Tillgång till IPv6-baserade webbtjänster från IPv4-nätverk genom användning av transparenta proxyservrar

UTKAST

Tony Karlsson
Åbo Akademi

10 april 2012

OBS! *Den här versionen av avhandlingen är ett utkast. Avhandlingen är under arbete, och delar av den slutliga texten fattas ännu. Beakta speciellt:*

- Ordningsföljden på innehållet borde motsvara den slutliga, men rubriker och delar av innehållet kan ännu förändras. Där större textavsnitt fattas eller ska redigeras framgår det ur ett kursiverat TODO-stycke.

- Källhänvisningarna är ofullständiga. För vissa textstycken kan de fattas helt, alternativt hänvisa till källor som ännu saknas i källförteckningen, [?].

- Textformatering, marginaler, radavstånd och dylikt kan avvika från den slutliga versionen. Många ord avstavas just nu felaktigt, eftersom detta för tillfället har gjorts automatiskt enligt felaktiga regler. Det här åtgärdas till den slutliga versionen.

- Bilder saknas helt. Eventuellt tillkommer några grafer till den slutliga versionen.

Referat

I takt med att bristen på IPv4-adresser ökar blir IPv6 av naturliga skäl allt vanligare. Det är sannolikt att webbtjänsteleverantörer inom en förutsebar framtid kommer att delvis frångå IPv4-adressering till förmån för IPv6. Därmed är det även oundvikligt att någon webbtjänst inom en relativt snar framtid kommer att finnas tillgänglig enbart med IPv6-adressering. Samtidigt finns det ett enormt antal lokala nätverk byggda på enbart IPv4-adressering.

Många små företag och organisationer kommer sannolikt inte att inom någon snar framtid prioritera en fullständig övergång från IPv4 till IPv6, inte minst av ekonomiska och resursmässiga skäl. Därmed uppstår en potentiell problemsituation, eftersom nativa IPv6-baserade webbtjänster inte kan nås från IPv4-nätverk utan specialarrangemang.

I detta arbete undersöks en metod att göra IPv6-baserade webbtjänster tillgängliga i IPv4-nätverk genom användning av transparenta proxyservrar. Avsikten är att göra IPv6-tjänster tillgänliga för nativa IPv4-klienter utan förutsätta några som helst aktiva åtgärder i klienterna.

TODO: *Mera konkret om avhandlingens innehåll och slutsats. Den här texten är också delvis råddig och behöver skrivas om lite.*

Innehåll

1 Inledning

TODO: *Hela inledningen är work in progress. Mellanrubrikerna är som reminders åt mig själv, och faller troligtvis bort i slutliga versionen.*

1.1 Övergången från IPv4 till IPv6

TODO: *Lite kort text om att IPv4-adresserna tar slut och den långsamma takten som IPv6 tas i bruk.*

1.2 Behovet av tillgång till IPv6-tjänster i IPv4-nätverk

TODO: *Text om hur småföretag och mindre organisationer inte har råd att förnya sina nätverk, och därför fortsätter använda IPv4 länge ännu.*

1.3 Syftet med denna avhandling

I detta arbete undersöks en specifik metod att möjliggöra tillgång till IPv6-baserade webbtjänster i lokala IPv4-nätverk. Avsikten är att undersöka en metod som fungerar i ett IPv4-baserat nätverk utan att kräva aktiva åtgärder i klienterna.

I arbetet undersöks den specifika metodens tillämplighet endast på HTTP-baserade webbtjänster, vilka utgör merparten av världens internettrafik.

TODO: *Här fattas mera utförlig text om avhandlingens syfte.*

2 Internetprotokollet (IP)

TODO: *Inledning för IP-kapitlet ännu under arbete. Texten här är bara reminders åt mig själv.*

Internetprotokollet IP (eng. *Internet Protocol*)...

ICANN *Internet Corporation for Assigned Names and Numbers* och IANA (*Internet Assigned Numbers Authority*)...

RIR (eng. *Regional Internet Registry*) - 5 olika, i Europa RIPE RCC (*Réseaux IP Européens Network Coordination Centre*).

RFC 1466: Guidelines for Management of IP Address Space från maj 1993 [1]

RFC 1881: IPv6 Address Allocation Management från december 1995 [2]

2.1 IPv4

Den fjärde versionen av internetprotokollet, IPv4, är ett över 30 år gammalt protokoll som används för majoriteten av datatrafiken på internet ännu idag [3]. IPv4 beskrevs av Postel i *RFC 760 - DoD standard Internet Protocol* i januari 1980 [4] samt senare i det ersättande dokumentet *RFC 791 - Internet Protocol* i september 1981 [5].

IPv4-adresser är 32 bitar (4 byte) långa. I läsbar form representeras IPv4-adresser som fyra heltal från 0 till 255, där varje heltal motsvarar en byte (8 bit) av adressen. Heltalen separeras med punkter, och inledande nollor i heltalen skrivs normalt inte ut. Exempelvis används den läsbara formen *173.194.32.35* för att representera den binära adressen *10101101 11000010 00100000 00100011* (hexadecimalt *ADC22023*). [5]

IPv4-adresser delas upp i två delar som kan vara av varierande längd, men som alltid tillsammans utgör 32 bitar. Den första delen (från vänster) är ett n bitar långt nätverksnummer, ett prefix, medan den andra delen utgör en 32-n bitars lokal adress inom nätverket. Antalet möjliga IP-adresser inom ett nätverk är således direkt relaterat till prefixets längd. Denna uppdelning av IP-adresser effektiverar sättet att skriva regler för routing, då man inte behöver beskriva rutter för alla enskilda IP-adresser, utan istället endast för

större helheter baserade på nätverksnummer. [3, 6]

Nätverkets storlek (dvs. prefixets längd) anges ofta med hjälp av en delnätsmask (eng. *subnet mask*). Delnätsmasker är 32 bitar långa värden där varje bit som representerar nätverksnumrets längd sätts till 1, och de övriga bitarna sätts till 0. För ett nätverk där prefixets längd är 24 bitar är delnätsmasken i binär form således *11111111 11111111 11111111 00000000*. I läsbar form representeras delnätsmasker enligt samma metod som IP-adresser. Med andra ord skrivs samma 24 bitars delnätsmask normalt i formen *255.255.255.0*, medan en 16 bitars delnätsmask skrivs i formen *255.255.0.0*. [3, 6]

I läsbar form representeras ett nätverksnummer eller prefix som den lägsta IP-adressen inom det specifika nätverket, med andra ord den IP-adress där den lokala adressdelen består av enbart nollor. Nätverksnumret skrivs ut med ett suffix bestående av ett snedstreck följt av ett heltal, vilket anger prefixets längd i antal bitar. Exempelvis beskriver *130.232.0.0/16* ett nätverk där nätverksnumret är 16 bitar långt, medan *77.234.108.0/24* beskriver ett annat nätverk med ett 24 bitar långt prefix. Nätverk är hierarkiska på så vis att ett nätverk kan ha flera underliggande delnät (eng. *subnet*). Exempelvis kan *130.232.67.0/24* anses vara ett delnät i nätverket *130.232.0.0/16*. [3, 6]

2.2 Bristen på IPv4-adresser

Vid en första anblick borde IPv4 med sin adresslängd på 32 bitar teoretiskt sett kunna erbjuda 2^{32}, alltså nästan 4,3 miljarder, unika IP-adresser. En del adresser och delnät har dock reserverats för speciella ändamål, exempelvis *broadcast*, *multicast* och *loopback*. Sådana specialområden har dokumenterats av IETF i *RFC 5735 - Special Use IPv4 Addresses* [7]. Då IP-adresser reserveras för organisationer i större delnät är det även praktiskt taget omöjligt att komma nära ett fullständigt utnyttjande av adressområdena; en organisation som har allokerats ett */16*-delnät använder eventuellt inte alla tillgängliga adresser, men eftersom hela delnätet har reserverats kan outnyttjade adresser inom det ändå inte tilldelas någon annan organisation. [3, 6]

I början på 1990-talet började användningen av internet kraftigt öka. Redan då förutsåg IETF att IPv4-adresserna inte skulle räcka hur långt som

helst. Man började då arbeta med olika potentiella lösningar på problemet. Delvis har bristen på IPv4-adresser kunnat undvikas bland annat på grund av av nätverksadressöversättning med NAT (eng. *Network Address Translation*). NAT beskrevs i *RFC 1631 - The IP Network Address Translator (NAT)* i maj 1994 [8] och senare i *RFC 3022 - Traditional IP Network Address Translator (Traditional NAT)* i januari 2001 [9]. Istället för att använda publika, globalt unika adresser för alla noder i ett nätverk gör NAT det möjligt att inom ett lokalt nätverk använda adresser ur exempelvis *192.168.0.0/16*-området (icke-ruttbara adresser, avsedda för lokal, privat användning [7]) och dela på en eller ett fåtal publika IP-adresser. Användningen av NAT och lokala IP-adresser minskar således behovet av publika adresser. [3,6]

NAT har stött på mycket kritik på grund av de många brister metoden lider av. Brister i NAT har uppmärksammats bland annat i *RFC 2993 - Architectural Implications of NAT* [10] och *RFC 3027 - Protocol Complications with the IP Network Address Translator* [11]. Exempelvis strider NAT mot principen att alla nätverksnoder på internet ska vara nåbara med globalt unika IP-adresser. Många kommunikationsprotokoll fungerar inte heller över NAT utan specialarrangemang. Trots alla problem och brister är användningen av NAT idag utbredd och ytterst vanlig, speciellt i hem och mindre företag. Stöd för NAT finns i de flesta modem och routers, och är ofta i bruk automatiskt. [3,6]

Trots användningen av NAT har förbrukningen att IP-adresser inte kunnat stoppas, utan enbart fördröjas. ***TODO:*** *Här fattas några meningar om när IPv4-adresserna tog slut hos ICANN/IANA osv...*

I takt med att tillgången på IPv4-adresser har försvårats har organisationer och internetleverantörer blivit tvungna att kringå problemet. Bland annat har internetleverantörer den senaste tiden börjat använda sig av NAT för kundanslutningar och leverera publika IP-adresser enbart till de kunder som själva efterfrågar sådana [**?**]. Det här tillvägagångssättet har benämningen CGN (eng. *Carrier-Grade NAT*) och kallades tidigare också LSN (eng. *Large-Scale NAT*) [**?**, 12]. CGN kan dock endast ses som en tillfällig lösning tills man börjar erbjuda nativa IPv6-anslutningar [**?**].

2.3 IPv6

TODO: *IPv6-avsnittet fattas helt. Texten här är endast reminders åt mig själv.*

IPv6 är en ny standard...

IPv6 har 128 bitars adresser...

Teoretiskt sett borde IPv6 med sina 128 bitars adresser kunna erbjuda 2^{128}, eller cirka $3{,}4*10^{38}$ unika IP-adresser.

2.4 Kompatibilitet mellan IPv4 och IPv6

TODO: *Kompatibilitet mellan IPv4 och IPv6, eller snarare brist på sådan. Kommer att nämna tunnlar, 6to4, NAT64 osv. generellt, snabbt och ytligt.*

3 Domännamnssystemet (DNS)

Trots att numeriska nätverksadresser fungerar väl för tekniska tillämpningar, kan de knappast anses vara speciellt användarvänliga. De flesta internetanvändarna skulle säkerligen se det som ytterst besvärligt att behöva komma ihåg 32 bitars IPv4-adresser, än mindre 128 bitars IPv6-adresser, för att exempelvis kunna besöka webbsidor. För att uppnå en viss grad av användarvänlighet krävs alltså en annan adresseringsmetod än IP-adresser. Den nuvarande lösningen för detta problem är domännamnssystemet DNS (eng. *Domain Name System*), som uppfanns år 1983. [3]

DNS gör det möjligt att adressera datorer i nätverk med användarvänliga värdnamn istället för IP-adresser, exempelvis genom att använda värdnamnet *www.abo.fi* istället för IPv4-adressen *130.232.212.56*. För att en klient ska kunna använda DNS förutsätts att den har en metod att genomföra namnförfrågningar. Normalt konfigureras en dator med en eller flera (för redundans och/eller lastbalansering) namnserveradresser till vilka förfrågningar kan skickas. Dessa adresser är IP-adresser som oftast pekar på internetleverantörens eller nätverksoperatörens cachande och rekursiva namnservrar. [**?**]

Då en klient vill kontakta en värdadress (t.ex. *www.abo.fi*) börjar den med att skicka en förfrågan till en av de konfigurerade namnservrarna. Ur användarens synvinkel är denna process automatisk och osynlig. Namnservern som får förfrågan tar reda på IP-adressen för den begärda värdadressen (t.ex. *130.232.212.56*) och meddelar denna adress till klienten. Då svaret har tagits emot känner klienten till den gällande IP-adressen för värdadressen, och kan därefter försöka ta kontakt med den. [**?**]

3.1 Domänhierarki

Strukturen för DNS följer en modell där värdnamn är hierarkiskt uppbyggda, och nivåerna i hierarkin separeras med punkter. Trots att värdnamn läses från vänster till höger är hierarkin för dem uppbyggd från höger till vänster. Hierarkin kan också visualiseras som ett träd, där roten är DNS-roten och löven är enskilda värdnamn. DNS är ett distribuerat system där ansvaret

för olika nivåer i DNS-hierarkin kan vara delegerat till olika aktörer, och förfrågningar gällande olika zoner, dvs. olika domäner och underdomäner i hierarkin, kan skötas och besvaras av olika servrar. [3, 6]

För exemplet *www.abo.fi* utgör *.fi* den högsta nivån, Finlands toppdomän. Ansvaret för Finlands toppdomän ligger hos Kommunikationsverket, som administrerar toppdomänen och beviljar domännamn som slutar på *.fi* [13]. Nästa steg i hierarkin är domännamnet *abo.fi*, som tillhör Åbo Akademi [**?**]. Hela adressen *www.abo.fi* utgör ett värdnamn i *abo.fi*-domänen. Värdnamn kan ha flera hierarkiska nivåer, t.ex. utgör adressen *www.cs.abo.fi* ett värdnamn under *cs.abo.fi*, som i sin tur är en underdomän i *abo.fi*-domänen. [3,6]

3.2 DNS-poster

Zoner i DNS-servrar innehåller DNS-poster (eng. *records*), som utgör uppgifter om de värdnamn och/eller underdomäner som zonen innehåller. Varje DNS-post består av ett namn, en tidsgräns i form av ett TTL-värde (eng. *Time To Live*), en klass, en typ och ett värde.

Namnet för en DNS-post utgör den primära söknyckeln vid namnförfrågningar. Namnet består av antingen ett värdnamn eller en domän eller underdomän, beroende på kontexten och posttypen. Exempelvis kan *www* vara namnet för en domänpost i *abo.fi*-zonen.

TTL-värdet används för att ange hur statiskt en DNS-posts innehåll är. Denna information kan utnyttjas för caching och behandlas mera under punkt 3.3.3.

Varje DNS-post har en angiven klass, som avser postens användningsområde. Syftet med detta fält är att göra det möjligt att använda DNS för flera olika ändamål. För all Internet-relaterad information är postklassen *IN*, som står för Internet. Övriga klasstyper omnämns i *RFC 5395 - Domain Name System (DNS) IANA Considerations* [14].

3.2.1 Posttyper

Det finns ett stort antal olika DNS-posttyper som används för olika ändamål. Här nämner vi kort de vanligaste, inklusive *A*- och *AAAA*-poster som utgör

de viktigaste typerna ur vår lösningsmetods synvinkel.

TODO: *Lite snabb info om CNAME, PTR, A, AAAA, MX-poster. Följande rader är reminders åt mig själv.*

En *A*-post innehåller en IPv4-adress för ett värdnamn.

En *AAAA*-post innehåller en IPv6-adress för ett värdnamn.

RFC 3596 - DNS Extensions to Support IP Version 6 [15] innehåller stuff om IPv6 DNS...

RFC 4472 - Operational Considerations and Issues with IPv6 DNS [16] är en annan nice källa.

3.2.2 Skillnaden mellan IPv4 och IPv6

TODO: *Här fattas text om A- vs. AAAA-poster, att de kan existera samtidigt och innehålla olika adresser.*

3.3 Namnservrar

Domämnnamnsystemet upprätthålls med hjälp av tre olika typer av namnservrar som alla uppfyller olika funktioner. Här behandlas kort dessa olika typer av servrar och deras uppgifter.

3.3.1 Rotservrar

Alla namnförfrågningar med DNS görs utgående från DNS-roten som, liksom allokeringen av IP-adresser, administreras av ICANN och deras enhet IANA. DNS-rotens innehåll består av information om alla toppdomäner och adresser till deras auktoritativa namnservrar, som tillsammans utgör den så kallade rotzonen (eng. *root zone*). Rotzonen tillhandahålls av DNS-rotservrarna, som upprätthålls av många olika organisationer. Rotservrarna är 13 till antalet, namngivna med bokstäver från A till M. Varje rotserver har ett värdnamn enligt modellen *bokstav.root-servers.net* (t.ex. *a.root-servers.net*) samt en IPv4-adress. I skrivande stund är 10 av rotservrarna även tillgängliga över IPv6. [3, 6, 17]

Trots att varje rotserver endast har en IP-adress (eller en IPv4-adress och en IPv6-adress) är de i praktiken uppbyggda med redundanta och distribuer-

ade metoder, vilket gör dem kontinuerligt tillgängliga. Största delen av rotservrarna är geografiskt distribuerade över många enskilda sajter med hjälp av *anycast*-teknik, vilket innebär att trafik till serverns IP-adress alltid styrs till den sajt som finns närmast klienten. Mest redundant är L-rotservern, som för tillfället är distribuerad över totalt 95 olika sajter. Tre av rotservrarna (I, J och K) har sajter i Helsingfors. Tillsammans består de 13 rotservrarna idag av totalt 299 olika sajter som tillsammans tillhandahåller DNS-rotzonen. [**?**, 17]

3.3.2 Auktoritativa namnservrar

Principen med delegerat ansvar för olika DNS-zoner innebär bland annat att DNS-rotzonen inte innehåller någon information om enskilda toppdomäners underliggande domäner, utan enbart adresser till toppdomänernas egna DNS-servrar. DNS-servrar som bär ansvaret för en DNS-zon kallas auktoritativa (eng. *authoritative*) för den zonen. Exempelvis är DNS-rotservrarna auktoritativa för rotzonen, medan Kommunikationsverkets servrar är auktoritativa för *.fi*-zonen. För Finlands toppdomän *.fi* innehåller DNS-roten alltså enbart hänvisningar till Kommunikationsverkets DNS-servrar, och exempelvis ingen information om domänen *abo.fi*. [**?**]

Den som har beviljats ett domännamn måste tillhandahålla minst två (för feltolerans och redundans) auktoritativa namnservrar för domänens DNS-zon. Dessa servrar ska vara allmänt tillgängliga och besvara namnförfrågningar för den specifika domänen. Adresser till en domäns auktoritativa namnservrar meddelas till den beviljande organisationen, som inför adresserna i toppdomänens DNS-zon. På detta sätt innehåller t.ex. *.fi*-zonen adresserna till *abo.fi*-zonens auktoritativa namnservrar, som i sin tur upprätthålls av domänens innehavare, i det här fallet Åbo Akademi. [**?**]

Normalt ger en auktoritativ namnserver endast svar som direkt ingår i dess egen zon. Då en förfrågan gällande värdnamnet *www.abo.fi* skickas till en rotserver besvarar den alltså inte hela förfrågan, utan svarar enbart med adresser till *.fi*-zonens auktoritativa servrar. Då samma förfrågan skickas till *.fi*-zonens auktoritativa servrar besvarar de förfrågan endast med adresser till *abo.fi*-zonens auktoritativa servrar. Först när samma förfrågan skickas

till någon av *abo.fi*-zonens auktoritativa servrar fås ett slutligt svar med IP-adresser för värdnamnet *www.abo.fi*. [?]

3.3.3 Cachande och rekursiva namnservrar

Om alla namnförfrågningar skulle behöva göras som ovan, dvs. i flera iterationer med början vid rotservrarna, skulle det kunna leda till vissa problem. Dels tar varje förfrågan tid att utföras och besvaras, dels är det lätt att se att namnservrar högt uppe i DNS-trädet, dvs. främst rotservrar, men också toppdomäners auktoritativa namnservrar, skulle bli belastade med väldigt stora mängder namnförfrågningar. Både för att snabba upp namnförfrågningar och undvika sådana överbelastningar för namnservrarna förlitar sig DNS-systemet långt på caching. [?]

Administratörer för en DNS-zon kan för alla DNS-poster definiera en tidsgräns i form av ett TTL-värde (eng. *Time To Live*). Tidsgränsen anger hur länge auktoritativa poster får cachas, dvs. hur länge ett svar kan mellanlagras och återanvändas före ett nytt auktoritativt svar behövs. Tidsgränsen kan således anses vara ett mått på hur beständiga DNS-posterna är. En post som varierar ofta kan ges en tidsgräns på bara en minut, medan en väldigt statisk post kan ha en tidsgräns på flera dagar. [?]

Rekursiva namnservrar besvarar inkommande namnförfrågan fullständigt. Istället för att besvara en förfrågan med adresser till auktoritativa namnservrar, utför en rekursiv namnserver själv alla iterationer av namnförfrågan, ända tills ett slutligt svar fås. Svaren från de enskilda iterationerna skickas inte till klienten, utan enbart det slutgiltiga svaret. [?]

Att en DNS-server är rekursiv gör att klienter får slutliga svar genom att göra endast en namnförfrågan, medan cachande DNS-servrar kan besvara upprepade förfrågningar snabbt. Tekniskt sett behöver en cachande namnserver inte vara rekursiv eller vice versa. Det normala är dock att internetleverantörer tillhandahåller namnservrar som är både cachande och rekursiva. Rekursiva, cachande namnservrar körs också i många modem och routrar. [?]

3.4 Domännamnssystemets säkerhetstillägg (DNSSEC)

TODO: *Här fattas text om DNSSEC och hela principen med det. Borde komma med, bara jag hinner skriva om det. Är ganska viktigt eftersom vår lösningsmetod inte fungerar med DNSSEC, eftersom vi modifierar eller fakear DNS-svar.*

Domännamnssystemets säkerhetstillägg DNSSEC (eng. *Domain Name System Security Extensions*)...

4 Hypertext Transfer Protocol (HTTP)

TODO: *Här ska HTTP-protokollet behandlas väldigt kort. Viktigast på grund av HTTPS:*

4.1 HTTPS

TODO: *Kunde kort skriva om hur HTTPS fungerar, och hur mycket som krypteras. Speciellt intressant/viktigt eftersom request headers är krypterade, vilket gör att man inte kan köra HTTPS genom transparenta proxyservrar (dock nog genom vanliga sådana). Borde komma med.*

5 Proxyservrar

En proxy eller proxyserver är en server som kan fungera som ombud eller förmedlare för datakommunikation med ett specifikt protokoll. Proxyservrar placeras mellan klienter och servrar, där de kan ta emot förfrågningar från klienter, förmedla förfrågningarna vidare till rätt servrar och slutligen förmedla svaren från servrarna tillbaka till klienterna. Ur klientens synvinkel är proxyn en server, medan proxyn agerar som klient sett ur den riktiga serverns synvinkel. [18, 19]

Normalt tar en webbläsare direkt kontakt till relevant webbserver då användaren försöker öppna en webbsida. Om webbläsaren har konfigurerats för att använda en HTTP-proxy kommer kontakten dock att istället tas till denna, oberoende av med vilken webbsida användaren har begärt kontakt. Alla HTTP-förfrågningar från webbläsaren görs således enbart till den konfigurerade proxyn. Proxyn tar emot alla HTTP-förfrågningar och kontaktar i sin tur den korrekta webbservern med HTTP-protokollet, på motsvarande sätt som webbläsaren själv skulle ha gjort utan proxyserver. Svaret som HTTP-proxyn får från webbservern förmedlas till klienten. I klienten hanteras ett svar från en HTTP-proxy lika som ett svar direkt från en webbserver. [18, 19]

5.1 Nyttan med proxyservrar

Proxyservrar kan utnyttjas för att erbjuda många olika fördelar. Här nämns endast ett par sådana egenskaper, med specifik beskrivning för hur de kan tillämpas på HTTP-proxyservrar. Detta görs för att ge en praktisk överblick av idén med proxyservrar och deras potential.

5.1.1 Caching

Proxyservrar kan erbjuda caching, vilket innebär att en HTTP-proxy själv kan lagra lokala, temporära kopior av förmedlade webbsidor. Caching kan därmed utnyttjas genom att upprepade förfrågningar till samma webbplats kan besvaras direkt av proxyservern. Då samma webbsidor inte behöver begäras på nytt från webbservern kan detta dels minska belastningen på

webbservern, dels på datanätverket mellan proxyservern och webbservern. Proxyservrar kan med andra ord utnyttjas för att minska belastningen på ansträngda eller överbelastade datalänkar, vilket kan vara praktiskt bland annat i mindre företag med långsamma internetuppkopplingar. [18, 19]

5.1.2 Filtrering

Eftersom en proxyserver har kontroll över alla HTTP-förfrågningar som skickas via den kan detta även utnyttjas för filtrering. En proxyserver kan t.ex. ha en lista på otillåtna URL:ar, dvs. adresser till webbsidor. Då en klient försöker besöka en otillåten sida kan proxyn därmed känna av detta fall och hantera situationen genom att t.ex. visa ett felmeddelande. Detta kan exempelvis utnyttjas för att i ett företags HTTP-proxy hindra tillgången till sociala medier, t.ex. Facebook och Twitter, eller för att i en lågstadieskola blockera tillgången till webbsidor som är olämpliga för barn. [3, 19]

5.1.3 Säkerhet

Proxyservrar har inte bara tillgång till de inkommande HTTP-förfrågningarna, utan självklart även till de svar som fås från själva webbservrarna. Genom att analysera innehållet i sådana svar före de skickas till klienterna kan man uppnå avancerad, innehållsbaserad filtrering. På detta sätt kan exempelvis proxyservern utnyttja ett antivirusprogram för att skanna innehållet i förmedlade webbsidor och på så vis automatiskt blockera nerladdning av virus eller trojaner. På samma vis kan man genom aktiv innehållsanalys blockera sidor vars innehåll är förenat med kända metoder för nätfiske (eng. *phishing*). En proxyserver kan med andra ord användas för att upprätthålla en viss nivå av säkerhet i ett nätverk. [19]

5.2 Transparenta proxyservrar

Vid normal användning av proxyservrar förutsätts att klienten konfigureras för ändamålet. Exempelvis kan en HTTP-klient (t.ex. en webbläsare) konfigureras för att göra alla HTTP-förfrågningar till en specifik, medvetet vald

proxyserver. I större nätverk med många klienter kan det dock vara tidskrävande eller i praktiken omöjligt att på ett ändamålsenligt sätt konfigurera alla klienter i nätverket. Ett alternativ till traditionell konfigurering för proxyanvändning är att implementera proxyn som en del av nätverksinfrastrukturen, som en transparent proxyserver. [19]

En transparent proxyserver är en proxyserver som har placerats så att all relevant trafik automatiskt styrs till och genom denna. Exempelvis kan en router konfigureras så att all HTTP-trafik styrs till en HTTP-proxy eller så att DNS-trafik styrs till en DNS-proxy. Detta gör det möjligt att påtvinga användningen av en proxyserver utan att den aktivt behöver konfigureras i klienterna. Användningen av en proxyserver blir således osynlig för användarna och klienterna, som inte ens behöver vara medvetna om att en proxyserver existerar. [19]

Transparenta proxyservrar kallas även uppfångande eller avlyssnande proxyservrar (eng. *intercepting proxies*) [19]. En annan, kanske ännu bättre beskrivande term, är påtvingade proxyservrar (eng. *forced proxies*) [**?**].

6 Lösningsmetod

TODO: *Lösningsmetoden är klar principiellt sett. De franska strecken nedan fungerar som guide för mitt eget skrivande. Här kommer vi att resonera oss fram till den slutliga lösningen, och samtidigt hantera de problem som uppstår på vägen.*

Den föreslagna metoden består av två delar; en HTTP-proxyserver samt en cachande rekursiv DNS-server. I det här kapitlet förklaras dessa servrars funktionalitet i detalj. Huvudproblemet vi försöker lösa är att klienter med enbart IPv4-nätverksanslutning inte kan kommunicera direkt med IPv6-nativa värdar.

IDEAS:

- IPv4 kan inte connecta direkt till IPv6

=> HTTP-proxyserver med både IPv4 och IPv6

- HTTP-proxyservern kan ta kontakt till IPv6-tjänster utanför det lokala nätverket

- IPv4-klienter kan internt ta kontakt till HTTP-proxyservern

=> IPv4-klienter kan kontakta IPv6-webbtjänstser via proxyn

- Förutsätter aktiva åtgärder i alla IPv4-klienter (konfigurera användning av proxyn)

- Går kanske inte att göra i alla klienter (t.ex. gamla applikationer eller inbyggda system kanske inte stöder proxyn)

=> Transparent/forced HTTP-proxy

- Hijacka all trafik på TCP port 80, styr den till HTTP-proxyn

- Kräver inga aktiva åtgärder i klienterna

- Kontakt till IPv6-webbtjänster från IPv4-klienter kommer att misslyckas eftersom IPv4-klienterna inte förstår DNS-svar (AAAA records) för IPv6-värdar

=> Modifierad cachande, rekursiv DNS-server

- Ger IPv4-uppgifter (A record) (= IP:n till den lokala HTTP-proxyn) för lookups som på riktigt bara har IPv6-uppgifter (AAAA records)

- HTTP-proxyn behöver inte vara transparent/forced, IPv4-klienter kan ansluta direkt till IPv4-webbtjänster.

- Förutsätter att klienterna använder rätt DNS-server, om de använder t.ex. OpenDNS eller Googles DNS-servrar kommer det inte att fungera

=> Transparent/forced DNS-proxy

- Hijacka all trafik på UDP port 53, styr den till den modifierade cachande, rekursiva DNS-servern

- Dålig praxis, har fått massiv negativ kritik i samband med SOPA/PIPA/ACTA och blockeringar av webbsidor genom DNS-filtrering

- Fungerar inte med DNSSEC

- Fungerar inte med HTTPS

- Fungerar enbart för HTTP

(Long story short, en hel del problem med HTTPS och DNSSEC...)

TODO: *En annan potentiell lösning, som jag avser diskutera i korthet, men som är dålig på grund av non-standard namespace:*

Alternativ: IPv4Gate @ http://www.sixxs.net/tools/gateway/

- Surfa till adresser http://whatever.ipv4.sixxs.org, t.ex. http://ipv6.google.com.ipv4.sixxs.org/.

- All trafik går okrypterad via tredje part = säkerhetsrisk

- Non-standard namespace (styr inte automatiskt t.ex. http://ipv6.google.com/ via gatewayen) = client side scripting, t.ex. AJAX kan ha problem med detta (sidan nåbar bara via annan URI än eventuellt hardcoded).

=> Tillämpa en egen lokal motsvarighet, t.ex. "ipv4gw.local"

- Minskar säkerhetsrisken, namespace-problemet kvarstår.

7 Sammanfattande diskussion

TODO: *Som rubriken säger. Som det ser ut nu kan vi konstatera att metoden går att tillämpa för en relativt kostnadseffektiv, tillfällig lösning inom en organisation där man inte har möjlighet att i nuläget ta i bruk IPv6 fullständigt. På grund av problemen med HTTPS (viktigt!) och DNSSEC (mindre viktigt, men värt att notera) är det däremot inte någon permanent lösning.*

Lista över förkortningar

CGN	Carrier-Grade NAT
DNS	Domännamnssystemet (eng. *Domain Name System*)
DNSSEC	Domännamnssystemets säkerhetstillägg (eng. *Domain Name System Security Extensions*)
HTTP	Hypertext Transfer Protocol
IANA	Internet Assigned Numbers Authority
ICANN	Internet Corporation for Assigned Names and Numbers
IETF	Internet Engineering Task Force
IP	Internetprotokollet (eng. *Internet Protocol*)
IPv4	Version 4 av internetprotokollet (eng. *Internet Protocol version 4*)
IPv6	Version 6 av internetprotokollet (eng. *Internet Protocol version 6*)
LSN	Large-Scale NAT
NAT	Nätverksadressöversättning (eng. *Network Address Translation*)
RIPE RCC	Réseaux IP Européens Network Coordination Centre
RIR	Regional Internet Registry
TTL	Time To Live
URL	Uniform Resource Locator

Referenser

[1] E. Gerich, "Guidelines for Management of IP Address Space," RFC 1466 (Informational), Internet Engineering Task Force, May 1993, obsoleted by RFC 2050. [Online]. Available: http://www.ietf.org/rfc/rfc1466.txt

[2] IAB and IESG, "IPv6 Address Allocation Management," RFC 1881 (Informational), Internet Engineering Task Force, Dec. 1995. [Online]. Available: http://www.ietf.org/rfc/rfc1881.txt

[3] A. Tanenbaum and D. Wetherall, *Computer Networks.* Pearson, 2010.

[4] J. Postel, "DoD standard Internet Protocol," RFC 760, Internet Engineering Task Force, Jan. 1980, obsoleted by RFC 791, updated by RFC 777. [Online]. Available: http://www.ietf.org/rfc/rfc760.txt

[5] ——, "Internet Protocol," RFC 791 (Standard), Internet Engineering Task Force, Sep. 1981, updated by RFC 1349. [Online]. Available: http://www.ietf.org/rfc/rfc791.txt

[6] L. Peterson and B. Davie, *Computer networks: a systems approach*, ser. The Morgan Kaufmann Series in Networking. Morgan Kaufmann, 2007.

[7] M. Cotton and L. Vegoda, "Special Use IPv4 Addresses," RFC 5735 (Best Current Practice), Internet Engineering Task Force, Jan. 2010. [Online]. Available: http://www.ietf.org/rfc/rfc5735.txt

[8] K. Egevang and P. Francis, "The IP Network Address Translator (NAT)," RFC 1631 (Informational), Internet Engineering Task Force, May 1994, obsoleted by RFC 3022. [Online]. Available: http://www.ietf.org/rfc/rfc1631.txt

[9] P. Srisuresh and K. Egevang, "Traditional IP Network Address Translator (Traditional NAT)," RFC 3022 (Informational), Internet Engineering Task Force, Jan. 2001. [Online]. Available: http://www.ietf.org/rfc/rfc3022.txt

[10] T. Hain, "Architectural Implications of NAT," RFC 2993 (Informational), Internet Engineering Task Force, Nov. 2000. [Online]. Available: http://www.ietf.org/rfc/rfc2993.txt

[11] M. Holdrege and P. Srisuresh, "Protocol Complications with the IP Network Address Translator," RFC 3027 (Informational), Internet Engineering Task Force, Jan. 2001. [Online]. Available: http://www.ietf.org/rfc/rfc3027.txt

[12] S. Jiang, D. Guo, and B. Carpenter, "An Incremental Carrier-Grade NAT (CGN) for IPv6 Transition," RFC 6264 (Informational), Internet Engineering Task Force, Jun. 2011. [Online]. Available: http://www.ietf.org/rfc/rfc6264.txt

[13] Kommunikationsverket. (2011, Feb.) Fi-domännamn. [Online]. Available: http://www.viestintavirasto.fi/sv/index/internet/fi-verkkotunnukset.html

[14] D. Eastlake 3rd, "Domain Name System (DNS) IANA Considerations," RFC 5395 (Best Current Practice), Internet Engineering Task Force, Nov. 2008, obsoleted by RFC 6195. [Online]. Available: http://www.ietf.org/rfc/rfc5395.txt

[15] S. Thomson, C. Huitema, V. Ksinant, and M. Souissi, "DNS Extensions to Support IP Version 6," RFC 3596 (Draft Standard), Internet Engineering Task Force, Oct. 2003. [Online]. Available: http://www.ietf.org/rfc/rfc3596.txt

[16] A. Durand, J. Ihren, and P. Savola, "Operational Considerations and Issues with IPv6 DNS," RFC 4472 (Informational), Internet Engineering Task Force, Apr. 2006. [Online]. Available: http://www.ietf.org/rfc/rfc4472.txt

[17] (2012) Root server technical operations assn. [Online]. Available: http://www.root-servers.org/

[18] C. Shiflett, *HTTP developer's handbook*, ser. Developer's library. Sams, 2003.

[19] D. Gourley and B. Totty, *HTTP: the definitive guide*, ser. Definitive Guides. O'Reilly, 2002.