EE 565 Homework 1: Link Layer

Instructor: Patrick Tague, Winter Quarter 2008

Homework Due: 10:30AM on 23 January 2008

---

1. Consider a string of n data bits and 1 parity bit computed as the single parity check of the n data bits. Suppose that at the receiver side, each bit is flipped independently with probability p. Give an expression for the probability $P(n, p)$ that the received string contains errors that are not detected. Using MATLAB or other software, plot the probability $P(n, p)$ as a function of n ranging from 2 to 50 for the three values $p = 1/30$, $p = 1/10$, and $p = 1/3$. As an optional extension, derive an approximation for $P(n, p)$ that doesn't require a summation over multiple terms and plot the approximation on the same set of axes.

2. The sender's link layer receives a data packet from the network layer containing the data 1010111 1111001 1111010 0100011.

   (a) *Two-Dimensional Parity Check*: Arrange the 28-bit data into a matrix by taking each 7-bit block above as a row. Compute the parity checks added by the two-dimensional parity check. Re-arrange the resulting data and parity bits by sequentially appending the resulting matrix rows, and give the resulting codeword X (data bits and parity check bits). Give an error pattern E of minimum weight (minimum number of 1's) such that the received bit string $Y \equiv X + E \pmod 2$ is a valid codeword, i.e. such that no error is detected when Y is received.

   (b) *Cyclic Redundancy Check*: Let the generator polynomial $g(D)$ be given by the product of $(D + 1)$ and the primitive polynomial $p(D) = D^4 + D^3 + 1$. Treat the 28-bit data as two 14-bit blocks b_1 and b_2 and compute the CRC c_1 of block b_1 and the CRC c_2 of block b_2. Compute the codeword X given by concatenating the bit strings b_1, c_1, b_2, and c_2 in that order. Given an error pattern E of minimum weight such that the received bit string $Y \equiv X + E \pmod 2$ is a valid codeword.

3. For each of the codewords X computed in parts 1(a) and 1(b) in the previous problem, perform bit-oriented framing of X using the bit-stuffing rule with the special character 01^60 at the beginning and end of each frame. For each resulting frame, compute the fraction of bits in the frame dedicated to overhead (parity bits and framing bits).

4. (Problem 2.33 from Bertsekas & Gallager's *Data Networks*) Suppose that the string 0101 is used as the bit string to indicate the end of a frame and the bit stuffing rule is to insert a 0 after each appearance of 010 in the original data; thus 010101 would be modified by stuffing to 01001001. In addition, if the frame proper ends in 01, a 0 would be stuffed after the first 0 in the actual terminating string 0101. Show how the string 11011010010101011101 would be modified by this rule. Describe the destuffing rule required at the receiver. How would the string 11010001001001100101 be destuffed?

5. Consider the sliding-window ARQ techniques with a window size n and sequence number SN and request number RN computed mod m, i.e. $SN, RN \in \{0, \dots, m-1\}$.

   (a) Demonstrate a case where the go back n ARQ algorithm fails when $m = n$.

   (b) Demonstrate a case where the selective repeat ARQ algorithm fails when $m = 2n - 1$.