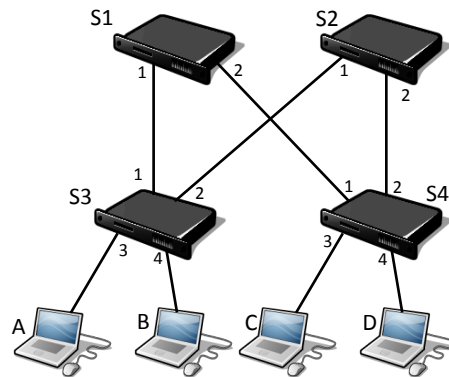


Homework 2

Due: 16 October, 4pm

Question 1 - Spanning Tree

Consider the following figure of a switched Ethernet network with two redundant core switches. Since it is switched, each individual wire is a separate physical LAN with only 2 nodes connected (on either end of the wire). The switches are numbered S1-S4, hosts named A-D, and ports on the switches numbered for each switch. In the following, refer to a port as either A-D (for the hosts), or Sn.m (e.g. S1.1) for the switches.



- a. We discussed the Spanning Tree Protocol (STP) in class. Why is it needed in this topology? Give an example of something bad that can happen without it.

[1 pt]

We need the Spanning Tree Protocol in this topology because there is at least one loop.

As an example of what can go wrong, node A may send a broadcast message that reaches S3.3. S3 copies this message to its ports S3.1, S3.2. Let's just look at one of them: S3.1 → S1.1 → S1.2 → S4.1 → S4.2 → S2.1, and back to S3. So this frame would circle the network forever, and the same thing would happen to the copy going out on S3.2. Since every broadcast message reaching either S3 or S4 would cause the same, the network capacity would quickly be exhausted.

- b. If the priority of the switches in the STP is their numeric id, what is the final state of the protocol? Give the final state by listing the state of each port (from one the three possible states - root, preferred, or disabled).

[2 pt]

The terminology here is in one-to-one correspondence to what we used in class, but let's use the one from class: root = root, preferred = designated, and disabled = blocking.

Port	State	Port	State
S1.1	Designated	S2.1	Root
S1.2	Designated	S2.2	Blocking
S3.1	Root	S4.1	Root
S3.2	Designated	S4.2	Designated
S3.3	Designated	S4.3	Designated
S3.4	Designated	S4.4	Designated

Sanity checks: there is one root port per switch; there is one designated port per LAN segment; there is one blocked port per loop; all ports attached to the root switch are designated ports.

S2 chooses its port 1 as the root port because it is connected to a switch with a lower id than port 2, and the two have same cost paths.

- c. What is the final state after switch S1 is removed? (You can pretend the ports on other switches that used to connect to S1 don't exist.)

[1 pt]

Port	State	Port	State
		S2.1	Designated
		S2.2	Designated
S3.2	Root	S4.2	Root
S3.3	Designated	S4.3	Designated
S3.4	Designated	S4.4	Designated

Note here that there is no blocking port, since there was no loop. The network was already a tree!

- d. In this new state (from c.), assuming that the switches are all learning switches, and that their tables are empty initially, list all transmissions in the network when A sends a frame to D, and when D replies to A. (Assume A and D know each other's MAC addresses, and list messages based on their source and destination ports, e.g., A→S3.3, S3.1→S2.1, etc.)

[1 pt]

From A to D: A→S3.3, S3.2 → S2.1, S3.4 → B, S2.2 → S4.2, S4.3 → C, S4.4 → D.

From D to A: D → S4.4, S4.2 → S2.2, S2.1 → S3.2, S3.3 → A.

The difference is that on the reverse path there is no broadcast, as the switches have learned the ports through which they heard from (and thus can send to) A.

Question 2 - Mystery Routing

Your friend wanted to optimize his home network and made it barely functional. Since you are taking 168, they asked you to go see if you can make it better. The network has four components: two computers, an Ethernet switch, and an IP router. The two nodes and the router are connected to the switch directly. (In reality, modern home routers are both switches and routers, but conceptually you can think of the two as separate components.)

They have the following configuration:

Node	IP Address	Netmask	Gateway
Laptop	192.168.1.6	255.255.255.224	192.168.1.1
Desktop	192.168.1.36	255.255.255.0	192.168.1.1
Router	192.168.1.1	255.255.255.0	not relevant

- a. The laptop is accessing the Internet just fine, but connections between the laptop and the desktop have been slower since he played with the configurations. You suspect that the router may be faulty, so you disconnect it, while leaving the switch and the two nodes connected. You notice that you can send UDP datagrams from the desktop to the laptop, *but not the other way around!* Why is that?

[2 pt]

You can send UDP datagrams from the desktop to the laptop because they are in the same physical LAN (both connected to the same switch), and from the desktop's point of view, the laptop is in the local subnet: $192.168.1.6 \& 255.255.255.0 == 192.168.1.36 \& 255.255.255.0$.

In the other direction, however, the laptop has a different netmask, which excludes the desktop from its subnet. So, the laptop thinks it can't reach the desktop directly (through the link layer), and that it needs to send the packet to an IP next hop. But, since you disconnected the router, there is no such next hop, and the datagram is not sent.

The netmask in the laptop is wrong: it does not correctly designate the directly reachable nodes!

- b. In an attempt to diagnose the problem, you reconnect the router, and you can send UDP datagrams in both directions now. Puzzled, you run traceroute from the laptop to the desktop. What is the result? What is the result of traceroute from the desktop to the laptop? Explain any differences.

[2 pt]

From the desktop to the laptop nothing changes when you connect the router. The traceroute would say: 192.168.1.36, 192.168.1.6.

From the laptop to the desktop, we saw that the laptop needs to send packets to the desktop through the gateway, as it thinks the desktop isn't in the same subnet. When the router receives the packet, it will rightfully send it to the desktop through the LAN, as, for the router, the desktop is in its local subnet. The traceroute would say 192.168.1.6, 192.168.1.1, 192.168.1.36.

As a curiosity, the router will also send an ICMP message to the laptop of type Redirect. This message informs the laptop that the router knows of a better route to reach the desktop, in this case directly. This message will be ignored by the laptop, as it will still think it can't use the desktop as a next hop, as it is not directly connected.

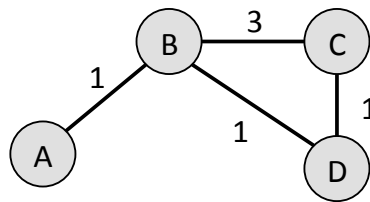
- c. You finally figure it out and fix the problem. What is the right way to fix the network?

[1 pt]

You change the network mask of the laptop to 255.255.255.0, which would correctly include the desktop. You really want to make sure all of the nodes connected to the same LAN will have the same subnet mask, to avoid these problems.

Question 3 - Routing Algorithms

Consider the network in the figure below, where numbers represent the costs of a link.



- a. (Link State) Write down the sequence of costs and next hops for each destination n from node A , i.e., $[n, C(A, n), \text{next}(n)]$, computed by each stage of running Dijkstra's algorithm on this graph. (Consider each stage to be the state of the table for each new node added to the set M . No need to write entries for which the cost is ∞ .)

[2 pt]

- 1: [A, 0, -]
 2: [B, 1, B]
 3: [D, 2, B]
 4: [C, 3, B]

- b. (Distance Vector) Assume the network has reached a stable state with a distance vector algorithm, and consider the routes **to A** from the rest of the network. Now assume node A's link to B goes down. Write down a sequence of events that will cause count-to-infinity to happen.

[2 pt]

The state of the network right before the change is (node: [dest, cost, next]):

- B: [A, 1, A]
 C: [A, 3, D]
 D: [A, 2, B]

This is assuming split horizon, in which you don't advertise to your parent.

Now B sends a message saying its cost to A has become infinity.

At the same time C sends a message saying its cost is 3. B then changes its cost to be $3 + 3 = 6$.

D receives B's message and switches its cost to ∞ .

B then sends an advertisement for cost 6, which D accepts and broadcasts. C accepts D's announcement, changing its cost to 7.

The cycle is now formed, and will remain in effect until they reach the value of infinity.

- c. List at least two advantages of path vector over distance vector protocols.

[1 pt]

- Path vector prevents loops from forming as a node will not connect to a path that has itself as a hop
- Path vector allows for policy-based route selection, as you have the entire path in the advertisement