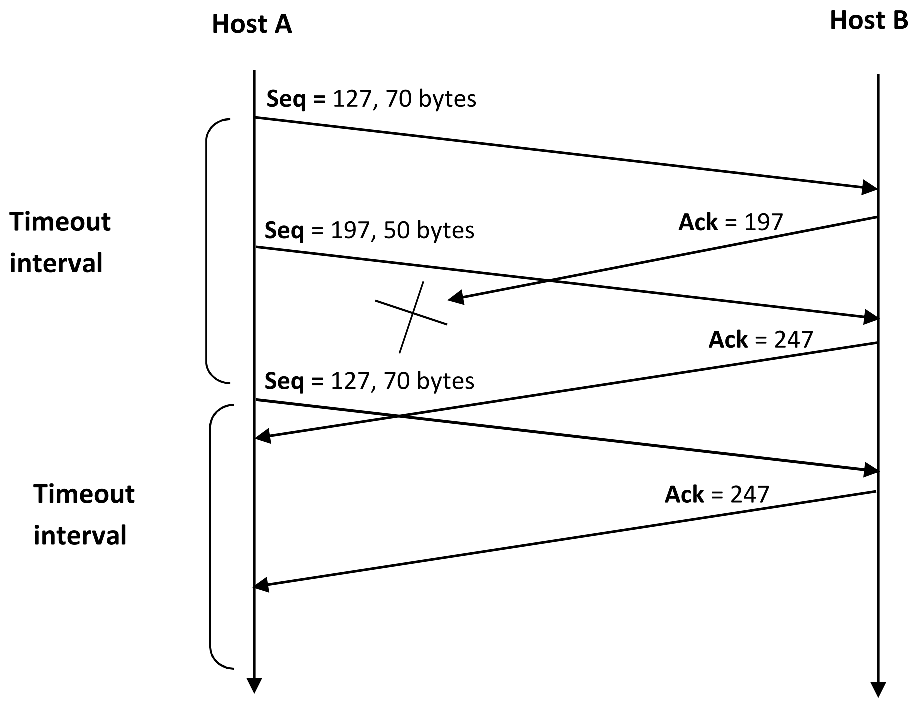
Homework 2 Solutions

1. **<u>P25:</u>**

   a. In the second segment from Host A to B, the sequence number is 197, source port number is 302 and destination port number is 80.
   b. If the first segment arrives before the second, in the acknowledgement of the first arriving segment, the acknowledgement number is 197, the source port number is 80 and the destination port number is 302.
   c. If the second segment arrives before the first segment, in the acknowledgement of the first arriving segment, the acknowledgement number is 127, indicating that it is still waiting for bytes 127 and onwards.
   d.



2. **<u>P34:</u>**

Note that there is a typo in the textbook.
part b, “5” is missing in “ $\cdot RTT$ ”.
“If the timeout values for all three protocol are much longer than $\cdot RTT$ ”
→
“If the timeout values for all three protocol are much longer than $5 \cdot RTT$ ”

a).
GoBackN:

A sends 9 segments in total. They are initially sent segments 1, 2, 3, 4, 5 and later re-sent segments 2, 3, 4, and 5.
B sends 8 ACKs. They are 4 ACKS with sequence number 1, and 4 ACKS with sequence numbers 2, 3, 4, and 5.

Selective Repeat:
A sends 6 segments in total. They are initially sent segments 1, 2, 3, 4, 5 and later re-sent segments 2.
B sends 5 ACKs. They are 4 ACKS with sequence number 1, 3, 4, 5. And there is one ACK with sequence number 2.

TCP:
A sends 6 segments in total. They are initially sent segments 1, 2, 3, 4, 5 and later re-sent segments 2.
B sends 5 ACKs. They are 4 ACKS with sequence number 2. There is one ACK with sequence numbers 6. Note that TCP always send an ACK with expected sequence number.

b). TCP. This is because TCP uses fast retransmit without waiting until time out.

3. **<u>P43:</u>**

Let W denote the max window size measured in segments. Then, W*MSS/RTT = 10Mbps, as packets will be dropped if the maximum sending rate exceeds link capacity. Thus, we have W*1500*8/0.1=10*10^6, then W is about 84 (ceiling of 83.3) segments.
As congestion window size varies from W/2 to W, then the average window size is 0.75W=63 segments. Average throughput is 63*1500*8/0.1=7.56Mbps.
84/2 *0.1= 4.2 seconds, as the number of RTTs (that this TCP connections needs in order to increase its window size from W/2 to W) is given by W/2. Recall the window size increases by one in each RTT.

4. **<u>P47:</u>**

a. The key difference between C1 and C2 is that C1's RTT is only half of that of C2. Thus C1 adjusts its window size after 100 msec, but C2 adjusts its window size after 200 msec.
Assume that whenever a loss event happens, C1 receives it after 100msec and C2 receives it after 200msec.

We further have the following simplified model of TCP.
After each RTT, a connection determines if it should increase window size or not. For C1, we compute the average total sending rate in the link in the previous 100 msec. If that rate exceeds the link capacity, then we assume that C1 detects loss and reduces its window size. But for C2, we compute the average total sending rate in the link in the

previous 200msec. If that rate exceeds the link capacity, then we assume that C2 detects loss and reduces its window size.
Note that it is possible that the average sending rate in last 100msec is higher than the link capacity, but the average sending rate in last 200msec is smaller than or equal to the link capacity, then in this case, we assume that C1 will experience loss event but C2 will not.

The following table describes the evolution of window sizes and sending rates based on the above assumptions.

	C1		C2	
Time (msec)	Window Size (num. of segments sent in next 100msec)	Average data sending rate (segments per second, =Window/0.1)	Window Size(num. of segments sent in next 200msec)	Average data sending rate (segments per second, =Window/0.2)
0	10	100 (in [0-100]msec]	10	50 (in [0-100]msec)
100	5 (decreases window size as the avg. total sending rate to the link in **last 100msec** is 150= 100+50)	50 (in [100-200]msec]		50 (in [100-200]msec)
200	2 (decreases window size as the avg. total sending rate to the link in **last 100msec** is	20	5 (decreases window size as the avg. total sending rate to the link in **last 200msec** is	25

	100= 50+50)		125= (100+50)/2 + (50+50)/2)	
300	1 (decreases window size as the avg. total sending rate to the link in last 100msec is 45= (20+25)	10		25
400	1 (no further decrease, as window size is already 1)	10	2 (decreases window size as the avg. total sending rate to the link in **last 200msec** is 40= (20+10)/2 + (25+25)/2)	10
500	2	20		10
600	3	30	3	15
700	1	10		15
800	2	20	1	5
900	3	30		5
1000	1 (decreases window size	10	2 (**increases** window size	10

	as the avg. total sending rate to the link in last 100msec is 35= (30+5)		as the avg. total sending rate to the link in **last 200msec** is 30= (20+30)/2 + (5+5)/2)	
1100	2	20		10
1200	3	30	3	15
1300	1	10		15
1400	2	20	1	5
1500	3	30		5
1600	1	10	2	10
1700	2	20		10
1800	3	30	3	15
1900	1	10		15
2000	2	20	1	5
2100	3	30		5
2200	1	10	2	10

Based on the above table, we find that after 2200 msec, C1's window size is 1 segment and C2's window size is 2 segments.

b. No. In the long run, C1's bandwidth share is roughly twice as that of C2's, because C1 has shorter RTT, only half of that of C2, so C1 can adjust its window size twice as fast as C2. If we look at the above table, we can see a cycle every 600msec, e.g. from 1400msec to 1900msec, inclusive. Within a cycle, the sending rate of C1 is (20+30+10+20+30+10)/6 = 120, which is twice as large as the sending of C2 given by (5+5+10+10+15+15)/6=60.

5. **Socket Programming Assignment 1:**

   See the separate html file.

6. **Wireshark TCP:**

   See the separate pdf file.