This homework is due at 11:59:59 PM on February 4, 2022 and is worth 3.75% of your grade.

Name: Melody Liu

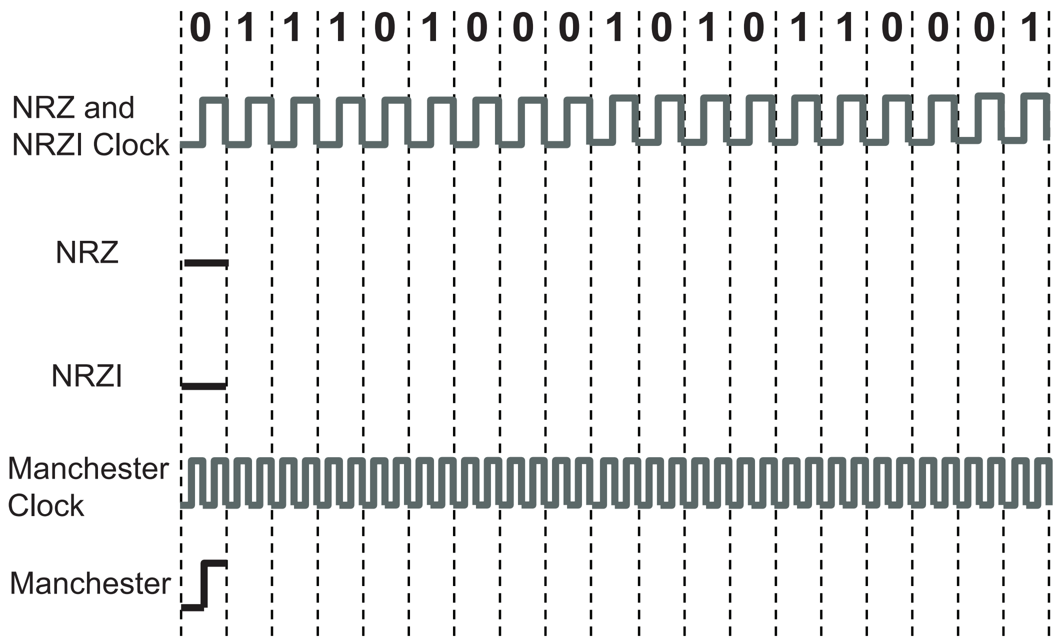
NUID (with leading zeros): 001390669

Problem	Possible	Score
1	15	
2	30	
3	15	
4	25	
5	15	
6	20	
Total	120	

1a. Briefly summarize the end-to-end argument in systems design (1-2 sentences). (5 pts)

1b. Suppose there is a proposal to add a "mobility" feature to the Internet, where end hosts could move between networks and the network devices (routers, switches, etc) would figure out where packets for the end host should be delivered without anyone they were communicating with knowing. What would the end-to-end argument say about where this proposal should be implemented? (10 pts)

2a. Draw in the NRZ, NRZI and Manchester encodings for the bit pattern below. The first bit has been encoded for you as an example. *Note that there are two clock signals shown, one corresponding to the NRZ and NRZI transmission, and one corresponding to the Manchester transmission.*



You can use Figure 2.10 of Peterson and Davie as a model. (10 pts)

2b. Apply the bit-stuffing protocol to the pattern below as if you were the sender and write down the resulting sequence in the boxes provided. You do not need to include any start frame/end frame sequences.

110100101111111101011111111

You may not need to use all of the boxes. (10 pts)

2c. If the bit pattern below is received at a bit-stuffing receiver, what is the interpretation of this pattern?

01111110101111111101101111100001111110

You may not need to use all of the boxes. If the bit sequence cannot be correctly decoded, write "Error" and explain why the sequence cannot be decoded. (10 pts)

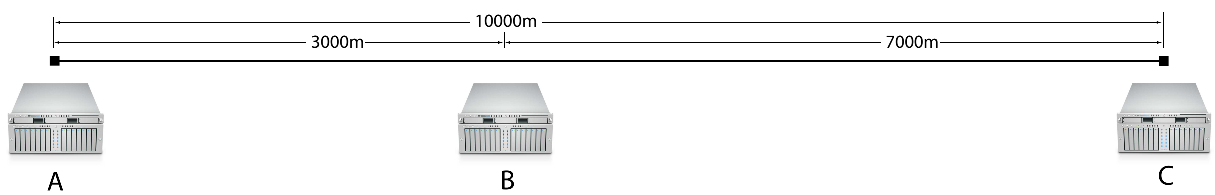
3a. Why do Ethernet packets have a *minimum* size in the Ethernet specification? (5 pts)

3b. How does Ethernet implement this minimum size requirement? What happens if a host wishes to transmit a smaller message? How does a receiver of such a message correctly understand what is going on? (5 pts)

3c. Why does the Ethernet protocol include a 64-bit preamble before every packet that consists of alternating 0s and 1s. (5 pts)

4a. Suppose that we have an Ethernet which has a bandwidth of 5 megabits/second. If the speed of light in copper is assumed to be 2.5×10^8 meters/second, what is the minimum frame size that we must select for a LAN of length 10,000 meters? *Note that there are 1000 bits in a kilobit, 1000 kilobits in a megabit, etc.* (10 pts)

4b. Suppose the layout of our LAN is as shown below.



What would happen if host A transmitted a frame that was smaller than this minimum frame size? Under what circumstances would problems occur? (10 pts)

4c. What is the minimum frame size that host B could send without any problems? (5 pts)

5a. Imagine that we are using a error detection scheme (Scheme A) that uses a single parity bit per *packet* (where packets can be up to 1,500 bytes in length). What is the overhead of this scheme? (3 pts)

5b. How "good" is Scheme A? How many bit errors could it detect? (2 pts)

5c. Imagine that there is a competing scheme (Scheme B) that uses a single parity bit per *byte*. What is the overhead of this scheme? Is it better or worse than Scheme A? (3 pts)

5d. How "good" is Scheme B? How does it compare to Scheme A? (2 pts)

5e. Suppose you were asked to deploy a network in an environment where there was significant electrical interference and noise. Which scheme would you recommend and why? (5 pts)

6. In class, we only discussed error *detection* schemes; there are also error *correction* schemes that can not only detect that a bit flip has occurred, but can also identify *which* bit has flipped. One such scheme is the *Hamming(7,4)* code, which can send 4 data bits using 7 transmitted bits, and can detect and correct all 1-bit errors. In brief, you can imagine this scheme as transmitting bits b_1, b_2, b_3, b_4 using parity bits p_1, p_2, p_3, where those parity bits are calculated as:

$$p_1 = \mathsf{parity}(b_1, b_2, b_4)$$

$$p_2 = \mathsf{parity}(b_1, b_3, b_4)$$

$$p_3 = \mathsf{parity}(b_2, b_3, b_4)$$

where the parity function returns the parity of the inputs (i.e., if the inputs have an odd number of 1s, it returns 1; otherwise it returns 0).

6a. Calculate the parity bits p_1, p_2, p_3 for the message $b_1 = 0, b_2 = 1, b_3 = 1, b_4 = 1$. (5 pts)

6b. Suppose that a receiver receives $b_1 = 0, b_2 = 0, b_3 = 1, b_4 = 1, p_1 = 1, p_2 = 0, p_3 = 1$. How can the receive determine *which* bit has flipped? (5 pts)

6c. Suppose that a receiver receives $b_1 = 0, b_2 = 1, b_3 = 1, b_4 = 1, p_1 = 1, p_2 = 1, p_3 = 1$. How can the receive determine *which* bit has flipped? (5 pts)

6d. What would happen at a receiver if *two* bits flipped? (5 pts)