Adressöversättning

- Det här kapitlet behandlar adressöversättning relativt djupt. Olika tekniker som NAPT och PAT tas upp. Likaså olika varianter av NAPT för att förklara problem som uppstår på grund av adressöversättning. Kapitlet är ett utdrag, konfigurationsexempel har utelämnats.
- Kapitlet fungerar bäst om du vet hur TCP/IP fungerar.

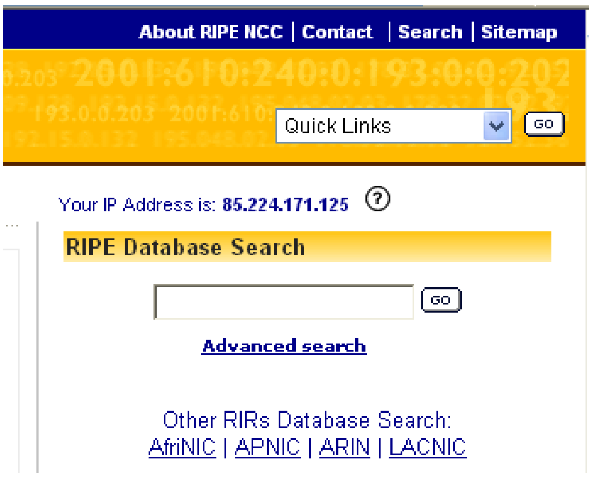
Att använda adressöversättning har blivit det vanligaste sättet att ansluta sig till Internet för privatpersoner och företag. Få bekymrar sig om att adressöversättning bryter mot ett av fundamenten inom IP: kommunikation ska ske ”end-to-end”. Detta gör att flera applikationsprotokoll utvecklats helt utan hänsyn till adressöversättning. Ur dessa utvecklares ögon slutar Internet vid den publika adressen. Om datorn eller gatewayen med en publik adress sedan skickar dataflödet vidare till en annan nod så sker detta inte på Internet längre.

Adressöversättning bryter mot principen att kommunicera ”end-to-end”.

Men adressöversättning har fördelar. Det råder delade meningar om hur ont det är om IP-adresser, men fyra miljarder adresser är en begränsning och de som delar ut adresser idag är restriktiva med utdelning av publika adresser, se kapitlet IP-nivån. Två andra skäl till att adressöversättning slagit igenom är säkerhet och administration. Med adressöversättning har din dator bytt adress på Internet och är inte en del av Internet. Alltså kan inte heller en inkräktare skicka paket hur som helst till din dator. Med publika adresser följer krav på dokumentation. Om ditt nät har vuxit ur de publika adresser du tilldelats kan det också bli svårt. Du tvingas byta adresser på 254 noder när den 255:e ska installeras. Problemet finns i princip även med privata adresser men väljer du adresser ur nätet 10.0.0.0 så har du ett par miljoner adresser att arbeta med.

Prova om du använder adressöversättning. Med kommandona ipconfig eller ifconfig kan du se den IP-adress som din dator har just nu. Men hur ser det ut på Internet? Surfa till www.ripe.net och se vilken IP-adress de ser på Internet. (Andra sajter som visar din externa IP-adress är till exempel whatsmyip.org eller www.ipv6.org.) Om du har en annan adress externt än internt så används adressöversättning.

Testa själv



Your IP Address is: 85.224.171.125

Det finns flera webbsidor som visar vilken IP-adress som efterfrågat webbsidan. Ett exempel är www.ripe.net (beskuren ovan).

Adressöversättning består egentligen av flera delar:

- mekanismen för att byta adress och eventuellt även portnummer
- privata adresser
- översättning av IP-adresser på applikationsnivå

Privata adresser

Adressöversättning har sina största förtjänster om det kombineras med privata adresser. Det finns flera privata adressblock, men de vanligaste är de som definierats i RFC 1918 ”Address Allocation For Private Internets”:

10.0.0.0 – 10.255.255.255
172.16.0.0 – 172.31.255.255
192.168.0.0 – 192.168.255.255

Dessa adressblock kallas även RFC 1918-adresser och svarta adresser. Tanken är att dessa adresser aldrig ska routas vidare ut på Internet. Dessa adressblock ska istället kunna återanvändas gång på gång. Ett företag kan ha tusentals noder, men utåt sett används bara en eller ett fåtal publika IP-adresser.

Adressöversättning är fullt möjlig även utan privata adresser. Tekniskt sett kan en organisation använda en annan organisations adresser internt bara de översätts när de skickas ut på Internet. Tekniken benämns ibland ”masquerading” och ligger nära spoofing. (Spoofing innebär att man medvetet byter IP-adresser. En nod kan störas ut med en massa paket, för att inte kunna spåra vem som skickat dessa byter angriparen avsändaradress.) Med RFC 1918 blev det lite mer ordning på detta och risken att adresser ska dyka upp på fel ställe minskade betydligt.

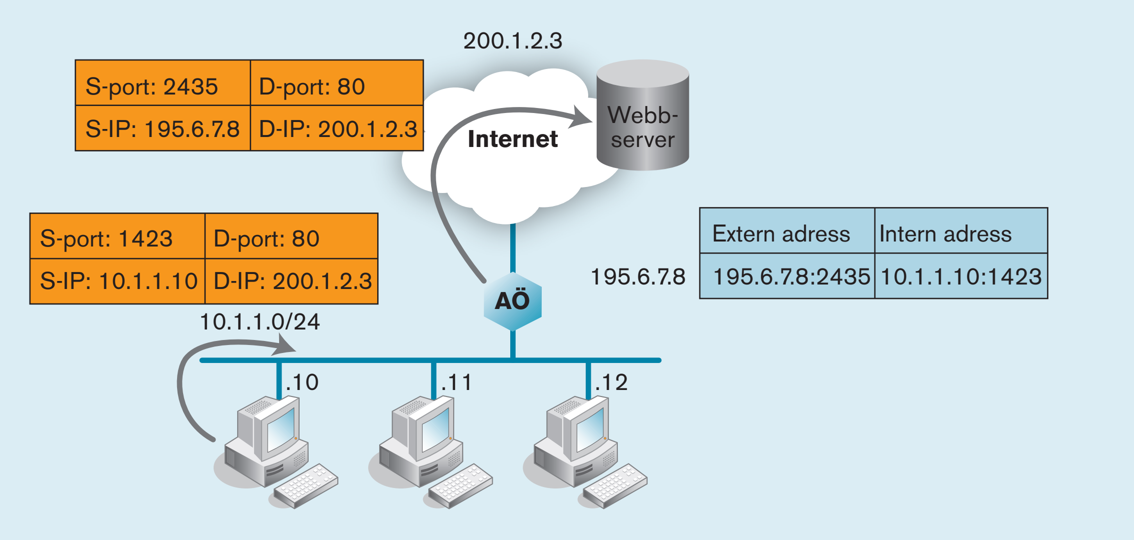
Adressöversättning fungerar även utan privata adresser.

NAPT

Det finns flera olika tekniker för adressöversättning och flera olika akronymer som bland annat NAT, NAPT, NPAT och PAT. De används lite olika av olika tillverkare (till exempel Cisco) och beskrivs olika i olika publikationer. Även termen masquerading används för generell adressöversättning. På datakomjargong används adjektivet ”nattad adress” men exakt vilken teknik som avses är ofta oklart. I detta kapitel används begreppen huvudsakligen i enlighet med RFC 2663 och 3022.

NAPT (Network Adress Port Translation) är den vanligaste formen av adressöversättning. Den kallas tyvärr ofta bara NAT. Med NAPT kan flera hundra noder dela på en extern IP-adress. Att det fungerar är inte mycket konstigare än att man på en dator kan starta fler fönster (eller processer) och hämta ner filer från fler olika servrar. Med hjälp av portnummer kan man adressera rätt program eller process.

NAPT är den vanligaste formen av adressöversättning. Och den benämns ofta bara NAT.



NAPT innebär att avsändarens IP-adress och portnr byts. AÖ står för adressöversättare.

Webbklienten 10.1.1.10 skickar ett paket till den externa webbservern 200.1.2.3. Webbklienten har adressöversättaren som default gateway. Adressöversättaren (AÖ) är konfigurerad för att byta adresser (en tabell där den externa adressen 195.6.7.8 ersätts med 10.1.1.10 och vice versa). Med NAPT kommer även portnummer i avsändarfältet att bytas. Adressöversättaren lägger även till en ny rad i sin tabell, den externa adressen 195.6.7.8:2435 ska mappas mot 10.1.1.10:1423.

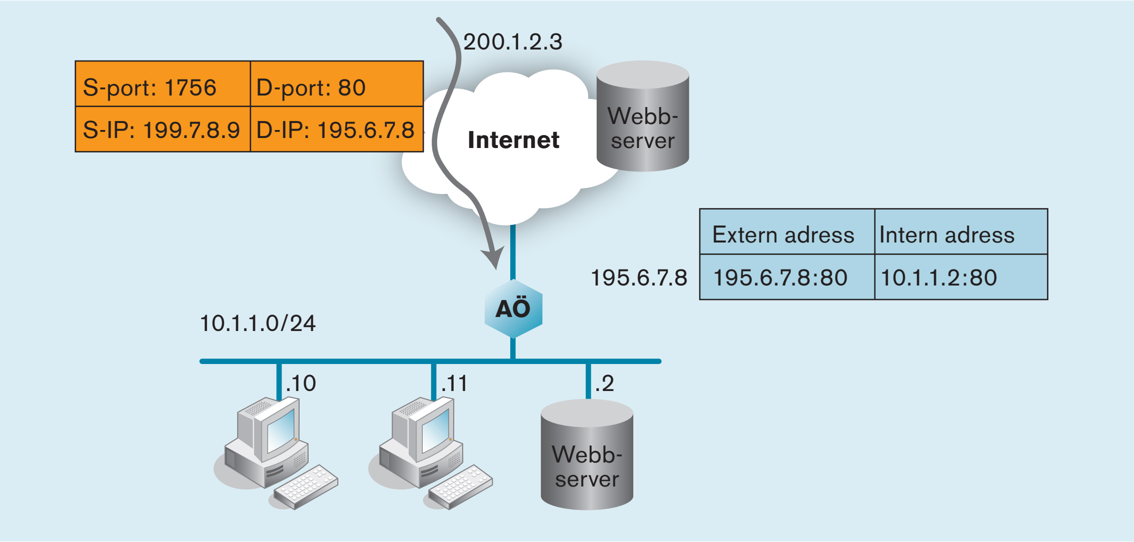
När paket kommer tillbaka från en server på Internet kan adressöversättaren utifrån portnumret avgöra vilken IP-adress och port på den interna sidan som ska ha paketet. Varje gång ett paket passerar som använder aktuell rad i tabellen så uppdateras en timer.

När timern räknat ner till noll raderas raden, timern arbetar i storleksordningen minuter. Adressöversättaren kan när TCP används även analysera sessionen och leta efter FIN och RST för att radera en sessionspost.

Webbservern uppfattar kommunikationen som en session initierad på adressöversättarens externa adress, det är denna adress som sparas i loggar och dylikt. En nod kan starta tusentals sessioner, det går inte att avgöra om de hanteras internt eller om de distribueras vidare på ett annat nätverk.

Den här typen av adressöversättning bygger på att datorn med en privat adress är den som initierar kommunikationen. Den fungerar alltså utmärkt när en webbklient ska hämta en webbsida från en server. Om kommunikationen istället skulle initieras från Internetsidan fungerar inte modellen. För servrar och peer-to-peer-nät behövs någonting annat.

Port Forwarding eller Port Address Translation – PAT



S-port: 1756	D-port: 80
S-IP: 199.7.8.9	D-IP: 195.6.7.8

Extern adress	Intern adress
195.6.7.8:80	10.1.1.2:80

Adressöversättaren konfigureras med hur port 80 ska skickas vidare.

Port Forwarding är ett sätt att göra datorer tillgängliga på Internet för anrop. I det här fallet konfigurerar vi adressöversättaren så att varje gång den får en anropsbegäran på port 80 så skickar adressöversättaren frågan vidare till 10.1.1.2 på insidan. Konfiguration kan sättas upp port för port så till exempel anrop på port 23 skickas till en annan privat adress. Port Forwarding fungerar som NAPT med den skillnaden att adressöversättaren konfigureras med hur portar ska översättas.

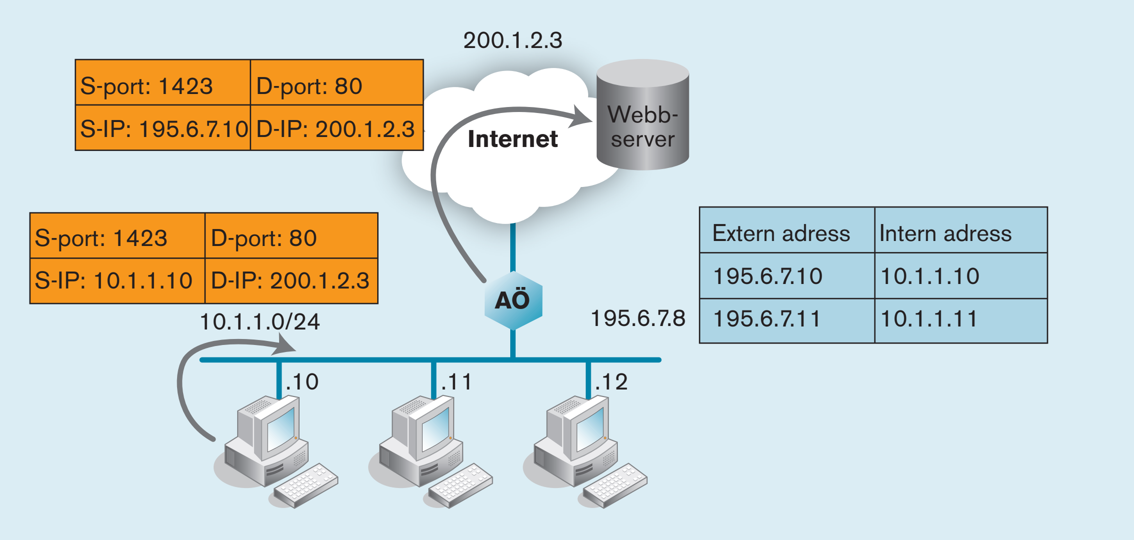
Tekniken har begränsningar. Hur ska vi konfigurera om vi vill sätta upp två publika webbservrar? Bägge använder samma välkända portnummer 80. Den vanligaste lösningen är att vi använder en ny extern port. Till exempel 79 som vi styr om på insidan till exempelvis 10.1.1.4:80. En annan lösning är att adressöversättaren tillåter att vi har flera externa adresser som vi kan styra till olika interna adresser.

Begreppet PAT är inte helt konventionellt, man kan se tekniken som vanlig NAPT med den egenskapen att portöversättningen konfigureras för hand.

En del tillverkare använder begrepp som DMZ eller virtuell server då en dator på insidan får ta emot alla anrop från utsidan. Ur ett säkerhetsperspektiv sitter denna nod direktansluten mot Internet på utsidan och ska ha en säkerhetsnivå som motsvarar detta. Enligt min personliga åsikt bör man använda denna lösning med försiktighet. Noder som sitter på Internet logiskt sett bör även göra det fysiskt.

Network Adress Translation – NAT

En annan adressöversättningsteknik använder inte byte av portnummer utan arbetar bara på IP-nivå. Adressöversättaren innehåller en tabell som antingen innehåller en tabell hur varje IP-adress ska översättas statiskt i förhållandet ett till ett, eller en pool av adresser som dynamiskt ska användas externt.



Extern adress	Intern adress
195.6.7.10	10.1.1.10
195.6.7.11	10.1.1.11

Adressöversättning på IP-nivå.

Denna typ av adressöversättning kräver lite mer konfigurationsarbete men möjliggör att flera servers av samma typ kan nås utifrån.

Tekniken kan också råka ut för att poolen av externa IP-adresser tar slut.

Observera, som sagt, att begreppet NAT ofta används även på tekniken NAPT. (I RFC 2663 och 3022 benämns denna teknik som Basic NAT.)

Adressöversättning ger ett visst skydd

En nod som sitter bakom en adressöversättare är svår att nå från Internet. Om en inkräktare försöker skicka paket till IP-adress 10.1.1.10 så stoppas det av Interleverantörer på vägen. Försöker man istället skicka paket till den externa adressen så finns det normalt bara slumpartade klientportar upplagda och de har redan en session igång så de svarar inte på vanliga SYN-anrop.

När virusen Sasser och Blaster härjade som värst i början av 2000-talet hade man problem på stora nätverk med publika adresser. När nya maskiner driftsattes hann man inte uppdatera operativsystemen innan maskinerna blev smittade. En lösning blev då att först sätta maskinerna på privata nät, uppdatera operativsystemen via Internet och sedan flytta dem till det publika nätet. Detta förfarande är fortfarande ”god sed” vid driftsättning av nya maskiner.

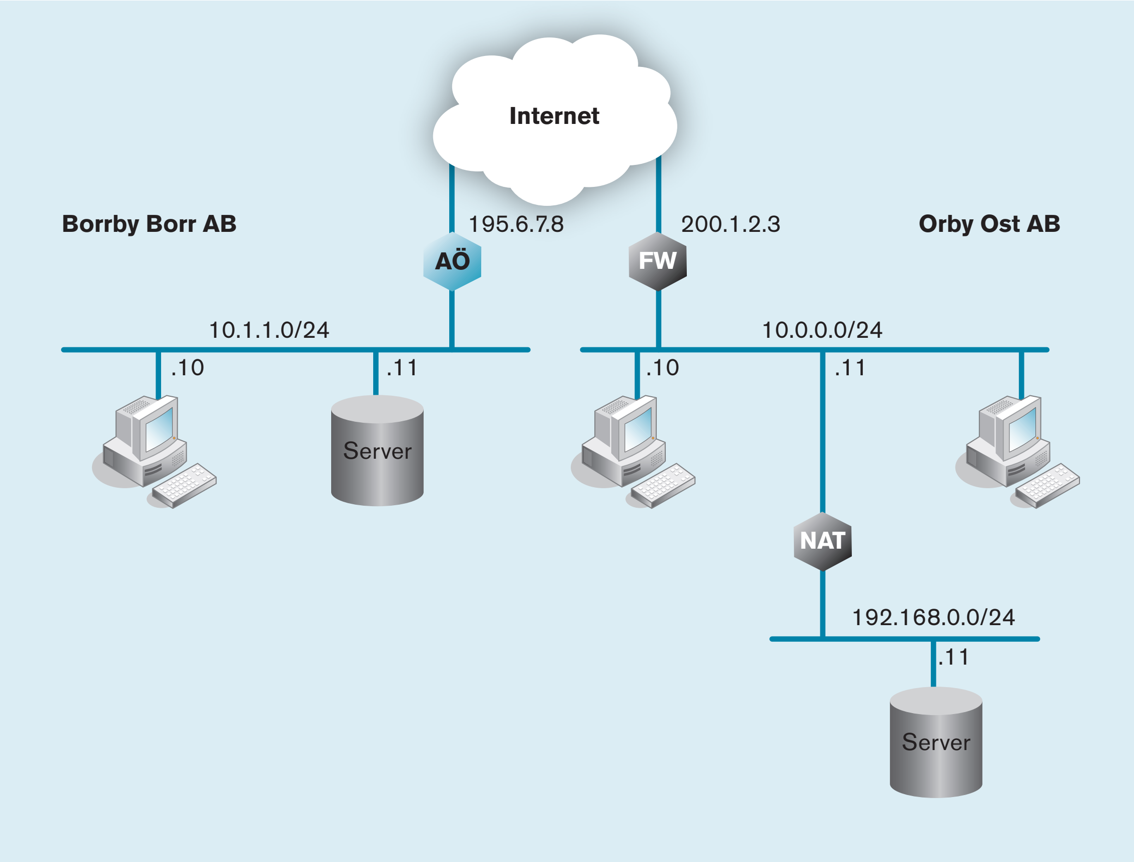
Problem med adressöversättning

Adressöversättning slog igenom snabbt på 1990-talet trots problem och tveksamheter. Citat från RFC 1631 från år 1994:

> *NAT may be a good short term solution to the address depletion and scaling problems. This is because it requires very few changes and can be installed incrementally. NAT has several negative characteristics that make it inappropriate as a long term solution, and may make it inappropriate even as a short term solution. Only implementation and experimentation will determine its appropriateness.*

Det som var en förtjänst med Internet, snabb kommunikation nod till nod kan idag bli mycket komplicerat.

Borrby Borr och Orby Ost vill koppla ihop sina affärssystem. Med publika adresser hade det varit lätt, speciellt innan brandväggarnas tid. Nu krävs en del arbete. Vi måste ta reda på vilka portar respektive affärssystem använder och öppna för dem i brandväggarna samt adressöversättarna. I Orby Ost-nätet har man dess-



utom dubbel adressöversättning, tekniskt sett fungerar detta men vid felsökning blir det svårarbetat. Ofta har också företag outsourcat drift av brandväggen, varje ändring ska lämnas på en viss blankett tre arbetsdagar i förväg för att sedan först ifrågasättas av driftleverantören som till slut motvilligt konfigurerar om enheten, och sedan toppar med att göra det felaktigt. Det blir en ny blankett och flera arbetsdagar till. Att få ihop en sådan hör koppling kan i praktiken ta flera veckor.

Adressöversättning gör att det kan bli svårt att nå tjänster mellan företag.

En annan liten avart som dykt upp är att operatörer börjat använda privata adresser. Vid en traceroute ser man resultat liknande:

```
C:\>tracert www.chalmers.se

Spårar väg till www.chalmers.se [129.16.221.8]
över högst 30 hopp:

1     1 ms     1 ms     1 ms  192.168.3.1
2    44 ms    31 ms    31 ms  10.244.128.193
3    21 ms    21 ms    21 ms  vlan6.sto21.se.isp.com
                             [195.54.116.245]
4    21 ms    21 ms    21 ms  sto2.se.isp.com [195.54.116.34]
```

Traceroute.

Operatörens användning av privata adresser medför dels att man riskerar krockar, operatörens kunder kanske tänkte använda nät 10.244.0.0/16 till något annat. Det medför också att inkommande översättning till servers (PAT) kräver medverkan från operatören.

Med dagens användning av IPv4 har vi gjort oss beroende av adressöversättning.

Två andra problem med adressöversättning är så intressanta att de behandlas separat: applikationer och ”wrap around”. Trots de problem som finns med adressöversättning så är adressöversättning mycket vanligt idag. En stor del av den nätverkutrustning som säljs klarar inte ens publika adresser på det interna nätet – adressöversättningen är alltid påslagen. Adressöversättning skyddar också svaga operativsystem och dåligt uppdaterade maskiner, vi har mer eller mindre gjort oss beroende av den.

Klienter som loggar på

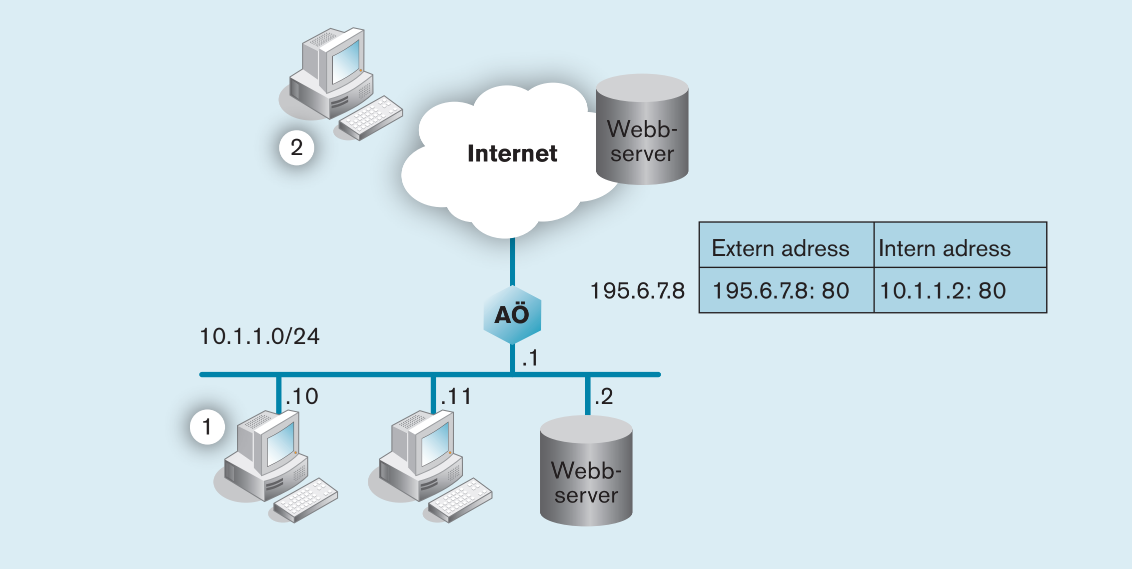
Många tjänster som innebär att en klient ska kunna anropas från Internet bygger på att klient-programmen registrerar sig på en publik server.

Adressöversättning har blivit så vanligt att tillverkarna måste få serverbaserade tjänster att fungera utan ”port forwarding” eller ”virtuella servrar”. En vanlig lösning är att tillverkarna lägger in att programmet alltid startas vid systemstart. Programmet börjar då med att anropa en server med en publik adress. Denna fråga passerar i princip alla brandväggar. Den publika servern håller reda på vilken IP-adress och port som skickade frågan och ute på Internet registreras det att ”Håkan Lindberg is on line”. Vill någon kontakta mig går frågan via den publika servern som skickar frågan vidare till min adressöversättare som vet vilken port och adress som ska få frågan på insidan.

Denna teknik fungerar givetvis men bygger på att klientprogrammen har en lista med kända servrar eller att användarna själv väljer en server. En annan lösning är att de klienter som sitter med publika adresser i princip blir mellanhänder (proxies) för klienter som använder adressöversättning.

Wrap around-problemet

I följande fall har vi en server med den interna IP-adressen 10.1.1.2. Vi har även konfigurerat upp en portöversättning så servern kan nås från Internet på IP-adress 195.6.7.8. Detta fungerar utmärkt men för noder som flyttar sig från den interna nätet till Internet eller åt andra hållet blir det problem. Vilken IP-adress som ska användas beror på var man befinner sig. Vilken IP-adress ska vi använda om vi vill nå vår interna server?



Extern adress	Intern adress
195.6.7.8: 80	10.1.1.2: 80

En server nås med två olika IP-adresser.

Klassik Internetteknik ger ett enkelt svar: vi ska använda den publika adressen. Det är bara den här adressen som är unik och som kan användas vid routing etc. Flera adressöversättare klarar att hantera det här problemet, som ofta kallas ”wrap around”. När en fråga kommer från insidan (fall 1 i bilden ovan) efter IP-adress 195.6.7.8 så översätts frågan. Adressöversättaren skickar ett paket från 10.1.1.1 till 10.1.1.2 som returneras av webbservern till 10.1.1.1. Lösningen i sig blir inte optimal, trots att klient och server sitter på samma nät så går alla frågor via default gateway.

Vissa adressöversättare kan inte hantera det här problemet, de skickar inte om frågan på det privata nätet. Då får man lösa det via DNS eller ett routingkommando. En lösning är att ha två namnservers, en på insidan om adressöversättaren och en annan på utsidan. Dessa två namnservers ger olika svar på frågan vad servern har för IP-adress.

Problemet med ”wrap around” är generellt på Internet men det saknas ett vedertaget begrepp för det. Det är även svårt att ta reda på hur tillverkare stöder denna funktion. Problemet kommer knappast minska då användningen av privata adresser ökar och man samtidigt vill göra servers tillgängliga från utsidan.

Adressöversättning och applikationer

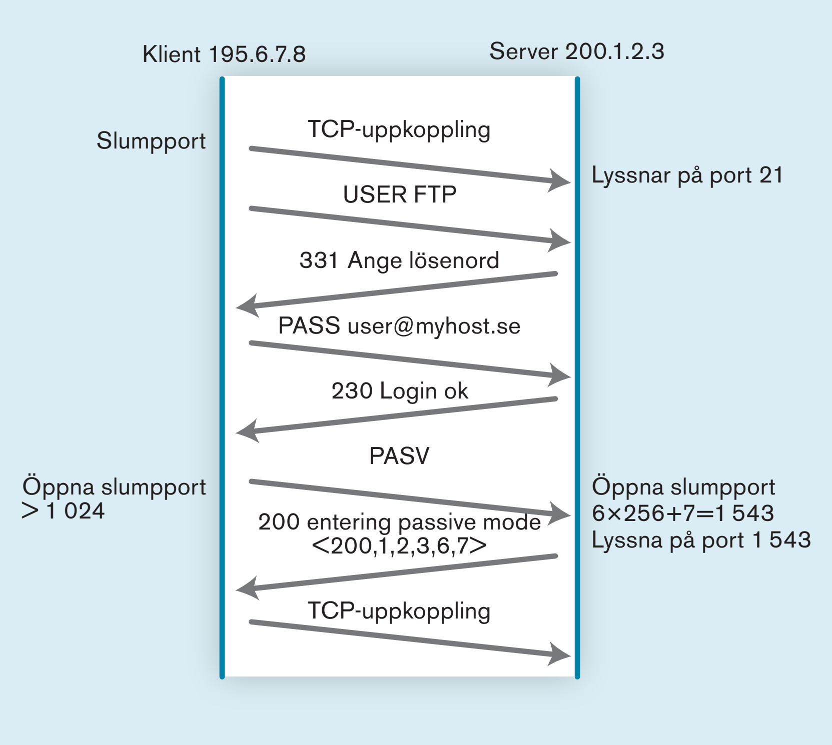
En adressöversättare behöver kunna mer än att byta adresser. I och med att IP-adressen ändras måste IP-headerns checksumma räknas om. Faktum är att även TCP-headerns checksumma behöver räknas om. Ett protokoll som får alldeles uppenbara problem med adressöversättning är ICMP. ICMP:s uppgift är ju att skicka meddelanden när till exempel routing inte fungerar. Om vi till exempel tänker oss ett felmeddelande av typ ”Destination unreachable” så bör en adressöversättare titta igenom innehållet och eventuellt byta ut adresser.

Det finns även en mängd applikationer och protokoll som stöter på problem vid adressöversättning: FTP, NetBIOS, Kerberos, SNMP, IPsec, SIP med flera. Problemet ligger huvudsakligen i att IP-adresser skickas som data på nivå 5–7. I flera fall kan problemet lösas ifall adressöversättaren har stöd för en så kallad applikationsgateway (ALG – Application Layer Gateway). En annan lösning är självklart att använda protokollen internt och inte låta dem passera en adressöversättning överhuvudtaget, detta gäller i viss mån NetBIOS och SNMP. Fler lösningar finns, se referenslistan.

För att belysa problemets komplexitet kan vi titta på hur FTP fungerar.

Oavsett om Passive Mode används eller inte (se kapitlet IP-baserade program) så skickas IP-adresser över på applikationsnivå. (Formatet som används är att 200,1,2,3,6,7 motsvarar IP-adress 200.1.2.3 portnummer 6 × 256 + 7 = 1 543.) IP-adresser skickas alltså inte i hexadecimalt format. Detta medför att en översättning kan ändra på längden (från 200.1.1.3 till 192.193.194.195). Inte bara checksummorna måste räknas om, datamängden påverkar även hur sekvens- och kvittensummer beräknas på TCP-nivån.

En applikationsgateway som säger sig hantera FTP måste hantera ett flertal olika fall: traditionell (aktiv) FTP och Passive Mode, bakomliggande servrar och bakomliggande klienter. Och då är FTP ett ganska enkelt protokoll. Adressöversättning behandlas



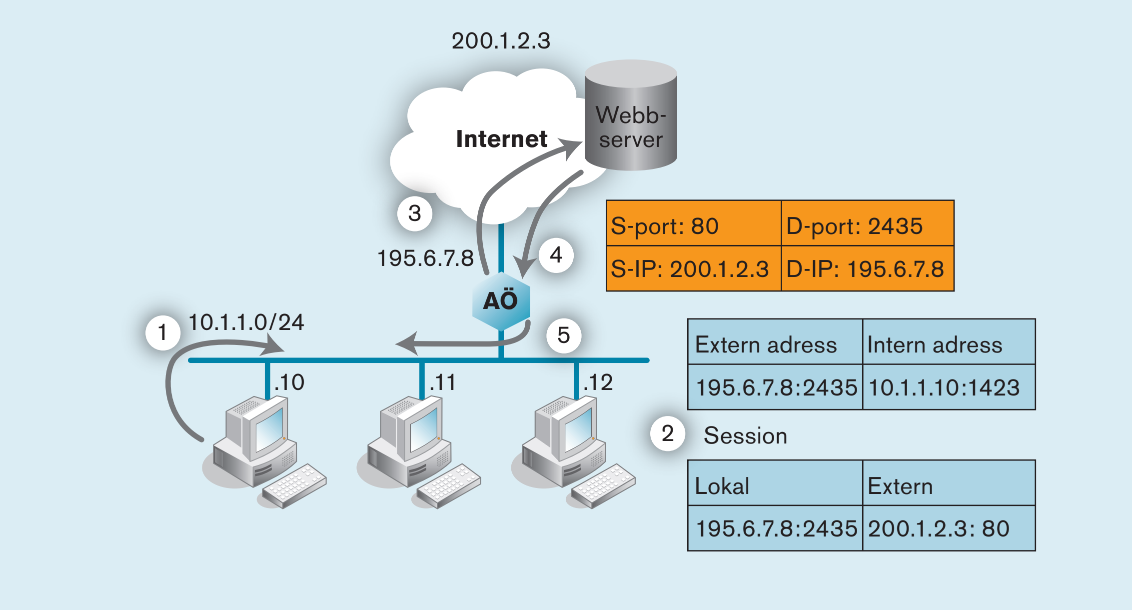
Handskakning under en FTP-sesssion. I detta fall med passive mode.

även i avsnitten om IP-telefoni och VPN. Här blir problemen ibland så komplexa att man delvis måste försöka undvika adressöversättning.

Fördjupning: tre olika former av NAPT

IP-telefoni är ett användningsområden som ökat och ökar. IP-telefoni har som sagts ovan problem med adressöversättning. För IP-telefoni används huvudsakligen transportprotokollet UDP, detta gäller SIP (Session Initiation Protocol) såväl som RTP (Real-time Transport Protocol). UDP är en utmaning inom adressöversättning då protokollet är förbindelselöst, det finns inga egentliga sessioner att hålla reda på. Istället bygger man funktionen kring timers (ett paket som kommer från en avsändare motsvarande det anrop adressöversättaren skickat ut för ett par sekunder sedan kan härledas till motsvarande privat adress och port).

Det finns tre huvudsakliga former av NAPT, vi ska ta och undersöka dem lite närmare. En adressöversättare startar nya sessioner mot de servrar den arbetar med (motsvarande information som man ser på en enstaka nod om man ger den kommandot netstat -an). Vi lägger därför till en tabell med fälten ”lokal” respektive ”extern” och noterar IP-adress och portnummer. Vi betecknar tabellen session, tekniskt sett finns ingen session om UDP används men vi hanterar som sagt detta med timers.



S-port: 80	D-port: 2435
S-IP: 200.1.2.3	D-IP: 195.6.7.8

Extern adress	Intern adress
195.6.7.8:2435	10.1.1.10:1423

Lokal	Extern
195.6.7.8:2435	200.1.2.3: 80

Bild: NAPT och adressöversättarens externa sessioner.

Den vanligaste typen av NAPT kallas för strutformad (full cone) . I samband med att paketet skickas ut externt lägger adressöversättaren upp en datapost med extern och intern IP-adress och port (punkt 2 i bilden). Efter att adressöversättaren lagt upp denna post så accepterar adressöversättaren alla inkommande paket på port 2 435 och skickar dem vidare till 10.1.1.10.

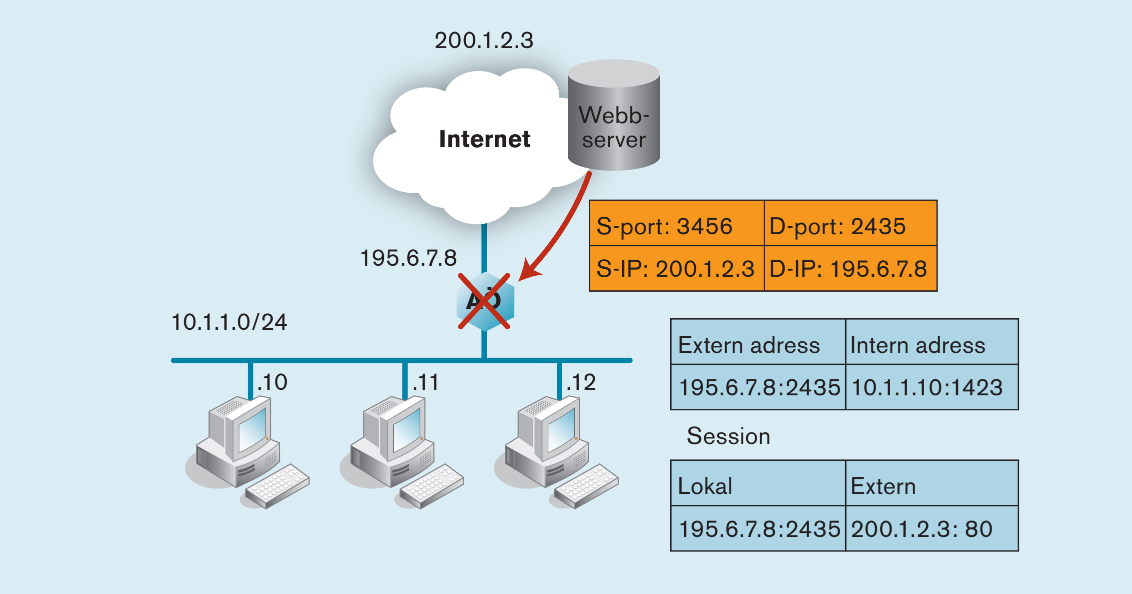
En annan typ av NAPT kallas för strikt, även begreppen partiell eller begränsad förekommer (restricted cone). Adressöversättaren accepterar alla inkommande paket på port 2 435 och skickar dem vidare förutsatt att avsändaren har rätt IP-adress (200.1.2.3 ovan). Skillnaden mot strutformad NAPT ligger alltså i hur den externa IP-adressen hanteras.

Symmetrisk NAPT fungerar inte med IP-telefoni. Och flera större brandväggar, som används i stora organisationer, använder symmetrisk NAPT.

Den tredje typen kallas för symmetrisk. Adressöversättaren accepterar inkommande paket på port 2 435 och skickar dem vidare förutsatt att avsändaren har rätt IP-adress och port. Anrop till 2 435 men med fel avsändaradress eller fel avsändarport släpps inte igenom. Denna typ av NAPT är implementerad i vissa ADSL-enheter och vissa råbarkade brandväggar.

Symmetrisk NAPT fungerar inte med inkommande IP-telefoni, medan de andra två metoderna går att få att fungera.

För att få IP-telefoni att fungera genom en adressöversättare behövs flera funktioner. Klienten behöver:



S-port: 3456	D-port: 2435
S-IP: 200.1.2.3	D-IP: 195.6.7.8

Extern adress	Intern adress
195.6.7.8:2435	10.1.1.10:1423

Session

Lokal	Extern
195.6.7.8:2435	200.1.2.3: 80

Symmetrisk NAPT accepterar bara paket från en viss IP-adress och ett visst portnummer.

1. Ta reda på sin externa adress innan den registrerar sig i växeln
2. Ta reda på vilken typ av adressöversättning som används: NAT eller NAPT, typ av NAPT
3. Hålla sessionen uppe så den inte ”timas ut” av adressöversättaren.

Dessa tre funktioner hanteras av protokollet STUN (Simple Traversal of UDP through NATs) beskriven i RFC 3489. STUN har blivit populärt då det inte kräver ändringar av hur brandväggar eller protokoll fungerar.

```
Source         Destination    Protocol Info
192.168.3.245   62.80.200.56   STUN   Message: Binding Request
192.168.3.245   62.80.200.56   STUN   Message: Binding Request
192.168.3.245   62.80.200.56   STUN   Message: Binding Request
. . .
192.168.3.245   62.80.200.56   STUN   Message: Binding Request
192.168.3.245   62.80.200.56   STUN   Message: Binding Request
192.168.3.245   62.80.200.56   STUN   Message: Binding Request
192.168.3.245   62.80.200.56   STUN   Message: Binding Request
192.168.3.245   62.80.200.55   STUN   Message: Binding Request
62.80.200.55    192.168.3.245  STUN   Message: Binding Response
192.168.3.245   62.80.200.56   STUN   Message: Binding Request
62.80.200.56    192.168.3.245  STUN   Message: Binding Response
192.168.3.245   62.80.200.55   STUN   Message: Binding Request
192.168.3.245   62.80.200.56   STUN   Message: Binding Request
192.168.3.245   62.80.200.56   STUN   Message: Binding Request
. . .
192.168.3.245   62.80.200.56   STUN   Message: Binding Request
192.168.3.245   62.80.200.55   STUN   Message: Binding Request
192.168.3.245   62.80.200.55   STUN   Message: Binding Request
192.168.3.245   62.80.200.55   STUN   Message: Binding Request
62.80.200.55    192.168.3.245  STUN   Message: Binding Response
62.80.200.55    192.168.3.245  STUN   Message: Binding Response
```

Ovan visas en inspelning av hur STUN fungerar. Vissa återkommande ”binding requests” har utelämnats. En privat adress anropar en STUN-server, servrarnas adresser har konfigurerats på klienten. En STUN-server måste inte svara, enligt RFC 3489 ska klienten fortsätta att skicka förfrågningar i drygt 30 sekunder innan den uppfattar kontakten som bruten. Klienten bör också skicka förfrågningar regelbundet för att upptäcka förändringar.

För att förstå vilken typ av NAPT som klienten sitter bakom behövs information från två STUN-servers, vilket syns i andra halvan av listan ovan. (de slutar på 55 respektive 56). Om vi undersöker de två typerna av paket från samma inspelning hittar vi:

```
Paket nr. 262
Ethernet II, Src: 00:0b:5d:c3:2f:98, Dst: 00:0f:cb:9f:4a:fa
Internet Protocol, Src: 192.168.3.245 Dst: 62.80.200.55
User Datagram Protocol, Src Port: 65340, Dst Port: 3478
   Source port: 65340
   Destination port: 3478
```

```
    Length: 28
    Checksum: 0xfc10 [correct]
        [Good Checksum: True]
        [Bad Checksum: False]
Simple Traversal of UDP Through NAT
    Message Type: Binding Request (0x0001)
    Message Length: 0x0000
    Message Transaction ID: FC160CF17571424CB2437EE5B7D081EC
```

Paket nr 263

```
Ethernet II, Src: 00:0f:cb:9f:4a:fa, Dst: 00:0b:5d:c3:2f:98
Internet Protocol, Src:.80.200.55, Dst: 192.168.3.245
User Datagram Protocol, Src Port: 3478 , Dst Port: 65340
    Source port: 3478
    Destination port: 65340
    Length: 96
    Checksum: 0x25c3 [correct]
        [Good Checksum: True]
        [Bad Checksum: False]
Simple Traversal of UDP Through NAT
    Message Type: Binding Response (0x0101)
    Message Length: 0x0044
    Message Transaction ID: FC160CF17571424CB2437EE5B7D081EC
    Attributes
        Attribute: MAPPED-ADDRESS
            Attribute Type: MAPPED-ADDRESS (0x0001)
            Attribute Length: 8
            Protocol Family: IPv4 (0x0001)
            Port: 65340
            IP: 85.224.171.125
        Attribute: SOURCE-ADDRESS
            Attribute Type: SOURCE-ADDRESS (0x0004)
            Attribute Length: 8
            Protocol Family: IPv4 (0x0001)
            Port: 3478
            IP: 62.80.200.55
        Attribute: CHANGED-ADDRESS
            Attribute Type: CHANGED-ADDRESS (0x0005)
            Attribute Length: 8
            Protocol Family: IPv4 (0x0001)
            Port: 3479
            IP: 62.80.200.56
```

Det första paketet, nr 262, innehåller inte mycket mer information än att det just är en ”Binding Request”. Portnumret som används är 3478 för STUN-servern. En klient kan sätta flaggor i sin förfrågan och be servern att svara till exempel en annan port

Den IP-adress som kontaktade STUN-servern lyfts upp två nivåer och blir applikationsdata. Därför kan det nu skickas till klienten i form av ett ”Binding Response” under attributet ”MAPPED-ADDRESS. Vidare skickas information om vilken IP-adress och port STUN-servern har samt den IP-adress (CHANGED-ADDRESS) som kan användas om klienten vill testa mot en annan server för att bedöma vilken adressöversättningsteknik som används.

Referenser

RFC 1631	The IP Network Address Translator (NAT)
RFC 1918	Address Allocation for Private Internets
RFC 2663	IP Network Address Translator (NAT) Terminology and Considerations
RFC 2694	DNS extensions to Network Address Translators (DNS_ALG)
RFC 3022	Traditional IP Network Address Translator (Traditional NAT)
RFC 3027	Protocol Complications with the IP Network Address Translation”
RFC 3235	NAT Friendly Application Design lines
RFC 3489	STUN – Simple Traversal of User Datagram Protocol (UDP) Through Network Address Translators (NATs)