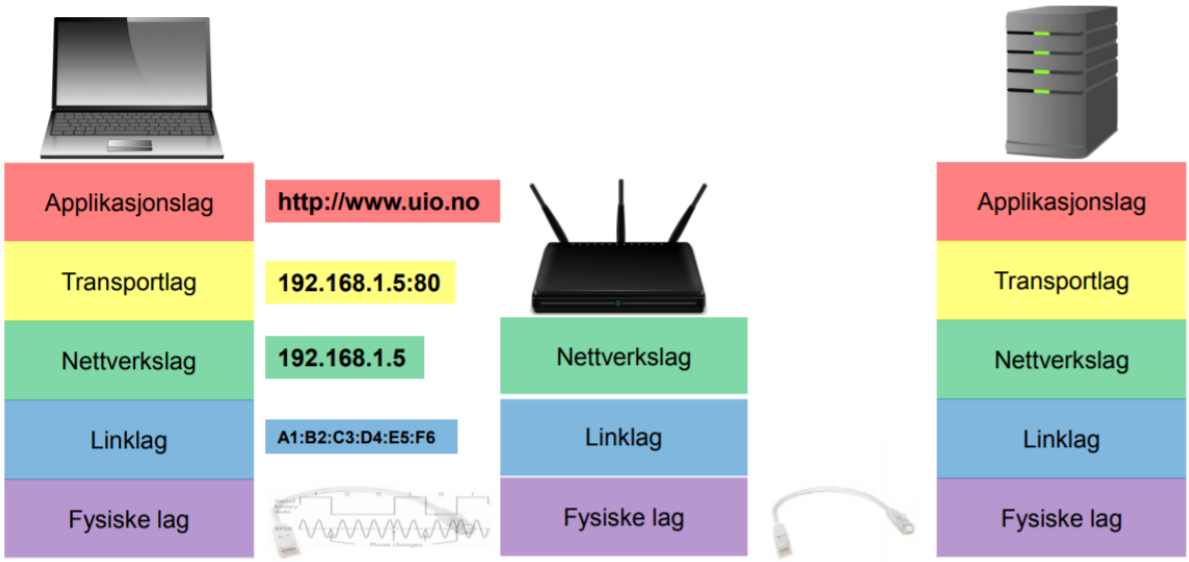
Seminar 11 (tjenester i internett)

Tjenester i Internett

https://s3-us-west-2.amazonaws.com/secure.notion-static.com/97b4311f-22d8-4499-93f3-da1a9d29a972/uke-44-tjenester-i-internett.pdf



Repetisjon: Lag 4 - Transportlaget

- **TCP**: HTTP, E-post, filoverføring, etc.
  - Oppsett av forbindelse (3-way handshake)
  - Pakkene leveres i riktig rekkefølge
  - Pålitelighet – Pakker sendes på nytt hvis kvitteringen (ACK) ikke kommer frem
  - Flytkontroll og metningskontroll
  - Feilsjekking av nyttelasten (sjekksum)
- **UDP**: Strømming av video og lyd
  - Tilkoblingsløs forbindelse
  - Ingen garantier, «Best-effort»

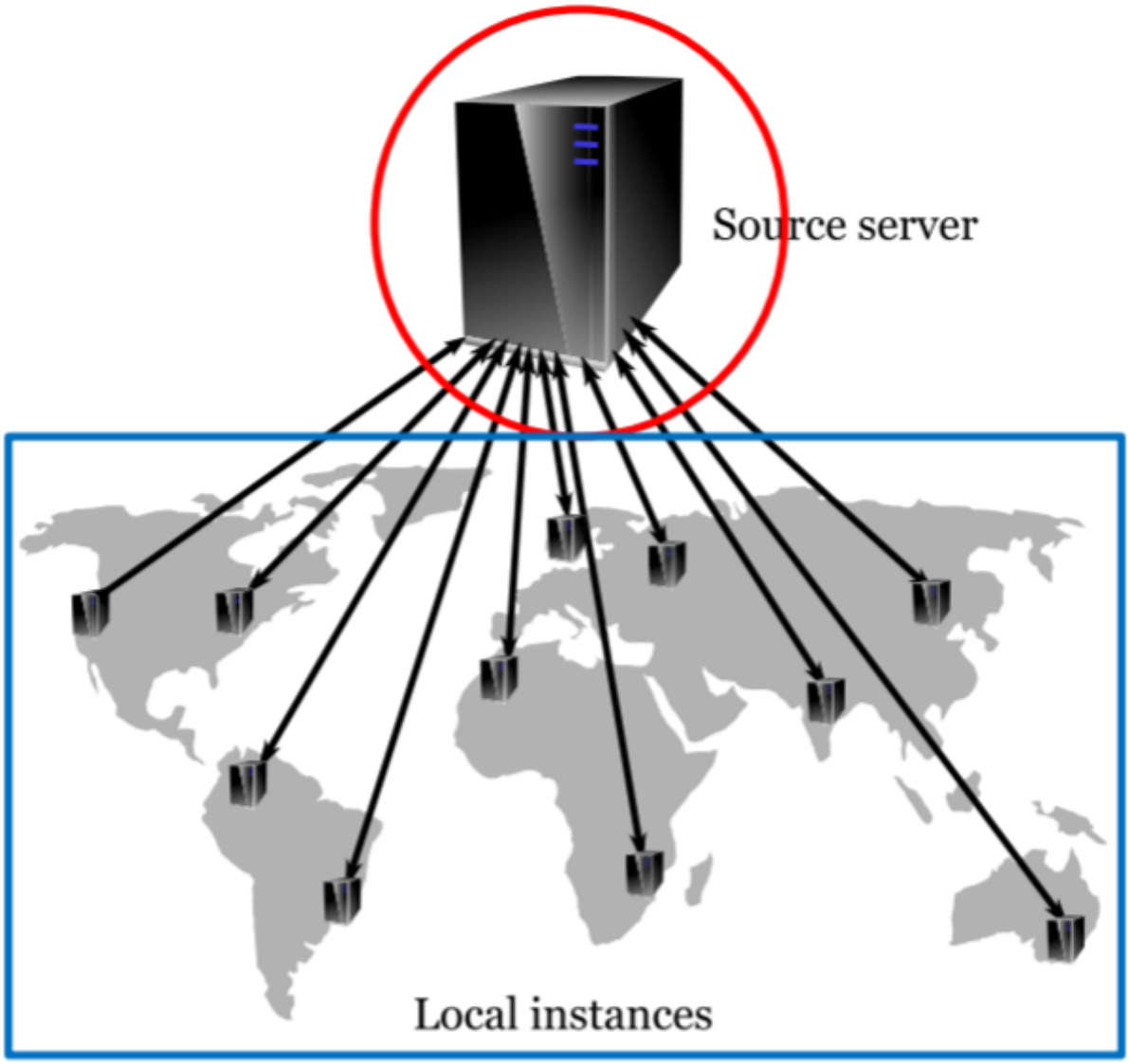
    - Feilsjekking av nyttelasten (sjekksum)
- Bruker «port» som en unik identifikator
    - Representeres med et 16-bit heltall (65535 unike adresser)

Forbindelsesstrategier

- Pull
    - Klienten ber tjeneren om en tjeneste (f.eks et dokument)
    - Eksempel: når du manuelt oppdaterer drivere for ditt skjermkortet, for eksempel
- Push
    - Tjeneren ”dytter” en tjeneste (f.eks en beskjed) til klienten.
    - Krever at det er en forbindelse fra før, eller at klienten lytter
    - Eksempel: når Windows varsler deg om en ny OS update
- Publish-subscribe
    - Variant av ”Push” der tjeneren dytter ut beskjeder til en gruppe av abonnenter (subscribers)
    - Eksempel: Windows varsler ikke bare deg, men alle som har samme OS-versjon

CDN (Content delivery Network)

- Baserer seg på en klient-tjener modell (en til flere)
    - Lokale servere funker som både klient og tjener, fordi de mottar content data fra source server og sender den videre til sluttbrukere.
- Innhold som skal leveres til klienter over hele verden kan repliseres til maskiner som ligger fysisk nær brukerne.
- Sparer kapasitet i backbonenettet
- Gir lavere forsinkelse på forespørsler
- Hindrer overbelastning av tjeneren
- MEN koster ekstra maskinvare og lagringsplass



CDN (en til flere) vs. Peer-to-peer (flere til flere)

Likheter

- Endesystemer kan være både klient og tjener
- Distribuert lagring
- Innholdet nærmere sluttbruker
- Begrenset lagring per endesystem

Forskjeller

- Ingen klare roller på endesystemene
- Ikke noe hierarkisk forhold

- Klienter vet nødvendigvis ikke hvor innholdet er
  - Trenger protokoller for å «oppdage innhold»
- Endesystemer i et P2P nettverk kommer og går

P2P: styrker og svakheter

+

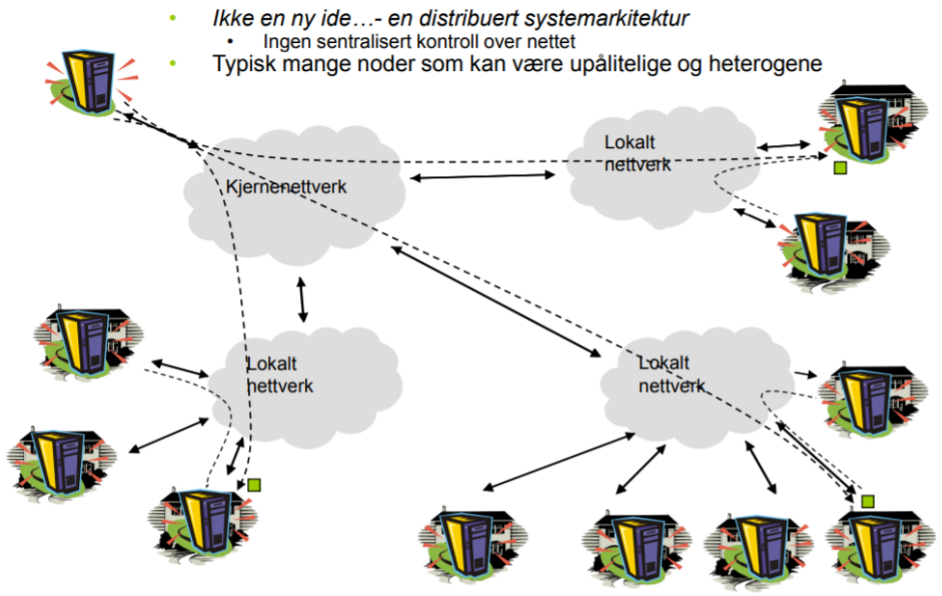
- Ideelt til å dele ressurser mellom enheter (peers)
- Ressurser som dele kan være:
  - CPU
  - Lagring/harddiskplass
  - Nettverk/lagring
  - CPU/lagring/nettverk
  - ... eller nesten hva som helst
- Tilgangen/fordelingen av ressurser styres av algoritmer
- Robust mot endringer i nettverket av peers

-

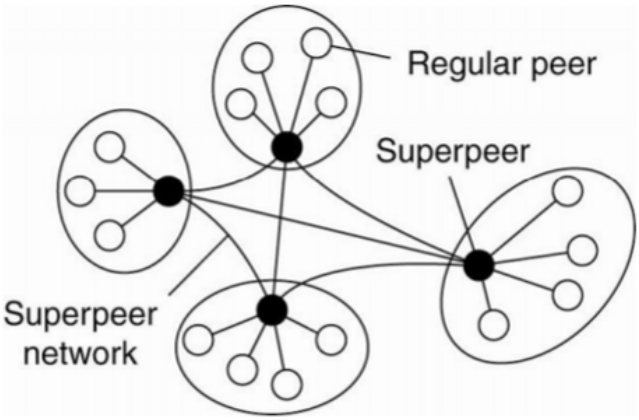
- Egner seg dårlig til applikasjoner som er avhengig av lav forsinkelse.
  - Onlinespill med høyt tempo
- Om man er avhengig av en viss grad av sentral kontroll
  - Hvis det er ingen master-bransjen på Git, så blir det kaos
- Peers kommer og går, så systemet ikke alltid er like stabil som før
- Kan være vanskelig å effektivt indeksere og finne igjen elementer

Varianter av P2P-baserte topologier

- Ren P2P
  - Alle peers er likeverdige



- Hybrid P2P
  - Noen sentrale noder som har mer funksjonalitet enn andre (sentrale tjenere eller ”superpeers”)
  - Kan være valgt fordi:
    - De ikke står bak brannmur og/eller NAT
    - De har mye av en ressurs, f.eks. båndbredde, harddiskplass, CPU regnekraft
  - Slike løsninger kan hjelpe til f.eks. å få ned forsinkelsen i en P2P-løsning

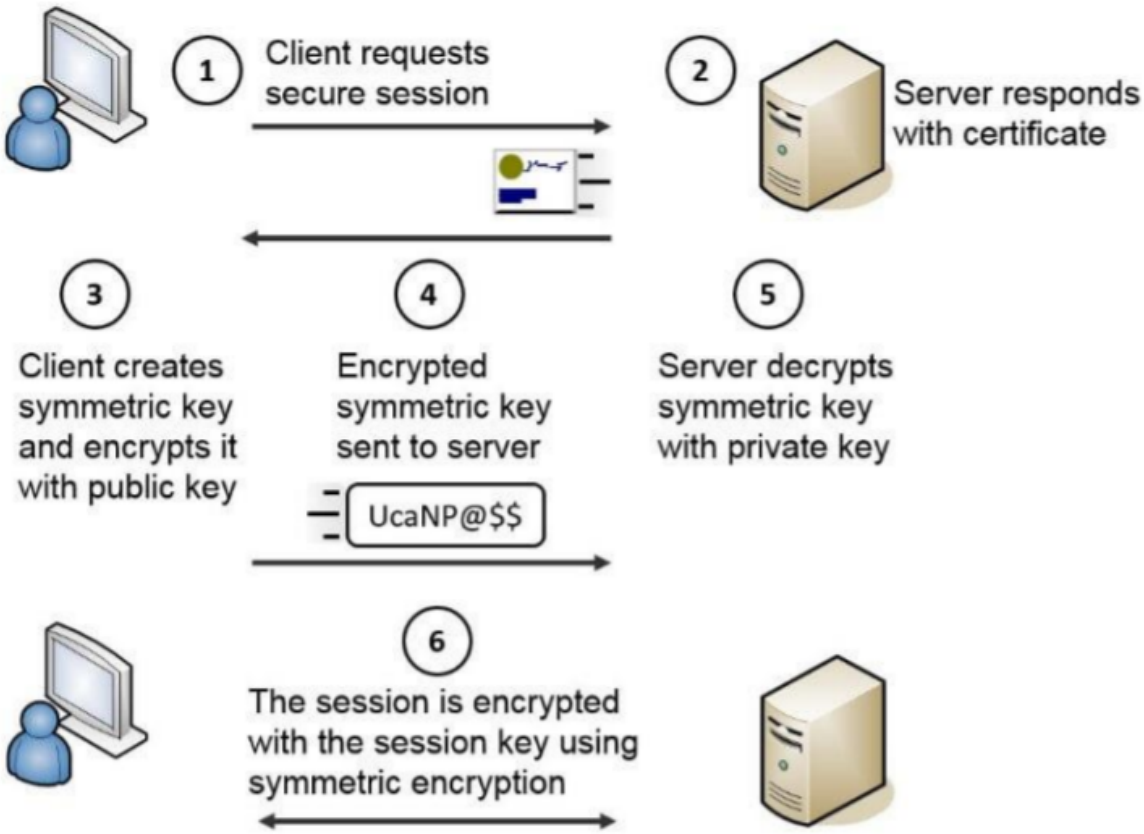


World Wide Web (www): HTTP-protokollen

- HTTP står for HyperText Transfer Protocol
- Applikasjonslagsprotokollen for Web

- Klient-/tjenermodell
  - **klient**: nettleser som spør etter, får og viser web-objekter
  - **tjener**: sender web-objekter som svar på forespørsler
- Fire hovedversjoner:
  - HTTP/1.0 (1990)
  - HTTP/1.1 (1999)
  - HTTP/2 (2015)
  - HTTP/3 (2018)
- HTTP bruker TCP som transport:
  - Klienten oppretter en TCP-forbindelse (socket) til tjeneren, port 80
  - Tjeneren godtar TCP-forbindelsen fra klienten
  - HTTP-meldinger (protokollmeldinger på applikasjonslaget) utveksles mellom nettleseren (HTTP-klient) og Webtjeneren (HTTP-tjener)
  - TCP-forbindelsen lukkes
- HTTP er “stateless”: tjeneren sparer ikke på tilstandsinformasjon om tidligere forespørsler

HTTP med SSL (hybrid kryptering)



Persistente og ikke-persistente TCP-forbindelser

- **Ikke-persistent TCP-forbindelse**: blir lukket så snart et svar er levert.
- **Persistent TCP-forbindelse**: gjenbruker den samme forbindelsen til flere forespørsler.

Datasikkerhet (continuation)

https://s3-us-west-2.amazonaws.com/secure.notion-static.com/a55b88d9-dcc6-4e7a-88c6-b2cf3a2ca0fe/uke-44-datasikkerhet.pdf

Ukeoppgaver

https://s3-us-west-2.amazonaws.com/secure.notion-static.com/40ca06bb-c122-4cbd-9b45-103634d6bb7b/uke45-datakom-sikkerhet-oppgaver.pdf

https://s3-us-west-2.amazonaws.com/secure.notion-static.com/ad51a700-7020-40d5-a668-4945bd779009/uke45-datakom-sikkerhet-losning.pdf