CSE 123: Computer Networks

Homework 1 Solutions

Out: 4/6, Due: 4/13

Total Points = 25

1. **MESSENGER Spacecraft [4 pts]**
   NASA launched a space probe, called MESSENGER, designed to study the planets Mercury and Venus. Assume MESSENGER uses an XBand (i.e., RF) communication system to achieve a 250 kbps point-to-point link between it and the Earth station. The distance between Earth and Mars is approximately 75 million km.
   a. Calculate the minimum RTT and the bandwidth-delay product for the link
      i. RTT = 2*dist/signal propagation speed = 2*(75*10^9 m) / (3*10^8 m/s) = 500 seconds
      ii. Bandwidth-delay product = 250*10^3 * 500 = 125*10^6 bits
          Will also accept => Bandwidth x RTT/2 = 62.5*10^6 bits
   b. Let us assume that the space probe sends 10MB of data back to earth periodically. What would be the total time for the Earth station to get the entire 10MB of data? (ie. the time taken for a single 10MB data transfer)
      i. Total time = transmit time + propagation time = (8*10*1024*1024) / (25*10^4) + 500 / 2 = 585.54432 seconds

   Assume the speed of light through outer space is 3x10^8 m/s

2 pts for part a

- 1 pt for the right formula/setting it up
- 1 pt for the correct answer
  - 0.5 pts if almost correct, but off due to an arithmetic error

2 pts for part b

- 1 pt for the right formula/setting it up
  - 0.5 pts if using 2-way propagation time without stated assumption that receiver will send acknowledgements
- 1 pt for the correct answer (or correct follow through if 0.5 pts from above)
  - 0.5 pts if almost correct, but off due to an arithmetic error or because units are wrong for file size and data rate

2. **Nyquist Theorem [3 pts]**
   The Nyquist Theorem gives us a lower limit on our sampling frequency in order to correctly reproduce a given input signal. For this problem assume the input signal has a frequency “f”.
   a. What is the minimum rate at which we would need to sample the input signal in order to be able to accurately reproduce the signal?
      i. 2f
   b. Explain why you cannot sample at a lower frequency.
      i. If you sample at a lower frequency, you cannot reproduce the original signal because it will cause aliasing in reconstructing the original signal. The lowest frequency signal that matches the sampling points will not be the same one that was actually transmitted.

1 pt for a for correctness
2 pts for b

- 2 pts for good answer that refers to being unable to reconstruct the original signal
  - 1 pt for a weak answer that at least hints in the right direction

3. **2B/9B Encoding Scheme [7 pts]**
   Similar to a 4B/5B encoding scheme let us concoct a new encoding 2B/9B. That is, every 2 bits of actual data will have a 9 bit code associated with it.

2B	**9B**
00	000000000
01	101010101
10	010101101
11	011110000

a. How many errors can be detected using the above set of codewords?
    i. Up to 3 because that's the smallest codeword distance
b. How many errors can be corrected if any?
    i. Correction bits = 2d+1 <= 4. Since d can be at most 1, 1 error can be corrected.
c. Is this an efficient encoding? Why or why not?
    i. No, not all codewords are equidistant
d. What is the efficiency of this 2B/9B encoding scheme?
    i. 2/9 = 0.222 => 22.2%
e. Consider another encoding scheme that is the same except for the encoding of "11" which becomes is instead "011110010". How many errors can be detected with the codewords in this scheme?
    i. Up to 4
f. With the scheme from e, how many errors can be corrected if any?
    i. 2d+1 <= 5, so 2 errors can now be corrected
g. With the scheme from e, is this an efficient encoding? Why or why not?
    i. Still not efficient because some encodings are 6 bits different

1 pt for a for correctness
1 pt for b for correctness
1 pt for c
- 0.5 pts for No
- 0.5 pts for codewords are not equidistant

1 pt for d for correctness
1 pt for e for correctness
1 pt for f for correctness
1 pt for g
- 0.5 pts for No
- 0.5 pts for codewords are still not equidistant (some 5, some 6)

4. **HDLC Framing [4 pts]**
   The following bit sequence arrives over the link

   0111 1111 1011 1111 0101 1011 1111 1011 1110 1101 1111 0011 1110 0111 1110

   or if you strictly followed how HDLC was defined in the book

   0111 1111 1011 1111 0101 1011 1111 1011 1110 1101 1111 0011 1110 0111 1110

   If the HDLC protocol was used for framing, and assuming an end frame was sent just before this sequence, mark the following
   a. Start of frame
      i. Yellow highlight
   b. End of frame
      i. Orange highlight
   c. Stuffed bits
      i. Cyan highlight
   d. Bits indicating errors
      i. Green highlight

In either of the above solutions:
1 pt for correct location(s) of start of frame(s)
1 pt for correct location of end of frame (not at all in the second case)
1 pt for correct location(s) of stuffed bits (not at all in the second case)
1 pt for correct location of error bit, with acceptable notations as follows

- The first bit at which the receiver will recognize an error "0111 1111"
- Marking the bit where a stuffed bit should have been "0111 1111"
- The whole chunk of the sequence that signifies an error "0111 1111"

**(0.5 pts instead of 1 for each of the above that are partially correct)

5. **CRC [7 pts]**

Suppose we want to transmit a message 1101 1001 0101 1001 and protect it from errors using the CRC generator x^8 + x^2 + x^1 + 1 ?

a. What is the CRC generator sequence? (The CRC generator polynomial represented in a bit sequence)?
    i. This CRC would be represented as 100000111
b. How many bits will the resulting frame check sequence be?
    i. 8 bits
c. What is the transmitted bit sequence (show your work)?

```
                    1101101101011001
          ------------------------
100000111(110110010101100100000000
          100000111|||||||||||||||
          ---------|||||||||||||||
           101101011||||||||||||||
           100000111||||||||||||||
           ---------||||||||||||||
             110110001||||||||||||
             100000111||||||||||||
             ---------||||||||||||
              101101101|||||||||||
              100000111|||||||||||
              ---------|||||||||||
                110101000|||||||||
                100000111|||||||||
                ---------|||||||||
                 101011111||||||||
                 100000111||||||||
                 ---------||||||||
                   101100000||||||
                   100000111||||||
                   ---------||||||
```

```
110011100||||
100000111||||
---------||||
 100110110|||
 100000111|||
 ---------|||
    110001000
    100000111
    ---------
     10001111
```

```
Transmitted sequence = 110110010101100110001111
```

d. How large a burst of errors can be detected?
   i. A burst of size 8

1 pt for a for correctness
1 pt for b for correctness
4 pts for c

- 1 pt for proper setup of “polynomial division” (including padding)
- 1 pt for using XOR operations for modulo-2 subtraction
- 1 pt for correct CRC (remainder)
  - 0.5 pts if incorrect due to obvious arithmetic error
- 1 pt for appending the CRC to the end of the message to get the “transmitted bit sequence”

1 pt for d for correctness