

## Problem 1

	source port numbers	destination port numbers
a) A → S	467	23
b) B → S	513	23
c) S → A	23	467
d) S → B	23	513

e) Yes.

f) No.

## Problem 24

a) True. Suppose the sender has a window size of 3 and sends packets 1, 2, 3 at  $t_0$ . At  $t_1$  ( $t_1 > t_0$ ) the receiver ACKS 1, 2, 3. At  $t_2$  ( $t_2 > t_1$ ) the sender times out and resends 1, 2, 3. At  $t_3$  the receiver receives the duplicates and re-acknowledges 1, 2, 3. At  $t_4$  the sender receives the ACKs that the receiver sent at  $t_1$  and advances its window to 4, 5, 6. At  $t_5$  the sender receives the ACKs 1, 2, 3 the receiver sent at  $t_2$ . These ACKs are outside its window.

b) True. By essentially the same scenario as in (a).

c) True.

a) True. Note that with a window size of 1, SR, GBN, and the alternating bit protocol are functionally equivalent. The window size of 1 precludes the possibility of out-of-order packets (within the window). A cumulative ACK is just an ordinary ACK in this situation, since it can only refer to the single packet within the window.

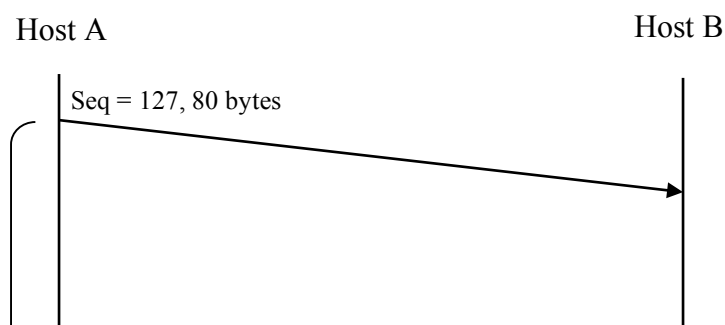
## Problem 27

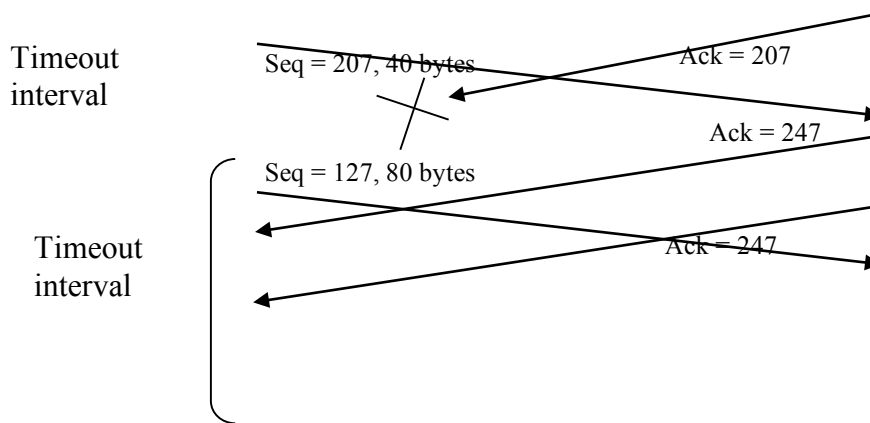
a) In the second segment from Host A to B, the sequence number is 207, source port number is 302 and destination port number is 80.

b) If the first segment arrives before the second, in the acknowledgement of the first arriving segment, the acknowledgement number is 207, 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)





### Problem 33

Let's look at what could go wrong if TCP measures `SampleRTT` for a retransmitted segment. Suppose the source sends packet P1, the timer for P1 expires, and the source then sends P2, a new copy of the same packet. Further suppose the source measures `SampleRTT` for P2 (the retransmitted packet). Finally suppose that shortly after transmitting P2 an acknowledgment for P1 arrives. The source will mistakenly take this acknowledgment as an acknowledgment for P2 and calculate an incorrect value of `SampleRTT`.

Let's look at what could be wrong if TCP measures `SampleRTT` for a retransmitted segment. Suppose the source sends packet P1, the timer for P1 expires, and the source then sends P2, a new copy of the same packet. Further suppose the source measures `SampleRTT` for P2 (the retransmitted packet). Finally suppose that shortly after transmitting P2 an acknowledgment for P1 arrives. The source will mistakenly take this acknowledgment as an acknowledgment for P2 and calculate an incorrect value of `SampleRTT`.

### Problem 36

Suppose packets  $n$ ,  $n+1$ , and  $n+2$  are sent, and that packet  $n$  is received and ACKed. If packets  $n+1$  and  $n+2$  are reordered along the end-to-end-path (i.e., are received in the order  $n+2$ ,  $n+1$ ) then the receipt of packet  $n+2$  will generate a duplicate ack for  $n$  and would trigger a retransmission under a policy of waiting only for second duplicate ACK for retransmission. By waiting for a triple duplicate ACK, it must be the case that *two* packets after packet  $n$  are correctly received, while  $n+1$  was not received. The designers of the triple duplicate ACK scheme probably felt that waiting for two packets (rather than 1) was the right tradeoff between triggering a quick retransmission when needed, but not retransmitting prematurely in the face of packet reordering.

### Problem 40

- a) TCP slowstart is operating in the intervals [1,6] and [23,26]
- b) TCP congestion avoidance is operating in the intervals [6,16] and [17,22]
- c) After the 16<sup>th</sup> transmission round, packet loss is recognized by a triple duplicate ACK. If there was a timeout, the congestion window size would have dropped to 1.
- d) After the 22<sup>nd</sup> transmission round, segment loss is detected due to timeout, and hence the congestion window size is set to 1.
- e) The threshold is initially 32, since it is at this window size that slow start stops and congestion avoidance begins.
- f) The threshold is set to half the value of the congestion window when packet loss is detected. When loss is detected during transmission round 16, the congestion window size is 42. Hence the threshold is 21 during the 18<sup>th</sup> transmission round.
- g) The threshold is set to half the value of the congestion window when packet loss is detected. When loss is detected during transmission round 22, the congestion window size is 26. Hence the threshold is 13 during the 24<sup>th</sup> transmission round.
- h) During the 1<sup>st</sup> transmission round, packet 1 is sent; packet 2-3 are sent in the 2<sup>nd</sup> transmission round; packets 4-7 are sent in the 3<sup>rd</sup> transmission round; packets 8-15 are sent in the 4<sup>th</sup> transmission round; packets 16-31 are sent in the 5<sup>th</sup> transmission round; packets 32-63 are sent in the 6<sup>th</sup> transmission round; packets 64 – 96 are sent in the 7<sup>th</sup> transmission round. Thus packet 70 is sent in the 7<sup>th</sup> transmission round.
- i) The threshold will be set to half the current value of the congestion window (8) when the loss occurred and congestion window will be set to the new threshold value + 3 MSS . Thus the new values of the threshold and window will be 4 and 7 respectively. Threshold is 21, and congestion window size is 1.
- j) round 17, 1 packet; round 18, 2 packets; round 19, 4 packets; round 20, 8 packets; round 21, 16 packets; round 22, 21 packets. So, the total number is 52.