

Datorn i världen, världen i datorn

*En introduktion till data-
och informationsteknik*

HANS LUNELL

Datakommunikation och nätverk

Hittills har vi enbart sett hur en enskild dator är uppbyggd och fungerar. Men som vi alla vet står datorer i förbindelse med varandra. De kan stå i samma rum eller byggnad eller på skilda kontinenter och ändå kommunicera med varandra. Det här kapitlet beskriver hur det går till. Datakommunikation är dock ett utomordentligt stort område och här kan vi bara skrapa litet på ytan av det.

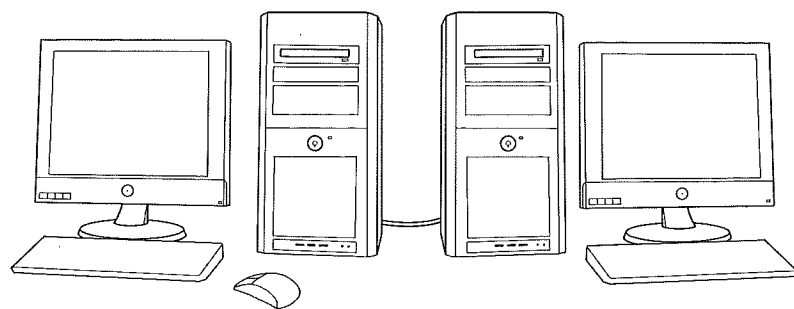
Området är stort av flera skäl. För det första är det komplext – att föra över en fil från ett hörn av jorden till ett annat är inte lätt, speciellt med tanke på att filen ska passera ett antal mellanliggande datorer av olika fabrikat och med olika egenskaper. Att göra detsamma med musik, röst eller video i realtid är förstås ännu mer komplicerat.

För det andra, och som en konsekvens av det första, är området omfattande – det finns en lång rad principer och standarder för hur kommunikationen ska gå till. Det är också ett område som är under ständig förändring genom att nya metoder läggs till de redan existerande.

Något av allt detta beskrivs i det här kapitlet. Första halvan handlar om datakommunikation generellt, medan den andra delen tar upp nätverk.

5.1 Hur går det till?

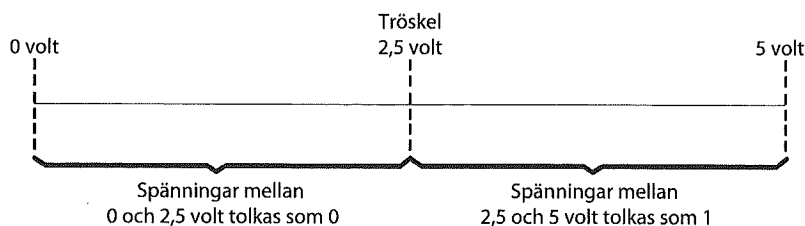
Hur kan två datorer utbyta data med varandra? Man kan bara genom att använda sunt förnuft komma fram till ungefär vad som krävs för att man ska kunna överföra t.ex. en fil från en dator till en annan. Det är vad det här avsnittet handlar om. Senare avsnitt beskriver de enskilda aspekterna närmare.



FIGUR 5.1 Två ihopkopplade datorer.

Först och främst måste det finnas en fysisk förbindelse mellan datorerna. Vi kan tänka oss att det helt enkelt är en kabel som är ansluten till dem, men det skulle också kunna vara en trådlös förbindelse, t.ex. radiovågor eller IR-ljus. Men då början och slutet på förbindelsen är elektriska bortser vi för tillfället från de andra möjligheterna. Vi tänker oss alltså en konfiguration som i figur 5.1.

Den ena datorn, avsändaren, kan via kabeln skicka elektriska pulser till den andra, mottagaren. Eftersom data lagras binärt, som ettor och nollor, är det naturligt att utforma pulserna så att en viss puls betyder 1 och en annan 0. Men för att det ska fungera krävs att mottagaren vet hur avsändaren sänder en etta resp. nolla. Exempelvis skulle man kunna komma överens om att pulser med spänningen 0 volt (dvs. ingen spänning alls) betyder 0 medan 5 volt betyder 1. Och eftersom avvikelser är oundvikliga i verkligheten måste man tillåta att värdena varierar inom ett intervall, t.ex. som i figur 5.2.



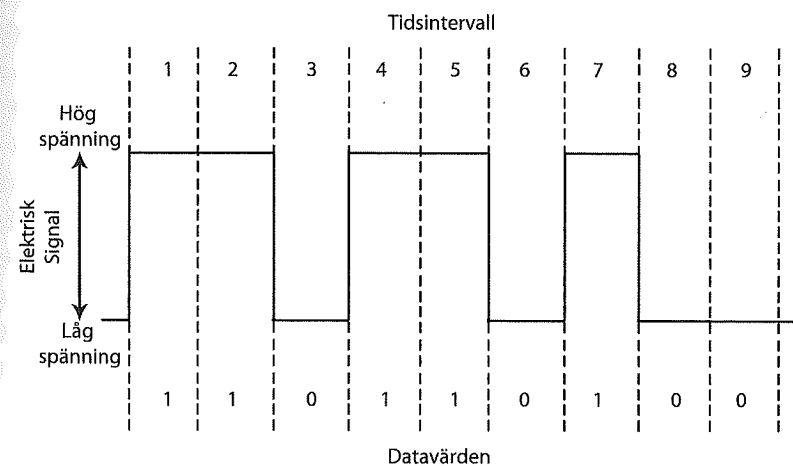
FIGUR 5.2 Spänningsintervall för att representera 0 och 1.

Dessutom krävs att sändare och mottagare är överens om vilken hastighet som används, så att mottagaren kan avgöra hur många nollor och ettor som sänds om spänningen är låg resp. hög under en viss tid. Det här illustreras i figur 5.3 där hög och låg spänning betyder följder av ettor och nollor.

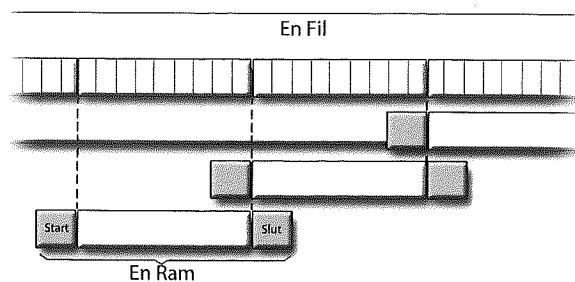
Slutligen måste mottagaren veta när det kommer data från den andra datorn. Det här kan göras på två sätt. Antingen skickar sändaren hela tiden data, som – när det inte finns något att överföra – helt enkelt betyder just detta: inga data. Det kallas *synkron* överföring. Eller också får förbindelsen vila under tiden ingen kommunikation äger rum. Man talar då om *asynkron* överföring. Innan överföringen kan komma igång måste i detta fall mottagaren ”väckas upp” med hjälp av en *startsekvens* bestående ett antal startbitar med ett överenskommet mönster. Startsekvensen ger också mottagaren möjlighet att synkronisera sig mot avsändaren. En *slutsekvens* talar om när filen är överförd.

Med de här sakerna på plats så har man definierat allt som krävs för att det teoretiskt sett ska vara möjligt att överföra t.ex. innehållet i en fil från den ena datorn till den andra. Tillsammans definierar de det *fysiska skiktet*.

Dataöverföring via det fysiska skiktet består alltså av att en avsändare skickar en ström av ettor och nollor som bildar startsekvensen följt av filen och slutsekvensen (vid asynkron överföring).



FIGUR 5.3 Pulståg för en binär signal.



FIGUR 5.4 Uppdelning av en fil i ramar.

Tyvärr räcker dock inte det fysiska skiktet. Det beror på att risken är för stor att filen inte kommer fram helt korrekt. Tänk på att även en liten mp3-fil innehåller flera tiotal miljoner bitar, så även med ett fel på 10 miljoner så skulle var och varannan fil komma fram felaktig.

I stället delar man upp filen i ett antal s.k. *ramar* som var och en har en start- och slutsekvens. Man skickar över och verifierar varje ram för sig. När alla ramar är mottagna fogar man ihop dem till en enda fil, och kan på det sättet återskapa en kopia av ursprungsfilen, se figur 5.4. Ramarnas utformning och innehåll definieras i det s.k. *länkskiktet*. I vårt tänkta fall med två direkt förbundna datorer räcker det med att båda datorerna vet hur länkskiktet är definierat för att de ska kunna kommunicera.

När man vill skicka data mellan datorer som inte är direkt förbundna med varandra kommer data att passera över mellanliggande utrustning, såsom andra datorer. Då behövs mer information än den som finns i länkskiktets ramar. För att ge plats åt den informationen gör man därför en ny uppdelning på en högre nivå, nätverksnivån. Här talar man om *paket* och förstås om *nätverksskiktet*.

Ett paket innehåller – förutom data – bl.a. identiteterna för avsändaren och mottagaren. Oftast numreras dessutom paketet eftersom det inte är säkert att de kommer fram i rätt ordning. Mottagaridentiteten används av mellanliggande utrustning för att bestämma vart filen ska skickas vidare, och ibland även av mottagaren för att upptäcka att paketet ska tas emot. Avsändaridentiteten används för att kvittera mottagna paket eller för att

begära att paketet sänds om. Och paketnumret används för att foga ihop paketen när alla paket har tagits emot.

På detta sätt byggs datakommunikationen upp i ett antal skikt ända tills man når den högsta nivån, där avsändaren skickar en webbsida (säg) och mottagaren presenterar den på skärmen. Den här högsta nivån kallas *applikationsskiktet* då det ju är knutet till en tillämpning eller applikation, i vårt tänkta fall en webbläsare.

Men hur vet mottagaren att filen som skickas över är riktig, dvs. att inget fel har introducerats under överföringen? Och hur vet avsändaren att mottagaren har tagit emot filen? För fel kan ju uppstå. Det kan lätt bli elektriska störningar som gör det svårt att avgöra om det har anlänt en etta eller en nolla. I värsta fall kan ju störningarna förorsaka att det som skulle vara en etta ser ut som en nolla eller tvärtom.

Svaret på den första frågan, om fel introducerats eller ej, är att avsändaren skickar med extra data som gör det möjligt för mottagaren att avslöja (vissa) fel i överföringen. Det finns många metoder för det här, men en – som både är enkel att beskriva och förstå – är att man tillfogar en *checksumma*. Enkelt uttryckt summerar man ihop alla bytes (oktetter) i paketets innehåll och skickar med summan. Mottagaren kan då göra samma beräkning och kontrollera att summan blir densamma.

Svaret på den andra frågan, hur avsändaren vet att paketen har tagits emot, är att mottagaren skickar en *kvittens* varje gång ett korrekt paket har anlänt, och en *negativ kvittens* om det innehåller fel. I det senare fallet kan avsändaren skicka om paketet och hoppas på att det går bättre. Om överföringen misslyckas många gånger, ger parterna upp.

Som synes av den här beskrivningen handlar dataöverföring och datakommunikation mycket om att de ingående parterna ska komma överens om hur det ska gå till. Det gäller hur nollor och ettor ska representeras, hur större datamängder (filer) ska delas upp i ramar och paket och hur paketen ska vara utformade, hur man ska upptäcka och rätta fel, och hur man kvitterar mottagna data. Ett gemensamt ord för allt detta är *protokoll*, och datakommunikation definierar protokoll på många olika nivåer. Dessutom finns det skilda protokoll för olika behov och uppgifter.

De följande avsnitten går in på olika modeller och protokoll för datakommunikation mer i detalj.

5.2 Protokoll och kommunikationsprincip

Det förra avsnittet gav en informell introduktion till kommunikation mellan datorer, introducerade några begrepp och skisserade några principer. I det här avsnittet ser vi på detta ur ett generellt perspektiv och beskriver vad som menas med "protokoll" och hur kommunikationen i princip går till.

PROTOKOLL

Ett protokoll är en uppsättning regler och överenskommelser för hur två parter ska kommunicera. Protokoll i den meningen finns överallt. Exempelvis finns regler för hur personer umgås. När man ses, hälsar man. I den västerländska kulturen sker det genom handskakning. Om det är första gången man ses, presenterar man sig. När man skiljs, tar man farväl, ofta också genom handskakning. I skolan ber en elev om lärarens uppmärksamhet genom att (tyst) vifta med handen. Mer formella exempel på protokoll finns också. Ett exempel är kommunikationen mellan pilot och flygledare inför en landning. Inom diplomatin används t.o.m. termen "protokoll" för att beteckna regler för hur olika ärenden ska hanteras.

Inom datakommunikation definierar alltså ett protokoll exakt vad och hur något ska kommuniceras. Definitionen måste vara exakt eftersom en dator inte kan gissa sig till vad en avsändare avser.

Protokoll finns, som sagt, på många olika nivåer, och även om de skiljer sig från varandra i komplexitet och innehåll – beroende på användningsområde – har de ofta flera gemensamma egenskaper. Typiskt är att de innehåller regler för:

- Hur den underliggande fysiska förbindelsen upptäckas
- Handskakning
- Hur man kommer överens om diverse kommunikationsegenskaper, t.ex. hastighet
- Vad som markerar start och slut på ramen eller paketet, dvs. start- och slutsekvenserna
- Hur paketet eller ramarna är formaterade, dvs. vilken information de innehåller, hur den är ordnad och hur den representeras

- Hur förstörd eller felaktigt formaterad information ska hanteras, dvs. upptäckas och eventuellt sändas om eller korrigeras
- Vad man gör om man tappar kontakt med den andra parten
- Hur man avslutar och kopplar ned förbindelsen

Inte alla protokoll innehåller allt detta, men alla innehåller en del. Några exempel kommer i ett senare avsnitt.

För att datakommunikation ska fungera måste protokoll finnas på alla använda nivåer. Och både avsändare och mottagare, liksom mellanliggande utrustning, måste kunna hantera alla protokoll upp till "sin" nivå. För avsändare och mottagare betyder det den högsta nivån. En sådan svit av sammanhängande protokoll brukar kallas en *protokollstack*.

Det finns många olika protokollstackar, men på senare tid har den s.k. *TCP/IP-stacken* (eller ibland bara *IP-stacken*) blivit en av de mest använda och kända. Det beror dels på att den används för internet, där den har sitt ursprung, men också på att den har kommit att användas i många andra sammanhang dessutom. Den beskrivs mer i detalj i de följande avsnitten.

KOMMUNIKATIONSPRINCIP

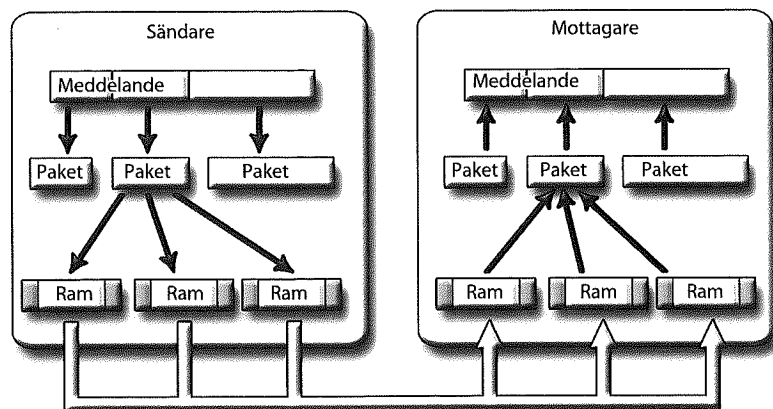
Den allmänna principen för all datakommunikation är att en applikation initierar dataöverföringen. Den initieras alltså i det högsta skiktet. Det kan röra sig om att sända e-post, ett chat-meddelande, en nedladdning av ett program, en låt eller en film, eller att hämta en sida på internet. Mer generellt säger man *meddelande* om det som ska överföras.

Det som händer under överföringen är i stora drag detta:

- Meddelandet delas upp i ett antal delar.
- Varje del läggs in i ett *paket* som består av en del data (den s.k. *nyttolasten*) och information för att man ska kunna återställa meddelandet (s.k. *pålägg*).
- Paketet skickas ett och ett till nästa nivå, där eventuellt en ny uppdelning och paketering görs, innan det skickas vidare nedåt.
- När paketet kommit ned till den nivån där det ska överföras till mottagaren, sker en sista uppdelning i *ramar* (ofta block av fix storlek) innan de skickas iväg.

- Om meddelandet passerar genom mellanliggande utrustning innan det når mottagaren, packar den upp varje paket eller ram precis så mycket som behövs för att den ska veta vart paketet ska skickas vidare. Mellanliggande utrustning bryr sig alltså inte alls om meddelandets innehåll, nyttolasten.
- Mottagande sidan gör precis tvärtom: den tar emot ramarna och bygger paket, som sedan skickas uppåt till närmast övre skikt.
- Det mottagande skiktet extraherar nyttolasten, kombinerar den till paket och skickar vidare uppåt.
- När paketen når den översta nivån, extraheras nyttolasten en sista gång och kombineras till det ursprungliga meddelandet.

Hela den här processen illustreras i figur 5.5.



FIGUR 5.5 Principerna för dataöverföring.

5.3 TCP/IP-stacken

En av de idag mest använda uppsättningar av protokoll för datakommunikation brukar gå under namnet TCP/IP-stacken. Det finns därför anledning att presentera den närmare. Den innehåller dessutom, i kraft av sitt stora genomslag, många begrepp och termer som förekommer i dagligt tal om datakommunikation.

Officiellt kallas TCP/IP-stacken för *Internet Protocol Suite*. Den bildar fundamentet för all kommunikation inom internet, men har också spritt sig till många andra nät. I de följande avsnitten presenteras kort dess allmänna struktur, dess kommunikationsmodell. Därefter går vi igenom de grundläggande dragen för protokollsvitens olika skikt.

TCP/IP-MODELLEN

När TCP/IP-stacken utvecklades var målet att möjliggöra kommunikation mellan datautrustning av olika slag. Man ville ha en enkel, robust modell för datakommunikation. En viktig aspekt av detta var att inte lägga ansvaret för saker som felkontroll, tillförlitlighet, tillstånd och återsändning på nätverket. I stället är det den utrustning som utnyttjar nätverket som har det ansvaret.

TCP/IP-stacken är, liksom andra protokollstackar, uppdelad i ett antal skikt som vart och har ett syfte och ändamål. Till skillnad från OSI-modellen, som presenteras på nästa sida, kan dock ändamålet med olika skikt i TCP/IP-stacken variera beroende på vilka protokoll som används. Exempelvis kan felkontroll placeras på olika nivåer.

Protokollsviten definierar fyra skikt som synes av tabell 5.1. Den definierar inget fysiskt skikt, som OSI-modellen, eftersom den är avsedd att vara oberoende av specifik hårdvara. Det har ändå tagits med här för att få en sammanhängande beskrivning av alla skikt.

TABELL 5.1 TCP/IP-stacken.

Skikt	Funktion	Beskrivning
4. Applikation	Process till process	Innehåller protokoll som gör det möjligt för program som är anslutna till nätet att kommunicera.
3. Transport	Dator till dator	Beskriver hur datorer kommunicerar med varandra för att föra över hela meddelanden. Inkluderar även t.ex. hur fel hanteras.
2. Internet	Över skilda nätverk	Definierar hur datakommunikation mellan skilda nätverk ska gå till.
1. Länk	Inom lokalt nätverk	Innehåller protokoll som gör att datorer inom ett och samma lokala nätverk kan kommunicera med varandra.

FAKTA

OSI-modellen

När datorer började kopplas ihop för cirka ett halvsekel sedan fanns det naturligtvis ingen standard för hur de skulle utväxla data. Tillverkare och organisationer, t.ex. myndigheter och universitet, uppfann sina egna metoder. Företag som IBM, Digital och Apple utvecklade alla sina egna lösningar, samtidigt som högskolor och universitet fortsatte att försöka hitta nya metoder. Det här förorsakade problem när datorer av olika fabrikat skulle kopplas ihop i s.k. heterogena nät. Två internationella standardiseringsorganisationer, ISO och ITU-T, initierade därför 1977 ett arbete för att skapa en internationell standard för datakommunikation. Resultatet blev OSI-modellen som är idag grundläggande för all datakommunikation.

OSI står för Open System Interconnection och är alltså en öppen standard för kommunikation mellan datorer. Det är i första hand en begreppslig modell – den beskriver ett sätt att tänka runt datakommunikation. Den kallas därför också OSI Reference Model. Den är praktisk att använda som referens när man t.ex. ska beskriva var ett visst kommunikationsprotokoll hör hemma.

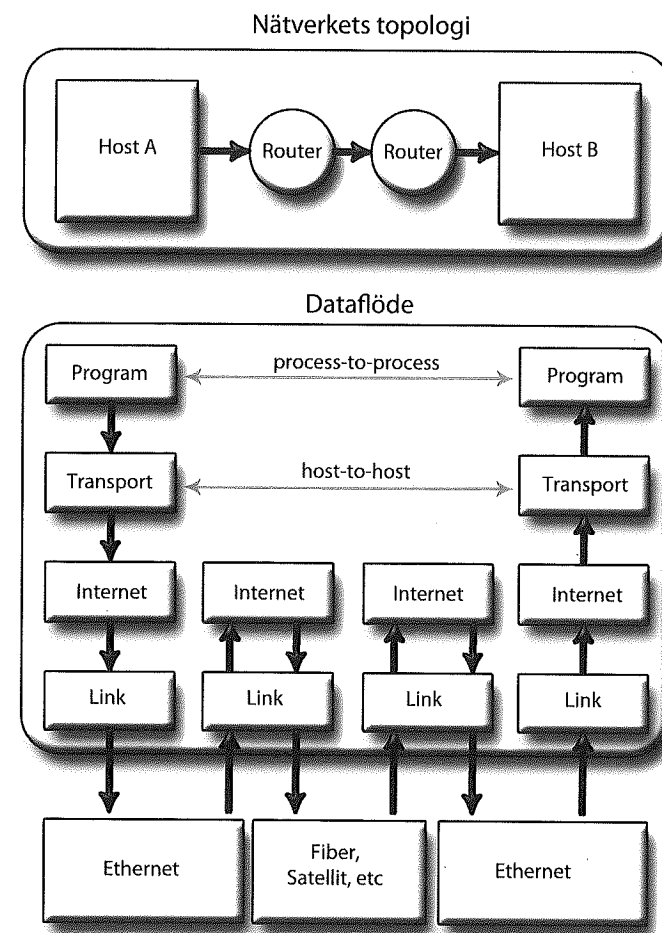
OSI-modellen organiserar datakommunikationen i ett antal skikt, där varje skikt har en klart avgränsad uppgift, löser uppgifter på en specifik nivå och har väldefinierade gränssnitt till skikten ovanför och under (i de fall det finns några). Den har sju skikt som tabellen här nedan visar.

Skikt	Dataenhet	Huvudsaklig eller typisk funktion
7. Applikation	Fil	Nätverksprocess till applikation
6. Presentation	Fil	Datarepresentation och kryptering
5. Session	Fil	Dator-till-dator-kommunikation
4. Transport	Segment	End-to-end-förbindelser och tillförlitlighet
3. Nätverk	Paket	Bestämning av transportväg och logisk adressering
2. Datalänk	Ram	Fysisk adressering
1. Fysiskt	Bit	Media, signal och binär överföring

När OSI-modellen var färdigutvecklad, hade redan TCP/IP-stacken börjat användas mer allmänt. OSI-modellen har därför inte fått någon större betydelse som implementationsmodell. Det är idag få protokollstackar som implementerar OSI-modellens alla sju skikt. Men som referensmodell står den sig än.

Kommunikation via TCP/IP-stacken mellan två datorer belägna i skilda nät fungerar i princip som i figur 5.6.

Ett program på avsändardatorn skickar sitt meddelande till transportskiktet som vidarebefordrar det till internetskiktet, där det delas upp i paket. Paketerna skickas vidare till länkskiktet där paketerna förpackas i ramar som skickas via det lokala nätverket till en router. Routern packar upp



FIGUR 5.6 Kommunikation via TCP/IP-stacken.

ramarna och paketen för att se vilket dator som är mottagare och skickar sedan ut ramarna på en förbindelse till mottagaren.

Det mottagande nätets router gör motsvarande analys av ramarna och paketen och skickar vidare ramarna till mottagande dator, som packar upp paketen och sätter ihop dem till det ursprungliga meddelandet.

De följande avsnitten ger en översikt över de olika skikten, från det lägsta till det högsta, med en beskrivning av vilka uppgifter som i typfallet löses och några exempel på protokoll på respektive nivå.

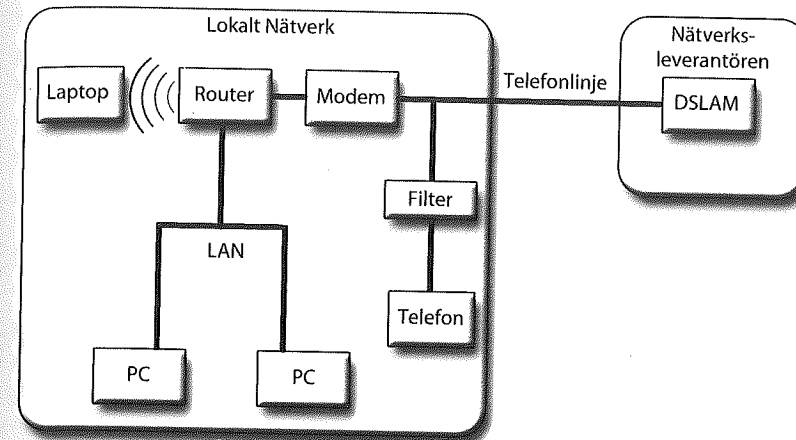
DET FYSISKA SKIKTET

I botten av varje kommunikationsmodell ligger ett fysiskt skikt. Det motsvarar ungefär vad som beskrevs i det inledande avsnitt vad som krävs för att den binära överföringen. Det fysiska skiktet specificerar alltså mekaniska, optiska och elektriska egenskaper.

Liksom för de övriga skikten, finns ett ansevärt antal protokoll för det här skiktet – vart och ett med sina unika egenskaper. RS-232, V.34, V.35, Ethernet och DSL är exempel på protokoll som en vanlig användare kan stöta på. De förklaras närmare i tabell 5.2.

TABELL 5.2 Exempel på protokoll för det fysiska skiktet.

Protokoll	Typ av standard	Kommentar
RS-232	Standard för seriell binär överföring	När det står att en dator, skrivare, eller annan utrustning har en seriell port är det nästan alltid RS-232 som avses.
V.34 och V.35	Standarder för modemöverföring	Förekommer ofta i t.ex. manualer som beskriver hur man tar sig ut på internet via ett modem.
Ethernet	Familj av standarder för överföring inom lokala nät	Standarden täcker både det fysiska skiktet och datalänkskiktet. Det finns för flera olika hastigheter, från 1 Mbit/s till 100 Gbit/s. Bredbandsnät använder sig ofta av Ethernet.
DSL, Digital Subscriber Line.	Familj av standarder för användning av det vanliga telefonnätet för dataöverföring samtidigt med vanlig telefoni	Tjänsten levereras oftast som ADSL, dvs. Asymmetric DSL, vilket innebär att överföring till datorn ("downloading") är väsentligt snabbare än från datorn ("uploading").



FIGUR 5.7 Användning av DSL.

En aspekt som kan vara värd att lägga märke till är att kontakterna utformas olika för olika standarder. Det här är tydligt när man jämför t.ex. en USB-kabel med en Ethernet-kabel eller en modem-kabel med en Firewire-kabel. Delvis beror skillnaderna på skilda behov, men det är också en säkerhetsåtgärd för att förhindra att man stoppar in fel kabel i ett uttag och skadar utrustningen, t.ex. en modemkabel i ett Ethernet-uttag.

I figur 5.7 illustreras hur en DSL-förbindelse kan sättas upp i ett hem eller ett litet kontor och hur man kombinerar DSL-modemet med en router för att skapa ett lokalt nät. Vid ingången till det lokala nätverket sitter en "splitter" som delar upp inkommande signaler i telefontrafik och datatrafik. Datatrafiken leds vidare till en router som vidarebefordrar data till den dator som anges i de inkommande paketen. Det kan både röra sig om fast uppkopplade datorer och tillfälliga uppkopplingar via ett trådlöst nätverk.

LÄNKSKIKTET

I länkskiktet skickas ramar direkt från en dator till en annan som befinner sig i samma lokala nät.

I det här skiktet finns inte många protokoll som man hör talas om som normal användare av en dator. Men som ett exempel på ett par protokoll på den här nivån kan man nämna PPP, ATM och Ethernet Data Link, se tabell 5.3.

TABELL 5.3 Exempel på standarder inom länkskiktet.

Protokoll	Typ av standard	Kommentar
ATM, Asynchronous Transfer Mode	Standard för att skicka digital information över telenätet	Meddelanden delas upp i celler av fix storlek (5 + 48 bytes) som skickas med konstant hastighet över telenätet.
Ethernet Data Link Protocol	Ethernet-standardens övre del, men länkskiktets lägre del	Meddelanden delas upp i ramar av variabel storlek som skickas över Ethernets fysiska skikt.
PPP, Point-to-Point Protocol	Standard för att etablera en direkt förbindelse mellan två datorer	PPP tillhör länkskiktets övre del och kan innehålla procedurer både för komprimering, kryptering och autentisering. PPP kapslas ofta in i Ethernet (PPPoE) eller ATM (PPPoA).

INTERNETSKIKTET

I TCP/IP-stackens internetskikt hittar man den andra delen av dess namn, IP, Internet Protocol. Det är alltså ett av de grundläggande protokollen i sviten. Egentligen handlar det dock om en familj av protokoll, där IP endast är ett av flera. Dessutom finns två versioner av protokollen, version 4 och 6, med förkortningarna v4 och v6. Version 4 växte fram under 1970-talet och är den första stabila versionen av protokollet. Den fastställdes 1978 och har alltsedan dess varit standardprotokollet för internet. Standarden för version 6 publicerades 1998, men fortfarande är v4 den dominerande standarden. I kapitel 25 som handlar om internet tas ytterligare aspekter upp på TCP/IP-stacken. Tabell 5.4 sammanfattar de olika protokollen i internetskiktet.

TABELL 5.4 Protokoll på internetsnivå i TCP/IP-stacken.

Protokoll	Typ av standard	Kommentar
IP, Internet Protocol	Kommunikationsprotokoll för att skicka paket (datagram) mellan två datorer anslutna till samma eller olika nätverk	Det grundläggande protokollet för överföring av data paketvis över internet. Finns i två versioner, IPv4 resp. IPv6.
ICMP, Internet Control Message Protocol	Protokoll för att skicka förfrågningar och felmeddelanden över internet	Används normalt enbart av nätverksfunktioner i operativsystem för att undersöka status i nätet. Finns i två versioner, ICMP resp. ICMPv6.

Protokoll	Typ av standard	Kommentar
IGMP, Internet Group Management Protocol	Protokoll för rundsändning (<i>broadcasting</i>) av data	Används för att utnyttja resurserna effektivt vid strömmande utsändning av t.ex. video eller för spel över nätet.
IPsec	Protokoll för säker överföring av data	Ser till att varje IP-paket krypteras och bekräftas genom autentisering. Används också för att initiera en förbindelse med handskakning, utbyte av lösenord, etc.

TRANSPORTSKIKTET

TCP/IP-stackens första förkortning hänför sig till transportskiktet; den står för Transmission Control Protocol. Men – liksom för internetskiktet – finns det för att täcka olika behov fler protokoll i transportskiktet.

Huvuduppgiften för transportskiktet är att erbjuda det översta skiktet, applikationsskiktet, en enkel kommunikationsmodell för programmen på de sändande och mottagande datorerna. Om tillämpningen är e-post, ska programmet kunna sända och ta emot hela meddelanden. Om det är ett chat-program så ska användaren ges illusionen att datorerna är direkt hopkopplade. Om det är ett program som tar emot radio eller musik, ska det verka som data strömmar utan avbrott in i programmet.

Protokollen i transportskiktet kan utformas på olika sätt. Typiska egenskaper som karaktäriserar (och skiljer) olika protokoll i det här skiktet är:

- **Uppkopplingsprincip.** En förbindelse kan vara uppkopplingsorienterad eller uppkopplingslös. En uppkopplingsorienterad förbindelse etableras innan avsändaren börjar skicka data. Avsändaren har alltså identifierat mottagaren och vet att den är beredd på att ta emot data. Vid uppkopplingslös överföring skickas data utan att en session upprättas.
- **Leveransordning.** Det normala på denna nivå är att paketen kan levereras i vilken ordning som helst och att det är mottagarens uppgift att lagra dem och sortera dem innan de levereras till applikationsskiktet. Därför numreras paketen.
- **Pålitlighet.** Protokollet avgör hur säker sändare och mottagare kan vara på att sända och mottaget meddelande stämmer överens. Här kan sändaren lägga till data som möjliggör felupptäckt eller felkorrigering, och mottagaren kan kvittera korrekt mottagna paket. Mottagaren kan

begära omsändning av felaktiga paket och sändaren kan skicka om paket som inte kvitteras. Man kan också välja att sända och ta emot utan någon större kontroll.

- *Byte-orientering.* Många tillämpningar förväntar sig data som en sekvens av bytes (oktetter). Protokoll omvandlar i allmänhet paket och segment till strömmar av bytes innan de levereras till applikationsskiktet.
- *Flödeskontroll.* Ett protokoll kan förses med funktioner så att mottagaren (eller annan utrustning mellan sändare och mottagare) kan varna om att inkommande flöde av data är större än vad som kan hanteras. Sändaren kan då tillfälligt stoppa sändningen tills det kommer ett klartecken från den som stoppade sändningen.
- *Processadressering.* En IP-adress anger enbart vilken dator som ett meddelande sänds till, men där kan ju flera program vara igång samtidigt (e-postläsare, webbläsare, chat, etc.). För att identifiera den mottagande processen, kompletteras paket med ett portnummer som är kopplat till det mottagande programmet.

De två vanligaste protokollen på den här nivån är TCP och UDP, som kort beskrivs i tabell 5.5, tillsammans med RTP, ett nyare protokoll för strömmande data.

TABELL 5.5 De vanligaste protokollen i transportskiktet.

Protokoll	Typ av standard	Kommentar
TCP, Transmission Control Protocol	Protokoll för uppkopplingsorienterad förbindelse mellan två datorer i ett nätverk	Det vanligaste protokollet på internet i detta skikt. Är pålitligt och används därför i sådana situationer där det är viktigt att data kommer fram oförändrade.
UDP, User Datagram Protocol	Protokoll för uppkopplingslös förbindelse mellan två datorer i ett nätverk	Kontrollerar inte att paket kommer fram eller om överföringsfel uppträder, men är effektivt i situationer där snabbhet är viktigare än korrekthet, t.ex. vid spel över internet.
RTP, Real-time Transport Protocol	Protokoll för strömmande data	Standard för att leverera audio och video över IP. Används för strömmande musik och video, liksom vid IP-telefoni och videokonferenser.

APPLIKATIONSSKIKTET

Det översta skiktet i TCP/IP-stacken är – liksom i OSI-modellen – applikationsskiktet. Det är alltså här det hela börjar och slutar. Eftersom det finns många olika slags tillämpningar som kräver anslutning till ett lokalt eller mer omfattande nät, så är det också naturligt att det finns många protokoll. Det är dock inte så att varje tillämpning har sitt eget protokoll. Det är i stället de krav som tillämpningen ställer på nätkommunikationen som avgör vilket protokoll som används.

Några av de vanligaste protokollen på denna nivå återfinns i tabell 5.6. De här är protokoll som kan ses av en normal användare, t.ex. när man surfar på internet eller skickar (eller tar emot) e-post. Vid sidan om dessa protokoll finns en lång rad protokoll som används för att administrera nät eller som enbart är åtkomliga direkt från ett terminalfönster.

TABELL 5.6 Några exempel på protokoll i applikationsskiktet.

Protokoll	Typ av standard	Kommentar
HTTP, Hypertext Transfer Protocol	Protokoll för överföring av webbsidor	Används för att föra över html-sidor från webbserverar till webbklienter. Klienten skickar en begäran till en server om att servern ska föra över en viss sida i html-format till mottagaren, dvs. klienten.
HTTPS, Hypertext Transfer Protocol Secure	Säker variant av HTTP-protokollet	Används oftast för ekonomiska transaktioner över nätet. Kommunikationen krypteras för att inga obehöriga ska kunna tjuvlyssna på den.
SMTP, Simple Mail Transfer Protocol	E-postprotokoll	Vanligt protokoll för e-post. Det är textbaserat, men använder sig bara av sju bitar till ett tecken vilket gör att t.ex. våra nationella tecken (å, ä, ö) måste de ges speciell kodning.
POP3, Post Office Protocol, Version 3	E-postprotokoll	Vanligt protokoll för e-post. Liksom SMTP textbaserat med enkla kommandon i klartext för att hantera e-posten.
IMAP4, Internet Message Access Protocol, Version 4	E-postprotokoll	Efterföljare till POP3. Det mest kapabla protokollet. Klarar av att hantera mappar, flaggor och sökning. Delar av meddelanden kan hämtas. Ny e-post meddelas.