

# CSE 123: Computer Networks

## Homework 2

Out: 10/18, Due: 10/25

**Total Points: 47.5**

### 1. The Sliding Window Protocol

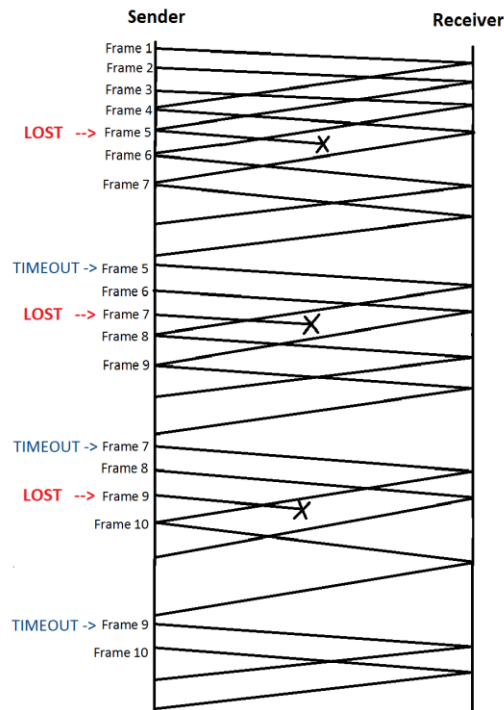
Assume that the sender's window size is 3. If we have to send 10 frames in total, and the channel of transmission is such that every 5th transmission is lost (but no ACKs), how many data transmissions (including both original transmissions and any necessary retransmissions) does the sender make in total if:-

- a. The go-back-N protocol is followed **(2 points)**
- b. The selective repeat protocol is followed **(2 points)**

Also, calculate the ratio of the effective number of frames sent to the total number of frames sent in each of the above cases. **(2 points – 1 point for each protocol)**

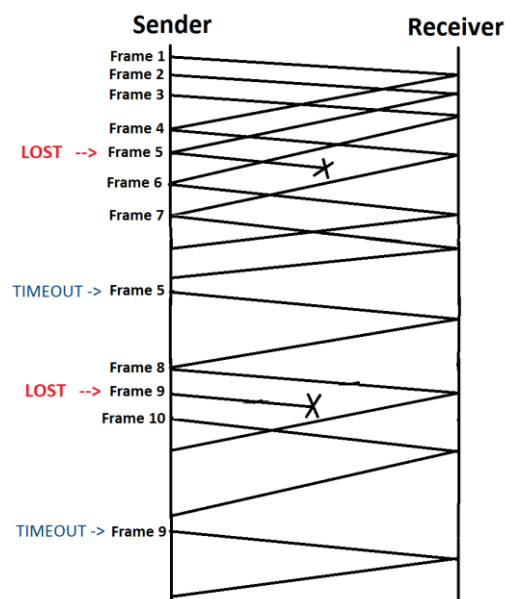
In this problem, for every 5th frame that is trying to be sent from the sender to receiver, an error in transmission occurs, and the frame is lost.

a. Given below is a diagram showing the sequence of transmissions from the sender to the receiver. The go-back-n protocol is followed, and every 5th frame transmitted is lost.



It can be seen that in order to transmit 10 frames of data from the sender to the receiver, 18 frames in total were sent. Hence the efficiency of the transmission is  $10/18 = 55.55\%$ .

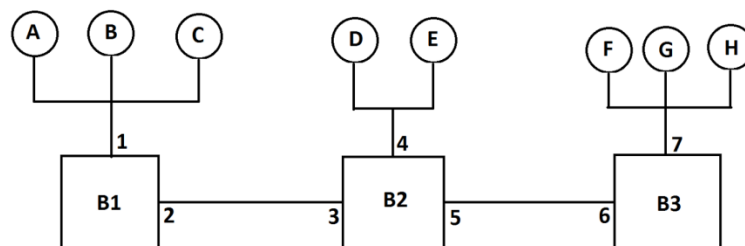
b. Given below is a diagram showing the sequence of transmissions from the sender to the receiver. The selective repeat protocol is followed, and every 5th frame being transmitted is lost.



It can be seen that in order to transmit 10 frames of data from the sender to the receiver, 12 frames in total were sent. Hence the efficiency of the transmission is  $10/12 = 83.33\%$ .

## 2. Learning Bridges

Consider the following topology, consisting of hosts A-H attached to learning bridges B1-B3, with their corresponding port numbers marked as shown.



If the following sequence of steps is followed, identify all the hosts that receive each message, assuming that the tables are empty for all bridges in the initial state.

- A sends a message to B (1 point)
- B sends a message to A (1 point)
- D sends a message to G (1 point)
- E sends a message to D (1 point)
- G sends a message to D (1 point)
- H sends a message to F (1 point)
- F sends a message to B (1 point)

Also, draw a simple forwarding table for each bridge after the above messages have been sent. (3 points – 1 point for each bridge)

- If A sends a message to B, all hosts receive it.
- If B sends a message to A, hosts A, B and C will receive it.
- If D sends a message to G, all hosts will receive it
- If E sends a message to D, hosts D and E will receive it
- If G sends a message to D, D, E, F, G, and H will receive it
- If H sends a message to F, all hosts will receive the message
- If F sends a message to B, all hosts will receive the message

The forwarding table maintained by each bridge will look like this:-

B1

Host	Port
A	1
B	1
D	2
F	2
H	2

B2

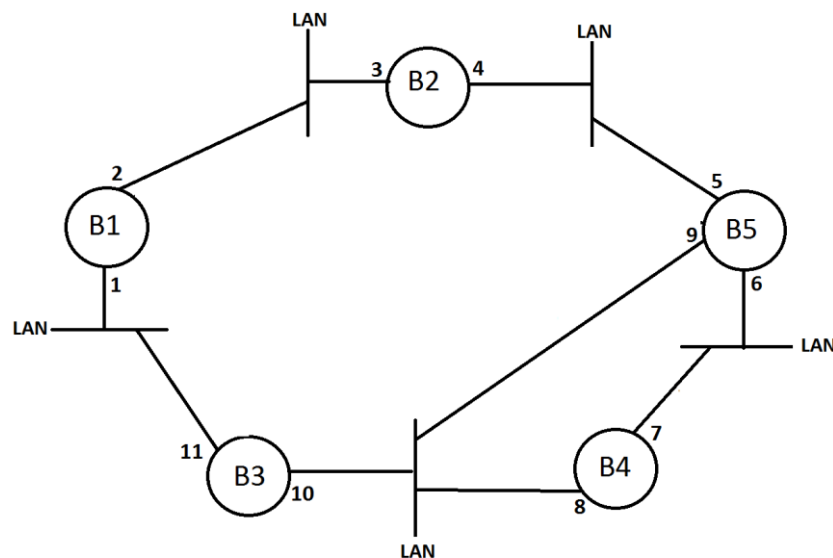
Host	Port
A	3
D	4
E	4
F	5
H	5
G	5

B3

Host	Port
A	6
D	6
F	7
G	7
H	7

### 3. The Spanning Tree Protocol

Consider the following network topology, where B1-B5 represent bridges, with their respective ports numbered.



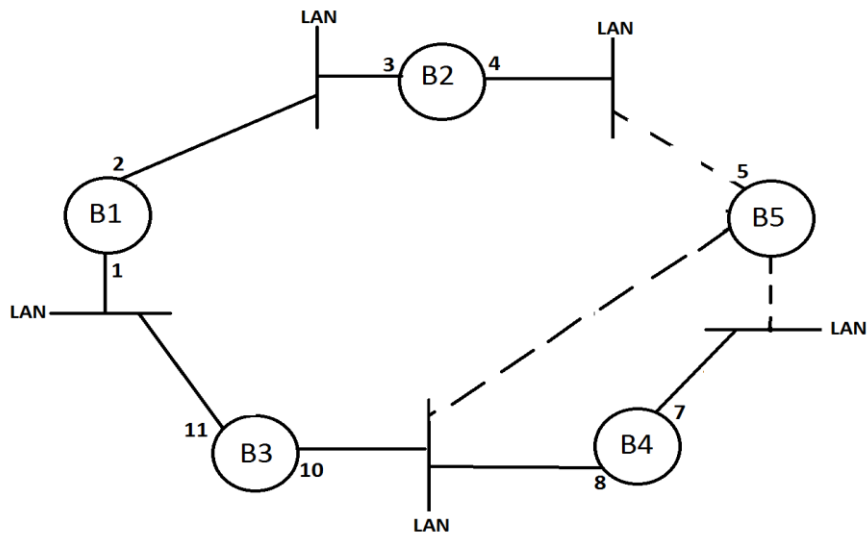
Using the spanning tree algorithm, find out the ports that will remain and the ports that will be turned off, and draw the resultant spanning tree. Provide an explanation for how the tree is

formed; including -- selecting the root node, sending configuration messages to its neighbors, and the cascading of messages between the bridges. Make sure to show the final configuration message each bridge will send, and explain which ports will be switched off. **(10 points - 1 point each for each bridge's configuration message, 1 point each for an explanation for the same)**

The following is an explanation of how the spanning tree is formed for this network:-

1. The spanning tree algorithm uses the bridge with the lowest ID as root. In this case, the root corresponds to B1. Therefore, all of B1's ports will be a part of the final spanning tree.
2. Each node sends periodic configuration messages to each other in the form of a tuple of order **(RootID, Distance to root, Bridge ID)**.  
Therefore, B1 sends the configuration message **(B1, 0, B1)** to its neighbors B2 and B3 through ports 1 and 2 respectively.
3. B2 and B3 receive the above configuration message. Even though these bridges may have believed themselves to be the root bridge prior to this, B1 will now become their root node as they have received a configuration message with a 'better' root (as it has a lower ID number). B2 adds 1 to the distance advertised by B1 and sends **(B1, 1, B2)** to its neighbor B5 through port 5. B3 does the same and sends **(B1, 1, B3)** to its neighbors B4 and B5 through port 10.
4. Since B5 receives messages from both B2 and B3, it will select B2 as its parent node as it has the lower ID number and will stop forwarding messages on ports 6 and 9 when it receives configuration messages from B4 and B3 respectively. Port 5 will switch off too, as the designated port for the LAN connected to B2 and B5 will be port 4 of bridge B2. The rest of the ports will continue to function, thereby removing the cycles in the graph.
5. The final configuration message each bridge will periodically send will be the following:-  
**B1 - (B1, 0, B1)**  
**B2 - (B1, 1, B2)**  
**B3 - (B1, 1, B3)**  
**B4 - (B1, 2, B4)**  
**B5 - (B1, 2, B5)**

Hence, the resulting spanning tree looks like this:-



## 4. Fragmentation

Suppose a router receives an IP packet of 552 bytes, and has to fragment the packet and forward the fragments across a network with an MTU of 300 bytes. Then, a subsequent router has to further forward the packet (and/or any resulting fragments) onto another network that has an MTU of 100 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 TCP header is 20 bytes and that of the IP header is also 20 bytes (i.e., there are no options), compute the values for the following fields in the IP headers for all of the fragments that traverse each network:-

- Length (5 points – 0.5 points for each of the 10 fragments)
- MF (2.5 points - 0.25 points for each of the 10 fragments)
- Offset (5 points – 0.5 points for each of the 10 fragments)

Consider the first network. Since the MTU is the size of the largest IP packet that can be carried, which is 300B, the maximum allowable payload is 280B.

Now, the payload that we need to send is  $552\text{B} - 20\text{B} = 532\text{B}$ . Hence, the message is fragmented into two - one of size  $280\text{B} + 20\text{B} = 300\text{B}$  and the other of size  $(532\text{B} - 280\text{B} = 252\text{B}) + 20\text{B} = 272\text{B}$  (because each fragment needs its own header). The fields in the IP header for the first packet are:-

- Length = 300  
MF = 1  
Offset = 0

And that for the second packet are:-

- Length = 272  
MF = 0  
Offset = 35

Offset has a value of 35 since the 280B offset is divided by 8 to fit into a field of 13 bits instead of 16.

Now, consider the second network, wherein the MTU is only a 100B. Hence, the maximum amount of payload data will be  $100\text{B} - 20\text{B} = 80\text{B}$ . Therefore, the fragment of size 300B, that contains 280B payload

data, will be further fragmented into fragments of size 100B, 100B, 100B, and 60B. The corresponding field values are as follows:-

- a.   Length = 100  
      MF = 1  
      Offset = 0
- b.   Length = 100  
      MF = 1  
      Offset = 10
- c.   Length = 100  
      MF = 1  
      Offset = 20
- d.   Length = 60  
      MF = 1 (as the MF bit of the fragment was originally 1)  
      Offset = 30

The second fragment of size 272B will be further fragmented into fragments of size 100B, 100B, 100B, and 32B. The corresponding field values are as follows:-

- a.   Length = 100  
      MF = 1  
      Offset = 35 (as the offset of the fragment was originally 35)
- b.   Length = 100  
      MF = 1  
      Offset = 45
- c.   Length = 100  
      MF = 1  
      Offset = 55
- d.   Length = 32  
      MF = 0  
      Offset = 65

## 5. Sequence Number Wraparound

Assuming that the Transmission Control Protocol (TCP) operates over a 2 Gbps link:-

- a. If TCP could utilize the full bandwidth of the link, assume that it chooses to send segments of 20B of data each, back-to back without any delay whatsoever. Also assume that there is no error in the channel, and there is no retransmission of packets. Taking into account that the size of the TCP header is 20 bytes, and ignoring other packet/frame headers or any other overhead from the network and link layers, how long would it take for the TCP sequence space to wrap around completely? **(5 points - 3 points for effective bandwidth utilization, 2 points for final answer)**
- b. If the maximum segment lifetime (MSL) is 2 minutes for the network, is there a potential for TCP to operate incorrectly? Why? **(4 points - 1 point for answer, 3 points for explanation)**

a. The given bandwidth is 2 Gbps. The TCP includes a header of 20B with the 20B payload, making the total message size 40B. Hence, the fraction of effective bandwidth being utilized is  $20/40 = 1/2$ . Therefore the effective bandwidth is

$$(1 / 2) * 2 \text{ Gbps} = 1 \text{ Gbps}$$

Now, since the sequence space for TCP is 32 bits, the total time it takes for the sequence numbers to completely wraparound is

$$\frac{2^{32} * 8 \text{ bits}}{1 \text{ Gbps}} = 34.36 \text{ seconds}$$

b. In order for data transmission across a network to occur without faults, the wraparound time for the TCP sequence number must exceed the MSL. Since the calculated wraparound time  $34.36\text{s} < 120\text{s}$ , there could be scenarios at the receiver end where multiple packets of the same sequence number could arrive. This means that the receiver will only select one of the packets and discard the rest of the seemingly indistinct packets, although in reality, all the packets have different payload data, making them unique.

TCP overcomes this by adding a unique 32-bit timestamp to extend the sequence space. A higher timestamp for a packet will indicate that it was sent by the sender more recently, and a lower timestamp will indicate that it was sent less recently; thereby making packets with the same sequence number unique to the receiver.