

Introduktion till Linux och små nätverk

Internet och dess utveckling

När man på 90-talet såg hur antalet datorer anslutna till internet ökade lavinartat, så var det rätt uppenbart att antalet lediga IPv4-nät och adresser minskade så snabbt så att de skulle ta slut rätt snart. För att lösa det problemet, samt några andra som man kommit på med IPv4, så skapades den nya versionen som fick namnet IPv6¹.

Det nya protokollet baserades till stora delar på IPv4, så att den kunskapen kunde utnyttjas med det nya protokollet IPv6. Man hade även sett att de routrar som finns på internet hade problem med att de behövde så mycket information om hur de olika näten satt ihop för att kunna skicka paketen rätt. Man noterade även att en ökad säkerhet relativt IPv4 var nödvändig, så att därför lade man till IPsec som en nödvändig del i IPv6. För lokala nätverk, NAT, noterades att det inte fanns något bra sätt att automatiskt ansluta och sedan nå internet att administratören måste förbereda detta. Det var några av problemen som löstes med IPv6.

Översikt av IPv6

För att lösa det första problemet, antalet adresser, så utökades antalet bitar i IPv6 från IPv4:s 32 bitar till IPv6:s 128 bitar. Det gör att varje kvadratmillimeter på jordens ytan skulle kunna få en egen IPv6-adress. Med detta ansåg man sig framtidssäkra internet.

Eftersom det finns så många adresser, så passade man även på att göra några förenklingar för att få effektiv routing för routrarna på internet. Som resultat av detta så är det starkt rekommenderat att man normalt använder en nätmask som är /64 för LAN. Dvs nod-delen i lokala nätverk i IPv6 har då 2^{64} nodadresser. Men det är för de enskilda LAN:en. Det är även designat så att varje kund skall ha fler än ett nät. Så de vanligaste uppdelningar för företag och privatpersoner från ISP:er är då /48 och /56.

Ur dessa nät så kan kunden sedan välja att dela upp sina egna nät som de vill, men att varje LAN skall då alltså vara /64 bitar. Detta gör att en kund som får /48, dvs 65532 nät och de med /56 får 256 nät att använda som de vill. Standarden för IPv6 säger explicit att man INTE skall dela upp med annat än dessa nätmasker till kunder.

Behöver man så många adresser? Det som är på gång nu är Internet of Things (IoT), som vi redan idag ser början av. Varje mobiltelefon är en IoT-enhet. Det kommer flera datorer, TV-apparater har nätverksanslutning, strömuttag kommer med nätverksanslutning, dammsugare, kylskåp, temperaturgivare och element kommer att bli nätverksanslutna. Så det kommer snabbt upp i många nätverksanslutna enheter.

För att ha någon kontroll på dem och förenkla brandväggsregler, så behöver de delas upp i flera nät. Dvs nät för temperaturstyrning, larm, köksutrustning, vanliga datorer, mobiltelefoner, bil/cykel, gästers enheter, personliga nätverk för de saker som man har på sig etc. Så då är det absolut nödvändigt med många IP-adresser.

IPv6 adresser

När IPv4 skrivs som fyra decimala tal åtskilda med en punkt så skrivs IPv6 ut med 8 st

1 Varför inte IPv5? Jo, det var redan reserverat för ett experimentellt protokoll som sedan inte utvecklades mer.

hexadecimala tal åtskilda med kolon. En IPv6-adress kan alltså se ut så här:

`2001:0db8:dead:beef:0012:0000:0000:0345/64` eller
`fe80:0000:0000:0000:a8a6:eea8:0e30:0d52/64`.

För att adresser inte skall bli så långa så kan de förkortas.

Inledande nollor kan raderas, så att adresserna ovan kan skrivas så här istället.

`2001:db8:dead:beef:12:0:0:345/64` eller
`fe80:0:0:0:a8a6:eea8:e30:d52/64`.

Sedan kan man även korta ner en grupp av ensamma nollor genom att ta bort dem och tillhörande `:`-tecken. Man kan bara göra det med en grupp nollor per adress. Varför kan man göra det? För att man vet att den totala längden skall vara 8 st grupper åtskilda med 7 st `:`-tecken. Så man kan ersätta en grupp på det viset. Skulle man göra det med två grupper, så vet man inte hur många grupper av nollor som skall skjutas in på respektive ställe. Så då kan adresserna ovan skrivas så här istället:

`2001:db8:dead:beef:12::345/64` eller `fe80::a8a6:eea8:e30:d52/64`.

Varje dator har normalt flera olika IPv6-adresser, nämligen `unicast` eller `anycast` samt `multicast`. IPv6 har inte någon `broadcast`-adress som ju finns i IPv4. Broadcast-adress går till alla datorer på ett lokalt nätverk, men det har alltså tagits bort ut IPv6 och använder istället mindre grupper av datorer med en gemensam `multicast`-adress istället. Mer om det senare.

`Unicast` och `anycast` används när två datorer skall kommunicera med varandra. `Unicast` är en dator på internet medans `anycast` bara är den semantiska skillnaden att flera datorer kan ha samma `unicast`-adress. Då går medelandet till den dator som är närmast. Exempel på `anycast` kan vara den DNS-servrar som hanterar top-domänerna. Dessa har bara en handfull IPv4 och IPv6-adresser, men det finns hundratals datorer som lyssnar och svarar på DNS-anrop till dessa adresser, men det är bara en i taget som svarar, dvs de är `anycast`-adresser.

`Multicast` är en grupp av datorer som delar på samma adress. Så när ett meddelande skickas till en `multicast`adress, så skickas det till alla datorer som lyssnar på just den adressen. Så en `multicast`adress kan man se som en adress för alla datorer med samma funktion. Så i ett LAN finns en `multicast`adress för alla IPv6-maskiner (`fe80::1`), för alla routers (`fe80::2`) etc.

`Broadcast`, som finns i IPv4 men inte i IPv6, innebär att man skickar till alla datorer i nätverket, inte en grupp. De funktionerna täcks in av `multicast` på ett bättre sätt.

Alla IPv6-adresser är sedan uppdelade i grupper beroende på om de är globala eller länk-lokala i det LAN som maskinen finns i. Kort kan man säga att hör adressen till nätet `2000::0/3` (inleds binärt med `0010B=2H` eller `0011B=3H`), så är den global och hör den till `fe80::0/10` så är adressen länk-lokal (bara användbar i LAN, den får inte skickas vidare av en router). Adresser i nätet `ff00::0/8` är olika former av `multicast`adresser. För mer information titta i referenserna.

Det finns några speciella adresser, som även finns motsvarande i IPv4. Några viktiga är `localhost`, vilket är adress `::1/128` (vilket i IPv4 motsvaras av adressen `127.0.0.1/8` i nätet `127.0.0.0/8`). Standardadress när ett program vill lyssna på alla nätverksenheter är `:::/0` (vilket i IPv4 motsvaras av `0.0.0.0`). Det kan inte användas i paket som skickas ut, utan används bara för program som skall öppna en adress för att lyssna. Exempelvis webbservrar eller filservrar.

Adresstilldelning

Som kund så får man normalt ett nät med /48 eller /56 som man sedan kan dela upp i /64-nät. Detta eftersom man vill ha flera nätverk (exempelvis ett för gäster, ett för köksmaskinerna, ett för sina personliga datorer i kläderna och ett för det vanliga nätverket). Detta eftersom varje LAN i IPv6 **skall** ha ett /64-nät (eller i *undantagsfall* kortare prefix). Det finns tillräckligt med nätadresser och nät i IPv6 för att det skall räcka för hela världen. Även om det bara är för en tunnel med två datorer, så är dessa tilldelade ett eget /64-nät.

Adresser till noder kan delas ut på liknande sätt som för IPv4, dvs statiska adresser och utdelade med DHCP-mekanism anpassad för IPv6, DHCPv6. Den statiska adressen sätts i installationsfilerna, `/etc/network/interfaces` för Debian. DHCPv6 är inte samma protokoll som DHCPv4, men de fungerar på liknande sätt. Så man behöver en DHCP-server som stödjer DHCPv6 för ett IPv6-nät.

Med IPv6 så finns en tredje metod som är unik för IPv6. Den innebär att noden frågar nätets routrar efter nätadress, nätmask, router och DNS-server. Det fungerar genom att routrar i nätet annonserar denna informationen med jämna mellanrum till alla datorer in i nätet. Det innebär då att datorn bara behöver anslutas till nätet för att få en fungerande IPv6-nätverk.

För att få en Debian-maskin att fungera som en sådan router, så kan man installera paketet `radvd`, som rätt inställt annonserar nätprefix och mask till nätverket som det är anslutet till. För att hantera DNS-inställningar på Linux-klienter så vill man även ha paketet `rdnssd` installerat. För att felsöka, så använder man lämpligen samma verktyg som för IPv4 men man har även tillgång till verktygen i paketet `ndisc6`. Det sista för att diagnosticera IPv6.

Man kan även notera att IPv6 är designat för att hantera många IPv6-adresser på en maskin, så det är normalt med två-tre länkllokala `fe80::/64`-adresser och två-tre globala `2000::/3`-adresser.

ICMPv6

Med IPv4 så behövdes ett protokoll som heter RARP för att koppla hop en nätverksanslutnings MAC-adress med de IP-adresser som använder den. Det är detta som man tittar på med kommandot `ip neighbour show`. I IPv6 så tog man och skrotade RARP och lade till funktionen i IPv6-versionen av ICMP istället.

Man lade även till en förenkling i IPv6 som gör att routrar inte får ändra storlek på paketen som den skickar. Så om en anslutning mellan två routrar inte klarar tillräckligt stora paket, så slänger routern paketet och skickar ett ICMPv6-meddelande tillbaka till första routern om att paketet var för stort. I IPv4 så kunde routern dela på paketet i två delar och skicka vidare dessa. Men det lade mer arbete på routern, så då har man skippat detta i IPv6 så att routrarna får mindre att göra.

På grund av tidiga felaktiga implementeringar av IPv4 under 90-talet så har många valt att helt filtrera bort ICMPv4-meddelanden i sina brandväggar. Det har gjort att exempelvis ICMP ECHO inte fungerar mot många nätverk och maskiner. Det går alltså inte att göra numera, eftersom så mycket viktig information finns i ICMPv6-meddelanden.

I övrigt fungerar ICMPv6 som ICMPv4 när det gäller dess funktionalitet.

Anslutning till Internet via IPv6

Alla moderna operativsystem har numera stöd för IPv6 direkt, så man behöver inte ställa in något

speciellt för att de skall fungera, om det nätverk som de kopplas till har IPv6-stöd. Om de inte har IPv6-stöd så kommer datorerna att använda IPv4 istället (om nätet har IPv4-stöd). Dvs man behöver aldrig stänga av IPv6.

Hur ansluter man då sitt nätverk eller sin maskin till IPv6? Antingen så har man otur och den ISP (Internet Service Provider, som exempelvis Telia eller Bredbandsbolaget) som man har inte tillhandahåller IPv6 eller så har man tur och så gör de det. Om de gör det, så är det bara att ställa in sin bredbandsdelare (router/brandvägg/NAT) så att den även delar ut IPv6-adresser till maskinerna i det lokala nätet. Har man fått en router från ISP:n, så är det troligen redan inställt.

Om man inte har sådan tur, så kan man skapa en tunnel så att man skickar all IPv6-trafik igenom den tunneln från sitt nät till internet via IPv4. En tunnel är som en nätverksanslutning direkt mellan två datorer, men den går via ett annat nätverk, som exempelvis IPv6 via IPv4-nätverk.

Det finns några företag som tillhandahåller tunnlar gratis. Ett sådant företag är Hurricane Electric, som tillhandahåller IPv6 via <http://tunnelbroker.net/>. Så med hjälp av dem så kan man låta sin Linux-maskin vara router för IPv6-nät till linux-maskinen och det LAN som den är ansluten till.

Transportskikten TCP och UDP i IPv6

Ett mål med IPv6 var att få det så lika som möjligt IPv4, så i princip är det enda som skiljer TCP och UDP i IPv6 och IPv4 adresserna. Övriga inställningar, som protokoll och portnummer, är med väldigt få undantag lika mellan IPv4 och IPv6.

Serverar (daemons)

En sak som kan vara bra att veta är att i URL:er så kan man inte skriva som man gjorde i IPv4, exempelvis <http://192.168.22.12/>, för att komma åt en server via IP-adressen. I IPv6 skriver man [http://\[2001:db8:dead:beef:12::345\]/](http://[2001:db8:dead:beef:12::345]/), dvs man **måste** omge IPv6-adressen med hakparenteser. Om man har stöd för IPv6 i DNS-servern så behöver man inte göra något annorlunda än för IPv4. Bara använda namnet som vanligt.

Detta gäller även när man konfigurerar serverar, så kan man ibland använda IPv4-adressen `0.0.0.0` för att tala om att servern skall lyssna på alla IPv4-nätverksadresser som servern har på alla interface. Motsvarande för IPv6 är `::` eller `0::0`.

Brandvägg

I Linux så använder man ju kommandot `iptables(8)` för att konfigurera IPv4-brandväggen. För att konfigurera en IPv6-brandvägg så använder man kommandot `ip6tables(8)`. Många program som ställer in brandväggen, som exempelvis `ufw(8)`, har numera bra stöd för IPv6, så det behöver man normalt inte bry sig om. Andra har delats upp i två versioner, en för IPv4 (exempelvis `shorewall(8)`) och för IPv6 (`shorewall6(8)`).

Notera att för de flesta program som man använder när man testar nätverk, som `ping`, `ip` och `netstat`, så måste man antingen använda en speciell version av kommandot, som `ping6`, eller så måste man lägga till en speciell växel, som exempelvis `-6` till kommandot `ip -6 addr`.

Så ställer man då in en brandvägg?

Använder man `ufw`, så kommer den att ställa in samma regler för IPv4 som för IPv6, så där behöver man inte göra något speciellt. Om man anger en IPv4-adress så kommer brandväggen naturligtvis bara att gälla för IPv4 och med en IPv6-adress bara för IPv6.

Använder man istället programmet `shorewall`, så finns det en speciell version för IPv6 som heter `shorewall6`. De har olika inställningsfiler, men de har motsvarande innehåll. Så vet man hur `shorewall` fungerar så är det lätt att få `shorewall6` att fungera.

Så ställer man då in sin router?

Om man har en router som klarar av att hantera IPv6, så behöver man inte ställa in så mycket. Men vanliga datorer skall normalt inte fungera som routrar, så de är normalt inställda att inte skicka vidare paket som inte skall till dem direkt. Därför måste man ställa in så att Linux skickar vidare paket, sk routning. Det gör man genom att ställa in i `/etc/sysctl.conf` så att Linux-inställningen sätts till ett, `net.ipv6.conf.all.forwarding=1`, dvs på motsvarande sätt som man gör för IPv4. Utan den inställningen så kommer paket som kommer till datorn och är adresserade till någon annan dator bara att slängas.

Notera att för mer avancerade nätverk, så behöver man speciella program som hjälper till att dynamiskt ställa in routnings-tabellerna eller så måste man konfigurera de olika routningarna i `/etc/network/interfaces`.

Tunnel?

Vad är en tunnel då? Det är en inställning som gör att man kan komma i kontakt mellan olika nät utan att datat går över internet i klartext. Exempel på tunnel är exempelvis VPN för att kryptera innehållet som skickas eller för att få IPv6 skickat till sitt lokala nät/dator.

För att konfigurera en tunnel till sin egna maskin så kan man ställa in dess parametrar i `/etc/network/interfaces` med exempelvis `up` respektive `down` direktiven. Där kan man skriva kommandon som skall exekveras när nätverket startar och stängs av.

Det finns många olika tunnlingssätt, som exempelvis *Teredo*. Men pga några problem med det protokollet, så bör man undvika det.

För att ställa in en IPv6 i IPv4-tunnel så skall man få brandväggen att släppa igenom protokollet 41 genom brandväggen, som heter *IPv6*. Notera att detta inte är TCP- eller UDP-port 41, utan det är *IP-protokollet* IPv6 som har värdet 41. TCP har värdet 6, UDP har värdet 17 och ICMP värdet 1.

För att öppna IPv4-brandväggen för protokollet så kan man lägga till följande i sin IPv4-router.

```
# iptables -I INPUT 1 -p 41 -s SERVER_IPV4_ENDPOINT -j ACCEPT
```

Annat alternativ är att definiera maskinen som är IPv6-router som *DMZ-host* i IPv4-routern. Då kommer all anslutning till routern från internet att skickas till den maskinen. Det som är lite tråkigt är att detta heter lite olika på olika routrar.

Mer om det finns i referenserna nedan.

Man kan även använda vissa program för att hantera en tunnel automatiskt, exempelvis med programmet `gogoc` (8). Då behöver man normalt inte justera IPv4-routers/brandväggens inställningar. Det man kan behöva är att skaffa sig ett konto och ändra i konfigurationsfilerna för

`gogoc i /etc/gogoc/gogoc.conf.`

Namnuppslagning

För att slippa använda IPv6-adresser (och IPv4-adresser), så kan man istället ge dem vanliga namn i nätverket.

Normalt så behöver man ställa in namnet och tillhörande IPv4 och IPv6-adress i `/etc/hosts`. Där finns det normalt ett IPv4-adress för datorns namn som är `127.0.1.1`. Detta bör bytas ut mot servermaskinens IPv4-adress i LAN:et om den har en statisk adress. Annars är det bäst att låta den vara.

När maskinen har en statisk IPv6-adress så kan man även lägga till den till maskinens namn i `/etc/hosts`.

Men för att andra maskiner än den själv skall kunna använda namnet, så kan man ibland lägga till det i dns-server/routers lista över namn-IP-nummer.

Vill man sedan att hela världen skall kunna hitta maskinen utan att använda IPv6-adressen, så måste man registrera ett DNS-namn samt i den registrera ett AAAA-fält som översätter från namn till IPv6-adress. För IPv4 så använder man ett A-fält istället. Dessa fält kan ibland uppdateras dynamiskt. Det kan då göras genom att skriva det kommando som behöver exekveras i ett up och down-direktiv i `/etc/network/interfaces`.

Summering

Vill man ha IPv6 bara till sin egna maskin, utan att skicka det vidare så behöver man inte sätta upp maskinen som router.

Vill man däremot dela IPv6 med flera maskiner i sitt nätverk, så måste man ställa in Linux-maskinen som en router för IPv6 och lämpligen sätta upp en korrekt brandvägg för IPv6 parallellt med IPv4. Det innebär att man även behöver konfigurera brandväggen i Linux-maskinen så att regeln FORWARD bara släpper igenom data som man är intresserad av. Eftersom NAT inte finns till IPv6, så behöver man i vart fall inte hantera det, vilket gör det lite lättare att konfigurera ett IPv6-nätverk.

Referenser

- <http://sv.wikipedia.org/wiki/Internetprotokoll>
- <http://sv.wikipedia.org/wiki/OSI-modellen>
- <http://sv.wikipedia.org/wiki/IPv6>
- <http://tunnelbroker.net/>
- <http://en.wikipedia.org/wiki/Netfilter>
- <http://en.wikipedia.org/wiki/Iptables>
- <http://en.wikipedia.org/wiki/Shorewall>
- http://en.wikipedia.org/wiki/Uncomplicated_Firewall
- <http://wiki.debian.org/DebianIPv6>
- <http://planetfoo.org/blog/archive/2012/04/23/using-an-ipv6-tunnel-broker-with-linux/>
- <http://www.idg.se/2.1085/1.387839/sa-enkelt-skapar-du-en-ipv6-tunnel>
- <http://www.networkworld.com/news/2010/050610-ipv6-tunnel-basics.html>
- <http://dnslookup.me/free-tunnel/free-ipv6-tunnels/>
- <http://www.tldp.org/HOWTO/Linux+IPv6-HOWTO/chapter-configuring-ipv6-in-ipv4-tunnels.html>
- <http://www.sevalecan.com/site/debianipv6>
- <http://madduck.net/docs/ipv6/>
- <http://www.chronos-tachyon.net/reference/debian-ipv6-and-hurricane-electric>
- <http://blog.endpoint.com/2012/03/ipv6-tunnels-with-debianubuntu-behind.html>
- <http://www.tunnelbroker.net/forums/index.php?topic=2589.0>

(Lite skoj)

- <http://xkcd.com/742/>
- <https://xkcd.com/865/>
- <http://xkcd.com/195/>
- <http://xkcdsw.com/3328>