Oversigt

- UDP
- Reliable Data Transfer
  - Go-Back-N
  - Selective Repeat
- TCP
  - Flow control
  - Congestion control
- Eksamensopgaver

Transport layer

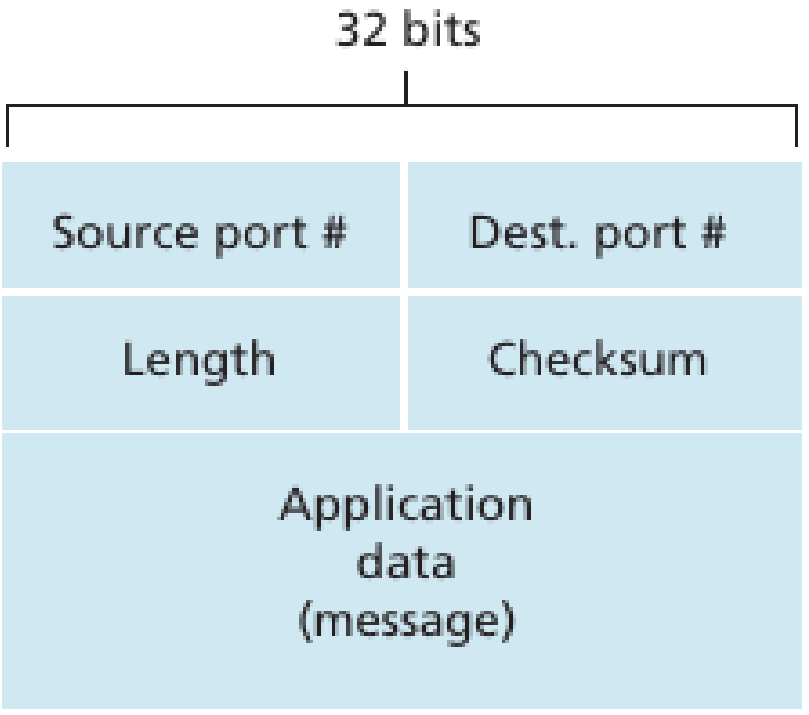
- Kommunikation mellem processer
- Afsender opdeler beskeder I mindre segmenter
- Modtager gendanner beskeder af segmenterne
- To primære protokoller: UDP og TCP

UDP

- 2 services:
  - process-to-process data delivery
  - error checking
- Unreliable data transfer:
  - ingen garanti for rækkefølge
  - eller om det overhovedet når frem

UDP

- UDP segment struktur
- Length: header + message
- Checksum: error detection



UDP Checksum

Word 1:	0110011001100000
Word 2:	0101010101010101

	1011101110110101
Word 3:	1000111100001100

	0100101011000010
checksum =	1011010100111101

Principles of Reliable Data Transfer

- Reliable data transfer:
  - garanteret ankomst (ved at gensende)
  - og i korrekt rækkefølge

Principles of Reliable Data Transfer

- Grundlæggende protokol: Stop and wait

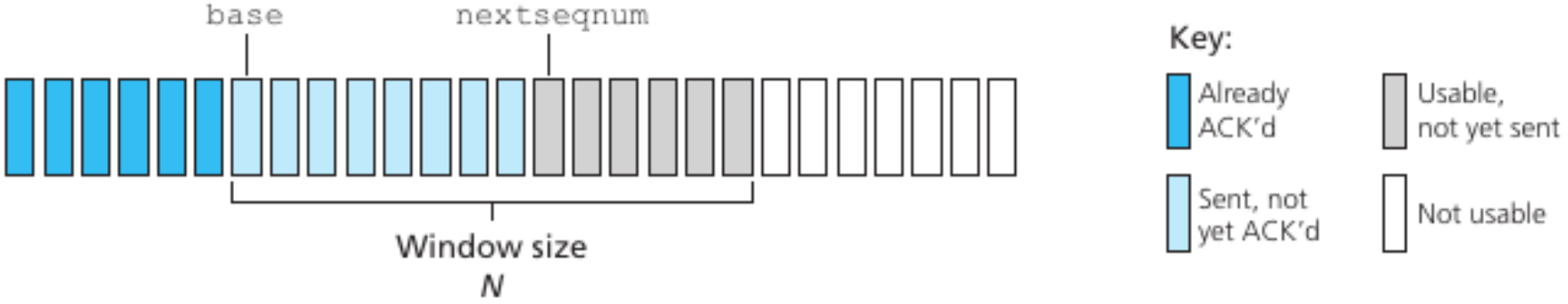
  - send pakke, vent på ACK eller timeout

  - Ikke specielt effektivt

  - vil gerne kunne sende flere pakker afsted inden der skal ventes

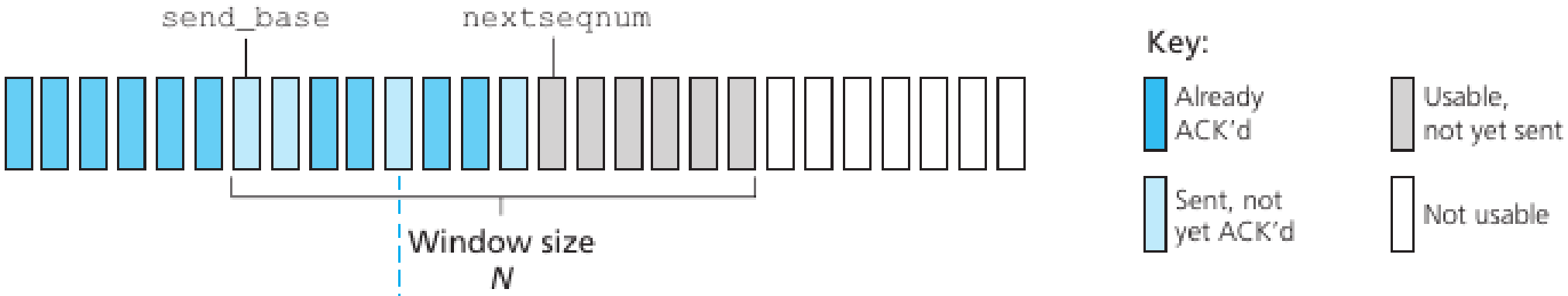
Principles of Reliable Data Transfer

- Go-Back-N:
  - max of *N* unacknowledged packets
  - Sender ACK for sidst korrekt modtaget
  - Smider out-of-order pakker væk



Principles of Reliable Data Transfer

- Selective-Repeat:
  - Window size som i GBN
  - Buffer out-of-order pakker
  - ACK for modtagne pakker



TCP

- Reliable data transfer:

  - garanteret ankomst

  - og i korrekt rækkefølge

- TCP control:

  - sikrer at links, routere og buffere ikke oversvømmes med data

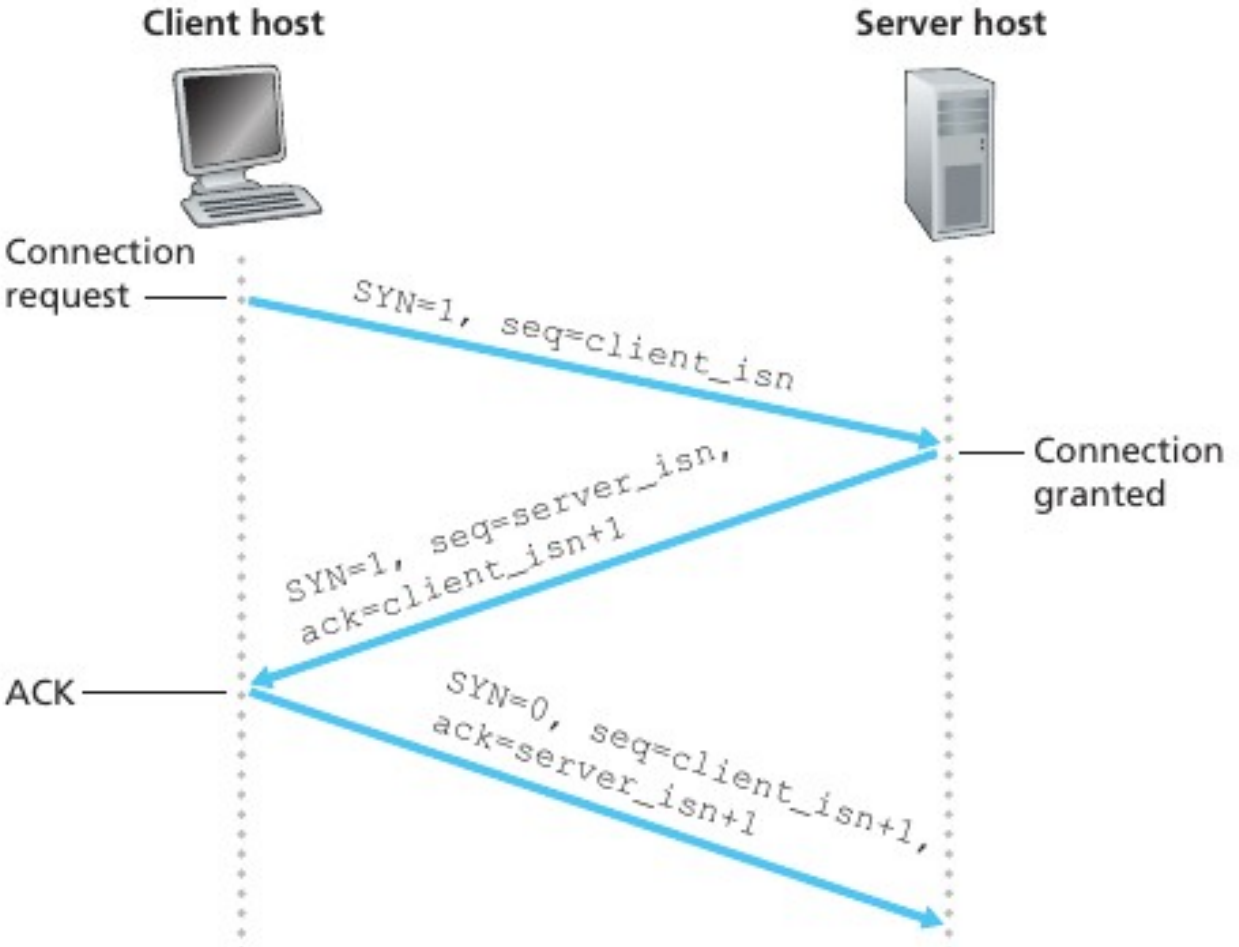
TCP

- Sequence number: nummeret af den første byte i segment (i den samlede data stream)
- ACK number: sequence num of next expected byte
- Kan være buffered eller ej (i praksis buffered)
- Fast retransmit

  - 3 duplicate ACKs

3-way handshake

- Connection - oriented



TCP Control

- TCP begrænser, hvor meget data vi sender afsted i forhold til:
  - Flow Control
  - Congestion Control

TCP Flow Control

- Sende for meget data i forhold til, hvor hurtigt modtageren bearbejder det

  - Overflow i modtagerens buffer

- Derfor *receive window* i headeren

  - hvor meget plads er der tilbage i bufferen

  - angives i TCP headeren

  - Udregnes: Buf_size – (last_b_received - last_b_read)

TCP Congestion Control

- Nu ikke receive buffer hos modtageren
- Men overflow af selve netværket:

  - sender mere data end hvad routers / links kan håndtere
- End-to-end congestion control

  - antagelser ud fra tabte pakker

TCP Congestion Control

- Congestion window, cwnd (hvor mange pakker sender vi per RTT), styret af *Loss events:*

  - opdages af timeout eller 3 dup ACKs

- 3 states:

  - slow start

  - congestion avoidance

  - fast recovery

TCP Congestion Control

- Slow start:
  - cwnd = 1 segment per RTT
  - For hver ACK, cwnd += 1 (exponential growth)
- Ved timeout loss:
  - set threshhold til cwnd/2
  - slow start igen, cwnd = 1
- Ved 3 dup ACKs, (ikke lige så drastisk, mange efterfølgende pakker er kommet frem):
  - set threshhold til cwnd/2
  - cwnd = (cwnd / 2) + 3
  - fast retransmit → fast recovery
- Når cwnd overstiger threshold → congestion control

TCP Congestion Control

- Congestion avoidance:
  - congestion kan være lige om hjørnet!
  - mere konservativ: cwnd += 1, hver RTT
- Ved loss events, samme som i slow start:
- Ved timeout loss:
  - set threshhold til cwnd/2
  - slow start igen, cwnd = 1
- Ved 3 dup ACKs, (ikke lige så drastisk, mange efterfølgende pakker er kommet frem):
  - set threshhold til cwnd/2
  - cwnd = (cwnd / 2) + 3
  - fast retransmit → fast recovery

TCP Congestion Control

- Fast recovery:

  - cwnd += 1, for hver duplicate ACK af pakke der fik TCP til at gå i fast recovery mode

- Ack for missing segment → Congestion avoidance

  - cwnd = threshold

- Timeout → Slow start

  - threshhold = cwnd / 2

  - cwnd = 1

TCP Congestion Control

- Congestion control algoritmen sikrer at TCP forbindelser deler et fælles link fair
- Forklaring i bogen afsnit 3.7.1

UPD vs TCP

- Kanon, hvorfor så overhovedet bruge UDP?

  - Ingen regulering af transmission rate

  - Mindre overhead

  - Multimedie applicationer, hvor enkelte tabte pakker ikke har den store betydning

  - Så bruges stadig selvom TCP dominerer

Opgaver

- Eksamenssæt
- Opgaverne fra øvelsesgangen om TCP er også gode
- Video på Absalon

References

- All figures from:
- KR: James F. Kurose, Keith W. Ross: Computer Networking: A Top-Down Approach International Edition, 7th and global edition