

Flere Detaljerte Funksjoner i Datanett

Tor Skeie

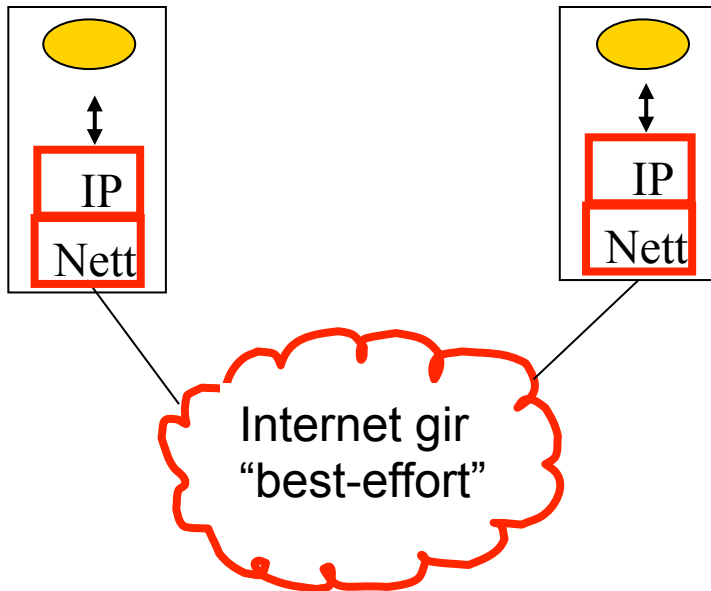
Email: tskeie@ifi.uio.no

Foiler fra Kjell Åge Bringsrud

Ennå litt mer detaljer:

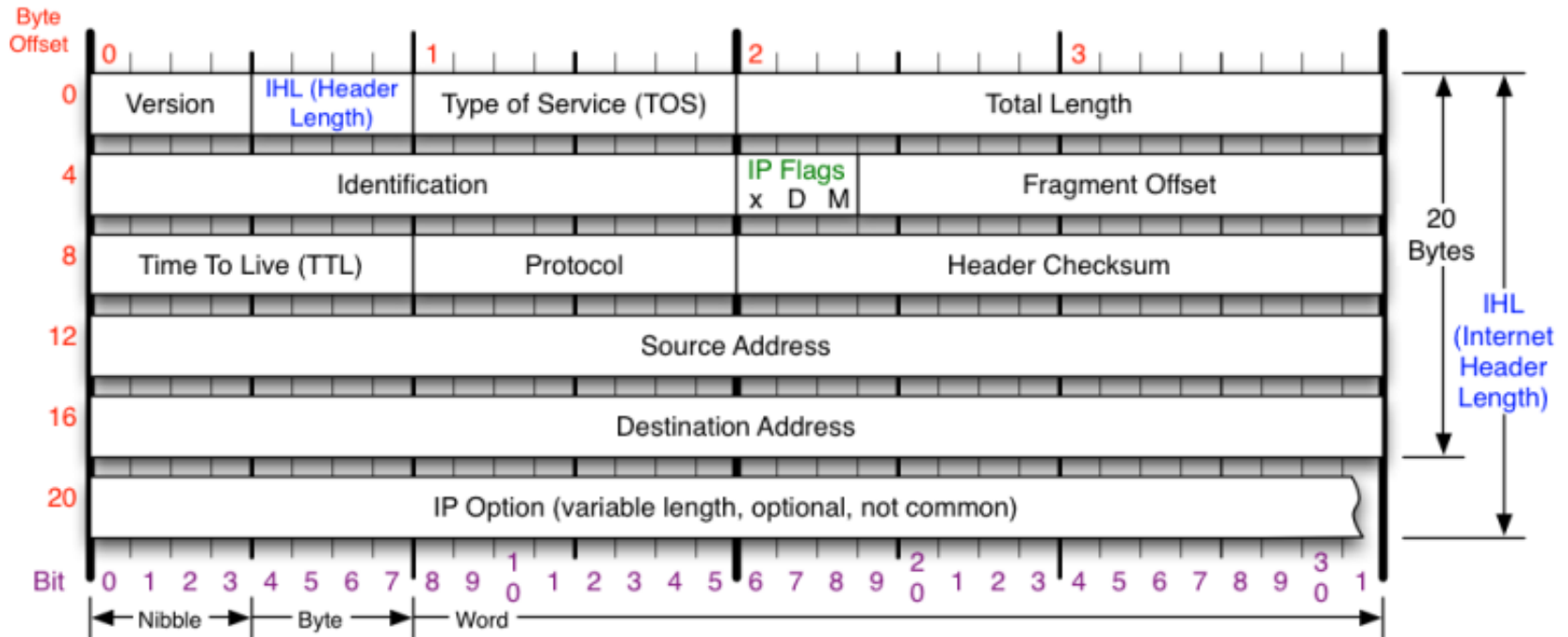
- IP og adressering
- TCP, UDP
- Øvre lag
- Applikasjonsprotokoller

Internet - IP gir “Best-effort” overføring



- Hva menes med “best-effort” overføring?
 - ingen garanti for vellykket overføring:
 - Ingen garantier om båndbredde og forsinkelse gjennom nettet
 - pakker kan bli borte (sjekksumfeil & rutefeil)
 - pakker kan komme frem i gal rekkefølge
 - pakker kan dupliseres

IPv4 header formatet



Version

Version of IP Protocol. 4 and 6 are valid. This diagram represents version 4 structure only.

Header Length

Number of 32-bit words in TCP header, minimum value of 5. Multiply by 4 to get byte count.

Protocol

IP Protocol ID. Including (but not limited to):

1 ICMP	17 UDP	57 SKIP
2 IGMP	47 GRE	88 EIGRP
6 TCP	50 ESP	89 OSPF
9 IGRP	51 AH	115 L2TP

Total Length

Total length of IP datagram, or IP fragment if fragmented. Measured in Bytes.

Fragment Offset

Fragment offset from start of IP datagram. Measured in 8 byte (2 words, 64 bits) increments. If IP datagram is fragmented, fragment size (Total Length) must be a multiple of 8 bytes.

Header Checksum

Checksum of entire IP header

IP Flags

x D M

x 0x80 reserved (evil bit)
D 0x40 Do Not Fragment
M 0x20 More Fragments follow

RFC 791

Please refer to RFC 791 for the complete Internet Protocol (IP) Specification.

IP adresser og vertsnavn

- Vertsnavn (host eller maskinnavn)
 - Mer nøyaktig fullstendig spesifiserte vertsnavn
 - Ser ut som safir.ifi.uio.no
 - Vert safir
 - I subdomene ifi, Institutt for Informatik
 - I domene uio, Universitetet i Oslo
 - I toppnivå-domene no, Norge
- Hvem bestemmer dette?
 - .no - IANA ga det til Uninett
(”Internet Assigned Numbers Authority”)
 - .uio - Uninett ga det til UiO
 - .ifi - USIT, UiO’ s drift, ga det til Ifi
 - safir - IFI drift ga det til maskinen

Adressestrukturer (IPv4)

- Formatet; 32 bit (4 oktetter):
- Uten struktur \Rightarrow problematisk routing
 - globalt unike adresser
 - to-nivå (hierarkisk) Internet adresse-struktur:
 - ✓ Nettverks-nr. + Host-nr.

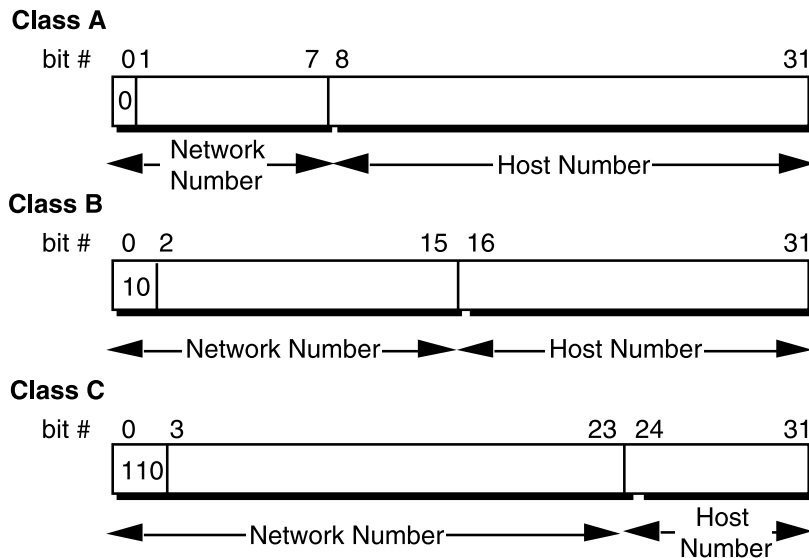


or



Klasseoppdeling av IPv4 adresser

- IP adresserommet er delt opp i tre klasser for å kunne støtte nettverk av forskjellige størrelser (kjent som “classful addressing”)
- Hver klasse setter en grense mellom nettverk prefiks og host nummer på ulike steder i 32-bits adressen



- Klasse A kan definere 126 (2^7-2) ulike nettverk, hvor hver av dem kan støtte 16777214 ($2^{24}-2$) hosts
- Klasse A legger beslag på 50% av det totale IPv4 unicast adresse-rommet
- Klasse B kan definere 16384 (2^{14}) ulike nettverk, hvor hver av dem kan støtte 65534 ($2^{16}-2$) hosts
- Klasse B konsumerer 25% av adressene
- Klasse C kan definere 2097152 (2^{21}) ulike nettverk, hvor hver av dem kan støtte 254 (2^8-2) hosts
- Klasse C konsumerer 12,5% av adressene

Klasseoppdeling av IPv4 adresser

Bit #

1110	Multicast
------	-----------

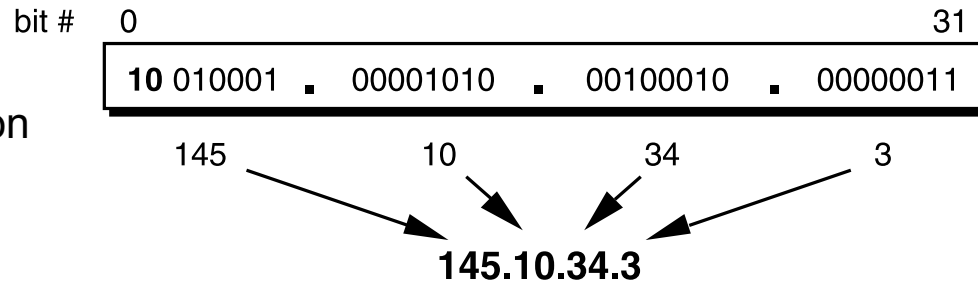
Klasse D

- 224.0.0.1 adresserer alle systemer i et LAN
- 224.0.0.2 adresserer alle routere i et LAN
- 224.0.0.5 adresserer alle OSPF-routere i et LAN

Dotted-decimal notasjon

- Dotted-decimal notasjon deler 32-bit IPv4 adressene inn i fire 8-bit felter (for å gjøre det lettere for oss å lese og skrive)

- Eksempel på en Klasse B adresse gitt med dot-notasjon



- Adresserommet til de tre klassene gitt med dot-notasjon

Address Class	Dotted-Decimal Notation Ranges
A (/8 prefixes)	1.xxx.xxx.xxx through 126.xxx.xxx.xxx
B (/16 prefixes)	128.0.xxx.xxx through 191.255.xxx.xxx
C (/24 prefixes)	192.0.0.xxx through 223.255.255.xxx

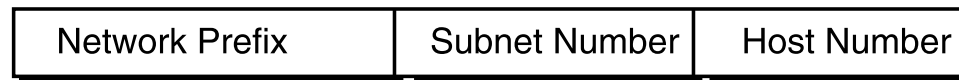
Subnetting

- Subnetting (RFC 950) ble introdusert for å overkomme problemene knyttet til det to-nivå IP adresse hierarkiet:
 - Internet tabellene begynte å bli for store
 - Nettverksansvarlige måtte få en ny (ekstra) IP adresse før et nytt nettverk (segment) kunne installeres
- Derfor ble det lagt til et nytt (ekstra) hierarki i Internet adressen kalt “Subnet Adresse Hierarki”

Two-Level Classful Hierarchy

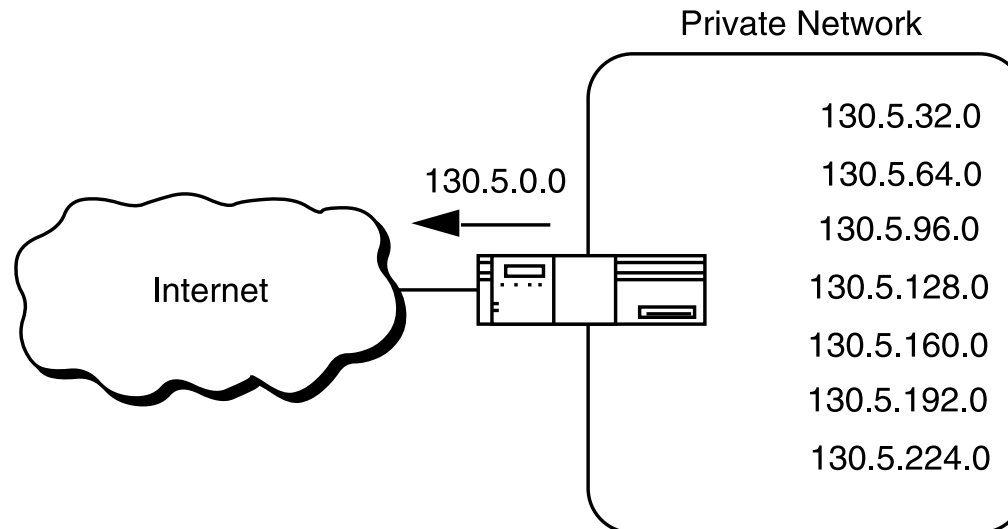


Three-Level Subnet Hierarchy



Subnetting

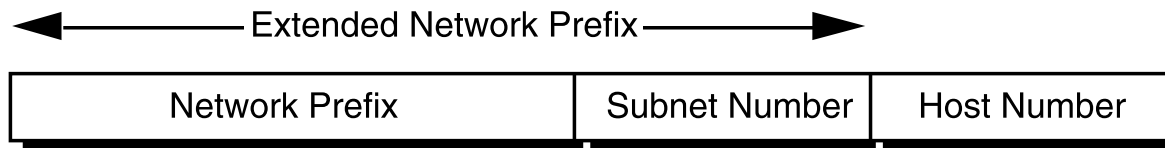
- Med subnetting kan flere logiske nettverk benytte samme IP adresse, f.eks. én enkel /16 klasse B adresse, som vist her:



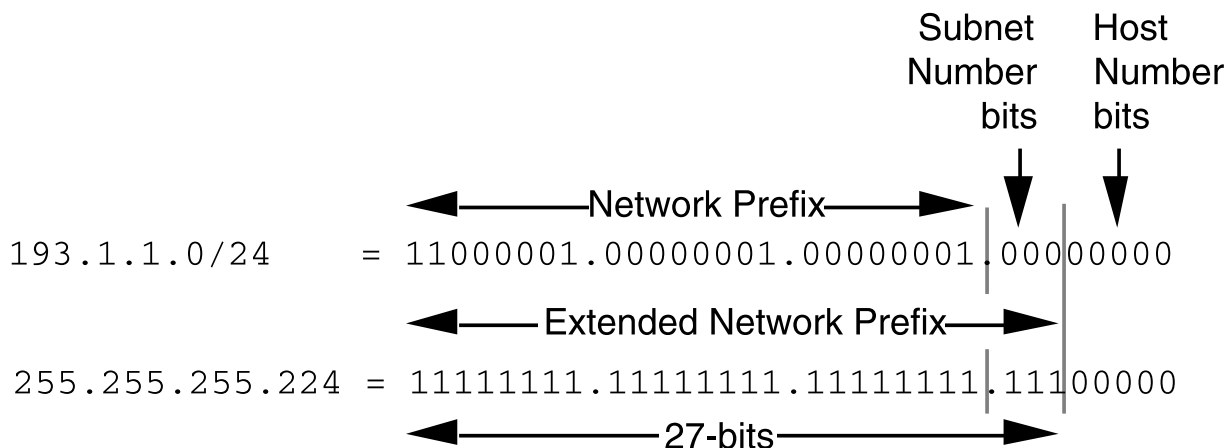
- Router vil akseptere all trafikk fra Internet med adresse 130.5.0.0, og framsende trafikken til de interne subnettverkene basert på den tredje oktetten (subnett nr.) – subnettene (nr.) er bare synlige innen lokalområdet

Utvidet Nettverk Prefiks

- Routere innen det subnettede lokalområdet vil benytte utvidet nettverk-prefiks



- Anta at en organisasjon har IP adresse 193.1.1.0/24 (klasse C adresse) og ønsker seg 6 subnet der hvert subnet kan adressere 30 maskiner
- /24 adressen må da subnettes med tre ekstra bit ($2^3 = 8 > 6$), dvs. det utvidede nettverk-prefikset blir på 27 bit, eller /27
- Et /27 utvidet nettverk-prefiks kan også uttrykkes i dotted-desimal notasjon som 255.255.255.224:



Subnetting eksempel

- Her vises de åtte subnett-nummerene fra eksempelet på forrige foil, hvor den understrekede delen identifiserer det utvidede nettverk-prefikset:

Base Net: 11000001.00000001.00000001 .00000000 = 193.1.1.0/24

Subnet #0: 11000001.00000001.00000001.000 00000 = 193.1.1.0/27

Subnet #1: 11000001.00000001.00000001.001 00000 = 193.1.1.32/27

Subnet #2: 11000001.00000001.00000001.010 00000 = 193.1.1.64/27

Subnet #3: 11000001.00000001.00000001.011 00000 = 193.1.1.96/27

Subnet #4: 11000001.00000001.00000001.100 00000 = 193.1.1.128/27

Subnet #5: 11000001.00000001.00000001.101 00000 = 193.1.1.160/27

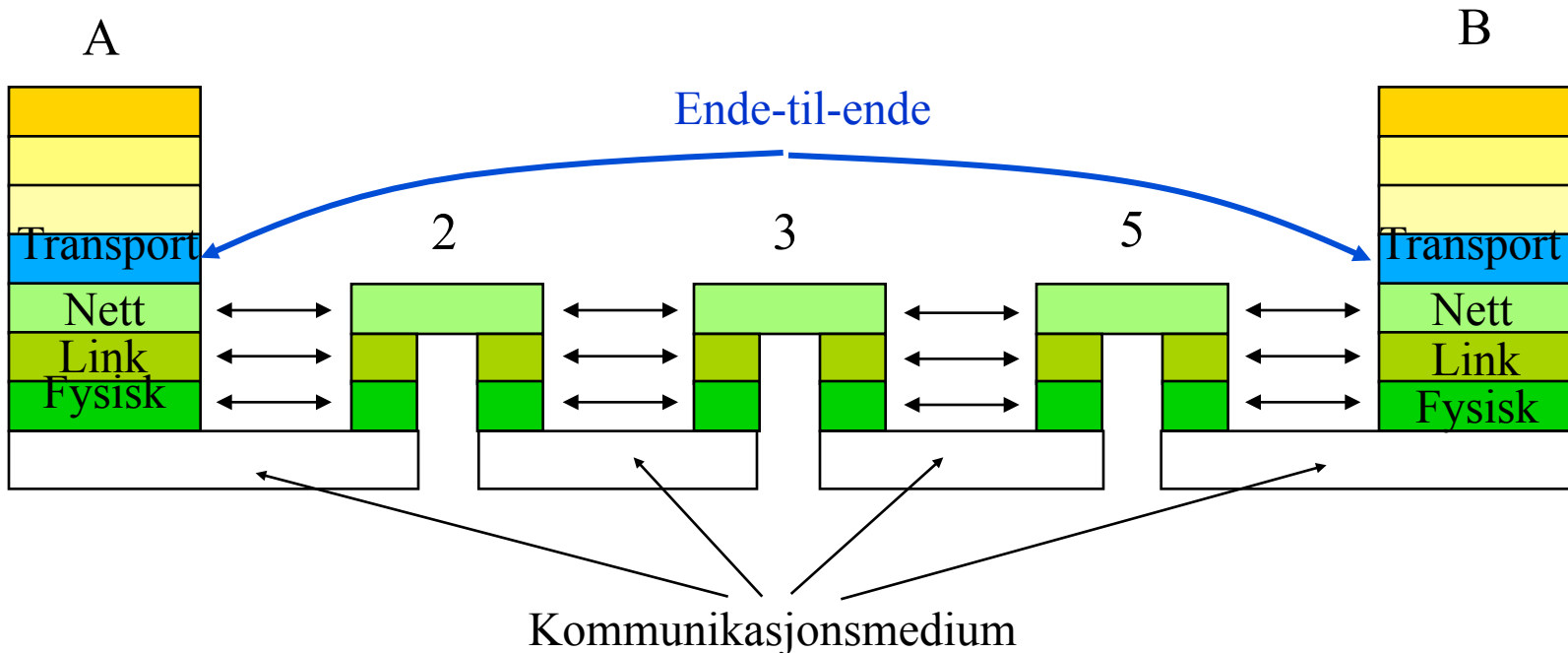
Subnet #6: 11000001.00000001.00000001.110 00000 = 193.1.1.192/27

Subnet #7: 11000001.00000001.00000001.111 00000 = 193.1.1.224/27

- Host nummer-delen består da av 5 bit som gjør at det da kan adresseres 32 (2^5) maskiner innen hvert av subnettene

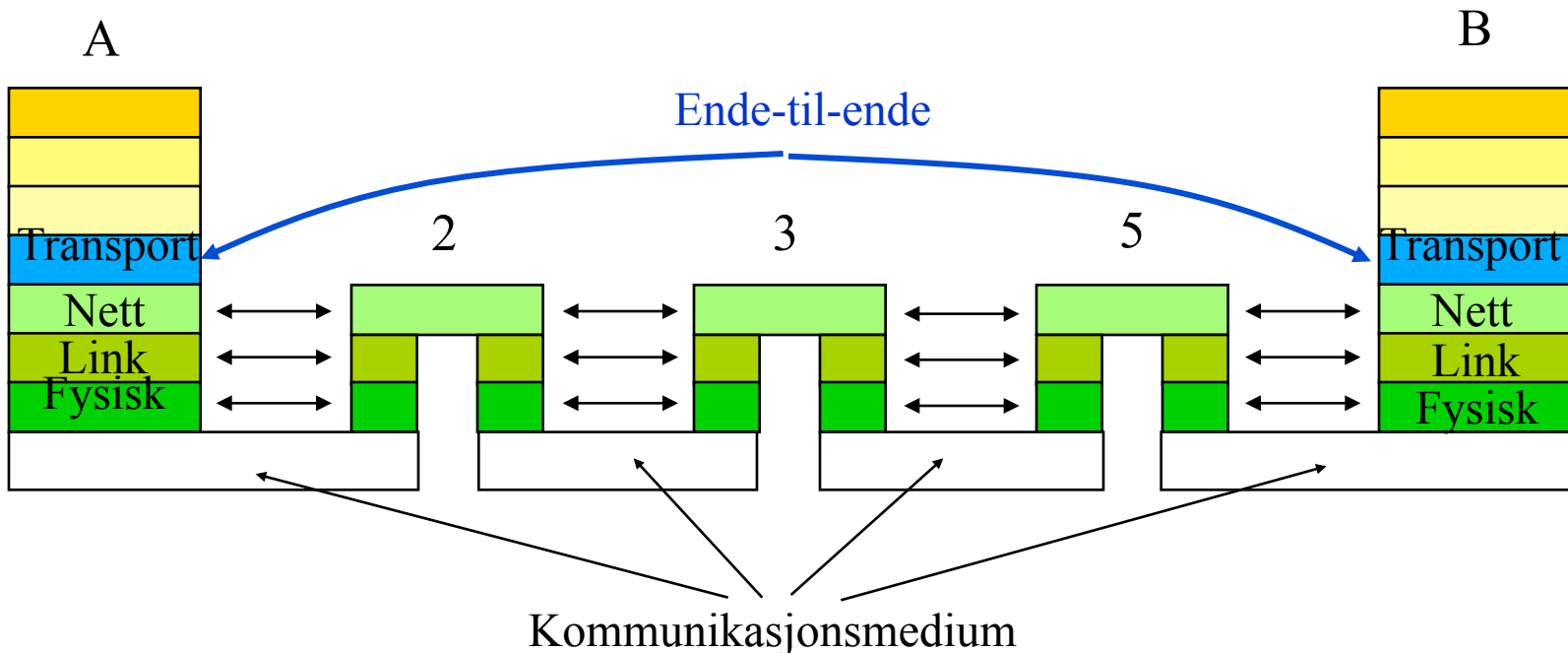
Tilbakeblikk

- Protokollstakk-modellen:
 - lagdelt / abstraksjonslag
 - et lag bygger på laget under og tilbyr en verdiøket tjeneste



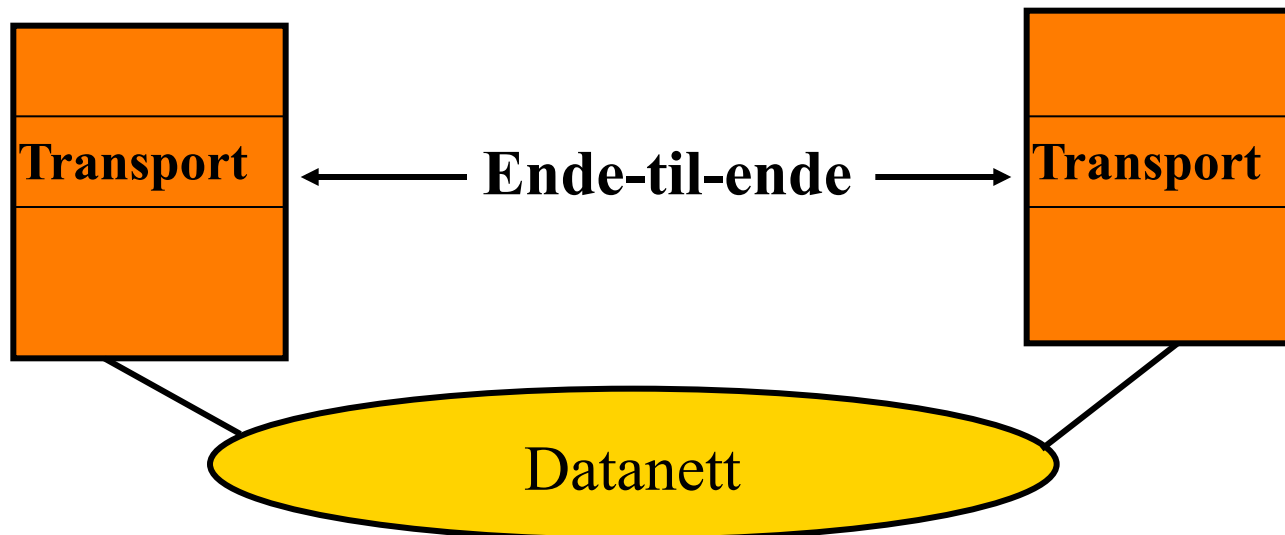
Transportlaget; egenskaper

- Det nederste laget i referansemodellen med ende-til-ende kontroll
- Transportlaget er hjertet i kommunikasjonen – står som garantist for pålitelighet
- Transportlaget fjerner den siste rest av nettavhengighet



Transportlaget; egenskaper

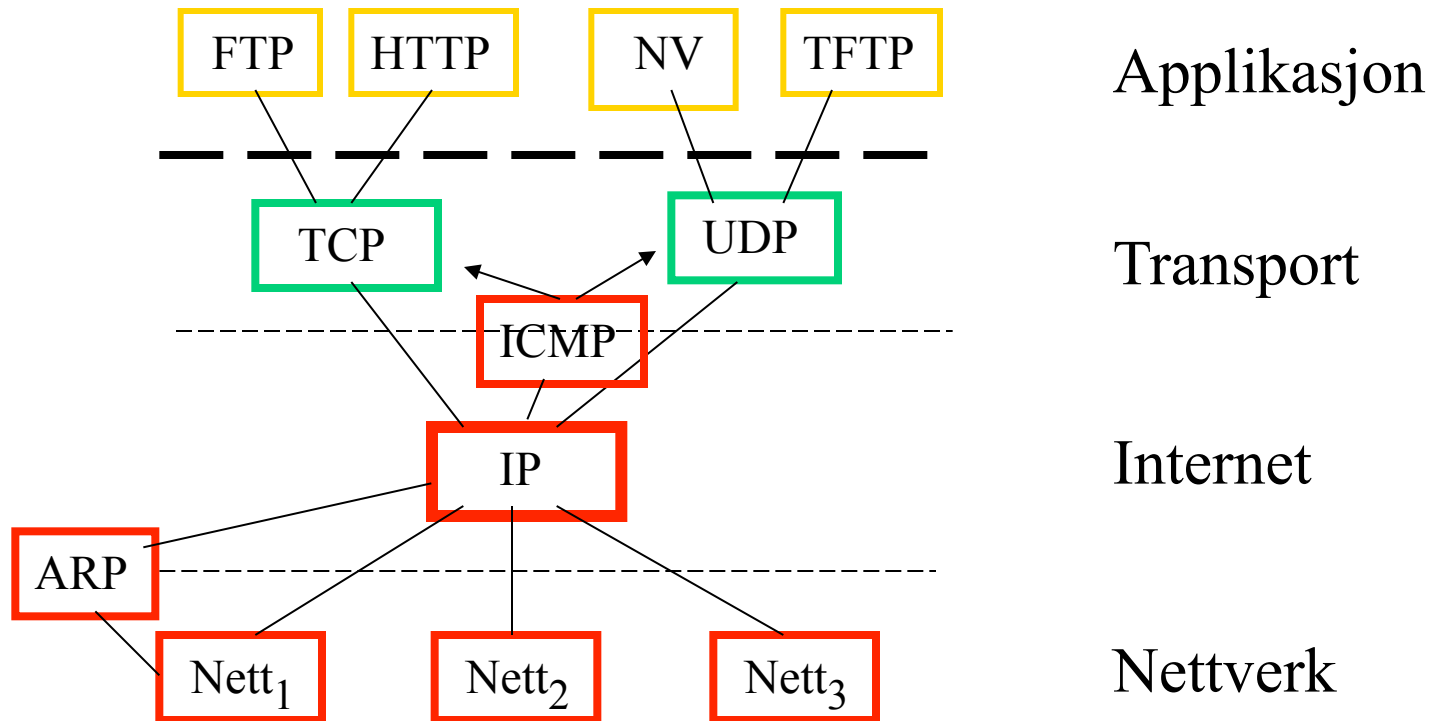
- Det nederste laget i referansemodellen med ende-til-ende kontroll
- Transportlaget er hjertet i kommunikasjonen – står som garantist for pålitelighet
- Transportlaget fjerner den siste rest av nettavhengighet



Grunnleggende transport-protokoll utfordringer

- For at transport-protokollen skal kunne garantere pålitelighet, må den kunne takle de verst tenkelige situasjoner
- Pålitelig etablering av en forbindelse:
 - Skille mellom gamle og nye instanser av en forbindelse
 - ”kræsje”-problematikker
- Pålitelig nedkopling av en forbindelse:
 - Uten tap av data
 - Halvt åpne forbindelser
- Flyt-kontroll
 - Ende-til-ende maskin
 - Maskin-til-nett

Internet arkitekturen



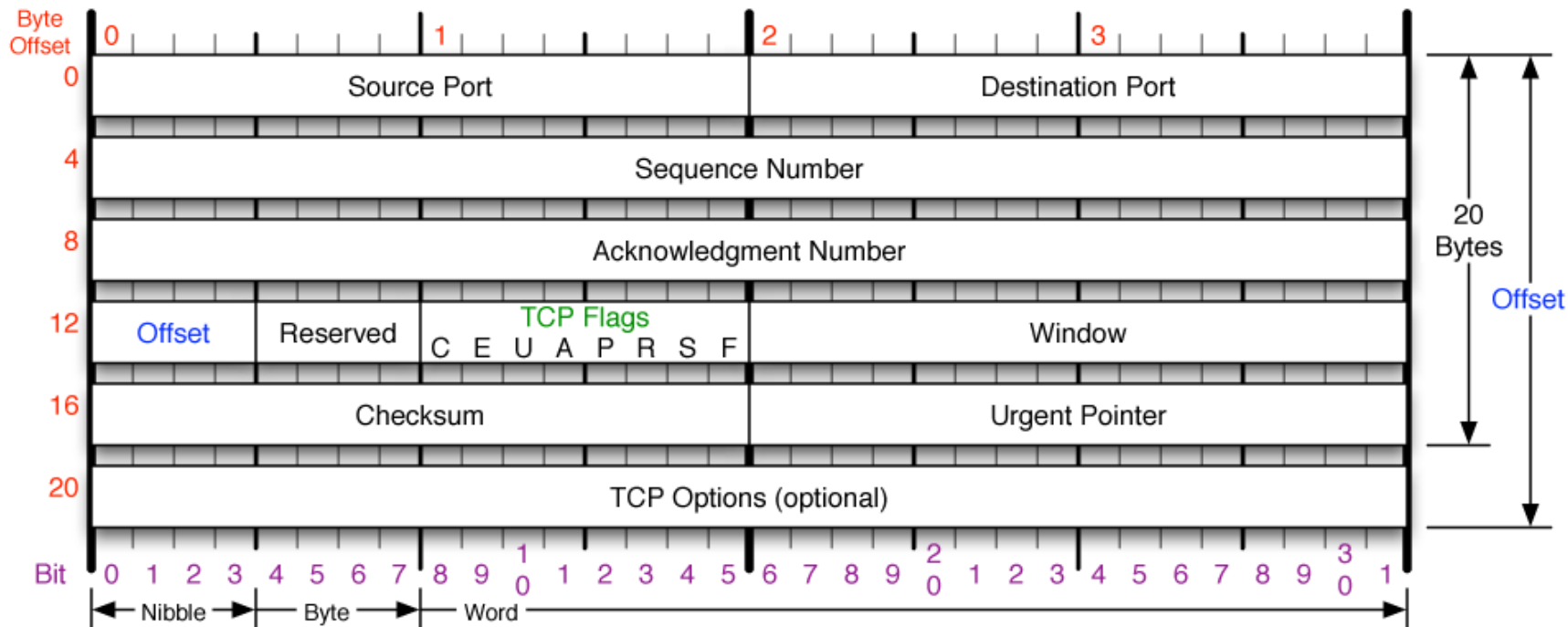
Pålitelig byte-strøm

TCP
(Transmission Control Protocol)

Funksjoner TCP

- forbindelses-orientert
- støtter en oktett-strøm mellom to prosesser
- full dupleks
- tilbyr multiplexing/demultiplexing vha portnummer
 - | for å håndtere multiple/samtidige applikasjoner
- flyt-kontroll
 - | hindrer sender å oversvømme mottaker
- metnings-kontroll
 - | hindrer sender å oversvømme nettet

TCP Header



TCP Flags

C E U A P R S F

Congestion Window

- C 0x80 Reduced (CWR)
- E 0x40 ECN Echo (ECE)
- U 0x20 Urgent
- A 0x10 Ack
- P 0x08 Push
- R 0x04 Reset
- S 0x02 Syn
- F 0x01 Fin

Congestion Notification

ECN (Explicit Congestion Notification). See RFC 3168 for full details, valid states below.

Packet State	DSB	ECN bits
Syn	00	11
Syn-Ack	00	01
Ack	01	00
No Congestion	01	00
No Congestion	10	00
Congestion	11	00
Receiver Response	11	01
Sender Response	11	11

TCP Options

- 0 End of Options List
- 1 No Operation (NOP, Pad)
- 2 Maximum segment size
- 3 Window Scale
- 4 Selective ACK ok
- 8 Timestamp

Checksum

Checksum of entire TCP segment and pseudo header (parts of IP header)

Offset

Number of 32-bit words in TCP header, minimum value of 5. Multiply by 4 to get byte count.

RFC 793

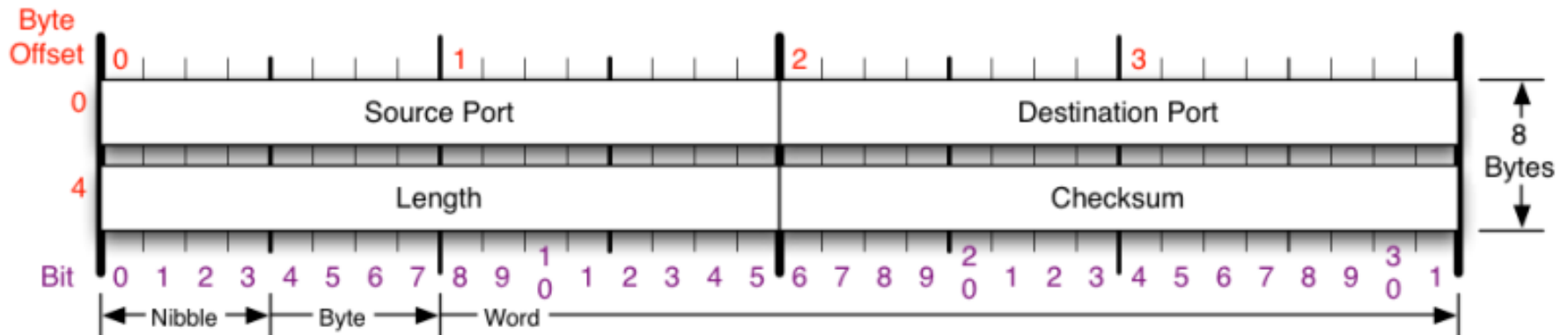
Please refer to RFC 793 for the complete Transmission Control Protocol (TCP) Specification.

”Upålitelig” overføring

UDP
(User Datagram Protocol)

UDP Header

- ingen flytkontroll;
- tjener-siden lytter på velkjente porter
- checksum: valgbar
- tilbyr multipleksing og demultipleksing ved hjelp av “Service aksess punkter” kalt porter;



Checksum of entire UDP segment and pseudo header (parts of IP header)

Please refer to RFC 768 for the complete User Datagram Protocol (UDP) Specification.

UDP Standard overføringskvalitet

- forbindelsesfri transport (datagram transport)
- “best-effort” overføring
 - **ingen garanti for vellykket overføring:**
 - pakker kan bli borte (sjekksumfeil & rutefeil)
 - pakker kan komme frem i gal rekkefølge
 - pakker kan dupliseres
 - pakker kan forsinkes unormalt

Nettverks-applikasjoner: noen begreper

- En **prosess** er et program som kjører i en vertsmaskin.
- I samme vertsmaskin kommuniserer to prosesser vha. **interprosess-kommunikasjon** definert av operativsystemet.
- Prosesser som kjører i **forskjellige** vertsmaskiner kommuniserer vha. en **applikasjonslags** protokoll
- En **brukeragent** (user agent) er et grensesnitt mellom brukeren og nettverks-applikasjonen.
 - ➔ **Web: navigatør (browser)**
 - ➔ **E-post: post-leser**
 - ➔ **streaming audio/video: media-spiller**

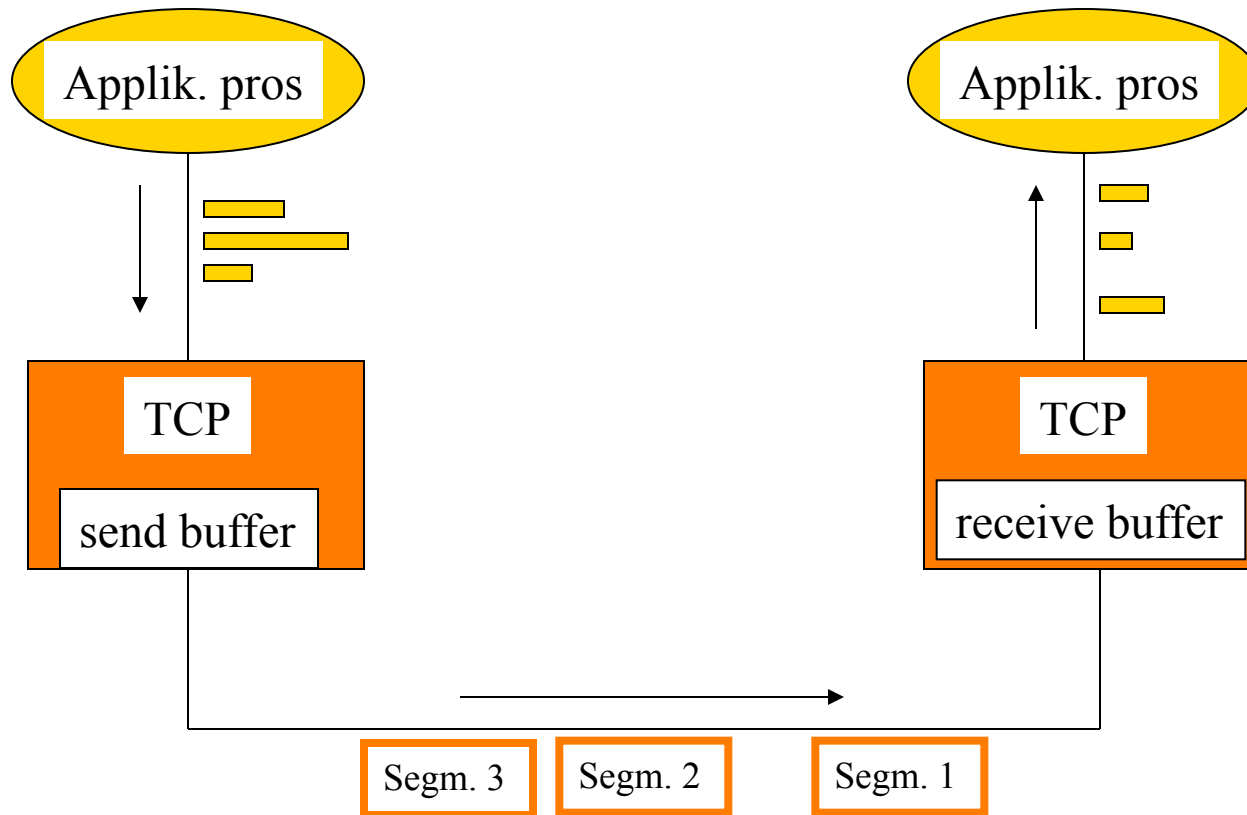
Vanlige applikasjons-behov

- garantere avlevering av pakker
- tillate vilkårlige pakkestørrelser
- avlevere pakker i riktig sekvens
- avlevere kun en kopi av hver pakke
- synkronisere samarbeidende prosesser til hverandre
- utøve flytkontroll
- støtte mange applikasjonsprosesser i hver maskin
- Noen apps har også krav om båndbredde og forsinkelse

Funksjonelle behov

- applikasjonsprosessenes behov:
 - navning av maskiner og tjenester
 - konvertering av navn til adresser
 - tilgang til kommunikasjonstjenesten (API):
 - service aksess punkt (SAP);
virtuelt tilknytningspunkt mellom
applikasjonsprosess og komm.hierarkiet
 - etablering, bruk, og nedkopling av forbindelser
 - spesifisere kvalitets-krav

Oktett-strøm mellom applikasjonsprosesser



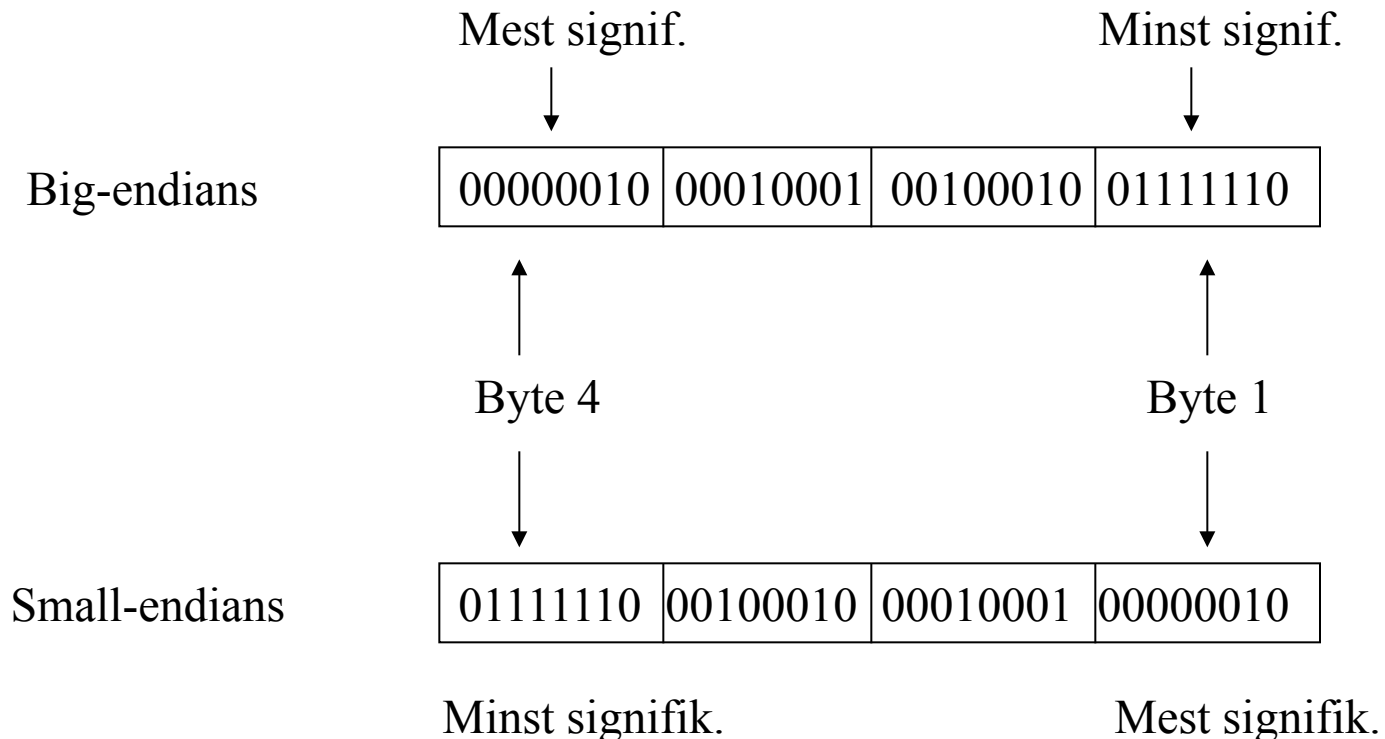
Overførings-syntaks

- inhomogene ende-systemer
 - ulik hardware
 - ulike operativsystemer
 - ulike programmerings-språk
- kommunikasjonen over nettet må foregå i en syntaks som begge sider oppfatter på samme måten

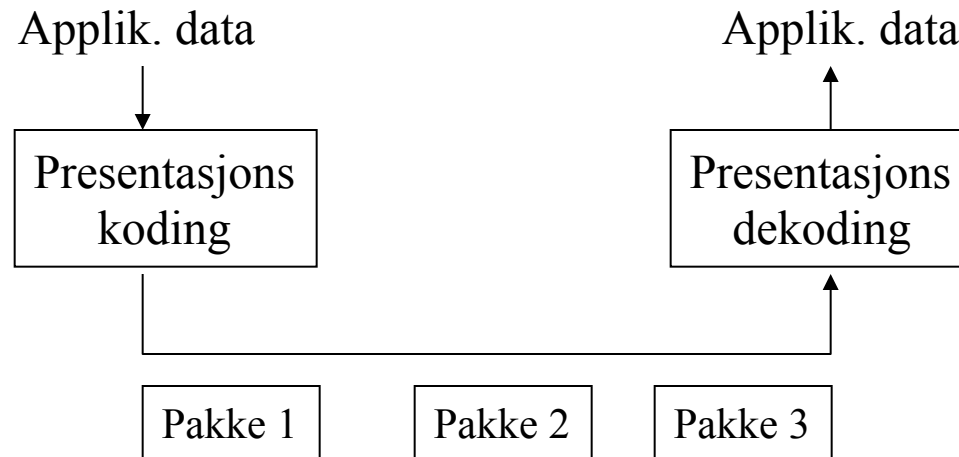
Eksempel på maskinvare-forskjeller

- Endianness refererer til rekkefølgen av individuelt adressbare deler (sub-komponenter) mhp representasjonen av større data komponenter som lagres i ekstern memory (eller sendt over en seriell forbindelse)

Integerverdien 34.677.374



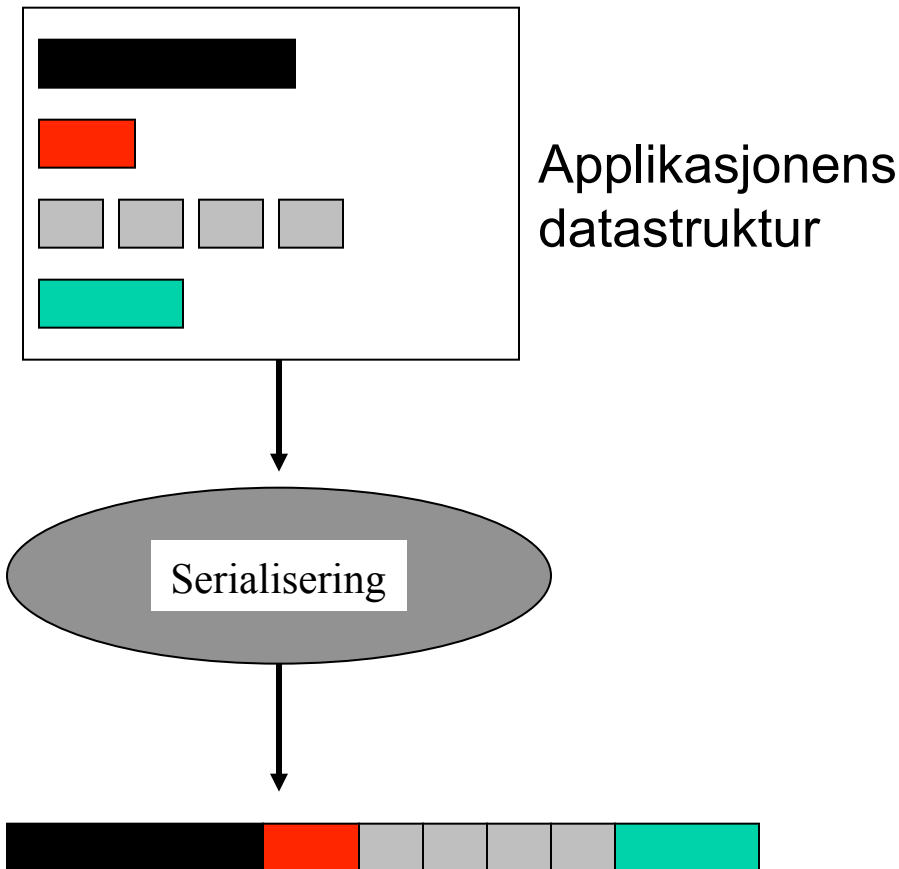
Overførings syntaks



Data må overføres i en overførings syntaks som begge sider oppfatter på samme måte!

Marshalling/serialisering

- Er prosessen med å transformere memory representasjonen av et objekt til et format som egner seg for transmisjon (og lagring)
- Benyttes typisk når data skal flyttes mellom programmer og maskiner



Marshalling

- konvertering
- serialisering
- innpakking

ASN.1 eksempler

(Abstract Syntax Notation One)

- ASN.1 er en standard og fleksibel notasjon som beskriver regler og strukturer for å representere, kode, sende og dekode data innen tele- og computer networking

Integer > 65 535

32-bits maskin; representert i et 32-bits ord

16-bits maskin; representert i to 16-bits ord

Med ASN.1 kodes hver datatype ved overføring som et trippel:

Type	Lengde	Verdi
------	--------	-------

INT	4	B1	B2	B3	B4
-----	---	----	----	----	----

Type	lengde	verdi	type	lengde	verdi	type	lengde	...
------	--------	-------	------	--------	-------	------	--------	-----

- Primitive typer:

- Boolean
- Integer
- Bit streng
- Oktett streng
- Null
- Objekt ID
- Real

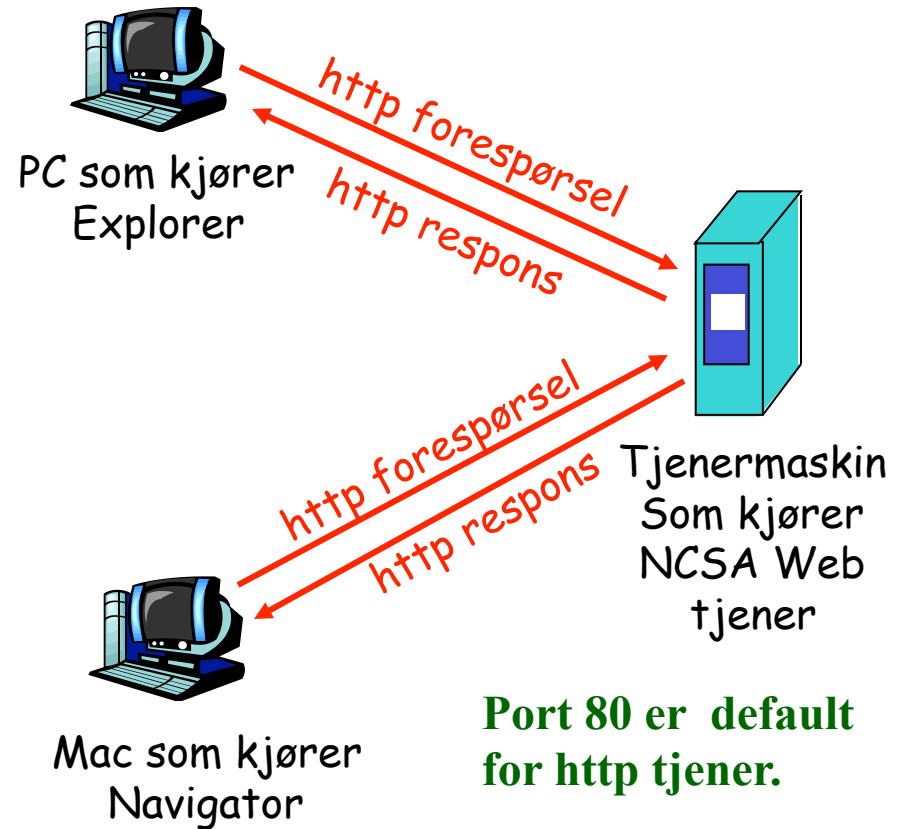
- Også struktur typer:

- Lister

- ...

Web-en: http protokollen

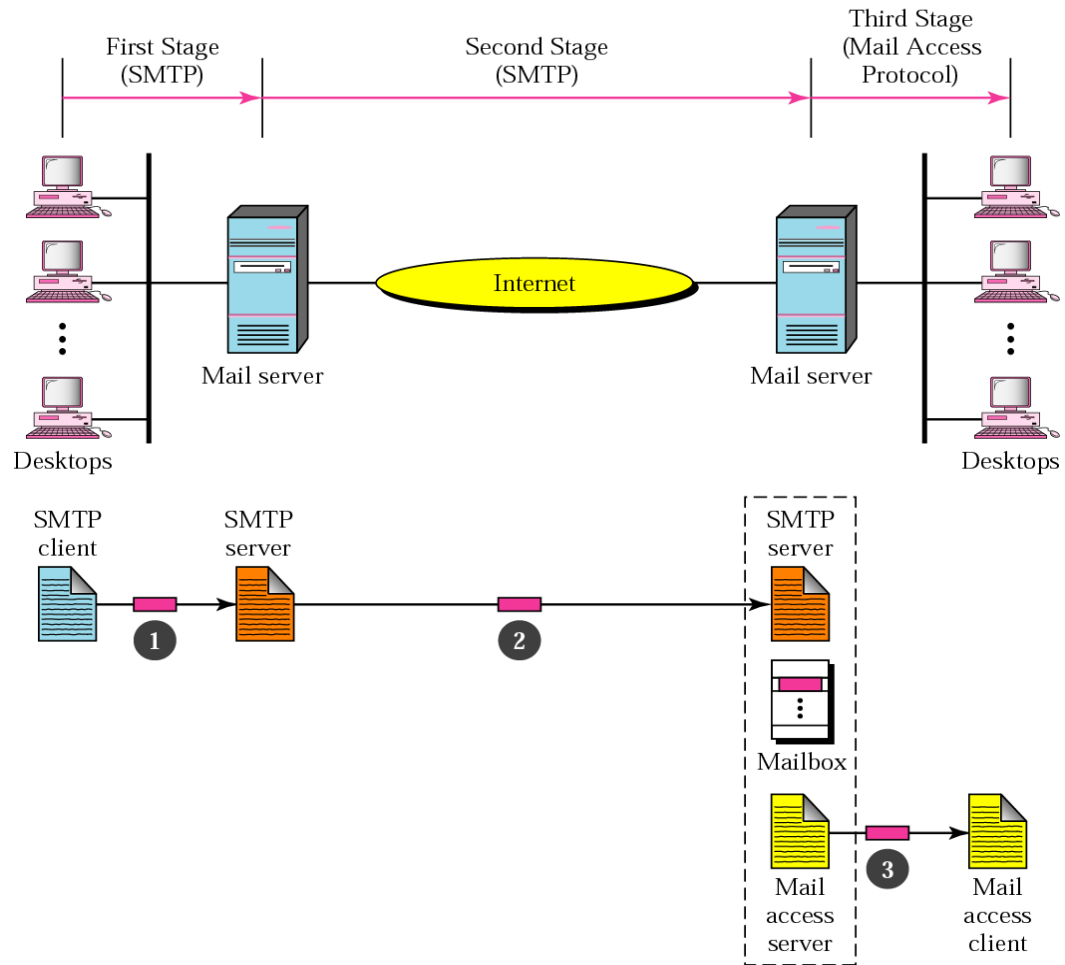
- http: hypertext transfer protocol
- Web er en applikasjonslags-protokoll
- klient/tjener modell
 - *klient*: navigator som ber om, mottar og "viser" Web objekter
 - *tjener*: Web tjener sender objekter som svar på forespørsler
- http1.0: RFC 1945
- http1.1: RFC 2616



http protokollen: mer

- http er “tilstandsløs”
- Tjeneren vedlikeholder ingen informasjon om tidligere klientforespørsler

Epost levering



Viktige datacom funksjoner

- Viktige og nødvendige funksjoner
 - adressering, routing, og fremsending
 - deteksjon og korreksjon av feil
 - ende-til-ende pålitelighet
 - metningskontroll
 - multicasting
 - overføringskvalitet

Noen viktige begreper

- databit samles i pakker (1000 → 10.000 bit)
- noder eller pakke-switcher
 - mellom-lagring og framsending
 - asynkron kommunikasjon
 - virtuelle linjer eller datagram
 - minst to linjer ut fra hver node
- endesystemer med et stort antall tjenester

Spørsmål fra Pål Halvorsen & co

- Spørre studentene på forelesningen om de kan delta i et kort eksperiment ved å se på YouTube videoer?
- Søke etter et tema og deretter bruke related-listen for å finne flere relaterte treff? En kort oppskrift er gitt her:

<http://emmy9.casa.umass.edu/logger.html>