TK1104\_Leksjon\_0x07\_ApplikasjonsLaget

**Applikasjonslaget**

* Hensikten med nettverket er å kjøre applikasjoner som fysisk er lokalisert på flere steder (distribuert)

**Klient/tjener**

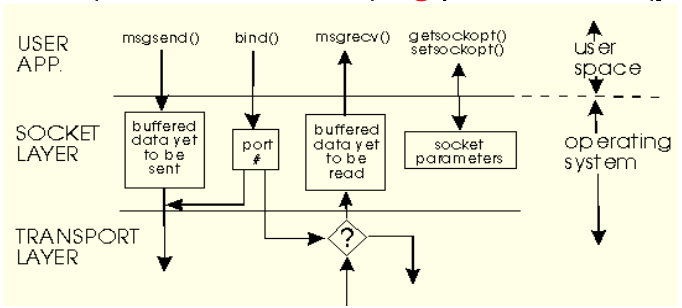
* **Klient**
  + Tar initiativet
  + Ber om en service fra tjeneren
  + På web er klienten browseren
* **Tjener**
  + Leverer etterspurt service til klienten
  + Står ”Alltid på”
  + Har en fast, velkjent adresse
  + Er ”Flaskehals” Fordi alle bruker den samme serveren/server-parken (lastbalansering mulig/nødvendig)

**Peer-to-Peer (P2P)**

* Minimalt/intet behov for at noen alltid står på
* Alle kan både be om og levere tjenesten
* BitTorrent, LimeWire, Skype
* Selv-skalerende
  + I et fildelingsnettverk vil hver klient også øke antall ”tjenere” og samlet kapasitet
* **Noen Problemer**
  + Opphavsrett og fildeling
  + ASDL, kabel, m.fl er laget for **asymmetriske** (klient/tjener) trafikk: mye ned, lite opplalsting. Problematisk for ISPer (internettleverandør)
  + Mange brukere struper opplasting og maksimerer nedlasting, noe som gjør P2P innefektivt.

**Sockets (API) – Application User Interface**

* Definerer forbindelsen (grensesnittet) mellom applikasjons og transport-laget
* **Socket = ”**internett API”
  + To prosesser kommuniserer med hverandre over internett ved å sende data inn i socket og lese data ut fra socket
* Adresse til ønsket kommunikasjons-partner dannes av IP-adresse (vertsmaskin-id) og port-nummer(prosess-id)

****

**Kritiske tjenestenivåer for applikasjoner**

* Tap av data
  + Noen applikasjoner tåler litt tap av data
    - Audio, video
  + Båndbredde /bit-rate (bps)
    - Noen applikasjoner må ha en viss båndbredde
      * Multimedia
      * Caching kan forbedre brukeropplevelsen, men kun dersom man ikke har sanntidskrav
      * Andre kan bruke båndbredden dynamkisk
        + Filoverføring
* Timing
  + Noen applikasjoner tåler ikke mye tidsforsinkelse (latency)
    - Sanntidsprosesser, spill