Sample Final

Directions: Write your name on the exam. Write something for every question. You will get some points if you attempt a solution but nothing for a blank sheet of paper. Problems take long to read but can be answered concisely.

Question	Maximum	Score
1	40	
2	15	
3	15	
4	15	
5	15	
Total	100	

1, Fundamentals, 40 points total, 4 points each: Give short 1 line answers for each question. If you are asked to give a reason for something, give the most important reason you can think of.

- **Addressing:** Why Ethernet addresses are 48 bits in length although most LANs have only 1000 stations.

- **Protocol Specifications:** Besides the specification of how the protocol responds to various events and the interfaces to higher and lower layers, what is the other major component of a protocol spec.

- **Ethernets:** Why Ethernet packet formats have a length field in the Ethernet header while other Local Area network protocols do not.

- **Bridges versus Routers:** Peter Protocol is building an application that needs low latency. Peter decides he wants his network to be full of routers although the bridges are slightly faster. Explain why.

- **Spanning Tree Protocol:** I barely mentioned in class that when a bridge dies, the bridges reconfigure their spanning tree so that other bridges that were part of a loop can now start working. Also, I did not tell you this in class, bridges time out learned addresses faster after a spanning tree topology change. Why?

- **CIDR:** Why new organizations get multiple consecutive class C addresses instead of randomly assigned addresses.

- **ARP:** Suppose your PC sits on a LAN with a router R. Suppose the router's Ethernet card is not working and a maintenance person, yanks out the card and replaces it with another one. Your PC has not been touched but you may find that you cannot access any web sites other than on your LAN for a day. Why?

- **Link State Routing:** Why a source S sending a link state packet may get a packet with source S and a higher sequence number than S is currently using.

- **Distance Vector Routing:** Hugh Hopeful suggests stopping the count-up of Distance Vector when the distance reaches the diameter of the network. The diameter of a network is the length of the longest shortest path between two nodes in the network. What is the problem with Hugh's suggestion.

- **Fragmentation:** One reason why IP routers are not encouraged to fragment packets.

2. Ethernet Protocols, 15 points: The Ethernet protocol is called CSMA-CD because it has has three main ideas: *carrier sense, collision detection, and backoff.* Briefly explain these ideas and why each idea allows the Ethernet to have better performance than ALOHA (which uses none of these ideas.) Recall that Aloha is the idea that you send whenever you feel like, detect collisions by not receiving an ack, and use a random backoff of fixed maximum duration. Only 1-2 line explanations and comparisons are needed. The points are allocated as follows.

- Ethernet Carrier sense (explanation of what it is: 2 point, why it improves over Aloha, 3 points)

- Ethernet collision detection (explanation: 2 points, why it improves over Aloha, 3 points)

- Ethernet backoff (explanation: 2 points, why it improves over Aloha, 3 points)

3, Bridging and Learning, 15 points: Hugh Hopeful notices that at very high speeds it is hard for bridges to learn information from the source addresses in every packet. So Hugh suggests that bridges look at source addresses only in multicast packets. Since routing endnode protocols typically ensure that endnodes send multicast packets (e.g., ARPs, OSI hellos), this should ensure that each bridge periodically hears a multicast packet from each endnode. Also, since multicast traffic is so much less than non-multicast traffic, the processing load on bridges to do learning will be considerably reduced. Peter Protocol, who is brought in as a consultant, points out that not all endnodes send multicast periodically.

- As usual, bridges will flood unknown destination frames. What is one disadvantage of using Hugh's scheme of learning from multicast messages only (based on Peter Protocol's comment).

- All IEEE 802 LANS are supposed to support the SYSID-REQ message. When a station X on a LAN sends a SYSID-REQ message to the broadcast address all stations are supposed to send a SYSID-RESP message back to X. This can be used, for instance, by a manager to find how many stations there are on a LAN. How can Hugh use the SYSID-REQ message to avoid Peter Protocol's objection.

- Would the SYSID scheme work well in a large Extended LAN with 8000 stations? Explain.

4. Modifying Routing to do Load balancing, 15 points: In the figure below, there are two equivalently good routes from $R0$ to $R6$, one through $R1$ and one through $R3$, both with cost 4. Most routing algorithms today will only choose (arbitrarily) one of the two routes. Thus $R0$ will choose to send to either $R1$ or $R3$ but not to both. However, if $R0$ is a high speed router and the links are slow it may be better for $R0$ to send some packets to $R1$ and some packets to $R3$, thus balancing the load on the two paths and getting more throughput. We are going to see what modifications we need to add to routing for load balancing.

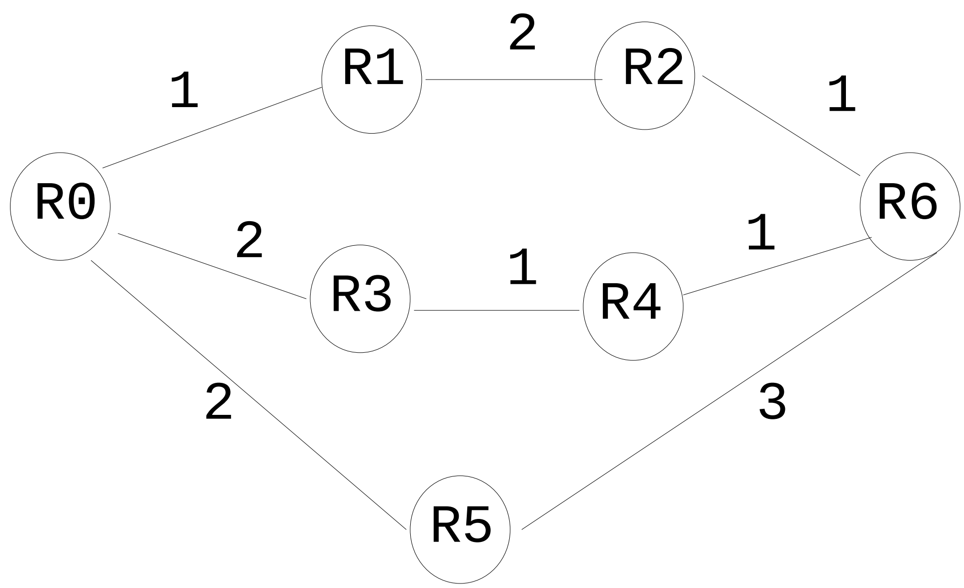

Figure 1:

- In distance vector routing, a router R computes the neighbor that is closest to a destination $R6$ using the distances sent by its neighbors as follows:

  The closest neighbor for destination D is the neighbor N such that $Distance(D, N) + Distance(R, N)$ is the smallest over all neighbors.

  How would you modify this protocol to *also* compute *all* neighbors that provide *equal cost* routes to destination D.

- It is also theoretically possible to not limit ourselves to equal cost paths. For example, in the figure above there is a path of cost 5 between $R0$ and $R6$ through $R5$. It seems that we could do better load balancing by having $R0$ send a small fraction of its packets through $R5$ as well. However, this kind of load balancing can lead to packet looping unless care is taken. Explain why.

5. Modifying Endnode Routing to do Load Balancing, 10 points: In the preceding page, we saw how to modify the router code to calculate equal cost routes. Now we turn to endnodes. In the figure below, we see that an endnode S on a LAN has two equally good routes (through either $R1$ or $R3$) to get to destination D. (The heavy lines represent LANs e.g., Ethernets). It is typically worth having S sending half its traffic to D to $R1$ and half to $R3$ because the LANs are much faster than the routers and the links between routers. Thus S needs to find out that $R1$ and $R3$ offer equally good paths to D so that it can split traffic among them. Notice that S must not choose $R5$ to split traffic to, because this only causes an extra hop.

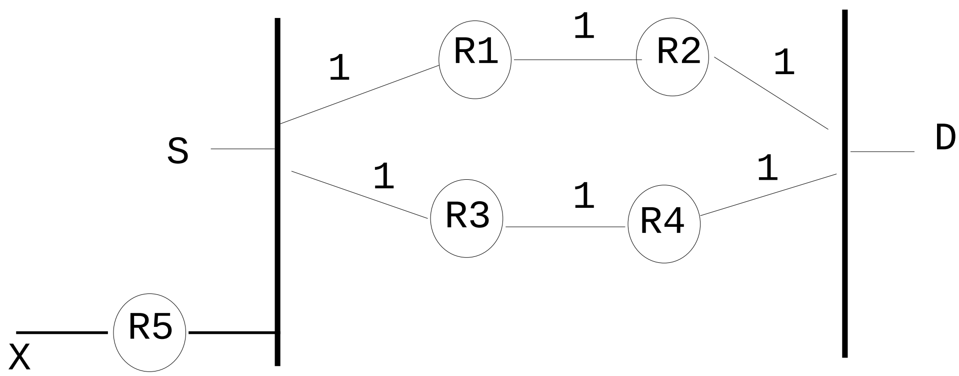

Figure 2:

The idea is to have a special QUERY message. If an endnode S has no information cached for D, S sends a QUERY to any router it knows about. The router sends back a REPLY with the list of routers that offer equal cost paths to D

- The algorithm used by a router to reply to a QUERY is trickier than you might think. It is obvious that $R5$ already knows that $R1$ and $R3$ are the best ways for $R5$ to get to D. However, S may choose to ask $R1$. How is $R1$ to know that $R3$ is also an equally good way to get to D? Assume the use of distance vector routing.
- Suppose S has a cache entry for D that says the best two routers are $R1$ and $R3$. Then the link from $R1$ to $R2$ crashes. $R1$ quickly calculates that the best route to D is through $R3$ but S may still have an old cache entry. How should $R1$ react when S sends a packet from D to $R1$. How can S use this information to update its cache?

```
The heart has its reasons of which reason knows nothing.

                             --- Pascal, The Pensees
```

I hope more than anything else in this course I have been able to convery some of my excitement for the field of networks to you. In the long run, a passion for learning and creating will win over mere intelligence. I enjoyed knowing all of you. Happy holidays!