UCSD CSE CS 123	Computer Networks

Final Exam

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	30	
2	10	
3	10	
4	10	
5	15	
6	10	
7	5	
8	10	
Extra	15	

1, Fundamentals, 30 points total, 3 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.

- **Media:** Why it is still cheaper to run twisted pair to workstations though the cost of fibre wire is fairly cheap.

- **Physical Layer:** If a designer of a physical link finds that there is Intersymbol Interference (ISI) when he tries to send bits over a link, what can the designer do? (Any one reasonable option will do.)

- **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.

- **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:** Although we did not tell you this in class, bridges time out learned addresses faster after a spanning tree topology change. 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.

- **ATM Networks:** Why a simple bit for the last cell in a packet is sufficient for ATM cell reassembly while IP needs an offset and a packet ID.

- **Congestion Control:** Why can the throughput of a network go to zero (congestion collapse) if too much traffic is allowed to enter the network?

- **Transport:** What resources does a transport connection consume at a workstation even when the user of the connection is not sending any data.

2. ATM and Multicasting, 10 points: It seems easy to treat an ATM network as as a big LAN (i.e., like an Ethernet) in order to run routing protocols like IP over ATM. However, while ATM does provide multicast virtual circuits, it does not provide multicasting the way Ethernet does. Recall that in a LAN every packet sent by a source goes to all nodes in the LAN.

- What problem can the lack of Ethernet style multicasting cause for routing protocols like IP or OSI? (Hint: think how protocols like ARP would work over the ATM network.)

- One solution to provide a Ethernet style multicasting facility for ATM is to create a *single* multicast virtual circuit that links all nodes in the ATM network. Then any multicast message can be sent on this multicast virtual circuit. However, Peter Protocol points out that reassembly of cells into frames is done on a per virtual circuit basis in AAL-5 and so this solution will not work without further changes. What specific problem is Peter concerned about?

- Can you think of any solutions to Peter's problem? (Just a few lines will do for a solution idea, however vague.)

3. BGP, 10 points: Explain briefly the main differences between BGP and Distance vector in terms of a) how routes are chosen b) how routes are considered to be unreachable.

4, Bridging and Learning, 10 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.

5. 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.

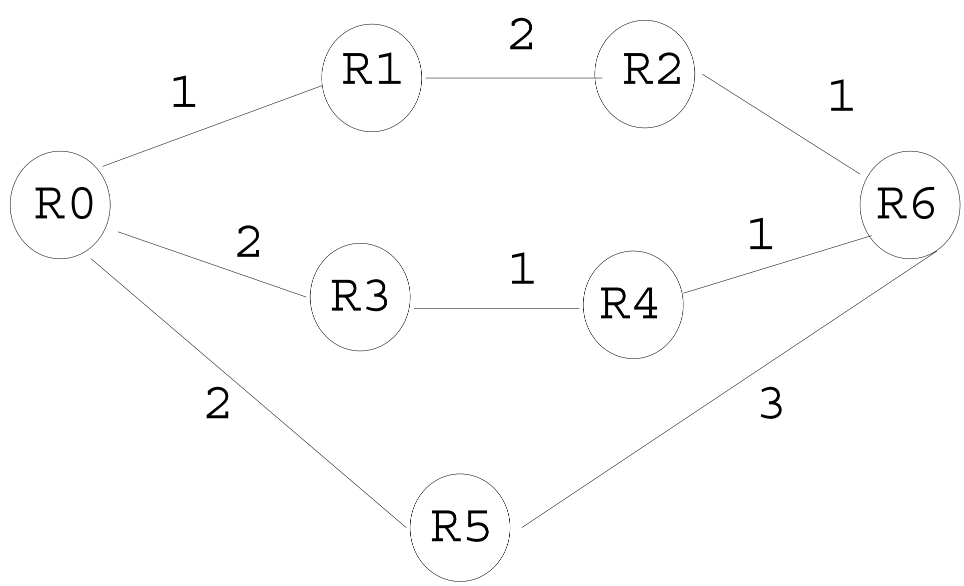

Figure 1:

- 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.

6. 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.

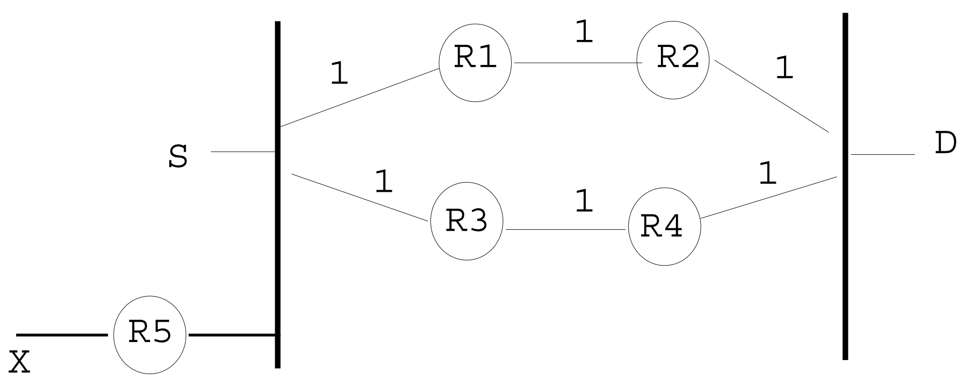

Figure 2:

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?

7. Modifying Transport Protocols to Deal with Load Balancing, 5 points: Hugh Hopeful uses a sliding window transport protocol. over a routing protocol and everything works fine. Hugh Hopeful later modifies the routing protocol so that it can do load balancing as shown above. However, he finds that performance actually decreases when he does load balancing.

- Why does performance go down?
- What simple change does Hugh need to make to his transport protocol implementation?

8. Reverse Path Congestion Control, 15 points: Recall that in congestion control we had two separate problems. A router had to *sense* congestion on a link and then send *feedback* to all sources using the link. In class, we described the DECbit/ECNbit scheme in which the bit is passed to the destination and then back to the source. Here we describe another scheme in which a congestion bit is passed from the router directly back to the source.

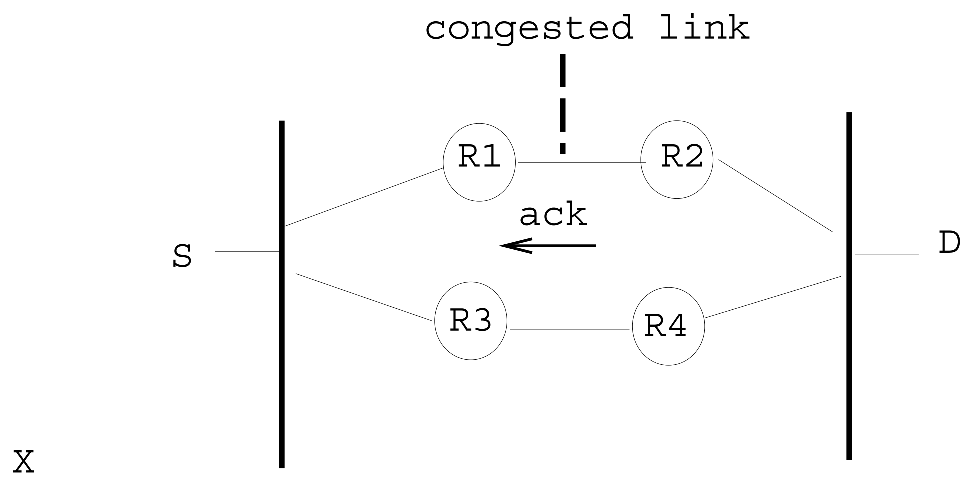

Figure 3:

In the figure below, assume that S is sending packets to D through $R1$ and all acks from D return on the same path. The *reverse path congestion rule* is as follows: if a packet p is received from a link l that is congested in the outbound direction, then a congested bit is set in the routing header of packet p. Thus in the figure, if the link from $R1$ to $R2$ gets congested, then the outbound queue at $R1$ and going to $R2$ will build up. When any packet from D to S arrives on this link (for example an ack), $R1$ will set the congested bit and this bit will get to S which then can send at a slower rate.

- What is one advantage of reverse path congestion control over the DECbit/ECNbit scheme?
- The correctness of reverse path congestion control depends on an assumption. What is the assumption and why does it not always hold for all routing algorithms?

`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!