CS164 — Winter 2000 Midterm

No aids allowed.

Problem 1. State whether each of the following statements is **true** or **false**.
(**Two points for each correct answer, –1 point for each incorrect answer.**)

a In a high speed ATM or SONET network, network speeds are described as "OC-3", "OC-12", "OC-48", etc., where the N in "OC-N" says the connection runs 2^N times faster than the basic speed of 1.5 Mbps.

FALSE – for two reasons, since the basic speed is 55 Mbps and the speeds increase linearly, rather than exponentially so that "OC-N" is N times 55 Mbps.

b A process running some layer-N protocol, $N > 1$, doesn't communicate directly with its peer on another host. Instead, it invokes functions provided by the service interface of a layer-$(N-1)$ protocol.

TRUE

c The acronym "CSMA/CD" stands for "Carrier-Sense Multiple Access with Collision Detection."

TRUE

d A parity code is designed for error *detection*, whereas a polynomial code (also called a CRC) is designed for error *correction*.

FALSE, it provides error detection, not error correction

e In a token ring, a station may need to start removing the front part of a packet before it has finished transmitting the end of that same packet.

TRUE

f Flow control algorithms protect the receiver by limiting the speed at which the transmitter can send data.

TRUE

g Gigabit Ethernet uses an 8B/10B code instead of a 4B/5B code, because it allows data to be sent twice as fast over the same wire.

FALSE – both an 8B/10B and a 4B/5B code expand the data stream by the same amount, 20%, so there is no inherent speed advantage. The only real difference is to increase the number of available non-data control symbols. NOTE: 8B/10B is a two-level code that is used for Gigabit Ethernet over a single optical fiber. A completely different encoding technique is used for Gigabit Ethernet over copper, which uses 5 signal levels in parallel over 4 wires.

h The Alternating Bit Protocol is a sliding window data link protocol that uses a window size of one frame.

TRUE

i In a token ring, the terms "early token release" and "multiple token operation" mean the same thing.

TRUE

j Bit stuffing is a technique for maintaining data transparency, which means the user can send any pattern of bits through the network (including a control symbol) without worrying about which patterns might cause the link to shut down.

TRUE

Problem 2. Recall that the distinctive feature in Manchester encoding that separates it from the other encoding schemes we talked about, such as NRZ and NRZI, is that Manchester always has state change in the middle of each bit (and possibly also at the boundary between bits), whereas the other schemes only allow state changes to take place at the boundary between bits.

a Briefly explain how to obtain the Manchester encoding for a message from its NRZ encoding and a clock signal that completes "low" / "high" cycle every bit time.

(5 POINTS) *NRZ encoding uses a fixed signal level for each bit, i.e., "low" for a logical-zero and "high" for logical-one. Assume the clock is a repeating pattern with a period of one bit time, in which the signal is "low" for the first half of each bit period and "high" for the second half of each bit time. Then Manchester is obtained by combining the NRZ data stream and the clock using the XOR operator.*

b Suppose Alice was not paying attention during class on the day we discussed the Manchester code, so she accidentally used the NRZI encoding of her message (instead of its NRZ encoding) in combination with the clock signal. Find the *simplest* feature in Alice's encoded data sequence that Bob could use to decode it. (HINT: draw a diagram!)

(5 POINTS) *The textbook shows the state change in NRZI to indicate a logical-one bit to occur in the middle of the period representing one bit, although in lecture I think I showed it at the start of the bit period. If we follow the textbook's rule, then the value of a logical-zero is constant over a bit, whereas the value of a logical-one changes in the middle of the bit, either from "low" to "high" or from "high" to "low". When we XOR this signal with the clock, then you can see that in Alice's code, there is a state change in the middle of a logical-zero, and no state change in the middle of a logical-one. If you follow the convention from lecture that has the state change at the start of a bit instead of at its center, then the above rule becomes the following. If there is a state change at the boundary between two bits, then the next bit is logical-one, otherwise the next bit is logical-zero.*

Problem 3. Manchester encoding may be thought of as a "1B/2B" code, since each data bit gets expanded into a two-bit channel code symbol of the form "01" or "10". Note that although there is always a change in the middle of the code word, it is possible for at most two consecutive bits to be the same (e.g., the pair of 1's that result if you send "01" followed by "10"). Find a "2B/3B" code in which pairs of data bits get expanded into a three-bit channel code symbol where (i) there is at least one change within the code word, and (ii) it is possible for at most three consecutive bits to be the same. (HINT: think of the restrictions we used in 4B/5B to limit the number of consecutive 0's).

(10 POINTS) *We must find four out of the eight possible 3-bit strings that satisfy both rules (i) and (ii). To satisfy the first rule, we cannot have all three bits in the codeword the same. Therefore, we must have either one or two changes in the bit value in each codeword. In the case of two changes, we only have two codewords available: "010" and "101". Clearly, both of these also satisfy the second rule since we can have at no consecutive bits the same inside either codeword, and the most consecutive bits we can have is two at the boundary between codewords. However, we still need to find two more codewords to complete our set, each of which will have only one change in bit value inside. These could be of the form "xyy" or "xxy". Let's select both additional codewords to have the same form, i.e., "011" and "100". This way we can a similar argument to the one for limiting the number of consecutive zeros in 5B/5B to show that the second condition holds. Specifically, we cannot have more than 3 consecutive zeros because no codeword ends with more than 2 zeros or starts with more than 1 zero, and similarly for consecutive ones.*

Problem 4. In this problem, we are going to estimate the limit on the bandwidth-delay product for a fiber optic channel caused by dispersion (i.e., spreading) of each pulse of light as it passes through the fiber. Therefore, we assume: (i) the average speed of light is 2×10^8 meters/sec through fiber; and (ii) if Alice transmits a narrow pulse of light in one end of the the fiber then the fastest part of the pulse travels 1% faster than average and the slowest part ofthe pulse travels 1% slower than average.

a Suppose Alice is transmitting data over the fiber at 100 Mbps (i.e., 10^8 bits/sec). What is the distance (in meters) between consecutive bits travelling along the fiber?

b After travelling what distance (in meters) over the fiber will the fastest part of one pulse for one catch up to the slowest part of the pulse for the previous bit? How does you answer change if we increase the data rate to 1 Gbps (i.e., 10^9 bits/sec)?

c How much variation in speeds could we tolerate in a system that uses a data rate of 10 Gbps and spans a distance of 1 kilometer if we require the gap between consecutive pulses of light can only shrink to half its original value?

Problem 5. Alice and Bob want use the polynomial 11011 compute a CRC that is added to the end of their messages for error detection. Find the CRC that will be appended to the message 10111100011.

5 POINTS) *This is a very straightforward application of the modulo-2 long division algorithm described in section 2.4, and illustrated in Figure 2-17. In this case, the remainder is 0011.*