Institutionen för teknik, matematik och datavetenskap – TMD

EXAMENSARBETE
2004:DS24

Introduktion av IPv6 i en medelstor organisations IPv4 nätverk

Amir Palic
Pekka Wikman

HÖGSKOLAN
TROLLHÄTTAN • UDDEVALLA

EXAMENSARBETE

Introduktion av IPv6 i en medelstor organisations IPv4 Nätverk

Amir Palic
Pekka Wikman

Sammanfattning

Syftet med denna rapport är att se hur IPv6 fungerar och om tekniken är mogen att implementeras i ett redan fungerande IPv4-nätverk.

I den första delen av rapporten undersöks anledningen till IPv6 uppkomst, hur den är uppbyggd och vilka förbättringar som gjort gentemot IPv4. Rapporten tar även upp olika Transitions metoder som används för tillfället för att möjliggöra samexistensen mellan IPv6 och IPv4.

Nästa del av rapporten behandlar hur installationen och konfiguration av de klienter, servrar och routerar som gjorts. Här förklaras även hur nätverkstotologin är tänkt samt vilka operativsystem och programvara som kommer att användas i nätverket.

För att kunna fastställa att nätverket fungerar som det är tänkt görs tester av nätverket och dess komponenter. I de fall där klienter, servrar eller routerar inte fungerar finns en utförlig beskrivning över felsökningen som gjorts. Det rapporten visar är att sätta upp ett IPv6-nätverk är inte svårare än att sätta upp än ett IPv4-nätverk. Men att få IPv6-nätverket att kommunicera med IPv4-nätverket och tvärtom kan medföra problem.

Utgivare:	Högskolan Trollhättan/Uddevalla, Institutionen för Teknik, Matematik och Datavetenskap, Box 957, 461 29 Trollhättan Tel: 0520-47 50 00 Fax: 0520-47 50 99 Web: www.htu.se		
Examinator:	Stefan Christiernin, HTU		
Handledare:	Robert Andersson, HTU		
Huvudämne:	Datavetenskap	**Språk**:	Svenska
Nivå:	Fördjupningsnivå 1	**Poäng**:	10
Rapportnr:	2004:DS24	**Datum**:	2004-08-02
Nyckelord:	IPv6, NAT-PT, DNS6, Nätverk, Implementation,		

DEGREE PROJECT

Implementation of IPv6 in a middle-sized organisation IPv4 network

Amir Palic
Pekka Wikman

Summary

The purpose with this report is to see if IPv6 works and if it's ready to be implemented into a working IPv4 network.

In the first part of the report IPv6 is examined, why it became to be, how it works and witch improvements have been made over IPv4. The report also explains the different translation mechanisms that are used to allow IPv6 and IPv4 networks to coexist.

The next part of the report deals with the installation and configuration of the various clients, servers and routers. An explanation of how the network topology is planed to be and witch operating systems and software is to be used in the network.

To be sure that the network is operating as planed series of test are made. In the cases where clients, servers or routers doesn't work a detailed description of search for reason for failing is reported. What this report shows is that setting up an IPv6 network isn't harder then setting up an IPv4 network. But getting an IPv6 network to communicate with an IPv4 network can be troublesome.

Publisher:	University of Trollhättan/Uddevalla, Department of Technology, Mathematics and Computer Science, Box 957, S-461 29 Trollhättan, SWEDEN Phone: + 46 520 47 50 00 Fax: + 46 520 47 50 99 Web: www.htu.se		
Examiner:	Stefan Christiernin, HTU		
Advisor:	Robert Andersson		
Subject:	Electrical Engineering	**Language**:	Swedish
Level:	Advanced	**Credits**:	10 Swedish, 15 ECTS credits
Number:	2004:DS24	**Date**:	August 2, 2004
Keywords	IPv6, NAT-PT, DNS6, Network, Implementation		

Förord

Detta examensarbete är en del av Data- & Systemvetenskapligt program 120p som vi genomfört på Institution för Informatik & Matematik på Högskolan Trollhättan/Uddevalla.

Det finns några personer vi skulle vilja tacka lite extra för den hjälp dom gett oss under arbetets gång.

Vi vill tacka Mats Lejon som tålmodigt hjälpt oss med många uppslag och idéer samt hjälp oss med diverse nätverksrelaterade problem. Vi vill tacka Christian Jiresjö för all hjälp vi har fått med diverse Linux relaterade problem. Ett stort tack går även ut till Micael Ebbmar för all hjälp med, allt från växel till kaffe till nätverks och Linux relaterade frågor. ”We bugged you with our bugs and you helped us!”

Vi vill även tacka våra handledare Stanislav Belenki och Robert Andersson för den hjälp vi fått med rapportskrivningen.

“failure depended on failure to locate failure in the network thus failure was eminent”

Pekka Wikman

Amir Palic

Innehållsförteckning

Bilagor

Nomenklatur

3G	Står för Tredje Generationen mobil teknologi. De tjänster som associeras med 3G är dess möjlighet att både kunna överföra röst samtal samt nedladdning av data (e-post, spel, filmer, musik osv.).
AH	Se Authentication Header för mer information.
ALG	Application Level Gateway. Används av NAT/NAT-PT för att tillåta klienter på utsidan/insidan initiera applikationsspecifika sessioner med klienter på insidan/utsidan.
Authentication Header	En av säkerhets mekanismerna inom IPsec. IPsec är en standardiserad del av IPv6.
Borderrouter	Router som är belägen mellan två nätverk.
DHCP	Dynamic Host Configuration Protocol. Används för att kunna tilldela dynamiskadresser till klienter i ett nätverk.
DNS	Domain Name System (Namnserver). Används i huvudsak till att översätta datorers IP-adress till datorn namn och tvärtom.
DNSv6	IPv6 anpassad DNS.
DSTM	Transitons mekanism där klienter med dubbla IP-stackar används.
Dynamic Host Configuration Protocol	Se DHCP för mer information.
Encapsulated Security Payload	En av säkerhets mekanismerna inom IPsec. IPsec är en standardiserad del av IPv6.
ESP	Se Encapsulated Security Payload för mer information.
EXT3	Third Extended Filesystem. Ett journalförande filsystem. Börjar bli mer och mer populärt inom Linux/Unix världen då det går att uppgradera till EXT3 från EXT2 som inte är journalförande.
ICMP	Se Internet Control Message Protocol för mer information.
IETF	Internet Engineering Task Force. Har ansvar för att ta fram och godkänna nya standarder inom Internet och nätverk.
Internet Control Message Protocol	Ett av protokollen i TCP/IP och som hanterar övervakningsinformation och felmeddelanden.
Internet Protocol version 4	Den nuvarande standarden. Kommer i framtiden att ersättas av IPv6.
Internet Protocol version 6	Är ämnad att ersätta den nuvarande standarden IPv4. Största anledningen till byte av standard är den otroligt mycket större adress rymden som IPv6 ger.
IP	Internet Protokoll. Paketförmedlande protokoll. Används av avsändare och mottagare klienterna för att kommunicera över nätverk.

IPsec	Säkerhets mekanism för IP protokoll.
IPv4	Se Internet Protocol version 4 för mer information.
IPv6	Se Internet Protocol version 6 för mer information.
Key Management Protocol	En förhandligs fas inom SA som bland annat etablerar nycklar mellan noder samt autentiserar noder.
KMP	Se Key Management Protocol för mer information.
MD5	standardiserad krypterings algoritm. 128 bits kryptering.
NAT	Network Adress Translator. Används för att låta ett stort nätverk vara uppbyggd med privata ip adresser och endast ha några få publika adresser som används vid kommunikation med omvärlden.
NAT-PT	Network Address Translation - Protocol Translation. Används för att tillåta trafik mellan IPv6 och IPv4 nätverk.
Network Adress Translator	Se NAT för mer information.
QoS	Se Quality of Service för mer information.
Quality of Service	rar till de protokoll som kör ovanpå IP lagret och tillgodoser tjänster som prioritering av paket.
SA	Se Security Association för mer information.
Security Association	Regler som de olika delarna inom IPsec måste uppfylla för att paketet skall kunna anses vara säkert.
Security Parameter Index	Används för att kontrollera paket som skickas via ett nätverk inte har blivit modifierade.
SIIT	Stateless IP/ICMP Translation Algorithm. Transitionsmekanism mellan IPv6-IPv4.
SPI	Se Security Parameter Index för mer information.
TCP	Förbindelse orienterad protokoll.
TEP	Se Tunnel End Point för mer information.
Tunnel End Point	Används inom DSTM. Det är hit paket skickas när de skall ut från nätverket.
UDP	Förbindelselöst protokoll, använder sig av IP

1 Inledning

I och med att persondatorer och inbyggda system kostar allt mindre kommer allt fler hem att ha fler och fler datorer i sig. Det finns många visioner om hur alla datorer i samhället skall kunna kommunicera med varandra. Man kommer att kunna fråga kylskåpet med hjälp av en mobiltelefon vad som finns i den och vad som skulle behöva köpas. Prata och se varandra när man talar i mobiltelefonen är redan möjligt med *3G (Tredje Generationen mobil telefoni)*. Men för att de och många andra visioner skall kunna slå in måste man bli av med en stor bromskloss, nämligen bristen på IP-adresser.

Detta är inget man märker av som vanlig användare. Det fungerar att koppla upp sig på nätet med datorn och allt verkar fungera. Men ofta är man uppkopplad med hjälp av nödlösningar (se Brist på IP-adresser) för att komma förbi bristen på adresser. Om man sedan tar i beaktning att alla 3G telefoner behöver en unik IP-adress där dessa nödlösningar inte egentligen är ett alternativ. Då förstår man att något måste göras. Och lösningen är *IPv6 (Internet protocol version 6*). IPv6 ger även inbyggd säkerhet och QoS (*Quality of Service*).

IPv6 är ingen direkt ny standard. Men den har inte fått någon större spridning. Detta arbete handlar om implementering av ett IPv6 nätverk (nätverk där endast IPv6 protokollet används) som skall finnas vid sidan av HTU: s (Högskolan Trollhättan/Uddevalla) existerande nätverk. En utvärdering av översättningsmekanism *NAT-PT (Network Address translator - Protocol Translator)* mellan IPv6 och *IPv4 (Internet protocol version 4)* kommer att göras. IPv6 stödet hos Bifrost router/brandvägg programvara baserad på Linux undersöks. Tester kommer att göras för att säkerhetsställa att allt verkligen fungerar som det skall, testerna görs med Iperf (Iperf är programvara för nätverkstester) och *DNS*-namnuppslagningar samt trace6 och ping6. Allt för att HTU skall kunna ta steget mot ett helt fungerande IPv6 nätverk.

I kapitel 3 beskrivs hur IPv4 och IPv6 fungerar samt de olika transitions teknikerna. Kapitel 4 dokumenterar installationen av mjukvara och operativsystem. Kapitel 5 redovisar testerna samt resultaten. Slutsatser samt diskussion redovisas i kapitel 6.

1.1 Bakgrund

IPv4 blev standard för ARPANET 1981. [1] Det har varit en standard som med hjälp av endast små modifikationer lyckats att bestå i över 20 år. Men med ett explosionsartad intresse för Internet på 90-talet började man inse att IPv4-adresser kommer ta slut. Så man började ta fram en ny standard kallad IPv6. Om ni undrar vart IPv5 tagit vägen så var det ett experimentellt protokoll kallad ST2. Men den har sedan länge övergivits. [2] IPv6 ger enormt många fler IP-adresser än vad IPv4 kan ge. IPv4 har $4 * 10^9$ unika IP adresser medan IPv6 har hela 3.4×10^{38} unika adresser.

Då IPv6 fortfarande är en ganska oprövad standard är många tekniska institutioner intresserade av den för att se hur den fungerar och börja förberedelser för långsamt skifte till IPv6.

1.2 Problembeskrivning

Arbetet omfattar installation och test av ett IPv6-nätverk (nätverk där all data trafik är IPv6-datatrafik) som skall kunna kommunicera med IPv4-nätverk, IPv4 till IPv6 och vice versa genom översättnings mekanism/er. Flera operativsystem (Linux, Microsoft Windows 2000 och Microsoft Windows server 2003 standard edition) installeras på klientmaskiner i IPv6-nätverket. Klienter används till att testa nätets funktionalitet, samtidigt fås en bild av hur komplicerat det är att installera operativsystemen med endast IPv6 stöd.

Transitions mekanism som kommer användas för kommunikation med IPv4-nätverket NAT-PT. Två utvärderingar av NAT-PT görs:

- en Linux-Server
- Cisco router 7507

En test av Bifrost (router/brandvägg program) i IPv6-nätverk görs, där Bifrosts IPv6 kompatibilitet testas. Utvärdering av Linux-Server NAT-PT är nödvändig då dess funktioner behövs i ett nätverk och kan ge ett alternativ till dyra nätverkslösningar. Vid den slutliga utvärderingen av arbetet bör man veta om och hur introduktion av IPv6 med transition mot IPv4 är möjlig på ett tillfredsställande sätt.

1.3 Avgränsningar

Tester av transitionsmekanismer kommer att begränsas till en Linux NAT-PT (minimal Red Hat 9) och en Cisco NAT-PT (Cisco 7507 router). Men fler transitonsalternativ kommer att tas upp och beskrivas i detta arbete.

DNS tester kommer att ske från Windows 2000, Red Hat 9 och Windows server 2003 standard edition maskiner. Detta för att testa operativsystem med inbyggt IPv6 stöd och en utan (Windows 2000).

Då Bifrost som standard stänger av all IPv6-trafik kommer endast trafik nödvändig för fastställande av dess IPv6 kompabilitet att tillåtas. Detta gör att nätverket inte behöver säkerställas mer än Bifrost gör som standard.

1.4 Mål

Målet med arbetet är att implementera ett IPv6-nätverk som klarar av att kommunicera med IPv4-nätverk genom NAT-PT. NAT-PT utvärderas på både vanlig dator med operativ systemet Linux och NAT-PT mjukvara installerad och Cisco 7507 router. Linux NAT-PT testas för att se om mjukvaru NAT-PT fungerar tillfredsställande då den

i så fall kan vara en kostnadseffektivt val framför routrar som Cisco 7507 router. Test av Bifrosts IPv6 kompatibilitet testas. Arbetet bör vid sitt slut utgöra start för skifte från IPv4 till IPv6.

2 metod

2.1 Val av metod

Valet av metoder i detta arbete är studier av befintliga arbeten och intervjuer, då kunskap om hur de olika delarna fungerar i ett IPv6-nätverk är viktigt. Samt hur kommunikationen med ett IPv4-nätverk skall gå till, och vilka skillnader som det finns mellan IPv6 och IPv4. Detta så man får en bild av vad som måste göras och hur det skall gå till redan innan man börjar. Arbetet kommer till en början att inrikta sig på studier av befintlig litteratur, läsa vetenskapliga arbeten och information från Internet. Sedan kommer arbetet att övergå mer i ett praktiskt arbete då all programvara samt nätverket skall byggas. Och slutligen skall allt testas och verifieras.

2.2 Informationsinsamling

Informationsinsamling kommer att riktas in på vetenskapliga arbeten, olika standarder (RFC :s) och aktuella artiklar. Då detta är ett område som utvecklas snabbt och nya teniker uppkommer konstant är dessa mera aktuella än vad böcker har en tendens att vara. Dock kommer böcker att användas för att finna djupare kundskaper innom olika områden så som IPv6 uppbyggnad.

2.3 Intervjuer

Dessa kommer att bli kompleterande till den ovan nämda informationsinsamlingen. Intervjuer kommer att användas för att få feedback från personer som arbetat med en viss teknik, som är intresserant för arbetet. Intervjuer kommer att ske via E-post.

Vi kommer även att ha kontakt med nätverksansvariga på HTU för att få information vi behöver samt disskutera hur och vad som skall testas.

2.4 Val av analysmetod

Det arbetet är inriktat på är hur bra transitionsfunktioner fungerar, hur stor belastning dom klarar av samt se hur pass användarvänligt det är med ansende på kommunikationen mellan de olika näten. För att en teknik skall anses fungera tillfredställande kommer den att behöva fungera närmast felfritt.

Tester kommer även att göras för att se hur bra DNS förfrågningar fungerar mellan dom olika näten och se hur bra en DNSv6 fungerar i jämförelse med en DNSv4. Tester som kommer att göras är mer ingående beskrivna i testplanen som gjorts, se kapitel 5.

2.5 Svårigheter i arbetet

Det finns flera svårigheter man kan stöta på under ett arbete som detta. Detta är några av dessa:

- Människor man intervjuar inte svarar på e-post.
- Bristfälliga installations guider.
- Svårigheter med att finna rellevant information.
- Hitta programvara för att kuna utföra testerna på ett tillfredställande sätt.
- Inte få programvara, hårdvara eller något annat som behövs för att nätverket skall fungera.
- Problem med att få operativsystem och/eller mjukvara att fungera i ett native IPv6-nätverk.

3 Förstudie

IPv4 har varit ett oerhört lyckat protokoll. Genom att vara skalbart och kunna utökas från nätverk på några enstaka klienter till ett världsomspännande nätverk har den stått sig bra. Men allt eftersom nya behov uppstår börjar IPv4 visa tecken på ålderdom. Så tidigt som på slutet av 80-talet började man förstå att adressrymden som IPv4 ger inte kommer att kunna tillgodose det behovet som skulle komma. Dock är detta endast en del i varför en ny standard behövs.

IPv6 är den standard som skall efterträda IPv4. Några av de viktigaste punkterna den nya standarden åtgärdar är brist på IP-adresser, säkerhet, prestanda och bristen på autokonfiguration. [3]

3.1 Brist på IP-adresser

Bristen på IP-adresser har varit känd länge. Dock har men kunnat lindra påverkan genom olika nödlösningar, dynamisk adress tilldelning och *NAT (Network Adress Translator)*. [4]

Dynamiskadresstilldelning är en teknik där klienten ansluter sig på nätverket och skickar ut meddelandet på nätverket om att den vill ha en IP-adress. Vartefter den får sig tilldelad en IP-adress ur en pool av adresser. Adresstilldelningen görs av *DHCP* (*Dynamic Host Configuration Protocol*). Klienten behåller den tilldelade adressen så länge den är ansluten mot nätverket. Dock är adressen reserverad en viss tid efter att klienten försvinner från nätverket. Detta om klienten åter skulle bli aktiv så ska den kunna få samma IP-adress. Tiden adressen är reserverad varierar efter inställningen på DHCP servern. Hur adresser delas ut kan ställas in väldigt flexibelt. Datorer kan alltid få samma IP-adress baserat på MAC-adresser (MAC-adress är nätverkskortets maskinadress, den är alltid den samma och unik för varje nätverkskort). [5]

NAT är en teknik som tillåter klienter med privata IP-adresser att nå ut till Internet. NAT fungerar genom att nätverket med privata IP-adresser finns på ena sidan och Internet/nätverk med publika IP-adresser på den andra sidan. Nätverket där klienter med privata adresser finns kommer i fortsättningen kallas privat nätverk och nätverk med klienter som har publika adresser kommer att kallas publikt nätverk. När klienter från det privata nätverket vill skicka datapaket till ett publikt nätverk passerar all trafik genom NATen. I NATen översätts den privata adressen till en slumpvis vald publik adress ur en pool med publika IP-adresser som NATen har reserverade för översättning av IP-adresser. Anta att en klient från det privata nätverket vill skicka datapaket till en klient i ett publikt nätverk. Paketet som skickas iväg kommer till NATen där den översätts. Efter översättningen skickar NATen ut datapaketet på det publika nätverket som då innehåller den publika IP-adressen som avsändaradress. NAT mekanismen sköter översättningen genom att ha tabeller över översättningar den gör. När det iväg skickade paketet tagits emot av mottagaren så skickas svaret som innehåller den översatta adressen som destinationsadress tillbaka till NATen. NATen ser efter om destinationsadressen på det mottagna paketet finns i dess översättnings tabell, när det hittas så översätts destinationsadressen till den privata adressen som det ursprungliga datapaketet skickades ifrån. Paketet skickas därefter ut på det privata nätverket och den mottas av klienten som ursprungligen skickade iväg paketet. Denna procedur upprepas så länge datapaket skickas mellan klienterna. Det finns flera nackdelar med denna teknik. Klienter på det publika nätverket kan inte påbörja en datasession med klienter som ligger innanför NATen i det privata nätverket. NATen kan dock konfigureras så att servrar kan ligga i det privata nätverket, i sådana fall använder sig NATen av portnummer för att veta vilken privat adress den ska skicka paket till.

Exempel:

En webbserver som ligger i det privata nätverket kör sin tjänst på port 80, om NATen tar emot data paket som ska till port 80 så vidareskickas dom paket till serverns privata adress. Antal samtidiga datasessioner mellan det privata och publika nätverk kan aldrig överskrida mängden publika adresser som NATen har i sin översättnings adresspool.

End-to-end kommunikation är inte möjlig då båda klienter tror sig skicka datapaket till NATen. En av dom största nackdelarna är att många applikationer inte är kompatibla med NAT tekniken [6]

3.2 Säkerhet

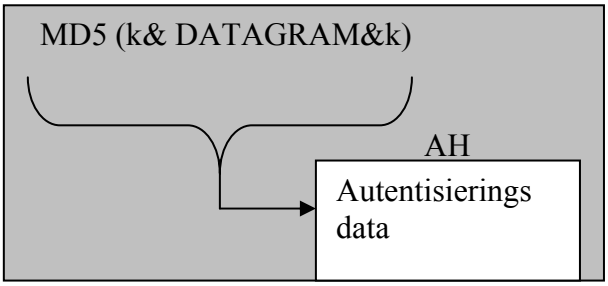
För att Internet ska bli så säker och tillförlitlig som möjligt så jobbar *Security Area of IETF (Internet Engineering Task Force)* med att förbättra Internets säkerhet, IPv6 är resultat av deras arbete. [7] Säkerhets mekanismer som det finns stöd för i IPv6 är: integritet, konfidentialitet, autentisiering på nätverksnivå (IP-datagram). För att lyckas uppnå säkerheten har den nya standarden två nya security payloads (Fält i IPv6-hedern

ämnad att nyttjas av säkerhetsprotokoll, se fig.3) AH (*Authentication Header)* och ESP (*Encapsulated Security Payload)*.

- AH, Autentierings och integritets mekanismer som upptäcker om ett paket har blivit ändrad under själva överförningen
- ESP, konfidentialitets mekanism som ser till att endast behöriga mottagare får tillgång till information.

3.2.1 Authentication Header

Fungerar genom att AH läggs till som payload (tillägg) i IP-datagrammet på de paket som behöver autentisiering. Endast noder som kommunicerar med varandra bryr sig om AH, alla andra noder som IP-datagram med AH passerar förmedlar de vidare utan att bry sig om. Med andra ord så kan IP-datagram passera IPv4-noder utan problem. AH skyddar hela datagrammet. Den gör det genom att få fram en hasch av datagrammet med *MD5* (MD5 är en standardiserad krypterings algoritm) och en nyckel (k). Nyckeln finns dock med på två ställen. Se fig 1.



Figur 1. Authentisierings data beräkning [8]

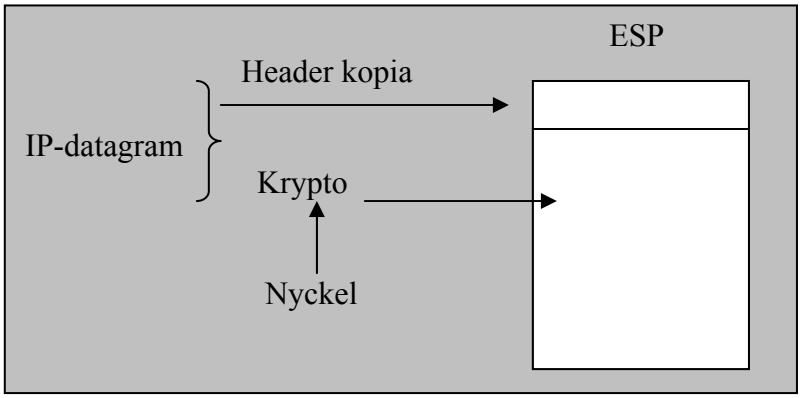
När mottagaren får paketet så validerar den paketet och endast då paketet blivit godkänd tas den emot. [8]

3.2.2 Encapsulating Security Payload

Vid sekretessbehov används ESP (Encapsulating Security Payload). Den är ämnad för konfidialitet och integritet på IP-lagret. Med ESP ökar kravet på datorkraft då krypteringen kräver mer av hårdvaran än AH. ESP är uppbyggd av klartext och krypterad text. Klartext är information om destinationen och dekrypteringen av ESP och Krypterade delen är den skyddade delen. ESP har två funktionella sätt: *Tunnel Mode* och *Transport mode*. [8]

- *Tunnel Mode*, sändaren Tar hela IP-datagrammet och krypterar det med den kända nyckeln. Den krypterade datan läggs in i ett ny IP-datagram som en ESP payload. Nya datagrammet sänds till mottagaren där endast destinationsfält är i klartext. Mottagaren kastar det mottagna datagrammet utom krypterade ESP payload och därefter dekrypterar ESP payload med den kända nyckeln. Se fig. 2. [8]

- *Transport Mode*, i Transport Mode tas hela paketet från *TCP* (Transmisson Controll Protocol) lagret och samma procces som i Tunnel Mode utförs. Skillnaden är att hela transportlagrets paket finns inne i ESP. [8]



Figur 2. ESP beräkning [8]

3.2.3 Säkerhets förhandling

För att AH, ESP samt alla säkerhets protokoll som kan dyka upp i och med den nya IP standarden (IPv6) ska kunna fungera måste dom kunna komma överens om vissa parametrar som till exempel: Funktionalitet, nycklar och så vidare. För att lyckas med detta så finns SA *(Security Association)* där förhandlig sker i två faser.

- Fas 1, IKE (Internet Key Exchange), Säkerhets parametrar förhandlas, etablerar nycklar, autentisierar noder och resulterar i ett nytt SA med säkerhets parametrar som möjliggör en säker informations utbyte inför fas 2 [8]
- Fas 2, SA, Förhandlar säkerhetsparametrar åt säkerhetsprotokoller som *IPsec*. Alla fas 2 förhandlingar måste förgås av fas1. IKE dokumentation rekommenderar att fas1 förhandlingar används till fler fas 2 förhandlingar för att på så sätt minimera den tid det tar för att sätta upp en specifik SA. [8]

SA är en samling regler som måste uppnås av nya säkerhetsprotokoll. För att SA ska kunna användas måste dom kommunicerande noderna komma överens om hemliga nycklar som ska användas. Detta sköts av KMP *(Key Management Protocol)*, den skyddar SA *Negotiation Protoco*l. [8][9][10]

3.2.4 IPsec

Ipsec arkitekturen är designat av IETF för att fungera på nätverkslagret, vilket medför att den kan användas av applikationer på alla lager ovanför. *IP* ligger på nätverkslagret (enligt OSI modellen) ensamt och finns endast i nätverkslagret måste all IP trafik förr eller senare passera detta lager. Och då kan man göra trafiken säker, oavsett vilken programvara man använder sig av. IPsec är kompatibelt både med IPv6 och med IPv4. I

IPv6 läggs säkerhetsmekanismer i extension headern och i IPv4 ligger dom i options fält. Se nedan för mer information om extention headers och options fält.

Säkerhet uppnås genom att kombinera flera olika säkerhetsmekanismer. AH tillgodoser autentisering och integritet åt paketet. ESP tillgodoser kryptering samt datainkapsling. För mer information om AH och ESP läs ovan.

När paketet skickas genom ett nätverk sätts nya headers till och gamla tas bort. I varje ny header finns en *SPI (security parameter index, används för datapaket kontroll)*. Den använder sig av säkerhetsprotokoll samt mottagaradressen för att få ett unikt värde. Detta för att kunna ge varje SA ett unikt värde. För kryptering av data används KPI. Se ovan för mer information. [9][10][11]

3.3 Headers

IPv4 designades för över 20 år sedan och de beslut man tog då var baserade på vad man hade för behov på den tiden. Många av dessa beslut har ingen praktiskt mening längre. Som man kan se på figur 3 och figur 4 har antalet fält i IPv6 reducerats i jämförelse med IPv4. I figurerna ser man att man har gått ifrån IPv4 12 olika fält till endast 8 i IPv6. De fält som är borttagna är markerade i figur 3. IPv6 paket har alltid en fast storlek till skillnad från IPv4 paket, de kan processas mycket snabbare av olika noder på dess väg genom nätverk.

Version	**IHL**	**TOS**	**Total length**
Identification	**Flags**	**Fragment offset**	
TTL	**Protocol**	**Header checksum**	
Source IP address			
Destination IP address			
Options			
Data			

Figur 3 IPv4 datagram [12]

Version: 4 bits lång. Version av protokollet som används.

IHL (Internet Header Length): 4 bits lång. Beskriver hur stor IP-headern är.

TOS (Type Of Service): 8 bits lång. Beskriver hur paketet skall handskas av olika nätverkskomponenter.

Total Length: 16 bits långt. Anger hela paketets storlek inklusive headern.

Identification: 16 bits långt. Ger varje paket i en ström en unik identifierare. Detta för att kunna sätta ihop meddelandet igen på rätt sätt om den blivit fragmenterad vid överföringen.

Flags: 4 bits lång. Dessa flaggor kan sättas om man har speciella önskemål om att paketet absolut inte eller att den får fragmenteras under sin väg genom nätverket.

Fragment offset: 12 bits lång. Om paketet blir fragmenterat anges dess rätta position i strömmen med detta värde.

TTL: 8 bits lång. . För varje router som paketet passerar sänks Hop limit värdet med ett. När paketet når noll kastas den.

Protocol: 8 bits lång. Anger vilket protokoll som finns i Data delen av paketet.

Header Checksum: 16 bits långt. Används för att kontrollera IP-headerns validitet. Men genom att TTL ändras med varje hop räknas detta värde om vid varje hop som görs.

Source IP adress: 32 bits IP-adress till avsändaren.

Destination IP adress: 32 bits IP-adress till mottagaren.

Options: Här kan man ge fler instruktioner för paketets hantering.

[13][14]

Version	**Trafic Class**	**Flow lable**
Payload length	**Next header**	**Hop limit**
Source adress		
Destination adress		
Data		

Figur 4. IPv6 Datagram

Version: 4 bits lång. Beskriver vilken version av IPv6 som används.

Trafic Class: 8 bits lång. Prioritets värde för paketet.

Flow lable: 20 bits stort. Används för att specificera speciella routing kommandon för en sekvens av paket från avsändaren till mottagaren.

Payload Length: 16 bits lång. Anger datafältets storlek.

Next header: 8 bits lång. Identifierar nästa header som kommer efter IPv6-headern.

Hop limit: 8 bits lång. För varje router som paketet passerar sänks Hop limit värdet med ett. När paketet når noll så kastas den.

Source adress: 16 bytes långt. IPv6 avsändaradress.

Destination adress: 16 bytes långt. IPv6 mottagarens adress.

[15]

Tabell 1 nedan visar vad som ändrats mellan de två protokollen. Vad fält heter i IPv4 och vad de heter i IPv6-datagram samt vilka ändringar/åtgärder man har gjort.

Tabellen nedan visar även hur och vart transitons metoder lägger in de olika fälten vid översättning. För mer information om översättning se **kapitel 3.4**

Ipv4	**Ipv6**	**Åtgärd**
Version	Version	Ingen åtgärd
IHL	---	Helt borttagen ut IPv6 då IPv6 har en fast header storlek.
TOS	Trafic class	Ersätts av Trafic class headern.
Total Length	Payload length	Total Legth fältet har gjorts om till Payload Legnth. Man räknar inte med header
Identification	Tillval	Dessa finns kvar men finns nu som tillval.
Flags	Tillval	Dessa finns kvar men finns nu som tillval.
Fragment offset	Tillval	Dessa finns kvar men finns nu som tillval.
TTL	Hop limit	Här är det endast namnet som ändrats för att förtydliga vad den gör.
Protocol	Next header	Har samma funktion i båda protokollen.
Header Checksum	---	Då IPv6 har en fast header storlek på 40 bytes behövs den inte kontrolleras.
Source IP address	Source adress	Likvärdiga i båda.
Destination IP Adress	Destination Adress	Likvärdiga i båda.
Options	Extension headers	Man har plockar bort options och satt in Extension headers. Läs mer om dessa i stycket nedan.

Tabell 1. Visar de ändringar som gjort mellan IPv4 och IPv6 headers.

3.3.1 Extension Headers

Dessa tillval i IPv6 är istället för Options fältet som finns i IPv4. Options fältet i IPv4 ersattes på grund av att den uppfattas som alldeles för svåranvänd, dessutom måste varje nod vid hopp analysera all information som finns i options fältet, vilket gör att behandlingen av paketet tar längre tid än nödvändigt. I IPv6 har man löst detta genom så kallade extension headers. Dessa ligger direkt efter IPv6-headern, routers som inte berörs av informationen ser dom som en del av data fältet och ignorerar dom. Detta gör att paketen skickas mycket effektivare. Det är i extension headers som säkerhets mekanismer kan ställas in. [16]

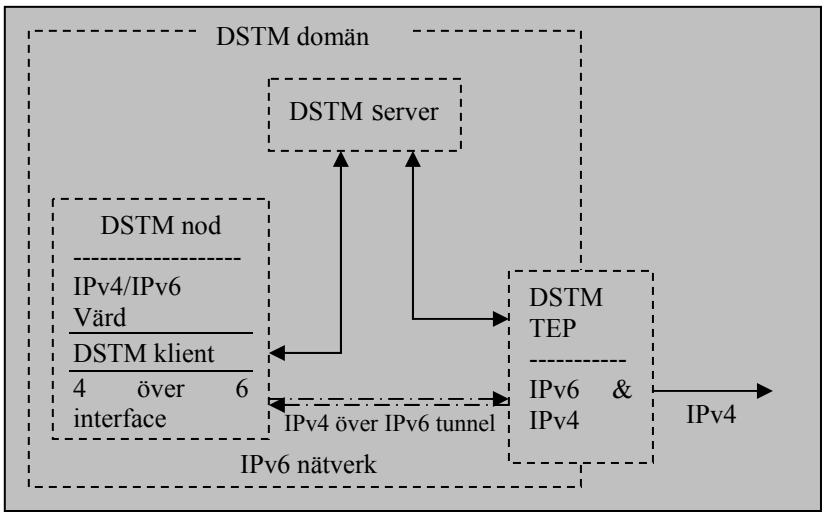
3.4 Transitions tekniker

IPv4 och IPv6 kommer att samexistera ett tag framöver men datorer och routrar med olika IP standarder kan inte kommunicera direkt med varandra. Lösningen blir att man får använda transitionstekniker för att möjliggöra kommunikation mellan IPv6 och IPv4. [17][18]

Det finns ett flertal olika standardiserade tekniker som kan användas för kommunikation mellan IPv4 och IPv6. Enligt studierna som gjordes för detta arbete bestämdes det att lösningen som passar bäst är NAT-PT. Även om NAT-PT är tekniken som används för detta arbete så finns flera olika tekniker beskrivna i texten nedan.

3.4.1 DSTM

DSTM (Dual Stack Transitions Mechanism) är transitions mekanism där alla värdar i IPv6 nätverket har dubbla IP-stackar, det vill säga både IPv4 och IPv6 stack. IP standard som används vid kommunikation mellan noderna bestäms av den mottagande noden, om noden inte klarar IPv6 standarden kommer IPv4 att användas. Målet med tekniken är att IPv6-noder i en IPv6 domän ska kunna kommunicera med IPv4-noder i en IPv4 domän utan att applikationer ska behöva anpassas till IPv6 standarden. I och med att maskiner med dubbla stackar resulterar i dubbla routing regler och dubbla brandväggar så tar DSTM itu med detta genom att tunnla IPv4-trafik över en IPv6 tunnel. På sådant sätt får man endast IPv6-datatrafik inom IPv6-nätverket samt att underhåll av nätverket blir enklare och routingutrustning behöver bara stödja en standard. Datatrafik ut från DSTM domänen måste passera genom en *TEP (Tunnel End Point)*. DSTM tekniken och komponenter beskrivs mer ingående i figur 5. [19]

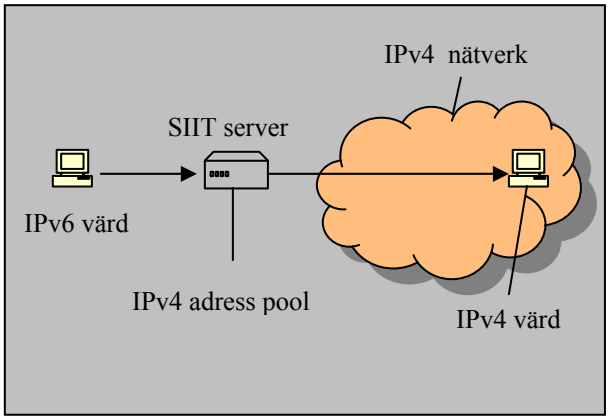


Figur 5. DSTM domän med DSTM komponenter, i DSTM domänen måste all datatrafik vara IPV6 trafik

- **DSTM server:** är en process, sköter IPv4 adressallokering till DSTM noder
- **DSTM klient:** är en process som hanterar den allokerade IPv4-adressen
- **DSTM nod:** Värd på nätverket med dubbla IP-stack (IPv4/IPv6), IPv4 över IPv6 tunnel och är en DSTM klient. Noden genererar enbart IPv6-trafik
- **DSTM TEP:** DSTM Tunnel end Point, IPv6-trafik som innehåller IPv4-datapaket routas hit för att kunna skickas till IPv4-nätverket.
- **IPv4 over IPv6:** Inkapsling av IPv4-datapaket i IPv6-datapaket, transporterar paket mellan noderna inom DSTM domänen

3.4.2 SIIT

SIIT (Stateless IP/ICMP Translation Algorithm) möjliggör kommunikation mellan IPv6 -noder och IPv4-noder där IPv6-noden inte har permanent IPv4-adress. IPv6-noden skickar datapaket till IPv4-noden genom SIIT servern genom att använda sig av en mappad IPv4-adress (mappad IPv4-adress är en IPv4-adress som refererar till en nätverksnod som inte använder IPv6-adress standard). [20] Transitionen som sker påminner om NAT (se kapitel 3 rubrik 3.1). Det vill säga att transitionen fungerar bara åt ett håll. IPv4-noder kan inte initiera kommunikation med IPv6-noder, se även figur 6.



Figur 6: IPv6 värden kommunicerar med IPv4 värden genom SITT servern , båda värdar använder endast en ip-adress standard.

Protokollöversättning görs genom att IPv6-headern översätts till IPv4-headern, routingutrustning på vägen mellan värdarna behöver inte köra dual stack läsningar. Även *ICMP (Internet Control Message Protocol)* medelanden mellan dom olika protokollen måste översättas [20]. För jämförelse mellan IPv4 och IPv6 headers se kapitel 3 rubrik 3.3.

Största nackdelen med tekniken är att den inte lämpar sig för lösningar där applikationer sänder adresser (applikationer som FTP). [18]

3.4.3 NAT-PT

NAT-PT transitionsmekanism fungerar som NAT vid adressöversättning. Undantaget är att NAT-PT även har möjlighet att fungera i båda riktningar och använder globala IPv4-adresser i sin adresspool (globala IPv4-adresser är registrerade IP-adresser som används på Internet där varje adress är unik). NAT beskrivs i kapitel 3 rubrik 3.1 i detta arbete. Adress översättning IPv4 till IPv6 och tvärtom är alltså möjlig. Protokollöversättning görs enligt SIIT modellen. [20]

All kommunikation mellan IPv4 och IPv6 noder sker genom NAT-PT och det rekommenderas att inga noder använder dubbla IP-stackar. Adressöversättningen görs genom att dom noder som kommunicerar mellan IPv6 och IPv4 domän passerar NAT-PT och där får globala IPv4-adresser. Sessionkontroll görs och IPv4 pooladresser binds till IPv6-adresser så att routingen blir helt transparent. NAT-PT stödjer inte applikationer som skickar IP-adresser i datapaket men detta åtgärdas med hjälp av *ALG* (*Applikation Level Gateway* – agent som tillåter kommunikation mellan IPv6 och IPv4 noder). För att NAT-PT ska kunna hantera sessionsinitieringar från IPv4 till IPv6 så måste en DNS-ALG användas och adresser till noder fås från *DNS* (Domain Name Server) genom DNS-namnförfrågningar. Alla noder som kommunicerar genom NAT-PT måste ha unika nätverksnamn i tillhörande nätverk.

Nackdelarna med att använda NAT-PT är som vid NAT och SIIT, all kommunikation mellan noderna måste ske genom NAT-PT, specifika ALGs måste användas beroende på vilken applikation som kommunicerar genom NAT-PT, end-to-end nätverklager säkerhet är inte möjlig, DNS översättning och DNSEC (DNS version 6 servrar signerar svar för att öka säkerheten i nätverk) från IPv4 noder till DNS server i IPv6 nätverket är inte möjlig då DNS-svar inte är signerade. [21]

3.4.4 6to4 tunneling

6to4 Tunneling är en teknik som tillåter ett IPv6-nätverk kommunicera med ett annat IPv6-nätverk över befintlig IPv4 nätverksstruktur. Detta görs genom att IPv6-datapaket kapslas in i IPv4-datapaket och skickas över IPv4-nätverk där dom av routrar behandlas som vanliga IPv4-datapaket. Först när paketen anländer till 6to4 routern hos destinations nätverk kommer ursprungliga IPv6-datapaketet att packas upp och skickas inom IPv6-nätverket till rätt IPv6-nod. Tekniken är tänkt att användas för att koppla ihop två grafiskt åtskilda IPv6-nätverk. Tekniken går dock att använda för att koppla enskilda IPv6-noder med IPv6-nätverk men då är Tunnel Broker teknik (teknik för automatiskt skapande av 6over4 tunnlar, standardiserad enligt rfc3053) att föredra. [22]

6to4 tekniken kräver inga ändringar i IPv4 routingen, en *borderrouter* måste finnas på IPv6-nätverk som kopplar nätverket mot en befintlig IPv4 struktur. Borderroutern måste vara en 6to4 router. Underhåll av routern och tunneln kräver lite av nätverksansvarig personal. [23]

3.5 Bifrost – brandvägg stöd för IPv6

”Bifrost Network Project har som mål är att utröna driftsäkerhet, nätprestanda, filterfunktioner, administration, datasäkerhet, skalbarhet och utvecklingsbarhet hos en router/brandväggen byggd på standard PC-hårdvaran med två (eller flera) Ethernetkort (helst DECs Tulip chip eller Intel 82543GC Gigabit Controller) och en flashdisk. Operativsystemet är en modifierad, minimal och optimerad Linux-distribution, med kärnan konfigurerad för routing och brandväggsfunktioner.” (ur Installationstips för Bifrost, http://www-alnarp.data.slu.se/bifrost/tips.html#H_intro)

Anledningen till att Bifrost (bifrost.slu.se) utvärderas är att ta reda på om den är tillräckligt kompatibel med IPv6 att kunna fungera i ett IPv6-nätverk. Bifrost Version 5.11 används då denna har stöd för IPv6. Valet av denna version beror på att det är den senaste när installationen gjordes samt att IPv6-tables stöd infördes så sent som i 5.10 och förhoppningsvis förbättrats ytterligare till 5.11. [24]

4 Installation

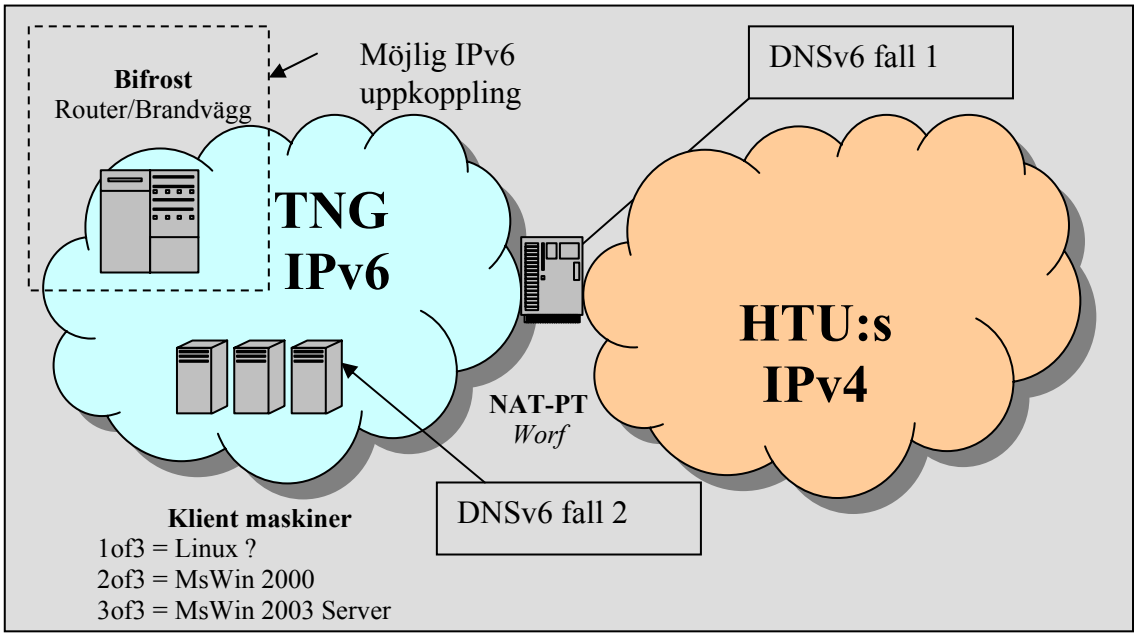
Nedan beskrivs installationen av de klienter samt servrar, routrar som gjorts i detta arbete. Även installation av nätverket redovisas, med dess design och fysiska topologi.

4.1 IPv6 nätverksdesign

IPv6-nätverket som byggs för detta arbete består av 3 klienter och en DNSv6. Möjlighet att ansluta nätverket till ett IPv6 native nätverk ges med borderrouter, routern är en Linuxmaskin som kör Bifrost den är samtidigt en brandvägg. För att sedan möjliggöra kommunikation med HTU:s IPv4-nätverk kommer NAT-PT att användas. NAT-PT kommer att köras på två olika sätt:

- Fall 1, NAT-PT körs på en Linuxmaskin
- Fall 2, NAT-PT körs på en Cisco 7507 router

Nätverks designen tydliggörs i figur 7.



Figur 7, IPv6 nätverket TNG(egna nätverket The Next Generation) med IPv6 klienter, DNS6, border router och NAT-PT kopplat mot HTU:s IPv4 nätverk.

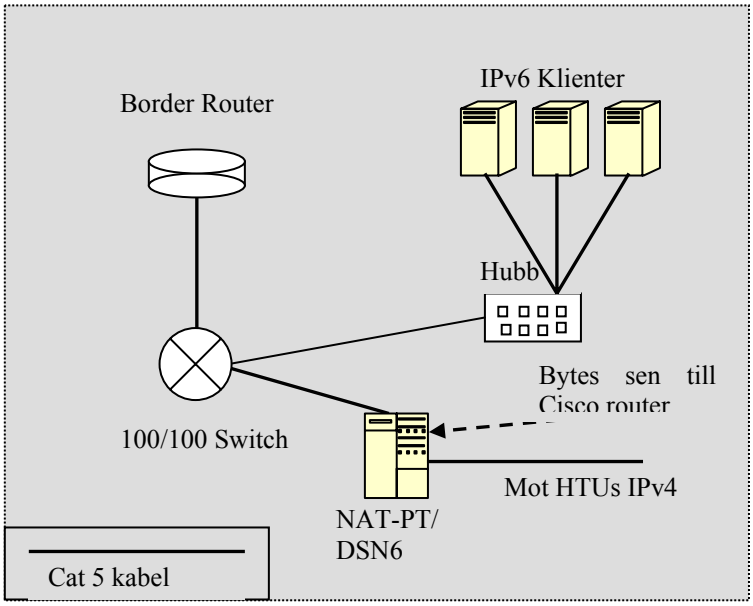
Arbetet utförs i flera steg.

1. Kabel dragning
2. Installation av IPv6 klienter
3. DNSv6 installation
4. NAT-PT installation (på en Linuxmaskin)
5. Installation och test av Bifrost
6. Test av nätverket och NAT-PT
7. Byte av NAT-PT, (körs på Cisco 7507 router)
8. Test av klienter, DNS samt NAT-PT mekanismen.

Alla steg förklaras ingående längre fram i arbetet i takt med att dom utförs.

4.2 Nätverkets fysiska topologi

I nätverket används Cat 5 kabel. Detta då nätverket har en Internet uppkoppling på 100 Mbit och all kommunikation inom nätverket och till HTUs IPv4-nätverk är 10/100 Mbit nätverk. En switch används i nätverket då routerns fiber till cat 5 adaptern inte är kompatibel med hubben som används i nätverket. Endast switchen kan inte användas på grund av dess fysiska placering.



Figur 8. Fysisk nätverks design

4.3 Klient installation

Klienterna som installeras i nätverket installeras med standard förvalda alternativ i den mån det är möjligt. Efter installation görs de uppgraderingar/inställningar som är nödvändiga för att få igång IPv6-stödet. Operativsystems installationer beskrivs mer ingående nedan. För varje installation redovisas IPv6-stödet med enkel test, testet innebär att operativsystem beroende kommando används för att visa nätverkets interface adress samt kommandot ping6 där klienten kontaktar sitt eget interface.

4.3.1 Windows 2000

Installation av Windows 2000 svensk version gjordes helt efter de förvalda inställningar som installationsprogrammet hade. Dock ändrades standard språket på tangentbords layouten från Engelsk till Svensk. Datorn fick namnet 1of3 för att kunna identifieras på nätverket. Klienten uppdaterades genom Microsofts Windows update med alla de senaste uppdateringarna (servicepack 4).

Då Windows 2000 inte har något inbyggt stöd för IPv6 måste ett utvecklingspacket för IPv6 (Microsoft IPv6 Technology Preview for Windows 2000) installeras. Vid installation varnade programmet för att fel version av servicepack var installerat. Det visade sig att utvecklingspaketet endast vad menat att fungera med servicepack 1. Genom att ändra på en post i filen Hotfix.inf kunde detta åtgärdas. I filen under [Version] skulle ändring göras från NTServicePackVersion=256 till NTServicePackVersion=1024. Efter ändringen dök fortfarande samma felmeddelande upp. Efter ytterligare efterforskningar visade det sig att utvecklingspaketet endast fungerar med engelsk version av Windows 2000.[25]

Efter formatering och installation av engelskt version av Windows 2000 efter samma premisser som innan, uppdateringar och ändring i Hotfix.inf gick utvecklingspaketet att installera.

4.3.2 Windows 2003 server standard

Installationen av Windows server 2003 standard edition gjordes efter samma premisser som Windows 2000, vi följde de föreslagna alternativen. Datorn fick namnet 2of3. Efter installation uppdaterades klienten genom Microsoft Windows Update hemsida. Efter detta aktiverades IPv6-stödet genom att lägga till IPv6 som ett protokoll på nätverkskopplingen.

4.3.3 Linux installation

Linuxdistribution som används är Red Hat 9 med 2.4.20 kärna. Installationen gjordes som server utan Open Office och utan multimedia program som är förvalda i Red Hat distributionen.

Exempel på multimedia program som valdes bort:

Xmms
Gimp

Dessa valdes bort då det förkortade installationen och inte behövs i detta arbete. Det som behövde läggas till är Development paket och kernel code, detta då det är nödvändigt att kompilera Red Hat Linux kärnan för att möjliggöra IPv6-stödet. Kärnan kompilerades utan den ursprungliga kärnkonfigurationsfil.

Det som kompilerades in i kärnan utöver det som finns i kärnan som standard (Red Hat 9 medföljande kärna (2.4.20-8), vald i installationen ”kernel source”) är:

Networking options

→the ipv6 protocol

Ipv6: Netfilter configuration

→ipv6 tables support

→mac address match support

→packet filtering

Då installationer av Linux gjordes med filsystemet *ext3* så kompilerades även detta in, dock är detta ingen krav för IPv6-funktionalitet. IPv6-stödet verifierades efter installation med ifconfig kommandot se listning 1.

```
eth0      Link encap:Ethernet  HWaddr 00:A0:24:50:0F:F5
          inet6 addr: 3002::3/64 Scope:Site
          inet6 addr: fe80::2a0:24ff:fe50:ff5/10 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:17864 errors:0 dropped:0 overruns:0 frame:0
          TX packets:2 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:100
          RX bytes:1978049 (1.8 Mb)  TX bytes:156 (156.0 b)
          Interrupt:5 Base address:0x220
```

Listning 1. Utmatning av kommandot *ifconfig eth0*

Permanenta nätverksinställningar gjordes i filen /etc/sysconfig/network-scripts/ifcfg-eth0. se listning 4, dock heter filen ifcfg-eth0 på klienten och inte ifcfg-eth1 som den gör på nat-pt datorn. Filen /etc/sysconfig/network listas i listning 2.

```
NETWORKING=yes
HOSTNAME=nat-pt.tng.htu.se
NETWORKING_IPV6=yes
IPV6FORWARDING=yes
IPV6AUTOCONF=no
```

Listning 2, IPV6FORWARDING=yes gör att alla noder agerar som router och vidareskickar paket i nätverket och IPV6AUTOCONFIG=no stänger av autokonfiguration av adresser så att datorer kan tilldelas adresser manuellt och inte från en borderrouter. [26][27]

4.4 DNS6 installation

Som namnserver installerades bind-9.2.3 med IPv6-stöd. Bind konfigurerades med configure-scriptet och flaggorna –with-openssl och –enable-ipv6. --with-openssl flaggan konfigurerar bind så att signerade zoner kan användas. –enable-ipv6 konfigurerar bind för IPv6 standarden, även om IPv6-stödet skall konfigureras automatiskt så användes denna flagga för att vara säker att IPv6-stödet installeras. Detta på grund av att datorn bind installerades på hade både IPv4 och IPv6 stacken installerad. Den agerar NAT-PT samtidigt som den är DNSv6-server.

Efter konfigurering av bind med kommandot ./configure –with-openssl –enable-ipv6 konfigurerades dns-server med enbart AAAA poster och inga A6 poster då A6 poster inte stöds av NAT-PT. Dns-server konfigurerades för loopback och tng (IPv6 domän) zon samt för baklänges namnupslag zoner. Konfigurations filer finns i bilaga A
Ett utdrag ur zon filen för tng domänen visas i listning 3. filen är skriven med enbart AAAA poster. [28][29][30]

```
TNGZON FILEN
……
tng.htu.se. SOA nat-pt.tng.htu.se. root.nat-pt.tng.htu.se. (
……
tng.htu.se. 1D IN NS nat-pt.tng.htu.se.
tng.htu.se. 1D IN AAAA 3002::2
nat-pt 1D IN AAAA 3002::2
2of3 1D IN AAAA 3002::3
3of3 1D IN AAAA 3002::4
bifrost 1D IN AAAA 3002::5
dns 1D IN AAAA 3002::2
```

Listning 3. Utdrag ut TNG zonfil. Den fullständiga Zonfilen kan ses i Bilaga A.

4.5 NAT-PT installation

Installationen göres som en Red Hat 9 installation, minimal installation. Datorn är utrustad med två nätverkskort där ena kortet är konfigurerad för IPv4 och den andra för IPv6. Installation av själva operativsystemet gjordes som vid klient installation av Linux klienten utom att NAT-PT funktioner och användande/underhåll bedömdes att inte behöva något grafiskt system.

Kärnan 2.4.20-8 kompilerades med följande alternativ utöver val som görs vid Linux klient installationer.

Networking options

→packet filtering

Ipv6: Netfilter configuration

TCP/IP Networking

→IP advanced router

(alla tillval under denna kategori skallaktiveras)

Efter kompileringen och omstart av systemet med den nya kärnan konfigurerades nätverkskortet i Linux kallad eth0 för IPv4 och eth1 för IPv6 trafik. Nedan listas konfigurations filer för båda nätverkskort. Listning 4 och 5.

```
DEVICE=eth0
ONBOOT=yes
IPV6INIT=no
BOOTPROTO=static
IPADDR=193.10.236.210
NETMASK=255.255.255.0
GATEWAY=193.10.236.1
```

Listning 4 Konfigurations fil ifcfg-eth0, konfigurerad för Ipv4 protokoll.

```
DEVICE=eth1
ONBOOT=yes
IPV6INIT=yes
IPV6ADDR=3002::2/64
```

Listning 5. konfigurations fil ifcfg-eth1, konfiguration för IPv6

Systemet konfigurerades ytterligare i filen /etc/sysconfig/network, listning 6.

```
NETWORKING=yes
HOSTNAME=nat-pt.tng.htu.se
FORWARD_IPV4=yes
NETWORKING_IPV6=yes
IPV6FORWARDING=yes
IPV6AUTOCONF=no
```

Listning 6. network filen konfiguration för både IPv6 och IPv4.

En ändring gjordes också i filen /etc/sysctl. På raden med IPv4 forwarding så måste värdet sättas till 1, på så sätt möjliggörs IPv4 forwarding i Linux kärnan. Efter en omstart av nätverkstjänsten så kan konfigurationen verifieras med kommandot ifconfig. Exempel på utmatning av kommandot ges i listning 7.

```
eth0      Link encap:Ethernet  HWaddr 00:C0:4F:A5:1C:00
          inet addr:193.10.236.210  Bcast:193.10.236.255
Mask:255.255.255.0
          inet6 addr: fe80::2c0:4fff:fea5:1c00/10 Scope:Link
          UP BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:17912 errors:0 dropped:0 overruns:0 frame:0
          TX packets:42 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:100
          RX bytes:1983024 (1.8 Mb)  TX bytes:3169 (3.0 Kb)
          Interrupt:11 Base address:0xec80

eth1      Link encap:Ethernet  HWaddr 00:A0:24:50:0F:F5
          inet6 addr: 3002::2/64 Scope:Site
          inet6 addr: fe80::2a0:24ff:fe50:ff5/10 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:17864 errors:0 dropped:0 overruns:0 frame:0
          TX packets:2 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:100
          RX bytes:1978049 (1.8 Mb)  TX bytes:156 (156.0 b)
          Interrupt:5 Base address:0x220

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:10 errors:0 dropped:0 overruns:0 frame:0
          TX packets:10 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:700 (700.0 b)  TX bytes:700 (700.0 b)
```

Listning 7. Utmatningen av ifconfig kommandot. Man ser att nätverkskortet eth0 är konfigurerat för IPv4 och eth1 är konfigurerat för IPv6.

[27][31]

4.5.1 NAT-PT Linux userspace based

NAT-PT Linux userspace based installeras på en dator med Linux operativsystem, installationen av Linux beskrivs ovan. NAT-PT programvaran hämtades från www.ipv6.or.kr/english/natpt-overview.htm. Uppackningen av programmet görs genom att skriva tar –xzf linux-usermode-natpt-src.tar.gz. Det första som måste göras sedan är att konfigurera IPv4_Adresses.list filen. Adresserna i denna lista används när paket kommer från IPv6-nätverket och skall skickas till IPv4-nätverket. Alla adresser i listan måste vara unika och routbara. Den version av NAT-PT som finns för tillfället kan maximalt hantera tjugo IPv4-adresser. Den klarar inte heller av port mappning. Den första adressen i IPv4_Adresses.list måste vara statisk mappad mellan DNSv6-adress och DNSv4 adress. Detta så paketen kan finna rätt vägen mellan de olika nätverken. Så det verkliga antalet tillgängliga adresser för översättning mellan de två nätverken är nitton.

Filen nat-pt_global.h filen kan behöva editeras något. Raden #define DEBUG skall sättas till OFF om ingen extra debug-information önskas. Om den är påslagen kommer NATen att skriva ut debug-information på skärmen. Raden #define IPv6_PREFIX_HOST kan behövas editeras ifall NAT-PT inte är standard routern för nätverket. Ifall den är det kan denna lämnas att vara som den är, annars måste ett IPv6-prefix sättas så klienterna kan routas till NATen.

Den sista filen som måste editeras är nat-pt.c filen. Det som måste ändras i denna fil är att ge *pIFnameIP4 som är interfacet som är kopplat mot IPv4-nätverket dess rätta eth namn samt ge *pIFnameIP6 som är interfacet mot IPv6-nätverket rätt namn. Efter dessa ändringar körs make och efter att kompileringen är klar kan demonen startas genom att skriva nat-pt. [32]

4.5.2 NAT-PT Cisco 7507 router

Routern som NAT-PT installeras på är en 7507 Cisco router med IOS version 12.3(8)T.

Routerns *FastEthernet 1/0/0* interface konfigureras med kommandot *ip address <ipv4 adress> <nätmask>*. Eftersom interfacet skall användas av NAT-PT så används kommandot *ipv6 nat*. Interface *FastEthernet 1/1/0* konfigureras med kommandot *ipv6 address <ipv6 adress/prefix>* och kommandot *ipv6 nat*. Kommandona utförs på respektive interface.

NAT-PT statiska och dynamiska översättningar görs i konfigurations läge. För att IPv6 klienter ska kunna kontakta IPv4 klienter och vice versa så måste det finnas mappningar mellan adresser. Kommandot för statisk mappning mellan IPv6 och IPv4 klienter är

ipv6 nat v6v4 source 3ffe:b00::1 192.168.0.1, IPv6 adressen 3ffe:b00::1 kommer översättas till 192.168.0.1 då IPv6-klienten försöker kontakta datorn med 3ffe:b00::1 adressen. För att översättningen skall fungera måste NAT-PT programvaran vara konfigurerad med en NAT-PT prefix. Prefixet används vid routing så att NAT-PT programvaran vet att destination adressen skall översättas. Prefixet måste vara 96 bitar lång och ha högre prioritet än adresser i IPv6-nätverket. Högre prioritet fås om NAT-PT prefix har högre adressbitar än adresser i IPv6-nätverket. Vid mappnigen som görs ovan är IPv6-adressen en adress som har NAT-PT prefix som prefix. NAT-PT prefix sätts med kommandot *ipv6 nat prefix <prefix/prefix length>*

För att IPv4-klienter ska nå IPv6-klienter används samma kommando där v6v4 byts ut mot v4v6 och en IPv4-adress mappas statisk mot en IPv6-adress. Översättningen görs då från en IPv4-adress till en IPv6-adress. IPv4 -dresser som är mappade till IPv6-adresser måste vara routbara i IPv4-nätverket och vara routade till routern som kör NAT-PT programvaran. För att dynamiska mappningar skall fungera måste pool adresser konfigureras. Pool adresser är en viss intervall av adresser som kommer användas vid översättning då dessa behövs. Dynamisk översättning kan användas först då DNS-server adresser finns statisk mappade så att kommunikation mellan IPv4 och IPv6 och vice versa klienter initieras med datornamn och inte adresser. IPv6 DNS-servern handhar IPv6-nätverket och IPv4 DNS-servern delegerar IPv6 domänen till den statisk mappade adressen. För att skapa en adresspool av IPv4-adresser som behövs vid kommunikation från IPv6 till IPv4 klienter används kommandot:

Ipv6 nat pool v4pool <start adress> <slut adress> prefix-length <prefix längd/ CIDR mätmask>

För att skapa en pool av IPv6-adresser används samma kommando men med IPv6-adresser och v4pool byts ut mot v6pool. Användning av access listor är möjlig och görs med kommandot:

ipv6 nat v6v4 source list pt-list pool v4pool

pt-list är en accesslist där endast bestämda klienter tillåts initiera förbindelse och få en IPv4-address ur v4pool adresspoolen.

Nedan listas utdrag ur router konfiguration i listning 8.

```
.....
interface FastEthernet1/0/0
 ip address 193.10.236.210 255.255.255.0
 half-duplex
 ipv6 nat
!
interface FastEthernet1/1/0
 no ip address
 full-duplex
 ipv6 address 3002::1/64
```

```
 ipv6 enable
 ipv6 nat
!
ip classless
ip route 0.0.0.0 0.0.0.0 193.10.236.1
no ip http server
!
......
access-list 15 permit any
ipv6 route 3FFE:B00::/96 FastEthernet1/1/0
ipv6 nat translation udp-timeout 600
ipv6 nat translation dns-timeout 600
ipv6 nat v4v6 source list 15 pool v6pool
ipv6 nat v4v6 source 193.10.192.40 3FFE:B00::1
ipv6 nat v4v6 pool v6pool 3FFE:B00::3 3FFE:B00::4 prefix-length 128
ipv6 nat v6v4 source list pt-list pool v4pool
ipv6 nat v6v4 source 3002::2 193.10.191.97
ipv6 nat v6v4 source 3002::3 193.10.191.98
ipv6 nat v6v4 pool v4pool 193.10.191.99 193.10.191.110 prefix-length
28
ipv6 nat prefix 3FFE:B00::/96
!
.....
ipv6 access-list pt-list
 permit ipv6 any any
```

Listning 8. Utdrag router configuration av interface och NAT-PT konfiguration. [33] Den fullständiga router konfigurations filen kan ses i Bilaga B

4.6 Installation av Bifrost

Installation av Bifrost görs med tomsrtbt (mini linux distribution, får plats på en diskett) [34]. Datorn bootas upp med tomsrtbt. Bifrost är en distribution skapad för att startas från en flashdisk. Men för denna rapport kommer installationen att göras direkt till hårddisken från en CD-skiva. Det vanliga tillvägagångssättet för installation av Bifrost är att installera Bifrost direkt från slu.se server. Men då datorn i frågan redan befinner sig i ett IPv6-nätverk är det enklare att installera från skiva. Installationen av Bifrost till den lokala hårddisken gjordes på följande sätt:

- Starta datorn med tomsrtbt diskett.
- Skapa katalogen Bifrost.
- Skapa en partition på hårddisken och mounta Bifrost katalogen till katalogen Bifrost.
- Packa upp Bifrost från CD-skivan direkt till Bifrost katalogen.
- Byt root över till hårddisken, verifiera uppackning och den nya root miljön.
- Tag ut diskett och CD-skiva ut datorn och starta om.

Konfigureringen av Bifrost görs genom scriptet /sbin/configure. Bifrost hemsidan (bifrost.slu.se) har noggranna beskrivningar hur konfigurationen skall gå till, dessutom

är konfigurations scriptet väldigt enkelt utformat. Se listning 9 nedan för IPv6-nätverks konfiguration på Bifrost:

```
Final Settings
--------------
The system has been configured with the following settings:

hostname: bifrost.tng.thn.htu.se

Desc: eth0
eth0: IP=3003::5
eth0: NM=64
eth0: BC=255.255.255.255
eth0: GW=3002::1/64

Desc: eth1
eth1: IP=3002::6
eth1: NM=64
eth1: BC=255.255.255.255

Allow telnet and ftp to the firewall
```

Listning 9. nätverks konfiguration för Bifrost

Då IPv6 stödet i Bifrost är relativt nytt finns det inga guider för hur det skall användas. Information om installation kan vara svår att få tag på men Bifrost är i grund och botten en vanlig Linux distribution även om den är en minimal sådan och specialanpassad. Kunskap om Linux operativsystem skall räcka för att installationen skall lyckas

Bifrost använder sig av ip6tables för att reglera trafik. Som standard är allt satt till DENY vilket gör att ingen trafik alls kan gå igenom Bifrosten. För att utröna hur bra IPv6-brandväggen fungerar stängs IPv4-stödet av, sedan konfigureras ip6tables för att tillåta IPv6-trafik. För att Bifrosten ska tillåta ICMPv6 trafik läggs denna regel till ip6tables (alla kommandon som ges till ip6tables måste göras som root): ip6tables –A INPUT –p ICMPv6 –j ACCEPT. –A INPUT betyder att kommandot skall läggast till i INPUT kedjan, -p ICMPv6 syftar på att det är protokollet ICMPv6 som berörs av regeln och –j ACCEPT betyder att Bifrosten får svara på ICMPv6 förfrågningar. För mer ingående förklaring hur iptables fungerar finns utförliga beskrivningar på http://www.netfilter.org/. Nedan i listning 10. listas ip6tables som finns på Bifrost datorn som man får fram genom kommandot ”ip6tables -L”:

```
Chain INPUT (policy ACCEPT)
Target      prot  opt         source      destination
ACCEPT      icmpv6            anywhere    anywhere
ACCEPT      tcp               3002::2     3002::6
ACCEPT      udp               3002::2     3002::6

Chain FORWARD (policy DROP)
Target      prot  opt         source      destination

Chain OUTPUT (policy DROP)
Target      prot  opt         source      destination
```

Listning 10.enkel ip6table regel på Bifrost som tillåter ICMP protokoll

Hur testerna av brandväggen gick till kan ses i kapitel 5.2.3.

5 Nätverks tester

Testerna som görs ska verifiera nätverkets funktionalitet. DNSv6 testas med avseende på prestanda och korrekthet. NAT-PT mekanismen analyseras och testas så att dess funktionalitet bevisas fungera. Fyra olika test områden har identifierats. Nätverket, DNSv6, NAT-PT och Bifrost testas. Mer specificerad testbeskrivning görs i underrubrikerna av detta kapitel.

5.1 IPv6 Nätverk

- Ping6 till IPv4 och IPv6 klienter

Kontakt inom TNG (The Next Generation) och andra IPv4 och IPv6 nätverk verifieras med applikationen Ping6 (*ping6* är en IPv6 version av IPv4s kommando *ping*).

- Belastnings test i TNG nätverket

Nätverkstrafik genereras i nätverket med nätverkstest programvara Iperf (nätverks test program). Detta test ger en bild av vad nätverket klarar då olika maskinvaru konfigurationer används i nätverket (10Mbit och 100Mbit nätverkskort). Testets resultat kan påverka NAT-PT och Bifrost belastnings test analys.

5.2 DNSv6

- DNS prestanda test

QueryPerf är en applikation som följer med Bind version 9. Den är gjord för prestanda mätningar på DNS servrar med avseende på antalet förfrågningar i sekunden.

Inmatningen till queryPerf görs med ett enkelt script och testresultat presenteras som en text fil. I resultatet får man information om hur många förfrågningar som har gjorts, hur många paket som gått förlorade, samt hur lång tid testet tog och genomsnittsmängd på namnuppslagningar i sekunden.

- DNS namnuppslagning

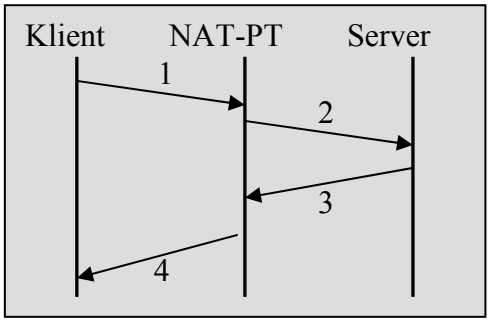
Namnuppslagning görs på den egna DNSv6 och utomstående DNSv6-serverar (definieras senare) där den utomstående servern används som referenspunkt.

Antalet namnuppslagningar som görs är 1000 förfrågningar. I testet används slumpvis valda IP-adresser. Testet görs från en Linux dator med applikationen *dig* och resultatet analyseras med applikationen *dif*. Inmatningen till *dig* görs med ett script som skapas egenhändigt.

- DNS namnuppslagning mot egen IPv6-server från IPv4 klient

Namnuppslagning görs från en IPv4-klient (från HTUs IPv4 nätverk), testet är ämnat verifiera funktionalitet och möjlighet att använda DNSv6 för att tillhandahålla service åt IPv4-klienter. Testlogik visas i figur 9. Server i detta fall är en DNSv6-server, klient är en IPv4-klientdator.

1. Meddelanden skickas till NAT-PT där de översätts till serverns IP-adresstandard
2. NAT-PT skickar vidare den översatta adressen till servern
3. Servern skickar tillbaka svaret till NAT-PT och översätter den tillbaka till klientens IP-adressstandard
4. NAT-PT skickar svaret till klienten.



Figur 9. Trafik flöde genom NAT-PT

5.3 NAT-PT

För trafikflöde genom NAT-PT i tester i detta avsnitt se figur 9.

- FTP-trafik från IPv6 till IPv4 och tvärtom genom NAT-PT

NAT-PT tester med FTP-trafik görs för att testa FTP ALG, samtidigt görs en NAT-PT belastningstest med FTP-trafik.

- Test av SSH trafik från IPv6 till IPv4 genom NAT-PT och tvärtom

Testet görs för att se om det är möjligt att ansluta till SSH servrar genom NAT-PT oavsett vilket nätverk man försöker ansluta sig från (IPv6 eller IPv4 nätverk). Testet är viktigt då SSH klienter kan användas för fjärradministrering av server datorer.

5.4 Test resultat

Windowsklienter som skulle ingå i IPv6-nätverket har visats sig inte kunna fungera med endast IPv6 IP-stacken installerad. DNS-förfrågningar från och till Windowsklienter kräver att dom görs med IPv4 protokollet. På grund av detta har Windowsklienter fått lämna plats åt Linuxklient (Red Hat 9), den installerades med samma konfiguration som den redan befintliga Linuxklienten.

NAT-PT mekanismer har inte fungerat med tillräcklig framgång för att dom planerade testerna skall kunna genomföras. Istället för de tester redovisas dom felsökningar och slutsatser av felsökningar som har gjorts.

DNSv6 tester har inte kunnat göras på grund av ovan nämnda skäl, tester som gjorts är att den egna zonen och dess reverserade zon fungerar och på så sätt styrka DNSv6 funktionalitet.

5.4.1 Test av IPv6 nätet

Verifieringen av nätverkets förbindelse inom IPv6-nätverket (TNG) görs. *Ping6* användes på så sätt att varje nod i nätverket körde kommandot mot alla andra noder i nätverket. Nedan i listning 11 och 12 visas resultat av kommandot mot en klient och borderrouter. Listningarna representerar gjorda tester i nätverket och att alla noder är igång och kontaktbara.

```
PING 3002::3(3002::3) 56 data bytes
64 bytes from 3002::3: icmp_seq=1 ttl=64 time=0.522 ms
64 bytes from 3002::3: icmp_seq=2 ttl=64 time=0.420 ms
64 bytes from 3002::3: icmp_seq=3 ttl=64 time=0.415 ms
--- 3002::3 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2003ms
rtt min/avg/max/mdev = 0.415/0.452/0.522/0.052 ms
```

Listning 11. Ping6 mot en klientnod i nätverket.

```
PING 3002::1(3002::1) 56 data bytes
64 bytes from 3002::1: icmp_seq=1 ttl=64 time=1.62 ms
64 bytes from 3002::1: icmp_seq=2 ttl=64 time=0.772 ms
64 bytes from 3002::1: icmp_seq=3 ttl=64 time=0.694 ms
--- 3002::1 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2042ms
rtt min/avg/max/mdev = 0.675/0.905/1.620/0.361 ms
```

Listning 12.ping6 mot borderroutern.

5.4.2 Test av NAT-PT Linux

NAT-PT installationen på Linux misslyckades vid kompilering, men efter mindre ändringar i källkoden så lyckades själva kompileringen. Kompileringen misslyckades på grund av att time.h filen inte var rätt deklarerad så som den var med 2.4.0-test9 kärna av Linux som NAT-PT som programvaran gjordes på. Efter ändring i källkoden lyckades kompileringen.

Källkods ändring som gjordes var:

#include <sys/time.h> ändrades till #include <bits/time.h>

Tester med *ping6* genom NAT-PT misslyckades. Programmets beteende vid körning var i det närmaste oförutsägbar. Vid ett antal tillfällen så gjorde programmet inga utmatningar till skärmen och vid andra gjorde den det. Utmatning till skärmen vid dom tillfällen det gjordes var: *setting interface eth1 in promiscuous mode.* Försök att få programvaran att dumpa (skriva) en loggfil med kommandot dokumenterat i själva dokumentationen misslyckades. Försök gjordes även att försöka kompilera en 2.4.0-test9 kärna på Red Hat 9, detta då NAT-PT programvaran utvecklades på denna. På grund av att medföljande applikationer inte var kompatibla med 2.4.0-test9 kärna, kunde inte kärnan kompileras. Att installera Red Hat 6.2 och försöka kompilera 2.4.0-test9 kärnan skulle innebära större chanser att lyckas. Det skulle däremot resultera i en för stor säkerhetsrisk med tanke på alla säkerhetshål som finns och arbete att säkerställa systemet skulle bli stort om ens möjlig. Med tanke på arbetets tidsbegränsning togs beslutet att överge NAT-PT på en Linuxmaskin och byta till Cisco 7507 router.

5.4.3 Test av NAT-PT Cisco router

NAT-PT mekanismen på Cisco routern fungerar såvida att den klarar ICMP protokoll översättning men inte *UDP (User Datagram Protocol)* och TCP protokoll. Omfattande felsökningar har gjorts utan framgång. Felsökningsstrategin var att övervaka inblandade komponenter samtidigt och försöka hitta fel i kommunikationen. Gjorda felsökningar redovisas nedan i detta kapitel.

Listning 13 nedan visar NAT-PT översättnings tabell med dom statiska mappningar av adresser.

```
 Router#sh ipv6 nat translations
Prot  IPv4 source              IPv6 source
      IPv4 destination         IPv6 destination
---   ---                      ---
      193.10.192.40            3FFE:B00::1

---   193.10.191.97            3002::2
      ---                      ---

---   193.10.191.98            3002::3
      ---                      ---
```

Listning 13. statiska mappningar på nat-pt

Nedan i listning 14 visas resultat av kommandot *traceroute* från en IPv4-nod till en IPv6-nod genom NAT-PT, endast ICMP protokollet används. *Traceroute* görs mot den i NAT-PT mappade IPv4-adressen. Listning 15 visar NAT-PT översättningstabellen efter en lyckad *traceroute*.

```
traceroute -I 193.10.191.97
traceroute to 193.10.191.97 (193.10.191.97), 30 hops max, 38 byte
packets
 1  193.10.192.2 (193.10.192.2)  1.174 ms  1.117 ms  1.080 ms
 2  193.10.236.210 (193.10.236.210)  0.638 ms  0.590 ms  0.527 ms
 3  193.10.191.97 (193.10.191.97)  1.610 ms  1.452 ms  1.313 ms
```

Listning 14 lyckad *traceroute* genom NAT-PT med endast ICMP protokollet.

```
Router#sh ipv6 nat translations
Prot  IPv4 source              IPv6 source
      IPv4 destination         IPv6 destination
---   ---                      ---
      193.10.192.40            3FFE:B00::1

---   193.10.191.97            3002::2
      193.10.192.40            3FFE:B00::1

---   193.10.191.97            3002::2
      ---                      ---

---   193.10.191.98            3002::3
      ---                      ---
```

Listning 15 NAT-PT översättningstabell där markerade fält visar dom dynamiskt skapade mappningar.

Efter en misslyckad *traceroute* där kommandot använder UDP protokoll ser NAT-PT översättnings tabell ut som visas i listning 16. Tabellen stämmer och ser ut som väntat vid en lyckad protokoll översättning. Inga svar har dock nått klienten som utförde *traceroute*, den får tillbaka ICMP meddelande av typ 35 (destination unreachable). För att ta reda på vart det går fel har övervakning gjorts på både routern och klienten som *traceroute* försöker nå. Detta kan ses i listning 17 och 18. Övervakningen gjordes samtidigt på både klienten och routern. Listning 19 visar mappning som görs på NAT-PT vid en traceroute. Test med att ställa en namnförfrågan till DNSv6 genom NAT-PT från en IPv4-klient har gjorts för att testa om DNS-ALG fungerar och klarar översättning, dock utan framgång. Efter all felsökning har inte felet hittats.

```
Prot  IPv4 source              IPv6 source
      IPv4 destination         IPv6 destination

udp   193.10.191.97,32769      3002::2,32769
      193.10.192.40,33434      3FFE:B00::1,33434

---   193.10.191.97            3002::2
      193.10.192.40            3FFE:B00::1

udp   193.10.191.97,32769      3002::2,32769
      193.10.236.2,33434       3FFE:B00::3,33434
```

Listning 16. utdrag ur NAT-PT översättningstabell, dynamiska mappningar har skapats. Den fullständiga NAT-PT översättnings tabellen kan ses i Bilaga D.

```
10:29:06.311614 3ffe:b00::1 > 3002::2: frag (0|18) 35148 > 33441: udp
10 [hlim 1]
10:29:06.311815 3002::2 > 3ffe:b00::1: [|icmp6]
8000.00:02:fd:04:dc:40 pathcost 0 age 0 max 20 hello 2 fdelay 15
10:29:11.305234 fe80::2a0:24ff:fe50:ff5 > 3002::1: icmp6: neighbor
sol: who has 3002::1
10:29:11.305932 3002::1 > fe80::2a0:24ff:fe50:ff5: icmp6: neighbor
adv: tgt is 3002::1 [class 0xe0]
10:29:11.309874 fe80::250:a2ff:fe57:3828 > 3002::2: icmp6: neighbor
sol: who has 3002::2 [class 0xe0]
10:29:11.309995 3002::2 > fe80::250:a2ff:fe57:3828: icmp6: neighbor
adv: tgt is 3002::2
10:29:11.312487 3ffe:b00::1 > 3002::2: frag (0|18) 35148 > 33442: udp
10 [hlim 1]
10:29:11.312629 3002::2 > 3ffe:b00::1: [|icmp6]
10:29:16.305999 fe80::250:a2ff:fe57:3828 > fe80::2a0:24ff:fe50:ff5:
icmp6: neighbor sol: who has fe80::2a0:24ff:fe50:ff5 [class 0xe0]
10:29:16.306120 fe80::2a0:24ff:fe50:ff5 > fe80::250:a2ff:fe57:3828:
icmp6: neighbor adv: tgt is fe80::2a0:24ff:fe50:ff5
10:29:16.312710 3ffe:b00::1 > 3002::2: frag (0|18) 35148 > 33443: udp
10 [hlim 1]
10:29:16.312840 3002::2 > 3ffe:b00::1: [|icmp6]
```

Listning 17. tcpdump på IPv6 klienten som *traceroute* utförs mot.

```
Router#debug ipv6 icmp
ICMP packet debugging is on
Router#
*Jul 28 08:02:40.903: ICMPv6: Received ICMPv6 packet from 3002::2,
type 136
*Jul 28 08:02:50.891: ICMPv6: Received ICMPv6 packet from
FE80::2A0:24FF:FE50:FF5, type 135
*Jul 28 08:02:55.891: ICMPv6: Received ICMPv6 packet from
FE80::2A0:24FF:FE50:FF5, type 136
```

Listning 18. övervakning av ICMP protokollet på routern.

Router#debug ipv6 nat

```
IPv6 NAT-PT debugging is on
Router#
*Jul  28  08:05:27.379:  IPv6  NAT:  udp  src  (193.10.192.40)  ->
(3FFE:B00::1), dst (193.10.191.97) -> (3002::2)
*Jul 28 08:05:27.379: IPv6 NAT: icmp src (3002::2) -> (193.10.191.97),
dst (3FFE:B00::1) -> (193.10.192.40)
*Jul  28  08:05:32.371:  IPv6  NAT:  udp  src  (193.10.192.40)  ->
(3FFE:B00::1), dst (193.10.191.97) -> (3002::2)
*Jul 28 08:05:32.375: IPv6 NAT: icmp src (3002::2) -> (193.10.191.97),
dst (3FFE:B00::1) -> (193.10.192.40)
*Jul  28  08:05:37.371:  IPv6  NAT:  udp  src  (193.10.192.40)  ->
(3FFE:B00::1), dst (193.10.191.97) -> (3002::2)
*Jul 28 08:05:37.375: IPv6 NAT: icmp src (3002::2) -> (193.10.191.97),
dst (3FFE:B00::1) -> (193.10.192.40)
```

Listning 19. övervakning av NAT-PT samtidigt som *traceroute* görs igenom den. Man ser översättning mellan IPv4 och IPv6 adresser

5.4.4 Test av DNSv6

Vid de initiala testerna av DNS uppstod problem. Klienterna frågade oavsett satt option i *dig* efter A6 record vid baklänges namnuppslagning. Felet bestod i att klienterna hade en för gammal version av *dig*. Efter uppdatering till samma *dig*-version som finns i bind (9.3) fungerade baklänges namnuppslagningarna.

Test av DNS gjordes med *dig* kommandot där namnuppslag gjordes mot klienter i IPv6-nätverket. Baklänges namnuppslagning för adressen som fås som svar på namnuppslagningen görs för att kunna verifiera baklänges namnuppslagnings-zonen. Testen styrker att DNS-servern är rätt konfigurerad för den egna zonen. Testet gjordes mot klienter med datornamn 2of3.tng.htu.se och nat-pt. Resultat av dig kommandot för namnuppslagningen kan ses i listning 20 och baklänges namnuppslagningen på 2of3.tng.htu.se kan ses i listning 21. Samt för nat-pt.tng.htu.se i listning 22 och 22.

```
[root@nat-pt root]# dig 2of3.tng.htu.se AAAA
......
;; QUESTION SECTION:
;2of3.tng.htu.se.               IN      AAAA

;; ANSWER SECTION:
2of3.tng.htu.se.        86400   IN      AAAA    3002::3

;; AUTHORITY SECTION:
tng.htu.se.             86400   IN      NS      nat-pt.tng.htu.se.

;; ADDITIONAL SECTION:
nat-pt.tng.htu.se.      86400   IN      AAAA    3002::2
......
```

Listning 20. Utdrag ur namnupslagning för 2of3.tgn.htu.se. Den fullständiga namnupslagningen kan ses i Bilaga D.

```
 [root@nat-pt root]# dig -x 3002::3
......
;; QUESTION SECTION:
;3.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.2.0.0.3.ip6.a
rpa.
IN PTR

;; ANSWER SECTION:
3.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.2.0.0.3.ip6.ar
pa.
86400IN PTR 2of3.tng.htu.se.

;; AUTHORITY SECTION:
0.0.0.0.0.0.0.0.0.0.0.0.2.0.0.3.ip6.arpa. 86400 IN NS
nat-pt.tng.htu.se.

;; ADDITIONAL SECTION:
nat-pt.tng.htu.se.      86400   IN      AAAA    3002::2
......
```

Listning 21 Utdrag ur baklänges namnuppslagningen på den erhållna adressen. Den fullständiga baklänges namnuppslagningen kan ses i Bilaga D.

```
root@nat-pt root]# dig nat-pt.tng.htu.se AAAA
......
;; QUESTION SECTION:
;nat-pt.tng.htu.se.              IN      AAAA

;; ANSWER SECTION:
nat-pt.tng.htu.se.      86400   IN      AAAA    3002::2

;; AUTHORITY SECTION:
tng.htu.se.             86400   IN      NS      nat-pt.tng.htu.se.
......
```

Listning 22. Utdrag ur namnuppslagningningen för nat-pt.tng.htu.se. Den fullständiga namnupslagningen kan ses i Bilaga D.

```
 [root@nat-pt root]# dig -x 3002::2
......
;; QUESTION SECTION:
;2.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.2.0.0.3.ip6.a
rpa.
IN PTR

;; ANSWER SECTION:
2.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.2.0.0.3.ip6.ar
pa.
86400IN PTR nat-pt.tng.htu.se.

;; AUTHORITY SECTION:
0.0.0.0.0.0.0.0.0.0.0.0.2.0.0.3.ip6.arpa. 86400 IN NS
nat-pt.tng.htu.se.
```

```
;; ADDITIONAL SECTION:
nat-pt.tng.htu.se.      86400   IN      AAAA    3002::2
......
```

Listning 23 utdrag ur baklänges namnuppslagningen på den erhållna adressen. Den fullständiga baklänges namnuppslagningen kan ses i Bilaga D.

6 Diskussion

Att Windows inte fungerar i ett IPv6 native nätverk är en stor besvikelse. Att Windows bara har delvis stöd men inte fullt ut är dåligt. Extra förvånande är att Windows server 2003 standard edition som är ett server operativsystem inte har fullt stöd är helt oförståeligt. Man får hoppas att Microsoft ordnar upp detta då de flesta persondatorer idag använder sig av Windows och ifall inte stödet finns med kan det bli en ny bromskloss.

Linuxoperativsystem har ett väl fungerande IPv6-stöd. Risken som Windows nu står inför är att de kommer att förlora på serversidan till Linuxservrar när IPv6 börjar komma igång på allvar.

Den stora besvikelsen i arbetet är misslyckandet med att få igång NAT-PT. Vi kommer troligtvis att fortsätta vårat arbete med NAT-PT, detta är ett intressant område och tekniken behövs verkligen i skiftet mellan IPv4-IPv6.

Arbetets omfattning kan i efterhand konstateras ha varit alldeles för stort. Arbetet kunde ha delats upp i två arbeten. Första arbetet där ett IPv6-testnätverk byggs upp med servrar (WWW, E-post, DNS, FTP, streaming video…). Andra arbetet kunde inriktas på att koppla ihop IPv6-testnätverket med ett befintligt IPv4-nätverk.

6.1 Slutsatser

Att det kommer att bli ett skifte från IPv4 till IPv6 inom den närmaste framtiden är det ingen som betvivlar. Det svåra kommer att vara att sätta upp temporära IPv6-nätverk i väntan på att hela Internet skall bli IPv6 anpassat.

Att sätta upp ett IPv6-nätverk skiljer sig inte markant från att sätta upp ett IPv4. Än så länge behöver man i Linuxkärnan välja att lägga till vissa IPv6-stöd för att det skall fungera. Men efter det kan det mer eller mindre fungera. Detta är möjligt om man har autokonfiguration påslagen på routern. Dock användes inte detta alternativ i detta arbete då NAT-PT måste veta exakt vilka datorer som finns i nätverket. Så klienternas interface konfigurerades manuellt. IPv6-applikationer är så pass bra i Linux att avsaknad av en applikation som klarar IPv4 men inte IPv6 aldrig uppstod. Linux som operativt system är långt före Windows när det gäller IPv6, Windows kan inte fungera i ett IPv6-nätverk utan att ha IPv4 IP-stacken installerad och det gjorde att Windows fick överges som operativsystem i detta arbete.

NAT-PT mekanism på Linuxmaskinen som användes i arbete visade sig vara otillräcklig och svår att konfigurera. Dokumentationen som finns tillgänglig på engelska är liten. Applikationen har även brister i källkoden och utvecklingen tycks vara nedlagd.

När det gäller konfigurationsmöjligheter och användarvänlighet är Cisco IOS klart överlägsen de i Linux NAT-PT. Det finns bra guider och hjälp med konfigureringen. Cisco uppgraderar sina IOS ständigt och nya implementationer och finesser och säkerhets uppdateringar läggs till ständigt i nya versioner. Dock är Cisco routrar relativt dyra samt uppgraderingar av IOS till en version som stödjer IPv6 kan bli kostsamma.

De planerade testerna som skulle göras i arbetet kunde inte genomföras på grund av arbetets resultat. Detta gör att själva utvärderingen av NAT-PT mekanismen inte kunde göras. Dock är det Cisco-tillämpningen den som ändå rekommenderas. Det är inte säkert att Cisco routern är den felande komponenten då dess NAT-PT mekanism är delvist fungerande. Trots all hjälp från både IT-kunnig personal och diverse forum på Internet kunde inte felet hittas.

DNSv6 fungerade utan större problem, se under kapitel Test resultat, test av DNS. Bifrost klarar av IPv6, även om testet som gjordes inte var omfattande, men i grund och botten är det en vanlig Linux distribution. Script som används för konfigurationer är anpassade för IPv6 och IPv4. Konfigurationsfilers placering i distributionen kan vara något förvirrande men är inte ett egentligt hinder. Med tanke på att distributionen av Bifrost testad (Bifrost test) i arbetet är den första IPv6 anpassade version och redan finns en ny version ute kan man nog anta att den bara blir bättre.

6.2 Rekommendationer till fortsatt arbete

Detta arbete har en hel del lösa trådar som kan utforskas och utvecklas. Det första är att få NAT-PT att fungera som det är tänkt. Rekommenderat är att få Cisco-routern att fungera i första hand då denna kan sköta större mängder klienter än Linux NAT-PT. Detta kan vara viktigt om HTU väljer att utvidga IPv6-nätverket.

Ett annat arbete som kan göras är att se hur IPv6-nätverket beter sig om man kopplar ihop allt som det var tänkt från början i detta arbete. Med Sunets IPv6 native uppkoppling via Bifrost och ha globalt routbara IPv4-adresser i NAT-PT poolen. Detta skulle göra att datakommunikation kan komma både via Sunets IPv6-uppkoppling samt HTU uppkoppling. Vad vi har kunnat finna är detta aldrig testat. Skulle vara intressant at se hur nätverket fungerar i en sådan miljö.

En utforskande rapport om IPv6 där man försöker tyda vad som kommer att hända när IPv6 är fullt utvecklat: Kommer IP-telefoni få ett uppsving, när kommer alla apparater att ha en IP-adress, hur kommer Telefoni bolagen att handskas med detta, hur ser säkerheten ut, hur skiljer sig IPv6 och IPv4 åt när det gäller implementering rent mjukvaru mässigt, vad finns det för risker med att program skrivs om till IPv6 utan att man tänker på säkerheten.

7 Källförteckning

[1]IPv4 (IP) - A Brief History,

Tillgänglig: <http://ntrg.cs.tcd.ie/undergrad/4ba2/ipng/gerd.ipv4.html>[2004-07-27]

[2]IP Adress, (senast uppdaterad 2004-06-23),

Tillgänglig: <http://en.wikipedia.org/wiki/IP_address>[2004-07-27]

[3] Loshin, Peter (1999). *IPV6 Clearly Explained*, San Francisco, Ca: Morgan Kaufmann

[4] Wegner, J.D.& Rockell, Robert, *IP addressing and subnetting: including IPv6*, Rockland, MA: Syngress

[5]RFC2131,Dynamic Host Configuration Protocol,

http://www.faqs.org/rfcs/rfc2131.html

[6] RFC1631, The IP Network Adress Translator,

http://www.faqs.org/rfcs/rfc1631.html

[7]IP Version 6 Working Group (ipv6) Charter, (senast uppdaterad 2004-06-18), Tillgänglig: <http://www.ietf.org/html.charters/ipv6-charter.html>[2004-07-27]

[8]Sierra,J.M.;Ribagorda,A.;Munoz,A.;Jayaram,N.;
Security Technology, 1999. Proceedings. IEEE 33rd Annual 1999 International Carnahan Conference on , 5-7 Oct. 1999

[9]Madalina Baltatu & Antonio Lioy (2000), IP Security Paper Summery, Tillgänglig: <http://staging.denison.edu/~bressoud/cs402-f03/summary-question/ipsec-summary.pdf>[2004-07-27]

[10]John Thomas and Adam J. Elbirt (2004), IPsec: How it works and why we need it, Computerworld 03-18, Tillgänglig:

<http://www.computerworld.com/securitytopics/security/story/0,10801,91312,00.html> [2004-07-27]

[11]IP Security (IPsec)(2004), Tillgänglig: <http://www.ind.alcatel.com/library/e-briefing/eBrief_IPSec.pdf>[2004-07-27]

[12] Michael Schorr (2004), IPv4 and IPv6: A Comparison, myITforum.com 2004-05-01, Tillgänglig: <http://www.myitforum.com/articles/16/view.asp?id=6720>[2004-07-27]

[13]The Case for IPv6 (1999), Tillgänglig:

<http://www.6bone.net/misc/case-for-ipv6.html>[2004-07-27]

[14] IPv4 Header, Tillgänglig:

<http://www.spacerobots.org/dennis/Headers/IPv4header.htm>[2004-07-27]

[15] IPv6, Internet Protocol version 6, Tillgänglig:

<http://www.networksorcery.com/enp/protocol/ipv6.htm>[2004-07-27]

[16] From options to extension headers, Tillgänglig:

<http://www.ngnet.it/e/ipv6proto/ipv6-proto-2.php>[2004-07-27]

[17]RFC1933, Transition Mechanisms for IPv6 Hosts and Routers,

http://www.faqs.org/rfcs/rfc1933.html

[18] Hossam Afifi, Laurent Toutain, Methods for IPv4-IPv6 transition, The Fourth IEEE Symposium on Computers and Communications 1999, 6-8 Juli

[19] Dual Stack Transition Mechanism (DSTM)(2002), Tillgänglig:

<http://www.ietf.org/proceedings/02mar/I-D/draft-ietf-ngtrans-dstm-07.txt>[2004-07-27]

[20]RFC2765, Stateless IP/ICMP Translation Algorithm (SIIT), http://www.faqs.org/rfcs/rfc2765.html

[21]RFC2766, Network Address Translation - Protocol Translation (NAT-PT), http://www.faqs.org/rfcs/rfc2766.html

[22]RFC3053, IPv6 Tunnel Broker, http://www.faqs.org/rfcs/rfc3053.html

[23]RFC3056, onnection of IPv6 Domains via IPv4 Clouds, http://www.faqs.org/rfcs/rfc3056.html

[24] Bifrost Network Project, (Senast uppdaterad 2004-03-15), Tillgänglig: http://bifrost.slu.se[2004-07-27]

[25] Frequently Asked Questions about the Microsoft IPv6 Technology Preview for Windows 2000, Tillgänglig:

<http://msdn.microsoft.com/downloads/sdks/platform/tpipv6/faq.asp>[2004-07-27]

[26] Ibrahim Haddad, Installing IPv6 with Linux Kernel

Tillgänglig: <http://www.linux.ericsson.ca/ipv6/kernel.html>[2004-07-28]

[27] Eric S. Raymond (2002), The Linux Installation HOWTO (Senast uppdaterad 2002-07-06), Tillgänglig: <http://mirrors.kernel.org/LDP/HOWTO/Installation-HOWTO/kernel.org>[2004-07-28]

[28]RFC3596, DNS extensions to support IP version 6,

[29]David Gordon & Ibrahim Haddad (2003), Building a Linux IPv6 DNS Server (senast uppdaterad 2003-10), Tillgänglig: http://www.linux.ericsson.ca/ipv6/dns_v6.pdf[2004-07-30]

[30]Betrand Buclin (2000), IPv6 DNS settings (senast uppdaterad 2000-01-31), Tillgänglig: <http://www.isi.edu/~bmanning/v6DNS.html>[2004-07-30]

[31] J. W. Atwood, Kedar C. Das, & Ibrahim Haddad, NAT-PT: Providing IPv4/IPv6 and IPv6/IPv4 Address Translation (2003),

Tillgänglig: < http://www.linux.ericsson.ca/ipv6/v4_v6_translation.pdf>[2004-07-27]

[32]Dokumentation medföljande packade källkoden för NAT-PT, Tillgänglig: <http://www.ipv6.or.kr/english/linux-usermode-natpt-src.tar.gz>[2004-07-29]

[33] Implementing NAT-PT for IPv6-Cisco IOS Software Releases 12.3 Mainline - Cisco Systems, Tillgänglig:

<http://www.cisco.com/en/US/products/sw/iosswrel/ps5187/products_configuration_guide_chapter09186a00801d6600.html>[2004-07-27]

[34]tomsrtbt home page, Tillgänglig: <http://www.toms.net/rb/tomsrtbt.FAQ>[2004-07-27]

A DNSv6 filer

Innehållsförteckning

NAMED:CONF

```
options {
directory "/var/named";
listen-on-v6 { any; };
};

controls {
inet 127.0.0.1 allow { localhost; };
};

// endast cache namnserver konfiguration
zone "." IN {
type hint;
file "master/named.ca";
};

// loopback namn uppslagning
zone "localhost" IN {
type master;
file "master/localhost.zone";
allow-update { none; };
};

// loopback baklÃ¤nges namn uppslagning
zone "0.0.127.in-addr.arpa" IN {
type master;
file "master/0.0.127.in-addr.arpa";
allow-update { none; };
};
```

```
// SÃ¤ker signerad zon fil
zone "tng.htu.se" IN {
type master;
//file "master/tng.htu.se.signed";
file "master/tng.htu.se";
};

// baklÃ¤nges namn uppslagning för zonen AAAA
zone "0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.2.0.0.3.ip6.arpa" IN {
type master;
file "master/0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.2.0.0.3.ip6.arpa";
};

// baklÃ¤nges namn uppslagning för zonen A6
// zone "tng.arpa" IN {
// type master;
// file "master/tng.rev";
// };
```

TNGZON fil

$TTL 86400

tng.htu.se. □O Anat-pt.tng.htu.se. root.nat-pt.tng.htu.se. (

2004062916 ; Serial number (yyyymmdd-num)

3H ; Refresh

15M ; Retry

1W ; Expire

1D) ; Minimun

tng.htu.se. 1D IN NS nat-pt.tng.htu.se.

tng.htu.se. 1D IN AAAA 3002::2

nat-pt 1D IN AAAA 3002::2

2of3 1D IN AAAA 3002::3

3of3 1D IN AAAA 3002::4

bifrost 1D IN AAAA 3002::5

dns 1D IN AAAA 3002::2

$INCLUDE "/var/named/master/Kzonekey.+157+31741.key"

TNGZON baklängesnamnuppslagning

```
$TTL 86400
$ORIGIN 0.0.0.0.0.0.0.0.0.0.0.0.2.0.0.3.ip6.arpa.
@ SOA nat-pt.tng.htu.se. root.nat-pt.tng.htu.se. (
2004063022 ; serial
3H ; Refresh
15M ; Retry
1W ; Expire
1D ) ; Minimum
0.0.0.0.0.0.0.0.0.0.0.0.2.0.0.3.ip6.arpa. IN NS nat-pt.tng.htu.se.
IN MX 10 root.tng.htu.se.
$ORIGIN 0.0.0.0.0.0.0.0.0.0.0.0.2.0.0.3.ip6.arpa.
2.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0 IN PTR nat-pt.tng.htu.se.
3.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0 IN PTR 2of3.tng.htu.se.
4.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0 IN PTR 3of3.tng.htu.se.
5.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0 IN PTR bifrost.tng.htu.se.
```

B NAT-PT filer

Innehållsförteckning

Cisco NAT-PT configuration

```
Building configuration...

Current configuration : 1596 bytes
!
version 12.3
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
service multiple-config-sessions
service single-slot-reload-enable
!
hostname Router
!
boot-start-marker
boot-end-marker
!
!
redundancy
 mode hsa
enable password XXX
!
no aaa new-model
ip subnet-zero
!
!
ip cef
!
!
```

```
interface FastEthernet1/0/0
 ip address 193.10.236.210 255.255.255.0
 half-duplex
 ipv6 nat
!
interface FastEthernet1/1/0
 no ip address
 full-duplex
 ipv6 address 3002::1/64
 ipv6 enable
 ipv6 nat
!
interface POS4/0/0
 no ip address
 shutdown
!
ip classless
ip route 0.0.0.0 0.0.0.0 193.10.236.1
no ip http server
!
!
!
access-list 15 permit any
snmp-server community mummalandet RW
snmp-server enable traps tty
ipv6 route 3FFE:B00::/96 FastEthernet1/1/0
ipv6 nat translation udp-timeout 600
ipv6 nat translation dns-timeout 600
ipv6 nat v4v6 source list 15 pool v6pool
ipv6 nat v4v6 source 193.10.192.40 3FFE:B00::1
```

```
ipv6 nat v4v6 pool v6pool 3FFE:B00::3 3FFE:B00::4 prefix-length 128
ipv6 nat v6v4 source list pt-list pool v4pool
ipv6 nat v6v4 source 3002::2 193.10.191.97
ipv6 nat v6v4 source 3002::3 193.10.191.98
ipv6 nat v6v4 pool v4pool 193.10.191.99 193.10.191.110 prefix-length 28
ipv6 nat prefix 3FFE:B00::/96
!
!
ipv6 access-list pt-list
 permit ipv6 any any
!
control-plane
!
!
line con 0
 password XXX
 login
line aux 0
line vty 0 4
 password XXX
 login
!
!
end
```

C Klient konfigurations information

Innehållsförteckning

2of3 ifconfig

```
eth0      Link encap:Ethernet  HWaddr 00:C0:4F:43:49:E8
          inet6 addr: 3002::3/64 Scope:Global
          inet6 addr: fe80::2c0:4fff:fe43:49e8/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:1098053 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1220 errors:0 dropped:0 overruns:0 carrier:986
          collisions:0 txqueuelen:100
          RX bytes:91286915 (87.0 Mb)  TX bytes:110161 (107.5 Kb)
          Interrupt:11 Base address:0xdc00

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:6831 errors:0 dropped:0 overruns:0 frame:0
          TX packets:6831 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:565002 (551.7 Kb)  TX bytes:565002 (551.7 Kb)
```

2of3 resolv.conf

```
search tng.htu.se
nameserver fec0:0000:0000:0202:0000:0000:0000:0001
```

NAT-PT resolv.conf

```
domain tng.htu.se
search tng.htu.se
nameserver 3002::2
```

NAT-PT ifconfig

```
eth0      Link encap:Ethernet  HWaddr 00:C0:4F:A5:1C:00
          inet addr:193.10.236.210  Bcast:193.10.236.255  Mask:255.255.255.0
          inet6 addr: fe80::2c0:4fff:fea5:1c00/10 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:657234 errors:0 dropped:0 overruns:0 frame:0
          TX packets:5576 errors:0 dropped:0 overruns:0 carrier:5
          collisions:0 txqueuelen:100
          RX bytes:74223806 (70.7 Mb)  TX bytes:806812 (787.9 Kb)
          Interrupt:11 Base address:0xec80

eth1      Link encap:Ethernet  HWaddr 00:A0:24:50:0F:F5
          inet6 addr: fe80::2a0:24ff:fe50:ff5/10 Scope:Link
          inet6 addr: 3002::2/64 Scope:Global
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:167532 errors:0 dropped:0 overruns:0 frame:0
          TX packets:206 errors:0 dropped:0 overruns:0 carrier:0
          collisions:1 txqueuelen:100
          RX bytes:13933385 (13.2 Mb)  TX bytes:22760 (22.2 Kb)
          Interrupt:5 Base address:0x220

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:1462 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1462 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:163095 (159.2 Kb)  TX bytes:163095 (159.2 Kb)
```

D Tester och felsökningar

Innehållsförteckning

NAT-PT namnuppslagning

```
[root@nat-pt root]# dig nat-pt.tng.htu.se AAAA

; <<>> DiG 9.2.3 <<>> nat-pt.tng.htu.se AAAA
;; global options:  printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 27402
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 0

;; QUESTION SECTION:
;nat-pt.tng.htu.se.            IN      AAAA

;; ANSWER SECTION:
nat-pt.tng.htu.se.      86400   IN      AAAA    3002::2

;; AUTHORITY SECTION:
tng.htu.se.             86400   IN      NS      nat-pt.tng.htu.se.

;; Query time: 10 msec
;; SERVER: 3002::2#53(3002::2)
;; WHEN: Fri Jul 30 22:31:25 2004
;; MSG SIZE  rcvd: 77
```

2of3 Namnuppslagning

```
[root@nat-pt root]# dig 2of3.tng.htu.se AAAA

; <<>> DiG 9.2.3 <<>> 2of3.tng.htu.se AAAA
;; global options:  printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 33712
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 1

;; QUESTION SECTION:
;2of3.tng.htu.se.               IN      AAAA

;; ANSWER SECTION:
2of3.tng.htu.se.        86400   IN      AAAA    3002::3

;; AUTHORITY SECTION:
tng.htu.se.             86400   IN      NS      nat-pt.tng.htu.se.

;; ADDITIONAL SECTION:
nat-pt.tng.htu.se.      86400   IN      AAAA    3002::2

;; Query time: 11 msec
;; SERVER: 3002::2#53(3002::2)
;; WHEN: Fri Jul 30 22:31:48 2004
;; MSG SIZE  rcvd: 110
```

NAT-PT baklänges namnuppslagning

```
[root@nat-pt root]# dig -x 3002::2

; <<>> DiG 9.2.3 <<>> -x 3002::2
;; global options:  printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 43575
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 1

;; QUESTION SECTION:
;2.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.2.0.0.3.ip6.arpa.
IN PTR

;; ANSWER SECTION:
2.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.2.0.0.3.ip6.arpa.
86400IN PTR nat-pt.tng.htu.se.

;; AUTHORITY SECTION:
0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.2.0.0.3.ip6.arpa. 86400 IN NS
nat-pt.tng.htu.se.

;; ADDITIONAL SECTION:
nat-pt.tng.htu.se.      86400   IN      AAAA    3002::2

;; Query time: 11 msec
;; SERVER: 3002::2#53(3002::2)
;; WHEN: Fri Jul 30 22:32:47 2004
;; MSG SIZE  rcvd: 163
```

2of3 Baklängesnamnuppslagning

```
[root@nat-pt root]# dig -x 3002::3

; <<>> DiG 9.2.3 <<>> -x 3002::3
;; global options:  printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 44074
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 1

;; QUESTION SECTION:
;3.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.2.0.0.3.ip6.arpa.
IN PTR

;; ANSWER SECTION:
3.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.2.0.0.3.ip6.arpa.
86400IN PTR 2of3.tng.htu.se.

;; AUTHORITY SECTION:
0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.2.0.0.3.ip6.arpa. 86400 IN NS
nat-pt.tng.htu.se.

;; ADDITIONAL SECTION:
nat-pt.tng.htu.se.      86400   IN      AAAA    3002::2

;; Query time: 11 msec
;; SERVER: 3002::2#53(3002::2)
;; WHEN: Fri Jul 30 22:32:57 2004
;; MSG SIZE  rcvd: 168
```