Solutions to Chapter 3 Problems

Q1. True or false

a) The size of the TCP rwnd never changes throughout the duration of the connection.
b) Suppose Host A is sending Host B a large file over a TCP connection. The number of unacknowledged bytes that A sends cannot exceed the size of the receiver buffer.
c) Suppose Host A is sending Host B a large file to Host B over a TCP connection. If the sequence number for a segment of this connection is *m,* then the sequence number for the subsequent segment will necessarily be *m*+1.
d) Consider congestion control in TCP. When the timer expires at the sender, the value of ssthresh is set to one half of its previous value.

(This question is taken from Kurose & Ross's book, Chapter 3 Review Problems 14(b,c,d), and 18

Solution:

a) F
b) T
c) F
d) F

Q2.

UDP and TCP use 1s complement for their checksum. Suppose you have the following three 8-bit bytes: 01010011, 01100110, 01110100. What is the 1s complement of the sum of these 8-bit bytes? (Note that although UDP and TCP use 16-bit words in computing the checksum, for this problem you are being asked to consider 8-bit sum.) Show all work. Why is it that UDP takes the 1s complement of the sum: that is, why not just use the sum? With the 1s complement scheme, how does the receiver detect errors? Is it possible that a 1-bit error will go undetected? How about a 2-bit error?

(This question is taken from Kurose & Ross's book, Chapter 3, Problem 3)

Solution:

Note, wrap around if overflow.

$$\begin{array}{rcccccccc} & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 1 \\ + & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 \\ \hline & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 1 \end{array}$$

$$\begin{array}{rcccccccc} & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 1 \\ + & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ \hline & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 \end{array}$$

One's complement = 1 1 0 1 0 0 0 1.

To detect errors, the receiver adds the four words (the three original words and the checksum). If the sum contains a zero, the receiver knows there has been an error. All one-bit errors will be detected, but two-bit errors can be undetected (e.g., if the last digit of the first word is converted to a 0 and the last digit of the second word is converted to a 1).

Q3.
Consider the rdt2.2 receiver in Fig. P3 and the creation of a new packet in the self-transition (i.e. the transition from the state back to itself) in the Wait-for-0-from below and the Wait-for-1-from-below states: sndpkt=make_pkt(ACK, 0, chksum) and sndpkt=make_pkt(ACK, 1, chksum). Would the protocol work correctly if this action were removed from the self-transition in the Wait-for-1-from-below state? Justify your answer. What if this event were removed from the self-transition in the Wait-for-0-from-below state? (hint: In this later case, consider what would happen if the first sender-to-receiver packet were corrupted)

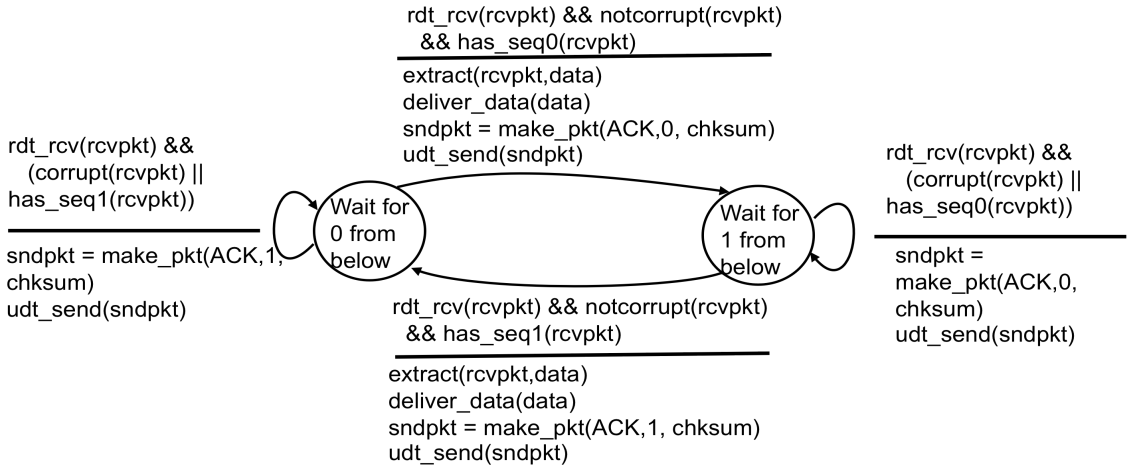

Fig. P3

(This question is taken from Kurose & Ross's book, Chapter 3 Problem 11)

Solution:
If the sending of this message were removed, the sending and receiving sides would deadlock, waiting for an event that would never occur. Here's a scenario:

- Sender sends pkt0, enter the "Wait for ACK0 state", and waits for a packet back from the receiver

- Receiver is in the "Wait for 0 from below" state, and receives a corrupted packet from the sender. Suppose it does not send anything back, and simply re-enters the 'wait for 0 from below" state.

Now, the ender is awaiting an ACK of some sort from the receiver, and the receiver is waiting for a data packet form the sender – a deadlock!

Q4.
Suppose the five measured SampleRTT values are 106ms, 120ms, 140ms, 90ms, and 115ms. Compute the EstimatedRTT after each of these SampleRTT values is obtained, using a value of α=0.125 and assuming that the value of EstimatedRTT was 100ms just before the first of these five samples were obtained. Compute also the DevRTT after each sample is obtained, assuming a value of β=0.25 and assuming the value of DevRTT was 5 ms just before the first of these five samples was obtained. Last compute the TCP TimeoutInterval after each of these samples is obtained.

(This question is taken from Kurose & Ross's book, Chapter 3 Problem 31)

Solution:

$$EstimatedRTT = xSampleRTT + (1-x)EstimatedRTT$$
$$DevRTT = y|SampleRTT - EstimatedRTT| + (1-y)DevRTT$$
$$TimeoutInterval = EstimatedRTT + 4 * DevRTT$$

After obtaining first sampleRTT is

$$EstimatedRTT = 0.125 * 106 + 0.875 * 100$$
$$= 100.75ms.$$
$$DevRTT = 0.25 * |106 - 100.75| + 0.75 * 5$$
$$= 5.06ms.$$
$$TimeoutInterval = 100.75 + 4 * 5.06$$
$$= 120.99ms.$$

After obtaining second sampleRTT = 120ms:

$$EstimatedRTT = 0.125 * 120 + 0.875 * 100.75$$
$$= 103.15ms.$$
$$DevRTT = 0.25 * |120 - 103.15| + 0.75 * 5.06$$
$$= 8ms.$$
$$TimeoutInterval = 103.15 + 4 * 8$$
$$= 135.15ms.$$

After obtaining Third sampleRTT = 140ms:

$$EstimatedRTT = 0.125 * 140 + 0.875 * 103.15$$
$$= 107.76ms.$$

$$DevRTT = 0.25 * |140 - 107.76| + 0.75 * 8$$
$$= 14.06ms .$$
$$TimeoutInterval = 107.76 + 4 * 14.06$$
$$= 164ms .$$

After obtaining fourth sampleRTT = 90ms:

$$EstimatedRTT = 0.125 * 90 + 0.875 * 107.76$$
$$= 105.54ms .$$
$$DevRTT = 0.25 * |90 - 105.54| + 0.75 * 14.06$$
$$= 14.42ms .$$
$$TimeoutInterval = 105.54 + 4 * 14.42$$
$$= 163.22ms .$$

After obtaining fifth sampleRTT = 115ms:

$$EstimatedRTT = 0.125 * 115 + 0.875 * 105.54$$
$$= 106.71ms .$$
$$DevRTT = 0.25 * |115 - 106.71| + 0.75 * 14.42$$
$$= 12.88ms .$$
$$TimeoutInterval = 106.71 + 4 * 12.88$$
$$= 158.23ms .$$

Q5.

Consider Fig. P5. Assuming TCP Reno is the protocol experiencing the behavior shown in the figure, answer the following questions.

a) Identify the interval of time when TCP slow start is operating.
b) Identify the interval of time when TCP congestion avoidance is operation.
c) After the 16th transmission round, is segment loss detected by a triple duplicate ACK or by a timeout?
d) After the 22nd transmission round, is segment loss detected by a triple duplicate ACK or by a timeout?
e) What is the initial value of ssthresh at the first transmission round?
f) What is the value of ssthresh at the 18th transmission round?
g) What is the value of ssthresh at the 24th transmission round?
h) During what transmission round is the 70th segment sent?
i) Assuming a packet loss is detected after the 26th round by the receipt of a triple duplicate ACK, what will be the value of the congestion window size and of ssthresh?
j) Suppose TCP Tahoe is used (instead of TCP Reno), and assume that triple duplicate ACKs are received at the 16th round. What are the ssthresh and the congestion window size at the 19th round?
k) Again suppose TCP Tahoe is used, and there is a timeout event at 22nd round. How many packets have been sent out from 17th round till 22nd round, inclusive?

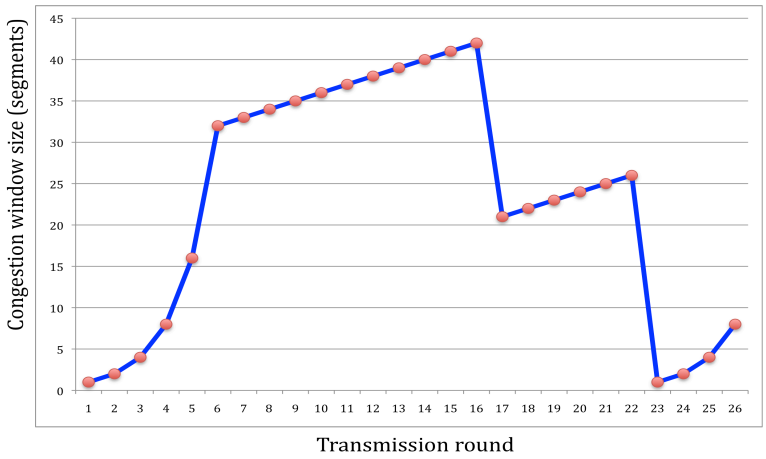

Fig. P5

(This question is taken from Kurose & Ross's book, Chapter 3 Problem 40)

Solution:

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 16th 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 22nd 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 windows size is 42. Hence the threshold is 21 during the 18th 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 windows size is 29. Hence the threshold is 14 (taking lower floor of 14.5) during the 24th transmission round.
h) During the 1st transmission round, packet 1 is sent; packet 2-3 are sent in the 2nd transmission round; packets 4-7 are sent in the 3rd transmission round; packets 8-15 are sent in the 4th transmission round; packets 16-31 are sent in the 5th transmission round; packets 32-63 are sent in the 6th transmission round; packets 64 – 96 are sent in the 7th transmission round. Thus packet 70 is sent in the 7th 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.
j) threshold is 21, and congestion window size is 1.

k) 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.