Indholdsfortegnelse

Forord

Denne notetsamling er udviklet til brug på alle danske uddannelser, hvor der er brug for en grundlæggende viden om datakommunikation. Bogen udgives on-line som pdf fil, og kan frit benyttes af alle som de ønsker det, delvis eller i uddrag. Ved kopiering skal der dog mindst kopieres en fuld side med header og footer, og der må ikke fjernes noget fra siderne.

Kapitel 1 starter med en kortfattet gennemgang af de nødvendige grundbegreber.

Kapitel 2 – 6 gennemgår lagende i TCP/IP protokolstakken med et lag for hver kapitel, som om det var en model.

Til sidst gennemgår kapitel 7 netværksrelaterede emner som firewall og NAT, der ikke umiddelbart kan placeres inde i protokolstakken.

Undervejs vil eksempler, øvelser og opgaver basere sig på en traditionel Linux i Command Line Interfacet (CLI). Hvor der konfigureres for at vise praktiske eksempler, baserer disse sig på en Debian distribution, der bl.a. også benyttes af Ubuntu, men ikke af Fedora. Dette er valgt for at holde kompleksiteten så lav som mulig, og rutinere eleverne i at benytte et CLI.

Årsagen til at vælge Linux er tingende er standardiserede og kun ændrer sig langsomt eller slet ikke. Årsagen til at vælge Debian er den simple at forfatteren arbejder en del med den. Der findes mange glimrende operativsystemer og distributioner at arbejde med, men dette er altså forfatterens valg.

Årsagen til at vælge at gennemgå TCP/IP som en protokolstak er, den er massivt udbredt, hvilket man ikke kan sige om OSI modellen. Kan du lide det kan du bruge denne bog, kan du ikke lide det kan du enten skrive en bog selv, nappe hvad du kan bruge fra denne notesamling, og supplere med andet, eller finde en anden bog, valget er dit.

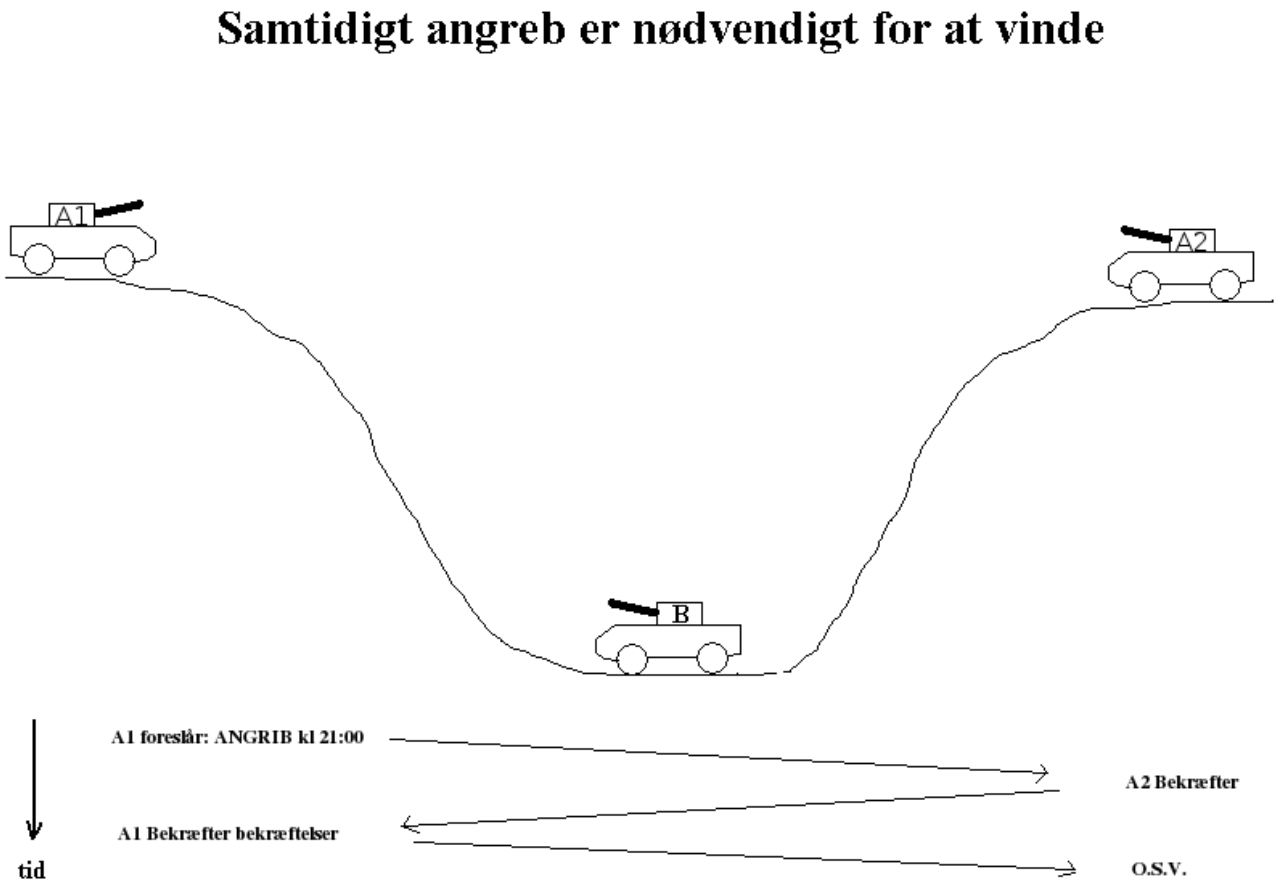
Forudsætninger

Det forventes eleven har:

- Kendskab til talsystemer.
- Kendskab til programmering.
- Brugerkendskab til CLI i operativsystemet Linux.
- Kendskab til fysik på folkeskole niveau.

Henrik Kressner
kressner@synkro.dk

1. Grundbegreber



Figur 1.1

Figur 1.1 viser problemstillingen ved kommunikation over et ikke 100% sikkert medie. Figuren viser to hære A og B. Hær B er Lidt stærkere en hver af A's hære, men A's samlede hære er meget stærkere end B's. Hvis A kan angribe B fra to sider samtidig kan A vinde, ellers vil A tabe. Hvordan kan Generalen hos A sikre sig at hans hære angriber samtidig, hvis han kun kan bruge fodordonnanser?

Han starter med at sende et signal fra den ene side til den anden, hvor han beordrer et angreb i morgen kl 07:00. For at sikre sig at hans kollega på den anden side har modtaget ordren, skal hans kollega sende en bekræftelse retur.

Vi kan konstatere: Uanset hvor mange gange der sendes en ordonnans frem og tilbage, så vil en af A's to generaler altid være i tvivl om hans kollega har modtaget signalet, derfor bør han ikke angribe. Det kan i øvrigt eftervises matematisk, at antallet af gange der sendes frem og tilbage, ikke påvirker sandsynligheden for en sikker overførsel af signalet.

Vi har altså det grundliggende problem, at vi aldrig kan være 100% sikker på en information kommer frem, men vi kan gøre en masse for at maksimere sandsynligheden for data kommer rigtigt frem.

1.1 Bits og byte

Den mindste dataenhed vi kan have i en digital computer er en bit. En bit kan have to tilstande, nul eller et, også kaldet sand eller falsk, hvor:

Falsk = 0 = Lav
Sand = 1 = Høj

Med en bit har vi 2 kombinationsmuligheder, med 2 bit har vi 4 kombinationsmuligheder, med 3 bit har vi 8 kombinationsmuligheder, med 4 bit har vi 16 kombinationsmuligheder o.s.v. Hver gang vi tilføjer en bit, fordobler vi antallet af kombinationsmuligheder.

Siden 1970'erne, hvor 8 bit processoren fremkom, har det været almindeligt antaget, at når man samler 8 bit, kaldes det en byte.

For at beskrive de store antal bit en moderne computer arbejder med, gør man brug af præfikser for hvor gang der er tre nuller bag et tal.

1 Kilobit = 1.024 bit <=> 1 Kilobyte = 1024 Byte

Normalt smider man de skæve tal væk og siger 1.000 bit = 1 kilobit.

1 Megabit	= 1.000.000 Bit	<=>	1 Megabyte = 1.000.000 Byte
1 Gigabit	= 1.000.000.000 bit	<=>	1 Gigabyte = 1.000.000.000 Byte
1 Terabit	= 1.000.000.000.000 bit	<=>	1 Terabyte = 1.000.000.000.000 Byte
1 Pentabit	= 1.000.000.000.000.000 bit	<=>	1 Pentabyte = 1.000.000.000.000.000 Byte

Vi regner normalt datatransmissions hastigheder i bit per sekund, som forkertes til bps. Jo højere antal bps, jo højere overførselshastighed.

Transmissionshastigheden kan også opgives med præfix som:

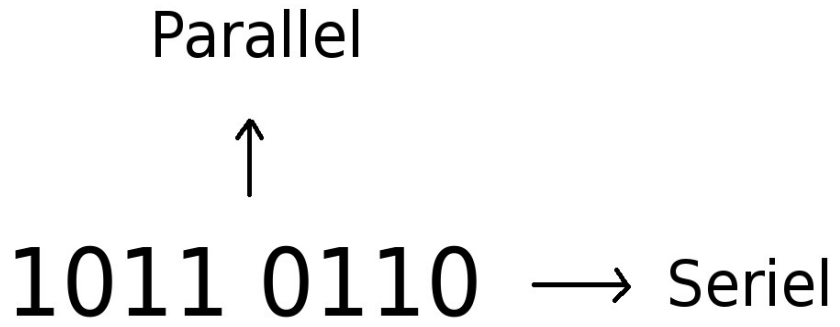
Kbps = Kilo bit per sekund
Mbps = Mega bit per sekund
Gbps = Giga bit per sekund
Tbps = Tera bit per sekund

Overførselshastighed kaldes i daglig tale også for båndbredde, for at undgå forvirring vil vi i denne notesamling undgå den formulering.

Opgave 1.1.1 Bestem overførselshastigheden for en lastbil fuld af DVD'er, der skal flytte bit fra København til Rom, hvilket er ca. 2.500 km. Du kan gå ud fra lastbilen lad er 16 m lang, 3m høj og 3 m bred.

1.2 Serial/parallel

Datatransmission kan deles op mellem parallel og seriel transmission.



Ved seriel kommunikation sendes en bit ad gangen over en linie, ved parallel transmission sendes flere bit samtidig, altså parallelt, og vi taler om bus bredden. En typisk bus bredde er en byte, altså 8 bit.

Parallel datakommunikation bruges generelt over korte strækninger, som inde i en CPU og i nærheden af CPU'en, altså på printkortet. Jo længere afstand jo mere besværligt er det at have en bus, hvorfor man bruger seriel kommunikation.

Alt andet lige er parallel datakommunikation busbredden gange hurtigere end seriel kommunikation, men hvis blot det går hurtigt nok, kan det være lige meget, derfor er seriel datakommunikation i dag meget udbredt.

1.3 Transmissionsformer

Simplex
Den simpleste form for datakommunikation er, når en taler, og en lytter, dette kaldes simplex.

Broadcast
Når en taler og mange lytter, kaldes det broadcast, hvilket er engelsk for radioudsendelse, altså en sender, og mange modtagere.

På engelsk kaldes en sender for en tranciever, hvilket forkortes til Tx, på engelsk kaldes en modtager en reciever, hvilket forkortes til Rx. Det er begreber der benyttes i datakommunikation.

Multicast
Multicast er en variation af broadcast, hvor man sender til en specifik gruppe, altså en sender, mange, men ikke alle, lytter.

Halv duplex
Hvis vi har et system hvor en kan tale og en anden lytte, til taleren holder op, derefter kan lytteren tale, og den første taler må så lytte, kalder vi det halv duplex.

Ved halv duplex er det er ikke tilladt at tale i munden på hinanden.

Begrebet kendes fra walki talkier.

Duplex
Når to parter taler sammen, og begge kan tale og lytte samtidig, kalder vi det duplex, eller fuld duplex. Med duplex kan man altså tale i munden på hinanden.

1.4 Modem

Modem er en sammentrækning af to engelske ord, Modulator og DEModulater.

En modulator er en teknisk ting der kan lægge et signal på en radiobølge, demodulatoren gør det modsatte, altså trækker et signal ud af en radiobølge.

Et modem benyttes når man ønsker at overføre et digitalt signal, over en analog forbindelse, for at kunne gøre det laver modemmet nuller og ettaller, om til bølger af bestemte frekvenser.

Eksempel:

Nul (Lav)	=	1.000 Hertz
ET (Høj)	=	2 .000 Hertz

På denne måde kan man overføre et datasignal over en almindelig telefonlinie, der har en båndbredde på 3.000 Hertz.

Forklaring: Båndbredde

Ordet båndbredde betyder hvilket bånd målt i Hertz der kan overføres. Hvis frekvensbåndet fra 17 Mhz til 21,5 Mhz kan overføres, er båndbredden: 21,5 – 17 = 4,5 Mhz.

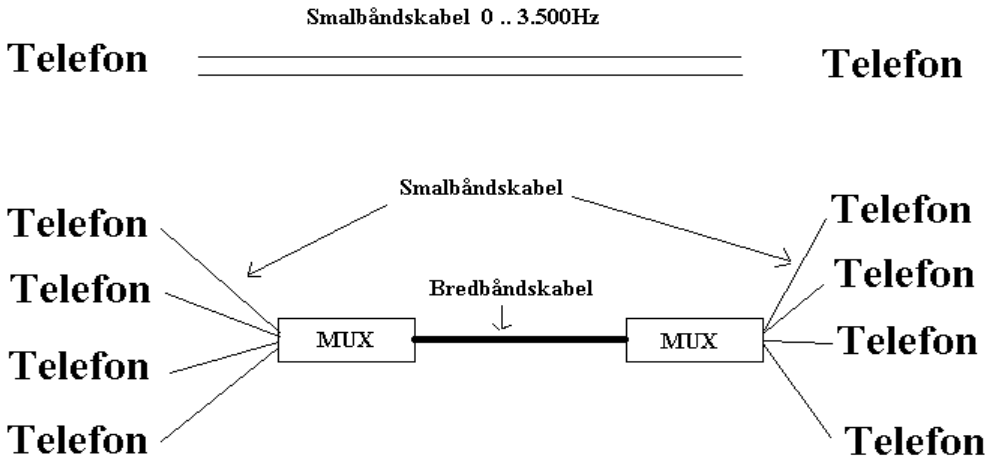
Der er en direkte matematisk sammenhæng mellem båndbredden målt i Hertz, og den teoretisk højest opnåelige hastighed målt i bit per sekund, men den er for kompleks at komme ind på her.

Modem teknikken blev opfundet i 1920'erne, til brug for de netop opfundne fjernskrivere.

Frem til slutningen af 1900 tallet, benyttede man mest PLL (Phased Locked Loop) i modemer, i dag bruger man hovedsageligt DSP'er (Digital Signal Processor) for at få det meste ud af en givet linie.

1.5 Multiplexning / Mux

Begrebet multiplexning forkortes MUX, svarer til begrebet multitasking, altså det forhold at man kan gøre flere ting (tilsyneladende) samtidig.



Figur 1.5.1

Der er to former for multiplexning: Frekvensmultiplexning og tidsmultiplexning.

Frekvensmultiplexning / FDM:
Frekvensmultiplexning går principielt ud på, at man sender flere bit samtidig (parallelt), men på hver sin frekvens.

Eksempel:
Send en Byte ad gangen. En Byte består af 8 bit kaldet bit1 til bit8

Bit	**Høj/Herz**	**Lav/Hertz**
1	1.100	1.150
2	1.200	1.250
3	1.300	1.350
4	1.400	1.450
5	1.500	1.550
6	1.600	1.650
7	1.700	1.750
8	1,800	1.850

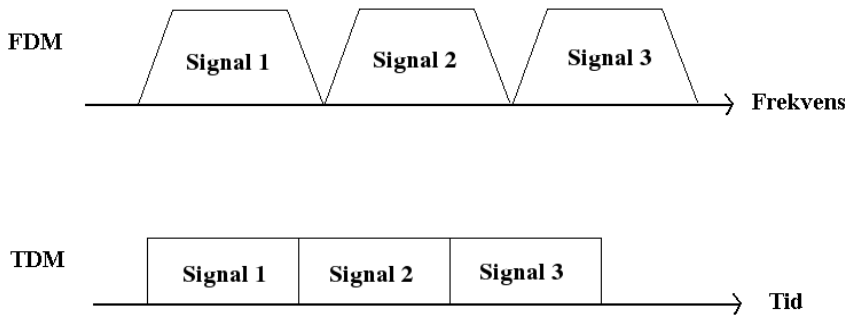
Ved frekvensmultiplexning sendes der altså ægte parallelt, ikke indbildt (Virtuelt)

Teknikken kaldes også bærefrekvens hvilket forkertes til BF.

Tidsmultiplexning / TDM;
Ved tidsmultiplexning sendes der først et signal i en kort tid, så et andet signal en kort tid, så et tredje signal en kort tid, hvorefter det første signal sendes en kort tid. Princippet er de samme som i en film der består af en række enkeltbilleder, vises de hurtigt nok ser det ud som en film.

Princippet kan naturligvis udviddes med uendelig mange signaler, men jo flere signaler man ”putter ind” (muxer) des længere tid tager det at komme en tur rundt, er der ikke båndbredde nok vil der opstå svartider, video vil rykke og samtalen vil hakke.

Tidsmultiplexning er nok den mest udbredte i dag, fordi den er nem og billag at lave.



Figur 1.5.2

1.6 Topologi

Topologi er en matematisk disciplin der er udviklet fra geometrien, inden for datakommunikation benytter vi begrebet til at beskrive opbygningen af et netværk, både det fysiske, og de virtuelle netværk.

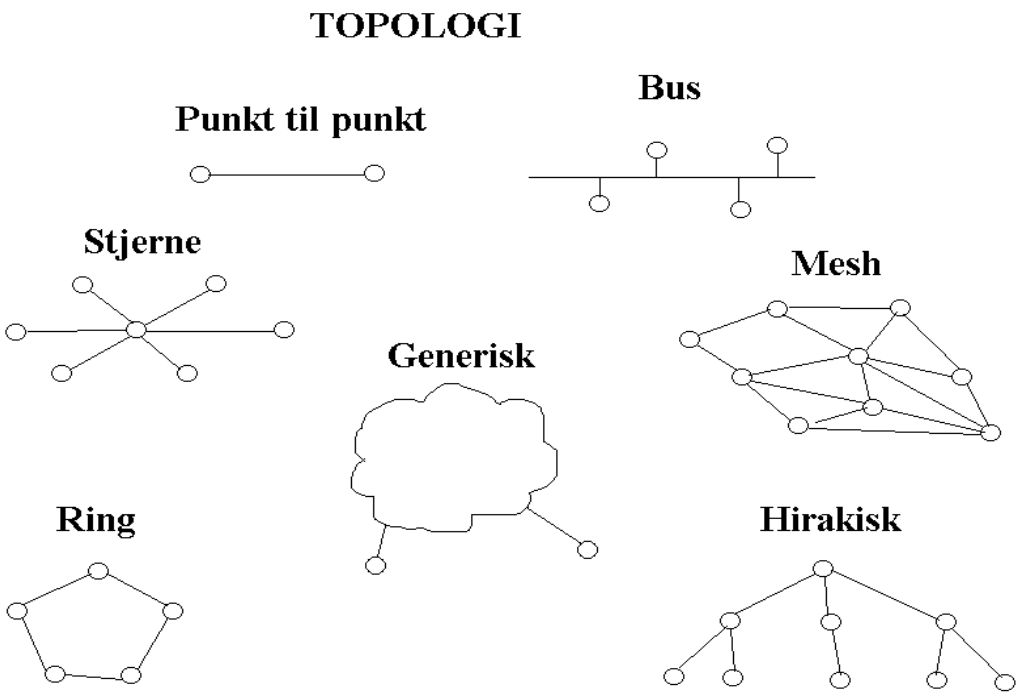
Først skal vi have nogle begreber på plads der stammer fra ISO 8802-3 suiten af standarder:

LAN (Local Area Network)
Benyttes om et netværk som er fysisk lokalt, og som ikke opererer over lag 2, altså datalink laget. (Se kapitel 3)

WLAN (Wireless Local Area Network)
Udtrykkes benyttes når man har et trådløst netværk i et lokalt område, der ikke benytter noget protokollag over lag 2, datalink laget.

MAN (Metropolitab Area Network)
Metropolitan betyder by, så et MAN er altså et netværk der dækker et byområde, det er underforstået det må bestå af flere LAN's.

WAN (Wide Area Network)
Dette er den største konfiguration mulig, hvilket i praksis er uendelig, Internettet er et godt eksempel på et WAN, det er telefonnettet også.



Figur 1.6.1

Punkt til punkt topologi
Når en computer er forbundet direkte til en anden, med et kabel eller et radiosignal, kaldes det en punkt til punkt forbindelse. Dette kan eksempelvis være når man tilslutter sin mobiltelefon til sin bærbare, med et USB kabel, eller via en trådløs forbindelse.

Bus topologi
En bus forbindelse er når alle computere kan høre alle andre computere på nettet, hvilket medfører, at alle computere skal have en adresse, det er en af protokolens opgaver at definere hvordan det skal ske.

Et bus system kan sammenlignes med et trådløst netværk, og kaldes også for ethernet. Det er i udbredt brug inden for LAN.

Stjernetopologi
Når den fysiske struktur er således, at der er et centralt punkt, og et lag derudover, kaldes det en stjerne, hvilket er et specialtilfælde af en hirakisk topologi.

I denne topologi vil hvert knudepunkt typisk være et LAN og/eller et WLAN, den samlede topologi vil altså typisk være et MAN eller et WAN

Hirakisk topologi
Når den fysiske struktur er hirakisk i mere end 2 lag, kaldes det en hirakisk topologi.

I denne topologi vil hvert knudepunkt typisk være et LAN og/eller et WLAN, den samlede topologi vil altså typsik være et MAN eller et WAN

Mesh Topologi
I Mesh topologien er flere knudepunkter samlet på en måde der ikke kan beskrives med anden betegnelse. Knudepunkterne vil typsik være LAN's, hvorfor et mesh ofte vil være et MAN eller et WAN.

Ring Topologi
Når computere eller knudepunkter er samlet i en ring, kaldes det sjovt nok en ring topologi. En ring kan have den ulempoe, at hvis den brydes et sted, vil hele nettet dø, men det kan håndteres i højere protokollag.

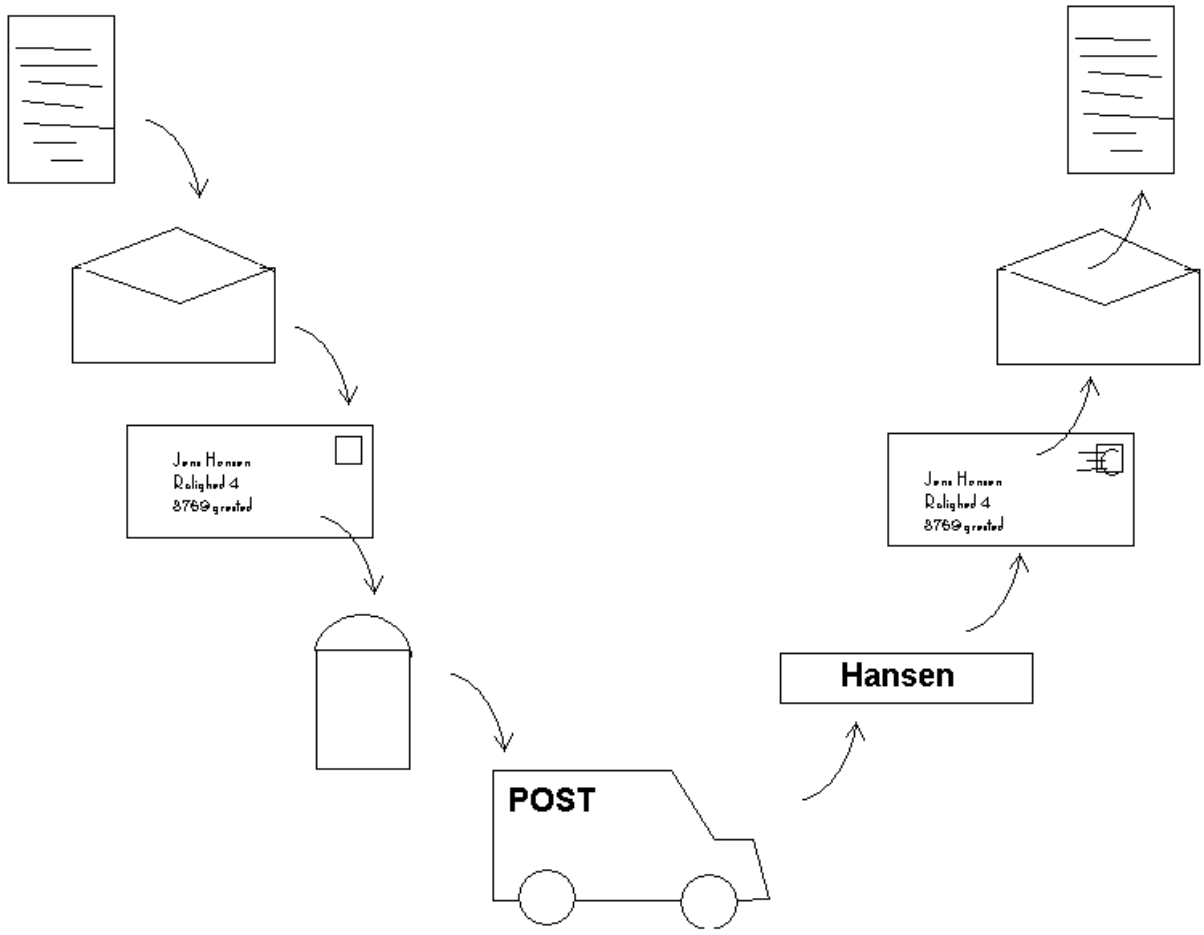
Generisk
Denne forklarer vi med en sky hvor vi kan ”tilkoble” enkelte computere eller hele LAN/MAN/WAN. Skyen har i årtier været brugt til at beskrive et stort ikke nærmere specificeret netværk. Den seneste tids markedsføring af ”skyen” er altså bare et nyt ord for dataanalfabeterne.

Topologierne kan blandes, så eksempelvis en del af et mesh er en ring, og/eller et hirakisk delelement.

Det skal understreges et netværk kan have en fysisk topologi, og en række helt forskellige virtuelle topologier, men for nuværende holder vi os til den fysiske topologi.

1.7 Protokoller

En protokol er en aftalt måde at kommunikere på.



Princippet i en kommunikation protokol kan sammenlignes med det at kommunikere via et brev. For at gøre programmeringen mere overskuelig deles opgaverne op i delopgaver, kaldet lag. Princippet i et lag er, at det korresponderer med begge sider. Lad os kigge på eksemplet:

Til at starte med skal vi gøre os klart, at vi har en modtager (kaldet Rx), og en sender (kaldet Tx).

For at sende en skriftlig meddelelse skal senderen først skrive det. Det at skrive et brev korresponderer til det at modtageren læser brevet. Når brevet er skrevet, skal der laves en hel del, der dybest set er ligegyldigt for både sender og modtager, men det er ikke ligegyldigt for postvæsenet.

Brevet skal puttes i en konvolut, og konvolutten skal lukkes. Det korresponderer med, at modtageren åbner konvolutten, og tager brevet ud, det er den samme operation, blot omvendt.

Når senderen har puttet brevet i konvolutten, skal han sikre sig, at der er en afsender og en modtager adresse på konvolutten. Dette korresponderer med, at en lovlydig modtager, ikke åbner et brev, før han har tjekket om det er til ham, og hvem det kommer fra. (For en computer er det spild af tid, at læse post der ikke er til den.)

Når senderen er sikker på hans post er pakket korrekt ind, lægger han det i postkassen og postvæsenet sørger for at det bliver lagt i modtagerens postkasse.

Bemærk: Der ligger en del opgaver i at få posten frem til modtageren, det abstraherer vi fra i dette eksempel, det er jo ikke vores problem.

Det at sende et brev indeholder altså en del overhead, (en for senderen dybest set overflødig information) men det er nødvendigt for at bringe posten frem til den rigtige modtager inden for en rimelig tid.

Overhead i et brev, er konvolutten, og alt det der står på den.

Opgave 1.7.1:

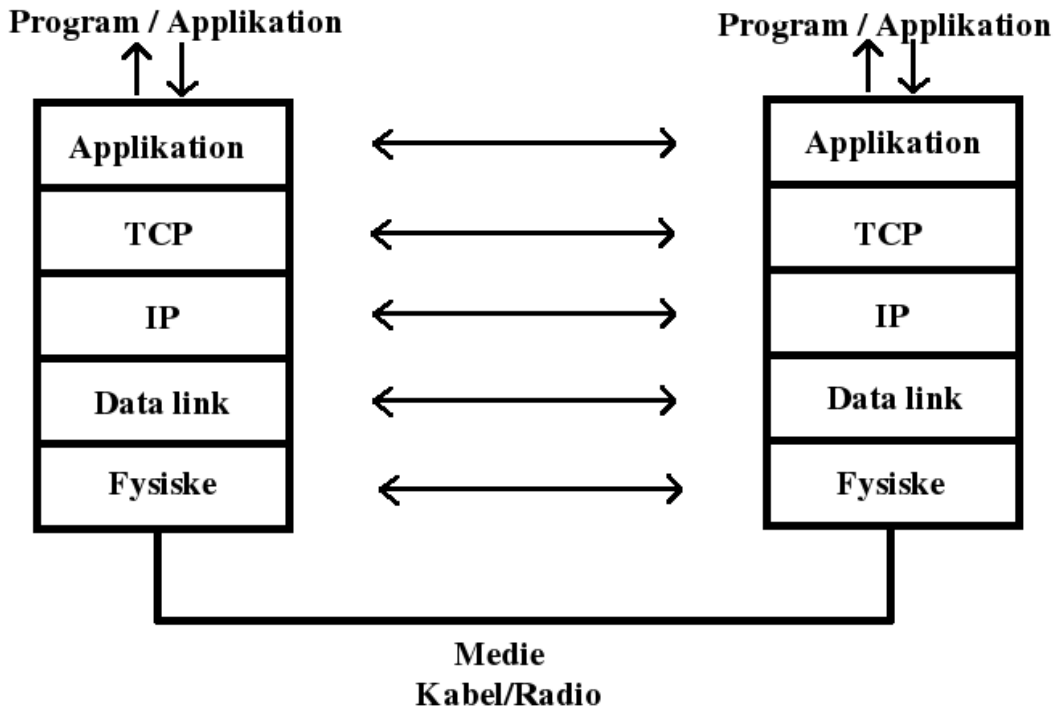
Hvordan fungerer protokollen hvis modtageren ikke eksisterer.

1.7.1 TCP/IP som protokolstak

TCP/ IP kan betragtes som en stak af kommunikations protokoller, der ligger oven på hinanden, hvor hvert lag korresponderer til det over og det under, samt det modstående protokollag, på samme måde som en konvolut forventer der kommer et brev i den fra det overliggende lag, og at der kommer et frimærke og en adresse på, til det underliggende lag.

De 5 protokollag kaldes:

Lag	Navn	Funktion
1	Det fysiske	Sikre elektriske forbindelse, og udformning af stik
2	Data Link	Sikre signal fra punkt af til B på samme LAN
3	Internet Protokol (IP)	Sikre transmission mellem LAN (Kaldes også routning)
4	Transmission Control Protokol (TCP)	Sikre transport fra afsender til modtager
5	Applikationslaget	Præsentere data for en applikation



Lad os straks fjerne en udbredt misforståelse: Applikationslaget er IKKE applikationen, det er en del af kommunikationsprotokollen.

Eksempelvis er et ftp program netop et program, der via API (Applikation Programming Intertface) kalder det øverste lag i protokolstakken, som altså er applikationslaget.

I TCP/IP er datakommunikation brudt i fem overordnede regler, hver af de fem regler har en samling af regler inde i sig, men de kan betragtes som en funktion man kalder, med en parameterliste, som udgør API'et.

Hver af de fem lag kommunikerer med modparten ved at tage de data laget har modtaget, og pakke dem ind efter regler givet i protokollen for det lag, og så sende de data ned til det næste lag.

Det omvendte sker når en computer modtager data, lag for lag stripper den de data af som de underliggende lag har indsat, for at ende op med de data der blev puttet ind i applikationslaget, af en applikation.

På samme måde som det kræver en applikation at sende data, kræver det en applikation for at modtage dem.

2. Det fysiske lag

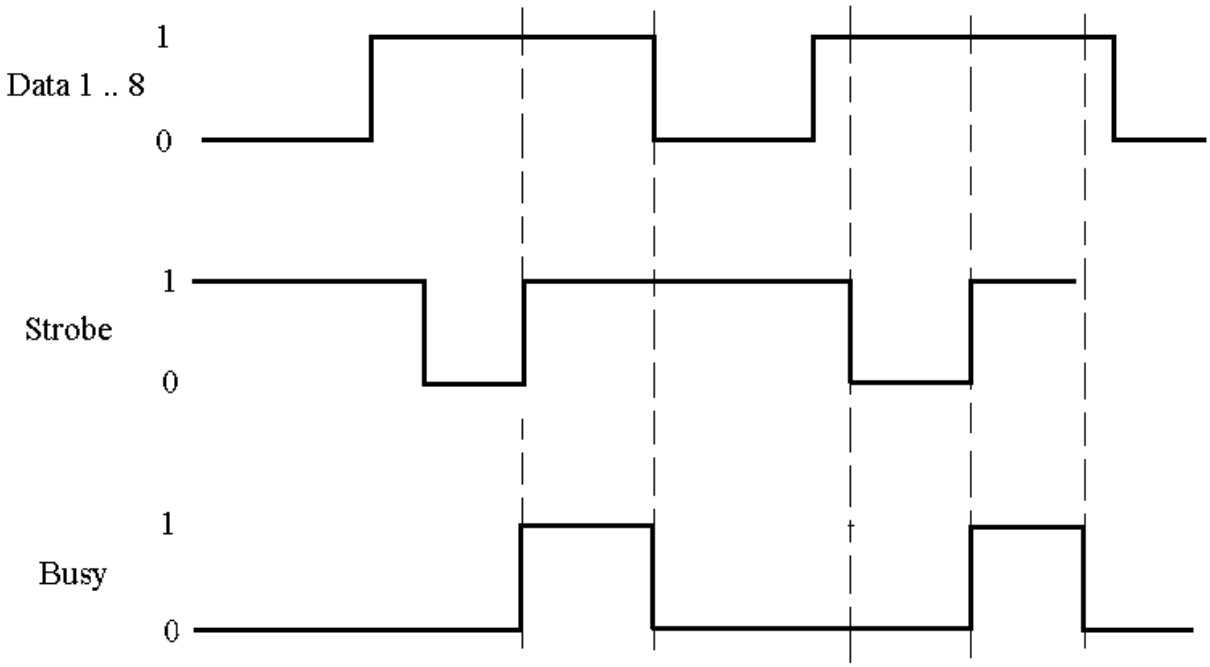
Som traditionen byder starter vi vores beskrivelse af protokolstakken for neden, med det fysiske lag, der ligger tæt på elektronikken.

Bit orienteret / Tegn orienteret protokol.

2.1 Centronix

Centronix er en byteorienteret protokol, hvilket betyder dataformatet er byte. Den var en udbredt måde at kommunikere med printere, frem til omkring årtusinde skiftet, hvor USB overtog.

Centronix er parallel datakommunikation der sender 8 bit, altså en byte, parallelt fra en afsender til en modtager. Timingsdiagrammet er vidst på figur 2.1.



Figur 2.1

For at benytte Centronix skal man have 10 ledere, en for hver bit, plus to handshake signaler, kaldet Strobe og Busy. Der er to spændingsniveauer, 0 (nul) volt som per defenition er digitalt nul, og +5 Volt, der per defenition er digitalt 1 (et).

De 8 databit og Strobe er output fra afsenderen, der typisk er en computer, og altså input for modtageren, der typisk er en printer.

Busy er output fra modtageren (printeren) og derfor input for afsenderen (computeren).

Busy er aktiv lav, hvilket betyder, at hvis der ikke er spænding (lav = usand) så er modtageren ikke klar, modtageren er Busy, og det nytter ikke at forsøge at stoppe nogen data ind i modtageren, modtageren lytter ikke.

Aktiv lav er naturligvis valgt ud fra, hvis modtageren er slukket, er der ikke noget signal der kan melde noget, Busy er derfor lav (Busy) og ingen forsøger at sende data til den, når den er slukket.

Når modtageren er klar til at modtage 1 (en) byte, sætter modtageren Busy høj, og afsenderen kan nu sætte sine databit, når alle databit er sat og stabile, (dette er en elektronik overvejelse), sænker afsenderen strobe til lav, hvilket betyder, at nu, og ikke før, må modtageren læse databitne.

Når modtageren har læst databitne, sætter modtageren Busy lav, og processen fortsætter til der ikke er mere at læse.

Denne teknik kaldes hardware handshake, fordi synkroniseringen sker ved elektriske signaler.

Hvis man ikke foretog nogen form for synkronisering, ville data gå tabt.

2.2 RS232

RS232 er en videreudvikling af det gamle fjernskriver system, oprindeligt udvidet fra et dataord på 5 bit, til et dataord på 7 bit, og senere 8 bit.

RS232 er en protokol på det fysiske lag til seriel datakommunikation. RS232 er altså en byte orienteret seriel protokol.

For at sikre korrekt modtagelse tilbyder protokollen en såkaldt paritetsbit, der vil finde en fejl i 50% af alle tilfælde, hvis der opstår en fejl. Denne teknik kaldes en fejldetekterende protokol.

RS232 er altså en byte orienteret seriel protokol, med fejldetektering.

I RS232 er 0 (nul) defineret til at være enhver spænding mellem +3 Volt til +15 Volt, modsat er enhver spænding mellem -3 til -15 Volt defineres som 1 (et). Spændinger mellem -3 Volt og +3 Volt må ikke forekomme, hvis de gør er data ubestemt, og vi har fejl.

Ved 50 bps må bitlængden være: 1/50 sekund = 20 mS

Ved en bps på 19.200 bps må bitlængden være: 1/19.200 = 52 µS

Både hardware og software handshake

Selv om RS232 i stor stil er erstattet af USB i PC verdenen, er protokollen stadig meget udbredt inden for teknisk IT.

2.3 USB

USB er en seriel protokol, der har overtaget fra RS232 i mange brugermiljøer.

2.4 ISO 8802-3 Ethernet

Coax længde versus tid ?

Ethernet IEEE 802-3 Frameformat det fysiske lag

PRE: Preample, 7 gange 10101010

SOF: Start Of Frame Delimiter 1 gang 10101011

Payload: Data fra Data link laget

2.5 AX25

2.6 ISO 11801 Struktureret kabelsystem

Den dyre del af en datainstallation er kabelanlægget, så for at sænke prisen på kabelanlæg etablerede det amerikanske telefon og data selskab AT&T i 1980'erne, et system de kaldte PDS

(Premessis Distributed System), som egentlig bare var en forenkling af IBM's tilsvarende, men ret kostbare system

Senere blev dette system standardiseret under ISO (Internationel Standardisation Organisation), som er et organ under FN (Forenede Nationer).

Det geniale ved ISO 11801 systemet var, og er, at et (1) sæt kabler kan bruges til alting. Man skulle ikke længere trække et kabel for at forbinde en computer med en server, et andet kabel for at forbinde en computer med en printer, og et tredje kabel for at forbinde en telefon med telefonnettet.

2.7 Bluetooth

2.8 Afrunding

Hvis du vi vide mere om forskellige teknikker på det fysiske lag er hardwarebogen et godt opslag:

http://www.hardwarebook.info/

3 Data link laget

På data link laget oprettes der en forbindelse mellem to enheder på samme lokalnetværk (LAN).

3.1 ISO 8802-3 Ethernet

Framet for data link laget er vist i figur 3.1.1

Ethermet IEEE 802.3 Frameformat Data link laget

Figur 3.1.1

DA: Destinations (modtager) Adresse længde 6 Byte

SA: Source (afsender) Addresse længde 6 Byte

Type: Oplyser typen af frame, længde 2 Byte

Payload: Data fra IP laget, længde 46-1500 Byte

CRC: Cyclic Redundans Code, fejldetekteringskode, længde 4 byte

Afsender og modtager adresser på dette lag i denn protokol kaldes også MAC adresser, hvilket er en forkortelse for Media Access Control address.

CSMA/CD: Carier Sence, Multible Access med Collision Detection, hvilket betyder: Før der sendes lyttes der på nettet, for at undersøge om der er nogen der sender, (Carier Sence), Der er mange der kan sende på det samme net (Multible Access), og hvis der detekteres en kollision (to eller flere sender samtidig) detekteres kollisionen, og der forsøges at sende igen.

Max 5 lag switche

3.2 ISO 8802-11

3.3 Bluetooth

Dette er en standard for trådløs datakommunikation over korte afstande, udviklet på dansk initiativ, heraf navnet, fra Harald Blåtand.

3.4 AX25

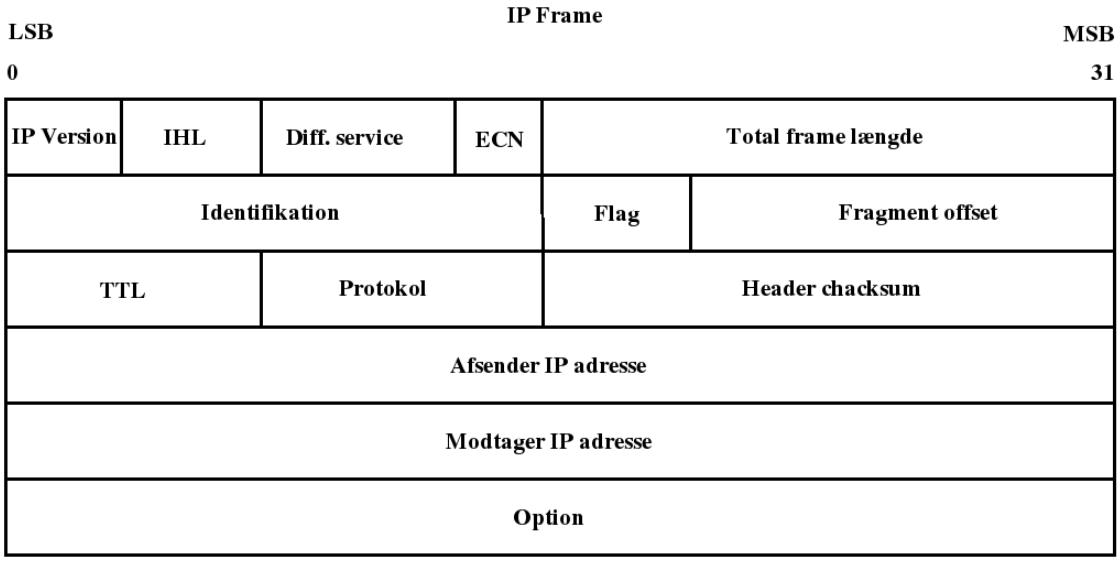
AX25 er benyttet af radioamatører til et amatørbaseret universelt Internet, nettet rækker faktisk ud af atmosfæren, da der er flere satellitter i systemet.

4 IP laget

IP er en forkortelse for Internet Protokol, altså en protokol der forbinder netværk, ligesom intercitytog er tog der forbinder byer.

IP lagets opgave er at skabe en forbindelse fra en host til en anden host, evt. over flere routere.

4.1 IP Frame



IP Version: 4 bit, beskriver protokollens versionsnummer 4 = IPv4, 6 = IPv6.

4.2 IP Notationsform / CIDR

Måden vi beskriver en IP adresse på kaldes CIDR, hvilket er en forkortelse for Classless Inter-Domain Routing.

Eksempel 1: 189.231.22.0/24 betyder de første 24 bit er fast, de resterende 8 kan variere, hvilket betyder vi har et subnet med 255 IP adresser, men her fragår 2 IP adresser.

189.231.22.255 er ikke en IP adresse, det er en broadcast og 189.231.22.0 er ikke en IP adresse, men et netværk.

Netmasken for dette net er 255.255.255.0, fordi de første 24 bit er faste, de sidste 8 bit er til hosts.

Eksempel 2: 189.41.17.16/30 betyder de første 30 bit er fast, der er altså i alt 4 IP adresser til rådighed fra 189.41.17.17 til 189.41.17.20, her er 189.41.17.16 netværket og 189.41.17.21 broadcast for det netværk.

Netmasken for dette net er 255.255.255.252, fordi de første 30 bit er faste, de sidste 2 bit er til hosts.

Hvis forskellige computere for et netværk ikke har orden i deres broadcast filter, vil der opstår netværksproblemer der kan have periodisk karakter.

IP range	**Beskrivelse**	**Reference**
0.0.0.0/8	Denne host på dette net.	RFC 5735
10.0.0.0/8	Private netværk	RFC 5735
100.64.0.0/10	Shared Address Space (NAT)	RFC 6598
127.0.0.0/8	Loopback	RFC 5735
169.254.0.0/16	Lokalt link	RFC 5735
172.16.0.0/12	Private netværk	RFC 5735
192.0.0.0/24	IETF Protocol Assignments	RFC 5735
192.0.2.0/24	TEST-NET-1	RFC 5735
192.88.99.0/24	IPv6 til IPv4 ”oversætter”	RFC 5735
192.168.0.0/16	Private netværk	RFC 5735
198.18.0.0/15	Network benchmark tests	RFC 5735
198.51.100.0/24	TEST-NET-2	RFC 5735
203.0.113.0/24	TEST-NET-3	RFC 5735
224.0.0.0/4	IP Multicast	RFC 5735
240.0.0.0/4	Reservered for fremtidigt brug.	RFC 5735
255.255.255.255	Broadcast, forwardes ikke.	RFC 5735

For IPv6 er princippet det samme, dog skrives den normalt i Hexadecimal form, hvor hver byte er adskilt af et kolon. Eksempel: 2301:67a:51:1g0:0f:20:5:1/64. Vi vil ikke komme nærmere ind på IPv6.

4.3 IP i praksis

Her vil vi se på et eksempel på hvordan man kan konfigurere netværket på en Debian. Husk at slå network-manager fra, hvis den kører, ellers er det den der bestemmer.

Eksempel på: /etc/network/interfaces

```
auto lo
iface lo inet loopback

# auto eth0
# iface eth0 inet dhcp

auto  eth0
iface eth0 inet static
     address 192.168.1.25
     broadcast 192.168.1.255
     netmask 255.255.255.0
     network 192.168.1.0
     gateway 192.168.1.1
     dns-nameservers 8.8.8.8
```

I dette eksempel har vi først erklæret vores loopback interface, det er tilrådeligt altid at have et loopback interface.

Derefter er der er erklæring for eth0 (det første kablede netværk på denne computer). Det er en erklæring om at starte netværket op ved at bede om en IP adresse hos en DHCP server ude på nettet, de 2 linier er udkommenteret, og ignoreres derfor af styresystemet.

Derefter følger konfiguration af eth0, startende med linien `auto eth0` det betyder denne konfiguration skal udføres ved start af computer.

`address 192.168.1.25`	Her defineres computerens IP adresse
`netmask 255.255.255.0`	Dette er netmasken for dette net .
`network 192.168.1.0`	Dette er netværket.
`broadcast 192.168.1.255`	Dette er broadcast adressen for dette net.
`gateway 192.168.1.1`	Dette er adressen på routeren ud af LAN'et.
`dns-nameservers 8.8.8.8`	Dette er adressen på navneserveren.

Det er ikke nødvendigt at at definere både netmasken, netværket og broadcast, da de kan udledes af hinanden, en er nok, men alle er taget med her her for eksemplets skyld.

Opgave 4.3.1: Konfigurer 2 computere som følger:

Computer 1: Adresse 192.168.10.1 Broadcast 192.168.10.255
Computer 2: Adresse 192.168.10.2 Broadcast 192.168.10.255

Lad hver af de to computere pinge hinanden på skift, og undersøg datapakkerne med en sniffer som eksempelvis tcpdump.

Passer placeringen af IP adresserne ?

Opgave 4.3.2: Tilpas opgave 4.3.1, så computer 2 ændrer adresse til 192.168.11 2

Kan de 2 computere nu pinge hinanden ?

Opgave 4.3.3: Tilpas opgave 4.3.2, så begge computere har broadcast adressen 192.168.255.255

Kan de 2 computere nu pinge hinanden ?

Opgave 4.3.4: Dan en fil på 100 KByte, flyt den fra en computer til en anden, undersøg pakke fragmenteringen ved hjælp af en sniffer.

a. Passer det et data link frame ikke må være større end 1.500 Byte ?

b. Passer det et IP frame ikke må være større end 65.536 Byte ?

5 TCP laget

6 Applikationslaget

7 Andet

7.1 Firewall

Begrebet firewall kommer fra flyvningens verden. Helt op til midt i 1970'erne, var man så vand til en flymotor brød i brand ved almindelig drift, typisk fordi noget spildt olie antændtes, at man placerede en firewall (brandvæg) mellem motoren, og resten af flyveren. Derved kunne man opnå en brand ikke bredte sig ud over motorrummet, i motorrummet var det ikke noget væsentligt der tog skade af en brand, motoren er jo varm alligevel, og man kunne typisk starte motoren igen, når branden havde slukket sig selv på grund af manglende næring.

Ovenstående problem var så udbredt, at for at krydse store øde områder, som Atlanten, skulle flyveren være udstyret med mindst 4 motorer, da man forventede der altid var mindst en motor der fejlede undervejs.

I da er det tilladt at krydse store havområder med en enkelt motor, eller sagt på anden måde, man har nu så meget kontrol over motorteknologien, at man kan stole på motorene.

En firewall til en computer fungerer efter samme princip. Fordi man ikke kan stole på styresystemet, er man nød til at bruge en firewall for at fange manglerne i styresystemet.

IPTABLES

7.2 Virus tjekker

Mange fejlmeldinger sløver agtpågivenheden, derfor databaseopslag, ingen forsøg på intilligense.

7.3 NAT Network Adreess Translation

7.4 VLAN / Virtual Local Area Network

Appendix A Kilder

Appendix B Stikordsregister

Stikordsregister