

En dummy guide til nettverk

Eirik Berg

November 2020

Introduksjon

Dette er en liten guide til nettverksbiten i IN1020 da jeg merker at mange (meg selv inkludert) synes denne biten er litt vanskelig. Dette lille prosjektet vil derfor forhåpentligvis være til nytte for deg som leser dette, men kanskje mest av alt for meg selv. La oss derfor sammen gjøre et grunt dypdykk i nettverk, og gjøre oss klare for eksamen!¹

NB!

Som en disclaimer vil jeg understreke at dette dokumentet er kraftig inspirert av både foiler og wikipedia. Henter man frem Wikipediasidene til flere av konseptene som diskuteres vil man kunne finne likheter. Dette dokumentet er ment som en samling av denne informasjonen på ett sted:)

¹Jeg skal ikke ta eksamen:)

Hva er datanettverk?

Datanettverk handler om at datamaskinene snakker sammen. Dette gjøres ved hjelp av flere komponenter:

- Tjenere (endesystem)
- Klienter (endesystem)
- Switcher (intermediate system)
- Routere (intermediate system)

Tjener (eng. server)

Tjener er enten en datamaskin eller et program. En tjenermaskin tilbyr nettverkstjenester (som lagring, utskrift, databaser, e-post og nettsider) til klienter i nettverket.

Klienter

Klienter er programvare på din personlige datamaskin (for eksempel Google Chrome™(derav ordet internet client som jeg innså akkurat nå)). Klienter og tjenere jobber sammen (kreativt kalt "klient/tjener-teknologi") hvor klienten ber om en tjeneste (opprettet forbindelse), og tjeneren leverer den tjenesten (svarer på forespørselen). Klient/tjener-prinsippet ligger til grunn for Internett-teknologi. Å sende og lese mail krever for eksempel en klient på din private maskin, som snakker med et tjenerprogram. Enkelt sagt snakker én klient med én tjener, mens én tjener snakker med flere klienter.

Switcher

En switch styrer datatrafikk mellom ulike noder i et nettverk, som PC, tjenere, skriver etc. Nå kommer det litt info om switcher hvor jeg skriver noen ord i kursiv. Disse ordene er forklart senere i dokumentet. Kanskje klarer jeg til å med å gjøre dem trykkbare? Spennende saker. Her kommer den infoen jeg snakket om:

Switchen tar imot informasjon i *frames*, ikke *pakker*. Den lagrer hvilken port dataene kom på, og hvilken *MAC-adresse* de ble sendt fra. Dersom MAC-adressen ligger i listen over allerede registrerte MAC-adresser, blir dataene sendt kun til den porten denne adressen er lagret på. Ellers blir den sendt videre til alle porter. Breath of the Wild er ikke tilgjengelig på en slik switch:(

Routere

En enhet som videresender datapakker gjennom datanettverk.

Person 1: "Æsj! Jeg har ikke nett as bro"

Person 2: "Har du prøvd å skru routeren av og på bro?"

Aksessmodeller

I tillegg til å være et vanskelig ord å skrive, er aksessmodeller enkelt forklart forskjellige prinsipper til hvordan man kobler seg opp på internett.

Klient/tjener

Denne er forklart ovenfor under "klienter" fordi jeg ikke så denne komme. En kjapp repetisjon kan ikke skade nå en halv side senere. Dette er den tradisjonelle aksessmodellen. Dette er nok også den de fleste av oss tenker på når vi hører ordet Internett; vi har en klient (for eksempel en nettleser) som kommuniserer med en tjener (server). Dette kan dog fort bli dyrt dersom strukturen blir stor da man må ha flere og flere tjenere.

Peer-to-Peer (P2P)

I en P2P aksessmodell er alle nodene i nettverket likeverdige, og alle kan nå hverandre. Man går altså ikke via en server og så til en annen sin PC.

CDN

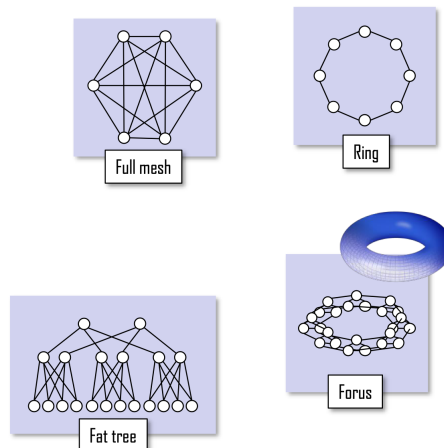
Nettverk kan distribueres i en struktur som heter CDN distribusjon (content delivery network). Dette er en hierarkisk struktur der populært innhold lagres på lokale tjenere, som har kontakt med regionale tjenere, som igjen kommuniserer med en kjernetjener. Dette er nytting struktur der båndbredden er begrenset (for eksempel fly).

Nettverkstopologier/-strukturer

Nettverkstopologi beskriver hvordan noder i et datanettverk er koblet sammen. Vi skal nå se nærmere på to slike topologier: Punkt-til-punkt og broadcast.

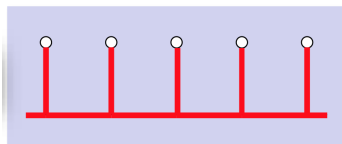
Punkt-til-Punkt

Punkt-til-punkt er en sammenhengende tilkobling mellom to noder. Punkt-til-punkt kan komme på flere former: stjernetopologi, treetopologi, ringtopologi, full mesh og forus for å nevne noen. Fellestrekket er at det er en serie av to noder som er koblet sammen.



Broadcast

I en broadcasttopologi kan flere enn to noder være koblet sammen, enten ved at alle er koblet sammen eller med en type "hub" i mellom seg. Dette betyr at når én node sender kan potensielt alle høre. Dette betyr også at det kan oppstå problemer hvis to noder sender samtidig, så dette må derfor koordineres.



Nettverkstyper

Det er nok ikke forventet å pugge alle disse (husk at hjelpemidler er tilgjengelige på eksamen), men det er nok lurt å bite seg merke i LAN og WAN.

- NFC - Near field communication
- BAN - Body area network
- PAN - personal area network
- LAN - Local area network
- SAN - Storage area network
- MAN - Metropolitan area network
- WAN - Wide area network

Avstand mellom punktene	Lokasjon	Eksempel
0,1 m og lavere	Kretskort	Multi-core prosessorer
1 m	Systemer	NFC, BAN, PAN
10 m	Rom	LAN, SAN
100 m	Bygninger	
1 km	Campus	
10 km	Byer	MAN
100 km	Land	WAN
1.000 km	Kontinenter	
10.000 km +	Planeter	

Protokoller

Hva er en protokoll?

En protokoll i IT-verden er et standardisert sett med regler som bestemmer tilkobling, kommunikasjon og dataoverføring mellom to endepunkter. Protokoller kan implementeres i både hardware og software, eller begge samtidig.

Når det kommer til datakommunikasjon skal vi nå se nærmere på TCP/IP-modellen. Dette er et sett med kommunikasjonsprotokoller som brukes på Internett og datanettverk. TCP/IP-modellen spesifiserer hvordan data bør pakkes, adresseres, sendes og mottas. Modellen brukes så og si overalt, så det er lurt å gjøre seg litt kjent med den. Vi gjør som en god diktator og splitter modellen i to og "hersker" over hvert av begrepene som bygger opp TCP/IP. Ikke en perfekt analogi, men nå er ikke jeg perfekt heller.

TCP

Transmission Control Protocol er en del av *transportlaget* i TCP/IP-modellen. TCP tilbyr pålitelig (den sier ifra til avsender om mottaker har mottatt data), ordnet og feilsøkt levering av en strøm av bytes mellom applikasjoner som kommuniserer via IP-nettverk (merket man at denne setningen var oversatt fra Wikipedia?). World Wide Web, mail, filoverføring med mer avhenger av TCP.

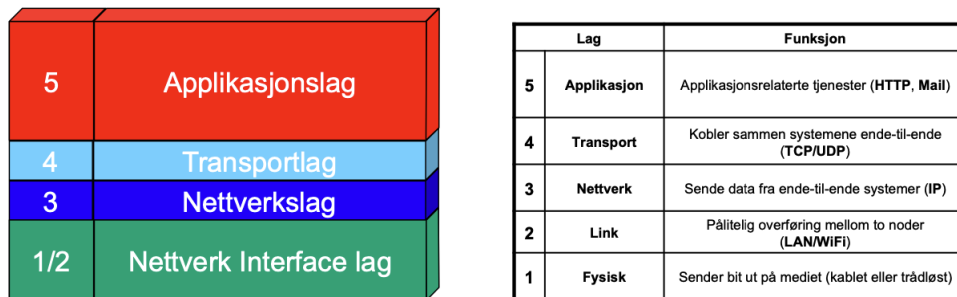
TCP krever en forbindelse mellom en klient og server før data kan sendes. Dette øker påliteligheten, men gjør øker latency (tidsbruken). UDP (User datagram protocol) er et eksempel på en protokoll som ikke krever denne forbindelsen; den er tilkoblingsløs. Vi kan se for oss en post-analogi. TCP ville vært som at postpersonen banker på døra di, bekrefter at du er den du er, og så gir deg posten. UDP blir mer som at postpersonen bare legger brevet i postkassa di (sånn som de vanligvis gjør). Det tar kortere tid, men sannsynligheten for en feilaktig levering øker.

IP

Internet Protocol er nok å anse som hovedelementet i TCP/IP-modellen. Det er på mange måter IP som setter opp tillater Internett slik vi kjenner det. IP har som jobb å levere pakker fra source host til destination host kun basert på IP-adresser. Den definerer metoder som brukes for å merke dataen som sender med blandt annet source- og destination-informasjon.

Lagdeling i TCP/IP-modellen

TCP/IP-modellen deles i fem lag: applikasjonslaget, transportlaget, nettverkslaget, linklaget og det fysiske laget.



Applikasjonslaget

Applikasjonslaget inkluderer protokoller som brukes av de fleste applikasjoner som brukes til data-/filoverføring over Internett. Eksempler på disse protokollene er Hypertext Transfer Protocol (HTTP), File Transfer Protocol (FTP), Simple Mail Transfer Protocol (SMTP) og Dynamic Host Configuration Protocol (DHCP). Eksempler på applikasjoner som benytter seg av disse er nettlesere, mail-applikasjoner, filoverføring (Airdrop) etc.

Transportlaget

Transportlaget sørger for datakanaler som applikasjoner bruker for mer oppgavespesifikk dataoverføring. Sentralt for IN1020 er hva TCP, og til en viss grad UDP, gjør da det er disse protokollene vi fokuserer på i kurset. Som nevnt ovenfor sørger TCP for at:

- Data ankommer i riktig rekkefølge
- Data har minimalt med feil
- Duplikat data ses vekk ifra
- Tapte eller forkastede data sendes på nytt
- Kontrollerer trafikken

UDP er mindre pålitelig og kommer derfor ikke med "garantier" på samme måte. Den brukes derfor mer for streaming der det er viktigere at dataen kommer tidnok heller enn at den er pålitelig. Se mer om UDP og TCP på side 13.

Nettverkslaget

Nettverkslaget kobler systemene ende-til-ende ved hjelp av ruting. Ruting er prosessen der man velger en sti for trafikken i et nettverk eller mellom/over flere nettverk. Rutingen kan være statisk eller dynamisk. Ved en statisk ruting er stiene satt opp manuelt i en tabell. Ved en dynamisk ruting fylles denne tabellen seg opp av seg selv. Ruting sørger også for metningskontroll, hvor det da sørges for at det ikke er for mange pakker på én sti. Den mest brukte nettverksprotokollen i dag er IP, versjon IPv4. Denne bruker en 32-bit adresse. IPv6 bruker 128 bit. En router jobber på dette laget.

Linklaget

Linklaget sørger for pålitelig overføring mellom to enheter. En switch jobber på dette laget. De vanligste linklagene er ethernet og wifi. Ved hjelp av MAC-adresser sendes data til riktig mottaker på det lokale nettverket. Dersom man er flere enheter koblet opp på det samme nettverket bruker switchen media access control (MAC)-adresser til å skille de forskjellige enhetene fra hverandre, og holde styr på hva som skal til hvilken enhet.

Det fysiske laget

Handler om signalrepresentasjonen av bits. Sørger for at 1-bit også blir mottatt som 1-bit og ikke 0-bit. Ser også på hvordan noder er koblet sammen, og hva slags spenning diverse enheter skal ha. Er på mange måter det som ligger til grunn slik at enheter er påskrudd og tillater videre kommunikasjon i lagene over.

Kort om dataenheter i TCP/IP-modellen

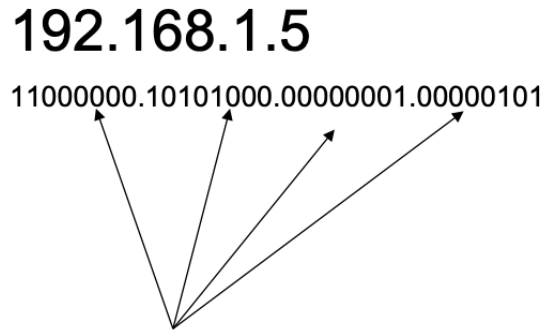
Dataenheter har forskjellige navn avhengig av hvilket lag vi befinner oss i:

- Segments brukes på transportlaget
- Packets/pakker brukes på nettverkslaget
- Frame brukes på linklaget
- Bits brukes på det fysiske laget

Det kan ses på som at bits settes sammen til frames, som settes sammen til pakker, som igjen settes sammen til segments.

Hva er en IP-adresse?

IP-adresser er på mange måter postadressen over internett. Man skiller gjerne mellom to typer: IP-adresser utenfor det lokale nettverket, og de innenfor. Er adressen adressert til en enhet innenfor det lokale nettverket leveres pakken direkte til den enheten. Er den utenfor viser IP-adressen til en router, som igjen får ansvaret for å sende pakken videre innenfor det lokale nettverket. En kan se på Internett som en samling med masse nettverk, der sjefene i hvert nettverk (routerne) kommuniserer med hverandre. En router får hjelp av sin gode venn switchen som filtrerer og videresender innad i nettverket. Routeren kan være koblet sammen med en HUB i tillegg til, eller istedenfor, en switch. En HUB videresender det som kommer inn på alle porter.



Et eksempel på en IP-adresse er vist ovenfor. Den er delt opp i fire oktetter (8 bits hver) hvor hver har en maksverdi på 255.

255.255.0.0

11111111.11111111.00000000.00000000

Ovenfor ser du et eksempel på en nettverksmaske. Masken angir hvilke bits som definerer et gitt subnett (subnett er et nettverk inni et annet nettverk). Bits som er satt til 0 kan varieres for å angi IP-adresser i subnettet. Bits som er satt til 1 angir delen av IP-adressen som definerer hvilket nettverk vertene tilhører. Nettverksmasken består alltid av en sammenhengende serie "1" og deretter "0".

Det er to vanlige måter å notere omfanget av et subnett:

- **Punktnotasjon:** For eksempel 192.168.1.0. Da må man også oppgi nettverksmasken: 255.255.255.0
- **CIDR:** Vanlig punktnotasjon først etterfulgt av hvor mange bits nettverksmasken består av: 192.168.1.0/24

"Ok", tenker du kanskje. "Hva skal jeg med dette?". Hva med å:

Regne med IP-adresser!

Dersom du har IP-adressen til en maskin i nettet og nettverksmasken, kan du vet å gjøre en bitvis AND operasjon mellom IP-adressen og nettverksmasken finne subnettadressen. Vi har:

11000000.10101000.00000001.00000101 AND
11111111.11111111.11111111.00000000 som gir

11000000.10101000.00000001.00000101 = **192.168.1.0**. Dette er den første IP-adressen i subnettet og brukes til å identifisere subnettet.

Ved å utføre en bitvis OR-operasjon mellom en maskin sin IP-adresse og bit komplement (bitvis invers) av nettverksmasken finner du kringkastingsadressen. En melding som sendes til kringkastingsadressen leveres til alle enheter som er koblet på samme LAN. Vi tar utgangspunkt i samme adresse og maske som i stad. Legg merke til at masken er invertert:

11000000.10101000.00000001.00000101 OR
00000000.00000000.00000000.11111111 som gir

11000000.10101000.00000001.11111111 = **192.168.1.255**

DHCP - Utdeling av IP-adresser

DHCP går ut på å automatisk dele ut IP-adresser. Det fungerer på denne måten:

Discovery

En klient (en datamaskin, telefon etc.) bruker kringkastingsadressen på sub-nettet til å spørre om det er noen som kan dele ut IP-adresser på subnettet. Serveren (kan være en hjemmerouter som kjører en DHCP-tjener) vil kunne gjøre dette.

Offer

Serveren svarer (med kringkastingsadressen) at IP-adresse x er tilgjengelig sammen med en liste over andre viktige adresser.

Request

Klienten svarer til alle at den IP-adressen virka kjempefin.

Acknowledge

Serveren ”skriver” opp at IP-adresse x nå er tatt i bruk av klienten og gir beskjed til klienten om dette.

Ord/begreper/konsepter

Jeg begynner nå å havne litt på overtid med dette dokumentet, så videre følger mer direkte forklaringer av ord og konsepter. Dette er da mer for å gi deg som leser noen knagger, og en oversikt over hva ting handler om. Kanskje burde dette vært taktikken min fra starten:)

ARP - Koblingen mellom nettverk- og linklaget

Address Resolution Protocol (ARP) kobler IP-adresser (nettverkslaget) opp mot MAC-adresser (linklaget). ARP fungerer derfor rett og slett som et oppslagsverk.

NAT - Network Address Translation

NAT muliggjør gjenbruk av IP-adresser. Dette ved bruk av private og offentlige IP-adresser. Alle enheter som er koblet opp på samme subnett har forskjellige IP-adresser. Dette er private adresser. Men subnettet i seg selv har også én felles IP-adresse (den offentlige). Så selv som to enheter med forskjellig IP-adresse ber om en tjeneste, ser det for tjeneren ut som om de har samme IP-adresse. Dette gjøres ved hjelp av NAT.

Litt mer TCP og UDP...

UDP tilbyr multiplexing over en IP-adresse og kontroll av nyttelasten (checksum). Multiplexing tillater flere applikasjoner å sende og motta data samtidig ved at signalene samles til ett signal. Før når man brukte telefoner sendtes signalene gjennom fysiske kabler. Istedenfor å ha individuelle kabler fra hver husstand til for eksempel et sentralbord, samlet man signalene i én kabel. Dette er eksempel på multiplexing. En sjekksum har som mål å finne eventuelle feil som kan ha oppstått på pakkene som leveres.

TCP tilbyr det samme som UDP men i tillegg pålitelighet (garantert levering), levering i rekkefølge, flytkontroll (så ting ikke sendes fortere enn mottakeren kan ta imot) og metningskontroll (for å dele nettverksressursene med andre datastrømmer).

DNS - Domain Name System

DNS kobler domenenavn sammen med IP-adresser. Dette betyr at istedenfor å skrive inn IP-adressen til serveren til nrk.no kan man bare skrive nrk.no. Dette gjør ting lettere:)

HTTP-protokollen

HTTP er kort for hypertext transfer protocol. Dette er applikasjonslagsprotokollen for Web. HTTP bruker TCP som transport ved at klienten oppretter en

TCP-forbindelse til tjeneren (dette kalles en socket). Tjeneren godtar deretter TCP-forbindelsen fra tjeneren. HTTP-meldinger utveksles så mellom nettle-seren (HTTP-klient) og Webtjeneren (HTTP-tjener). Til slutt lukkes TCP-forbindelsen. Hva betyr dette? La oss si at jeg skriver inn `www.nrk.no/tv` (jeg er ikke sponsa av nrk (de ville ikke gi meg jobb (det går bra:())) i nettleseren min:

- HTTP-klienten, eller nettleseren min, oppretter en TCP forbindelse ved hjelp av IP-protokollen til en spesiell port på en annen vert (tjeneren (dette er som regel port 80)).
- Tjeneren på `www.nrk.no` lytter etter TCP-forbindelser. Den godtar meg og min forespørsel, og gir beskjed til klienten min om dette (føles godt å bli godtatt av noen).
- Nettleseren sender så en HTTP request message (med URL) til TCP-forbindelsen.
- HTTP-tjeneren mottar forespørselen, lager en response message med det som spørres etter (`/tv`), og sender meldingen med TCP-forbindelsen.
- HTTP-tjeneren lukker TCP-forbindelsen.
- Nettleseren min mottar svarmelding med HTML-fil, og viser frem resultatet for meg. Disse stegene gjentas frem til jeg har funnet Nytt på Nytt og er fornøyd. I alle fall i 20 minutter².

²Til videostreaming så er det strengt tatt UDP-forbindelser som opprettes, men dette er bare en eksempliggjøring av prosessen. Jeg burde dog ha valgt et bedre eksempel, men dette kan ingen av oss gjøre noe med nå.