CSCI 415 Computer Networks
Homework 3
Due 02/27/08

Saad Mneimneh
Computer Science
Hunter College of CUNY

Problem 1: Error correcting codes
Suppose that a parity check code has a minimum distance d. Define the distance of a code word and a received string (of the same length) as the number of bit positions in which the two are different (that's the Hamming distance).

(a) Show that if the distance between a code word and a given string is less than $d/2$, the distance between any other code word and the given string must exceed $d/2$.

(b) Show that if a decoder maps a given string into a code word at smallest distance from the string, all combinations of fewer than $d/2$ erros will be corrected.

Problem 2: CRC
(a) For the generator polynomial $g = 110011$ and the data bits (message) $m = 11100011$ find the CRC and the transmitted string T (since g is 6 bits, i.e. a polynomial of degree 5, $L = 5$ and the CRC should be 5 bits).

(b) Suppose $g = 1001$ and the received $T = 1010101$, did any transmission errors occur?

(c) Suppose $g = 101$ and the received $T = 1100110$, did any transmission errors occur?

(d) Suppose $g = 1011$ and $m = 10010$. Give the shift register implementation of the CRC generator and show the register sequence for generating the CRC with the above value for m.

(e) Suppose an error occurs only in the data bits (CRC bits unchanged). Argue that this error may not be detected.

(f) Suppose an error occurs only in the CRC bits. Argue that this error must be detected.

Problem 3: Stop and Wait variants

(a) Consider the Stop and Wait protocol such that:

- the receiver DLC sends a frame with request RN upon the receipt of each frame from the sender DLC
- the sender DLC sends frame SN upon the receipt of every frame from the receiver DLC with request $RN = SN$ (even if frame SN has not timed out yet)

Show by an example that if the sender times out, each packet can be sent twice (or more) over the network, which may cause congestion and more timeouts... (that was the problem with TFTP in 1980s). Suggest a fix to the protocol to alleviate this problem.

(b) Consider a Stop and Wait protocol where:

- instead of using sequence numbers, the sender uses one bit to signify whether the frame is original or a re-transmission (e.g. 0: original, 1:re-transmission)
- the receiver sends either an ACK (frame received with no errors) or a NAK (frame received with error) without any request number

Assume that frames are never lost. The sender simply waits (never times out) for either an ACK or a NAK or an erroneous frame.

Show by example that this strategy does not work correctly no matter what rule the receiver DLC uses for accepting frames.

(c) (optional) For scenario (b), what if the sender uses a counter instead of just a single bit, to indicate the re-transmission number. Therefore, it uses a 0 for the original frame, and any subsequent re-transmission of the frame takes a new number? What if we allow messages to be lost and the sender to timeout?