ECE 465: Computer Network Protocols and Applications
Homework 8
Solution
Dr. Paris

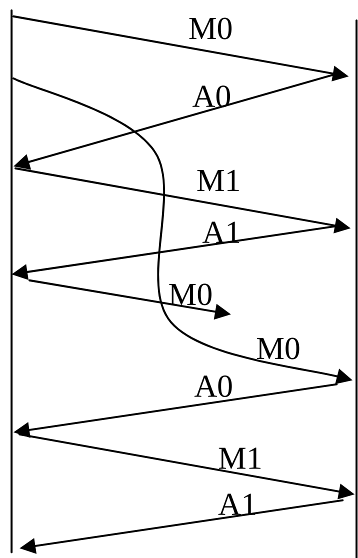
Chapter 3 Problems

Problem 9.

The protocol would still work, since a retransmission would be what would happen if the packet received with errors has actually been lost (and from the receiver standpoint, it never knows which of these events, if either, will occur).

To get at the more subtle issue behind this question, one has to allow for premature timeouts to occur (which is not what the question asked, sorry). In this case, if each extra copy of the packet is ACKed and each received extra ACK causes another extra copy of the current packet to be sent, the number of times packet n is sent will increase without bound as n approaches infinity.

Problem 10.



old version of M0 accepted!

Problem 12.

It takes 8 microseconds (or 0.008 milliseconds) to send a packet. in order for the sender to be busy 90 percent of the time, we must have

$$util = 0.9 = (0.008n)/30.016$$

or n approximately 3377 packets.

Problem 16.

a) Here we have a window size of N=3. Suppose the receiver has received packet k-1, and has ACKed that and all other preceeding packets. If all of these ACK's have been received by sender, then sender's window is [k, k+N-1]. Suppose next that none of the ACKs have been received at the sender. In this second case, the sender's window contains k-1 and the N packets up to and including k-1. The sender's window is thus [k-N,k-1]. By these arguments, the senders window is of size 3 and begins somewhere in the range [k-N,k].

b) If the receiver is waiting for packet k, then it has received (and ACKed) packet k-1 and the N-1 packets before that. If none of those N ACKs have been yet received by the sender, then ACK messages with values of [k-N,k-1] may still be propagating back. Because the sender has sent packets [k-N, k-1], it must be the case that the sender has already received an ACK for k-N-1. Once the receiver has sent an ACK for k-N-1 it will never send an ACK that is less that k-N-1. Thus the range of in-flight ACK values can range from k-N-1 to k-1.

Problem 19.

a) True. Suppose the sender has a window size of 3 and sends packets 1, 2, 3 at $t0$. At $t1$ $(t1 > t0)$ the receiver ACKS 1, 2, 3. At $t2$ $(t2 > t1)$ the sender times out and resends 1, 2, 3. At $t3$ the receiver receives the duplicates and re-acknowledges 1, 2, 3. At $t4$ the sender receives the ACKs that the receiver sent at $t1$ and advances its window to 4, 5, 6. At $t5$ the sender receives the ACKs 1, 2, 3 the receiver sent at $t2$. These ACKs are outside its window.

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

c) True.

d) 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 21.

Denote $EstimatedRTT^{(n)}$ for the estimate after the *n*th sample.

$$EstimatedRTT^{(1)} = SampleRTT_1$$

$$EstimatedRTT^{(2)} = xSampleRTT_1 + (1-x)SampleRTT_2$$

$$EstimatedRTT^{(3)} = xSampleRTT_1$$

$$+ (1-x)[xSampleRTT_2 + (1-x)SampleRTT_3]$$
$$= xSampleRTT_1 + (1-x)xSampleRTT_2$$
$$+ (1-x)^2 SampleRTT_3$$

$$EstimatedRTT^{(4)} = xSampleRTT_1 + (1-x)EstimatedRTT^{(3)}$$
$$= xSampleRTT_1 + (1-x)xSampleRTT_2$$
$$+ (1-x)^2 xSampleRTT_3 + (1-x)^3 SampleRTT_4$$

b)

$$EstimatedRTT^{(n)} = x\sum_{j=1}^{n-1}(1-x)^j SampleRTT_j$$
$$+ (1-x)^n SampleRTT_n$$

c)

$$EstimatedRTT^{(\infty)} = \frac{x}{1-x}\sum_{j=1}^{\infty}(1-x)^j SampleRTT_j$$
$$= \frac{1}{9}\sum_{j=1}^{\infty}.9^j SampleRTT_j$$

The weight given to past samples decays exponentially.

Problem 27.

a) TCP slowstart is operating in the intervals [1,6] and [23,26]
b) TCP congestion advoidance 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 slowtart 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 16, the congestion windows size is 42. Hence the threshold is 21 during the 18th 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; packets15-31 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 congestion window and threshold will be set to half the current value of the congestion window (8) when the loss occurred. Thus the new values of the threshold and window will be 4.

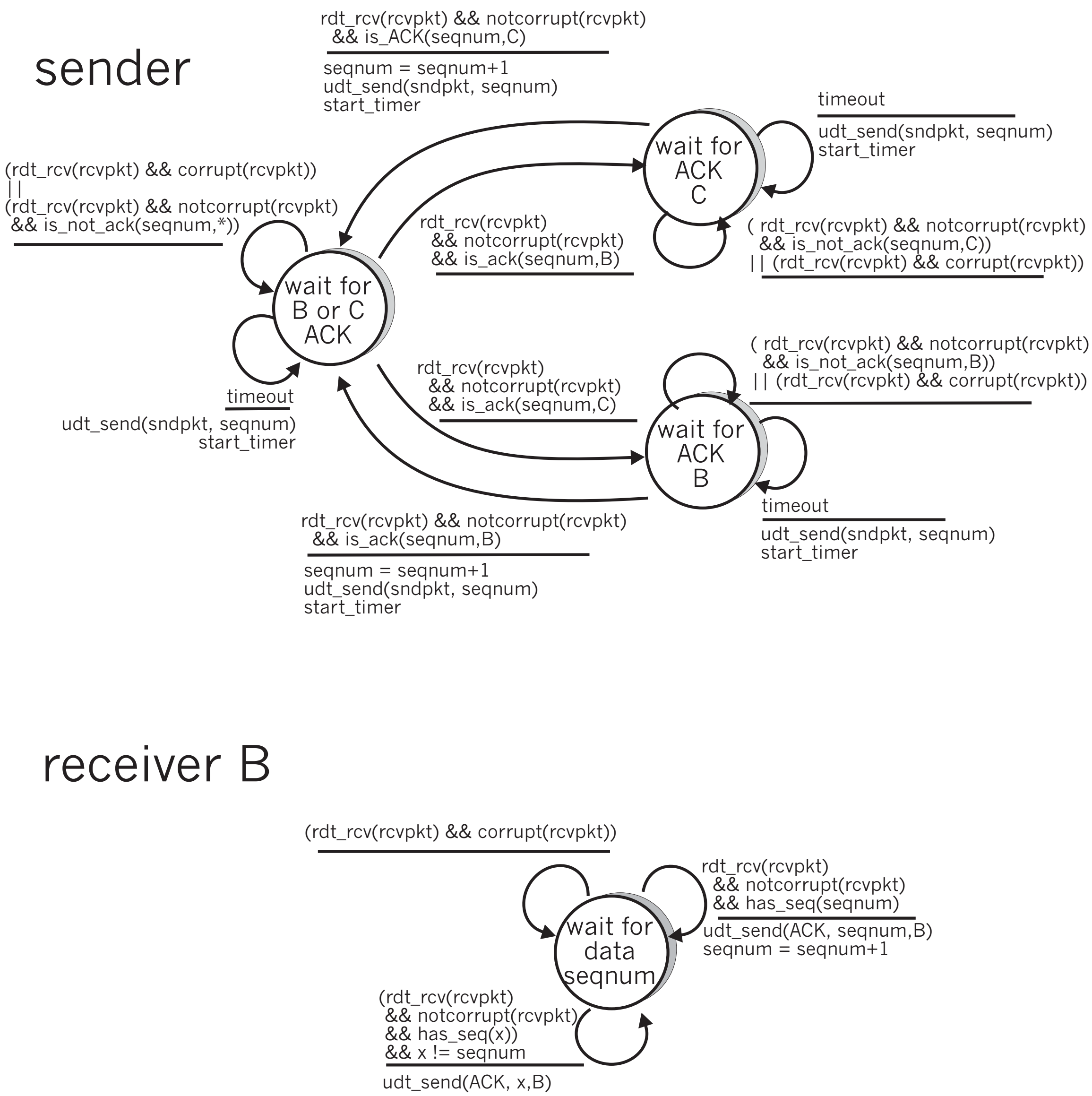


Figure 3. Sender and receiver for Problem 3.12