IP – grunder och arkitektur

- Kapitlet beskriver grunddragen i hur IP fungerar samt hur under- och överliggande nivåer fungerar. Arkitekturen i TCP/IP beskrivs och hur TCP/IP relaterar till OSI-modellen. Skiktade arkitekturer förklaras kortfattat.
- Lokala nätverk förklaras med fokus på modern Ethernetteknik. Samverkan mellan IP, LAN och WAN i form av protokollet ARP gås igenom.
- Kapitlet är tänkt att kunna läsas fristående.

I datorernas barndom fanns det knappast en egen bransch som kallades datakommunikation. Däremot har det alltid varit viktigt hur information kan överföras, ett fundament inom en dator är ju att information hämtas, bearbetas och lagras. Själva grunden för detta är informationsöverföring. Claude Shannon, som lade grunden för den matematiska teorin för hur datorer skulle kommunicera med sitt banbrytande verk ”The mathematical theory of communication” på 1940-talet, hade fokus på hur information ska överföras.

Idag är datakommunikation en etablerad bransch men den har förändrats helt på 20 år. Förr så handlade branschen mycket om hur olika system skulle kunna utbyta information via speciell utrustning: protokollkonverterare. Denna typ av lösning blev mer och mer komplex, om det fanns 15 olika system och ett nytt slog igenom på marknaden så behövdes ju 15 nya protokollkonverterare. Idag använder vi nästan alltid TCP/IP-protokollen och alla system kan utbyta information.

När alla system i stort sett pratar samma protokoll behövs sällan protokollkonvertering. Men om alla system kan kommunicera med varandra så blir ju det ett problem i sig. Ska vi ha kontroll på vilka system som pratar med varandra så krävs konfiguration och en helt ny bransch har uppstått: nätverkssäkerhet. Tekniskt sett kan min dator utbyta information med Pentagon och Säkerhetspolisen. Klart att de vill ha kontroll på hur jag kan komma åt deras system. Och jag kan själv vara rätt intresserad av att begränsa deras möjligheter att komma åt min information. Det som skiljer våra system åt är ett par användarkonton med lösenord och kanske digitala certifikat. Vidare skiljs vi åt av ett regelverk, jag får inte ens försöka komma åt deras system och det finns förhoppningsvis även reglerat hur de får komma åt mitt.

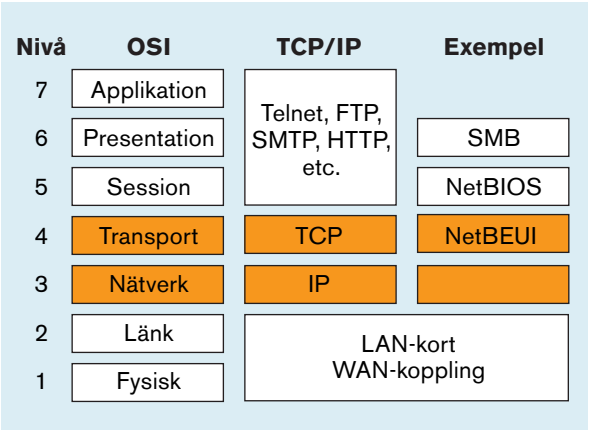
Arkitektur och standarder

Standarder är viktiga. När jag på min dator startar en webbläsare och skriver in webbadressen www.bolag.se så förväntar jag mig att det ska fungera. Men för att det ska fungera krävs att en massa saker ska vara standardiserat. På något sätt sitter min dator ihop med webbservern så vi måste komma överens om hur kontakter och kablage ska se ut (fysisk nivå). Vi måste komma överens om adresser och namn så att jag hamnar på rätt ställe (nätverksnivå). Hur ska vi dela mediet med andra så att inte alla sänder på en gång (länknivån)? Webbsidan ska delas upp i en massa små paket. Hur stora ska de vara? Hur ska de hamna rätt? Hur ska de sättas ihop igen? Om ett paket kommer bort när ska det sändas om? De sista frågorna hanteras av transportnivån. Vi måste vara överens om hur bilder och tecken ska tolkas (presentationsnivå). Och eventuellt ska vi anpassa kodningstekniken efter förbindelsens kvalitet.

En fördel med en skiktad arkitektur är att vi bestämt på vilken nivå varje funktion ska utföras så att vi till exempel inte försöker komprimera data som redan är komprimerat.

En standard för kommunikation är OSI, Open System Interconnection. Arbetet har bedrivits av ISO sedan 1980-talet och syftar dels till att ange en referensmodell med sju nivåer och dels till att ange standarder inom alla sju nivåer. I praktiken har TCP/IP ersatt OSI som praktisk tillämpning av en öppen och oberoende standard för kommunikation. OSI har till stor del blivit just en referens.

TCP/IP har också blivit ett praktiskt bevis på att skiktade arkitekturer fungerar. Det tar lite tid att göra saker i olika skikt men resultatet blir flexibelt och modulärt. Genom att beskriva gränssnittet mellan till exempel nivå två och tre så kan vi enkelt byta ut ett protokoll på nivå två eller tre mot att annat, gränssnittet är detsamma. Idén är inte konstig, få av oss vet hur en fax fungerar men vi vet hur vi ska använda den – hur gränssnittet fungerar.



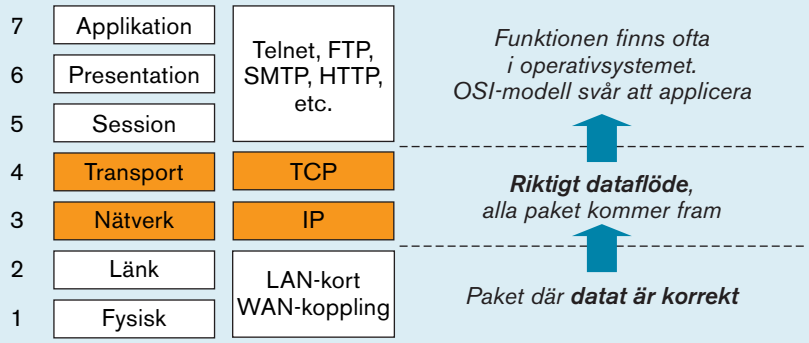
OSI-modellen och TCP/IP bygger bägge på skiktade arkitekturer. TCP motsvarar nivå fyra enligt OSI och IP nivå tre.

Kostnaden för flera skikt blir en hierarki som kan bli omständlig. En programmerare bör inte skriva kod som går direkt ut på en kommunikationsport och börjar skicka ettor och nollor. Istället blandas protokoll från flera skikt in och ska lägga till overhead och komplexitet. Vi får overhead på overhead vilket tar plats från vår egentliga nyttolast. Att kom-

municera över IP blir en kompromiss. Hur mycket information ska varje nivå tillåtas lägga till (hur lång header)? Hur långa paket? Hur ofta ska vi skicka kvittenser? Hur ofta ska vi sända om vid problem? Ska vi utveckla en teknik för att skicka till exempel audio över trådlös länk i 700 MHz-bandet så går det att hitta en mer effektiv teknik än att använda TCP/IP?

Men med TCP/IP får vi en mängd fördelar. Alla slags applikationer kan kommunicera över alla slags lokala nätverk och fjärrförbindelser. IP blir en gemensam nämnare för modern kommunikation. När vi kommunicerar via TCP/IP får vi en mängd funktioner på köpet för att hantera omsändningar, varierande fördröjning, adressering och routing. Ena gången kommunicerar vi med en server på vårt lokala nät, vi har gott om bandbredd, kort fördröjning och ett fåtal överföringsfel. Nästa gång kommunicerar vi med en server i en annan världsdel, för att hitta dit krävs drygt trettio routrar. Vi har ont om bandbredd, lång fördröjning och vart och vartannat paket försvinner på vägen.

TCP/IP ger en möjlighet för alla slags program att kommunicera över alla slags WAN (globala nät som ägs av operatörer) och LAN (lokala nätverk som Ethernet).



Tjänster från nivå två respektive fyra.

I modern datakommunikation kan IP lita på att nivå två hittar överföringsfel i varje länknivåpaket. Det förekommer bitfel (att en nolla felaktigt tolkas som en etta eller tvärtom) på grund av brus, svaga signaler och andra störningar. Men på nivå två lägger vi till en checksumma som beräknas baserat på de bitar som överförs. En mottagare beräknar checksumman och jämför med mottaget värde. Om de inte stämmer så kastas paketet. Jämför checksumman med sista siffran i personnumret, den kan bara visa att något är fel och inte användas för att rätta felet.

På nivå tre, nätverksnivån, lägger vi in funktioner för en global adresseringsfunktion. Routrar arbetar på nivå tre med IP-adresser för att kunna skicka paketen till rätt mottagare.

Paket kan komma bort på nivå två och tre. För att kunna erbjuda ett korrekt dataflöde lägger vi till nivå fyra, transportnivån. TCP tar emot data från överliggande nivå, delar upp det i lagom

stora paket och numrerar dem. Mottagande TCP-stack använder dessa sekvensnummer för att kontrollera att allt data kommit fram och vid behov begära omsändningar.

Två viktiga principer vid skiktade arkitekturer är också:

- Vi får inte hoppa över en nivå. Nivå fyra ska inte prata direkt med nivå två. Behöver detta ske får det ske via nivå tre.
- Sändare och mottagare kommunicerar med motsvarande nivå till exempel TCP till TCP. En TCP-stack på en maskin ska alltså inte ställa frågor till ett Ethernetkort på en annan maskin.

Dessa två är principer som man eftersträvar. TCP/IP är å andra sidan utvecklat av pragmatiker så om det behövs gör man någon gång små avsteg från dessa principer.

Allt över IP och IP överallt

Routrar använder mottagarens IP-adress för att avgöra hur ett IP-paket ska skickas vidare.

IP arbetar på nätverksnivå, protokollets viktigaste uppgift är routing (vägval). Med hjälp av IP-adresser kan rätt mottagare och avsändare adresseras och routrar tittar på denna adress för att avgöra hur paket ska skickas vidare.

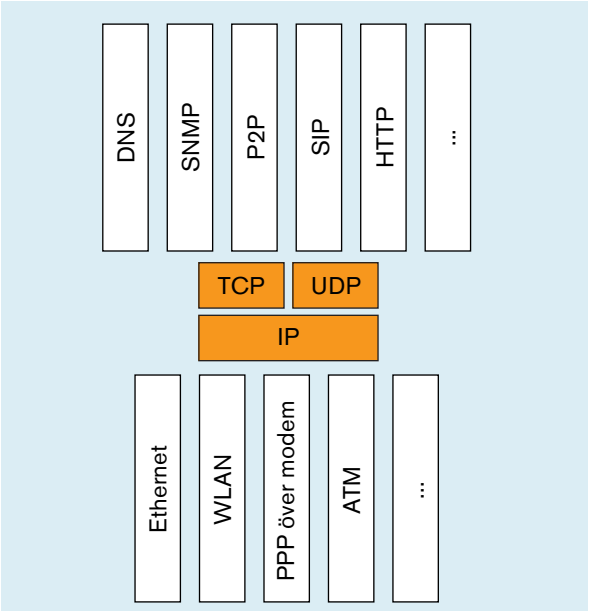
När en router skickar ut ett paket på ett gränssnitt kan den få problem med paketets storlek. Routern kan ha tagit emot ett paket med längden 1 024 byte men den länknivå som utgör nästa hopp accepterar inte större paket än 512 byte. Routern måste vid sådana tillfällen dela upp paket i mindre fragment och motsvarande process kallas fragmentering. Varje fragment hanteras som ett eget paket men det märks upp så att mottagaren kan sätta ihop de olika fragmenten till ett IP-paket igen. Exakt hur fragmentering fungerar behandlas i kapitlet IP-nivån.

Den här avsnittet heter inte ”Allt över TCP/IP och TCP/IP överallt” utan ”Allt över IP och IP överallt”. Det finns tillfällen då vi inte vill använda TCP, protokollet UDP (User Datagram Protocol) har funnits med sedan starten. Nya transportnivåprotokoll har också utvecklats som till exempel SCTP. Och IP kan även bära andra nivå tre-protokoll. Men vi har mycket att vinna på att använda en gemensam adressering. IP handlar ju ytterst om att vi sätter en siffra på saker och denna siffra kan vi använda för att hitta fram. Mycket blir enklare om vi gör detta med en gemensam standard: IP.

IP-adresser är strukturerade så att de kan slås ihop till nätverksadresser, vilket ger hanterliga routingtabeller.

IP har en struktur på sina adresser och det är detta som gör protokollet routbart. Vill vi bara ha en unik adress så kunde vi använt MAC-adresser (Media Access Control, se nästa avsnitt), men i så

fall skulle routrarna fått arbeta hårt. Med IP-adresser behöver routrarna inte veta varje enstaka IP-adress, istället arbetar man huvudsakligen med nätnummer. En router som ska föra paket vidare till IP-adresserna 195.6.7.8 samt 195.6.7.250 använder normalt det faktum att de tillhör samma nät 195.6.7.0. Detta gör att routingtabellerna kan kortas ner betydligt. För att detta ska fungera måste IP-adresser delas ut och administreras centralt. Dessutom måste självklart en IP-adress som används ute på Internet vara unik.



IP har en central roll om alla slags applikationer ska kunna kommunicera över alla slags WAN och LAN.

För att förstå IP:s genomslagskraft kan man dels hänvisa till att arkitekturen och protokollen är väldigt generella. Som många andra nätverkstekniker drar IP också nytta av något som brukar beskrivas som Metcalfe's lag nyttan ökar med kvadraten på antalet användare. Det här är ingen exakt lag (hur mäter man nytta?) men den förklarar varför en teknik som blir dominerande tar över en marknad helt. De som väljer att inte använda tekniken eller kommer in sent får ett kraftigt underläge. Det gäller inte bara IP utan även GSM och Ethernet.

En publik IP-adress delas ut centralt och är unik.

Givetvis finns det brister i Internetprotokollen. Men vi kan jämföra TCP/IP med engelska. Ska vi titta på vilket språk som är första språk för flest antal människor borde vi lära våra barn kinesiska. Som alla naturliga språk innehåller engelska oregelbundenheter och konstigheter vilket det tar många år att lära sig. Engelska är dessutom ovanligt svårt att stava, ordlistor kom tidigt och man var inte alltid så konsekvent. Det lär finnas 14 sätt att stava sh-ljudet, varav några finns i orden shoe, sugar, passion, ambitious, ocean och champagne. Det lär finnas 38 stavningar som uttalas ”air” eller ”aire” eller ”heir”. Och så har till exempel ordgruppen ”ough” fått en mängd olika uttal som through, though, thought, tough, plough och två – tre till. Trots det är engelska ohotat som internationellt språk för närvarande. Alla försök att hitta på nya språk för internationellt utbyte har misslyckats trots fördelar som fonetisk stavning och regelbunden grammatik.

Den främsta konkurrenten till IP som protokoll på nätverksnivå är IP själv. Idag använder vi IP version fyra (IPv4). Version sex (IPv6) finns framtagen och används i vissa delar av Internet. IPv6 är tänkt att kunna samexistera med IPv4 men det finns skill-

nader i hur protokollen fungerar. Det finns även principiella problem när Internet pratar två nätverksprotokoll. Hur ska vi till exempel hantera om jag har IPv4 och vill skicka e-post till en server som bara har en IPv6-adress?

Funktioner på TCP och UDP-nivå

IP-nivån ser till att paketen kommer fram till mottagaren, och protokollet ICMP (Internet Control Message Protocol) hjälper till om problem uppstår.

För att få ett korrekt dataflöde behövs dock lite mer. IP är förbindelselöst, vi skickar data till mottagaren utan att kontrollera om mottagaren är redo att ta emot data eller om den överhuvudtaget går att nå just nu. Om vi vill ha en förbindelseorienterad överföring får vi lägga till TCP (Transmission Control Protocol). TCP handskakar innan den egentliga dataöverföringen sker, tre paket utväxlas mellan klient och server och på så sätt kan vi vara säkra på att bägge parter är redo att överföra data.

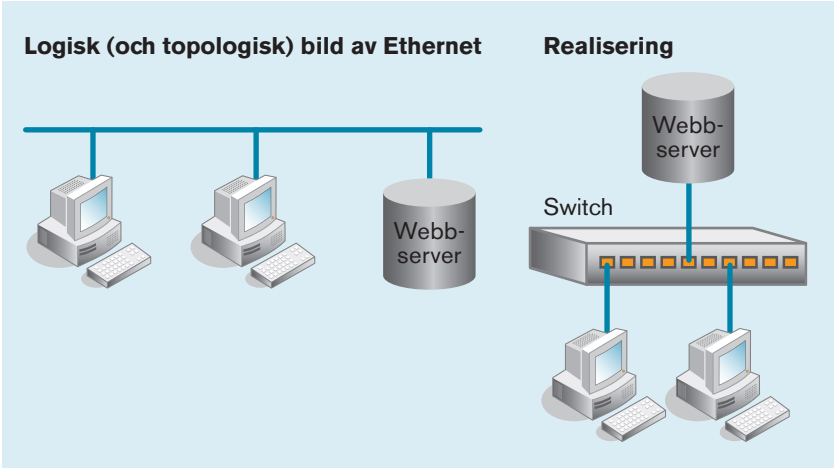
TCP lägger även till så kallade portnummer. Men hjälp av dessa kan flera applikationer dela på TCP/IP-stacken. Portnumret talar om vilken applikation (eller program eller process) som ska ta emot datapaketet. Samma server kan därför både vara webbserver (och svara på port 80) och telnetserver (port 23). Även på klienter används portnummer, det är därför vi kan surfa till flera servers på en gång eller överföra flera filer mot olika servrar simultant.

Det är även TCP:s uppgift att dela upp dataflödet i lagom stora delar och numrera dem. Med hjälp av dessa sekvensnummer kan mottagaren sedan meddela vilka paket som kommit fram och vilka som behöver sändas om.

Ibland behöver vi inte all denna funktionalitet. Då använder vi istället UDP (User Datagram Protocol). UDP är förbindelselöst och använder inga sekvens- eller kvittensnummer. UDP:s främsta uppgift blir bara att lägga till portnummer för att adressera rätt applikation. Ordet ”User” bär ingen speciell betydelse, protokollet lär från början ha haft namnet ”Unreliable” (otillförlitligt) men när TCP/IP började kommersialiseras tyckte man inte att den gamla betydelsen dög.

Lokala nätverk

Idag är Ethernet helt ohotat som teknik i lokala nätverk, det finns i princip inget alternativ förutom möjligen trådlösa nät – WLAN. Och även WLAN har en hel del gemensamt med Ethernet. ”Nät-



Logisk buss men fysisk stjärna.

verkskort" har blivit ett vanligt namn på Ethernetkort. Ethernets framgångssaga beror delvis på att Ethernet är en enkel teknik som det har varit lätt att bygga på, de brister som finns med Ethernet är dessutom väl kända. Ett annat skäl är Ethernets fantastiska utveckling. Från 10 Mbit/s i slutet av 1970-talet till 10 Gbit/s.

Vi ritar fortfarande Ethernet som en logisk buss, alla kan prata med alla. Vi väljer vilken station vi vill prata med genom att i Ethernetheadern ange avsändar- och mottagardress. Tekniskt sett kommer alla stationer att ha möjlighet att läsa paketets innehåll. Men Ethernetkorten filtrerar sedan bort de paket som är avsedda för en annan mottagare, på så sätt behöver inte operativsystem eller programvara arbeta med onödigt data. Klassiskt Ethernet (IEEE 10Base-5 och 10Base-2) använde en koaxialkabel som fysiskt medium, så den fysiska bilden av Ethernet stämde bra med den logiska.

Ethernetkretsar filtrerar bort paket som är avsedda för andra mottagare än vår MAC-adress.

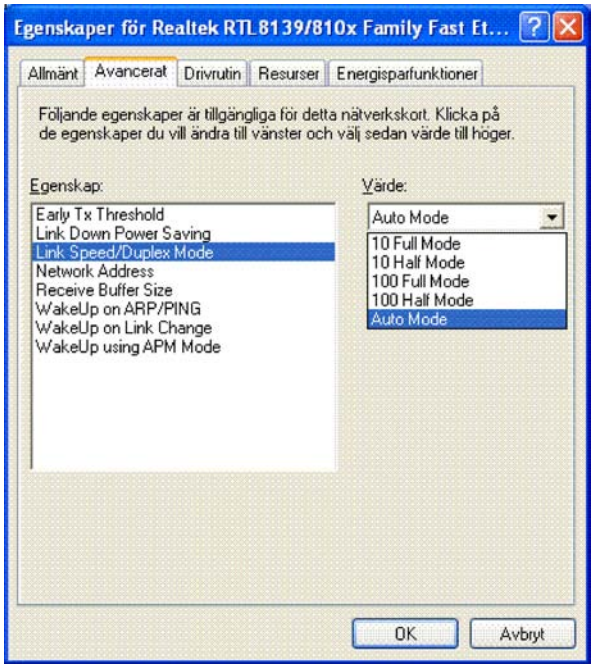
Idag byggs Ethernet som stjärnformade nät. I centrum sitter en Ethernetswitch eller en hubb. Logiskt sett kan man gärna ha bilden av en buss framför sig. Kabeltypen som används är så gott som uteslutande partvinnad kabel eller fiber och hastigheten finns i flera varianter från 10 Mbit/s till 10 Gbit/s. Den vanligaste standarden är 100Base-T (100 Mbit/s över partvinnad kabel, T som i twisted pair). Byter vi sista bokstaven mot ett F så avser standarden fiber. Byter vi första siffran mot 10 eller 1 000 så avser standarden 10 respektive 1 000 Mbit/s.

100Base-T kallas ofta "Fast Ethernet", det finns egentligen fler varianter. Den vanligaste standarden betecknas egentligen 100Base-TX.

100Base-T kallas ofta Fast Ethernet och avser en standard om 100 Mbit/s över partvinnad kabel.

Att bygga ett Ethernetbaserat nätverk har blivit lätt. Standarder har medfört billig och driftsäker utrustning. Ethernet har gått från svårhanterligt rörmokeri med koxialkablage till "plug and play" med färdigt partvinnat kablage. Att Ethernet är lätt att bygga innebär dock inte att det inte kan krångla.

1. Ethernet består av flera standarder. Mycket av detta löser sig genom autokonfigurering men det kan bli problem när all utrustning ska lyssna och ställa in sig. Ibland kan man vara tvungen att ställa in hastigheten på ett lokalt nätverk manuellt.



Typiska Ethernet-inställningar.

2. När partvinnat slog igenom fanns möjlighet till full duplex, ett kabelpar används för sändning och ett för mottagning. Samma princip som fall 1, utrustning ska klara av detta själv men vissa fabrikat fungerar dåligt ihop.

3. Partvinnat kablage fanns ursprungligen i två varianter: rakt kablage avsett att användas mellan en hubb och en dator och korsat kablage avsett för kommunikation dator till dator eller hubb till hubb. Många tillverkares utrustning kan idag ställa om automatiskt för korsat kablage men ibland fungerar det inte. Länklampan ska lysa i bägge ändar, annars kan man misstänka problem med kablaget.

4. Det är normalt med en viss paketförlust på Ethernet, men den bör bara ligga på någon eller några procent. På mer avancerad utrustning skiljer man på tidiga och sena (late eller out-of-windows collisions) kollisioner. För långt kablage eller för många kopplingar mellan switchar och hubbar leder till sena kollisioner, dessa ska inte förekomma på ett väl fungerande Ethernet.

5. Prispressen på Ethernetutrustning har lett till att utrustningen har minimala buffertar på varje port. Om två paket kommer in samtidigt måste utrustningen buffra paketet tills bussen blir ledig. Lågprisutrustning kan dessutom ha svårigheter med att hantera flera MAC-adresser på samma port. Detta gör att paket försvinner, och applikationer kan beskyllas för att fungera dåligt när felet egentligen ligger i det lokala nätet.

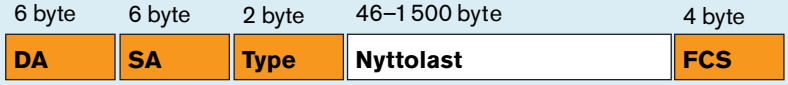
För att förstå varför paketförluster uppstår på Ethernet behöver vi titta lite noggrannare på den accessmetod som används. Accessmetoden anger hur stationerna ska dela på den gemensamma kanalen. För Ethernet heter denna metod CSMA/CD (Carrier Sense Multiple Access with Collision Detection). Tekniken är i princip

enkel och påminner om hur (väluppfostrade) gäster delar på möjligheten att samtala under en middag. Alla har tillgång till en och samma kanal (Multiple Access). Lyssna först, om det är ledigt (Carrier Sense) så sänd. När du sänder så lyssna samtidigt, om avsänt meddelande skiljer sig från det du avlyssnar (Collision Detection) så beror det på en krock, någon annan sänder samtidigt. Avbryt i så fall genast, vänta en slumptid och försök på nytt.

Styrkan i den här algoritmen är dess enkelhet. Den kan lätt göras tusen gånger snabbare bara elektroniken hänger med. Den bygger heller inte på att någon station i nätet tar kontrollen utan alla stationer är jämlikar. Men det har en del inbyggda problem. Tekniskt sett kan vi aldrig garantera hur lång tid det tar innan ett meddelande kommer fram, det beror på vad de andra stationerna vill sända. Även om stationerna följer regelverket till punkt och pricka så kan två stationer börja sända i stort sett samtidigt innan de hinner uppfatta den andra stationens sändning. Därför uppstår kollisioner på Ethernet. Den huvudsakliga lösningen på detta problem är att inte bygga för stora nät, tekniskt sett säger man att man minskar kollisionsdomänen, samt att se till att ha gott om bandbredd. Ju mer bandbredd desto snabbare går det att skicka meddelandet och desto mindre risk att någon annan vill sända samtidigt.

Ethernets paketformat

Det finns flera olika standarder för hur Ethernetpaket kan se ut, detta var ett komplext problem under 1980-talet med olika Ethernet-inkapslingar. Idag används främst Ethernet typ II som är enkel att förstå.



DA Destination Address, mottagaradress
SA Source Address, avsändaradress
Type Detta fält anger vilken typ av nyttolast som bärs
FCS Frame Check Sequence. Checksumma

Fälttyper inom Ethernet typ II.

Inom Ethernet typ II har vi en minimal header bestående av 14 byte uppdelade på destinationsadressen (DA), avsändaradressen (SA) samt typfältet. Datafältet består av 46–1 500 byte och kan innehålla viss utfyllnad (padding). På slutet lägger vi till en 32-bitars checksumma och kommer totalt upp i en paketlängd om 64 till 1 518 byte. Ethernetpaket har alltså både en minimal och maximal längd.

Typfältet används för att deklarera vilken typ av datainnehåll ramen innehåller. Fältet består av två byte. Till exempel betyder 0x0080 att innehållet är IP.

Både avsändar- och mottagaradresserna består av 6 byte eller 48 bitar. Den här typen av adresser kallas MAC-adresser där MAC står för Media Access Control, även begreppet fysisk adress används. En MAC-adress kan till exempel vara 00-50-DA-69-D7-35.

Varje Ethernetkort har en unik MAC-adress om 48 bitar som till exempel 00-50-DA-69-D7-35. Då de första tre byten är specifika för tillverkaren (3COM i detta fall) skrivs de ibland på formatet 3COM-69-D7-35.

En tillverkare av Ethernetkort registrerar sig hos organisationen IEEE och tilldelas en egen nummerserie (de tre första byten). Sedan ansvarar tillverkaren för att varje kort eller chip får ett unikt löpnummer, de sista tre byten. Varje MAC-adress är alltså unik och kan användas för att adressera just ett unikt Ethernetkort, ett WLAN-kort, ett Token Ring-kort eller en Bluetooth-enhet.

Ethernet typ II har som sagt en väldigt enkel header. Det finns tillägg till standarden för att hantera olika logiska nät (VLAN) och prioriteringsfunktioner. Standarderna för dessa kallas 802.1Q respektive 802.1p.

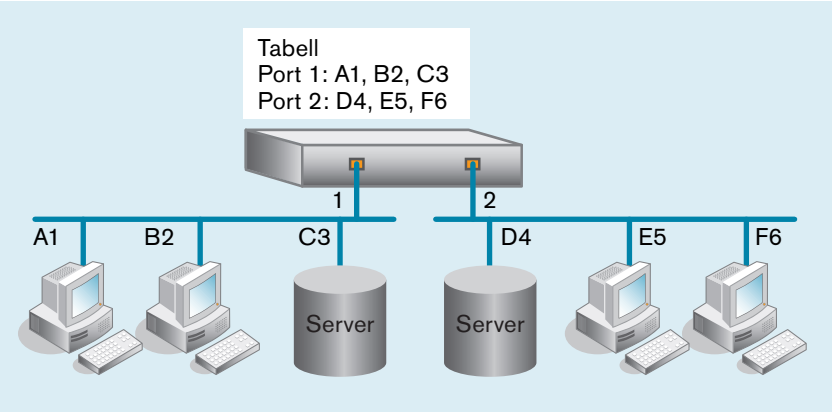
Ethernethubbar och Ethernetswitchar

En nackdel med äldre Ethernetarkitekturer som de koaxialbaserade 10Base-2 och 10Base-5 blev uppenbar när näten blev mer spridda och mer omfattande. Ett dåligt nätverkskort eller en glappande kontakt kunde drabba ett stort antal användare eftersom signalen inte kom vidare. Nätet blev aldrig bättre än dess svagaste länk.

En annan typ av lokalt nätverk, Token Ring, hade också problem med driftsäkerhet och där hade man löst det genom att bygga en centralt placerad enhet mitt i nätet: ett nav (hub). Nätverkskort eller kablage som fungerar dåligt kopplas bort av hubben. Konceptet spreds även till Ethernet. Hubben kan dessutom fungera som förstärkare (repeater) innan signalen skickas ut på varje enskild nätverksport.

Ethernet blev snabbt populärt och näten blev större och större. Ju större nät desto mer sannolikt att två noder vill sända samtidigt och risken för kollisioner ökar. En lösning på detta blev så kallade Ethernetbryggor. En brygga bygger upp tabeller genom att titta på avsändaradresser, på så sätt lär den sig vilka noder som sitter på respektive port.

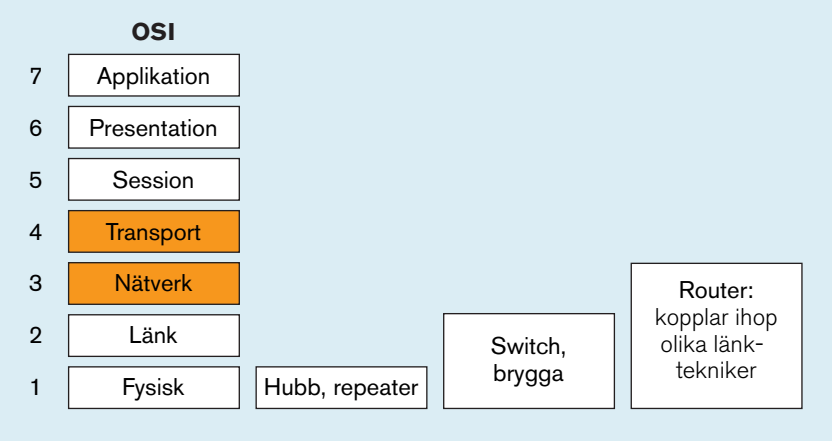
Fördelen med en brygga är att A1, B2 och B3 kan kommunicera för fullt utan att det stör det högra nätet. Vi får två separata så kallade kollisionsdomäner. Om däremot A1 vill skicka data till E5 så ser bryggan att detta paket måste vidarebefordras. Men för noderna är det helt transparent vilket segment noderna sitter på.



En brygga bygger själv upp tabeller med vilka adresser som sitter på respektive port.

I början av 1990-talet fanns multiportsbryggor men de var dyra. Tillverkaren Kalpana började marknadsföra sina multiportsbryggor som ”Ethernetswitchar” med stor framgång då priset per port var förhållandevis lågt. Switcharna lärde sig MAC-adresser på samma sätt som bryggor men man införde också ett snabbare sätt att skicka Ethernetpaket vidare. Bryggor läste in hela paketet och kontrollerade checksumman innan det skickas ut, så kallad store-and-forward. Men tekniskt sett kan vi skicka ut paketet så fort vi läst av Ethernetheadern eller till och med bara mottagaradressen. En vanlig teknik kallas för fragment free, då läser switchen in 64 byte innan det skickas ut. Vid 10 Mbit/s skulle det ta cirka 1,2 millisekunder att läsa in 1 518 byte, att läsa in 64 byte tar istället cirka 50 mikrosekunder. En avsevärd förbättring. Observera att en modern switch också måste kunna hantera store-and-forward för att klara olika hastigheter på respektive port.

Idag är Ethernetswitchar mycket billiga. De har i princip slagit ut hubbar från marknaden. Även utrustning som säljs som hubbar är i teknisk mening ofta switchar. Ur säkerhetssynpunkt är switchar att föredra då de lär sig vilken eller vilka MAC-adresser som



Genom OSI-modellen kan man tydliggöra skillnaden mellan hubbar, switchar och routrar.

sitter inkopplade på respektive port och bara skickar information till denna adress (samt multi- och broadcast), detta gör det svårare att avlyssna trafik till och från andra stationer. I labbsituationer kan dock hubbar vara att föredra då all trafik skickas ut på alla portar, det gör det lättare att analysera trafik.

Man kan tydliggöra skillnaden mellan hubbar, switchar och routrar genom OSI-modellen. En hubb jobbar på nivå ett genom att göra en ny kopia på paketet, switchen arbetar på nivå två genom att använda MAC-adresser. Routern jobbar på nätverksnivå med IP-adresser.

IP och samverkan med länknivån

För att adressering ska fungera ska varje nod i nätverket ha en unik adress. På länknivå är MAC-adresser vanligast. Ger du din Windows-dator kommandot ”ipconfig” så visar den vilken eller vilka MAC-adresser som används. MAC-adresser kallas ibland även fysisk adress.

```
Ethernet-kort Local Area Connection:

        Anslutningsspecifika DNS-suffix .
        Beskrivning . . . . . . . . . . . : Realtek RTL8139
        Fysisk adress . . . . . . . . . . : 00-0B-5D-C2-2F-68
        DHCP aktiverat  . . . . . . . . . : Nej
        Autokonfiguration aktiverat  . . . : Ja
        IP-adress . . . . . . . . . . . . : 192.168.3.245
        Nätmask . . . . . . . . . . . . . : 255.255.255.0
        Standard-gateway  . . . . . . . . : 192.168.3.1
        DHCP-server . . . . . . . . . . . : 192.168.3.1
        DNS-servrar . . . . . . . . . . . : 192.168.3.2
```

Ett exempel på resultat från kommandot ”ipconfig /all”. Vi kan se både IP- och MAC-adress. MAC-adresser kallas även fysiska adresser.

Vi skulle kunna använda MAC-adresser för att bygga ett globalt nätverk men en fördel med IP-adresser är att de har en hierarkisk uppbyggnad, den första delen av adressen används för att peka på vilket nätverk adressen hör till.

Jämför IP-adresser med postadresser. Vi skulle kunna använda personnummer för att hitta rätt mottagare, men det blir ett hårt arbete för postverket. Istället använder vi gatuadresser, jämför med IP-adresser. Det är bara den sista instansen som har kontroll på exakt var ”Kungsgatan 12 i Stockholm” ligger, de som ligger

tidigare i kedjan tittar mer på Stockholm och Kungsgatan. Detta gör att adresseringstabellernas storlek kan hållas nere.

Det finns tre principiellt olika sätt att adressera en eller flera mottagare. Begreppen används lite olika i olika tekniker men principerna är de samma. Dels kan vi adressera en enstaka nod, detta kallas för unicast, och är den vanligaste adresseringsmetoden. Sedan kan vi adressera alla noder i ett nätverk, detta kallas broadcast. Broadcast används mycket i lokala nätverk, en nod kan fråga efter alla servertjänster av en viss typ. En nod kan till exempel fråga efter alla skrivarservers, alla noder på nätet kontrollerar om de ska svara. De som svarar skickar tillbaka information innehållande namn och typ av tjänst. Detta visas som en lista i klienten, och det är sedan slutanvändaren som väljer vilka server som ska användas. Kommunikationen övergår sedan till vanlig unicast.

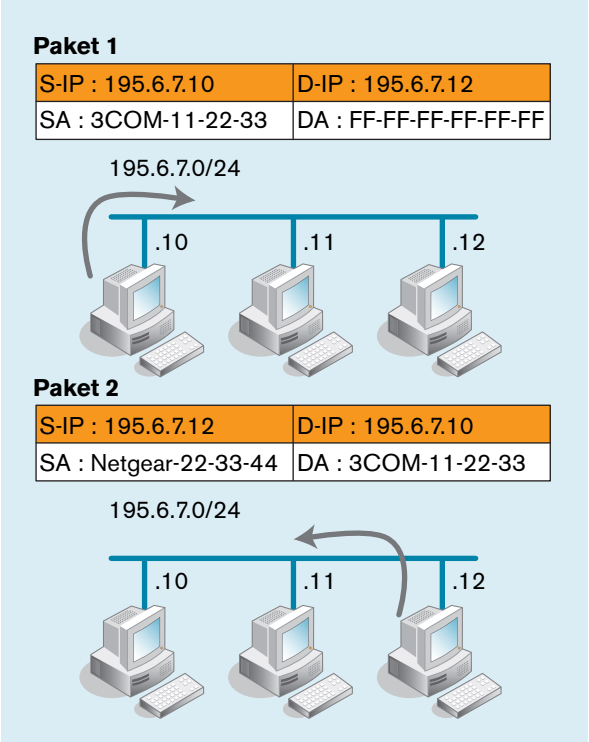
Broadcast fungerar inte på Internetnivå, det skulle bli svårhanterligt om en nod frågar efter ”alla skrivare”. Därför begränsar alla routrar broadcastfrågor så de inte vidarebefordras.

Ett mellanting mellan dessa två tekniker är multicast. I multicast adresserar vi flera noder, alla noder som tillhör en viss multicastgrupp. I multicasting finns också flera klasser med olika räckvidd (scope) som exempelvis lokal länk, organisation eller global.

ARP

Om en dator ska skicka IP-paket till en annan behöver den ta reda på mottagarens MAC-adress. Annars skulle paketet filtreras bort av mottagarens länknivå. Protokollet för att lösa detta problem kallas Address Resolution Protocol, ARP och används bland annat på Ethernet, Token Ring och WLAN. En motsvarande mekanism finns även inom Frame Relay för att finna så kallade DLCI-värden, där den kallas Inversed ARP.

Noden med IP-adress 195.6.7.10 behöver ta reda på MAC-adressen för nod 195.6.7.12. Den skickar då en så kallad ARP Request med en mottagaradress på Ethernetnivå där alla bitar satts till 1



ARP-handskakning. Begreppen i denna bild används genomgående: SA och DA står för Source och Destination Adress (MAC-nivå). S-IP och D-IP står för Source respektive Destination IP. Handskakning visas ovan på nivå två och tre.

(FF-FF-FF-FF-FF-FF). Alla Ethernetkort tar då emot paketet och granskar innehållet, en ARP Request. Den mottagare som har rätt IP-adress besvarar med en ARP Response och skickar då tillbaka sin MAC-adress till avsändaren.

Resultaten cachas av bägge noderna. När nod 195.6.7.12 får ARP-frågan så mellanlagrar den MAC-adressen tillhörande 195.6.7.10. På motsvarande sätt kommer nod 195.6.7.10 att mellanlagra resultatet, i detta fall MAC-adressen tillhörande 195.6.7.12. Bägge noderna tar för givet att ARP-handskakningen efterföljs av kommunikation mellan noderna, så cachen sparas i ett par minuter.

Du kan se innehållet i arp-cachen genom kommandot "arp -a".

```
C:\Documents and Settings>arp -a
Interface: 192.168.111.2 --- 0x3
  Internet Address      Physical Address      Type
  195.6.7.12            00-90-7f-22-33-44     dynamic
```

Med kommandot "arp -a" visas innehållet i arp-cachen. Du kan även testa vilka växlar som finns till kommandot. Till exempel finns möjlighet att lägga in fasta mappningar mellan IP- och MAC-adresser.

Nedan visas resultatet av en LAN-analys från en ARP-fråga och motsvarande svar. Analysen har gjorts med programmet Wireshark.

Paket 1
Ethernet II,
Destination: Broadcast (ff:ff:ff:ff:ff:ff)
Source: **Fujitsu_c3:2f:98 (00:0b:5d:c3:2f:98)**
Type: ARP (0x0806)
Address Resolution Protocol (request)
Hardware type: Ethernet (0x0001)
Protocol type: IP (0x0800)
Hardware size: 6
Protocol size: 4
Opcode: request (0x0001)
Sender MAC address: Fujitsu_c3:2f:98
Sender IP address: 195.6.7.10
Target MAC address: 00:00:00:00:00:00
Target IP address: 195.6.7.12

Paket 2
Ethernet II
Destination: Fujitsu_c3:2f:98 (00:0b:5d:c3:2f:98)
Source: 3comEuro_9f:4a:fa (00:0f:cb:9f:4a:fa)
Type: ARP (0x0806)
Address Resolution Protocol (reply)
Hardware type: Ethernet (0x0001)
Protocol type: IP (0x0800)
Hardware size: 6
Protocol size: 4
Opcode: reply (0x0002)
Sender MAC address: 3comEuro_9f:4a:fa
Sender IP address: 195.6.7.12
Target MAC address: Fujitsu_c3:2f:98
Target IP address: 195.6.7.10

Exempel på en ARP-fråga och motsvarande svar. Trafiken har spelats in med en LAN-analysator.

Lägg märke till att Ethernet version II används och att typfältet anger att innehållet är ARP. Lägg också märke till hur Wireshark gör en översättning från de tre första byten i en MAC-adress till vilken tillverkare som registrerat denna serie.

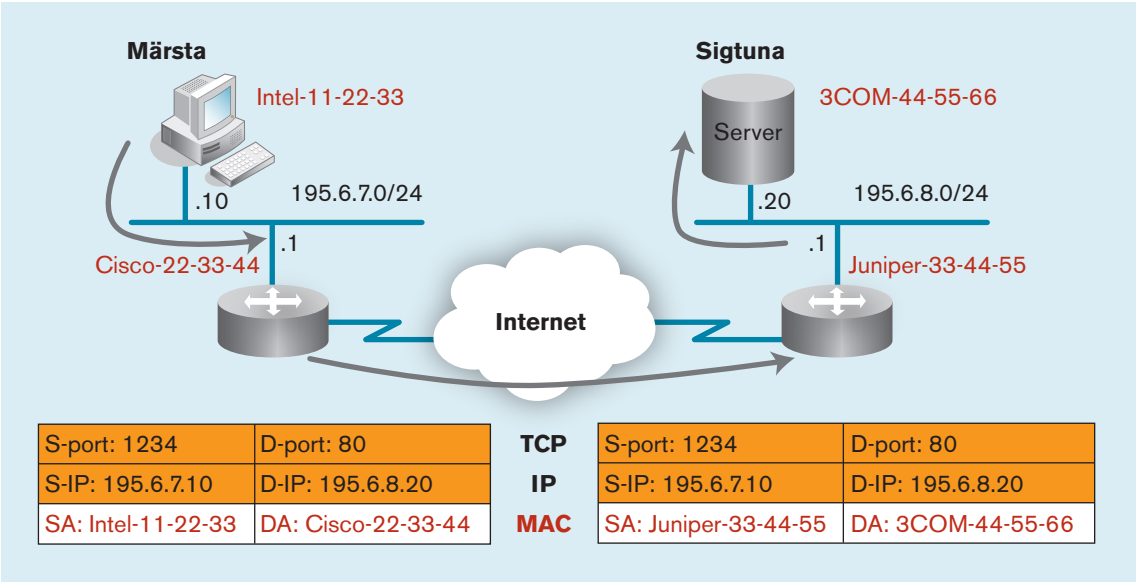
ARP och säkerhet

ARP är en grundläggande mekanism i moderna nätverk och på ett stort nätverk förekommer flera ARP-frågor i sekunden. Mekanismen är stabil och mycket transparent, vi behöver sällan fundera på den. Ändå finns det principiella säkerhetsproblem med funktionen.

En ARP-fråga skickas ut till alla noderna på ett nätverk. En nod som innehåller skadlig kod eller ett hackarverktyg kan välja att svara på ARP-frågor och ange sin egen MAC-adress som den eftersökta. Avsändaren kommer då att skicka alla paket till denna nod, förutsatt att den hann svara innan rätt nod skickade sitt svar. Noden kan nu agera mellanhand och analysera och kopiera alla paket till och från avsändaren.

Samverkan mellan TCP, IP och Ethernet

Vi ska titta på hur nivå två till fyra samverkar vad det gäller adressering. Om det verkar krångligt så var lugn, vi kommer tillbaka till samtliga funktioner lite senare. I Märsta vill en klient skicka



Adressering på Ethernet-nivå och IP-nivå frigör sig från varandra. MAC-adresserna används bara lokalt.

över kommandot ”GET / HTTP/1.0” till en webbserver i Sigtuna (detta skulle få webbservern att skicka HTML-text från sin webbsida). Webbservern lyssnar på port 80, och klienten skickar från port 1234. Det är mekanismen portnummer på transportnivån som gör att en nod kan hantera en mängd dataströmmar mot andra noder simultant, den kan se vilket program som ska ha datainnehållet.

Lägg märke till att adresseringen på länknivå (MAC) och nätverksnivå (IP) frigör sig från varandra. IP-adresserna är globala och de är de samma över alla typer av länkar. MAC-adresserna används bara lokalt, trots att vi ska till Sigtuna så är det routern som finns på det lokala nätverket i Märsta som vi adresserar. MAC-adresser är alltså inte intressanta för ett brandväggsskydd medan IP-adresser och portar kan användas.