

UiO : Institutt for informatikk
Det matematisk-naturvitenskapelige fakultet

IN1020 - Introduksjon til datateknologi

Forelesning – 18.11.2021

Oppsummering

Håkon Kvale Stensland



simula

tfj

Nettverksdelen - Pensum

- Relevante kapitler fra boka (se pensumliste)
- Alt presentert på forelesningene
- Ukeoppgaver
- Obligatorisk oppgave 1

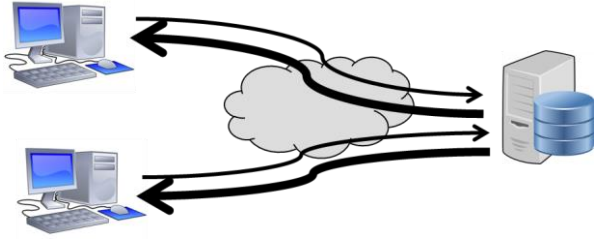
- **NB!** Tema som ikke nevnes i denne oppsummeringen er allikevel pensum!

Protokoller i nettverk

- En protokoll definerer strukturen på beskjeder sendt over et nettverk
- Hvorfor trenger vi protokoller?

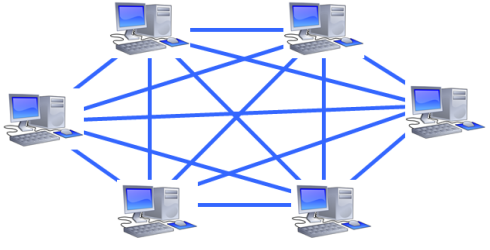
Aksessmodeller: Klient-tjener

- Klienter ber om en tjeneste (oppretter en forbindelse)
- Tjenere leverer tjenesten (svarer på forespørselen)



Aksessmodeller: Peer-to-Peer (P2P)

- Alle noder er likeverdige
- Alle noder kan nå hverandre
- Eierskapet er distribuert



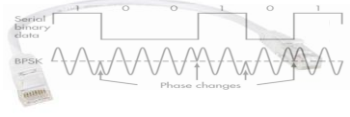
Lagene i Internett (TCP/IP referansemodellen)





Applikasjonslag	**http://www.uio.no**		Applikasjonslag
Transportlag	**192.168.1.5:80**		Transportlag
Nettverkslag	**192.168.1.5**	Nettverkslag	Nettverkslag
Linklag	**A1:B2:C3:D4:E5:F6**	Linklag	Linklag
Fysiske lag		Fysiske lag	Fysiske lag



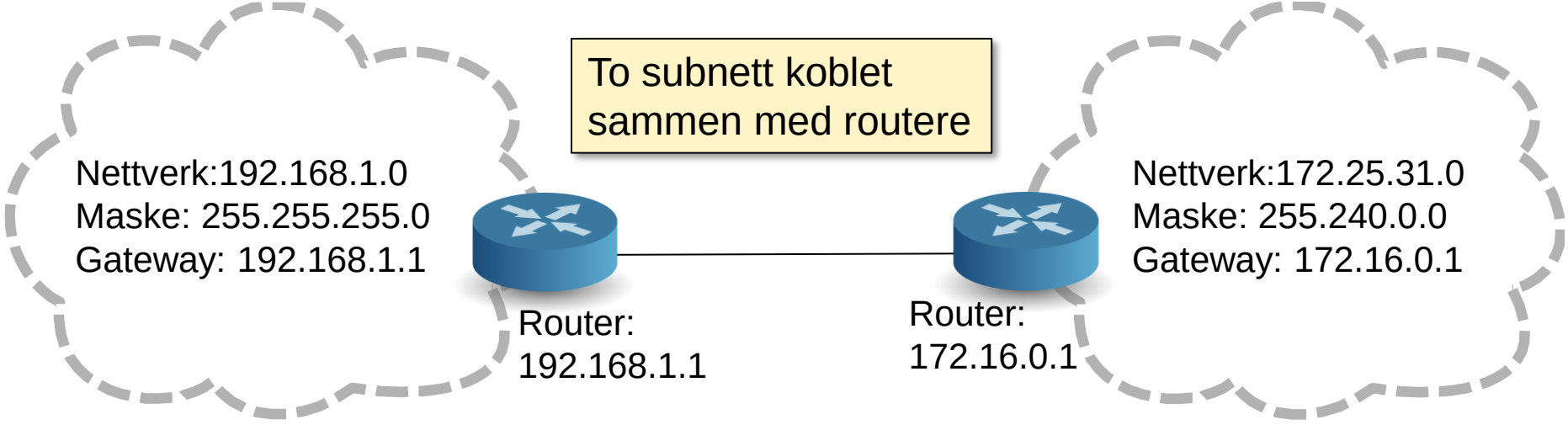




Lokalnettverk (LAN), subnett og broadcast



- Internett er en sammenkobling av mindre, separate nettverk.
- Koblet sammen med switcher og/eller HUBer.
  - Kunne regne seg frem til nettmaske og broadcast adresser.

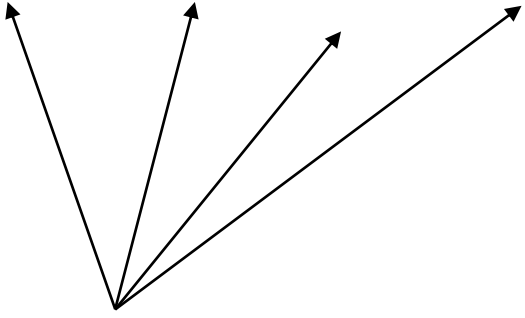


IP-adresser (IPv4)

IP-adresse

192.168.1.5

11000000.10101000.00000001.00000101

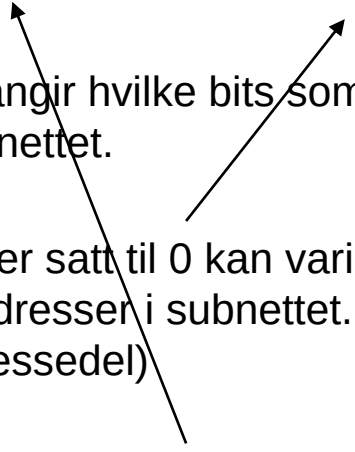


Oktetter:
Består av 8 bits hver. Maks verdi for hver oktett er 255

Nettverksmaske

255.255.0.0

11111111.11111111.00000000.00000000



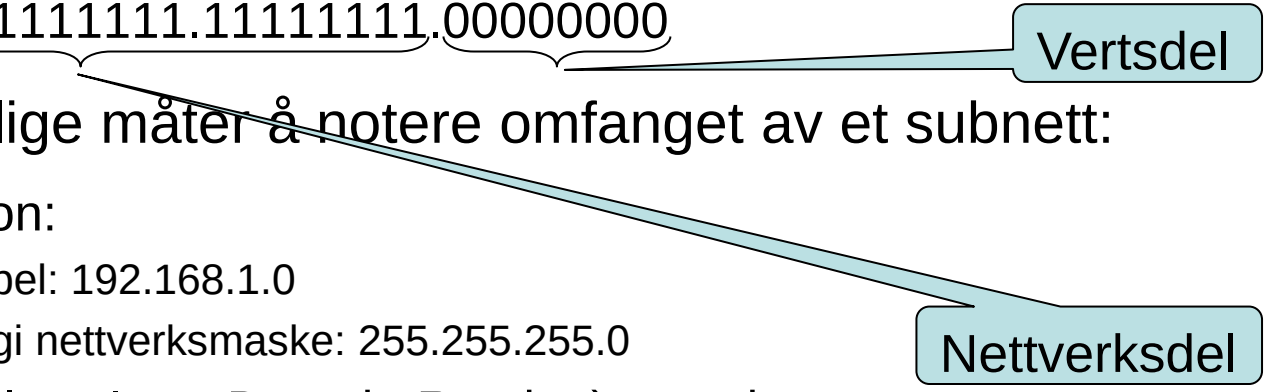
Masken angir hvilke bits som definerer dette subnettet.

Bits som er satt til 0 kan varieres for å angi IP-adresser i subnettet. (vertsadressedel)

Bits som er satt til 1 angir delen av IP-adressen som definerer hvilket nettverk vertene tilhører.

CIDR- og punktnotasjon av subnett

- Nettverksmasken består alltid av en sammenhengende serie "***1***" deretter en sammenhengende serie "***0***"
  - Eks: 255.255.255.0
  - 11111111.11111111.11111111.00000000
- Det er to vanlige måter å notere omfanget av et subnett:
  - Punktnotasjon:
    - For eksempel: 192.168.1.0
    - Må da oppgi nettverksmaske: 255.255.255.0
  - CIDR (Classless Inter-Domain Routing) notasjon:
    - 192.168.1.0***/***24
    - Vanlig punktnotasjon først.
    - Tallet etter skråstreken angir hvor mange bits nettverksmasken består av



Regne ut subnettet fra en IP + nettverksmaske

En maskin i nettet har IP 192.168.1.5 = `11000000.10101000.00000001.00000101`

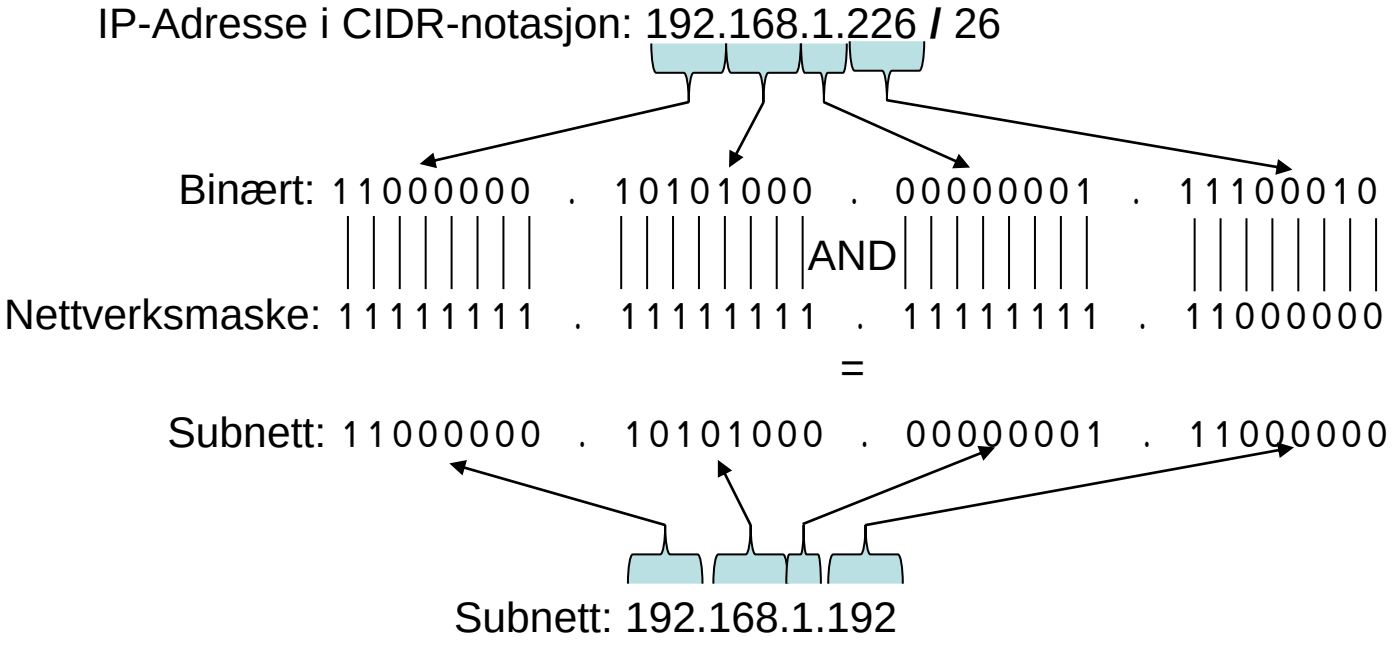
Nettverksmasken er 255.255.255.0 =.`11111111.11111111.11111111.00000000`

For å finne subnettadressen til maskinen må du gjøre en bitvis ***AND**-operasjon* mellom IP-adressen og nettverksmasken.

11000000.10101000.00000001.00000000 = **192.168.1.0**

Dette er den første IP-adressen i subnettet og brukes til å identifisere subnettet.

Eksempel: subnettadresse fra IP / nettverksmaske



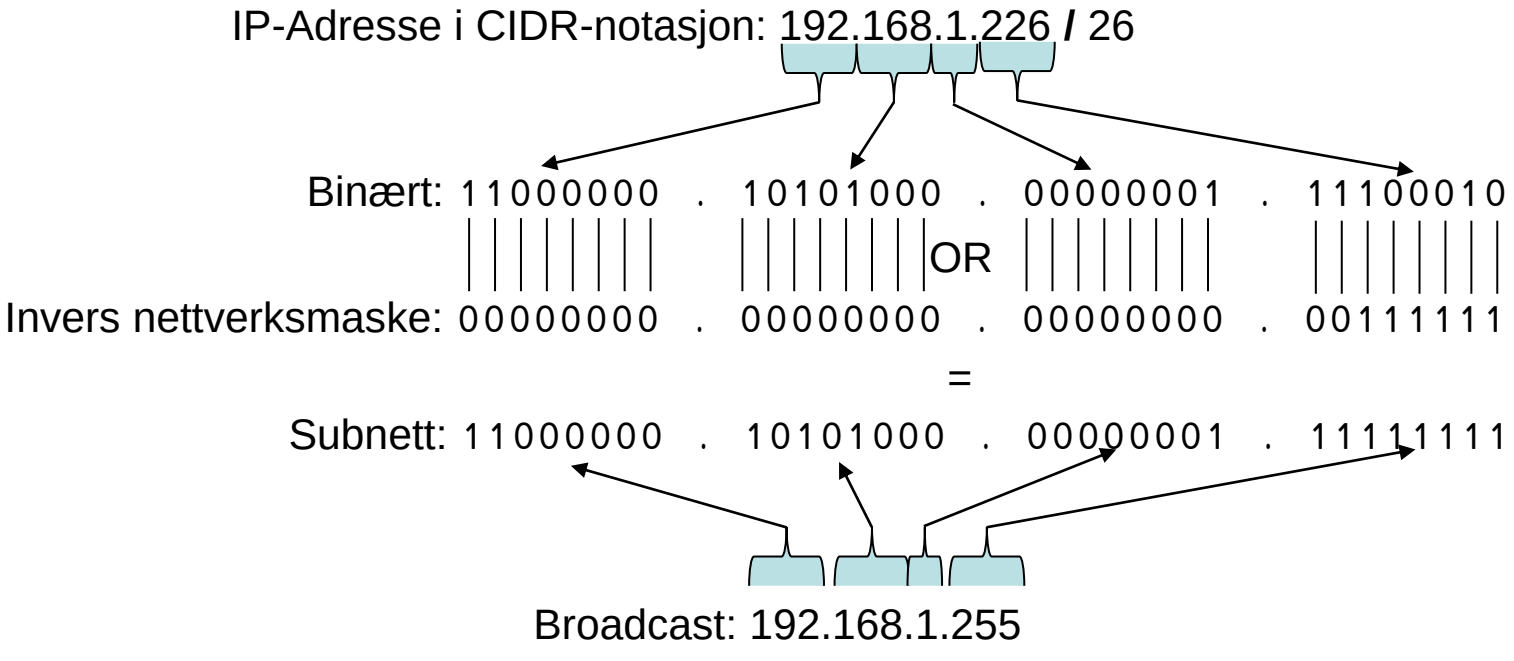
Kringkasting (send til alle)



- En melding som sendes ut på en spesiell adresse.
- Leveres til alle enheter som er koblet på samme LAN (nettverk):
  - Linklaget (MAC): FF:FF:FF:FF:FF:FF
  - IP/Internett: 255.255.255.255

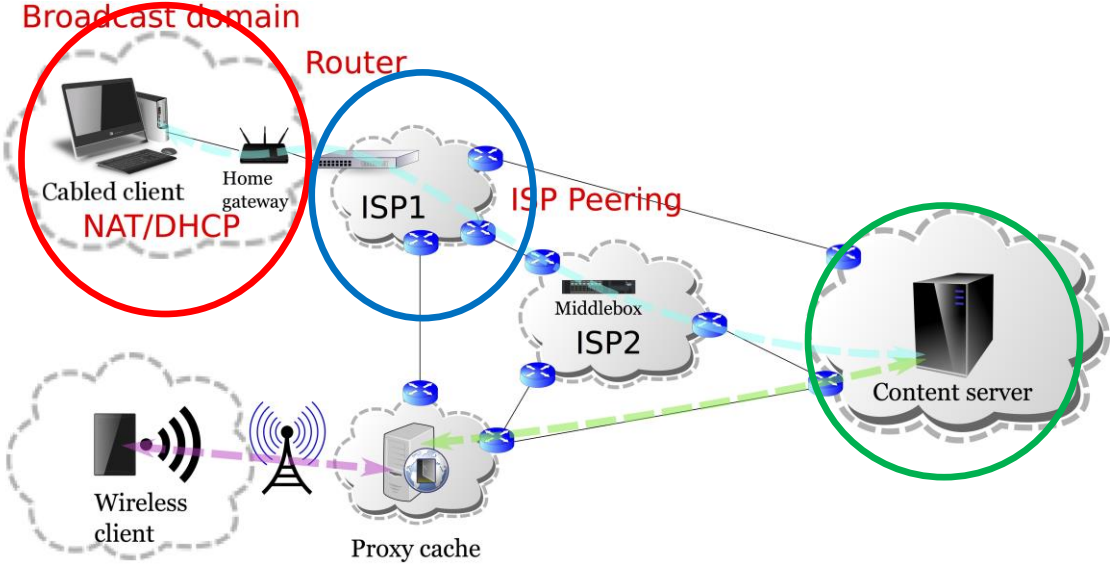
  - For en maskin på et subnett, finner du kringkastingsadressen ved å gjøre en bitvis ***OR**-operasjon* mellom maskinens IP-adresse og bit komplement (*bitvis invers*) av nettverksmasken.
    - Eks: IP-adresse 192.168.1.5 nettverksmaske: 255.255.255.0
    - (192.168.1.5) OR (0.0.0.255) = 192.168.1.255

Eksempel: kringkastingsadresse fra IP / nettverksmaske



Fysisk plassering av innholdet

Cachenivå	Fysisk beliggenhet	Est. RTT
Lokal Proxy	Organisasjon / LAN	< 10ms
Content Delivery Network	ISP	< 50ms
Original datakilde	Internett	~10ms - ~250ms



ARP – Koblingen mellom nettverk og IP

- Kjenne til hvordan ARP fungerer, og hvorfor vi trenger denne protokollen.
- For at IP skal fungere, må avsenderen vite hvilken MAC-adresse pakken skal sendes til.
- Address Resolution Protocol(*ARP*) kobler *IP* (Internett) og *MAC* (Linklaget).



Én IP-adresse – mange porter

Hvordan kan en IP adresse brukes til mange tjenester?
Adressering i transportlaget.

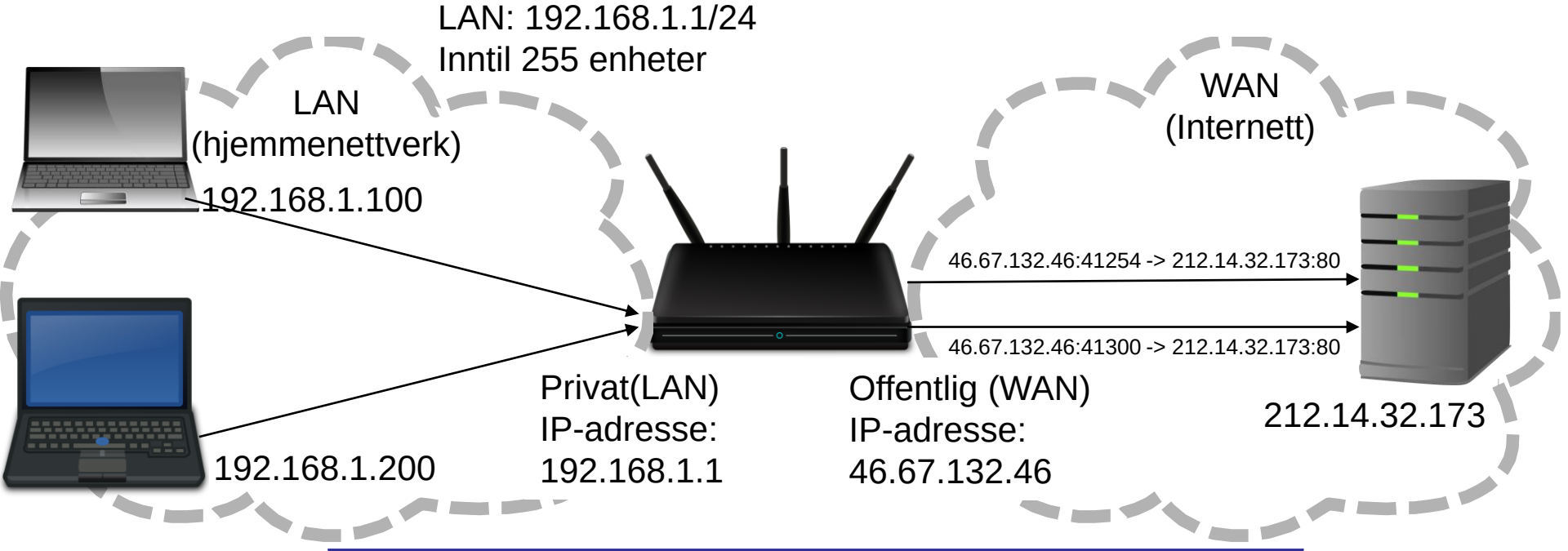


IP: 192.168.1.5

Port	Tjeneste
0	Reservert
1	tcpmux
…	
22	SSH
…	
25	SMTP
…	
1024-49151	Brukerporter
49152-65535	Dynamisk / privat

Transportprotokollene (**UDP, TCP**) implementerer ”porter” som muliggjør totalt 65535 samtidige forbindelser på én IP-adresse

NAT – Network Address Translation



Kilde IP	Mottaker	Oversatt adresse
192.168.1.100	212.14.32.173:80	46.67.132.46:41254
192.168.1.200	212.14.32.173:80	46.67.132.46:41300

Transmission Control Protocol (TCP)

- Forbindelsesorientert
  - Settes opp ved et 3-veis-håndtrykk
    - SYN-SYN+ACK-ACK (se figur)
- Flytkontroll
  - Ikke sende fortere enn mottageren kan ta imot
- *Metningskontroll*
- Byte-strøm og levering i rekkefølge
- Pålitelighet
  - Implementert ved at bekreftelser på hver pakke sendes tilbake fra mottakeren
- Feilsjekking av nyttelasten (sjekksum)

TCP Segment Header Format

Bit #	0 – 7	8 – 15	16 – 23	24 – 31
0	Source Port		Destination Port	
32	Sequence Number			
64	Acknowledgment Number			
96	Data Offset	Res	Flags	Window Size
128	Header and Data Checksum		Urgent Pointer	
160...	Options			

UDP Datagram Header Format

Bit #	0 – 7	8 – 15	16 – 23	24 – 31
0	Source Port		Destination Port	
32	Length		Header and Data Checksum	

User Datagram Protocol (UDP)

- Forbindelsesløst
  - Ikke oppsett av forbindelse på forhånd
- Ingen *flytkontroll* eller *metningskontroll*
  - Pakker sendes ut så fort som mulig
- Ingen garanti for rekkefølgen
- Ingen garanti for pålitelighet
- Feilsjekking av nyttelasten (sjekksum)

TCP Segment Header Format

Bit #	0 – 7	8 – 15	16 – 23	24 – 31
0	Source Port		Destination Port	
32	Sequence Number			
64	Acknowledgment Number			
96	Data Offset	Res	Flags	Window Size
128	Header and Data Checksum		Urgent Pointer	
160...	Options			

UDP Datagram Header Format

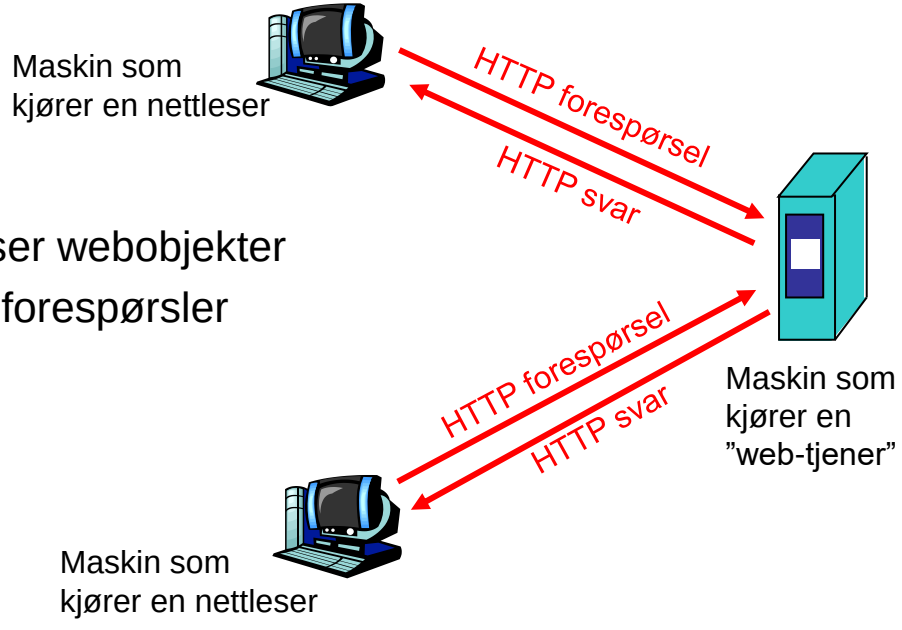
Bit #	0 – 7	8 – 15	16 – 23	24 – 31
0	Source Port		Destination Port	
32	Length		Header and Data Checksum	

World Wide Web (www): HTTP-protokollen

HTTP: **H**yper**T**ext **T**ransfer **P**rotocol

- Applikasjonslagsprotokollen for Web
- Klient-/tjenermodell
  - *Klient:* nettleser som spør etter, får og viser webobjekter
  - *Tjener:* sender webobjekter som svar på forespørsler

- Fire hovedversjoner:
  - HTTP/1.0 (1990)
  - HTTP/1.1 (1999)
  - HTTP/2 (2015)
  - HTTP/3 (2018)



HTTP-protokollen

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”

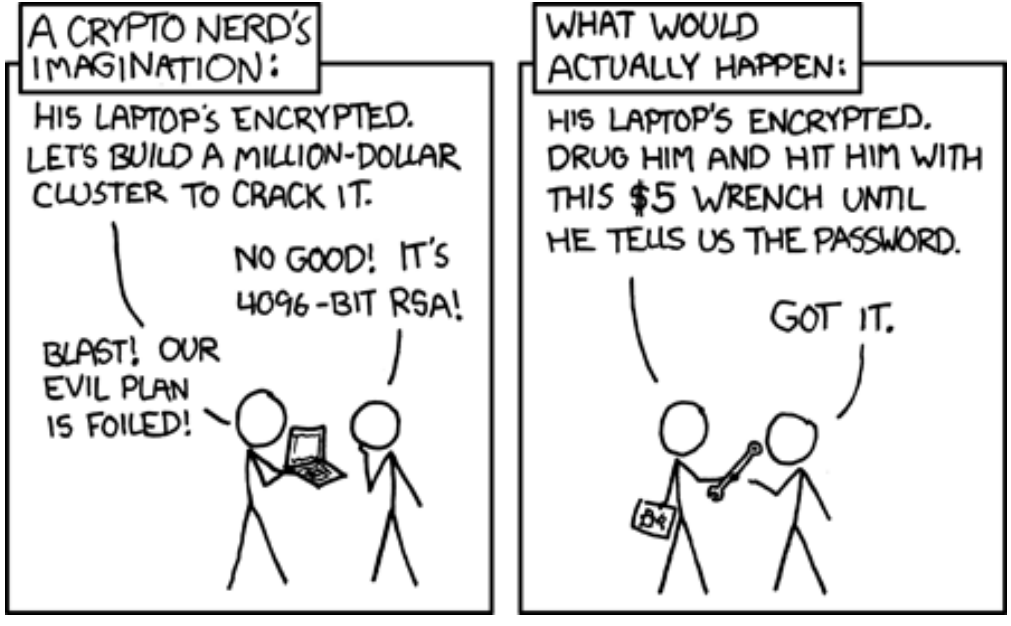
- Tjeneren sparer ikke på tilstandsinformasjon om tidligere forespørsler

Protokoller som sparer på ”tilstand” er komplekse!

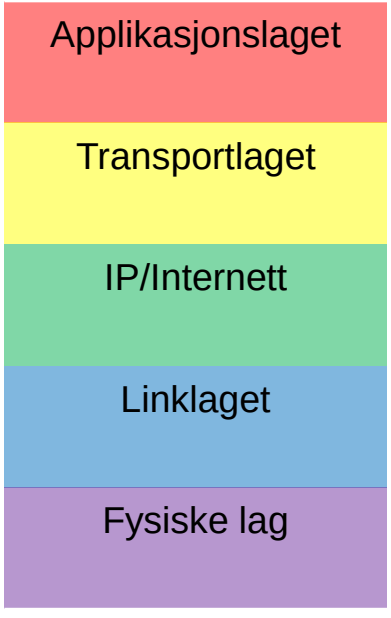
- Tilstanden må vedlikeholdes
- Om en tjener eller klient ”kræsjer”, kan tilstanden bli ulik mellom dem. Da må den gjennoprettes.

Kryptografi

- Hvorfor trenger vi kryptografi?
- Forskjellige teknikker for kryptering:
  - Hemmelig nøkkel (symmetrisk) kryptering.
  - Offentlig nøkkel (asymmetrisk) kryptering.
- Hash-algoritmer.



Kryptering / sikkerhet i nettverket



Secure Sockets Layer – kryptering for ende-til-ende
Applikasjoner – f.eks nettbank eller butikker.

F.eks. *tcpcrypt*– har som mål at alle TCP-forbindelser som settes opp skal være kryptert

VPN (IPSEC etc.) – kobler to subnett sammen så det fungerer som ett LAN selv om de er fysisk adskilt

Kryptering på flere lag gjør det vanskeligere for uvedkomne å lytte til kommunikasjonen.
Adressen (avsender / mottakeren) er vanskelig å kryptere, da routere må vite hvor pakken skal leveres.