Jason Lubrano - Problem Set #2

Due Tuesday Oct 8th before the class

**Problem 1 (6pts)**

Explain collision domain and broadcast domain with respect to a hub, switch, and a router.

- Hub – A hub creates and expands the collision domain and broadcast domain. A hub operates at physical layer and is very dumb. When one node uses a hub, all of the other nodes connected to the hub must wait. The hub gets a signal from one node and forwards it to all the other nodes. Collision occurs when two nodes sends signals at the same time; therefore a hub creates a collision domain. If the collision domain gets too large, there are major traffic problems. Daisy chaining hubs expands the collision domain.
- Switch – A switch separates collision domain but creates broadcast domain. Traditional switches operate at the data link layer. A Switch is much smarter than a hub because it gives every conversation the full bandwidth of the network. When node A is talking, only it's receiver B can hear. And when node C is talking, only it's receiver D can hear. Conversations become one-to-one. Both A and C can talk at the same time and no collision can occur. That is how switch separates the collision domain. A switch keeps track of a source address table to keep track of connections between nodes. The broadcast domain is a group of nodes that can reach each other by a broadcast. If node A sends the broadcast, then node B, C, and D can hear it.
- Router – A router separates both a collision domain and broadcast domain. A router operates at the network layer. Routers only forward IP packets based on the IP address. Any conversations and broadcasts between multiple LANs connected to a router will not be heard by other LANs. If two LANs are connected through a router, LAN-A will not hear conversations or broadcast messages from LAN-B, and vice versa.

**Problem 2 (5pts)**

Consider the following networked computers connected by Bridge X and Y. Bridge X has interface 1,2 and3. Bridge Y has interface 1 and 2. Assume at the beginning the address tables of Bridge X and Y are all empty. Write down the address tables of Bridge X and Y after the following communication finished.

1. A send a packet to C
2. B send a packet to D
3. C send a packet to E

4. E send a packet to A
5. D send a packet to A

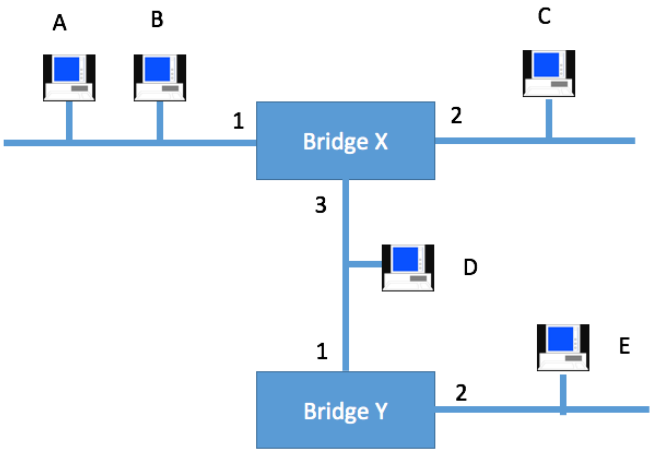



Figure 1

| Bridge X | |
|---|---|
| Address | Interface |
| C | 2 |
| D | 3 |
| E | 3 |
| A | 1 |

| Bridge Y | |
|---|---|
| Address | Interface |
| E | 2 |
| A | 1 |

**Problem 3 (5pts)**

Given the extended LAN shown in Figure 2, indicate which ports are not selected by the spanning tree algorithm. Note that the bridge with the smallest ID becomes a root.

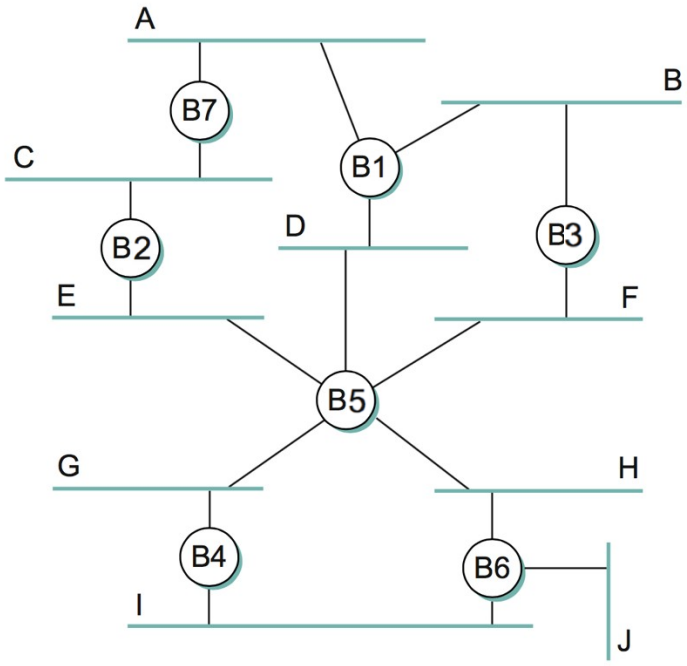
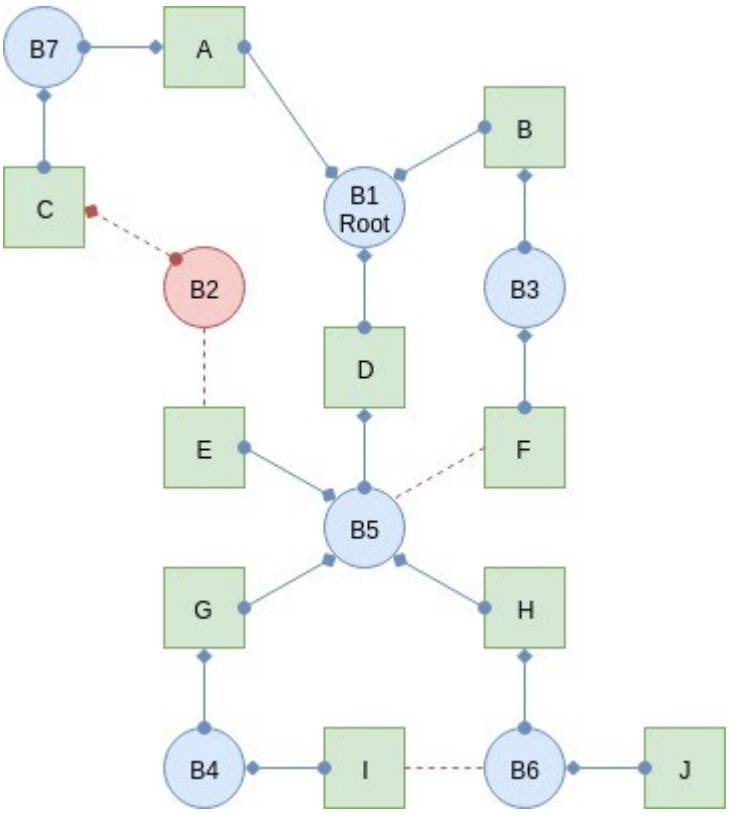
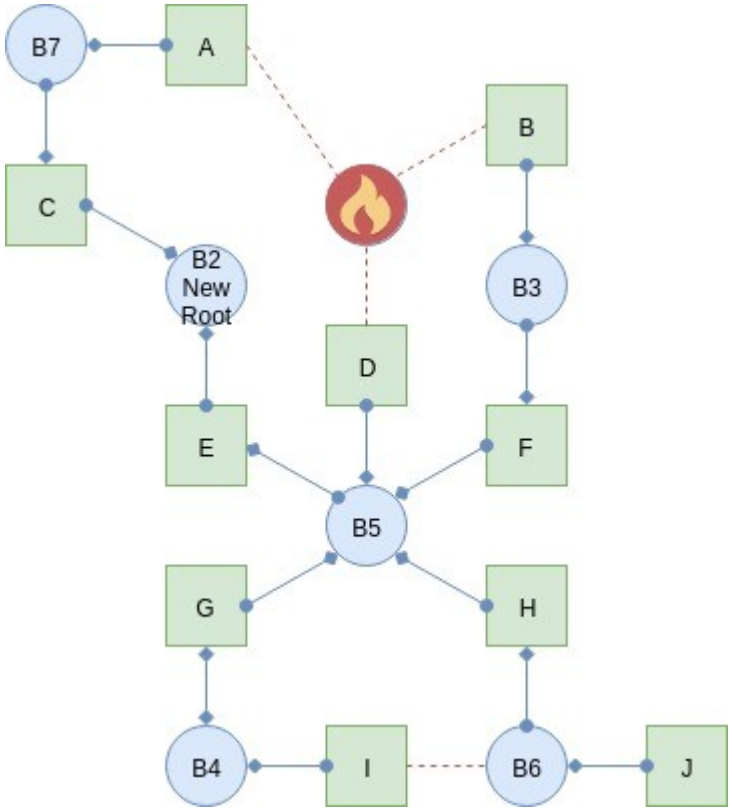

A
B
B7
B1
C
D
B3
B2
E
F
B5
G
H
B4
B6
I
J

Figure 2


B7
A
B
C
B1
Root
B2
B3
D
E
F
B5
G
H
B4
I
B6
J

**Problem 4 (5pts)**

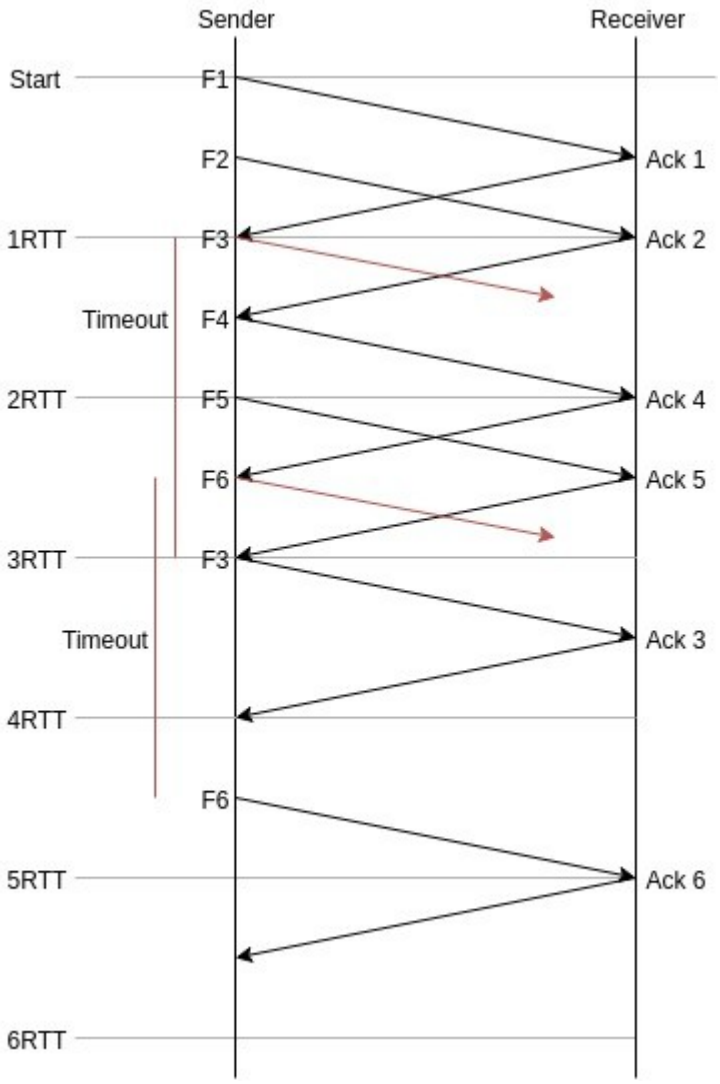
Still considering Figure 2. If Bridge B1 suffers catastrophic failure. Again indicate which ports are not selected by the spanning tree algorithm.




**Problem 5 (6pts)**

Draw a time line diagram for the sliding window algorithm with SWS = RWS = 3 frames, for the following situations. Use a timeout interval of about 2 × RTT. And assume 2 frames must be send ½ RTT apart which means if everything is normal Sender will receive ACK and then immediately send the next frame.

Frames 3 and 6 are lost on their first transmissions. Draw the algorithm with time line diagram till Frame 6 is sent. (6pts)




**Problem 6 (6pts)**

Consider the GBN protocol with a sender window size of N=4 and a sequence number range of 1,024. Suppose that at time t, the next in-order packet that the receiver is expecting has a sequence number of k. Assume that the medium does not reorder messages. Answer the following questions:

(a) What are the possible sets of sequence numbers inside the sender's window at time t? Justify your answer. (2 pts)

The window is size of N=4. Assume the receiver has received packets [k, k+N-1], [k, k+3] and sent the ACK for each of these to the sender, meaning all but the last packet in the window has an ACK followup. This is the last frame before the sender moves on to the next window. Consider the case where the sender has sent all of the packets but has not received any ACKs. The window of the sender in this case the bound is where he started minus the window to

the number before. In mathematical notation [k-N, k-1] or [k-4, k-1] because it did not receive any ACKs so the window can not move on at all. We can take the first two bounds of where the window is the lowest (no ACKs) to where it is the largest (All but one ACK) to find the possible starting sequence- [k-N, k] or [k-4, k]. Any other scenario will put the sender in this sequence.

(b) What are all possible values of the ACK field in all possible messages currently propagating back to the sender at time t? Justify your answer. (2 pts)

Starting off with a similar scenario as (a), we can assume the receiver has received all messages except for message k. Receiving message k makes the window change to the beginning. If the receiver has not received message k, then it has received number N-1 out of N messages for that window, which is messages k-1. We are at [k-N, k-1], or [k-4, k-1] messages received and an ACK sent out for them. For the first message of a window the receiver must send an ACK for is message k-N-1 = k-5. When the ACK for k-5 is sent the receiver will get the next message from the next window, k-1, and the window will move past it. The receiver's sequence will exist between [k-5, k-1].

(c) With the Go-Back-N protocol, is it possible for the sender to receive an ACK for a packet that falls outside of its current window? Justify your answer with an example. (2 pts)

Yes. When the sender's timeout is shorter than the RTT, it can receive messages while being in the next frame. With a window size of 4 the sender sends all four packets at the start. Shortly after, the receiver sends the ACKs for all four messages. Before the ACK messages reach the sender, the timeout happens and the sender re-sends packets 1-4. The ACKs from the first set of messages reach the sender and the sender now sends messages 5-9. At that time, the receiver has received the second set of 1-4 messages, and sends the ACK for those again. The ACK for messages 1-4 hits the sender when the window is for message 5-9.

**Problem 7 (10 points)**

(a) Is 10.72.0.255/255.255.254.0 a valid IP address for a host? **[2pts]**

255.255.254.0 = 23 bit mask = 510 hosts (512-2); This is a valid address since the Network IP is 10.72.0.0, and the broadcast would set every bit after 23 to 1 so the broadcast would be 10.72.1.255.

(b) Divide the 10.72.0.0/16 subnets into five large networks of 8192 IPs each, 8 medium-sized networks of 2048 IPs each, and 10 small sized networks of 128 IPs each. **[6pts]**

/16 = 255.255.0.0. 8192 IPs mean our mask is 1...1.1...1.11100000.0 and we need to borrow 3 bits. Now our mask is 255.255.224.0, or /19.
First Network 10.72.0.0 /19
Second Network [10.72.001]00000.0 /19 → 10.72.32.0 /19
Third Network [10.72.010]00000.0 /19 → 10.72.64.0 /19

Fourth Network [10.72.011]00000.0 /19 → 10.72.96.0 /19
Fifth Network [10.72.100]00000.0 /19 → 10.72.128.0 /19

/16 = 255.255.0.0 = 1...1.1...1.0.0. 2048 IPs mean our mask will be 255.255.111110000.0 → 255.255.248.0, or /21. Continuing on from the previous 10.72.128.0 with a /21 mask:
First Network [10.72.10100]000.0 /21 → 10.72.160.0 /21
Second Network [10.72.10101]000.0 /21 → 10.72.168.0 /21
Third Network [10.72.10110]000.0 /21 → 10.72.176.0 /21
Fourth Network [10.72.10111]000.0 /21 → 10.72.184.0 /21
Fifth Network [10.72.11000]000.0 /21 → 10.72.192.0 /21
Sixth Network [10.72.11001]000.0 /21 → 10.72.200.0 /21
Seventh Network [10.72.11010]000.0 /21 → 10.72.208.0 /21
Eight Network [10.72.11011]000.0 /21 → 10.72.216.0 /21

/16 = 255.255.0.0. 128 IPs mean the mask will be 255.255.255.128, or /25. Same methodology as before:
First Network [10.72.11100000.0]0000000 /25 → 10.72.224.0 /25
Second Network [10.72.11100000.1]0000000 /25 → 10.72.224.128 /25
Third Network [10.72.11100001.0]0000000 /25 → 10.72.225.0 /25
Fourth Network [10.72.11100001.1]0000000 /25 → 10.72.225.128 /25
Fifth Network [10.72.11100010.0]0000000 /25 → 10.72.226.0 /25
Sixth Network [10.72.11100010.1]0000000 /25 → 10.72.226.128 /25
Seventh Network [10.72.11100011.0]0000000 /25 → 10.72.227.0 /25
Eighth Network [10.72.11100011.1]0000000 /25 → 10.72.227.128 /25
Ninth Network [10.72.11100100.0]0000000 /25 → 10.72.228.0 /25
Tenth Network [10.72.11100100.1]0000000 /25 → 10.72.228.128 /25

(c) Is 192.168.2/23 and 192.168.3/23 representing the same subnet? Please justify your answer. **[2pts]**

Yes. [x.x.0000001]0 /23 and [x.x.0000001]1 /23 both belong to the same subnet.

**Problem 8 (8 points)**

An organization has been assigned the prefix 192.168.1.0/23 and wants to form subnets for 4 departments which have the following number of hosts:

Department A: 130 hosts

Department B: 120 hosts

Department C: 60 hosts

Department D: 31 hosts

(a) Give a possible arrangement of subnet masks to make this possible. **[5pts]**

Dept A:
[192.168.00000001].00000000 → 192.168.1.0 /24
/24 = 255.255.255.0

Dept B:
[192.168.00000001.0]0000000 → 192.168.1.0 /25
/25 = 255.255.255.128
Dept C:
[192.168.00000001.10]000000 → 192.168.1.128 /26
/26 = 255.255.255.192
Dept D:
[192.168.00000001.110]00000 → 192.168.1.192 /27
/27 = 255.255.255.224

| Dep. | Network | Subnet mask |
|---|---|---|
| A | 192.168.1.0 /24 | 255.255.255.0 |
| B | 192.168.1.0 /25 | 255.255.255.128 |
| C | 192.168.1.128 /26 | 255.255.255.192 |
| D | 192.168.1.192 /27 | 255.255.255.224 |

(b) Suggest what the organization might do if department C grows to 65 hosts. **[3pts]**

Since department C will be growing to 65 hosts, then it will need more space since the current subnet will not be able to hold all the IPs. Since department B only needs 120 IPs, it can use the extra ones from that subnet.

Dept A:
[192.168.00000001].00000000 → 192.168.1.0 /24
/24 = 255.255.255.0
Dept B:
[192.168.00000001.0]0000000 → 192.168.1.0 /25
/25 = 255.255.255.128
Dept C1:
[192.168.00000001.1]1111011 → 192.168.1.251 /25
/26 = 255.255.255.128
Dept C2:
[192.168.00000001.10]000000 → 192.168.1.128 /26
/26 = 255.255.255.192
Dept D:
[192.168.00000001.110]00000 → 192.168.1.192 /27
/27 = 255.255.255.224

| Dep. | Network | Subnet mask |
|---|---|---|
| A | 192.168.1.0 /24 | 255.255.255.0 |
| B | 192.168.1.0 /25 | 255.255.255.128 |
| C1 | 192.168.1.251 /25 | 255.255.255.128 |
| C2 | 192.168.1.128 /26 | 255.255.255.192 |

| | | |
|---|---|---|
| D | 192.168.1.192 /27 | 255.255.255.224 |

**Problem 9 (12 points)**

For the network given below in Figure 3, give global distance-vector tables for each node when:

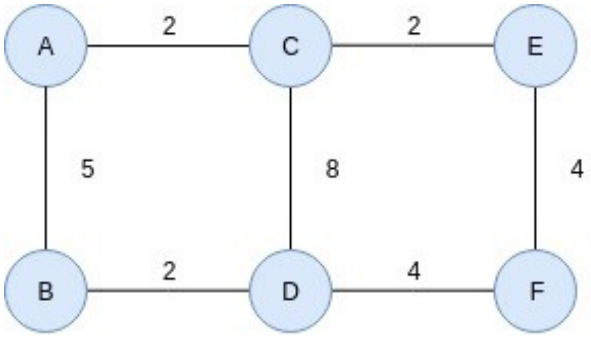



Figure 3

(a) Each node knows only the distance of its immediate neighbors. **[4pts]**

| Node | A | B | C | D | E | F |
|---|---|---|---|---|---|---|
| A | 0 | 5 | 2 | Inf. | Inf. | Inf. |
| B | 5 | 0 | Inf. | 2 | Inf. | Inf. |
| C | 2 | Inf. | 0 | 8 | 2 | Inf. |
| D | Inf. | 2 | 8 | 0 | Inf. | 4 |
| E | Inf. | Inf. | 2 | Inf. | 0 | 4 |
| F | Inf. | Inf. | Inf. | 4 | 4 | 0 |

(b) Each node has reported the information it had in the first step to its immediate neighbors. **[4pts]**

| Node | A | B | C | D | E | F |
|---|---|---|---|---|---|---|
| A | 0 | 5 | 2 | 7 | 4 | Inf. |
| B | 5 | 0 | 7 | 2 | Inf. | 6 |
| C | 2 | 7 | 0 | 8 | 2 | 6 |

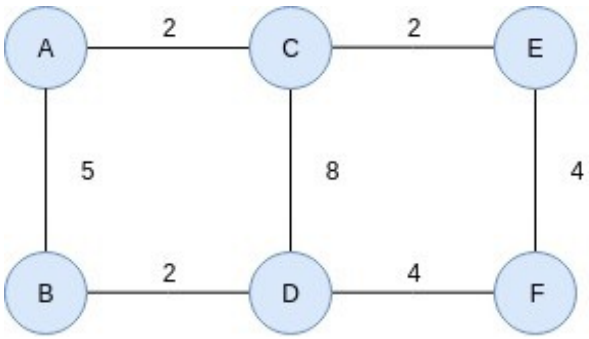
| | | | | | | |
|---|---|---|---|---|---|---|
| D | 7 | 2 | 8 | 0 | 8 | 4 |
| E | 4 | Inf. | 2 | 8 | 0 | 4 |
| F | Inf. | 6 | 6 | 4 | 4 | 0 |

(c) Repeat step (b) one more time. **[4pts]**

| Node | A | B | C | D | E | F |
|---|---|---|---|---|---|---|
| A | 0 | 5 | 2 | 7 | 4 | 8 |
| B | 5 | 0 | 7 | 2 | 9 | 6 |
| C | 2 | 7 | 0 | 8 | 2 | 6 |
| D | 7 | 2 | 8 | 0 | 8 | 4 |
| E | 4 | 9 | 2 | 8 | 0 | 4 |
| F | 8 | 6 | 6 | 4 | 4 | 0 |

**Problem 10 (8 points)**

Again for the network graph in Figure. 3. Show how the link-state algorithm builds the routing table for node D.




(a) Show the detailed steps with the link-state algorithm. **[5pts]**

Link state algorithm is split between two phases. The first phase is reliable flooding. In the beginning, each node only knows the cost to its neighbor. At the end of reliable flooding, each node knows the entire graph. Phase 2 is route calculation. Each node will use Dijkstra's algorithm on the graph to calculate optimal routes to all nodes. Node D will tell all the other nodes in the topology that it is connected to B, C, and F with their respective weights.

(b) Show the final routing table of node D. **[3pts]**

| Step | Confirmed | Info coming in |
|---|---|---|
| 0 | (D,0,-) | |
| 1 | (D,0,-) | (B,2,B), (C,8,C), (F,4,F) |
| 2 | (D,0,-), (B,2,B), (C,8,C), (F,4,F) | (A,7,B), (E,8,F) |
| 3 | (D,0,-), (B,2,B), (C,8,C), (F,4,F), (A,7,B), (E,8,F) | |

**Problem 11 (6 points)**

The network graph is shown in Figure. 4.

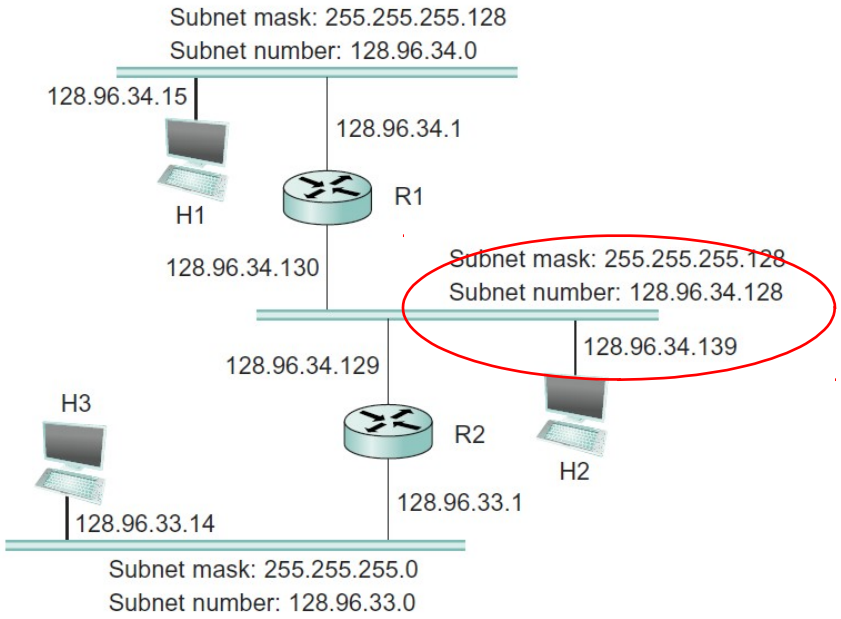



Figure 4

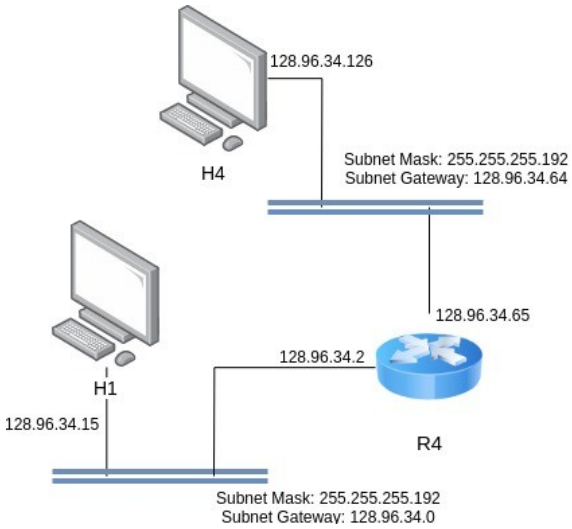
(a) Host H1 sends a packet to the destination 128.96.34.126. Explain how this packet traverses in the network described below. You need to describe who received the packet and what are their reactions. **[2pts]**

The destination 128.96.34.126 belongs to the same subnet as H1. When H1 sends the packet, all of the host will receive and drop the packet, except the intended receiver. The MAC address mismatching tells all other hosts to drop the packet. When the MAC address matches with the intended host, the message gets processed. R1 will not forward any packet since it is on the same subnet mask.

(b) Host H3 sends a packet to the destination 128.96.34.250. Explain how this packet traverses in the network. **[2pts]**

The destination 128.96.34.250 is not on the same subnet as H3. The packet gets sent to R2. R2 then forwards that packet to subnet 128.96.34.128. Similar to before, the packet gets sent to all host on that subnet. Only the intended host will receive and process the messages. All of the other hosts will drop the packet because the MAC addresses don't match up.

(c) The subnet of H1 has now two different teams and would like to split it into two subnets. Please add one more subnet and add R3 and change the network configurations as you need. Note that you are allowed to modify the network as least disruptive as possible. **[2pts]**




* Leads to R1 and everything is the same as the figure above.

**Problem 12 (8 points)**

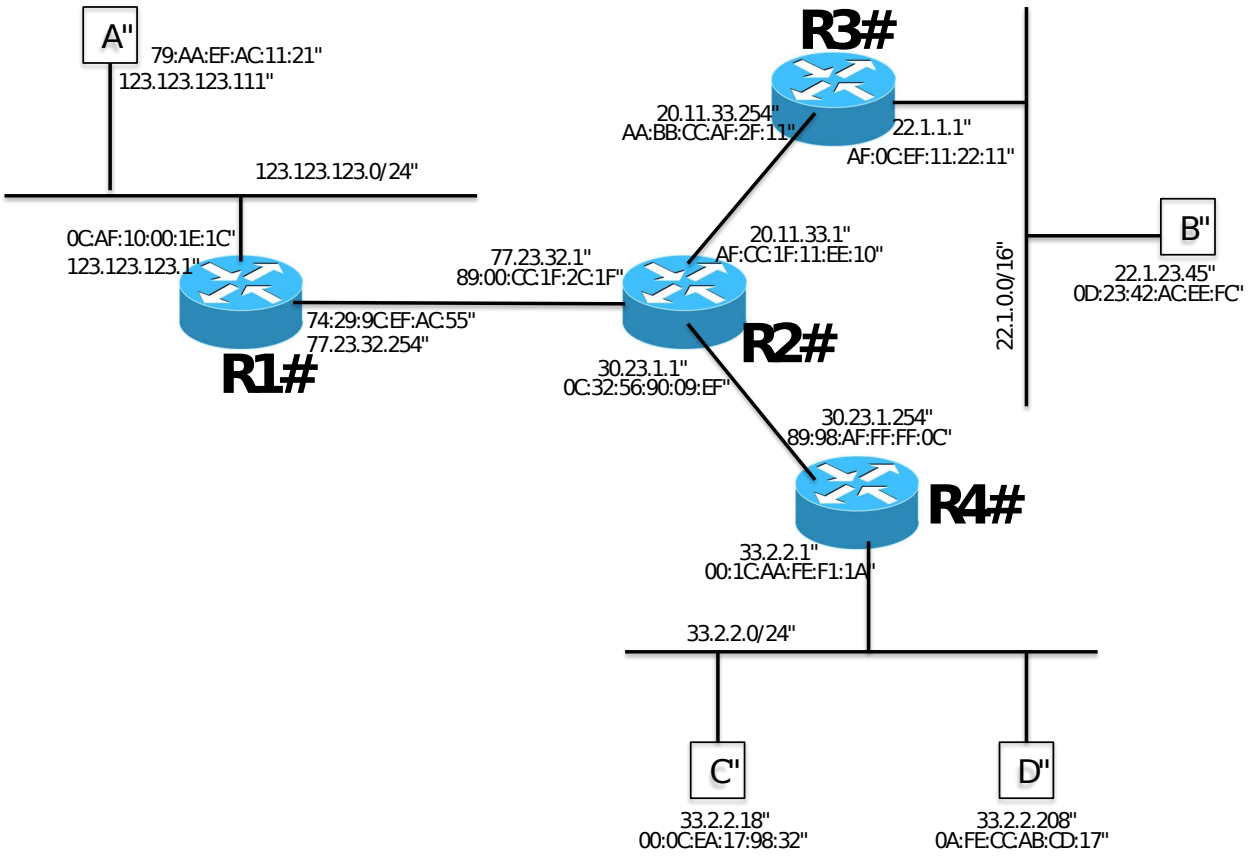



Figure. 5

Above in Figure 5 is the network graph with 4 routers (R1, R2, R3, R4) and 4 hosts (A, B, C, D). Each router interfaces and hosts are labeled with both IP and MAC address, Routing is enabled so that any two hosts can communicate with each other and also the default gateway of each host is set to its gateway router.

(a) Suppose that B send an IP packet to C through R3, R2, R4. Write down the IP packet's content (src MAC, dst MAC, src IP, dst IP) along the path in the Table given below: **[4pts]**

| | Src MAC | Dst MAC | src IP | dst IP |
|---|---|---|---|---|
| B → R3 | 0D:23:42:AC:EE:FC | AF:0C:EF:11:22:11 | 22.1.23.45 | 32.2.2.18 |
| R3 → R2 | AA:BB:CC:AF:2F:11 | AF:CC:1F:11:EE:10 | 22.1.23.45 | 32.2.2.18 |
| R2 → R4 | 0C:32:56:90:09:EF | 89:98:AF:FF:FF:0C | 22.1.23.45 | 32.2.2.18 |
| R4 → C | 00:1C:AA:FE:F1:1A | 00:0C:EA:17:98:32 | 22.1.23.45 | 32.2.2.18 |

(b) When A sends out an ARP query for its default gateway, what is the reply to that query? **[2pts]**

When sending a packet to any machine on the public internet, the packet is sent to the MAC address of the router interface that is the default gateway. In this case, the response to A's ARP query would be 0C:AF:10:00:1E:1C

(c) Suppose the routers use link-state routing protocol, what will be R3's routing table entries? **[2pts]**

| Step | Confirmed | Info coming in |
|---|---|---|
| 0 | (R3,-) | |
| 1 | (R3,-) | (R2,R2), (B,B) |
| 2 | (R3,-), (R2,R2), (B,B) | (R1,R2), (R4,R2) |
| 3 | (R3,-), (R2,R2), (B,B), (R1,R2), (R4,R2) | (A,R1), (C,R4), (D,R4) |
| 4 | (R3,-), (R2,R2), (B,B), (R1,R2), (R4,R2), (A,R1), (C,R4), (D,R4) | |

**Problem 13 (6 points)**

Suppose a computer just boot up, connected to wireless network and successfully obtained IP, gateway and DNS address. Now it wants to access www.yahoo.com from its browser. Describe the sequence of packets exchanged to and from this computer until the webpage starts to load. (include what kind of protocol is used and what is the content of the packets)

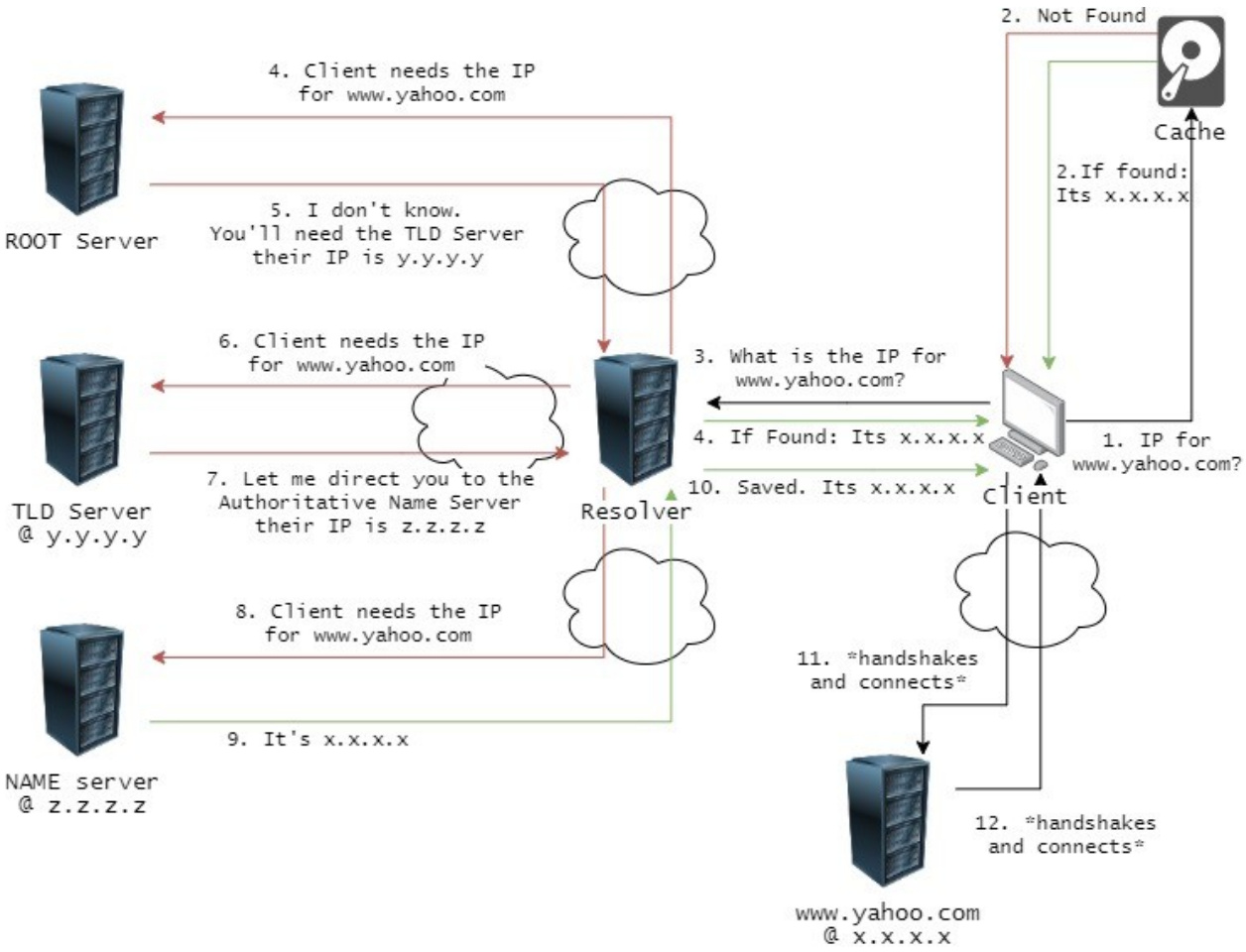
DNS stands for Domain Name System. This process resolves names to numbers; in other words, it resolves www.yahoo.com to a number, the IP address, because the only thing computers know are numbers. If the user already knew the IP address to yahoo.com, then they can type in the IP address. A DNS server works very similarly to a phonebook. It searches the domain and replies back with the specified number, just like how people search for a name rather than a phone number first.

Once the user has the IP address, the computer can retrieve the webpage. There are a couple of different routes to resolve a webpage, from resolving it to the local computer to several steps requiring connection to the root server and down.

The process starts when the client types www.yahoo.com into their browser and hits enter. The computer tries to find the IP address for that domain in its cache memory. If the OS can't find it, then the request is sent to the resolver server owned by the client's ISP. The resolver server searches its database for yahoo.com. If the resolver server can't find the IP address, then the request gets sent to the root server.

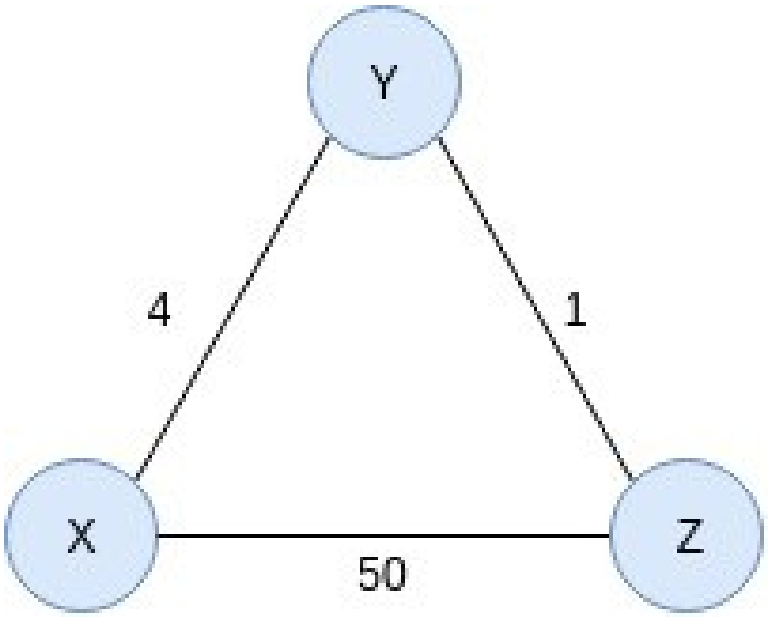
The root servers are the root of a DNS hierarchy. The root server doesn't know the IP address for www.yahoo.com, but it knows where to find it. The root

server directs the resolver to the top-level domain (TLD) server for the ".com" domain. The resolver then asks the TLD server for the IP address for www.yahoo.com. When the TLD server receives the request for yahoo.com, the TLD server doesn't know the address, but the TLD directs the resolver to the authoritative name server. So then, the resolver asks the authoritative name server for the IP address for www.yahoo.com. When the authoritative name server receives the query for the address, it replies with the IP address for that domain to the resolver server. Once the resolver receives the IP address, it stores it in the cache memory, so it doesn't have to go through those steps again. Finally, the resolver sends the IP address to the client's computer, and the client can retrieve the webpage. The exchange of packets between all of the servers uses the UDP protocol.




**Problem 14 (9 points)**

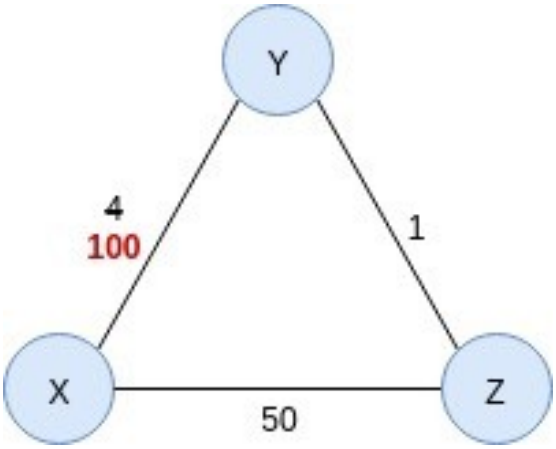
Consider the simple network in Figure 5 below. X, Y and Z are routers and their link costs are as specified. Assume the network uses a Distance Vector algorithm is used. Y's and Z's routing tables are look like Table 2.




| Node Y/Distance | Via X | Via Z |
|---|---|---|
| X | 4 | 6 |

| Node Z/Distance | Via X | Via Y |
|---|---|---|
| X | 50 | 5 |

(a) Now Let assume the cost of link X-Y suddenly changed to 100. Please write down the Y's and Z's routing table regarding distance to X, after Y updates this information to Z and then Z updates its information back. **[3pts]**



| $D^Y$ | X | Z |
|---|---|---|
| X | 4 | 6 |

| $D^Z$ | X | Y |
|---|---|---|
| X | 50 | 5 |

$t_0$

| $D^Y$ | X | Z |
|---|---|---|
| X | 4 | 6 |

| $D^Z$ | X | Y |
|---|---|---|
| X | 50 | 5 |

$t_1$

| $D^Y$ | X | Z |
|---|---|---|
| X | 4 | 6 |

| $D^Z$ | X | Y |
|---|---|---|
| X | 50 | 7 |

$t_2$

| $D^Y$ | X | Z |
|---|---|---|
| X | 4 | 8 |

| $D^Z$ | X | Y |
|---|---|---|
| X | 50 | 5 |

$t_3$

| $D^Y$ | X | Z |
|---|---|---|
| X | 4 | 6 |

| $D^Z$ | X | Y |
|---|---|---|
| X | 50 | 9 |

$t_4$

| $D^Y$ | X | Z |
|---|---|---|
| X | 4 | 51 |

| $D^Z$ | X | Y |
|---|---|---|
| X | 50 | 51 |

$t_{46}$

The algorithm will continue until the distance from x to y through z is 51. The reason being is because it would be useless to try another path as the one before, ie X → Y → Z → Y. We can save a hop and use X → Z → Y instead. Bad news travels slowly, so the algorithm would continue on until this happened though. I skipped steps t4 through 46 because it only increments by 1.

Good News Travels Fast

| $D^Y$ | X | Z |
|---|---|---|
| X | 100 | 6 |

| $D^Z$ | X | Y |
|---|---|---|
| X | 50 | 5 |

$t_0$

| $D^Y$ | X | Z |
|---|---|---|
| X | 1 | 6 |

| $D^Z$ | X | Y |
|---|---|---|
| X | 50 | 5 |

$t_1$

| $D^Y$ | X | Z |
|---|---|---|
| X | 4 | 6 |

| $D^Z$ | X | Y |
|---|---|---|
| X | 50 | 2 |

$t_2$

| $D^Y$ | X | Z |
|---|---|---|
| X | 3 | 8 |

| $D^Z$ | X | Y |
|---|---|---|
| X | 50 | 5 |

$t_3$

| $D^Y$ | X | Z |
|---|---|---|
| X | 4 | 6 |

| $D^Z$ | X | Y |
|---|---|---|
| X | 50 | 4 |

$t_4$

| $D^Y$ | X | Z |
|---|---|---|
| X | 4 | 4 |

| $D^Z$ | X | Y |
|---|---|---|
| X | 50 | 5 |

$t_5$

(b) Please write down the Y's and Z's routing table regarding X after Y updates this information to Z again and then Z updates back again. **[3pts]** How many updates did Y get until its distance to X have converged with Distance Vector algorithm? **[3pts]**

When the link went to 100, the distance vector algorithm had to iterate 46 times. This is explained in (a). The algorithm will continue until the distance from x to y through z is 51. The reason being is because it would be useless to try another path as the one before, ie X → Y → Z → Y. We can save a hop and use X → Z → Y instead. Bad news travels slowly, so the algorithm would continue on until this happened though. I skipped steps t4 through 46 because it only increments by 1.

When the link went back to normal, it only had to iterate about 5 times.