# CSC/ECE 570 Section 001

# Spring 2021

# Homework #3

**Keywords:** Data Link Layer, Framing, Error Correction, Error Detection

Name 1: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Student ID 1: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Name 2: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Student ID 2: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

## Instructions

* You can do this homework in groups of two (at most). Only one submission per group.
* The total number of points is 48.
* You must answer all questions for full credit.
* Use only this paper for your answers, in the space provided.
* The due date is as posted on the web page (please submit your answers through Wolfware).

# Questions: Answer the following questions. Justify your answers and be as precise as possible. Do not make unnecessary assumptions.

**[1]** **[4 points]** The following character encoding is used in a data link protocol:

A: 01000111 B: 11100011 FLAG: 01111110 ESC: 11100000

Show the bit sequence transmitted (in binary) for the four-character frame A B ESC FLAG when each of the following framing methods is used:

(a) Byte count.

(b) Flag bytes with byte stuffing.

(c) Starting and ending flag bytes with bit stuffing.

1. 01000111 11100011 11100000 11100000 01111110
2. 01000111 01111110 11100011 11100000 11100000 01111110
3. 0100011111010001111100000011111010

**[2]** **[4 points]** The following data fragment occurs in the middle of a data stream for which the byte-stuffing algorithm described in the text is used: A B ESC C ESC ESC ESC FLAG FLAG D. What is the output after stuffing?

A B ESC ESC C ESC ESC ESC ESC ESC ESC ESC FLAG ESC FLAG D

**[3] [4 points]** Calculate the 16-bit Checksum for the text of 8 characters (“Network”) at the sender. Also show how it be decoded in receiver if there are no transmission errors. USE ASCII (refer ASCII table ) to change each byte to a 2 digit hexadecimal number.

**[4] [4 points]** A bit string, 011110111110110110, needs to be transmitted at the data link layer. What is the string transmitted after bit stuffing?

0111101111100110110

**[5] [4 points]** To provide more reliability than a single parity bit can give, an error-detecting coding scheme uses one parity bit for checking all the odd-numbered bits and a second parity bit for all the even-numbered bits. What is the Hamming distance of this code?

Making one change to any valid character cannot generate another valid char- acter due to the nature of parity bits. Making two changes to even bits or two changes to odd bits will give another valid character, so the distance is 2.

**[6] [4 points]** Sixteen-bit messages are transmitted using a Hamming code. How many check bits are needed to ensure that the receiver can detect and correct single-bit errors? Show the bit pattern transmitted for the message 1111001100110101. Assume that even parity is used in the Hamming code.

We need 5 parity bits.

P1 P2 M3 P4 M5 M6 M7 P8 M9 M10 M11 M12 M13 M14 M15 P16 M17 M18 M19 M20 M21

011111100011001010101

**[7] [4 points]** One way of detecting errors is to transmit data as a block of n rows of k bits per row and add parity bits to each row and each column. The bit in the lower-right corner is a parity bit that checks its row and its column. Will this scheme detect all single errors? Double errors? Triple errors? Show that this scheme cannot detect some four-bit errors.

A single error will cause both the horizontal and vertical parity checks to be wrong. Two errors will also be easily detected. If they are in different rows, the row parity will catch them. If they are in the same row, the column parity will catch them. Three errors will also be detected. If they are in the same row or column, that row’s or column’s parity will catch them. If two errors are in the same row, the column parity of at least one of them will catch the error. If two errors are in the same column, the row parity of at least one of them will catch the error. A 4-bit error in which the four error bits lie on the four corners of a rectangle cannot be caught.

**[8] [4 points]** Suppose that data are transmitted in blocks of sizes 1000 bits. What is the maximum error rate under which error detection and retransmission mechanism (1 parity bit per block) is better than using Hamming code? Assume that bit errors are independent of one another and no bit error occurs during retransmission.

From Eq. (3-1), we know that 10 check bits are needed for each block in case of using Hamming code. Total bits transmitted per block is 1010 bits. In case of error detection mechanism, one parity bit is transmitted per block. Suppose error rate is *x* per bit. However, a block may encounter a bit error 1000*x* times. Every time an error is encountered, 1001 bits have to be retransmitted. So, total bits transmitted per block is 1001 + 1000*x* × 1001 bits. For error detection and retransmission to be better, 1001 + 1000*x* × 1001 *<* 1010. So, the error rate must be less than 9 x 10-6

**[9] [4 points]** Suppose that a message 1000 1100 1010 0011 is transmitted using Internet Checksum (4-bit word). What is the value of the checksum?

To obtain the checksum, we need to calculate the ones complement sum of words, which is same as sum modulo 24 and adding any overflow of high order bits back into low-order bits:

0011 + 1010 = 1101  
1101 + 1100 = 1001 + 1 = 1010 1010 + 1001 = 0011 + 1 = 1100.

So, the Internet checksum is 1100

**[10] [4 points]** What is the remainder obtained by dividing x7 + x4 + 1 by the generator polynomial x2+1?

By Dividing x7 + x4 + 1 by x2+1 we get x5 - x3 + x2 + x as quotient and reminder is -x2 – x +1

Hence answer is -x2 – x +1

**[11] [4 points]** Find the status of the following generators related to two isolated, single-bit errors

1. x+1
2. x4 +1
3. x7 + x6 + 1
4. x15 + x14+ 1
5. No xi is divisible by (x+1) hence single-bit errors can be detected. But any two errors next to each other cannot be detected.
6. This generator cannot detect any two error that are four positions apart
7. This is a good choice as it detects
8. This polynomial cannot divide xt + 1 if t is less than 32,768. A codeword with two isolated errors up to 32,768 bits apart can be detected by this generator

**[12] [4 points]** A 1024-bit message is sent that contains 992 data bits and 32 CRC bits. CRC is computed using the IEEE 802 standardized, 32-degree CRC polynomial. For each of the following, explain whether the errors during message transmission will be detected by the receiver:

(a) There was a single-bit error.

(b) There were two isolated bit errors.

(c) There were 18 isolated bit errors.

(d) There were 47 isolated bit errors.

(e) There was a 24-bit long burst error.

(f) There was a 35-bit long burst error.

The CRC checksum polynomial is of degree 32,

(a) Yes. CRC catches all single-bit errors.

(b)  Yes. CRC catches all double-bit errors for any reasonably long message.

(c)  No. CRC may not be able catch all even number of isolated bit errors.

(d)  Yes. By making ( x +1 ) as a factor of G(x) CRC catches all odd number of isolated bit errors.

(e)  Yes. CRC catches all burst errors with burst lengths less than or equal to 32.

(f) No. CRC may not be able to catch a burst error with burst length greater than 32.