EECS 122: Introduction to Communication Networks

Midterm 1 Solutions

Problem 1. (2 points) Two nodes are connected by a point-to-point link with a propagation delay of 1 μs and a data rate of 10 Mbps. How long after one node begins sending a 1000 byte packet will it be completely received by the other?

> We first need to get the packet onto the link (this is the transmission time), then we need to get the last bit across the link (this is the propagation delay). Or thought of another way, we first need to get the first bit across the link (propagation time), then we need to get the packet off the link and onto the destination node (transmission time). Either way the total time is the sum of transmission time and propagation delay.
>
> $$\text{transmission time} = 1000\text{ bytes} \cdot \frac{8\text{ bits}}{\text{byte}} \cdot \frac{\text{second}}{10 \times 10^6\text{ bits}} \cdot \frac{10^6\ \mu\text{s}}{\text{second}} = 800\ \mu\text{s}$$
>
> $$\text{propagation time} = 1\ \mu\text{s}$$
>
> $$\text{total time} = 801\ \mu\text{s}$$

Problem 2. (2 points) If a mail transfer agent wishes to send a message to foo@bar.com, it must open a connection to port 25 of some IP address. How does it determine the IP address? (A sentence or two is sufficient.)

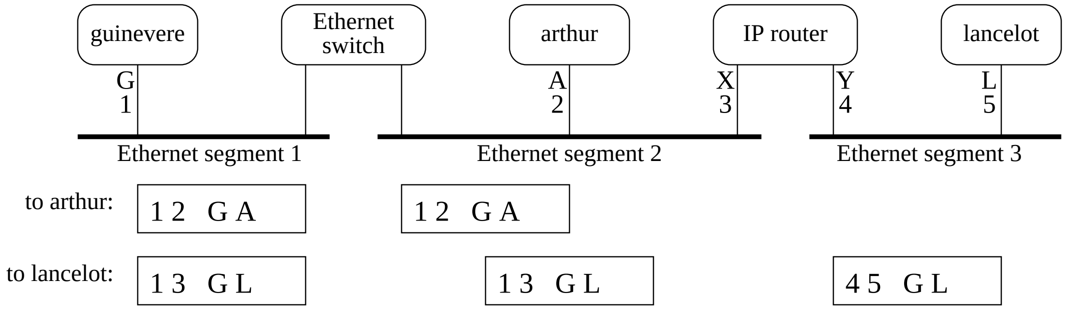
> The MTA first looks up the mail exchanger for the domain name (bar.com) using DNS (the domain name system). The mail exchanger is returned as a host name, so the MTA then looks up the IP address for that hostname, again using DNS.
>
> (Notice that the MTA might end up talking to two different name servers, because the name of the mail exchanger for bar.com need not end with “bar.com”.)
>
> Answers that neglected to talk about mail exchangers, or neglected to talk about DNS (or name servers), received at most one point, possibly less if there were other mistakes (like looking up “foo@bar.com”, which is not a domain name and therefore cannot be lookup up in DNS).

Problem 3.

a) (2 points) Label the figure below to show all addresses that exist in the network. Use letters to represent IP addresses and integers to represent Ethernet addresses.

One possible correct answer is shown below. Every interface has both an Ethernet address and an IP address, except that an Ethernet switch is transparent, so it has no addresses. An Ethernet switch is a drop-in replacement for a hub, so frames are never addressed to the switch, only to hosts (including routers), and the switch copies the frames without altering them. We did not deduct any points if Ethernet address were assigned to the switch, but we did deduct a half point if any IP addresses were assigned to the switch, because it is a link-layer device and has no network layer.

Since the router has two interfaces, it needs two sets of addresses. At the Ethernet layer, this is simply because the Ethernet address physically resides on the interface hardware, and there are two interfaces. At the IP layer, it's because an IP address has the form *net.host*, and each interface is on a different network (the whole purpose of a router is to forward datagrams from one network to another). Assigning only one Ethernet address and one IP address to the router was a one-point deduction; assigning two Ethernet addresses and only one IP address was a half-point deduction.



b) (2 points) Suppose guinevere sends a packet to arthur. In each rectangle in the first row, write all addresses that appear in the headers as the packet traverses the Ethernet segment above the rectangle.

Hosts use the network layer to send packets to other hosts, and the network layer uses the link layer to send these packets one hop at a time. So the packet will be an IP datagram (with source and destination IP addresses) encapsulated inside an Ethernet frame (with source and destination Ethernet addresses). The switch is transparent, so the packet is the same on Ethernet segments 1 and 2.

One common mistake was to leave out the IP addresses, which cost one point. Even for hosts on the same network, the IP layer is still present and functioning (because the transport layer above the IP layer knows how to talk to the IP layer, but not how to talk to the link layer). This was illustrated in the solution to homework 3 problem 2.

c) (2 points) Now suppose guinevere sends a packet to lancelot. Fill in the second row of rectangles as in part b.

> The IP datagram contains source and destination IP addresses of guinevere and lancelot, but this time it is forward in two hops. For the first hop, it is encapsulated inside an Ethernet frame going from guinevere to the router; for the second hop, it is encapsulated inside an Ethernet frame going from the router to lancelot. Again, the switch is transparent, so the packet is the same on Ethernet segments 1 and 2.
>
> The IP addresses were worth one point, and the Ethernet addresses were worth one point. If the router was assigned only one Ethernet address, we already deducted a whole point for that in part a, so we did not deduct again for it here. (If we accidentally deducted twice for this mistake, tell us.)

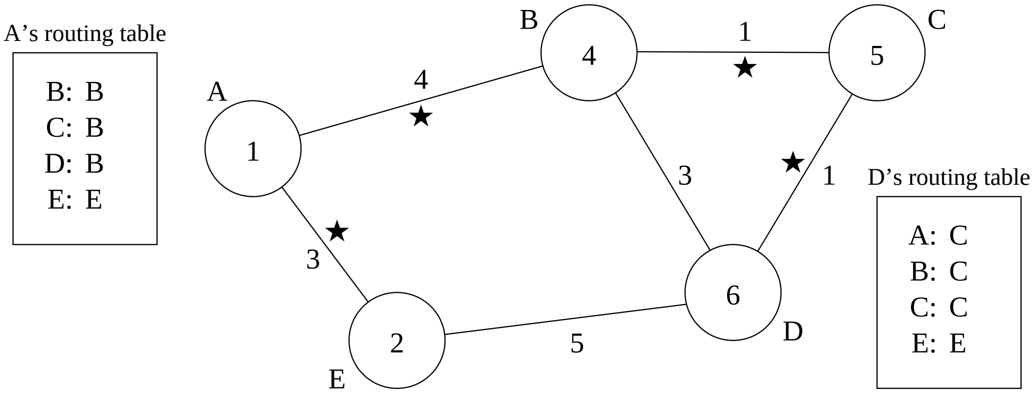
d) (2 points) How does guinevere decide whose Ethernet address to send each packet to, and how are these addresses learned? (Briefly describe the mechanisms in a few sentences.)

> There are two questions—*whose* address, and how is it learned? Notice that in part b guinevere sends the frame to the destination (arthur), while in part c guinevere sends the frame to the router. This choice is based on whether the destination is on the same network as the source, which is determined using the source and destination IP addresses and the netmask. This was discussed in section 3.3.2 of the textbook, and explained more clearly on the “Internet model” web page (which was linked from the Announcements section), and was integral to the solution of homework 3 problem 10.
>
> Regardless of whose Ethernet address is chosen, it is discovered using ARP (the address resolution protocol). The source broadcasts a message on the local network asking who has a particular IP address, and the owner of that IP address sends a response containing its Ethernet address.
>
> One point was awarded for the first question, and one point for the second. For the second question, either mentioning ARP or describing how it works was sufficient. For the first question, recognizing that the frame will be addressed to the destination or the router depending on whether the destination is on the local network was worth a half-point, and describing the correct method for making this determination earned the other half-point. A common answer was to use ARP to determine whether the destination was local, but ARP is not used for that, it is used only to map IP addresses to link-layer addresses. The destination IP address determines whether the destination is local, and failure to get an ARP response means the host or the network is down.

Problem 4.

a) **(2 points)** The nodes below have been running a shortest-path routing protocol (such as RIP or OSPF) long enough that the paths have stabilized. Show the routing tables for nodes A and D. Use a separate row for every destination node, indicating the next node on the path.

> The answer is shown below. A routing table maps destinations to next hops, where a next hop simply tells the router how to forward the packet to the next router. A next hop does not include the entire path that the packet will follow. In real life, a next hop is an interface ID (like "my 3rd interface"), plus the address of the next router on the path if that interface is attached to a multipoint link. The picture below shows no interface IDs or link IDs, so the problem asked for the next node. One point was awarded for demonstrating an understanding of what a routing table is, and another point was for indicating the correct shortest paths.



A's routing table

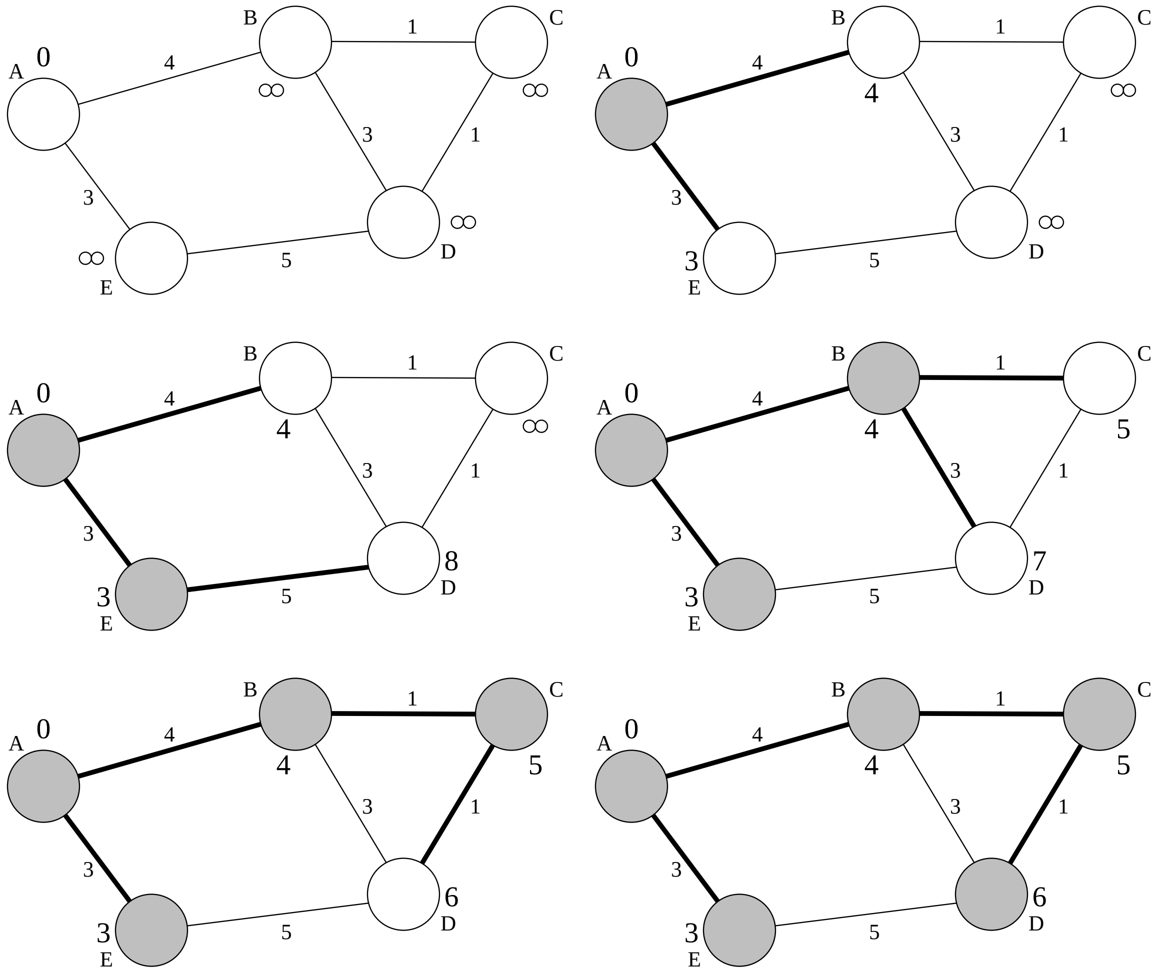
B: B
C: B
D: B
E: E

D's routing table

A: C
B: C
C: C
E: E

b) **(2 points)** Execute Dijkstra's shortest-path algorithm with node A as the source. Number the nodes in the figure in the order they are added to the shortest-path tree (starting with node A as number 1), and draw stars near the edges belonging to the final shortest-path tree.

> This was very similar to homework 4 problem 2, except that this problem did not require a diagram for each step of the algorithm. To illustrate the answer, however, each step is shown below. The algorithm is explained in section 3.4.2 of the textbook. The ordering of the nodes was worth one point, and identifying the shortest-path tree was worth one point.



Problem 5. (2 points) Consider a shared Ethernet that uses a single coaxial cable (no repeaters). Suppose the maximum propagation delay from any station to any other is 10 μs. What is the *channel acquisition time*? In other words, how soon after a station begins to transmit can it be sure that its frame will not collide with another (if it hasn't already)?

> Call the transmitting station T_1. If another transmitting station T_2 is going to collide with T_1, it must start transmitting less than 10 μs after T_1 starts, otherwise it will hear T_1 and not transmit. After T_2 starts, it will take no more than 10 μs before T_1 hears that another station is transmitting (detects a collision). Therefore, if there is going to be a collision, T_1 must hear it no more than 20 μs after it starts transmitting. In other words, if T_1 hears no collision during the first 20 μs, then there cannot be a collision.
>
> Notice that the receiver(s) of the frames are not involved in collision detection. A stations detects a collision simply by hearing that it is not the only one transmitting. The problem stated that there were no repeaters (also known as hubs), but the answer would have been the same if there were hubs, which are designed to make the multiple segments look to the stations like a single shared broadcast medium.
>
> The correct answer (20 μs) was worth a point and a half. A correct explanation added another half point, but an incorrect explanation subtracted a half point.