

INF105 - 15
UDP / TCP / IP

David Grellscheid

5-layer model

- **Application**: **HTTP**, SMTP, FTP, SSH, DNS
  lett å lage nye protokoller her
  end system <-> end system
- Transport
- Network
- Link
- Physical

5-layer model

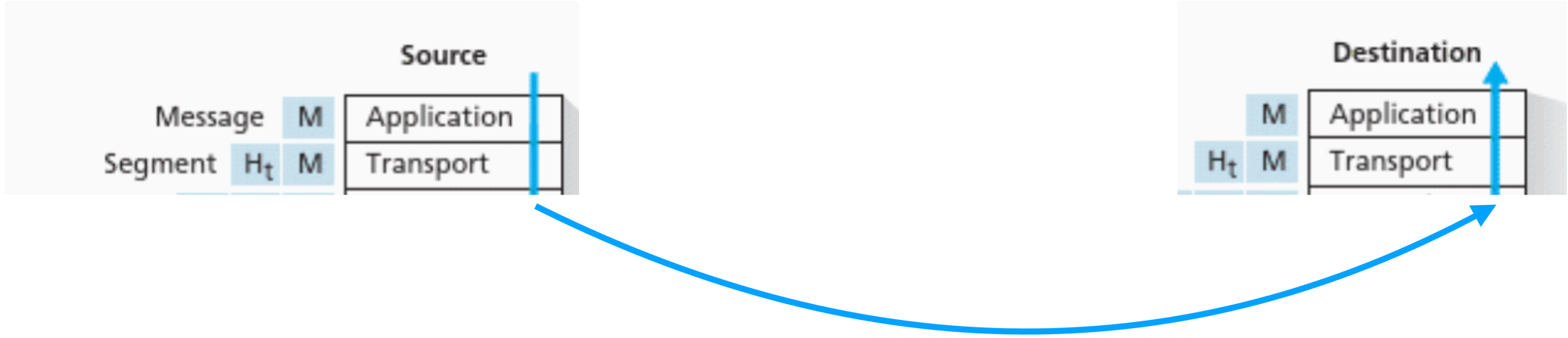
- Application: HTTP, SMTP, FTP, SSH, DNS
- **Transport**: **TCP** (connection) eller **UDP** (connectionless)
  sender "segments" fra end system <-> end system
- Network
- Link
- Physical

Transport layer

- Network layer (layer 3) sender pakker mellom 2 enheter
- En enhet kan ha mange applications i App.layer (layer 5)
- Transport layer er mellomleddet som gjør dette mulig,
  sender segments mellom 2 prosesser som kjører på enhetene
  - enklest: UDP (RFC 768)
- Transport layer også skal sikre pålitelig kommunikasjon
  - TCP (RFC 793)
- Transport-layer aktivitet skjer på end-systems, rutere ikke bryr seg

Transport layer

- Bryter opp application-melding i mindre "segments"
- legger til transport-informasjon



- på internett: enten UDP eller TCP
- applikasjon velger transport-metoden og en ledig "port"

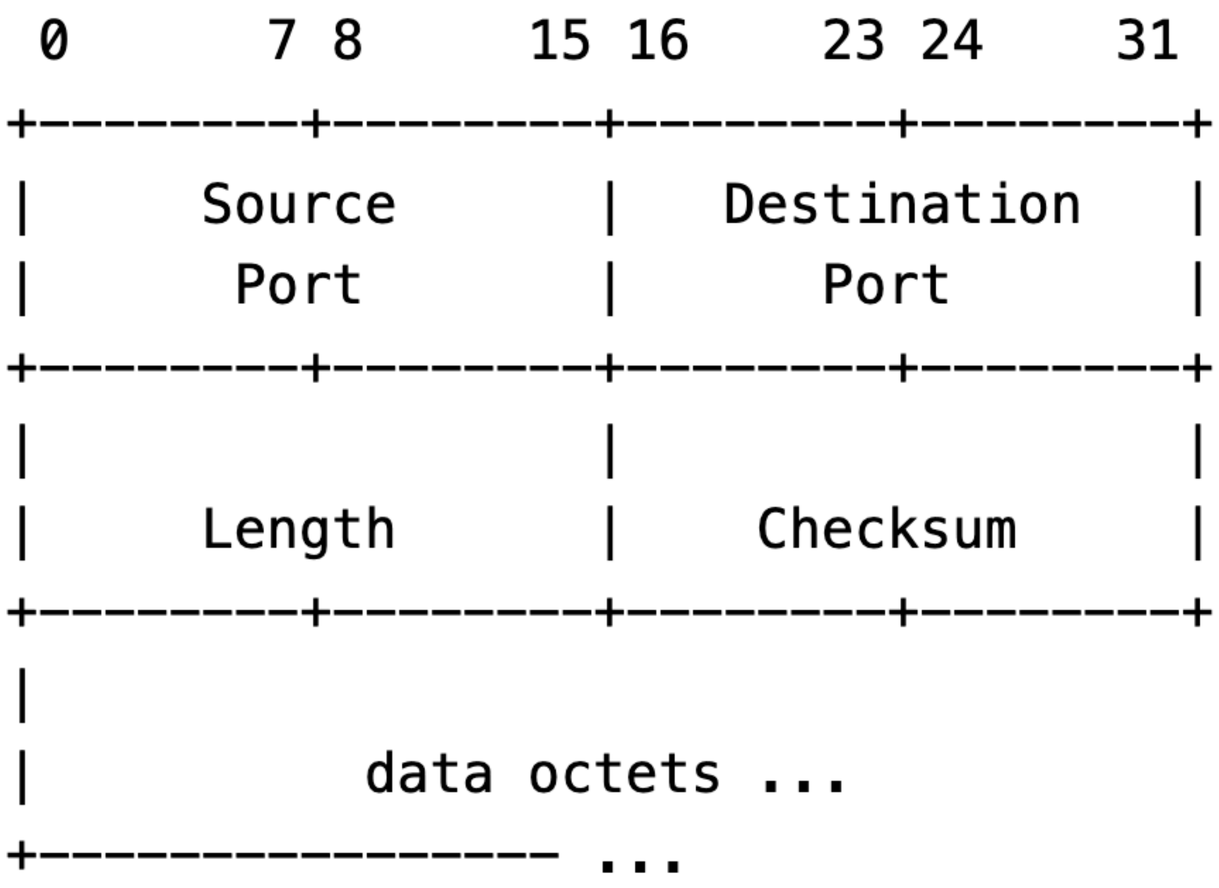
ports

- En applikasjon "binder" til en socket på hver side, informasjonen i transport header brukes til å velge riktig app



UDP

- RFC 768 (1980)
- introduserer "Port"-koseptet
- checksum inkluderer IP-adressene, mot feilsendte pakker



User Datagram Header Format

8 byte

UDP

- Når UDP får data fra applikasjon, blir det segmentert, UDP-header lagt til, og sendt videre til nettverk
- ingen tilbakemelding om segmentet kom frem
- fordel: liten overhead, bare 8 byte i tillegg
- brukes til applikasjoner der noen tap av informasjon kan håndteres på layer 5 (audio / video live chat)

TCP

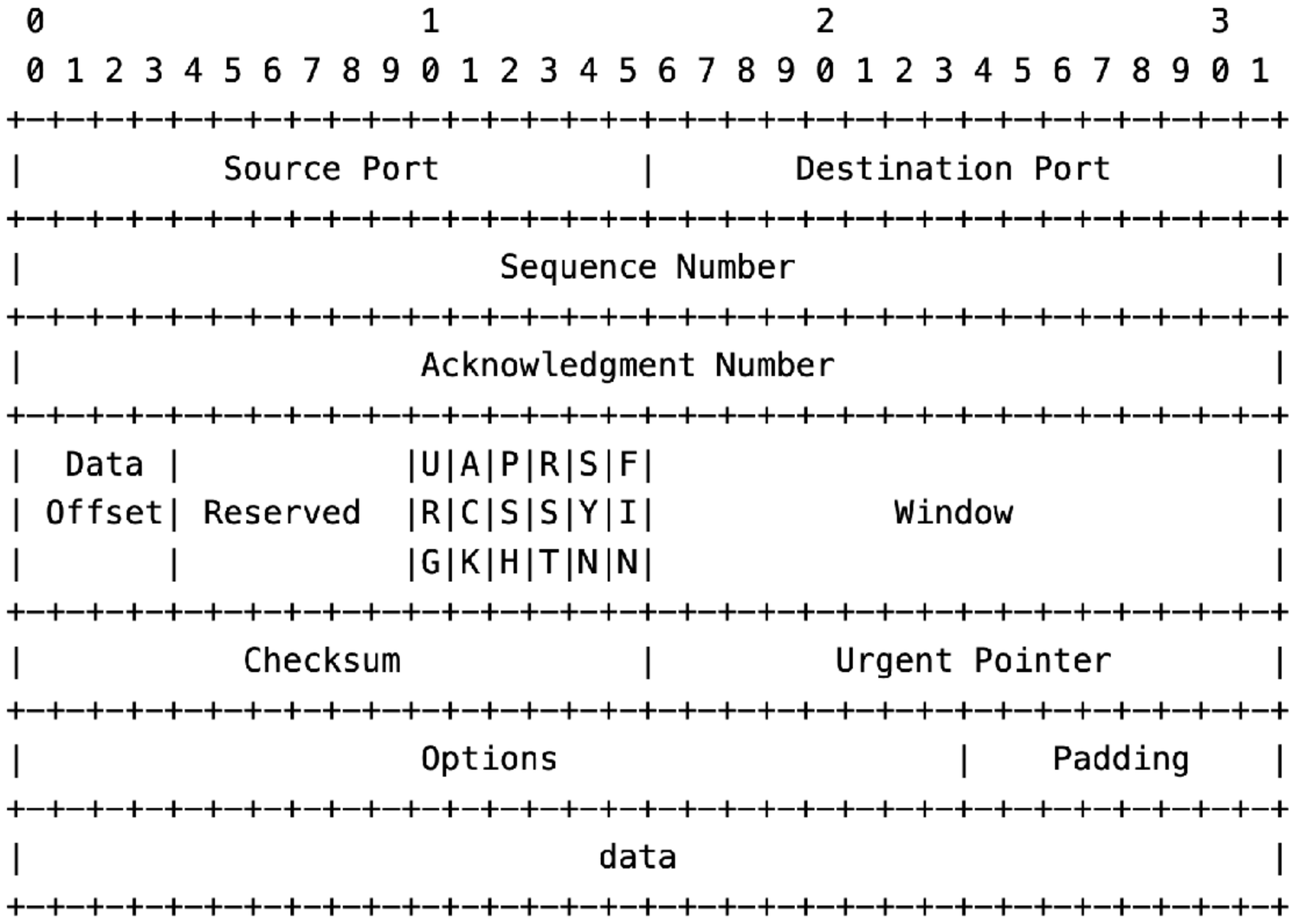
- Med TCP på layer 4 later vi som at vi har en pålitelig forbindelse
- selv om layer 3 (IP) kan ikke garantere det
- Må
  - sjekke at et segment kom frem, og
  - er uendret
  - hvis ikke, sende på nytt

TCP

- sjekke at segmentet kom frem
  - sende ACK (bekreftelse) tilbake
  - men hva om selve ACK ikke kommer frem?
  - ACK må si hvilket segment blir bekreftet
- segmentene må nummereres
- hvis ingen/feil ACK, må segmentet sendes på nytt

TCP

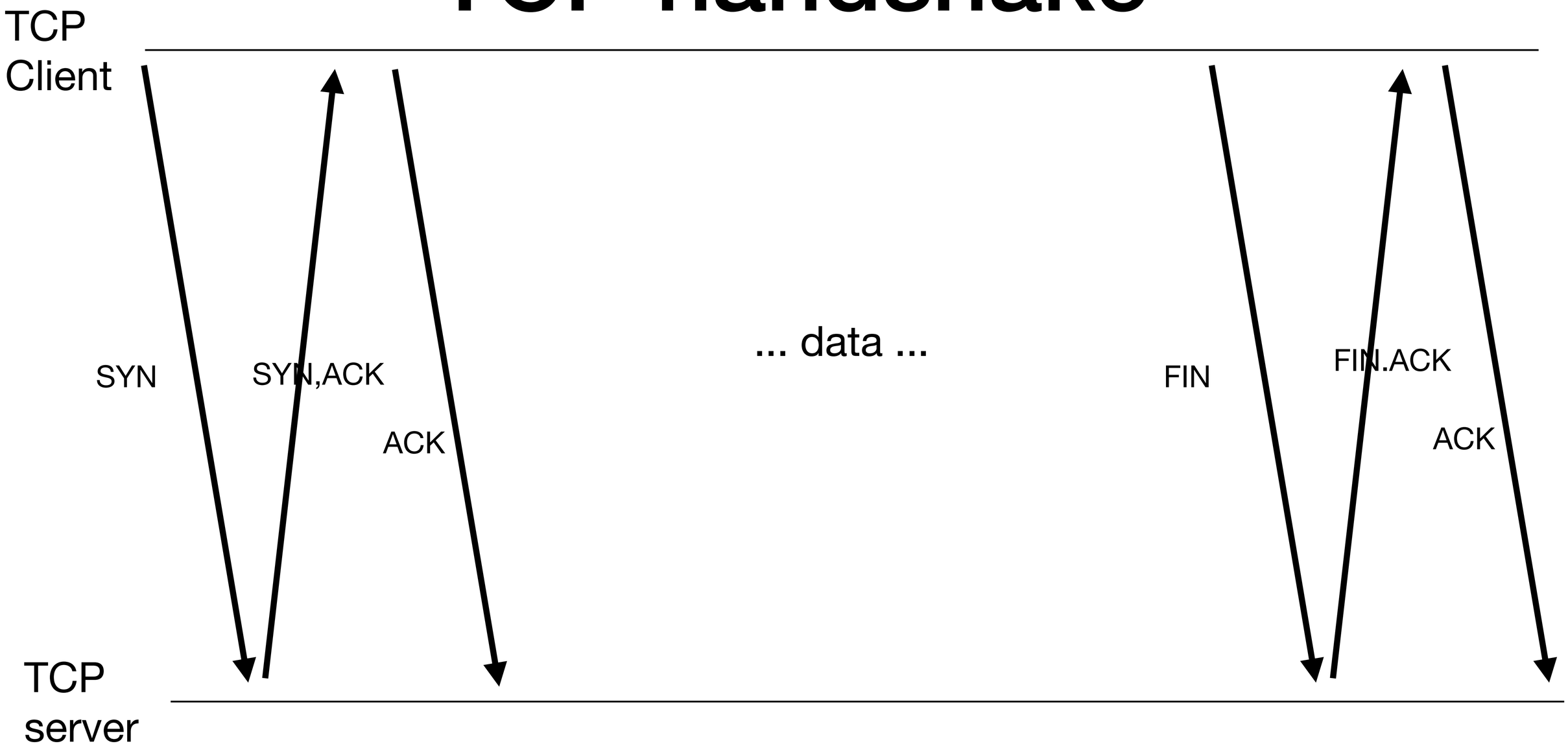
- TCP/IP fra 1974
- RFC 791 (1981)
- også har ports
- nummerert sekvens
- Flags:
  SYN, ACK, FIN, RST, ...
- ...



TCP Header Format

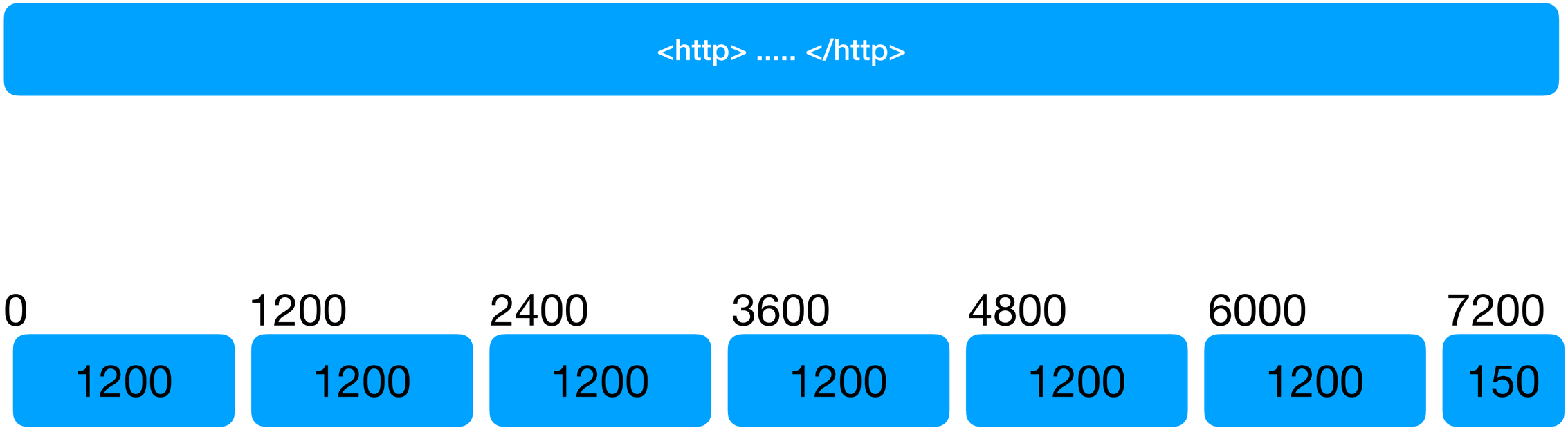
24 byte

TCP handshake



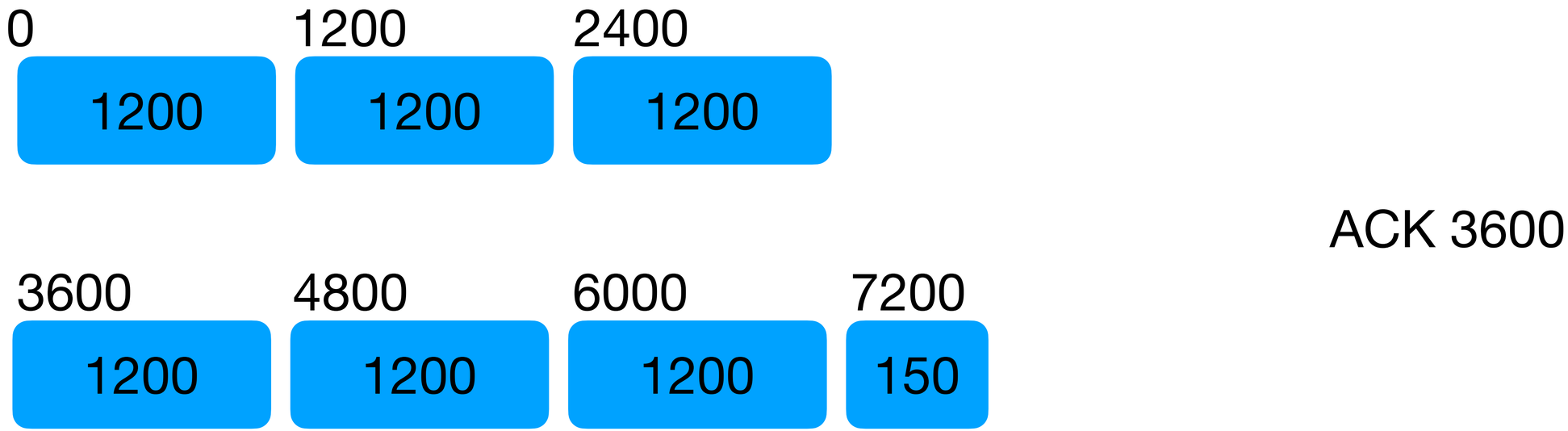
TCP data

- En sortert sekvens av segments
- sekvensnummer til en segment = antallet bytes i meldingen så langt

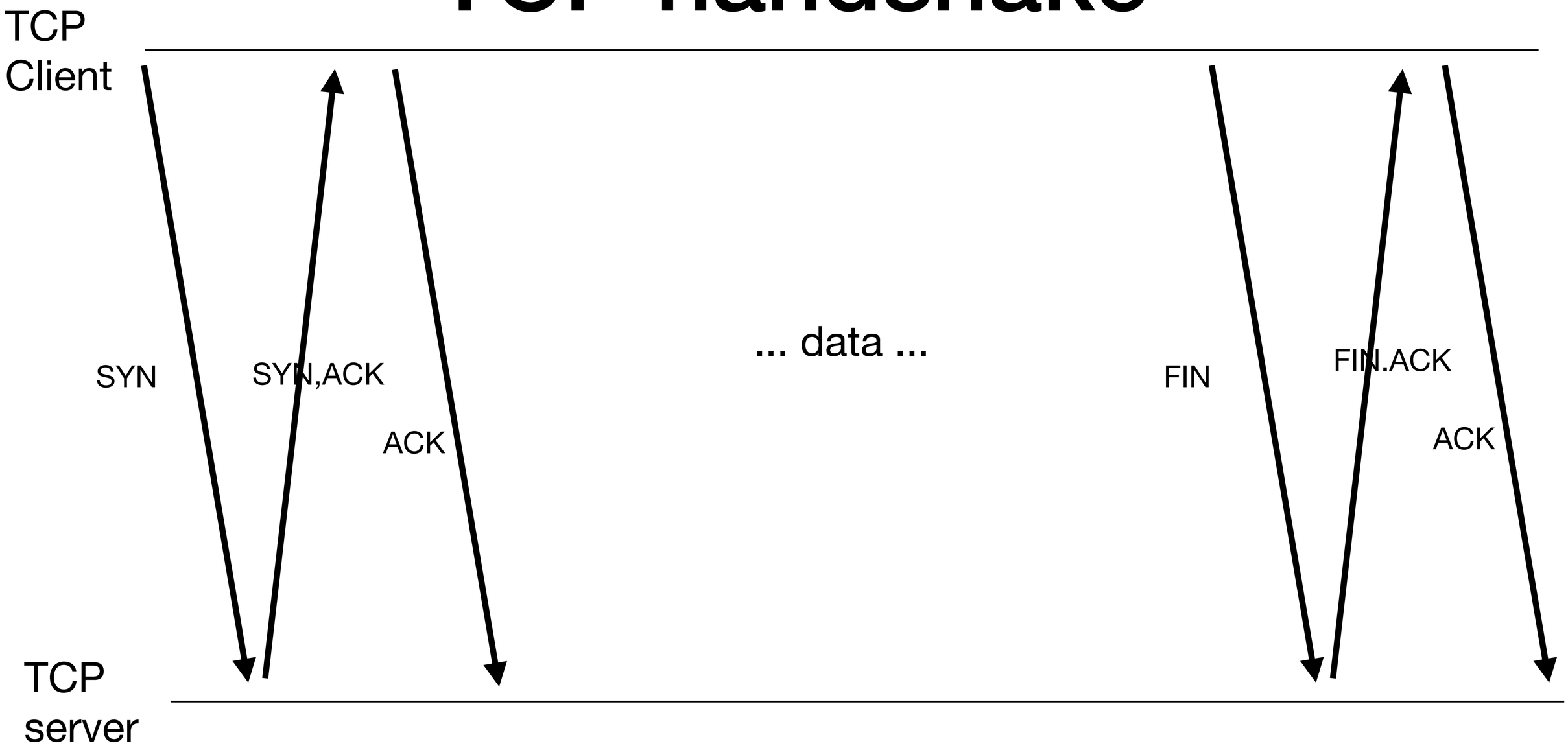


TCP ACK

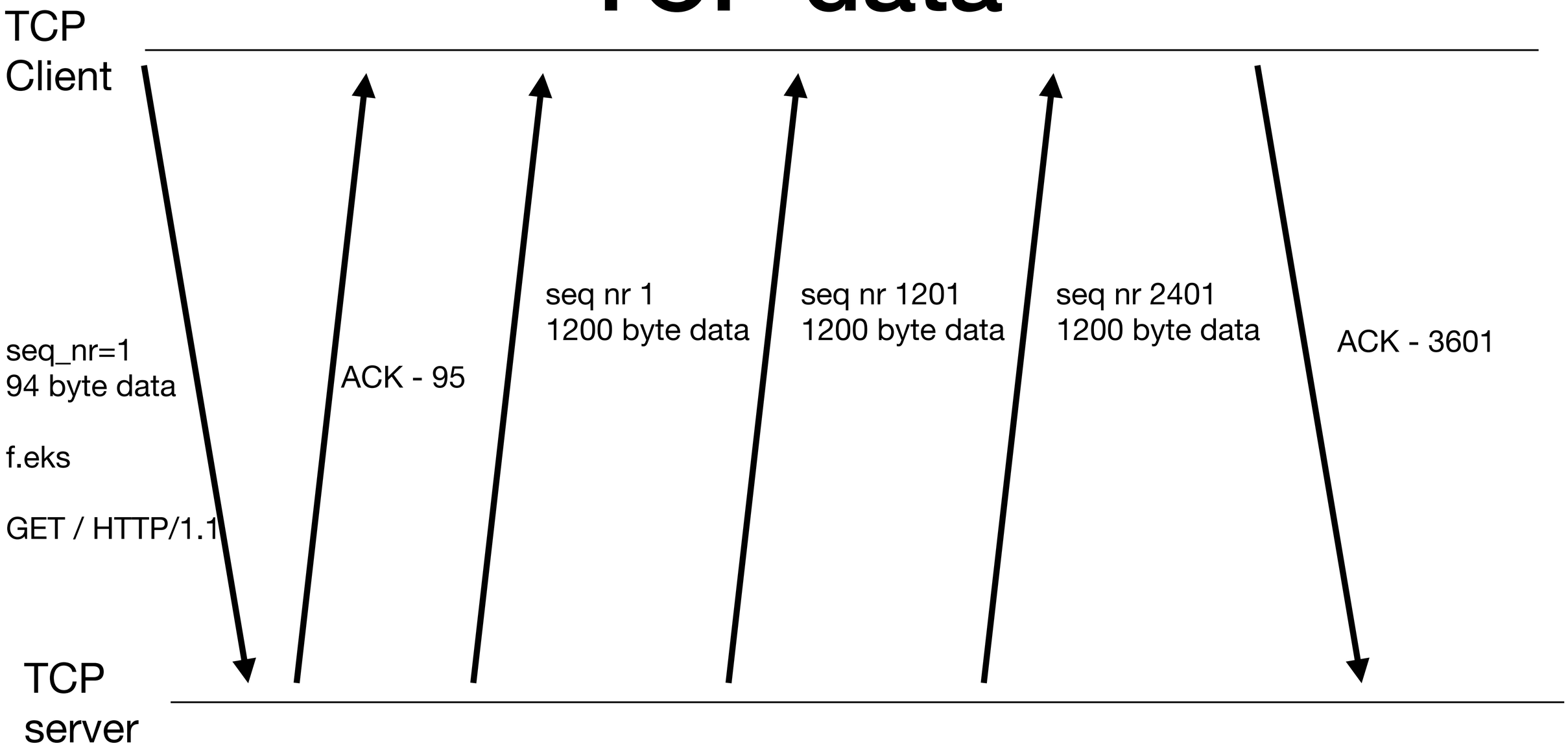
- Bekreftelse ACK inneholder **neste** sekvensnummer som kan mottas



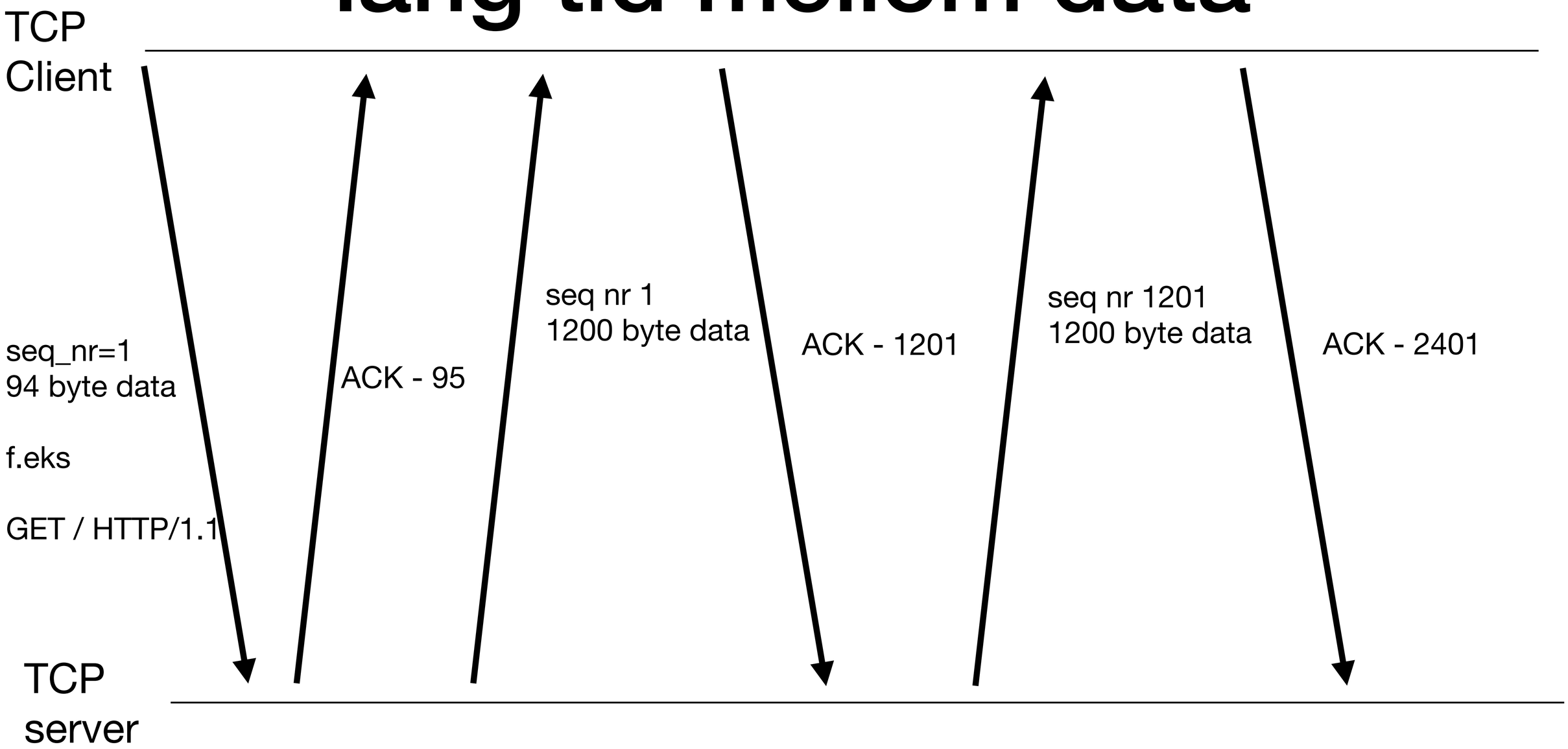
TCP handshake



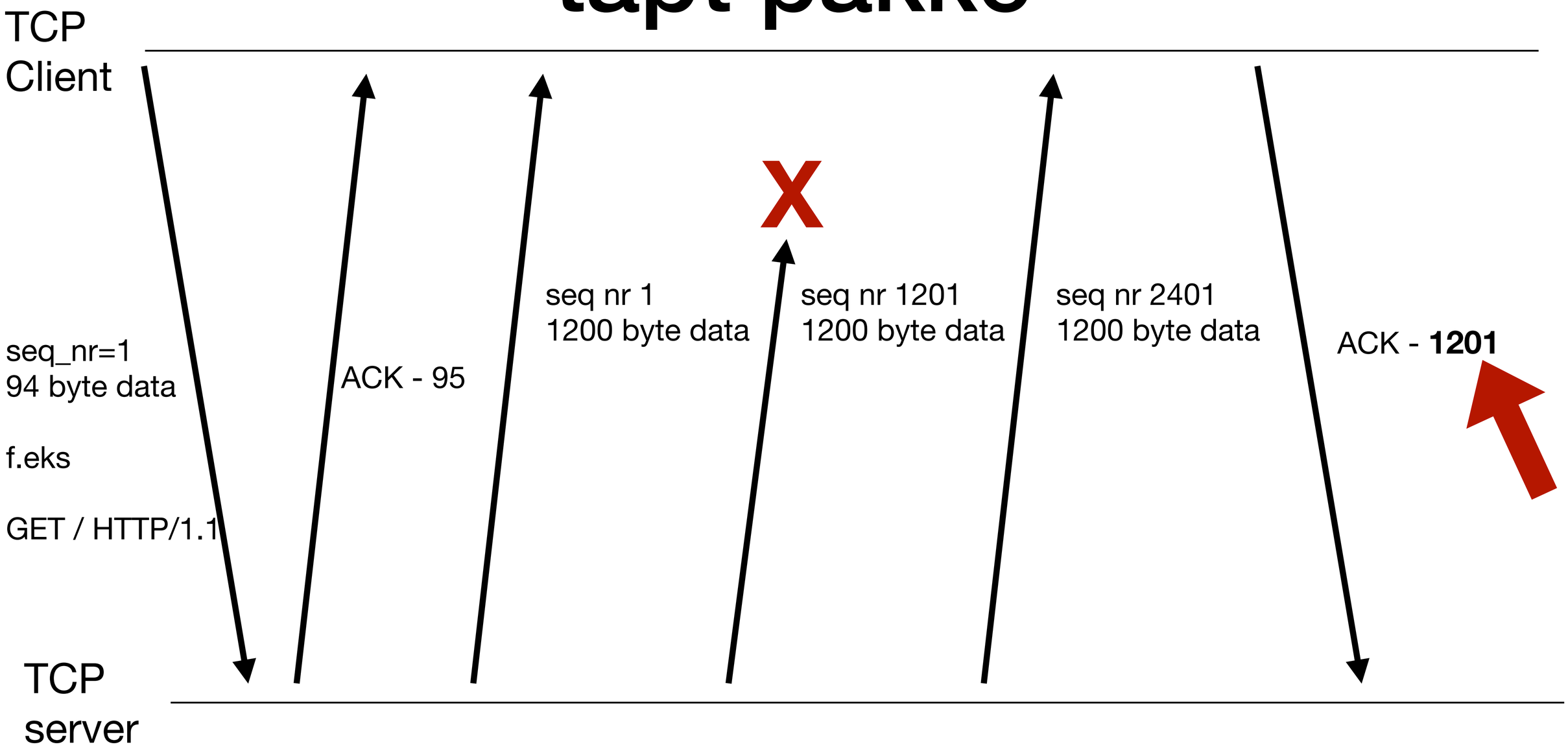
TCP data


TCP
Client
seq_nr=1
94 byte data
f.eks
GET / HTTP/1.1
ACK - 95
seq nr 1
1200 byte data
seq nr 1201
1200 byte data
seq nr 2401
1200 byte data
ACK - 3601
TCP
server

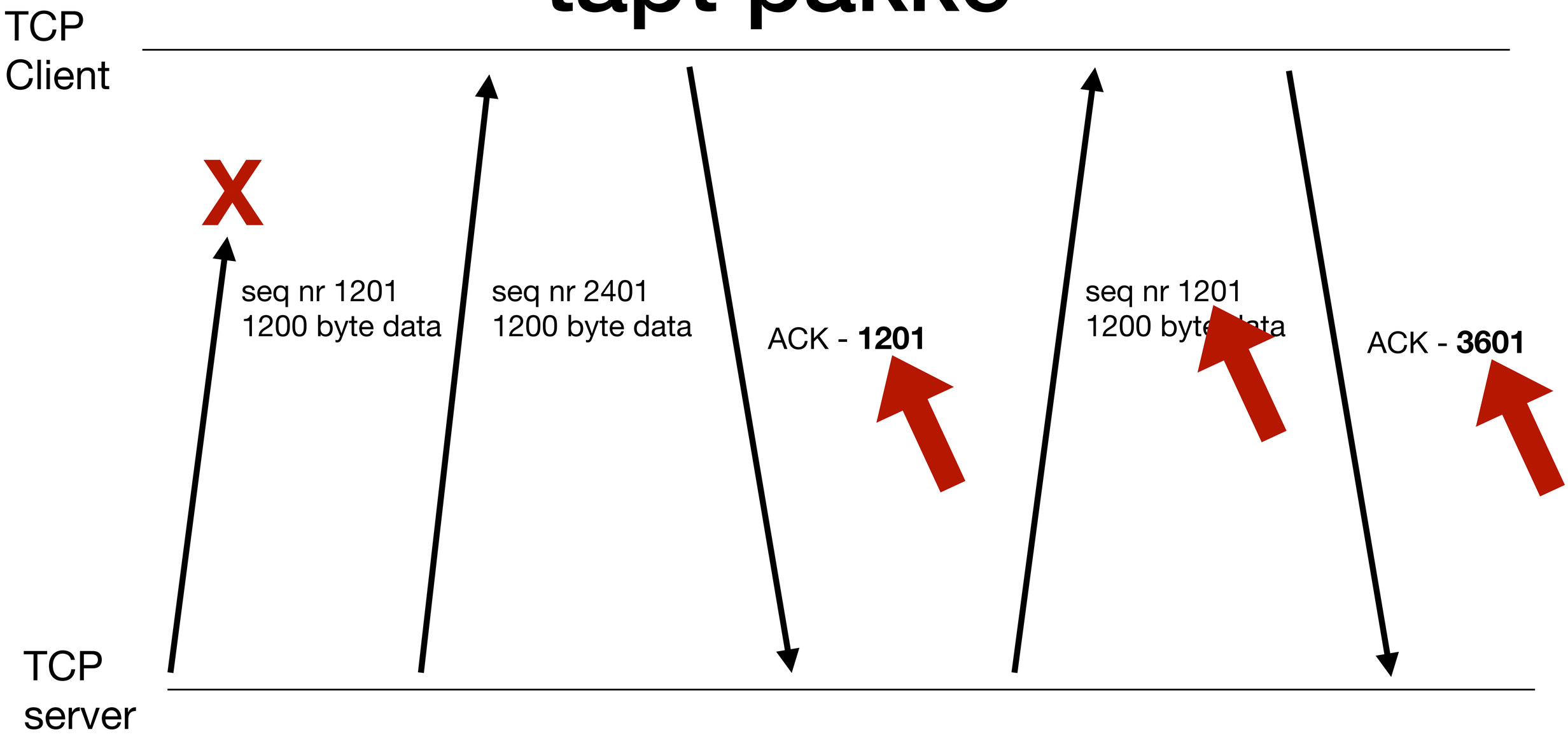
lang tid mellom data



tapt pakke


TCP
Client
seq_nr=1
94 byte data
f.eks
GET / HTTP/1.1
ACK - 95
seq nr 1
1200 byte data
X
seq nr 1201
1200 byte data
seq nr 2401
1200 byte data
ACK - 1201
TCP
server

tapt pakke


TCP
Client
X
seq nr 1201
1200 byte data
seq nr 2401
1200 byte data
ACK - 1201
seq nr 1201
1200 byte data
ACK - 3601
TCP
server

TCP

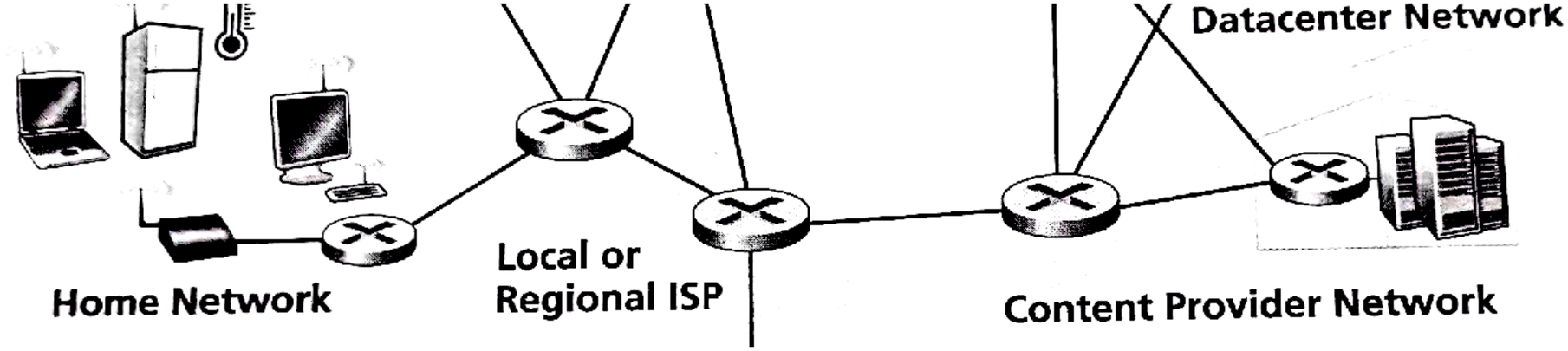
- mange flere features: congestion control, ...
- out of scope here

5-layer model

- Application: HTTP, SMTP, FTP, SSH, DNS
- Transport: TCP / UDP
- **Network**: **IP** (ingen andre valg på Internet)
  sender "datagrams / packets"
  mellom end systems fra en router til den neste
- Link
- Physical

Layer 3 - IP

- Første gang vi må se på flere steg innimellom enhetene
- **Ruter**



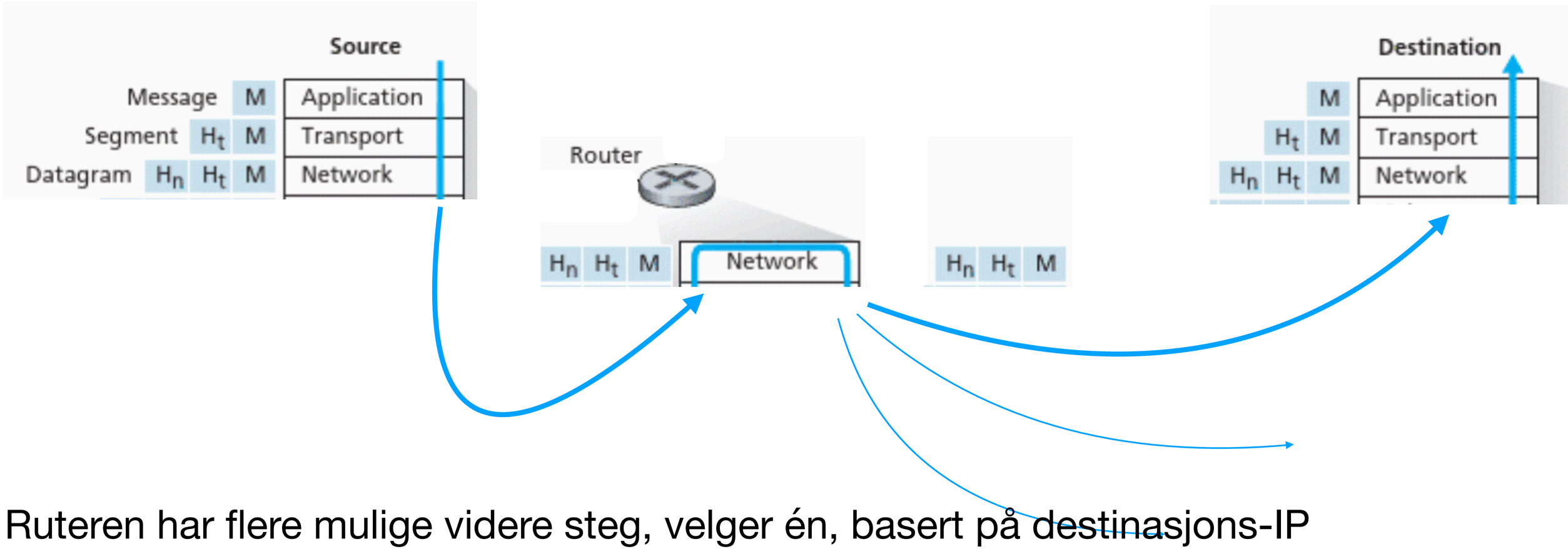
Kurose/Ross,
"Computer Networking,
a top-down approach",
Pearson 2022

Layer 3 - IP

- RFC 791 (1981), basert på TCP/IP ideen fra 1974
- grovt sagt: hver enhet får en **IP-adresse**
- bedre formulert:
  hver (fysisk) nettverkstilkobling til en enhet får en **IP-adresse**

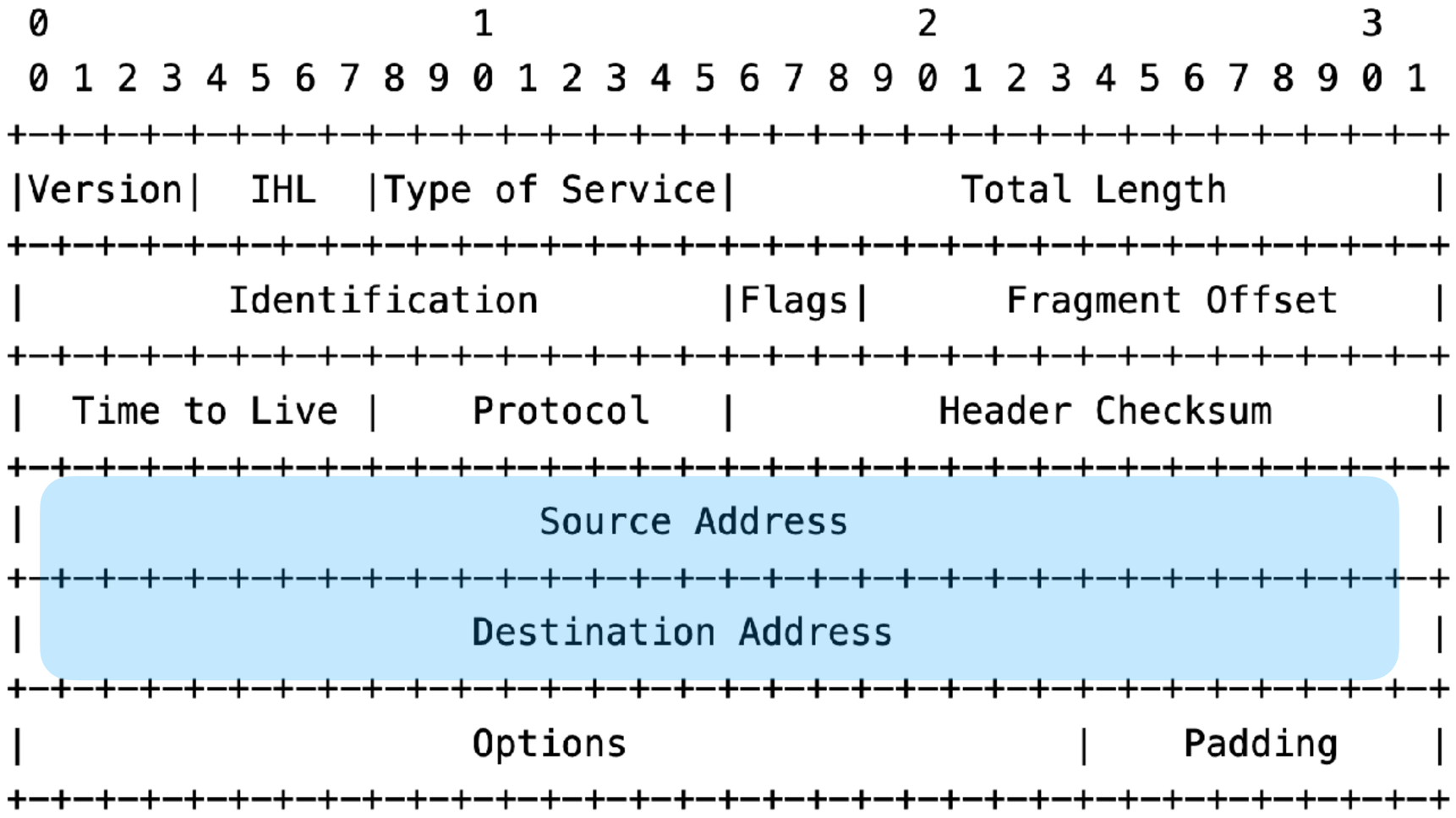
Network Layer - IP

- IP-adressene brukes av **rutere** underveis for å sende pakker til riktig destinasjon



Ruteren har flere mulige videre steg, velger én, basert på destinasjons-IP

Layer 3 - IP



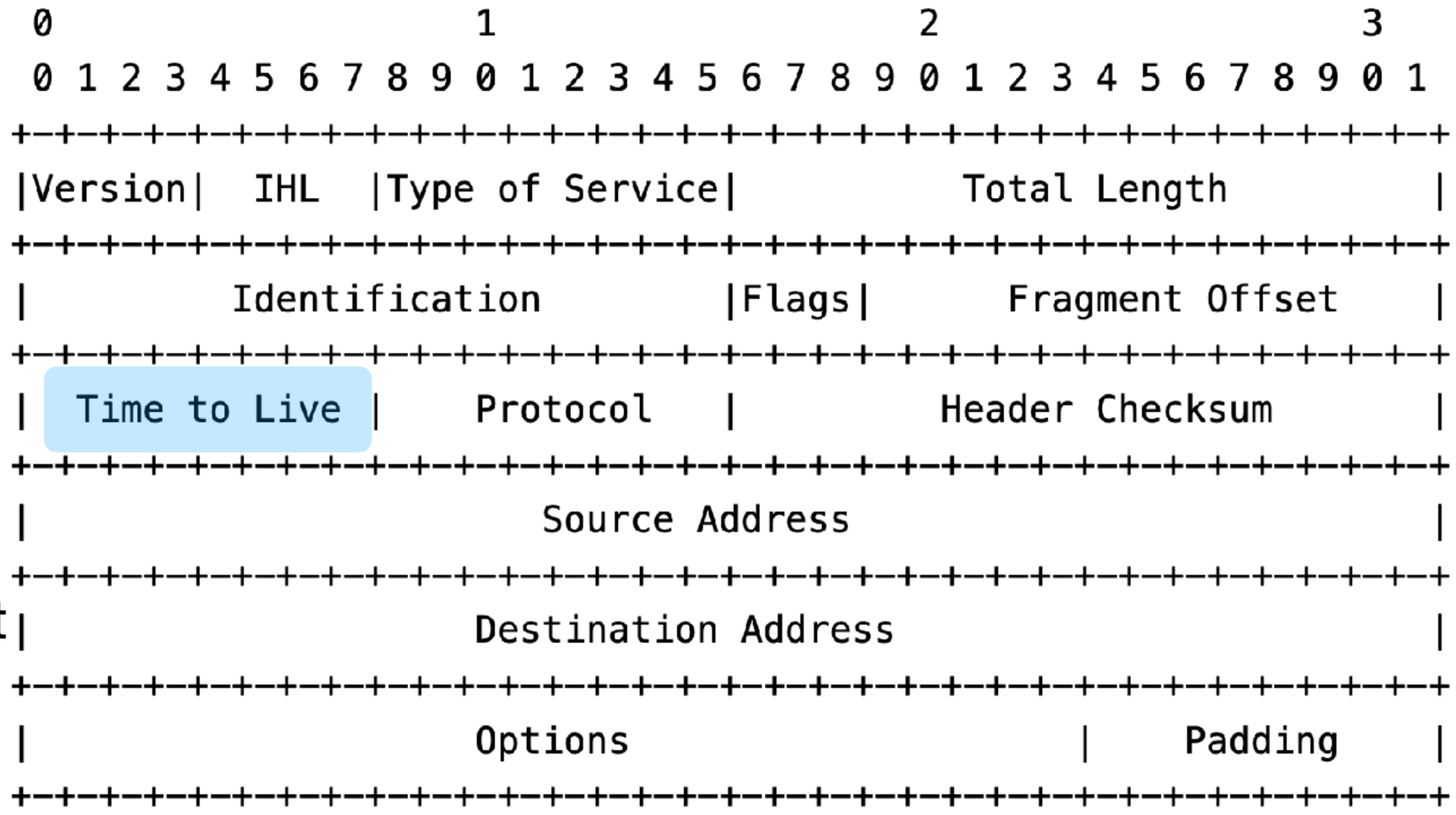
Example Internet Datagram Header

24 byte

Layer 3 - IP

telles ned
av hver ruter

unngå pakker
som går rundt
uendelig

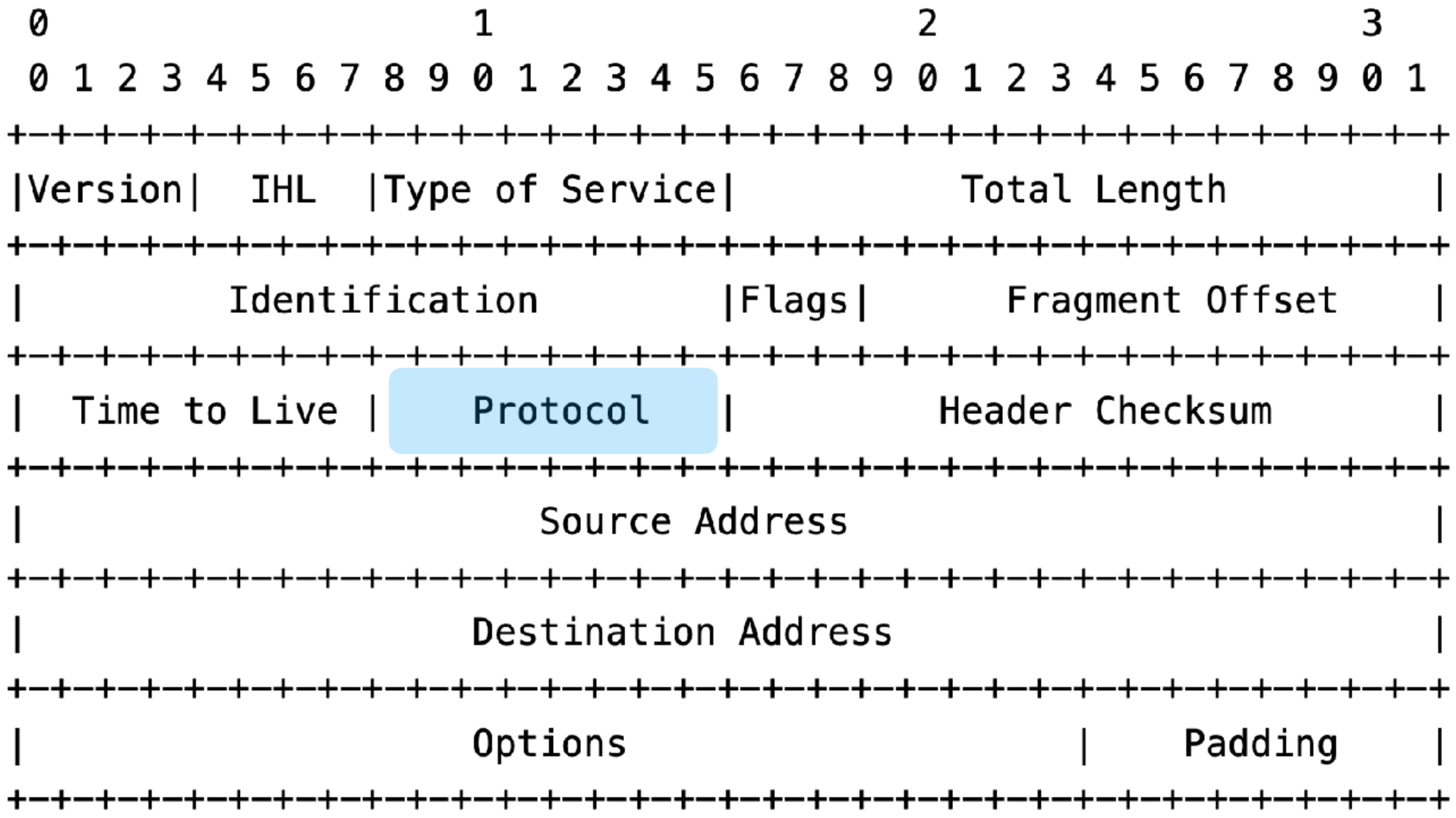


Example Internet Datagram Header

Layer 3 - IP

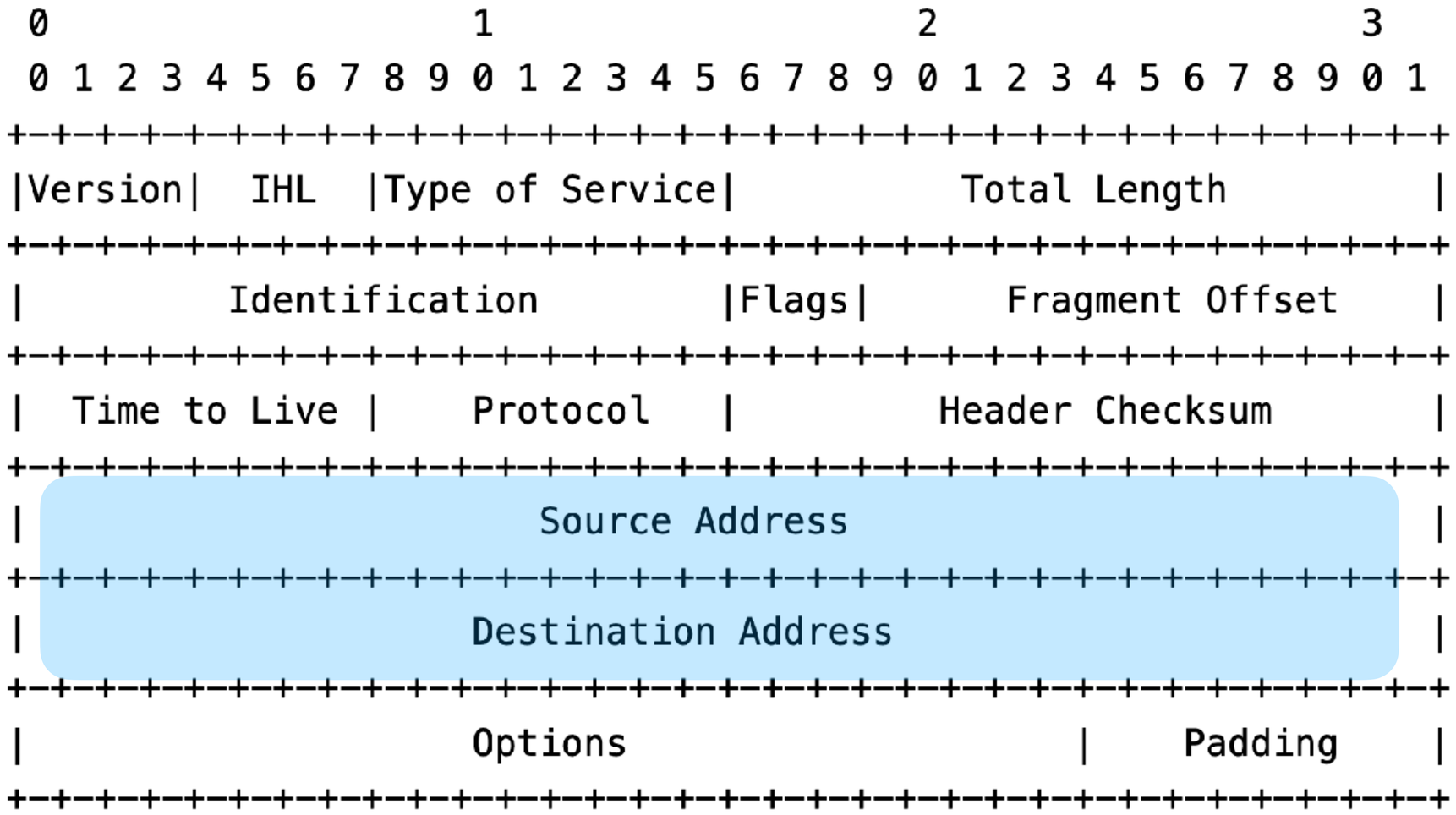
UDP, TCP

...



Example Internet Datagram Header

Layer 3 - IP



4 byte

4 byte

Example Internet Datagram Header

IP-adresser

- 4 byte til en adresse
- Vanlig notasjon skjer i base-10: 0-255 separert med '**.**'
- 127.0.0.1 158.39.201.59 8.8.8.8
- for å uttrykke et fast prefix (**subnet**): bruk **/** med antallet faste **bit**
- f.eks **192.168.64**.0**/24**
  alle adresser mellom **192.168.64**.0 og **192.168.64**.255
- **192.168.64.** er 3 * 8 = 24 bit

IP-adress routing

- Rutere bruker en **routing table**
- **dest.address -> (fysisk) nettverkstilkobling**
- **dest.address -> gateway**

IP-adresser

- Adresser i et subnet som 198.51.100.0/24 er tildelt fra nettleverandør
- top-nivå: ICANN
- ISPs kunngjør til andre på internet hvilke subnett de kan rute videre (BGP-protokollen)
- intern routing videre fra ruter til ruter frem til enheten

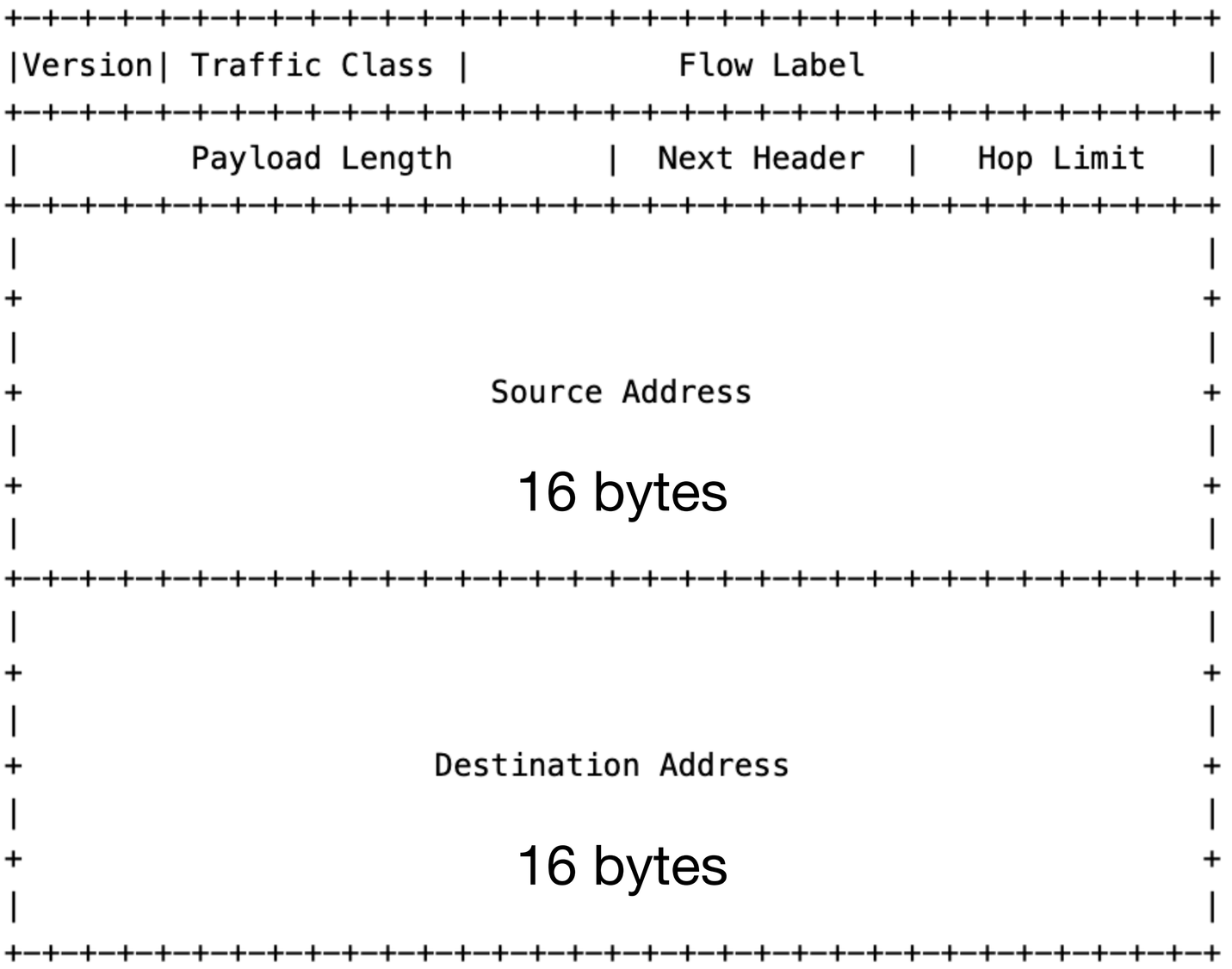
IP-adresser

- Noen ranges er reservert til ulike formål
- https://en.wikipedia.org/wiki/List_of_reserved_IP_addresses
- 127.0.0.0/8 - loopback (selve maskin)
- 192.168.0.0/16 172.16.0.0/12 10.0.0.0/8 - private, interne nettverk
- 169.254.0.0/16 - link-local uten tildelt IP

IP-adresser

- 4 byte i en adresse:
- maksimal 2^32 = bare 4,3 milliarder unike IP-adresser
- denne versionen av IP kallles for IPv4
- de fleste regionene i verden er tom for IPv4-adresser
- ulike løsninger: private nettverk bak NAT, dynamisk tildeling, ...
- ny version IPv6. Ikke kompatibel med IPv4.
- Layer 3 har nå 2 ulike valg: IPv4 eller IPv6. Applikasjoner må velge

IPv6-adresser



40 bytes

IPv6-adresser

- en adresse har 128 bit istedenfor 32 bit
- Vanlig skrivemåte i hex-base: 8 grupper med 2 byte (16 bit)

  0:0:..:0 kan avkortes med ::

  - 2001:700:2:8300::24b8
- ::1 - loopback (selve maskin)
- Private brukere burde få minst en /64 fra hjemme-ISP
  2001:db8:bacf:5010::/64
- 2^64 = 10^19 ledige adresser til hjemmenettverk

IPv6-adresser - Politikk

- IPv6 gjør **alle** enhetene adresserbare igjen,
  ideen bak internett av direkt desentral kommunikasjon mellom 2 enheter
- stabil drift i > 25 år nå
- fortsatt mange som ikke er klar til å bytte (github, BBC, ...)
- chicken-egg problem: hvem skal bytte først?
  særlig i regioner der IPv4 er fortsatt tilgjengelig
- dual-stack der an app prøver både IPv4 og IPv6 skjuler problemet

ICMP: meldinger om IP-layer

- ICMP er en enkel protokoll som sitter på toppen av IP direkte (ved siden av UDP/TCP)
- brukes for meldinger om IP routing
  - ***ping***: echo request / echo reply
  - Destination network unreachable
  - Destination host unreachable
  - ...



brukes til feilsøking i layer 3

f.eks virker *ping -c3 1.1.1.1* ?

5-layer model

