

### Direct Link Networks and Error Correction Codes

Assigned reading: Peterson and Davie: Chapter 2.1 – 2.4. All problems carry equal weight.

#### 1. Noisy Channel Data Rates

The decibel is a measure of the ratio between two signal levels:  $N_{db} = 10 \log_{10} (P_2/P_1)$ , where  $N_{db}$  = the number of decibels,  $P_1$  = the input power level and  $P_2$  = the output power level.

- a. A telephone line is known to have a loss of 20db. The input signal power is measured as 0.50 watt and the output noise is measured as 8  $\mu$ watt. Using this information, calculate the output signal-to-noise ratio in dB.

$P_1$  = the input power level = 0.50 watt, and  $P_2$  = the output power level that we need to find.

$$10 \log (P_2/P_1) = -20\text{dB}$$

$$P_2/P_1 = 0.01$$

Since  $P_1 = 0.50$  watt,  $P_2 = 0.0050$  watt

$$\text{SNR} = 0.0050/(8 \times 10^{-6}) = 625$$

$$\text{SNR}_{\text{dB}} = 10 \log (625) = 27.95 \text{ dB}$$

- b. What is the capacity of this phone line with a frequency range of 100 Hz – 1000 Hz?

Using Shannon's law

$$C = B \log_2 (1 + S/N)$$

$$C = (1000-100)\log_2 (1 + 625)$$

$$C = 900 \times 6.43935 = 5795.415$$

- c. If the attenuation rate of this phone line is 2db/km, and the minimum output signal is 0.00025 watt, given the input signal from part a), how long can the phone line be before requiring a repeater?

$$10 \log (P_2/P_1) = 10 \log (0.00025/0.50) = -33.01\text{dB}$$

$$\text{Max length} = 16.505\text{km.}$$

#### 2. Encoding, Channel Capacity and Modulation

- a. Show the NRZ, Manchester, NRZI and 4B/5B encoding signals (the resulting NRZI signal for 4B/5B), using a diagram similar to that in Figure 2.10, p. 82, P&D, for the data bit sequence 1011 1101 0011 0111 0110. To be definite, suppose the NRZI signals begin at low voltage.
- b. What signal-to-noise ratio (in dB) is needed to put a T4 carrier (274.176 Mbit/s) on a 100-MHz line?

Using Shannon's law

$$C = B \log_2 (1 + S/N).$$

$$274.176 * 10^6 = 100 * 10^6 * \log_2 (1 + S/N).$$

$$S/N = 2^{2.74176} - 1 = 5.688.$$

$$\text{dB} = 10 \log (S/N).$$

$$\text{dB} = 10 \log (5.688).$$

$$\text{dB} = 7.549 \text{ dB.}$$

- c. Suppose it was possible to send 8 different types of signals on a link (0,0,0), (0,0,1) (0,1,0) ... (1,1,1). How many bps can such a link achieve at 1200 baud?

There are 8 symbols. So, each symbol carries 3 bits.  
 Baud rate = symbols per sec = 1200  
 So, data rate = 3\* 1200= 3600bps.

- d. A modem constellation diagram has data points at (0, 1) and (0, 2). Does the modem use phase modulation or amplitude modulation?

Phase shift between the 2 points is 0. Only the amplitude changes.  
 So, the modem uses Amplitude Modulation(AM).

### 3. Framing

Consider the data bit sequence 0010 0110 1011 0100 1010 1001 1101 1011 1101 1111 0000 1110. In this problem, you are required to frame these bits in three different ways. (*In the solution, the bold bits are those added extra because of the framing*)

- a. Frame the above bits with byte stuffing as used in the BISYNC protocol. You need show only the body (including stuffed bytes) and the sentinel bits. DLE is 10 (decimal), STX is 2, and ETX is 3.

No modifications needed

STX 0000 0010	Body 0010 0110 1011 0100 1010 1001 1101 1011 1101 1111 0000 1110	ETX 0000 0011
------------------	---------------------------------------------------------------------	------------------

- b. Second, frame the bits using bit stuffing as defined by the HDLC protocol. Again, you need show only the (stuffed) data bits and the sentinel bits.

Flag 01111110	Body 0010 0110 1011 0100 1010 1001 1101 1011 1101 1111 <b>0</b> 0000 1110 1110	Flag 01111110
------------------	-----------------------------------------------------------------------------------	------------------

- c. Third, frame the bits using the SONET STS-1 protocol.

HHH	Body 1101 1001 0100 1011 0101 0110 0010 0100 0010 0000 1111 0001
-----	---------------------------------------------------------------------

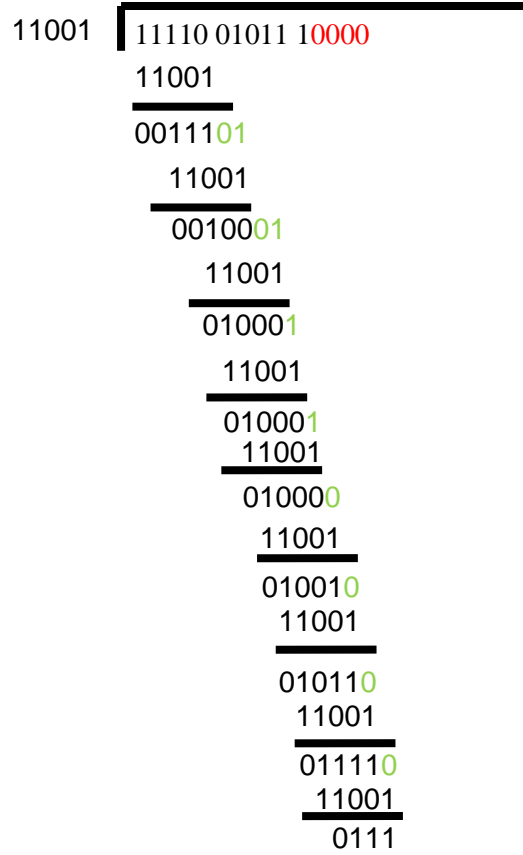
*HHH indicates the overhead bytes. The XOR is done with a sequence of 1's.*

- d. Calculate the framing efficiency (as a percentage of real data per bit sent) of your answers to (a), (b), and (c).

For (a), 64 bits are transmitted for 48 bits. So efficiency is  $48/64 = 75\%$   
 For (b), 65 bits are transmitted for 48 bits. So efficiency is  $48/65 = 73.84\%$   
 For (c), 60 bits are transmitted for 48 bits. So efficiency is  $48/60 = 80\%$

### 4. Error Detection

- a. A CRC is constructed to generate a 4-bit checksum for an 11-bit message. The generator polynomial is  $x^4 + x^3 + 1$ . Encode the data bit sequence 11110010111. Now assume that bit 4 (counting from the least significant bit) in the code word is in error and show how the error is detected.



M(x) = 11110010111  
T(x) = 111100101110000  
C(x) = 11001  
R(x) = 0111  
P(x) = 111100101110111 (These is the transmitted bits)

```

11001 | 11110 01011 11111
      11001
      -----
      0011101
        11001
        -----
        0010001
          11001
          -----
          010001
            11001
            -----
            010001
              11001
              -----
              010001
                11001
                -----
                010001
                  11001
                  -----
                  010001
                    11001
                    -----
                    010001

```

Remainder  $\neq 0$  and thus the error can be detected.

- b. The bit sequence **101100100001** corresponds to the polynomial  $x^{11}+x^9+x^8+x^5+1$ . Divide this polynomial by the CRC generator polynomial  $x^4+1$  and report the remainder as a polynomial. Is the bit sequence correctly encoded with the given generator (i.e., is the remainder 0)?

```

10001 | 101100100001 0000
      10001
      -----
      0011101
        10001
        -----
        011000
          10001
          -----
          010010
            10001
            -----
            00011001
              10001
              -----
              010000
                10001
                -----
                00001000

```

Remainder  $\neq 0$  and thus the sequence cannot be encoded correctly.

- c. A block of bits with  $n$  rows and  $k$  columns uses horizontal and vertical parity bits for error detection. Suppose that *exactly* 4 bits are inverted due to transmission errors. Derive an expression for the probability that such an error will be undetected. Assume  $nk > 16$ .

*Hint:* Using basic Combinatorics, count the number of cases in which you will not be able to detect bit inversions.

Describe an error pattern as a matrix of  $n$  rows by  $k$  columns. Each of the correct bits is a 0 and each of the incorrect bits is a 1. With 4 errors per block, each block will have exactly 4 1s. How many such blocks are there? There are  $nk$  ways to choose where to put the first 1 bit,  $nk - 1$  ways to choose the second, and so on, so the number of blocks is  $nk(nk - 1)(nk - 2)(nk - 3)$

*Undetected errors only occur when the four 1 bits are at the vertices of a rectangle.*

First we look at the case of forming 1 rectangle. Using cartesian coordinates, every one bit is at a coordinate  $(x, y)$  where  $0 \leq x < k$  and  $0 \leq y < n$ . Suppose that the bit closest to the origin (the lower left hand vertex) is at  $(p, q)$ . The number of legal rectangles is  $(k-p-1)(n-q-1)$ . Then the total number of rectangles can be found by summing this formula for all possible  $p$  and  $q$  which is

$$N = \sum_{p=0}^{n-2} \sum_{q=0}^{n-2} (k-p-1)(n-q-1)$$

The probability of an undetected error is then the number of such rectangles divided by the number of ways to distribute the four bits =

$$\frac{N}{nk(nk - 1)(nk - 2)(nk - 3)}$$

## 5. Internet checksum

Suppose that one byte in a buffer covered by the Internet Checksum algorithm needs to be decremented (e.g. header hop count field). Give an algorithm to compute the revised checksum without re-scanning the entire buffer. Your algorithm should consider whether the byte in question is low order or high order.

You can assume that you are given the original checksum, the position of the byte which is to be decremented and the value by which the byte will be decremented.

### Basic Algorithm:

Take the 1's complement of the checksum. If the byte to be decremented by  $x$  is of lower order, then subtract  $x$  from it else subtract  $2^8 * x$  from it using 1's complement arithmetic. Taking the 1's complement of the result gives the checksum of the modified buffer.

## 6. Networking Utilities

Read the manual pages for the Unix utility `netstat` on an `eelnxxx.ews.uiuc.edu` machine. Try running the command using various options. (Not all options at the same time!). Among other things, you can use the command: `netstat -an` to displays the currently active network connections (both incoming and outgoing), routing tables, and a number of network interface statistics. In particular you are required to find and report the following statistics **along** with the particular options you were required to use to obtain this information.

- a. the local and remote socket addresses (where a socket address is a host address plus a port number) for three TCP sockets in use (any status) on an `eelnx` workstation.

command: `netstat -anv`

- b. the Ethernet address of eelnx24 and eelnx25. (Hint: You needn't log onto these stations. You might have to ping them though, in order to make sure the eelnx machine you are on has recently seen an Ethernet packet from each of the two machines.)

command: netstat -p

- c. the number of packets sent and the number of packets received by a eelnx machine in a 20 second interval (report the machine name and time of day for this data collection). Use the primary network interface for this purpose. You are required to report how you determined which is the primary interface of the machine you were using. (Hint: Use netstat!).

command: netstat -i -I <main\_interface> 10

Note: In future problem sets you will be asked to explore the Unix networking utilities nslookup, ifconfig, and arp.

**Solution for 2a**

