Sammanfattning Datorkommunikation

Philip Larsson

2013-10-22

Innehåll

1 Allmänt

1.1 OSI-modellen

OSI-modellen används som en referensmodell idag.

7	Applikation
6	Presentation
5	Session
4	Transport
3	Nät
2	Länk
1	Fysisk

1.2 TCP-IP modellen

Eftersom protokollen för Internet utvecklades oberoende av arbetet med OSI-modellen så innehåller de inte samma skikt.
Skikt 5, 6 och 7 är i TCP-IP modellen endast ett skikt: applikationsskiktet. TCP/IP modellen definierar inte heller något länkskikt eller fysiskt skikt.

Applikation
Transport
Nät

1.3 Router, switch, hub

- Hubb
  Jobbar på fysiska lagret. Det den får in på en port skickar den ut till alla andra portar.

- Switch
  Jobbar på länk lagret. Fungerar inledningsvis som en hubb, men lär sig vilken dator som sitter på respektive port. Den sparar detta i en addresstabell.

- Router
  Jobbar på nätlagret. Som switch men har även en routingtabell över närmaste väg. Används när man ska koppla ihop nät. Tillhandahåller även andra funktioner som t.ex. brandvägg.

2 Skikt i OSI-modellen

1. **Fysiska skiktet**
   Ansvarigt för att skicka bitar mellan två noder som är kopplade via en fysisk länk. Arbetar med bitar.

2. **Länk skiktet**
   Ansvarigt för att överföra ramar från en nod till nästa över en länk. Arbetar med ramar.

3. **Nätskiktet**
   Ansvarigt för att skicka paket mellan en sändar-värd och en mottagar-värd (som kan vara kopplade på olika nät). Arbetar med paket.

4. **Transportskiktet**
   Ansvarigt för att skicka data mellan två applikationsprocesser. Använder portnummer för att separera applikationer. Arbetar med segment.

5. **Sessionsskiktet**
   Ansvarigt för styrning och synkronisering av dialogen mellan sändar och mottagar process. Arbetar med data.

6. **Presentationsskiktet**
   Ansvarigt för översättning, komprimering och kryptering av applikationsdata. Arbetar med data.

7. **Applikationsskiktet / tillämplingsskiktet**
   Avsvarigt för att tillhandahålla användartjänster (t.ex. e-post och filöverföring). Det enda skiktet användaren behöver se. De andra 6 lagren inklusive tillhörande protokoll har till uppgift att betjäna protokollen inom applikationsskiktet. Arbetar med data.

3 Begrepp inom det fysiska lagret

Data: information vi vill överföra.
Signal: så som data är representerat när det skickas över en länk.

3.1 Bandbredd

- Analog definition: frekvensbandet på kanalen (mäts i Hz).
- Digital definition: antalet bitar per sekund som kanalen kan överföra (bps). Kallas även för bit-rate.

3.2 Störningar

När en signal färdas över en länk kommer den att försämras p.g.a.. störningar. Olika typer av störningar:

- Dämpning: signalenergin minskar.
- Distortion: signalen ändrar sin form.
- Brus: slumpmässiga störningar som kan bero på brus alstrade i elektroniska kretsar, eller brus från atmosfären.

3.3 Prestanda

- Throughput / genomströmmning:
  Ett systems verkliga kapacitet. Mäts i bps.
- Fördröjning (latency):
  Hur lång tid det tar för ett system att hantera ett meddelande eller paket mellan sändare och mottagare.
  Är en summa av utbredningstid, transmissionstid, kötid och betjäningstid.

3.4 Digital kommunikation

Handlar om att skicka data i form av bitar (ettor och nollor) över en fysisk länk. Bitarna kan antingen representeras av digitala signaler (digital transmission) eller av analoga signaler (analog transmission).

3.4.1 Digital transmission

Ettor och nollor representeras av olika spänningsnivåer eller energinivåer. Processen att konvertera digital data till digitala signaler kallas för linjekodning. Tre metoder för linjekodning:

- **Non-return-to-zero (NRZ)**
  En etta motsvaras av en amplitudnivå och en nolla motsvaras av en annan amplitudnivå. Brukar vara så att en nolla representeras av en positiv spänningsnivå och en etta en negativ spänningsnivå.
  Kan bli problem med synkroniseringen mellan sändare och mottagare när flera ettor och nollor kommer efter varandra.

- **Manchester**
  En bit representeras av en spänningsändring, så en nolla representeras av en positiv spänningsnivå till negativ och en etta representeras av en negativ spänningsnivå till positiv.[1]

- **Differential Manchester**
  Om signalen följer förra bitens mönster så är det en nolla. Om mönstret bryts så blir det en etta. Första tecknet (biten) vet man oftast inte och skriver ett frågetecken under den.

3.4.2 Analog transmission

Vid analog transmission så representeras ettorna och nollorna av förändringar av en sinusvåg. Sinusvågen karaktäriseras av amplitud, fas och frekvens och genom att variera dessa storheter kan man skapa olika sinusvågor som kan representera etta och nolla. Vi har tre olika moduleringstekniker:

- **Amplitudmodelering** (amplitude shift keying, ASK)
  Amplituden ändras så att en etta representeras av en annan amplitud än en nolla. Är dock känslig för störningar.

- **Frekvensmodelering** (frequency shift keying, FSK)
  Ettor och nollor representeras av olika frekvenser. Mindre känslig för störningar än ASK.

- **Fasmodulering** (phase shift keying, PSK)
  Ettor och nollor representeras av olika fas.

4 Begrepp inom länkskiktet

Länkprotokoll tillhandahåller funktioner för att hantera en länkförbindelse, feldetektering, felhantering och flödeskontroll för data som skickas öven en förbindelse. I länkprotokollen brukar termen ram (eng. frame) användas istället för paket. Anledningen till detta är att paketen ”ramas is” av flaggor. Flaggorna underlättar när mottagaren ska detektera vart ramen börjar och slutar.

[1] Tänk att det är ”bra” att gå från negativ till positivt, så då en 1:a.

4.1 Bitstuffing

Om en flagga består av bitmönstret 0111 1110 måste vi se till att samma mönster inte finns i ramen. Detta löser man med bitstuffing (bitutfyllnad). Om det kommer fem ettor efter varandra så stoppar man in en nolla efter femte ettan. Sekvensen 0111 1110 0 blir alltså 0111 1101 00. Mottagaren tar bort den nolla som följer fem ettor i rad för att få fram det ursprungliga meddelandet (alltså ramen).

4.2 Feldetektering

Ett bitfel är när mottagaren tolkar en etta som en nolla eller tvärt om. Detta måste detekteras och det gör man genom att lägga på extra bitar som beräknas fram genom datan. Tre feldetekteringsmetoder:

Paritetsbit (Simple Parity-Check Code)

Om jämn paritet används så lägger sändaren till en etta om det finns ett ojämt antal ettor i paketet, annars lägger sändaren till en nolla. Så om data innehåller 5 stycken 1:or så blir paritetsbiten 1.

Cyklisk redundanscheck (CRC)

En annan feldetekteringsmedtod som är bättre än paritetsbit. Går till så här:

- Först och främst har mottagare och sändare kommit överens om ett *generatorpolynom* ($C(x)$) som används för att verifiera meddelandet.
- Sen skriver vi bitmönstret som ett polynom. T.ex. så blir 0011 1010 $x^5 + x^4 + x^3 + x^1$ och detta kallas $M(x)$.
- Sedan multiplicerar vi $M(x)$ med x^k där k är högsta exponenten i generatorpolynomet. Så om $C(x)$ är x^3+x^2+1 så är $k = 3$. $M(x)\times x^k = B(x)$
- Därefter modulo 2 delar vi $B(x)$ med $C(x)$ och får ett svar $f(x)$ samt en rest $R(x)$. $R(x)$ adderas till $B(x)$ och då får vi meddelandet som ska skickas, $P(x)$.
  $R(x)$ skrivs i lika många bitar som k är. T.ex. om $R(x) = x^2 + 1$ så skrivs det som 101.

Mottagaren delar sedan det mottagna meddelandet med generatorpolynomet $C(x)$ och om ingen rest ges så är paketet helt korrekt (resten skrivs som $E(x)$).

Kontrollsumma (checksum)

Går till så här:
Adderar alla bitar med varandra, oftast 8 bitar + 8 bitar. Om nån bit ”skjuts över” (carry bit) så adderas den också. Meddelandet är alltså komplementet (till den summa man får) tillsammans med de ursprungliga bitarna. Komplementet får man genom att byta ut alla nollar mot ett och vice versa.
Exempel:
Vi har: 10101001 10111001
Det som skickas iväg är ursprungliga meddelandet samt checksumman, alltså: 10101001 10111001 10011100.

Det mottagaren gör är att addera ihop allt och om summan blir bara ettor så gick paketet fram utan bitfel.
Exempel:
Tar emot 10101001 10111001 10011100

4.3 Felhantering

2 olika sätt. Antingen be om ett nytt paket eller försöka rätta de fel som detekterades. Om man ska kunna be om ett nytt paket måste varje paket kunna identifieras. Det gör de med sekvensnummer.
I en omsändningsalgoritm är det vanligt att alla paket som kommer fram korrekt bekräftas (**ACK**:as). Mottagaren talar om vilket sekvensnummer (paket) den väntar på.
Man kan också tänka sig en negativ bekräftelse, d.v.s. att man **NAK**:ar de sekvensnummer vars paket man inte har tagit emot korrekt.

4.3.1 Automatic-repeat-request (ARQ) algoritmer

- **Stop-and-wait ARQ**
  När ett paket har skickats så måste sändaren få en bekräftelse för det paketet innan den kan skicka nästa paket. Tekniken gör att det tar lång tid att överföra en större mängd data.

- **Go-back-N ARQ**
  Kan skicka flera paket åt gången. Sändaren har ett sändfönster.
  S_F = Sliding window first
  S = Paket som senast skickades

S_L = Sliding window last
Sändarfönstret måste vara mindre än sekvensnumret.

- **Selective repeat ARQ**
  Bekräftar paket den tar emot men kan även skicka ett NAK om något paket gick förlorat.

4.4 Point-to-Point Protocol (PPP)

Ett länkprotokoll.
Kapslar in IP-paket, etablera, testa och konfigurera länk via LCP eller förbereda för nätprotokoll via NCP.

4.5 Multiplexering

Multiplexering används när fysiska länkar behöver delas av flera förbindelser. Alltså att en fysisk kanal delas upp i flera kanaler.

Duplex

- Simplex: Kommunikation i en riktning. T.ex. TV-sändningar.
- Halv-duplex: Kommunikation i båda riktningar men bara en i taget. T.ex. walkie talkie.
- Full-duplex: Kommunkiation i båda riktningar samtidigt. T.ex. telefoni.

4.5.1 Metoder för multiplexering

- **Frekvensmultiplexering (FDM)**
  Länkens bandbredd delas upp i flera frekvensband. Varje kanal får sedan sitt egna frekvensband.
- **Tidsmultiplexering**
  Finns *Synkron* tidsmultiplexering och *statisk* multiplexering. Båda går till så att länken delas upp i ramar som innehåller tidsluckor, och varje kanal får en egen tidslucka. Varje ram brukar börja med en synkroneringsbit som alternerar mellan 0 och 1.
  Skillnaden mellan *synkron* och *statisk* tidsmultiplexering är att om en kanal inte har något att skicka i *synkron* tidsmultiplexering så kommer tidsluckan att vara tom.

I *statisk* tidsmultiplexering får kanalerna endast tillgång till tidsluckan när den har något att sända. Om kanalen inte har något att sända så får de andra kanalerna mer kapacitet.

4.6 Medium Access Control (MAC) Protokoll

Alla värdar på en länk/nät måste ha samma regler för hur och när de ska få sända data på länken. Dessa regler kallas för accesmetoder (medium access control).

4.6.1 Accesmetoder

Kan delas upp i två grupper, controlled access och random access:

- **Controlled access**
  Terminalerna kommer överens om vem som får skicka och när. En terminal måste ha tillstånd att sända data. Ex. polling, token ring.

- **Random Access**
  Alla terminaler sköter sig själva och bestämmer själv när de får skicka. Kollisioner kommer förmodligen att ske. Ex. Aloha, CSMA/CD, CSMA/CA.

Och de är:

- **Polling**
  Har en ”master” dator och en eller flera ”slav” datorer. All data som ska skickas måste gå via mastern, och får endast skickas på begäran av mastern.

- **Token ring**
  Datorerna är kopplade i en ring och använder sig av en token (typ stafettpinne) som bestämmer när man får sända. Fördelar med denna metod är att det inte sker några kollisioner och att all kapacitet används och delas lika. Nackdelar är att systemet kraschar om token försvinner eller om en dator kopplas bort.

- **Aloha**
  Om två terminaler vill skicka måste det gå via centraldatorn, men skickar så fort de har något att skicka (till skillnad mot polling som väntar på tillstånd att få sända). Kollisioner kommer att uppstå.

- **CSMA / CD**
  Carrier Sense Multiple Access / Collision Detection. Om någon dator vill skicka på länken så gör den det. Den lyssnar och om länken är ledig skickar den. Om datorn märker att en kollision har inträffat (om t.ex. två datorer skickade samtidigt) så väntar datorn en slumpmässig tid och försöker sedan igen.

- **CSMA / CA**
  Carrier Sense Multiple Access / Collision Avoidance. Som CSMA/CD men används där det är svårt att veta om en kollision har ägt rum. Försöker undvika kollisioner helt genom att t.ex. först skicka en RTS (request to send) till accesspunkten och sedan vänta på en CTS (clear to send) innan den börjar skicka data.

4.7 IEEE standardiseringsprojekt 802.

Varje dator som kopplas in till ett 802.x nät måste ha en unik address som är inbränd i nätverkskortet. Kallas MAC-address (ibland också fysisk adress) och är 6 byte lång.

- **802.3x**
  Ethernet en samling standardiserade metoder för trådade LAN

- **802.11x**
  Wlan standarder

IEEE 802.11 använder accessmetoden CSMA/CA. När en station detekterat att mediet är ledigt, väntar den en viss tid, kallad *distributed interface space* (**DIFS**). Om mediet fortfarande är ledigt efter DIFS skickar stationen en Request To Send (RTS) till mottagarstationen. Mottagaren av RTS väntar en viss tid, kallat *short interface space* (**SIFS**) och skickar sedan en Clear To Send (CTS) till sändarstationen. Efter att sändarstationen har tagit emot ett CTS väntar den tiden SIFS och skickar sedan sin dataram. När mottagarstationen har tagit emot sin dataram väntar det tiden SIFS och skickar sedan ett ACK.

4.8 Nättopologier

- Buss
- Ring
- Stjärna

4.9 Unicast och broadcast

Unicast: en sändare och en mottagare.
Broadcast: en sändare och flera mottagare.

4.10 Spread Spectrum

Teknik för trådlösa länkar med mycket störningar.

- **Frequency Hopping Spread Spectrum (FHSS)**
  Hoppar mellan frekvenser och skickar lite på varje frekvens.
- **Direct Sequence Spread Spectrum (DSSS)**
  En bit kodas med flera bitar som är bestämda av sändare och mottagare.

5 Begrepp inom nätskiktet

Nätadressen behövs för att vi ska veta vart nånstans datorn finns. Består av *nät-id* och *värd-id*. Nät-id identifierar de nät som enheten är kopplad till. Värd-id identifierar enheten själv.
Router används för att skicka data mellan näten. Den gör beslut om bästa väg för paketen (routing, se nedan). Routing-beslut baseras på nät-identiteten inte värd-identiteten.
Forwarding är att skicka vidare paket beroende på nätadressen, och det är det routern gör.
Routing är den process som routern gör när denna väljer vilken väg som paketen i nätverket ska ta.

5.1 IP - ett nätprotokoll

Enda nätverksprotokollet som får användas på Internet. Finns två versioner som används idag, IPv4 och IPv6.
IP har ingen felhantering eller flödeskontroll[2], utan är ett så kallat *best effort protokoll*[3] Varje värd måste ha en unik address, en så kallad IP-adress.

5.1.1 IPv4

IP-adresserna delas upp i klassindelad adressering och klasslös adressering.

[2]ICMP hjälper med detta
[3]Best effort protokoll: ger inga garantier för om paketet kommer fram i rätt ordning, utan fel eller över huvud taget.

Klassindelad addressering (classful addressing)

IP-addresserna delas in i klasser vars nät-id och värd-id har olika storlek beroende på hur stort nätet är.

Klasslös addressering (classless addressing)

Man separerar nät-id och värd-id med hjälp av en mask. Masken består av 32 bitar och en etta indikerar att addressbiten på motsvarande plats ingår i nätidentiteten.

Exempel:
mask: 255.255.192.0
IP-adress: 130.235.188.247
en etta indikerar att addressbiten ingår i nät-id. Alltså är vårt nät-id = 130.235.128.0 och värd-id är då resten = 0.0.60.247.

Man kan också skriva masken efter IP-addressen separerat med ett snedstreck "/". Exempelvis 130.235.188.247/18 anger att de 18 första bitarna i IP-adressen är nät-id.

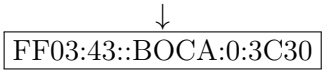
5.1.2 IPv6

Fördelar med IPv6 gentemot IPv4:

- Fler addresser (128 bitars addresser)
- Konstant headerlängd förenklar routingprocessen
- Stöd för realtidsdata
- Stöd för datasäkerhet såsom autentisering och kryptering

IPv6 adresser består av 128 bitar (16 byte) och kan förkortas.
Man kan ta bort nollor som inte ändrar värdet (d.v.s. 0043 är samma sak som 43) och förkorta följder av nollor och ersätta dem med dubbla kolon. Men man kan bara förkorta nollor till kolon en gång, annars går det inte att återskapa den ursprungliga adressen.
Exempel:

FF03:0043:0000:0000:0000:B0CA:0000:3C30
↓
FF03:43:0:0:0:B0CA:0:3C30



5.2 Address Resolution Protocol

Jobbar på lager 2-3.
Används för att mappa en IP-address till en MAC-address. Varje värd har en ARP-tabell där den lagrar MAC-IP adresser.
Det går till så här:

- Dator1: *ARP Request* till *broadcast* ”Vem har IP x.x.x.x.?”
  Skickar med sin IP-address och MAC-address i ARP Requesten.
- Dator2: *ARP Reply* svarar dator1 ”Jag har IP x.x.x.x och min MAC-address är z.z.z.z.z.z ”

Dator1 och dator2 sparar varandras MAC-IP adresser i tabellen så de inte behöver fråga nästa gång de ska kommunicera.

5.3 Routing Algoritmer

5.3.1 Distance Vector

Varje router i nätet har en egen bild över hur nätet ser ut.
Routrarna skickar sin routingtabell till sina grannar, och detta gör den med jämna intervall eller vid en förändring.
Enkelt → låga krav på CPU och minne.

5.3.2 Link State

Skickar info om vilka länkar routern ansluter till och vilka de närmaste grannarna är. Alla routrar har samma karta över nätet och får på egen hand räkna ut bästa väg till olika destinationer. Skickar info om förändringar har skett, men inte annars.
Mer avancerat → högre krav på CPU och minne.

5.4 ICMP - ett nätverksprotokoll

Ett hjälpprotokoll för IP som ska hjälpa till med felmeddelanden och förfrågningar.
ICMP meddelandet packas in i ett IP paket.

6 Begrepp inom transportskiktet

6.1 TCP - ett transportprotokoll (Transmission control protocol)

Används när en applikation vill ha en kontrollerad och tillförlitlig dataöverföring. Skapar en överföringssession.

6.2 UDP - ett transportprotokoll (User Datagram Protocol)

Är precis som IP ett *Best effort* protokoll. Lämpar sig för applikationer där hastigheten är viktigare än att något paket försvinner eller kommer i fel ordning, t.ex. i spel. [4]

7 Begrepp inom applikationsskiktet

Två grundläggande användarmodeller, Client Server och Peer to Peer.

Client Server

”Traditionell” metod, där alla klienter hämtar data från en server.

Peer to Peer

Användare fungerar både som en klient och en server, och data skickas mellan klienterna. [5]

7.1 WWW Applikation

Baseras på klient server modellen.
Består av tre delar:

- **Webbsidor**
  - Statiska: HTML
  - Dynamiska: PHP m.fl.
- **URL**
  En standard för att namnge webbsidor.

[4] Notering: UDP tillför egentligen inte med så många funktioner till transportskiktet.
[5] Ibland används en hybrid mellan båda användarmoddellerna.

- **HTTP**
  Applikationsprotokoll som används för att hämta webbsidor från en server.
  Använder en TCP-förbindelse.

7.2 Domain Name System (DNS)

Sköter mappning från symboliska namn till IP-addresser[6]. En värd vet alltid IP-addressen till närmaste DNS Server.

8 Telenäten

Accessnäten är analoga. Från telefon till lokalstation överförs talet analogt i frekvensområdet $0 - 4$ kHz.
Kärnnäten är digitala. *PCM* sker i lokalstationerna.

8.1 Pulse Code Modulation (PCM)

Utvecklades för telekombranschen för att digitalisera telekomnätet.
Sker i tre steg:

1. **Sampling**
   Man mäter signalen vid vissa tidpunkter. Om högsta frekvensen är n Hz, så måste man sampla med $2n$ Hz (*Nyqvist*)

2. **Kvantisering**
   Avrundar mätvärdena från samplingen till ett begränsat antal amplitudnivåer. Många mätvärden behöver förmodligen avrundas och leder därmed till ett *kvantiseringsfel*.

3. **Kodning**
   Alla avrundade mätvärden kodas till binära tal.
   Nu är den analoga signalen en följd av binära tal som kan lagras i datorn.

8.2 Mobilnät

Nätet är geografiskt indelat i celler. I varje cell finns det en basstation (stor antenn). Varje cell får ett visst frekvensband. Frekvensbandet återanvänds i

[6]Metafor: telefonbok. Den mappar något som är lätt för människor att komma ihåg (namn) med ett nummer (telefonnummer).

flera celler.

Flera mobilterminaler måste dela på samma frekvensband i en cell. Då behövs metoder för multipel access:

- **Frequency Division Multiple Access (FDMA)**
  Det tillgängliga frekvensbandet delas upp i ett stort antal radiokanaler, var och en med sin bärfrekvens.
- **Time Division Multiple Access (TDMA)**
  Kanalerna turas om att skicka i olika tidsluckor.
- **Code Division Multiple Access (CDMA)**
  Många kanaler skickar på samma frekvens, men varje kanal har en unik kod så att det går att skilja dem åt.