

IP-adresser.

En introduksjon for elever ved Oslo By Steinerskole.

Dette dokumentet forutsetter at du allerede har lest ”Protokoller, en introduksjon for elever ved Oslo By Steinerskole” og ”Tallsystemer, en introduksjon for elever ved Oslo By Steinerskole”, samt kompendiet ”Logiske operasjoner på binære tall”.

Introduksjon:

Nå i år 2002 er Internet et helt annet nettverk enn det som var da det ble etablert tidlig på 80-tallet. I dag er Internet det største datanettverket i verden og det doubler sin egen størrelse ca. hver 9.de måned. Man ser stadig flere og flere websider og det er vel ikke mange firmaer, med respekt for seg selv, igjen som ikke har en webside. Vi hører stadig oftere eldre mennesker som klager og sier ”trur’em at alle har Internet, eller...” Vel, ikke alle har Internet, men det blir mer og mer vanlig. Norge ligger nesten i verdenstoppen (helt på topp, i skrivende stund, ligger Syd Korea ☺) og det bare øker. Yngre mennesker i dag er ofte oppvokst med Internet og World Wide Web og tar det som en selvfølge at det er der. Husk at WWW ikke ble oppfunnet før i 1993. Det er ikke mer enn ca. 10 år siden. Vi kan nok trygt si at Internet bare så vidt har begynt. Det kommer til å øke dramatisk de neste årene, det er i alle fall jeg 100 % sikker på.

Det er ikke bare den private bruken av Internet som kommer til å øke. Etter hvert som nettet øker vil det dukke opp flere nye tjenester og noen vil forsvinne. Stadig flere firmaer vil ikke bare ha en hjemmeside, de vil ha direkte handel med kunder som ikke kan komme til butikken over nettet. I større kjeder er det vanlig at kassaapparatene er koblet til nettet, noe som gir ledelsen et meget sterkt verktøy for å holde kontroll over omsetningen. Kombinert med kamera er det nesten en total innsikt.

Vi kan også tenke oss borettslag og småbåthavner, eller parkeringsplasser hvor det settes opp et kamera, bildene sendes hele tiden til en webside og dersom noen ser på websiden og observerer en ”skummel person” kan man ringe politi eller vaktseksjon. (For ikke å snakke om brann og brannvesen!).

Det er en direkte sammenheng mellom verdien av Internet og antall maskiner som kobler seg på. Jo flere det er, jo bedre ressurser kan du finne og jo flere kunder kan et firma nå frem til.

Skaleringsproblemet:

Det er to enkeltfaktorer som begrenser veksten av selve Internet. Det er:

- Det er ikke nok IP adresser (i IP versjon 4).
- Rutingen (evnen til å sende datapakke fra et nettverk til et annet nettverk) mellom de forskjellige nettverkene som er koblet sammen blir vanskelig fordi de er så mange nettverk og rutingtabellene blir kjempestore.

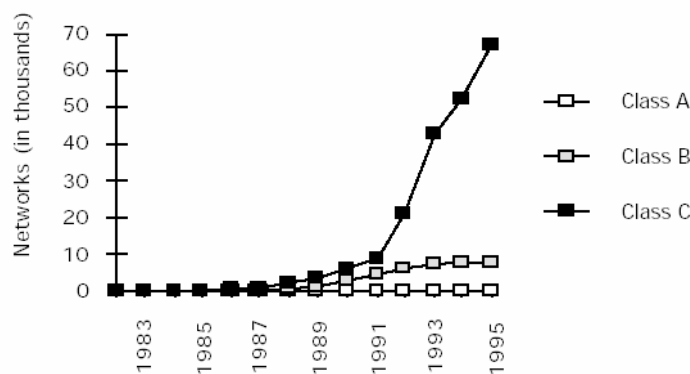
Det første problemet her er at vi holder på å slippe opp for adresser. Internet (gjelder også andre nettverk) er slik at alle maskiner som kobles til Internet MÅ ha en egen, unik IP adresse. Ingen kan ha en adresse som er i bruk på en annen maskin. IP protokollen i versjon 4 (det er den vi bruker i dag) er en 32 bits adresse. Det betyr at vi maksimalt kan ha 2^{32} adresser, eller 4 294 967 296 maskiner koblet til Internet samtidig. Fire milliarder tohundreogtjue millioner ni hundreogsekstisytusen tohundreogtjue maskiner, og så holder vi på å slippe opp? Ja, vi gjør det. Husk at alle maskiner også rutere og (mange) switcher og lignende skal ha IP adresser også. Det er ingen helt umiddelbar fare for at vi går tom for adresser, men det vil komme dit i løpet av noen få år. Det finnes metoder for å omgå denne problemstillingen og vi skal se nærmere på dette litt senere.

En av grunnene til at vi allerede har begynt å få problemer med mangel på adresser er at IP adressene ikke utnyttes på beste måte bestandig. Dette skal vi også se nærmere på.

Det er mye bekymring å spore hos ekspertene om dette. Litt av problemet (ganske mye forresten) ligger i det begrepet som kalles adresseklasser, eller klassemessig adressering. Dette skal vi naturligvis se nøye på også.

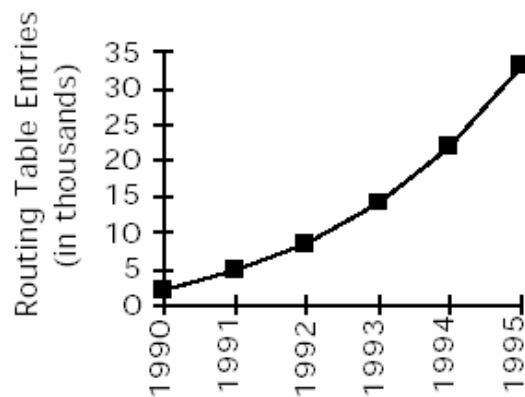
Dersom problemene med IP adresser ikke løses innen få år vil det kunne oppstå en situasjon hvor nye abonnenter ikke får IP adresse og dermed ikke får koblet seg inn på Internet.

Det er dessverre fullt...



Økning i antall registrerte nettverk i klasse A, B og C (1983-1995)

Det andre store problemet som har dukket opp ligger på et annet nivå. Det er veksten i rutingtabellene på Internet. De store ruterne i Internets backbone, de ruterne som for eksempel Telenor benytter der trafikken skal rutes over Atlanterhavet og til andre gigantiske noder som samler og videresender datapakker fra kontinent til kontinent, disse ruterne må ha oppdaterte rutingtabeller for HELE Internet! De senere årene har antall rutere gått rett til vær. I desember 1990 var det 2100 rutere, i 1995 var det 30000. Hvor mange det er i dag vet jeg faktisk ikke, men det er garantert flere hundre tusen.



Vekst i Internets rutingtabeller (1990-1995)

Tilsvarende ser vi at veksten i rutingtabellene fra 1990 til 95 bare går en vei, nesten rett til værs. Dessverre så er det ikke slik at dette kan løses bare ved å sette inn flere rutere eller øke RAM og prosessorkraft i ruterne. Det er andre faktorer som spiller inn, først og fremst det at Internet er blitt veldig dynamisk. Det endres hele tiden, nye maskiner og nettverk kommer til, andre forsvinner. Dette krever enormt mye computerkraft og enormt mye menneskelig innsats. Dersom rutingtabellene får lov til å bare vokse fritt, vil ruterne måtte droppe deler av tabellene og enkelte deler av Internet vil plutselig ikke være tilgjengelig.

Det finnes imidlertid en løsning på dette problemet som er effektiv og langsiktig. Det er IPng Internet Protocol Next Generation, eller IP v6. Mens vi venter på at IP v6 kommer og overtar for IP v4 (som er den vi har i dag) vil det komme til å være behov for massevis av lapping, flikking og smarte omgåelser for å få Internet til å holde seg oppe og tilby de tjenestene (og flere fremtidige) som vi er vant med å ha. All denne lappingen og flikkingen vil være ”smertefull” og kan komme til å endre på fundamentale konsepter i Internet.

NB! Det er en forskjell på hvordan man skriver tall i USA og i Norge. Her hjemme bruker vi ”,” (komma) som desimalmerke. Det vil si det tegnet vi bruker for å indikere at det ikke lenger er hele tall, men deler av tall vi angir. For eksempel vil vi på norsk skrive den greske bokstaven π ”PI” 3,14, mens man i USA ville skrive 3.14, altså med punktum i stedet for komma. Dette er det lurt å være observant i forhold til, hvis du leser engelskspråklig litteratur. Dessverre er ikke all litteratur oversatt til norsk, og spesielt innen datalitteratur er det lite som er oversatt. Når du støter på uttrykket ”dotted decimal notation” betyr det at det brukes (amerikansk) desimalmerke mellom grupper av tall. Altså grupper av tall skilles ad ved bruk av punktum. For eksempel:

193.45.211.134

Legg merke til at det er ikke punktum verken før, eller etter tallrekken. Dersom det er et punktum til slutt betyr det bare at setningen er slutt, det har ingenting med tallrekken å gjøre.

Når det gjelder IP adresser er det slik at desimalmerket (det amerikanske, altså punktum og ikke komma) denne gangen bare brukes som et tegn, ikke for å angi at det faktisk er snakk om fraksjoner, eller desimaler, om du vil. Dette betyr at en adresse som for eksempel 193.45.211.134 IKKE er et tall med desimaler (og eventuelle flerleddede underdesimaler, hvis det finnes noe slikt i det hele tatt da...)

En IP(Internet Protocol) adresse er et ”tall” som er unikt (det vil si ingen andre kan ha det samme ”tallet”) på Internet. Dette tallet identifiserer maskinen på Internet. Det er ikke bare datamaskiner som har IP-adresser på Internett, det gjelder også en lang rekke andre komponenter som hjelper til å få trafikken til å gå, for eksempel rutere. Vi skal komme tilbake til dem, men kort sagt er de en slags trafikkonstabler som sørger for å videresende ditt signal til riktig maskin på Internett. Et annet eksempel kan være en nettverksprinter eller et avansert Web-kamera.

En IP (Internet Protokoll) adresse er en unik identifikator for en ”node” eller en ”host” på Internett som det går an å få kontakt med. (Forutsetter at maskinen ikke er skrudd av). En IP adresse er et 32 bits binært tall som vanligvis skrives som fire desimale tall, adskilt med et punktum. For eksempel: 148.122.232.223. Hver av disse fire tallgruppene, som kalles oktetter, representerer et åtte bits binært tall, og dermed er laveste mulige verdi 0, og høyeste mulige verdi 255. Dette gir $2^8=256$ mulige verdier, husk at 0 er en verdi. (Se ”Tallsystemer, en introduksjon for elever ved Oslo By Steinerskole”).

Det er lurt å forsøke å se dette i binær form, mange finner plutselig ut at dette er ganske innlysende da. Andre klør seg litt i hodet, men så kommer det ofte litt etter litt allikevel.

Eksemplet fra i sted: 148.122.232.223 blir, hvis vi oversetter det til binære tall:

148.	122.	232.	223
10010100.	1111010.	11101000.	11011111

Alle IP adresser består av to deler. En del som angir hvilket nettverk du tilhører, og en som angir ”noden” (En node er for eksempel en PC, en ruter eller en nettverksprinter). IP adresser er inndelt i klasser. Det finnes fem klasser. Hvilken klasse, og hvilken subnetmaske som er brukt avgjør hvilket nettverk noden tilhører. Denne kombinasjonen av IP adresse og subnetmaske avgjør OGSÅ hvilken del, eller hvor stor del, av adressen som tilhører nettverket og hvilken, eller hvor stor del av adressen som tilhører noden.

Adresseklasser;

(binære tall angis heretter som xb (for eksempel 0111b=7), hexadesimale tall angis som xh (for eksempel FFh = 255) og desimale tall har ingen angivelse (for eksempel 32 = 32).

- ❑ Klasse A er 00000001b- 01111110b, eller 1 – 126. (for eksempel 100.200.201.201 = (alle=b)1100100. 11001000. 11001001. 11001001
- ❑ Klasse B er 10000000b- 10111111b, eller 128-191. for eksempel 182.xxx.xxx.xxx = 11000001b.xxxxxxxxxb.xxxxxxxxxb.xxxxxxxxxb
- ❑ Klasse C er 11000000b – 11011111b eller 192 – 223.
- ❑ Klasse D er 11100000b – 11101111b eller 224 – 239
- ❑ Klasse E er 11110000b – 11111110b eller 240 – 254

Legg merke til at det blir stadig flere ”ledende ett-tall”, altså at adressene (i binær form) begynner med stadig flere ett-tall.

- ❑ Klasse A begynner med 0
- ❑ Klasse B begynner med 10
- ❑ Klasse C begynner med 110
- ❑ Klasse D begynner med 1110
- ❑ Klasse E begynner med 11110

Adresser som begynner med 01111111b, eller 127 er reservert. Disse brukes for å teste IP på din egen maskin og kalles en ”loopback adresse” fordi det signalet du sender, ”loopes”, går i sirkel tilbake til deg selv. Dette kan du teste selv, ved å ”pinge” 127.0.0.1. Det er din egen, interne IP adresse på din egen maskin. Det er mange programmer som bruker denne IP adressen for å få til en samtale med din PC. For å gjøre dette i praksis trenger du et DOS-vindu, eller en ”kommandolinje”. Gå til start/programmer/tilbehør/kommandolinje og skriv inn:

```
C:\>ping 127.0.0.1
```

Du får opp følgende:

```
Pinging 127.0.0.1 with 32 bytes of data:
```

```
Reply from 127.0.0.1: bytes=32 time<10ms TTL=128  
Reply from 127.0.0.1: bytes=32 time<10ms TTL=128  
Reply from 127.0.0.1: bytes=32 time<10ms TTL=128  
Reply from 127.0.0.1: bytes=32 time<10ms TTL=128
```

Ping statistics for 127.0.0.1:

Packets: Sent = 4, Received = 4, Lost = 0 (0% loss)

Approximate round trip times in milli-seconds:

Minimum = 0ms, Maximum = 0ms, Average = 0ms

Dette betyr at din maskin svarer deg, du får "reply" og alt er OK.

Akkurat som adresser som begynner med 127 er reservert er også alle adresser i D og E klassene reservert. Vi kan ikke bruke disse på Internett, og vi bør ikke bruke dem hjemme heller!! Dette er faktisk litt viktig, og vi skal komme tilbake til hvorfor. I mellomtiden sier vi bare at klasse D er reservert for multicasting og klasse E er reservert for eksperimentelt og for fremtidig bruk. Det er, i tillegg til disse, tre andre adresser (nettverk) som er reservert. Det er adresser som begynner med 10.0.0.0, 172.16.0.0 og 192.168.0.0. Disse er til privat bruk og kan ikke benyttes på Internett. Alle forespørsler fra en maskin som har en av disse adressene vil bli avvist på Internet. Mer informasjon om dette finner du ved å søke etter "RFC 1918" på Internett.

RFC står for "Request For Comments" og det er disse RFC som er "fasit" for hva som kan og hva som ikke kan gjøres innenfor TCP/IP.

RFCer er ofte skrevet på en veldig teknisk måte, noen av dem er utrolig detaljerte og kan derfor være vanskelige å forstå. Det er likevel sterkt å anbefale at man leser i alle fall de mest sentrale. For eksempel de som setter standarden for e-post, NetBIOS, TCP, UDP, IP samt flere interessante.

Klasse A nettverk (/8 prefiks):

Hvert eneste klasse A nettverk har et nettverksprefiks som er 8 bit lang og den biten som er MSD (most significant bit, som står lengst til venstre) er alltid satt til 0 og derefter kommer et nettverksnummer som er 7 bit langt. Dette følges av en hostnummer-del som er 24 bit lang. Hostnummer er det samme som nodeadresse.

I våre dager regnes det ikke lenger som moderne å snakke om et klasse A nettverk, man kaller det et "/8" nettverk. Dette uttales "slash-åtte-nettverk" fordi det er 8 bits i nettverksprefikset. (Se seksjonen klasseløse nettverk lenger ned i dokumentet).

Klasse A er en spesiell klasse, det er plass til enormt mange noder i et LAN, men det er ikke så mange nettverk i klassen. Det største antallet vi kan ha er $2^7 - 2 = 126$ nettverk. Vi trekker fra 2 fordi det er ulovlig med et nettverksprefiks som består av bare 0 eller bare 1. Disse er reservert for spesielle formål. Vi kunne altså hatt $2^7 = 128$ minus 0.0.0.0 nettverket og sittet igjen med 127 nettverk. Imidlertid er det nettverket som heter 127.0.0.0 reservert for noe som kalles loopback. Det er et "falskt" nettverk og brukes til å teste TCP/IP adresse konfigurasjonen på maskiner, pluss en del andre formål, men det kan ikke brukes som adresser på Internet. Vi har derfor 126 nettverk i klasse A, eller vi kan si at det finnes 126 stk. /8 nettverk.

Hvert av nettverkene i klasse A har plass til 2^{24} hostadresser (vi har 24 bit i hostnummeret) og det vil si 16 777 214 maskiner på et og samme nettverk (LAN).

/8 nettverk har til sammen (vi legger sammen all 126 nettverk) plass til $2^{31} = 2147483648$ individuelle hostadresser. IP adresse v4 er 32 bit lang 2^{32} og det er 4294967296 adresser. Vi ser dermed at /8 nettverkene, eller klasse A nettverkene har halvparten av alle tilgjengelige IP adresser i hele verden.

Klasse B nettverk (/16 prefiks):

Hver klasse B adresse har en 16 bit nettverkprefiks hvor de to bitsene som er lengst til venstre er satt til 1-0 og dermed er det 14 bits igjen til nettverksnumre og 16 bits igjen til hostnumre. Klasse B nettverk kalles /16 nettverk fordi det er 16 bits i nettverksadressen (de er der selv om de to første er fastsatt på forhånd). Det finnes 2^{14} klasse B nettverk (16384 stk). Det er mulig å få så mange som $2^{16}-2$ hostadresser. Altså 65 534 stk. Hvis ser etter vil vi finne ut at en klasse B adresse (/16) totalt har 30 bits å benytte seg av (det er 32, men de første to er fastsatt) er det plass til 2^{30} mulige adresser. Det er 1 073 741 824 adresser og det utgjør 25 % av alle tilgjengelige adresser i hele verden.

Klasse C nettverk (/24 prefiks):

Hver klasse C adresse har en 24 bits nettverkprefiks hvor de tre MSD er satt til henholdsvis 1-1-0, og en 21 bit stor del til nettverksnummeret ($24-3=21$). Hostnummerdelen er 8 bits lang. Vi kaller klasse C nettverk for /24 nettverk fordi den totale nettverkprefiksen er 24 bits lang. Vi ser nå raskt, for nå har vi fått litt trening i dette, at $24-3=21$ og det er derfor 2^{21} mulige nettverk, altså 2 097 152 forskjellige LAN med plass til 2^8-2 hoster. Det blir 254 maskiner maksimalt på et nettverkssegment. Ettersom hele /24 blokken har en teoretisk mulighet for $2^{21}= 536\ 870\ 912$ adresser og det utgjør 12.5 % av alle IP adresse adresser i hele verden.

Hallo?!?:

Hva er dette? Klasse A har 50 %, klasse B har 25 % og klasse C har 12,5 % av alle adressene. Hvis vi legger sammen det får vi jo bare 87,5 %. Hvor er resten av adressene? Hvorfor kan vi ikke bare ta dem i bruk hvis det er mangel på adresser? Det er jo 12,5 % igjen og det er 1/8 av adressene!

Jo da, de er der. Det finnes en klasse D og en klasse E også.

Klasse D har de første fire bitsene satt til 1-1-1-0 og brukes til multicasting. (se RFC 1112). Klasse E har de fire første bitsene satt til 1-1-1-1 og er reservert for eksperimentelt bruk.

Hvis du forsøker å bruke disse to klassene hjemme kan du få en del uventede komplikasjoner og de vil ikke bli rutet på Internet uansett.

Vi skal se på hvordan klassene avgjør hvor stor del av IP adressen som tilhører nettverket (N) og hvor stor del som tilhører noden (n).

- Klasse A -- NNNNNNNN.nnnnnnnn.nnnnnnnn.nnnnnnnn
- Klasse B -- NNNNNNNN.NNNNNNNN.nnnnnnnn.nnnnnnnn
- Klasse C -- NNNNNNNN.NNNNNNNN.NNNNNNNN.nnnnnnnn
- Klasse D og E dropper vi nå, de er allikevel ulovlige.

I eksempeladressen vi brukte tidligere; 148.122.232.223 er en klasse B adresse (148 er i gruppen 128 – 191), og nettverksadressen utgjøres av 148.122.y.z og nodeadressen er w.x.232.223. For å spesifisere hva som er nettverksadressen er det vanlig å sette nodeadressen til bare null. I vårt eksempel vil altså 148.122.0.0 være nettverksadressen til 148.122.232.223.

Nettverksadressen kan aldri brukes på en node, det vil si det er ikke mulig å ha en maskin på Internett med adressen 148.122.0.0, det er en nettverksadresse! En node kan altså ikke ha en adresse som har 0 i nodeadressen. Det er heller ikke tillatt å ha en adresse hvor det binære tallet i nodeadressen er bare 1-tall. Altså: 0 og 255 (11111111b=255) kan ikke brukes. Ellers er det greit, og vi ser da at det er plass til 254 mulige adresser ved å forandre på den siste oktetten.

($2^8 = 256$ minus 0 og minus 255 = 254 muligheter.)

Hvis den delen av IP adressen som tilhører noden består av bare 1 tall, vil dette være adresserangens såkalte broadcast-adresse. En adresserange er for eksempel 193.122.14.0. Den har 254 adresser fra 193.122.14.1 til 193.122.14.254. Det betyr at et signal fra en maskin til den maskinen som heter 193.122.14.123 bare blir tatt imot av 193.122.14.123. endes det derimot en melding til broadcastadressen 193.122.14.255 vil denne meldingen bli hørt og lest av alle maskiner fra 193.122.14.1 til 193.122.14.254, så sant de er skrudd på da... ☺.

Subnetmasker:

Alle noder på et nettverk som bruker TCP/IP må ha en subnetmaske. Alle må ha det! Subnetmasken brukes for å avgjøre om en datapakke skal til en node på det lokale nettverket (LAN), eller om pakken skal til en maskin på et annet nettverk.

Subnetmasken er, som IP adressen, en 32 bits binær verdi. Den angis, akkurat som IP adressen som fire desimale tall med et punktum imellom. Et eksempel på en typisk subnetmaske kan være: 255.255.255.0. Dette er den såkalte default subnetmasken for en klasse C adresse. En "default subnet mask" benyttes dersom nettverket IKKE er delt opp i subnett, dersom det er delt opp bruker vi en "custom subnet mask".

Alle nettverk må altså ha en subnetmaske. Dersom nettverket ikke er delt opp i subnett bruker vi default subnetmasken, men hva den er varierer med hvilken klasse adressen tilhører.

Subnetmasken brukes for å "skjule" eller "maskere" en del av IP adressen. Den delen som maskeres er den delen vi kaller nettverksadressen.

I subnetmasken er alle de bits (binær form) som tilsvarer nettverksadressen satt til 1, alle som tilhører nodeadressen er satt til 0. Hvis vi ser på en klasse C adresse med default subnetmaske kunne vi ha:

IP adresse	193.157.235.6
subnetmaske	255.255.255.0

Hvor 193.157.235 er nettverksadressen og 6 er nodeadressen.

Hvis vi nå setter opp disse tallene i binær form vil du straks se noe smart...

11000001.10011101.11101011.00000110	(IP adressen)
11111111.11111111.11111111.00000000	(subnetmaske)

Hvis vi ser på default subnet mask for klasse A, B, og C binært får vi:

Kl. A	11111111.00000000.00000000.00000000	desimal: 255.0.0.0
Kl. B	11111111.11111111.00000000.00000000	desimal: 255.255.0.0
Kl. C	11111111.11111111.11111111.00000000	desimal: 255.255.255.0

Hvis vi ser på en klasse B adresse kan vi sette den opp slik som dette:

IP adresse	131.107.16.200
Subnetmaske	255.255.0.0
Nettverksadresse	131.107.y.z
Nodeadresse	w.x.16.200

Hemmeligheten:

Det store problemet for IP ligger i å bestemme om en datapakke skal til en lokal node (på et LAN) eller om den skal til en node på et annet nettverk, altså om den skal fraktes over et WAN. Hvis den skal lokalt er det bare å sende den ut på nettverket vårt, men hvis den ikke skal til en lokal adresse må den jo sendes til ruterer, eller gatewayen om du vil. Det er ”porten” ut fra vårt lokale nettverk. Husk det lokale nettverket er der hvor alle sitter og lytter på mediet og hvis det kommer svevende en datapakke forbi som har riktig adresse, hentes innholdet opp fra nettverket og inn til selve datamaskinen. Slik er det ikke på Internet, der er det rutere som bestemmer hvor datapakker skal sendes, slik at de kan komme frem til det nettverket hvor den noden som skal ha dataene befinner seg. Når datapakkene kommer frem til riktig nettverk vil gatewayen (ruterer) der sende dem ut på det lokale nettverket. Der vil de vanligvis få hjelp av en switch til å finne raskt og greit frem til destinasjonsnoden.

For å avgjøre om en datapakke skal lokalt eller til et annet nettverk bruker TCP/IP en logisk operasjon på to binære tall. To ganger. Derefter sammenligner den resultatet av de to operasjonene.

Hvis de er like, skal pakken lokalt. Hvis de er ulike skal pakken til et annet nettverk og den går til ruterer / gatewayen og sendes ut på Internet.

Når TCP/IP starter opp, og det gjør den (protokollstakken) når maskinen starter, er noe av det første TCP/IP gjør å foreta en AND operasjon på sin egen IP adresse. Resultatet av operasjonen blir lagret i RAM.

Når IP skal avgjøre om en IP pakke skal til en lokal node eller en node på et annet nettverk blir den nye destinasjonsadressen ANDet med den samme subnetmasken som i sted. Hvis resultatene er like, skal pakken lokalt, er de forskjellige skal den til et annet nettverk. Da går den til gatewayens IP adresse for videre behandling.

AND operasjonene foregår internt i TCP/IP og er vanligvis ikke noe du trenger å utføre manuelt. For å være ærlig trenger du aldri å gjøre det, men du trenger å forstå hvordan dette foregår. Derfor skal vi se på noen eksempler.

For å ANDe IP adressen og subnetadressen sammenlignet IP de binære tallene etter et bestemt mønster. Hvis du er usikker på de logiske operasjonene anbefales det at du går tilbake og leser kompendiet ”Logiske operasjoner på binære tall” en gang til.

Helt kort; slik er mønsteret:

Bitkombinasjon	Resultat
1 AND 1	1
1 AND 0	0
0 AND 0	0
0 AND 1	0

Altså: 1 AND 1 er 1, alle andre kombinasjoner blir 0.

Vi tar et eksempel: (159.224.7.129 med default subnetmaske)

IP adresse:	10011111	11100000	00000111	10000001
Subnetmaske:	11111111	11111111	00000000	00000000
Resultat:	10011111	11100000	00000000	00000000

Øvelse:

Forsøk å ANDe avsenderadressen med nettverkets subnetmaske først, derefter ANDer du destinasjonsadressen med den samme subnetmasken.

Avsenderadresse:	10011001	10101010	00100101	10100011
Subnetmaske:	11111111	11111111	00000000	00000000
Resultat:				

Destinasjonsadresse:	11011001	10101010	10101100	11101001
Subnetmaske:	11111111	11111111	00000000	00000000
Resultat:				

- Er resultatene like?
- Er destinasjonsadressen på det lokale nettverket eller på et annet nettverk?
- Hvorfor?

Vi skal se mer på subnetmasker og hvordan vi beregner nettverk senere.

Shout or route?

Dette er et meget viktig spørsmål for en datamaskin som vil sende en datapakke over et nettverk til en annen maskin. Shout (å rope) betyr at pakken sendes ut på det lokale nettverket med mottagerens IP adresse (eller adresse fra en annen protokoll) i datapakkens header. Pakken sendes da ut på nettverket og (vi forutsetter Ethernet) ”flyte” rundt til den kommer til riktig mottagermaskin. Der blir innholdet kopiert og lastet opp og inn i mottagermaskinen. Dette gjør maskinen dersom den logiske AND operasjonen gav 100 % likt resultat når vi ANDet avsenderadresse/subnetmaske og mottageradresse/subnetmaske.

Hva skjer dersom den logiske AND operasjonen gav ulikt resultat når vi ANDet avsenderadresse/subnetmaske og mottageradresse/subnetmaske? Da vet vi jo at mottageren ikke er på det lokale nettverket, men befinner seg et eller annet sted på Internet, eller et annet privat nettverk. Hva gjør vi da?

Løsningen:

Istedenfor å sende pakken ut på nettverket, hvor den vil vandre rundt til den ikke får lov til å vandre rundt lenger (bestemmes av TTL faktoren), så pakker vi hele pakken inn i en ny IP pakke og sender den til IP adressen til ruterens vår! Den vil umiddelbart skrelle av den ytterste IP pakken og sjekke sine egne rutingtabeller for å se hvor den skal ekspedere pakken videre. Derefter vandrer pakken videre ut på Internet, fra ruter til ruter, inntil den kommer frem til det nettverket og den noden den egentlig skal til. Det er ikke slik at disse pakkene ”flyter fritt” på Internet, de blir sendt til bestemmelsesstedet ved bruk av et par protokoller (RIP [Routing Information Protocol] og OSPF [Open Shortest Path First]).

Adressering med IP versjon 6.0

Det vi bruker på Internet nå er IP versjon 4. Den er, som vi nettopp har sett, basert på 32 bits adresser. Det gir en mulighet for 2^{32} IP adresser, eller 4294967296. Det er ikke så veldig mange, og vi holder på å slippe opp for adresser. Det er ingen helt umiddelbar fare, men om få år vil dette være et problem av dimensjoner. Den nåværende IP protokollstakken er ikke blitt endret eller oppgradert siden 70-tallet. Det sier jo litt om hvor genialt systemet er. Allikevel var det ingen som forutså den enorme veksten Internet har fått og hvilket behov for adresser det er blitt. Det er jo ikke bare arbeidsstasjoner som trenger IP adresser, det kan være alt fra rutere, Web-kamera eller et kjøleskap (!) som er koblet opp til Internet. Internet er fremdeles bare i en sped begynnelse.

Dette problemet med mangel på adresser er grunnen til at IP versjon 6 ble lavet. Den kalles også IPng (IP next generation).

Der IP v.4 bruker 4 oktetter, bruker IP v.6 16 oktetter. Når vi skriver IP6 adresser skriver vi dem som 8 oktettpar separert med : [kolon] oktettparene er angitt heksadesimalt.

Eksempel på IP v.6 adresse:

AECC : 3E40 : B3FF : 7301 : AE52 : 56C4 : BA21 : AAF8

IP versjon 6 har altså 128 bits adresser, det vil si 2^{128} , eller mer enn $3 * 10^{38}$. Det vil si et 3-tall med 38 nuller bak. Det er sannsynligvis nok for en lang stund. Du kan jo forsøke å regne ut 2^{128} på kalkulatoren din?

Både adressestrukturen og strukturen til datapakken er annerledes i IP6 enn i IP4 og de to systemene er IKKE kompatible. Det er mye som gjør det vanskelig å sette i gang og bruke IP6 helt uten videre. Eftersom IP4 og IP6 ikke er kompatible betyr det at alt fra nettverkskort og lag 3 switcher til trancievere og rutere må byttes ut med nytt utstyr som støtter IP6. Dette er selvsagt dyrt, men det er også et problem at det foreløpig ikke er så mange fabrikker som produserer utstyr slik at dersom vi skulle bestemme oss for å bytte i morgen, ville det ikke være et nettverkskort å oppdrive...

IP6 er også annerledes i strukturen i selve pakken. Adressefeltet er større, men selve headeren er enklere, men har mulighet for å gjøre meget mer enn headeren i IP4. Det er lagt inn støtte for fremtidige egenskaper, det hele er altså ”foroverkompatibelt”.

I den grad man opplever IP6 kompatible servere eller rutere på Internet i dag så er de enten hybride, det vil si at de støtter både IP4 og IP6, eller at IP6 pakkene kamufleres inne i IP4 pakker og sendes over Internet som IP4 pakker, blir skrelt når de kommer frem og derefter blir de atter behandlet som IP6 pakker.

IP6 forklares i detalj i RFC 1883.

Klasseløse IP adresser:

Som vi har sett er IP adressene delt opp i klasser, avhengig av hvor stor del av adressen som kan benyttes til nettverksadresse og hvor stor del som kan benyttes til nodeadresse. Dette systemet virker, men ikke spesielt bra. Det kan hende du trenger mer enn en klasse C adresse (256 IP, 254 nodeadresser), men ikke sikkert at du trenger så mye som en Klasse B adresse (65536 IP, 65534 nodeadresser). Da kunne vi tenke oss at vi setter opp en IP adresse med en subnetmaske som ikke er en default subnet mask, men en tilpasset eller "custom" subnet mask.

I en klasse C adresse er det 8 bits i nodeadressefeltet og 24 bits i nettverksadressefeltet. 2^8 gir 256 IP adresser (vi kan bare bruke 254). Tilsvarende vil et klasse B nettverk ha 16 bits i nettverksadressefeltet og 16 bits i nodeadressefeltet.

Vi kan da angi en klasse C adresse slik: 193.212.14.251/24 fordi det er brukt 24 bits til nettverksadressefeltet. Da er det 8 bits igjen til nodeadresser. ($2^8=256 -2= 254$ noder). Tilsvarende for en klasse B adresse: 148.132.230.12/16 fordi det er 16 bits i nettverksadressefeltet. Da er det 16 bits igjen til nodeadresser. ($2^{16}=65536 -2=65534$). Binært ser det slik ut:

193.212.14.251/24

IP adresse: 11000001.11010100.00001110.11111011

Subnetmaske: 11111111.11111111.11111111.00000000

148.132.230.12/16

IP adresse: 10010100.10000100.11100110.00001100

Subnetmaske: 11111111.11111111.00000000.00000000

Tenk deg nå følgende forslag:

148.132.230.12/20

IP adresse: 10010100.10000100.11100110.00001100

Subnetmaske: 11111111.11111111.11110000.00000000

Vi tar bare halvparten av 3. oktett i subnetmasken. Det betyr at det er 20 bits til nettverksadressefeltet og 12 bits til nodeadressefeltet. $2^{12}= 4096 -2=4094$ noder, og det kunne jo passe bedre, eller hva? Ved å flytte litt frem og tilbake i den binære subnetmasken og angi alle tall til venstre for der du befinner deg som 1 og alle til høyre for null kan du lage så små eller store nettverk du vil, uten å være bundet til de forskjellige klassene. På denne måten kan man allokere (tildele) IP adresser på en mer "dynamisk" måte. Dermed tar det litt lengre tid før vi slipper opp for IP adresser, men det som virkelig redder oss er bruken av "private IP adresser".

Private adresser forklares i detalj i RFC 1918.

Øvelse:

Sett opp disse adressene binært (med binær subnetmaske☺) og regn ut hvor mange noder du kan ha på nettverket:

➤ 146.245.15.15/24

➤ 195.111.8.212/26

➤ 10.1.1.3/8

➤ 10.1.1.3/28

➤ 148.122.52.131/18

Øvelse 2:

Tenk deg nå at du snur litt på flisen og regner ut hvor mange nettverk du kan ha med de IP adresse/subnetmaske kombinasjonene over? Regn ut hvor mange nettverk det er i hvert av tilfellene.