Implementing IPv6 in an Organization

Marya Steenman & Jan van Lith

7th February 2005

Master program System and Network Administration
University of Amsterdam

In cooperation with SURFnet

SURFnet

Abstract

During a research project done in 2004 there was concluded that the migration to IPv6 could be done without too many problems. Because this research was done on paper it would be interesting to test these conclusions in practice. This is what this report and project is about.

IPv6 is the next generation internet protocol. Its main benefit is a larger address space and enhanced security options. It is not yet possible to completely migrate to IPv6, but several Transition and Interoperability mechanisms are available to have Ipv6 and IPv4 coexist together in the same infrastructure.The SNB- lab will make use of dual-stack mechanisms offered in IPv6 capable hosts.

Making an inventory of the current situation is very advisable, because it can serve as a guidance for the rest of the project. Doing the work beforehand results in a good view of the amount of work that has to be done.

For the SNB-lab we made an inventory, and based on this inventory we were able to tell which software would not be able to implement IPv6.

The introduction of IPv6 in an organization takes several steps. One of them is designing a new infrastructure which will be IPv6 compatible. When an IPv6 infrastructure is going to be designed take into account the following aspects:

Numberplan RFC 2374 describes how to divide the address space into subnets. Decisions about which prefixes should or should not be announced have to be made.

Hardware considerations Network hardware needs to be IPv6 compatible.

Testing Test the new network design on a small scale, because the impact can not be predicted.

In the SNB-lab migrating the infrastructure was without problems. The only unexpected thing we encountered were the double IPv6 addresses assigned to each interface due to lacking configuration of router announcements of prefixes.

After implementing the IPv6 infrastructure the next step is for hosts (server/ clients) to be able to make IPv6 based connectivity. In order for this to work IPv6 has to be activated on every host. After establishing basic connectivity it is good to know how the hosts (and applications) deal with the IPv6 connectivity. The following tests can be done:

- Name resolving
- Remote login
- File Transfer
- Web browsing
- Mail sending and receiving

When running these tests in the SNB-lab we discovered that basic connectivity was no problem on any of the platforms.

Migrating different services to IPv6 depends on the possibility of the software version whether this is possible or not. Most newer software versions have IPv6 implemented.

Contents

Chapter 1

Introduction

The discussion about IPv6 has started a long time ago. The need for a replacement of its predecessor (IPv4) has been acknowledged even longer. Developers more and more implement IPv6 into their software and network parties like Cisco have adopted IPv6 as well.

Why is it so difficult for companies (and even ISPs) to implement IPv6? Research done by students of the FNWI (Faculty Physics Mathematics and Computer Science) faculty of the University of Amsterdam [2] have concluded that implementing IPv6 does not have to be as difficult as people think it is going to be.

SURFnet is an organization which aims to provide Internet connectivity to educational and research organizations throughout the Netherlands. It is also very active when it comes to IPv6 and it provides a backbone capable of routing IPv6 traffic. In order for their customers to implement IPv6, hesitation towards implementing IPv6 has to be taken away. The research done by A. Ahrouch and S. Ezzine ([2] is only done in literature. Our goal is to test these conclusions by a practical research, in order to take away some of the hesitation. This research is done in the context of the project RP1 at the Master SNB (System and Network Administration) at the University of Amsterdam.

1.1 About this document

This document is divided in two parts, one is theoretical and one is practical. The theoretical part is a description of encountered problems and a guideline that can be used by companies when migrating to IPv6. The practical part is the story about the SNB-lab migrating to IPv6.

The chapters are chronological and match the steps we have taken to introduce IPv6 in the SNB-lab.

1.2 Acknowledgements

We would like to thank SURFnet for the free IPv6 gigabit connection. Especially we would like to thank Wim Biemolt and Mendel Mobach for their suggestions about this project and for reviewing the report.

Chapter 2

Research Project

SURFnet wanted to know if the implementation of IPv6 in an IPv4 environment gives any problems. This part describes what we are going to research, what not, what preconditions there are and how we planned the work to be done. Our project definition therefore is:

What problems do you encounter when implementing IPv6 in an IPv4 environment such as that of the organization SNB?

This question will be answered by answering our research questions. The following research questions are answered in this project.

- What kind of problems do you encounter in the area of network devices, hosts- and server connectivity and applications when implementing IPv6?
- How does these problems reflect on the IPv4 environment?
- How does the IPv6 network on application level, reflects to the IPv4 network.

Because of the short period of time there is to complete this project, it has some boundaries. The project consist of:

- The setting up of the infrastructure in the SNB lab. This means: inventorisation, implementing a numberplan, configuring and connecting the network devices.
- Adding IPv6 functionality on the following Operating Systems: Mac OS X, a Linux distribution, a BSD distribution, Windows XP, Windows 2003 server.
- Research and adding of IPv6 functionality for the following services: DNS, SMTP, SSH, IMAP, LDAP, HTTP and DHCP.

Not a part of this project will be:

- If the there's no IPv6 support (discovered in the inventorisation) for a service and/or applications IPv6 functionality will not be added.

Preconditions of our project are:

- When the implementation of IPv6 on the services takes to long or when it's not possible, the next service will be implemented so that not too much time will be spend.
- The IPv4 infrastructure will not be touched. All the services that are now available on IPv4 will also be available after the project.

After setting the boundaries for this project the scheduling has to be done. The work has to be done in two weeks. The steps that need to be taken are as follow:

step 1 Inventarisation of the infrastructure and services in the SNB lab.

step 2 Development of the IPv6 network and an IPv6 numberplan

step 3 Implementing and testing of the IPv6 infrastructure.

step 4 The testing of IPv6 functionality on Operating Systems.

step 5 Adding IPv6 functionality for the services.

step 6 The writing of a report.

Chapter 3

Introduction IPv6

3.1 History

In the future there will be a shortage in IPv4 addresses (and address space) and some flaws in the protocol stack a next generation IP protocol was being developed[1]. Discussion and development on IPv6 started in the 1990s by the IETF (Internet Engineering Task Force). The first official recomendation on IPv6 was approved by the Internet Ingineering Steering Group in 1994.

3.2 IPv6 characteristics

The main differences between IPv4 and IPv6 are:

Address space The IPv6 address space is 128 bits which gives a larger address space. There are three types of addresses Unicast , Multicast en Anycast.

Autoconfiguration mechanisms Autoconfiguration gives the possibility to configure hosts automatically. IPv6 addresses can be based on an prefix offered by the router and the host MAC address.

Header format(extensions and options) The IPv6 header differs form the IPv4 header in that it is of fixed length. The header is 40 bits long an includes two 16 byte addresses. This fixed header size results in higher processing speed. The IPv4 header fields which are not part of the IPv6 header have become optional en are handled in so called Extension headers.

Authentication and privacy IPv6 offers support for authenitcation and privacy.

Labeling Packets can be labeled to offer additional services like QoS.

3.3 Introducing IPv6

It is difficult to migrate from IPv4-only to IPv6-only, because IPv6 is not fully adopted in all systems currently being developed. Developers took into account that for IPv6 to succeed, transition to IPv6 should go without difficulties and without dependencies towards any aspect of the IPv4 infrastructure. This is why several Transition and Interoperability mechanisms are available. There are three categories: Dual Stack techniques, Tunneling techniques, Translation techniques.

3.3.1 Address notation

An IPv6 address has 128-bits, or 16 bytes. The address is divided into eight 16-bit hexedecimal blocks, seperated by colons. An example of an IPv6 unicast address:

```
2001:610:158:1500:20a:95ff:fede:d038
```

If there are zero's in the address then you can place a double colon, under the condition that you use it only once. IPv4 has subnetmasks that specifies the bits of the IPv4 address that belong to he network ID. IPv6 has prefixlengths, it specifies how many bits of the address specify the prefix. RFC 2374 [5] shows the 128 bits devided in three groups, table 3.1 shows these groups. RFC 2374 also explains how

name	bits
Public Topology	1-48
Site Topology	49-64
Interface Identifier	65 - 128

Table 3.1: IPv6 address scheme

subnets and hostadresses are constructed. The bits representing the Site Topology are to be used for designing subnets, the Interface Identifier bits are used to identify interfaces of different systems (one per interface). This means a total of 65536 subnets can be created each containing up to aproximately 20 trilion nodes.

3.4 CASE SNB-lab

3.4.1 SNB-lab: Introducing IPv6

The SNB-lab is an educational organisation which relies on the Hogeschool van Amsterdam for its IPv4 connectivity. This is why we chose to implement IPv6 in Dual Stack operation. The SNB-lab is connected through an IPv6 router with the SURFnet backbone. Two networks (IPv4 and IPv6) will coexist together in the SNB-lab and migration to native IPv6 will occur little by little.

Chapter 4

Inventory

Taking the inventory of the current infrastructure gives an outline of which software and hardware is part of the project. It is important that every part of the current situation is being described in detail. A detailed description of the infrastructure gives a good insight in which hard en software is to beIPv6 enabled. This outline can be used to determine which software or hardware is IPv6 compatible and which software has to be upgraded or migrated in a later stage. Table 4.1 gives an overview of which aspects to take into account when making an inventory.

System	Hardware	OS	Software	IPv6 capable
Network device	IPv6 compatible	Version	N/A	N/A
Server	IPv6 compatible	Version and Patches	Version and Patches	Y or N
Client	IPv6 compatible	Version and Patches	Version and Patches	Y or N

Table 4.1: Inventory list

4.1 CASE SNB-lab

4.1.1 SNB-lab: Inventory

The results of the inventory taken in our SNB lab is outlined in table 4.2.

System	**Hardware**	**OS**	**Software**	**IPv6 capable**
Network device	IPv6 compatible	Version and Patches	Software	IPv6 capable
Router	V	Experimental IOS 12.3	N/A	N/A
Server	IPv6 compatible	Version and Patches	Version and Patches	IPv6 capable
Siena	V	Macintosh Darwin 7.5	OpenLDAP 2.1.22	Y
Firenze	V	Macintosh Darwin 6.8	Apache 1.3.29	N
			Postfix 2.1.3	N
			Bind 9.2.2	Y
			OpenSSH_3.4p1	N
			Xinetd 2.3.10	Y
Carleone	V	OpenBSD	OpenSSH	Y
Pisa	V	Gentoo 2.4.25	Apache 2.0.48	Y
			OpenSSL 0.9.7d	Y
			OpenSSH_3.9p1	Y
Lucca	V	Gentoo 2.4.27-sparc	OpenSSH_3.9p1	Y
			Bind 9.2.2-P1	Y
			Sendmail 8.12.11	Y
Client	IPv6 compatible	Version and Patches	Version and Patches	IPv6 capable
Linux	V	Gentoo 2004-3	Chapter 6	Y
		Debian 2.6	Chapter 6	Y

Table 4.2: Inventory SNB-lab

Chapter 5

IPv6 infrastructure

The implementing IPv6 in an organization takes several steps. One of them is designing a new infrastructure which will be IPv6 compatible.

5.1 Number plan

Designing an IPv6 infrastructure implies, amongst others, designing a number plan. This number plan represents all future IPv6 subnets and addresses. The current IPv4 number plan can be translated to an IPv6 number plan or the transition to IPv6 can be used to design a new plan.

Deviding the IPv6 addres scheme into subnets is almost done the same way as with IPv4. The main difference is the fact that addresses used in IPv6 are 128 bits.

When designing a number plan we took several things in consideration:

- number of subnets
- types of systems in a subnet
- types of services on systems in a subnet

5.1.1 Hardware considerations

Hardware that is part of the infrastructure (routers, switches and hubs) have to be IPv6 compatible. The routers are the most important systems, because these systems are dealing with routing IPv6 traffic between the different (sub)networks. When implementing IPv6 you have to check the IPv6 compatibility of these devices.

5.1.2 Testing

Testing the new infrastructure must be done on a small scale, because the impact on the existing environment can not be predicted. We have tested the possible impact on a configuration consisting of two nodes and an IPv6 compatible router. Two interfaces of the router were configured for different IPv6 subnets and each subnet had one host attached to it. We used this configuration mainly to see what would happen if the router was switched on. When implementing IPv6 in an environment where lots of filtering/firewalling is done you have to be aware to also configure this for IPv6.

5.2 CASE: SNB lab IPv6 infrastructure

5.2.1 number plan

The number plan for the SNB-lab is designed for the future situation see appendix A.1. Several servers are going to be migrated and additional subnets will be introduced. The result will be a number plan

that can not be fully implemented. This is not a problem, because it is possible to just implement a small part of the plan and when the new situation is implemented, see appendix B.1, the rest of the plan can be introduced accordingly.

The IPv6 prefix assigned to the SNB-lab is 2001:610:158::/48. Bits 49 to 64 are to be used for creating subnets see table 3.1. Table 5.1 gives a number plan for the SNB-lab.

Subnet	Category	Prefix	Site	Interface	Mask
Productie Clients		**2001:610:158**	**1000**		
	Network Devices		1100		/64
	Servers		1200		/64
	Clients		1500	RA = ON	/64
Experiment Clients		**2001:610:158**	**2000**		
	Network Devices		2100		/64
	Servers		2200		/64
	Clients		2500	RA = ON	/64
Admin		**2001:610:158**	**3000**		
	Network Devices		3100		/64
	Servers		3200		/64
	Clients		3500	RA = ON	/64
Productie		**2001:610:158**	**4000**		
	Network Devices		4100		/64
	Servers		4200		/64
	Clients		4500	RA = ON	/64
Experiment		**2001:610:158**	**5000**		
	Network Devices		5100		/64
	Servers		5200		/64
	Clients		5500	RA = ON	/64

Table 5.1: SNB subnet-number plan

In table 5.1 you can see the 5 different subnets the SNB-lab will have in the future. Different systems in every subnet will have a different /64 range in the Site Topology.

Router interfaces:	IPv6 address:
Surfnet	**2001:610:00ff:0010::2/64**
Productie Clients	**2001:610:158:1100::1/64**
	2001:610:158:1200::1/64
RA = ON	**2001:610:158:1500::1/64**
Experiment Clients	**2001:610:158:2100::1/64**
	2001:610:158:2200::1/64
RA = ON	**2001:610:158:2500::1/64**
Admin	**2001:610:158:3100::1/64**
	2001:610:158:3200::1/64
RA = ON	**2001:610:158:3500::1/64**
Productie	**2001:610:158:4100::1/64**
	2001:610:158:4200::1/64
RA = ON	**2001:610:158:4500::1/64**
Experiment	**2001:610:158:5100::1/64**
	2001:610:158:5200::1/64
RA = ON	**2001:610:158:5500::1/64**

Table 5.2: SNB router interface configuration

In table 5.2 you can see the configuration of the interfaces for the different subnets. Each subnet is attached to one interface (Our router firmware does not support vlan routing). The first address is the identifier of the interface, the next three addresses are the gateway addresses for the subnets for the different systems within each subnet.

Servername:	IPv6 address:
Siena	**2001:610:158:1200::11/64**
LDAP	2001:610:158:1200::1:389/64
Firenze	**2001:610:158:1200::10/64**
SSH	2001:610:158:1200::1:22/64
DNS	2001:610:158:1200::1:53/64
IMAP	2001:610:158:1200::1:143/64
POP	2001:610:158:1200::1:110/64
Lucca	**2001:610:158:3200::10/64**
SSH	2001:610:158:3200::1:22/64
SMTP	2001:610:158:3200::1:25/64
DNS	2001:610:158:3200::1:53/64
NTP	2001:610:158:3200::1:123/64

SYSLOG	2001:610:158:3200::1:514/64

Carleone	**2001:610:158:3200::2/64**
SSH	2001:610:158:3200::2:22/64

Pisa	**2001:610:158:1200::13/64**
SSH	2001:610:158:1200::2:22/64
HTTP	2001:610:158:1200::2:80/64

Table 5.3: SNB server-number plan

In table 5.3 you can see the configuration of the interfaces of the servers. The first address is the fixed IPv6 address of the server and is based on the last octet if the IPv4 address of the same server (this way it is easy to remember which address belongs to which server).

The next addresses are *service* based. In this case the address ends with the port number of the service and is preceded by an number that indicates the next server on which the service is running in the same subnet, compare Firenze SSH and Pisa SSH.

```
Firenze: 2001:610:158:1200::1:22
Pisa: 2001:610:158:1200::2:22
```

This way addresses of services are not associated to a server but a service and can easily be migrated to another system. In IPv4, services must be bounded to an server interface and have an DNS record pointing towards this interface. In IPv6, it is recommended to bind your services to an IPv6 address and pointing your service DNS record to an service IPv6 address instead of an server interface address. Doing this will prevent clients from connecting to a service that is not available on IPv6, causing delay and even timeouts. An example: Firenze has an A record firenze.os3.nl pointing to 145.92.24.10 and an AAAA record firenze.os3.nl pointing to 2001:610:158:1200::10 The www record of os3.nl is also pointing to 145.92.24.10. Assume that firenze isn't able to implement IPv6 on his HTTP daemon then the DNS should not resolve the www.os3.nl record to an IPv6 address so the AAAA record of www.os3.nl must not be added. Beware of services, not IPv6 enabled, which have a CNAME record and are pointing towards a host that has an AAAA record. We recommend making an A record for this service.

This table also shows the current situation in which only two subnets are available.

5.2.2 Hardware considerations

The SNB-lab runs both IPv4 and IPv6 in dual stack operation. The IPv4 traffic is routed via the Hogeschool van Amsterdam and our IPv6 traffic via Surfnet. This is why there has been decided not to use a dual-stack router, but an IPv6-only router.

5.2.3 Testing

We have tested the possible impact on a configuration consisting of two nodes and an IPv6 compatible router. Two interfaces of the router were configured for different IPv6 subnets and each subnet had one host attached to it. After switching on the router the following situation occurred:

- Hosts were not automatically configured.
- No IPv6 related actions occurred.

The router we used was a cisco router running IOS 12.3. The configuration of this router is described in the same book we used to get general information on IPv6 [1]. It described that for IPv6 connectivity to be working IPv6 packet forwarding needed to be enabled.

```
ipv6 unicast-routing
```

```
nero#sh ipv6 interface eth 2
Ethernet2 is up, line protocol is up
IPv6 is enabled, link-local address is FE80::2E0:1EFF:FEBB:E05A
Global unicast address(es):
2001:610:158:1100::1, subnet is 2001:610:158:1100::/64
2001:610:158:1200::1, subnet is 2001:610:158:1200::/64
2001:610:158:1500::1, subnet is 2001:610:158:1500::/64
Joined group address(es):
FF02::1
FF02::2
FF02::1:FF00:1
FF02::1:FFBB:E05A
MTU is 1500 bytes
ICMP error messages limited to one every 100 milliseconds
ICMP redirects are enabled
ND DAD is enabled, number of DAD attempts: 1
ND reachable time is 30000 milliseconds
ND advertised reachable time is 0 milliseconds
ND advertised retransmit interval is 0 milliseconds
ND router advertisements are sent every 200 seconds
ND router advertisements live for 1800 seconds
Hosts use stateless autoconfig for addresses.
```

Figure 5.1: router IPv6 interface

The second time we run the router with its new configuration (IPv6 packet forwarding enabled) everything was working as we expected. Testing with the two connected hosts we discovered that every host received an address of each subnet configured on the interface it was attached to (our number plan shows that one interface is configured with multiple addresses). This problem was easily solved by turning of prefix announcements for specific subnets on each interface, see figure 5.2 which shows the configuration of one of the interfaces of the router.Now everything was working properly and we connected the router to the IPv4 Experimental network to see how hosts would react to this. As expected and tested in the test configuration nothing went wrong. Every host received one prefix and connectivity was established. At this point the remaining IPv4 subnets were connected to the router. As of this moment the IPv4 and IPv6 infrastructure coexist together. As a simple but fun test we connected to www.kame.net and www.ipv6.surfnet.nl.

We also tested what would happen when two hosts on a different subnet but in the same LAN connect to eachother. ICMPv6 takes care of this situation and the router sends one IMCPv6 redirect to each host. This ICMPv6 packet contains the link-local address of the other host. Now the hosts know the link-local address of eachother and connect directly without intervention of the router.

No problems concerning firewalling where encountered due to no firewall presence.

```
nero#sh run interface eth 2
Building configuration...

Current configuration : 350 bytes

interface Ethernet2
no ip address
ipv6 address 2001:610:158:1100::1/64
ipv6 address 2001:610:158:1200::1/64
ipv6 address 2001:610:158:1500::1/64
ipv6 nd prefix 2001:610:158:1100::/64 no-advertise
ipv6 nd prefix 2001:610:158:1200::/64 no-advertise
end
```

Figure 5.2: router interface configuration

Chapter 6

Operating Systems

When introducing IPv6 into an IPv4 environment you have to be sure you're operating systems are ready for this. That is why it is usefull to test IPv6 on some Operating Systems. Is it hard to configure IPv6? Are there any applications using IPv6? Is IPv4 still working? The following paragraph will give an answer to this kind of questions. The testing of operating systems was narrowed down to the following systems:

- Debian (Linux distribution)
- openBSD (BSD distribution)
- Mac OS X
- Windows XP/2003

Testing IPv6 connectivity will be done by some tests described below. Testing whether IPv4 connectivity is still available is also done.

Basic connectivity To test basic connectivity. The `ping` command for IPv4 and the `ping6` command for IPv6 are executed to check if other machine's are giving ICMP echo replies and thus if there is basic connectivity. Also the `traceroute` and `traceroute6` command are issued to check if routing is ok.

Name Resolving Testing if name resolving is still operational. The `dig` command on *nix-based Operating Systems and the `nslookup` command on Windows Operating systems are used to check name resolving.

Remote Login Testing of remote login capabilities. The commands `ssh` and `telnet` are used to test if remote login works on IPv6 an still works on IPv4

File Transfer File transfers are tested with `ftp`.

Web Testing of web browsing facilities. Testing is done by surfing to an IPv6 enabled website and one IPv4 only website with several browsers.

Mail Testing of mail sending and receiving capabilities. IMAP, POP and SMTP are tested on both IPv4 and IPv6 with several mail client software.

All the results on the testing are outlined per Operating System for the SNB-lab case.

6.1 Conclusion

Windows XP/ 2003, Linux distributions, BSD distributions, and Mac OS X will work fine on an IPv6 enabled environment. When connecting to services they will first use IPv6 before IPv4. There is a lot

of software that is IPv6 ready. The only thing you have to watch out for is whether the interface allows router advertisements. Windows has also no big problems with IPv6. Windows XP and Windows 2003 don't offer IPv6 support for file and print sharing. All operating systems have in common that software is the key to using IPv6 or staying at IPv4 but basic connectivity is not a problem for any of the operating systems tested. The main conclusion is that the operating systems have no problem with IPv6 in their IPv4 environment. All the tasks the systems could do before the implementation of IPv6 they can still be done in a dualstack environment.

6.2 CASE: SNB-lab

6.2.1 Linux Distribution

Linux kernels version 2.2 and above have an built in IPv6 implementation [3]. Loading of IPv6 (installed in the kernel) at startup depends on the linux distribution installed. Next step is to check if the interface that is connected to your LAN is accepting router advertisements. The command `sysctl -A |grep ipv6` is showing some installed preferences about IPv6 on a Linux distribution. Here you can change a lot of IPv6 preferences. When accept_ra = 1 or accept_radv = 1, the OS is accepting router advertisements an the command `ifconfig -a` will tell if the interface is having an IPv6 address. If you want static IPv6 addresses you have to set the `autoconf = 0` and `accept_ra = 0` with the command `sysctl -w net.ipv6.conf.all.accept_ra=0` and `sysctl -w net.ipv6.conf.all.accept_ra=0` command. This prevents router advertisements to be accepted and autoconfiguration of an IPv6 address on all interfaces. Now the address can be added statically. In debian you can do this in `/etc/network/interfaces`, so that the IPv6 address still will be configured after a reboot. This is what you need to add:

```
iface eth0 inet6 static
    address         2001:610:158:1000:dead::1
    netmask         64
    gateway         2001:610:158:1000::1
```

Don't forget to add the gateway! After testing you can see that with Debian is that it is first trying to connect at IPv6 address and when there's no AAAA record the IPv4 address is tried. You see this with an ftp session, a web session, and a mail session. This feature differs per Linux distribution and software used. All the tests where completed successfully. Only some old software doesn't work with IPv6 but will still work with IPv4.

6.2.2 BSD distribution

OpenBSD 3.6 has installed IPv6 functionality by default [3]. OpenBSD distribution doesn't accept router advertisements so you have to allow these.
Executing `sysctl -w net.ipv6.conf.eth0.accept_ra=1` will enable router advertisements. This is slightly different then in Debian. If you want to add static IPv6 addresses you have to do this the same way as in Linux. The configuration file that holds the network interfaces in openBSD is in `/etc/hostname.xl0`(xl0 is the name of the interface we used) here you can add a static address. We added the follwing line to our hostname.xl0: `inet6 2001:610:158:1200::11 prefixlen 64`. The gateway has to be added in
`/etc/rc.local`. We added the following line:
`route add -inet6 default 2001:610:158:1200::1` All the results of the tests where good. BSD has no problem handling IPv6 and is much the same as a Linux distribution.

6.2.3 Mac OS X

Mac OS X has no problem handling IPv6. It is installed by default from Mac OS X 10.2 [3]. The only test that wasn't successful was the web browsing with it's standard browser "Safari" but after a few configurations this browser can handle IPv6. type in your terminal `defaults write com.apple.Safari`

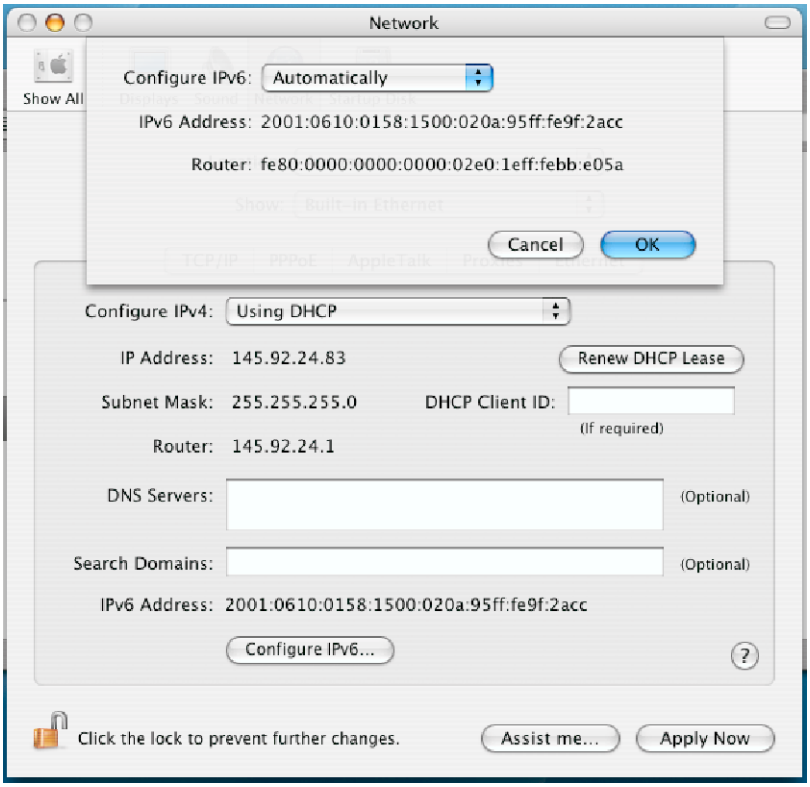

Figure 6.1: Configuring static IPv6 address on Mac OS X.

`IncludeDebugMenu 1`. This will include a debug menu in Safari. In this menu you have to deselect the http: (Simple Loader). Now its connecting also on IPv6. The most software on a Mac is IPv6 compatible. It is easy to configure static IPv6 addresses on Mac OS X. Open *system preferences*, click on *Network* and then double click on the interface you would like to configure. Now press configure IPv6 and select manual and insert your static IPv6 address. Figure 6.1 shows that you don't need to configure a gateway because Mac OS X has learned the link-local address of the router.

6.2.4 Windows XP/2003

Windows XP is by default IPv6 ready [4]. From Servicepack 1 or above you can install the IPv6 protocol in Network Neighborhood and you can get support from Microsoft. Without a servicepack installed you have to install IPv6 at the commandline and you have a developer preview, thus no support. In Windows 2003 server this protocol is installed by default. Adding static IPv6 addresses in Windows XP or 2003 is no problem. `netsh interface ipv6 add address "Local Area Connection" 2001:610:158:1200::48` configures the static IPv6 address and `netsh interface ipv6 add route ::/0 "Local Area Connection" 2001:610:158:1200::1` configures the gateway. With the `netsh` command you can change many more IPv6 preferences. Two tests where not successful on IPv6: The mail software (outlook, outlook express) isn't IPv6 ready but if you use other software like Thunderbird it is. And name resolving can't be done over IPv6. Windows XP and Windows 2003 don't offer IPv6 support for file and print sharing.

Chapter 7

Services

This section describes the problems that where encountered when enabling IPv6 in the following services:

- DNS
- SMTP
- SSH
- IMAP
- LDAP
- HTTP
- DHCP

There will be a detailed outline about the implementation of IPv6 in all these services in the CASE SNB section.

7.1 DNS

DNS takes care of name resolving. In an IPv4 environment DHCP configures the interface with IPv4 DNS servers. To make use of IPv6 to connect to a DNS server you have to manual configure the IPv6 address (this isn't possible with Microsoft XP). It isn't possible to let DHCP issue an IPv6 DNS server. That's why native IPv6 is no option in an DHCP environment. The configuration of a DNS server to make it resolve IPv6 hosts isn't difficult. There are two new DNS record types defined for IPv6 hosts.

- The AAAA record, described in RFC 1886 [6]. It stores a 128-bit IPv6 address. This record is similar to the A record in IPv4. The corresponding reverse lookup domain is IP6.INT and/or IP6.ARPA. IP6.INT was developed for easy storing of reverse information and will deprecate.
- The A6 record, described in RFC 2874 [7]. In cooperation with the DNAME record[8] they support renumerable and aggregatable IPv6 addressing. These record types are deprecating and must not be used.

It is easy to add these records in the DNS servers zone-file. It is nearly the same as adding IPv4 addresses. The only thing you have to watch out for is when a host connects to a hostname that has an A and an AAAA record. Some hosts will first try the IPv6 address and then the IPv4 address(depends on distribution). So it is advisable to add AAAA records only when the service on the server, where the record is pointing to, is working correctly under IPv6 otherwise this causes lack in connecting to it.

7.2 SMTP

There are a lot of SMTP daemons. The first thing you have to do is checking if the daemon your using supports IPv6 [3]. In our situation it wasn't possible to implement IPv6 on our SMTP mail server because of our operating system. A patch couldn't be applied to update our SMTP mail server to support IPv6 because the operating system doesn't support the new version of our SMTP server. Not only the version of your SMTP mailserver is important but also the version of your operating system. Windows standard DNS server works over IPv6 after a few configurations. See the case SNB for more details.

7.3 SSH

There are several patches for SSH to enable IPv6 support. Most of the systems are using openssh and are keeping there version up to date. Therefore it isn't much of a problem to get ssh working. And if you have an old openssh version you can upgrade it.

7.4 IMAP

IMAP is used for clients to access there mail. There are several IMAP servers and you have to check whether you have support for IPv6. In our case the IMAP server worked with the xinetd daemon [**?**]. This daemon supports IPv6 and after some configurations IMAP listens on IPv6. Also our pop server listened on IPv6 after making some changes to the configuration of to the xinetd daemon.

7.5 LDAP

LDAP software that has IPv6 support differs per vendor. During the inventory it became clear which version of LDAP is being used and can be sorted out whether or not this distribution supports IPv6.

7.6 HTTP

The ability for webservers to talk IPv6 is available in different distributions. Two widely used distributions are Apache webserver and Microsoft IIS. Both support IPv6 and for version 1.3 of the apache webserver patches are available for IPv6 communication.

7.7 DHCPv6

Stateless autoconfiguration of hosts (ip addressing without DHCP servers) is easy to use, but extra parameters can not be offered. DHCP offers the option of sending extra parameters (like DNS information) and is called statefull autoconfiguration. DHCPv6 is the IPv6 version of DHCP.

7.8 CASE SNB

7.8.1 DNS

In the SNB-lab we have Bind. Bind 8.4 or higher has IPv6 support. Older versions of bind 8 need a patch [3]. In the SNB-lab both DNS server are bind version 9.2.2. Bind has one configuration file called `named.conf`. In this file you have to add the line `listen on ipv6 { any; };` in the options tag to make the DNS server listen on IPv6 addresses. It is not possible to listen on just one interface. When issuing the command `dig @2001:610:158:1200::10 www.kame.net any` you can test if the dns server is really listening on IPv6.

Now IPv6 hosts can be added to the DNS server. This is done by adding the AAAA record in the zone file. An example of our zone file.

```
$ORIGIN         os3.nl.
$TTL            8H

@               IN SOA  lucca.capo.os3.nl. hostmaster.os3.nl. (
                                2005020200      ; serial
                                4H              ; refresh
                                30M             ; retry
                                7D              ; expire
                                20M             ; minimum
)

lucca.capo IN AAAA     2001:610:158:3200:0:0:0:10
lucca.capo IN A     139.162.25.10
www IN      AAAA   2001:610:158:3200:0:0:2:80
www IN A 139.162.24.10
```

This is the same as adding IPv4 addresses with A records. The service www points to an service IPv6 address. If the webserver moves to another server you can delete the IPv6 address on the old server and add it on the new, without changing any DNS records.

The PTR record of an IPv6 hosts is also very much the same as in IPv4.

```
$ORIGIN 8.5.1.0.0.1.6.0.1.0.0.2.ip6.arpa.
$TTL   8H
@   IN SOA  lucca.capo.os3.nl. hostmaster.os3.nl. (
                                2005013102 ; serial
                                4H         ; refresh
                                30M        ; retry
                                7D      ; expire
                                20M        ; minimum
                                )

$INCLUDE        2001:0160:0158.reverse
```

To also enable the IP6.INT zone we copied this zone file and renamed the origin to `8.5.1.0.0.1.6.0.1.0.0.2.ip6.int.` In the 2001:0160:0158.reverse file you can add the hosts. Our 2001:0160:0158.reverse file looks like this.

```
0.1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.2.3 IN PTR lucca.capo.os3.nl.
0.8.0.0.2.0.0.0.0.0.0.0.0.0.0.0.2.3 IN PTR www.os3.nl.
```

As we where curious how Microsoft's DNS server would handle IPv6 we also configured DNS on Windows 2003. For DNS to listen over IPv6 some configuration has to be done. It is described in Administering IPv6 [9]. The following has to be done:

1. Install Windows Support Tools.
2. Open Command Prompt.
3. Type the following command: `dnscmd /config /EnableIPv6 1`
4. Restart the DNS Server service.

It wasn't any problem to add AAAA records. Microsoft's DNS server has no problem handling DNS request on IPv6.

7.8.2 SMTP

Our postfix version is 2.1.3 there is a patch available for IPv6 connectivity. But our Darwin 7.3 or higher supports the new version of postfix and upgrading to a new OS version isn't an option because of the many special software. [10] That's why we tried to setup a sendmail server to work on IPv6 and IPv4. You have to have sendmail version 8.9.1 or higher for IPv6 support [3] they are already compiled with IPv6 support. We compiled sendmail with the -DNETINET6 flag on and added `DAEMON_OPTIONS('Port=smtp, Name=MTA, Family=inet6')dnl` in the sendmail.mc file. Compile the sendmail.mc file with m4 to sendmail.rc (`m4 sendmail.mc > sendmail.cf`) and restart sendmail. Now it is accepting connections on IPv6.

7.8.3 SSH

Our servers are using the ssh version 2 protocol with openssh version 3.x and are listening on IPv6. We didn't need to change anything. When SSH port forwarding fails on IPv6 it will not try to connect on IPv4. This problem can be solved by connecting with the `ssh -4` option.

7.8.4 IMAP

We use UW-IMAP4rev1 [11] and this application uses the xinetd daemon [**?**] for listening on interfaces. The xinetd 2.1.8.8pre series or higher has IPv6 support so we didn't had any problem making IMAP available for IPv6. Xinetd makes IPv6 available in legacy environments such as our MAC Darwin 6.8. In the imap file, that can be found in the xinetd.d directory, we added *IPv6* to the flags option.

```
service imap
{
        socket_type     = stream
        wait            = no
        user            = root
        server          = /usr/bin/imapd
        groups          = yes
        flags           = REUSE IPv6
}
```

The same can be done with the ipop3 daemon of UW-IMAP4rev1.

```
service pop3
{
        socket_type     = stream
        wait            = no
        user            = root
        server          = /usr/bin/ipop3d
        groups          = yes
        flags           = REUSE IPv6
}
```

Now xinetd will listen on all interfaces at IPv4 and IPv6 ports 143 and 110.

We also checked the standard pop mail server of Microsoft Windows 2003 but this pop server has no support for IPv6.

7.8.5 OpenLDAP

OpenLDAP supports IPv6 from version 2.0 [3]. Our LDAP server Siena (see Table 4.2 has OpenLDAP 2.1.22 installed and IPv6 was enabled by default. Testing the server on IPv6 went without any problems.

7.8.6 HTTP

Apache webserver

Apache supports IPv6 from version 2.0. Earlier versions can be made IPv6 aware through patches. The main webserver of the SNB-lab is Firenze 4.2 is running an older version of the Apache webserver. Theoretical the version can be patched, but because of some specific configurations and the use of webobjects this would be a difficult job. An normal installation of Apache can be patched easily. In the future Firenze will only be a filesserver and no longer an webserver. When the new webserver is built an IPv6 enables webserver will be installed.

Pisa another server running Apache is IPv6 enabled by default.

Microsoft IIS

The SNB lab mainly runs on different *nix distributions, because a great part of the customers of SURFnet use Microsoft based software we tested the Microsoft IIS webserver. IIS version 6.0 supports IPv6 out of the box. No configuration (apart from installation) has to be done.

7.8.7 DHCPv6

In the SNB-lab we use stateless autconfiguration of the hosts, this works fine. In the future we would like to use the more flexible DHCPv6. We looked at the DHCPv6 version distributed by Sourceforge. It contains a server and an client daemon which is installed on the machines who want to communicate with the server. We couldn't get this to work also due to the amount of time available for this project.

Chapter 8

conclusion

Throughout the document it becomes clear that introducing IPv6 in an IPv4 environment is no problem. Changing the infrastructure to a dualstack environment went without problems.

On a Linux distribution, a BSD distribution or MAC OS X there is a lot of software available that supports IPv6. It depends on the version and the platform it is installed on whether or not IPv6 can be enabled. Microsoft on the other hand has more software that has no IPv6 support, but alternatives are available and everything still works on IPv4.

It will take some time before everything has IPv6 support, until than both IPv6 and IPv4 can coexist together without any problems. Therefore it is advisable to implement IPv6 as much as possible, because sooner or later the migration from IPv4 to IPv6 has to be made. Important when deciding to implement IPv6 (eventhough it is partially) is to plan everything very carefully. Especially when it comes to services it is important to know whether or not the services installed and configured in your situation are capable of handling IPv6.

In our opinion it is not difficult to implement IPv6 in an IPv4 environment and if there are any hesitations left, the report shows that migration can go without difficulties.

Appendix A

Future infrastructure

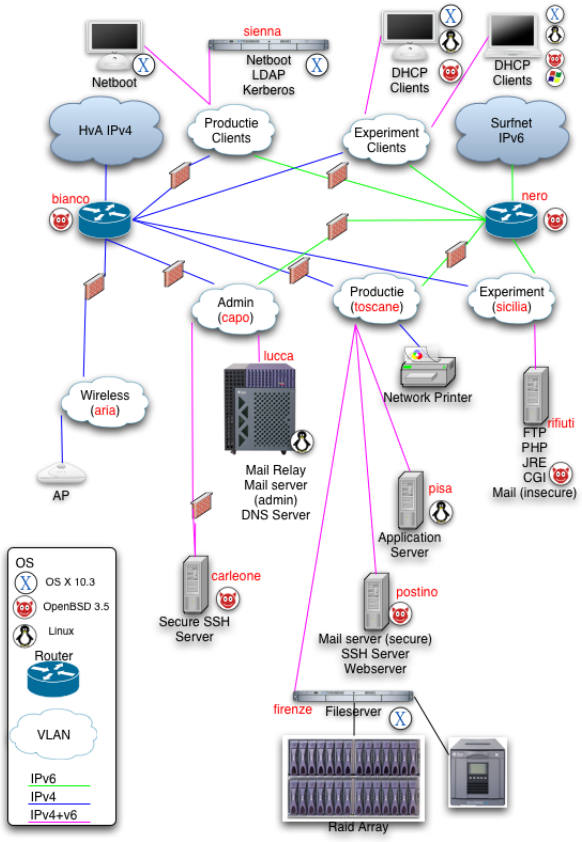

Figure A.1: Future SNB network

Appendix B

Current infrastructure

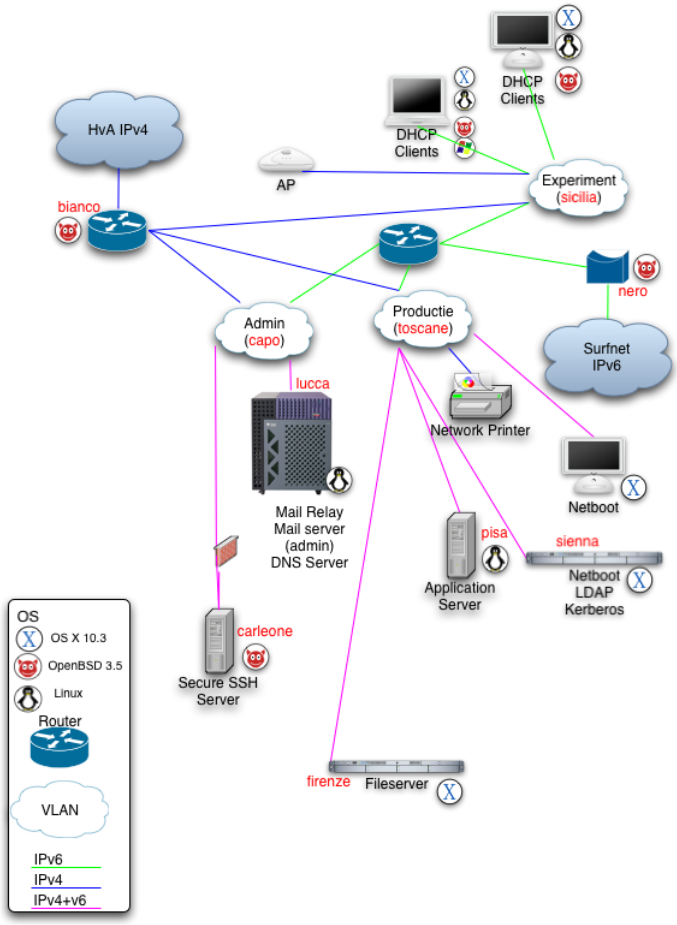

Figure B.1: Current SNB network

Bibliography

[1] Silvia Hagen. *IPv6 Essentials*.First Edition. O'Reilly, 2002.

[2] A. Ahrouch and S. Ezzine. *IPv4 to IPv6 Migration*. UvA 2004

[3] IPv6.org - http://www.ipv6.org/

[4] FAQ about the IPv6 protocol for Windows XP - http://www.microsoft.com

[5] RFC 2374 - http://rfc.net/rfc2374.html

[6] RFC 1886 - http://rfc.net/rfc1886.html

[7] RFC 2874 - http://rfc.net/rfc2874.html

[8] RFC 2672 - http://rfc.net/rfc2672.html

[9] IPv6 administration - http://www.microsoft.com

[10] Postfix IPv6 support http://www.postfix.org/IPV6_README.html

[11] UW IMAP - http://www.washington.edu/imap/

[12] xinetd - http://www.xinetd.org