# Exercise 8

Florian Tegeler

# TCP congestion control

- N=200, RTT=200ms, MSS=1000 bytes, sender just sent a complete window!
  - a) Assuming no loss, what is the throughput (in terms of MSS and RTT and in terms of Megabit/s) of this message exchange?

$$throughput = \frac{segments \cdot MSS}{RTT} = \frac{200 \cdot 8000 Bit}{0.2s} = 8000000 \frac{Bit}{s} = 8 \frac{MBit}{s}$$

# TCP congestion control cont'd

- b) Suppose TCP is in its congestion avoidance phase. Assuming no loss, what is the window size (in terms of segment) after the N = 200 segments are acknowledged?
- From the lecture:
  - When **`CongWin`** is above **`Threshold`**, sender is in congestion-avoidance phase, window grows linearly.
  - **`CongWin`** is in units of MSS

# TCP sender congestion control

| State | Event | TCP Sender Action | Commentary |
|---|---|---|---|
| Slow Start (SS) | ACK receipt for previously unacked data | CongWin = CongWin + MSS, If (CongWin > Threshold) set state to "Congestion Avoidance" | Resulting in a doubling of CongWin every RTT |
| Congestion Avoidance (CA) | ACK receipt for previously unacked data | CongWin = CongWin+MSS * (MSS/CongWin) | Additive increase, resulting in increase of CongWin by 1 MSS every RTT |
| SS or CA | Loss event detected by triple duplicate ACK | Threshold = CongWin/2, CongWin = Threshold, Set state to "Congestion Avoidance" | Fast recovery, implementing multiplicative decrease. CongWin will not drop below 1 MSS. |
| SS or CA | Timeout | Threshold = CongWin/2, CongWin = 1 MSS, Set state to "Slow Start" | Enter slow start |
| SS or CA | Duplicate ACK | Increment duplicate ACK count for segment being acked | CongWin and Threshold not changed |

# TCP congestion control cont'd

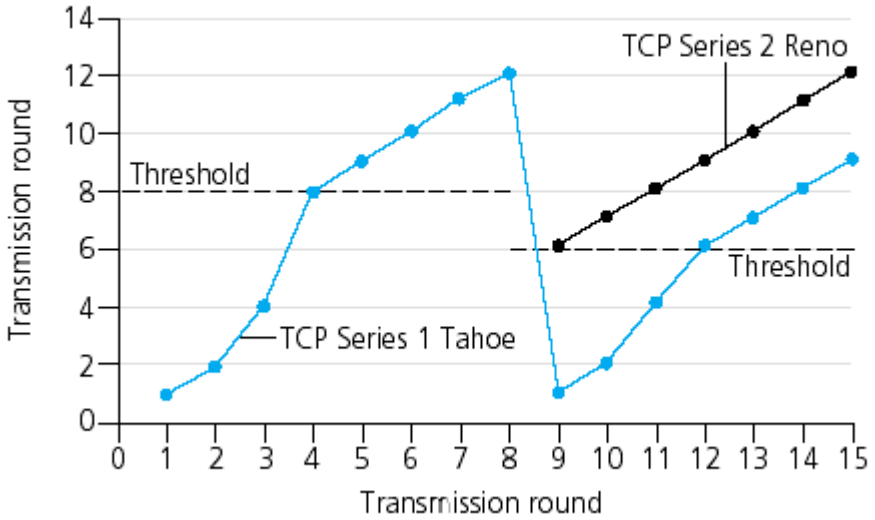
- b) Suppose TCP is in its congestion avoidance phase. Assuming no loss, what is the window size (in terms of segment) after the N = 200 segments are acknowledged? (CongWin = CW)

- in one RTT: $$CW = CW + MSS \cdot \left( \frac{MSS}{CW} \right)$$

- Each ack increases the CW by MSS/CW, which is 8000Bit/200=40Bit. As 200 acks arrive, the window is increased by 8000Bit which is exactly 1MSS, therefore CW=200+1!

# TCP-Reno and Tahoe

- What is the difference between the two congestion control algorithms TCP-Tahoe and TCP-Reno?
- Difference in handling time-outs and triple duplicate acks!
-

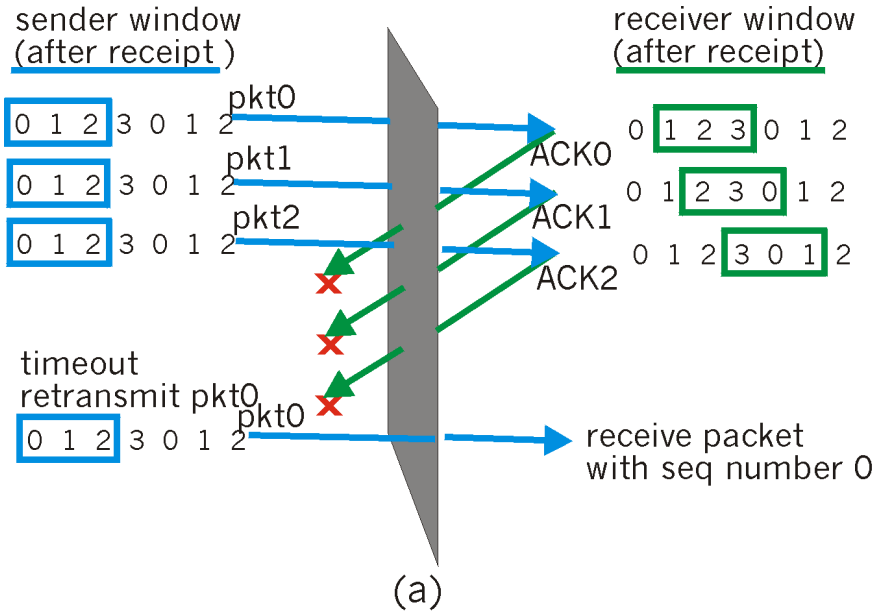


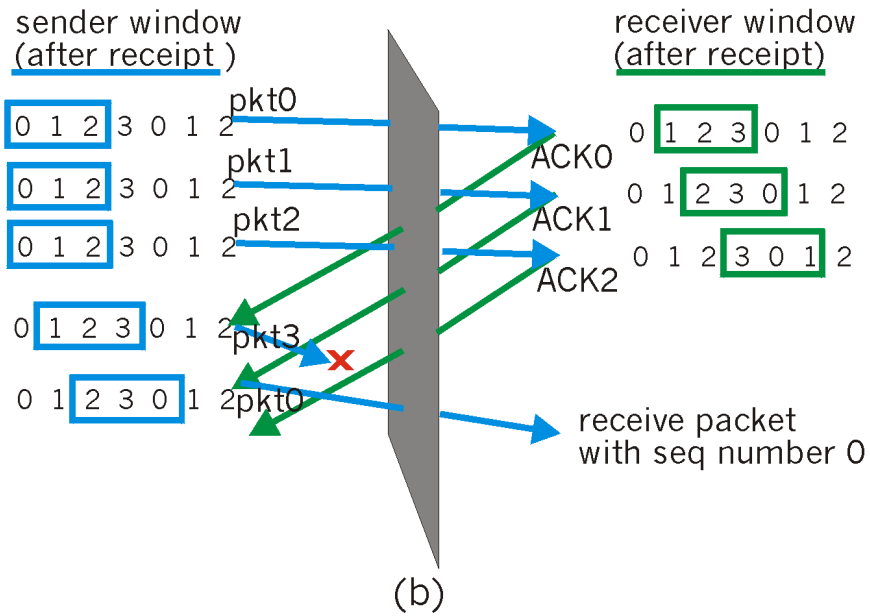

- Tahoe always down to 1MSS, Reno distinguishes: 3 duplicate ACKs-> go down to 50% then CA, timeout -> go down to 1MSS

# Selective Repeat

- Please explain the selective repeat dilemma and name a solution to prevent its occurrence.

- Dilemma occurs on a limited sequence range and large window size. Solution: Window size should be maximally half of the sequence range!


sender window
(after receipt )
receiver window
(after receipt)
pkt0
pkt1
pkt2
ACK0
ACK1
ACK2
timeout
retransmit pkt0
pkt0
receive packet
with seq number 0
(a)


sender window
(after receipt )
receiver window
(after receipt)
pkt0
pkt1
pkt2
pkt3
ACK0
ACK1
ACK2
pkt0
receive packet
with seq number 0
(b)

# TCP vs. UDP

- Please name at least three differences between UDP and TCP.
  - TCP is connection oriented, UDP is not
  - TCP is a reliable data transfer protocol, UDP is not reliable
  - TCP enables in-order delivery, UDP does not guarantee in-order deliver
  - UDP has less overhead (lightweight) compared to TCP (heavy load due to ordering, window maintenance etc...)
  - TCP uses flow control, UDP does not
  - TCP uses congestion control, UDP does not

# Choosing a protocol

- If you would like to transfer a file, which transport protocol would you use? Which protocol would you use for voice traffic?
  - File: TCP as it is reliable, in-order delivery. Receiver can directly pipe data contents into file
  - Voice: UDP as it is lightweight, small in-orders cannot be heard and reliability has no advantage if delivery takes to long

# TCP fast retransmit

- Please explain TCP fast retransmit.
  - Time-out period  often relatively long:
    - long delay before resending lost packet
  - Detect lost segments via three duplicate ACKs.

Fast retransmit:

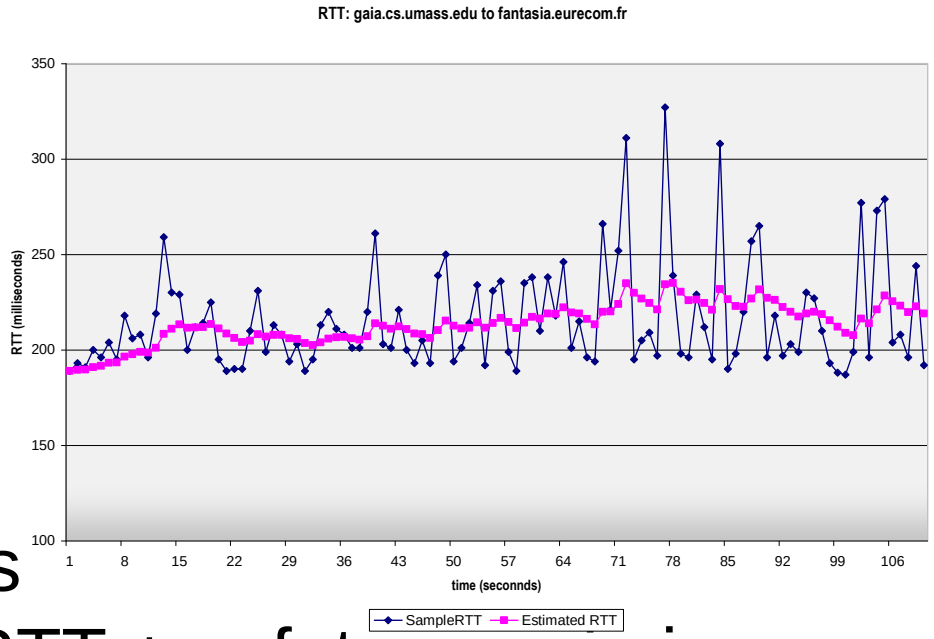
resend segment before timer expires, directly after receiving three duplicate acks

# Flow vs. congestion control

- What is the difference between flow control and congestion control?
- Flow Control: Prevent overwhelming the receiver by sending too much data. Reduce sending rate if receiver's buffer fills up.
- Congestion Control: React on congestion in the network (on the path to the receiver). Reduce sending rates based on congestion observation (deduction by seeing delayed acks, lost acks etc.)

# Estimated vs. sampled RTT

- Why is an EstimatedRTT used to calculate the TCP timeout instead of the recently sampled RTT?
  - Exponential weighted moving average
  - influence of past sample decreases exponentially fast




- SampleRTT fluctuates too much. EstimatedRTT + safety margin is a safer guess to set the timer.