EECS 122: Introduction to Communication Networks

Homework 8 Solutions

Solution 1. In the absence of transmission errors, alternating bit delivers one message per round-trip time, while a sliding window protocol with a window of W packets delivers W messages per round-trip time. So the throughput and efficiency simply increase by a factor of W, except that if the throughput reaches the channel rate (the efficiency reaches 100%) then further increases in W do not help. The results are therefore:

window size	throughput	efficiency
10 packets	999.60 packets/second	0.35%
100 packets	9996.0 packets/second	3.5%
1000 packets	99,960 packets/second	35%
10,000 packets	284,090 packets/second	100%

The throughput in the last row is the maximum packet rate of the channel (1 packet$/\tau$, see the solution to homework 7 problem 2).

Solution 2.

a) Let M be the boolean matrix corresponding to the transmitted message (including the parity bits) and let $M(i,j)$ denote the i-th row, j-th column element of M. Let M' be the message received. Possibly, $M' \neq M$. We assume that M' differs from M in at most three elements.

If M' and M differ in only one element, say $M'(i_1,j_1) \neq M(i_1,j_1)$, then the parity bits of row i_1 and column j_1 are both wrong.

If M' and M differ in two elements, say $M'(i_1,j_1) \neq M(i_1,j_1)$ and $M'(i_2,j_2) \neq M(i_2,j_2)$, then: either $i_1 \neq i_2$, in which case the parity bits of rows i_1 and i_2 are both wrong; or $j_1 \neq j_2$, in which case the parity bits of columns j_1 and j_2 are both wrong.

If M' and M differ in three elements, say $M'(i_1,j_1) \neq M(i_1,j_1)$, $M'(i_2,j_2) \neq M(i_2,j_2)$ and $M'(i_3,j_3) \neq M(i_3,j_3)$, then: either all rows i_1, i_2 and i_3 are different, in which case the parity bits of all three rows are wrong; or two rows among i_1, i_2 and i_3 are the same, in which case the parity bit of the other row is wrong; or $i_1 = i_2 = i_3 = i$, in which case the parity bit of row i is wrong.

b)

$$M = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \qquad M' = \begin{pmatrix} 1 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

c) Assuming at most one bit error, if $M'(i_1, j_1) \neq M(i_1, j_1)$, then the parity bits of row i_1 and column j_1 are both wrong, indicating which element was inverted.

d) If two bit errors occur in the same row, then the parity bits for the two columns will be wrong, but the parity bits for all rows will be correct, so the decoder cannot know which row the errors are in. Similarly if the errors are in the same column. If the errors are in neither the same row nor the same column, the parity bits for both columns and both rows will be wrong, but the decoder cannot know whether the errors are in the upper-left and lower-right intersections, or in the upper-right and lower-left intersections.

Solution 3.

```
11001001001010100000000  |  100000111
100000111                |
---------                |
010010101                |
 100101010               |
 100000111               |
 ---------               |
 000101101               |
    101101101            |
    100000111            |
    ---------            |
    001101010            |
      110101001          |
      100000111          |
      ---------          |
      010101110          |
       101011100         |
       100000111         |
       ---------         |
       001011011         |
         101101100       |
         100000111       |
         ---------       |
         001101011       |
           110101100     |
           100000111     |
           ---------     |
           010101011     |
            101010110    |
            100000111    |
            ---------    |
            001010001    |
              101000100  |
              100000111  |
              ---------  |
              001000011  |
                10000110 |
```

Figure 1: CRC division for problem 3a.

a) We have to divide (without carry) $M \cdot 2^8$ by $G = 100000111$, where

$$M \cdot 2^8 = 1100100100101010 \cdot 2^8 = 110010010010101000000000$$

The division (in bit-string form) is shown in figure 1. The remainder is found to be $R = 10000110$. So, the transmitted codeword is

$$T = M \cdot 2^8 + R = 110010010010101010000110$$

```
1111000100101010100001l0  |  100000111
100000111                 |
---------                 |
011100101                 |
 111001010                |
 100000111                |
 ---------                |
 011001101                |
  110011011               |
  100000111               |
  ---------               |
  010011100               |
   100111000              |
   100000111              |
   ---------              |
   000111111              |
      111111101           |
      100000111           |
      ---------           |
      011111010           |
       111110100          |
       100000111          |
       ---------          |
       011110011          |
        111100111         |
        100000111         |
        ---------         |
        011100000         |
         111000000        |
         100000111        |
         ---------        |
         011000111        |
          110001110       |
          100000111       |
          ---------       |
          010001001       |
           100010010      |
           100000111      |
           ---------      |
           000010101      |
               101010110  |
               100000111  |
               ---------  |
               001010001  |
```

Figure 2: CRC division for problem 3b.

b) The received codeword is $T' = 111100010010101010000110$. The decoder computes the remainder of the division of T' by G as shown in figure 2. The remainder is found to be $E = 1010001$.

c) The receiver knows an error has occurred because E is not zero.

The receiver cannot generally correct the error. The statement "the 1s in E tell where the error occurred" in the lecture slides is not generally correct. If the added "noise" (corruption by the channel) is divisible by G then the resulting E does not indicate where the errors occurred. In any case, CRC is mainly used for error detection, not correction (see slide on CRC properties).

Solution 4.

a) The encoded message (output of the automaton) corresponding to input $M = 101001$ is $T = 110100011111$ (the automaton follows the path $00 \to 10 \to 01 \to 10 \to 01 \to 00 \to 10$).

b) The received message is $T' = 101100011111$. In figure 3 you can see the tree computed by the decoder. The decoder chooses the path $00 \to 10 \to 01 \to 10 \to 01 \to 00 \to 10$, which has penalty $1+1+0+0+0+0 = 2$. This path corresponds to the automaton input 101001 (which is the original message) and this is what the decoder outputs.

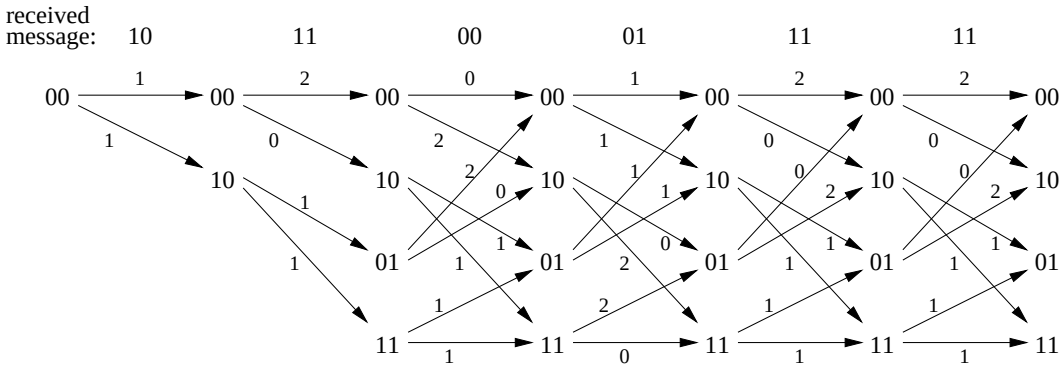

Figure 3: Convolutional decoding for problem 4b.

c) Assume the error was in one of the last two bits, say, the received message is $T'' = 110100011110$. Then, the decoder would find two paths with same penalty 1, namely, the correct path (above) and the path $00 \to 10 \to 01 \to 10 \to 01 \to 00 \to 00$. If the decoder chooses the second path, then it outputs the wrong message 101000.

Solution 5.

a) In our time-slotted model (which is an approximation of what really happens), the source sends a window of one packet during the first round-trip time, two during the second, and so on, until it sends 33 packets during the 33rd and the last packet is dropped, causing the cycle to repeat. The throughput is $1+2+\ldots+31+32+32=$ 560 packets per cycle, and each cycle is 33 round-trip times. The channel can carry 32 packets per round-trip time, so the efficiency is

$$\frac{\text{throughput}}{\text{channel rate}} = \frac{\frac{560 \text{ packets}}{\text{cycle}} \cdot \frac{\text{cycle}}{33 \text{ RTT}}}{\frac{32 \text{ packets}}{\text{RTT}}} \doteq 53.0\%$$

b) The larger the channel capacity n, the less it matters that the window ramps from 1 to $n+1$ rather than 0 to n. The average window size approaches $n/2$, so the efficiency approaches $1/2$.

c) The threshhold for transitioning from slow start to congestion avoidance might start out large, but it will be reduced after the first loss or two, and the connection will settle into a repeating pattern: the source will grow the window from 1 packet to 16 in powers of 2 (slow start), then grow it 1 packet at a time (congestion avoidance) up to 33, which causes a loss, and then the cycle repeats. Since the efficiency is in terms of the average throughput, the behavior at the beginning of the connection doesn't matter. The throughput is $1+2+4+8+16+17+18+\ldots+31+32+32=455$ packets per cycle, and each cycle is $5+16+1=22$ round-trip times. The efficiency is

$$\frac{\text{throughput}}{\text{channel rate}} = \frac{\frac{455 \text{ packets}}{\text{cycle}} \cdot \frac{\text{cycle}}{22 \text{ RTT}}}{\frac{32 \text{ packets}}{\text{RTT}}} \doteq 64.6\%$$

d) First, let's assume that the channel capacity n is a power of 2. If n is large, the time spent in slow start ($\log_2 n$ round-trip times) is negligible compared to the time spent in congestion avoidance ($n+1$ round-trip times). The window is almost always ramping from $n/2$ to $n+1$, so as n increases the average window size approaches $3n/4$, so the efficiency approaches $3/4$. This is an acceptable solution.

If n is slightly less than a power of 2, the window reaches only about $n/4$ before switching to congestion avoidance. The window is almost always ramping from $n/4$ to $n+1$, so as n increases the average window size approaches $5n/8$, so the efficiency approaches $5/8$.

If n is unrestricted, then as n increases the efficiency takes on all values between $5/8$ and $3/4$ infinitely often.