. The connection of IPv6 only devices stays a bit lower, at 978 kbps, which could be expected assuming that iPerf does not require much overhead and that both protocols can opt for the lightest possible header (20 bytes in IPv4 and 40 in IPv6).

Between these two results are those of the connections made with NAT64, with 983 kbps in the direction of the IPv4 domain to the IPv6 domain, and slightly higher, 987 kbps when the path travelled is the opposite. It seems that the translation from IPv6 to IPv4 is slightly faster than from IPv4 to IPv6. Perhaps one reason for this is that, since IPv6 headers are likely to be larger than IPv4 headers, it takes longer to write the new header data after translation than it does to remove or examine the previous header. It is surprising that better results are observed with NAT64 than with a pure IPv6 communication. It is possible that this is a small anomaly due to licensing restrictions and the virtual environment, however many tests were performed, all under the same conditions, and all of them were faster using NAT64. Another possible explanation is that GNS3 has a prefixed link extension (since no parameter was found that would allow this to be changed) and that it takes into consideration the time it takes for a packet to be transmitted through the link. In this way, IPv4 packets, since they probably have a smaller header, should be transmitted more quickly; and the configuration with NAT64 has two sections in which one of them transmits IPv4 packets and the other IPv6 while pure IPv6 only transmits IPv6 packets. Note that this is only a possible theory and that it is unknown why a higher throughput was observed with NAT64 than with IPv6 alone.

If it is assumed that the results seen here correspond to reality and that, despite the limitation imposed by the license, these numbers are scalable to higher bandwidths, then the results of the connection with NAT64 are approximately 1,6% slower than

with pure IPv4 and 0,5% faster than with pure IPv6 in relation to the connection from IPv4 to IPv6; and 1,2% slower than pure IPv4 and 0,9% faster than pure IPv6 in relation to connection from IPv6 to IPv4.

On a low scale like the one shown, these differences seem little significant, but if extrapolated to the throughput handled in many of Uninett's links, it can mean a difference of several hundred Mbps in each of the nodes connecting to neighboring networks (this is explored in the next Section 3.5).

However, it should be remembered that what is shown here was done in a virtual environment and also very limited, so it is likely that these numbers cannot be extrapolated directly to reality. It is therefore advisable to carry out a detailed study, apart from this, and with real machines, to ensure the difference in flow rate that the presence of NAT64 in points with high traffic may represent. It should then be assessed by the ISP whether the benefits of this technology will compensate for these losses.

3.3 Uninett network

As mentioned earlier, Uninett is the national ISP for universities and research institutions of Norway. Its activities include to provide of Internet access and to act as secretary, coordinator and counsellor to those institutions interconnected by its network. It was founded in 1993 although it started as a project organization in 1976. Among its achievements is to have been responsible to administrate the Internet domain .no since it was established in 1987 [Unia]. It places its headquarters at the Teknobyen Innovation Centre in Trondheim

Nowadays, Uninett has grown to be a state-owned company responsible of the National Research and Education Network (NREN) of Norway. This network spreads along the country and connects more than 150 Norwegian educational and research institutions, including the Norges Teknisk-Naturvitenskapelige Universitet (NTNU) and the university of Oslo among many others, with more than 300.000 users. Uninett's network is connected through NORDUnet to the worldwide Internet. NORDUnet is an international collaboration that interconnects the NRENs of the Nordic countries (Norway, Sweden, Finland, Denmark and Iceland) between them and to the Internet. Figure 3.6 shows a map of Uninett's network. On it, it can be appreciated the four exit points to NORDUnet, which are accessed through Oslo, Narvik and Finnmark.

Uninett offers a variety of tools in [Unic] that provide information of the state of its network and the traffic that traverse it.

It is possible to have a glimpse at the state of its edge networks by comparing

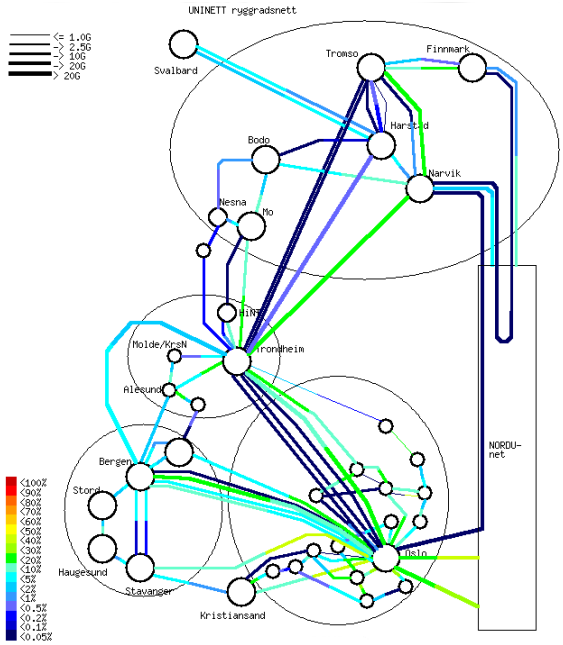


Figure 3.6: Uninett's network map

the number of IPv6 with IPv4 hosts in each one of the institutions. They provide this information in the form of graphs like the ones showed in the figure 3.7 and of all the institutions connected to Uninett

Not all the data presented shows realistic measures. Some of the institutions merged and then the data of them both is taken together. Some others have had a change of domain, that's why some domains stop reporting data at some time. Also, if the institution itself doesn't route IPv6 traffic, it won't count the IPv6 hosts (which in this case have to be dual stack) since they will be using their IPv4 addresses to communicate. A few hosts may appear however, if they also monitor the Uninett uplink, but much less than expected considering that all, or at least most, modern computers and smartphones are dual stack devices.

To calculate the number of IPv4 and IPv6 hosts, each of the institutions connected to Uninett have a monitor station that logs the contents of the IPv4 *Address Resolution Protocol* (ARP) and IPv6 *Neighbor Discovery* (ND) caches of each off the routers in the network. Then, the logs are matched with the *Media Access Control* (MAC) addresses to calculate the total number of hosts and the number of hosts using IPv6.

Table 3.2 shows the peak number of IPv6 hosts and peak number of total hosts

Figure 3.7: Samples of hosts data collected in [Unic]

during the lapse of two years, starting in February 2018, and the percentage that IPv6 hosts represents for several of the Uninett institutions. Those with an "*NR*" value in the table are the ones that show less than around the 10% of IPv6 hosts, number inferior than the one expected (see 2.3) and therefore it is assumed that those institutions either don't route IPv6 traffic and that the few hosts counted belong to Uninett network or only few of their routers report IPv6 usage, and therefore considered not reliable to the analysis as they do not correspond to reality. The data of the *NR* institutions have not been taken under consideration when calculating the total numbers and percentage for reliability reasons. Those that haven't reported any data in more than six months have been completely ignored.

In all institutions under consideration, both the number of IPv6 hosts and total hosts follow the same distribution, so the peak number in both of them happen at the same date and are therefore, considered reliable.

Half of the institutions from which it is possible to retrieve data are not routing IPv6 traffic. This tells that many of the Uninett's edge networks still need to be updated in order to fully support IPv6. Nevertheless, attending to the ones that are providing data about IPv6 hosts the results are positive. In most of the institutions more than 50% of the hosts have IPv6 addresses, with some others above the 70% or even 90% of adoption, being the most worrying scenario the Universitet i Bergen,

[1]In this case, a change in the domain name of the servers prevented data from being collected for a period of several months. However, once data began to be received from the new domain, the current data was still consistent with that from before the change was made

Table 3.2: Number of IPv6 hosts in Uninett institutions

Institutions	Peak number of IPv6 hosts	Peak number of hosts	Percentage of IPv6 hosts
Norges Teknisk-Naturvitenskapelige Universitet [1]	53.887	71.442	75,427%
Universitetet i Agder	12.787	20.417	62,629%
Universitetet i Bergen	16.559	34.825	47,549%
Oslomet	18.496	20.242	91,374%
Uninett	1.892	3.192	59,273%
Høgskolen i Sørøst-Norge	11.727	15.121	77,554%
Høgskolen i Ostfold	6.435	6.990	92,06%
Høgskolen i Innlandet	6.487	11.090	58,494%
Høgskolen I Molde	1.671	2.468	67,706%
Studentsamskipnaden I Trondheim	2.508	6.178	40,595%
Nord Universitetet (NR)	8	4.604	0.17%
Universitetet i Stavanger (NR)	9	1.427	0,63%
Kunsthøgskolen i Oslo (NR)	0	1.712	0%
Høgskolen i Volda (NR)	0	3.346	0%
Høgskolen på Vestlandet (NR)	761	6.741	11,28%
Arkitektur og Designhøgskolen i Oslo (NR)	218	2.264	9,62%
Norges Musikkhøgskole (NR)	0	1.171	0%
Sàmi Allaskuvla Samisk Høgskole (NR)	0	1.045	0%
Total (excluding NR)	**132.449**	**191.965**	**68,996%**

which, despite of the big number of hosts doesn't reach the 50% of IPv6 users.

Looking at the total numbers the perception is more positive. Almost a 69% of IPv6 support among the hosts at the institutions (ignoring those which data would interfere to a realistic measure, so the real number may vary). It is expected that this number will continue growing quickly as the old machines, not compatible with IPv6 will eventually fail and need a replacement. Probably it won't need many years for the 31% gap to almost completely disappear.

Uninett also offers different measurements performed on the links that make up its network. They are accessed by clicking on them from the overall map (figure 3.6) and they offer different interesting values such as traffic in Gbps, number of discards or percentage of IPv6 packets among others, being the last one the most interesting for this analysis. The data is presented in the form of a graph by hours, like the one in figure 3.8 and it is updated daily.

Table 3.3 shows the average IPv6 utilization of ten of the most important transited links within Uninett's network, ignoring those that go to NORDUnet which will be analysed in section 3.5. These links have been chosen either for being the ones that carry the most traffic or for connecting important emplacements within Norway. The average values in the table have been calculated taking under consideration the data

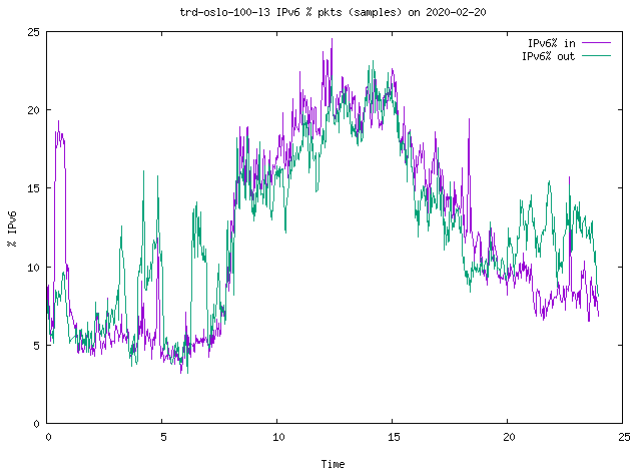


Figure 3.8: Sample of link data collected in [Unic]

of 4 weeks (from the 26/02/2020 until the 24/03/2020) to have a more consistent average. Some links were down during some specific days due to scheduled works, this was taken under consideration while calculating the average by not considering these days. The term "out" in the header refers to the traffic going from the link A to the link B while "in" refers to the traffic from link B to link A.

Table 3.3: Number of IPv6 hosts in Uninett institutions

Link A	**Link B**	**IPv6 out max (%)**	**IPv6 out min (%)**	**IPv6 in max (%)**	**IPv6 in min (%)**
Oslo	Trondheim	16,1642	6,175	17,7214	5,3321
Oslo	Stavanger	14,4035	0,7464	16,2464	0,9357
Oslo (Tullin)	Bergen	3,1695	0,3326	3,6565	0,6347
Oslo	Kristiansand	13,2392	1,0785	17,8321	1,3964
Trondheim	Ålesund	17,125	6,4357	4,8392	1,3035
Trondheim	Bergen	11,4428	0,5589	17,3928	0,7053
Trondheim	Narvik	55,6785	10,5714	57,0714	13,4642
Trondheim	Lillehammer	1,43	0,7292	1,1360	0,1639
Tromsø	Narvik	40,9107	8,8	41,2321	7,0892
Tromsø	Finnmark	1,1235	0,2367	0,4153	0,1253

Generally, in all the measures taken, the maximum peaks occurred in the hours with the highest traffic volume (often from 12:00 to 15:00) while the minimum peaks occurred in the hours with the lowest traffic (from 23:00 to 06:00 mainly).

It is worth noting the great difference in the use of IPv6 between some links and others. One of the reasons that may explain this is the lack of IPv6 in some of the academic institutions. Some of the lower results can be found in the link between Tromsø and Finnmark. This makes sense if it is compared with the data of the table 3.2, which shows how the Nord Universitet in Tromsø has a negligible amount of IPv6 hosts while the Universitet i Tromsø (which has installations in Finnmark) didn't show any data related with the amount of hosts and their kind, but after this results it is expected that the Universitet i Tromsø is likely to don't route IPv6 traffic or to have a very limited infrastructure.

Mostly, links coming from Oslo or Trondheim, which showed a high percentage of IPv6 compatible hosts, also show a high utilization of IPv6 within its traffic. There are a few cases in which the IPv6 utilization shown here may look not consistent with the number of available hosts for those institutions, like the link between Trondheim ad Bergen, which, looking at the percentage of IPv6 hosts at the NTNU and the Universitet i Bergen, it could be expected a much higher IPv6 utilization. A possible answer for this is that these links are not used exclusively to communicate between institutions, and they may carry a lot of traffic from other sources that affects to the overall measure. The mentioned link between Trondheim and Bergen often carries an average of 15 Gbps of traffic, which positions it as one of the most used links within the Uninett network. The low percentage of IPv6 use makes sense, since it probably carries a lot of IPv4 traffic from many institutions that do not support IPv6.

Considering the number of institutions that are likely not to route IPv6 traffic (considering the lack of data from their network), the results seen in these links are quite positive. Some of these links are among the ones that carry the most traffic within Uninett network and they show and IPv6 utilization above the 10% or 15% in many cases, while others like the links connecting Narvik with Tromsø and Trondheim showcase a very high percentage, probably due that many of the IPv6 traffic from the north part of Uninett network is routed through those links.

At this time, it seems that those academic institutions that can, they direct much of their traffic through IPv6. As more institutions update their network to support IPv6, this type of traffic will increase. None of the links in Uninett's core network show a high percentage of occupation (they are normally between the 40% and 50% at most). This can be seen by looking at the link colours and the legend at the bottom left of the figure 3.6. Therefore, it is not expected that a large increase in IPv6 traffic from the current state will result in the saturation of any of these links. In addition, it is expected that IPv4 traffic will decrease in the same proportion as IPv6 traffic increases.

Figure 3.9 shows the data available in APNIC [APN], already mentioned in

chapter 2. According to them, 21,66% of Uninett users are able to access to the Internet using IPv6. It is necessary to take into account the limited number of samples carried out to calculate such a quantity. However, these results are in line with what could be expected from the previous analyses. There is a much greater number of IPv6 users than those served (thanks to the dual stack they can maintain their connections through IPv4), considering the numbers calculated in 2.3, it can be expected that around 95-98% of user devices within Uninett edge networks would be able to connect to an IPv6 network. The difference between this number and the one calculated by APNIC or the numbers shown in 3.2, to a large extent, may be due to the limitations of edge networks. Without a dedicated infrastructure in many of the institutions, hosts are forced to make their connections over IPv4. On the other hand, as seen in Section 2.3, much of the current Internet is not yet IPv6 compliant, which means that IPv4 is often preferred over IPv6 for reaching the destination, even when an IPv6 infrastructure exists within the network.

ASN	AS Name	IPv6 Capable	IPv6 Preferred	Samples
AS205016	HERNLABS	99.74%	98.33%	10,161
AS31169	SOGNENETT-AS Providing fiber and wireless access	71.56%	70.14%	211
AS52157	VITNETT-AS	55.80%	55.80%	138
AS35132	ENIVEST-AS	53.58%	53.03%	726
AS42708	PORTLANE www.portlane.com	48.86%	0.00%	88
AS29492	EIDSIVA-ASN	47.46%	47.03%	1,854
AS2119	TELENOR-NEXTEL Telenor Norge AS	38.44%	38.21%	27,818
AS224	UNINETT UNINETT, The Norwegian University & Research Network	21.66%	21.51%	637
AS2116	ASN-CATCHCOM	7.57%	7.54%	6,117
AS8542	BKK-DIGITEK-AS8542 Norway	3.90%	3.90%	205

Figure 3.9: Uninett data available in [APN]

3.4 Edge networks

Despite the unfinished state of transition of the edge networks, Uninett's core network is already fully compatible with IPv6. The edge networks are in a continued state of development which eventually will lead to have all of them fully compatible with IPv6. Until that time comes, it is of interest of Uninett to know if it is feasible to stop supporting the IPv4 part of its core network and rely only on the IPv6 part. Whether if this comes after the edge networks have fully turn into completely operative IPv6 networks or if they use different transition technologies until that time comes is beyond the scope of this document. What is of Uninett's interest is to know if they can rely on transition technologies so the communication in the core part can be driven through IPv6 and remain IPv6 or turn to IPv4 depending of the necessity when they go through one of their exit points to the Internet.

Even with this, it is important for Uninett to calculate the increase in IPv6 traffic that this would mean for its network, as there may be network nodes that are not prepared to handle the increase of IPv6 traffic, specially those that acts as gateways

for the edge networks. In that case, a traffic redirection or update of some of these nodes may be necessary to handle the new situation.

The following is a traffic and user study focusing on the NTNU and the Universitetet i Agder (UiA) networks, to get some idea of the changes it would entail. This study was possible thanks to the collaboration of these institutions and the data collected by the Network Administration Visualized (NAV) software and facilitated by Uninett. However, this study is limited to only two institutions out of all those that make up the Uninett network and even the data used to calculate the data presented here is limited.

Figure 3.10 shows the number of IPv4 and IPv6 addresses detected at NTNU's subnets during the lapse of two weeks. The number of IPv6 addresses is much higher than the number of IPv4 addresses every day. However, this contrasts when comparing with the number of MAC addresses, which is higher for IPv4 subnets, and matches the results seen in table 3.2 (note that the numbers shown in that table are peak numbers in the lapse of two years, so they are not exactly the same as shown here). However, the gap between MAC addresses (and therefore devices) using IPv4 and IPv6 is much smaller than the gap between the number of IPv6 and IPv4 addresses. This indicates that IP address reuse is much higher on IPv4 networks than on IPv6 networks, since fewer IPv6 users have used many more addresses than those used by a larger number of IPv4 users.

Taking the numbers of the 05/03/2020 (fig 3.10) for instance. On it, it was counted a total amount of 68.012 different MAC addresses that were connected to an IPv4 subnet against the 48.840 that connected to an IPv6 one. So, despite what was shown in table 3.2, it is actually higher the amount of equipment using IPv4 than IPv6. On that day in particular it was a 39,45% higher. Without having a list of all the MAC addresses connected to all the subnetworks (which was not given nor asked for privacy concerns) it is not possible to know how many of these machines were connected only to the IPv6 network, only to the IPv4 one or used both of them. However, it is possible to compare with the data in table 3.2, which has already matched the IP addresses with the MAC addresses internally to avoid taking the same machine into account several times. By doing that comparison it is possible see that the number of IPv4 connected machines is very close to the peak amount number of total hosts at the NTNU, and therefore it is possible to assume that a large majority of IPv6 machines are in fact dual stack devices using both IPv4 and IPv6 networks.

However, the difference between the number of IP addresses and MAC addresses is surprising, especially in the case of the IPv6 network. Figure 3.11 compares these numbers for some arbitrary days. In it, it can be seen that, while the number of IP

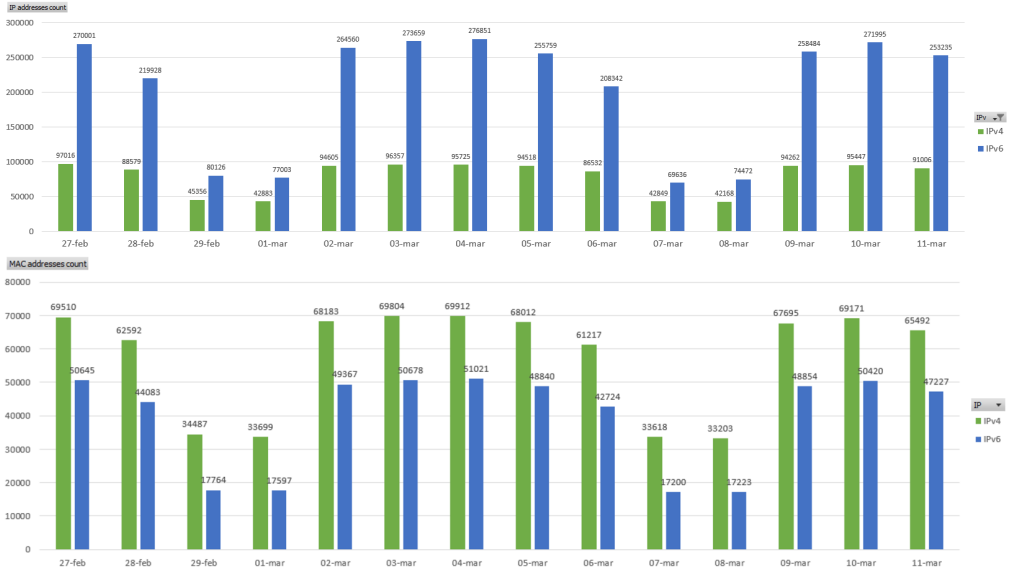


Figure 3.10: Number of IP addresses (above) and MAC addresses (below) connected to NTNU's IPv4 and IPv6 network

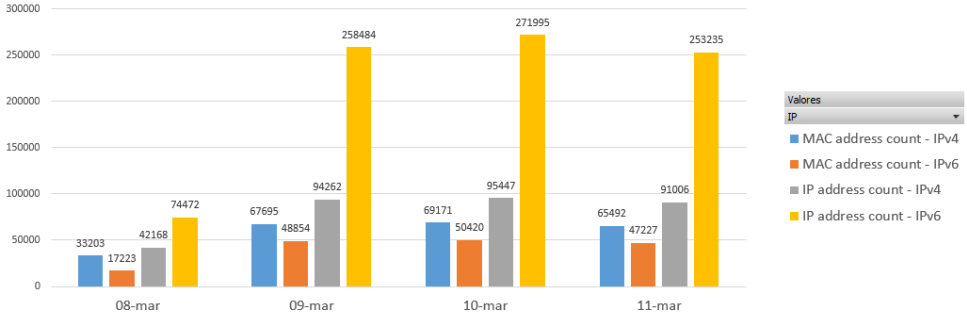


Figure 3.11: Comparison between the number of MAC addresses and IP addresses in the IPv4 and IPv6 domains at NTNU

addresses is slightly higher than the number of MAC addresses for the IPv4 network, in the case of IPv6 the IP addresses are up to five times higher than the MAC addresses. This, once again, highlights the huge difference of size in the address space from one technology to the other. The IPv4 space is, at present, extremely limited, the use of addresses has to be highly optimized, as well as the use of technologies that allow them to be more widely reused. IPv6 on the other hand, may be seen as

almost limitless. Subnets have a much larger address space, which means that when a device connects to a subnet, it is unlikely that the assigned address has already been used by another of the devices connected to the same subnet or by the same machine before, and therefore causing that large difference in number between some addresses and others.

Similar conclusions can be drawn after analysing the data from users of the UiA (figures 3.12 and 3.13). This time the leap between IPv4 and IP6 users is greater, being about 2,5 times higher the number of IPv4 users based on the collected MAC addresses while at the NTNU it was just a 36% higher. This difference is probably due to the use of a less prepared IPv6 infrastructure (perhaps it is not available for the whole campus) rather than to the fact that the number of IPv6-ready users is much lower in one university than in the other. However, without a further study of its network it is not possible to conclude.

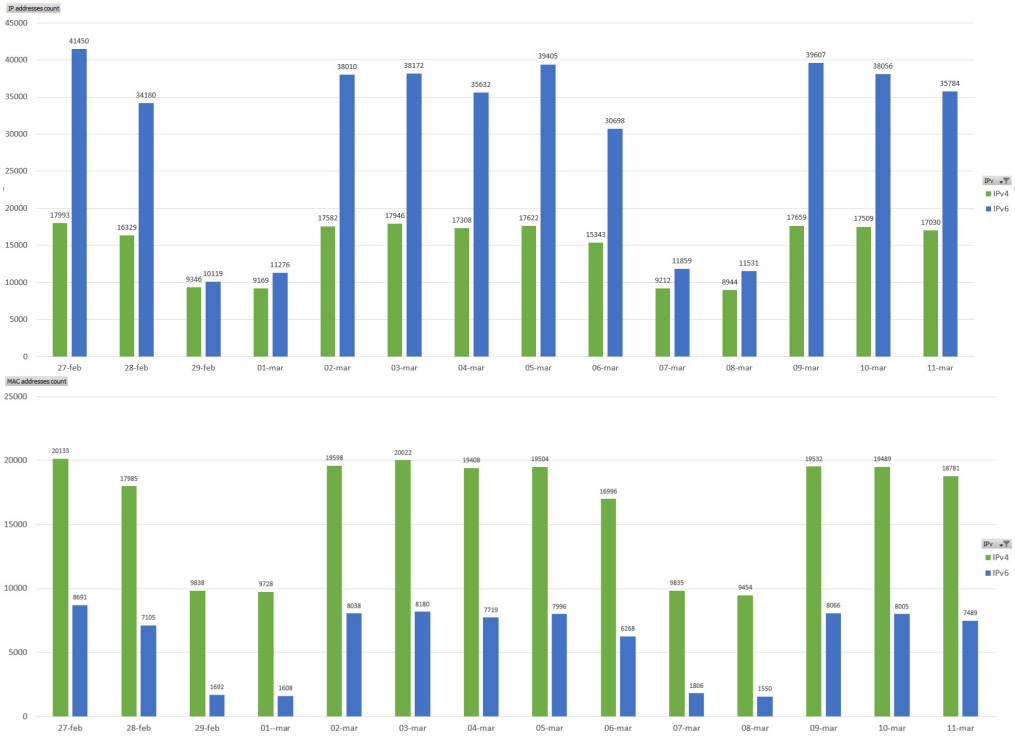


Figure 3.12: Number of IP addresses (above) and MAC addresses (below) connected to UiA's IPv4 and IPv6 network

Again, the number of IPv4 users is almost equal to the total number of users that is shown in table 3.2, and therefore a large percentage of IPv6 users are expected to

be dual stack users. The differences between the numbers shown here and those in the table above are due both to the difference in dates when the measurements were taken and to the fact that the number shown in the table is the peak number. Once again, and despite the lower proportion of IPv6 users, the number of IPv6 addresses used far exceeds that of IPv4, being slightly more than the double of it.

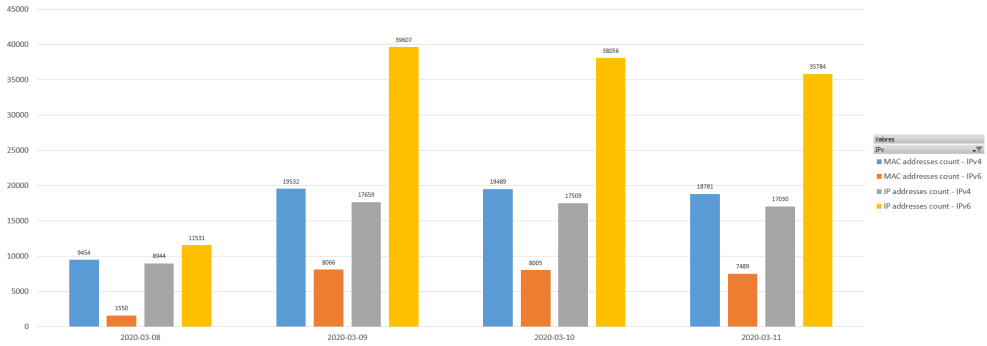


Figure 3.13: Comparison between the number of MAC addresses and IP addresses in the IPv4 and IPv6 domains at UiA

IPv6 does not only bring improvements in the address space but it also simplifies the scheme of the network. NTNU's network consists of 1142 IPv4 LANs and 490 IPv6 LANs while UiA is conformed by 192 IPv4 LANs and 67 IPv6 LANs, having most of the last (all of them for UiA) an address space greater than the entire IPv4 address space. In part, this reduction may be due to the fact that the IPv6 network is less developed than IPv4 in both cases. But it is also due to the freedom that comes with having more address space. It is likely that some of the IPv4 LANs share the same physical space but are subdivided into a greater number of networks in order to favour a better distribution of addresses, a problem that does not exist with IPv6. This big reduction (less than the half) will bring a simplification in the network administration and probably in its security when the transition from IPv4 to IPv6 is complete.

It was also possible to analyse a data flow of the Uninett network from the 11/03/2020 using the SiLK analysis suite. SiLK is a collection of command-line tools for processing Flow records created by the routers. These commands allow to filter, sort and count records and then export them in an easy to read format [CER]. Appendix B shows in a simplified way an example of the scripts used to obtain the following data, once again from the NTNU and UiA.

These scripts filter the traffic of the NTNU and UiA institutions by using all the IPv4 and IPv6 LAN addresses and divide the traffic into incoming and outgoing by

using some predefined types. The data flow belongs to a working Wednesday, so it is assumed that most working days follow similar statistics. Taking into account that during data filtering, the range of IPs used was denoted as source or destination addresses, these statistics should not be affected by other traffic not belonging to the users of these institutions but which is routed through nodes common to their networks, as shown in figure 3.6. The results are shown in the figure 3.14 and 3.15. Data such as these can help understand the extent to which IPv6 is preferred over IPv4 when a user can choose both, or the extent to which IPv6 traffic would increase if all network traffic from these institutions were routed using that protocol.

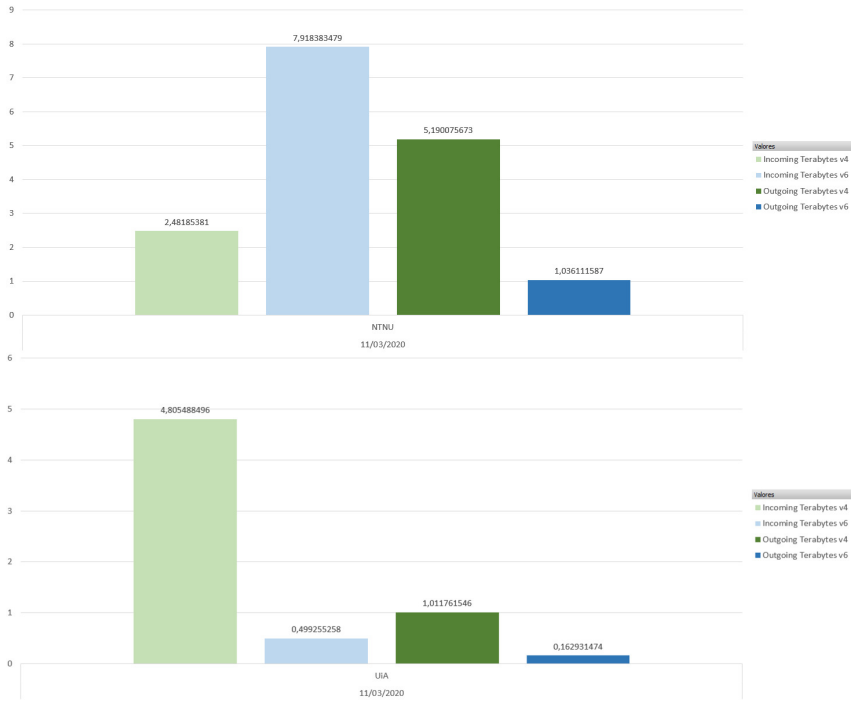


Figure 3.14: Incoming and outgoing traffic of NTNU and UiA shown in terabytes, collected over 24 hours

The NTNU for example shows a large volume of downstream IPv6 data compare with IPv4, but the opposite is true for upstream traffic. This, although uncertain, could be due to the fact that many websites involving a large volume of downstream traffic such as Youtube or Netflix are fully IPv6-compatible. As seen before, about 75% of NTNU users are IPv6 compatible. It is therefore likely that this large volume of IPv6 download data is due to the use of this type of websites by the majority of users, while the remaining IPv4 volume could be due to the use of similar websites by the remaining 25% of users as well as a large number of websites that are not yet

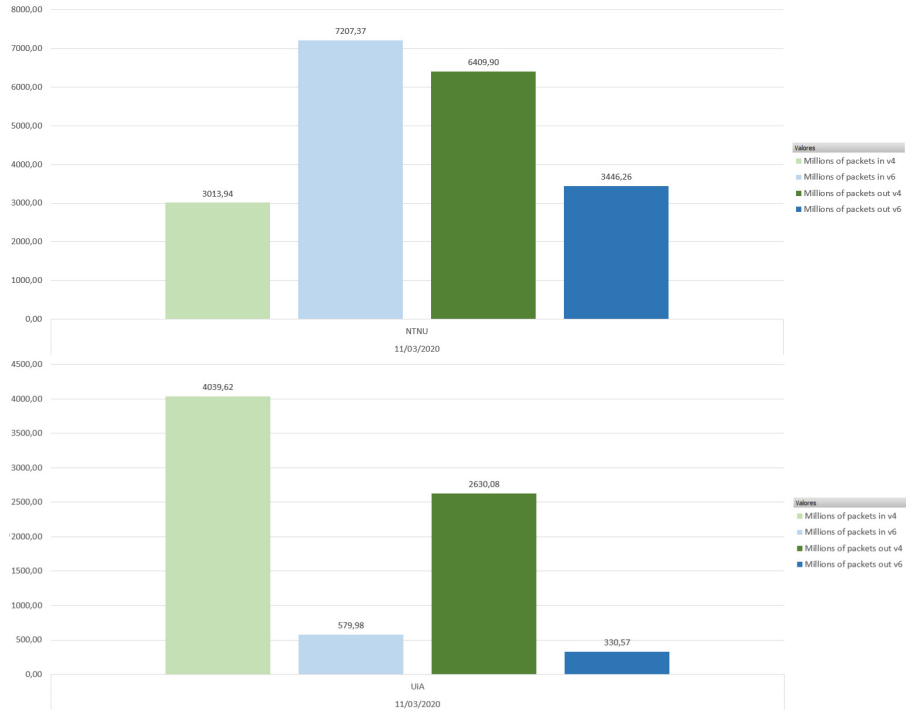


Figure 3.15: Incoming and outgoing traffic of NTNU and UiA shown in millions of packets, collected over 24 hours

accessible via IPv6. However, it is positive to see how such a large volume of traffic is conducted over IPv6 at present. Specifically, for the data shown from 11/03/2020, downstream IPv6 traffic represents 76,13% of the total while upstream IPv6 traffic represents 16,64%. With respect to the large difference between IPv4 and IPv6 traffic in the upstream, it is unlikely that this is all due to the upstream traffic of clients communicating with servers, but rather that the NTNU is likely to be hosting a number of services (such as web sites) that are only accessible via IPv4. Consistently, the difference in the number of packages of both technologies, both upstream and downstream, is similar to that of traffic volume.

On the other hand, and as expected from the reduced presence of IPv6 users, the UiA shows a much lower volume of traffic conducted over IPv6. Specifically and for that day, downstream and upstream IPv6 traffic represent 9,41% and 13,87% of the total, respectively. A similar allocation between IPv4 and IPv6 traffic is expected for those institutions with a similar percentage of IPv6 users. Looking at the data in the table 3.2 it could be expected for Universitetet i Bergen, Studentsamskipnaden I Trondheim or Høgskolen i Innlandet to have a similar behaviour that the one seen at

UiA, this is those where half or less of the total users are IPv6 compatible.

Fortunately, some of the institutions with the largest number of users are also those with a greater presence of IPv6. At Oslomet, Høgskolen i Sørøst-Norge or Høgskolen i Ostfold for instance, it is to be expected that the allocation will be more similar to that seen with the NTNU. It is therefore foreseeable that the increase in IPv6 traffic that would occur if Uninett's IPv4 network were to be shut down would be mixed. Some of the institutions that generate the most traffic already carry a large part of it over IPv6, so in their case the increase would not be very large. On the other hand, for other institutions that appear to be less prepared internally for IPv6, this increase in traffic would be considerably greater.

As mentioned before, it is important for Uninett to carry out a more in-depth study than the one presented here on the incoming and outgoing traffic of these institutions in order to take the necessary measures, if any, to prepare the core network. There are institutions for which it has not been possible to get an idea of the IPv6 traffic they handle or the number of devices compatible with the protocol, and others for which it has been only possible to compare them with others with similar IPv4/IPv6 users ratio.

3.5 Core network

Stopping or reducing dependence on IPv4 is something that every ISP should look for, as it facilitates any development/change on the network, as it only needs to be compatible with one of the technologies. The reduced complexity can easily mean an improvement on security, especially with the IPv6 improvements. It also translates to an economic benefit, reducing the number of required machines and/or their complexity and reducing drastically the number of IPv4 addresses needed by the ISP, which could sell the remaining ones. Besides the operational simplification by reducing the number of IP addresses (the IPv4 ones) from the network.

As mentioned before, Uninett's network has four different exit points to the Internet, all of them connected to the NORDUnet network, two are in Oslo, and the other two are in Narvik and Finnmark, they can be appreciated in the figure 3.6. These points could be the bottle neck if they want to rely only on their IPv6 core network, as these points would have to handle all the outgoing traffic (as they currently do) with the difference that they must also take care of translating the communication from IPv6 to IPv4 whenever this is necessary. As seen in previous sections, many communications need to be carried out over IPv4 today, whether due to network deficiencies, services, etc, it is therefore to be expected that, although these can be carried over IPv6 within the Uninett network, they will subsequently need to be translated into IPv4 in order to be able to reach their destination. If this

were the case, these translations would be carried out at the borders of the Uninett network, at one of the four exit points. Also, for those translations and after the results seen in the previous chapter, NAT64, together with DNS64 is the chosen technology to carry them out.

It is important, however, to know the amount of traffic handled at these nodes in each direction, as well as the percentage of IPv6 traffic from this, since this will affect the solution to be proposed. The table 3.4, shows an average of the measurements taken at the four links connecting the Uninett and NORDUnet networks over the course of one week (13/03/2020 - 19/03/2020). The measures were obtained from [Unic] and they show the maximum and minimum peaks in the total amount of traffic that is handled by the links and the maximum and minimum percentage of IPv6 of that traffic. Again, "*out*" refers to the outgoing traffic from link A to link B and "*in*" to the incoming traffic. In link A, the name in parentheses refers to the name of the router by Uninett. Some measures shown values at the order of 10^{-6} and are therefore approximate to 0 and considered negligible.

Table 3.4: Measures in Uninett's exit points

Link A	Link B	Traffic out max (Gbps)	Traffic out min (Gbps)	Traffic in max (Gbps)	Traffic in min (Gbps)	IPv6 out max (%)	IPv6 out min (%)	IPv6 in max (%)	IPv6 in min (%)
Oslo (Tullin-gw1)	NORDUnet	8,3571	1,5	16,8571	6,2142	9,0285	2,4857	0,4428	0,1285
Oslo (Oslo-gw1)	NORDUnet	12,4285	3,4285	22,4285	4,7857	14,5714	5,5714	17,8571	4
Narvik (Narvik-gw2)	NORDUnet	1,4928	0,4785	2,3	0,4428	≈0%	≈0%	≈0%	≈0%
Finnmark (Utsjok-gw1)	NORDUnet	0,1328	0,01	0,9971	0,1357	≈0%	≈0%	≈0%	≈0%

The first impression after looking at the data is that most of the outgoing traffic on the Uninett network comes through the links from Oslo. This includes almost all IPv6 traffic (in practice, all of it). Finnmark and especially Narvik also handle a large amount of traffic, but since this is practically all IPv4, it is less interesting when looking at the current capabilities of the network.

The two Oslo links handle more incoming than outgoing traffic, about twice as much as the outgoing, which makes sense considering that users usually use much more traffic on the downstream that in the upstream and that most of Uninett traffic comes from regular users. Of all this traffic, outgoing IPv6 traffic is distributed unequally between the two links, which is about 2.5 times greater in the case of Oslo-gw1. Incoming IPv6 traffic is also unbalanced, with 50 times more incoming IPv6 traffic in Oslo-gw1 than in Tullin-gw1. All of this assuming that the data obtained during the week in which they were recorded is representative of normal network usage.

This asymmetric traffic distribution may be due both to the capacities of the

nodes studied and to the capacity of their links or nearby nodes. It is therefore recommended that the network in the future can continue to behave in the same way. A change when trying to distribute more traffic by one of these nodes, although the node in question would be able to do it, could saturate to nearby nodes, what would affect negatively to the global functioning of the network.

In general, during this chapter, the operating principle of NAT64 has been observed, as well as certain limitations that it use implies. However, despite these limitations, it is still seen as a promising technology, in that these limitations are either not critical or an IPv4 configuration in the client environment could solve them, as explained above. As far as the Uninett network is concerned, it is too vast to be able to carry out a detailed study in the time frame of this project. It is a network whose core part extends over a whole country and has numerous edge networks. However, the work seen here, although it does not allow for the design of a solution that is totally adjusted to the Uninett scenario, where the benefits of the work could be maximum; it does allow for a general understanding of the type of connections that are made as well as the volume of traffic and distribution between IPv4 and IPv6 of the same. All this data is very valuable in order to design a solution, which is presented in a more general way than desired, allowing, in the event that the stakeholder shows interest in it, to carry out more studies that allow it to be adapted to the needs of each case. This is shown in the following chapter.

Chapter 4
Results and discussion

On 1.2 the following knowledge question was formulated:

"Is it feasible for a nation wide ISP to move away from IPv4 and run an IPv6 core network only?"

and the simplest answer possible for it is: it can be.

Many transition technologies have been developed over the years, many of them being evolutions of the former and solving some of their cons. Different techniques have emerged to approach this transition such as tunnelling or package translation and, nowadays, it is considered that these technologies are sufficiently developed to be able to depend on them on a large scale.

Of course these have their shortcomings that have been observed in the Chapters 2 and 3, and it is therefore up to the stakeholders to assess to what extent these shortcomings have a strong impact on them.

IPv6 has also been deployed since 1994 and it is increasingly rare to see protocols or operating systems that do not support it. Most of the large technology companies provide a large part of their services through IPv6 and the number of IPv6 compatible end devices continues to grow, it is expected that the percentage of user-side end-devices compatible with IPv6 exceeds the 90% as seen in section 2.3.

However the gap between IPv6 and IPv4 still exists and is of considerable size. Much of the Internet traffic today is still carried over IPv4 and many medium and small networks are not yet IPv6 compliant, making it necessary to continue using IPv4 to some extent so as not to limit the scope of networks. It is therefore considered that it is not yet possible, or at least not advisable, for an ISP to depend exclusively on IPv6, due to all the limitations that this would bring to its users. However, it is also considered that through appropriate transition technologies that allow interaction with other IPv4 networks, the presence of IPv4 in an ISP network could be greatly

reduced, disappearing even in certain parts, with all the advantages that this could entail.

The other knowledge question formulated in 1.2 was:

"What combination of relevant IPv6 transition technologies can be used to handle WAN-traffic on a nation wide scale?"

As mentioned several times in different parts of the project, the global health crisis of COVID-19 has negatively impacted the performance of the work. Several planned studies have had to be simplified and reduced in scope due to working conditions and unexpected resource constraints. Various stakeholders in the work have been affected and forced to reallocate resources and time, thus reducing what was originally intended for the work. In general, this has affected the final model, being less specific than desired, and above all, reducing the scope of testing related to it. It is therefore important to emphasize once again that the model presented should not be taken as final but as a guide, whose work should be taken up in the future if it is seen to have potential.

However, the following sections will try to answer the previous knowledge question for the case of Uninett and could be extrapolated to other stakeholders. It is of course left to the stakeholder to assess, in addition to the potential negative impact this might have, whether the benefits of this network model would outweigh the necessary investment.

4.1 Results

Knowing that Uninett's core network is already fully IPv6 compatible, and with a view to relying exclusively on it and being able to "turn off" the core IPv4 part, there are several important factors to consider.

First of all, from a structural point of view there are two types of points in Uninett network that must be considered, those where the core network connects to neighboring networks, and those where the core network connects to the external networks, as well as the state of these in relation to the transition. Both points are fundamental in order to design future scenarios for Uninett and both need different solutions.

In addition, and as has been said, due to the incompatibility of IPv4 and IPv6 by design, it is necessary to assess the impact that the limitations that these incompatibilities could produce would have on users and the network.

During the previous Chapter 3, these impacts and the points mentioned above

have been studied. However, these are limited studies that should be reviewed in greater depth in order to make some type of decision. Nevertheless, the data observed in them allows to suggest certain scenarios which are based on the assumption that these data are actually extrapolated to reality.

With what has been seen in Section 3.2, it can be assumed that the use of technologies such as NAT64, or very similar technologies, should not have a significant impact on the speed or quality of communications. In addition, due to the low percentage of use of the Uninett network links (the most loaded links are between 30-40% of their maximum capacity according to Uninett data (figure 3.6)), it can be deduced that the nodes that connect these links are probably not working at full power or, if not, are easily upgradable. In this way, if the reduction in speed that the use of NAT64 may imply is considered unacceptable, these nodes could be upgraded to alleviate it. In any case, in view of the data seen in 3.2, it is not believed that this is a significant worsening or that it will limit the overall performance of the network.

In relation to the experience of use with the above-mentioned technology, it is more complicated to determine its impact. As seen in Section 3.1, the use of NAT64 was generally positive, however certain software was found to be incompatible with the use of the technology. The most serious problems found were generally due to the licensing of the use of certain dedicated software and it is believed that it was caused due to the software client forcing IPv4 connections. With some of them, despite being initially a problem, solutions were found (such as the case of Mathlab with the license servers, or Endnote with the licensed installation file). It is vitally important to study the possibility of applying these solutions to all academic software that involves this type of problems, or, failing that, to look for compatible alternatives. Given the nature of Uninett's external networks (mostly academic or research) it is likely that the use of this type of software is not something optional or easily replaceable. With regard to the other compatibility problems that were found, none of them were a serious handicap since there were compatible alternatives, such as the case of video call services. However, it is especially important to stress the limitations of the study in this particular section since there may be many protocols or types of connections that have not been taken into account and that could be greatly affected if dependency on IPv4 is cut.

With respect to the points that connect the Uninett network with other neighbouring ones, there are only four of them that, as commented in the Section 3.5, connect to the NORDUnet network; from Oslo, Narvik and Finnmark. These four points should continue to work as before, i.e. they should be able to direct all incoming and outgoing traffic in a similar proportion as they do now. The internal configuration of the NORDUnet network is unknown but it is assumed that the nodes connected to these four points are fully IPv6 compatible, although taking

into account the near-zero IPv6 traffic observed at two of these points (Narvik and Finnmark, table 3.4), this should be studied further. The need to distribute traffic in the same way as up to now comes from ensuring that intermediate nodes in the core network do not become saturated or a bottleneck for the network. Transitioning to a single IPv6 core network would involve all existing IPv4 traffic being converted to IPv6 traffic, but this should not result in a significant increase in the total amount of traffic. Perhaps a small increase due to the size of the headers, but nothing that is considered significant. In addition, simplifying the processing of IPv6 packets with respect to IPv4 (e.g. only senders fragment the packets), could mitigate this difference. Currently the nodes at these points do not have the capabilities to perform these translations, so they should be upgraded or changed.

At the time of writing, external networks are not ready for the full transition to IPv6. There is much heterogeneity in the state of IPv6 adoption, and while there are some that already direct much of their traffic over IPv6 there are many others that do not. Two non-exclusive options should be considered here, as one could apply to certain institutions and the other to others. In the event that a decision is made to dispense with an IPv4 core network, in order to ensure that these external networks do not remain isolated from the rest of the network, they will have to either match the bet by strongly developing their IPv6 network in order to depend exclusively on it, or depend on the use of some transition technologies that allows an IPv4 island to have connectivity with a large IPv6 network. The case of each one of Uninett's affiliated institutions should be carefully studied to see which option to choose, but it is likely that for those that already have a strong IPv6 infrastructure the first option would be better; while for those whose IPv6 network is extremely limited, it would be better to upgrade only the points that connect it to the core network while allowing time for their IPv6 network to develop naturally.

Nevertheless, there are two serious problems in limiting external networks to IPv6 only, and that is those end devices that are not compatible with the technologies and the existing limitations of no longer relying on IPv4, as discussed above. In table 3.2 it can be seen that most of the equipment is compatible with the protocol (note that, as mentioned on section 3.3, the numbers on that table are very likely to show a very pessimistic view of the network due to limitations on IPv6 networks, in reality they should be much higher); and in section 2.3 it was made an approximate calculation of the percentage of user-side devices that should be IPv6 compatible. However, there is still a niche of users that would be isolated if the IPv4 network is eliminated. In the case of equipment belonging to academic institutions, it may be possible to update it at the same time as the network so that there are no devices left that are not IPv6 compatible. However, it is likely that there are users and students with old, non-compatible devices who cannot be forced to change their equipment in any way. In addition, it is possible that there is software that is incompatible with IPv6 at the

moment and that may be of vital importance in some fields and irreplaceable.

With this in mind and in relation to the external networks, only the use of transition technologies is recommended, and explored here, to ensure that no user of the network is left without connection and that no critical functions that may be required by any user are rendered impossible. This would make possible to continue expanding their IPv6 networks in the background and reduce dependence on IPv4. The number of IPv4 subnets and addresses could be reduced and used mainly in laboratories and working environments, to ensure that there are means to overcome the limitations that exclusive use of IPv6 may have for certain users.

Below, and under all these premises, a scenario is presented as a suggestion to carry out the transition to IPv6 only for Uninett's core network. It is recalled that although the Uninett case has been worked on throughout the thesis, the following model can be carried out by ISPs with similar overall configuration and traffic distributions. Also, for privacy reasons, CAPEX and OPEX values for the current Uninett nodes could not be accessed.

4.2 Proposed scenario

4.2.1 Core network exit points

To ensure connectivity with external IPv4 networks, the use of NAT64 is suggested at the four network exit points of Uninett, Oslo, Finnmark and Narvik. In these points, maximum outgoing traffic values of around 8.5, 12.5, 1.5 and 0.2 Gbps have been observed for Tullin-gw1, Oslo-gw1, Narvik-gw2 and Utsjok-gw1 respectively, and 17, 22.5, 2.3 and 1 Gbps for incoming traffic. At the moment, it can be seen that the Narvik-gw2 and Utsjok-gw2 machines are working far below their maximum capacities. This may be due to future network expansion projects in that area but it is not known if there is a reason for this. A cheaper possibility that should be considered by the stakeholders is to apply NAT64 only to part of the exit points and distribute the IPv6 traffic to the remaining. However, this possibility should be studied carefully since such a large traffic redistribution could saturate some network nodes and it is therefore not considered here.

The NAT64 configuration at each of the points can follow the one at figure 4.1, with a single NAT64 node connecting to the dual stack router which acts as exit point for Uninett to neighbouring IPv4 and IPv6 networks. The representation of the neighboring networks is simplified in the figure for better visualization. Uninett's neighboring network is NORDUnet which is understood to be at least largely dual stack, as is the case with many large networks, the boundary between IPv6 and IPv4 Internet is not as defined in reality. As already mentioned in section 2.7.4, the

presence of DNS64 servers by Uninett is necessary so that they can convert A records into AAAA when necessary to reach IPv4 destinations. Connecting the NAT64 node to the network exit point allows it to handle IPv4 and IPv6 traffic in the same way as before, thus simplifying the necessary configurations. In addition, this allows any slowdowns that may be caused by the NAT64 translation to affect only the originally IPv4 traffic and not to all of it, which would not be the case if the NAT64 node was directly connected to the neighbouring network.

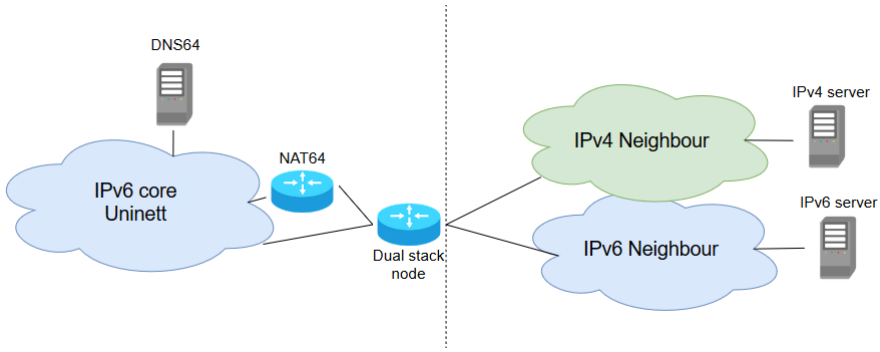


Figure 4.1: Core network proposed model of exit point configuration

NAT64 nodes should be able to handle at least the amount of IPv4 traffic currently handled by the exit points. It is expected that with the increased IPv6 focus of the core network, this originally IPv4 traffic will be reduced, especially if the edge networks are able to reduce the presence of IPv4 in their networks. However, to ensure the correct operation of the network in the coming years, it is recommended that these nodes be able to handle more traffic, since over time the traffic tends to grow and it is still uncertain to what extent the percentage of IPv4 traffic could decrease.

In view of the data in the table 3.4 it can be calculated that Tullin-gw1, Oslo-gw1, Narvik-gw2 and Utsjok-gw1 handle a current average of 16.8, 18,4, 2.3 and 1 Gbps of incoming IPv4 traffic and 7.6, 10.6, 1.5 and 0.13 Gbps of outgoing IPv4 traffic, respectively, and as maximum values.

With that in mind, major vendors in the market offer several solutions with NAT64 capabilities, Cisco's ASR 1000 series [Sysa] or Juniper's SRX [Netb] for instance. Of these, several configurations are possible to suit the requirements of each point. In this case, nodes with a capacity of at least 20 and 5 Gbps for incoming and outgoing traffic, respectively for the case of Tullin-gw1; 25 and 12 Gbps for Oslo-gw1; 5 and 5 Gbps for Narvik-gw2; and 3 and 1 gbps for Utsjok-gw1 respectively. However, since these manufacturers do not offer public prices it is not possible to make an approximate calculation of the total cost. Anyway, when suggesting these models

no exhaustive market research has been done. What is suggested here is merely indicative and there may be better options from other vendors.

4.2.2 Customer network entry points

With respect to connectivity to Uninett's customer networks, it is assumed here, based on the data available in [Unib], that there are currently 19 external networks connected to the Uninett core network. As mentioned in Section 3.3, this number may vary as certain sites combine with each other and it is unknown how this affects their networks and the way they are connected to Uninett core network, or there may be others that are not reporting data to [Unib]. In any case, the number considered here is that of all of them that in [Unib] continue to report data at present and, in case it is a different number, it is easily scalable. In addition, these 19 networks here will be treated independently. It is possible that some of them share, or may share access points which could reduce costs, but here it is assumed that this is not the case and that each of them needs a solution independent from the others.

As mentioned above, it is considered that all these external networks should retain their IPv4 networks, at least to some extent, until it can be ensured that all their users have IPv6 support. It should be possible to route as much IPv6 traffic as possible in order to accelerate this transition and to be able to reduce the number of IPv4 addresses that these networks need, while IPv4 traffic should be converted before reaching the core network in order to be able to transit that IPv6 network. NAT64, which was the focus technology during the development of the thesis, does not fit the needs of the connection point between the edge and the core network, since it is a technology designed so that the start of communication takes place on the IPv6 side. As Uninett is an ISP that provides Internet to academic institutions, almost all of its end users act as clients (although it is true that it also has a significant amount of outgoing traffic, as seen in 3.4, largely due to the services offered by Uninett's customer networks) and they are the ones who initiate communications with another server, usually an external one. Therefore, communications will normally begin on one of the external networks, either through IPv4 or IPv6, will be transmitted over IPv6 through the core network and will later reach its destination on networks outside Uninett, with the possibility of being transformed from IPv6 to IPv4 depending on the reachability of that destination.

There is therefore a need for technology that allows packets to be transmitted from an IPv4 domain to IPv6. In 2.4, a number of technologies were explored. Of these, the tunnelling mechanisms were not considered appropriate for the case study because of their limitations in many cases and the need for extensive infrastructure, which in the case of a large ISP can be very costly. Regarding the translation technologies, there is one that seems to fit the needs of the design, although it does

not have official certification, as commented in 2.8.3, that is NAT46.

Figure 4.2 shows a tentative suggestion on how the use of NAT46 may work. On it, two User Equipments (UE) are located in one of the edge networks and each one has access to either the IPv6 or IPv4 part of it. When the one connected to the IPv6 network initiate a connection, it will be done normally and the packets will be transmitted to the core network. On the other hand, when this is done through IPv4, there will be a NAT46 node at the exit of the IPv4 network that is in charge of the translation to convert these packets to IPv6 so that they can go to the core network through the IPv6 edge network (since this will facilitate that there is only one link needed connecting the edge and the core network). Since the technology requires the presence of DNS-ALG servers, these could be provided by Uninett at the core network, in order to reduce the total amount of DNS needed by the whole network. The NAT46 node would be connected to an IPv6 node which would in turn connect to the core network. This ensures that the expected reduction in speed on the IPv4 network, due to translations, will not affect the rest of the IPv6 traffic.

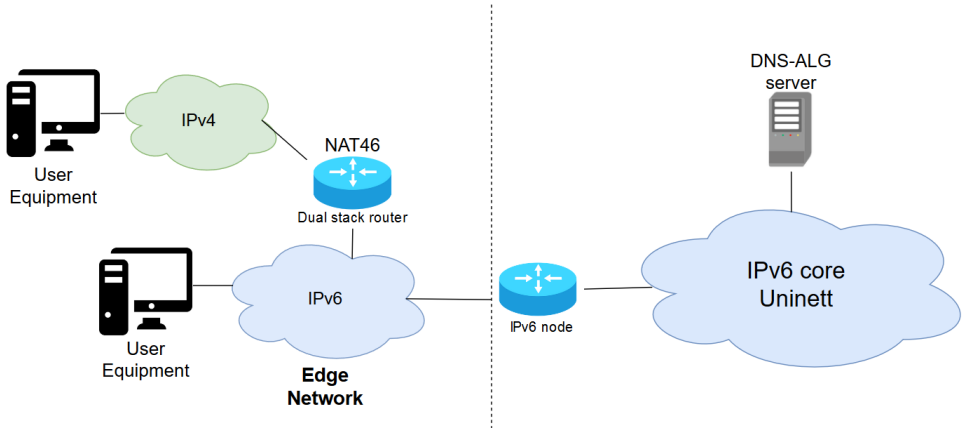


Figure 4.2: Edge network model of connection

Because the address associations are 1:1 for both domains, this technology has no advantage in reducing IPv4 addresses. However, because the translation would be done within Uninett's private network, or the institution itself, it is possible that all IPv4 addresses used within Uninett's domain are private addresses. Each institution could have a range assigned to it and the full spectrum of IPv4 could be used internally. Subsequently, the NAT64 nodes that would act as a gateway to the Internet would allow the reuse of IPv4 addresses and could have a smaller IPv4 pool than the one used by Uninett at present, thus reducing the total number of global IPv4 addresses required by Uninett.

There is a possibility that some of the edge networks already have nodes that are compatible with this technology, in any case this is unknown so it must be assumed that an investment of at least 19 NAT46 nodes should be made to take care of the translation of IPv4 traffic to IPv6 in each of the institutions. The case of each of the edge networks has not been explored so various combinations may be possible depending on the circumstances. It is possible that one may have several connection points to Uninett, in which case the possibility of enabling a single connection point as suggested in the figure above, or splitting IPv6 and IPv4 traffic to access the core network through different access points, should be studied. As already mentioned, the configuration required may be different for each of the 19 networks and may affect the solution economically, as nodes with need of higher capacity will be more expensive than other with lower requirements. Here it is assumed that the configuration in which a single NAT46 node can handle all the IPv4 traffic in each institution is possible in all cases. The presence or absence of internal DNS servers in each of the edge networks should not influence, therefore it is a datum that is not taken into consideration, however it is also unknown the number, if any, of DNS-ALG servers within Uninett core network. Another investment should be necessary in order to have enough DNS-ALG servers to handle all queries from the IPv4 domain.

Cisco Systems with its 4000 series [Sysb] and Juniper Networks with its SRX devices [Netb] offer a wide variety of NAT46 capable devices, with different specifications regarding throughput, power consumption, expansion ports or memory that could possibly fit for the requirements of each of the edge networks of Uninett or the interested ISP. Since the requirements of each of the edge networks are unknown, and the manufacturers do not offer public price data, it is not possible to suggest a model for each case, but it is good to highlight the wide range of products that cover this possible need.

Another possibility to solve the connectivity of the edge networks, is the use of a variant of the XLAT technology (commented during the section 2.7.1), called 464XLAT. As described in [MKB13], it is a translation mechanism that through the addition of prefixes and suffixes, same way as already explained in 2.7.4, transforms IPv4 addresses into IPv6 ones. Besides, the 464XLAT does not require the use of any special DNS, as communications can go either through the IPv6 or IPv4 domains and connect with a regular DNS in order to get the record of the desired destination.

464XLAT differentiates between two types of nodes: those that are on the client side and perform a 1:1 translation between IPv4 and IPv6 addresses, which are called Customer-side Translators (CLAT); and those that connect the ISP's network to other neighboring networks and perform an N:1 translation between IPv6 and IPv4 domains. In practice, the latter function as NAT64 and are called Provider-side Translator (PLAT). CLAT does not necessarily need to work with PLAT, depending on the objectives of the network it is possible to dispense with PLAT or it can be replaced by NAT64 as both perform the same function, so the model proposed for the connection between the core network and neighbouring networks remains valid. However a major problem that has been encountered with 464XLAT is that it hasn't been found any vendor which manufactures nodes with CLAT, but only with PLAT support. Device-level solutions exist, as systems such as Android or Windows in their latest versions can be configured to work as CLAT, that only translates the incoming and outgoing packets of the device itself [LAC]. However, because there would still be a high number of devices that could not perform this configuration, it is not considered a feasible solution. There are also free operating systems for routers that support CLAT technology, OpenWrt for instance [Opea]. A more economical solution than the one mentioned above with NAT46, but also more laborious to apply, would be the installation of the aforementioned operating system and the configuration of the device as CLAT in all those routers that provide service to IPv4 LANs that want to be maintained in all academic institutions. In this case, the dependency of the institutions with their IPv6 network would be greater since these LAN routers would translate IPv4 packets to IPv6 and therefore it is necessary for the routers to be dual stack and to be connected to the institution's IPv6 network. For institutions whose IPv6 network is not sufficiently developed, this is probably not a valid option, although they could try to quickly develop their IPv6 network if they find it feasible. Figure 4.3 shows a possible representation of this type of connection.

Since the translation made in the 464XLAT is reduced to adding prefixes and suffixes over the IPv4 addresses, it would be convenient that the IP address of the DNS server be composed of an IPv4 address along with the prefixes and suffixes used by Uninett to form the true IPv6 address. This way IPv4 hosts could try to reach that "non-real" IPv4 address, and after crossing the 464XLAT it would become a real IPv6 address. Of course, this would only apply for those institutions that do not have their own DNS servers, which can happen in the smallest ones. In the figure 4.3, the 464XLAT router is a dual stack router that would be in charge of processing the IPv4 traffic of the LAN to which the UE is connected, and to transform it into IPv6 traffic so it can go through the IPv6 Uninett core network.

The economic impact of this option should be restricted almost exclusively to the time required to make all devices of interest compatible and operational with CLAT. This is due to the low performance requirements of OpenWrt and its compatibility

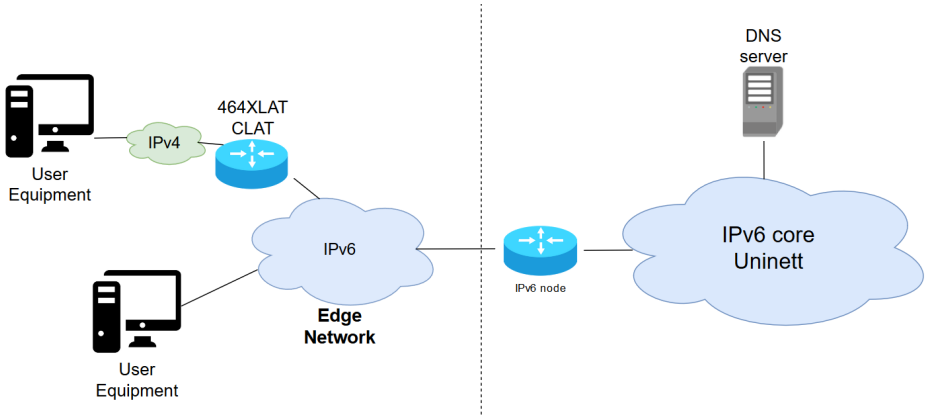


Figure 4.3: Edge network model of connection with CLAT

with a large number of architectures [Opeb]. It is expected that the vast majority of machines to be upgraded will be compatible and that the number of new equipment to be purchased will be minimal. Of course the situation may vary depending on the ISP. In the case of Uninett, it is expected that a very high number of devices will require this upgrade (in 3.4 the number of IPv4 LANs mentioned was of 1142 just for NTNU). However, this could be a good opportunity to restrict IPv4 access to specific areas and thus reduce the number of IPv4 subnets, the number of devices to be upgraded, and the number of IPv4 addresses required by the institution and by Uninett, which in the end this reduction of IPv4 addresses is one of the final objectives of changing the Uninett network model.

However there has been encountered some problems with this type of configuration. First of all, if an IPv4-only host tries to connect to an IPv6-only host. With 464XLAT there is no translation system from AAAA DNS records to A DNS records, as there is with NAT64 and DNS64 or with NAT46 and DNS-ALG in reverse. Therefore, although the IPv4-only host might be able to connect to any type of DNS server, if it only obtains AAAA records it will not be able to initiate a communication to that destination. In any case this is not considered a serious problem due to the very low probability that such a situation is believed to occur. Although it is possible that IPv4-only hosts are found in some of the edge networks (more likely because there is no IPv6 network available, than because the host is not IPv6 compatible), it is very rare that it will try to connect to an IPv6-only host. Most servers on the Internet are dual stack or IPv4-only, as they try to get a maximum reachability, so it is not considered that this limitation in communication will have an impact to take into account. Another perhaps more critical problem is in the use of free software as a solution, because, in the event of any serious problem in the devices with OperWrt,

support would not be available.

As mentioned above, each network should be studied in detail if this model is to be applied in reality. It is possible that a combination of the previously proposed technologies may be the most optimal option at Uninett, and that each edge network may opt for one or another option depending on its circumstances. It is possible that some of them already have the infrastructure to choose to use NAT46 while for others it is more economical to depend on CLAT. This decision may be largely influenced by the state of development of the IPv6 network in each case, with NAT46 being the least dependent on that network. The importance of valuing each network independently is therefore stressed, since the solution at the other end, depending on NAT64, is valid for both options. Nevertheless, although it is believed that the option with 464XLAT is important to be highlighted, in case some stakeholder can find it suitable for their interests, because of its cons and its high dependence on a heavily developed IPv6 network it is not recommended in this case and the solution through NAT46 is preferred.

4.2.3 Complete model

If the suggested models are applied, the whole model of the network would be similar to that of the figure 4.4, where *Edge Network* represents any of the institutional customer networks of Uninett and only one of them is represented in aim of simplicity.

Thanks to the use of DNS64 and DNS-ALG servers, as well as NAT46 and NAT64 translators, any kind of connection should be possible. Hosts from IPv4 or IPv6 should be able to establish connections to IPv4 or IPv6 destinations indiscriminately and without any really important limitations (beyond certain IPv4 and IPv6 functions that would be lost with translations). The investment needed to realize this model can vary greatly depending on each ISP but can be reduced to:

- As many NAT64 nodes as there are contact points between the core network and other neighbouring networks.
- As many NAT46 nodes as edge networks are available to the ISP.
- Enough DNS-ALG and DNS-64 servers to serve the entire network.
- The operational and power costs of all of the above.

It is likely that some of the above infrastructure (especially that related to DNS servers) is already available from the ISP, thus reducing overall costs. Also, as already mentioned, it is possible to redistribute pure IPv6 traffic to be driven by certain nodes and separate it from that which has been translated, or will be translated

to reach its destination, thus reducing the number of NAT64 nodes needed. Or there may be the possibility of joining several edge networks with a single NAT46 node, if its layout permits, and thus reduce the number of NAT64 nodes. Numerous combinations are possible which must be adapted to the specific situation of each ISP.

Formalizing the above in the form of mathematical expressions for any ISP, the total investment can be simplified by:

Cost of upgrading edge networks:

$$T_e = \sum_{i \in N_c} C_{nat46}[max(T_o(ipv4, i), T_i(ipv6, i))] + O(nat46) - \sum_{j \in S_{nat46}} C_{nat46}[B_{nat46}(j)] + O(nat46)$$

Cost of upgrading core network exit points:

$$T_c = \sum_{i \in N_p} C_{nat64}[max(T_o(ipv6, i), T_i(ipv4, i))] + O(nat64) - \sum_{j \in S_{nat64}} C_{nat64}[B_{nat64}(j)] + O(nat64)$$

Cost of DNS-ALG servers:

$$T_{dnsalg} = \sum_{i \in N_{dnsalg}} C_{dnsalg}[B_{dnsalg}(i)] + O(dnsalg) - \sum_{j \in S_{dnsalg}} C_{dnsalg}[B_{dnsalg}(j)] + O(dnsalg)$$

Cost of DNS64 servers:

$$T_{dns64} = \sum_{i \in N_{dns64}} C_{dns64}[B_{dns64}(i)] + O(dns64) - \sum_{j \in S_{dns64}} C_{dns64}[B_{dns64}(j)] + O(dns64)$$

Where:

N_c: set of customer edge networks.
N_p: set of points in contact with neighbour networks.
N_{dnsalg}: set of necessary DNS-ALG servers.
N_{dns64}: set of necessary DNS64 servers.
S_x: set of available units of type $x \in \{nat46, nat64, dnsalg, dns64\}$.
$B_x[i]$: capacity of unit i of type $x \in \{nat46, nat64, dnsalg, dns64\}$.
$C_x[c]$: investment cost of unit of type $x \in \{nat46, nat64, dnsalg, dns64\}$ with (symmetric) capacity c.
$O[x]$: operational and power cost of unit of type $x \in \{nat46, nat64, dnsalg, dns64\}$.
$T_o(p, i)$: peak traffic load from protocol $p \in \{ipv4, ipv6\}$ out of network/point i.
$T_i(p, i)$: peak traffic load from protocol $p \in \{ipv4, ipv6\}$ into network/point i.

As for the benefits, the fundamental one is a high decrease in the presence of IPv4 in the network. As for the core network, the presence of IPv4 would be reduced to zero, thus simplifying it, reducing operational costs and perhaps even dispensing

with certain nodes, thus reducing the total cost of the network. However, the greatest benefit in terms of protocol change would be found in the reduction of IPv4 addresses required by the ISP. As has been mentioned many times, the number of available IPv4 addresses is becoming smaller while the number of devices that require an Internet connection is increasing. ISPs around the world are having difficulties squeezing out the address space they have available to serve all their users. The use of NAT64 at the exit of the network would allow a better reuse of the address space, putting an end to this problem. Furthermore, thanks to the joint use of NAT64 and NAT46, the entire IPv4 spectrum could be used internally as private addresses, while the required number of global IPv4 addresses would be lower. In addition to solving the current problem, decreasing the number of IPv4 addresses required would allow the ISP to sell the remaining ones, which can reach a high market value due to their scarcity and could cover, at least to some extent, the investment needed for the new model.

Again, these economical benefits can be simplified in the form of a mathematical expression:

$$W = (A_{ipv4} - \sum_{i \in N_c} M_{ipv4}(i) * a_{ipv4}(i)) * V_{ipv4} + N_r * o_{ipv6}$$

Where:

W: total benefits for the ISP.
$a_{ipv4}(x)$: fraction of hosts in network x in need of global IPv4 address.
o_{ipv6}: per router factor for operational cost reduction from dual stack to IPv6-only.
N_r: number of routers in core network.
$M_{ipv4}(x)$: number of IPv4 only hosts in network x.
V_{ipv4}: market value of IPv4 address.
A_{ipv4}: number of IPv4 addresses currently controlled by the ISP.

The model presented here is a model prepared for the years to come. As the presence of IPv6 increases in the edge networks, and in the Internet in general, the ISP's IPv4 address pool can be further reduced, as well as, at a certain point, reducing the complexity of the network by removing the translation nodes when they are no longer needed. However, there are hundreds of factors to be taken into account when presenting a specific model for the case of a particular ISP. Here the case of Uninett has been taken as a reference, however the high number of variables to be taken into account, as well as the limited scope of the project, by its very nature, have prevented a more concrete solution from being proposed for Uninett. Any ISP that may find interest in the model presented should carry out a detailed analysis of each of its edge networks, an exhaustive market analysis in terms of the availability

of models and a further study of the risks that these technologies may entail in order to apply them in the most optimal way to reality. In any case, the previously shown expressions of costs and benefits of the model, can represent a tool set of value in the decision processes for ISPs planning for the future.

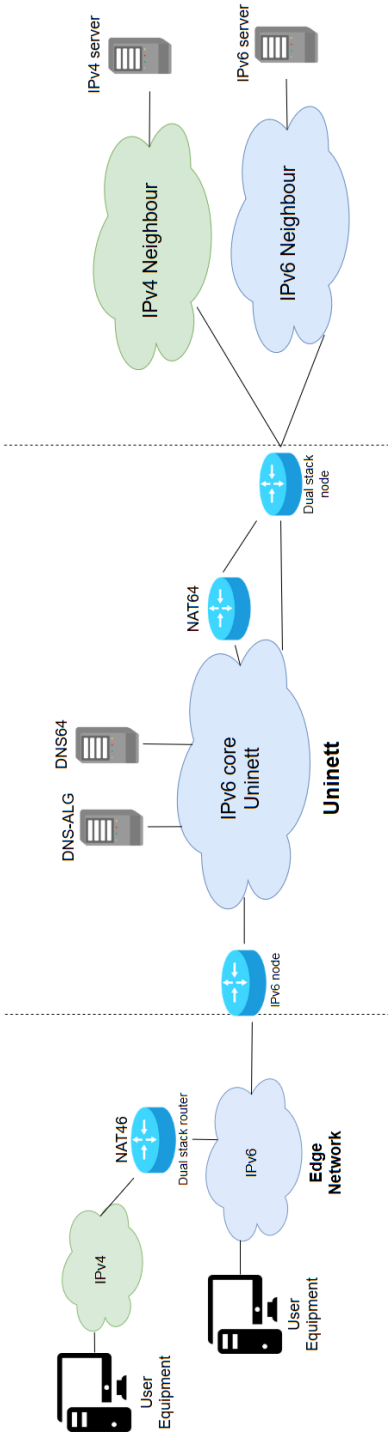


Figure 4.4: Whole network model

Chapter 5
Conclusion

This thesis has largely served to ascertain the state of IPv6 at present. It is curious how a technology that seems clear to represent the future of communications is advancing so slowly. Despite the long road ahead, a large part of today's Internet and its users are finally compatible with the technology. It is time for networks to be upgraded to operate entirely with IPv6.

It is important to remember that with the nearby Internet of Things, the number of devices that require an Internet connection will increase very soon and very quickly. Most of these devices do not require a high bandwidth or very stable connections for its proper functioning, so it is likely that the total network traffic will not increase too much, however what is certain is that all these new devices will require IP addresses to obtain service.

There are therefore a number of factors that are beginning to increasingly drive a migration to IPv6. For this reason, and because of the facilities that exist today in terms of transition technologies and greater compatibility with IPv6 in general, although the growth of IPv6 has always been constant and slow, it could accelerate over the next decade.

With all the advances made in transition technologies over the years, it seems that we have finally reached a point where it is possible to depend on these technologies and on IPv6 to provide service to users and reduce dependence on IPv4. Dependence that, by the way, is increasingly difficult to maintain due to the limited resources available in terms of available addresses. In view of the growing options for IPv6 accessibility and the increasing demand for IPv4 addresses, it may be a strategic move for an ISP to make the effort to migrate its network model to one similar to that proposed during the previous chapter. Not only is this a model that would allow the ISP to be prepared for the years to come, since it would have a strong IPv6 infrastructure that would sooner or later become indispensable, reducing dependence on IPv4 would allow it to be more flexible when it comes to gradually updating

the IPv4 part of the network, and in addition, the high prices that IPv4 addresses currently reach would allow, in the event of selling part of the current IPv4 address pool, to finance to a large extent the investment required to carry out the network model. In addition, it is expected that over time ISPs will eventually migrate to IPv6 in one way or another, making IPv4 networks those that need options to transition to an IPv6 Internet. Furthermore, when a very high state of IPv6 adoption is reached, IPv4 addresses will lose their value, so it will be the early adopters of this type of model who will find the greatest benefits in it. In any case, it is expected that there will still be many years left until the situation described above.

Although the model presented here is believed to be suitable for the case of Uninett and similar ISPs, it has repeatedly stressed the importance of taking this study further in order to put it into practice in reality. In addition to the special conditions during development, studying the network model of a network as large as Uninett has proved to be too ambitious to be carried out by one person and for such a limited time. There are a number of points, which have been mentioned during the thesis, that should be re-evaluated and strengthened, and many others that have barely scratched the surface.

Based on the results observed, the feasibility of the project is believed. However, it would be innocent to take the limited results observed here as maximum certainties. By means of a more in-depth study, carried out by the interested parties, it might be possible to ensure the viability of the project or to make small adjustments to ensure it. Uninett and the other interested parties are recommended to carry out this study, as it is believed that continuing the work started here could bring great benefits in the medium and long term.

References

[AD07] Cedric Aoun and E Davies. Reasons to move the network address translator-protocol translator (nat-pt) to historic status. Technical report, RFC 4966, July, 2007.

[Aka] Akamai. Akamai IPv6. https://www.akamai.com/uk/en/resources/our-thinking/state-of-the-internet-report/state-of-the-internet-ipv6-adoption-visualization.jsp#networks. Accessed: 2020-03-03.

[Alea] Alexa. Alexa Top 50 sites Norway. https://www.alexa.com/topsites/countries/NO. Accessed: 2020-06-01.

[Aleb] Alexa. Alexa Top 500 sites. https://www.alexa.com/topsites. Accessed: 2020-03-07.

[ALG16] T Anderson, Redpill Linpro, and F Gont. Ip/icmp translation algorithm. *IETF RFC7915, June*, 2016.

[Ali12] Amer Nizar Abu Ali. Comparison study between ipv4 & ipv6. *International Journal of Computer Science Issues (IJCSI)*, 9(3):314, 2012.

[APN] APNIC. APNIC IPv6. https://stats.labs.apnic.net/ipv6/. Accessed: 2020-03-06.

[ASS+11] Mohammad Aazam, Adeel M Syed, Syed Atif Hussain Shah, Imran Khan, and Muhammad Alam. Evaluation of 6to4 and isatap on a test lan. In *2011 IEEE Symposium on Computers & Informatics*, pages 46–50. IEEE, 2011.

[BAD14] Olabenjo Babatunde and Omar Al-Debagy. A comparative review of internet protocol version 4 (ipv4) and internet protocol version 6 (ipv6). *arXiv preprint arXiv:1407.2717*, 2014.

[BHB+10] Congxiao Bao, Christian Huitema, Marcelo Bagnulo, Mohamed Boucadair, and Xing Li. Ipv6 addressing of ipv4/ipv6 translators. *RFC6052*, 2010.

[BMvB+11] Marcelo Bagnulo, Philip Matthews, Iljitsch van Beijnum, et al. Stateful nat64: Network address and protocol translation from ipv6 clients to ipv4 servers. *IETF, April*, pages 2070–1721, 2011.

[BP05] Marc Blanchet and Florent Parent. Ipv6 tunnel broker with the tunnel setup protocol (tsp). *work in progress*, 2005.

[Bri] Martin Brinkmann. Guidance for configuring IPv6 in Windows for advanced users. https://www.ghacks.net/2011/02/06/how-to-enable-ipv6-on-windows-xp/. Accessed: 2020-06-12.

[BSMVB11] Marcelo Bagnulo, Andrew Sullivan, Philip Matthews, and Iljitsch Van Beijnum. Dns64: Dns extensions for network address translation from ipv6 clients to ipv4 servers. *IETF, April*, 2011.

[Car11] Brian Carpenter. Advisory guidelines for 6to4 deployment. Technical report, RFC 6343, August, 2011.

[CD+98] Alex Conta, Stephen Deering, et al. Generic packet tunneling in ipv6 specification, 1998.

[CER] CERT. SiLK documentation. https://tools.netsa.cert.org/silk/docs.html. Accessed: 2020-05-20.

[CJ99] Brian Carpenter and Cyndi Jung. Transmission of ipv6 over ipv4 domains without explicit tunnels, 1999.

[Con] Condor. Debian GNU/Linux on an IBM ThinkPad 600E. https://condor.depaul.edu/jkristof/debian-tp600e.html. Accessed: 2020-06-13.

[DCOPLF07] J De Clercq, D Ooms, S Prevost, and F Le Faucheur. Connecting ipv6 islands over ipv4 mpls using ipv6 provider edge routers (6pe). *Internet Engineering Task Force RFC*, 4798, 2007.

[DDWL11] Alain Durand, Ralph Droms, James Woodyatt, and Y Lee. Dual-stack lite broadband deployments following ipv4 exhaustion. *IETF RFC6333, August*, 201(1), 2011.

[DFGL01] Alain Durand, Paolo Fasano, Ivano Guardini, and Domenico Lento. Ipv6 tunnel broker. *RFC3053*, 1, 2001.

[DH81] S Deering and R Hinden. Rfc 1883: Internet protocol. *Version*, 6:1995–10, 1981.

[EvB98] J. H. P. Eloff and Suzi van Buuren. Framework for evaluating security protocols in a banking environment. *Computer Fraud and Security*, January 1998.

[EVJC13] Martin Elich, Petr Velan, Tomas Jirsik, and Pavel Celeda. An investigation into teredo and 6to4 transition mechanisms: Traffic analysis. In *38th Annual IEEE Conference on Local Computer Networks-Workshops*, pages 1018–1024. IEEE, 2013.

[Fac] Facebook. Facebook IPv6. https://www.facebook.com/ipv6/?tab=ipv6. Accessed: 2020-03-03.

[GA19] Ignacio Rey Gallo-Alcántara. Ipv6 only national backbone. Project report in TTM4502, Department of Information Security and Communication Technology, NTNU – Norwegian University of Science and Technology, Dec. 2019.

[Gooa] Google. Google IPv6. https://www.google.com/intl/es/ipv6/statistics.html. Accessed: 2020-02-28.

[Goob] Google. Google Scholar. https://scholar.google.com/.

[Har] Jeff Harrington. Ipv6 end station addressing: Choosing slaac or dhcp. *NYSERNet*.

[HLFT94] Stan Hanks, Tony Li, Dino Farinacci, and Paul Traina. Generic routing encapsulation (gre). Technical report, RFC 1701, October, 1994.

[Hui06] Christian Huitema. Teredo: Tunneling ipv6 over udp through network address translations (nats). Technical report, RFC 4380, February, 2006.

[iana] iana. iana Number Resources. https://www.iana.org/numbers. Accessed: 2020-01-30.

[ianb] iana. RIPE NCC IPv6 Allocations. https://www.iana.org/numbers/allocations/ripencc/ipv6/. Accessed: 2020-01-30.

[iPe] iPerf. iPerf - The ultimate speed test tool for TCP, UDP and SCTP. https://iperf.fr/. Accessed: 2020-05-01.

[Kim17] Pyung Soo Kim. Analysis and comparison of tunneling based ipv6 transition mechanisms. *International Journal of Applied Engineering Research*, 12(6):894–897, 2017.

[KTA07] Sinchai Kamolphiwong Kuljaree Tantayakul, Robert Elz and Touchai Angchuan. Mobility mechanism between mipv4 and mipv6. Technical report, Centre for Network Reach (CNR) Department of Computer Engineering, Faculty of Engineering, Prince of Songkla University, Thailand, 2007.

[LAC] LACNIC. IPv6 Transition Mechanisms Status. https://www.lacnic.net/innovaportal/file/2907/1/ipv6-trans-mechs-status_v1-short-jordi-palet.pdf. Accessed: 2020-06-20.

[LBB11] Xing Li, Congxiao Bao, and Fred Baker. Ip/icmp translation algorithm. *Internet Engineering Task Force, RFC*, 6145, 2011.

[LBD+13] Xing Li, C Bao, W Dec, O Troan, S Matsushima, T Murakami, and T Taylor. Mapping of address and port using translation (map-t). *Internet Engineering Task Force, Internet-Draft (Work in Progress), Draft-ietf-Softwire-map*, 1, 2013.

[Mar] Net MarketShare. Operating System Market Share. https://netmarketshare.com/operating-system-market-share.aspx. Accessed: 2020-06-12.

[Mau10] Antti Maula. A review and qualitative analysis of ipv6 and ipv4 interoperability technologies. In *Seminar on Internetworking*, pages 2–6, 2010.

[Mic] Microsoft. Guidance for configuring IPv6 in Windows for advanced users. https://support.microsoft.com/en-us/help/929852/guidance-for-configuring-ipv6-in-windows-for-advanced-users. Accessed: 2020-06-12.

[MKB13] M Mawatari, M Kawashima, and C Byrne. 464xlat: Combination of stateful and stateless translation. *Internet Engineering Task Force, Internet-Draft (Work in Progress), draft-ietf-v6ops-464xlat-10*, 2013.

[NCC] RIPE NCC. What is IPv4 Run Out? https://www.ripe.net/manage-ips-and-asns/ipv4/ipv4-run-out. Accessed: 2010-09-30.

[Neta] Juniper Networks. Configuring NAT46 on SRX devices. https://kb.juniper.net/InfoCenter/index?page=content&id=KB33559&actp=METADATA. Accessed: 2020-06-21.

[Netb] Juniper Networks. SRX550 Services Gateway. https://www.juniper.net/us/en/products-services/security/srx-series/srx550/. Accessed: 2020-06-21.

[NG+05] Erik Nordmark, Robert Gilligan, et al. Basic transition mechanisms for ipv6 hosts and routers. Technical report, RFC 4213, October, 2005.

[Nik15] Mendi Nikkhah. On the adoption dynamics of internettechnologies: Models and case studies. *Publicly Accessible PennDissertations*, 2015.

[NNS06] T Narten, E Nordmark, and W Simpson. Rfc 2461 neigbhour discovery for ip version 6 (ipv6), 1998. *URL reference: http://www. ietf. org/rfc/rfc2461. txt*, 2006.

[Opea] OpenWrt. 464XLAT. https://openwrt.org/packages/pkgdata/464xlat. Accessed: 2020-06-20.

[Opeb] OpenWrt. OpenWrt Supported devices. https://openwrt.org/supported_devices. Accessed: 2020-06-20.

[P+81] Jon Postel et al. Rfc 791: Internet protocol. *RFC791*, 1981.

[PB15] C Pignataro and R Bonica. S. krishnan," ipv6 support for generic routing encapsulation (gre). Technical report, RFC 7676, DOI 10.17487/RFC7676, October 2015,< http://www. rfc-editor. org . . . , 2015.

[PLCC15] R Penno, Y Lee, G Chen, and M Chen. Internet engineering task force (ietf) r. despres request for comments: 7600 rd-iptech category: Experimental s. jiang, ed. *IETF RFC7600, July*, 2015.

[Res] ResearchGate. ResearchGate. https://www.researchgate.net/.

[SC11] Nejc Skoberne and Mojca Ciglaric. Practical evaluation of stateful nat64/dns64 translation. *Advances in Electrical and Computer Engineering*, 11(3):49–54, 2011.

[SH99] Pyda Srisuresh and Matt Holdrege. Ip network address translator (nat) terminology and considerations, 1999.

[SHP09] Mohd Khairil Sailan, Rosilah Hassan, and Ahmed Patel. A comparative review of ipv4 and ipv6 for research test bed. In *2009 International Conference on electrical engineering and informatics*, volume 2, pages 427–433. IEEE, 2009.

[Sta] StatCounter Global Stats. Operating System Market Share Worldwide. https://gs.statcounter.com/os-market-share#monthly-201905-202005. Accessed: 2020-06-12.

[STAH99] P Srisuresh, G Tsirtsis, P Akkiraju, and A Hefferman. Dns extensions to network address translators (dns alg). Technical report, RFC 2694, Internet Engineering Task Force, 1999.

[Sysa] Cisco Systems. Cisco 1000 Series Aggregation Services Routersy. https://www.cisco.com/c/en/us/products/routers/asr-1000-series-aggregation-services-routers/index.html. Accessed: 2020-06-21.

[Sysb] Cisco Systems. Cisco 4000 Family Integrated Services Router Data Sheet. https://www.cisco.com/c/en/us/products/collateral/routers/4000-series-integrated-services-routers-isr/data_sheet-c78-732542.html. Accessed: 2020-06-21.

[Sysc] Cisco Systems. Cisco CSR1000v data sheet. https://www.cisco.com/c/en/us/products/collateral/routers/cloud-services-router-1000v-series/data_sheet-c78-733443.html. Accessed: 2020-06-01.

[Sysd] Cisco Systems. Cisco Systems IPv6. https://6lab.cisco.com/index.php. Accessed: 2020-03-07.

[Syse] Cisco Systems. Connectivity Between IPv4 and IPv6 Hosts Using Stateless NAT 46 . https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/ipaddr_nat/configuration/xe-16-10/nat-xe-16-10-book/iadnat-46.html. Accessed: 2020-06-21.

[Sysf] Cisco Systems. IP Addressing: NAT Configuration Guide. https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/ipaddr_nat/configuration/xe-16/nat-xe-16-book/iadnat-stateful-nat64.html. Accessed: 2020-05-01.

[Sysg] Cisco Systems. Planning Guide for Cisco Jabber 11.6 - Requirements to Support IPv6 in Android. https://www.cisco.com/c/en/us/td/docs/voice_ip_comm/jabber/11_6/cjab_b_planning-guide-cisco-jabber-116/cjab_b_planning-guide-cisco-jabber-116_chapter_010.html.

[Sys98] Cisco Systems. Isatap tunnel support for ipv6. *Interface and Hardware Component Configuration Guide, Cisco IOS XE Release 3S*, 1998.

[TDL+13] O Troan, Wojciech Dec, Xing Li, Congxiao Bao, Satoru Matsushima, Tetsuya Murakami, and T Taylor. Mapping of address and port with encapsulation (map). *Internet Engineering Task Force, Internet-Draft (Work in Progress), Draft-Ietf-Softwire-MAP-04*, 2013.

[TGT+08] Fred Templin, T Gleeson, D Thaler, et al. Intra-site automatic tunnel addressing protocol (isatap). *draft-ietf-ngtrans-isatap-02. txt (work in progress)*, 2008.

[TS00] George Tsirtsis and Pyda Srisuresh. Rfc2766: Network address translation-protocol translation (nat-pt), 2000.

[TT10] W Townsley and O Troan. Ipv6 rapid deployment on ipv4 infrastructures (6rd)–protocol specification. Technical report, RFC 5969, August, 2010.

[Ubua] Ubuntu. Ubuntu 14.04.6. https://wiki.ubuntu.com/TrustyTahr/ReleaseNotes/ChangeSummary/14.04.6. Accessed: 2020-06-13.

[Ubub] Ubuntu. Ubuntu releases. https://wiki.ubuntu.com/Releases. Accessed: 2020-06-13.

[Unia] Uninett. 10 years as an independent enterprise. https://www.uninett.no/sites/drupal.uninett.no.uninett/files/webfm/publikasjoner/digital.brytningstid.pdf. Accessed: 2020-02-25.

[Unib] Uninett. IPv6 usage in higher education (HE) in Norway. https://stats.uninett.no/ipv6stat/. Accessed: 2020-03-20.

[Unic] Uninett. Uninett statistics. https://stats.uninett.no. Accessed: 2020-03-05.

[UOUS03] Satoshi Uda, Nobuo Ogashiwa, Yojiro Uo, and Yoichi Shinoda. Ipv6 support on mpls networks: experiences with 6pe approach. In *2003 Symposium on Applications and the Internet Workshops, 2003. Proceedings.*, pages 226–231. IEEE, 2003.

[W3C] W3Counter. Browser and Platform Market Share. https://www.w3counter.com/globalstats.php?year=2020&month=5. Accessed: 2020-06-12.

[Wen12] Richard Wentk. *Mac OS X Lion Server Portable Genius*. John Wiley and Sons, 2012.

[Wie14] Roel J Wieringa. *Design science methodology for information systems and software engineering*. Springer, 2014.

[YPCK12] Se-Joon Yoon, Jong-Tak Park, Dae-In Choi, and Hyun K Kahng. Performance comparison of 6to4, 6rd, and isatap tunnelling methods on real testbeds. *International Journal on Internet & Distributed Computing Systems*, 2(2), 2012.

Appendix A
GNS3 configuration

This appendix explains all the commands and settings necessary for the performance study of the Section 3.2 both for the Ubuntu containers and the router. The commands are presented between quotes ("") and those parts of the commands that should not be understood literally but represent something, such as an address or a name, are presented between asterisks (**).

For the Ubuntu containers the configuration and functions applied were minimal:

In order to configure the network interface and provide IP addresses, it was decided to edit the *Interfaces* file, which is located in the etc/network directory and is loaded every time the containers are initialized. Figure A.1 shows the *Interfaces* file for both containers.

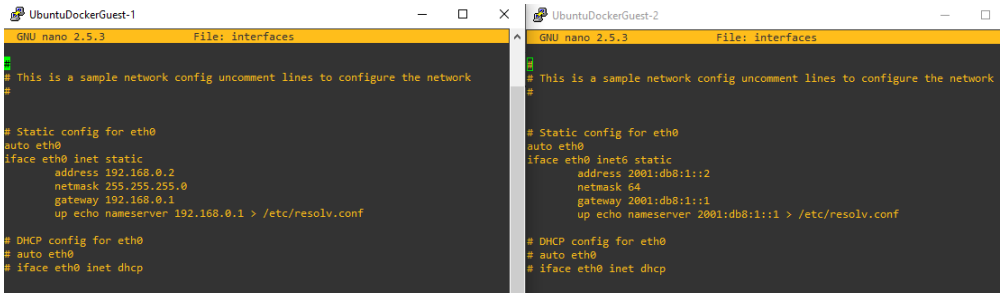


Figure A.1: *Interfaces* file with IPv4 (left) and IPv6 (right) configuration

To test the connectivity the tool iPerf3 was used. iPerf3 allows to set one machine as a listener (server) and the other to send data streams to the listener (client). It is always tried to obtain the maximum achievable bandwidth and after the test, it presents measures of bandwidth and packet loss among others [iPe]. To use it the following commands were used:

- *"iperf3 -s"*, to listen to data sent with iperf.
- *"iperf3 -c *IP address*"*, to send data streams to the specified address.

In the CSR 1000v router, a bit more complex configuration had to be applied:

- *"conf terminal"* is required to access the device's configuration mode.
- *"ipv6 unicast-routing"* is used to enable the forwarding of IPv6 packets.
- *"int *name of the interface*"* to access the configuration of a specific interface.
- *"ip address *the IPv4 address* *the subnet mask*"* can be used within the configuration of an interface to assign it an IPv4 address and its subnet mask.
- *"ipv6 enable"* and *"ipv6 address *the IPv6 address*"* are used instead to assign an IPv6 address.

Once the needed interfaces are configured, the next step is the configuration of NAT64 [Sysf]:

- *"nat64 enable"* is used to enable Stateless NAT64 on an interface of the router. Needs to be done on all the interfaces related with the whole operation.
- *"nat64 v4v6 static *IPv4 address* *IPv6 address*"* defines an static pair of translated addresses where the first defined address is an IPv4 address of a real interface and the last address is the IPv6 address that will be given to packets from the first interface when accessing the IPv6 domain.
- *"nat64 v6v4 static *IPv6 address* *IPv4 address*"* is analogous to the previous one but to give an IPv4 address to an IPv6 interface.

There are several different ways to do this configuration. It is possible for example to set up an address pool to be used when needed with *"nat64 v4 pool *name of the pool* *first IPv4 address of the pool* *last IPv4 address of the pool*"* or to establish an IPv6 prefix to be given to IPv4 packets when translated with *"nat64 prefix stateless *IPv6 prefix*"*. The method explained here is the one considered the most basic and simple for the case study.

Appendix B
SiLK code

Simplified and commented version of script using SiLK to obtain data. This example was used to get outgoing data from a network and is easily extrapolated to select incoming data.

Listing B.1: Bash script

```
DATA="Path to directory containing dataflows"
SILKCONF= "Path to configuration file"
START=2020/03/11:00
END=2020/03/12:00
BIN=3600

rwfilter --type=out,outweb,outicmp --start=$START --end=$END
--scidr="set of CIDR addresses" --pass=passout.rw  --fail=failout.rw
--data=$DATA
#Filtering of the whole data flow. In this case it is done based
#on the type of the records and its source IP addresses.


rwcount passout.rw --bin-size=$BIN --no-col --column-separator=";"
> countoutuiav4.txt
#The results of the counting are exported in a .txt file
```

rwfilter: tool to select SiLK Flow records form the data repository and filter them based on the following switches.

–type: data type selected to be filtered. Here, outweb, outicmp and out refer to outgoing traffic using TCP ports 80, 443, 8080, using the ICMP protocol, or which does not fall into the previous categories respectively.

–start: day and hour of the data flow from which data are selected.

–end: day and hour of the data flow from which data is no longer selected.

–scidr: source IP address or CIDR block.

–pass: file in which to write the filtered data .

–fail: file in which to write the non filtered data .

–data: root directory in which the data flow is stored.

rwcount: tool that counts previously filtered data to facilitate its reading.

–bin: size of each time bin in which data is added together. For this example one hour is used.