CSE 123: Computer Networks - Spring 2019 - Homework 2
(Due 04/29 @ 10pm on Gradescope)

Name: ______________________ PID: ______________

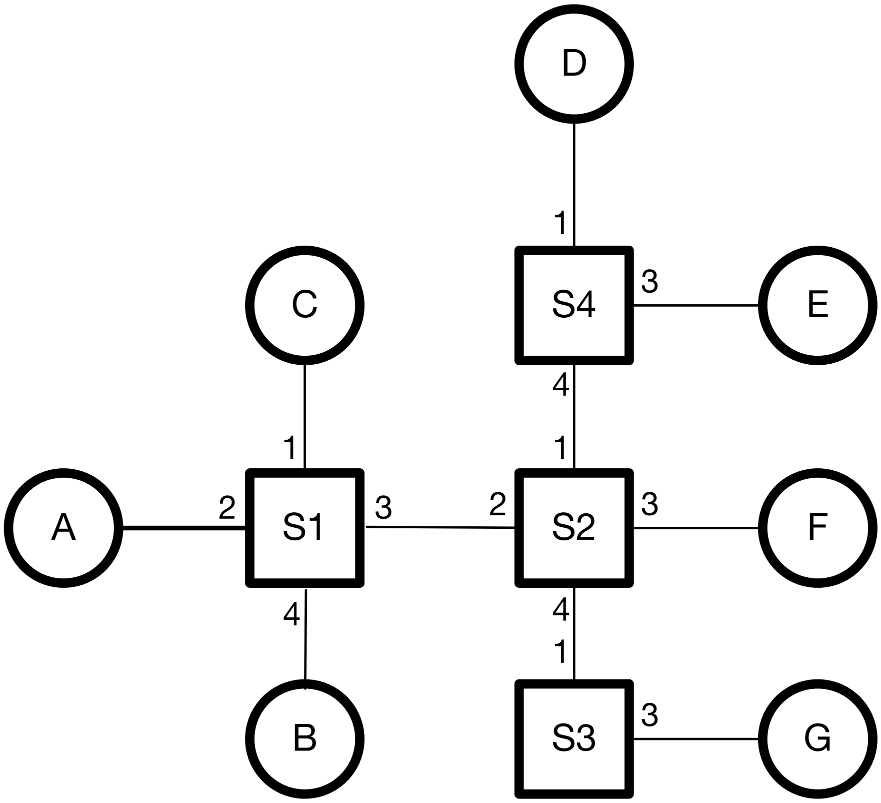
UCSD email: ______________________________

For the multiple choice questions, please fill in the circle for the correct answer. For the free-response questions, please write clearly and ensure that your answer fits in the box provided.

Turn in before 04/29 at 10pm on Gradescope.

Switching/Bridging

1. (14 points) Consider hosts A, B, C, D, E, F, G and learning switches/bridges S1, S2, S3, S4, with their corresponding port numbers marked as shown. Assume the switches were just powered on so their forwarding tables are empty. Also assume that entries added to each forwarding table do not have a timeout.



(a) (6 points) Given that hosts send packets to each other in the following order, list all of the hosts and switches that receive each packet.

1. `M1`: Host A sends a packet to Host B
   B, C, D, E, F, G, S1, S2, S3, S4
2. `M2`: Host A sends a packet to Host C
   B, C, D, E, F, G, S1, S2, S3, S4
3. `M3`: Host E sends a packet to Host A
   S4, S2, S1, A
4. `M4`: Host C sends a packet to Host A
   S1, A
5. `M5`: Host B sends a packet to Host C
   S1, C
6. `M6`: Host G sends a packet to Host E
   S3, S2, S4, E

(b) (6 points) Fill in the forwarding table for switch S2 as it will look after the above six packets have been sent (note: some entries may be unused).

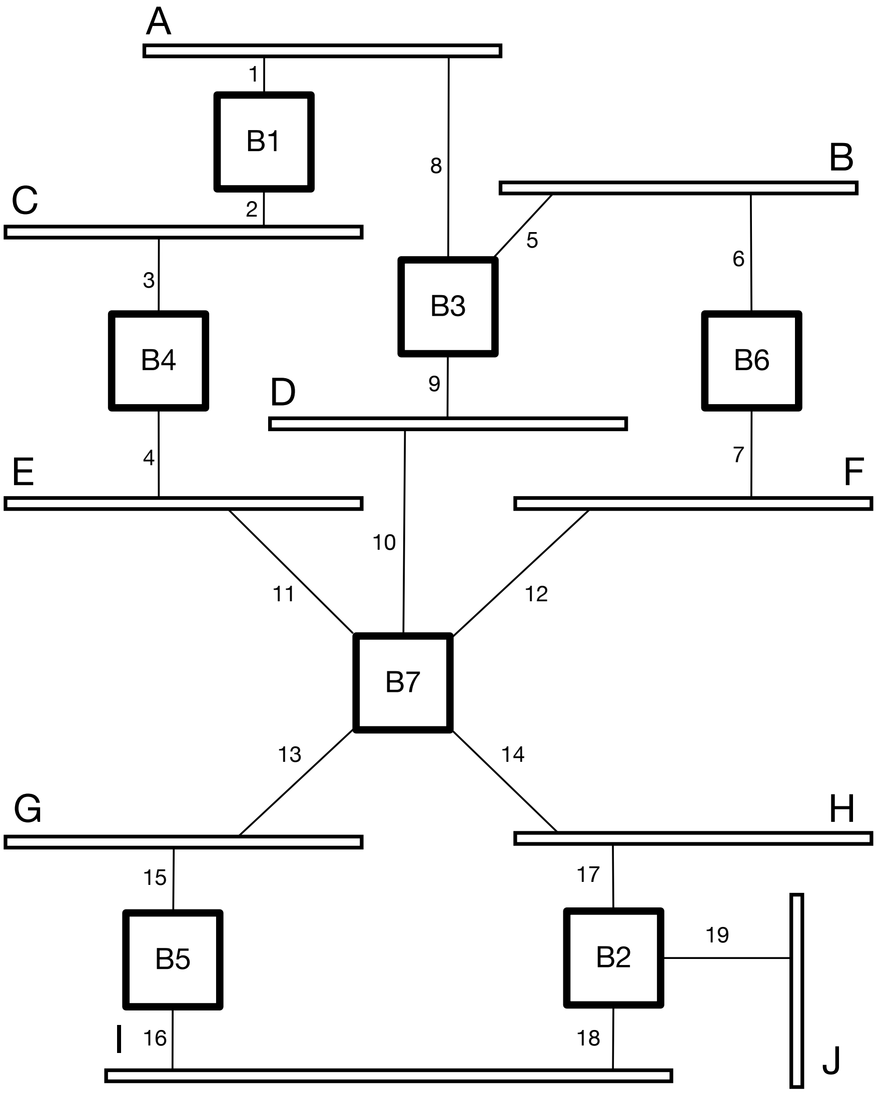
Host	**Port**
A	2
E	1
G	4

(c) (2 points) With this forwarding table, will switch S2 know the exact port to send a packet when it receives a packet from host A destined to any other host on the network? **Answer: False**

○ True
○ False

Spanning Tree Protocol

2. (15 points) Consider the LAN topology shown below, where B1-B8 are bridges with their corresponding links numbered.



(a) (5 points) After the spanning tree algorithm converges, what links will be deactivated?

- **B7 will disable port 11 because for LAN E, B4 is the lowest cost path to the root.**
- **B7 will disable port 12 because for LAN F, B6 is the lowest cost path to the root.**
- **B5 will disable port 16 because for LAN I, B2 is the lowest cost path to the root.**

(b) (6 points) After the spanning tree algorithm converges, what are the configuration messages sent by bridges B4, B5, and B7? Use the following notation for spanning tree messages from bridge X which claims to be distance d from root bridge Y: "(Y, d, X)". For example, the first message sent by bridge B7 is (B7, 0, B7).

- **(B1, 1, B4)**
- **(B1, 3, B5)**
- **(B1, 2, B7)**

(c) (4 points) Assume that some time after the spanning tree protocol converges, bridge B1 dies (lets out the magic smoke). What is the new root bridge? What links are deactivated by the spanning tree algorithm with this new root?

B2 will be the new root bridge because it has the lowest bridge number.

- **B7 will disable port 13 because for LAN G, B5 is the lowest cost path to the root.**
- **B6 will disable port 6 because for LAN B, B3 is the lowest cost path to the root.**

Links 1 and 2 will also be deactivated due to the dead B1, so it is OK to mention them.

IP Fragmentation

3. (10 points) Suppose a router receives an IP packet from a network with an MTU of 1,500 bytes. The packet has the following fields set in its header:

Length = 360	ID = 77	MF = 0	DF = 0	Offset = 0	TTL = 7

Now the router has to send this packet to a network with a smaller 300 byte MTU. Note that the MTU refers to the largest link-layer frame, the IP header always consumes 20 bytes of the frame.

(a) (5 points) How many fragments will the router transmit?

2

(b) (5 points) For each fragment, list the fields in their IP headers: Length, ID, MF, DF, offset, and TTL. Note that the "offset" indicates position in the current fragment (in bytes/8).

Length = 300	**ID = 77**	**MF = 1**	**DF = 0**	**Offset = 0**	**TTL = 6**
Length = 80	**ID = 77**	**MF = 0**	**DF = 0**	**Offset = 35**	**TTL = 6**

Project 1 Debug

4. (10 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) (5 points) Which issue does this function fail to address?

The 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) (5 points) Please rewrite the function to fix the problem in part (a).

```
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;
}
```