John Carroll

COMP 4320 Computer Networks - Homework 3

1. Question
   1. We saw that the sender needs sequence numbers so that the receiver can tell if a data packet is a duplicate of an already received data packet. In the case of ACKs, the sender does not need this info (i.e., a sequence number on an ACK) to detect a duplicate ACK. A duplicate ACK is obvious to the rdt3.0 receiver, since when it has received the original ACK it transitioned to the next state. The duplicate ACK is not the ACK that the sender needs and hence is ignored by the rdt3.0 sender.
2. Question
   1. Given SRTT values are 146ms, 110ms, 135ms, 85ms, 92ms α=0.15 then ERTT=120ms. We have to compute ERTT.  
      Assuming we have received n RTT, (with 1 being the most recent, 2 being the second most recent, etc.)  
        
      EstimatedRTT(n) = (1-α)\*EstimatedRTT(n-1) + α\*SampleRTT(1)

(EstimatedRTT(n): EstimatedRTT after receiving ACK of nth packet.)  
  
ERTT1 = SRTT3  
ERTT2 = (1-α) \* ERTT1 + α\*SRTT2(1-α)\*SRTT3 + α\*SRTT2  
ERTT3 = (1-α) \* ERTT2 + α\*SRTT1(1-α)\*((1-α)\*SRTT3 + α\*SRTT2) + α\*SRTT1  
By applying this formula,

EstimatedRTT(current)=(1-α)\*EstimatedRTT(previous) + α \* SampleRTT(recent)  
We get the following,

120+(1-0.15)(146)=244

ERTT =192.25  
Computing DEVRTT,

DevRTT=(1-β)\*DevRTT+β\*|SampleRTT-EstimatedRTT|(typically, β=0.3)  
Applying this formula, we then get,

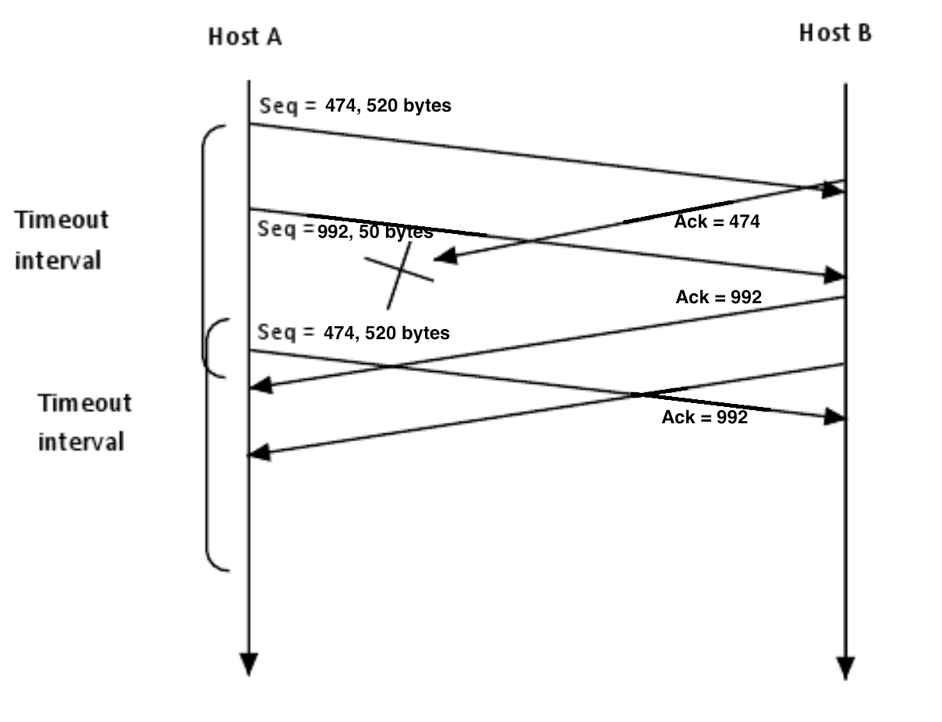
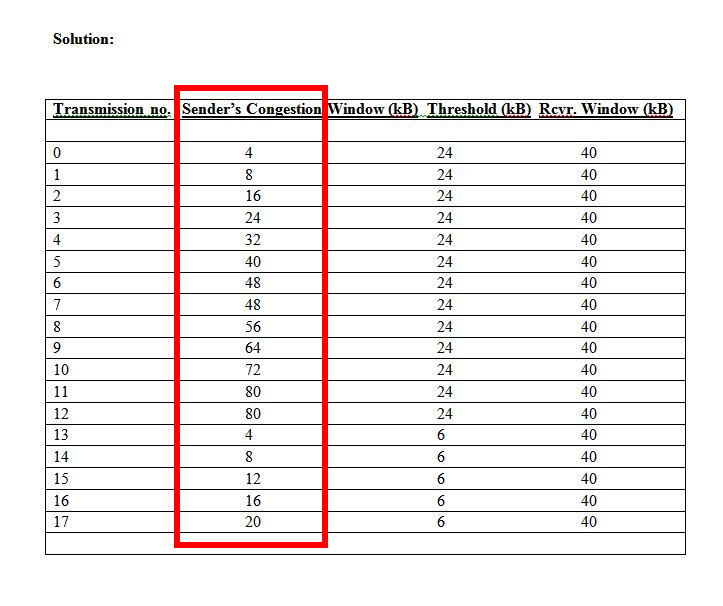
=0.7\*7+0.7\*|146-120|=23.1ms  
Similarly, we apply the formula for the rest of the four values(SRTT\_ERTT)

Answers: 23.1, 11.9, 17.5, 29.4, 54.5  
The TimedOutInterval has,

TimeOutInterval = EstimatedRTT + 4\*DevRTT,

TimeOutInterval = 192.5 + 4\*(136.4)

TimeOutInterval = 738.1ms

1. Question
   1. Parts
      * + 1. In the second segment from Host A to B, the sequence number is 994, source port number is 2350 and destination port number is 4270.
          2. If the first segment arrives before the second, in the acknowledgement of the first arriving segment, the acknowledgement number is 994, the source port number is 4270 and the destination port number is 2350.
          3. If the second segment arrives before the first segment, in the acknowledgement of the first arriving segment, the acknowledgement number is 474, indicating that it is still waiting for bytes 474 and onwards.
          4. [Image]
2. Question
   1. In order to avoid the scenario of Figure 3.27, we want to avoid having the leading edge of the receiver's window (i.e., the one with the “highest” sequence number) wrap around in the sequence number space and overlap with the trailing edge (the one with the "lowest" sequence number in the sender's window). That is, the sequence number space must be large enough to fit the entire receiver window and the entire sender window without this overlap condition. So we need to determine how large a range of sequence numbers can be covered at any given time by the receiver and sender windows.   
      Suppose that the lowest-sequence number that the receiver is waiting for is packet *m*. In this case, its window is *[m, m+w-1]* and it has received (and ACKed) packet *m-1* and the *w-1* packets before that, where *w* is the size of the window. If the sender has yet received none of those *w* ACKs, then ACK messages with values of *[m-w, m-1]* may still be propagating back. If the sender has received no ACKs with these ACK numbers, then the sender's window would be *[m-w, m-1].*   
      Thus, the lower edge of the sender's window is *m-w*, and the leading edge of the receiver’s window is *m+w-1*. In order for the leading edge of the receiver's window to not overlap with the trailing edge of the sender's window, the sequence number space must thus be big enough to accommodate *2w* sequence numbers. That is, the sequence number space must be at least twice as large as the window size, *k ≥ 2w*.
3. Question
   1. [Image]
4. Question
   1. Parts
      * + 1. ***GBN***: A sends 16 segments in total. They are initially sent segments 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 and later re-sent segments 5, 6, 7, 8, 9, and 10. B sends 15 ACKs. They are 9 ACKs with sequence number 1, 2, 3, 4 and 6 ACKs with sequence numbers 5, 6, 7, 8, 9, 10.   
             ***SR***: A sends 11 segments in total. They are initially sent segments 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 and later re-sent segments 5. B sends 10 ACKs. They are 4 ACKs with sequence number 1, 2, 3, 4, 6, 7, 8, 9, 10 and there is one ACK with sequence number 5.   
             ***TCP***: A sends 11 segments in total. They are initially sent segments 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 and later re-sent segments 5. B sends 10 ACKs. They are 4 ACKs with sequence number 5 and there is one ACK with sequence numbers 11.
          2. It is TCP. Because the TCP uses fast retransmit and there is no waiting time.