CSE3214 Computer Network Protocols and Applications

Chapter 3
Examples and Homework Problems

Example 1 (R11)

Suppose that the roundtrip delay between sender and receiver is constant and known to the sender. Would a timer still be necessary in protocol rdt 3.0? Explain.

Solution to Example 1

- A timer would still be necessary in the protocol rdt 3.0. If the round trip time is known then the only advantage will be that, the sender knows for sure that either the packet or the ACK (or NACK) for the packet has been lost, as compared to the real scenario, where the ACK (or NACK) might still be on the way to the sender, after the timer expires. However, to detect the loss, for each packet, a timer of constant duration will still be necessary at the sender.

Example 2 (P15)

Support Host A sends two TCP segments back to back to Host B over a TCP connection. The first segment has sequence number 90; the second has sequence number 110.

a. How much data is in the first segment?

b. Suppose that the first segment is lost but the second segment arrives at B. In the acknowledgment that Host B send to Host A, what will be the acknowledgment number?

Solution to Example 2

(a) 20 bytes
(b) Acknowledgment number = 90

Example 3 (R27)

Host A and B are communicating over a TCP connection, and Host B has already received from A all bytes up through byte 126. Suppose Host A then sends two segments to Host B back-to-back. The 1st and 2nd segments contain 80 and 40 bytes of data, respectively. In the 1st segment, the seq # is 127, the source port is 302, and the destination port number is 80. Host B sends an acknowledgment whenever it receives a segment from Host A.

a) In the 2nd segment sent from A to B, what are the seq #, source port number, and destination port number?

b) If the 1st segment arrives before the second segment, in the acknowledgment of the first arriving segment, what is the acknowledgment number, the source port number, and the destination port number?

c) If the 2nd segment arrives before the first segment, in the acknowledgment of the 1st arriving segment, what is the acknowledgment number?
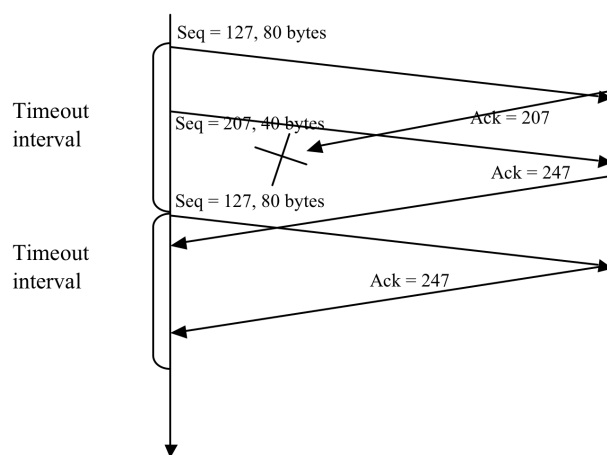
Example 3 (R27)

Host A and B are communicating over a TCP connection, and Host B has already received from A all bytes up through byte 126. Suppose Host A then sends two segments to Host B back-to-back. The 1st and 2nd segments contain 80 and 40 bytes of data, respectively. In the 1st segment, the seq # is 127, the source port is 302, and the destination port number is 80. Host B sends an acknowledgment whenever it receives a segment from Host A.

d) Suppose the two segments sent by A arrive in order at B. The 1st acknowledgment is lost and the 2nd acknowledgment arrives after the first time-out interval. Draw a timing diagram, showing these segments and all other segments and acknowledgment sent. (Assume there is no additional packet loss.) For each segment in your figure, provide the sequence number and the number of bytes of data; for each acknowledgment that you add, provide the acknowledgment number.

Solution to Example 3

- 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.
- 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.
- 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.

Solution to Example 3 (con't)

Seq = 127, 80 bytes

Timeout interval

Seq = 207, 40 bytes

Ack = 207

Ack = 247

Seq = 127, 80 bytes

Timeout interval

Ack = 247

Homework Problems

Solutions will be available on course website:

http://wiki.cse.yorku.ca

Problem 1: True or False? (R14,18)

1. The size of the TCP rwnd never changes throughout the duration of the connection.
2. 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.
3. 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 $m +1$.
4. 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.

Problem 2 (P3)

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?

Problem 3 (P11)

Consider the rdt2.2 receiver in Fig. P3(next page) 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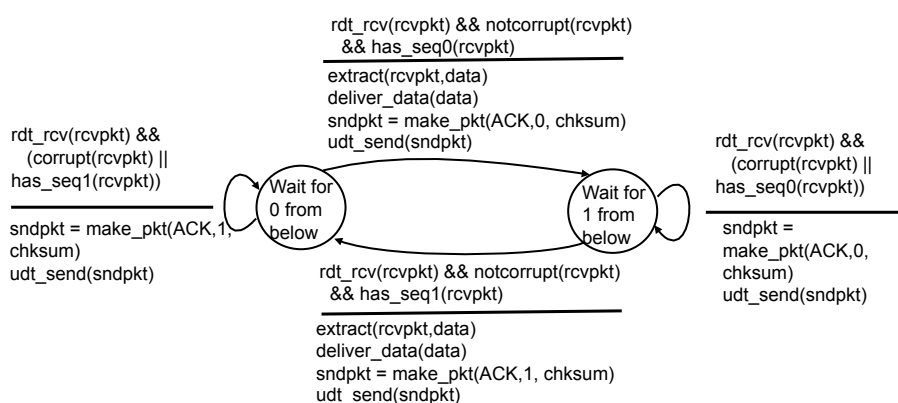 below and the Wait-for-1-from-below states: sndpkt=make_pkt(ACK, 0, chksum) and sndpkt=make_pkt(ACK, 1, chksum). Would the protocol work correctly if this action were removed from the self-transition in the Wait-for-1-from-below state? Justify your answer. What if this event were removed from the self-transition 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)

Problem 3, Fig. P3

rdt_rcv(rcvpkt) && notcorrupt(rcvpkt)
&& has_seq0(rcvpkt)
―――――
extract(rcvpkt,data)
deliver_data(data)
sndpkt = make_pkt(ACK,0, chksum)
udt_send(sndpkt)

rdt_rcv(rcvpkt) &&
(corrupt(rcvpkt) ||
has_seq1(rcvpkt))
―――――
sndpkt = make_pkt(ACK,1, chksum)
udt_send(sndpkt)

rdt_rcv(rcvpkt) &&
(corrupt(rcvpkt) ||
has_seq0(rcvpkt))
―――――
sndpkt = make_pkt(ACK,0, chksum)
udt_send(sndpkt)

Wait for 0 from below

Wait for 1 from below

rdt_rcv(rcvpkt) && notcorrupt(rcvpkt)
&& has_seq1(rcvpkt)
―――――
extract(rcvpkt,data)
deliver_data(data)
sndpkt = make_pkt(ACK,1, chksum)
udt_send(sndpkt)

Problem 4 (P31)

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.

Problem 5 (P40)

Consider Fig. P5(at the end of question). 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?

Problem 5 (con't)

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

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?

Problem 5 Fig. P5