Computer Networks WS20/21

Exercise 4

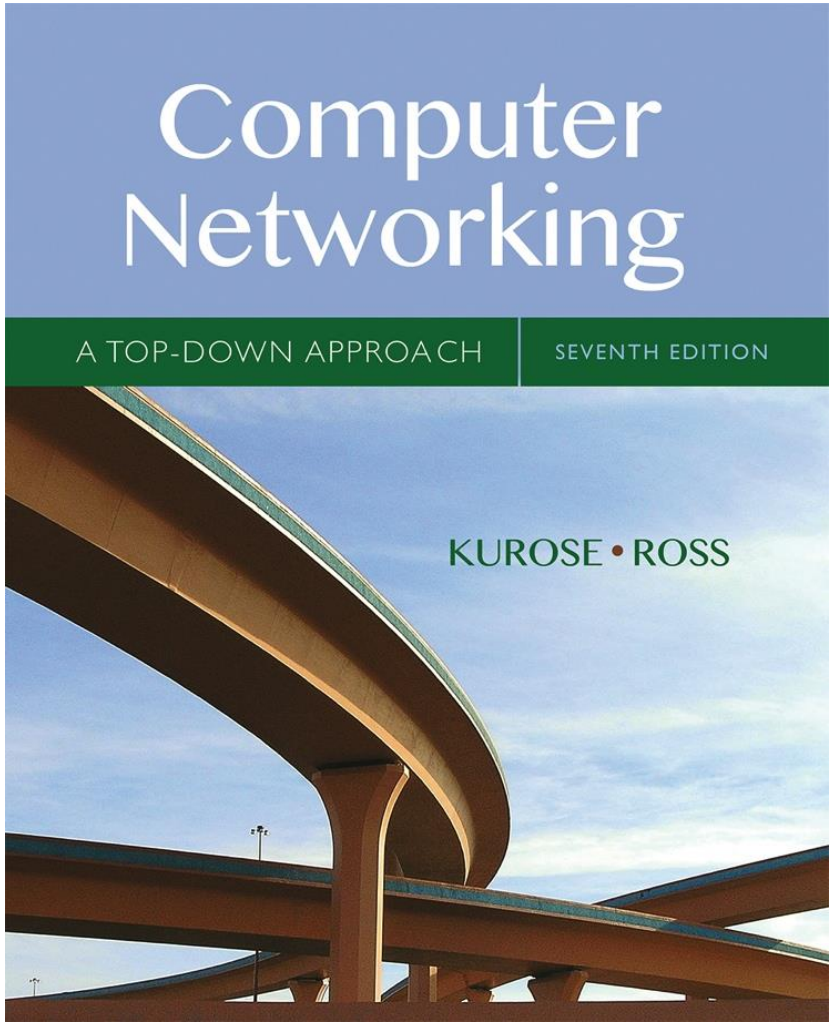
Recommendation

Try to borrow (or buy) this book:

Computer Networking: A Top Down Approach

7th edition. Jim Kurose, Keith Ross, Pearson, 2019.

It is very good to understand!



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?

  - Sender has just sent a complete window size of segments:

    Number of segments = N = 200

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

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 congestion window is above threshold, sender is in congestion-avoidance phase, window grows linearly
  - MSS is the unit of the congestion window

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: $$CW_{n+1} = CW_n + \mathrm{MSS} \cdot \left(\frac{\mathrm{MSS}}{CW_n}\right)$$

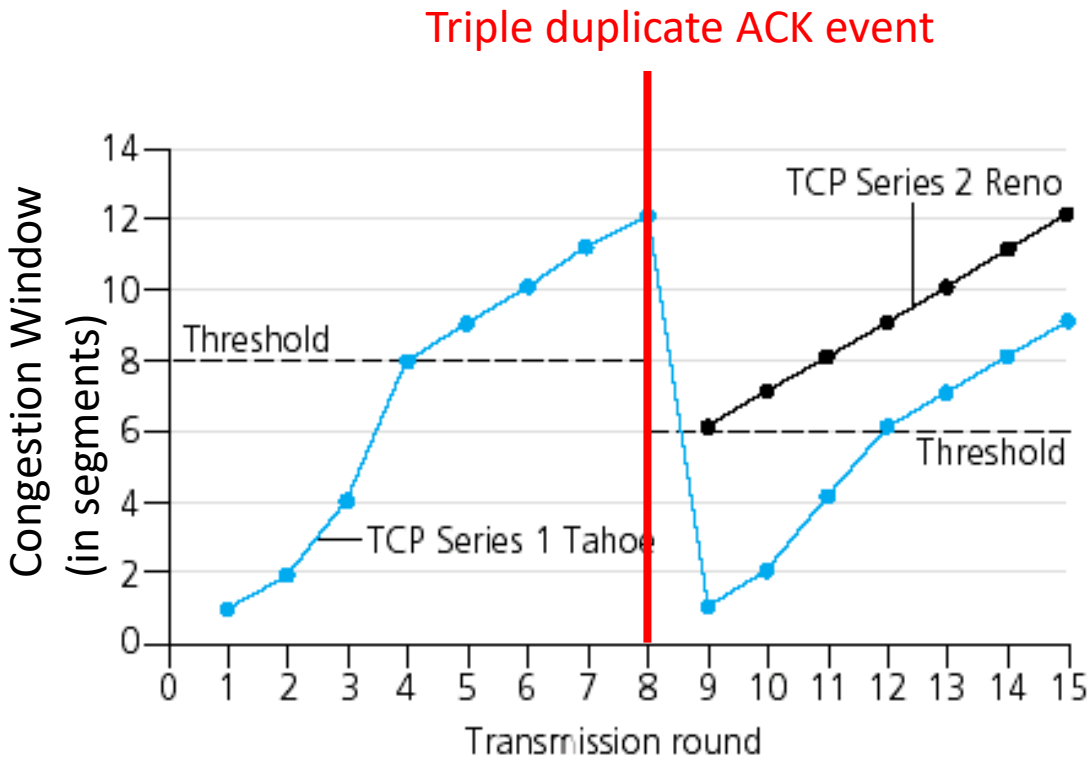
- Each ACK increases the CW by MSS/CW, which is 8000Bit/200=40Bit. As 200 ACKs arrive, the window is increased by 200*40Bit = 8000Bit which is exactly 1MSS, therefore CW=200+1 MSS = 201 MSS!
- Congestion Avoidance: Additive increase, resulting in increase of CongWin by 1 MSS every RTT

TCP-Reno and Tahoe

- What is the difference between the two congestion control algorithms TCP-Tahoe and TCP-Reno?

TCP-Reno and Tahoe

- Difference in handling loss events (time-outs and triple duplicate ACKs)!
- Tahoe: older TCP version
  - When time-out or triple duplicate ACKs: Tahoe cuts the congestion window always to 1MSS and enters slow-start phase
- Reno: newer TCP version
  - 3 duplicate ACKs: cuts down to 50% of congestion window and then enters congestion avoidance phase
  - Time-out: cuts down to 1MSS and enters slow-start phase (just like Tahoe)

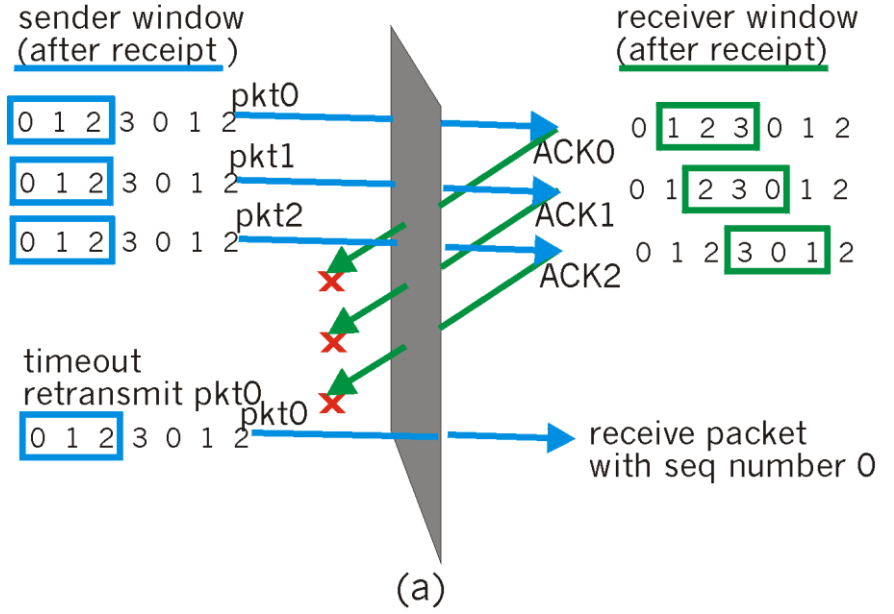


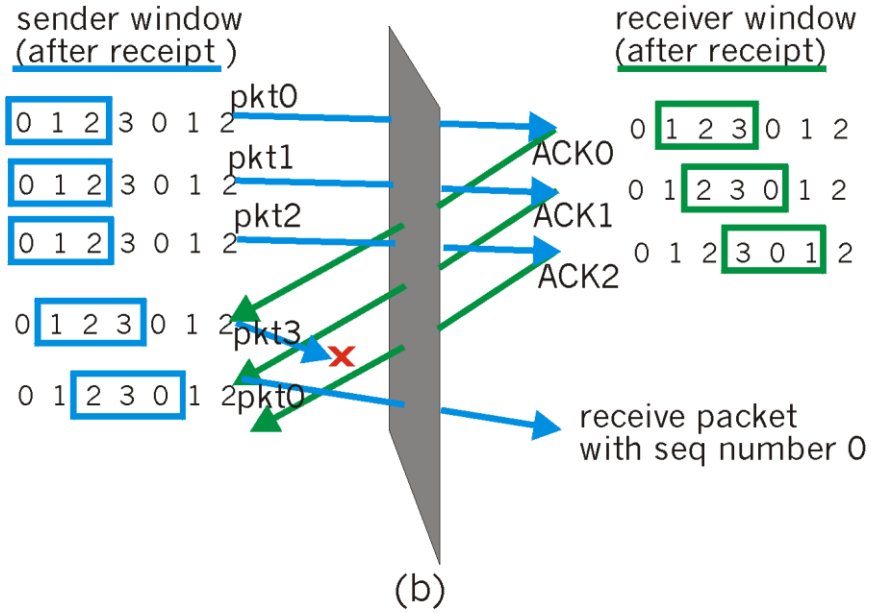
Selective Repeat

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

Selective Repeat

- Two different scenarios with packet sequence number 0, 1, 2, 3 and window size of 3:
  1. Receiver's ACK messages get lost
     -> Sender retransmits packet with sequence number 0 after timeout
  2. ACKs don't get lost, but packet number 3 gets lost
     -> Sender transmits next packet with number 0
- Receiver sees no difference in these scenarios. It can not see whether the arriving packet is new or a duplicate retransmitted packet
- Dilemma occurs on a limited sequence range and a (compared to this range) relatively large window size
- Solution: Window size should be maximally half of the sequence range!





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 and has in-order delivery. Receiver can directly pipe data contents into the file in the right order
- Voice: UDP as it is lightweight, small out-of-order packets cannot be heard and reliability has no advantage if delivery takes too long

TCP fast retransmit

- Please explain TCP fast retransmit.

  - Sender's time-out period is often relatively long

    => Long delay before resending lost packet
  - Use duplicate ACK: ACK that reacknowledges a segment for which the sender has already received an earlier acknowledgment
  - If the TCP sender receives three duplicate ACKs for the same data, it takes this as an indication that the segment following the segment, that has been ACKed three times, has been lost
  - Sender can retransmit this lost packet even before timer runs out

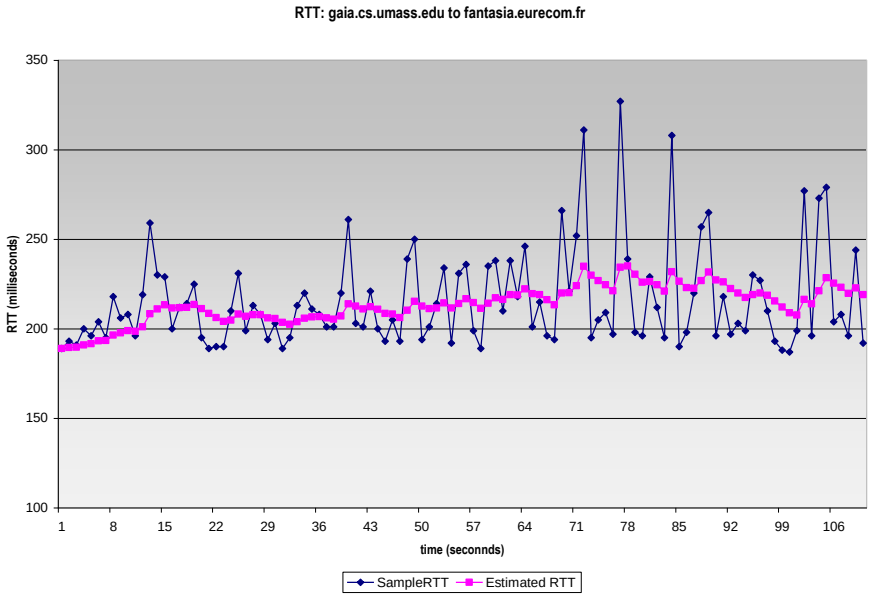
Flow vs. congestion control

- 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: Reacting on congestion in the network (on the path to the receiver). Reducing 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?



- SampleRTT:
  - Measured time from current segment transmission until ACK receipt
- EstimatedRTT:
  - 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.

Any Questions?

Mail us:

Yachao Shao: yachao.shao@cs.uni-goettingen.de
Fabian Wölk: fabian.woelk@cs.uni-goettingen.de