CSCI 4171 Networks and Communications
CSCI 6704 Advanced Topics in Networks
Fall 2024
Assignment No. 2
Date Given: Tuesday, September 24, 2024
Date Due: Monday, October 7, 2024, 11.59 PM on Brightspace

This assignment has two programming questions from Module 3: LAN Technologies.

Submission: One ZIP file uploaded on Brightspace. Please see instructions at the end of this assignment.

Submission Deadline: Monday, October 7, 2024, 11.59 PM

Grace Time: Submissions will be accepted until 4.59 AM on Tuesday, October 8, 2024 without late penalty.

Late Penalty: Submissions received after the grace time will be subject to a 10% per day late penalty, for up to 5 days. For example, if you submit the assignment on Tuesday, October 8 at 12 noon and your score is 8/10, it will be reduced to 7.2/10. Submissions past five days after the grace submission time will not be accepted. The submission portal will close on Sunday, October 13, 4.59 AM.

Dropping of Assignments: One out of six assignments can be dropped during the semester. No SDA submission required.

This assignment has two programming questions. In the first exercise, you will "simulate" various digital encoding schemes, and in the second exercise, you will emulate the bit stuffing protocol to understand how it works.

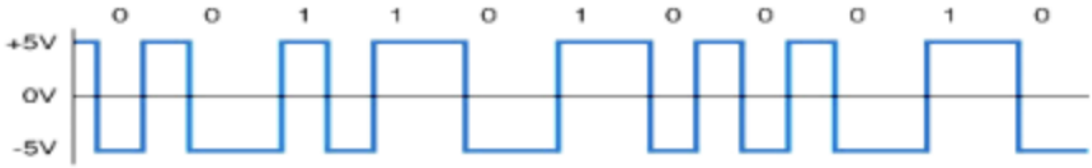
Programming Exercise 1: Write a program that accepts a binary string as input and generates waveforms using a GUI corresponding to the following encodings of the input binary string:
a) Unipolar
b) NRZ
c) Manchester

For example, if the input binary string is 0 0 1 1 0 1 0 0 0 1 0, your program should display the following waveforms using a GUI. For Manchester, you may assume that the signal is "High" to begin with.

Unipolar

NRZ

Manchester



Notes on programming:

1. Your output need not look exactly like the ones shown above, as long as it simulates each type of encoding. You can use any similar type of GUI output.
2. The code may be written in Java, Python, C++ or C.
3. You can use standard libraries (e.g., JavaSwing API) for generating the waveforms.
4. In your submission folder, you must include the source code (for example, .java, .py) and a README.txt file for the marker with instructions on how to run your program.

Programming Exercise 2 <Bit Stuffing Program> In this exercise, you will be writing a simple program that does the following:
a) Read a String of hex digits
b) Convert the String into a String of binary numbers
c) Perform bit stuffing on the binary String
d) Unstuff the bits from the binary String
e) Produce the original hex String

Here's an example to illustrate the program's operation:

Input: ABEFFFF
Conversion to binary: 10101011111011111111111111111
After bit stuffing: 101010111110011111101111101111101
After bit unstuffing: 10101011111011111111111111111
Output: ABEFFFF

Your program must have at least four methods/functions:
a) convert Hex String to a binary String; b) convert binary String to stuffed binary String c) convert stuffed binary String to original binary String and d) convert binary String to Hex String

Note: You can represent all the data types using Strings (you are not actually creating raw binary data).

You may use Java, C, C++ or Python as your programming language.
Submit the source code, and at least three samples of your program run in your submission package.

What to submit: You must submit one zip file that contains the following:

a) Source code(s) for Exercise 1
b) README text file for Exercise 1
c) Sample input and output for Exercise 1
d) Source code(s) for Exercise 2
e) README text file for Exercise 2
f) Sample input and output for Exercise 2

Make sure that your **full name and banner ID** appears on the top of the text documents and in comments in your source codes.