University of California, Riverside
CS 164 Midterm Test –Fall 1995
50 Minutes, Closed Book.

1. [5 × 3 points = 15 points total] Define the following terms:
   a) store-and-forward delay
   b) cut-through switching
   c) cumulative acknowledgement
   d) idle token
   e) roll-call polling

2. [15 points total] Measurements of data traffic on a computer network frequently show a *bimodal* packet size distribution. That is, most of the packets are either very short or very long—almost none have some intermediate length.
a) [4 points] State *two reasons* (based on application requirements and on the way protocols are designed, respectively) why we might expect to see bimodal packet sizes on a network.
b) [4 points] One particular measurement study found that 80% of the packets on a network were short, but 80% of the data bytes are carried in long packets. Find the length in bytes for the long packets, assuming the short packets are 100 bytes long. (You may assume that all framing overhead, such as headers and checksums, have been included in the count of data bytes.)
c) [2 points] What is the average packet length?
d) [5 points] What is the *mean residual life* of the length of a packet?

3. [2+2+2+4 points = 10 points total] Consider the stop-and-wait data link layer ARQ algorithm operating over a noisy radio channel. For *each* of the following suggested changes to the system *while keeping all other system parameters the same*:
   a) increasing the data rate
   b) increasing the error rate
   c) increasing the number of bits in the sequence number field in the header
   d) increasing the frame length (think carefully about this one!)

State whether it will result in an increase or decrease in the channel efficiency (i.e., the proportion of time it is being used to deliver valid frames). Justify your answers.

4. [2+3+5 points =10 points total] Alice and Bob are exchanging data link frames using the **selective repeat** ARQ. Assume that Alice's transmit window also Bob's receive window are both set to **8 frames**, and that **individual acknowledgments** are used. You may assume that the transmission time for a separate ACK is negligible in comparison to a normal data frame, and that the round-trip propagation delay is 4 times large than the transmission time for a frame.
a) What is the minimum number of bits that can be used for the sequence numbers?
b) Trace the sequence of frame numbers and ACKs sent if a bit error damages frame 2.
c) If acknowledgements are piggybacked on reverse traffic, what is the maximum time that Bob should wait for a reverse frame to arrive before sending a separate ACK.

University of California, Riverside
CS 164 Midterm Test –Fall 1995

*****SOLUTIONS*****

1. [5 × 3 points = 15 points total] Define the following terms:

a) store-and-forward delay

In a packet switched or message switched networks, this refers to the extra time spent sitting at each intermediate node (or switch, router, IMP, etc) across the network because the entire frame (up to the very last bit) must have arrived at the node before it begins transmitting the first bit of the frame along the next hop. Thus, the delay is proportional to the packet size.

b) cut-through switching

This is a technique for reducing the effects of store-and-forward delay. The node determines the next hop for the frame as soon as the destination field in the header has arrived. If that next hop is currently idle, transmission of the incoming frame on its next hop begins immediately, without waiting for the entire frame to have arrived. This reduces the delivery time on a quiet network, which is good. However, since transmission on the next hop begins before the node has seen the whole frame (and hence checked its CRC), the node will retransmit damaged frames as well as valid ones.

c) cumulative acknowledgement

A message from the receiver back to the sender saying that all items consecutively from the beginning to the given item have been recieved. Note the distinction between all items that have been sent, versus all items whose sequence numbers are less than or equal to the given value, since the sender may have send items out of order.

d) idle token

A special bit pattern (or control symbol) used in token ring networks to indicate the place where an attached host can add new packets to the end of the train of packets circulating around the ring. You could also describe it as a capability such that the current holder of the idle token is in control of the network, where you hold it be converting the symbol to a busy token and release it be generating a new one.

e) roll-call polling

A technique used in master-slave systems where the master sends a request to send data to each of the slaves, in turn. Each slave responds either by sending some data, if it has any, followed by an end-of-response message, or by sending a poll-reject message if it has no data. The response times for such systems tend to be very slow because it takes so long to go all the way around the list in search of the slaves that actually have some data.

2. [15 points total] Measurements of data traffic on a computer network frequently show a *bimodal* packet size distribution. That is, most of the packets are either very short or very long—almost none have some intermediate length.

a) [4 points] State *two reasons* (based on application requirements and on the way protocols are designed, respectively) why we might expect to see bimodal packet sizes on a network.

i) Application requirements: There are some applications, like interactive terminal traffic, which naturally use short packets. Conversely other applications, like file transfer, involve bulk data transfer and work best with large packets.
ii) Since networks are not completely reliable, acknowledegments are needed. If Alice is sending a big file to Bob, then Alice will use large data packets, but Bob will use short acknowledgment packets.
b) [4 points] One particular measurement study found that 80% of the packets on a network were short, but 80% of the data bytes are carried in long packets. Find the length in bytes for the long packets, assuming the short packets are 100 bytes long. (You may assume that all framing overhead, such as headers and checksums, have been included in the count of data bytes.)
Let L be the length of a long packet. Also, for convenience let N be the total number of packets seen during the measurement study. (We don't need it, and it will cancel out of our formulas, but it makes the explanation easier.) From the first statement, we have that the total number of short and long packets must be 0.8N and 0.2N, respectively. Now using the second statement, we have

$$0.2N*L = 0.8*(0.8N*100 + 0.2N*L),$$

or

$$L=4*80 +0.8L$$

or

$$L=320/0.2=1,600 \text{ bytes}$$

c) [2 points] What is the average packet length?

0.8*100+0.2*1,600=400 bytes

d) [5 points] What is the *mean residual life* of the length of a packet?

The residual life formula is E[X**2]/(2*E[X]). Therefore, the answer is (0.8*100**2+0.2*1600**2)/800 = 650 bytes

3. [2+2+2+4 points = 10 points total] Consider the stop-and-wait data link layer ARQ algorithm operating over a noisy radio channel. For *each* of the following suggested changes to the system *while keeping all other system parameters the same*:

a) increasing the data rate
b) increasing the error rate
c) increasing the number of bits in the sequence number field in the header
d) increasing the frame length (think carefully about this one!)

State whether it will result in an increase or decrease in the maximum channel efficiency (i.e., the proportion of time it is being used to deliver valid frames). Justify your answers.

The efficiency of stop-and-wait is determined by looking at what happens over a "cycle" where delay is introduced because of the time required for 1) Alice to transmit a frame to Bob, 2) the last bit of the frame reach Bob, 3) Bob to transmit an ACK, and 4) the last bit of the ACK to reach Alice.

a) If we increase the data rate, then the propagation times stay the same but the time spent transmitting the same frame and ACK at a higher data rate go down, so the maximum efficiency goes does.

b) If we increase the error rate, then the length of each cycle stays the same, but fewer of those cycles contain valid frames, so the maximum efficiency goes down.

c) If we increase the number of bits in the ACK field, then we could support a larger transmit and receive window. However, stop-and-wait uses a window size of one, by definition, so nothing will change. (Yes, there is an infintessimal loss in efficiency because the transmission times are a few bit times longer, but this isn't worth getting excited about.)

d) If we increase the frame length, then at first the efficiency goes up because the transmit portion of a cycle increases while the propagation time stays the same. However, as you keep increasing the frame length, you are also increasing the probability that a frame contains at least one bit error, since the error rates are per bit and you have more bits in a loner frame. Eventually, the additional efficiency from increasing the frame size in comparison to the propagation delay gets dominated by the loss in efficiency due to a higher frame error rate, and the maximum efficiency starts going down. There was a long discussion of this in my lecture slides.

4. [2+3+5 points =10 points total] Alice and Bob are exchanging data link frames using the **selective repeat** ARQ. Assume that Alice's transmit window also Bob's receive window are both set to **8 frames**, and that **individual acknowledgments** are used. You may assume that the transmission time for a separate ACK is negligible in comparison to a normal data frame, and that the round-trip propagation delay is 4 times larger than the transmission time for a frame.
a) What is the minimum number of bits that can be used for the sequence numbers?

The results from Knuth's paper were that the minimum sequence number modulus was the sum of the transmit window size and receive window size, which is 16 in this case. Thus, we need at least 4 bit sequence numbers.

b) Trace the sequence of frame numbers and ACKs sent if a bit error damages frame 2.

If Alice begins transmitting frame 1 at time 0 (measured in units of a frame transmit time), then she will finish frame 1 (and begin frame 2) at time 1, finish frame 2 (and begin frame 3) at time 2, and so on. Meanwhile, since the round-trip propagation delay is 4, Bob will finish receiving frame 1 at time 3, finish receiving frame 2 at time 5, etc, and hence that Alice should receive Bob's ACK for frame 1 right after time 5, his ACK for frame 2 right after time 6, and so on.

Thus, if (only) frame 2 is damaged by a bit error, then we will see the following sequence of frame numbers and ACKs sent by Alice and Bob:

Time:	0	1	2	3	4	5	6	7	8	9	10
Alice:	1	2	3	4	5	6	2	7	8	. . .	
Bob:	-	-	-	1	-	3	4	5	6	2	7

c) If acknowledgements are piggybacked on reverse traffic, what is the maximum time that Bob should wait for a reverse frame to arrive before sending a separate ACK.

The goal here is to make sure that Alice doesn't get left with nothing new in her transmit window, causing her transmitter to either stall (i.e., sit there sending nothing) or send an unnecessary duplicate. Since Alice has a transmit window size of 8 frames, she must get the ACK info for the given frame N before it is time to start transmitting frame $N+8$. Since it takes Alice 1 time unit to transmit frame N, 4 time units for the roundtrip propogation delay, and it takes 1 more time unit for Bob to transmit the reverse piggyback frame that contains the ACK for frame N, that means that Bob cannot delay the sending of the ACK for more than 2 time units, in the hope that some reverse traffic arrives.