CSE 123: Computer Networks

Homework 2 Solutions
Total Points: 30

Student Name:
PID:
UCSD email:

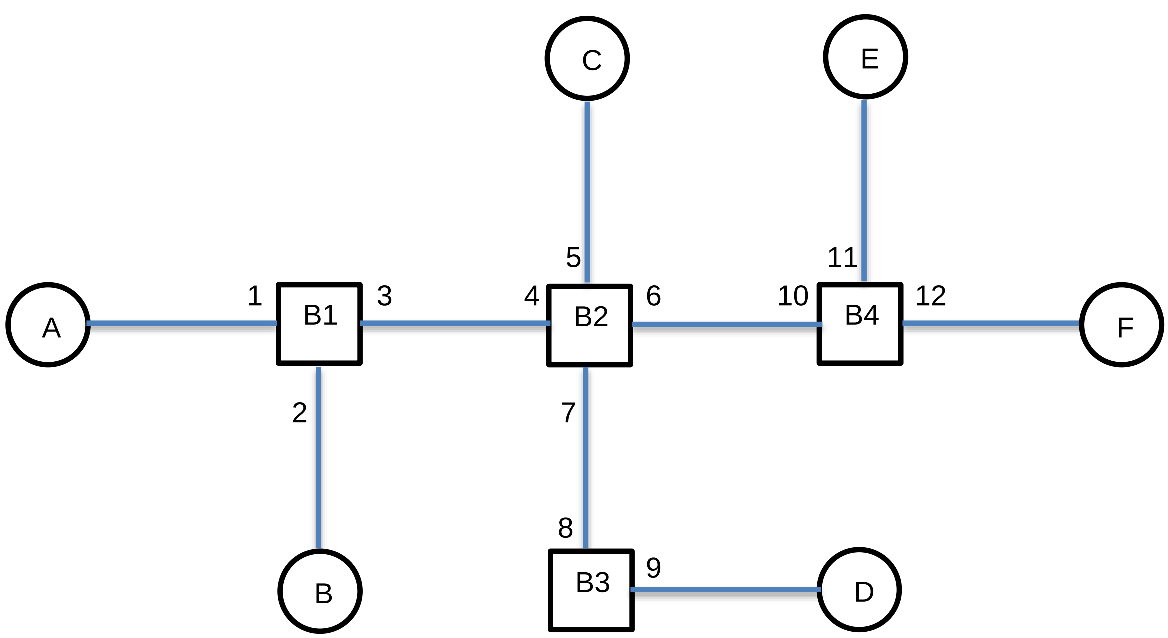
Instructions:

Turn in a **physical copy** at the beginning of the class on 10/26.

Problems:

1. **Learning Bridges (7 Points)**

Consider hosts A, B, C, D, E, F and learning bridges B1, B2, B3, B4, with their corresponding port numbers marked as shown. Assume the forwarding tables are empty for all the bridges. Also assume that entries added to each forwarding table do not have timeout.



a) If the following sequence of steps is followed, for each step, list all the **bridges** that receive each message. **(1 Point Each)**

i. Host A sends a message to Host B.
B1, B2, B3 and B4 (Each bridge floods the message)

ii. Host E sends a message to Host A.
B1, B2 and B4. (When B2 receives message from B4, it searches for "A" in its forwarding table and only forwards the message to B1)

iii. Host D sends a message to Host B.
B1, B2, B3 and B4. (No bridges know about B, so all bridges flood this message)

iv. Host F sends a message to Host D.
B2, B3 and B4. (When B2 receives message from B4, it searches for "D" in its forwarding table and only forwards the message to B3)

v. Host C sends a message to Host F.
B2 and B4. (When B2 receives message from C, it searches for "F" in its forwarding table and only forwards the message to B4)

b) Fill the forwarding table for bridge B1, after the above 5 messages (from part a) have been sent. **(2 Points; 1 Point if write at least one correct pair)**

B1 Forwarding Table

Host	Port
A	1
D	3
E	3

Also B2, B3 and B4's forwarding tables are listed below (Not Graded).

B2 Forwarding Table

Host	**Port**
A	4
C	5
D	7
E	6
F	6

B3 Forwarding Table

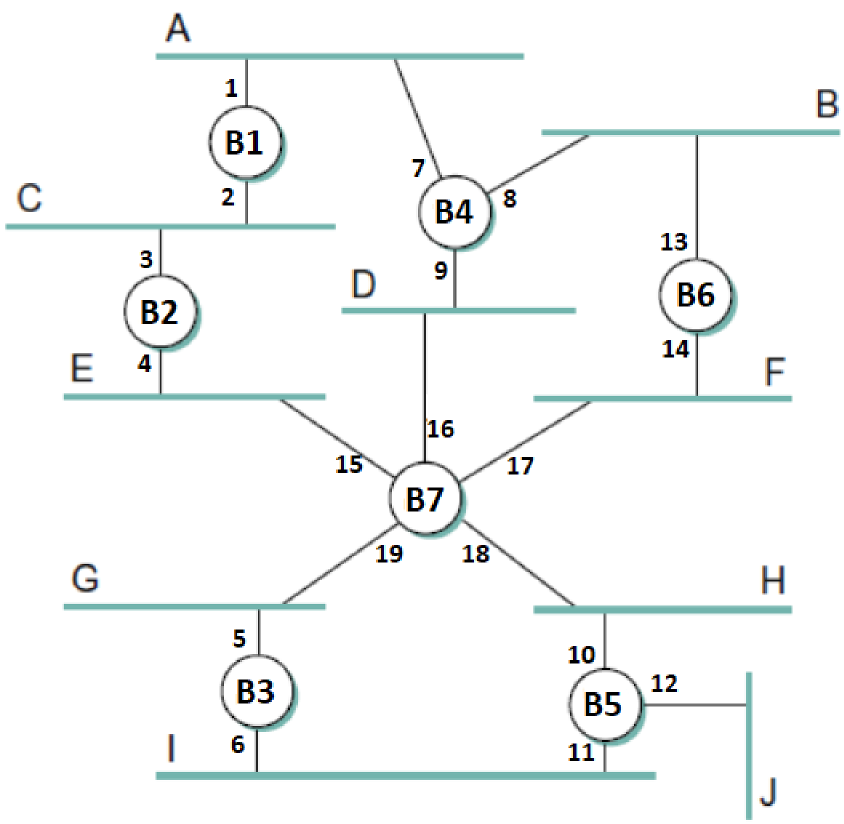
Host	**Port**
A	8
D	9
F	8

B4 Forwarding Table

Host	**Port**
A	10
C	10
D	10
E	11
F	12

2. Spanning Tree Protocol (6 Points)

Consider the extended LAN topology shown below, where B1-B7 represent bridges, with their corresponding ports numbered.



a) We denote a configuration message from node X in which it claims to be distance d from root node Y as (Y, d, X). For example, the first configuration message sent by bridge B7 is (B7, 0, B7). After the spanning tree algorithm converges, what are the configuration messages sent by bridge B2, B3 and B7, respectively? **(3 Points; 1 Point for each correct message)**

B2: (B1, 1, B2)
B3: (B1, 3, B3)
B7: (B1, 2, B7)

b) Indicate which ports are deactivated by the spanning tree algorithm. **(1 Points)**

11, 16 and 17.

c) Now assume bridge B1 does not participate in the spanning tree algorithm and **drops** all spanning tree algorithm messages.

(i) What is the new root bridge? **(1 Point)**
B2

(ii) Which ports are deactivated by the spanning tree algorithm? **(1 Point)**
11, 13 and 14. (Ports 1 and 2 should not be in the answer, but there are no penalties if included)

3. IP Fragmentation (7 Points)

Suppose a router receives an IP packet. Some fields of the IP header of this IP packet are shown as follow,

Length = 592	ID = 23	MF = 0	DF = 0	Offset = 0	TTL = 3

Now the router has to fragment this packet and forward the fragments across a network with an MTU of 300 bytes. Here, the MTU refers to the size of the largest packet that can be carried in a link-layer frame. If the size of the IP header is 20 bytes.

a) How many fragments will the router transmit? **(1 Point)**
3

b) For each fragment, list the values for the following fields in the IP headers: Length, ID, MF, DF, Offset and TTL. Note that "Offset" field indicates position of current fragment (in bytes/8). **(6 Points; 2 Points for each frame with all correct values; 1 Point for each frame that has partial correct values)**

IP header of Fragment 1:

Length = 300	ID = 23	MF = 1	DF = 0	Offset = 0	TTL = 2

IP header of Fragment 2:

Length = 300	ID = 23	MF = 1	DF = 0	Offset = 35	TTL = 2

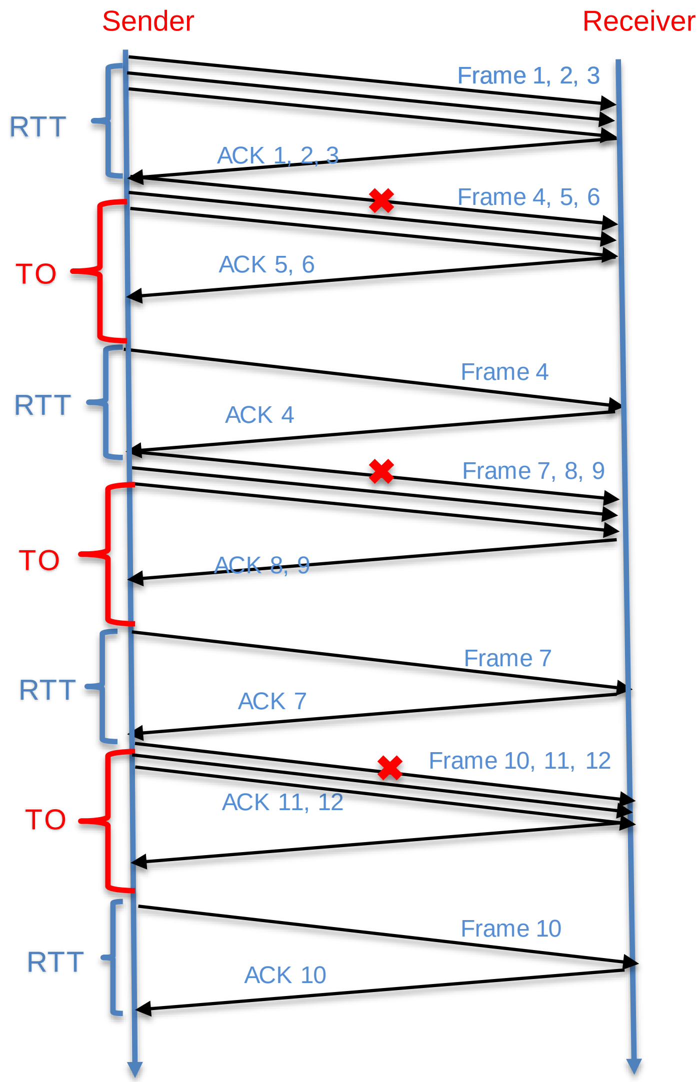
IP header of Fragment 3:

Length = 32	ID = 23	MF = 0	DF = 0	Offset = 70	TTL = 2

4. The Sliding Window Protocol (4 Points)
A sender and a receiver are transmitting data frames using the Selective Repeat Protocol. Assume Round Trip Time (RTT) is 6ms. The sender has a timeout of 8ms. The sender's window size is 3. The receiver drops every 4th transmission received (receiver will not send acknowledgement for that data frame); Sequence number is correctly encoded in all data and acknowledgement frames. The sender and receiver can transmit multiple frames at the same time. The sender transmits the first frame at time 0.

a) How many frames including duplicated ones, has the sender already transmitted when it receives an acknowledgement of the 10th frame from the receiver? **(2 Points)**
15 or 13 (Assume the sender transmits at most 10 different frames)

b) Compute the time (in ms) when the sender receives an acknowledgement of the 10th frame from the receiver. **(2 Points)**



T = 4 * RTT + 3 * TO = 48ms

5. IP Addressing (4 Points)

Consider the IP address 192.168.213.214. Suppose that class-based addressing is applied.

a) What type of networks would this IP address be? (2 Points)

   Class C

b) If the network administrator decides to break the network into 8 different subnets, what is the subnet number (address) of the subnet to which this IP address would be attached? (2 Points)

   192.168.213.192

6. Project 1 Debug (4 Points)

One of your classmates is working on project 1, and writing a function is_ack_in_window(uint8_t ACK, uint8_t LAR, uint8_t LFS). Here is the function that your classmate wrote.

```
// Returns true of ACK is inside of the sender's window. Return false otherwise.
static bool is_ack_in_window(uint8_t ACK, uint8_t LAR, uint8_t LFS)
{
	return LAR < ACK && ACK <= LFS;
}
```

This is a sender side function that checks whether the acknowledgement number (ACK) of a received frame is within the sender's sliding window. If it is not, the function should return false, and thus the sender knows to drop the frame. LAR is the acknowledgment number of the Last Acknowledgment Received (left bound of the sender's sliding window), and LFS is the sequence number of Last Frame Sent by the sender (right bound of the sender's sliding window). Sequence and acknowledgement numbers are both 8-bit long. The sender and the receiver have the same window size.

a) Which issue does this function fail to address? **(2 Points)**

Sequence number may wrap around, and thus LAR might be numerically larger than LFS. For example, assume that SWS = 7, LAR = 250, LFS = 1 ((uint8_t) (250+7) == 1), and a received ACK equals 252. In this case, ACK is a valid number since it is inside of the sliding window. However, the above function will return false.

b) Please rewrite the function to fix the problem in part a. **(2 Points)**

```
static bool is_ack_in_window(uint8_t ACK, uint8_t LAR, uint8_t LFS)
{
	uint8_t left = ACK – LAR;
	uint8_t right = LFS – LAR + 1;
	return left < right;
}
```