CSE 322: Computer Networks Sessional

Assignment 4: Error Detection and Correction

In this assignment, you will implement error detection (using CRC checksum) and error correction (using Hamming distance), which is a part of data link layer. To understand the fundamentals, carefully go through Section 3.2 of the textbook [1]. The requirements for this assignment are summarized below.

There are 4 inputs to your program.

1. *data string*, which is the string to be transmitted.
2. *m*, which indicates the number of bytes in a row of the data block. (Note that, in the reference book, *m* denotes number of data bits, which is different from our denotation.)
3. *p*, which indicates the probability of each bit being toggled during the transmission.
4. *generator polynomial*, which is used for calculating and verifying CRC checksum.

The tasks to be implemented are as follows.

1. If the size of the data string is not a multiple of *m*, append the padding character (~) to the data string accordingly. Print the updated data string.

2. Create the data block, which contains the 8-bit binary representation of the ascii codes of the characters of the data string. Each row contains ascii codes of *m* characters. The first row shows the first *m* characters, the second row shows the next *m* characters, and so on. Print the data block. Note that, there will be *l/m* rows in the data block, where *l* is the length of the padded data string.

3. Add check bits to correct at most one-bit error in each row of the data block (according to page 195, figure 3-7 of [1]). Print the updated data block. Note that, the check bits must be shown in a different color (green) [2].

4. Serialize the above data block in column-major manner. Print the serialized data bits.

5. Compute the CRC checksum of the above bit stream using the generator polynomial (according to page 197-198, figure 3-8 of [1]). Append the checksum to the bit stream. This is the frame to be transmitted. Print the frame. Note that, the appended checksum must be shown in a different color (cyan).

6. Simulate the physical transmission by toggling each bit of the stream with a probability of *p*. Print the received frame. Note that, the erroneous bits must be shown in a different color (red).

7. Verify the correctness of the received frame using the generator polynomial (according to page 198 of [1]). Print the result of the error detection procedure.

8. Remove the CRC checksum bits from the data stream and de-serialize it into the data-block in a column-major fashion. Print the data block. Note that, the erroneous bits must be shown in a different color (red).

9. Correct the error in each row according to the method described in page 195 of [1]. Observe that, if there is more than one error in a row, this error correction mechanism will fail to correct the error. Print the data block after correcting the errors and removing the check bits.

10. From the bits of the data block, compute the ascii codes of the characters. Print the data string.

Marking Criteria:

Error detection (CRC): 15
Error Correction (Hamming Code): 20
Output formatting and Compatibility: 15
Total: 50

Others:

1. Carefully go through the test cases provided to better understand how your code should work. Your submitted code will be tested against a different set of test cases to test for correctness.

2. Note that, you do not need to implement communication (sockets, etc.) between the peers. Just simulate the transmission by introducing random errors as discussed in step 6 above.

3. The output format (color differences, new lines, etc.) should exactly match the test case files. Deviation from the expected output format will result in heavy penalty.

4. In practice, a layer would add either error correcting or error detecting codes but not both. In this assignment, contrary to standard practice, you will implement both error correction and detection for learning/educational purpose.

Submission Guideline:

Create a folder with your student id (7 digit). Put all your source file(s) inside the folder. Zip that folder. This should create a .zip file with your student id. Submit that zip file in moodle. ***Failing to adhere to this guideline will result in straight 30% penalty. And, as usual, in case any kind of academic dishonesty/ unfair means is found, you are sure to receive -100% penalty. Recurring offenses will result in an "F".***

References:

[1] Section 3.2, Computer Networks, 4th edition, Andrew S. Tanenbaum

[2] Changing color in windows console: http://www.cplusplus.com/forum/beginner/102810/