THE IMPLEMENTATION OF IPV6 IN A PRODUCTION NETWORK

Dark Fibre in Amsterdam, Westerpark

By Martijn Paul van Overbeek

THE IMPLEMENTATION OF IPV6 IN A PRODUCTION NETWORK

By Martijn Paul van Overbeek

A thesis submitted to

Hogeschool van Amsterdam 2004

in partial fulfillment of the requirements
for the degree of

Bachelor Degree Computer Science

Presented January 26, 2005
Commencement February 2005

Hogeschool van Amsterdam

Bachelor of Science thesis of Martijn Paul van Overbeek presented on the 26th of January

Approved by _____
Chairperson of Supervisory Committee

_____

_____

_____

Program to Offer Degree _____ Authorized

Date _____

Chairperson of the Supervisory Committee: Jorien Schreuder

Department of Computer Science and Electronic Engineering

A thesis about the approach to implementing IPv6 in a production ISP network

COPYRIGHT

ACKNOWLEDGEMENTS

First of all I would like to thank my girlfriend Jordan for being so supportive and helping me with semantics and spelling. I would also like to thank my parents, Toon and Heleen van Overbeek, for there support during my study. Special thanks go out to my ex colleague and manager of CIPC Mike Janssen. Without his knowledge and adequate handling, I would not have been able to finish the project in the 6 months it took me. Special thanks also go out to Wouter van Diepen and Mark Heersink my companions and competitors during my study at the Hogeschool van Amsterdam. Both of them helped me a lot in finishing my Bachelor in Computer Science. Last but not least I would like to thank my tutor and student counselor Jorien Schreuder for her support during my entire career at the Hogeschool van Amsterdam.

ABSTRACT

We have reached the boundaries of the IP protocol in the present internet. In this thesis the ease of implementing IPv6, the successor of the current IP protocol, in an IPv4 network of an internet provider (CIPC B.V.) is investigated. The thesis is divided in a theoretical and a practical part. In the theoretical part improvements of IPv6 over its predecessor and new features it exhibits are reviewed. The practical part revolves around the actual implementation of an IPv6 network. After reviewing the current network and formulating an IP plan, the implementation is carried out by starting at the lowest layers of the OSI model. The practical part finishes with research on the interoperability between IPv6 and IPv4 and tests that show the difference between a Windows XP machine and a FreeBSD machine. This thesis has shown how promising IPv6 can be as the successor of IPv4. Despite the numerous improvements on the IP protocol in IPv6, the implementation of it could reveal limitations in the existing hardware. However, every ISP should encourage the implementation of IPv6 in its network. With knowledge about IPv6 a company could be a step ahead on its competitors. Also, as IPv6 is still in a developmental stage, embracing IPv6 might be influential on the further development of the protocol.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Number *Page*

PREFACE

In the early 1990s efforts were started to develop the successor of the IPv4 protocol. The most important reason for this was future lack of sufficient IPv4 addresses.

The Internet Engineering Task Force (IETF) started a project called IP next generation (IPng) to work on a new IP protocol. The proposals where examined in 1993 at a convention in Toronto, Canada. Out of all these proposals one was chosen and adopted IPv6. Since then the IETF, along with numerous enthusiasts, were and continue to test and develop IPv6.

In 1996 the IETF started 6bone, an IPv6 test bed network to enable various IPv6 testing as well as to assist in the transitioning of IPv6 into the Internet. In 1999 the Regional Internet Registries (RIR) started assigning IPv6 prefixes to production environments. In 2000 numerous vendors started bundling IPv6 software with their mainstream products. In 2001 Microsoft announced the availability of IPv6 in their latest operating system Windows XP.

Despite the efforts so far, in September 2004 less then 800 companies have an IPv6 prefix assigned to their Autonomous System (AS). With more then a 30,000 AS worldwide and more then a billion internet users this means IPv6 is far from being accepted. To jump on the IPv6 bandwagon now means a technological advantage on the competition. That is why CIPC, my current employer , chooses to implement IPv6. This thesis will describe the process of implementation.

GLOSSARY

RFC. Request For Comments. An RFC is a standard document describing protocols, systems, or procedures used by the Internet community.

Internet Draft. Internet-Drafts are working documents of the Internet community, its areas, and its working groups. An Internet Draft is the preliminary of an RFC.

Node. A device that implements IPv6.

Host. Any node that is not a router.

Router. A node that forwards IP(v6) packets not explicitly addressed to itself.

Switch. A device that directs network packets to the port they are intended for, without broadcasting them to all connections.

Link. A communication facility or medium over which nodes can communicate at the link layer.

Interface. A node's attachment to a link.

Packet. A packet is a standardized unit of data. In network communications a packet generally consists of a "header" with identifying information and a "body" containing the data to be transmitted.

Header. A header is the part of a packet containing information of source and destination of the packet.

POP. A Point-Of-Presence (POP) is an access point from one place to the rest of the Internet.

VLAN. Virtual Local Area Network. A logical subgroup within a local area network that is created with software to facilitate the flow of data to these subgroups.

DOT1Q. Part of the IEEE 802.1 working group. DOT1Q stands for 802.1q and concerns with virtual lans. A DOT1Q taq is a special taq in a link layer frame which places it in a specific VLAN.

CPE. Customer Premises Equipment. The DSL equipment located at the customer premises, either a DSL modem or router for Internet access.

Radius. Remote Authentication Dial-In User Service is a client/server protocol and software that enables remote access servers to communicate with a central server to authenticate dial-in users and authorize their access to the requested system or service.

AS. An Autonomous System is a network or set of networks under one administrative authority such as a company or an organization.

Gateway. A network point that acts as an entrance to another network.

IGP. The Interior Gateway Protocol is a protocol for exchanging routing information between gateways (hosts with routers) within an Autonomous System.

EGP. The Exterior Gateway Protocol is a protocol for exchanging routing information between two neighbor gateway hosts (each with its own router) in a network of autonomous systems.

ARP. Address Resolution Protocol. Internet protocol used to map an IP address to a MAC address.

DHCP. Dynamic Host Configuration Protocol is a method for a device to dynamically assign IP addresses to nodes from a central server.

QOS. Quality of Service. A network device capabilities that provide some guarantee of performance such as traffic delivery priority, speed, latency, or latency variation.

RIR. A Regional Internet Registries are Non-profit organizations responsible for distributing IP addresses on a regional level to internet service providers and local registries

MTU. The Maximum Transmission Unit is a parameter that determines the largest datagram than can be transmitted by an IP interface.

Transit Provider. A transit provider provides a full IP routing table to its customer.

INTRODUCTION

Purpose of this thesis:

During the course of this thesis I want to investigate the process of the implementation of IPv6 in a production environment. The production environment will be the network of an Internet Service Provider.

Thesis Question:

The question I hope to answer is: How straightforward is the implementation of IPv6 in an IPv4 network? Which facets, if any, could be improved to help the implementation?

Thesis goal

The goal is the implementation of IPv6 in an existing IPv4 network. The thesis will be divided in 2 parts, a theoretical part and a practical part. The theoretical part will rely on information from RFCs and the books "IPv6 Essentials" [Hagen, O'Reilly, 2002] and "Implementing Cisco IPv6 Networks" [Desmeules, Cisco, 2003]. The practical part will rely on information from the Cisco Website and again the book "Implementing Cisco IPv6 Networks". The theoretical part will start with an overview of IPv6 and the differences with the former protocol IPv4. In the second chapter I will dive deeper in IPv6 and its header format. The third chapter will deal with the address scheme and different address types. The fourth chapter will complete the theoretical part and deals with configuring IPv6. This chapter also takes notice of the importance of ICMPv6. The practical part starts at chapter five and spins off with describing the network architecture of CIPC. In this chapter the implementation strategy will be formulated as well. In the sixth chapter the IP plan will be formulated. The IP plan will focus on the distribution of IP addresses in the network. The seventh chapter deals with the

physical implementation of IPv6 in the network. This chapter describes the implementation with the OSI layers as a guideline. The last chapter of the practical part will deal with the portability between IPv6 and IPv4. The thesis will end with the conclusion and recommendations.

Chapter 1

IP VERSION 6 OVERVIEW

This chapter starts with defining the IPv6 specifications. This is illustrated by first looking at the shortcomings of IPv4. After this step the solutions for these problems that IPv6 offers will be examined. This chapter ends with looking at some of the drawbacks of IPv6.

IPv4 shortcomings

To illustrate the IPv6 features I briefly want to name out some of the most important shortcomings of IPv4.

Address space.

IPv4 has a limited amount of addresses. 32 bit gives a maximum of 4,294,967,296 addresses. This is less then 2 addresses for every 3 people in the world. On top of this a large bulk of the address space can't be used because of early technical decisions; reserving them for private network and loopback addresses, multicast, and unspecified future uses. As a result of these decisions some of the limitations have been programmed into devices; working around these limitations will require substantial amounts of re-engineering to increase the amount of available address space. Finally, due to the fact that organizations in the United States were very influential in the creation of the Internet and its use of TCP/IP, large IP address blocks were allocated to these organizations.

Routing table.

In the beginning of the internet a company was only able to choose between 3 different IP classes: class A (2^{24} addresses), class B (2^{16} addresses) or a class C (2^{8} addresses). For this reason the routing table of the backbone routers of the

internet where of sheer size. For these reasons Classless Inter-Domain Routing (CIDR) [**RFC 1518**, Rekhter and Li, 1993] was introduced. With CIDR it is possible to assign prefixes from 13 to 27 bits (in CIDR term a /13 to a /27). CIDR also implements hierarchical routing aggregation to minimize routing table entries. This means that a single high-level route entry can represent many lower-level routes in the global routing tables. Despite the efforts the routing table is still rather big in IPv4. A typical Border Gateway Protocol (BGP) routing table contains about 140,000 routes.

Security.

Another drawback of IPv4 is the lack of native security options in the protocol stack. Many add-ons and extensions are available but not part of the native IPv4 stack. This makes IPv4 hosts susceptible to attacks of all kinds.

No auto configuration support.

In IPv4 the nodes are not able to configure a public IP address by themselves. The only way to configure an IP address is by using a DHCP server or by manual configuration.

No quality of service.

Quality of service (QOS) is not part of the standard IPv4 header. It can be implemented but requires all the network devices to be QOS ready. On top of that it is extra overhead on the data packet.

IPv6 the solutions

Address space.

Concerning the limited amount of addresses in IPv4, IPv6 is made out of a 128 bit address space. This means there are $5.7x10^{28}$ addresses for every person in the world. That, off course, is plenty to solve the problem with the limited amount of addresses in IPv4.

Routing table.

IPv6 uses multiple levels of hierarchy. This helps to solve two issues. The great amount of addresses give the Regional Internet Registries, from now on called RIRs, enough address space to effectively assign IP blocks to ISPs. The effective assigning of addresses keeps the routing table relatively small.

Security.

In IPv6 security is natively implemented which means security is part of the IPv6 protocol stack. This means that IPv6 hosts can be made less susceptible to attacks of all kinds.

Autoconfiguration.

The IPv6 node is able to configure itself with help of specific ICMP messages from routers and neighboring nodes.

Quality of service.

QOS is part of the standard IPv6 header. All IPv6 ready devices can automatically handle the IPv6 data packet.

Mobile device support.

In IPv4 the ability to handle mobile support is limited. In its successor the mobile device support is standard.

IPv6 drawbacks

Incompatible with IPv4.

All nodes require a new stack and routers require a new routing engine. Although growing, the stack still has a limited availability,. It is also not always very easy to implement IPv6 support on hardware-based routers and other types of network equipment. Not all hardware has been made IPv6 ready.

Limited amount of software applications.

With software development and IPv6 there is a typical chicken and the egg dilemma. The developers won't develop because IPv6 is not widespread as of yet.

The users will not migrate because not much software has been developed yet. For instance the IPv6 stack of Windows2003 server still has limited functionality.

Synopsis

Despite the fact that a lot of negative issues related with IPv4 have been solved in IPv6, incompatibility with the former protocol and lack of software development will cause defer in the migration process to IPv6.

Chapter 2

IP VERSION 6 IN DEPTH

In this chapter the format of the IPv6 header will be described. The standard header fields will be examined as well as the extension header fields. This chapter will start with the header fields that are no longer present in the new protocol.

Removed header fields

Compared to its predecessor the IP header has undergone some major improvements. Although a standard IPv6 header is larger then a standard IPv4 header, it has been streamlined and 6 header fields are removed. The following header fields are removed:

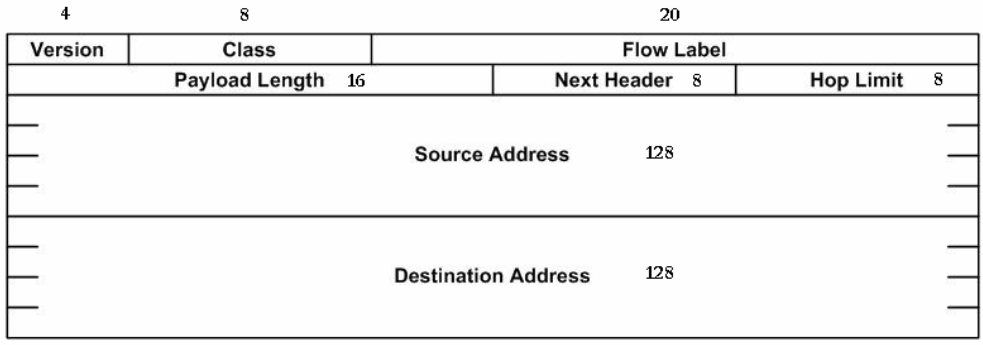
- **Header length**. This field is no longer present in IPv6 because the standard IPv6 header has a fixed size.
- **Header checksum**. This field has been removed because the data link layer and transport layer protocols already have checksums for error detection.
- **Identification, flag and fragments offsets**. These fields handle the fragmentation in IPv4 packets. The IPv4 packets get fragmented if a network does not support the packet size. In the new protocol the packet size is determined before sending of the actual data. This is handled by a process called path MTU discovery (PMTU) where MTU stands for Maximum Transmission Unit.
- **Option and padding**. The option field used to add extra functionality to IPv4 is replaced by extension headers. The option

field was limited by the maximum size of the IPv4 header which is 40 bytes. The padding field, used to line the option header field to 32 bit, is no longer needed.

The IPv6 standard header fields

The standard IPv6 header contains 8 header fields [**RFC 2460**, Deering and Hinden, 1998]. The header has a total length of 40 bytes. Figure 1 shows the IPv6 header. The header is divided in 32 bit parts.

Figure 1: IPv6 header



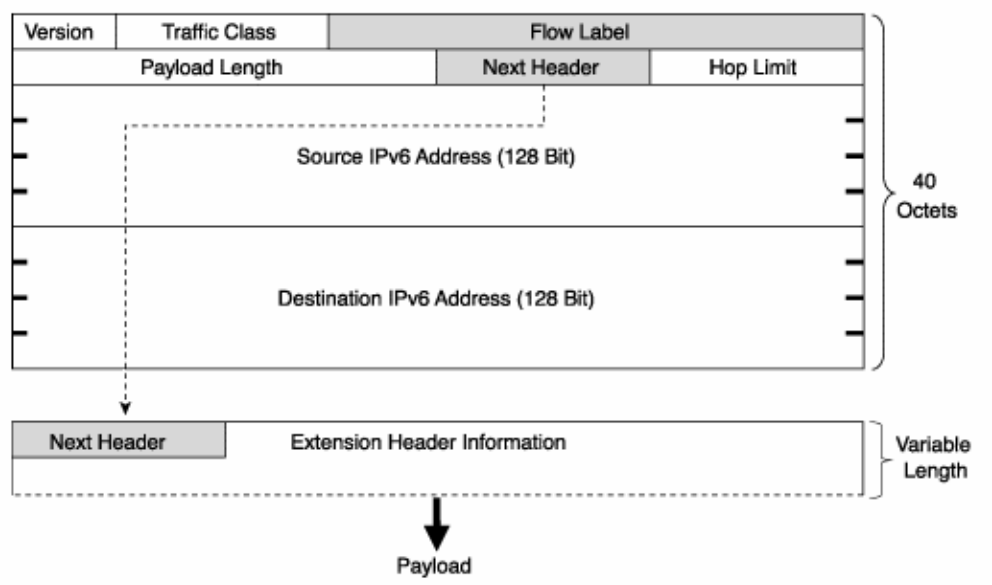
- **Version** is 4 bit in length. It only contains the version number. In the IPv6 header the number is off course 6.
- **Class** of traffic class is an 8 bit field. This field is equivalent to the type of service field in IPv4. It implements quality of service into IPv6. This field tags the packet so it is specified how it should be handled.
- **Flow Label** is 20 bits in length**.** Currently the use of this field is still highly experimental. It is meant to label packets so they can be classified as belonging to a certain flow. The Flow Label notifies routers that the packet needs special handling.

- **Payload length** is 16 bits in length. It contains the length of the IPv6 packet without the standard header in bytes. A simple calculation shows that the maximum payload length is 2^{16} (65535) bytes.
- **Next header** is 8 bits in length. This header field is similar to the Protocol Type field of IPv4. This header gives information about the extension headers attached to the standard IPv6 header. It also tells where the actual *Upper Layer Protocol Data Unit*, TCP or UDP or ICMPv6 starts. It uses the protocol numbering as it is assigned by the Internet Assigned Numbers Authority [IANA]. The protocol numbering can be found in Appendix A.
- **Hop limit** is 8 bits in length. The header field similar to the TTL field in IPv4. When passing through a router the value is decremented by 1.
- **Source Address** 128 bits in length. The originating address.
- **Destination Address** 128 bits in length. The target address for the packet (also 128 bits).

Extension headers

An IPv6 header can contain zero, one or multiple extension headers. An extension header always starts with a next header field. In the IP packet the extension headers are daisy chained (see figure 2) together with the next header fields as is shown in figure 2. The headers have to be in a certain order and all of them have their distinct next header value. Each extension header is 64 bits in length.

Figure 2: IPv6 standard header with daisy chained extension header



Eight extension headers are described in RFC 2460. The following is a brief synopsis of the extension headers:

- **Hop-by-Hop Options header**. If this header is present the first next header field has the value 0. It is the only header which is processed by intermediate routers and nodes. All other headers are only meant to be processed by the destination of the packet. The header field is used for jumbogram packets and router alert messages. Unlike its predecessor IPv6 is able to handle packets greater then 65535 bytes. These packets are called jumbogram packets. This feature is especially helpful in networks with a very large MTU (for instance in gigabit Ethernet environments).The Hop-by-Hop options field makes an MTU possible of 4,294,967,295 bytes. The router alert

feature is used if the destination requires special processing by intermediate routers along the delivery path.

- **Destination Options header**. This header contains information specifically aimed at the destination address. If IPsec headers are available in the IP packet more the one destination options header might be present. It is proposed to be of importance in mobile IPv6. This subject falls out of the scope of this thesis, however information can be found in the related RFCs, such as the RFC: "Mobility Support in IPv6" [**RFC 3775**, Perkins, Johnson et al., 2004].
- **Routing header**. This header can by used by an IPv6 source node if it wants to force a packet to use a specific list of intermediate routers on the way to its destination. As already mentioned earlier, only the Hop-by-Hop Options header may be read by intermediate routers and nodes. If the routing header is used, the final destination address is put as the destination address in the routing header and the first intermediate router is the destination address in the standard IPv6 header. In this way the routing header will be processed by the intermediate routers and the selected routing path will be used.
- **Fragment header**. Fragmentation by routers is not permitted in IPv6. As already mentioned briefly, the process PMTU adjusts the MTU. The node will only send packages that can be handled by the intermediate routers. In a case that the node needs to send a package bigger then the determined PMTU the fragmentation header is used. The packages are adjusted in size and numbered with the fragmentation header.
- **Authentication header**. This header is used in IPsec to provide authentication, data integrity and replay protection. It is identical to the header in IPv4.

- **Encapsulation Security Payload header**. This header provides the same features as the Authentication header and adds to that confidentiality.

Synopsis

Now that the new header has been reviewed and some the features of IPv6 are revealed it becomes clear that IPv6 has some mayor improvements over IPv4. The streamlined header and use of the extension headers make IPv6 more flexible then its predecessor.

Chapter 3

THE IP VERSION 6 ADDRESSING

This chapter will start with describing the form and notation of an IPv6 address. After that the different address types will be described.

IPv6 address form

An IPv6 address consists of 128 bits. The address is divided in 8 blocks of 16 bits. Because of the size of each block, the form has changed from a decimal form to a hexadecimal form. Every block has values ranging from 0000 (binary value: 0000 0000 0000 0000) to FFFF (binary value: 1111 1111 1111 1111). The IPv6 address can be written in different forms [**RFC 3513**, Deering and Hinden, 2003]. The three most common forms will be discussed in this chapter. The forms are presented in table 1.

Table 1: IPv6 address notation

IPv6 address notation	Notation
default	3ffe:2000:0102:3432:0000:0000:000f:fe10
compressed (leading 0s)	3ffe:2000:102:3432:0:0:f:fe10
compressed (16 bits fields)	3ffe:2000:0102:3432::000f:fe10
compressed (mixed)	3ffe:2000:102:3432::f:fe10
with embedded IPv4 address (compatible)	::192.168.0.254
with embedded IPv4 address (mapped)	::ffff:192.168.0.254
with embedded IPv4 address (compatible IPv6 default)	::c0a8:00fe

Default representation

An IPv6 address can be written in different forms. In its default form it contains eight 16 bit fields completely filled with hexadecimal values and separated by colons.

Compressed representation

It is common for IPv6 addresses to contain long strings of 0. To make reading and writing easier, IPv6 addresses can be written in a compressed form. Two forms of consecutive 0 values can be compressed: Successive 16 bit fields filled with 0s and leading 0s in 16 bit fields can be compressed. Consecutive 16 bits fields filled with 0s can be replaced with :: (double colon). Only one :: is permitted per IPv6 address. Leading 0s in 16 bit blocks can be removed to simplify the address however one 0 must remain per 16 bit field. A combined form is also possible.

IPv6 with an embedded IPv4 address

A 32 bit IPv4 address can be embedded in a IPv6 address. In this notation the first part, consisting of 96 bits, is written in hexadecimal notation and the second part, the last 32 bits, is written in decimal notation. It is also possible to convert the IPv4 address in a hexadecimal notation. Two versions of embedded IPv4 addresses in an IPv6 address are used. These versions are:

- IPv4-compatible address
- IPv4-mapped address

The IPv4-compatible address is used to establish an automatic tunnel to carry IPv6 packets over an IPv4 network. The address consists of 96 bits of 0s with the IPv4 address attached to it.

The IPv4-mapped address is only used locally by a node which has both an IPv4 and an IPv6 address. The address consists of 80 bits of consecutive 0s followed by 16 bits with the value 1. The IPv4 address is attached to this.

IPv6 Subnetting

As in IPv4, an IPv6 address has a subnet part and an address part. The address follows the CIDR notation. This means an IPv6 address can be divided in an address part which is also known as the interface-ID and the prefix-length. An example of an address with prefix is:

Example: 3ffe:2000:0102:3432:0000:0000:000f:fe10/64

In the example the actual prefix is:

3ffe:2000:0102:3432::/64

The "/64", in the example, means that the first 64 bits are set and can not be changed. In the example 64 bits can be used for the interface-ID. In general you can say that an IPv6 address contains a subnet of "n" bits and a interface-ID of 128-n bits.

IPv6 address types

IPv6 addresses are identifiers for interfaces. There are three different types:

- **Unicast**: An identifier for a single interface. A packet sent to a unicast address is delivered only to the interface identified by that address.

- **Anycast**: An identifier for a set of interfaces (typically belonging to different interfaces). A packet sent to an anycast address is delivered to one of the interfaces identified by that address (the "nearest" one, according to the routing protocols' measure of distance)

- **Multicast**: An identifier for a set of interfaces (typically belonging to different nodes). A packet sent to a multicast address is delivered to all interfaces identified by that address.

In comparison with IPv4 the broadcast is no longer present. It has been replaced by multicast. Both unicast and multicast contain many subtypes which all can be identified by their own bit pattern (see table 2). These two address types will be reviewed in more detail.

Table 2: Address type identification

Address type	Binary prefix	IPv6 notation
Unspecified	00...0 (128 bits)	::/128
Loopback	00…1 (128 bits)	::1/128
Multicast	11111111	FF00::/8
Link-local unicast	1111111010	FE80::/10
Site-local unicast	1111111011	FEC0::/10
Global unicast	(everything else)	

Unicast

The unicast address may or may not have an internal structure. With an internal structure is meant that the address contains a subnet prefix from the link it is attached to. There are several types of unicast addresses. The following address types can be described:

- **Unspecified address**: The address 0:0:0:0:0:0:0:0 (or shortened ::) is called unspecified. It may never be assigned to any node. This address is used as a source address of an IP packet of an initializing host which hasn't learned its own address yet.
- **Loopback address**: The address 0:0:0:0:0:0:0:1 (or shortened ::1) is called the Loopback address. This address is used by a node to send packages to itself. Mostly this address is used for testing purposes in software development.
- **OSI network layer ported addresses**: IPv4 addresses can be embedded in IPv6. It is also possible to embed the ISO NSAP addresses and IPX in

IPv6. Prefixes for the latter two are already defined however, the draft definition, motivation and usage are still under study.

- **Aggregatable Global Unicast Addresses**: These addresses represent the bulk of the IPv6 address space. Their function can be considered similar to the public IPv4 addresses.
- **Link local addresses**: Link local addresses are addresses which can only be used in the context of the link it is attached to. For example a direct cross link between a router and a host. Every IPv6 enabled interface automatically gets one Link local address assigned to it. Link local addresses play an important role in auto configuration and neighbor discovery, two mechanisms which will be described in the next chapter.
- **Site local addresses**: Site local addresses can be considered similar to the private IP ranges used in IPv4. These addresses are not assigned by default. Site local addresses are only meant for traffic within sites itself. They may never be routed to the internet. Although the Site local addresses are similar to IPv4 private ranges NAT is considered undesirable and while writing this thesis Site Local addresses became deprecated[**RFC 3879**, Huitema and Carpenter, 2004].

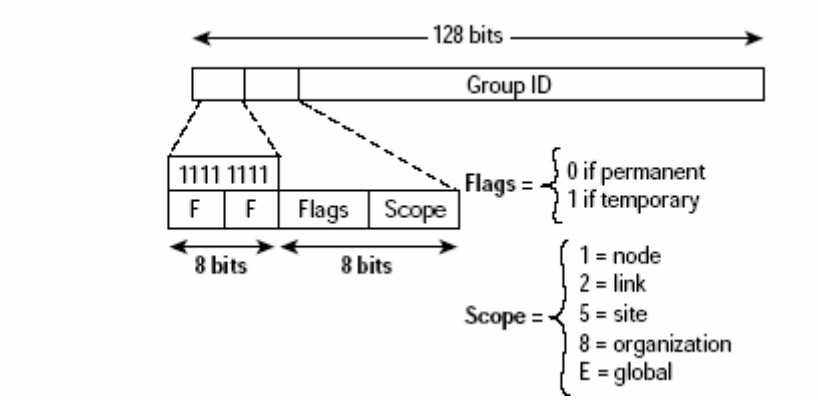
Multicast addresses

Multicast means sending a packet to multiple destinations. The main goal of multicast is to use the network bandwidth efficiently. It is used for many purposes in IPv6 among which auto configuration and neighbor discovery are two examples.

A multicast address can be divided in 4 parts, figure 3 illustrates this. The first part consists of 8 bits which are all set to 1. The second part, 4 bits long, is called the flags. It can have two hexadecimal values either 0 of 1. The flag indicates if the address is a permanent multicast address, assigned by IANA or a non-

permanent multicast. The third part is called the scope. This part determines the network boundaries to which a multicast packet can travel. E.g. a node scope multicast means the packet may not leave the node and a link local node may not leave the link where it and other nodes are attached to. The group ID also plays an important role in multicast addresses. The Group ID value determines if it is a reserved address. These addresses, also known as *multicast assigned addresses*, play an important role IPv6. All nodes and routers are instructed in their IPv6 stack to recognize these addresses. A list of all the multicast assigned addresses can be found in Appendix B.

Figure 3: The multicast address format



Synopsis

An IPv6 address can be presented in different forms. There are three address types which can be subdivided in extended different types. Especially the unicast and the multicast have subtypes which play an important role in the IPv6 protocol stack. In the next chapter more of that will be revealed.

Chapter 4

IP VERSION 6 CONFIGURATION

This chapter starts with a brief introduction to ICMPv6. This will be the bridge to the principal parts of this chapter the Neighbor Discovery Protocol and autoconfiguration of nodes using stateless autoconfiguration.

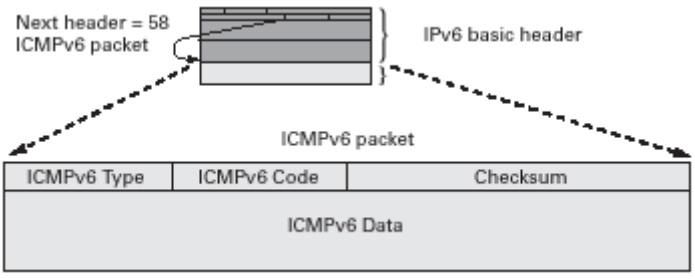
ICMPv6

The Internet Control Message Protocol (ICMP) is used to give information about the health of a network. ICMPv6, the successor of ICMP, is used in IPv6 and contains a lot of new features [**RFC 2463**, Conta and Hinden, 1998]. It is heavily used and lies at the foundation of most important improvements of IPv6. This chapter will show how ICMPv6 is used in some of these improvements.

Message format

An ICMPv6 packet is attached to an IPv6 header and its extension headers. It is identified by next header "type" 58. An ICMPv6 packet can be split up in four parts, as is shown in figure 4 and described on the next page.

Figure 4: ICMPv6 packet



- **Type**: This 8 bits field defines the type of message. There are two classes of messages in ICMPv6; error messages and informational messages. The class is decided by the highest order bit. A '0' means it is an error message and a '1' means it is a informational message. This means 128 error types can be specified and 128 informational types can be specified as well. Not all 256 types are defined yet.
- **Code**: The code field depends on the type field. It contains extra information related to the type field.
- **Checksum**: This field contains a computed value that is used to detect data corruption during transport.
- **ICMPv6 data**: This field is not always used in ICMPv6. When used it contains information important to the destination node.

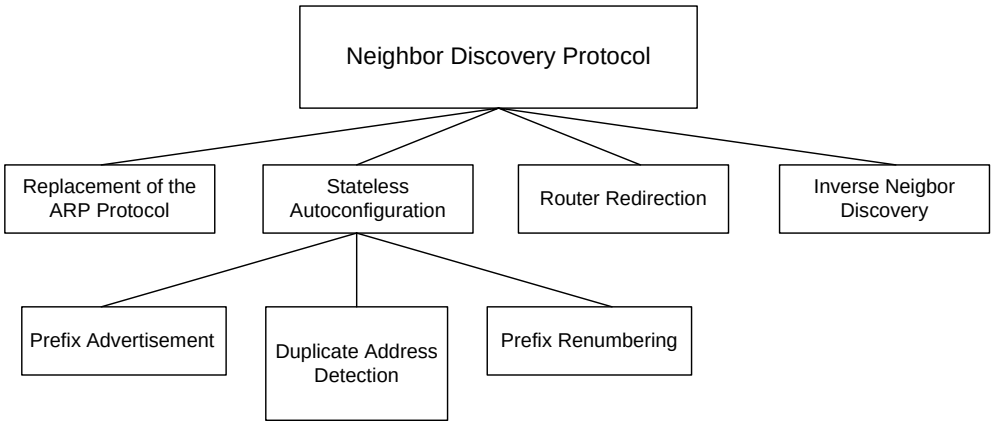
Neighbor Discovery Protocol

The Neighbor Discovery Protocol (NDP) [**RFC 2461**, Narten, Nordmark et al., 1998; **RFC 3122**, Conta, 2001] is a complete new protocol only available in IPv6. It combines tasks previously performed by protocols in IPv4 and adds to these new mechanisms. For its mechanisms NDP makes heavy use of all kinds of ICMPv6 message types and multicast. The purpose of the protocol is to determine the link-layer addresses for neighbors known to reside on attached links and to quickly purge cached values that become invalid. NDP is also used by hosts to find routers that are willing to forward packets on their behalf. Finally the protocol is also used by hosts to keep track on changes in link-layer addresses of neighboring nodes. When a router or the path to the router is interrupted the node will use NDP to actively search for alternatives.

Neighbor Discovery Protocol dissection

NDP can be divided in to four separate basic mechanisms. From these four mechanisms Stateless Autoconfiguration will receive extra attention later in this chapter. Figure 5 illustrates the mechanisms part of NDP:

Figure 5: Neighbor Discovery Protocol



- **Replacement of the Address Resolution Protocol (ARP)**: ARP, the mechanism to detect neighbors in IPv4 has been removed from IPv6. A new mechanism, that uses a mixture of ICMPv6 messages and multicast addresses, determines the link-layer address on the local link of nodes.

- **Stateless Autoconfiguration**: This mechanism allows nodes on the local link to configure IPv6 addresses themselves. It uses a mix of ICMPv6 messages and multicast addresses.

- **Router redirection**: The router sends ICMPv6 messages to inform a node that a better router address is available on the same local link to reach its final destination

- **Inverse Neighbor Discovery**: The mechanism in which a node sends information about its IPv6 addresses in response to ICMPv6 messages

on the local link. This mechanism uses a mix of ICMPv6 and multicast addresses.

Configuring IPv6

Like IPv4 there are two methods to configure an IPv6 address: Manual configuration and autoconfiguration. Manual configuration is mainly used for servers and routers and is not of importance in this chapter. There are three different methods of autoconfiguration namely: Statefull autoconfiguration, stateless autoconfiguration and a mix between these two. In Statefull autoconfiguration a DHCPv6 server is used to configure nodes. This chapter will for the most part deal with stateless autoconfiguration as that is where the real power of IPv6 is revealed.

Autoconfiguration

Stateless autoconfiguration is designed with the following goals in mind:

- Individual machines should no longer require manual configuration.
- Small sites should no longer require a Statefull server as a prerequisite for communication
- Hosts in large networks should be able to obtain the route and prefix without Statefull autoconfiguration.
- Hosts should have the ability to renumber their addresses

In the process of autoconfiguration the node will lease an address for a certain lifetime. At the end of its lifetime the address will become invalid to the node. An IP address can have 3 different stages:

1. **Tentative address**: This is an address that has not yet been assigned

2. **Preferred address**: This is the address that has been assigned to an interface and that can be used without any restrictions.

3. **Deprecated address**: The use of this address is discouraged but not forbidden. A deprecated address might be one whose lifetime is about to expire. It can still be used to continue an already opened communication.

The process of autoconfiguration can be described as a two phase process. In the first phase the node will try to generate a link local address. The link local address is formed by appending the interface-ID to the link local prefix. Before the address can be used by the node it must be verified by that node that the address in not already in use. This is done by sending special ICMP messages called *neighbor soliciting messages* (NS). These messages are sent using the *solicited-node* multicast address (see Appendix B for a list of the multicast address groups) as the destination address and the unspecified address as the source address of the IPv6 packet. The ICMPv6 data field contains that tentative address. If any node already uses that address it will send an ICMP message that is called *neighbor advertising message* (NA). This process is called Duplicate Address Detection (DAD). Either the IP address has to be manually configured or it is unique on the local link and fully functional.

The next phase of autoconfiguration involves determining the type of autoconfiguration that should be used. The node will start with sending ICMPv6 messages called *router soliciting messages* (RS). In these RS messages the destination address of the IP header is set to the all router multicast. If a router is present and configured to respond to these types of messages, it will send *router advertisement messages* (RA) as a response. Else Statefull autoconfiguration should be invoked. A router advertisement message contains the information that is needed by the node. This information can be subtracted from the ICMPv6 data field. The following information can be obtained from a router advertisement message:

- Information about the type of Statefull autoconfiguration (if any) that should be used.
- Information whether or not to use stateless autoconfiguration.
- If the RA applies to autoconfiguration the prefix information and lifetime of the address are submitted in a special option field.

When a node has received a RA it will process the message and will either generate a site local address, a global unicast address or both.

As is already mentioned, the node will also incorporate lifetime information. This is especially important in a process called site renumbering. Because of the lifetime information the ability exists of an address to time-out. Before it times out it will go through the stages of being a preferred address to a deprecated address and eventually an invalid address. If the router advertises new prefixes the node will have new preferred addresses. The ability to carry preferred and deprecated addresses at the same time will help the site renumbering to work smoothly. It will also give upper-layer protocols such as TCP an UDP the chance to change their communication to a host to its new address.

Synopsis

ICMP messages play an important role in IPv6 and especially in the Neighbor Discovery Protocol. Part of the protocol is a process called stateless autoconfiguration, a process that adds great flexibility in the auto configuration of IPv6 nodes. Due to this process nodes have the ability to configure themselves without a managed server (DHCP server). The incorporation of lifetime information in IPv6 nodes helps in renumbering processes.

Chapter 5

THE CIPC NETWORK

This chapter will start with documenting the entire network of CIPC. After the network has been documented the focus will lay on the implementation strategy of IPv6 in the network.

CIPC

To illustrate the entire network of CIPC the first thing that should be looked at is the business CIPC offers to its customers. The core business of the company is connectivity in a broad sense. CIPC specializes in layer 2 VPN solutions (Crossnet) and security. The Crossnet will be described in more detail later in the chapter. CIPC also carries regular services like DNS, hosting and mail.

The network

Figure 6 gives a detailed look at the CIPC network. The network consists of three POPs that are connected with each other by an Ethernet star network. The POPs are: TeleCity, GlobalSwitch and CyberCenter. The POPs have more or less the same architecture. Layer 2 service providers deliver customer connections to the Subscriber Management System (SMS) of CIPC. The SMS can be seen as the bridge between layer 2 virtual circuits and layer 3 addresses. It authenticates the customer connections at a specific radius server and if authenticated, a layer 3 address is applied to the customer and aggregated through the core routers.

POPs

Each POP has its own server park containing radius authentication servers, DNS servers, syslog servers and timeservers. Other services are more specific to each POP.

- TeleCity: Two IPv4 transit providers are connected in the switch. Three layer 2 service providers are connected. Two of those are ATM and one is Ethernet. Also connected to this site are firewalls for private WANs and IPv4 transit customers.

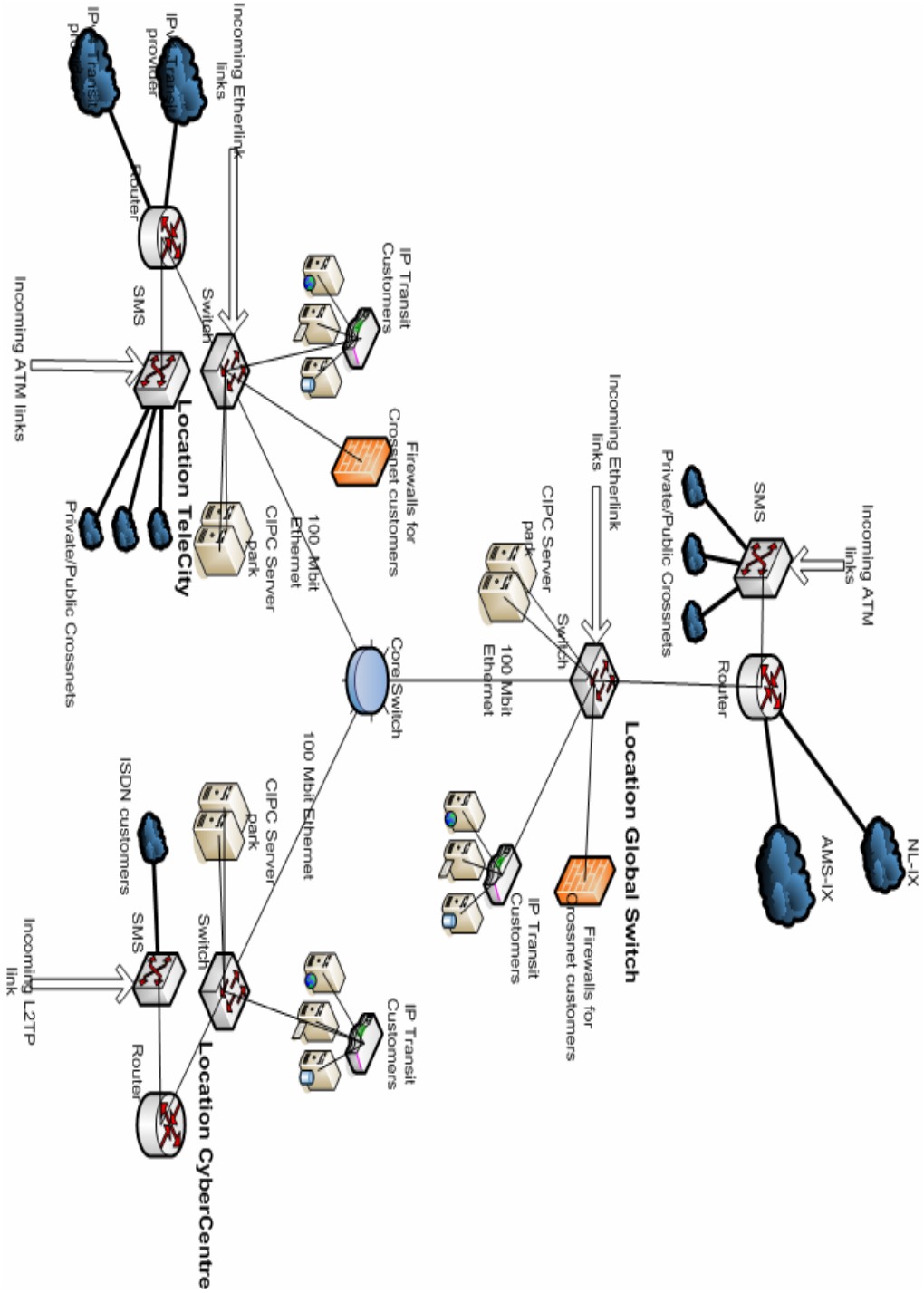
- GlobalSwitch: This POP harbors the connections with the NL-IX and the AMS-IX. Two layer 2 service providers are connected. One of those is ATM and one is an Ethernet provider. Firewalls for private WANs and IPv4 transit customers are connected as well.

- CyberCenter: This is the smallest POP of the three. It contains one layer 2 connection and only connects the ISDN customers. It also provides transit to some customers. On top of that it contains a backup transit line but this line is insufficient as a backup solution for all the customers.

Crossnet

Crossnet is a unique VPN solution used by CIPC. It differs from solutions of its competitors. The Crossnet solution makes it possible to connect various layer 2 techniques in a single VLAN. Because the techniques are connected at layer 2 the VLAN that is created is completely separate from the internet. The Crossnet solution creates no extra overhead. This increases the performance compared to IPsec solutions over the internet. These solutions add extra overhead to the IP header. Examples of these solutions are Multiprotocol Label Switching (MPLS) and IPsec using Encapsulated Security Payload (ESP).

Figure 6: The CIPC Network



Implementation strategy

The implementation can be divided in two parts. The first part is getting the CIPC Core network IPv6 ready. The second part will be bringing IPv6 to the customers. To get the Core network IPv6 ready the OSI model will be followed. This means the implementation starts at layer 2 (link layer) and finishes if layers 4 to 7 transport/session/presentation/applications) work well with IPv6. Looking at this "layer by layer", implementation of IPv6 basically is a three steps process:

- **Step 1: Making the network Layer 2 ready (link layer):** This layer concerns the Ethernet switches CIPC uses. Basically all the devices in the Core network are attached to them. Although these devices are called layer 3 switches the actual usage is at layer 2. The switches are concerned with creating 802.1Q (DOT1Q) VLANs which plays a mayor role in the CIPC network architecture. Compatibility with the IPv6 Ethernet frames is vital.
- **Step 2: Creating Layer 3 connectivity (network layer):** To further propagate IPv6 through and outside the CIPC network the routers must be IPv6 enabled. Most (backbone) routers only need minor adjustments and software upgrades before being IPv6 ready. After enabling IPv6 a transit provider need to be contracted to obtain the full routing table. At the same time the peering contracts can start to encourage free traffic between IPv6 providers.

- **Step 3: Enabling basic ISP services on Layer 4 to 7 (transport to application layer):** Full connectivity in the Core network is established if applications like DNS, timeservers, syslog servers and radius servers will be prepared to work with IPv6. As a minimal condition at least DNS should be IPv6 enabled.

To bring IPv6 to the customer's two approaches are made. The first approach is testing our SMS and radius authentication system for being capable of handling IPv6 packets. If both are capable, all customers can have native IPv6 subnets routed to their Customer Premise Equipment (CPE). This means all IPv6 transport within the CIPC network is carrier out using IPv6 packets only, no encapsulation of IPv6 packets has to be used. The second approach will deal with the possibility of tunneling IPv6 subnets over IPv4. This approach will only be viable if the first one fails.

Synopsis

In three steps IPv6 connectivity is possible in the CIPC Core network. Further research on the SMS and radius will reveal how connectivity on CPE's can be established. Before the actual implementation takes place an IP plan will be presented. This is the subject of the next chapter.

Chapter 6

IP PLAN CIPC

In this chapter the IP plan will be defined. An IP plan concerns with the division of the assigned IPv6 address space. After a short introduction to RIPE the IPv6 plan is presented. This plan is created in accordance with the guidelines as formulated by the IETF and RIPE itself.

RIPE and its jargon

RIPE is the abbreviation of Réseaux IP Européen which means European IP network. RIPE is a non profit organization and the RIR responsible for ISPs in Europe and parts of Africa. The IP plan, presented in this chapter, will follow rules and policies as formulated by RIPE [**ripe-267**, Inomata, 2003]. To get familiar with the jargon used by RIPE, some terms need to be explained:

- Allocation: To allocate means to distribute address space to Internet Registries for the purpose of subsequent distribution by them.
- Address aggregation: The distribution of IP blocks.
- Address Assignment: The delegation of global IPv6 subnets to customers.
- HD-ratio:

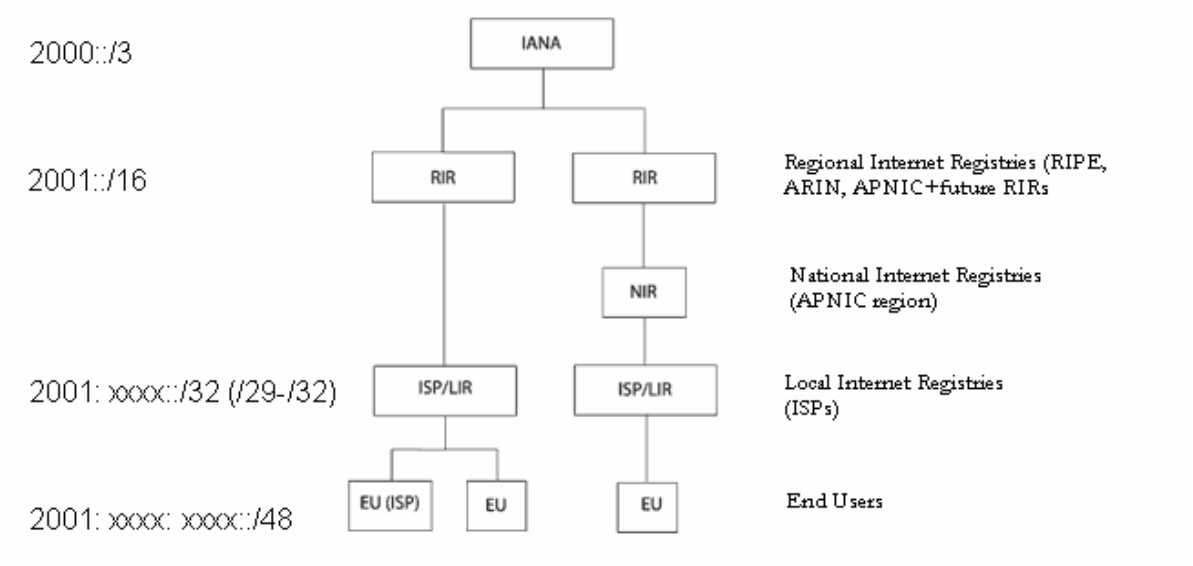
$$\text{HD-ratio} = \frac{\text{Log (number of assigned /48 subnet to custs.)}}{\text{Log (number of total /48 subnets)}}$$

IPv6 address management

Allocation process

The IETF has designated 2000::/3 as the initial global unicast space that Internet Assigned Numbers Authority (IANA) may allocate to the RIRs. The allocation of address blocks is a hierarchal process. The process begins with the allocation of address blocks by IANA to the RIRs and end with the assignment of IPv6 subnets from Local Internet Registries (LIR) or ISPs to end users. Figure 7 illustrates this process.

Figure 7: The allocation process



Global unicast space

The first IP block, which has been designated for allocation, is 2001::/16. From this block RIRs have been assigned 2001::xxxx::/23 subnets where x is a hexadecimal value (the assignment table can be viewed in Appendix C for reference).

Allocation policy

In the allocation policy of IPv6 address blocks RIPE is targeted to be uniform to all LIRs. The size of the internet provider determines the address space it gets allocated. The minimal allocation size is /32, however RIPE will reserve a /29 for each of its LIRs. Due to the reservation of a /29 each LIR can expand its address space by extending its subnet. A LIR should for the most part assign a /48 subnet to its customers. This means that a LIR can roughly assign 65,000 customers per /32. New space gets assigned to the LIR if a HD-ratio has reached the value 0.8. With the first assigned /32 this takes about 8,000 customers.

Assignment policy CIPC

The main priority in designing an IP plan is transparency. Predefined rules are a necessity. The focus will lay on defining easy interpretable bit patterns. To assist the memory, mnemonic methods are used to assign addresses to the CIPC server park. The assignment of IPv6 addresses to CIPC customers can be divided in two parts:

- Dividing the block on a POP basis
- Assigning the addresses on customer demand.

Dividing the block op a POP basis

At the moment CIPC is connected at 3 POPs: TeleCity, GlobalSwitch and CyberCenter. All broadband customers are connected in TeleCity. The CyberCenter carries the ISDN dial up connections and might be phased out in the future. GlobalSwitch carriers no customers at the moment, it is prepared however, to connect broadband connections as well. To assign the addresses to the customers the address block will be divided into eight parts. To keep it as easy as possible each POP will receive two /35 subnets (see table 3). Notwithstanding, there is no reason for a more difficult plan due to the HD-ratio.

Table 3: Address allocation to POPs

IPv6 Prefix	NLA ID Binary Values	Allocated to
2001:40e0:0000	000x xxxx xxxx xxxx	TeleCity
2001:40e0:2000	001x xxxx xxxx xxxx	TeleCity
2001:40e0:4000	010x xxxx xxxx xxxx	GlobalSwitch
2001:40e0:6000	011x xxxx xxxx xxxx	GlobalSwitch
2001:40e0:8000	100x xxxx xxxx xxxx	CyberCentre
2001:40e0:A000	101x xxxx xxxx xxxx	CyberCentre
2001:40e0:C000	110x xxxx xxxx xxxx	Future assignment
2001:40e0:E000	111x xxxx xxxx xxxx	Future assignment

Assigning the addresses on customer demand

The assignment of IP addresses to customers will follow the guidelines defined by [**RFC 3177**, Baker, Carpenter et al., 2001]. These guidelines include the following:

- /48 in the general case, except for very large subscribers
- /64 when it is known that one and only one subnet is needed by design
- /128 when it is absolutely known that one and only one device is connecting.

CIPC implementation

- All DSL customers will receive a /48 subnet.
- Co-located customers that will use only one subnet will receive a /64 subnet.
- Co-located customers which only have one device co-located and that are not planning to expand this in the future will receive a /128

All of these three subnets, mentioned above, will be implemented, per POP, in a /35 subnet. Each /35 contains 8192 /48 subnets. If the 48th bit is set to 1 the subnet will be used for the /64 subnet. If the 64th bit is set to 1 the subnet will be used for /128 customers. Table 4 illustrates the division process.

Table 4: Division of a /35 prefix

General Division /35 subnets Hexadecimal View (z=hex. Value 0,2,4,6,8,a,c,e)	
Begin	**End**
2001:40e0:z000::/48	2001:40e0:(z+1)ffe::/48

General Division /35 subnets Binary View (bit 33 to 48) (x=bin. value)	
Begin	**End**
xxx0 0000 0000 0000	xxx1 1111 1111 1110

General Division /64 subnets Hexadecimal View (z=hex. Value 0,2,4,6,8,a,c,e)	
Begin	**End**
2001:40e0:(z+1)fff:0000	2001:40e0:(z+1)fff:fffe

General Division /64 subnets Binary View (bit 49 to 64) (x=bin. value)	
Begin	**End**
0000 0000 0000 0000	1111 1111 1111 1110

General Division /128 subnets Hexadecimal View (z=hex. Value 0,2,4,6,8,a,c,e)	
2001:40e0:(z+1)fff:ffff:0:0:0:0:/128	2001:40e0:(z+1)fff:ffff:ffff:ffff:ffff:ffff:/128

Assigning to the CIPC Core Network and server park

The very first IPv6 /48 IP block will be assigned to CIPC. This block is partly meant for the Server Park and Core Network. To make the network easy interpretable the following steps are taken:

- Assigning each POP a /60 for the infrastructure

- Assigning a /60 for all multi POP spanning VLANs
- Within each /60 subnet the division of blocks should be the same e.g. the loopback of the router always has the same hexadecimal value.
- Create recognition by attaching the last two hexadecimal values to a certain network device/service.
- CIPC servers will receive a Site Local address for management purposes that is

The complete IP plan can be found in Appendix D.

Synopsis

Address aggregation and assignment of IPv6 blocks to customers goes in accordance with the documents written by RIPE and the internet community. Using well defined bit patterns and mnemonic methods of addresses helps to make IPv6 addresses easier to read.

Chapter 7

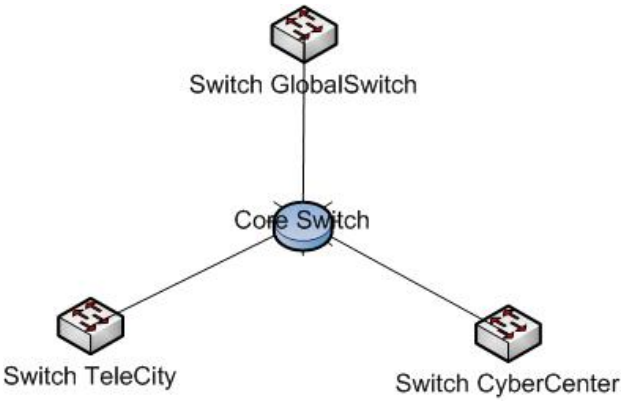
CORE IMPLEMENTATION

In this chapter the experiences with the implementation of IPv6 in the core network will be presented. This chapter consists of three parts. The first part deals with the layer 2 implementation. The second concerns with the layer 3 implementation results will be presented. The chapter ends with the results of implementing IPv6 on layers 4 to 7.

Layer 2 Implementation

At layer 2 the Core network is connected by four switches. Figure 8 illustrates this. The figure shows that the network resembles a star; all three switches in the POPs are connected to one core switch. By acknowledging the difference in IPv6 Ethernet frames and doing manufacturer research, the CIPC network eventually accomplishes layer 2 availability.

Figure 8: Switched Network CIPC



The CIPC IPv4 network is built on multiple VLANs. To come to a successful implementation of IPv6 the following has to be stated:

- All switches should be IPv6 ready
- Implementation of IPv6 should not have any impact on the existing IPv4 network

Ethernet frames

Ethernet frames are a layer 2 medium to transport IP packets. IPv6 packets are transmitted in standard Ethernet frames. The Ethernet header contains the Destination and Source Ethernet addresses and the Ethernet type code, which, in the case of IPv6, must contain the hexadecimal value 86DD [**RFC 2464**, Crawford, 1998].

Manufacturer research

To implement IPv6 at CIPC all four switches should be able to process the IPv6 specific Ethernet type. Off course there should be no impact on the existing VLANs. Two switches, out of four, were not able to handle IPv6 packets. Only one switch could be made IPv6 ready with a firmware upgrade. The two others had to be replaced to make the whole core IPv6 ready.

Layer 2 ready

The core was considered layer 2 ready when all four switches where able to handle IPv6 packets. It took us several weeks to carefully test and conduct the implementation. The testing involved the cloning of the switch configuration to a test situation in a test lab. All VLANs where tested and checked to verify if IPv6 packets could be sent over the switch.

Layer 3 implementation

The layer 3 implementation consists of enabling IPv6 at our three Core routers. After physically enabling IPv6 on the Core routers, the routing protocols BGP (Border Gateway Protocol) and OSPF (Open Shortest Path First) for IPv6 will be implemented. The layer 3 implementation will be conducted in three steps

- Making all three Core routers IPv6 ready
- Implementing the exterior gateway protocol BGP on the Core routers
- Implementing the interior gateway protocol OSPF

Software

All three Core routers needed to be updated to a new software version to be able to handle IPv6 packets. Due to the larger size of the software, all routers where equipped with flash disks containing the new software version. The new software version had no impact on the running configurations in the Core routers.

IP plan implementation.

When all routers where upgraded they where configured with IPv6 addresses according to the IP plan. IPv6 was enabled on the Core routers on a global basis. The general philosophy for Core Networks is to assign static IP addresses to sub interfaces of the routers. These sub interfaces, which implement dot1q, communicate with other network devices in separate VLANs. The entities in these VLANs are all directly connected to the Core network. For this reason Neighbor discovery, a feature only found in IPv6, is disabled in the Core network.

BGP implementation

CIPC uses BGPv4+ on all Core routers to obtain the full IPv4 routing table. This version of BGP is capable of handling IPv6 as well. To implement BGP on all routers the following checklist should be completed:

- Check if memory availability on all three routers is sufficient to handle the IPv6 full routing table
- Building filters (prefix-lists)
- Differentiating of routes
- Setting up internal BGP (iBGP)

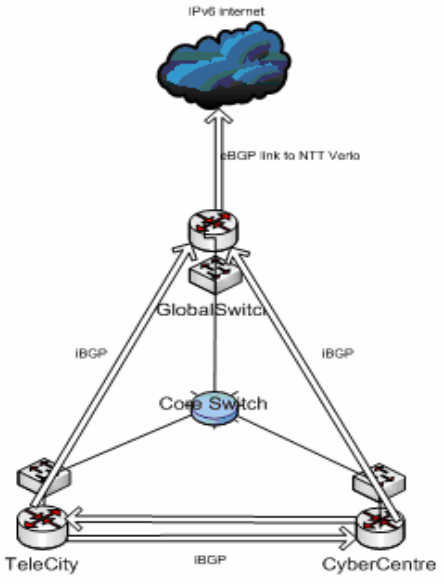
All three core routers had about 30 Megs of free memory. Around 50% of the total memory usage was consumed by the IPv4 BGP table. This table contains about 160,000 routes. At the time of writing there are about 700 companies worldwide that have an IPv6 block allocated. This means it is safe to assume the IPv6 routing table can easily fit in the 30 Megs of free memory.

The GlobalSwitch router was chosen as the POP for the interconnect with an IPv6 transit provider, namely NTT Verio. Before setting up a exterior BGP (eBGP) peer with NTT Verio a prefix list was made for incoming and outgoing routes. This list made sure only valid routes equal to or smaller than a /48 are accepted and only our own network prefix was advertised. The prefix-lists can be found in Appendix E.

After the full routing table was acquired, peering agreements were made with all parties connected to the AMS-IX. AMS-IX traffic is considered to be free. For this reason special filters where built, that are called route maps, to distinguish this traffic from the "expensive" traffic to NTT Verio [Döring, 2004].

iBGP was implemented on TeleCity, CyberCentre and GlobalSwitch to make the full routing table available on all three Core routers. After the implementation was successfully completed around 550 IPv6 routes where received from NTT Verio. Figure 9 shows the implementation of eBGP and iBGP. The direction of the arrow points to the device that provides the full routing table.

Figure 9: BGP implementation in the Core network



OSPF implementation

The Core network of CIPC has OSPFv2 implemented as the underlying routing protocol. The advantage of an underlying routing protocol is that no static routes are needed in the Core routers. Static routes can easily cause routing problems when mistakes are made with them or when links go down. OSPFv2 for IPv4 consumes around 8% of the available memory in the Core routers. IPv6 uses OSPFv3[Cisco, 2003]. This version can operate along with OSPFv2 on a router. This means there is no impact on the OSPF v2 routing table. Due to the initial setup of IPv6 and the comprehensive IP plan the impact of implementing

OSPFv3 will be limited. Less the 0.2 % extra of the memory was consumed by the core routers.

Layer 4-7 implementation

The layer 4-7 implementation merely focuses on providing DNS and web services on IPv6. For an internet provider those are the primary necessities.

Experimental server

None of the production servers were considered suitable to function as a web and or DNS test server. Important reasons are that production servers should not be used for testing purposes. On top of that none of the servers where IPv6 ready.. Therefore the strategy was chosen to install a new server that would function as a DNS and web server.

Bind 9

Berkeley Internet Name Domain (BIND) is an implementation of the Domain Name System. It is used on the vast majority of Name Servers on the Internet. Since version 8 [Hensema, 2004] it offers limited support for IPv6, however it is not able to listen to IPv6 data for querying. Version 9 of the program is fully featured for IPv6.

Because IPv6 addresses differ greatly from IPv4 addresses some additions were made to DNS. Two important additions will be reviewed:

- AAAA (Quad A) records
- Reverse lookups

AAAA records

These records are the IPv6 counterparts of A records. An A Record, short for Address Record, maps an IP address to a domain name. The result of an answer to a query can be seen on the next page.

Quad A record:

www IN AAAA 2001:40e0:10:a10::1

Reverse lookups

The main difference with its IPv4 counterpart is that delegation is set to 4 bits instead of 8 bits, which means a dot per hexadecimal value. The second difference is that the zone is part of ip6.arpa instead of in-addr.arpa. A reverse lookup is divided in two parts; a zone and the PTR record.

Implementation of the server and BIND 9

A new server was build using Debian 3.0 as the operating system. Unfortunately a release candidate was used. No stable version was released yet so the release candidate was as close as it gets. The installation was straightforward and applying security was not much different then a regular UNIX based machine with IPv4. The installation of the bind packages and configuring it to listen on IPv6 was as easy as a regular BIND installation. In less then a day the production server was production ready.

Apache web server implementation

The apache web server is the most popular web server on the internet. Since version 2.0 it offers native IPv6 support. Version 2.052 was installed on the Debian server. By just following the online manual it easily listened on port 80 for IPv6.

Synopsis

Through all levels of the OSI model adjustments needed to be made to devices before IPv6 could be implemented. It was noticeable that not all hardware could implement IPv6. Appendix F gives a detailed picture of the IPv6 network of CIPC.

Chapter 8

CONNECTING THE CUSTOMER TO IPV6

This chapter will describe the last part of the project. It will start with describing the process of getting IPv6 to customers. After that, the implementation of Network Address Translation Protocol Translation (NAT-PT) [**RFC 2766**, Tsirtsis and Srisuresh, 2000] will be reviewed. The chapter will end with briefly reviewing a FreeBSD machine and a Windows XP machine on the IPv6 internet.

Connecting the customers to IPv6 internet

Connecting customers to IPv6 in the same manner in which

IPv4 addresses are delegated to customers was not possible. The SMS, the device that applies the IP addresses to the CPEs is not compatible with IPv6. As this is the most expensive Network Device in the CIPC network, replacement with an IPv6 compatible device was not an option. The only option was to tunnel over IPv4. Tunnels are generally used to carry incompatible protocols or specific data over an existing network. In this case tunneling was used to transport IPv6 packets over an IPv4 network. There are three options of tunneling possible between devices [Desmeules, 2003]:

- **Host to host** Isolated hosts with dual stack can establish a tunnel. This architecture only allows the establishment of IPv6 sessions between hosts.

- **Host to router** Isolated hosts with a dual stack can establish a tunnel with a router. The router may have native IPv6 connectivity. This

architecture allows the establishment of end to end connections between the hosts and any IPv6 destination.

- **Router to router** Routers with a dual stack can establish a tunnel with other routers with a dual stack. Routers can be used to interconnect island of IPv6 hosts. Any hosts connected to one of these routers can establish native IPv6 end-to-end connections.

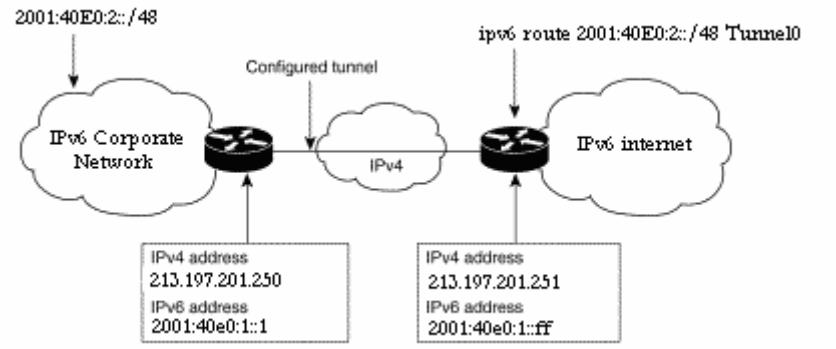
Deploying the router to router tunnel

The main goal of this thesis is to deliver IPv6 to the customer in its most compatible form. This means it should be delivered on the router. Only this option gives the customer the choice to design its whole network in a transparent way using IPv6. Two methods for deploying a router to router tunnel within an ISP network are available today. These methods are:

- Configured tunnel using the IPv6ip method
- Generic Router Encapsulation (GRE) tunnel

The MTU of a GRE tunnel is lower then the MTU of the configured tunnel using IPv6ip. On top of that most routers CIPC uses can only handle the IPv6ip method. Therefore the configured tunnel was used as the bridge between the IPv6 islands. To implement configured tunnels between customers and the Core network a special router was placed in TeleCity. In theory one of the Core routers could be used for this purpose too. Due to pollution of the configuration and the extra workload on CPU and memory, CIPC chose the option of installing a router specifically for this purpose. Figure 10 is an illustration of how this tunneling works:

Figure 10: IPv6 over IPv4 configured tunnel



The router at the edge of the IPv6 internet routes a /48 network to the corporate network through the tunnel.

IPv4 interoperability of IPv6 networks

The last assignment that is part of this thesis is the interoperability of IPv6 networks with IPv4. This means looking for methods of IPv6 hosts to interact with the IPv4 internet. The two methods that can be distinguished are:

- Application-level gateways (ALGs)
- NAT-PT

Application-level gateways

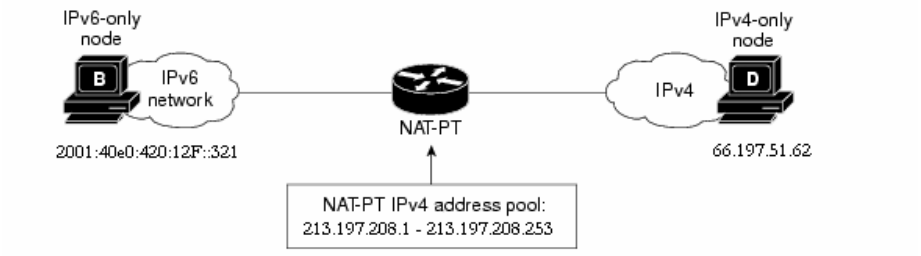
ALGs are gateways with dual stack support at the boundary of the IPv6 network. It allows IPv6-only nodes to interact with nodes that are IPv4-only. An example of an ALG is a dual stack proxy that is used in an IPv6-only network. The proxy server gets requests over IPv6 from IPv6 nodes. It will then try to reach the

destination web server over IPv6. If that is not possible it will reach the server over IPv4.

NAT-PT

NAT-PT was designed as a mechanism to allow IPv6-only nodes to communicate with IPv4 devices. The mechanism resembles Network Address Translation (NAT) for IPv4; a host behind the NAT-PT device is able to talk to IPv4 hosts using only one or more IPv4 addresses of the NAT-PT device. Within NAT-PT support for numerous applications is being deployed similar to the support for applications in NAT for IPv4. NAT-PT has the same restrictions that apply to IPv4 NAT. For instance, NAT-PT does not provide end-to-end security and the NAT-PT router can be a single point of failure in the network. Different types of NAT-PT can be identified; There is static NAT-PT and dynamic NAT-PT. Within dynamic NAT-PT there is the ability to overload the IPv4 address (also known as PAT) which means that a single IPv4 address can translate IPv4 addresses for more than one IPv6 host. NAT-PT is mostly used in conjunction with DNS-ALG. DNS ALG intercepts DNS requests sent by an IPv6 host and translates IPv4 records into IPv6 records. For the purpose of this thesis dynamic NAT-PT combined with DNS-ALG was chosen without the usage of PAT. To implement NAT-PT in an IPv6 network a /96 prefix within the IPv6 network that is going to be translated is needed. The /96 needs to be routed to the NAT-PT device. Figure 11 illustrates the usage of dynamic NAT-PT.

Figure 11: Dynamic NAT-PT



CIPC NAT-PT implementation

A special router was installed at CIPC that was dedicated for NAT-PT /DNS ALG purposes. In each IPv6 network a /96 prefix was routed to this specific device. After implementing NAT-PT with DNS-ALG *dig* queries to IPv4 addresses gave quad A records as answers. All the answers showed the IPv6 addresses that are part of the /96 prefix. Browsing the internet using NAT-PT on an IPv6-only node was surprisingly easy.

Working with IPv6

Two systems were tested for there IPv6 compatibility and flexibility, a FreeBSD machine (5.2.1) and a Windows XP professional SP2 machine. The two machines where connected directly to a DSL router with IPv6 Neighbor Discovery enabled. The systems where tested without and with NAT-PT enabled. Only browsing was tested.

FreeBSD:

To install FreeBSD an internet connection is needed. The installation is an interactive procedure and during the installation procedure a question pops up concerning the installation of IPv6 on the system. After confirming the installation of IPv6, the IPv6 stack was installed. After providing an IPv6 name server the system was able to download the packages and install FreeBSD using a native IPv6 ftp server.

Without NAT-PT enabled hardly any sites where reachable. As an example the following well known sites could not be reached:

www.nu.nl

www.google.com

www.cnn.com

The only mayor site that could be reached was:

www.lundia.nl

Windows XP:

Windows XP is installed using a CD-ROM. After installation windows needs to be registered. This is not possible using an IPv6-only network connection. Another problem that rose was the inability to use an IPv6 DNS server. In the IPv6 stack of Windows XP it is not possible to provide an IPv6 DNS server. DNS requests are sent over IPv4. Due to these DNS issues browsing the internet using an IPv6-only connection is not possible with Windows XP.

Synopsis

Building a native IPv6 network with the existing hardware was not possible. With specific hardware CIPC was able to tunnel IPv6 prefixes to customers. IPv6-only hosts could be made interoperable with the IPv4 internet using NAT-PT and DNS-ALG. From the Operating Systems FreeBSD and Windows XP, only the first one could be used in an IPv6-only network.

Chapter 9

CONCLUSION AND RECOMMENDATIONS

I started this thesis with the purpose to investigate the process of the implementation of IPv6 at CIPC, a business to business Internet Provider. The goal was to find an answer to the question how straightforward the implementation was of IPv6 in an IPv4 network. In the theoretical part IPv6 was compared with its predecessor and the new features and improvements were reviewed. The practical part presented the roll out of IPv6 at CIPC and showed which problems were encountered and how they were dealt with. My findings will be divided in a theoretical part and a practical part. I will end this chapter with recommendations.

Conclusion

Theoretical findings

In the first four chapters it becomes clear that IPv6 can become a worthy successor of IPv4. The IP protocol has been drastically reengineered which directly reveals its pro's and cons.

Numerous pro's can be stated about IPv6, among those off course the larger address space and the features of the Neighbor Discovery Protocol. However the real power lies in the fact that everyone who is working on IPv6 knows the strengths and weaknesses of IPv4 and used this knowledge to engineer IPv6.

One of the cons is that IPv6 requires most of the hard- and software to be reengineered. Another big problem compared to IPv4 is that it is still under heavy construction. As an example I name the suggested Site Local address. During

writing of my thesis this address type became deprecated [**RFC 3879**, Huitema and Carpenter, 2004]. This heavy construction causes software developers and their clients to become reluctant to design and build software that is IPv6 ready.

Practical findings

Despite the large address space of IPv6 there is a clear division between the prefix-ID and the Interface-ID in an IPv6 address. This makes the development of an IP plan much easier. The large address space helps in making allocation to customers much more transparent.

Seeking a painless implementing of IPv6 exposed the limitations of the hardware we used at CIPC. After relocating some of hardware, CIPC was able to implement IPv6 in the network. Limitations in the existing hardware had caused some drawbacks in the implementation.

Implementing basic services like DNS and Web services were relatively painless, although it should be stated that CIPC only uses UNIX based machines for these purposes. These services were not tested using Microsoft products.

Portability between IPv4 and IPv6 is available but rather limited. Numerous methods are still being developed. At CIPC, NAT-PT in combination with DNS-ALG was tested. It worked fine for regular browsing and SMTP but failed for applications that use IP addresses in the payload of their IP datagram's. The NAT-PT router is unaware of their existence and will not process these addresses. Examples of these packets are ftp control sessions that carry IP addresses in their payload. For ftp though an ALG has been developed within NAT-PT.

Looking at Operating Systems it becomes clear that the UNIX implementation of IPv6 is further than the Microsoft implementation. A good example is the fact

that a Windows XP machine can only send DNS queries over IPv4. This makes an IPv6-only node running Windows XP barely workable on the internet. With the machine running UNIX it was possible to browse the internet using only an IPv6 address.

Recommendations

I would recommend every ISP to start deploying an IPv6 network as soon as possible. The sooner an ISP starts with deploying an IPv6 network the better the chances are that they get a technological stronghold over their competitors. Because IPv6 is still being developed and adjusted today, staying close to the technology is also highly advisable and might even be influential on the development of the protocol as well. All companies that claim to be progressive in their technical management should encourage the development of an IPv6 network. Furthermore it should be stated that the acceptance of IPv6 on the internet depends on the adoption by the ISPs. Fast adoption by ISPs is vital because one important thing I learned during the course of this thesis is that IPv6 is far from being accepted.

When setting up an IPv6 network there is always room for improvement. In the case of CIPC I started with the IP plan. Due to limitations in the hardware this might not have been the best idea. The IP plan had to be revised numerous times before coming to its final version. In its most ideal form a company should adjust its network so IPv6 could be delivered natively to customers. Due to the current lack of applications and demand I would suggest companies to not invest ample funds in new IPv6-ready hardware immediately but to try using the existing network devices for implementation of IPv6. I would urge all companies to start the implementation as soon as possible therefore paving the way for IPv6 to become dominant over the existing internet protocol IPv4.

BIBLIOGRAPHY

Hagen, S. IPv6 Essentials (2002), O'Reilly.

Desmeules, R. IPv6 Integration and Coexistence Strategies (2003), Cisco Press.

Cisco.(2004), www.cisco.com.

RFC 1518. Rekhter, Y. and T. Li. An Architecture for IP Address Allocation with CIDR.(1993), IETF, ftp://ftp.rfc-editor.org/in-notes/rfc1518.txt.

RFC 2460. Deering, S. E. and R. M. Hinden. Internet Protocol, Version 6 (IPv6) Specification.(1998), IETF, ftp://ftp.rfc-editor.org/in-notes/rfc2460.txt.

IANA. www.iana.org.

RFC 3775. Perkins, C. E., D. B. Johnson, et al. Mobility Support in IPv6.(2004), IETF, ftp://ftp.rfc-editor.org/in-notes/rfc3775.txt.

RFC 3513. Deering, S. E. and R. M. Hinden. Internet Protocol Version 6 (IPv6) Addressing Architecture.(2003), IETF, ftp://ftp.rfc-editor.org/in-notes/rfc3513.txt.

RFC 3879. Huitema, C. and B. Carpenter. Deprecating Site Local Addresses.(2004), IETF, ftp://ftp.rfc-editor.org/in-notes/rfc3879.txt.

RFC 2463. Conta, A. and R. M. Hinden. Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification.(1998), IETF, ftp://ftp.rfc-editor.org/in-notes/rfc2463.txt.

RFC 2461. Narten, T., E. Nordmark, et al. Neighbor Discovery for IP Version 6 (IPv6).(1998), IETF, ftp://ftp.rfc-editor.org/in-notes/rfc2461.txt.

RFC 3122. Conta, A. Extensions to IPv6 Neighbor Discovery for Inverse Discovery Specification.(2001), IETF, ftp://ftp.rfc-editor.org/in-notes/rfc3122.txt.

ripe-267. Inomata, A. e. a. IPv6 Address Allocation and Assignment Policy.(2003), RIPE, http://www.ripe.net/ripe/docs/IPv6policy.html.

RFC 3177. Baker, F., B. Carpenter, et al. IAB/IESG Recommendations on IPv6 Address Allocations to Sites.(2001), IETF, ftp://ftp.rfc-editor.org/in-notes/rfc3177.txt.

RFC 2464. Crawford, M. Transmission of IPv6 Packets over Ethernet Networks.(1998), IETF, ftp://ftp.rfc-editor.org/in-notes/rfc2464.txt.

Döring, G. IPv6 BGP filter recommendations.(2004), Gert Döring, http://www.space.net/~gert/RIPE/IPv6-filters.html.

Cisco. Implementing OSPF for IPv6.(2003), Cisco Systems, http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/122newft/122t/122t13/IPv6/IPv6imp/sa_ospf3.htm#1076395.

Hensema, E. Configuratie van de Bind 9 DNS-server.(2004), Erik Hensema, http://www.hensema.net/docs/bind9/.

RFC 2766. Tsirtsis, G. and P. Srisuresh. Network Address Translation - Protocol Translation (NAT-PT).(2000), IETF, ftp://ftp.rfc-editor.org/in-notes/rfc2766.txt.

Desmeules, R. IPv6 Integration and Coexistence Strategies. Implementing Cisco IPv6 Networks (IPV6), (234-244, 254-255, 262-273), Cisco Press.(2003).

Appendix A

PROTOCOL NUMBERING

This information can also be obtained from IANA. In IPv4 there is a field,called "Protocol", to identify the next level protocol. This is an 8bit field. In IPv6 this field is called the "Next Header" field.

Assigned Internet Protocol Numbers:

Number	Protocol
0	HOPOPT IPv6 Hop-by-Hop Option
1	ICMP Internet Control Message
2	IGMP Internet Group Management
3	GGP Gateway-to-Gateway
4	IP IP in IP (encapsulation)
5	ST Stream
6	TCP Transmission Control
7	CBT CBT
8	EGP Exterior Gateway Protocol
9	IGP any private interior gateway (used by Cisco for their IGRP)
10	BBN-RCC-MON BBN RCC Monitoring
11	NVP-II Network Voice Protocol
12	PUP PUP
13	ARGUS ARGUS
14	EMCON EMCON
15	XNET Cross Net Debugger
16	CHAOS Chaos
17	UDP User Datagram
18	MUX Multiplexing
19	DCN-MEAS DCN Measurement Subsystems
20	HMP Host Monitoring
21	PRM Packet Radio Measurement
22	XNS-IDP XEROX NS IDP
23	TRUNK-1 Trunk-1
24	TRUNK-2 Trunk-2
25	LEAF-1 Leaf-1
26	LEAF-2 Leaf-2

27	RDP Reliable Data Protocol
28	IRTP Internet Reliable Transaction
29	ISO-TP4 ISO Transport Protocol Class 4
30	NETBLT Bulk Data Transfer Protocol
31	MFE-NSP MFE Network Services Protocol
32	MERIT-INP MERIT Internodal Protocol
33	SEP Sequential Exchange Protocol
34	3PC Third Party Connect Protocol
35	IDPR Inter-Domain Policy Routing Protocol
36	XTP XTP
37	DDP Datagram Delivery Protocol
38	IDPR-CMTP IDPR Control Message Transport Proto
39	TP++ TP++ Transport Protocol
40	IL IL Transport Protocol
41	IPv6 IPv6
42	SDRP Source Demand Routing Protocol
43	IPv6-Route Routing Header for IPv6
44	IPv6-Frag Fragment Header for IPv6
45	IDRP Inter-Domain Routing Protocol
46	RSVP Reservation Protocol
47	GRE General Routing Encapsulation
48	MHRP Mobile Host Routing Protocol
49	BNA BNA
50	ESP Encapsulating Security Payload
51	AH Authentication Header
52	I-NLSP Integrated Net Layer Security TUBA
53	SWIPE IP with Encryption
54	NARP NBMA Address Resolution Protocol
55	MOBILE IP Mobility
56	TLSP Transport Layer Security Protocol with Kryptonet key management
57	SKIP SKIP
58	IPv6-ICMP ICMP for IPv6
59	IPv6-NoNxt No Next Header for IPv6
60	IPv6-Opts Destination Options for IPv6
61	any host internal protocol
62	CFTP CFTP
63	any local network
64	SAT-EXPAK SATNET and Backroom EXPAK
65	KRYPTOLAN Kryptolan
66	RVD MIT Remote Virtual Disk Protocol
67	IPPC Internet Pluribus Packet Core
68	any distributed file system
69	SAT-MON SATNET Monitoring
70	VISA VISA Protocol

71	IPCV Internet Packet Core Utility
72	CPNX Computer Protocol Network Executive
73	CPHB Computer Protocol Heart Beat
74	WSN Wang Span Network
75	PVP Packet Video Protocol
76	BR-SAT-MON Backroom SATNET Monitoring
77	SUN-ND SUN ND PROTOCOL-Temporary
78	WB-MON WIDEBAND Monitoring
79	WB-EXPAK WIDEBAND EXPAK
80	ISO-IP ISO Internet Protocol
81	VMTP VMTP
82	SECURE-VMTP SECURE-VMTP
83	VINES VINES
84	TTP TTP
85	NSFNET-IGP NSFNET-IGP
86	DGP Dissimilar Gateway Protocol
87	TCF TCF
88	EIGRP EIGRP
89	OSPFIGP OSPFIGP
90	Sprite-RPC Sprite RPC Protocol
91	LARP Locus Address Resolution Protocol
92	MTP Multicast Transport Protocol
93	AX.25 AX.25 Frames
94	IPIP IP-within-IP Encapsulation Protocol
95	MICP Mobile Internetworking Control Protocol
96	SCC-SP Semaphore Communications Sec. Protocol
97	ETHERIP Ethernet-within-IP Encapsulation
98	ENCAP Encapsulation Header
99	any private encryption scheme
100	GMTP GMTP
101	IFMP Ipsilon Flow Management Protocol
102	PNNI PNNI over IP
103	PIM Protocol Independent Multicast
104	ARIS ARIS
105	SCPS SCPS
106	QNX QNX
107	A/N Active Networks
108	IPComp IP Payload Compression Protocol
109	SNP Sitara Networks Protocol
110	Compaq-Peer Compaq Peer Protocol
111	IPX-in-IP IPX in IP
112	VRRP Virtual Router Redundancy Protocol
113	PGM PGM Reliable Transport Protocol
114	any 0-hop protocol

115	L2TP Layer Two Tunneling Protocol
116	DDX D-II Data Exchange (DDX)
117	IATP Interactive Agent Transfer Protocol
118	STP Schedule Transfer Protocol
119	SRP SpectraLink Radio Protocol
120	UTI UTI
121	SMP Simple Message Protocol
122	SM SM
123	PTP Performance Transparency Protocol
124	ISIS over IPv4
125	FIRE
126	CRTP Combat Radio Transport Protocol
127	CRUDP Combat Radio User Datagram
128	SSCOPMCE
129	IPLT
130	SPS Secure Packet Shield
131	PIPE Private IP Encapsulation within IP
132	SCTP Stream Control Transmission Protocol
133	FC Fibre Channel
134	RSVP-E2E-IGNORE
135-254	Unassigned
255	Reserved

Appendix B

IPV6 MULTICAST ADDRESSES

IPv6 multicast addresses are defined in "IP Version 6 Addressing Architecture" [RFC3513]. This defines fixed scope and variable scope multicast addresses. The rules for assigning new IPv6 multicast addresses are defined in [RFC3513]. IPv6 multicast addresses not listed below are reserved. The current fixed IPv6 multicast addresses are listed below.

Fixed Scope Multicast Addresses

These permanently assigned multicast addresses are valid over a specified scope value:

Multicast Node scope assigned addresses

Node-Local Scope	**Purpose**	**Source**
FF01:0:0:0:0:0:0:1	All Nodes Address	[RFC2373]
FF01:0:0:0:0:0:0:2	All Routers Address	[RFC2373]

Multicast Local scope assigned addresses

Link Local Scope	Purpose	Source
FF02:0:0:0:0:0:0:1	All Nodes Address	[RFC2373]
FF02:0:0:0:0:0:0:2	All Routers Address	[RFC2373]
FF02:0:0:0:0:0:0:3	Unassigned	[JBP]
FF02:0:0:0:0:0:0:4	DVMRP Routers	[RFC1075,JBP]
FF02:0:0:0:0:0:0:5	OSPFIGP	[RFC2328,Moy]
FF02:0:0:0:0:0:0:6	OSPFIGP Designated Routers	[RFC2328,Moy]
FF02:0:0:0:0:0:0:7	ST Routers	[RFC1190,KS14]
FF02:0:0:0:0:0:0:8	ST Hosts	[RFC1190,KS14]
FF02:0:0:0:0:0:0:9	RIP Routers	[RFC2080]
FF02:0:0:0:0:0:0:A	EIGRP Routers	[Farinacci]
FF02:0:0:0:0:0:0:B	Mobile-Agents	[Bill Simpson]
FF02:0:0:0:0:0:0:C	SSDP	[Kostic]
FF02:0:0:0:0:0:0:D	All PIM Routers	[Farinacci]
FF02:0:0:0:0:0:0:E	RSVP-ENCAPSULATION	[Braden]
FF02:0:0:0:0:0:0:16	All MLDv2-capable routers	[RFC3810]
FF02:0:0:0:0:0:1:1	Link Name	[Harrington]
FF02:0:0:0:0:0:1:2	All-dhcp-agents	[RFC3315]
FF02:0:0:0:0:0:1:3	Link-local Multicast Name Resolution	[Aboba]
FF02:0:0:0:0:0:1:4	DTCP Announcement	[Vieth, Tersteegen]
FF02:0:0:0:0:1:FFXX:XXXX	Solicited-Node Address	[RFC2373]

Appendix C

IPV6 TOP LEVEL AGGREGATION IDENTIFIER ASSIGNMENTS

The source of this appendix can be found at IANA.

TLA Identifier Assignments

Top Level Aggregator (TLA) Identifiers are defined in [RFC2374] and are assigned from the Format Prefix (FP) 001 (binary) in [RFC2373].

TLA ID assignments:

IPv6 Prefix	FP	TLA Binary Value	TLA Hex	Assignment
2000::/16	001	0 0000 0000 0000	0x0000	Reserved
2001::/16	001	0 0000 0000 0001	0x0001	Sub-TLA Assignments [RFC2450]
2002::/16	001	0 0000 0000 0010	0x0002	6to4 [RFC3056]
3FFE::/16	001	1 1111 1111 1110	0x1FFE	6bone [RFC2471] (phased out 6-Jun-06 [RFC3701])
3FFF::/16	001	1 1111 1111 1111	0x1FFF	Reserved

Sub-TLA Identifier Assignments

Sub-TLA Identifiers are defined in [RFC2450] and are assigned out of TLA ID 0x0001. Note that use of the Reserved field to create the Sub-TLA field is specific to TLA ID 0x0001. It does not affect any other TLA ID.

Sub-TLA Identifier Assignments

3	13	13	19
FP	TLA-ID	Sub-TLA	NLA-ID
001	0 0000 0000 0001	xxxx xxxx xxxx x	xxx xxxx xxxx xxxx xxxx

As specified in [RFC2450], Sub-TLA ID assignments are done in blocks. The initial Sub-TLA ID assignments to IP address registries are in blocks of 64 Sub-TLA IDs. These assignments are listed below.

Sub-TLA ID assignments:

IPv6 Prefix	sub-TLA Binary Values	Allocated to	Date
2001:0000::/23	0000 000X XXXX X	IANA	Jul-99
2001:0200::/23	0000 001X XXXX X	APNIC	Jul-99
2001:0400::/23	0000 010X XXXX X	ARIN	Jul-99
2001:0600::/23	0000 011X XXXX X	RIPE NCC	Jul-99
2001:0800::/23	0000 100X XXXX X	RIPE NCC	May-02
2001:0A00::/23	0000 101X XXXX X	RIPE NCC	Nov-02
2001:0C00::/23	0000 110X XXXX X	APNIC	May-02
2001:0E00::/23	0000 111X XXXX X	APNIC	Jan-03
2001:1000::/23	0001 000X XXXX X	(future assignment)	
2001:1200::/23	0001 001X XXXX X	LACNIC	Nov-02
2001:1400::/23	0001 010X XXXX X	RIPE NCC	Feb-03
2001:1600::/23	0001 011X XXXX X	RIPE NCC	Jul-04
2001:1800::/23	0001 100X XXXX X	ARIN	Apr 03
2001:1A00::/23	0001 101X XXXX X	RIPE NCC	Jan-04
2001:1C00::/23	0001 110X XXXX X	RIPE NCC	May-04
2001:1E00::/23	0001 111X XXXX X	RIPE NCC	May-04
2001:2000::/23	0010 000X XXXX X	RIPE NCC	May-04
2001:2200::/23	0010 001X XXXX X	RIPE NCC	May-04
2001:2400::/23	0010 010X XXXX X	RIPE NCC	May-04
2001:2600::/23	0010 011X XXXX X	RIPE NCC	May-04
2001:2800::/23	0010 100X XXXX X	RIPE NCC	May-04
2001:2A00::/23	0010 101X XXXX X	RIPE NCC	May-04
2001:2C00::/23	0010 110X XXXX X	RIPE NCC	May-04
2001:2E00::/23	0010 111X XXXX X	RIPE NCC	May-04
2001:3000::/23	0011 000X XXXX X	RIPE NCC	May-04

2001:3200::/23	0011 001X XXXX X	RIPE NCC	May-04
2001:3400::/23	0011 010X XXXX X	RIPE NCC	May-04
2001:3600::/23	0011 011X XXXX X	RIPE NCC	May-04
2001:3800::/23	0011 100X XXXX X	RIPE NCC	May-04
2001:3A00::/23	0011 101X XXXX X	RIPE NCC	May-04
2001:3C00::/23	0011 110X XXXX X	(reserved *)	Jun 04
2001:3E00::/23	0011 111X XXXX X	(reserved *)	Jun-04
2001:4000::/23	0100 000X XXXX X	RIPE NCC	Jun 04
2001:4200::/23	0100 001X XXXX X	ARIN	Jun-04
2001:4400::/23	0100 010X XXXX X	APNIC	Jun-04
2001:4600::/23	0100 011X XXXX X	RIPE NCC	Aug 04
2001:4800::/23	0100 100X XXXX X	ARIN	Aug-04
2001:4A00::/23	0100 101X XXXX X	RIPE NCC	Oct-04
. . .	. . .	. . .	. . .
2001:5000::/23	0101 000X XXXX X	RIPE NCC	Sep-04
2001:5200::/23	0101 001X XXXX X	RIPE NCC	Sep-04
2001:5400::/23	0101 010X XXXX X	RIPE NCC	Sep-04
2001:5600::/23	0101 011X XXXX X	RIPE NCC	Sep-04
2001:5800::/23	0101 100X XXXX X	RIPE NCC	Sep-04
2001:5A00::/23	0101 101X XXXX X	RIPE NCC	Sep-04
2001:5C00::/23	0101 110X XXXX X	RIPE NCC	Sep-04
2001:5E00::/23	0101 111X XXXX X	RIPE NCC	Sep-04
. . .	. . .	. . .	. . .
. . .	. . .	. . .	. . .
. . .	. . .	. . .	. . .
2001:FE00::/23	1111 111X XXXX X	(future assignment)	

Appendix D

IP PLAN CIPC

General Division IPv6 blocks:

General Division /35 subnets Hexadecimal View (z=hex. Value(0,2,4,6,8,a)	
Begin	**End**
2001:40e0:z000::/48	2001:40e0:(z+1)ffe::/48
General Division /35 subnets Binary View (bit 33 to 48) (x=bin. value)	
Begin	**End**
xxx0 0000 0000 0000	xxx1 1111 1111 1110
General Division /64 subnets Hexadecimal View (z=hex. value)	
Begin	**End**
2001:40e0:(z+1)fff:0000	2001:40e0:(z+1)fff:fffe
General Division /64 subnets Binary View (bit 49 to 64) (x=bin. value)	
Begin	**End**
0000 0000 0000 0000	1111 1111 1111 1110
General Division /128 subnets Hexadecimal View (z=hex. value)	
2001:40e0:(z+1)fff:ffff:0:0:0:0:/128	2001:40e0:(z+1)fff:ffff:ffff:ffff:ffff:ffff:/128

Block allocation per POP:

IPv6 prefix	Allocated to:	NLA ID Binary Values (bit 33 to 48) x=binary value
2001:40e0:0000::/35	TeleCity	000x xxxx xxxx xxxx
2001:40e0:2000::/35	TeleCity	001x xxxx xxxx xxxx
2001:40e0:4000::/35	Global Switch	010x xxxx xxxx xxxx
2001:40e0:6000::/35	Global Switch	011x xxxx xxxx xxxx
2001:40e0:8000::/35	CyberCentre	100x xxxx xxxx xxxx
2001:40e0:a000::/35	CyberCentre	101x xxxx xxxx xxxx
2001:40e0:c000::/35	Future Assignment	110x xxxx xxxx xxxx
2001:40e0:e000::/35	Future Assignment	111x xxxx xxxx xxxx

IP block I division TeleCity

2001:40e0:0000::	**/35**		**2001:40e0:1fff:0000::**	**/48**	
IPv6 subnet	**Prefix**	**Customer Name**	**IPv6 subnet**	**Prefix**	**Customer Name**
2001:40e0:0000::	/48	CIPC	2001:40e0:1fff:0000::	/64	Xirmo
2001:40e0:0001::	/48	Tunnelblok eindpunten	2001:40e0:1fff:0001::	/64	
2001:40e0:0002::	/48		2001:40e0:1fff:0002::	/64	
2001:40e0:0003::	/48		2001:40e0:1fff:0003::	/64	
2001:40e0:0004::	/48		2001:40e0:1fff:0004::	/64	
2001:40e0:0005::	/48		2001:40e0:1fff:0005::	/64	
2001:40e0:0006::	/48		2001:40e0:1fff:0006::	/64	
2001:40e0:0007::	/48		2001:40e0:1fff:0007::	/64	
2001:40e0:0008::	/48		2001:40e0:1fff:0008::	/64	
2001:40e0:0009::	/48		2001:40e0:1fff:0009::	/64	
2001:40e0:000a::	/48		2001:40e0:1fff:000a::	/64	
2001:40e0:000b::	/48		2001:40e0:1fff:000b::	/64	
2001:40e0:000c::	/48		2001:40e0:1fff:000c::	/64	
2001:40e0:000d::	/48		2001:40e0:1fff:000d::	/64	
2001:40e0:000e::	/48		2001:40e0:1fff:000e::	/64	
2001:40e0:000f::	/48		2001:40e0:1fff:000f::	/64	
2001:40e0:0010::	/48		2001:40e0:1fff:0010::	/64	
2001:40e0:0011::	/48		2001:40e0:1fff:0011::	/64	
2001:40e0:0012::	/48		2001:40e0:1fff:0012::	/64	
2001:40e0:0013::	/48		2001:40e0:1fff:0013::	/64	
2001:40e0:0014::	/48		2001:40e0:1fff:0014::	/64	
2001:40e0:0015::	/48		2001:40e0:1fff:0015::	/64	
2001:40e0:0016::	/48		2001:40e0:1fff:0016::	/64	
2001:40e0:0017::	/48		2001:40e0:1fff:0017::	/64	
2001:40e0:0018::	/48		2001:40e0:1fff:0018::	/64	
2001:40e0:0019::	/48		2001:40e0:1fff:0019::	/64	
2001:40e0:001a::	/48		2001:40e0:1fff:001a::	/64	
2001:40e0:001b::	/48		2001:40e0:1fff:001b::	/64	
2001:40e0:001c::	/48		2001:40e0:1fff:001c::	/64	
2001:40e0:001d::	/48		2001:40e0:1fff:001d::	/64	
2001:40e0:001e::	/48		2001:40e0:1fff:001e::	/64	
2001:40e0:001f::	/48		2001:40e0:1fff:001f::	/64	
2001:40e0:0020::	/48		2001:40e0:1fff:0020::	/64	
……	/48		……..	/64	
2001:40e0:1fffe::	/48		2001:40e0:1fff:fffe::	/64	

Begin block	**/128**	**End block**	**/64**
2001:40e0:1fff:ffff:	/128	2001:40e0:1fff:ffff::ffff:ffff:ffff:ffff	128

IP block II division TeleCity

2001:40e0:2000::	**/35**		**2001:40e0:3fff:0000::**	**/48**	
IPv6 subnet	**Prefix**	**Customer Name**	**IPv6 subnet**	**Prefix**	**Customer Name**
2001:40e0:2000::	/48	CIPC	2001:40e0:3fff:0000::	/64	Xirmo
2001:40e0:2001::	/48	Tunnel enpoint	2001:40e0:3fff:0001::	/64	
2001:40e0:2002::	/48		2001:40e0:3fff:0002::	/64	
2001:40e0:2003::	/48		2001:40e0:3fff:0003::	/64	
2001:40e0:2004::	/48		2001:40e0:3fff:0004::	/64	
2001:40e0:2005::	/48		2001:40e0:3fff:0005::	/64	
2001:40e0:2006::	/48		2001:40e0:3fff:0006::	/64	
2001:40e0:2007::	/48		2001:40e0:3fff:0007::	/64	
2001:40e0:2008::	/48		2001:40e0:3fff:0008::	/64	
2001:40e0:2009::	/48		2001:40e0:3fff:0009::	/64	
2001:40e0:200a::	/48		2001:40e0:3fff:000a::	/64	
2001:40e0:200b::	/48		2001:40e0:3fff:000b::	/64	
2001:40e0:200c::	/48		2001:40e0:3fff:000c::	/64	
2001:40e0:200d::	/48		2001:40e0:3fff:000d::	/64	
2001:40e0:200e::	/48		2001:40e0:3fff:000e::	/64	
2001:40e0:200f::	/48		2001:40e0:3fff:000f::	/64	
2001:40e0:2010::	/48		2001:40e0:3fff:0010::	/64	
2001:40e0:2011::	/48		2001:40e0:3fff:0011::	/64	
2001:40e0:2012::	/48		2001:40e0:3fff:0012::	/64	
2001:40e0:2013::	/48		2001:40e0:3fff:0013::	/64	
2001:40e0:2014::	/48		2001:40e0:3fff:0014::	/64	
2001:40e0:2015::	/48		2001:40e0:3fff:0015::	/64	
2001:40e0:2016::	/48		2001:40e0:3fff:0016::	/64	
2001:40e0:2017::	/48		2001:40e0:3fff:0017::	/64	
2001:40e0:2018::	/48		2001:40e0:3fff:0018::	/64	
2001:40e0:2019::	/48		2001:40e0:3fff:0019::	/64	
2001:40e0:201a::	/48		2001:40e0:3fff:001a::	/64	
2001:40e0:201b::	/48		2001:40e0:3fff:001b::	/64	
2001:40e0:201c::	/48		2001:40e0:3fff:001c::	/64	
2001:40e0:201d::	/48		2001:40e0:3fff:001d::	/64	
2001:40e0:201e::	/48		2001:40e0:3fff:001e::	/64	
2001:40e0:201f::	/48		2001:40e0:3fff:001f::	/64	
2001:40e0:2020::	/48		2001:40e0:3fff:0020::	/64	
……	/48		……..	/64	
2001:40e0:3fffe::	/48		2001:40e0:3fff:fffe::	/64	
Begin block		**/128**	**End block**		**/64**
2001:40e0:3fff:ffff:		/128	2001:40e0:3fff:ffff::ffff:ffff:ffff:ffff		128

IP block I division GlobalSwitch

2001:40e0:4000::	**/35**		**2001:40e0:5fff::**	**/48**	
IPv6 subnet	**Prefix**	**Customer Name**	**IPv6 subnet**	**Prefix**	**Customer Name**
2001:40e0:4000::	/48		2001:40e0:5fff:0000::	/64	Xirmo
2001:40e0:4001::	/48		2001:40e0:5fff:0001::	/64	
2001:40e0:4002::	/48		2001:40e0:5fff:0002::	/64	
2001:40e0:4003::	/48		2001:40e0:5fff:0003::	/64	
2001:40e0:4004::	/48		2001:40e0:5fff:0004::	/64	
2001:40e0:4005::	/48		2001:40e0:5fff:0005::	/64	
2001:40e0:4006::	/48		2001:40e0:5fff:0006::	/64	
2001:40e0:4007::	/48		2001:40e0:5fff:0007::	/64	
2001:40e0:4008::	/48		2001:40e0:5fff:0008::	/64	
2001:40e0:4009::	/48		2001:40e0:5fff:0009::	/64	
2001:40e0:400a::	/48		2001:40e0:5fff:000a::	/64	
2001:40e0:400b::	/48		2001:40e0:5fff:000b::	/64	
2001:40e0:400c::	/48		2001:40e0:5fff:000c::	/64	
2001:40e0:400d::	/48		2001:40e0:5fff:000d::	/64	
2001:40e0:400e::	/48		2001:40e0:5fff:000e::	/64	
2001:40e0:400f::	/48		2001:40e0:5fff:000f::	/64	
2001:40e0:4010::	/48		2001:40e0:5fff:0010::	/64	
2001:40e0:4011::	/48		2001:40e0:5fff:0011::	/64	
2001:40e0:4012::	/48		2001:40e0:5fff:0012::	/64	
2001:40e0:4013::	/48		2001:40e0:5fff:0013::	/64	
2001:40e0:4014::	/48		2001:40e0:5fff:0014::	/64	
2001:40e0:4015::	/48		2001:40e0:5fff:0015::	/64	
2001:40e0:4016::	/48		2001:40e0:5fff:0016::	/64	
2001:40e0:4017::	/48		2001:40e0:5fff:0017::	/64	
2001:40e0:4018::	/48		2001:40e0:5fff:0018::	/64	
2001:40e0:4019::	/48		2001:40e0:5fff:0019::	/64	
2001:40e0:401a::	/48		2001:40e0:5fff:001a::	/64	
2001:40e0:401b::	/48		2001:40e0:5fff:001b::	/64	
2001:40e0:401c::	/48		2001:40e0:5fff:001c::	/64	
2001:40e0:401d::	/48		2001:40e0:5fff:001d::	/64	
2001:40e0:401e::	/48		2001:40e0:5fff:001e::	/64	
2001:40e0:401f::	/48		2001:40e0:5fff:001f::	/64	
2001:40e0:4020::	/48		2001:40e0:5fff:0020::	/64	
……....	/48		…….	/64	
2001:40e0:5ffe::	/48		2001:40e0:5fff:fffe::	/64	
Begin block		**/128**	**End block**		**/64**
2001:40e0:5fff:ffff:		/128	2001:40e0:5fff:ffff::ffff:ffff:ffff:ffff		128

Block II Division GlobalSwitch

2001:40e0:6000::	**/35**		**2001:40e0:7fff::**	**/48**	
IPv6 subnet	**Prefix**	**Customer Name**	**IPv6 subnet**	**Prefix**	**Customer Name**
2001:40e0:6000::	/48		2001:40e0:7fff:0000::	/64	Xirmo
2001:40e0:6001::	/48		2001:40e0:7fff:0001::	/64	
2001:40e0:6002::	/48		2001:40e0:7fff:0002::	/64	
2001:40e0:6003::	/48		2001:40e0:7fff:0003::	/64	
2001:40e0:6004::	/48		2001:40e0:7fff:0004::	/64	
2001:40e0:6005::	/48		2001:40e0:7fff:0005::	/64	
2001:40e0:6006::	/48		2001:40e0:7fff:0006::	/64	
2001:40e0:6007::	/48		2001:40e0:7fff:0007::	/64	
2001:40e0:6008::	/48		2001:40e0:7fff:0008::	/64	
2001:40e0:6009::	/48		2001:40e0:7fff:0009::	/64	
2001:40e0:600a::	/48		2001:40e0:7fff:000a::	/64	
2001:40e0:600b::	/48		2001:40e0:7fff:000b::	/64	
2001:40e0:600c::	/48		2001:40e0:7fff:000c::	/64	
2001:40e0:600d::	/48		2001:40e0:7fff:000d::	/64	
2001:40e0:600e::	/48		2001:40e0:7fff:000e::	/64	
2001:40e0:600f::	/48		2001:40e0:7fff:000f::	/64	
2001:40e0:6010::	/48		2001:40e0:7fff:0010::	/64	
2001:40e0:6011::	/48		2001:40e0:7fff:0011::	/64	
2001:40e0:6012::	/48		2001:40e0:7fff:0012::	/64	
2001:40e0:6013::	/48		2001:40e0:7fff:0013::	/64	
2001:40e0:6014::	/48		2001:40e0:7fff:0014::	/64	
2001:40e0:6015::	/48		2001:40e0:7fff:0015::	/64	
2001:40e0:6016::	/48		2001:40e0:7fff:0016::	/64	
2001:40e0:6017::	/48		2001:40e0:7fff:0017::	/64	
2001:40e0:6018::	/48		2001:40e0:7fff:0018::	/64	
2001:40e0:6019::	/48		2001:40e0:7fff:0019::	/64	
2001:40e0:601a::	/48		2001:40e0:7fff:001a::	/64	
2001:40e0:601b::	/48		2001:40e0:7fff:001b::	/64	
2001:40e0:601c::	/48		2001:40e0:7fff:001c::	/64	
2001:40e0:601d::	/48		2001:40e0:7fff:001d::	/64	
2001:40e0:601e::	/48		2001:40e0:7fff:001e::	/64	
2001:40e0:601f::	/48		2001:40e0:7fff:001f::	/64	
2001:40e0:6020::	/48		2001:40e0:7fff:0020::	/64	
……....	/48		…….	/64	
2001:40e0:7ffe::	/48		2001:40e0:7fff:fffe::	/64	
Begin block		**/128**	**End block**		**/64**
2001:40e0:7fff:ffff:		/128	2001:40e0:7fff:ffff::ffff:ffff:ffff:ffff		128

Block II Division CyberCenter

2001:40e0:8000::	**/35**		**2001:40e0:9fff::**	**/48**	
IPv6 subnet	**Prefix**	**Customer Name**	**IPv6 subnet**	**Prefix**	**Customer Name**
2001:40e0:8000::	/48		2001:40e0:9fff:0000::	/64	
2001:40e0:8001::	/48		2001:40e0:9fff:0001::	/64	
2001:40e0:8002::	/48		2001:40e0:9fff:0002::	/64	
2001:40e0:8003::	/48		2001:40e0:9fff:0003::	/64	
2001:40e0:8004::	/48		2001:40e0:9fff:0004::	/64	
2001:40e0:8005::	/48		2001:40e0:9fff:0005::	/64	
2001:40e0:8006::	/48		2001:40e0:9fff:0006::	/64	
2001:40e0:8007::	/48		2001:40e0:9fff:0007::	/64	
2001:40e0:8008::	/48		2001:40e0:9fff:0008::	/64	
2001:40e0:8009::	/48		2001:40e0:9fff:0009::	/64	
2001:40e0:800a::	/48		2001:40e0:9fff:000a::	/64	
2001:40e0:800b::	/48		2001:40e0:9fff:000b::	/64	
2001:40e0:800c::	/48		2001:40e0:9fff:000c::	/64	
2001:40e0:800d::	/48		2001:40e0:9fff:000d::	/64	
2001:40e0:800e::	/48		2001:40e0:9fff:000e::	/64	
2001:40e0:800f::	/48		2001:40e0:9fff:000f::	/64	
2001:40e0:8010::	/48		2001:40e0:9fff:0010::	/64	
2001:40e0:8011::	/48		2001:40e0:9fff:0011::	/64	
2001:40e0:8012::	/48		2001:40e0:9fff:0012::	/64	
2001:40e0:8013::	/48		2001:40e0:9fff:0013::	/64	
2001:40e0:8014::	/48		2001:40e0:9fff:0014::	/64	
2001:40e0:8015::	/48		2001:40e0:9fff:0015::	/64	
2001:40e0:8016::	/48		2001:40e0:9fff:0016::	/64	
2001:40e0:8017::	/48		2001:40e0:9fff:0017::	/64	
2001:40e0:8018::	/48		2001:40e0:9fff:0018::	/64	
2001:40e0:8019::	/48		2001:40e0:9fff:0019::	/64	
2001:40e0:801a::	/48		2001:40e0:9fff:001a::	/64	
2001:40e0:801b::	/48		2001:40e0:9fff:001b::	/64	
2001:40e0:801c::	/48		2001:40e0:9fff:001c::	/64	
2001:40e0:801d::	/48		2001:40e0:9fff:001d::	/64	
2001:40e0:801e::	/48		2001:40e0:9fff:001e::	/64	
2001:40e0:801f::	/48		2001:40e0:9fff:001f::	/64	
2001:40e0:8020::	/48		2001:40e0:9fff:0020::	/64	
……..	/48		……	/64	
2001:40e0:9ffe::	/48		2001:40e0:9fff:ffe::	/64	
Begin block		**/128**	**End block**		**/64**
2001:40e0:9fff:ffff:		/128	2001:40e0:9fff:ffff::ffff:ffff:ffff:ffff		128

Block II division CyberCenter

2001:40e0:a000::	**/35**		**2001:40e0:bfff::**	**/48**	
IPv6 subnet	**Prefix**	**Customer Name**	**IPv6 subnet**	**Prefix**	**Customer Name**
2001:40e0:a000::	/48		2001:40e0:bfff:0000::	/64	
2001:40e0:a001::	/48		2001:40e0:bfff:0001::	/64	
2001:40e0:a002::	/48		2001:40e0:bfff:0002::	/64	
2001:40e0:a003::	/48		2001:40e0:bfff:0003::	/64	
2001:40e0:a004::	/48		2001:40e0:bfff:0004::	/64	
2001:40e0:a005::	/48		2001:40e0:bfff:0005::	/64	
2001:40e0:a006::	/48		2001:40e0:bfff:0006::	/64	
2001:40e0:a007::	/48		2001:40e0:bfff:0007::	/64	
2001:40e0:a008::	/48		2001:40e0:bfff:0008::	/64	
2001:40e0:a009::	/48		2001:40e0:bfff:0009::	/64	
2001:40e0:a00a::	/48		2001:40e0:bfff:000a::	/64	
2001:40e0:a00b::	/48		2001:40e0:bfff:000b::	/64	
2001:40e0:a00c::	/48		2001:40e0:bfff:000c::	/64	
2001:40e0:a00d::	/48		2001:40e0:bfff:000d::	/64	
2001:40e0:a00e::	/48		2001:40e0:bfff:000e::	/64	
2001:40e0:a00f::	/48		2001:40e0:bfff:000f::	/64	
2001:40e0:a010::	/48		2001:40e0:bfff:0010::	/64	
2001:40e0:a011::	/48		2001:40e0:bfff:0011::	/64	
2001:40e0:a012::	/48		2001:40e0:bfff:0012::	/64	
2001:40e0:a013::	/48		2001:40e0:bfff:0013::	/64	
2001:40e0:a014::	/48		2001:40e0:bfff:0014::	/64	
2001:40e0:a015::	/48		2001:40e0:bfff:0015::	/64	
2001:40e0:a016::	/48		2001:40e0:bfff:0016::	/64	
2001:40e0:a017::	/48		2001:40e0:bfff:0017::	/64	
2001:40e0:a018::	/48		2001:40e0:bfff:0018::	/64	
2001:40e0:a019::	/48		2001:40e0:bfff:0019::	/64	
2001:40e0:a01a::	/48		2001:40e0:bfff:001a::	/64	
2001:40e0:a01b::	/48		2001:40e0:bfff:001b::	/64	
2001:40e0:a01c::	/48		2001:40e0:bfff:001c::	/64	
2001:40e0:a01d::	/48		2001:40e0:bfff:001d::	/64	
2001:40e0:a01e::	/48		2001:40e0:bfff:001e::	/64	
2001:40e0:a01f::	/48		2001:40e0:bfff:001f::	/64	
2001:40e0:a020::	/48		2001:40e0:bfff:0020::	/64	
……..	/48		……	/64	
2001:40e0:bffe::	/48		2001:40e0:bfff:ffe::	/64	

Begin block	**/128**	**End block**	**/64**
2001:40e0:bfff:ffff:	/128	2001:40e0:bfff:ffff::ffff:ffff:ffff:ffff	128

CIPC network division

IPv6 address /64	Device	Purpose	Device	IPv6 variant 2	Management Segment
TeleCity					
2001:40e0:0000:0010::ff	r1	loopback r1			
2001:40e0:0000:0011::ff	r1	p2p rb1.net.cipc.nl	rb1	2001:40e0:0000:0011::1	
2001:40e0:0000:0012::ff	r1	subnet ns3.net.cipc.nl	ns3	2001:40e0:0000:0012::aa	
2001:40e0:0000:0012::ff	r2	subnet ns3.net.cipc.nl	radius	2001:40e0:0000:0012::7	
2001:40e0:0000:0012::ff	r2	subnet ns3.net.cipc.nl	syslog	2001:40e0:0000:0012::9	
2001:40e0:0000:0012::ff	r2	subnet ns3.net.cipc.nl	timeserver	2001:40e0:0000:0012::b	
CyberCenter					
2001:40e0:0000:0020::ff	r2	loopback r2			
2001:40e0:0000:0021::ff	r2	p2p rb2.net.cipc.nl	rb2	2001:40e0:0000:0021::1	
2001:40e0:0000:0022::ff	r2	subnet ns2.net.cipc.nl	ns2	2001:40e0:0000:0022::aa	
2001:40e0:0000:0022::ff	r2	subnet ns2.net.cipc.nl	radius	2001:40e0:0000:0022::7	
2001:40e0:0000:0022::ff	r2	subnet ns2.net.cipc.nl	syslog	2001:40e0:0000:0022::9	
2001:40e0:0000:0022::ff	r2	subnet ns2.net.cipc.nl	timeserver	2001:40e0:0000:0022::b	
GlobalSwitch					
2001:40e0:0000:0030::ff	r3	loopback r3			
2001:40e0:0000:0031::ff	r3	p2p rb3.net.cipc.nl	rb3	2001:40e0:0000:0031::1	
2001:40e0:0000:0032::ff	r3	subnet ns1.net.cipc.nl	ns1	2001:40e0:0000:0032::aa	fec0:40e0:0000:0032::aa
2001:40e0:0000:0032::ff	r3	subnet ns1.net.cipc.nl	radius	2001:40e0:0000:0032::7	
2001:40e0:0000:0032::ff	r3	subnet ns1.net.cipc.nl	syslog	2001:40e0:0000:0032::9	
2001:40e0:0000:0032::ff	r3	subnet ns1.net.cipc.nl	timeserver	2001:40e0:0000:0032::b	

Device recognition table:

numbering last hexadecimal values	
Redback	01
Radius	07
Syslog	09
Timeservers	0b
DNS	aa
Router	ff

CIPC network

IP blok		
2001:40e0:0000:1000::	/64	CIPC Corp
2001:40e0:0000:2000::	/64	CIPC NOC

Management Network CIPC

fec0:0000:0000:0010::/64	**Management address space Telecity**	**Port**	**Location**
fec0:0000:0000:0010::1	SMS	4/0	Telecity
fec0:0000:0000:0010::d	Switch	managementport	Telecity
fec0:0000:0000:0010::ff	Core 1	fastethernet1/0.1000	Telecity
fec0:0000:0000:0020::/64	**Management address space Cybercenter**		
fec0:0000:0000:0020::1	SMS	2/0	Cybercenter
fec0:0000:0000:0020::d	Switch	managementport	Cybercenter
fec0:0000:0000:0020::54	Radius 4		Cybercenter
fec0:0000:0000:0020::ff	Core 2	fastethernet1/0.1000	Cybercenter
fec0:0000:0000:0030::/64	**Management address space Globalswitch**		
fec0:0000:0000:0030::1	SMS		Globalswitch
fec0:0000:0000:0030::d	Switch	managementport	Globalswitch
fec0:0000:0000:0030::54	Radius 4		Globalswitch
fec0:0000:0000:0030::ff	Core 3	fastethernet6/0.1001	Globalswitch

Tunnel endpoints

IP blok /64	**Klantnaam**
2001:40e0:0001:0000::	tunneltest
2001:40e0:0001:0001::	tunnel mike
2001:40e0:0001:0002::	
2001:40e0:0001:0003::	
2001:40e0:0001:0004::	
2001:40e0:0001:0005::	
2001:40e0:0001:0006::	
2001:40e0:0001:0007::	
2001:40e0:0001:0008::	
……..	
2001:40e0:0001:fffe::	

Appendix E

IPV6 PREFIX-LISTS AND ROUTE MAPS

The announce list and bogons

This prefix list makes sure only the CIPC IPv6 block will be announced in BGP sessions. It is also needed to filter out undesired prefixes.

IPv6 prefix-list announce-ip6 seq 5 permit 2001:40E0::/32
IPv6 prefix-list announce-ip6 seq 10 deny ::/0 le 12

The IPv6-bogons list makes sure only certain prefixes are allowed:

IPv6 prefix-list IPv6-ebgp-permitted description IPv6-bogons
IPv6 prefix-list IPv6-ebgp-permitted seq 5 permit 3FFE::/18 ge 24 le 48
IPv6 prefix-list IPv6-ebgp-permitted seq 10 permit 3FFE:4000::/18 ge 32 le 48
IPv6 prefix-list IPv6-ebgp-permitted seq 15 permit 3FFE:8000::/22 ge 28 le 48
IPv6 prefix-list IPv6-ebgp-permitted seq 20 deny 2001:DB8::/32 le 128
IPv6 prefix-list IPv6-ebgp-permitted seq 25 permit 2001::/16 ge 19 le 48
IPv6 prefix-list IPv6-ebgp-permitted seq 30 permit 2002::/16
IPv6 prefix-list IPv6-ebgp-permitted seq 35 deny ::/0 le 128

Route maps

Route maps are used to set a preference on a path of a route. At CIPC it was used to make AMS-IX traffic preferred over IPv6 transit to NTT Verio. The standard preference in a Cisco router for BGP traffic is 100. If the route map has a preference higher the 100 it is preferred over the standard.

route-map amsix-ip6-in permit 10
description AMSIX IP6 transit
set local-preference 255

Setting up a BGP peer on the AMS-IX

Setting up a BGP peers requires both the prefix lists and the route map. On the AMS-IX it is also recommended to send the *next-hop-self* command with. The command ensures that your router is the next-hop for the prefixes you receive on the AMS-IX. The *soft-reconfiguration inbound* command ensures that not the whole

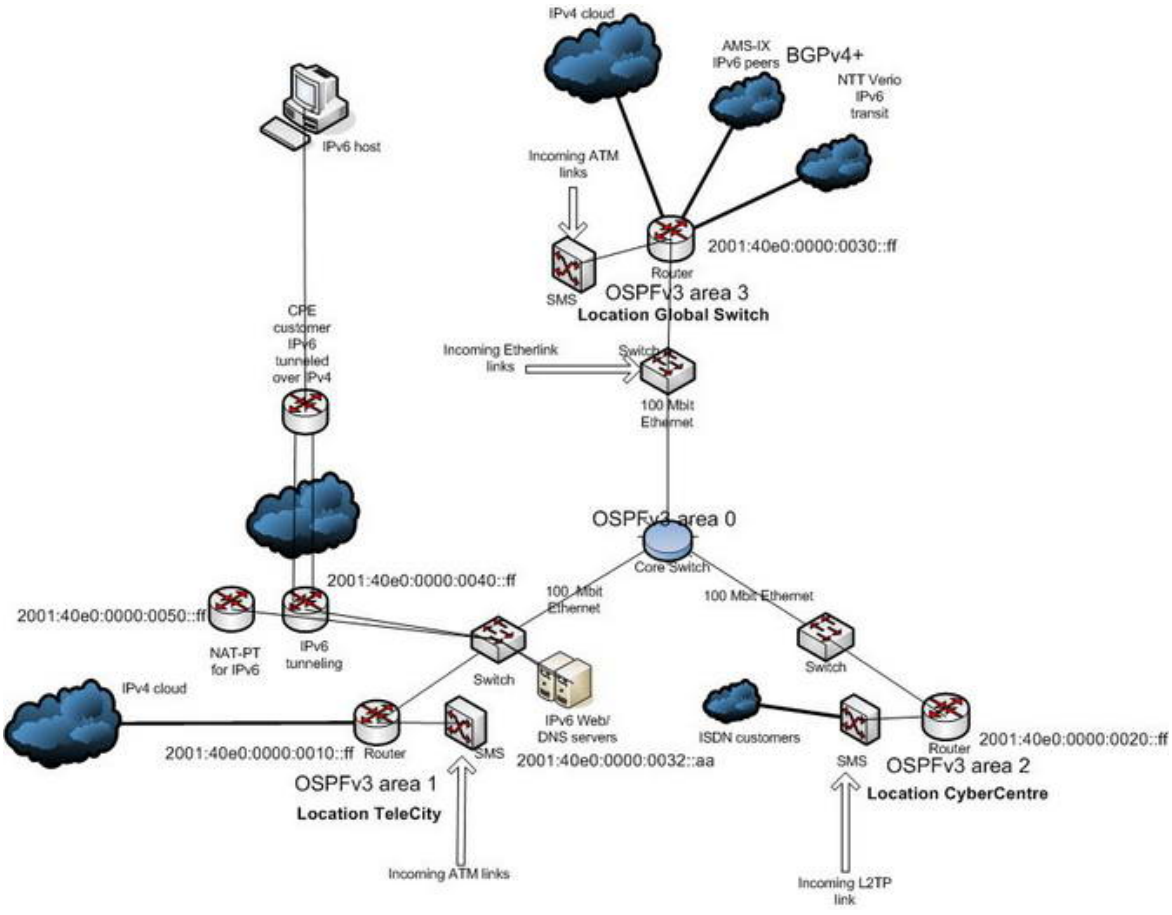
BGP session with a peer has to set up when something in BGP changes at either site.

neighbor peers-amsix-ip6 activate
neighbor peers-amsix-ip6 next-hop-self
neighbor peers-amsix-ip6 soft-reconfiguration inbound
neighbor peers-amsix-ip6 prefix-list IPv6-ebgp-permitted in
neighbor peers-amsix-ip6 prefix-list announce-ip6 out
neighbor peers-amsix-ip6 route-map amsix-ip6-in in

Appendix F

THE FINAL PICTURE

Figure 10: The CIPC Network



INDEX