CSCI 415 Computer Networks
Homework 3
Solution

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 $d/2$.

ANSWER: By contradiction: Let x denote the string, and y and z be two codewords. If the distance between x and y is less than $d/2$, and the distance between x and z is less than or equal to $d/2$, then the distance between y and z is less than d, contradicting the assumption that the code has a minimum distance of d.

(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.

ANSWER: Let x be the string, and y be the codeword with minimum distance d' to that string. Because fewer than $d/2$ errors occurred, $d' < d/2$. From part (a), such codeword y is unique. Therefore, y must be the codeword that was transmitted.

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).

ANSWER: The remainder of division of 1110001100000 by 110011 is 11010, which constitute the CRC bits.

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

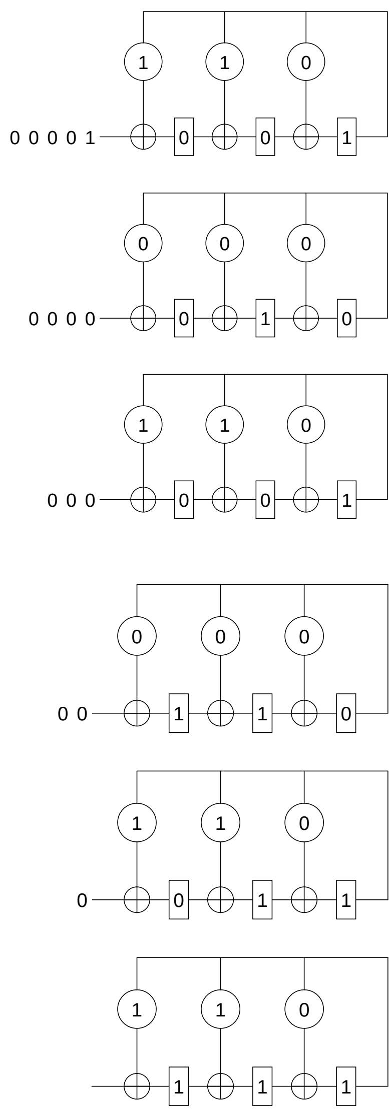
ANSWER: The remainder of division of 1010101 by 1001 is 110. Since it is different than 0, there must have been an error in transmission.

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

ANSWER: The remainder of division of 1100110 by 101 is 00. This is interpreted as no error.

(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.

ANSWER:



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

ANSWER: Let $T(x)$ be the transmitted message (i.e. $K + L$ bits). If we add $g(x)x^L$ to $T(x)$, we obtain $T'(x)$ which is a multiple of $g(x)$. Note that only the data bits (first K bits) in T are affected by the addition of $g(x)x^L$, since the number of CRC bits is L (Assume $K > L$)

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

ANSWER: Let $T(x)$ be the transmitted message and $T'(x)$ be the received message. Since error occurs only in CRC bits, $e(x) = T(x) - T'(x)$ is a degree $L - 1$ polynomial. Therefore, $e(x)$ is not divisible by $g(x)$.

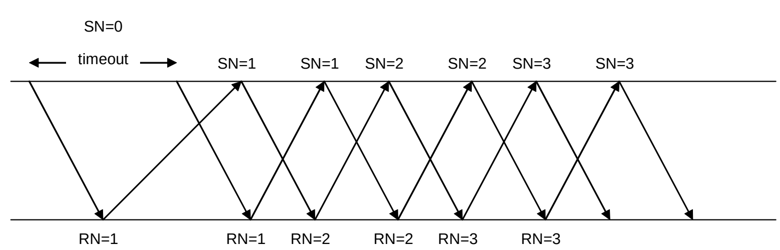
Problem 3: Stop and Wait variants

(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.

ANSWER:



To alleviate this problem, the sender can do the following: Only respond with a packet if $RN > SN$ (timeout takes care of retransmission).

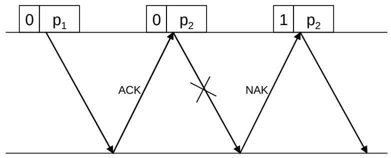
(b) Consider a Stop and Wait protocol where:

- 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

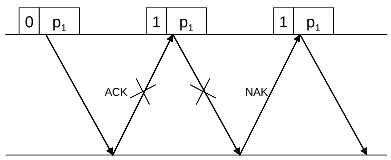
Assume that frames are never lost. The sender simply waits (never times out) for either an ACK or a NAK or an erroneous frame.

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

ANSWER: Consider the following scenario where packet 2 is received with error and is retransmitted.



The following scenario, where the ACK is received with error by the sender, is indistinguishable to the receiver from the previous one.



No matter what the receiver decides (packet 1 or packet 2), one of the scenarios will cause an error. If the receiver decides to NAK, it is only delaying the decision because all subsequent packets are going to be retransmission of the exact same one (with the same bit).