# HW6 Solutions

**Problem 1**

1 1 1 0 1

0 1 1 0 0

1 0 0 1 0

1 1 0 1 1

1 1 0 0 0

**Problem 3**

Unsigned ASCII code for “Networking”:

0100 1110, 0110 0101,

0111 0100, 0111 0111,

0110 1111, 0111 0010,

0110 1011, 0110 1001,

0110 1110, 0110 0111.

0100 1110 0110 0101

+ 0111 0100 0111 0111

------------------------------

1100 0010 1101 1100

+ 0110 1111 0111 0010

------------------------------

0011 0010 0100 1111 (with wrap around)

+ 0110 1011 0110 1001

-----------------------------

1001 1101 1011 1000

+ 0110 1110 0110 0111

------------------------------

**0000 1100 0010 0000** (with wrap around)

The one's complement of the sum is 1111 0011 1101 1111.

**Problem 7**

1. Without loss of generality, suppose ith bit is flipped, where 0<= i <= d+r-1 and assume that the least significant bit is 0th bit.

A single bit error means that the received data is K=D\*2r XOR R + 2i. It is clear that if we divide K by G, then the reminder is not zero. In general, if G contains at least two 1’s, then a single bit error can always be detected.

1. The key insight is that G can be divided by 11 (binary), but any binary number of odd-number of 1’s cannot be divided by 11. Thus, any sequence of (not necessarily contiguous) odd-number of bit errors cannot be divided by G. Thus, any odd number of bit errors is detectable.

Note 1: Here “divided” refers to modular-2 division, without borrowing.

Note 2: Proof of “any binary number of odd-number of 1’s cannot be divided by 11”:

We prove by contradiction using the concept of “generator polynomial”. Let E(x) be the generator polynomial of the error, e.g., E(x) = xi + xj + 1 denoting bit errors in the (i+1)-th bit, the (j+1)-th bit, and the first bit (i>j>0). G(x) = x3+1 is the generator polynomial of G=1001. Suppose that E can be divided by 11, i.e., E(x) contains a factor of (x+1). Then we have E(x) = (x+1) F(x), for some polynomial F(x). In particular, E(1) = (1+1) F(1). The lefthand side equals 0 (modular-2 arithmetic), but the righthand side equals 1 as E(x) contains an odd number of terms (corresponding to an odd number of bit errors). We have a contradiction. Thus, E(x) cannot contain a factor of (x+1), i.e., E cannot be divided by 11.

**Problem 8**

1. Let *E(p)* denote the efficiency under transmission probability *p*.











 

Thus



**Problem 10**

1. A’s average throughput is given by pA(1-pB).

Total efficiency is pA(1-pB) + pB(1-pA).

1. A’s throughput is pA(1-pB)=2pB(1-pB)= 2pB- 2(pB)2.

B’s throughput is pB(1-pA)=pB(1-2pB)= pB- 2(pB)2.

Clearly, A’s throughput is not twice as large as B’s.

In order to make pA(1-pB)= 2 pB(1-pA), we need that pA= 2pB / (1+ pB).

1. A’s throughput is 2p(1-p)N-1, and any other node has throughput p(1-p)N-2(1-2p).

**Problem 20**

1. Let  be a random variable denoting the number of slots until a success:

,

where  is the probability of a success.

This is a geometric distribution, which has mean . The number of consecutive wasted slots is  that







efficiency 

b)

Maximizing efficiency is equivalent to minimizing , which is equivalent to maximizing . We know from the text that  is maximized at .

c)

efficiency 

efficiency 

d) Clearly,  approaches 1 as .

**Problem 31**

(The following description is short, but contains all major key steps and key protocols involved.)

Your computer first uses DHCP to obtain an IP address. You computer first creates a special IP datagram destined to 255.255.255.255 in the DHCP server discovery step, and puts it in a Ethernet frame and broadcast it in the Ethernet. Then following the steps in the DHCP protocol, you computer is able to get an IP address with a given lease time.

A DHCP server on the Ethernet also gives your computer a list of IP addresses of first-hop routers, the subnet mask of the subnet where your computer resides, and the names and addresses of local DNS servers (if they exist).

Since your computer’s ARP table is initially empty, your computer will use ARP protocol to get the MAC addresses of the first-hop router.

Your computer first will get the IP address of the web server you would like to download the webpage from. Your computer will use DNS protocol to find the IP address of the web server. This involves constructing a DNS query, encapsulated into a UDP segment, then an IP datagram with destination address being the local DNS server’s IP address (obtained from DHCP), and finally an Ethernet frame with the destination MAC address being the first-hop router’s MAC address (obtained from ARP). The first-hop router and subsequent routers (if any) will route the query to the local DNS server. If the local DNS server does not have a cached entry for the requested web server name, it will query the DNS hierarchy and then respond to your computer with the IP address of the web server.

Once your computer has the IP address of the web server, then it will establish a TCP connection to the server using the obtained IP address and destination port number 80. Establishing the connection involves your computer sending a TCP SYN packet, the server responding a TCP SYNACK packet, and your computer sending an ACK for the SYNACK (3-way handshake). The last handshake, ACK, can carry the HTTP request for the desired webpage in the payload. Specifically, the HTTP request message will be segmented and encapsulated into TCP packets, and then further encapsulated into IP packets, and finally encapsulated into Ethernet frames. Your computer sends the Ethernet frames destined to the first-hop router. Once the router receives the frames, it passes them up into IP layer, checks its routing table, and then sends the packets to the right interface out of all of its interfaces.

Then your IP packets will be routed through the Internet via BGP and IGPs until they reach the web server.

The server hosting the Web page will send back the web page (the base html) to your computer via a HTTP response message. The message will again be encapsulated into TCP packets and then further into IP packets. Those IP packets follow IP routes and finally reach your first-hop router, and then the router will forward those IP packets to your computer by encapsulating them into Ethernet frames.

If the web page contains embedded objects (e.g., images, video clips, animations), your computer will parse the base html and send out additional HTTP requests for each of these objects, by default in a batch using the same TCP connection (persistent HTTP with pipelining). If web caching is used, then your computer will send out a conditional HTTP Get request instead of a normal HTTP Get, and the web cache/proxy server will serve the request locally if it has an updated copy of the requested web page.