CS/ECE 438: Communication Networks for Computers Spring 2018
Problem Set 2 Due: start of class, Wednesday, February 14th

All problems carry equal weight. To receive full credit, show all of your work.

Physical Layer

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 0.25 watt and the output noise is measured as 10 µwatt. Using this information, calculate the output signal-to-noise ratio in dB.

   P_1 = the input power level = 0.25 watt, and P_2 = the output power level that we need to find.
   $10 \log (P_2/P_1) = -20dB$
   $P_2/P_1 = 0.01$
   Since $P_1 = 0.25$ watt, $P_2 = 0.0025$ watt

   $SNR = 0.0025/(10 \times 10^{-6}) = 25000$
   $SNR_{dB} = 10 \log (25000) = 43.99$ dB

b. What is the capacity of this phone line with a frequency range of 300 Hz – 4800 Hz?

   Using Shannon's law
   $C = B \log_2 (1 + S/N)$
   $C = 4500 \log_2 (1 + 16)$
   $C = 4500 \times 14.61 = 65745$

c. If the attenuation rate of this phone line is 8 dB/km, and the minimum output signal is 0.005 watt, given the input signal from part a), how long can the phone line be before it requires a repeater?

   $10 \log (P_2/P_1) = 10 \log (0.005/0.25) = -16.99$ dB
   Max length = 2.12 km.

2. Encoding

a. Bit and baud rates. Suppose, it is possible to send 128 different types of signals on a link, and that there is no noise. How many bits per second (bps) can such a link achieve at 4000 baud?

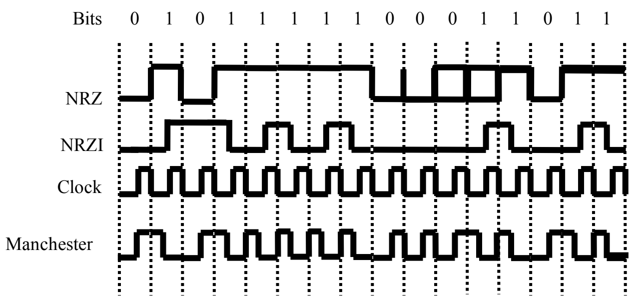
   $\log_2 128$ bits per symbol at 4000 symbols per second is 28000 bps.

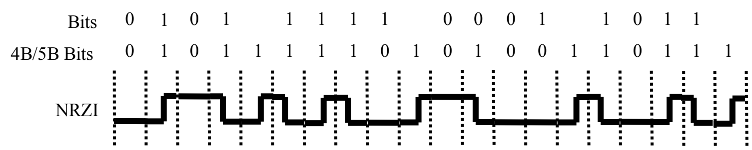
b. **SNR**. What signal-to-noise ratio (in dB) is needed to put a 7 Gbps carrier on a 500-MHz line? (Note: for line speeds in networking, giga-, mega-, kilo- indicate powers of 1000, not 1024.)

   Shannon's law says $C = B \log_2(1 + S/N)$. Solving for S/N, we have
   $S/N = 2^{C/B} - 1 = 2^{(7*10^9)/(500*10^6)} - 1 = 8191$
   In decibels, this ratio is $10 \log_{10} 8191 = 39.13$

3. Encoding and Channel Capacity

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 the class slides, for the data bit sequence 1111 0001 1011. To be definite, suppose the NRZI signals begin at low voltage.





b. In 1962, Bell Labs introduced the first version of their Transmission System 1 (T-1). Subsequent specifications carried multiples of the basic T1 data rates. What signal-to-noise ratio is needed to put a T3, 672channel, carrier on a 75-MHz line?

Using Shannon's law
$C = B \log_2 (1 + S/N)$.
$44.736*10^6 = 75*10^6 * \log_2 (1 + S/N)$.
$S/N = 2^{0.59648} - 1 = .512$
$dB = 10 \log (S/N)$.
$dB = 10 \log (.512)$.
$dB = 2.907\ dB$.

4. Modulation

a. A modem constellation diagram has data points at the following coordinates: (1, 1), (1, -1), (-1, 1), (-1, -1), (2, 2), (2, -2), (-2, 2), (-2, -2), (3, 3), (3, -3), (-3, 3), (-3, -3), (4, 4), (4, -4), (-4, 4), and (-4, -4). How many bps can a modem with these parameters achieve at 1600 baud?

There are 16 symbols. So, each symbol carries 4 bits.
Baud rate = symbols per sec = 1600
*So, data rate = 4 * 1600 = 6400 bps.*

b. A modem constellation diagram has data points at (-12, 4) and (-48, -16). Does the modem use phase modulation and/or amplitude modulation? Explain your answer.

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

5. Framing

Consider the data bit sequence 0000 1010 1111 1111 0101 0000 0110 0000 1111 1111 0000 1110. In this problem, you will frame these bits in three ways.

a. First, frame the 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 ASCII character 16 (decimal), STX is 2, and ETX is 3.

STX 0000 0010	Body **0000 1010** 0000 1010 1111 1111 0101 0000 0110 0000 1111 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 0000 1010 1111 1011 1010 1000 0011 0000 0111 1101 1100 0011 10	Flag 01111110

c. Third, frame the bits into 8-bit RS-232 characters. Use “0” to represent start bits and “1” to represent stop bits.

Start 1	Data 0000 1010	Stop 0	Start 1	Data 1111 1111	Stop 0	Start 1	Data 0101 0000	Stop 0	Start 1	Data 0110 0000	Stop 0	Start 1	Data 1111 1111	Stop 0	Start 1	Data 0000 1110	Stop 0

d. Now, counting only the bits that you wrote, calculate the efficiency (as a percentage of real data per bit sent) of your answers to (a), (b), and (c).

In part (a), 72 bits were sent for 48 bits of data, so the efficiency is 48/72 = 66.66%. In part (b), 67 bits were sent for 48 bits of data, for an efficiency of 72.73%. In part (c), 60 bits were sent for 48 bits of data, for an efficiency of 80%.

6. 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 **10000110100**. Now assume that bit 4 (counting from the most significant bit) in the code word is in error and show how the error is detected.

```
11001 | 100001101000000
        11001
        -----
        010011
         11001
         -----
          10101
          11001
          -----
           011000
            11001
            -----
            000011000
                11001
                -----
                00001000 = R(x)
```

M(x) = 10000110100
T(x) = 100001101000000
R(x) = 1000
P(x) = 100001101001000

```
11001 | 100001101001100
        11001
        -----
        01001
         11001
         -----
          010101
           11001
          ------
           011000
            11001
           ------
            000011001
                11001
               ------
                00000100 → not zero, hence error
```

b. The bit sequence **10010110011** corresponds to the polynomial $x^{10}+x^7+x^5+x^4+x+1$. Divide this polynomial by the CRC generator polynomial x^3+1 and report the remainder as a polynomial. Is the bit sequence correctly encoded with the given generator (i.e., is the remainder 0)?

```
1011 | 10010110011000
       1011
       ----
       001001
         1011
        -----
         001010
           1011
          -----
           0001011
              1011
             -----
              0000000 → correctly encoded
```

c. Suppose a 4 bit CRC is appended to an n bit message according to the CRC polynomial $x^4 + x + 1$. The encoded message thus has $n + 4$ bits. What is the largest value of n such that any double bit error can be detected? (Hint: any error sequence corresponds to a polynomial that is the product of C(x) and some other polynomial.)

We first seek the shortest nonzero sequence that when multiplied by 10011 using no carries (i.e., polynomial multiplication) produces a sequence with only two 1's. The sequence should start and end with a 1. The idea is to select the other bits of the multiplier so that the product also has 1's for the first and last bits, and all other bits 0.

```
          10011    Do multiplication without carries
     1.........??1    Fill in multiplier to cause 0's in product
     -------------
          10011
         10011
        10011       These are shifted the right
       10011        amounts to get an even number
     10011          of 1's in each column except
    10011           first and last
   10011
 10011
 ---------------
1000000000000001    This is 10011 * 100110101111, so is a valid codeword
```

Thus, the shortest undetected error sequence with two 1's has 16 bits. Therefore, in order to prevent undetectable double bit errors, the largest value of n possible is such that n + 4 = 15, or n = 11 data bits.

7. Networking Utilities

a. The Unix utility `ping` can be used to find the round trip time (RTT) to various Internet hosts. See the man page for `ping` to see how to use it and the -s option with other options to see how you can control the time between ICMP packet transmissions, and to display the resulting round trip times. Upon interrupting execution of `ping`, the min, average and maximum RTT will also be displayed. Here is what you turn in:

Report the average (average over ten pings) round trip times for pings to the following domains:

cs.illinois.edu:	~ 0.257ms
www.illinois.edu:	~ 0.756ms
www.nps.gov:	~ 8.856 ms
www.cambridge.uk:	~ 62.772 ms
sydney.edu.au:	~289.255ms

Note: slight variations in these numbers might happen, due to the fact that, as we will learn in this class, Internet routes are highly dynamic, and our probe packets must compete with real traffic, too.

b. The Unix utility `traceroute` is like ping, but it sends packets that are limited to go one hop, then two hops, then three hops, and so on, towards a given destination, and the intermediate routers are reported. Read the man page for `traceroute` and experiment with it. Try `traceroute` on the servers that you just pinged. Report the number of routers that are encountered along the way. Answer these questions: what is the relation between this number and the RTT? Is the number of hops the only factor affecting the RTT?

cs.illinois.edu:
www.illinois.edu:
www.nps.gov:
www.cambridge.uk:
sydney.edu.au:

(these counts are off by one with respect to the count given by traceroute, which accounts for the final destination, too).

Obviously the number of routers encountered has an influence on the RTT. The CS website is directly accessible from the ews machines, since it is located in the same building. The central webserver is in another location on campus, which requires a few hops. In this case the longer delay is a consequence of the queuing and processing time. The number of hops to Stanford and Australia is similar. In this case the much longer delay is caused by the long distance the signal must travel, too.

Note: slight variations in these numbers might happen, due to the fact that, as we will learn in this class, Internet routes are highly dynamic. However, it is very unlikely that reaching our www server will ever take longer than reaching the Stanford one.