

# Scriptie

## *IPv6 multihoming*

Opleiding: elektrotechniek/telematica, HVU



Afstudeerbedrijf: Amsterdam Internet Exchange



Naam: Andree Toonk  
Studentnummer: 1109024  
Datum: 20-05-2003  
Examinator: Dhr Uiterwijk  
Bedrijfsbegeleider: Henk Steenman

---

# FACULTEIT NATUUR & TECHNIEK

Elektrotechniek/Telematica



---

## AFSTUDEEROPDRACHT 2002-2003

---

**Titel:** IPv6 multihoming

### Examinandus

Naam : Andree Toonk  
student-nummer : 1109024  
Email-adres : Andree@Toonk.nl  
tel-nummer : 06-22790396

### Examinator

Naam : Kees Uiterwijk  
Email-adres : C.Uiterwijk@fnt.hvu.nl  
tel-nummer : 030 2308184

### Bedrijfsgegevens

Naam	: AMS-IX bv (Amsterdam Internet exchange)	
Afdeling	:	<b>Bedrijfsbegeleider</b>
Straat/nr	: Westeinde 12	Naam : Henk Steenman
Postcode/plaats	: 1017 NZ Amsterdam	Functie : technisch directeur
Land	: Nederland	tel-nummer : +31 20 305 89 99
		Email-adres : Henk.Steenman@ams-ix.net

### Omschrijving van de opdracht:

#### Inleiding

IPv6 is de nieuwe versie van het Internet protocol and de opvolger van de huidige versie IP versie 4 (IPv4). Zie RFC 1883 en andere relevante RFCs.

De Amsterdam Internet Exchange (AMS-IX) is op dit moment bezig met de introductie van IPv6 in haar services. Activiteiten die op dit gebied plaatsvinden zijn het beschikbaar maken van het Exchange platform voor IPv6 en het beschikbaar maken van de ondersteunende diensten (zoals webserver, e-mail, netwerk management etc) onder IPv6.

#### Opdracht.

In het kader van het bovenstaande moet er aan de volgende zaken gewerkt worden.

1: Het beschikbaar maken van de beheers ondersteunende software voor IPv6.

In de praktijk betekent dit:

- De AMS-IX webserver moet beschikbaar moet zijn voor IPv6 en op IPv6 queries antwoorden.
- De AMS-IX mailserver moet beschikbaar zijn op IPv6.
- De verschillende mailinglijsten die gebruikt worden om met groepen van leden en klanten van AMS-IX te communiceren, dienen beschikbaar gemaakt te worden voor IPv6
- Integratie van IPv6 in het AMS-IX infrastructuur management platform

2: Een van de grootste problemen met IPv6 op het moment is het vorm geven aan een oplossing voor zogenaamd "multi-homing" van een site. Dit betekent dat een site haar IP services niet afhankelijk wil laten zijn van één upstream provider, maar van verschillende. Op het moment is het niet duidelijk hoe dat te doen in IPv6 in het geval een site niet beschikt over een zogenaamde "/32". Dit is een stuk adres ruimte dat globaal routeerbaar is. Omdat AMS-IX niet over een dergelijke adres ruimte beschikt is het noodzaak om een oplossing voor multi-homing te vinden. Dit is een erg belangrijk onderwerp voor AMS-IX. De oplossing ligt voor ons of politiek waarbij wij toch een stuk global routable adres space krijgen of technisch waarbij alle machines adressen krijgen van verschillende providers. De afstudeer opdracht richt zich met name op dat laatste. Hoe meerdere adressen te gebruiken op machines en met name, hoe adres selectie toegepast dient te worden en hoe dit te implementeren (wanneer gebruiken we welke adressen en hoe doen we dat ?).

## Samenvatting

De huidige exponentiële groei van het Internet en het dreigende tekort aan IPv4 adressen, heeft tot gevolg dat IPv4 niet langer voldoet. Daarom is IPv6 ontwikkeld, IPv6 is de laatste versie van het Internet Protocol en is ook wel bekend als IPng. Er zijn een aantal grote verschillen tussen IPv4 en IPv6, het meest opvallende verschil is de grote van adressen, dit zou het adres tekort op Internet moeten oplossen.

De IPv6 header is een stuk gestroomlijnder dan de IPv4 header, alle niet strikt noodzakelijk velden zijn uit de basic-header weggelaten en zijn vervangen door extensie headers.

IPv6 is voor eindgebruikers een stuk eenvoudiger geworden door de autoconfiguratie, dit is een vereenvoudigde versie van DHCP. Een ander groot voordeel van IPv6 is dat broadcasts niet meer bestaan, deze functie is overgenomen door multicast adressen, dit komt de performance van het netwerk ten goede. In IPv6 zijn verschillende soorten adressen, zo zijn er link-local adressen, site-local adressen en adressen welke op het hele Internet te gebruiken zijn (global adressen).

Het ICMP protocol heeft voor IPv6 een metamorfose ondergaan, deze is in de nieuwe versie een stuk krachtiger geworden en heeft een aantal belangrijke nieuwe functies gekregen, o.a. de functie van ARP zoals we dat in IPv4 kennen.

Het netwerk van de Amsterdam Internet Exchange is nu geheel dualstack. Alle netwerk componenten en de belangrijkste servers zijn nu beschikbaar voor IPv6. Op de routers zijn IPv6 access-lists geïmplementeerd en BGP is geconfigureerd voor IPv6 prefixen. Servers als de DNS, Web en SMTP server zijn nu beschikbaar voor IPv6.

Een groot probleem met IPv6 is multihoming, de manier zoals dit in IPv4 werd gedaan is door schaalbaarheidsproblemen niet mogelijk in IPv6. Er zijn verschillende voorstellen gedaan om multihoming in IPv6 toch mogelijk te maken. Elk voorstel heeft zijn eigen specifieke kenmerken en voor- en nadelen. Het probleem met de meeste van deze voorstellen is dat deze op dit moment nog niet in te voeren zijn.

Om multihoming bij AMS-IX toch mogelijk te maken, wordt gebruik gemaakt van meerdere adressen per host. Hierbij dienen zich echter wel een aantal problemen aan zoals, wanneer moet welk adres gebruikt worden en hoe wordt gedetecteerd dat een bepaald adres niet meer te gebruiken is. Om deze problemen op te lossen is een script geschreven, welke deze processen op de juiste wijze kan beïnvloeden. Hoewel deze oplossing nog steeds niet helemaal optimaal is, is het totdat de “wijze mannen” van IETF iets beters bedacht hebben, een werkbare oplossing.

## Voorwoord

Deze scriptie dient ter afsluiting van mijn Afstudeer stage bij de Amsterdam Internet Exchange, te Amsterdam. IPv6 zal in de komende paar jaren zijn intrede gaan doen in allerhande netwerken. Ik heb bij de Amsterdam Internet Exchange de kans gekregen om in een professionele omgeving ervaring op te doen met het nieuwe Internet protocol.

Mijn dank gaat dan ook uit naar referenten en collega's, die mij regelmatig met adviezen en opbouwende kritiek hebben bijgestaan. In het bijzonder gaat mijn dank uit naar Arien Vijn, waar ik voor alle vragen mbt IPv6 terecht kon en Steven Bakker bij wie ik voor al mijn vragen over Linux/Unix terecht kon.

Uiteraard wil ik ook Henk Steenman, mijn begeleider bij AMS-IX bedanken en Dhr. Uiterwijk als begeleider namens de opleiding.

Last but not least, mijn jaargenoten waarmee altijd van gedachten gewisseld kon worden.

Andree Toonk  
Amsterdam, mei 2003

Het bestuur van de Stichting Hogeschool van Utrecht aanvaardt geen aansprakelijkheid voor de schade voortvloeiende uit het gebruik van enig gegeven, hulpmiddel of procédé in dit verslag beschreven.

Vermenigvuldiging zonder toestemming van zowel de opleiding E/T van de Hogeschool van Utrecht als de auteur is niet toegestaan.

# Inhoud

<b>INLEIDING</b> .....	<b>5</b>
<b>2 VERSCHILLEN MET IPV4</b> .....	<b>8</b>
<b>3 IPV6 HEADER</b> .....	<b>9</b>
<b>4 IPV6 HEADER VS IPV4 HEADER</b> .....	<b>11</b>
5.1 HOP-BY-HOP OPTION HEADER.....	14
5.2 DESTINATION OPTION HEADER .....	15
5.3 ROUTING HEADER .....	15
5.4 FRAGMENT HEADER .....	15
5.5 AUTHENTICATION HEADER (AH).....	17
5.6 ENCAPSULATION SECURITY PAYLOAD-HEADER (ESP).....	17
<b>6 IPV6 ADRESSERING</b> .....	<b>18</b>
6.1 IPV6 PREFIX .....	19
6.2 IPV6 ADRES TYPE .....	19
6.3 AGGREGATABLE GLOBAL UNICAST ADDRESSES.....	21
6.4 LINK-LOCAL UNICAST ADRESSEN.....	23
6.5 SITE-LOCAL UNICAST ADRESSEN.....	23
6.6 MULTICAST ADRESSEN .....	24
6.7 INTERFACE ID .....	26
6.8 VERPLICHTE ADRESSEN .....	27
<b>7 ICMP V6</b> .....	<b>28</b>
NEIGHBOR DISCOVERY .....	28
<b>8 HET AMS-IX NETWERK</b> .....	<b>33</b>
8.1 AMS-IX NETWERK.....	33
8.2 NETWERKEN OP HET SWITCH PLATFORM .....	34
8.3 AMS-IX NETWERKEN .....	38
8.4 HET AMS-IX NETWERK IN IPV4 .....	39
8.5 IPV4 ADRESSERING IN DE EXCHANGE NETWERKEN.....	40
8.6 IP ADRESSERING IN DE OVERIGE AMS-IX NETWERKEN .....	41
8.7 ROUTERING .....	42
<b>9 IPV6 INTEGRATIE IN HET AMS-IX NETWERK</b> .....	<b>44</b>
9.1 ROUTER .....	44
9.3 MAILSERVERS .....	48
9.4 DNS .....	50
9.5 OVERIGE SERVICES.....	52
9.6 IPV6 ADRESSERING .....	55
<b>10 MULTIHOMING</b> .....	<b>57</b>
10.1 VERSCHILLENDE SOORTEN MULTIHOMING .....	57
10.2 REDENEN VOOR MULTIHOMING .....	58
<b>11 MULTIHOMING MET IPV4</b> .....	<b>60</b>
<b>12 MULTIHOMING MET IPV6</b> .....	<b>62</b>
<b>13 POTENTIËLE IPV6 MULTIHOMING OPLOSSINGEN</b> .....	<b>64</b>
13.1 AANPASSINGEN IN DE TRANSPORT LAAG .....	65
13.2 IDENTIFIERS EN LOCATORS .....	67
13.3 MOBILE IPV6 .....	70
13.4 GEOGRAFISCHE ADRES TOEWIJZING .....	72
13.5 PROVIDER INDEPENDENT ADDRESSING GEBASSEERD OP AS NUMMER .....	74
13.6 EXCHANGE BASED AGGREGATION .....	74
13.7 SAMENWERKING TUSSEN VERSCHILLENDE ISP'S .....	75

---

<b>14</b>	<b>MULTIHOMING BIJ AMS-IX</b> .....	<b>76</b>
14.1	INGRESS FILTERING .....	76
14.2	SOURCE ADDRESS SELECTION .....	78
14.3	DESTINATION ADDRESS SELECTION .....	80
14.4	ESTABLISHED CONNECTIONS .....	82
14.5	MULTIHOMING SCRIPT .....	83
<b>15</b>	<b>CONCLUSIE</b> .....	<b>85</b>
	<b>REFERENTIES</b> .....	<b>87</b>
	<b>BIJLAGE I</b> .....	<b>90</b>
	<b>BIJLAGE II</b> .....	<b>103</b>
	<b>BIJLAGE III</b> .....	<b>110</b>

## Inleiding

IPv6 is de nieuwe versie van het Internet protocol en de opvolger van de huidige versie IP versie 4 (IPv4); IPv6 is beschreven in RFC 1883. De Amsterdam Internet Exchange (AMS-IX) is op dit moment bezig met de introductie van IPv6 in haar services. Activiteiten die op dit gebied plaatsvinden zijn het beschikbaar maken van het Exchange platform voor IPv6 en het beschikbaar maken van de ondersteunende diensten (zoals webserver, e-mail, netwerk management etc) onder IPv6.

De opdracht is te verdelen in twee onderdelen:

### *Integratie van IPv6 in de AMS-IX infrastructuur*

Het is eerste punt, is het beschikbaar maken van netwerk en de servers voor IPv6. In de praktijk betekent dit dat de AMS-IX webserver moet beschikbaar moet zijn voor IPv6 en op IPv6 queries antwoorden. De AMS-IX mailserver moet beschikbaar zijn op IPv6. De verschillende mailinglijsten die gebruikt worden om met groepen van leden en klanten van AMS-IX te communiceren, dienen beschikbaar gemaakt te worden voor IPv6 Integratie van IPv6 in het AMS-IX infrastructuur management platform.

### *Ipv6 multihoming*

Een van de grootste problemen met IPv6 op het moment is het vorm geven aan een oplossing voor zogenaamd "multi-homing" van een site. Dit betekent dat een site haar IP services niet afhankelijk wil laten zijn van één upstream provider, maar van verschillende.

Op het moment is het niet duidelijk hoe dat te doen in IPv6 in het geval een site niet beschikt over een zogenaamde "/32". Dit is een stuk adres ruimte dat globaal routeerbaar is. Om een /32 te krijgen moet een organisatie aan bepaalde eisen voldoen, dit heeft voornamelijk te maken met het aantal klanten wat de organisatie bediend. Denk hierbij bijvoorbeeld aan een ISP met klanten die allemaal een ipadres moeten hebben. Afhankelijk van het aantal klanten wordt een adresruimte toegewezen. Op dit moment is de route aggregatie op Internet (met IPv6) .service provider based., de ISP.s (Internet Service Providers) krijgen een blok adres ruimte van een Regional Internet Registries (RIRs) en kennen hun klanten een reeks toe uit die adres ruimte. De ISP.s sturen vervolgens slechts een route voor elk toegewezen adres block naar andere netwerken. Deze vorm van route aggregatie maakt het mogelijk, om miljoenen organisaties te verbinden met het Internet en er tevens voor te zorgen dat de entry's in de route tabellen gelimiteerd blijven tot net iets meer dan honderd duizend destination prefixes.

Helaas, werkt provider-based aggregation niet met netwerken die meer dan een connectie naar het Internet hebben (multihomed). In de IPv4 omgeving wordt dit opgelost met provider onafhankelijke adresruimte, welke over alle twee de uplinks wordt bekend gemaakt. Het netwerk is dan in de global routing table bekend en bereikbaar via verschillende AS-PATHs. In IPv6 bestaat er geen provider indepedended adres ruimte, althans niet voor kleinere organisaties, zoals AMS-IX. Als dit wel zou worden toegestaan, zouden organisaties bijv hun eigen /48 (relatief kleine reeks) gaan adverteren, dan zullen de route tabellen zo groot worden dat het voor routers onwerkbaar wordt. Er staan dan zoveel prefixen in de route tabellen, dat het te lang zou duren om alle routes door te rekenen.



Omdat AMS-IX geen globaal routeerbare adres reeks krijgt toegewezen, is het noodzaak om een oplossing voor multi-homing te vinden. Dit is een erg belangrijk onderwerp voor AMS-IX. De oplossing ligt of politiek waarbij wij toch een stuk global routable adres space krijgen of technisch waarbij alle machines adressen krijgen van verschillende providers. De afstudeer opdracht richt zich voornamelijk op dat laatste. Hoe meerdere adressen te gebruiken op machines en hoe adres selectie toegepast dient te worden en hoe dit te implementeren (wanneer worden welke adressen gebruikt en hoe wordt dit gerealiseerd?).

## 1 IPV6 protocol

IPv6 is de laatste versie van het Internet Protocol en is ook wel bekend als IPng (next generation). Sinds het ontstaan van Internet wordt er met IP gewerkt als netwerk (layer3) protocol, de huidige versie van het protocol, versie 4, is nauwelijks gewijzigd sinds de publicatie in RFC 791 in 1981. IPv4 heeft zich in al die jaren bewezen als een robuust en simpel protocol welke eenvoudig toe te passen is in allerlei netwerken. We zijn bekend geworden, met het ontwerpen en “troubleshooten” van dit netwerk, echter aan alle mooie dingen komt een eind, zo ook aan IPv4.

De huidige exponentiële groei van het Internet en het tekort aan IPv4 adressen, heeft tot gevolg dat IPv4 niet langer voldoet. Niet alleen het tekort aan IPv4 adressen was een rede voor een nieuwe versie van IP, ook was er behoefte aan andere zaken zoals: eenvoudige configuratie, security op IP niveau en betere ondersteuning voor “real time” verkeer(QOS). Al deze zaken hebben er toe geleid dat er een nieuw protocol is ontwikkeld.

Al in het begin van de jaren negentig, is een begin gemaakt door “the Internet Engineering Task Force” (IETF), met het ontwikkelen van een nieuwe versie van het IP protocol. Door verschillende mensen is een voorstel gedaan voor de opvolger van IPv4, al deze initiatieven zijn bestudeerd door de IETF. Uiteindelijk is in 1994 een advies uitgebracht door de Network Working Group, met daarin een aanbeveling voor een nieuw protocol. De aanbeveling beschrijft onder andere een eenvoudigere IP-header en een hiërarchisch adres structuur welke vergaande route-aggregatie toestaat en tevens groot genoeg is voor de verwachte groei van Internet. Deze aanbevelingen zijn beschreven in RFC 1752. De uiteindelijke protocol specificaties zijn in december 1995 gepubliceerd in RFC 1883, deze RFC heeft als titel: “Internet Protocol, Version 6 (IPv6) Specification”. Een logische vraag zou zijn, “waarom versie 6, wat is er met versie 5 gebeurd?”. IPv5 was al in gebruik voor een experimenteel streaming protocol (RFC1819), vandaar het versie nummer 6.

Op dit moment begint IPv6 langzaam te integreren in de productienetwerken. Het Surfnet netwerk is hier een goed voorbeeld van, SURFnet5 is al een tijdje dual stack (zowel IPv4 als IPv6). Een tijd lang is er geëxperimenteerd met IPv6 op SIXBONE, dit is een experimenteel IPv6 netwerk, waarop bedrijven konden aansluiten om ervaring op te doen met IPv6. Veel bedrijven verlaten nu SIXBONE (3FFE:: adressen) en gaan over naar productie adressen (2001:: adressen). Kortom: IPv6 komt eraan!

## 2 Verschillen met IPv4

Er zijn een aantal grote verschillen tussen IPv4 en IPv6, de belangrijkste verschillen worden hierna in het kort behandeld.

De voornaamste reden voor een nieuwe versie van IP, was de te krappe adres ruimte in IPv4. In IPv4 zijn er 32 bits beschikbaar voor adressen en hoewel hiermee theoretisch ruim 4 miljard hosts zijn te adresseren, blijkt dit getal in de praktijk lang niet haalbaar. Om de adresruimte efficiënter te gebruiken, is CIDR (Classless InterDomain Routing) ingevoerd en zijn technieken als NAT ontwikkeld. Dit is slechts een tijdelijke oplossing, een definitieve oplossing zou de invoering van IPv6 moeten zijn. Deze voorziet in 128 bits adresruimte, dit zou ruim voldoende moeten zijn voor de voorziene groei.

De IPv6 header is een gestroomlijnde versie van de IPv4 header, velden welke niet of nauwelijks worden gebruikt zijn verwijderd, of optioneel geworden. Er zijn velden toegevoegd voor betere ondersteuning voor real-time traffic. De headers van IPv6 pakketten hebben een vaste lengte van 8 bytes, plus 2 keer 16 bytes (128bits) adressen voor de source en destination, de totale lengte komt daarmee op 40bytes. Dit is 2 maal zo groot als de IPv4 header, deze is 20bytes groot. Toch is de IPv6 header eenvoudiger dan de IPv4 header, want hoewel de adressen 4 keer zo groot zijn, is de totale header slechts 2 maal zo groot.

De IPv4 header heeft een speciaal veld voor optionele eigenschappen. Dit “option field” was bedoeld voor oa. security opties en source routing, deze opties worden in de praktijk niet of nauwelijks gebruikt. In IPv6 daarentegen is geen option veld, maar wordt gebruik gemaakt van zogenaamde extensions headers. Deze worden na de “mainheader” toegevoegd, een logisch gevolg is, dat er geen length field meer nodig is, omdat de mainheader altijd een vaste lengte heeft.

IPv6 biedt native support voor authenticatie en voorziet in functies waarmee de data integriteit kan worden gewaarborgd. Bij de ontwikkeling van IPv6 is er veel aandacht besteed aan beveiliging op IP niveau. Als resultaat daarvan is IPsec ontwikkeld en hoewel deze in eerste instantie is ontwikkeld voor IPv6, wordt deze nu ook gebruikt in IPv4.

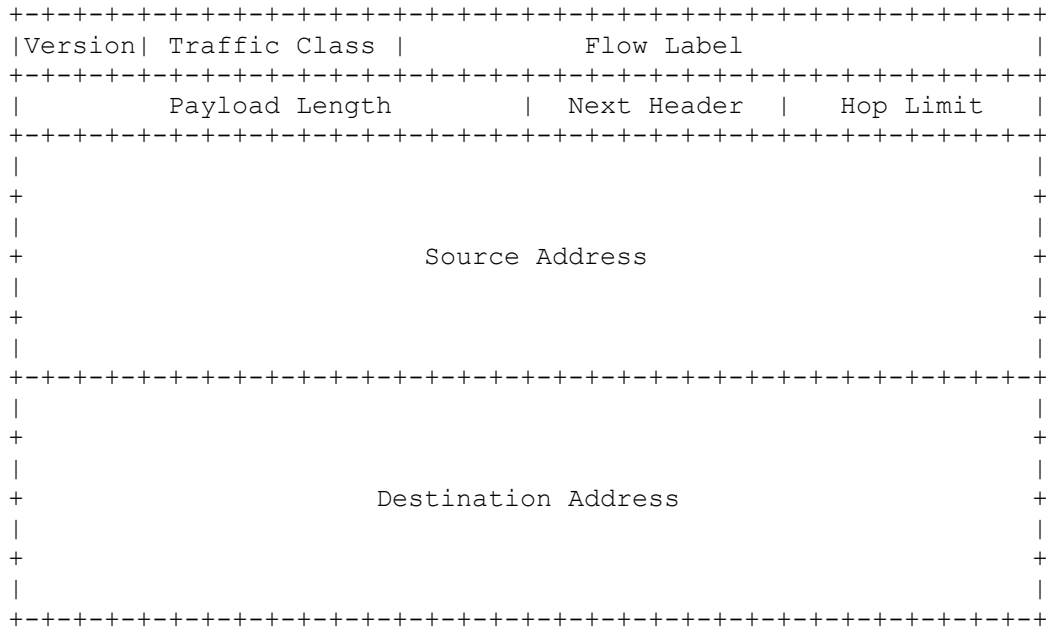
Een belangrijke eis bij de ontwikkeling van IPv6 was, dat het gebruik voor eindgebruikers eenvoudiger moest worden. Daarom voorziet IPv6 in meerdere mogelijkheden voor autoconfiguratie. Zo is er het zogenaamde ‘stateless autoconfiguration’, waarmee iedere hosts zonder kennis van het netwerk een adres kan krijgen (plug and play). Er is ook een mogelijkheid tot statefull autoconfiguration, waarbij gebruik gemaakt wordt van een DHCP server.

### 3 IPv6 Header

Zoals al in het voorgaande hoofdstuk is beschreven, bestaat de IPv6 header uit een vaste lengte van 40 Bytes, waarvan 32 bytes worden gebruikt voor het source en destination adres, er blijven dus nog 8 bytes over voor de overige header informatie.

De meest recente versie van de IPv6 header-specificatie staat beschreven in RFC 2460. [1]

De header lay-out is weergegeven in figuur 1.



**Figuur 1**

Wat de functie van de verschillende velden is, wordt in het volgende gedeelte besproken.

#### **Version**

Dit is een 4 bits veld, met de waarde 6. Dit staat voor IP versie nummer 6.

#### **Traffic Class**

Dit is een 8 bits veld vergelijkbaar met het “type of service” veld, zoals dit in IPv4 bekend is. Dit veld kan worden gebruikt voor realtime data en QOS, de waarde van het veld geeft aan wat voor soort verkeer het pakket bevat. De waarde van dit veld heeft op iedere router een bepaalde prioriteit, wat als gevolg kan hebben dat het pakket behandeld wordt met een bepaalde voorrang of juist niet.

#### **Flow Label**

Het 20 bits flowlabel veld in de IPv6 header, kan door een afzender gebruikt worden om een reeks pakketten van een label te voorzien. Pakketten die gemerkt worden, vereisen allemaal dezelfde behandeling door een router. Het kan bijvoorbeeld allemaal realtime traffic zijn. Routers kunnen hierdoor deze pakketten sneller verwerken, omdat niet de hele header bekeken hoeft te worden, als een flowlabel bekend is, wordt deze meteen geforward.

#### **Payload Length**

De payload lengte geeft aan hoeveel bytes aan data er in het dataveld zit. Dit is een 2 bytes groot veld, wat inhoudt dat de maximale payload 64KB kan zijn. In IPv4 is het vergelijkbare header veld

---

het “Total length field”, in IPv4 wordt de lengte van de header ook meegerekend. Het payload length veld in IPv6, is exclusief de lengte van de header.

### **Next Header**

In IPv4 heet dit veld het protocol veld, in IPv6 heeft dit een wat meer voor de hand liggende naam gekregen, het next header field. Dit veld is een 8bits veld, welke aangeeft welk protocol wordt getransporteerd (bovenliggende protocol). Hiervoor worden dezelfde waarden als in IPv4. Dus voor tcp het getal 6 en udp 17. Als de next header een extension header is, kan aan de hand van de waarde van het next header veld worden bepaald, wat voor type extensionheader dat is.

### **Hop Limit**

Het 8bits hop limit veld is gelijk aan het ttl veld zoals dat in IPv4 bestaat. Dit is een 1byte waarde, iedere router die een pakket forward, vermindert de waarde van dit veld met 1.

Wanneer dit veld de waarde 0 bereikt, wordt het pakket weggegooid; dit heeft als doel dat pakketten niet eindeloos in het netwerk rond blijven zwerven. Wanneer een pakket wordt weggegooid, wordt een ICMP time exceed message verstuurd naar de afzender van het originele pakket.

### **Source Address**

Het 128 bits source address. Weergegeven in hexadecimale getallen.

### **Destination Address**

Het 128 bits destination address. Dit adres kan een unicast, multicast of anycast adres zijn.

## 4 IPv6 header vs IPv4 header

Er zijn een aantal grote verschillen tussen de IPv4 header en de IPv6 header. De IPv6 header is een gestroomlijnde versie van de IPv4 header. Een aantal velden uit de IPv4 header bestaan niet meer in de base-header van IPv6. Het kan zijn dat deze velden geheel verwijderd zijn uit de IPv6 header, of dat deze in zgn. extension headers zijn opgenomen.

In figuur 2 en 3 staan beide headers afgebeeld. Wat direct opvalt, is het verschil in het aantal velden, IPv4 heeft veel velden in vergelijking met IPv6, waardoor de IPv6 header een stuk eenvoudiger te verwerken is door routers.

```

+-----+-----+-----+-----+-----+-----+-----+-----+
|Version| IHL  |Type of Service|           Total Length           |
+-----+-----+-----+-----+-----+-----+-----+-----+
|           Identification           |Flags|           Fragment Offset   |
+-----+-----+-----+-----+-----+-----+-----+-----+
| Time to Live | Protocol |           Header Checksum         |
+-----+-----+-----+-----+-----+-----+-----+-----+
|           Source Address           |
+-----+-----+-----+-----+-----+-----+-----+-----+
|           Destination Address      |
+-----+-----+-----+-----+-----+-----+-----+-----+
|           Options                   |           Padding                 |
+-----+-----+-----+-----+-----+-----+-----+-----+

```

**Figuur 2: IPv4 header format**

```

+-----+-----+-----+-----+-----+-----+-----+-----+
|Version| Traffic Class |           Flow Label           |
+-----+-----+-----+-----+-----+-----+-----+-----+
|           Payload Length           | Next Header | Hop Limit |
+-----+-----+-----+-----+-----+-----+-----+-----+
|
+
|
+           Source Address
|
+
|
+-----+-----+-----+-----+-----+-----+-----+-----+
|
+
|
+           Destination Address
|
+
|
+-----+-----+-----+-----+-----+-----+-----+-----+

```

**figuur 3: IPv6 header format**

Het eerste wat opvalt, is natuurlijk de grootte van de adres velden, welke van 32 bits naar 128 bits zijn vergroot.

Van een aantal velden is slechts de naam veranderd en niet de functie. De naam van het Time to live veld (TTL), is in IPv6 veranderd naar Hop Limit. De grote van dit veld is even groot gebleven (8bits, 255hops).

Het protocol veld zoals dit in IPv4 wordt gebruikt, heeft in IPv6 een andere naam gekregen, het Next Header veld. Ook de naam van het type of service veld is veranderd in Traffic Class.

Een aantal velden welke in de IPv4 header zijn gedefinieerd, komen niet meer in de IPv6 header voor. Velden welke uit de base header zijn verwijderd zijn:

1. Fragmentation field
2. Option field
3. header checksum
4. Header length field

Een opvallend gegeven is het ontbreken van de Fragmentation fields ("Identification", "Flags" en "Offset"). In IPv4 wordt de fragmentatie verzorgd door routers als het next-hop netwerk een kleinere MTU heeft dan het originele netwerk. In IPv6 is deze verantwoordelijkheid verschoven naar de hosts welke de pakketten verzenden (source hosts). Tijdens het begin van de communicatie zal de source host gaan onderzoeken, wat de kleinste mtu waarde in het traject naar het eindpunt is en zal aan de hand daarvan de pakket grote bepalen. Het onderzoeken van de MTU size op een bepaald traject wordt "path MTU discovery" genoemd. Een tweede optie is om niet aan path mtu discovery te doen, maar alleen pakketten te versturen van 1280 bytes. De IPv6 specificaties schrijven namelijk voor dat ieder netwerk een minimale mtu van 1280 bytes moet hebben. Dit kan voor systemen met een minimale IPv6 implementatie (bijv een bootROM) handig zijn, deze kunnen eenvoudig pakketten zenden van 1280 bytes, zonder dat de intelligentie voor pmtu discovery nodig is.

Het option field is verwijderd uit de IPv6 header en extensions headers zijn hiervoor in de plaats gekomen. Door het verwijderen van het variabele optie veld, is de IPv6 header "fixed" geworden, dat wil zeggen dat deze altijd een vaste lengte heeft. Het header lenght field is hiersdoor niet langer noodzakelijk. "fixed headers" zijn bovendien veel makkelijker te verwerken door routers, wat als gevolg heeft dat de verwerking van IPv6 headers aanzienlijk sneller gaat.

Het meest opvallend is wellicht wel het ontbreken van de header checksum. Dit betekent dat de IPv6 headers niet langer gecontroleerd worden op hun juistheid. Dit is gedaan met het oog op de toekomst, men is er vanuit gegaan dat de kwaliteit van de netwerken steeds beter zal worden. Bovendien bevatten de meeste layer2 (datalink) protocollen tegenwoordig een vorm van een error detectie/correctie mechanisme. Het verwijderen van de checksum heeft als logisch gevolg, een snellere verwerking van de pakketten door routers. De verantwoordelijkheid voor de controle wordt nu dus bij boven en onderliggende protocollen neergelegd. De in IPv4 optionele header checksum in UDP is in IPv6 dan ook niet langer optioneel maar verplicht [1].

Een interessant detail is overigens, dat er router fabrikanten zijn die er voor gekozen hebben de IPv4 header checksum niet te controleren. Dit wordt gedaan om de performance van de routers te verhogen.

Er zijn niet alleen velden verdwenen, er is ook een veld bijgekomen, dit is het 20 bits flowlabel veld. Samen met het vroegere "Type of service" (nu Traffic Class) moet het Flow Label Quality of Service mogelijk maken. Traffic Class geeft aan welke gewenste prioriteit dit pakket heeft. Het begrip "flow" moet als volgt geïnterpreteerd worden: "een aantal pakketten die samen één datastroom volgen, vormen een flow". Het flow-label dient ter identificatie van deze stroom en moet het mogelijk maken om resources te preallokeren. Alle pakketten afkomstig van hetzelfde source adres en met hetzelfde flow-label worden geacht tot die flow te behoren. Natuurlijk hangt de implementatie van Quality of Service niet van IPv6 alleen af; IPv6 blijft een pakket-geschakeld netwerkprotocol, dat afhankelijk is van de onderliggende data-link laag.

Al deze wijzigingen in de IPheader, hebben als doel de headers efficiënter in te richten waardoor ze eenvoudiger en dus sneller door routers te verwerken zijn. Eventuele optionele extensies, zijn verwerkt in extension headers, welke in het hoofdstuk 5 behandeld worden.

## 5 extension headers

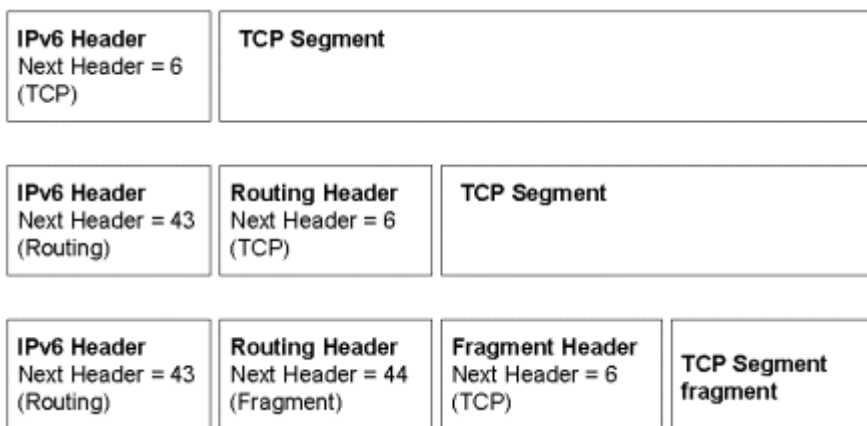
Het “option” veld zoals deze in IPv4 bestaat, bestaat niet langer in de IPv6 header. Hiervoor in de plaats zijn de extension headers gekomen. Het option field, is niet langer opgenomen omdat deze nauwelijks werd gebruikt. De routers moeten dit wel iedere keer verwerken voordat deze het pakket kan forwarden. Dit is erg inefficiënt, vandaar dat dit in IPv6 met behulp van extension headers is opgelost. Op dit moment zijn er zes extension headers gedefinieerd (RFC 2460 [2]), in de toekomst kunnen dit er nog meer worden. Extension headers komen meteen na de basic IPv6 header, dus voor routers die de extension headers niet hoeven te verwerken, zijn deze gewoon een gedeelte van de payload en worden dus ook niet geanalyseerd. Dit verhoogt de forwarding performance van de router. De enige extension header die door iedere router verwerkt dient te worden is de hop-by-hop extension header.

De volgende extension headers moeten door iedere IPv6 host herkend worden:

- Hop-by-Hop Options header
- Destination Options header
- Routing header
- Fragment header
- Authentication header
- Encapsulating Security Payload header

Een “normaal” IPv6 pakket bevat geen extension headers. Als er een speciale behandeling van het pakket nodig is, worden er één of meerdere extension headers toegevoegd. Iedere extension header moet een veelvoud van 8bytes (64bit) groot zijn. Waar nodig kan “padding” worden gebruikt om aan deze eis te voldoen.

In figuur 4 een voorbeeld weergegeven, van een aantal typische IPv6 pakketten.



Figuur 4

Het eerste pakket bevat geen extension headers, het next header field heeft hier de waarde 6, wat betekent dat de payload tcp data bevat. Het tweede pakket is een IPv6 pakket met één extension header, dit wordt aangegeven door het getal 43 in het next header field, dit betekent dat de volgende header een routing header is. De routing header heeft op zijn beurt weer een next header field met de waarde 6, wat betekent dat deze wordt gevolgd door een tcp segment.

Het laatste voorbeeld geeft een IPv6 header weer, gevolgd door een routing header, een fragment header en als laatste een tcp segment.

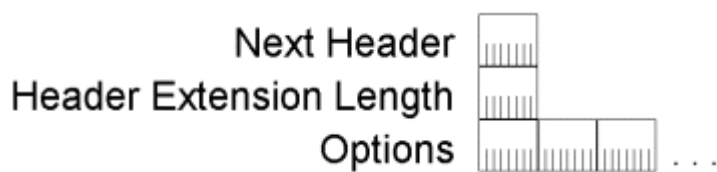


Zoals in het voorgaande voorbeeld is te zien, kan ieder pakket meer dan één header bevatten, iedere header geeft mbv. het next header field aan wat de volgende header is, dit wordt de “header chain” genoemd. Deze header chain wordt behandeld in de volgorde waarin deze binnen komt. Aan deze volgorde zijn regels gesteld. De volgorde dient als volgt te zijn:

- IPv6 header
- hop by hop option header
- destination option header
- routing header
- fragmentation header
- authentication header
- encrypted security payload header
- destination option header
- upper layer header (zoals, TCP of UDP).

### 5.1 Hop-by-hop option header

De hop-by-hop option header bevat optionele informatie, wat door iedere node moet worden verwerkt. De hop-by-hop header wordt aangegeven met een next-header waarde van “0”. In het figuur 5 is de hop-by-hop option header weergegeven.



Figuur 5

De hop-by-hop option header bestaat uit een next header field, een header extension length field en een option field welke één of meerdere opties bevat. Het header extension length field is een 8 bits groot veld, welke de totale grote van de option header weergeeft in bytes.

Het laatste veld is het option veld, de lengte van dit veld is variabel en wordt aangegeven door het header extension length field.

De option extension headers worden gebruikt voor padding. Pad1 en PadN zorgen hier voor de alignment. Een processor die zijn informatie in hapklare blokken krijgt, heeft een veel kortere verwerkingstijd nodig dan wanneer de informatie onhandig over de blokken is verdeeld. Een 32bits processor werkt bijvoorbeeld veel sneller wanneer de informatie in blokken van 32 bit wordt aangeboden. De bovenstaande twee opties voegen respectievelijk 1 en een willekeurig aantal (=N) toe aan het begin van de header zodat het datagram correct uitgelijnd is.

Een derde optie is de jumbo payload length optie. Het veld payload length in de eerste header kan geen waarden groter dan 65535 bytes representeren. Is het datagram echter groter, dan wordt deze waarde op 0 gesteld. Daarmee wordt doorverwezen naar de jumbo payload option, welke de grootte weergeeft, exclusief de 40 bits header.

De Router Alert option (Option Type 5), wordt gebruikt wanneer het pakket speciaal behandeld dient te worden, deze optie wordt onder andere gebruikt door protocollen als het Resource Reservation Protocol (RSVP) of door het Multicast Listener Discovery (MLD)protocol.

## 5.2 Destination option header

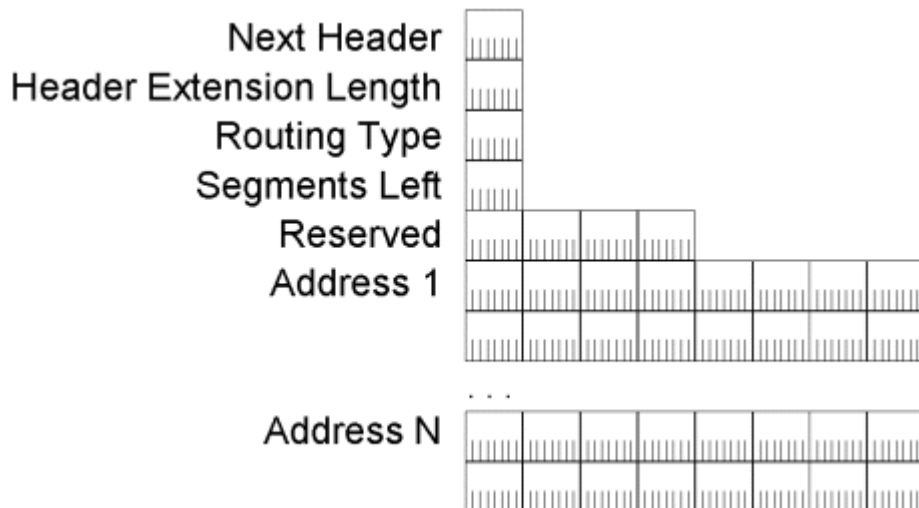
Deze extension header wordt weergegeven door de waarde 60 in het next header field. De destination option header is vergelijkbaar met de hop-by-hop option header. Het verschil is dat de destination option header alleen door de destination wordt verwerkt.

## 5.3 Routing header

Het kan zijn dat de bron van het datagram de bezorging niet aan het netwerk wil overlaten, maar zelf ideeën heeft over hoe het datagram bezorgd dient te worden. Met deze extensie geeft de zender aan via welke route het datagram verzonden moet worden. Momenteel is maar één type routing header gedefinieerd: type 0. Type 0 (beschreven in RFC 1883 [3]) bestaat simpelweg uit een lijst met adressen die de totale route vormen.

De source host maakt een lijst van alle routers die de totale route vormen en geeft als destination address de eerste router uit de lijst. Bij ieder volgend punt wordt nu de destination vervangen door de volgende router in de lijst. Tevens wordt het veld “segments left” met één verlaagt.

Het pakket weer gegeven in figuur 6:



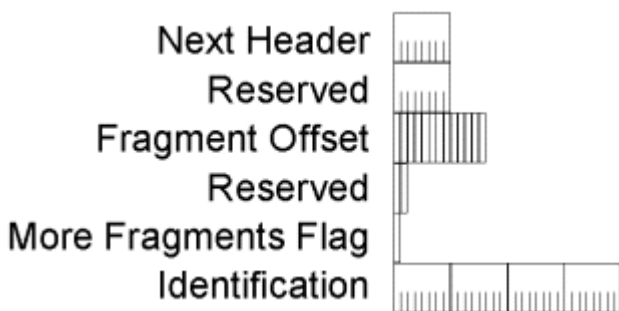
figuur 6

## 5.4 Fragment Header

In IPv6 zijn hosts zelf verantwoordelijk voor de fragmentatie van pakketten. In IPv4 wordt dit gedaan door routers onderweg, maar om de performance van routers te verbeteren is besloten de verantwoordelijkheid voor fragmentatie te verleggen naar de “end-hosts”. Hosts maken daarbij gebruik van een proces wat Path MTU discovery heet. Een host verstuurd standaard pakketten met een mtu-size van de eigen link. Wanneer deze een icmp packet to big bericht terug krijgt van een tussenliggende router, verkleint de host de pakketten naar de grote welke als optie in het icmp bericht is mee gestuurd. De source host gaat er van uit dat het eerste pakket verloren is gegaan en verstuurd het opnieuw met de aangegeven mtu.

Als de payload, welke door de hogere laag aan IPv6 wordt doorgegeven, groter is dan de link MTU of de Path MTU, dan dient dit pakket gefragmenteerd te worden. Dit kan bijvoorbeeld voorkomen bij UDP pakketten. Bij TCP is de kans hierop aanzienlijk kleiner omdat TCP mechanisme als TCP MSS (Maximum Segment Size) gebruikt, dit mechanisme bepaald wat de maximum grote van TCP pakketten mag zijn.

De fragment header ziet er uit zoals in figuur 7 is weergegeven.



figuur 7

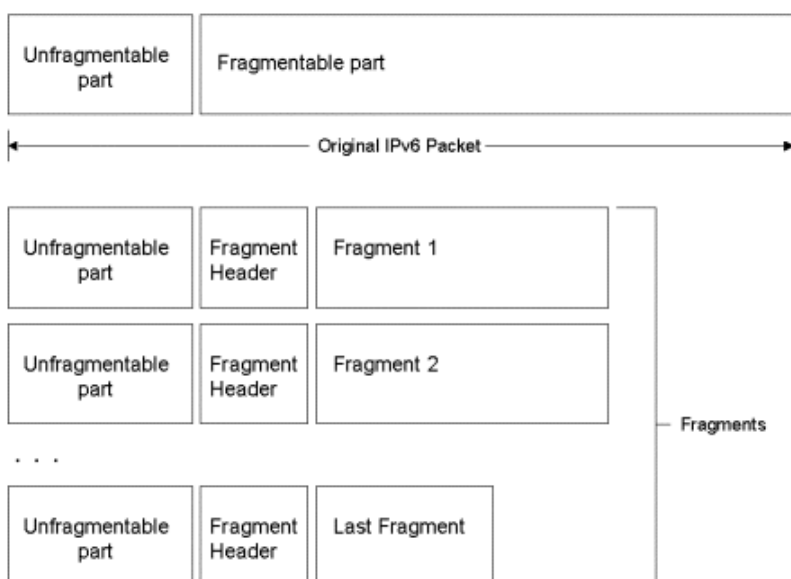
De belangrijkste velden zijn het fragment offset veld, de M vlag en de identification. Het fragment offset veld bevat de offset waarde vanaf het eerste pakket (in eenheden van 8bytes). In feite wordt heermee bedoeld hoeveel data er al verzonden is, als dit pakket het eerste gedeelte van een gefragmenteerd pakket is, zal de offset de waarde 0 hebben. Een tweede pakket zal een waarde van 175 kunnen hebben, wat dus betekend dat er al 1400 bytes (175\*8bytes) verzonden zijn. De M vlag, staat voor “More Fragments”. Als dit veld een waarde van 1 heeft, betekend dit dat er nog meer pakketen komen. Als dit veld een waarde van 0 heeft, betekend dit, dat dit het laatste gefragmenteerde pakket is.

Het Identification veld (32 bits) geeft aan bij welk originele pakket een gefragmenteerd gedeelte hoort. Zo kan de ontvanger meerdere ontvangen gefragmenteerde pakketen onderscheiden. Gefragmenteerde pakketten die bij hetzelfde originele pakket horen, hebben hetzelfde ID.

Wanneer een IPv6 pakket gefragmenteerd wordt, wordt deze verdeeld in een gedeelte wat gefragmenteerd kan worden en een gedeelte wat niet gefragmenteerd kan worden.

Het “unfragmentable” gedeelte moet worden verwerkt door iedere router op de weg naar de eind bestemming. Dit gedeelte bestaat uit de IPv6 header, de hop-by-hop option header en de routing header. Het “fragmentable” gedeelte van het originele IPv6 pakket, hoeft alleen op de uiteindelijke bestemming te worden verwerkt.

Een gefragmenteerd IPv6 pakket bestaat dus uit een unfragmentable gedeelte, de fragment header en een gedeelte van het fragmentable stuk. Schematisch ziet dit er als volgt uit, zie figuur 8:



figuur 8

### **5.5 Authentication Header (AH)**

Deze header vormt een mechanisme dat een cryptografische checksum berekent over onderdelen van de IPv6 header, de extensionheaders en de payload. Met behulp van de Authenticatie header, weet een ontvanger zeker dat het pakket van de originele afzender komt.

### **5.6 Encapsulation Security Payload-header (ESP)**

Deze header is altijd de laatste, niet versleutelde header van een pakket. Het geeft aan dat de rest van het pakket is versleuteld en verstrekt de geautoriseerde destination host voldoende informatie om de payload te kunnen ontcijferen.

## 6 IPv6 Adressering

Het meest opvallende verschil tussen IPv4 en IPv6 is de lengte van de adressen. De adressen zijn in IPv6 niet alleen een stuk langer geworden, van 32 bits naar 128 bits, ze zien er zo op het eerste gezicht ook heel anders uit. De IPv4 notatie is decimaal, de notatie voor IPv6 is hexadecimaal. Dit is gedaan omdat als IPv6 adressen decimaal zouden worden genoteerd, het adres erg lang zou worden. Door gebruik te maken van hexadecimale notering wordt het adres wat korter en wat overzichtelijker. Dit resulteert in minder kans op fouten.

In het onderstaande voorbeeld worden alle mogelijke DNS entry's opgevraagd die voor melix.ams-ix.net bestaan:

```
host -t any melix.ams-ix.net
melix.ams-ix.net      AAAA    2001:610:140:A604:0:0:0:3
melix.ams-ix.net      AAAA    2001:7B8:200:A604:0:0:0:3
melix.ams-ix.net      A       193.194.136.3
```

Zoals te zien is bestaan er 2 soorten DNS entry's voor melix.ams-ix.net, namelijk twee keer een AAAA record voor de IPv6 adressen en een A record voor het IPv4 adres.

Er is meteen een verschil te zien tussen de twee soorten adressen, het IPv6 adres is een stuk langer dan het IPv4 adres en in plaats van een punt als scheidings teken wordt er een dubbele punt gebruikt.

Omdat de adressen ondanks dat er een hexadecimale notatie wordt gebruikt, nog steeds erg lang zijn, zijn er een aantal regels waarmee de adressen kunnen worden afgekort.

Het volledige IPv6 adres voor melix.ams-ix.net is:

**2001:0610:0140:A604:0000:0000:0000:0003**

Het adres wordt dus weergegeven in acht velden van ieder 16 bit (hexadecimaal).

Er zijn een tweetal regels waarmee adressen kunnen worden ingekort,

1. bij elke veld mogen de begin cijfers "0" (leading zeros) worden weggelaten  
een voorbeeld hiervan is:

**2001:610:140:A604:0:0:0:3** ipv **2001:0610:0140:A604:0000:0000:0000:0003**

Het adres wordt hierdoor al een stuk korter en dus overzichtelijker.

2. een aantal opvolgende nullen, mag worden vervangen door "::"  
een voorbeeld hiervan is:

**2001:610:140:A604::3** ipv **2001:610:140:A604:0:0:0:3**

Hierdoor wordt het adres weer een stuk korter, de "::" houd in het adres moet worden aangevuld met nullen, tot dat het adres uit 128 bits bestaat. Dit is ook meteen de reden dat dit maar één keer kan worden toegepast in een adres. Wanneer deze afkorting vaker zou worden gebruikt weet een resolver niet hoeveel nullen op welke plaats horen.

Door deze twee mogelijkheden tot het afkorten van adressen, worden IPv6 adressen een stuk handelbaarder, echter ze blijven in de meeste gevallen een stuk langer dan de IPv4 adressen.

## 6.1 IPv6 Prefix

De IPv6 prefix is het linker gedeelte van het IPv6 adres, de prefix representeert het netwerk adres. Een IPv6 prefix wordt net zo weergegeven zoals in de IPv4/CIDR notatie. Het begint met een netwerk ID gevolgd door de lengte van het netwerk ID in bits.

Een voorbeeld van hoe dit in IPv4 wordt gedaan is: 192.168.3.0/24, wat betekent dat het netwerk adres 192.168.3.0 is en een lengte van 24 bits heeft. In de IPv6 situatie ziet dit er vergelijkbaar uit: 2001:610:140:A604::/64, dit betekent dat het netwerk ID 2001:610:A604 is en een lengte van 64 bits heeft, wat betekent dat er nog 64 bits over blijven voor host op dit netwerk.

## 6.2 IPv6 adres type

In IPv4 kennen we drie soorten adressen, namelijk unicast, multicast en broadcast. In IPv6 bestaan geen broadcast adressen meer. In plaats van broadcast adressen worden multicast adressen gebruikt. Het afschaffen van het broadcast adressen is een goede zaak, want broadcasts zorgen op menig netwerk voor onnodig veel overhead. Voor elk broadcast adres die een host voorbij ziet komen, moet deze de processor interrumpen. In 90%, of wellicht meer in grotere netwerken, van de gevallen is deze broadcast niet voor de betreffende host van toepassing. De adres typen welke in IPv6 gebruikt worden zijn: unicast, multicast en anycast adressen.

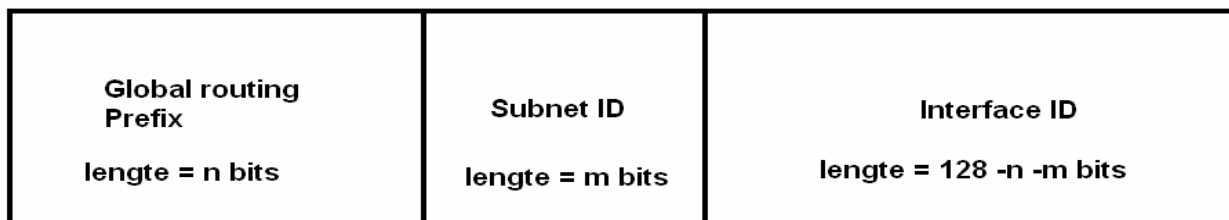
*Unicast* is een adres wat uniek is voor een bepaalde machine binnen een netwerk. Een pakket voor een unicast adres wordt alleen verzonden naar dat adres. Een voorbeeld van een unicast adres in IPv4 is 145.89.82.50 een voorbeeld van een unicast adres in IPv6 is 2001:610:140:2202::3

*Multicast adres*, hiermee bereik je een bepaalde groep machines binnen het netwerk. Een multicast adres is een adres waarmee een groep unicast adressen geïdentificeerd wordt. Een pakket wat wordt verstuurd naar een multicast adres zal bij alle leden van een multicast groep worden afgeleverd.

*Anycast* is een soort kruising tussen unicast en multicast. Een willekeurige verzameling nodes kan worden aangemerkt als een anycast-groep. Alle nodes uit zo'n groep hebben hetzelfde adres. Een IP-pakket met een anycast-adres als bestemming wordt slechts bij één node afgeleverd, meestal is dat de dichtstbijzijnde node uit de anycast-groep met dat adres.

Een ander groot verschil is dat in IPv4 een interface normaal gesproken één adres krijgt toegewezen, in IPv6 heeft een interface eigenlijk altijd meer dan één adres.

In figuur 9 staat het formaat van een standaard IPv6 adres afgebeeld.



Figuur 9

De global routing prefix wordt gebruikt om het type adres (zoals multicast) of de adres reeks te identificeren. Het subnet ID wordt gebruikt om een link binnen een site te identificeren, het begrip link is het zelfde als een subnet zoals dat in IPv4 gebruikt wordt.

Het is toegestaan meerdere subnets ID's op een link te gebruiken. Ten slotte wordt de Interface ID gebruikt om een node te kunnen identificeren. De interface ID moet daarom uniek zijn op een link.

Zoals hierboven beschreven staat, wordt het type adres bepaald door de waarde van de Global routing prefix. In RFC 2373 [4] is een lijst met prefixen en het bijbehorende type adres gedefinieerd.

Allocation -----	Prefix (binary) -----	Prefix (Hex)	Fraction of Address Space -----
Reserved	0000 0000	::0/128	1/256
Unassigned	0000 0001		1/256
Reserved for NSAP Allocation	0000 001		1/128
Reserved for IPX Allocation	0000 010		1/128
Unassigned	0000 011		1/128
Unassigned	0000 1		1/32
Unassigned	0001		1/16
Aggregatable Global Unicast Addresses	001	2::/125	1/8
Unassigned	010		1/8
Unassigned	011		1/8
Unassigned	100		1/8
Unassigned	101		1/8
Unassigned	110		1/8
Unassigned	1110		1/16
Unassigned	1111 0		1/32
Unassigned	1111 10		1/64
Unassigned	1111 110		1/128
Unassigned	1111 1110 0		1/512
Link-Local Unicast Addresses	1111 1110 10	FE80::/10	1/1024
Site-Local Unicast Addresses	1111 1110 11	FEC0::/10	1/1024
Multicast Addresses	1111 1111	FF00::/8	1/256

**Tabel 1**

Zoals blijkt uit tabel 1, is het overgrote deel van de adres ruimte (80%) “Unassigned”, deze zijn voor toekomstig gebruik. Afhankelijk van de begin waarde van een adres kan dus bepaald worden wat voor type adres het is. De belangrijkste type worden in de volgende paragraaf behandeld.

### 6.3 Aggregatable Global Unicast Addresses

Aggregatable Global Unicast Addresses beginnen altijd met een binaire waarde “001”. Deze adressen worden gebruikt om host mee te adresseren op het Internet, vandaar de naam global. Deze adressen zijn dus overal te gebruiken, in tegenstelling tot bijvoorbeeld link local en site local adressen. De adressen zijn “aggregatable”, dwz dat er een bepaalde vorm van hiërchie in moet zitten. Een Aggregatable Global Unicast Addresses is opgebouwd als in figuur 10:



figuur 10

#### Prefix

Altijd 001 voor een Aggregatable Global Unicast Addresses.

#### TLA ID

Top-Level Aggregation Identifier

#### RES

Gereserveerd voor toekomstig gebruik.

#### NLA ID

Next-Level Aggregation Identifier

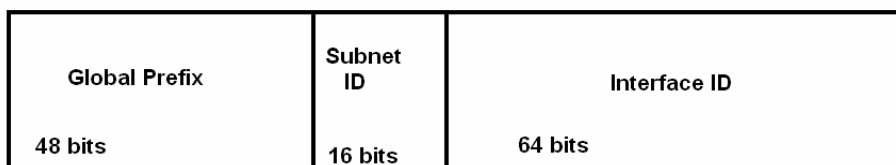
#### SLA ID

Site-Level Aggregation Identifier, deze reeks wordt gebruikt om te subnetten binnen een site.

#### INTERFACE ID

Interface Identifier, dit is een unieke identifier binnen een subnet/link

Het adres formaat zoals hierboven is beschreven, vind je in het overgrote deel van de documentatie over IPv6 terug. Dit formaat wordt echter meer zo gebruikt. Men realiseerde zich dat dit adres formaat niet het juiste is. Dit model limiteerde het aantal “upstream provider-independent ISP’s” tot 8192 ( $2^{13}$ ), hoewel dit een nobel en vooruitstrevend idee was om de groei van de route tabellen aan te pakken, bleek dit getal niet realistisch.



figuur 11

Het huidige formaat ziet er vaak zoals in figuur 11 uit. Waarbij een organisatie een /48 krijgt toegewezen, bijv 2001:610:140::/48 deze kan dan zelf gesubnet worden in 65536 subnetten van 64bits. Echter deze getallen van 48, 16 en 64 bits liggen niet vast.



## Adres toewijzing

De Internet Assigned Numbers Authority (IANA) is verantwoordelijk voor de uitgifte van IPv6 adressen. IANA delegeert wanneer nodig grote stukken adres ruimte aan RIRs, dit zijn Regional Internet Registries. In europa is er RIPE (Réseaux IP Européens), in noord Amerika is er Arin, de APNIC voor Azië en LACNIC voor zuid Amerika.

De LIRs wijzen op hun beurt weer adres ruimte toe aan zogenaamde LIR, Local Internet Registries, LIRs zijn vaak grote ISP's. In IPv4 zijn de toegewezen adres blokken aan LIRs vaak een /16 tot een /20. In IPv6 zijn dit vaak blokken van 32 bits, een zogenaamde "slash tweeëndertig" (/32). De LIRs, vaak ISP's, wijzen blokken adres ruimte toe aan kleine ISP's, organisaties en uiteraard aan de consumenten. Een goed voorbeeld hiervan in Nederland is de Internet Service Provider XS4ALL, bij deze provider kunnen alle klanten een /48 aan IPv6 adres ruimte aanvragen.

De huidige toewijzing door IANA is als volgt:

IPv6 Prefix	Assignment
2000::/16	Reserved
2001::/16	Sub-TLA Assignments [RFC2450]
2002::/16	"6to4" [RFC3056]
3FFE::/16	6bone Testing [RFC2471]
3FFF::/16	Reserved

Zoals te zien is, wordt de reeks 2001::/16 gebruikt voor toewijzingen aan RIRs. De reeks 2002::/16 wordt gebruikt voor de 6to4 adressen, de 3FFE reeks wordt gebruikt voor sixbone adressen. 6to4 adressen, zijn specieale IPv6 adressen welke worden samen gesteld aan de hand van een IPv4 adres, deze beginnen altijd met 2002:: en dan het IPv4 adres. Zo kan iedereen die een IPv4 adres heeft, zich zelf ook een IPv6 adres toewijzen. 3FFE adressen horen bij SIXBONE. Het SIXBONE project heeft in het allereerste begin van IPv6 gedient als test netwerk.

De 2001::/16 wordt zoals gezegd gebruikt voor de Aggregatable Global Unicast Addresses, Deze prefixen zijn als volgt verdeeld:

IPv6 Prefix	Allocated to	Date
2001:0000::/23	IANA	Jul 99
2001:0200::/23	APNIC	Jul 99
2001:0400::/23	ARIN	Jul 99
2001:0600::/23	RIPE NCC	Jul 99
2001:0800::/23	RIPE NCC	May 02
2001:0A00::/23	RIPE NCC	Nov 02
2001:0C00::/23	APNIC	May 02
2001:0E00::/23	APNIC	Jan 03
2001:1200::/23	LACNIC	Nov 02
2001:1400::/23	RIPE NCC	Feb 03

## 6.4 Link-Local Unicast Adressen

De adressen welke een prefix hebben van FE80::/10, zijn link local adressen. Dit zijn speciale unicast adressen welke alleen geldig zijn op een bepaalde link en worden niet gerouteerd. Ze worden onder andere gebruikt voor autoconfiguration en neighbor discovery. Uiteraard zijn deze adressen ook prima te gebruiken op een netwerk zonder routers. Het formaat van een link local adres is afgebeeld in figuur 12:



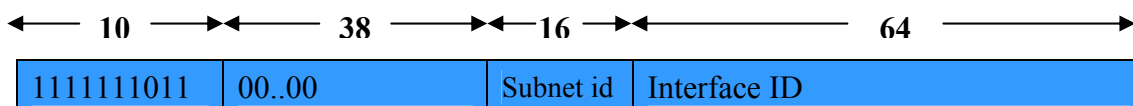
Figuur 12

Het adres begint altijd met FE80 gevolgd door 54 bits aan nullen en vervolgens de interface identifier. Een voorbeeld van een link local adres is: fe80::202:44ff:fe50:ae46  
De interface ID is hier dus 202:44ff:fe50:ae46. Dit is een waarde welke wordt samengesteld aan de hand van het 48 bits mac adres en het EUI-64 algoritme.

In IPv4 worden private adressen uit RFC 1918 gebruikt om private netwerken te kunnen gebruiken. Een voorbeeld hiervan is 192.168.2.4, in IPv6 kunnen hiervoor link local adressen worden gebruikt.

## 6.5 Site-Local Unicast Adressen

Hoewel de private adres reeks uit RFC 1918 niet over internet gerouteerd kan worden, kunnen de adressen wel gebruikt worden om binnen een site verschillende subnetten te creëren en hiertussen kan dan gerouteerd worden. In IPv6 kunnen hiervoor de zogenaamde Site local adressen gebruikt worden. Deze zijn in tegenstelling tot link local adressen wel routeerbaar, echter alleen binnen een site. Deze link local adressen kunnen dus niet gebruikt worden voor communicatie over het Internet. Het formaat van de site local adressen is weergegeven in figuur 13.

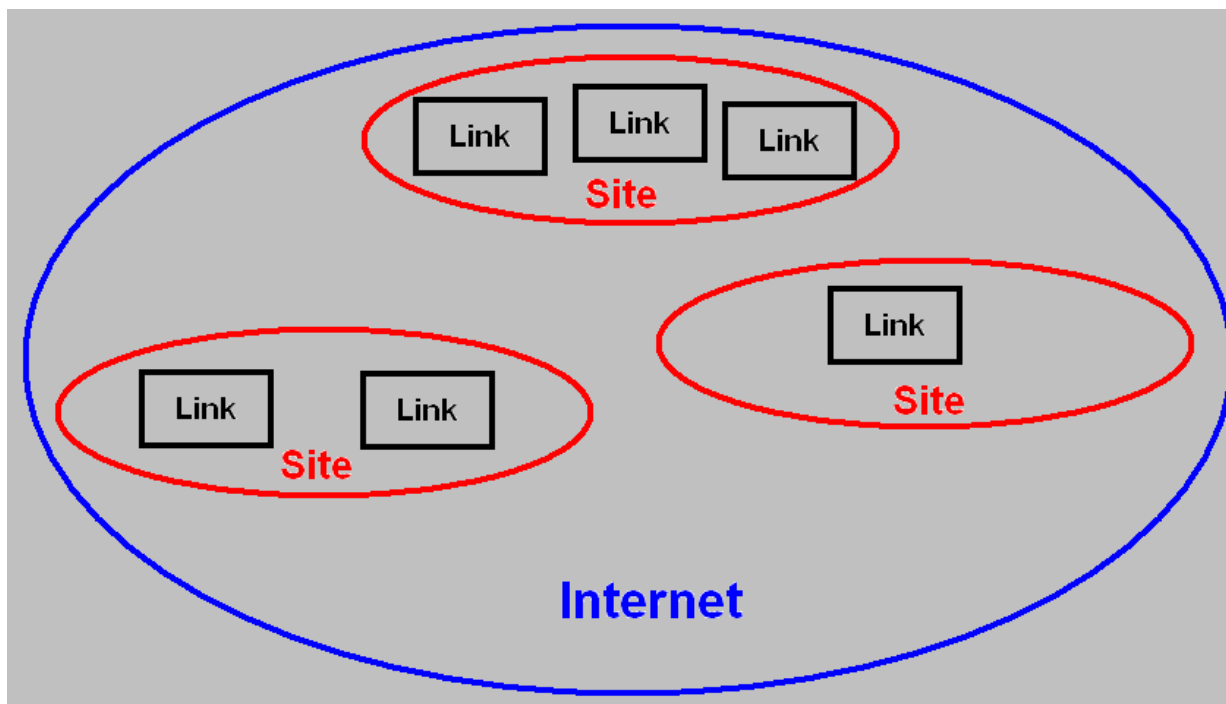


Figuur 13

Een site local adres zal altijd beginnen met FEC0: gevolgd door 38 bits aan nullen. Vervolgens een 16 bits getal welke het subnet ID aangeeft. Ten slotte weer de 64 bits interface ID.

Site local adressen zijn ideaal voor situaties, waarin alle nodes binnen een site met elkaar dienen te communiceren maar absoluut niet met het Internet verbonden mogen worden.

Schematisch ziet het verhaal van link local adressen en Site local adressen eruit zoals is afgebeeld in figuur 14 op de volgende pagina.



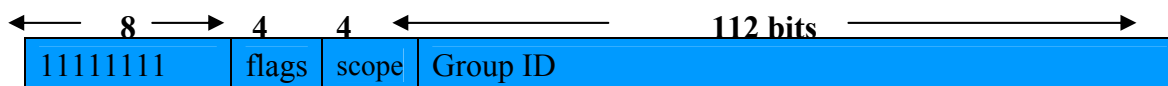
figuur 14

De grote cirkel is het Internet, het Internet is opgebouwd uit een groot aantal netwerken. Deze netwerken zijn als de kleiner cirkels getekend, dit zijn de zogenaamde site's. Een site kan op zijn beurt weer bestaan uit één of meerdere links. Deze links zijn de subnetten binnen een netwerk en hebben over het algemeen een prefixlengte van 64 bits.

## 6.6 Multicast adressen

Met multicast adressen bereik je een bepaalde groep nodes binnen het netwerk. Een multicast adres is een adres waarmee een groep unicast adressen geïdentificeerd wordt. Een pakket wat wordt verstuurd naar een multicast adres zal bij alle leden van een multicast groep worden afgeleverd. Een bepaalde node kan meer dan één multicast adres hebben en dus tot meerdere multicast groepen behoren. Alle taken waarvoor in IPv4 broadcast adressen gebruikt werden, worden nu mbv multicast adressen gedaan. Multicast werd ook al gebruikt in IPv4 (de klasse D adressen) echter voor IPv6 is multicast herzien en verbeterd.

Bekende multicast adressen zijn onder andere het adres FF02::1 wat betekent, alle nodes op de link of FF02::2 waarmee alle routers op de link geadresseerd worden. Het formaat van een multicast adres is te zien in figuur 15:



Figuur 15

Het eerste byte van het adres geeft aan dat dit een multicast adres is. De volgende 4 bits worden gebruikt voor vlaggen, op dit moment wordt er van de vier bits slechts één gebruikt. De eerste drie bits zijn voor toekomstig gebruik, de laatste bit geeft aan of dit een permanent adres is (toegewezen door de IANA, well known multicast adres) of een tijdelijk adres. Als het laatste bit een waarde van nul heeft, is het een well known adres; een waarde van één betekent een tijdelijk adres.

Het scope veld bestaat uit 4 bits, dit geeft de scope van het multicast adres aan. Dit veld geeft aan, hoever het de reikwijdte van het adres is. Hiervoor zijn 16 mogelijkheden (4 bits), afgesproken is hoe hoger de waarde van het scope veld hoe groter de scope (reikwijdte). De volgende waarden, weergegeven in tabel 2 zijn gedefinieerd:

0	Reserved
1	node-local scope/ interface local
2	link-local scope
3,4	Unassigned
5	site-local scope
6,7	Unassigned
8	organization-local scope
9,A,B,C,D	Unassigned
E	global scope
F	Reserved

**Tabel 2**

Het laatste veld geeft de Group ID aan, hiervoor zijn 112 bits beschikbaar. In RFC 2375 [5] zijn een aantal permanente multicast adressen beschreven, een aantal daarvan zijn:

### Node-Local Scope

FF01:0:0:0:0:0:0:1 All Nodes Address  
FF01:0:0:0:0:0:0:2 All Routers Address

### Link-Local Scope

FF02:0:0:0:0:0:0:1 All Nodes Address  
FF02:0:0:0:0:0:0:2 All Routers Address  
FF02:0:0:0:0:0:0:3 Unassigned  
FF02:0:0:0:0:0:0:4 DVMRP Routers  
FF02:0:0:0:0:0:0:5 OSPFIGP  
FF02:0:0:0:0:0:0:6 OSPFIGP Designated Routers  
FF02:0:0:0:0:0:0:7 ST Routers  
FF02:0:0:0:0:0:0:8 ST Hosts  
FF02:0:0:0:0:0:0:9 RIP Routers  
FF02:0:0:0:0:0:0:A EIGRP Routers  
FF02:0:0:0:0:0:0:B Mobile-Agents  
FF02:0:0:0:0:0:0:D All PIM Routers  
FF02:0:0:0:0:0:0:E RSVP-ENCAPSULATION  
FF02:0:0:0:0:0:1:1 Link Name  
FF02:0:0:0:0:0:1:2 All-dhcp-agents  
FF02:0:0:0:0:1:FFXX:XXXX Solicited-Node Address

### Site-Local Scope

FF05:0:0:0:0:0:0:2 All Routers Address  
FF05:0:0:0:0:0:1:3 All-dhcp-servers  
FF05:0:0:0:0:0:1:4 All-dhcp-relays  
FF05:0:0:0:0:0:1:1000 Service Location

Multicast adressen kunnen alleen als destination adres gebruikt worden en nooit als source adres.

### 6.6.1 Solicited-node multicast adres

Het solicited-node adres is een speciaal multicast adres welke iedere host moet hebben. Dit adres wordt gebruikt bij het duplicate address detection process (DAD) en neighbor discovery. Het solicited-node multicast adres heeft altijd de prefix FF02:0:0:0:0:1:FF00::/104 deze prefix wordt door een node aangevuld met de laatste 24 bits van zijn IPv6 adres. Een voorbeeld van zo'n adres is FF02:0:0:0:0:1:FF1e:D213.

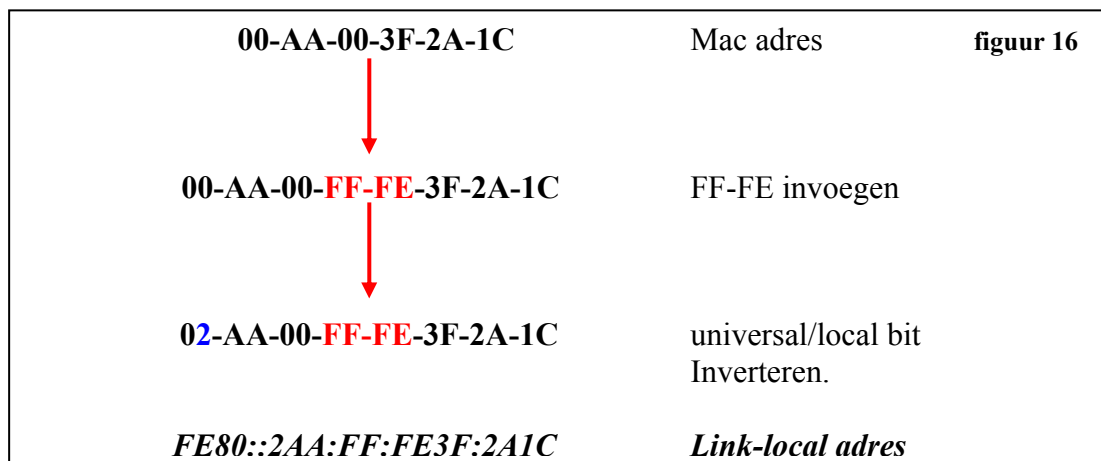
### 6.6.2 Multicast group management

Om bij te houden welke nodes bij welke multicast groepen horen, is het Internet Group Management Protocol (IGMP) ontwikkeld. IGMP versie 1 en 2 worden gebruikt in IPv4. Dit protocol wordt gebruikt om bij te houden welke unicast adressen bij welk multicast adres horen. Tevens wordt dit protocol gebruikt voor de zogenaamde joins en leaves (lid worden en afmelden bij een multicast groep). Het multicast listener discovery protocol, zoals het protocol in IPv6 wordt genoemd, is gebaseerd op IGMPv2 maar maakt nu gebruik van ICMP berichten (ICMP type 130 t/m 132).

## 6.7 Interface ID

In de voorgaande paragrafen zijn reeds een aantal adres typen besproken, deze worden steeds gekenmerkt door hun prefix. Deze prefixen (link-local, site-local en Aggregatable Global Unicast) worden steeds aangevuld met een interface ID. Deze interface ID wordt automatisch samen gesteld aan de hand van het mac adres van de node en de prefix, volgens de EUI64 standaard.

Het mac adres bestaat uit 48 bits, de interface identifier is een 64bits getal, dit betekent dat het mac adres moet worden aangevuld met 16bits. In het volgende voorbeeld, heeft host A een ethernet MAC adres: "00-AA-00-3F-2A-1C". Om dit adres te converteren naar een EUI-64 adres, wordt er altijd het 16 bits getal FF-FE tussen het derde en vierde byte ingevoegd. Dit heeft het volgende resultaat 00-AA-00-FF-FE-3F-2A-1C, vervolgens wordt het universal/local bit, dit is het 7<sup>e</sup> bit van de eerste byte geïnverteerd. Het eerste byte in binaire vorm ziet er als volgt uit 00000000, nadat we het 7<sup>e</sup> bit hebben geïnverteerd krijgt dit byte de waarde 00000010 (0x02). Het uiteindelijk resultaat is dan 02-AA-00-FF-FE-3F-2A-1C. Het bijbehorende link-local adres van deze interface zou dan worden: "FE80::2AA:FF:FE3F:2A1C" Schematisch ziet dit eruit als in figuur 16.



## 6.8 Verplichte adressen

Er zijn een aantal adressen waar naar elke host dient te luisteren:

1. Voor iedere interface een link-local adres.
2. een loopback adres [::1]
3. “all nodes multicast address” [FF02::1]
4. Het solicited-node adres [FF02:0:0:0:1:FFxx:xxxx]

In het geval van een router, dient deze nog een tweetal extra adressen hebben:

5. het subnet router anycast adres. (wordt in de praktijk niet gebruikt).
6. “all routers multicast address” [FF02::2]

Hieronder een voorbeeld van alle adressen van een cisco router:

```
Rtr1#sh ipv6 interface FastEthernet0/1
FastEthernet0/1 is up, line protocol is up
IPv6 is enabled, link-local address is FE80::205:DCFF:FE66:1006
Description: Service LAN1
Global unicast address(es):
  2001:610:140:A604::1, subnet is 2001:610:140:A604::/64
  2001:7B8:200:A604::1, subnet is 2001:7B8:200:A604::/64
Joined group address(es):
  FF02::1 (all nodes multicast address)
  FF02::2 (all routers multicast address)
  FF02::1:FF00:1 (solicited-node multicast address)
  FF02::1:FF66:1006 (solicited-node multicast address)
MTU is 1500 bytes
ICMP error messages limited to one every 100 milliseconds
ICMP redirects are enabled
Input features: ACL
Output features: ACL
Inbound access list V6-FROM-SERVICE-LAN
Outgoing access list V6-TO-SERVICE-LAN
ND DAD is enabled, number of DAD attempts: 1
ND reachable time is 30000 milliseconds
ND advertised reachable time is 0 milliseconds
ND advertised retransmit interval is 0 milliseconds
ND router advertisements are sent every 200 seconds
ND router advertisements live for 1800 seconds
Hosts use stateless autoconfig for addresses.
```

## 7 ICMP v6

Iedereen die een beetje bekend is met IP, weet dat ICMP een grote hulp kan zijn bij het troubleshooten van een IP netwerk. ICMP genereert berichten welke informatie verschaffen over pakketten die te groot zijn, of wanneer een host/netwerk niet te bereiken is. In de IPv6 wereld is ICMP nog krachtiger geworden, het heeft namelijk een aantal extra functies gekregen.

Zo is het nu verantwoordelijk voor het zoeken van een datalink adres (MAC adres) bij een IPv6 adres en andersom. Deze functie werd in de IPv4 situatie uitgevoerd door het (R)ARP protocol. ICMPv6 heeft deze functie overgenomen en uitgebreid. Het ICMP proces wat ARP vervangt wordt in IPv6 Neighbor Discovery (ND) genoemd. Een tweede belangrijke uitbreiding is de functie van het multicast listener discovery protocol. Dit ICMP proces is de vervanger van IGMP zoals we dat in IPv4 kennen. Een laatste belangrijke nieuwe feature in ICMPv6, is de ondersteuning voor Mobile IP. Er zijn verschillende type ICMP messages, welke het proces van mobile IP mogelijk maken. Meer informatie over mobile IP is te lezen in het hoofdstuk Mobile IP (hoofdstuk 13.3).

ICMP heeft een ware metamorfose ondergaan, het is een zeer krachtig zoniet een onmisbaar protocol geworden. Overhead van andere protocollen zoals ARP en IGMP zijn hiermee komen te vervallen. Dit heeft als gevolg dat het blokkeren van ICMP verkeer in een firewall of router wel eens erg vervelend kan uitpakken voor de netwerk gebruikers. In IPv4 netwerken was het voor een hoop netwerk beheerders niet ongewoon om ICMP verkeer te blokken. Hoewel dit hooguit een vervelend probleem was, was het geen onoverkomelijk probleem. Wanneer dit zelfde in IPv6 netwerken gedaan zal worden, zal de netwerkbeheerder voor een aantal onaangename verrassingen komen te staan. Niet alleen is het blokkeren van ICMPv6 vervelend voor wat betreft het troubleshooten van netwerk problemen, maar het belemmert de juiste werking van IPv6 ook. Dit komt omdat ICMPv6 een aantal belangrijke functies vervult. Het proces Path MTU discovery, is hiervan een mooi voorbeeld. pMTU discovery is verantwoordelijk voor het bepalen van de maximale MTU grote op de weg naar het eindpunt. Wanneer dit proces niet juist kan worden uitgevoerd als gevolg van het blokkeren van 'ICMP packet to big berichten' betekent dit dat communicatie nooit zal plaatsvinden. Ook het gebruik van multicast, wat een veel gebruikt wordt in IPv6, kan dan niet langer succesvol verlopen.

### ***Neighbor Discovery***

Zoals gezegd heeft ICMP naast zijn oude functies een aantal nieuwe functies gekregen. Één ervan is het neighbor discovery proces welke wordt beschreven in RFC 2461 [6].

De functies van het Neighbor Discovery proces zijn de volgende:

1. Bepalen van link-layer adressen, van andere nodes op dezelfde link.
2. Het zoeken van routers op een link
3. Het bijhouden van de bereikbaarheidsinformatie van nodes op dezelfde link.

Om deze functies te kunnen vervullen zijn er verschillende typen ICMP pakketten ontworpen. In het hierop volgende gedeelte zullen de deze functies worden beschreven. Dit wordt gedaan door het proces van autoconfiguratie beschrijven, hierin zitten alle aspecten van neighbor/router discovery verwerkt. Door middel van Autoconfiguratie kunnen hosts op een netwerk zich zelf (stateless) voorzien van een IPv6 adres, zonder zelf vooraf kennis te hebben van het netwerk

## Neighbor Solicitation en Neighbor advertisements.

Wanneer een IPv6 enabled host opstart, zal deze als eerste kijken of de hosts zijn link local adres kan maken met behulp van zijn mac address deze is alsvolgt: fe80::E1U-64-ID een voorbeeld hiervan is het adres fe80::202:44ff:fe23:453c. De betreffende host moet wel zeker weten dat het link local adres uniek op de link is. IPv6 heeft hiervoor een mechanisme, welke controleert of het adres al bestaat op de link, dit proces heet het duplicate address detection (DAD)proces.

De host zal een neighbor solicitation message verturen (ICMP 135) welke naar het solicited-node multicast adres (FF02:0:0:0:1:FF00::/104 + laatste 24 bits van ip adres) wordt verstuurd welke bij het link-localadres hoort.

In dit geval is het solicited-node multicast adres FF02:0:0:0:1:FF23:453c. Het source adres is :: oftewel allemaal nullen, want het adres is nog niet bekend. In het ICMP bericht wordt het tentative address als optie meegegeven. Dit is het target adres, het adres waarvan de duplicatie wordt getest. Een tentative adres wordt in de betreffende RFC alsvolgt beschreven:

```
tentative address - an address whose uniqueness on a link is being
    verified, prior to its assignment to an interface. A tentative
    address is not considered assigned to an interface in the usual
    sense. An interface discards received packets addressed to a
    tentative address, but accepts Neighbor Discovery packets
    related to Duplicate Address Detection for the tentative
    address.
```

Normaal gesproken vervult de het neighbor discovery proces, de taak van ARP met behulp van neighbor solicitation en neighbor advertisement berichten, in dit geval wordt het ook gebruikt voor DAD.

Het neighbor solicitation pakket ziet er alsvolgt uit:

```
Ethernet II, Src: 00:02:44:23:45:3c, Dst: 33:33:ff:23:45:3c
  Destination: 33:33:ff:23:45:3c (IPv6-Neighbor-Discovery_ff:23:45:3c)
  Source: 00:02:44:23:45:3c (Surecom__23:45:3c)
  Type: IPv6 (0x86dd)
Internet Protocol Version 6
  Version: 6
  Traffic class: 0x00
  Flowlabel: 0x00000
  Payload length: 24
  Next header: ICMPv6 (0x3a)
  Hop limit: 255
  Source address: :: (::)
  Destination address: ff02::1:ff23:453c (ff02::1:ff23:453c)
Internet Control Message Protocol v6
  Type: 135 (Neighbor solicitation)
  Code: 0
  Checksum: 0xac66 (correct)
  Target: fe80::202:44ff:fe23:453c (fe80::202:44ff:fe23:453c)
```

Als er binnen een bepaalde time-out niemand reageert op het neighbor discovery packet, wordt het adres als uniek beschouwd en wordt het link local adres toegewezen aan de interface.

Wanneer een andere host op het netwerk dit link local adres al gebruikt, dan zal deze host op het neighbor solicitation verzoek reageren met een neighbor advertisement met daarin zijn link local adres, in dit geval stopt de host met de autoconfiguratie.



Wanneer het link local adres is toegewezen aan de interface, zal de host verder gaan met de volgende stap in de autoconfiguratie. Deze volgende stap is het op zoek gaan naar een router in het netwerk. De host verstuurd een router solicitation message met als doel een router te vinden op het netwerk welke de host kan helpen met de autoconfiguratie.

Normaal gesproken maakt een router zich met bepaalde intervallen bekend door middel van router advertenties. Een host kan hier ook om vragen, zodat daar niet op gewacht hoeft te worden. Het verzoek wordt gedaan door het verzenden van router solicitation berichten. Als source adres wordt het linklocal adres van de host gebruikt, het destination adres is het multicast adres ff02::2 (alle routers op de link het bijbehorende ethernet multicast-adres is 33-33-00-00-00-02). De opbouw van het pakket is als volgt:

```
Ethernet II, Src: 00:02:44:23:45:3c, Dst: 33:33:00:00:00:02
  Destination: 33:33:00:00:00:02 (IPv6-Neighbor-Discovery_00:00:00:02)
  Source: 00:02:44:23:45:3c (Surecom__23:45:3c)
  Type: IPv6 (0x86dd)
Internet Protocol Version 6
  Version: 6
  Traffic class: 0x00
  Flowlabel: 0x00000
  Payload length: 16
  Next header: ICMPv6 (0x3a)
  Hop limit: 255
  Source address: fe80::202:44ff:fe23:453c (fe80::202:44ff:fe23:453c)
  Destination address: ff02::2 (ff02::2)
Internet Control Message Protocol v6
  Type: 133 (Router solicitation)
  Code: 0
  Checksum: 0x686b (correct)
  ICMPv6 options
    Type: 1 (Source link-layer address)
    Length: 8 bytes (1)
    Link-layer address: 00:02:44:23:45:3c
```

Met een router solicitation verzoek, wordt gevraagd aan alle routers om zich bekend te maken en te vertellen welke prefixen gebruikt kunnen worden voor autoconfiguratie. Als er een router op de link aanwezig is, zal deze onmiddellijk reageren met een router advertisement. Het destination adres is het multicast adres ff02::1 (alle nodes) dit is zo bij een periodieke router advertisement, of het adres van de node welke zojuist een router solicitation bericht heeft verstuurd. Een voorbeeld van een router advertisement pakket is te zien op de volgende pagina.

```
Internet Control Message Protocol v6
  Type: 134 (Router advertisement)
  Code: 0
  Checksum: 0x8510 (correct)
  Cur hop limit: 64
  Flags: 0x00
    0... .... = Not managed
    .0.. .... = Not other
    ..0. .... = Not Home Agent
    ...0 0... = Router preference: Medium
  Router lifetime: 30
  Reachable time: 0
  Retrans time: 0
  ICMPv6 options
    Type: 3 (Prefix information)
    Length: 32 bytes (4)
    Prefix length: 64
    Flags: 0xe0
      1... .... = Onlink
      .1.. .... = Auto
      ..1. .... = Router Address
      ...0 .... = Not site prefix
    Valid lifetime: 0x00278d00
    Preferred lifetime: 0x00093a80
    Prefix: 2001:888:1357::
  ICMPv6 options
    Type: 1 (Source link-layer address)
    Length: 8 bytes (1)
    Link-layer address: 00:a0:c9:66:ec:7c
```

In een router advertisement bericht (ICMP type 134) worden verschillende opties meegegeven. Een voorbeeld hiervan is de hoplimit. Daarna volgen er een aantal 1bit opties (vlaggen). De eerste vlag is de “M flag” als deze een waarde 1 heeft betekend dit, dat de host stateful configuratie moet gebruiken. Met stateful configuratie wordt bedoeld, DHCP zoals in IPv4. Er is ook een IPv6 versie van DHCP in ontwikkeling. De volgende vlag is de “O flag” (other), als deze vlag gezet is betekend dit, dat DHCP niet alleen gebruikt wordt om een IPadres te leren maar ook bijvoorbeeld het adres van een nameserver. De volgende 6 bits worden niet gebruikt, en dienen 0 te zijn. Het veld “router lifetime” geeft de geldigheid van de router als default router aan.

Er zijn op dit moment 3 verschillende type optie velden beschikbaar.

1. source link-layer address information
2. target link-layer address information
3. Prefix information
4. redirected header
5. mtu size welke gebruikt dient te worden op netwerken met een variabele mtu size ( zoals tokenring).

Het derde punt is erg belangrijk voor autoconfiguratie, alle netwerken die de betreffende router interface kent, worden geadverteerd. Een host welke zicht wil autoconfigureren zal adressen creëren voor ieder van de geadverteerde prefixen. Het optie veld van de prefix informatie vinden bevat het volgende:

Prefix lenght, in dit geval is dat 64 bits gevolgd door een aantal vlaggen.

Alles bij elkaar ziet een pakket er als volgt uit:

- Onlink, als dit bit gezet is, kan het voor onlink determination worden gebruikt.
- Auto (autonomous address-configuration flag), Als dit bit gezet is, betekend dit, dat het adres gebruikt mag worden als global adres, de prefix wordt dan aangevuld, met interface indetifier.

- Valid Lifetime, de tijd dat het adres geldig is.
- Preferred Lifetime, de tijd dat het adres als preferred adres wordt gebruikt
- Prefix, de prefix.

Een router advertisement, met prefix informatie, kan meerder prefixen bevatten, dit is voor multihomed host erg handig. De host zal dan adressen creëren voor iedere geadverteerde prefix.

Nadat de host de prefix informatie heeft ontvangen van de router, kan deze het adres gaan gebruiken. Maar voor dat de host het adres echt gaat gebruiken, moet deze checken of het adres nog niet gebruikt wordt door een andere host op het netwerk. Net zoals dit gebeurde bij link local adressen wordt ook voor global adressen gecontroleerd of het adres uniek is op de link (DAD).

Ook nu weer wordt een neighbor solicitation pakket gestuurd naar het solicited-node multicast address, die bij het adres hoort welke de host wil gebruiken. Als er een host antwoord betekend dit dat het adres al bezet is, als er binnen een bepaalde tijd niet wordt geantwoord, gaat de host er van uit, dat het adres nog niet in gebruik is en zal hij zich zelf het adres toewijzen. Nu ook dit globale adres uniek is, kan de host communiceren met andere hosts op het netwerk

## 8 Het AMS-IX netwerk

Een gedeelte van de afstudeeropdracht is, om de services van de Amsterdam Internet Exchange beschikbaar te maken voor IPv6. Zodat IPv6 hosts van zowel Internet als het interne netwerk gebruik kunnen maken van de AMS-IX services. Services die via IPv6 benaderbaar dienen te zijn, zijn om te beginnen uiteraard de website ([www.ams-ix.net](http://www.ams-ix.net)) en de mailserver, om te zorgen dat deze services over IPv6 benaderd worden, is het noodzakelijk dat de DNS server ook queries over IPv6 kan ontvangen en beantwoorden. Dit is slechts een gedeelte van de services in het netwerk van de Amsterdam Internet Exchange, voor de algehele invoering van IPv6 in het netwerk van de AMS-IX is een stappenplan gemaakt, zie hiervoor bijlage I.

Om een beter inzicht te krijgen in hoe het netwerk van de Amsterdam Internet Exchange is opgebouwd zal eerst een beschrijving van de huidige infrastructuur worden gegeven.

### 8.1 AMS-IX netwerk

Het netwerk van de Amsterdam Internet Exchange is opgebouwd uit verschillende onderdelen. Dit zijn onder andere onderdelen voor dienstverlening aan derde partijen, de zgn. exchange netwerken en een onderdeel voor ondersteunende diensten, zoals de webserver en het kantoor. Deze onderdelen zijn gedefinieerd als verschillende (V)LANs.

Het complete netwerk kan om te beginnen worden onderverdeeld in 2 delen:

1. Netwerken die bedoeld zijn voor klanten /leden van de AMS-IX, om verkeer uit te wisselen. Dit zijn de exchange netwerken op het switch platform.
2. Netwerken die bedoeld zijn voor de eigen en ondersteunende diensten van de Amsterdam Internet Exchange.

Gebruikers van het switch platform van de Amsterdam Internet Exchange, zijn leden. Deze leden hebben allemaal een verschillende achtergrond, op het ISP lan, zijn het voornamelijk Internet Providers, die daar IP verkeer uitwisselen.

Het GRX (GPRS Roaming Exchange) lan, wordt gebruikt door carriers om GPRS data uit te wisselen (GPRS roaming). Er is ook een vlan beschikbaar speciaal voor mobiele operators, die hier MMS (Multimedia Messaging Services) berichten met uitwisselen.

MMS is de opvolger van SMS, in plaats van alleen tekst kun je met MMS ook beeldberichten vanaf je mobiele telefoon versturen. In eerste instantie zijn dat foto's, over een half jaar volgen videofilmmpjes met geluid.

In dit hoofdstuk zal uitéén gezet worden, wat de functies van de verschillende netwerken zijn, hoe ze verband houden met elkaar en wat hun plaats in het netwerk is. Daarnaast zal worden ingegaan op, hoe de adressering en routing voor IPv4 in de netwerken is gerealiseerd.

## 8.2 Netwerken op het switch platform

De Amsterdam Internet Exchange is een plaats waar Internet providers (ISP's) verkeer met elkaar kunnen uitwisselen. Door onderlinge afspraken kunnen aangesloten providers efficiënt en tegen relatief lage kosten data met elkaar uitwisselen. Maar het zijn niet alleen ISP's die op de AMS-IX data met elkaar uitwisselen, ook mobiele operators zijn aangesloten om de AMS-IX, zij wisselen hier MMS en GPRS verkeer met elkaar uit.

Met een aansluiting op de AMS-IX, wordt bedoeld een aansluiting op een van de vier switches van de Amsterdam Internet Exchange. Deze switches zijn onderling verbonden door middel van "trunks" met een capaciteit van 10Gb/s

Het switch platform wordt gebruikt om verschillende soorten verkeer met elkaar uit te wisselen. Zo zijn er verschillende Internet providers die hier grote hoeveelheden Internet verkeer (IP) met elkaar uit wisselen. De mobiele telecom-providers die hier MMS verkeer uit wisselen en mobiele carriers die GPRS verkeer uit wisselen. Voor de verschillende soorten verkeer zijn op de switches verschillende VLANs gedefinieerd.

De volgende productie VLANs zijn op de switch aanwezig:

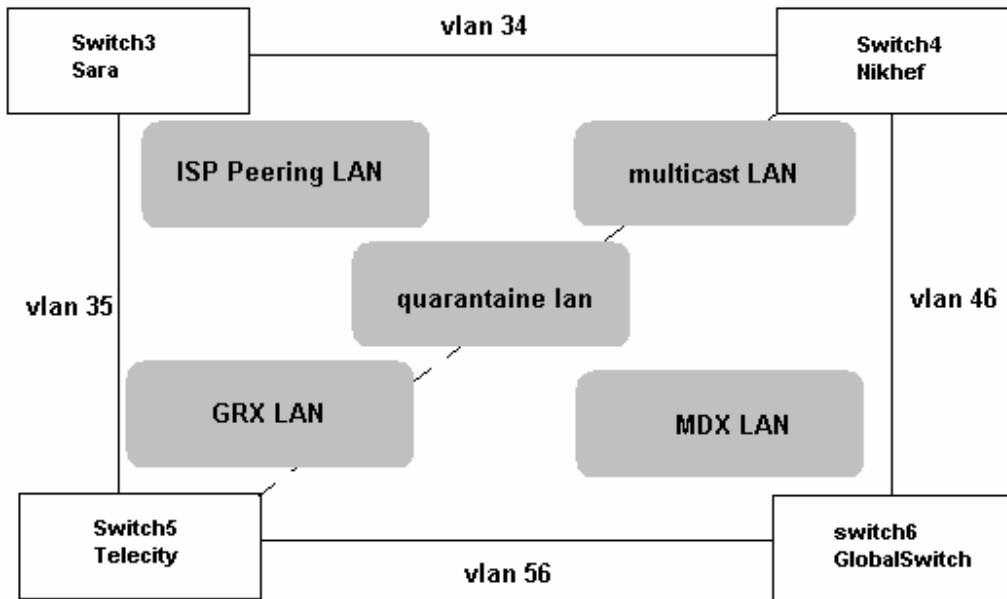
1. ISP peering LAN
2. GRX LAN
3. MDX LAN
4. Multicast LAN

Er zijn ook nog andere VLANs op de switches gedefinieerd, dit zijn echter geen productie netwerken. Deze VLANs hebben voornamelijk een beheers ondersteunende functie, zo zijn er vlands die zijn gecreëerd om de trunks tussen de switches te testen en is er een quarantaine VLAN.

De volgende niet productie VLANs zijn op de switch aanwezig:

1. quarantaine VLAN
2. vlan 34
3. vlan 35
4. vlan 46
5. vlan 56

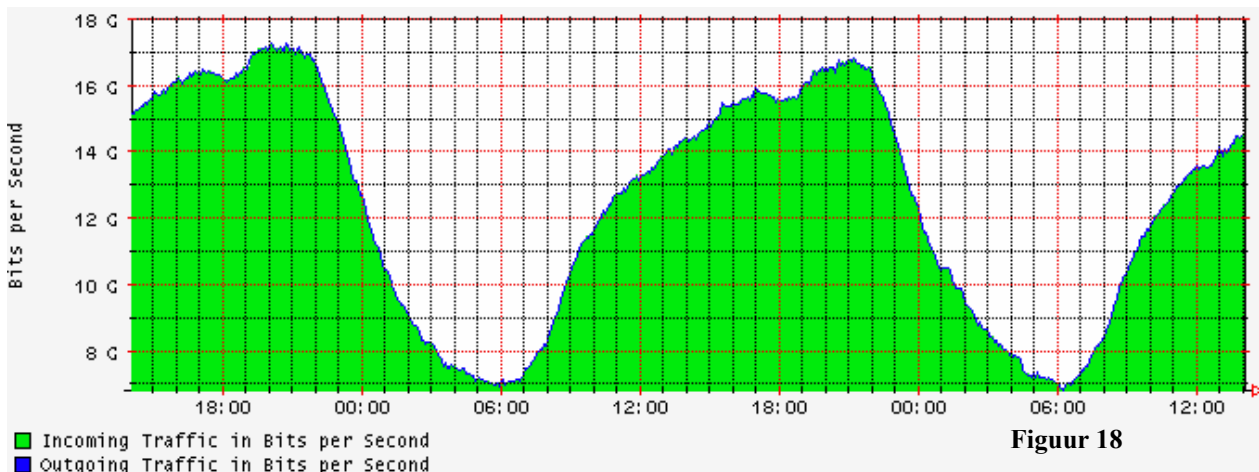
Schematisch ziet dit er als volgt uit, zie figuur 17:



8.2.1

### ISP Peering LAN

In dit netwerk zitten de ISP's die verkeer met elkaar uitwisselen. Dit VLAN heeft als VLAN-ID 501. Wanneer een ISP een aansluiting wil op de AMS-IX dan krijgt deze een switchport toegewezen welke lid is van het ISP peering VLAN. Dit VLAN is het belangrijkste VLAN op het switching platform, het totale verkeer wat op dit moment (mei 2003) over deze exchange gaat is ongeveer 17Gb/s (Dit is 17 Gbit/s in en 17 Gbit/s uit) zie figuur 18. Een groot deel van het nationale IP verkeer van Nederland wordt over dit VLAN getransporteerd.



Figuur 18

## 8.2.2 Multicast LAN

Er is op het switching platform een apart VLAN gedefinieerd welke speciaal bedoeld is voor instellingen/bedrijven die willen experimenteren met multicast verkeer.

Multicast biedt de mogelijkheid om netwerkpakketten naar meer dan één bestemming tegelijk te sturen. Ethernet gebruikers kennen het begrip *broadcast*: de mogelijkheid om een pakket naar alle machines op het (lokale) net te sturen. Op het wereldwijde Internet bestaat geen broadcast mogelijkheid, als de mogelijkheid tot broadcast op het wereldwijde Internet wel bestond werd het een chaos. Want met de miljoenen hosts die het Internet tegenwoordig kent zou het sturen van een enkel pakket naar al die hosts een enorme belasting voor het netwerk veroorzaken. Multicast is een selectievere variant van broadcast: een pakket wordt alleen naar een geselecteerde groep bestemmingen verstuurd. Dit Vlan wordt nog niet echt actief gebruikt, dit is dan ook terug te zien in de totale hoeveelheid verkeer, wat gemiddeld zo'n 10kb/s is met af en toe een piek van een paar Mb/s.

## 8.2.3 GRX LAN

Dit vlan is een "exchange point" voor carriers, hier wisselen de verschillende carriers GPRS data met elkaar uit. De verschillende mobiele carriers hebben een aansluiting op de AMS-IX switch, deze switch poorten zijn ingedeeld in het grx vlan. Zo kunnen de verschillende carriers eenvoudig en veilig GPRS data uitwisselen.

Gemiddeld gaat er ongeveer 500kb/s aan GPRS data over dit vlan. Speciaal hiervoor is een nieuw top level domain opgericht, namelijk .gprs

De root servers voor het .gprs domein, bevinden zich ook in dit vlan. Het beheer van deze servers is de verantwoordelijkheid van AMS-IX.

## 8.2.4 MDX LAN

Het MDX netwerk wordt door mobiele telecom aanbieders gebruikt om MMS verkeer over uit te wisselen. Dit netwerk staat helemaal los van het Internet. Wanneer een KPN abonnee een MMS bericht verstuurd naar een bijvoorbeeld een O2 abonnee, dan zal dat via het MDX vlan gebeuren.

Dit netwerk is ongeveer hetzelfde als het GRX lan, echter het MDX netwerk is voor MMS verkeer en het grx lan voor GPRS verkeer.

## 8.2.5 Quarantaine LAN

Wanneer een bedrijf een aansluiting wil op het ISP/peering netwerk, dan wordt zijn switchport in eerste instantie in een apart VLAN geplaatst, het quarantaine VLAN.

Dit VLAN is geen productie netwerk en wordt gebruikt om het verkeer wat de router verstuurt te analyseren. Zo kan worden bekeken of de router geen "verboden verkeer" verstuurd. Verboden verkeer is onder andere non-ip traffic en onnodig broadcast verkeer. Het enige broadcast verkeer wat nodig en toegestaan is op het ISP-peering LAN zijn ARP requests.

Ook wanneer een bepaalde klant verdacht wordt, van het veroorzaken van problemen op de exchange, wordt deze in dit VLAN geplaatst, zodat er rustig kan worden geanalyseerd wat er precies op deze switch poort gebeurt.

## 8.2.6 Trunk vlans

Dit zijn de vlans tussen de 4 switches. Iedere trunk link heeft een apart vlan, om zodoende te kunnen testen of de trunk links nog goed werken. Met bepaalde intervallen wordt de bereikbaarheid van ipadressen binnen de test vlans getest, als dit niet meer goed gaat, betekend dit waarschijnlijk dat er een trunk is uitgevallen.

De nummering van de vlans is als volgt:  
VLAN 34 voor de trunk tussen switch 3 en 4  
VLAN 35 voor de trunk tussen switch 3 en 5  
VLAN 46 voor de trunk tussen switch 4 en 6  
VLAN 56 voor de trunk tussen switch 5 en 6

Deze Vlans wordt alleen voor beheers doeleinde gebruikt.



## 8.3 AMS-IX netwerken

Het tweede gedeelte van de infrastructuur van AMS-IX bestaat uit netwerken die een “ondersteunende” functie hebben, deze netwerken worden dus niet gebruikt voor exchange doeleinden en zijn dus niet op het switching platform gedefinieerd.

Dit zijn de volgende netwerken:

1. Office netwerk
2. Management netwerk
3. Test netwerk
4. Service netwerk.

Elk van deze netwerken heeft een eigen specifieke functie, welke in de volgende paragrafen zullen worden toegelicht.

### 8.3.1 Office LAN

Dit is het kantoor netwerk aan het westeinde. Op het kantoor bevindt zich onder andere het NOC, waarvandaan het beheer van de switches wordt gedaan. Het kantoor netwerk is via een glasvezel verbinding verbonden met een router op de globalswitch locatie, mocht deze verbinding, om wat voor reden dan ook, onbruikbaar worden dan is er een back-up verbinding. Deze verbinding is een SDSL verbinding naar SARA. Het Office netwerk wordt door middel van accesslists gedeeltelijk afgeschermd van het Internet, zodat niet iedereen zomaar het kantoor netwerk kan bereiken. De belangrijkste hosts in dit netwerk zijn idifix (de router / firewall /vpn server) en panoramix. Panoramix is de intranet server, hierop draaien ook de pop en imap services.

### 8.3.2 Service LAN

De mailservers, webservers en DNS servers bevinden zich in een het service LAN. Dit netwerk is speciaal ontworpen, voor servers die rechtstreeks vanaf het Internet bereikbaar moeten zijn, dit wordt ook wel de DMZ (Demilitarized zone) genoemd. Op de router die dit netwerk met het Internet verbindt, zijn regels geïmplementeerd die ervoor zorgen dat alleen die services bereikbaar zijn die nodig zijn. Dus alleen de poorten voor smtp, DNS en web staan open.

### 8.3.3 Management LAN

De servers uit het service lan, zoals de webserver en de mailserver. Hebben meerdere interfaces. Eén van deze interfaces bevindt zich in het management LAN, zodat deze ook via ssh en andere services bereikbaar zijn. Ook de switches hebben een interface op dit vlan zitten, zodat deze ook bereikbaar zijn via ssh en telnet. Het management LAN is alleen bereikbaar vanaf het kantoornetwerk. Het is dus een afgeschermd netwerk, dit zorgt ervoor dat niet iedereen toegang heeft tot de switches. Voor elke switch is een pc met Linux als OS ingericht, deze Linux pc wordt gebruikt om metingen uit te voeren op iedere switch. Deze Linux pc's, hebben adressen die thuis horen in het management lan.

Tevens is er voor iedere Switch een console server beschikbaar. Mocht er iets met de bereikbaarheid van de switches gebeuren, dan is iedere switch door middel van deze (out of band) consoleservers altijd nog bereikbaar.

### 8.3.4 Test LAN

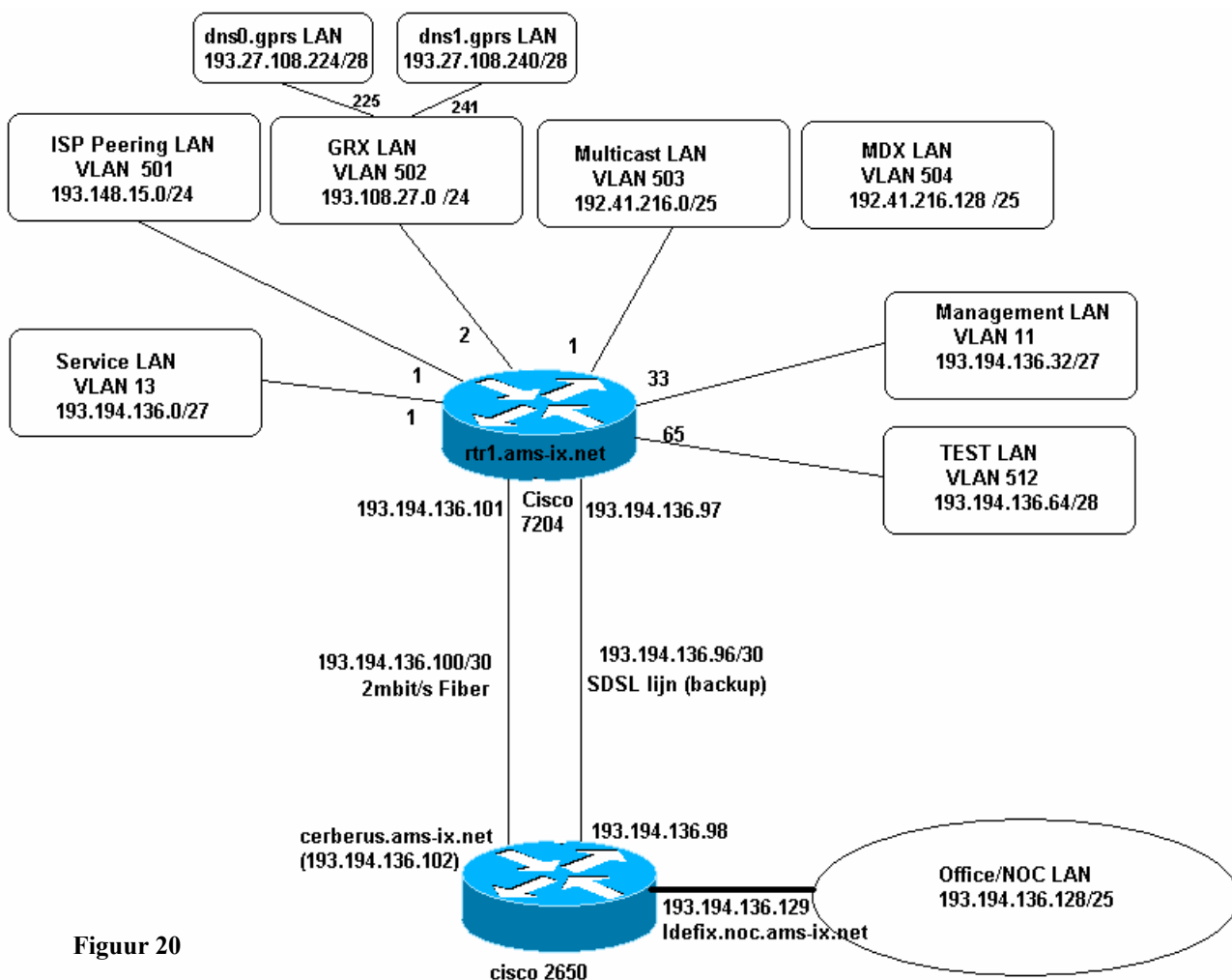
In test netwerk worden servers geplaatst die niet per definitie een belangrijke functie voor ASM-IX hebben. Er staan verschillende servers in dit netwerk.

fulliautomatix.noc.ams-ix.net (193.194.136.66) is onder andere de back-up mailserver en DNS. Deze machine is ook bekend als sirius1.galact-ix.net, ir-baboon.monkey-mind.net en ns1.nlnog.net. Op deze machine worden ook nog enkele “prive websites” gehost. In dit vlan stond ook de tijdelijke IPv6 tunnel server.

## 8.4 Het AMS-IX netwerk in IPv4

Het netwerk is onder te verdelen in verschillende vlans, maar hoe houden deze vlans nu verband? Wat is de adressering die gebruikt wordt voor de verschillende netwerken?

Al dit soort zaken zijn eenvoudig terug te vinden in een layer 3 ontwerp van het netwerk. In figuur 20 staat het layer 3 schema van het AMS-IX netwerk.



Figuur 20

In het figuur 20, is duidelijk te zien welke vlans welke IP-reeksen toegekend gekregen hebben. Hoe de IP adressering in IPv4 voor de productie VLANs is gerealiseerd, is in de volgende paragraaf beschreven.

## 8.5 IPv4 adressering in de exchange netwerken

ISP peering LAN: Over het ISP peering LAN, wordt al het verkeer van de verschillende ISP's verzonden. Hiervoor is de volgende reeks gereserveerd 193.148.15.0 /24 Dit biedt de mogelijkheid tot 254 aansluitingen in dit netwerk. Klanten die een aansluiting hebben op de Amsterdam Internet Exchange, krijgen voor hun router(s) een IP-adres uit deze reeks. Deze adressen worden door de ISP's gebruikt om hun BGP/peering sessions mee op te zetten.

GRX LAN: Voor het GRX LAN wordt de volgende reeks gebruikt: 193.108.27.0 /24. Dit netwerk is verder gesubnet in de volgende netwerken:

Prefix	Purpose
<u>193.27.108.0/25</u>	<u>GRX Peering LAN</u>
<u>193.27.108.128/26</u>	<u>Unused</u>
<u>193.27.108.192/27</u>	<u>Unused</u>
<u>193.27.108.224/28</u>	<u>dns0.gprs LAN</u>
<u>193.27.108.240/28</u>	<u>dns1.gprs LAN</u>

Het netwerk dat wordt gebruikt voor het uitwisselen van GPRS verkeer, het grx peering lan, is 25 bits groot. Dit biedt de mogelijk tot 126 potentiële aansluitingen. De DNS root servers voor het .gprs domein zijn ook in dit VLAN opgenomen, er zijn hiervoor 2 servers. Namelijk dns0.gprs en dns1.gprs. In werkelijkheid zijn de twee DNS netwerken één netwerk. Echter er is rekening mee gehouden dat deze in de toekomst gescheiden worden, vandaar dat ze ieder in een ander subnetwerk zijn ingedeeld.

MDX LAN: Voor dit VLAN wordt het subnet 192.41.216.128/25 gebruikt. Dit biedt de mogelijkheid tot max 126 potentiële aansluitingen.

Multicast LAN: Voor het multicast LAN is de reeks 192.41.216.0 /25 gereserveerd.

## 8.6 IP adressering in de overige AMS-IX netwerken

Service LAN: Dit is het netwerk voor oa. de AMS-IX webserver, voor dit netwerk wordt de reeks 193.194.136.0/27 gebruikt. Dit houdt in dat het maximale aantal servers in dit netwerk 13 is (14 – routerinterface).

Management LAN: het subnet hiervoor is: 193.194.136.32/27. Alle servers uit het service lan en de switches (sw3/sw6) hebben een adres uit deze reeks.

Test LAN: De IP reeks voor het test lan is 193.194.136.64/28

Office/NOC LAN: Dit is het kantoor netwerk, voor dit netwerk is de volgende reeks gereserveerd “193.194.136.128/25”  
Het office netwerk is verder gesubnet, dit is gedaan om in het kantoor netwerk ook bepaalde reeksen voor vaste machines te kunnen gebruiken.  
Dit ziet er als volgt uit:

Range	Type	Description
<a href="#">129-141</a>	Static	Servers, routers, staging, etc.
<a href="#">142-171</a>	dhcp-dynamic	Guests, etc.
<a href="#">172-176</a>	static/dhcp-host	Reserved for Henk Steenman
<a href="#">177-181</a>	static/dhcp-host	Reserved for Romeo Zwart
<a href="#">182-186</a>	static/dhcp-host	Reserved for Arien Vijn
<a href="#">187-191</a>	static/dhcp-host	Reserved for Steven Bakker
<a href="#">192-250</a>	dhcp-host/dynamic	Office range
<a href="#">251-253</a>	static/dhcp-host	Office servers

Static: Statically allocated and configured

dhcp-host: Statically allocated but configured through DHCP

dhcp-dynamic: Dynamically allocated and configured with DHCP

## 8.7 Routing

Het switch platform van AMS-IX bestaat uit 4 switches welke op 4 verschillende locaties staan. Sara, Nikhef, Globalswitch en Telecity. In de colocation van Globalswitch staan de servers van AMS-IX. Bij globalswitch, hangt tevens een cisco 7204 router, welke verantwoordelijk is voor de routing van het ams-ix netwerk. Deze router zorgt ervoor dat alle VLANs bereikbaar zijn, of juist afgeschermd zijn en alleen bereikbaar zijn vanaf bepaalde netwerken. Deze router heeft als hostname rtr1.ams-ix.net en is de centrale router in het netwerk, zie figuur 21 voor layer3 overzicht.

### 8.7.1 Verbinding naar Internet

Rtr1-ams-ix.net, is verantwoordelijk voor BGP peering met een groot deel van onze klanten. Van een aantal van onze klanten krijgt rtr1.ams-ix.net de hele Internet route tabel, wat betekent dat AMS-IX transit van hun krijgen. Al het verkeer van en naar Internet gaat via rtr1.ams-ix.net.

### 8.7.2 Verbinding naar kantoor netwerk

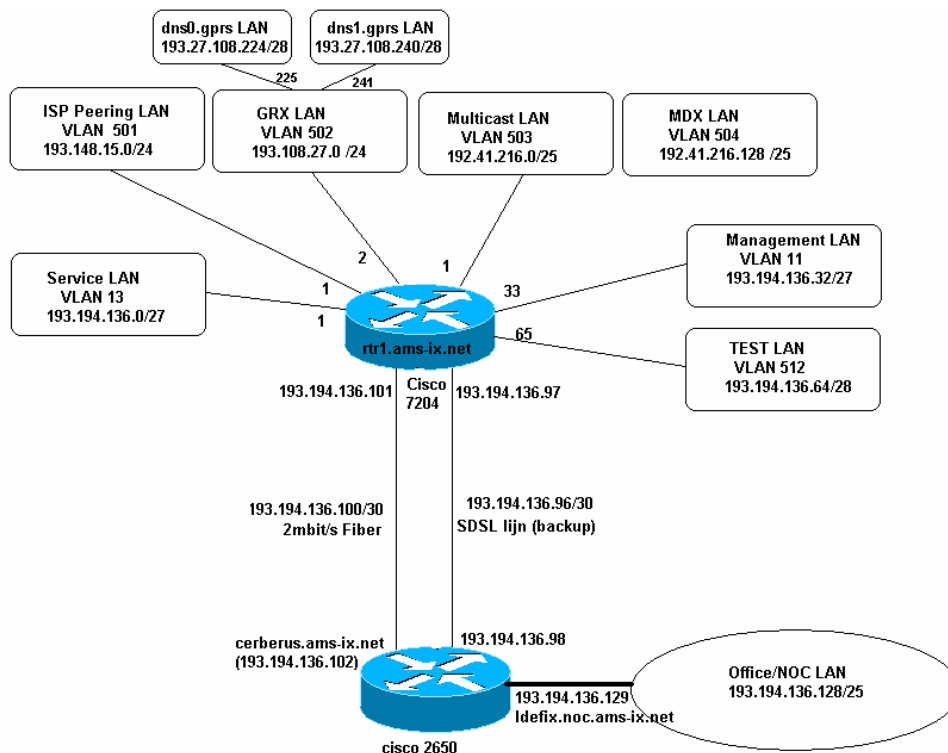
De office locatie van de Amsterdam Internet Exchange is gevestigd in het centrum van Amsterdam. Het kantoor netwerk is met een glasvezel (2Mbit/s) verbonden met rtr1.ams-ix. Mocht deze glasvezel verbinding op een of andere manier niet meer werken, dan is er een back-up verbinding beschikbaar. Deze back-up verbinding is een SDSL verbinding naar SARA, beide verbindingen worden afgenomen van Colt-telecommunicatie diensten.

De router op het kantoor heet idfix en is verantwoordelijk voor de routing naar rtr1.ams-ix.net, of via de glas verbinding, of via SDSL verbinding (zie fig 21). Op de kantoor router (idfix) zijn ook een aantal access-lists gedefinieerd welke het kantoor netwerk enige bescherming bieden. Voor medewerkers van de Amsterdam Internet Exchange, is het mogelijk om vanaf thuis een VPN verbinding op te zetten naar het AMS-IX office netwerk. Idefix heeft dus naast alleen de routing, ook een functie als firewall en VPN (IP-SEC) server.

### 8.7.3 Verbinding naar overige netwerken

Omdat ieder netwerk bereikbaar moet zijn voor het NOC, het network operations centre, heeft rtr1.ams-ix.net een verbinding naar elke VLAN, zie fig 21.

De gehele routing verloopt dus via rtr1.ams-ix.net.



Figuur 21

Op rtr1.ams-ix.net zijn access\_lists geïmplementeerd om ervoor te zorgen dat niet alle netwerken rechtstreeks vanaf Internet toegankelijk zijn, andere netwerken mogen maar gedeeltelijk toegankelijk zijn (service LAN). Al dit soort zaken zijn op rtr1.ams-ix.net geïmplementeerd.

Rtr1.ams-ix.net is een cisco 7200 router, zie figuur 22.



Figuur 22

## 9 IPv6 integratie in het AMS-IX netwerk

In het vorige hoofdstuk is uitéén gezet hoe het netwerk van de Amsterdam Internet Exchange is opgebouwd. In dit hoofdstuk zal worden beschreven hoe IPv6 is ingevoerd in het netwerk van AMS-IX, welke stappen hiervoor ondernomen zijn en hoe het IPv6 nummer plan eruit ziet.

Het uitgangspunt is dat alle belangrijke services, het kantoor netwerk en het de management netwerken via IPv6 bereikbaar dienden te zijn.

Er is een document gemaakt, waarin een plan is beschreven voor de invoering voor IPv6 in de AMS-IX netwerk omgeving. Hierin staat beschreven welke routers/servers wanneer in het AMS-IX IPv6 netwerk opgenomen zullen worden, en wat hiervoor noodzakelijk is,

Dit plan richt zich op de invoering van IPv6 in het service lan, office lan en management lan. Uiteraard dienen de routers welke de netwerken met elkaar verbinden ook geschikt gemaakt te worden. Het plan van aanpak hiervoor is beschreven in het “AMS-IX IPv6 integratie plan”, zie bijlage I (pagina 93). In het “AMS-IX IPv6 integratie plan” is ook een IPv6 nummer plan te vinden voor het AMS-IX netwerk.

### 9.1 Router

De eerste stap was het IPv6 gereed maken van de netwerk componenten in het AMS-IX netwerk. Hoewel de beide routers, idifix en rtr1, al geschikt waren voor IPv6 was er nog een groot probleem met de security. Met de IOS versies welke op de routers draaide was het niet mogelijk extended access-lists te configureren, waardoor het IPv6 netwerk niet goed afgeschermd kon worden, dit was een groot gat in de beveiliging. De eerste stap was dan ook de upgrade van één van de routers. Er is voor gekozen om rtr1.ams-ix.net te upgraden omdat deze een centrale functie heeft en over ruim voldoende geheugen beschikt, (Rtr1.ams-ix.net is de core router van AMS-IX). Deze router is verantwoordelijk voor de routing van het AMS-IX netwerk, op deze router zitten de verschillende netwerken aangesloten, waaronder het kantoor netwerk en het service LAN, in het service netwerk staan o.a. de webservice en de mailserver.

De router is inmiddels voorzien met, de op dat moment meest actuele versie van het Cisco IOS, IOS versie 12.2(13)T1, met als feature pack IP/FW/IDS. De Firewall feature set biedt de mogelijkheid om wat uitgebreider te filteren. Zonder deze feature set is het alleen mogelijk extended access-lists te implementeren. Er is echter gekozen om zogenaamde reflective access-lists te gebruiken, hiervoor is de FW feature nodig.

Reflective access-lists bieden de mogelijkheid om dynamisch poorten te openen voor inbound verkeer, maar dan alleen voor verkeer waar door een interne host om gevraagd is.

#### **Voorbeeld:**

*Host A op het kantoor netwerk van AMS-IX vraagt de website [www.google.com](http://www.google.com) op.*

*Daarvoor zet Host A een TCP verbinding op naar [www.google.com](http://www.google.com) naar poort 80.*

*Op dat moment maakt de router automatisch een access-list aan welke terug komend verkeer van [www.google.com](http://www.google.com) poort 80 naar Host A en de bijbehorende poort toe staat. Deze ACL is geldig voor zolang de sessie duurt of tot dat een bepaalde Time-out verloopt*

Deze manier van filtering lijkt op het eerste gezicht veel op het gebruik van de established parameter welke bij extended access-lists gebruikt kan worden. Echter het gebruik van reflective

access-lists is toch net iets gebruiksvriendelijker en veiliger. Het grootste voordeel is dat het niet alleen met TCP verbindingen werkt, maar ook voor UDP verkeer.

Voor elk VLAN zijn reflective access-lists aangemaakt, hieronder staat hoe dit voor het office netwerk is gedaan, zie figuur 23:

```
ipv6 access-list INBOUND-v6office
remark " -----IPv6 security----- "
remark "allow anything to leave office network, reflect these to REFLECTOUT"
permit tcp 2001:7B8:200:2202::/64 any reflect REFLECTOUT
permit tcp 2001:610:140:2202::/64 any reflect REFLECTOUT
permit udp 2001:610:140:2202::/64 any reflect REFLECTOUT
permit udp 2001:7B8:200:2202::/64 any reflect REFLECTOUT
permit icmp any any

ipv6 access-list OUTBOUND-v6office
remark " allow ICMP traffic in, and all the reflected connections "
permit icmp any any
evaluate REFLECTOUT
remark " allow ssh, imaps, smtp and dns to panoramix "
permit tcp any host 2001:610:140:2202::2 eq 22
permit tcp any host 2001:7B8:200:2202::2 eq 22
permit tcp any host 2001:610:140:2202::2 eq 993
permit tcp any host 2001:7B8:200:2202::2 eq 993
permit tcp any host 2001:610:140:2202::2 eq 25
permit tcp any host 2001:7B8:200:2202::2 eq 25
permit tcp any host 2001:610:140:2202::2 eq 53
permit tcp any host 2001:7B8:200:2202::2 eq 53
permit udp any host 2001:610:140:2202::2 eq 53
permit udp any host 2001:7B8:200:2202::2 eq 53
remark " allow ssh to idefix"
permit tcp any host 2001:7b8:200:2202::1 eq 22
permit tcp any host 2001:610:140:2202::1 eq 22

remark " -----IPv6 security----- "
```

**Figuur 23**

Voor alle uitgaande verbindingen wordt een tijdelijke regel gemaakt voor het terug komende verkeer. Alle uitgaande verbindingen worden “ge-reflect” naar een access-list REFLECTOUT, bij al het inkomende verkeer wordt dan bekeken of dit bij een al tot stand gekomen verbinding hoort (evaluate REFLECTOUT), als dat zo is, wordt het verkeer toegestaan.

Voor het service en management lan zijn soortgelijke regels geïmplementeerd.

Router1 is tevens verantwoordelijk voor de routing naar Internet, hiervoor wordt BGP als routings protocol gebruikt. De Amsterdam Internet Exchange peert met bijna alle aangesloten partijen op de AMS-IX. Nog lang niet alle peers op de exchange maken gebruik van IPv6, echter degene die al wel IPv6 gebruiken hebben ook BGP sessies met AS1200 (AS nummer van de Amsterdam Internet Exchange) opgezet.

BGP versie 4, zoals deze op de meeste routers is geïmplementeerd biedt ondersteuning voor zogenaamde Multiprotocol Extensions. Dit stelt BGP in staat om niet alleen maar met IPv4 netwerken te kunnen werken, maar ook met andere layer3 protocollen (IPv6, IPX, etc...). Hierover zijn twee RFC's verschenen, RFC 2545 *Use of BGP-4 Multiprotocol Extensions for IPv6 Inter-Domain Routing* [7] en RFC 2858 *Multiprotocol Extensions for BGP-4* [8].



De Multiprotocol extensions voor BGP maken het dus mogelijk voor verschillende layer3 protocollen, verschillende BGP sessies op te zetten. In een BGP pakket is hiervoor een speciaal veld, welke aangeeft welk layer3 NLRI (network layer Reachability information) er wordt verzonden. Op rtr1 zijn zowel BGP sessies over IPv4 als over IPv6 opgezet. BGP sessies worden geïdentificeert met behulp van router ID's, in de praktijk wordt hiervoor nog wel het IPv4 adres gebruikt. Dit komt omdat in het protocol 32bits zijn gedefinieerd voor het router ID, het ligt dus voor de hand het zelfde ID te gebruiken als voor IPv4 BGP sessies worden gebruikt, dit is vaak een IPv4 loopback adres.

Door gebruik te maken van de Multiprotocol Extensions van BGP, is het ook mogelijk IPv6 netwerk informatie naar IPv4 peers te verzenden, dit geldt ook andersom dus naar IPv6 peers kunnen IPv4 routes worden verstuurd.

In het figuur 24 staat een stuk Cisco configuratie, van hoe BGP voor IPv6 geconfigureerd kan worden.

```
router bgp 1200
  bgp router-id 193.148.15.1
  !
  neighbor ipv6peers peer-group
  neighbor 2001:7F8:1::A500:3265:1 remote-as 3265
  neighbor 2001:7F8:1::A500:3265:1 description XS4ALL - SARA
  neighbor 2001:7F8:1::A500:1103:1 remote-as 1103
  neighbor 2001:7F8:1::A500:1103:1 description SURFnet
  address-family ipv6
  neighbor ipv6peers activate
  neighbor ipv6peers soft-reconfiguration inbound
  neighbor ipv6peers route-map AS1200-IPV6-IN in
  neighbor ipv6peers route-map AS1200-IPV6-OUT out
  neighbor 2001:7F8:1::A500:3265:1 peer-group ipv6peers
  neighbor 2001:7F8:1::A500:1103:1 peer-group ipv6peers
  network 2001:610:140::/48
  network 2001:7B8:200::/48
  network 2001:7F8:1::/48
  exit-address-family
```

Figuur 24

In de bovenstaande BGP configuratie, wordt als routerID een IPv4 adres gebruikt, het ASnummer is 1200. In het voorbeeld staan 2 IPv6 peers, dit zijn AS3265 en AS1103 met de bijbehorende IPv6 adressen. Deze peers zijn in een peergroup ingedeeld dit is eenvoudiger voor BGP configuratie met een groot aantal peers. De geadverteerde netwerken zijn de prefixen die AMS-IX heeft toegewezen gekregen van BIT en Surfnet. Deze zijn in de configuratie te herkennen aan het "network" commando, hier had dus ook een IPv4 prefix tussen kunnen staan.

## 9.2 Webservers

In het netwerk van de Amsterdam Internet Exchange zijn een aantal webservers aanwezig. Uiteraard is er de webserver voor de website van de Amsterdam Internet Exchange, maar ook de intranet server en de web interface voor de mailinglist server.

De webservers welke gebruikt worden zijn Apache webservers, dit is de meest gebruikte webserver op Internet. De nieuwste versie van Apache is versie 2.0, deze versie heeft in tegenstelling tot de 1.3.x serie native ondersteuning voor IPv6. Het is ook mogelijk de 1.3.X serie te gebruiken hiervoor dient dan wel eerst de source code gepatched te worden.

Gekozen is voor het gebruik van de Apache 2.0.x serie, mede omdat deze tegenwoordig standaard worden meegeleverd met RedHat installaties.

### Configuratie

Om de webserver ook te laten “binden” aan IPv6 sockets zijn een aantal regels configuratie nodig. De configuratie file heet httpd.conf, de specifieke IPv6 configuratie ziet er als volgt uit:

Als eerste dient in de globale configuratie opgegeven te worden op welke adressen de webserver moet binden, zie figuur 25. Dit zijn dus de adressen en poortnummers waarna de http verzoeken verstuurd zullen worden.

```
Listen [2001:610:140:a604::2]:80
Listen [2001:7b8:200:a604::2]:80
Listen 193.194.136.6:80
```

Figuur 25

In dit geval zal apache luisteren op drie verschillende adressen, twee daarvan zijn IPv6 adressen. Omdat de meeste webservers meerdere virtualhosts draaien, moet hiervoor een soort gelijke configuratie gedaan worden, zie figuur 26.

```
NameVirtualHost 193.194.136.6
NameVirtualHost [2001:610:140:a604::2]
NameVirtualHost [2001:7b8:200:a604::2]
```

Figuur 26

Vervolgens moet er per virtualhost wat configuratie worden gedaan, een voorbeeld hiervan is te zien in figuur 27.

```
<VirtualHost 193.194.136.6 [2001:7b8:200:a604::2] [2001:610:140:a604::2]>
  ServerAdmin      webmaster@ams-ix.net
  DocumentRoot     /var/www/html/ams-ix/data
  DirectoryIndex   index.html index.htm home.html
  ServerName       www.ams-ix.net
  ScriptAlias      /cgi-bin/ /var/www/html/ams-ix/cgi-bin/
  <Directory /var/www/html/ams-ix/data>
    AllowOverride All
  </Directory>
  ErrorLog         /var/log/www/ams-ix/error.log
  CustomLog        /var/log/www/ams-ix/access.log combined
  EnableSendfile Off
</VirtualHost>
```

Figuur 27

Deze virtualhost luistert op alle drie de gedefinieerde IP adressen, waarvan er twee IPv6 adressen zijn. De overige informatie is standaard, zo is er opgegeven waar de documentroot zich bevindt, wat de servername is (www.ams-ix.net in dit geval) en waar de logging gedaan moet worden. De optie “ EnableSendfile Off” heeft te maken met een bug in de IPv6 stack van Linux, deze optie is een “workaround” .

Na deze configuratie wijzigingen is de webserver geschikt voor IPv6, uiteraard dient het systeem zelf al wel geschikt te zijn voor IPv6 en met de juiste IPv6 adressen geconfigureerd te zijn.

Tot voor kort draaide de website van de Amsterdam Internet Exchange (<http://www.ams-ix.net>), op een server welke niet geschikt was voor IPv6. Dat wil zeggen dat het operating system wat gebruikt wordt (Solaris 7 SunOS 5.7) is standaard niet geschikt voor het gebruik van IPv6. Om de website [www.ams-ix.net](http://www.ams-ix.net) toch voor IPv6 cliënts beschikbaar te stellen is er een reversed proxy geïmplementeerd op een webserver die wel geschikt is voor IPv6. De configuratie hiervoor is te zien in figuur 28.

```
<VirtualHost [2001:610:140:a604::2] [2001:7b8:200:a604::2]>
  ServerAdmin      noc@ams-ix.net
  ServerName       www.ams-ix.net
  ProxyRemote      http://www.ams-ix.net/ http://www.ams-ix.net/
  ProxyVia         On
  ProxyPass        / http://www.ams-ix.nl/
  ProxyPassReverse / http://www.ams-ix.nl/
</VirtualHost>
```

Figuur 28

Het AAAA record voor [www.ams-ix.net](http://www.ams-ix.net) wijst naar het IPv6 adres van de webserver welke dienst doet als reversed proxy. De verzoeken komen dus bij de reversed proxy binnen, deze haalt vervolgens de pagina over IPv4 op bij [www.ams-ix.nl](http://www.ams-ix.nl) (hiervoor bestaat alleen een IPv4 adres) en geeft deze vervolgens terug aan de originele IPv6 cliënt. Door dus gebruik te maken van een proxy principe, is een IPv4 only website toch te bereiken via IPv6.

Inmiddels is [www.ams-ix.net](http://www.ams-ix.net) verhuisd naar een nieuwe server, deze is nu wel rechtstreeks via IPv6 te benaderen.

### 9.3 Mailservers

Bij AMS-IX zijn meerdere mailservers in gebruik, melix is de primaire mailserver, de andere dienen als backup mailserver. Mailserver worden in de DNS weergegeven als MX records, voor AMS-IX zien deze mx records er als volgt uit:

```
host -t mx ams-ix.net
ams-ix.net mail is handled by 30 fulliautomatix.noc.ams-ix.net.
ams-ix.net mail is handled by 10 melix.ams-ix.net.
ams-ix.net mail is handled by 20 panoramix.noc.ams-ix.net.
```

```
host -t mx ams-ix.nl
ams-ix.nl mail is handled by 20 fulliautomatix.noc.ams-ix.net.
ams-ix.nl mail is handled by 10 melix.ams-ix.net.
```

Zoals hierboven is te zien, zijn er voor de domeinen van AMS-IX verschillende MX records gedefinieerd in de DNS servers, ieder met een verschillende prioriteit. Voor AMS-IX wordt de mail afgehandeld door 3 verschillende mailservers, ieder met een verschillende prioriteit.

- melix.ams-ix.net
- panoramix.noc.ams-ix.net
- fulliautomatix.noc.ams-ix.net

Het voordeel van meerdere mailservers is dat er altijd een backup server is. De prioriteit (dit zijn de getallen die worden meegegeven bij het opvragen van de MX record hierboven) geeft aan welke mailserver de primaire server is. De primaire mailserver voor de beide domeinen is melix.ams-ix.net. Tijdens de migratie naar IPv6 is er voor gekozen in eerste instantie alleen melix.ams-ix.net te upgraden. De andere 2 servers kunnen in een later stadium ge-upgrade worden. Het is verstandig hiermee een tijdje te wachten, zodat eventuele bugs en/of configuratie fouten ontdekt kunnen worden.

Bovendien is het zo dat de meeste mailservers, zo zijn geconfigureerd, dat als ze de mail niet via IPv6 kunnen afleveren, ze dit via een IPv4 connectie zullen doen. Door deze aanpak kan AMS-IX er van verzekerd zijn dat de mail altijd aan blijft komen.

Bij de Amsterdam Internet Exchange wordt gebruik gemaakt van de MTA (smtp server) software “postfix”; deze server software is voor Unix /Linux machines. Normaal gesproken wordt deze bij AMS-IX vanuit een RedHat pakket (RPM) geïnstalleerd. De standaard postfix RPM biedt geen ondersteuning voor IPv6. Er is om deze reden een eigen RPM gemaakt uit de originele source code van postfix en een IPv6 patch [9]. De configuratie van postfix is grotendeels hetzelfde als voor een IPv4 installatie. De enige extra configuratie is het opgeven van de lokale IPv6 netwerken waarvoor de mailserver dient te relaysen. Dit ziet er als volgt uit, zie figuur 29:

```
mynetworks = 193.194.136.0/24, localhost, [::1]/128, 127.0.0.1/32,  
             [2001:7b8:200:2202::]/64, [2001:610:140:2202::]/64
```

Figuur 29

Deze configuratie wordt gedaan in het configuratie bestand main.cf (/etc/postfix/main.cf).

Zoals is weergegeven in figuur 29, zijn er 3 IPv6 prefixen opgegeven; één is het loopback adres [::1], de overige twee zijn de prefixen welke gebruikt worden op het office LAN.

Na de installatie en configuratie is de smtp server geschikt voor IPv6. Op dit moment zijn er gemiddeld zo'n 10 smtp servers per dag waarmee melix.ams-ix.net over IPv6 communiceert.

## 9.4 Dns

Het upgraden van software op de verschillende servers heeft geen enkele zin, als er niemand over IPv6 deze services opvraagt. Omdat te laten gebeuren zijn er dns entry's nodig om de IPv6 adressen bekend te maken. De DNS server voor het ams-ix domain zijn:

1. nemix1.ams-ix.net
2. nemix2.ams-ix.net

Voor de DNS server software wordt gebruik gemaakt van de software van ISC (Internet Software Consortium), dit is op Internet de meest gebruikte DNS server software.

De laatste versie van BIND (versie 9), biedt native ondersteuning voor IPv6. De software hoeft dus niet aangepast of gepatched te worden. Om de DNS server ook op IPv6 sockets te laten binden, zijn een paar kleine configuratie wijzigingen noodzakelijk.

Als eerste dient opgegeven te worden in de configuratie file named.conf, dat de server ook op zijn IPv6 adressen dient te luisteren, dit dient opgegeven te worden in het "options" gedeelte van de config file met het statement *listen-on-v6 { any; };*. Een voorbeeld configuratie is te zien in figuur 30.

```
options {
    directory "/var/named";
    allow-transfer { other-dns; };
    allow-recursion { ams-ix-clients; };
    listen-on-v6 { any; };
    notify-source 193.194.136.4;
};
```

Figuur 30

De nameserver is geconfigureerd met een aantal access-lists, zodat niet iedereen de zone transfers en recursive lookups kan doen. De ACL's moeten dus aangepast worden zodat ook de juiste IPv6 prefixen worden toegestaan. Hoe de ACL's er uitzien, is te zien in figuur 31.

Nu de Nameserver gereed is voor DNS queries over IPv6, moeten er ook DNS records aangemaakt worden voor de IPv6 adressen. In IPv4 worden A records gebruikt om een DNS naam aan een IP adres te koppelen. Een voorbeeld daarvan is: [www.ams-ix.net](http://www.ams-ix.net) deze hostname heeft als IP adres 193.194.136.2. De IPv6 equivalent van een A record is het AAAA record (quad-A).

[www.ams-ix.net](http://www.ams-ix.net) heeft twee quad-A record, namelijk:

```
andree@panoramix[master]$ nslookup -sil
> set type=aaaa
> www.ams-ix.net
Server:                ::1
Address:               ::1#53

www.ams-ix.net  has AAAA address 2001:7b8:200:a604::2
www.ams-ix.net  has AAAA address 2001:610:140:a604::2
```

```
acl localhost6 { // All local IPv6 addresses.
  ::1;
};

acl local-only { // All local interfaces (v4 & v6).
  localhost;
  localhost6;
};

acl ams-ix { // AMS-IX addresses
  193.194.136.0/24; // AMS-IX IPv4 LANs.
  2001:610:140::/48; // AMS-IX IPv6 LAN (1).
  2001:7b8:200::/48; // AMS-IX IPv6 LAN (2).
  local-only; // This host's interfaces.
};

acl other-dns {
  // Your secondary DNS servers go here...
  192.87.106.101; // NS1.SURFnet.nl
  192.87.36.2; // NS2.SURFnet.nl
  145.41.1.167; // NS3.SURFnet.nl
  193.176.144.128/28; // NL domain registration.
  ams-ix;

  213.136.0.66; // ns.bit.nl
  213.136.0.77; // ns2.bit.nl
  62.250.7.46; // ns3.bit.nl
};

acl ams-ix-clients { // Clients that we consider "local"
  // This list includes really local addresses, that are part
  // of the AMS-IX infrastructure, and various "home" systems.
  ams-ix;
  194.109.222.156;
  213.84.213.66;
  213.84.188.208;
};
```

Figuur 31

Voorbeelden van AAAA records uit de zone file van ams-ix.net, zijn te zien in figuur 32.

nemix1	1D	IN	AAAA	2001:610:140:2202::2
nemix1	1D	IN	AAAA	2001:7b8:200:2202::2
nemix2	1D	IN	AAAA	2001:610:140:a604::4
nemix2	1D	IN	AAAA	2001:7b8:200:a604::4
www	1D	IN	AAAA	2001:610:140:a604::2
www	1D	IN	AAAA	2001:7b8:200:a604::2
melix	1D	IN	AAAA	2001:610:140:a604::3
melix	1D	IN	AAAA	2001:7b8:200:a604::3

Figuur 32

## 9.5 Overige services.

In de voorgaande paragrafen zijn reeds de belangrijkste services behandeld. Uiteraard zijn er nog een aantal andere services welke veel gebruikt worden, maar dan vooral door de medewerkers van de Amsterdam Internet Exchange zelf. Voorbeelden hiervan zijn de SSH servers op verschillende Linux servers. De OpenSSH servers welke worden gebruikt in het AMS-IX netwerk zijn allemaal geschikt voor communicatie over IPv6, het enige wat hiervoor gedaan dient te worden is het systeem zelf IPv6 geschikt te maken.

Bij AMS-IX worden voornamelijk RedHat Linux servers gebruikt, deze zijn allemaal geschikt voor IPv6 het enige wat nog gedaan dient te worden is de machine configureren met de juiste IPv6 adressen.

Eerst dient er in de globale netwerk configuratie file opgeven te worden, dat IPv6 support in de kernel geladen dient te worden.

Dit wordt gedaan, door in het configuratie bestand `/etc/sysconfig/network` de regel `"NETWORKING_IPV6=yes"` op te nemen.

Vervolgens kan er per interface nog wat specifieke IPv6 configuratie worden opgegeven, een voorbeeld daarvan is te zien in figuur 33: inhoud van `/etc/sysconfig/network-scripts/ifcfg-eth1`

```
DEVICE=eth1
<IPv4 configuratie weg gelaten>
IPV6INIT=yes
IPV6ADDR=2001:610:140:a604::2/64
IPV6ADDR_SECONDARIES=2001:7b8:200:a604::2/64
IPV6_ROUTER=no
IPV6_AUTOCONF=yes
IPV6FORWARDING=no
```

Figuur 33

Zoals te zien is in figuur 34, heeft deze machine twee IPv6 adressen en doet ook nog aan autoconfiguratie.

Het uiteindelijk resultaat is dan ook dat deze host 4 (global unicast) IPv6 adressen heeft op de interface eth1.

```
eth1
  Link encap:Ethernet  HWaddr 00:30:48:24:FC:3D
  inet6 addr: 2001:610:140:a604::2/64 Scope:Global
  inet6 addr: 2001:7b8:200:a604::2/64 Scope:Global
  inet6 addr: fe80::230:48ff:fe24:fc3d/10 Scope:Link
  inet6 addr: 2001:610:140:a604:230:48ff:fe24:fc3d/64 Scope:Global
  inet6 addr: 2001:7b8:200:a604:230:48ff:fe24:fc3d/64 Scope:Global
  UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
  RX packets:416466 errors:0 dropped:0 overruns:0 frame:0
  TX packets:401801 errors:0 dropped:0 overruns:0 carrier:0
  collisions:0 txqueuelen:100
  RX bytes:95829952 (91.3 Mb)  TX bytes:181283213 (172.8 Mb)
```

Figuur 34

Door gebruik te maken van zowel autoconfiguratie als statische adressen, hoeft er geen default gateway te worden opgegeven, omdat nu de router welke de prefixen adverteert, gebruikt wordt als gateway.

Andere services welke nu onder IPv6 beschikbaar zijn, zijn onder andere de FTP server op de webserver. De IPv6 enabled FTP server biedt tevens support for EPSV and EPRT commando's [10]. Dit zijn commando extensions voor de reeds gedefinieerde PSV en PRT commando's, zodat ze nu ook voor IPv6 gebruikt kunnen worden. Deze uitbreiding van de commando's waren nodig omdat de het formaat van de IPv6 adressen een stuk langer is dan die van IPv4. Omdat de parameters van de beide commando's IPv6 adressen zijn, moest er een nieuw commando in het leven geroepen worden. In figuur 35 is een voorbeeld van een ftp sessie over IPv6 te zien, waarin de nieuwe extensies worden gebruikt.

```
ftp 2001:610:140:a604::2
Connected to 2001:610:140:a604::2.
220 (vsFTPD 1.1.3)
Name (2001:610:140:a604::2:andree): andree
331 Please specify the password.
Password:
230 Login successful. Have fun.
ftp> ls
229 Entering Extended Passive Mode (|||44528|)
150 Here comes the directory listing.
```

Figuur 35

Het is nu dus mogelijk om met een FTP cliënt welke geschikt is voor IPv6 de content van de website bij te werken. Uiteraard is de FTP server ook nog gewoon over IPv4 bereikbaar.

Medewerkers van de Amsterdam Internet Exchange, kunnen hun email ophalen met behulp van verschillende soorten protocollen. Er kan gekozen worden uit POP3 of IMAP2, deze twee services zijn ook via een beveiligde SSL verbinding te benaderen pop3s imap2s. Vanaf Internet zijn alleen de met ssl beveiligde verbindingen te gebruiken om de mail op te halen (pop3s en imaps).

De mailserver software welke hiervoor gebruikt wordt is de courier mailserver. Volgens de specificaties zou deze ondersteuning moeten hebben voor IPv6. Helaas kan deze software niet goed overweg met de IPv6 implementatie van SUN Solaris en werkt deze software alleen op Linux systemen correct met IPv6.

Om de POP3(s) en IMAP2(s) servers toch ook voor IPv6 beschikbaar te maken, wordt er gebruik gemaakt van een 6to4 proxy. Dit is een relatief eenvoudig stukje software (800 regels C++ code) welke op de opgegeven IPv6 poorten luistert en al het verkeer forward naar een opgegeven IPv4 adres en poort. De 6to4 proxy wordt opgestart met een aantal parameters, zie onderstaande voorbeeld:

```
./ipv6tunnel ALL6 995 193.194.136.132 995 > pops3-ipv6tunnel.log 2>&1 &
```

In het bovenstaande voorbeeld luistert de 6to4 proxy op al zijn IPv6 adressen (ALL6) op poort 995. Al het verkeer wat ontvangen wordt op die betreffende sockets wordt geforward naar het IPv4 adres 193.194.136.132 poort 995, al het terug komende verkeer wordt uiteraard terug gestuurd. Alle verbindingen worden tevens netjes gelogd, zodat net zoals IPv4 connecties altijd kan worden nagegaan vanaf waar is ingelogd.

Het principe van een 6to4 of 4to6 proxy is vrij eenvoudig, maar in allerlei situaties in te zetten.

Het is een ideale oplossing voor gevallen waarbij de server software nog niet geschikt is voor IPv6, maar waarvoor het toch wenselijk is dat de betreffende service ook onder IPv6 beschikbaar is.

Nu alle servers beschikbaar zijn voor IPv6 cliënten is het zaak de hosts op het kantoor netwerk te voorzien van een IPv6 adressen. De hosts op het kantoor netwerk bestaan voornamelijk uit



Windows XP, Apple en Linux operating systems. Al deze operating systems zijn geschikt voor IPv6. De host op het kantoor netwerk zullen allemaal voorzien worden van 2 adressen, dit wordt gedaan door middel van autoconfiguratie. De betreffende prefixen zullen worden geadverteerd door de kantoor router Idefix, de configuratie hiervoor is te zien in figuur 36:

```

idefix#sh run | begin interface FastEthernet0/1
interface FastEthernet0/1
  description AMS-IX Office/NOC LAN
  ip address 193.194.136.129 255.255.255.128
  speed 10
  full-duplex
  ipv6 enable
  ipv6 address 2001:610:140:2202::1/64
  ipv6 address 2001:7B8:200:2202::1/64
  ipv6 nd ra-interval 30
  ipv6 nd prefix-advertisement 2001:610:140:2202::/64 600 300 autoconfig
  ipv6 nd prefix-advertisement 2001:7B8:200:2202::/64 600 300 autoconfig
  no cdp enable
!

```

**Figuur 36**

Hoewel cisco routers standaard aan router advertisements doen en daarbij vanzelf de bijbehorende prefixen adverteren, is er toch voor gekozen de default instelling wat aan te passen.

De Router Advertisements worden met de bovenstaande configuratie om de 30sec verzonden.

De prefixen die worden verzonden zijn 2001:610:140:2202::/64 en 2001:7B8:200:2202::/64, welke een preferred lifetime van 5 minuten hebben en een valid lifetime van 10 minuten.

Hosts op het netwerk zullen zich zelf nu automatisch configureren met de juiste prefix, voor een windows-XP host ziet dat er uit als in figuur 37:

```

C:\>ipconfig
Windows IP Configuration

Ethernet adapter Local Area Connection:

    Connection-specific DNS Suffix  . : noc.ams-ix.net
    IP Address. . . . . : 193.194.136.171
    Subnet Mask . . . . . : 255.255.255.128
    IP Address. . . . . : 2001:7b8:200:2202:191d:a7f8:c616:a2d5
    IP Address. . . . . : 2001:7b8:200:2202:210:a4ff:fe1:893d
    IP Address. . . . . : 2001:610:140:2202:191d:a7f8:c616:a2d5
    IP Address. . . . . : 2001:610:140:2202:210:a4ff:fe1:893d
    IP Address. . . . . : fe80::210:a4ff:fe1:893d%4
    Default Gateway . . . . . : 193.194.136.129
                                fe80::207:ebff:fe46:48e1%4

```

**Figuur 37**

de andere adressen zijn zogenaamde “anonymous” adressen. Deze zijn in het leven geroepen om de privacy te beschermen, door het gebruik van de anonymous adressen is het niet mogelijk de identiteit van iemand te achterhalen aan de hand van zijn of haar mac adres.

Dit anonymous adres wordt met bepaalde intervallen ook veranderd, de specificaties zijn beschreven in RFC 3041.

Windows-XP kiest bij voorkeur de anonymous adressen voor uitgaande verbindingen. Het gebruik van anonymous adressen door Windows XP kan met het volgende commando worden uitgeschakeld: “ipv6 -p gpu UseAnonymousAddresses no”.

## 9.6 IPv6 adressering

Er zijn voor de Amsterdam Internet Exchange 2 IPv6 reeksen van ieder 48 bits beschikbaar, namelijk 2001:7b8:200:2202/48 en 2001:610:140:2202/48

inet6num	Range	LIR
2001:0610:0140::/48	2001:0610:0140:0000:0000:0000:0000:0000 2001:0610:0140:ffff:ffff:ffff:ffff:ffff	surfnet
2001:07B8:200::/48	2001:07b8:0200:0000:0000:0000:0000:0000 2001:07b8:021f:ffff:ffff:ffff:ffff:ffff	Bit

Deze beide /48 prefixen worden op dezelfde manier gesubnet, dus alleen de eerste 48 bits zullen verschillen op een host. Bijvoorbeeld de kantoor router (idefix) heeft de volgende 2 adressen:

1. 2001:7B8:200:2202::1
2. 2001:610:140:2202::1

Dus alleen de eerste 48 bits van de adressen verschillen, deze identificeren de providers welke deze reeksen routeren.

Deze adres ruimte wordt in eerste instantie onderverdeeld in 16 verschillende netwerken met ieder een prefixlenght van 52 bit. Deze netwerken kunnen dan gebruikt worden voor verschillende bedrijfs onderdelen. In het nummer plan is ruimschoots rekening gehouden met evt toekomstige groei. Op dit moment worden er van deze 16 prefixen 2 gebruikt.

Een reeks voor de offices en een voor de POPs.

Use	Prefix
Reserved	2001:xxxx:xxxx:0000::/52
Reserved	2001:xxxx:xxxx:1000::/52
<b>Offices</b>	<b>2001:xxxx:xxxx:2000::/52</b>
Reserved	2001:xxxx:xxxx:3000::/52
Reserved	2001:xxxx:xxxx:4000::/52
Reserved	2001:xxxx:xxxx:5000::/52
Reserved	2001:xxxx:xxxx:6000::/52
Reserved	2001:xxxx:xxxx:7000::/52
Reserved	2001:xxxx:xxxx:8000::/52
Reserved	2001:xxxx:xxxx:9000::/52
<b>POPs</b>	<b>2001:xxxx:xxxx:a000::/52</b>
Reserved	2001:xxxx:xxxx:b000::/52
Reserved	2001:xxxx:xxxx:c000::/52
Reserved	2001:xxxx:xxxx:d000::/52
Reserved	2001:xxxx:xxxx:e000::/52
Reserved	2001:xxxx:xxxx:f000::/52

Er zijn op dit moment vier POPs (point of presence), namelijk Sara, Nikhef, Telecity en GlobalSwitch. De andere 12 prefixen blijven gereserveerd voor toekomstig gebruik.

POPs	Prefix
reserved	2001:xxxx:xxxx:a000::/56
...	
reserved	2001:xxxx:xxxx:a200::/56
SARA	2001:xxxx:xxxx:a300::/56
NIKHEF	2001:xxxx:xxxx:a400::/56
TeleCity	2001:xxxx:xxxx:a500::/56
Global	
Switch	2001:xxxx:xxxx:a600::/56
reserved	2001:xxxx:xxxx:a700::/56

... reserved | 2001:xxxx:xxxx:af00::/56

De prefixen voor de verschillende VLANs komen uit de reeks die in het nummerplan is toegewezen aan GlobalSwitch(2001:xxxx:xxxx:a600::/56). De keuze hiervoor is logisch, omdat bij globalswitch de servers van AMS-IX staan.

Voor de AMS-IX VLANs wordt de volgende adressering gebruikt

VLAN	Prefix
Management LAN	2001:xxxx:xxxx:a602::/64
Service LAN	2001:xxxx:xxxx:a604::/64
Test LAN	2001:xxxx:xxxx:a607::/64

Er zijn dus ook in deze reeks nog een aantal /64 prefixen gereserveerd voor toekomstig gebruik.

Voor de Office locaties zijn de volgende /56 prefixen gereserveerd:

Offices	
Reserved	2001:xxxx:xxxx:2000::/56
Reserved	2001:xxxx:xxxx:2100::/56
Westeinde	2001:xxxx:xxxx:2200::/56
Reserved	2001:xxxx:xxxx:2300::/56
...	
Reserved	2001:xxxx:xxxx:2f00::/56

De reeks die op de huidige office locatie wordt gebruikt is, 2001:xxxx:xxxx:2202::/64

Dus hosts en servers op de office locaties krijgen een IPv6 adres uit deze reeks.

Het complete nummerplan staat vermeld in bijlage I.

## 10 Multihoming

Veel organisaties zijn tegenwoordig multihomed omdat hun Internet verbinding steeds belangrijker wordt, veel communicatie en betalingen worden in deze tijd over het Internet gedaan. Bedrijven worden dus steeds meer afhankelijk van hun Internet verbinding. Om de verbinding met Internet te betrouwbaar mogelijk te laten zijn, zijn veel organisaties en ISP's multihomed. Met de term multihoming wordt het volgende bedoeld [12]:

*A "multihomed" site is one with more than one transit provider.  
"Site-multihoming" is the practice of arranging a site to be multihomed.*

### 10.1 Verschillende soorten multihoming

Verschillende organisaties implementeren multihoming om verschillende redenen. Dit wordt gedaan omdat organisaties erg verschillen in grootte en groepen welke ze bedienen.

De eerste manier om te multihomen, is door de hosts binnen een site te voorzien van één IP adres en vervolgens deze adres reeks bereikbaar maken door verschillende ISP's met behulp van routerings protocollen (routing based).

Een tweede mogelijkheid om te multihomen, is door nodes te voorzien van meerdere adressen. Deze adressen zijn afkomstig van verschillende ISP's (host based). Als er één ISP door een storing niet langer te gebruiken is, kan het alternatieve adres gebruikt worden.

De eerste multihoming toepassing stelt een aantal eisen aan de "site", over het algemeen wordt deze manier voornamelijk toegepast door grotere partijen. Het routerings protocol wat hier wordt gebruikt is eigenlijk altijd BGP. Om BGP te gebruiken dient een site over een AS nummer te beschikken, om een AS nummer te krijgen dient er aan een aantal eisen voldaan te worden. Bovendien dient een site over een eigen IP reeks te beschikken. Deze zgn. PI space (Provider Independed) wordt vervolgens geadverteerd in de globale routing tabel. Om een PI adres reeks te krijgen in IPv4 dient een organisatie aan te kunnen tonen dat ze deze daadwerkelijk nodig heeft. De minimale grote is een /24 afhankelijk van de verwachte groei kan deze prefix kleiner zijn (betekend meer adressen).

In de IPv6 situatie is het voor veel partijen die nu een eigen PI IPv4 prefix hebben, niet langer mogelijk een Provider Independed reeks te krijgen, wat als gevolg heeft dat deze partijen afhankelijk worden van een ISP, hierover later meer.

ISP zullen over het algemeen multihoming bewerkstelligen door transit in te kopen bij een transit provider en zich zelf aansluiten op een Internet exchange zoals AMS-IX. In de praktijk zal het er op neer komen dat een hoop nationaal verkeer over de exchange zal gaan. Al het overige verkeer of wanneer er problemen zijn met de aansluiting naar/of op de exchange zelf, zal worden gerouteerd via de transit provider.

Deze scriptie zal voornamelijk gericht zijn op bedrijven/organisaties die voor hun Internet connectiviteit afhankelijk zijn van één of meerdere ISP's. Deze bedrijven/organisaties zelf zullen dus niet een groot aantal peers hebben, afgezien van de ISP's waarvan ze de Internetverbinding afnemen.

Dit zullen veelal bedrijven zijn die zelf niets met Internet doen maar er wel afhankelijk van zijn. Een voorbeeld zou kunnen zijn, een willekeurige research bedrijf wat voor veel van haar onderzoek afhankelijk is van Internet. Omdat ze in grote mate afhankelijk zijn van het Internet wil een dergelijk organisatie niet van één provider afhankelijk zijn. De netwerken van deze organisaties of bedrijven zullen verder benoemd worden als sites.

ISP zijn vaak groot genoeg om zelf multihoming te realiseren, door middel van een aansluiting op een Internet exchange en één of meerdere transit provider(s). ISP's zijn vaak ook groot genoeg om stuk globaal routereerbaar adres ruimte te krijgen, dit in tegenstelling tot de sites welke verder op besproken zullen worden.

## 10.2 Redenen voor multihoming

Een organisatie kan verschillende redenen hebben om multihomed te zijn [12], de belangrijkste redenen worden in de volgende paragrafen behandeld.

### 10.2.1 Redundancy

Wanneer een site multihomed is, is deze in zekere mate beschermd tegen het onbereikbaar worden vanaf “de rest” van het Internet. Dit kan verschillende oorzaken hebben, voorbeelden hiervan zijn:

1. Physical link failure  
Denk hierbij aan een kapotte glasvezel kabel, of een kapotte router.
2. Logical link failure  
Dit kan bijvoorbeeld een fout geconfigureerde router interface zijn, of een software bug op de routers.
3. Routing protocol failure  
Zoals een BGP peer reset, of een fout geconfigureerde peer.
4. Transit provider failure  
Een upstream provider die problemen ondervindt, heeft tot gevolg dat de klanten niet langer bereikbaar zijn via deze ISP.
5. Exchange failure,  
De Internet Exchange waar de ISP is aangesloten kan problemen krijgen met het exchange platform, al het verkeer wat normaal gesproken over de exchange verloopt, zal nu een andere route moeten nemen.

Het is niet ondenkbaar dat één van de hierboven genoemde punten zou kunnen voorkomen, een multihomed site is door de tweede uplink altijd gegarandeerd van een tweede pad.

### 10.2.2 Load Sharing

Een site heeft de mogelijkheid één uplink als back-up route te gebruiken en pas in te schakelen als de normale link niet langer beschikbaar is. Echter wanneer een site beschikt over meerdere uplinks, welke verzorgd kunnen worden door verschillende ISP's of Transit providers, beschikt de site over de mogelijkheid om traffic te verdelen over de verschillende uplinks. Hierbij is outbound loadsharing is een stuk eenvoudiger te realiseren dan Inbound loadsharing.

### 10.2.3 Performance

Een willekeurig bedrijf kan er voor kiezen om al het mail verkeer over de wat langzamere ADSL link te laten verlopen, terwijl real-time applicaties of SSH verkeer over een leased-line gaan. Voor email is de beschikbare bandbreedte triviaal, echter tijdens een SSH sessie is het bijzonder vervelend om steeds te moeten wachten tot de karakters in beeld verschijnen.

Deze keuzes kunnen ook afhangen van de performance van de upstream providers, wanneer één van deze partijen performance problemen ondervindt, kan er voor gekozen worden om verkeer over de alternatieve ISP te routeren.

### 10.2.4 Policy

Een organisatie kan er voor kiezen bepaalde protocollen over een specifieke ISP te routeren. Deze keuze kan gemaakt worden op basis van verschillende overwegingen, een voorbeeld is bijvoorbeeld dat een bepaalde ISP minder betrouwbaar is dan de andere. Wat betreft performance en privacy (netwerk ontwerp), belangrijk verkeer kan dan over de leased line worden verstuurd. Onbelangrijk verkeer zoals NNTP (news), wat meestal high volume is kan dan over de minder betrouwbare maar goedkopere aansluiting verlopen. Een ISP of bedrijf kan een aantal redenen hebben om bepaalde type verkeer over verschillende lijnen te routeren, al deze redenen zijn gebaseerd op een policy.

### 10.2.5 Independence

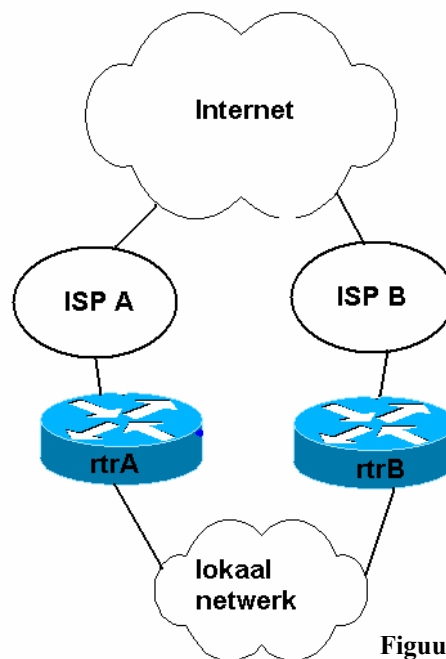
Het afhankelijk zijn van één partij welke verantwoordelijk is voor de Internet connectiviteit, is voor veel bedrijven een doorn in het oog. Het onafhankelijk willen zijn van één ISP's kan meerdere redenen hebben, dit kunnen zowel technische als politiek/economische redenen zijn.

Technische redenen zullen voornamelijk gebaseerd zijn op de redundancy, een bedrijf kan ook een aantal andere redenen hebben om onafhankelijk te zijn. Wanneer een organisatie beschikt over eigen IP adressen (Provider Independent), kan er zonder al te veel problemen van ISP worden gewisseld. Dit kan bijvoorbeeld voorkomen wanneer een ISP opgeheven wordt of wanneer een ISP ineens strenge regels gaat stellen met betrekking op monitoring of traffic filtering. Een voor de hand liggende reden kan natuurlijk ook zijn wanneer een alternatieve ISP, een goedkoper alternatief kan bieden of betere SLA's biedt.

## 11 Multihoming met IPv4

Wanneer een bedrijf of instelling zijn netwerk wil verbinden met het Internet, zijn hiervoor verschillende mogelijkheden. De meest voor de hand liggende mogelijkheid is een contract af te sluiten met Internet Service Provider (ISP), alle netwerk connectiviteit wordt dan verzorgd door de ISP. Maar wat nu als de ISP problemen heeft met zijn netwerk, of dat de router naar de ISP gaat kapot? Dit heeft tot gevolg dat het hele bedrijfsnetwerk onbereikbaar is geworden. Voor een hoop bedrijven is deze situatie onacceptabel. Zeker in deze tijd waar een hoop zaken via het Internet worden gedaan, moet een netwerk altijd bereikbaar zijn. Dit is één van de redenen voor bedrijven om te “multihomen”.

Een mogelijke oplossing is een tweede aansluiting afnemen, bij voorkeur van een tweede provider. Hierdoor is een bedrijf niet afhankelijk van één provider. De situatie is schematisch weergegeven in figuur 38.



in

Figuur 38

en

Het lokale netwerk heeft twee “exit routers” dit zijn rtrA en rtrB. Deze zijn verbonden met ISPA ISPB. Dit ontwerp heeft het volgende voordeel: wanneer zich een probleem voordoet met provider A, rtrA of de link naar ISP A, zal het verkeer via ISPB gerouteerd worden, zodoende wordt gebruik gemaakt van de redundancy feature. Uiteraard kan dit ontwerp ook worden uitgevoerd met maar één router, echter hiermee wordt wel een single point of failure gecreëerd. Want wanneer de router om wat voor reden dan ook niet juist functioneert, is zowel de uplink naar ISPa als ISPB weg.

Om een netwerk failure te detecteren en het verkeer om te leiden via een ander netwerk, is een detecterings mechanisme nodig. Hiervoor worden routerings protocollen gebruikt, het protocol wat wordt gebruikt voor routing op het Internet is het Border Gateway Protocol, version 4 (BGP4). Dit protocol is verantwoordelijk voor onder andere het bekend maken van de lokale netwerk prefixen aan de upstream ISP's (ISPa en ISPB). ISPa en ISPB worden in BGP termen “peers” genoemd. Tussen deze peers wordt informatie uitgewisseld over welke netwerken via welke routers (peers) te bereiken zijn. BGP heeft ook een detectie mechanisme, welke detecteert of een peer nog bereikbaar is of niet. Wanneer het BGP protocol op de router ontdekt dat bijvoorbeeld ISPa onbereikbaar is geworden, zal deze al het verkeer routeren via ISPB.

Om met BGP te kunnen werken, is het noodzakelijk over een eigen AS nummer te beschikken en een eigen IP adres reeks te hebben. Een AS nummer is een 16 bits getal waarmee een organisatie op het Internet geïdentificeerd wordt. Natuurlijk heeft een organisatie ook een stuk eigen adres ruimte nodig, welke deze kan adverteren. Zo'n adres ruimte is een Provider Independent reeks, dit wil zeggen dat deze reeks niet een gedeelte is van een andere adres reeks van een andere ISP. Om dit netwerk te kunnen gebruiken, zal deze geadverteerd moeten worden in de Internet route tabel, dit wordt ook wel de Default Free Zone genoemd. Alle routers in de core van het Internet moeten beschikken over de gehele Internet route tabel waarin dus ook een entry is opgenomen voor onze organisatie, zodat bekend is hoe het netwerk van de organisatie te bereiken is.

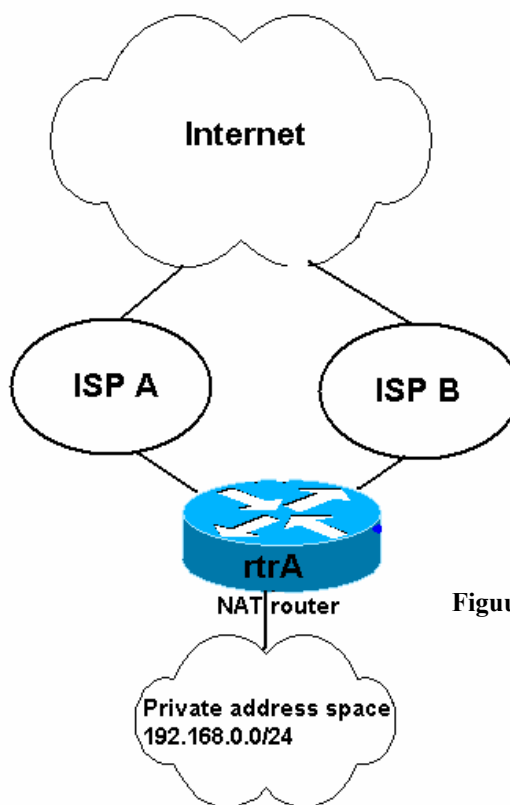
Dit is hoe multihoming in de IPv4 situatie wordt gerealiseerd, de eigen netwerk(en) via meerdere ISP/upstreams bekend maken in de Internet route tabel (DFZ). Zoals gezegd is het noodzakelijk de beschikking te hebben over een PI (provider Independent) adres reeks. Deze kan aanvraagd worden bij de daarvoor verantwoordelijke organisaties. Voor Europa is dat RIPE (Réseaux IP Européens) voor de overige wereld delen zijn er ARIN (American Registry for Internet Numbers), APNIC (Asia Pacific Network Information Centre) en LACNIC voor zuid Amerika.

Er is een tweede, relatief simpele manier waarop multihoming in IPv4 gerealiseerd kan worden. Deze manier wordt vooral geïmplementeerd door wat kleinere organisaties, welke niet voor een PI adres ruimte in aanmerking komen en niet de kennis voor BGP en alle bijkomende zaken hebben.

Een organisatie kan van 2 verschillende ISP's een verbinding afnemen. Bijvoorbeeld ISPA ADSL/mx-stream en ISPB kabel verbinding (Chello). Het interne netwerk kan bestaan uit een Ethernet netwerk waar private address space wordt gebruikt [13]. Dit interne netwerk zal dan door middel van een NAT/Firewall gekoppeld worden met zowel de DSL verbinding en/of de kabel uplink, zie voor een schematische weergave figuur 39.

In de praktijk wordt er vaak maar één verbinding tegelijkertijd gebruik, en wordt de tweede ISP pas gebruikt als er problemen zijn met de verbinding naar of bij de eerste ISP.

Uiteraard is het ook mogelijk beide verbindingen tegelijkertijd te gebruiken en zo load sharing te implementeren. Echter een reeds tot stand gekomen verbinding zal altijd via één ISP lopen, dit heet per destination load balancing (ipv per packet load balancing).



Figuur 39



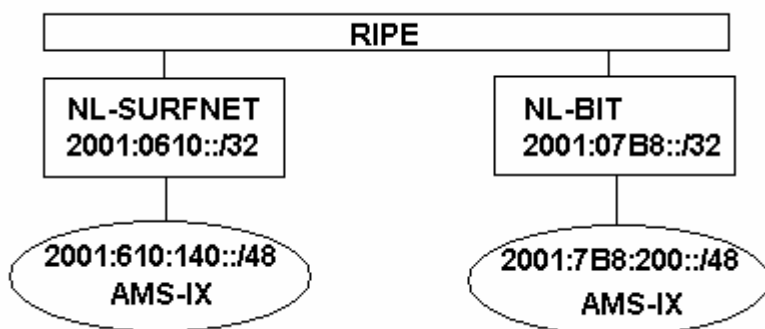
## 12 Multihoming met IPv6

IPv6 is ontworpen om de beperkingen van IPv4 (oa. het tekort aan adres ruimte en security) te verbeteren. Deze beperkingen zijn met de komst van IPv6 grotendeels opgelost, maar helaas is er een probleem bijgekomen namelijk multihoming.

Multihoming zoals dat in IPv4 wordt gerealiseerd, is theoretisch ook mogelijk in IPv6, echter met een beperkte schaalbaarheid. In het initiële ontwerp van IPv6, is gespecificeerd dat er in IPv6 veel meer gebruik gemaakt moet worden van aggregatie. Deze aggregatie is provider-based. Provider Independent (PI) space is voor IPv6 erg moeilijk te krijgen, wanneer een organisatie PI adres ruimte wil krijgen moet deze aan een aantal strenge eisen voldoen. Deze eisen hebben voornamelijk te maken met het aantal klanten en of potentiële nodes, die de organisatie zal moeten voorzien van Internet access. Denk hierbij bijvoorbeeld aan een ISP met klanten die allemaal een IP adres moeten hebben. Afhankelijk van het aantal klanten wordt een adresreeks toegewezen.

Over het algemeen krijgen Providers een prefix toegewezen met een lengte van 32 tot 35 bits, normaal gesproken krijgen de klanten van deze providers op hun beurt weer een /48 toegewezen.

De Amsterdam Internet Exchange heeft van Surfnets de reeks 2001:610:140::/48 toegewezen gekregen en van BIT 2001:7b8:200::/48, zie figuur 40.



Provider aggregatie

Figuur 40

Het is echter niet de bedoeling dat AMS-IX deze prefixen zelf gaat adverteren met BGP, zoals dat in de IPv4 architectuur gedaan wordt. Als iedere organisatie zijn /48 prefix zou gaan adverteren, dan zal de Internet route tabel op den duur veel te groot worden, routers hebben te weinig rekenkracht en geheugen om deze tabellen door te kunnen rekenen. Vandaar dat de verplichting tot aggregatie is ingevoerd, in principe mogen prefixen welke niet geaggregeerd zijn, dit is dus ook een /48, niet in de Internet route tabel voorkomen.

Door deze maatregel zullen de route tabellen niet “exploderen”, echter door deze maatregel dient zich ook een nieuw probleem aan, namelijk hoe multihoming te implementeren wanneer een organisatie niet de beschikking heeft PI adres ruimte (bijv een /32).

In de praktijk zal het er op neer komen dat een groot aantal organisaties (waarschijnlijk zelfs de meerderheid van de sites) welke nu multihomed zijn met behulp van BGP, in de IPv6 architectuur niet langer multihomed kunnen zijn. Dit omdat ze niet beschikken over een zogenaamde “slash 32” (/32) of een “slash 35” (/35). Dit heeft tot gevolg dat de sites welke nu multihomed zijn, in de IPv6 architectuur weer afhankelijk zullen zijn van één provider, met alle daarbij te bedenken nadelen (zie hoofdstuk 10.2, Redenen voor multihoming).

Om dit probleem aan te pakken heeft de IETF een speciale werkgroep in het leven geroepen, de “multi6” werkgroep [14]. Helaas heeft deze werkgroep tot nu toe niets geproduceerd, behalve requirement (ietf-multi6-multihoming-requirements-05 [12] ) waarin beschreven is waaraan een potentiële oplossing zal moeten voldoen.

Uit onvrede over de resultaten van de multi6 groep, zijn een aantal mensen begonnen met een eigen werkgroep IPv6mh[15], welke niet wordt gestuurd door de IETF. Op de mailinglist van deze werkgroep worden veel meer discussies gevoerd over mogelijke oplossingen, het lijkt er dan ook op dat deze werkgroep zo langzamerhand de functie van de Multi6 groep aan het overnemen is. Echter een concrete oplossing, of een advies van deze groep, is op korte termijn nog niet te verwachten. Hieruit kan geconcludeerd worden dat het multihoming probleem erg moeilijk is om op te lossen. De moeilijkheid ligt voornamelijk in de schaalbaarheid van de verschillen oplossingen, de security problemen welke een aantal potentiële oplossingen met zich meebrengen of de impact op reeds bestaande (transport) protocollen.

In het volgende hoofdstuk zullen een aantal potentiële oplossingen besproken worden, elk van deze oplossingen heeft zijn specifieke voor en nadelen.

### 13 Potentiële IPv6 multihoming Oplossingen

Er zijn meerdere potentiële oplossingen voor het IPv6 multihoming probleem mogelijk. Iedere oplossing heeft zijn eigen voor en nadelen, sommige zullen te ver gezocht zijn, of vereisen het ontwerp van een nieuw protocol, andere voorstellen zijn heel slim van opzet maar hebben security problemen.

In dit hoofdstuk zullen een aantal verschillende soorten oplossingen beschreven worden. Er zijn door een groot aantal mensen potentiële oplossingen aangedragen, alleen diegene die mijns inziens redelijkerwijs haalbaar zijn zullen beschreven worden.

De meeste oplossingen hebben als basis dat een node over meerdere “Provider aggregated” IPv6 adressen beschikt. Een Site zal in een dergelijke opzet verbonden zijn met een tweetal (of meer) ISP’s, van elk van deze ISP zal de organisatie een /48 ontvangen. Iedere node heeft nu de beschikking over meerdere globaal routeerbare adressen.

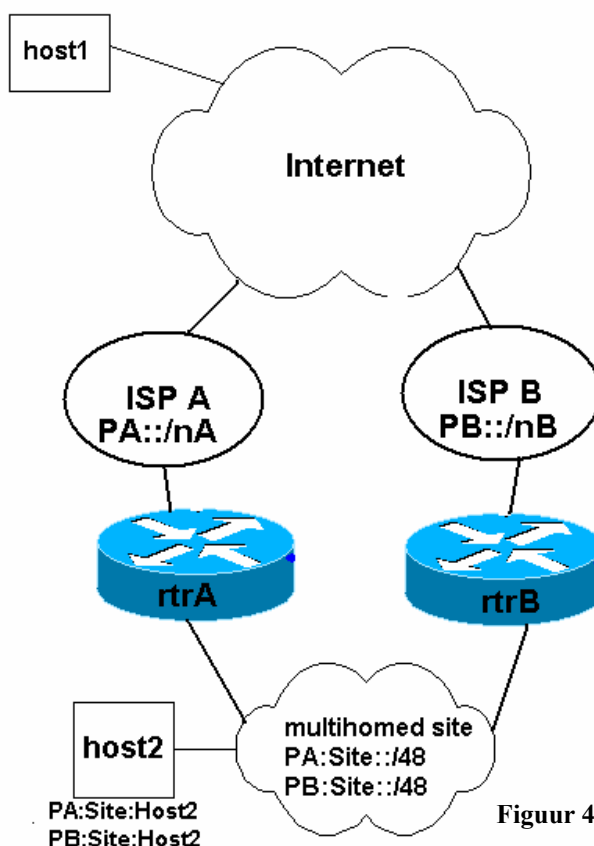
Figuur 41 geeft een multihomed site weer in een IPv6 omgeving, deze situatie is vergelijkbaar met de huidige situatie bij de Amsterdam Internet Exchange.

De multihomed site heeft de beschikking over twee Provider Aggregatable (PA) prefixen (dit zijn prefixen welke zijn toegewezen door de provider), namelijk PA::/nA en PB::/nB. Dus een Provider Aggregatable prefix van ISPA en een Provider Aggregatable prefix van ISPB. In de AMS-IX situatie zijn dit 2 prefixen van beide 48 bits, uit de IP reeks van Surfnet en Bit.

Om gebruik te maken van de voordelen van multihoming, moeten de hosts in de multihomed site (ams-ix netwerk) via beide ISP’s te bereiken zijn. Dit betekent dat iedere host een adres moet hebben uit beide reeksen: PA:Site:Host2 PB:Site:Host2.

De site/hosts zijn nu multihomed omdat ze zowel via ISPA als ISPB te bereiken zijn. Mocht er een probleem zijn bij één van de upstream providers, bijvoorbeeld ISPA, dan kan host1 (een host ergens op het Internet) nog steeds host2 (een host binnen het AMS-IX netwerk) benaderen via ISPB door het destination adres PB:Site:Host2 te gebruiken. Dit lijkt op het eerste gezicht goed te werken, maar toch zijn er een aantal grote nadelen aan deze configuratie.

De beschreven situatie laat bijvoorbeeld niet toe dat reeds opgezette verbindingen blijven bestaan op het moment dat een ISP wegvalt. Als host1 en host2 communiceerde via ISPA door gebruik te maken van het adres PA:Site:Host2 en de situatie doet zich voor dat ISPA vervalt (door bijvoorbeeld een linkfailure), dan zal dit tot gevolg hebben dat verbinding tussen host1 en host2 verloren zal gaan. Dit komt doordat connecties op de transport laag hun eindpunten identificeren op



Figuur 41

basis van het IP adres, dit als gevolg dat tijdens de duur van de verbinding de IP adressen niet mogen veranderen. Routing over het netwerk van ISPB zal niet werken, omdat ISPB alleen zijn eigen adres reeks routeert (PB::/nB). Pakketten voor een adres uit de reeks PA::/nA zullen altijd via ISPA gerouteerd worden, op het moment dat ISPA problemen heeft, zullen deze adressen dus niet meer te bereiken zijn, dit is het gevolg van het behouden van de aggregatie.

Als de aggregatie behouden blijft, kan de IPreeks dus niet gerouteerd worden via ISPB. Dit is een nadeel van het behouden van de aggregatie. Daar staat wel weer tegenover, dat het behouden van de aggregatie als voordeel heeft dat de grote van de route tabellen binnen de perken blijft. Wanneer IPv6 helemaal uitgerold is en de aggregatie is gebroken, zullen routers nooit alle routes kunnen bevatten (te weinig capaciteit).

### **13.1 Aanpassingen in de transport laag**

Zoals in de vorige paragraaf beschreven is, is de meest voor de hand liggende optie voor multihoming in IPv6 het gebruik maken van meerdere adressen per node. De node (host1) kan nu kiezen welk adres deze gebruikt. Wanneer de verbinding met ISPA verloren gaat en de node (host1) had verbindingen met een server ergens op Internet, dan zal deze verbinding verloren gaan. Theoretisch gezien zou host1 zijn alternatieve adres welke bij ISPb hoort kunnen gebruiken voor de rest van de communicatie. Echter hier tredt het probleem op dat TCP zijn connecties identificeert aan de hand van het sourceIPaddress, destinationIPaddress, sourceport en destinationport. Wanneer host1 zijn sessie wil voortzetten met zijn alternatieve adres dan zal de server dit niet herkennen als de voorgaande sessie, dit omdat het source adres is veranderd. TCP zal deze sessie dus herkennen als een nieuwe connectie omdat het peer adres is veranderd.

#### **13.1.1 Wijziging in de TCP stack**

Om toch het renumbering principe (het principe zoals hiervoor is beschreven, het gebruik van meerdere adressen per node) te kunnen gebruiken, zijn er voorstellen gekomen welke beschrijven dat de huidige één op één mapping tussen het adres en TCP connecties wordt veranderd naar de mogelijkheid om meerdere adressen op te geven. Wanneer een host een aantal alternatieve adressen voor z'n peer kent, kan het peer adres veranderen naar één van de bekende alternatieve adressen wanneer blijkt dat het originele adres niet langer bereikbaar is. Zodoende valt de TCP connectie niet weg, de verbinding zal voortbestaan echter nu met andere adressen. Het originele voorstel komt van Peter R. Tattam van Trumpet Software International Pty Ltd, de auteur draagt in zijn voorstel ook meerdere implementatie mogelijkheden aan. Een voorbeeld hiervan is dat tijdens het opzetten van de TCP verbinding (three way handshake) er alternatieve adressen worden meegestuurd. Zodoende wordt er tijdens het opzetten van de verbinding al onderhandeld over eventuele alternatieve adressen.

Dit idee is door een groep studenten uit Noorwegen reeds uitgewerkt, zij hebben een alternatieve TCP stack geschreven waarin de mogelijkheid bestaat om meerdere alternatieve adressen te gebruiken. Hieruit is gebleken dat het concept werkt en dat het wat de C(++) code betreft niet heel veel werk is. Meer informatie hierover is te vinden op:

<http://www.vermicelli.pasta.cs.uit.no/ipv6/students/troels/html/thesis/node27.html>

De TCP implementatie dient hiervoor wel te worden aangepast en omdat TCP wereldwijd gebruikt wordt zal het niet eenvoudig zijn de gemeenschap er van te overtuigen dat zulke wijzigingen nodig zijn. De gevolgen voor de security zijn minimaal vergeleken met de huidige TCP implementatie. Potentieel zijn er een hoop security issues te bedenken door dat de adressen mogen veranderen,

denk maar eens aan het hijacken (overnemen) van een TCP verbinding. Echter door dat de alternatieve adressen tijdens het opzetten van de verbinding al worden vast gelegd en tijdens de verbinding niet meer gewijzigd kunnen worden zal dit risico in de praktijk erg klein zijn.

### 13.1.2 Sctp

Het Stream Control Transmission Protocol (SCTP) [16] is een nieuw IP transport protocol en werkt op de zelfde laag als UDP en TCP. De laatste twee protocollen voorzien op dit moment bijna alle Internet applicaties van transport laag functies. Net als TCP is SCTP een betrouwbaar transport protocol, wat inhoudt dat de data die wordt afgeleverd altijd in de juiste volgorde aankomt en geen fouten bevat. Net als TCP is SCTP een connection-oriented protocol, wat betekent dat er een relatie tussen de twee end-points wordt opgebouwd, deze relatie wordt onderhouden voor zolang de connectie duurt. SCTP lijkt dus erg op TCP, maar het heeft een aantal belangrijke uitbreidingen. De twee belangrijkste features van SCTP zijn: *multi-streaming* and *multi-homing*. Op het begrip: multi-streaming zal verder niet ingegaan worden, echter de SCTP feature multi-homing zal hieronder verder worden toegelicht.

De wijzigingen in de TCP stack zoals in de voorgaande paragraaf is beschreven, is grotendeels al verwerkt in SCTP. SCTP geeft een end-point de mogelijkheid om meerdere IP adressen te ondersteunen. Dit betekent dat een SCTP verbinding een netwerk “failure” kan overleven.

In de huidige implementatie voorziet SCTP nog niet in de mogelijkheid om aan loadsharing te doen, er wordt één adres als “primary address” gekozen, dit adres wordt gebruikt om data te versturen. In het geval van hertransmissie van data wordt vervolgens het alternatieve adres gebruikt, zo wordt de mogelijkheid om het eindpunt te bereiken vergroot.

Om de multihoming te ondersteunen, wordt tijdens de initiële SCTP setup een lijst met adressen uitgewisseld welke kunnen worden gebruikt voor communicatie, dit kunnen zelfs meerdere IPv4 en IPv6 adressen zijn. Iedere SCTP host moet dus op verschillende adressen verkeer behorende bij één sessie kunnen ontvangen en verzenden, de gebruikte poortnummers zullen tijdens de gehele sessie het zelfde blijven.

De wijzigingen welke in TCP ingevoerd zouden kunnen worden om tijdens een renumbering een tcp sessie te overleven, zijn eigenlijk al geïmplementeerd in het SCTP protocol. Daarnaast biedt SCTP nog een aantal andere voordelen, een nadeel is echter dat SCTP nog nauwelijks gebruikt wordt en daadwerkelijk implementaties van SCTP staan nog in de kinderschoenen.

Een groot bijkomend voordeel is dat IPv4 adressen en IPv6 adressen bij het gebruik van SCTP door elkaar gebruikt kunnen worden tijdens een SCTP sessie.

SCTP voorziet in een aantal security verbeteringen wat betreft security ten opzichte van TCP. Het protocol is ontworpen met een “security cookie mechanism”, welke SYN-flood attacks voorkomt.

SCTP is een protocol wat betreft “running code” nog in de kinderschoenen staat. Voor Linux is er een kernel patch beschikbaar. In bijlage II, staat de uitwerking van een proef die is uitgevoerd met de nieuwste Linux kernel 2.5.67 en de sctp patch. In de proef zijn testen gedaan met betrekking op de multihoming feature van sctp.

Er is daarbij gebruik gemaakt van een software pakket welke gebruikt kan worden als echo server/cliënt. Bij het starten hiervan dienen een aantal parameters opgegeven te worden, onder andere een primair adres en een aantal alternatieve adressen waaraan lokaal gebind kan worden.

Getest is wat er gebeurt wanneer het primaire adres onbruikbaar wordt, blijft de sessie inderdaad bestaan en wat is de impact?

De uitwerking van deze test staat zoals reeds vermeld in bijlage II, uit de proef kan in ieder geval geconcludeerd worden dat de multihoming feature van SCTP in combinatie met de gebruikte software prima werkt.

### 13.2 Identifiers en locators

In RFC 2101[17] wordt beschreven hoe het gebruik van IPv4 adressen in de loop van de tijd is veranderd. Deze RFC maakte een belangrijk onderscheid tussen “locators” en “identifiers”. In IPv4 wordt voor beide begrippen het volledige IPv4 adres gebruikt. In een InternetDraft [18] worden de begrippen identifier en locator als volgt beschreven:

Identifier: A value that indicates the sender of a packet, or the intended recipient of a packet, i.e. the rightmost eight bytes of the address is an identifier.

**Opmerking: hoewel hier 8bytes worden genomen voor de identifier, kan dit in principe ook een ander getal zijn (bijv 64 bytes).**

Locator: A field in a packet header that is used by the routing subsystem to deliver a packet to the link on which a destination resides.

Locators en Identifiers zijn dus twee verschillende begrippen met ieder een eigen doel. Het doel van de identifier is om een node uniek te identificeren tijdens een sessie, de identifier mag niet veranderen. Een identifier dient minimaal net zo lang mee te gaan als de duur van de communicatie tussen 2 nodes. TCP gebruikt het IP adres van zijn peer als identifier, deze moet tijdens de gehele sessie het zelfde blijven.

Locators worden gebruikt om een host op het internet te lokaliseren, de locator wordt dus gebruikt door het routerings mechanisme op Internet. Routerings mechanisme houden bij welke netwerken via welke paden bereikt kunnen worden. In tegenstelling tot een Identifier, hoeft een locator niet tijdens de gehele sessie het zelfde te blijven.

Door een scheiding te maken tussen de Locator en Identifier mag de waarde van de Locator veranderen, zolang de peer maar weet wat die nieuwe locator is.

Door dat er in de huidige TCP/IP implementaties geen onderscheid wordt gemaakt tussen locators en identifiers heeft dit tot gevolg, dat als de Locator tijdens een sessie veranderd ook de Identifier veranderd. De IPv4 en IPv6 adressen fungeren dus zowel als identifier en locator.

IP adressen zijn geen ideale Locators of Identifiers, het zou een goed idee zijn deze twee functies te scheiden. In de ideale situatie blijft een identifier altijd toegewezen aan één node en wordt nooit opnieuw toegewezen aan andere node. Dit is ook niet nodig omdat de identifier altijd uniek dient te zijn. De locator wordt gebruikt om uit te zoeken waar een pakket naar toe moet, een tijdelijke levensduur van de locator is geen enkel probleem. Wanneer een host van netwerk veranderd, dit bijvoorbeeld als gevolg van een hernummering, heeft dit automatisch tot gevolg dat de Locator ook veranderd.

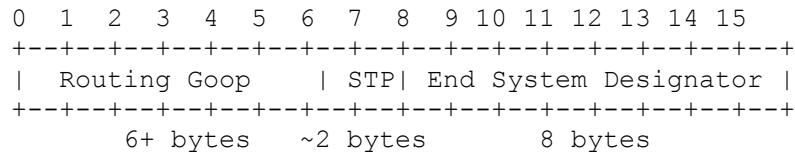
Er zijn verschillende voorstellen welke het principe van Locators en Identifiers gebruiken om het multihoming probleem op te lossen.

### 13.2.1 GSE proposal

Eén van de voorstellen waarin het principe van Locators en Identifiers is uitgewerkt is het GSE voorstel. GSE staat voor Global, Site and End-System Designator en is gebaseerd op het idee om een IPv6 adres op te delen in verschillende delen.

#### GSE Address Format

In tegenstelling tot zowel IPv4 als IPv6 waarin de locators en identifiers het zelfde zijn, stellen de auteurs van GSE voor om het IPv6 adres op te delen in de volgende delen:



Een typisch IPv6 adres bestaat uit 16 bytes en is onderverdeeld in drie delen, namelijk de prefix (48 bits), een site ID (16bits) en tot slot de interface ID (64 bits). De eerste 8 bytes wordt het "Routing Stuff" genoemd de laatste 8 bytes wordt gebruikt voor de identificatie.

Het principe van het GSE voorstel is dat de End-System Designator (ESD) alleen wordt gebruikt voor de identificatie van een interface. Het "routing stuff" gedeelte van het adres wordt, hoewel het wel wordt gebruikt om een pakket af te leveren op de juiste locatie, niet gebruikt ter identificatie van een sessie.

De communicerende end-points gebruiken voor de identificatie van hun sessies alleen de End System Designator; in het geval van TCP, worden peers dus geïdentificeerd op hun source en destination ESD, samen met de bijbehorende poortnummers.

Als globaal uniek End-System Designator(ESD), zou de "EUI-64" identifier gebruikt kunnen worden. Deze ID's zouden wereldwijd uniek moeten zijn omdat ze afgeleid zijn van het MAC-adres, in de praktijk blijkt echter dat dit niet altijd het geval is.

De auteurs van het GSE voorstel, stellen voor dat de end nodes zich niet bewust hoeven te zijn van de gebruikte routing-goop (de toegewezen prefix). De nodes in een site zouden als prefix gebruik kunnen maken van een sitelocal prefix (FEC0::/8); wanneer de pakketten de site verlaten dient de border router deze te herschrijven naar de juiste prefix. Wanneer PrefixA niet meer te gebruiken is, bijvoorbeeld door een probleem bij ISPA, dan zou de router de sitelocal prefix kunnen vervangen door prefixB. Het zelfde geldt uiteraard voor inbound pakketten, hierbij wordt de globale routable prefix vervangen door de gebruikte site local prefix.

Wanneer GSE geïmplementeerd zou worden, betekend dit dat het renumbering process in het geval dat een provider niet langer gebruikt kan worden, een stuk eenvoudiger wordt. Alleen op de border routers dienen hiervoor wijzigingen aangebracht te worden. Dit proces zou zelfs geautomatiseerd kunnen worden door middel van een protocol.

### 13.2.2 Een alternatief

Het herschrijven van de pakketen op de border routers is wellicht een wat al te vooruitstrevend idee, “echte end-to-end” communicatie gaat hiermee verloren. Daarom is er een alternatief mogelijk, waarbij wel gebruik gemaakt wordt van de End-System Designator(ESD) welke gevormd wordt door de laatste 64 bits (dmv EUI-64), maar waarbij niet de prefix wordt herschreven. TCP kan nu nog steeds gebruik maken van de identifiers om de sessie mee te identificeren en locators kunnen worden gebruikt om het juiste destination netwerk te vinden.

In tegenstelling tot het originele GSE idee, zouden end-nodes zich wel bewust moeten zijn van hun global routable unicast adres. In het geval van een renumbering kan de host de alternatieve prefix gebruiken als locator. Het wisselen van het adres maakt voor de TCP stack wederom niet uit omdat sessies worden geïdentificeerd door de identifier; het identifier gedeelte van het adres kan het zelfde blijven.

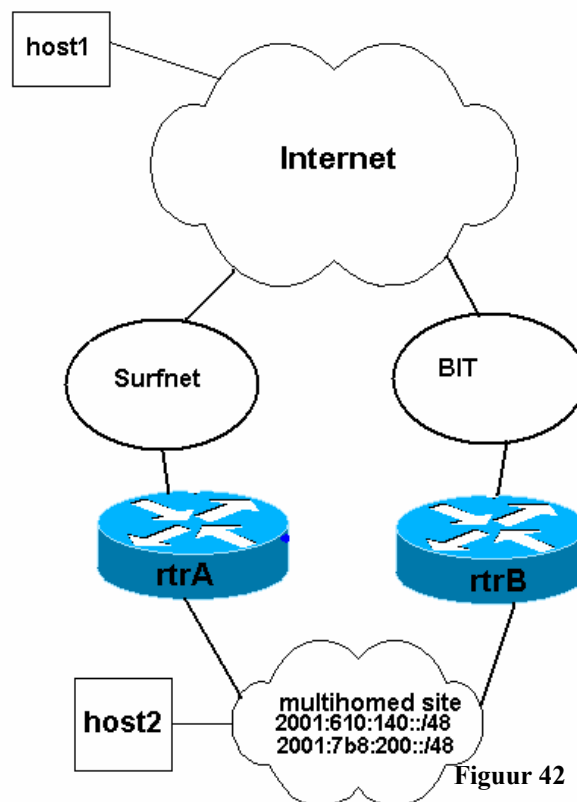
Een multihomed netwerk in IPv6 zou er als volgt uit kunnen zien, zie figuur 42.

Het IPv6 adres van host2 in dit voorbeeld is 2001:610:140:2202:a00:20ff:feec:8a24. De eerste 64 bits doen dienst als locator (prefix + site ID) en ziet er als volgt uit: 2001:610:140:2202. De identifier wordt gevormd door de laatste 64 bits (EUI-64) en ziet er als volgt uit a00:20ff:feec:8a24.

Als door gevolg van een storing bij ISP A/surfnet kan niet langer gebruik gemaakt worden van de surfnet locator 2001:610:140:2202. Host2 kan in dit geval switchen naar de prefix van Bit en dus 2001:7b8:200+site-id als locator gaan gebruiken zonder dat de sessie verloren gaat want de identifier blijft het zelfde.

Het implementeren van locators en identifiers brengt wel een aantal additionele zaken met zich mee, waarover nagedacht dient te worden. Een identifier alleen is niet voldoende om een pakket af te leveren, hiervoor is ook de bijbehorende locator nodig. Het probleem is hoe worden de locators aan de identifiers gekoppeld? Een tweede potentieel probleem is, dat door alleen de identifier door te geven aan een hogere laag, de kans op spoofing (je voor doen als iemand anders) toe neemt.

Er zijn een hoop mensen en organisaties bezig geweest met verschillende uitwerkingen van het scheiden van locators en identifiers. Twee initiatieven zijn zeker het vermelden waard, LIN6 en HIP. LIN6, dit staat voor location independent Addressing for IPv6. LIN6 is ontwikkeld voor mobiele nodes, welke steeds van netwerk/locatie veranderen. Weer een andere uitwerking is HIP, het Host Identity Payload and protocol. Alle pakketten blijven hetzelfde, maar de transportlaag gebruikt alleen de identifier om de sessie te identificeren. Het grote verschil met andere uitwerkingen is dat HIP erg gericht is op security. Er wordt gebruik gemaakt van een 4way handshake waarin alle parameters voor de end-to-end encryptie worden uitgewisseld. Het grote voordeel is dat het daardoor erg veilig is. Er zijn op dit initiatieven om dit protocol te laten werken op diverse platformen, zoals NetBSD en Linux.



Figuur 42

locator:identifier  
2001:610:140:2202:a00:20ff:feec:8a24



### 13.3 Mobile IPv6

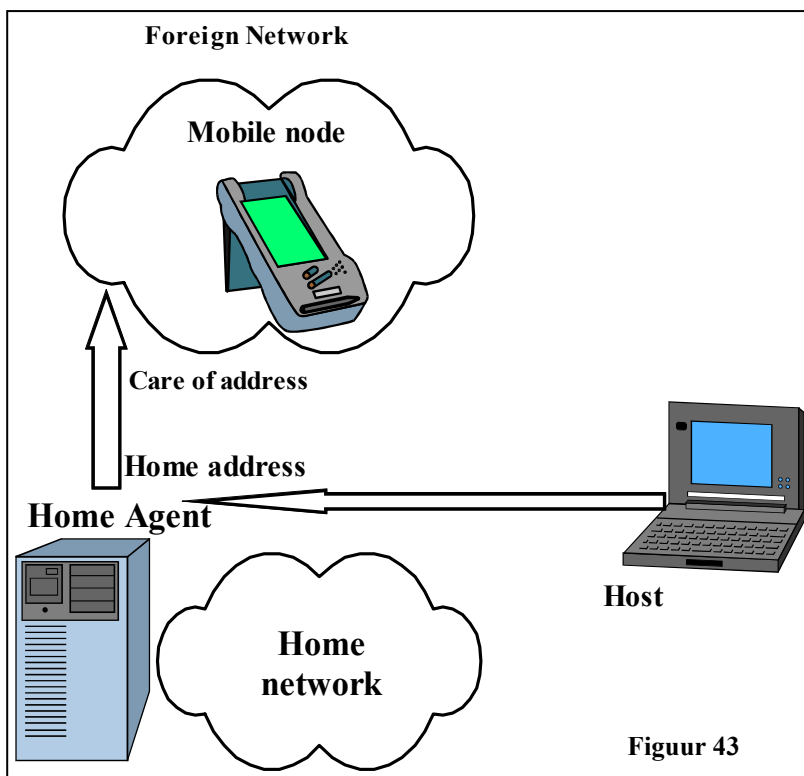
Tegenwoordig is het gebruik van mobiele telefonie enorm populair, bijna elke inwoner van Nederland heeft een mobiele telefoon. IPv6 is ontworpen met de gedachte dat het tekort van IPv4 adressen mede veroorzaakt zou worden door de enorme toename van mobiele apparatuur. Al deze mobiele nodes zullen in de toekomst voorzien moeten worden van IPv6 adressen, dit zijn niet alleen mobiele telefoons maar ook PDA's en andere mobiele apparatuur.

Het grote probleem van Mobile nodes is dat ze steeds veranderen van locatie. Hoewel dit natuurlijk het principe van mobile nodes is, is het een probleem om steeds op het zelfde IP adres bereikbaar te blijven. Het feit dat mobile nodes zich steeds verplaatsen, heeft tot gevolg dat ze steeds op een ander mobiel netwerk terecht kunnen komen. Stel je voor dat je een PDA hebt met een 802.11b (WIFI) interface en een General Packet Radio Service (GPRS) interface. Vanuit je hotelkamer is er verbinding met het Wireless netwerk van het hotel. Maar wanneer je het hotel verlaat en dus niet langer gebruik kunt maken van het WIFI netwerk, wordt er overgegaan naar een GPRS connectie zonder dat de bestaande connecties verloren gaan.

Om dit mogelijk te kunnen maken is er een RFC gemaakt waarin beschreven hoe dit gerealiseerd kan worden, RFC 2002 [19] deze RFC is geschreven voor IPv4. Voor IPv6 is er ook een Draft beschikbaar [20].

De probleem omschrijving voor mobile IP is vergelijkbaar met het multihoming probleem in IPv6. Het is daarom zeker eens interessant om te kijken hoe het probleem van mobile IPv6, wordt opgelost en in hoeverre dit ook toepasbaar is voor Site-multihoming.

Een mobile node wordt altijd voorzien van twee IPv6 adressen, het eerst adres is het "home adres" dit is een statisch adres en verandert niet. Het tweede IPv6 adres wordt het "care-of adres" genoemd. Dit adres is afhankelijk van het netwerk waarmee het op dat moment verbonden is. Om deze twee adressen goed te kunnen gebruiken wordt er gebruik gemaakt van een Home agent. Wanneer de mobiele node niet verbonden is met zijn thuis netwerk, verzameld de Home agent alle IP packets die naar het home adres worden verstuurd, de home agent forward deze pakketten vervolgens naar het care of address van de mobile node. Wat de Home agent dus eigenlijk doet is een tunneling onderhouden met het care of adres en de pakketten hierdoor versturen, zie figuur 43.



Iedere keer wanneer een mobile node van netwerk verandert, dient deze zijn nieuwe care of address door te geven aan de home agent. Dit Registration request bevat een aantal parameters en vlaggen welke iets zeggen over de tunnel en de registration Lifetime.

De combinatie van het home address, het care of address en de registration lifetime wordt een binding genoemd. Wanneer de mobile node van netwerk veranderd verstuurd deze een zogenaamde binding update waarin het nieuwe care of address wordt opgegeven.

Deze aanpak brengt nog al wat potentiële security problemen met zich mee. Het kan nu voorkomen dat een 3<sup>e</sup> host zich voor doet als de mobile node en een binding update verstuurd uit naam van de originele mobile node. Wanneer deze binding update zonder verdere controle wordt geaccepteerd door de home agent, zullen alle daarop volgende pakketten naar de 3<sup>e</sup> host worden verstuurd. Dit is uiteraard onacceptabel, er moet dus een security mechanisme worden gebruikt welke garandeert dat de binding update ook daadwerkelijk van de originele mobile node komt.

Hiervoor kunnen digitale handtekeningen worden gebruikt op basis van een 128 bits MD5 algoritme.

In zekere zin is mobile IPv6 ook gebaseerd op het principe van het scheiden van de locators en de identifier. Het care of address wordt gebruikt als locator en mag dus veranderen, het home address is de identifier en blijft de gehele sessie hetzelfde.

Hoewel Mobile IPv6 nog steeds verder ontwikkeld wordt heeft het zeker potentie, maar in hoeverre is dit toepasbaar op Multihomed sites?

Helaas is deze oplossing niet echt toepasbaar als oplossing voor het multihoming probleem.

Ten eerste is het niet bepaald schaalbaar, in het geval van een uitval van één van de upstream ISP's, zal de home agent een groot aantal tunnels dienen op te zetten, voor klein netwerk zal dit niet echt een groot probleem zijn, maar wanneer een site enkele honderden tot wellicht duizenden nodes heeft zal dit hoogst waarschijnlijk echt te veel worden.

Ten tweede is er het probleem van de bereikbaarheid van de home agent zelf. De home agent zal op een plaats in het netwerk geplaatst dienen te worden waar deze geen last heeft van de ISP outage, dit zal meestal niet het geval zijn. Daarnaast is er ook nog het probleem van de security, ook het systeem van digitale handtekeningen is niet geheel waterdicht te maken, tevens wordt alles zo een stuk ingewikkelder gemaakt.

### 13.4 Geografische adres toewijzing

Adressen toe laten wijzen aan de hand van geografische kenmerken, is een voorstel wat al enkele jaren speelt. Zelfs in de Ontwikkeling van IPv4 heeft men hieraan gedacht en ook nu zijn er weer diverse voorstellen voor geografische adres toewijzing. Het idee op zich is in principe een goed idee, maar toch kleven hier een aantal nadelen aan.

De ideeën zijn er op gebaseerd dat elke regio, bijvoorbeeld een wereld deel, een aparte prefix krijgt toegewezen. Een voorbeeld zou kunnen zijn dat Europa de prefix 2004::/16 en noord Amerika 2005::/16. De route tabellen verschillen dan per regio, in de Europese route tabellen staan dan een hoop specifieke prefixen voor Europese netwerken en telkens een ge-agregeerde route voor de overige wereld delen. Een voorbeeld van een dergelijke routetabel is te zien in figuur 44.

#### Voorbeeld Geografisch gebaseerde routing

Prefix	Nexthop
2004:43:d4a::/48	2004:07b8:200:A500:1200::b
2004:1234:567::/48	2004:07b8:200:A500:550::9
2004:678:43ac::/48	2004:07b8:200:A500:2340::ab
2004:a73d::/32	2004:07b8:200:A500:1200::6
2004:d48:2434::/48	2004:07b8:200:A500:1430::3
2004:893:dca:123::/64	2004:07b8:200:A500:1320::b
2004:abdd:334:23::/64	2004:07b8:200:A502:1232::13
2004:800:900:bab::/64	2004:07b8:200:A503:456::2
2005::/16	2004:07b8:2d0:Ad03:426::3
2006::/16	2004:074d:4d0:bd03:426::4
2007::/16	2004:0123:575:303:abc::42

Figuur 44

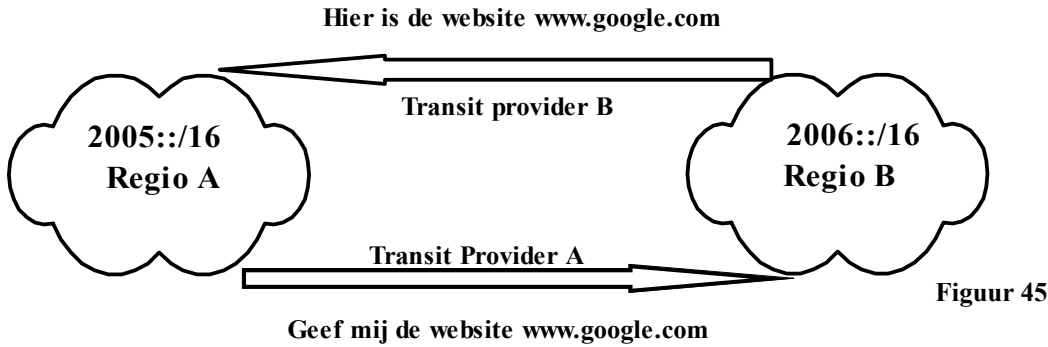
Het grote voordeel van de aanpak is dat het aantal entry's in de global routing table enorm verlaagd wordt. De Europese routers bevatten alleen specifieke prefixen van netwerken die zich in Europa bevinden. Voor de overige werelddelen is een ge-agregeerde route, over deze route zal al het verkeer voor bijvoorbeeld Amerika verlopen. Op deze manier hoeven de routers in Europa niet op de hoogte te zijn van meer specifieke routes in noord Amerika. Het voordeel daarvan is dat er een hoop ruimte vrij komt in de global routing table. Deze ruimte kan vervolgens gebruikt worden voor meer specifieke routes voor Europese netwerken.

In de huidige situatie is het niet toegestaan een /48 te adverteren, dit is gedaan omdat anders de route tabellen te vol zouden raken waardoor deze zouden "exploderen". Het verbod op het adverteren van een /48 prefix heeft ook als gevolg dat een hoop organisaties niet kunnen multihomen op de manier waarop dat in de IPv4 situatie werd gedaan (je prefix adverteren naar meerdere upstreams). Door deze geografische aanpak te gebruiken, zou deze manier wel weer gebruikt kunnen worden.

Er zijn echter ook een aantal problemen aan deze manier van adres toewijzing en routing.

Er kunnen een aantal Transit providers zijn, welke deze geo prefixen zullen adverteren aan hun klanten, een bedrijf kan bijvoorbeeld transit afnemen van Level3 (een transit provider), deze adverteert de geoprefixen (2004::/16, 2005::/16, etc...) door middel van BGP aan de klant. Al het verkeer wat de klant voor Amerika heeft zal dus verlopen via Level3. Echter dit is één-richtings verkeer, de vraag is hoe het verkeer terug komt vanuit Amerika. Stel nu dat de website in Amerika waarvan informatie is opgevraagd een andere transit provider dan Level3 heeft. Het verkeer zal dan

via deze alternatieve transit provider terug komen. Deze transit provider levert nu transit aan iemand die daar niet voor betaalt. Waar het dus op neer komt is dat je transit betaalt voor het upstream verkeer en al het terug komende verkeer zal moeten worden betaald door de zendende partij, schematisch is dit nog eens weergegeven in figuur 45. De vraag is of dit gaat werken en of providers hiermee akkoord zullen gaan.



Wanneer een organisatie een adres reeks zal aanvragen bij bijvoorbeeld Ripe zal deze een reeks krijgen welke binnen de geografische prefix van Europa vallen. Maar het kan natuurlijk heel goed zijn dat deze organisatie een grote multinational is en kantoren over de hele wereld heeft. Een kantoor in Japan zal dan gebruik maken van IP adressen welke uit de Europese reeks komen, want deze reeks heeft het bedrijf toegewezen gekregen. Wanneer de Japanse werknemers van dit Europese bedrijf een Japanse website zullen opvragen, zal de routing verre van optimaal verlopen. Al het verkeer zal dan via Europa verlopen, omdat dit een Europese prefix is. Het zelfde voorbeeld gaat op voor een aantal grote Amerikaanse ISP zoals AOL (america online). Deze ISP hebben inbel klanten over de hele wereld, de klanten zullen altijd een IP adres krijgen uit de Amerikaanse reeks, ook al zitten ze in Australië.

De manier waarop het Internet nu is opgebouwd is erg robuust, als er ergens een fiber kapot gaat wordt al het verkeer vrij snel via een andere route gerouteerd. Als er om wat voor reden dan toch een probleem is ergens, is er een klein gedeelte van het Internet niet te bereiken. De kans bestaat dat door gebruik te maken van een aantal ge-agreerde routes voor verschillende continenten te gebruiken, er tijdens een probleem bij een transit providers, één of meerdere continenten helemaal niet meer bereikbaar zijn.

Tevens is er het probleem van her nummering, stel nu dat een organisatie verhuist van locatie. Dit zou betekenen dat het hele bedrijfsnetwerk hernummerd dient te worden, dit is dikwijls een klus waar bedrijven niet op zitten te wachten. Hoewel er in de bovenstaande voorbeelden gebruik is gemaakt op adres toewijzing per wereld deel, bestaan er ook voorstellen die dit doen op basis van het land of regio waarin een site zich bevindt. Op deze manier worden de genoemde problemen alleen maar groter.

### 13.5 Provider Independent Addressing gebaseerd op AS nummer

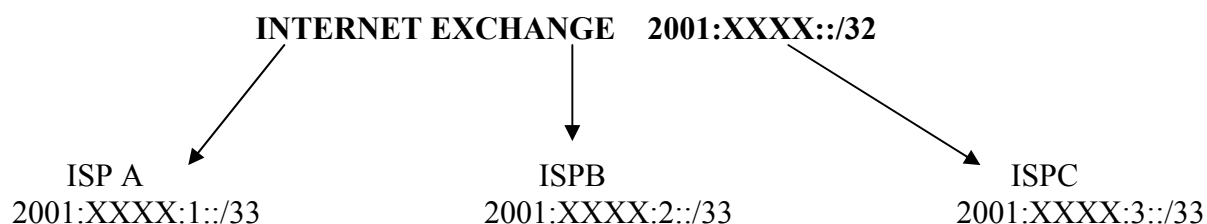
De Internet draft [21], beschrijft een manier waarop sites welke nu een eigen AS nummer hebben een stuk Provider Independent (PI) adres ruimte kunnen krijgen. Het idee is er op gebaseerd dat er een aparte prefix wordt gedefinieerd voor deze adressen. Bijvoorbeeld de prefix 2000::/16 gevolgd door het AS nummer. Een voorbeeld van zo'n prefix voor AS1200 is: 2000:4B0::/32 (1200 = 0x4B0).

Het AS nummer is een 32 bit getal waarvan er nu 16 worden gebruikt. Dat geeft 65536 AS nummers, waarvan een deel private AS space is. Er kunnen dus maximaal 65000 van de prefixen in de route tabel terecht komen. Op dit moment zijn er ongeveer 20.000 AS nummers toegewezen, waarvan er slechts 12.000 in de route tabel te zien zijn. Dus reëel gezien zou dit tot 12.000 extra prefixen en de IPv6 route tabel leiden. Dit is 10% van de huidige grote (120.000 prefixen) van de IPv4 route tabel.

Duidelijk mag zijn, dat dit geen definitieve oplossing is, maar het geeft in ieder geval de organisaties welke nu multihomed zijn de mogelijkheid dit te blijven doen na de migratie naar IPv6. De impact op de global routing table is mijn inziens acceptabel. Zeker wanneer gekeken wordt naar het feit dat op deze manier vrijwel alle organisaties welke nu multihomed zijn (want om te kunnen multihomen is een AS nummer nodig), opnieuw voorzien kunnen worden van PI adres ruimte. Tegenstanders van dit idee, zijn bang dat wanneer dit idee ingevoerd wordt er een enorme vraag naar AS nummers zal komen, met als gevolg dat deze opraken.

### 13.6 Exchange based aggregation

De manier waarop op dit moment wordt geaggregeerd is provider based. In feite betekent dit, dat binnen een provider een aantal specifieke prefixen bekend zijn, maar dat er slechts één aggregated prefix wordt geadverteerd naar peers toe. Een mogelijkheid is om de aggregatie te verleggen van de providers naar een Internet Exchange. In deze situatie krijgt een exchange een bepaalde prefix toegewezen van bijvoorbeeld RIPE. Alle klanten (ISP's) die zijn aangesloten op de Internet Exchange krijgen hun IPspace uit de reeks van de Internet Exchange. Bijvoorbeeld



Het voordeel van het aggregatie punt verleggen naar de Internet Exchange is dat Sites nu multihomed kunnen zijn tussen de verschillende ISP's die op de Internet Exchange zijn aangesloten. Zo kan een site aangesloten zijn op zowel ISPA als ISPB en door middel van BGP zijn /48 Advertiseren. Het probleem van deze vorm van aggregatie is de upstream. Wie verzorgt de transit naar de andere Exchanges. Het idee is, dat de Internet Exchange in deze situatie ook verantwoordelijk wordt voor de upstream. In de praktijk kan het natuurlijk zo zijn dat een Internet Exchange hier niet op zit te wachten.

### **13.7 Samenwerking tussen verschillende ISP's.**

Een ander mogelijkheid zou een samenwerkingsverband kunnen zijn tussen verschillende ISP's. Twee of meer Internet Service Providers, zouden met z'n tweeën een multihoming abonnement kunnen aanbieden aan klanten.

Speciaal voor dit doel zou een /32 bij RIPE kunnen worden aangevraagd. Deze prefix dienen beiden providers onafhankelijk van elkaar te adverteren in de internet route tabel. Wat er in resulteert dat de prefixen via beide ISP te bereiken zijn. Een klant zou een zogenaamd "Multihoming abonnement" kunnen afnemen bij deze ISP's. De klant krijgt vervolgens een /48 uit de multihoming IP-reeks van de twee ISP's. De klant dient dan ook naar beide ISP's een BGP sessie op te zetten. Zodoende kan een Site toch multihomed zijn met een /48, zonder de aggregatie te breken.

De vraag is natuurlijk, of ISP's dit willen gaan aanbieden, want ze dienen hiervoor wel samen te werken met hun concurrent. Een mogelijk struikelblok is de vraag, "wie wordt er als primaire transit provider gebruikt?". Wanneer ISPA in een stabiele situatie het meeste verkeer krijgt te verwerken, heeft deze uiteraard ook recht op meer vergoeding. Over dit soort zaken zullen de ISP's, eventueel in samenwerking met de klanten, goede afspraken te maken.

Het is het een uniek product welke ISP zouden kunnen aanbieden. De toekomst zal leren of ISP's hier wat in zien.

## 14 Multihoming bij AMS-IX

In het vorige hoofdstuk zijn een aantal potentiële oplossingen aangedragen, dit zijn oplossingen die op dit moment actief besproken worden op de mailinglijsten en bij de IETF.

Helaas is geen van deze oplossing op dit moment toepasbaar, tot nu toe zijn deze oplossingen dus nog toekomst muziek. De oplossingen die al wel toepasbaar zijn, zullen nooit goed werken omdat bijv de SCTP door nog geen één Operating System geïmplementeerd is.

Om toch tot een (gedeeltelijke) oplossing te komen, is het volgende bij de Amsterdam Internet Exchange geïmplementeerd.

Gekozen is voor een host based oplossing, dit ziet er als volgt uit:

hosts zijn voorzien van meerdere adressen (PA), wat inhoudt dat de aggregatie niet gebroken hoeft te worden, dit is een groot voordeel. Er wordt nu dus gebruik gemaakt van meerdere ISP's. Een host beschikt nu over meerdere adressen, hogere lagen zoals applicaties, moeten zelf keuzes maken welk adres ze wanneer als source adres gebruiken. Een groot voordeel van deze oplossing is dat deze op korte termijn geïmplementeerd kan worden. Echter deze oplossingen brengt een aantal nog op te lossen problemen met zich mee, deze problemen zullen in de volgende paragrafen worden besproken, tevens worden eventuele oplossingen daarvoor besproken.

### 14.1 Ingress filtering

Een groot probleem bij deze oplossing, het gebruik van meerdere PA adressen, is dat veel ISP's ingress filtering geïmplementeerd hebben op hun routers. Ingress filtering houdt in, dat alle pakketten die het ISP netwerk verlaten gecontroleerd worden op hun source adres zie ook [22]. Als het goed is, zijn dit altijd adressen uit de reeks van de ISP, wanneer dit niet het geval is, wordt het pakket gedropped.

Ingress filters wordt door ISP's gebruikt om IP spoofing tegen te gaan. Onder IP spoofing wordt verstaan, het verzenden van IP pakketten met een vals source adres. IP spoofing wordt veel gebruikt bij Denial of Service (DOS) attacks, waarbij door de aanvaller een grote hoeveelheid TCP/SYN pakketten naar bijvoorbeeld een webserver verstuurd met een vals source adres. De webserver zal hier op reageren met TCP/SYN-ACK pakketten deze worden echter verstuurd naar een host die hier nooit om gevraagd heeft of wellicht is de betreffende hosts niet eens te benaderen. Als de vloed van TCP/SYN pakketten maar snel genoeg gaat, zal de webserver binnen afzienbare tijd "out of resources" zijn, met als gevolg dat andere cliënten niet meer bediend kunnen worden. Voor sommige systemen kan dit zelfs een crash van de webserver tot gevolg hebben.

Normaal gesproken is de implementatie van Ingress filtering door ISP's dus een goede zaak, echter in het geval van multihoming veroorzaakt dit een probleem. Uitgaande van de situatie geschetst in figuur 46.

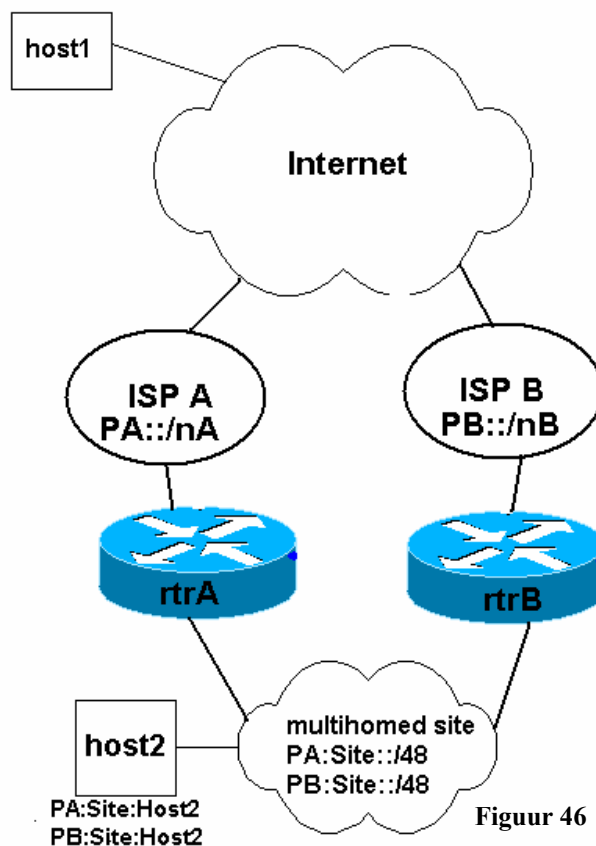
Stel dat host2 het source-adres PA:Site:host2 kiest voor de communicatie met host1. De default route van host2 wijst naar rtrB, wanneer rtrB het pakket forward naar ISPB, zal deze onderweg worden gedropped als ISPB ingress filtering toepast omdat het source adres niet in de reeks van ISPB ligt.

Het betreffende pakket zal op deze manier nooit bij host1 aankomen. Een mogelijke oplossing voor dit probleem is de implementatie van source-based routing. Dit is een techniek die aan de hand van het source adres de next hop bepaald.

In deze situatie zouden dus zowel rtrA als rtrB moeten kijken naar het source adres van alle pakketten die binnenkomen op hun interne interface. Wanneer het source adres niet bij de ISP hoort welke aan de betreffende router is verbonden, moet de router het pakket doorsturen naar de andere router. In het geval dat slechts één exit router wordt gebruikt, kan mbv. source routing de juiste exit interface worden gekozen.

Het implementeren van source based routing op de exit routers zou het probleem wat veroorzaakt wordt door het gebruik van ingress filtering kunnen oplossen. Een alternatieve oplossing is de IPv6 stack van hosts met wat meer intelligentie uit te rusten. Deze uitbreiding van de IPv6 stack is dan verantwoordelijk voor de keuze van de default route welke afhankelijk is van het gekozen source adres. Bij source adres PA:site:host2 hoort dan als default gateway rtrA, voor het source adres PB:site:host2 dient rtrB als default router gekozen te worden. Per adres moet een host dus een default gateway weten, de default gateways zouden theoretisch dezelfde router kunnen zijn of zelfs het zelfde adres kunnen zijn, in het geval een upstream ISP geen ingress filtering toepast op zijn netwerk.

Een laatste mogelijkheid is om afspraken te maken over Ingress filtering met de upstream ISP's. Een klant zou met zijn ISP kunnen afspreken dat de ISP voor hem meerdere IPv6 prefixen toe staat en dus minder streng is met ingress filtering. Hiervoor dient wel een bepaalde vertrouwensband te bestaan tussen de ISP en de klant, dit zal niet altijd het geval zijn.





## 14.2 Source address selection

Een multihomed host heeft de beschikking over meerdere adressen, wanneer de host een verbinding wil opzetten met bijv een webserver ergens op het Internet, zal deze een keuze moeten maken uit een aantal potentiële source adressen. RFC 3484 [23] beschrijft hoe address selection in IPv6 zou moeten plaats vinden en beschrijft een aantal algoritme voor adres selectie.

Een voorbeeld van adres selection zoals deze in RFC 3484 [23] beschreven is:

Source addresses: 2001:aaaa:aaaa::a of 2007:0:aaaa::a of fe80::a

Destination address: 2001:cccc:cccc::c

Volgens deze RFC zou de host het adres 2001:aaaa:aaaa::a als source address moeten kiezen. Dit omdat deze dit source address het meest overeen komt met het destination address (longest matching prefix) en van het zelfde type adres is. Het source adres fe80::a is een link local address, terwijl het destination address een global aggregatable unicast address is, dit zijn dus verschillende type adressen.

In de praktijk blijkt dat het address selection mechanisme zoals dit in de RFC is beschreven, door lang niet elke IPv6 stack wordt toegepast. Een mooi voorbeeld hiervan is de standaard IPv6 stack van Linux, deze lijkt helemaal niet aan source address selection te doen, maar gebruikt altijd het zelfde source address voor communicatie. Het gekozen source address in Linux is altijd het laatst geleerde adres. Dit kan een handmatige geconfigureerd adres zijn of een adres geleerd via een router advertisement. Systemen met OpenBSD, FreeBSD en WindowsXP lijken een stuk beter met source address selection om te gaan. In de onderstaande testen wordt dit bewezen:

In het eerste voorbeeld wordt een traceroute gedaan vanaf een host met FreeBSD als operating system, de host beschikt over de volgende adressen:

```
ifconfig x10 inet6
x10: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> mtu 1500
    inet6 fe80::260:97ff:fe96:69a7%x10 prefixlen 64 scopeid 0x1
    inet6 3ffe:8114:2000:1394:260:97ff:fe96:69a7 prefixlen 64 autoconf
    inet6 2001:888:1357:0:260:97ff:fe96:69a7 prefixlen 64 autoconf
```

De host is geconfigureerd met 3 IPv6 adressen, één link local adres fe80::260:97ff:fe96:69a, En twee global aggregatable unicast adressen, namelijk 2001:888:1357:0:260:97ff:fe96:69a7 en het sixbone adres 3ffe:8114:2000:1394:260:97ff:fe96:69a7

Als eerste wordt er een traceroute6 naar de host [www.ams-ix.net](http://www.ams-ix.net) gedaan:

```
traceroute6 to www.ams-ix.net (2001:610:140:604::2) from
2001:888:1357:0:260:97ff:fe96:69a7, 30 hops max, 12 byte packets
 1  2001:888:1357::1  1.061 ms  0.817 ms  0.789 ms
 2  xs4all855.ipv6.xs4all.nl  16.066 ms  17.118 ms  20.562 ms
 3  26.ge-0-2-0.xr1.pbw.xs4all.net  7.741 ms  7.335 ms  7.112 ms
 4  0.ge-0-3-0.xr1.sara.xs4all.net  7.022 ms  7.685 ms  8.134 ms
 5  rtr1.ipv6.ams-ix.net  8.624 ms  7.435 ms  7.913 ms
 6  2001:610:140:604::2  7.856 ms  7.924 ms  8.917 ms
```

Zoals te zien is in het bovenstaande voorbeeld is het destination address 2001:610:140:604::2, het gekozen source address is 2001:888:1357:0:260:97ff:fe96:69a7, wat klopt volgens de regel van “the longest matching prefix”.

Het zelfde experiment wordt vervolgens gedaan naar een destination address uit de sixbone reeks:

```
traceroute6 to ipv6.klingon.nl (3ffe:8114:1000::50f) from
3ffe:8114:2000:1394:260:97ff:fe96:69a7, 30 hops max, 12 byte packets
 1 3ffe:8114:2000:1394::1 1.115 ms 0.809 ms 0.793 ms
 2 2001:888:0:3::4242 17.161 ms 16.973 ms 16.553 ms
 3 2001:888:0:3::1 7.385 ms 7.442 ms 7.726 ms
 4 2001:888:2:1::1 7.476 ms 7.516 ms 7.029 ms
 5 2001:6e0::2 7.426 ms 6.961 ms 7.497 ms
 6 3ffe:8114:1000::4ac 7.272 ms 7.11 ms 7.207 ms
 7 3ffe:8114:1000::50f 19.984 ms 19.002 ms 25.697 ms
```

Zoals verwacht wordt hierbij het source address 3ffe:8114:2000:1394:260:97ff:fe96:69a7 uit de sixbone reeks gekozen.

Hieruit kan geconcludeerd worden dat FreeBSD en Openbsd zich aan de standaard houden, ook Solaris8 en WindowsXP lijken zich aan de regel van “the longest matching prefix” te houden.

Het is belangrijk te begrijpen welk source address een host kiest wanneer deze de keuze heeft uit verschillende adressen. In de volgende situatie beschikt host2 over 2 adressen. PA:Site:Host2 en PB:Site:Host2, er vanuit gaande dat het “ingress filtering probleem” nu niet van toepassing is, zal de host2 een source address kiezen welke het meest overeenkomt met het destination address. Maar wat gebeurt er wanneer ISPA of de link naar ISPA problemen ondervindt met als gevolg dat deze niet langer bruikbaar is. Dit betekent dat het adres PA:Site:Host2 niet langer gekozen moet worden als source adres. Dit adres is namelijk niet meer te bereiken. Er moet dus een methode ontwikkeld worden waardoor host2 het adres wat bij ISPA hoort (PA:Site:Host2) niet langer als source address kiest.

Wanneer dit probleem geprojecteerd wordt op de situatie bij AMS-IX, is te zien dat in de kantoor omgeving alle hosts voorzien zijn van een “auto-configured address”. De hosts hebben zichzelf dus geconfigureerd nadat ze Router advertisement berichten hebben ontvangen van de kantoor router. Hierin maakt de kantoor router bekend welke prefixen geldig zijn op het kantoor netwerk.

Aan de hand van deze informatie, hebben alle hosts op het kantoor netwerk zich zelf voorzien van een “auto-configured address”, deze adressen zijn voorzien van een bepaalde life-time. Deze lifetime is te bekijken, door de interface informatie op te vragen, in Linux ziet dit er als volgt uit:

```
ip -6 addr show dev eth0
eth0: <BROADCAST,MULTICAST,UP> mtu 1500 qdisc pfifo_fast qlen 100
inet6 2001:888:1357:0:202:44ff:fe23:453c/64 scope global dynamic
      valid_lft 566sec preferred_lft 266sec
inet6 3ffe:8114:2000:1394:202:44ff:fe23:453c/64 scope global dynamic
      valid_lft 566sec preferred_lft 266sec
inet6 fe80::202:44ff:fe23:453c/10 scope link
```

In het voorbeeld op de vorige pagina is te zien dat deze host is voorzien van twee autoconfigured adressen, in de regel daaronder is te zien wat de bijbehorende valid lifetime en preferred lifetime is. De host zal tijdens communicatie met een andere IPv6 host een keuze maken uit één van de 2 bovenstaande adressen, ook al is één van de adressen niet meer te routeren.

Een host moet dus weten dat tijdens een upstream outage, hij de bijbehorende adressen niet meer moet gebruiken. Nu is het natuurlijk mogelijk de adressen welke niet meer gebruikt kunnen worden handmatig te verwijderen. Echter dit is niet bepaald een schaalbare oplossing, wat een veel mooiere oplossing is, is de router de betreffende prefix te laten adverteren met een lifetime van 0sec.

Hierdoor zal het autoconfigured adres de status deprecated krijgen (zie onderstaande voorbeeld).

```
eth0: <BROADCAST,MULTICAST,UP> mtu 1500 qdisc pfifo_fast qlen 100
  inet6 2001:888:1357:0:202:44ff:fe23:453c/64 scope global dynamic
        valid_lft 581sec preferred_lft 281sec
  inet6 3ffe:8114:2000:1394:202:44ff:fe23:453c/64 scope global deprecated dynamic
        valid_lft 1sec preferred_lft -19sec
  inet6 fe80::202:44ff:fe23:453c/10 scope link
```

De protocol specificaties beschrijven dat een adres niet langer als source address gekozen mag worden als deze de status deprecated heeft. Voor nieuwe connecties zal vanaf dat moment altijd het andere adres gekozen worden.

Het is dus zaak een outage van één van upstream ISP te detecteren en de bijbehorende prefix door middel van router advertisements terug te trekken door met een lifetime van nul seconde te adverteren.

### 14.3 Destination Address selection

Net zoals een “normaal” werkstation in een multihomed netwerk meerdere IPv6 adressen heeft, zullen de verschillende servers binnen het multihomed netwerk ook voorzien worden van verschillende adressen. Normaal gesproken zullen de servers met beide adressen PA:Site:server1 en PB:Site:server1 in de nameserver vermeld staan. Een voorbeeld hiervan is de webserver van de Amsterdam Internet Exchange:

```
host -t aaaa www.ams-ix.net
www.ams-ix.net      AAAA  2001:610:140:604:0:0:0:2
www.ams-ix.net      AAAA  2001:7B8:200:604:0:0:0:2
```

Zoals te zien is in het bovenstaande voorbeeld, staat www.ams-ix.net in de nameserver met twee adressen vermeld. In een stabiele situatie (zowel ISPA als ISPB zijn bereikbaar) is dit geen probleem. In tegendeel zelfs, door meerdere DNS entry's wordt nu door middel van round robin, load balancing toegepast waarbij het verkeer over de verschillende ISP verdeeld wordt. De ene keer zal een cliënt het adres PA:Site:server1 gebruiken om de webserver te benaderen terwijl de volgende keer deze cliënt het adres PB:Site:server1 gebruikt.

Hier dient zich een potentieel probleem aan, wanneer één van de ISP's of de link naar de ISP's problemen ondervindt, is één van de adressen niet langer bereikbaar. Omdat een server met alle twee de adressen in de nameserver staat, zal theoretisch gezien 50 % van de verzoeken naar het adres PA:Site:server1 gaan en 50% naar PB:Site:server1. Wanneer het adres PA:Site:server1 onbereikbaar wordt, zal dit tot gevolg hebben dat 50% van de verzoeken naar de server niet aankomen.

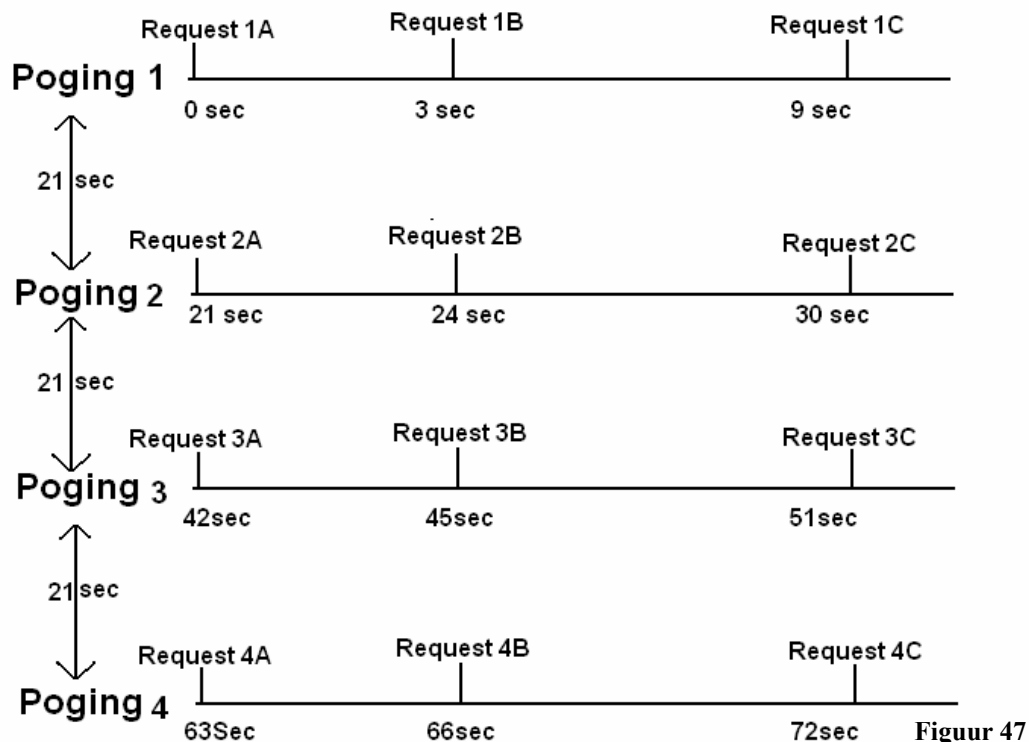
Er is een klein experiment gedaan om te onderzoeken hoe een willekeurige cliënt hiermee om gaat. Het volgende is de situatie:

- Een webserver is met meerdere AAAA records bekend in DNS server  
 In het geval van dit experiment waren dit 6 AAAA records, namelijk  
 PA:Site1:server1  
 PB:Site1:server1  
 PA:Site2:server1  
 PB:Site2:server1  
 PA:Site3:server1  
 PC:Site3:server1

Alleen op het adres PB:Site1:server1 luistert een webserver

- Met een IPv6 enabled browser (Internet Explorer 6) werd de website opgevraagd.
- Tijdens dit proces is het netwerk verkeer gemonitord

**Bevindingen:** Als eerste wordt er een query naar de nameserver gestuurd met het verzoek bij de naam (URL) het IPv6 adres te zoeken (aaaa query). De nameserver reageert op dit verzoek en komt met meerdere IPv6 adressen (zes in het geval van het experiment). Precies in dezelfde volgorde als waarin de adressen ontvangen worden, wordt geprobeerd een http connectie op te zetten. Als eerste wordt er een TCP ack verstuurd naar het adres PA:Site1:server1, vrijwel meteen daarna komt van een upstream router, een ICMP address unreachable bericht terug. Wat inhoudt dat het adres niet te bereiken is, dit kan wel kloppen omdat 5 van de 6 adressen niet gebruikt worden. 3 sec later verstuurd de cliënt nogmaals een TCP/SYN naar het adres PA:Site1:server1, zoals te verwachten is volgt ook op dit pakket onmiddellijk een ICMP address unreachable bericht. De browser wacht vervolgens 6 sec en doet dan nog een laatste poging voor dit adres. De cliënt heeft nu tot 3 keer toe geprobeerd verbinding te krijgen met poort 80 op het adres PA:Site1:server1, dit is tot drie keer toe niet gelukt. De cliënt last nu een time-out in van 12 seconden en probeert het tweede adres uit het rijtje. Zo zal Internet Explorer alle adressen proberen, schematisch ziet dit er als in figuur 47.



Figuur 47

Voor dat Internet Explorer een tweede adres zal gaan proberen is al 21 seconde verstreken, als het juiste adres het vierde adres is, betekend dit dat de cliënt een minuut moet wachten totdat de betreffende pagina geladen is! Dit scenario is uiteraard onacceptabel, waarschijnlijk zullen veel mensen niet eens de 21 seconde voor dat het tweede adres geprobeerd wordt afwachten.

Uit het voorgaande kan geconcludeerd worden, dat het niet is toegestaan dat een DNS server adressen adverteert welke niet langer bereikbaar zijn. Wanneer er een verbinding probleem wordt geconstateerd bij één van de providers dient de zone file van het betreffende domein te worden aangepast, zodat het adres niet langer wordt geadverteerd. Nu zullen alle verzoeken direct naar het juiste adres gaan.

Een alternatieve oplossing is een verandering in de applicatie, deze zou direct nadat er icmp bericht "port/address unreachable" is ontvangen en er is een ander adres beschikbaar meteen het alternatieve adres moeten proberen. Zodoende kunnen binnen een seconde meerdere adressen geprobeerd worden.

### 14.4 Established connections.

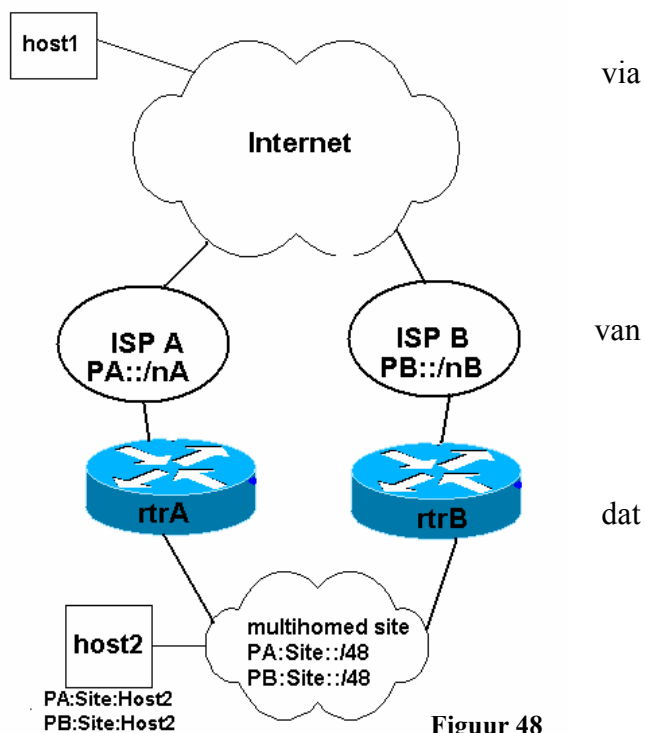
Wanneer een cliënt een verbinding heeft opgebouwd naar een server (host1) met het source adres PA:Site:Host2 en problemen treden op bij ISPA of de link naar ISPA (figuur 48), heeft dit tot gevolg dat de verbinding niet langer juist zal functioneren.

Ondanks dat er een redundant pad aanwezig is rtrb, zal de verbinding verloren gaan. Dit omdat host1, host2 met het adres PA:Site:Host2, via ISPA zal proberen te bereiken. Dit probleem is niet op te lossen door gebruik te maken van meerdere adressen. Host1 weet namelijk niet beter dan dat het PA:Site:Host2 kan bereiken over het netwerk ISPA.

Dit alles heeft dus tot gevolg dat verbindingen niet bestand zijn tegen een ISP outage of een link failure naar de ISP. Het gevolg hiervan is de verbinding opnieuw opgezet dient te worden, met het 2<sup>o</sup> adres, in dit geval het adres van ISPB PB:Site:Host2.

Nu is de vraag hoe ernstig dit is, wat is de impact hiervan op verschillende applicaties?

Hierin kan een tweedeling gemaakt worden tussen twee soorten verbindingen, namelijk verbindingen van korte en lange duur. Een voorbeeld van een verbinding van lange duur is een SSH connectie, deze kan in principe enkele uren tot zelfs dagen zoniet weken blijven bestaan. Een voorbeeld van een korte verbinding is een http verbinding. Wanneer een webpagina wordt opgevraagd zal voor iedere pagina een nieuwe verbinding worden opgezet. Tijdens het "surfen" kan als source adres de ene keer adres A worden gebruikt en de volgende keer adres B, de impact van het uitvallen van een ISP blijft nu beperkt tot niet kunnen laden van de betreffende pagina. Wanneer dit adres vervolgens wordt terug getrokken en de eindgebruiker zal de pagina vervolgens "refreshen"/opnieuw opvragen, dan zal de cliënt de pagina opvragen met het alternatieve adres en nu zal de eindgebruiker de pagina te zien krijgen. Het zelfde geldt voor het ophalen van email met



Figuur 48

bijvoorbeeld pop(s) of imap(s). De gemiddelde gebruiker met een continue Internet verbinding, zal zijn email om de 5minuten ophalen bij de mailserver. Als er een ISP outage plaats vindt, zal de gebruiker dit merken door een time out tijdens zijn pop(s) imap(s) sessie, na deze time-out zal het vanzelf goed gaan omdat de applicatie het andere IPv6 adres kiest als source adres.

Voor connecties van korte duur en welke steeds opnieuw worden established, zullen clients dus relatief weinig problemen ondervinden tijdens een probleem bij of naar één van de upstream ISP's. Echter, voor verbindingen welke van lange duur zijn en niet steeds opnieuw worden opgezet, is dit geen oplossing.

### **14.5 Multihoming script**

Zoals in de voorgaande paragrafen besproken is, zijn er aantal problemen met de tot nu toe enige manier om IPv6 multihoming te implementeren (host based). De besproken onderwerpen zijn:

1. Ingress Filtering
2. Source address selection
3. Destination address selection
4. Established connections.

Om een host based oplossing te realiseren voor de situatie zoals deze bij AMS-IX is, moeten een tweetal zaken worden aangepakt. Namelijk punt 2 source address selection en punt 3 destination address selection.

Punt 1, het ingress filterings probleem is in de AMS-IX situatie geen issue. De 2 providers waarvan AMS-IX transit krijgt, lijken niet aan ingress filtering te doen. Bovendien adverteert AMS-IX (tegen de regels van een Best Current Practice document in) beide /48 prefixen naar haar BGP peers. Dit betekent dat een aantal bestemmingen niet via de AMS-IX IPv6 transit providers BIT en Surfnets zullen gaan. Bovendien krijgt AMS-IX van een aantal partijen de hele IPv6 route tabel, wat inhoudt dat er transit wordt verkregen. Voor sites die niet over deze luxe beschikken, is source-based routing een oplossing.

Punt 4, Established connections heeft betrekking op de reeds opgezette verbindingen. Hiervoor is op dit moment geen oplossing mogelijk. Toch is dit mijns inziens geen onoverkomelijk probleem. De meeste sessie van langere duur, welke vanaf het AMS-IX netwerk worden opgezet zullen binnen het eigen netwerk blijven. Een voorbeeld van sessies van langere duur zijn SSH, Telnet en FTP, deze sessies zullen ook binnen AMS-IX regelmatig worden gebruikt. De AMS-IX switches en routers worden bijvoorbeeld door middel van Telnet en SSH beheert. Deze sessies blijven dus binnen het eigen netwerk, als er een probleem is met de verbinding naar of met de ISP zelf, heeft dit geen invloed op de bereikbaarheid van deze PA IPv6 adressen vanaf binnen het AMS-IX netwerk. Het grootste deel van de dagelijkse werkzaamheden zal door een ISP outage dus niet worden onderbroken.

Voor de langdurige sessie naar hosts die buiten het eigen netwerk staan is er geen oplossing. Het zelfde geldt voor hosts van buiten het AMS-IX netwerk naar hosts binnen het AMS-IX netwerk. Bedenk hierbij wel dat dit ook het geval is, bij de tweede multihoming oplossing zoals deze in IPv4 wordt geïmplementeerd. De methode waarbij gebruik gemaakt wordt van 2 ISP's waaraan het interne netwerk via NAT gekoppeld is met het Internet. Ook dan zullen de verbindingen verloren gaan bij een ISP outage. Dus wat dat betreft zal een multihomed site niets winnen of verliezen.

Voor de overige punten, source en destination address selection is een Unix Bash script, gemaakt welke er voor zorgt dat deze selectie juist verloopt. Het script is te zien in Bijlage III.

Wat het script doet is het volgende: met een bepaald interval (default is 5 seconde), wordt de bereikbaarheid van het netwerk via de verschillende providers getest. Dit wordt gedaan door een IPv6 host op Internet te pingen (ICMPv6), dit wordt steeds gedaan met 2 verschillende source adressen. Tijdens de eerste test wordt een adres uit de Surfnets reeks als source adres gebruikt. Als de echo reply aankomt, betekent dit, dat de verbinding via Surfnets nog goed functioneert.

Het zelfde wordt gedaan met een source adres uit de IP reeks die aan ons is toegewezen door BIT. Wanneer er geen echo reply terug komt betekent het, dat de IP reeks welke op dat moment getest wordt niet langer bereikbaar is. Wanneer dit is geconstateerd dient er wat te gebeuren, zodat het source en destination address selection proces de juiste IPv6 adressen kiezen.

Om er voor te zorgen dat hosts op het kantoor netwerk niet langer het IP adres kiezen welke niet te gebruiken is, moet deze de status “deprecated” krijgen. Dit wordt gerealiseerd met behulp van de Router Advertisements welke worden verzonden door de kantoor router. Deze router adverteert normaal gesproken twee prefixen, één uit de surfnets reeks en één uit de bit reeks. De IP reeks welke nu niet langer te gebruiken is wordt geadverteerd met een lifetime (preferred en valid) van 0 seconde. Dit heeft tot gevolg dat de IPv6 stack van de hosts op het office netwerk deze adressen de status deprecated geven en dit adres niet langer als source adres voor nieuwe uitgaande verbindingen zal kiezen. Zodoende wordt het source adres selection proces beïnvloed, met als gevolg dat nu altijd een source IP adres gekozen welke op dat moment routable is.

Om ervoor te zorgen dat de router in een dergelijk geval de betreffende prefix gaat adverteren met 0 seconde, zal het script inloggen op de router en de configuratie voor deze prefix wijzigen.

Destination address selection, zal vooral van belang zijn wanneer clients van buiten het AMS-IX netwerk verbinding proberen te krijgen met één van de AMS-IX servers. Een voorbeeld hiervan is een cliënt ergens op Internet, welke de website [www.ams-ix.net](http://www.ams-ix.net) probeert op te vragen. Als eerste zal de cliënt een DNS query doen voor [www.ams-ix.net](http://www.ams-ix.net). De cliënt zal hierop normaal gesproken twee antwoorden terug krijgen van de DNS server, één Surfnets adres en één BIT adres.

In een stabiele situatie is elke server voorzien van 2 DNS entry's, in het geval een probleem met of BIT of Surfnets, dient de DNS server niet langer 2 adressen te adverteren, maar alleen diegene die op dat moment nog bereikbaar is. Zodoende wordt altijd een geantwoord met een adres welke op dat moment te bereiken is en het destination address selection proces in goede banen geleid.

Wat het script dus doet, is in het geval van een ISP failure, de entry's in de DNS server aanpassen zodat alleen de bereikbare adressen nog in de DNS server bekend zijn. Dit wordt gedaan door de zone file van [ams-ix.net](http://ams-ix.net) aan te passen en de DNS servers te reloaden.

Een voorwaarde is wel, dat de verschillende AAAA entry's, een korte Time To Live (TTL) hebben. Bijna iedere DNS server op Internet cached zijn query's, wanneer een verzoek voor een bepaald adres éénmaal is gedaan zal deze een bepaalde tijd in de cache van de nameserver blijven bestaan. Zodoende hoeft niet iedere keer opnieuw een query naar de verantwoordelijke DNS server te worden gestuurd. Dit bevordert de efficiency van het DNS proces. Hoe lang een bepaalde DNS entry mag worden gecached wordt bepaald door de TTL van de entry. Wanneer de TTL van een gecachte entry is verlopen, dient de DNS server de query opnieuw te doen. In dit geval is gekozen voor een TTL van 1 minuut. Zou deze tijd langer worden gekozen, dan kan dit tot gevolg hebben dat het destination address selection proces het verkeerde adres terug krijgt van een DNS server, dit als gevolg van een inmiddels ongeldige cache entry.

De impact van uplink outage, op de zgn. korte sessies is ongeveer gelijk in de IPv4 en de IPv6 situatie. In IPv6 zal het een kwestie van seconden zijn voor dat een cliënt weet dat hij een bepaald adres niet langer als source adres mag gebruiken. In de IPv4 situatie moet er gewacht worden tot de routerings protocollen het probleem geconstateerd hebben en tot het opnieuw geconvergeerd heeft. Deze tijden, convergence time vs. deprecation time zullen elkaar niet veel ontlopen, uiteraard is dit afhankelijk van het routerings protocol en de grootte van het netwerk.

## 15 Conclusie

Na 4 maanden full time met allerhande IPv6 zaken bezig te zijn geweest, heb ik een degelijke kennis van de IPv6 protocol specificaties gekregen. Deze kennis heb ik getracht op te schrijven in het eerste gedeelte van deze scriptie. In het tweede gedeelte is beschreven hoe de integratie van IPv6 in het netwerk van de Amsterdam Internet Exchange is aangepakt en verlopen. Vol goede moed is destijds gestart met de upgrade van diverse services. Hoewel het merendeel in één keer goed is gegaan, waren er ook een aantal upgrades die niet geheel zonder problemen verliepen. Het merendeel van deze problemen is toe te schrijven op de IPv6 implementaties van de verschillende Operating systems. Onder andere op Solaris en Linux maar ook op de Mac ben ik tegen een aantal bugs aangelopen. Hieruit blijkt eens te meer dat IPv6 nog niet helemaal uit ontwikkeld is. Uiteraard zijn er ook goede ervaringen, zoals de IPv6 implementatie van WindowsXP, deze is verrassend stabiel en conform de IPv6 standaarden.

Nadat de diverse services van de Amsterdam Internet Exchange geschikt gemaakt waren voor IPv6, is begonnen met het bijhouden van statistieken. Deze gegevens van oa. de IPv6 hits op [www.ams-ix.net](http://www.ams-ix.net), de smtp sessies over IPv6 en de hoeveelheid IPv6 verkeer is in grafieken uiteengezet. Hiervoor zijn zelfs scripts ontwikkeld welke de gegevens verzamelen, er is namelijk nog niet of nauwelijks software beschikbaar om het IPv6 gebruik te monitoren. Ondanks de relatief korte tijd waarin actief statistieken worden verzameld over IPv6 gebruik op het AMS-IX netwerk, kan toch voorzichtig geconcludeerd worden dat het gebruik ervan toeneemt. Helaas is nog niet alle server software geschikt voor IPv6. Gelukkig is wel te zien dat voornamelijk de populaire internet server software zoals Bind (dns) en Apache (http) in hun meest recente releases default IPv6 ondersteuning bieden. Voor een groot gedeelte van de overige software zijn vaak patches beschikbaar. Deze worden vrijwillig beschikbaar gesteld door ontwikkelaars over de hele wereld. Alle services waarvoor geen patch beschikbaar is, kunnen door middel van een 6to4 proxy over het algemeen toch nog via IPv6 beschikbaar gesteld worden. De invoering van IPv6 in het AMS-IX netwerk kan dan ook als geslaagd beschouwd worden. Zowel het service, management en het office netwerk zijn nu geschikt voor zowel IPv4 als IPv6.

De problematiek mbt IPv6 multihoming is een complex probleem, welke zeker niet binnen een korte tijd opgelost zal worden. Door Multihoming op dezelfde manier te implementeren als dat in IPv4 wordt gedaan, zullen problemen mbt. de schaalbaarheid van de route tabellen optreden. In deze scriptie zijn een aantal mogelijke oplossingen aangedragen. De vraag of en welke van deze oplossingen het uit eindelijk zal gaan halen blijft voorlopig onduidelijk. De beste oplossingen zijn lange termijn oplossingen. Veel mensen zijn het erover eens dat er een nieuwe versie van TCP moet komen. Hoewel dat oorspronkelijk niet de bedoeling was, zal het voor een aantal problemen een perfecte oplossing zijn, zo ook voor multihoming in IPv6. Echter de invoering van een nieuw transport protocol, bijvoorbeeld SCTP, zal lang duren en niet door iedereen zomaar aanvaard worden. Het gevolg van het vervangen van TCP door een modernere variant welke ondersteuning biedt voor oa de scheiding tussen locators en identifiers, zal zijn dat alle applicaties herschreven zullen moeten worden. Dit is dus duidelijk een lange termijn oplossing, waarvan nog maar de vraag is, of "men" daar uiteindelijk tot over zal gaan. Mijns inziens het gebruik van prefixen welke worden afgeleid van het ASnummer, de beste oplossing welke op korte termijn kan worden geïmplementeerd. Het enige wat hiervoor gedaan dient te worden, is de toewijzing van een nieuwe prefix voor organisatie met een eigen AS.



De manier waarop nu een IPv6 multihomed netwerk is gerealiseerd bij AMS-IX, is het gebruik van meerdere prefixen. Iedere host kan is op dit moment dus via 2 ISP's bereikbaar. Hoewel dit geen ideale oplossing is, de established verbindingen gaan namelijk verloren. Is het voor AMS-IX op dit moment de beste oplossing.

Al met al heb ik in de afgelopen 4 maanden een hoop geleerd over IPv6 en het troubleshooten van IPv6 netwerken. Tevens heb ik een goed inzicht gekregen in de architectuur van Internet en de functie van een Internet Exchange.

## Referenties

- [1] RFC 2460, hoofdstuk 8. Upper-Layer Protocol Issues  
Internet Protocol, Version 6 (IPv6) Specification  
December 1998  
S. Deering, R. Hinden
  
- [2] RFC 2460  
Internet Protocol, Version 6 (IPv6) Specification  
December 1998  
S. Deering, R. Hinden
  
- [3] RFC 1883  
Internet Protocol, Version 6 (IPv6) Specification  
December 1995  
S. Deering, R. Hinden
  
- [4] RFC 2373  
IP Version 6 Addressing Architecture  
July 1998  
S. Deering, R. Hinden
  
- [5] RFC 2375  
IPv6 Multicast Address Assignments  
July 1998  
S. Deering, R. Hinden
  
- [6] RFC 2461  
Neighbor Discovery for IP Version 6 (IPv6)  
December 1998  
T. Narten, E. Nordmark, W. Simpson
  
- [7] RFC 2545  
Use of BGP-4 Multiprotocol Extensions for IPv6 Inter-Domain Routing  
March 1999  
P. Marques, F. Dupont
  
- [8] RFC 2858  
Multiprotocol Extensions for BGP-4  
June 2000  
T. Bates, Y. Rekhter, Y. Rekhter, D. Katz
  
- [9] <http://www.ipnet6.org/postfix.html>  
Postfix with IPv6 and TLS
  
- [10] RFC2428  
FTP Extensions for IPv6 and NATs  
September 1998  
M. Allman, S. Ostermann, C. Metz

- 
- [11] RFC3041  
Privacy Extensions for Stateless Address Autoconfiguration in IPv6  
January 2001  
T. Narten, R. Draves
- [12] draft-ietf-multi6-multihoming-requirements-05  
Goals for IPv6 Site-Multihoming Architectures  
<http://www.ietf.org/internet-drafts/draft-ietf-multi6-multihoming-requirements-05.txt>
- [13] RFC1918  
Address Allocation for Private Internets  
February 1996  
Y. Rekhter, B. Moskowitz, B. Moskowitz, G. J. de Groot, E. Lear
- [14] Site Multihoming in IPv6 (multi6)  
<http://www.ietf.org/html.charters/multi6-charter.html>
- [15] ipv6mh  
<http://arneill-py.sacramento.ca.us/ipv6mh/>
- [16] RFC2960  
Stream Control Transmission Protocol  
October 2000  
R. Stewart, Q. Xie, K. Morneault, C. Sharp, H. Schwarzbauer, T. Taylor, I. Rytina,  
M. Kalla, L. Zhang, V. Paxson
- [17] RFC2101  
IPv4 address behavior today  
February 1997  
B. Carpenter, J. Crowcroft, Y. Rekhter
- [18] draft-ietf-ipngwg-esd-analysis-05  
Separating Identifiers and Locators in Addresses.  
An Analysis of the GSE Proposal for IPv6  
October, 1999  
Matt Crawford, Matt Crawford, Thomas Narten, John W. Stewart,  
Lixia Zhang
- [19] RFC 2002  
IP Mobility Support  
October 1996  
C. Perkins
- [20] draft-ietf-mobileip-ipv6-21.  
Mobility Support in IPv6  
February 26, 2003  
D. Johnson, C. Perkins, J. Arkko

- [21] draft-savola-multi6-asn-pi-00  
Multihoming Using IPv6 Addressing Derived from AS Numbers  
January 2003  
P. Savola
  
- [22] RFC2267  
Network Ingress Filtering  
January 1998  
P. Ferguson  
D. Senie
  
- [23] RFC3484  
Default Address Selection for Internet Protocol version 6 (IPv6)  
February 2003  
R. Draves

## Bijlage I

### AMS-IX IPv6 integratie plan

Om het netwerk van de Amsterdam Internet Exchange IPv6 geschikt te maken en de services ook onder IPv6 beschikbaar te maken is het volgende plan van aanpak geschreven.

Het migreren van het AMS-IX netwerk naar dual stack IPv4 / IPv6, kan het beste in 4 stappen gedaan worden.

1. De eerste stap, in deze fase moeten alle netwerk componenten ge-upgrade worden, zodat de security gegarandeerd kan worden.
2. De tweede stap is de publiek toegankelijke services van de Amsterdam Internet Exchange zowel dmv IPv4 als IPv6 benaderbaar te maken. Zodoende kunnen mensen zowel via een Ipv4 als een Ipv6 adres gebruik maken van de services van de Amsterdam Internet exchange.
3. De derde stap is het kantoor/noc netwerk beschikbaar maken voor IPv6. Dit betekent dat de gebruikers op het kantoor, als ze dit willen, gebruik kunnen maken van IPv4 en/of IPv6.
4. De laatste stap is om ook de overige, niet publiek toegankelijke, servers geschikt te maken voor IPv6.

### Netwerk componenten

Het netwerk van de AMS-IX bestaat uit 2 routers, namelijk rtr1 en idifix.

Beide routers zijn al geschikt voor IPv6, dwz. Deze hebben een IPv6 adres en routeren IPv6 pakketten.

Helaas is er nog een probleem mbt de security. De IOS versies die nu gebruikt worden op beide routers ondersteunen nog geen extended access-lists. Dit brengt een aantal grote problemen met zich mee. Het netwerk is namelijk zo ontworpen, dat access-lists op de routers de verschillende VLANs afschermen van het Internet. Zo is het bijvoorbeeld niet toegestaan om vanaf Internet op het management lan te komen, het management netwerk is alleen toegankelijk vanaf het kantoornetwerk. Het kantoornetwerk op zijn beurt is weer niet geheel toegankelijk vanaf Internet, idifix heeft namelijk een firewall functionaliteit tbv de kantoor omgeving.

Deze restricties moeten ook in Ipv6 behouden blijven, het is dus noodzakelijk dat de routers extended access-lists ondersteunen, standard access-lists volstaan niet.

De eerste fase van het IPv6 integratie plan, is dan ook de beide routers te upgraden naar een nieuwere versie van het Cisco IOS, welke wel extended access-lists ondersteunen, zodat de veiligheid van het netwerk gewaarborgd blijft.

Mocht dit niet mogelijk zijn, bijv door dat de hardware dit niet ondersteund ( bijv te weinig geheugen), dan moet er naar een andere oplossing worden gekeken, dit zou bijv ook gedaan kunnen worden door een \*nix of BSD achtige machine.

Wanneer dit probleem zich alleen voordoet op idifix, moet worden bekeken in hoeverre het mogelijk is de firewall functionaliteit tbv. het kantoor netwerk, te verschuiven naar rtr1.

Wanneer de routers zijn ge-upgrade, dienen deze van de juiste adressen te worden voorzien, dit is voor een gedeelte al gerealiseerd (idifix-rtr1 netwerk).

## Publieke servers

Dit zijn de servers die voor iedereen rechtstreeks toegankelijk zijn vanaf Internet.

Je kunt hierbij denken aan de website van de Amsterdam Internet Exchange maar ook de mailservers en daarbij hoort uiteraard ook de dns server, want als er geen IPv6 dns queries gedaan kunnen worden, heeft het gereed maken van de website voor IPv6 geen nut.

De servers van de AMS-IX die vanaf Internet bereikbaar zijn, staan merendeels in het daarvoor gecreëerde vlan, de dmz, ofwel het service lan. In dit lan staan twee servers, namelijk server1.AMS-IX.net en melix.AMS-IX.net. Beide servers worden zeer binnenkort vervangen door nieuwe installaties. Server1 wordt fysiek vervangen door een nieuwe machine namelijk matrix, melix krijgt nieuwe harde schijven, met daarop een verse Red Hat 8 installatie.

## Webserver

De webserver zal komen te draaien op de nieuwe machine “matrix.ams-ix.net”.

Deze machine is voorzien van Red Hat8, deze versie van Red Hat is geschikt voor IPv6.

Als webserver software wordt apache gebruikt. De versie die is geïnstalleerd, is apache2.

Deze versie van apache is geschikt voor IPv6, het enige wat hiervoor gedaan hoeft te worden is de webserver een IPv6 adres geven en juist te configureren, zodat deze ook naar IPv6 adressen zal gaan luisteren.

## Mailserver

De mail voor het domein ams-ix.net en ams-ix.nl wordt op dit moment als volgt afgehandeld:

```
host -t mx ams-ix.net
ams-ix.net mail is handled by 30 fulliautomatix.noc.ams-ix.net.
ams-ix.net mail is handled by 10 melix.ams-ix.net.
ams-ix.net mail is handled by 20 panoramix.noc.ams-ix.net.
host -t mx ams-ix.nl
ams-ix.nl mail is handled by 20 fulliautomatix.noc.ams-ix.net.
ams-ix.nl mail is handled by 10 melix.ams-ix.net.
```

Voor AMS-IX wordt de mail dus afgehandeld door 3 verschillende mailservers.

- melix.ams-ix.net
- panoramix.noc.ams-ix.net
- fulliautomatix.noc.ams-ix.net

De primaire mailserver voor de beide domeinen is melix. Tijdens de migratie naar IPv6 zal deze als eerste gedaan moeten worden, zodat de andere 2 als backup mailserver kunnen blijven fungeren.

De andere 2 servers kunnen daarna gebeuren. Het is verstandig hiermee een tijdje te wachten, zodat eventuele bugs, configuratie fouten ontdekt kunnen worden.

Bovendien is het zo dat de meeste mailservers, zo zijn geconfigureerd, dat als ze de mail niet via IPv6 kunnen afleveren, ze dit via een IPv4 connectie zullen doen.

Bij de Amsterdam Internet Exchange wordt gebruik gemaakt van de MTA software “postfix”, normaal gesproken, wordt deze vanuit een RedHat pakket (rpm) geïnstalleerd.

Echter de standaard rpm's voor postfix bieden geen ondersteuning voor IPv6.

Er zijn 2 mogelijkheden om postfix aan IPv6 sockets te laten binden, postfix uit de source installeren met de benodigde IPv6 patch, of een rpm source package gebruiken, welke ook ondersteuning biedt voor IPv6.

Ik denk dat de laatste optie voorkeur verdient, omdat zo alles netjes vanuit rpm packages geïnstalleerd wordt en het overzicht beter is.

De source packages zijn te vinden op de volgende site: <http://postfix.w10.org/en/available-packages/>

Als deze software bewezen heeft stabiel te zijn, kan het vervolgens geïnstalleerd worden op panoramix en fullieautomatix.

## **Dns**

Het upgraden van software op de verschillende servers heeft geen enkele zin, als er niemand over IPv6 deze services opvraagt. Omdat te laten gebeuren zijn er dns entry's nodig om de IPv6 adressen bekend te maken. De DNS server voor het ams-ix domain zijn:

3. nemix1.ams-ix.net
4. nemix2.ams-ix.net

voor de dns server software wordt gebruik gemaakt van de software van ISC (Internet Software Consortium), dit is op Internet de meest gebruikte dns server software.

Nemix1 is dezelfde machine als panoramix, dit is de intranet server (voor dns van buitenaf benaderbaar). Op panoramix draait op al bind9, “bind” versie9 is een dns server welke ook op IPv6 sockets kan luisteren, een upgrade is dus niet noodzakelijk.

Echter, wel dienen de juiste dns entry's te worden aangemaakt voor de verschillende services, denk hierbij aan MX record en AAAA records.

De secundaire nameserver is nemix2, dit is dezelfde machine als melix. De nieuwe installatie van melix is ook voorzien van bind9, dus ook op deze machine is geen upgrade van bind nodig. De zone files zal nemix2 dmvs axfr (zone transfers) ontvangen, dus de zone files hoeven alleen op de nemix1 te worden aangepast.

## **Mailing list**

De mail wordt zoals hierboven al vermeld staat door melix afgehandeld, ook de mailinglist-server draait hierop. Voor de mail hoeft dus niets meer gedaan te worden, echter de bijbehorende webinterface dient nog wel juist geconfigureerd te worden.

De melix server wordt opnieuw geïnstalleerd met een RPM van apache2, welke al voor IPv6 geschikt is, het enige wat nog gedaan dient te worden is de configuratie van apache aan te passen.

## AMS-IX kantoor netwerk

De derde fase van het IPv6 integratie plan, is het geschikt maken van het kantoor netwerk voor IPv6. Dit betekent dat de kantoor werkplekken zowel via Ipv4 als via Ipv6 connectiviteit naar “buiten” hebben.

De meeste werkstations die op het NOC gebruikt worden zijn al geschikt voor IPv6, dit zijn apple machines, linux werkstations en windows XP/2000 pc's.

In IPv6 bestaat de mogelijkheid dat hosts zich zelf configureren dmv autoconfiguration. De benodigde (prefix/route) informatie wordt dan bekend gemaakt door de router in het netwerk. Deze functie zal ook door de router in het kantoor netwerk ondersteund moeten worden. Deze functionaliteit is op dit moment al geïmplementeerd op de router, dus hier hoeft als het goed is niets meer voor gedaan te worden.

Wel dienen er access-lists te worden geïmplementeerd op de kantoor router, zodat het netwerk dmv een `firewall` is afgeschermd van het (IPv6) Internet.

Als ook dat gedaan is, dan is het kantoor/noc netwerk geschikt voor IPv6 communicatie met het Internet.

## Overige servers

Nu het AMS-IX kantoor netwerk geschikt is voor Ipv6, kan begonnen worden met de upgrade van de beheers ondersteunende servers/software. Dit zijn bijvoorbeeld de ssh servers op de verschillende machines, de FTP daemon op de webserver tbv het uploaden van content, etc.

Het upgraden van deze servers kan het beste per vlan gedaan worden.  
In onderstaande tabel staat wat per vlan geupgrade dient te worden in deze fase.

### ***VLAN 13, service LAN***

#### **Matrix**

1. time server dient via ipv6 te kunnen synchroniseren en queries kunnen beantwoorden
2. bereikbaarheid via ssh

#### **Melix**

1. time server dient via IPv6 te kunnen synchroniseren en queries kunnen beantwoorden
2. bereikbaarheid via ssh

### ***VLAN 11, management LAN***

#### **nitelix.noc.ams-ix.net** (meet server nikhef)

1. voorzien van een IPv6 adres

#### **glotelix.noc.ams-ix.net** (meet server global switch)



1. voorzien van een IPv6 adres

**teltelix.noc.ams-ix.net** (meet server telicity)

1. voorzien van een IPv6 adres

**satelix.noc.ams-ix.net** (meet server sara)

1. voorzien van een IPv6 adres

**melix-mg.noc.ams-ix.net** (management interface melix)

1. voorzien van een IPv6 adres

**matrix-mg.noc.ams-ix.net** (management interface matrix)

1. voorzien van een IPv6 adres

In het management LAN staan ook 4 terminalserver (op elke locatie een), en bij sara en nikhef staan UPS systemen, uitgezocht moet worden of deze te upgraden zijn, zodat deze via IPv6 bereikbaar zijn.

### **Office/NOC LAN**

In fase 3 is het kantoor netwerk geschikt gemaakt voor dual stack hosts, in deze fase zullen de servers binnen het kantoor netwerk geschikt gemaakt gaan worden.

**panoramix.noc.ams-ix.net.**

Dit is de intranet server, de services die geupgrade zullen gaan worden zullen zijn:

1. als eerste voorzien van een IPv6 adres
2. aaaa record voor panoramix.noc.ams-ix.net)
3. apache upgraden naar of apache2 of apache 1.3.27 (gepatched).
4. Pop3
5. Imap2

Er moet bekeken worden of het wenselijk, dan wel mogelijk is de volgende hosts te upgraden zodat ze IPv6 compatible zijn:

- rome.noc.ams-ix.net (193.194.136.130) cisco switch
- assurancetourix.noc.ams-ix.net (193.194.136.131) Cisco Aironet AP
- offix.noc.ams-ix.net (193.194.136.138) terminal server in het lab
- arachnix.ams-ix.net
- HP printer
- ams001.office.ams-ix.net (193.194.136.254) NT4 server

### **VLAN 512, Test LAN**

In dit lan staan de 3 servers

1. fulliautomatix.noc.ams-ix.net
2. perix.ams-ix.net
3. veesix.ams-ix.net

**fulliautomatix.noc.ams-ix.net** (193.194.136.66). Op deze server draait op dit moment: ssh, www, smtp, pop, imap en dns.

Deze machine is ook bekend als:

ir-baboon.monkey-mind.net (193.194.136.67) = fulliautomatix

sirius1.galact-ix.net (193.194.136.68) = fulliautomatix

ns1.nlnog.net. (193.194.136.71) = fulliautomatix

Hierop draaien wat websites, moet dit over naar IPv6?

**perix.ams-ix.net** (193.194.136.69) Lijkt alleen ssh op te draaien, is voor een project van iemand buiten AMS-IX (netflow achtig iets)

**veesix.ams-ix.net** (193.194.136.70) Dit is/was de AMS-IX IPv6 tunnel server.

Deze wordt binnenkort opgeheven.

## **Uiteindelijke resultaat**

Uiteindelijk zullen de op de volgende systemen de volgende services op IPv6 werken.

### ***service LAN***

**Host: Matrix**

<b>Service</b>	<b>Ipv6 compatible</b>
Ssh	Fase 4
http	Fase 2
https	Fase 2
Ntp	Fase 4

**Host: Melix**

<b>Service</b>	<b>Ipv6 compatible</b>
Ssh	Fase 4
http	Fase 2
https	Fase 2
Ntp	Fase 4
Dns	Fase 2
Sntp	Fase 2

### ***management LAN***

**Host: nitelix.noc.ams-ix.net**

<b>Service</b>	<b>Ipv6 compatible</b>
Ssh	Fase 4

**Host: glotelix.noc.ams-ix.net**

<b>Service</b>	<b>Ipv6 compatible</b>
Ssh	Fase 4

**Host: teltelix.noc.ams-ix.net**

<b>Service</b>	<b>Ipv6 compatible</b>
Ssh	Fase 4

**Host: satelix.noc.ams-ix.net**

<b>Service</b>	<b>Ipv6 compatible</b>
Ssh	Fase 4

**Host: sarfix.noc.ams-ix.net (terminal server sara)**

Service	Ipv6 compatible
Ssh	Optioneel

**Host: nilfix.noc.ams-ix.net (terminal server nikhef)**

Service	Ipv6 compatible
Ssh	Optioneel

**Host: telfix.noc.ams-ix.net (terminal server telecity)**

Service	Ipv6 compatible
Ssh	Optioneel

**Host: glofix.noc.ams-ix.net (terminal server global-switch)**

Service	Ipv6 compatible
Ssh	Optioneel

**Host: sarazonderprefix.noc.ams-ix.net (ups sara)**

Service	Ipv6 compatible
telnet	Optioneel
http	Optioneel

**Host: niksgeenprikfix.noc.ams-ix.net (ups nikhef)**

Service	Ipv6 compatible
telnet	Optioneel
http	Optioneel

**Host: melix-mg.ams-ix.net (management interface melix)**

Service	Ipv6 compatible
Ssh	Fase 4
http	Fase 2
https	Fase 2
Ntp	Fase 4
Dns	Fase 2
Smtpt	Fase 2

**Host: matrix-mg.ams-ix.net (management interface matrix)**

Service	Ipv6 compatible
Ssh	Fase 4
http	Fase 2
https	Fase 2
Ntp	Fase 4

## Office/NOC LAN

**Host:** panoramix.noc.ams-ix.net.

Service	Ipv6 compatible
Ssh	Already done
ftp	Already done
http	Fase 4
https	Fase 4
Dns	Fase 2
Pop	Fase 4
Imap	Fase 4

**Host:** rome.noc.ams-ix.net (cisco switch)

Service	Ipv6 compatible
telnet	Optioneel
http	Optioneel

**Host:** assurancetourix.noc.ams-ix.net (Cisco Aironet AP)

Service	Ipv6 compatible
telnet	Optioneel
http	Optioneel

**Host:** offix.noc.ams-ix.net (terminalserver in het lab)

Service	Ipv6 compatible
telnet	Optioneel
Ssh	Optioneel

**Host:** ams001.office.ams-ix.net (nt4 server) (193.194.136.254)

Service	Ipv6 compatible
Smtip	Optioneel

**Host:** HP printer

Service	Ipv6 compatible
telnet	Optioneel
http	Optioneel

**Host:** Arachnix

Service	Ipv6 compatible
telnet	Optioneel
http	Optioneel

## IPv6 nummer plan

De Amsterdam Internet Exchange, heeft 2 IPv6 reeksen, deze zijn beide 48 bits.  
De twee reeksen zijn:

1. 2001:0610:0140::/48 (LIR is Surfnets)
2. 2001:07B8:200::/48 (LIR is BIT)

Er worden 2 reeksen gebruikt, zodat de servers via meerdere adressen bereikbaar zijn,  
Voor het geval het ene adres niet bereikbaar is, dit gaat volgens het round robin principe.  
De exacte uitwerking volgt in het tweede gedeelte van mijn afstudeer opdracht.

Er is al een IPV6 nummer plan gemaakt door Arien, dit nummer plan kan wat mij betreft gewoon  
gebruikt blijven worden, omdat ik denk dat deze goed doordacht is.  
Deze ziet er als volgt uit.

inet6num	Range	LIR
2001:0610:0140::/48	2001:0610:0140:0000:0000:0000:0000:0000 2001:0610:0140:ffff:ffff:ffff:ffff:ffff	surfnets
2001:07B8:200::/48	2001:07b8:0200:0000:0000:0000:0000:0000 2001:07b8:021f:ffff:ffff:ffff:ffff:ffff	bit

Deze adres ruimte wordt als volgt onderverdeeld:

use	Prefix
reserved	2001:xxxx:xxxx:0000::/52
reserved	2001:xxxx:xxxx:1000::/52
<b>Offices</b>	<b>2001:xxxx:xxxx:2000::/52</b>
reserved	2001:xxxx:xxxx:3000::/52
reserved	2001:xxxx:xxxx:4000::/52
reserved	2001:xxxx:xxxx:5000::/52
reserved	2001:xxxx:xxxx:6000::/52
reserved	2001:xxxx:xxxx:7000::/52
reserved	2001:xxxx:xxxx:8000::/52
reserved	2001:xxxx:xxxx:9000::/52
<b>POPs</b>	<b>2001:xxxx:xxxx:a000::/52</b>
reserved	2001:xxxx:xxxx:b000::/52
reserved	2001:xxxx:xxxx:c000::/52
reserved	2001:xxxx:xxxx:d000::/52
reserved	2001:xxxx:xxxx:e000::/52
reserved	2001:xxxx:xxxx:f000::/52

## Office

De adres ruimte voor offices wordt verder onderverdeeld voor de verschillende offices, Op dit moment is er maar een office, dat is aan het Westeinde, de overige reeksen zijn voor toekomstig gebruik.

Offices	
reserved	2001:xxxx:xxxx:2000::/56
reserved	2001:xxxx:xxxx:2100::/56
<b>Westeinde</b>	<b>2001:xxxx:xxxx:2200::/56</b>
reserved	2001:xxxx:xxxx:2300::/56
...	
reserved	2001:xxxx:xxxx:2f00::/56

Op dit moment wordt dus alleen het netwerk voor het westeinde gebruikt, deze wordt onderverdeeld in diverse /64 netwerken.

Westeinde	
reserved	2001:xxxx:xxxx:2200::/64
reserved	2001:xxxx:xxxx:2201::/64
<b>office LANs</b>	<b>2001:xxxx:xxxx:2202::/64</b>
reserved	2001:xxxx:xxxx:2203::/64
...	
reserved	2001:xxxx:xxxx:22f2::/64
<b>transit - SARA</b>	<b>2001:xxxx:xxxx:22f3::/64</b>
reserved	2001:xxxx:xxxx:22f4::/64
reserved	2001:xxxx:xxxx:2f52::/64
<b>transit - GS</b>	<b>2001:xxxx:xxxx:22f6::/64</b>
reserved	2001:xxxx:xxxx:22f6::/64
...	
reserved	2001:xxxx:xxxx:22ff::/64

Voor de point to point verbindingen, wordt gebruik gemaakt van een /64, dit zijn de fiber verbinding naar global switch en de SDSL verbinding naar SARA.

In het office LAN zal gebruik gemaakt gaan worden van autoconfiguratie, behalve voor de servers, en de medewerkers die graag een vast IP adres willen hebben

## Pops

Elke Pop krijgt een /56 toegewezen, dit is als volgt onderverdeeld:

<b>POPs</b>	
reserved	2001:xxxx:xxxx:a000::/56
...	
reserved	2001:xxxx:xxxx:a200::/56
SARA	2001:xxxx:xxxx:a300::/56
NIKHEF	2001:xxxx:xxxx:a400::/56
TeleCity	2001:xxxx:xxxx:a500::/56
Global	
Switch	2001:xxxx:xxxx:a600::/56
reserved	2001:xxxx:xxxx:a700::/56
...	
reserved	2001:xxxx:xxxx:af00::/56

## GlobalSwitch

Al onze servers, staan bij global switch, het nummer plan voor de verschillende vlands is als volgt opgezet:

<b>Global Switch</b>	
reserved	2001:xxxx:xxxx:a600::/64
reserved	2001:xxxx:xxxx:a601::/64
<b>Mng. LAN</b>	<b>2001:xxxx:xxxx:a602::/64</b>
reserved	2001:xxxx:xxxx:a603::/64
<b>DMZ</b>	<b>2001:xxxx:xxxx:a604::/64</b>
reserved	2001:xxxx:xxxx:a605::/64
reserved	2001:xxxx:xxxx:a606::/64
<b>Test LAN</b>	<b>2001:xxxx:xxxx:a607::/64</b>
reserved	2001:xxxx:xxxx:a608::/64
...	
reserved	2001:xxxx:xxxx:a6ff::/64

Welke Ipv6 adressen de servers / hosts / routers uiteindelijk uit de betreffende reeks krijgen kan tzt. worden bekeken.

2 reeksen zijn:  
2001:610:140::48  
2001:7b8:200::/48

### **Kantoor netwerk 2001:xxxx:xxxx:2202::/64**

Idefix interface FastEthernet0/1

2001:610:140:2202::1/64  
2001:7B8:200:2202::1/64

Panoramix:

Twee statische adressen en twee autoconfigured

2001:610:140:2202::2/64  
2001:7b8:200:2202::2/64  
2001:610:140:2202:a00:20ff:feec:8a24/128  
2001:7b8:200:2202:a00:20ff:feec:8a24/128

Melix2

Twee statische adressen en twee autoconfigured

Vanaf deze machine worden ook diverse check gedaan, waarin de verbinding van de verschillende ISP wordt getest.

2001:610:140:2202::200/64  
2001:7b8:200:2202::200/64  
2001:610:140:2202:203:47ff:feae:f75c/64  
2001:7b8:200:2202:203:47ff:feae:f75c/64

Idefix

interface FastEthernet0/0  
ipv6 address 2001:7B8:200:22F3::2/64  
ipv6 address 2001:610:140:22F3::2/64

interface Serial0/0:0

ipv6 address 2001:610:140:22F6::2/64  
ipv6 address 2001:7B8:200:22F6::2/64

### **Service LAN**

2001:xxxx:xxxx:a604::/64  
2001:610:140:a604:/64  
2001:7b8:200:a604:/64

De servers zullen ook allemaal een autoconfigured adres krijgen,  
Zodat ze de routes automatisch leren.

Rtr1

2001:610:140:a604::1/64  
2001:7b8:200:a604::1/64

Matrix

2001:610:140:a604::2/64  
2001:7b8:200:a604::2/64



Melix

2001:610:140:a604::3/64

2001:7b8:200:a604::3/64

nemix2

2001:610:140:a604::4/64

2001:7b8:200:a604::4/64

**Management lan**

2001:xxxx:xxxx:a602::/64

2001:610:140:a602::/64

2001:7b8:200:a602::/64

rtr1 interface FastEthernet3/0

2001:610:140:a602::1/64

2001:7b8:200:a602::1/64

nitelix.noc.ams-ix.net 41

2001:610:140:a602::41/64

2001:7b8:200:a602::41/64

glotelix.noc.ams-ix.net 44

2001:610:140:a602::44/64

2001:7b8:200:a602::44/64

teltelix.noc.ams-ix.net 49

2001:610:140:a602::49/64

2001:7b8:200:a602::49/64

satelix.noc.ams-ix.net 50

2001:610:140:a602::50/64

2001:7b8:200:a602::50/64

melix-mg.noc.ams-ix.net 47

2001:610:140:a602::47/64

2001:7b8:200:a602::47/64

matrix-mg.noc.ams-ix.net 48

2001:610:140:a602::48/64

2001:7b8:200:a602::48/64

## BIJLAGE II

Tijdens deze proef is gebruik gemaakt van het software pakket sctp\_darn, deze software wordt bij de Linux kernel patch geleverd.

De patch zelf is te downloaden op <http://lksctp.sourceforge.net/> en dient gebruikt te worden met de Linux 2.5.67 kernel. Nadat de kernel gepatched en op nieuw gecompileerd is, is de nieuwe Linux kernel uitgerust met sctp ondersteuning en als het goed is geschikt voor sctp applicaties.

Om te testen wordt gebruik gemaakt van een klein demo programma, welke door de makers van de patch beschikbaar is gesteld. Het is een simpel cliënt/server programma waarmee tekst kan worden verzonden. Tekst die op de cliënt wordt ingevoerd verschijnt bij de server in het scherm. De software is geschikt voor zowel IPv4 als IPv6 (en theoretisch is dit zelfs door elkaar te gebruiken).

Eerst starten we de server:

```
./sctp_darn -H 10.0.0.1 -B 10.0.0.2 -P 9001 -l
```

Het primaire adres is hier 10.0.0.1 als alternatief adres geven we 10.0.0.2 op, de server luistert op poort 9001. de “-l” betekend listen mode.

Vervolgens de cliënt:

```
./sctp_darn -H 10.0.0.3 -P 9000 -B 10.0.0.4 -h 10.0.0.1 -p 9001
```

De cliënt gebruikt als primair adres 10.0.0.3 en als alternatief adres 10.0.0.4, het poortnummer welke de cliënt gebruikt is 9000. Vervolgens geven we het adres van de server op en het bijbehorende poortnummer. De “-s” betekent send mode.

Deze proef is op één machine uitgevoerd, de server en cliënt zijn dus fysiek de zelfde machine \*, de ip configuratie ziet er als volgt uit:

```
eth0:1   Link encap:Ethernet  HWaddr 00:02:44:23:45:3C
         inet addr:10.0.0.1  Bcast:10.255.255.255  Mask:255.0.0.0
         UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
         Interrupt:11 Base address:0xfc00

eth0:2   Link encap:Ethernet  HWaddr 00:02:44:23:45:3C
         inet addr:10.0.0.2  Bcast:10.255.255.255  Mask:255.0.0.0
         UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
         Interrupt:11 Base address:0xfc00

eth0:3   Link encap:Ethernet  HWaddr 00:02:44:23:45:3C
         inet addr:10.0.0.3  Bcast:10.255.255.255  Mask:255.0.0.0
         UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
         Interrupt:11 Base address:0xfc00

eth0:4   Link encap:Ethernet  HWaddr 00:02:44:23:45:3C
         inet addr:10.0.0.4  Bcast:10.255.255.255  Mask:255.0.0.0
         UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
         Interrupt:11 Base address:0xfc00
```

\* De proef is ook op 2 verschillende hosts uitgevoerd. Het is dus niet zo dat dit alleen werkt als de proef op één host wordt uitgevoerd.

Nu zowel de cliënt als de server draaien kunnen we de software gebruiken.  
Links is de server te zien, rechts de cliënt.

```
./sctp_darn listening...

****TEST PRINT MESSAGE****
SNDRCV
sinfo_stream 0
sinfo_ssn 0
sinfo_flags 0x0
sinfo_ppid 0
sinfo_context 0
sinfo_tsn      3537489583
sinfo_cumtsn  0
Body:  dit is een test.
****END TEST PRINT MESSAGE****
```

```
./sctp_darn ready to send...
10.0.0.1:9001> dit is een test
```

Tijdens deze communicatie is met een protocol analyzer het verkeer gemonitord. In de onderstaande afbeelding is te zien hoe de sctp verbinding wordt opgezet en hoe de alternatieve adressen tijdens de handshake (INIT & INIT\_ACK) worden vastgelegd.

```
Internet Protocol,Src Addr: 10.0.0.3 (10.0.0.3), Dst Addr: 10.0.0.1 (10.0.0.1)
Stream Control Transmission Protocol
Source port: 9000
Destination port: 9001
Verification tag: 0x00000000
Checksum: 0x278d086d (correct Adler32)
INIT chunk requesting for 10 outbound streams and accepting up to 65535 inbound streams
Identifier: INIT (1)
Flags: 0
Length: 48
Initiate tag: 0xd39abf18
Advertised receiver window credit (a_rwnd): 32768
Number of outbound streams: 10
Number of inbound streams: 65535
Initial TSN: 824695255
IPV4 address parameter
  Parameter type: IPv4 address (0x0005)
  Parameter length: 8
  IP Version 4 address: 10.0.0.3 (10.0.0.3)
IPV4 address parameter
  Parameter type: IPv4 address (0x0005)
  Parameter length: 8
  IP Version 4 address: 10.0.0.4 (10.0.0.4)
Supported address types parameter reporting 1 address type
  Parameter type: Supported address types (0x000c)
  Parameter length: 6
  Supported Address Types (1 address type)
    Supported address type: 0x0005 (IPv4 address)
  Padding: 2 bytes
ECN parameter
  Parameter type: ECN (0x8000)
  Parameter length: 4
```

```
Internet Protocol, Src Addr: 10.0.0.1 (10.0.0.1), Dst Addr: 10.0.0.3 (10.0.0.3)
Stream Control Transmission Protocol
  Source port: 9001
  Destination port: 9000
  Verification tag: 0xd39abf18
  Checksum: 0xa03e1dd6 (correct Adler32)
  INIT_ACK chunk requesting for 10 outbound streams and accepting up to 10 inbound streams
  Identifier: INIT_ACK (2)
  Flags: 0
  Length: 204
  Initiate tag: 0x7ffad5b5
  Advertised receiver window credit (a_rwnd): 32768
  Number of outbound streams: 10
  Number of inbound streams: 10
  Initial TSN: 1990936521
  IPv4 address parameter
    Parameter type: IPv4 address (0x0005)
    Parameter length: 8
    IP Version 4 address: 10.0.0.1 (10.0.0.1)
  IPv4 address parameter
    Parameter type: IPv4 address (0x0005)
    Parameter length: 8
    IP Version 4 address: 10.0.0.2 (10.0.0.2)
  State Cookie Parameter with 160 bytes cookie
    Parameter type: State cookie (0x0007)
    Parameter length: 164
    State cookie (160 bytes)
  ECN parameter
    Parameter type: ECN (0x8000)
    Parameter length: 4
```

Nu de verbinding is opgezet kan het verkeer worden verzonden, dit ziet er als volgt uit:

```
10.0.0.3 -> 10.0.0.1      SCTP DATA
10.0.0.1 -> 10.0.0.3      SCTP SACK
```

Zoals te zien is gaat nu de data van de cliënt 10.0.0.3 naar de server 10.0.0.1, hierop komt een Acknowledgement terug.

Vervolgens wordt het ip adres van de server verwijderd (ifconfig eth0:1 down).

De sessie blijft bestaan, en een bericht die nu op de cliënt wordt ingetikt komt nog steeds op de server aan, met de protocol analyzer is het volgende waar te nemen:

```
10.0.0.3 -> 10.0.0.2      SCTP DATA
10.0.0.2 -> 10.0.0.3      SCTP SACK
```

Zoals te zien is heeft het protocol waargenomen dat het adres 10.0.0.1 niet langer bereikbaar is, de cliënt heeft nu het adres 10.0.0.2 als destination adres gekozen.

De volgende stap is het verwijderen van één van de 2 adressen van de cliënt, het adres 10.0.0.3 wordt verwijderd (ifconfig eth0:3 down). Berichten die nu worden ingetikt op cliënt komen nog steeds aan op de server! Met de protocol analyzer is het volgende waargenomen:

```
10.0.0.4 -> 10.0.0.2      SCTP DATA
10.0.0.2 -> 10.0.0.4      SCTP SACK
```

De cliënt heeft het source adres gewijzigd van 10.0.0.3, naar 10.0.0.4 (zijn alternatieve adres).

Tijdens deze proef zijn alle twee de primaire adressen verwijderd en toch hield de sessie stand!

Hieruit kan geconcludeerd worden dat de multihoming feature goed werkt in combinatie met IPv4.

De volgende stap is om dezelfde proef te doen, maar nu met IPv6 als layer3 protocol.  
Tijdens deze proef wordt gebruik gemaakt van sixbone adressen en adressen uit de xs4all reeks.

	<b>Primair adres</b>	<b>Secundair adres</b>
Server	2001:888:1357::2	3ffe:8114:2000:1394::2
Cliënt	2001:888:1357::3	3ffe:8114:2000:1394::3

De software wordt als volgt gestart:

Server:

```
./sctp_darn -H 2001:888:1357::2 -P 9000 -B 3ffe:8114:2000:1394::2 -h  
2001:888:1357::3 -p 9001 -s
```

Cliënt:

```
./sctp_darn -H 2001:888:1357::3 -B 3ffe:8114:2000:1394::3 -P 9001 -l
```

De Sessie wordt opgebouwd en er wordt een bericht van de cliënt naar de server verstuurd.

Dit ziet er als volgt uit:

```
2001:888:1357::3 -> 2001:888:1357::2 SCTP INIT  
2001:888:1357::2 -> 2001:888:1357::3 SCTP INIT_ACK  
2001:888:1357::3 -> 2001:888:1357::2 SCTP COOKIE_ECHO DATA  
2001:888:1357::2 -> 2001:888:1357::3 SCTP COOKIE_ACK  
2001:888:1357::2 -> 2001:888:1357::3 SCTP SACK
```

de init en init\_ack berichten zijn uitgewerkt op de volgende pagina terug te zien:

```
Internet Protocol Version 6
  Version: 6
  Traffic class: 0x00
  Flowlabel: 0x00000
  Payload length: 84
  Next header: SCTP (0x84)
  Hop limit: 64
  Source address: 2001:888:1357::3 (2001:888:1357::3)
  Destination address: 2001:888:1357::2 (2001:888:1357::2)
Stream Control Transmission Protocol
  Source port: 9000
  Destination port: 9001
  Verification tag: 0x00000000
  Checksum: 0x17500b21 (correct Adler32)
  INIT chunk requesting for 10 outbound streams and accepting up to
65535 inbound streams
  Identifier: INIT (1)
  Flags: 0
  Length: 72
  Initiate tag: 0x5bd43d44
  Advertised receiver window credit (a_rwnd): 32768
  Number of outbound streams: 10
  Number of inbound streams: 65535
  Initial TSN: 624882465
  IPV6 address parameter
    Parameter type: IPv6 address (0x0006)
    Parameter length: 20
    IP Version 6 address: 2001:888:1357::3 (2001:888:1357::3)
  IPV6 address parameter
    Parameter type: IPv6 address (0x0006)
    Parameter length: 20
    IP Version 6 address: 3ffe:8114:2000:1394::3
(3ffe:8114:2000:1394::3)
  Supported address types parameter reporting 2 address types
    Parameter type: Supported address types (0x000c)
    Parameter length: 8
    Supported Address Types (2 address types)
      Supported address type: 0x0005 (IPv4 address)
      Supported address type: 0x0006 (IPv6 address)
  ECN parameter
    Parameter type: ECN (0x8000)
    Parameter length: 4
```

Interessant gegeven is dat te zien is, dat zowel IPv4 als IPv6 adressen gebruikt kunnen worden. Eventueel kunnen deze zelfs door elkaar gebruikt worden.

```
Internet Protocol Version 6
  Version: 6
  Traffic class: 0x02
  Flowlabel: 0x00000
  Payload length: 304
  Next header: SCTP (0x84)
  Hop limit: 64
  Source address: 2001:888:1357::2 (2001:888:1357::2)
  Destination address: 2001:888:1357::3 (2001:888:1357::3)
Stream Control Transmission Protocol
  Source port: 9001
  Destination port: 9000
  Verification tag: 0x5bd43d44
  Checksum: 0xdd8b2113 (correct Adler32)
  INIT_ACK chunk requesting for 10 outbound streams and accepting
  up to 10 inbound streams
  Identifier: INIT_ACK (2)
  Flags: 0
  Length: 292
  Initiate tag: 0x416d0619
  Advertised receiver window credit (a_rwnd): 32768
  Number of outbound streams: 10
  Number of inbound streams: 10
  Initial TSN: 436508471
  IPV6 address parameter
    Parameter type: IPv6 address (0x0006)
    Parameter length: 20
    IP Version 6 address: 2001:888:1357::2 (2001:888:1357::2)
  IPV6 address parameter
    Parameter type: IPv6 address (0x0006)
    Parameter length: 20
    IP Version 6 address: 3ffe:8114:2000:1394::2
    3ffe:8114:2000:1394::2)
  State Cookie Parameter with 224 bytes cookie
    Parameter type: State cookie (0x0007)
    Parameter length: 228
    State cookie (224 bytes)
  ECN parameter
    Parameter type: ECN (0x8000)
    Parameter length: 4
```

Wanneer een minuut lang gemonitord wordt, zijn worden ook de sctp hartbeats zichtbaar:

```
2001:888:1357::3 -> 2001:888:1357::2 SCTP HEARTBEAT
2001:888:1357::2 -> 2001:888:1357::3 SCTP HEARTBEAT_ACK
2001:888:1357::2 -> 2001:888:1357::3 SCTP HEARTBEAT
2001:888:1357::3 -> 2001:888:1357::2 SCTP HEARTBEAT_ACK
3ffe:8114:2000:1394::2 -> 3ffe:8114:2000:1394::3 SCTP HEARTBEAT
3ffe:8114:2000:1394::3 -> 3ffe:8114:2000:1394::2 SCTP HEARTBEAT_ACK
3ffe:8114:2000:1394::3 -> 3ffe:8114:2000:1394::2 SCTP HEARTBEAT
3ffe:8114:2000:1394::2 -> 3ffe:8114:2000:1394::3 SCTP HEARTBEAT_ACK
```

Standaard worden de 2001 adressen gebruikt, dit zijn namelijk de primaire adressen.

De volgende stap is het verwijderen van het primaire adres van de cliënt:

```
ip -6 addr delete 2001:888:1357::3/64 dev eth0
```

Vervolgens wordt er weer een bericht van de cliënt naar de server verstuurd:

```
3ffe:8114:2000:1394::3 -> 3ffe:8114:2000:1394::2 SCTP DATA
3ffe:8114:2000:1394::2 -> 3ffe:8114:2000:1394::3 SCTP SACK
```

Zoals te zien is nu wordt er gebruik gemaakt van het 3ffe adres, tevens was er een ICMP bericht te zien, met het bericht dat 2001:888:1357::3 niet bereikbaar was.

De volgende stap is om ook op de server het 2001 adres te verwijderen:

```
ip -6 addr delete 2001:888:1357::2/64 dev eth0
```

Nu wordt er weer een bericht van de cliënt naar de server verstuurd, het volgende is nu te zien:

```
3ffe:8114:2000:1394::3 -> 3ffe:8114:2000:1394::2 SCTP DATA
3ffe:8114:2000:1394::2 -> 3ffe:8114:2000:1394::3 SCTP SACK
```

De failovers gaan zonder dat de cliënt en/of server software hier iets van merkt.

Geconcludeerd kan worden, dat door gebruik te maken van sctp ipv tcp een hitless failover mogelijk is. In een latere test is ook het gebruik van IPv4 en IPv6 adressen door elkaar getest.

Geconcludeerd kan worden dat het terug vallen op een IPv4 adres ook mogelijk is.



## Bijlage III

### all-prefixes

In het bestand all-prefixes, staan de prefixen welke getest worden:

```
2001:610:140:2202::3
2001:7b8:200:2202::3
```

### telnet-router

Het script telnet-router, wordt gebruikt om in te loggen op de router, en ziet er als volgt uit.

```
#
# 08-03-2003 Andree Toonk, AMS-IX
# telnet-router sysname
# where:
# sysname - The system to which you wish to telnet
#
# Use this with a "pipe" or "redirect" to allow sending a stream of commands
# to a remote system via telnet.
# ./telnet-router idefix < input
#
#
# Program
#
TFILE=/tmp/tsc.$$
echo '#!/bin/sh' > $TFILE
echo '(' >> $TFILE
sed -e 's/\(.*\) /sleep 1 \; echo "\1"/' >> $TFILE
echo 'cat - ) | telnet' $1 >> $TFILE
chmod 700 $TFILE
$TFILE
rm $TFILE
```

### ifV6test

Dit is het main script, vanuit dit script worden de andere aangeroepen. Het script ziet er als volgt uit:

```
#!/usr/local/bin/bash
# Andree Toonk
# Ams-ix Mon Apr 17
cd /home/andree/v6

while true;do
action=false

cat all-prefixes |
while read line ;
do
PREFIX=`echo $line |cut -d : -f 1,2,3,4`::/64
CHECK=ok; /sbin/ping6 -c1 -S $line www.ipng.nl >/dev/null || CHECK=FOUT ;

if [ "$CHECK" = "FOUT" ]; then
echo "`date` ERROR ipng.nl is niet te bereiken met $PREFIX, Let's try
www.ipv6.bieringer.de";
# just to be sure, let's try to ping a second address
CHECK2=ok; /sbin/ping6 -c1 -S $line www.ipv6.bieringer.de >/dev/null ||
CHECK2=FOUT ;

if [ "$CHECK2" = "FOUT" ]; then
echo "`date` ERROR! $PREFIX is niet te gebruiken";
#eerst checkken of de prefix al uit de config is gehaald (lifetime is 0)
dorouter=false
echo "`date` checking if config is already been changed"
grep $PREFIX deprecated-prefix || dorouter=true
echo "`date` dorouter is $dorouter"
```

```
#als er nog met lifetime 300 wordt gedaverteerd moet dit veranderd worden naar
lifetime 0sec
    if [ "$dorouter" = "true" ]; then
        echo "`date` config has not been changed yet"
        rm input-tmp
        cp input input-tmp
        echo "ipv6 nd prefix-advertisement $PREFIX 0 0 autoconfig" >>input-tmp
        echo exit >>input-tmp
        echo exit >>input-tmp
        echo exit >>input-tmp
        # de input file input-tmp wordt nu naar de router geschreven
        echo "`date` removing prefix from advertising list"
        ./telnet-router idefix <input-tmp
        echo $PREFIX >> deprecated-prefix
        #Dns aanpassen
        echo "doDNS $PREFIX error"
        ./doDNS $PREFIX error
    else
        echo "`date` prefix has already been deprecated"
    fi
fi
else
    #echo "`date` OKIDOKI $PREFIX"
    action=false
    sleep 1
    grep $PREFIX deprecated-prefix && action=true
    #als de prefix weer bereikbaar is, dan weer terug zetten in config
    if [ "$action" = "true" ]; then
        echo "`date` de Prefix $PREFIX is weer te gerbuiken, wijzigen in router"
        rm input-tmp
        cp input input-tmp
        echo "ipv6 nd prefix-advertisement $PREFIX 600 300 autoconfig" >>input-
tmp
        echo exit >>input-tmp
        echo exit >>input-tmp
        echo exit >>input-tmp
        # de input file input-tmp wordt nu naar de router geschreven
        ./telnet-router idefix <input-tmp
        #Dns aanpassen
        echo
        ./doDNS $PREFIX ok
        tmpprefix=`echo $PREFIX |cut -d / -f1`
        cat deprecated-prefix |grep -v $tmpprefix > deprecated-prefix
    fi
fi
done;
sleep 5
done
```

## doDNS

Het script doDNS wordt aangeroepen vanuit ifV6test wanneer de DNS server aangepast dient te worden.

```
#!/usr/local/bin/bash
cd /home/andree/v6

BIT="2001:7b8:200:2202::/64"
SURFNET="2001:610:140:2202::/64"

# probleem bij een van de upstreams.
if [ $2 = "error" ];then

    if [ $SURFNET = $1 ]; then
        echo "error bij surfnet!"
        cat dns-start > ipv6.include
        cat dns-bit >> ipv6.include
        scp ipv6.include andree@panoramix:/export/home/andree/
    fi

    if [ $BIT = $1 ]; then
        echo "error bij bit!"
        cat dns-start > ipv6.include
        cat dns-surfnet >> ipv6.include
        scp ipv6.include andree@panoramix:/export/home/andree/
    fi
fi

#probleem weer opgelost dns terug zetten
if [ $2 = "ok" ];then

    if [ $SURFNET = $1 ]; then
        echo "alls weer ok bij surfnet!"
        cat dns-start > ipv6.include
        cat dns-surfnet >> ipv6.include
        cat dns-bit >> ipv6.include
        scp ipv6.include andree@panoramix:/export/home/andree/
    fi

    if [ $BIT = $1 ]; then
        echo "alles weer ok bij bit!"
        cat dns-start > ipv6.include
        cat dns-surfnet >> ipv6.include
        cat dns-bit >> ipv6.include
        scp ipv6.include andree@panoramix:/export/home/andree/
    fi
fi
```

Op panoramix staan nu de juiste DNS include zone files.

Hier dient dan de dns server herstart te worden, dit kan ook automatisch vanuit het script worden gedaan.

Het script ifV6test samen met telnet-router maakt gebruik van meerdere input files, een voorbeeld van een dergelijke file ziet er als volgt uit:

### Cat input-tmp

```
username
password
enable
enabla-pass
conf t
interface fa 0/1
ipv6 nd prefix-advertisement 2001:610:140:2202::/64 600 300 autoconfig
exit
exit
exit
```

## cat input2

```
username
password
enable
enabla-pass
sh run | include ipv6 nd
exit
```

## rc.multi6

Het geheel kan worden bestuurd mbv. een rc script, dit script heeft de volgende opties:

```
./rc.multi6
Usage: {start|stop|restart|status|showlog|showrtr}
```

De werking van het script is als volgt:

```
#!/usr/local/bin/bash

case "$1" in
  start)
    echo -ne "Starting multihoming script\n"
    /home/andree/v6/ipv6test > logfile &
    ;;

  stop)
    echo -ne "Stopping multihoming script\n"
    kill `ps aux |grep ipv6test |grep -v grep |awk '{print $2}'`
    kill `ps aux |grep rc.multi6 |grep -v grep |awk '{print $2}'` >>/dev/null
    ;;

  restart)
    sh $0 stop
    sh $0 start
    ;;

  status)
    status=down
    ps -aux |grep /home/andree/v6/ipv6test |grep -v grep && status=up > /dev/null
    echo -ne "Status of ipv6test script is $status \n"
    ;;

  showlog)
    lines=`cat /home/andree/v6/logfile |wc -l`
    if [ $lines -le 20 ]; then
      cat /home/andree/v6/logfile;
    else less /home/andree/v6/logfile
    fi
    ;;

  showrtr)
    cd /home/andree/v6
    ./telnet-router idfix < input2
    ;;

  *)
    echo "Usage: {start|stop|restart|status|showlog|showrtr}"
    exit 1
    ;;
esac
exit 0
```