Comp 4320 Homework 3

Haden Stuart – has0027

1.) Sequence numbers are primarily used to tell whether a packet is a duplicate or not. We also know that once an ACK packet is received, the receiver moves onto the next state. Knowing this, we can come to the conclusion that ACK packets do not need a sequence number because once the ACK packet is received, the receiver moves to the next state which will cause any duplicate ACK packets to be ignored.

2.) For each sample we will use the following equations:

EstimatedRTT = α * SampleRTT + (1-α) * EstimatedRTT

DevRTT = β * |SampleRTT – EstimatedRTT| + (1-β) * DevRTT

TimeoutInterval = EstimatedRTT + 4 * DevRTT

After Sample #1:

EstimatedRTT = 0.15 * 146 + (1 - 0.15) * 120 = **123.9ms**

DevRTT = 0.3 * |146 – 123.9| + (1 – 0.3) * 7 = **11.53ms**

TimeoutInterval = 123.9 + 4 * 11.53 = **170.02ms**

After Sample #2:

EstimatedRTT = 0.15 * 110 + (1 - 0.15) * 123.9 = **160.315ms**

DevRTT = 0.3 * |110 –160.315| + (1 – 0.3) * 11.53 = **22.17ms**

TimeoutInterval = 160.315 + 4 * 22.17 = **249.0ms**

After Sample #3:

EstimatedRTT = 0.15 * 135 + (1 - 0.15) * 160.315 = **156.52ms**

DevRTT = 0.3 * |135 – 156.52| + (1 – 0.3) * 22.17 = **21.98ms**

TimeoutInterval = 156.52 + 4 * 21.98 = **244.44ms**

After Sample #4:

EstimatedRTT = 0.15 * 85 + (1 - 0.15) * 156.52 = **145.79ms**

DevRTT = 0.3 * |85 – 145.79| + (1 – 0.3) * 21.98 = **33.62ms**

TimeoutInterval = 145.79 + 4 * 33.62 = **280.27ms**

After Sample #5:

EstimatedRTT = 0.15 * 92 + (1 - 0.15) * 145.79 = **137.72ms**

DevRTT = 0.3 * |92 – 137.72| + (1 – 0.3) * 33.62 = **37.25ms**

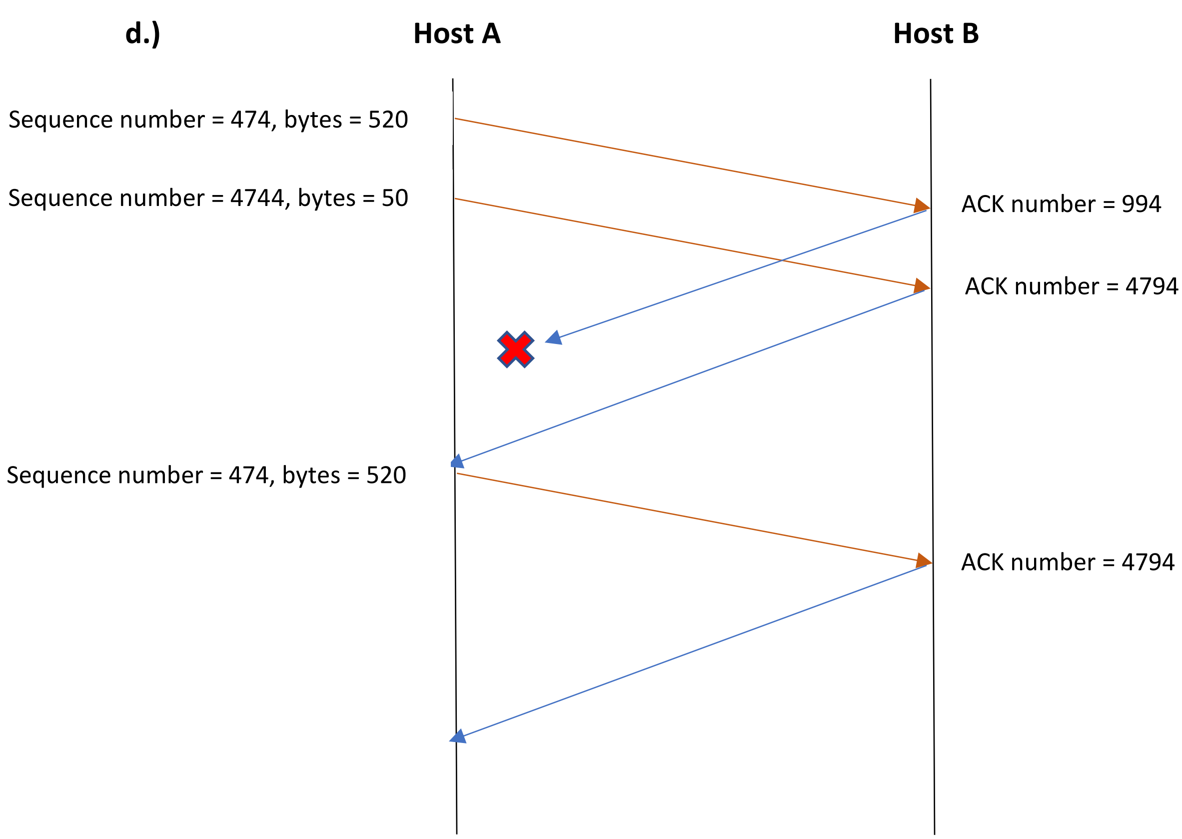
TimeoutInterval = 137.72 + 4 * 37.25 = **286.72ms**

3.) **a.)** Sequence number = 474 + 4270 = **4744**
Source port number = **2350**
Destination port number = **4270**

b.) Sequence number = **474**

c.) Sequence number = **4744**
Source port number = **4270**
Destination port number = **2350**

d.)



4.) To avoid this problem, we will need to determine the range of numbers that can be sent at a given time by both the sender and receiver windows. Let us assume that the lowest sequence number of the packet that the receiver is waiting for is k. This means that the receiver window must be [k, k+w-1]. If the receiver does not receive any of the packets then the ACK values that are sent back will be [x-w, x-1]. Since the sender does not receive any ACK packets, the sender window will be [x-w, x-1]. This gives us the lower edge of the sender's window, x-w, and the next edge of the receiver's window, x+w+1. For these two not to overlap, the window must give at least 2w sequence numbers. This leaves us with the final answer being, **X ≥ 2w.**

5.) In order to find the size of the sender's window we will need to go through the first 18 attempts. We can put the given data, initial threshold = 24kB and MSS = 4kB and receiver window = 40kB, into the table below:

	Transmit Number	**Sender's Window**	**Threshold**	**Receiver's Window**
	0	4	24	40
	1	8	24	40
	2	16	24	40
	3	24	24	40
	4	28	24	40
	5	32	24	40
	6	36	24	40
Triple Dupe ACK	7	40	24	40
	8	20	20	40
	9	24	20	40
	10	28	20	40
	11	32	20	40
Timeout	12	36	20	40
	13	4	18	40
	14	8	18	40
	15	16	18	40
	16	18	18	40
	17	22	18	40

6.) a.)

For GBN:

Host A sends 10 segments, where the 5th segment gets lost. This leaves us with the following segment transmissions: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 5, 6, 7, 8, 9, 10. Since the 5th segment never arrived, the ACK was never sent for it meaning we have to resend that segment along with all segments following it. This leaves us with **16 segments sent.**

Host B sends an ACK after each segment is received. If a segment is not received, then it does not move to the next state. This means that Host B sends the following ACKs: 1, 2, 3, 4, 4, 4, 4, 4, 4, 4, 5, 6, 7, 8, 9, 10 = **15 ACKs sent.**

For SR:

Host A sends 10 segments. where the 5th segment gets lost. Instead of sending that segment and every segment after, we just resend the segment that got lost. This leaves us with the following segment transmissions: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 5. This leaves us with a total of **11 segments sent**.

Host B sends an ACK after each segment is received. Segment 5 is lost so we will send each ACK after and then send an ACK for the resend of segment 5. This will look like: 1, 2, 3, 4, _, 6, 7, 8, 9, 10, 5, which leaves us with a total of **10 ACKs sent.**

For TCP:

Host A sends 10 segments, where the 5th segment gets lost. After all segments have been sent, we must resend the 5th segment since that was the last expected ACK. The segment transmissions would look like: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 5. This leaves us with a total of **11 segments sent.**

For TCP, the ACK messages sent are for the next expected packet, so the transmissions would look like: 1, 2, 3, 4, 4, 4, 4, 4, 4, 10. Meaning we have **10 ACKs sent.**

b.) TCP will have the shortest time interval to transmit all 10 segments successfully due to the speed at which TCP is able to retransmit without having to wait until timeout.