Nettverk: Historie, repetisjon og lagdeling

Time 1: historie og repetisjon

Time 2: Lagdeling/TCP-IP-modellen

Time 1:

Vi fortsetter der vi slapp i nettverksdelen av pensumet ved å se litt på historisk teknologi som har bidratt til utviklingen av internett og nettverk generelt.

Telegraf:

Blant de mest kjente er morse-telegrafen. Der brukes elektriske signaler til å sende koder, som mottas med en elektromagnet og oversettes til klartekst av et menneske eller en maskin. Det var også dette systemet som ble brukt i Norge, mellom 1855 og 2003.

Telegraf er ikke veldig eksamensrelevant, men det er jo litt gøy, da. Er du interessert i telegraf og hvordan den funker? Her er litt mer info: https://snl.no/telegrafi

Telefon:

Til å begynne med ble telefoner solgt og installert par-vis. En telefon kunne kun ringe én annen. Dette var vanskelig å skalere, så løsningen var å ha et sentralbord der alle som skal ringe kommer og blir koblet videre til riktig sted. Funker litt som en nettverk-switch, egentlig. Dette er heller ikke en optimal løsning, særlig da det var mennesker som utførte oppgavene på sentralbordet. Når det er sagt bruker vi fremdeles samme prinsipp i dag, med at man først når noe sentralt, før man når dit man skal.

Forskjell mellom telegraf og (gammeldags) telefon:

Telegrammer sendes via message switching og telefonsamtaler foregår via circuit-switching.

Circuit switching behøver ingen adresse, siden den krever en dedikert linje, som er tilkoblingsorientert.

Message-switching er ikke tilkoblingsorientert, som betyr at man ikke trenger en dedikert linje, men data sendes i stede fra punkt til punkt i nettverket helt til det når frem dit det skal. Dette betyr at det er flere veier til hver node i nettverket, som er bra i tilfelle linjesvikt. Dette ligner veldig på dagens internett, som bruker packet-switching. Dette er en modifisert versjon av message-switching, der data sendes i mindre “pakker” i stede for å sende alt på en gang.

ARPANET 1967:

Står for Advanced Research Projects Agency, som var organisasjonen bak ARPANET-et. Det var en utvidelse av tjenester som allerede var oppfunnet i mindre skala og verdens første *packet switch-network.* I dag tenker vi på det som en forgjenger av dagens Internett. Utenfor USA var Norge først til å koble seg på. 😊

Standardisering:

Med mange forskjellige nett er det viktig å etablere standarder. Det er takket være standarder at vi i dag kan kommunisere med folk over nesten hele verden via internett.

Internetworking 1973:

Mange av dagens standarder kommer herfra. "Best effort" og stateless routers, som brukes for å koble nettverk sammen og mye mer.

En fin take-away fra time 1 er nok det at fantes ikke noen enkelt oppfinner som ene og alene utviklet teknologien vi bruker i dag. Vi ser at i datateknologi leder en ting til en annen og selv om det utvilsomt har vært flere svært innflytelsesrike individer i den historien har ingen av dem oppfunnet hjulet, men gjerne basert teknologien sin på godt etablerte teorier og systemer.

Repeter:

Klient, tjener, og intermediate systems, som er switch, router

Topologier.

Klient-tjener og peer-to-peer

Time 2:

Protokoll: Definerer hvordan beskjeder skal sendes over nettet.

- Hvordan skal maskinvaren oppføre seg?
- Hvordan skal beskjeden finne frem?
- Garantier for levering?
- Håndtering av kø, tap og andre problemer.

Lagdeling:

Vi har snakket litt om lag tidligere. Å organisere i lag gjør det enklere implementere, bytte ut, troubleshoote og fikse nettverk. Ulempen med lagdeling er at man risikerer å løse noen oppgaver flere ganger.

I in1020 snakker vi mest om TCP-IP-modellen. I tillegg nevner vi OSI-modellen, som har to lag ekstra enn TCP-IP. Denne var en slags forgjenger, men er ikke i bruk og heller ikke er pensum.

TCP-IP-modellen:

Beskjeder sendes fra lag til lag. Når jeg sender en beskjed fra min datamaskin starter den i applikasjonslaget og sendes ned til transportlaget, så til nettverkslaget, deretter linklaget og så til det fysiske laget. Når den jeg sendte beskjeden til mottar den vil beskjeden først nå det fysiske laget deres, jobbe seg opp i lagene helt til det når applikasjonslaget hos mottakeren, der et menneske kan lese av beskjeden.

Man kan også sende beskjeder uten direkte tilkobling. Da har vi et fysisk, link og nettverkslag mellom avsender og mottaker, som beskjeden får gjennom.

Nr	Lag	Funksjon	Navn på dataenhet
5	Applikasjon	Applikasjonsrelaterte tjenester (HTTP, Mail, filoverføring, streaming.. osv)	Data units
4	Transport	Kobler sammen systemene ende-til-ende (TCP/UDP)	Segments
3	Nettverk	Sende data fra ende-til-ende-systemer (IP) Her jobber routern din.	Packets
2	Link	Overføring mellom to noder (LAN/WIFI) En switch jobber her. Både ethernet og WIFI bruker MAC-adresser, som switchen leser av.	Frame
1	Fysisk	Sender bits via kabel eller trådløst Sørger for at bits overføres riktig Enveis-kommunikasjon (Half-duplex) Toveis-kommunikasjon (Full-duplex) *	Bits

*Half-duplex betyr at man kun kan sende data en vei av gangen, mens full-duplex betyr at man kan sende begge veier på en gang. Tenk walkietalkie vs telefon.

TCP brukes til HTTP, mail og filoverføring. TCP har garantier for at pakkene leveres i riktig rekkefølge. Hvis en pakke går tapt blir den sendt på nytt. Dette skjer ved at mottaker sender en “acknowledgement” tilbake, når den mottar en pakke. Hvis en pakke sendes uten at sender mottar en slik annerkjennelse sendes pakken på nytt. Hvis pakken bare var forsinket eller lignende blir duplikater uansett slettet. Tilkoblingen her kalles en 3-way handshake.

UDP brukes til å sende bilde og lyd, slik som zoom og Netflix. UDP har ikke garantier, slik som TCP, siden det ikke er krise om noen pixeler i et filmklipp ikke kommer frem. Dette kalles “best effort"-levering av data. Forbindelsen er tilkoblingsløs.