Francisco Quezada

This work is mine unless otherwise cited.

[2]

a) ack=10
b) ack=30
c) 40 bytes
d) seq=30
e) ssthresh = cwnd/2 = 40/2 = 20
f) cwnd > ssthresh → 40>20 → congestion avoidance

[3]
a) 10
b) 0
c) no
d) slow start

[4]
a) 80/2 = 40
b) 80
c)seq=20
d) congestion avoidance

[5]
a)
slow start:
- 1-4
- 9-12
- 14-17
- 32-35
- 39-40

congestion avoidance:
- 4-8
- 12-13
- 17-19
- 20-28
- 29-31
- 35-38

fast recovery:
- 14-32 (TCP Reno; according to book Fig.3.53)

b)
- 8-9, timeout
- 13-14, timeout
- 19-20, triple ack
- 28-29, triple ack
- 31-32, timeout

38-39, timeout

c)
ssthresh is (“looks”) the same at:
4, 12, 17, 20, 35 → ssthresh = 8
ssthresh changes at:
29 → ssthresh = 11

[6]
a) Not in a reliable data transfer protocol (TCP) because it does not allow the sender to send out of order packets and if they are sent the receiver asks to resend them. There is an exception, let's say if a packet is lost then it would be out of order (e.g. packets 1,2,3 are sent and packet 2 is lost so packets 1,3 are received, thus out order; tcp receiver asks to resend packets). Although, recall that the packets have no sequence numbers in this scenario so it may be a possibility.
b) Corruption of acknowledgements, packets, etc. → retransmission; I do not know if it possible in this scenario because we do assume there are no bit errors.
c) Allowing the receiver to send negative acknowledgements (NAKs)