Alex Wauck
Chapter 3 Homework

R16

After typing 'R', the following segments are sent:
Host A to Host B: Seq=43, ACK=80, data='R'
Host B to Host A: Seq=80, ACK=44, data='R'
Host A to Host B: Seq=44, ACK=81

P33

a

Slow start appears to be in effect during the intervals 1-5 and 23-26, as the graph starts low, but appears to increase exponentially.

b

Congestion avoidance appears to be in effect during the intervals 5-16 and 18-22, as the graph increases linearly during these intervals.

c

The graph decreases by half after round 16, which is indicative of fast recovery, so it must have been triggered by three duplicate ACKs.

d

The graph decreases to its minimum after the 22nd round, so the connection must now be in slow start mode, meaning it detected a timeout.

e

The connection appears to transition from slow start to congestion avoidance after round 6, where the congestion window appears to be 32. So, the threshold must have been 32.

f

The congestion window appears to be at 42 right before the fast recovery drop, so the threshold at the 18th round must be 21.

g

At round 24, the window has been reduced to the minimum due to a timeout. Before the timeout, the window was 26, so the threshold should now be 13.

h

Round 1 sends 1 segment, round 2 sends 2 segments, round 3 send 4 segments, etc. So, by the 6th round, $1+2+4+8+16+32 = 63$ segments have been sent. The 7th round sends 33 segments, so the 70th segment will be sent during the 7th round.

i

If a triple duplicate ACK is encountered after the 26th round, the threshold will be set to half the size of the congestion window, and the window will be set to the threshold. Since the window was 8, it and the threshold are now 4.

3.x1

1. False. Packets often contain more than 1 byte of data, so the sequence number for the next (by sequence number) packet will most likely have a sequence number that is more than 1 greater. Even if we disregard sequence, and merely consider the next packet to arrive, it will either be the next in the sequence (and so have a substantially higher number) or a retransmission, in which case it will have the same sequence number.

2. False. If a TCP sender's receive window is smaller than its congestion window, then it must wait after sending as much data as the receive window.

3. False. A TCP receiver will respond to a corrupt packet by ACKing the last good packet.

4. True

5. False. In congestion-avoidance mode, TCP will increase its congestion window by a fraction of the maximum segment size.

6. False. If segments continue to go through fast enough after the loss, three duplicate ACKs may be received before the lost segment times out.

3.x3

If the second part of a divided ACK takes an unusually long time to arrive, the sender may erroneously conclude (due to a timeout) that the segment from 2000 to 2459 was dropped. With congestion control, the sender may erroneously enter slow-start mode due to the same condition. With congestion control and no losses, delays, or reorderings, the extra ACK will result in an incorrect congestion window size increase. Only one segment will have been received, but it will be ACKed twice, causing the window to increase twice for the one segment. This may result in hitting the threshold or causing congestion sooner than would otherwise have been expected. For example, if proper congestion window for a given set of network conditions is 33 segments, the threshold is 32, and the current window is 16, then normal ACKing would result in a new window size of 32 segments. The subsequent congestion-avoidance increase would take the congestion window to 33 segments. With the divided ACK, the new window size would be 64, which would probably be way too much.