Computer Networks Group

University of Göttingen, Germany

Homework 8

(Due on 12:00am, Thursday, Dec. 17th, 2009)

1. A) Please calculate the utilization rate of the sender U_{sender} for the stop and wait protocol scenario:

   Packet size (L): 2 000 bits (including header)
   Transmission rate (R): 2 Mbit/s
   Round Trip Time (RTT): 200 ms

   B) How big would the window size need to be for channel utilization U_{sender} to be greater than 95%?

   Answer A): $d_{trans} = \frac{L}{R} = \frac{2\,000 \frac{bits}{packet}}{20^6\ bits/sec} = 0{,}001 \frac{packets}{second}$

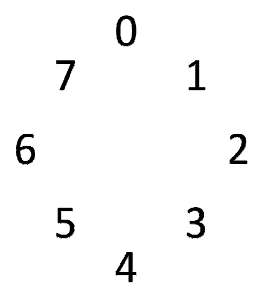
   $$U_{sender} = \frac{L/R}{RTT} = \frac{0{,}001}{0{,}2} = 0{,}005 = 0{,}5\%$$

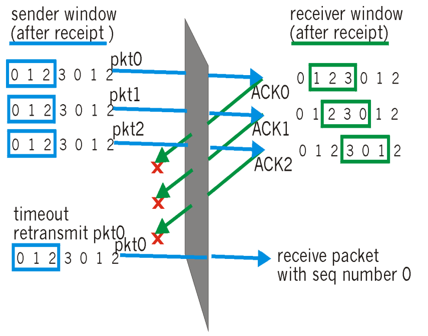
   Notice: We ignore the (negligible) transmission time in the division for simplicity. Usually RTT + L/R should be in the denominator.

   *Answer B): Now use pipelined protocol, i.e. send up to N packets without ACK. d_{trans} multiplies times 190 to achieve U_{sender} = 95% (0,5% * N = 95% -> N = 190).*

2. Selective Repeat Dilemma. A limited sequence range leads to the dilemma of the receiver not being able to distinguish whether a new packet is received or an old (lost/corrupt) packet is retransmitted. What is the cause for this dilemma, discuss the solution.

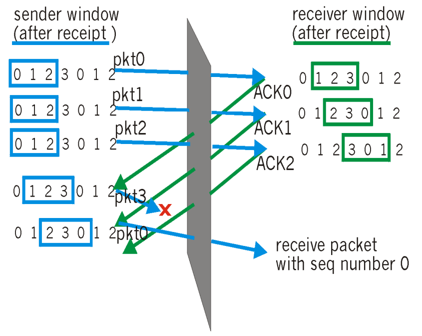
Answer: The dilemma occurs on a limited sequence range and a large window size.





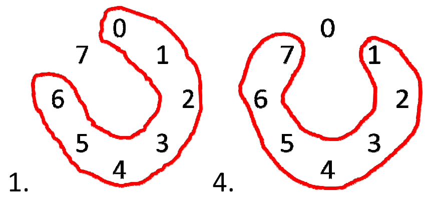
(a)

The Sequence range can be seen as a circle, as soon as the last [7] sequence number has been used, the first [0] is reused.

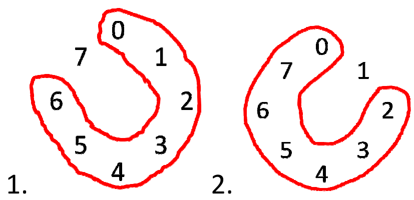


(b)

Here a window size of [range – 1] is too large.



Sender



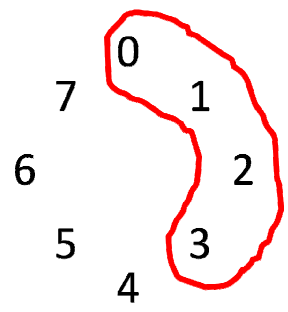
Receiver

Sender:

- 1. Transmission of n packets (7).
- 3. ACK for packet 0 has been lost on network, timer expires and sender resends packet 0.
- 4. ACK for packet 1 arrives at sender, window slides forward.

Receiver:

- 2. Packet 0 and packet 1 has been received, ACK's sent to sender, receiver window slides forward.
- 5. Receiver expects new packet with sequence number 0. **BUT**: The old packet with sequence number 0 is received.



Window size of n = [range/2] is more appropriate. **But why?**

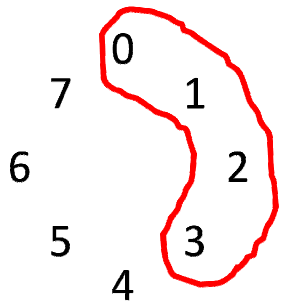
Oldest "in-flight" packet (Packet 0) is significant for the dilemma.
Worst case: all packets arrive at receiver, no AKS's arrive at the sender.

Case: In a chat, assuming each character is transferred in a separate segment the following would be delivered.

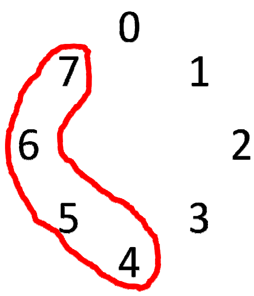
Sender Sends: "HI BILL" and retransmits H
Receiver delivers: "HI BILLH"

- Receiver slides n=[range/2] forward. Next expected packets packet 4-7.
- Sender resends all packets 0-3.
- Receiver resends ACK's.
-

Window size of [range/2] makes sure that start of sender window and end of receiver windows do not overlap in worst case of all los ACK's.



Sender cannot send more than [range/2] packets.



Receiver cannot slide more than [range/2] packets forward.

3. TCP Congestion Control. Suppose that in TCP, the sender window is of segment size N = *200*, the base of the window is at sequence number 600, and the sender has just sent a complete window size of segments. Let RTT be the sender-to-receiver-to-sender round trip time of 200 ms and Maximum Segment Size MSS = 1 000 **bytes**.

   a) Assuming no loss, what is the throughput (in terms of MSS and RTT) of this message exchange?

   Answer: Since the sender windows is 200 segments. The sender can thus send 200 segments, each of size 8000 Bit every 200ms sec. The throughput is 200 8000Bits/0,2 sec. = 8 mn Bits/s or 8 Megabit/s.*

   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?

   Answer: TCP handles congestion window in MSS units, it starts in slow start mode with cwnd of only one MSS. i.e. one segment.

   Slow Start mode:

$$cwnd = 1MSS$$

$$\frac{new\ ACK}{cwnd = cwnd + MSS}$$

Congestion Avoidance Mode: $\frac{new\ ACK}{cwnd = cwnd + MSS * (\frac{MSS}{cwnd})}$

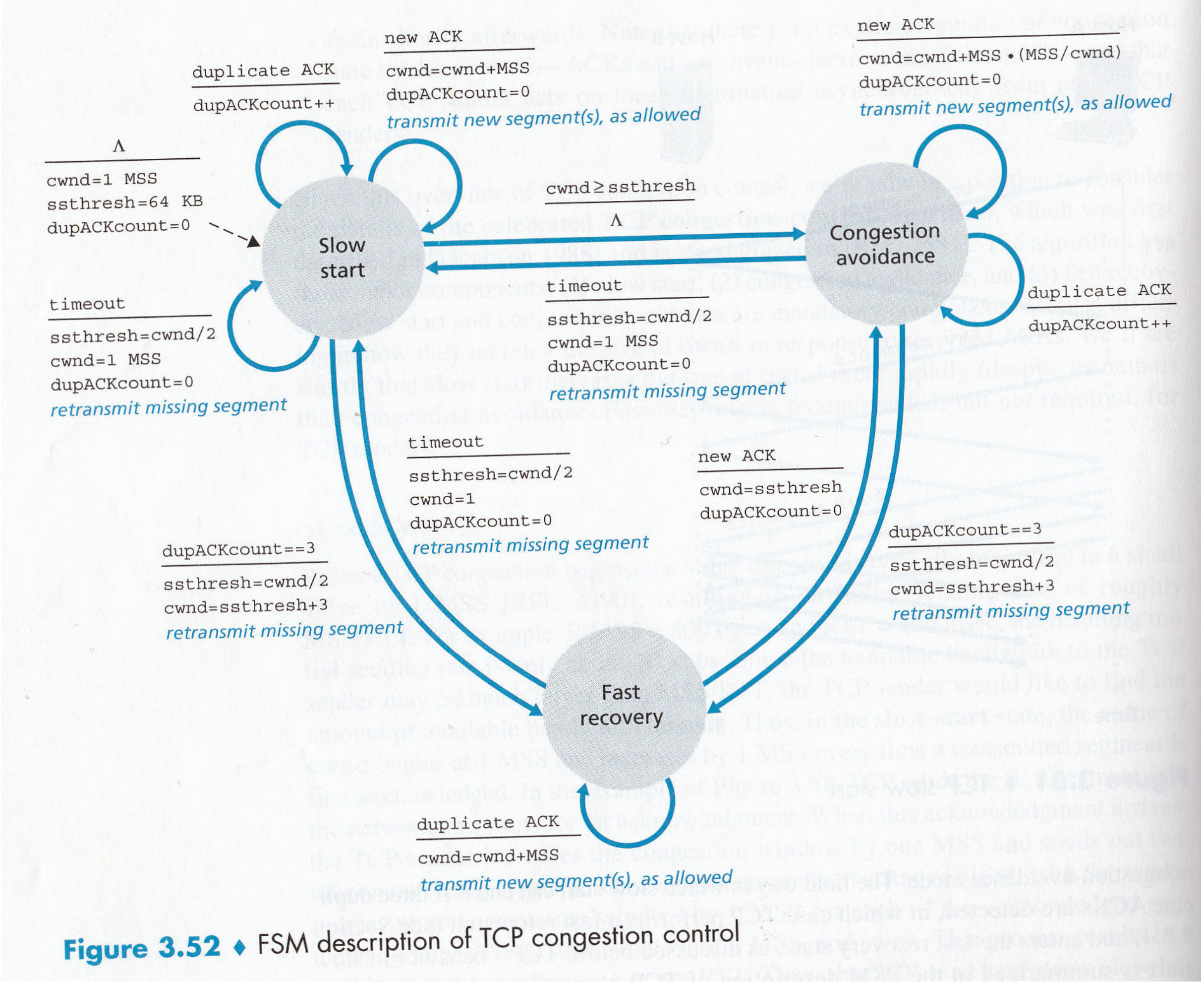


Figure 3.52 ♦ FSM description of TCP congestion control

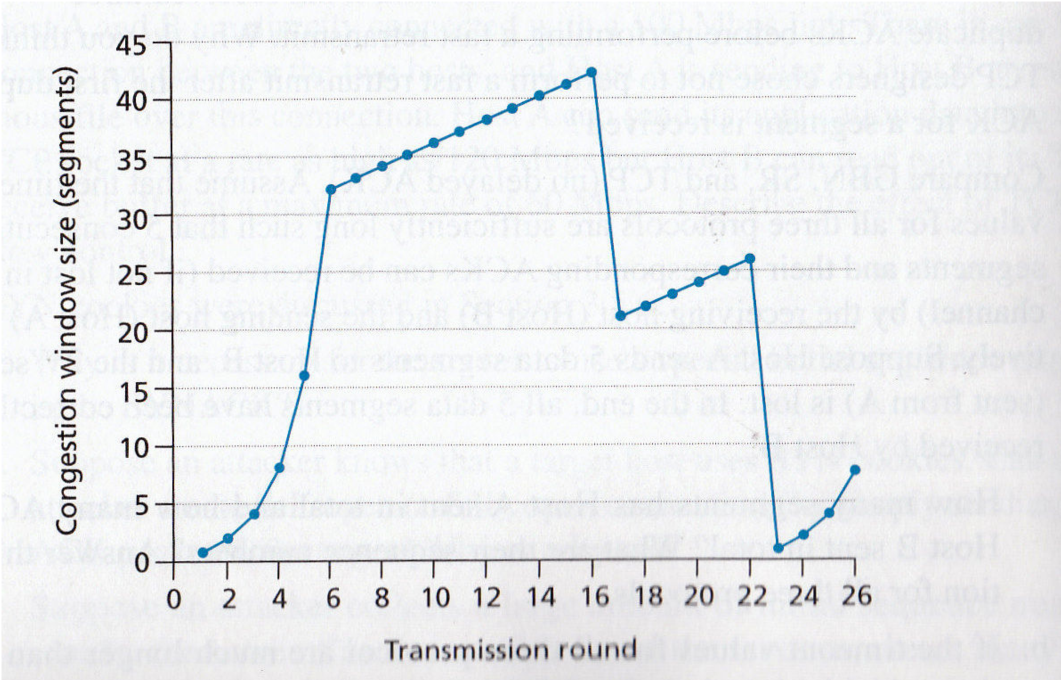
Source: Computer Networks, Kurose, Chapter 3, Figure 3.52, Fifth Edition

*Each ACK that arrives in congestion avoidance mode leads to incrementation of cwnd by MSS/cwnd, a fraction of one MSS, in this case 8000 Bit/200 = 40 Bits. After all 200 ACK arrived, cwnd will have increased to additional 8000 Bit (40 Bit * 200 ACK's) which is one MSS.*

Notice: N = cwnd/MSS in this case the window size has increased to **N + 1 = 201**.

4. Please consider the following figure, TCP Reno is used with fast recovery mode.
   a. In what interval of time does slow start mode operate?
   b. Identify the intervals of time when TCP congestion avoidance is operating.
   c. After the 16th transmission round, is segment loss detected by a triple duplicate ACK or by a time out?

d. After the 22nd transmission round, is segment loss detected by a triple duplicate ACK or by a time out?



Answer:

a. 0-6 (because of exponential (doubling) congestion window growth.
b. 6-16 (because of linear (additive) congestion window growth.
c. Triple duplicate ACK (because congestion window is set to half of threshold and grows additively again).
d. Timeout (congestion window is reset to 1 MSS).

Telematik - Computer Networks and the Internet

WS2009/2010