Universitetet i Sørøst-Norge

6105 Windows Server og datanett

Leksjon 8b TCP/IP del 2: Transportlaget – TCP og UDP

- Transportlagets oppgaver
- Adressering i transportlaget
- TCP – Transmission Control Protocol
- UDP – User Datagram Protocol
- TCP/IP verktøy i Windows

Pensum

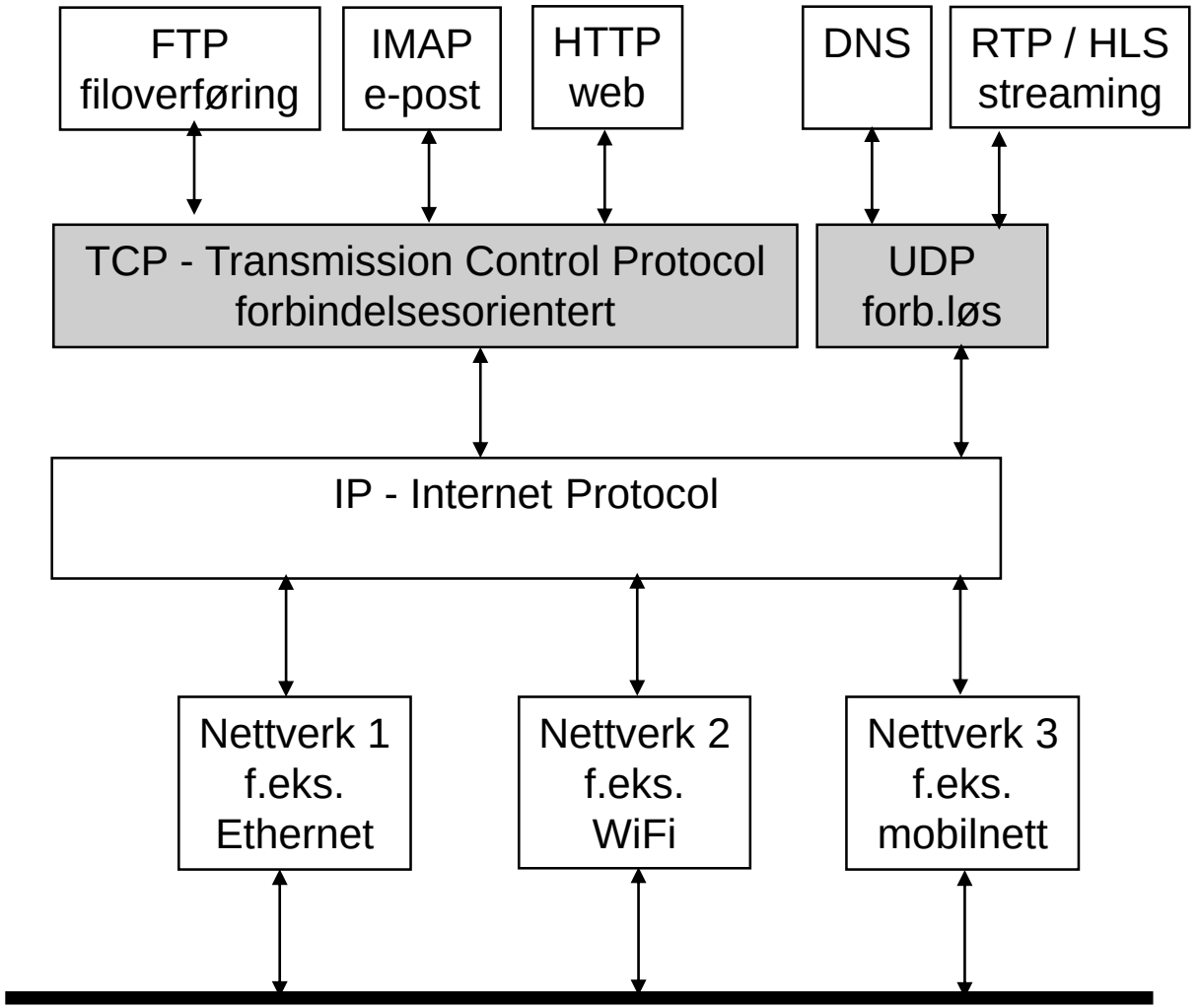
- Kvisli: *Datakommunikasjon og maskinvare*, kap. 3.3 og 3.4

Linker

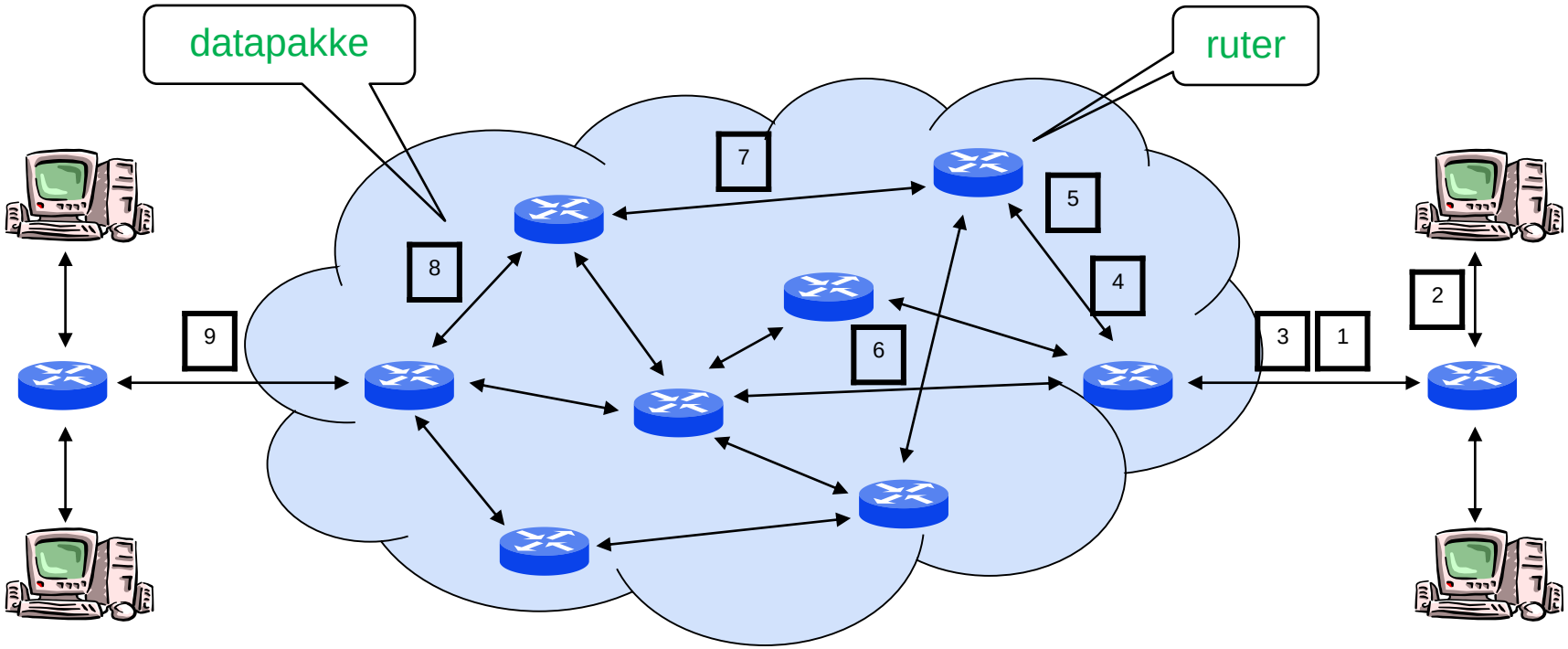
http://www.iana.org/
http://www.iana.org/assignments/port-numbers
http://no.wikipedia.org/wiki/TCP
http://en.wikipedia.org/wiki/Transmission_Control_Protocol
http://en.wikipedia.org/wiki/User_Datagram_Protocol
http://www.hardware.no/artikler/tcp_ip-guide/2296

Transportlaget i TCP/IP

OSI modell	TCP/IP modellen
Lag 5-7	Applikasjonslag app.protokoller
Lag 4	Transport lag
Lag 3	Nettverkslag (Internetlag)
Lag 1 og 2	Link/linjelag



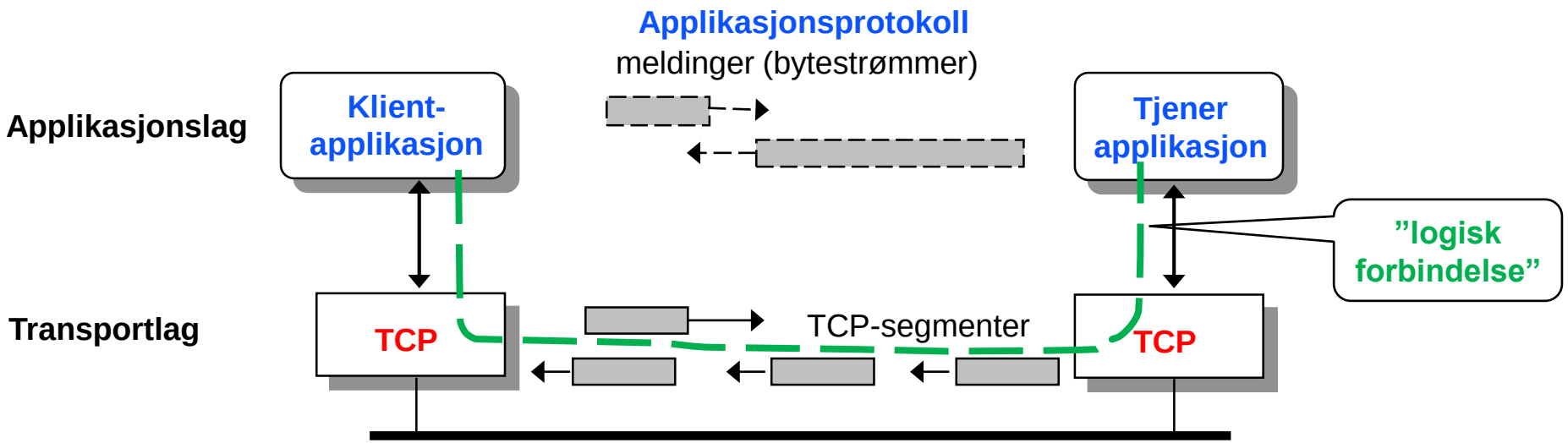
Pakkesvitsjede nettverk (repetisjon)



© Bjørn Klefstad: Innføring i datakommunikasjon

- **Kalles også ”datagramnettverk”**
  - Data overføres som separate pakker med begrenset størrelse
  - Pakkene kan sendes (rutes) forskjellige veier gjennom nettet
  - Rutere sørger for at pakkene sendes "beste" veg gjennom nettet
- **Internett (IP-protokollen) er et pakkesvitsjet nettverk**

Transportlagets oppgaver



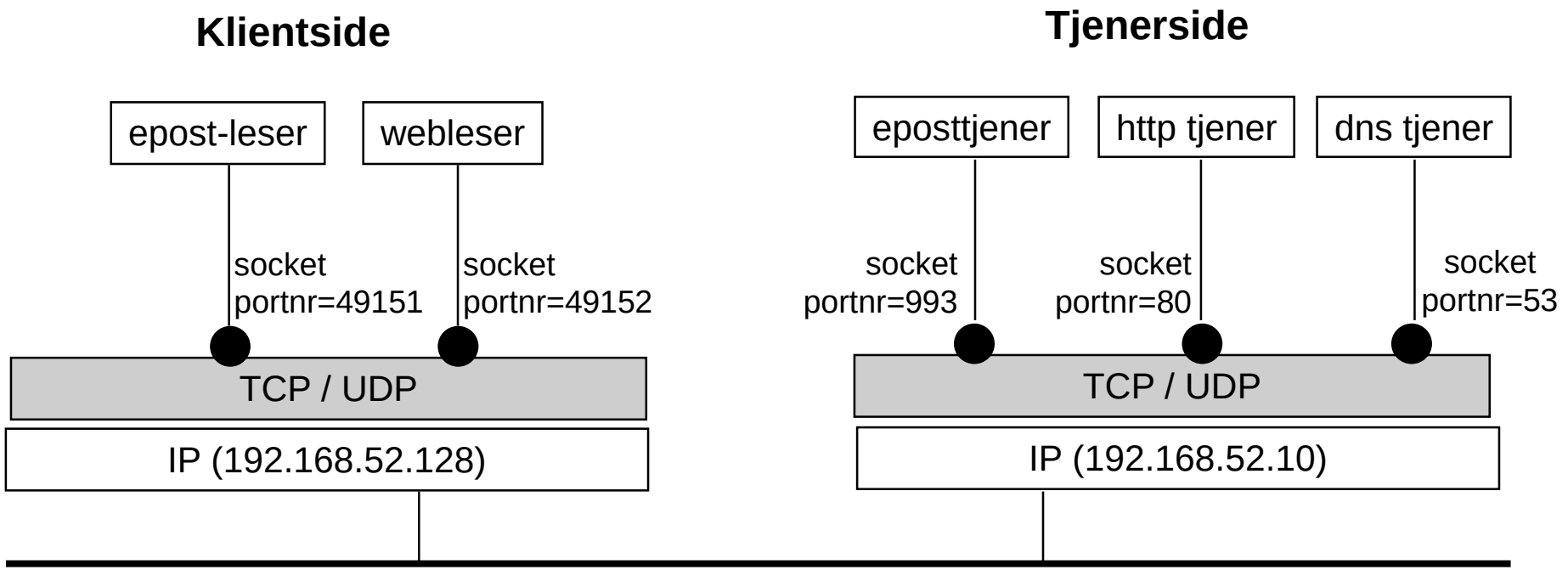
- **”Ende-til-ende forbindelse” mellom applikasjoner, dvs:**
  - Overfører data mellom applikasjoner som om de var direkte sammenkoblet
- **Er avhengig av et underliggende nettverk (nettverkslag)**
  - Nettverkslaget (IP) må kunne bringe segmentne fram til TCP på riktig maskin i nettet
  - TCP er ”bindeledd” mellom applikasjonene og nettverket

Transportlagets oppgaver

- **Applikasjonslaget**
  - Kommuniserer ved hjelp av meldinger (ofte som lesbar tekst)
  - Meldinger er sammenhengende bytestrømmer
  - Meldingene / bytestrømmene kan være svært lange
    - » f.eks. ved overføring av en stor e-postmelding, fil eller html-side

- **Transportlaget (TCP)**
  - Etablerer en logisk forbindelse mellom applikasjonene
  - Deler applikasjonenes meldinger opp i ”segmenter” (pakker), som kan sendes i nettet
  - Adresserer segmentene (med portnummer) til riktig mottakerapplikasjon
  - Garanterer pålitelig og feilfri overføring av data mellom applikasjonene
  - Takler køsituasjoner og varierende overføringskapasitet i nettet

Adressering i transportlaget



Klientprogrammer

- får tildelt portnummer ”dynamisk” av TCP
- vanligvis mellom 49151og 65535
- kalles også ”kortlivede portnummer”:
  - tildeles når forbindelsen etableres
  - "frigjøres" når forbindelsen kobles ned

Tjenerprogrammer

- tjenerprogrammet bestemmer portnummer
- ”lytter” på faste (kjente) portnummer
  - » ”velkjente” portnummer: 0-1023
  - » ”registrerte” portnummer 1024 – 49150
- portnummer er vanligvis fast for applikasjonen

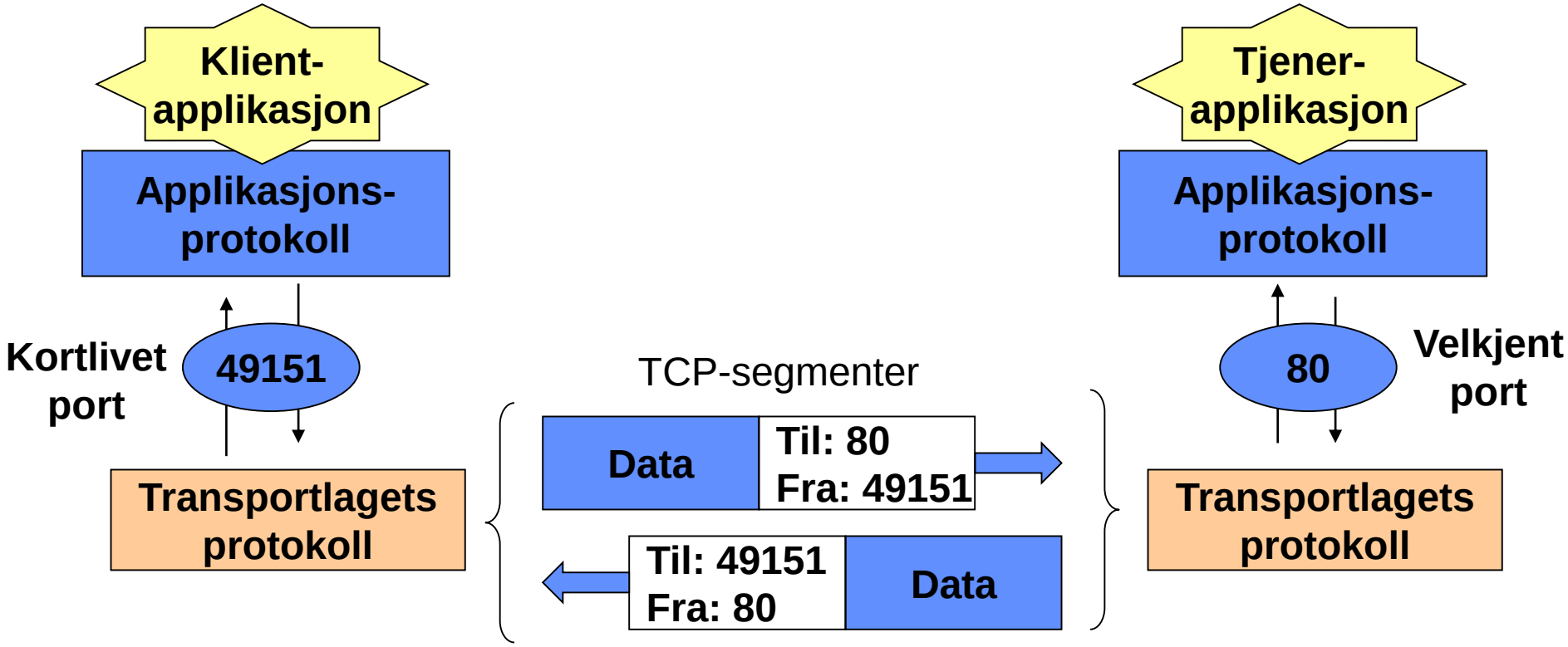
Adressering i transportlaget

- **Flere applikasjoner bruker TCP/IP på samme maskin!**
  - IP-adresser identifiserer hver maskin i nettet.
  - TCP må kunne adressere hver enkelt applikasjon på én maskin.
  - TCP må kunne sende data mottatt fra nettet til riktig applikasjon.

- **Socket / port**
  - Applikasjonene kobler seg til TCP gjennom hver sin "logiske port" (*socket)*
    - » Kan betraktes som et *kontaktpunkt* for applikasjoner mot nettet
  - Hver port (socket) på en maskin har et 16 bits ***portnummer*** (0-65535)
  - En socket adresseres med en ***socketadresse***
    - » Eks: 192.168.52.10:80
    - » socketadresse til webtjener på din Windows Server
  - Portnr kan angis i URL'er i en nettleser
    - » http://server2016.mittdomene.local:80 eller http://192.168.52.10:80
    - » (nettleseren vet at http-protokollen bruker port 80 som standard)
  - Må angi portnr i URL hvis webtjener er installert på et annet portnr:
    - » http://server2016.mittdomene.local:81 eller http://192.168.52.10:81

socketadresse = IP-adresse:portnr

Adressering i transportlaget



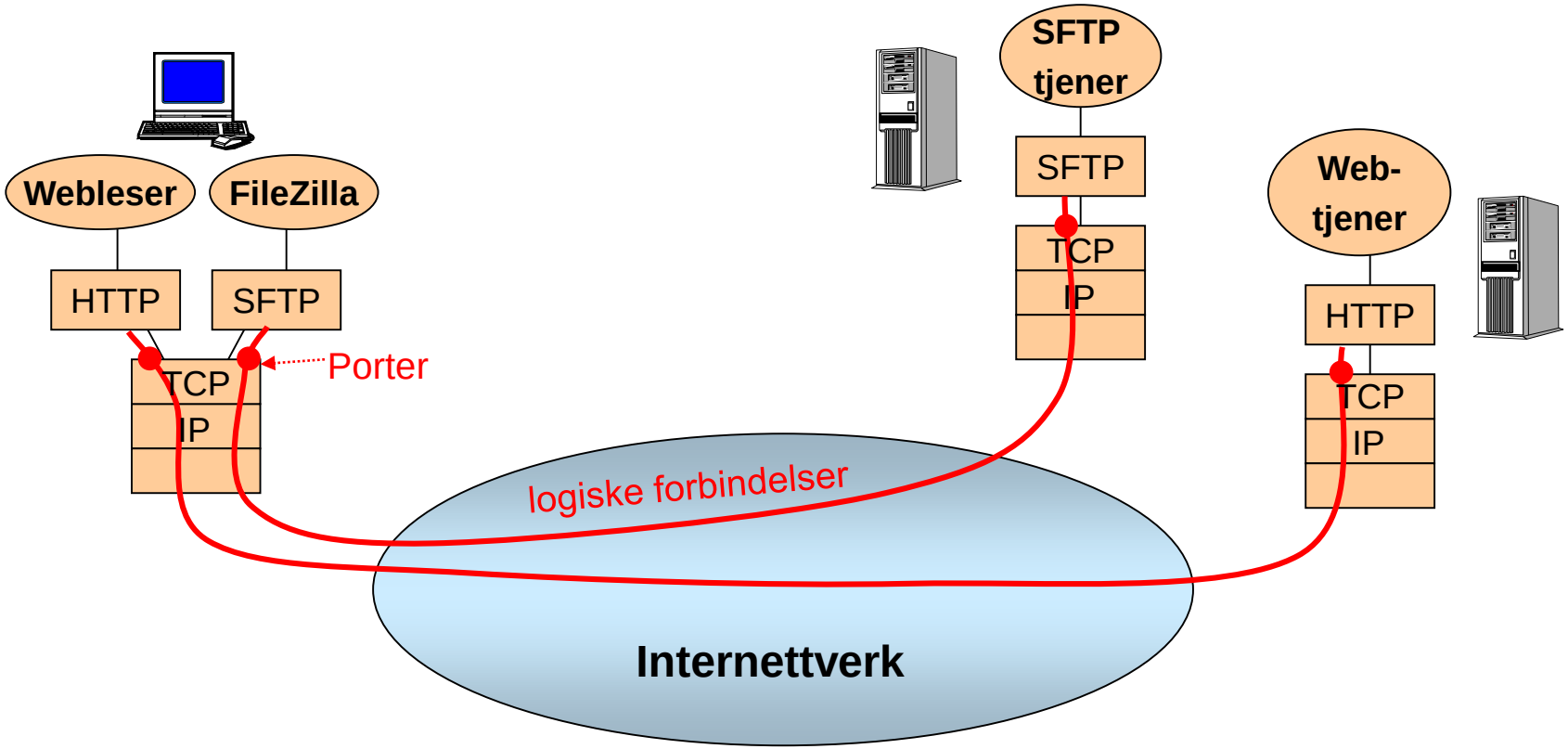
Figur: Olav Skundberg: Innføring i datakommunikasjon, Tisip, 2008

- I pakkenettverk som IP-nett må alle pakker inneholde informasjon om hvor pakken skal og hvor den kommer fra.
- På transportlaget brukes *portnummer* som adresse i pakken (segmentet).

Kjente portnr til noen applikasjoner

Applikasjon/protokoll	Bruksområde	TCP	UDP
BitTorrent	fildeling	6881-6999	
Counter Strike	spill	27020-39	1200, 27000-15
DNS	navneoppslag	53	53
FTP	filoverføring	20 / 21	
HTTP / HTTPS	websider	80 / 443	
IMAP4	e-post	143/ 993	
iTunes	musikk	3689	
Kazaa	fildeling	1214	
LDAP	katalogtjeneste	389	389
MS SQL Server	database	1433	1433
MSN Messenger	chat	1863 m.fl.	1863 m.fl.
MySQL Server	database	3306	3306
Napster	musikkdeling	6699	
Oracle database	database	1521	1521
POP3	e-post	110	
Remote Desktop	fjernpålogging	3389	
SMTP	e-post	25	
World of Warcraft	spill	3724	
Xbox Live	spill	3074	3074

TCP er forbindelsesorientert



Figur: Frode Sørensen: *Innføring i nettverk*, IDG Books Norge, 2011

TCP etablerer *logiske forbindelser* mellom kommunikasjonspartene (klient og tjener)

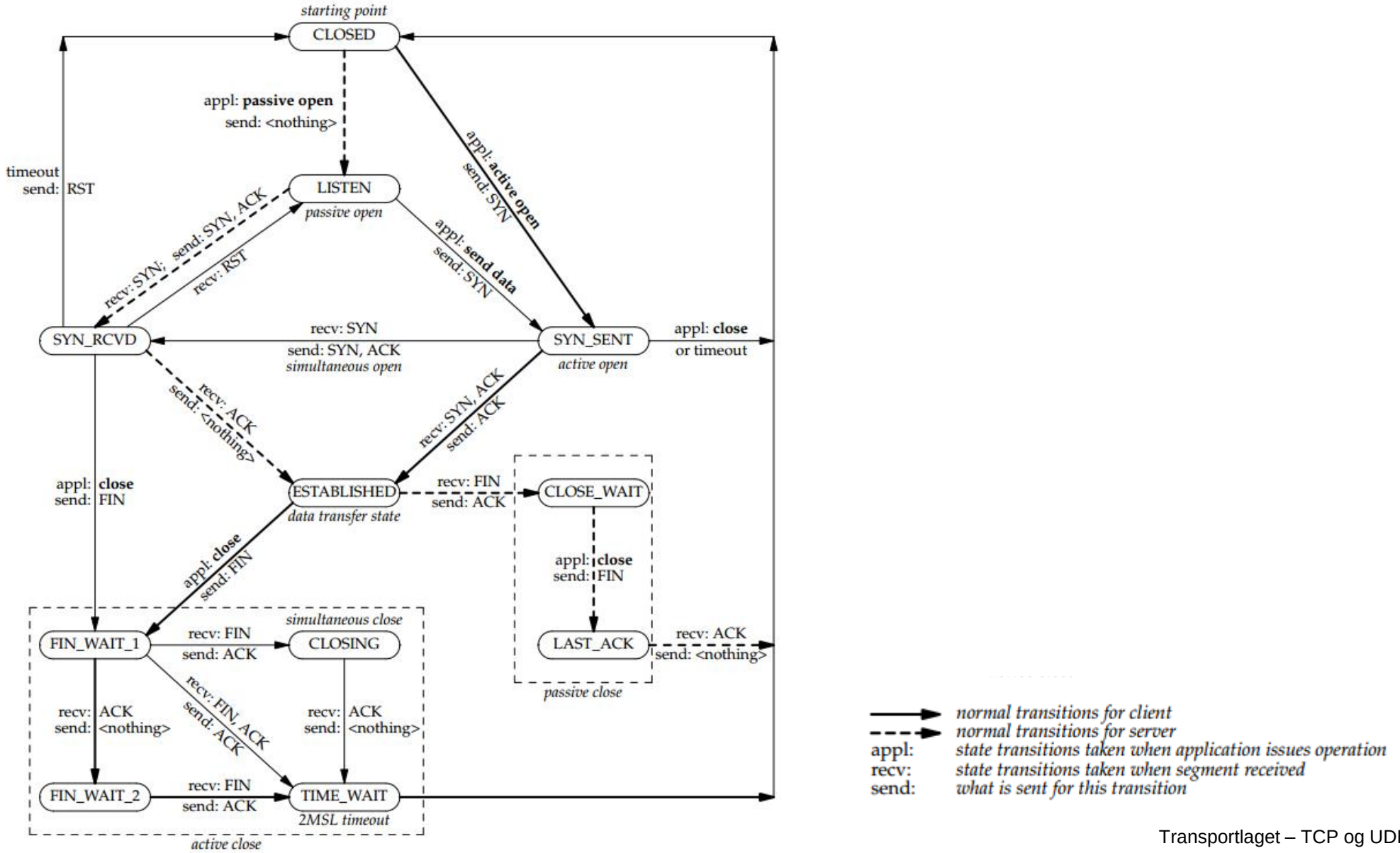
TCP er forbindelsesorientert

- **Logisk forbindelse =**
  - en tenkt kommunikasjonskanal mellom to socketer (porter)

- **Forbindelsen må etableres før dataoverføring kan starte**

- **Tre faser:**
  1. Etablere (åpne) en TCP forbindelse - alltid klienten som initierer
  2. Sende og motta data på forbindelsen - kan vare kort eller lenge
  3. Koble ned (lukke) TCP forbindelsen - klient eller tjener som bestemmer når

Tilstander i TCP

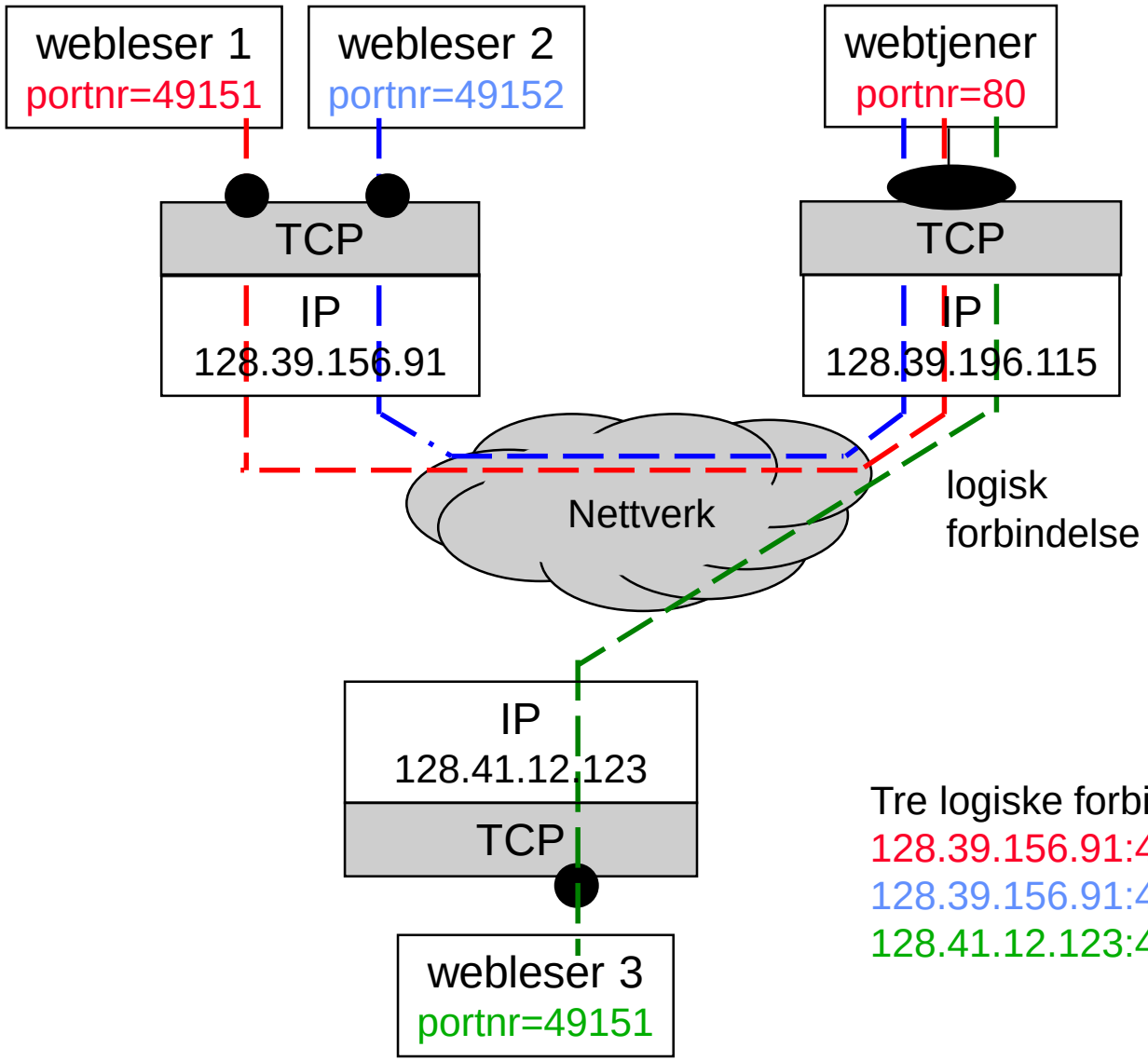


Tilstander i TCP

- **Tilstandskontroll**
  - TCP holder orden på hvilken ”fase” forbindelsene er i
  - Hver TCP socket har en tilstand avhengig av hvilken ”fase” den er i.

- **De viktigste tilstander en TCP-port kan ha:**
  - LISTENING (Server-)socketen lytter på sitt portnr.
Ingen forbindelse er etablert ennå.
  - ESTABLISHED Forbindelsen er etablert (åpen) og data kan sendes.
  - TIME_WAIT Lokal applikasjon har bedt om nedkobling.
Venter til at motparten har mottatt bekreftelse.
  - CLOSE_WAIT Motparten har bedt om nedkobling.
Venter på at lokal applikasjon skal avslutte.
  - CLOSED Socketen er lukket (ikke i bruk).

Logiske forbindelser



Socketpar

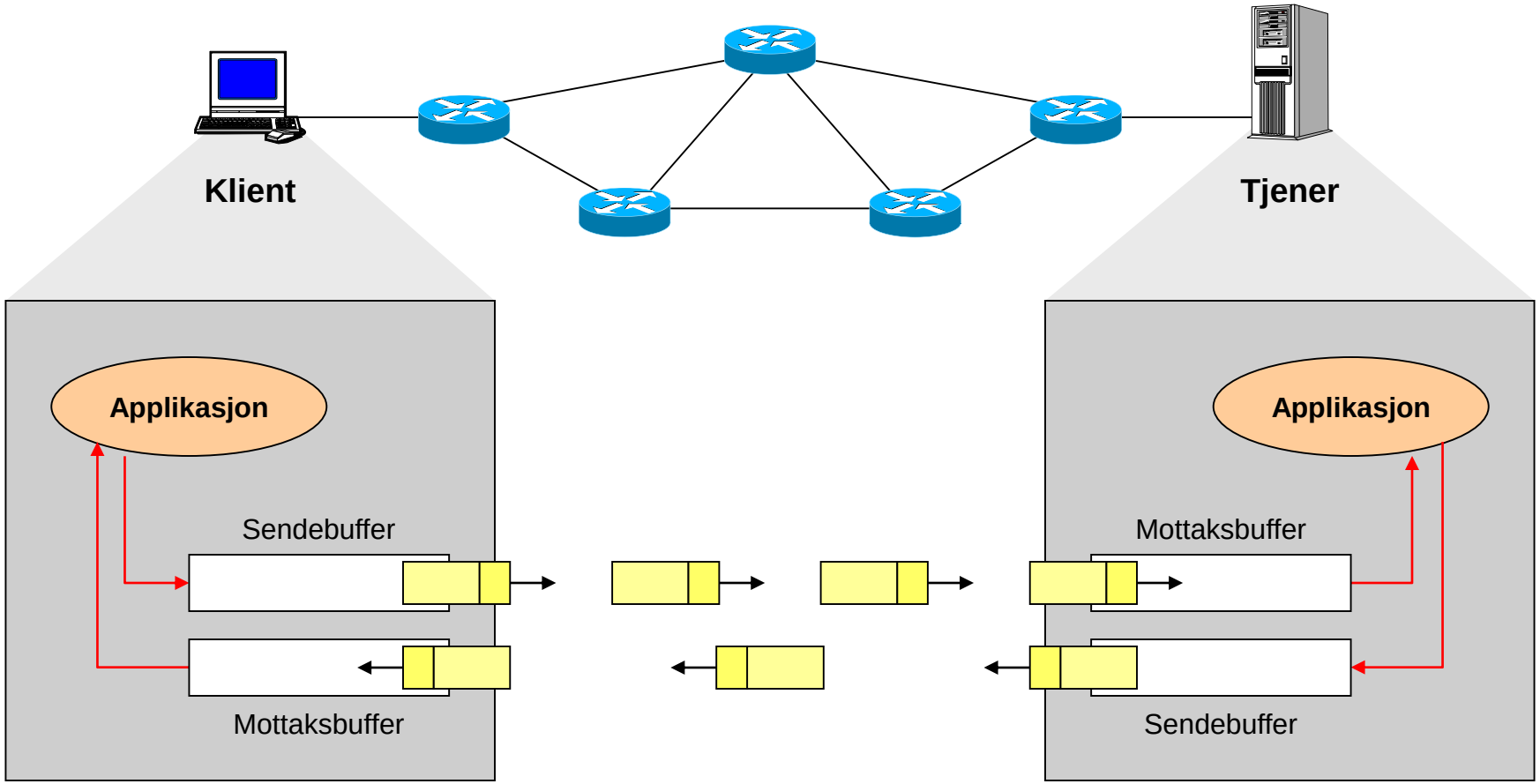
Tre logiske forbindelser
128.39.156.91:49151 - 128.39.196.115:80
128.39.156.91:49152 - 128.39.196.115:80
128.41.12.123:49151 - 128.39.196.115:80

Logiske forbindelser

- **TCP etablerer en *logisk forbindelse* mellom klient og tjener**
  - Forbindelsen identifiseres entydig med et *socketpar*, dvs. socketadressene til klient og tjener
  - Eksempel:
    - » 128.39.156.91:49151 - 128.39.196.115:80
    - » 128.39.156.91:49152 - 128.39.196.115:80
    - » 128.41.12.123:49151 - 128.39.196.115:80

- **En applikasjon kan ha flere logiske forbindelser gjennom samme socket**
  - men til forskjellige maskiner, eller applikasjonsinstanser (porter).
  - Eksempel:
    - » en webtjener kan betjene mange webklienter via port 80.
    - » hver klient har entydig socketadresse.

TCP er to-veis full dupleks



Figur: Frode Sørensen: *Innføring i nettverk*, IDG Books Norge, 2011

TCP er to-veis (full dupleks)

- **En TCP forbindelse tillater sending og mottak av data samtidig**
  - Hver side av forbindelsen har et *sendebuffer* og et *mottaksbuffer*
  - Sending og mottak foregår uavhengig av hverandre
  - Klient og tjener kan sende samtidig om det er ønskelig

- **TCP kan betjene flere logiske forbindelser samtidig**
  - *Multiplekser* flere forbindelser over ett og samme IP lag
  - Hver forbindelse har sitt eget sende- og mottaksbuffer
  - Hver åpen socket bruker derfor noe plass i internminne (RAM)

Oppsummering TCP

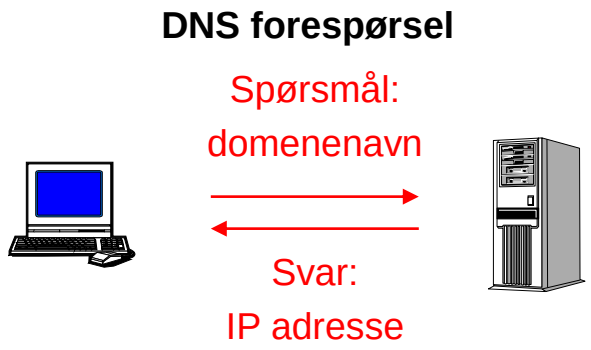
- **Forbindelsesorientert**
  - Klient må etablere forbindelse før dataoverføring
  - Klient eller tjener kobler ned når dataoverføring er ferdig
- **Pålitelig overføring**
  - TCP kan oppdage feil ved å bruke en sjekksum
  - Mottaker sender kvittering for korrekt mottatte data
  - Manglende kvittering = feil i pakken → Avsender sender pakken på nytt
- **Bytestrømorientert tjeneste**
  - Avsender deler opp lange meldinger i pakker (segmenter)
  - Pakkene (segmentene) nummereres med sekvensnummer
  - Mottaker setter sammen pakkene i rett rekkefølge til hele meldinger
- **«Ende-til-ende» transport mellom applikasjoner**
  - Adresserer pakker med portnummer til mottakerapplikasjonen

Forbindelsesløs transportprotokoll

Noen applikasjoner stiller andre krav til transportlaget enn det TCP tilbyr:

Forbindelsesløs tjeneste

- Korte forespørsler som krever raske (korte) svar
- Tidkrevende å koble opp og ned forbindelse
- Eks: DNS forespørsler

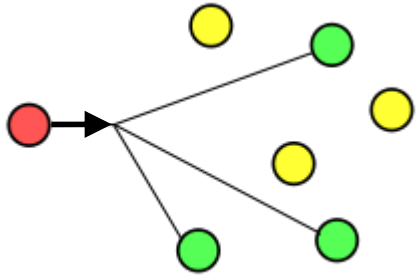


Rask, men ikke feilfri tjeneste

- Noen applikasjoner tåler små feil, men krever rask overføring
- Eks: Streaming av lyd og video, IP telefoni

Multicasting

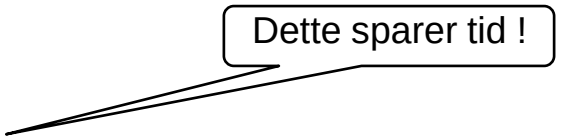
- Noen applikasjoner skal sende fra én avsender til mange mottakere samtidig
- Eks: IPTV



Her er det behov for noe annet enn TCP!

UDP protokollen

UDP = User Datagram Protocol



- **Forbindelsesløs tjeneste**
  - Sender data uten å etablere logisk forbindelse først – ingen "tilstander"
  - Ingen kontroll på om mottakerapplikasjonen ”finnes”, og kan ta i mot pakkene

- **Pakkeorientert tjeneste (mottar pakker fra applikasjonlaget)**
  - UDP kan ikke dele en bytestrøm opp i pakker (segmenter)
  - Applikasjonene må dele bytestrømmen i pakker før de leveres til UDP
  - I noen applikasjoner er datamengdene så små at de får plass i én pakke
    - » F.eks. DNS

- **Adresserer pakkene til riktig applikasjon**
  - UDP bruker samme adresseringsmekanisme som TCP
  - Tjenere lytter på en port (socket) med fast portnummer
  - Klientprogrammet får en port (socket) med ”kortlivet” portnr når det startes.
  - Mulighet for ”multicast-adressering”, dvs. flere mottakere av pakkene

UDP protokollen

- **Ikke pålitelig dataoverføring (garanterer ikke feilfrihet)**
  - Sender data og ”håper det beste”
  - Kan oppdage feil med sjekksum, men retter aldri feilen
    - » kan bruke eller kaste pakker som inneholder feil – ikke tid til å sende på nytt
    - » ber ikke avsender om å sende på nytt – ingen kvitteringer
  - Applikasjonene må rette eventuelle feil, dersom det er ønskelig...

- **Effektiv og rask**
  - Ingen tidsbruk for opp- og nedkobling av logiske forbindelser
  - Lite ekstra tidsbruk til beregning av sjekksummer, kvitteringer o.l.
  - Velegnet for ”tidskritiske sanntidsapplikasjoner”
    - » lydoverføring: IP-telefoni og musikkstreaming
    - » bilde/lydoverføriing: videosamtaler, videostreaming, IP-TV
    - » on-line spill

TCP/IP verktøy i Windows

- **ipconfig**
  - vise eller endre lokal TCP/IP konfigurasjon på maskinen
  - Eks: ipconfig /all – viser all konfigurasjonsinformasjon om alle nettkort
- **ping**
  - Sjekker om en maskin kan nås på nettet og er i stand til å svare
  - Eks: ping home.hit.no – ber om svar fra maskinen home.hit.no
- **route**
  - viser innholdet i maskinens lokale IP rutertabell
  - Eks: route PRINT – viser alt innhold i rutingtabellen
- **tracert**
  - sjekker og viser ruten fra lokal maskin til en angitt maskin
  - Eks: tracert home.hit.no – viser alle rutere på vegen fram til maskinen home.hit.no
- **pathping**
  - likner tracert, men viser også statistikk over pakketap (tar lengre tid)
  - Eks: pathping home.hit.no – viser alle rutere på vegen fram til maskinen home.hit.no
- **netstat**
  - viser statistikk over etablerte TCP/IP forbindelser
  - Eks: netstat –a viser alle sockets som er i bruk på maskinen