COMP/ELEC 429/556 Fall 2017
Homework #1

Assigned 9/28/2017

Due 10/12/2017 11:55pm

Submit Electronically to Canvas

(Hard deadline, no slip day may be used)

This homework is worth 10% of your final grade

IMPORTANT: To be completed by each student individually. See syllabus for policy details. Type set your responses either in the space provided in this document or in a separate document. No hand-written responses will be accepted.

Name: ______________________

Email: ______________________

Student ID: ___________________

Notations: K=1,000, M=1,000,000, G=1,000,000,000, 1 byte (B) = 8 bits (b).

Problem 1 (20 points) – A student wants to measure as accurately as possible the time it takes to send 1GB, in 10KB chunks, to a remote computer using a `SOCK_STREAM` socket. To do so, he writes a sender program which contains the following code fragment:

```
/* some standard code to create a socket
 * and to establish a connection
 */
... code omitted ...

fcntl(send_socket, F_SETFL, O_NONBLOCK);
gettimeofday(&start, NULL);
for (i=0; i<100000; i++) {
         chunk = malloc(10000);
         send(send_socket, chunk, 10000, 0);
         free(chunk);
}
gettimeofday(&end, NULL);
/* compare start and end to obtain
 * the elapsed time
 */
```

There are two fundamental design flaws in this code fragment. One causes the program to sometimes send fewer than 1GB. Another one causes the program to be very inaccurate. Identify these flaws and provide a revised code fragment that removes these flaws. You may find the `man` pages on CLEAR for the various system calls useful.

The first flaw is that the `send_socket` has been placed in non-blocking mode. As a result, each `send` call may return without actually sending 10KB of data. The second flaw is that `malloc` and `free` are repeatedly called within the loop, so the measured time includes irrelevant operations.

```
/* some standard code to create a socket
 * and to establish a connection
 */
... code omitted ...

chunk = malloc(10000);
gettimeofday(&start, NULL);
for (i=0; i<100000; i++) {
         send(send_socket, chunk, 10000, 0);
}
gettimeofday(&end, NULL);
/* compare start and end to obtain
 * the elapsed time
 */
free(chunk);
```

Problem 2 (10 points) The weakness of the NRZI bit representation scheme is that a long sequence of 0's in the input bit stream will cause the electrical signal to contain no voltage transition for a long period of time, thus making it difficult for the receiver to detect clock drifts. One solution to this problem is to translate the input bit stream into an encoded bit stream that eliminates long sequences of 0's.

Design a way to translate every 3-bit sequence in the input bit stream to a 4-bit encoded sequence such that the resulting encoded bit stream is guaranteed to contain no more than 2 consecutive 0's. Write down your answer in the shaded boxes below.

3-bit input sequence		**4-bit encoded sequence**
000	→	0111
001	→	1001
010	→	1010
011	→	1011
100	→	0101
101	→	1101
110	→	1110
111	→	1111

There are several possible correct answers. One is provided above. The strategy is to choose a set of 4 bit codes such that the number of leading and trailing zeros in each code is less than or equal to one, thus when the 4 bit codes are concatenated, there can be no more than two consecutive zeros.

Problem 3 (29 points) -- Suppose you are designing a multi-access (i.e. broadcast) wired network using CSMA/CD (Carrier Sense Multiple Access/Collision Detect) mechanisms similar to the Ethernet for media access control. Assume that in your design, the network's link speed is 15Mbps, frame sizes may range from 200 bytes to 2000 bytes. Also assume that data signal propagates on the network link at a speed of $2*10^8$ meters per second.

(a) To guarantee that a collision is detected, what is the maximum allowable link length (in meters) between any pair of hosts connected to the same link? Show your work. (15 points)

To guarantee that a collision is detected, the link length should be provisioned for the worst case scenario. The worst case is when A sends a message, immediately before A's first bit arrives at B, B starts sending and collides with A. For A to detect the collision, A must transmit until B's first bit has propagated to A, which is bounded by the transmission time of the smallest allowed packet (200B). Let d be the maximum link length, then

$2*d/(2*10^8) < 200*8/(15*10^6)$
d < 10,667 meters

(b) State the reasons why broadcast Ethernet, where all hosts on the network share one single channel (i.e. wire) and CSMA/CD (carrier sense multiple access/collision detect) is used to arbitrate media access among the hosts, cannot (1) support a large number of hosts and still maintain high throughput, or (2) support hosts spread across a large geographic area. (14 points)

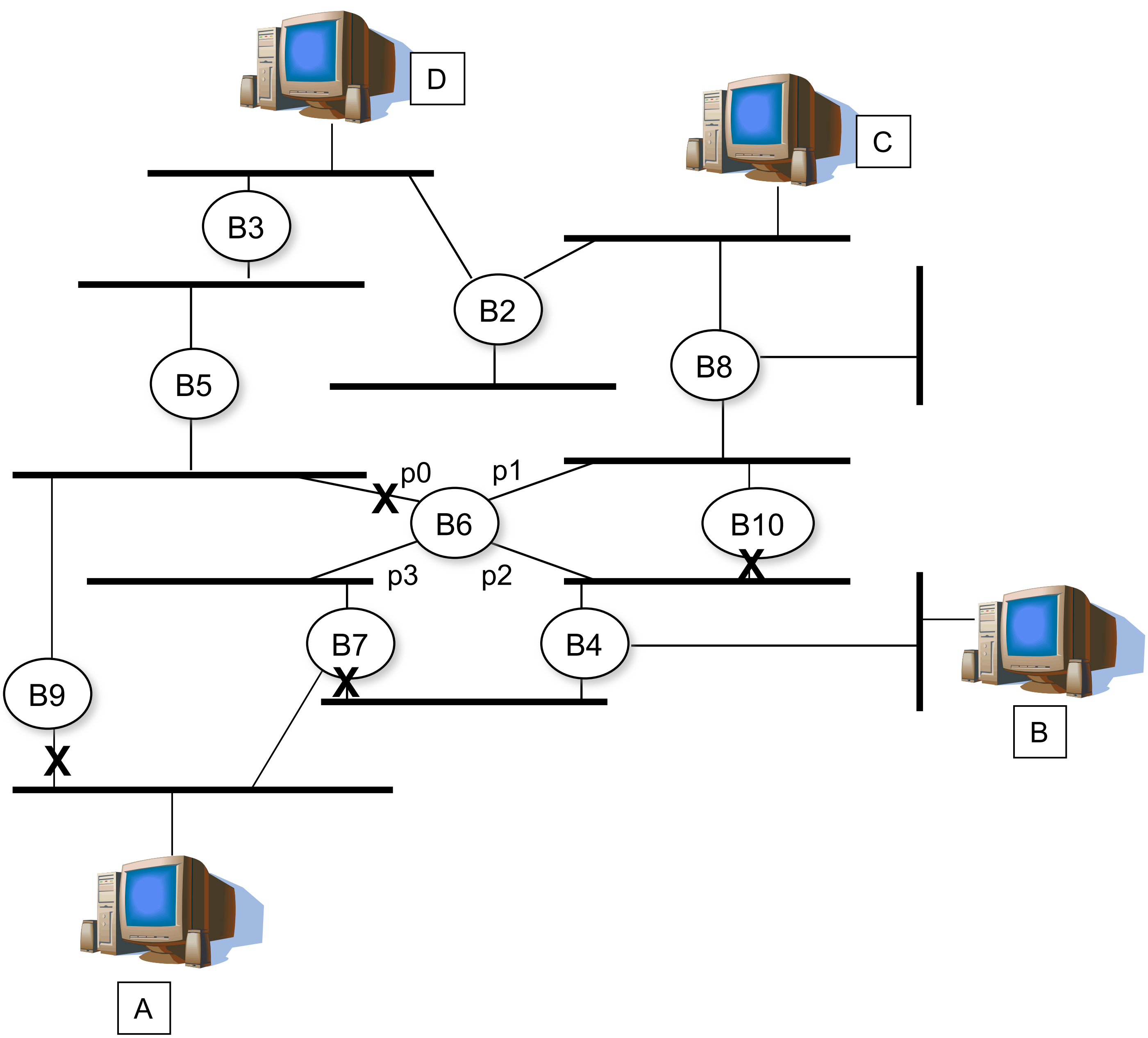
(1) Since the medium is shared and has a limited bandwidth, throughput will be very poor due to collisions when a large number of hosts are sharing the medium.
(2) To implement collision detection correctly, the maximum propagation delay of the network must be limited and related to the minimum packet size, thus as the physical length of the network increase, the minimum packet size must also increase, leading to inefficiency when sending small messages.

Problem 4 (20 points) Consider a bridged Ethernet shown below. There are 4 computers A, B, C, and D in the network. The ports of bridge B6 are labeled p0, p1, p2, p3 respectively. The bridge IDs are the numeric values. Assume the forwarding tables of all bridges are currently empty.

(a) Indicate which ports are blocked by the bridge spanning tree protocol by putting an "X" over the corresponding ports. (10 points)

(b) With the spanning tree established, suppose A transmits a single packet addressed to B, and C transmits a single packet addressed to D. Explain how the Ethernet bridges forward these packets and how they learn forwarding table entries. Use the space on the next page to write your answer. (7 points)

(c) After the transmissions of the two packets have been completed and the network is idle, what is the content of the forwarding table at bridge B6? (3 points)



(blank page for answer)

If a bridge doesn't have a forwarding table entry for the destination address in a packet, it floods the packet on all its ports except the port on which the packet is received. The packet will eventually reach its destination. Each bridge notices the incoming port P of a flooded packet and the source address X in that packet. The bridge then adds a forwarding table entry that uses port P for subsequent forwarding to address X.

Dest	Port
A	p3
C	p1

Problem 5 (21 points): Suppose Sammy is designing a new sliding window based reliable transport protocol for use in the Internet. In particular, in Sammy's design, he uses (i) a 4 bit packet sequence number to label each packet, (ii) a retransmission timeout of 20ms, and (iii) a window size of 14 packets.

Provide three detailed reasons in terms of the protocol's correctness or performance to argue that Sammy's design is a poor design.

Possible answers:

4 bit sequence number limits the window size to only 8 packets, which severely limits maximum throughput.

Fixed retransmission timeout of 20ms means when RTT >> 20ms, every packet is transmitted unnecessarily, wasting bandwidth; when RTT << 20ms, a lost packet will not be retransmitted quickly, reducing throughput.

A 4 bit sequence number is not enough to implement a window size of 14 packets correctly.

A fixed window size of 14 packets may not properly match the available bandwidth of the network.