Computer Science 15-441: Networks
Mid-Term Exam (A), Fall 2004

1. **Please read the entire exam before starting to write. This should help you avoid getting bogged down on one problem.**
2. Be sure to put your name and Andrew ID below *and also* put your Andrew ID at the top of *each* following page.
3. If you have a question about the exam, please write it down on the card you were provided with and then raise your hand.
4. This is a closed-book in-class exam. You may not use any reference materials during the exam.
5. You must complete the exam by the end of the class period.
6. Answer all questions. The weight of each question is indicated on the exam. Weights of question *parts* are estimates which may be revised during the grading process and are for your guidance only.
7. Please be concise in your answers. You will receive partial credit for partially correct answers, but truly extraneous remarks may count against your grade.
8. **Write legibly even if you must slow down to do so!**. If you spend some time to *think clearly* about a problem, you will probably have time to write your answer legibly.

Andrew Username	
Full Name	

Question	Max	Points
1.	**25**	
2.	**20**	
3.	**10**	
4.	**15**	
5.	**20**	
6.	**20**	
	110	

1. [25 points] Data Transfer

Calculate the total time required to transfer a 1000-KB file in the following cases, assuming an RTT of 100 ms, a packet size of 1 KB and an initial 2 * RTT of "handshaking" before data is sent. Ignore overhead due to headers.

(a) [7 points] The bandwidth is 1.5 Mbps, and data packets can be sent continuously.

$T_{total} = T_{handshake} + T_{prop} + T_{transfer}$
$T_{handshake} = 2 * RTT = 200ms$
$T_{prop} = RTT = 100ms$ `(includes both directions)`
$T_{transfer} = (1000packets) * (1024B/packet * 8b/B)/(1.5Mb/s) = (8096b)/(1500b/s) = 5.4s$
$T_{total} = 5.7s$

(b) [8 points] The bandwidth is 1.5 Mbps, but after we finish sending each data packet we must wait one RTT before sending the next.

`Same as above, except for an additional 100ms RTT delay between each packet.`
`Thus` $T_{total} = 5.7s + (999packets) * (100ms/packet) = 105.6s$.

Most homes are connected by a copper twisted pair to a telephone company's central office. Analog-modem dialup and DSL are two data services provided by the telephone company over these copper wires. However, modem speeds are usually around 30Kbps while DSL speeds may be several Mbps.

(c) [5 points] Use Shannon's Theorem, $C = B\log_2(1+\frac{S}{N})$, to explain why there is big a speed difference between analog modem service and DSL even though they run over the same twisted pair.

DSL and analog modems use different frequency ranges. In particular, the modem uses the voice frequency range with a bandwidth of B=3khz, whereas for DSL, B may be several Mhz.

(d) [5 points] Imagine you and your cousin live in different neighborhoods of the same town and use the same DSL modems, provided to you by local telephone company's DSL service department as part of the same monthly service plan, to connect to different ports on the same central-office DSLAM. Yet your cousin's Internet connection always seems to be faster than yours. Explain.

The lines from the central office to each of your houses may be different lengths or qualities, which would give each of you different S/N ratios.

2. [20 points] Short answer.

   (a) [4 points] Two nodes are directly connected by a single point-to-point link which is error-free. As part of a networking lab experiment, you have instrumented your system to record very fine-grained timestamps at the instant when a packet is accepted by the link layer for transsmision and also at the instant when the link layer has completed transmission of the packet. Your system stores the start and end timestamps for each packet transmission, along with the 20-byte IP header, in a log for later analysis. Imagine your surprise when you find that IP datagrams of exactly the same size frequently require different amounts of time for the link layer to transmit them. How is this possible ?

   The link or physical layer protocols may be performing bit-stuffing. Thus, the identical-length bit sequences provided to the link layer by the network layer may turn into different length sequences at the physical layer.

   (b) [6 points] While all the wired Ethernet variants use CSMA/CD, "wireless Ethernet" (IEEE 802.11) does not. Why was this choice made ?

   1. The inverse-square falloff of transmitted signal strength implies that the signal seen by a receiver next to the transmitter will be much stronger than that seen by a distant receiver. In the best case, if a node listens to its own transmissions, its signal will be so much stronger than a colliding signals from a distant transmitter that the colliding signal will not be heard. Additionally, some receiver hardware can be damaged by too high a received signal strength, so many radios turn off their receiver when transmitting.
   2. Wireless networks suffer from the ‘‘hidden node problem.’’ Imagine node A on a mountain, with nodes B and C on opposite sides. Node A can detect when B and C are transmitting at the same time, but B will not be able to hear C at all and vice-versa. Even if B and C could do CSMA/CD, the best they could do is to avoid colliding with A's transmissions, but they have no way of avoiding colliding with each other.

(c) [5 points] List two advantages of OSPF over RIP:

1. Each node has complete knowledge of all link states in the network. This allows for faster convergence after a link failure and avoids the count to infinity problem.
2. Since each node has complete knowledge, it is easier to extend OSPF for load balancing.

(d) [5 points] List two advantages of RIP over OSPF:

1. Requires less storage at each node. (no complete link state table)
2. More efficient and simpler algorithm at each node.
3. Changes in topology (i.e., link/node failure) can be fixed locally. In OSPF, the changes must be propagated to all nodes.

3. [10 points] Connectivity Matrix

Given the network in figure 1, what information about the network would node B know in the two cases below? We are not asking you to generate the routing table for B, instead we want to know what node B knows about the connectivity of the network. For example, does it know if node A is connected to node C and the cost of that link? For each case fill in the connectivity matrix (you need fill in only the upper half).

- If node B knows that two nodes are directly connected, then fill in the cost.
- If node B knows that two nodes are not directly connected, then fill in with an X.
- If node B does not know if two nodes are directly connected, then fill in with a U.

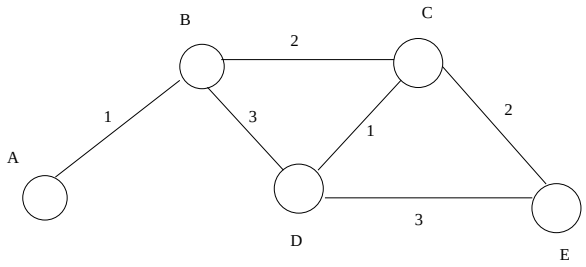

Figure 1: Network layout

(a) [5 points] All nodes in the network run a link-state protocol

	A	B	C	D	E
A	0	1	X	X	X
B	•	0	2	3	X
C	•	•	0	1	2
D	•	•	•	0	3
E	•	•	•	•	0

Table 1: Link-State

(b) [5 points] All nodes in the network run a distance-vector protocol *without* split horizon.

	A	B	C	D	E
A	0	1	U	U	U
B	•	0	2	3	X
C	•	•	0	U	U
D	•	•	•	0	U
E	•	•	•	•	0

Table 2: Distance-Vector

4. [15 points] Fragmentation

Figure 2 shows a network consisting of six nodes. The links between the nodes have different MTUs. The MTU for the links A-B and E-F is 9000 Bytes. Node A sends an IP packet of size 2604 Bytes to F. The packet undergoes fragmentation in the network, and F receives three fragments of size 1500 Bytes, 572 Bytes, and 572 Bytes (not necessarily in that order). The sizes include the sizes of both the IP header and the payload.

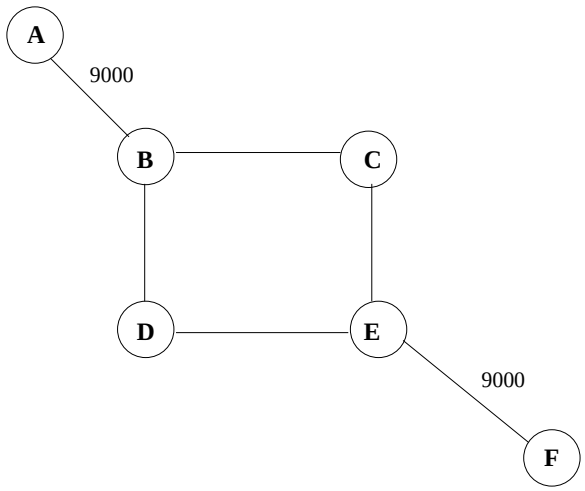

Figure 2: Fragmentation Network

(a) [10 points] Suggest possible MTU values for the links B-C, B-D, C-E, and D-E, and explain what path in the network did each fragment traverse.

Possible MTU values for the links are:
B-C: 1500B
B-D: 576B
C-E: 1500B
D-E: 576B

(9000B is the MTU of gigabit ethernet jumbo frames, 1500B is the MTU for ethernet, and 576B is the MTU for PPP.)

The large fragment followed the path B-C, C-E and E-F. The two smaller fragments followed the path B-D, D-E, and E-F.

(b) [5 points] Suggest a plausible (or at least defensible) reason why the fragments might have chosen the paths you specified above.

The packet is fragmented at B and one fragment of size 1500B (20B header + 1480B payload) is sent over the link B-C. Now suppose the link B-C goes down, so B sends data to F over the link B-D. B fragments the second fragment into two fragments of size 572B (20B header + 552B payload) and sends them over the link B-D.

5. [20 points] Ethernet

   (a) [5 points] Explain why Ethernet has a maximum packet size.

   Multiple answers are acceptable here. One is ‘‘head-of-line blocking’’ which is a fancy way to say that if one station transmits a very long packet then potentially urgent data at other stations will ‘‘wait in line’’ for an unacceptably long time.

   Another answer is ‘‘jabber detection’’: since a hub is a single collision domain, if one station suffers a hardware failure or a run of thin-net somehow couples to too much electrical noise, a port which transmits nonsense continuously can take down an entire network. Smart hubs can decide after seeing too many bits in a row from one port to electrically disconnect that port for a period of time so the rest of the network can be used.

   (b) [7 points] Explain why Ethernet has a *minimum* packet size.

   To ensure that all parties in a collision receive at least some interference before finishing transmission, so that the collision will not go undetected; see Section 2.6.2 of the text.

(c) [8 points] Imagine you are the chief technical officer of the networking services division of the Florida state government, and that you are in the position of evaluating bids from companies who want to rebuild the Florida state government's IP network infrastructure after the recent series of devastating hurricanes.

One company submits a proposal to do the whole job with Ethernet switches, using routers only to connect to the rest of the Internet. They begin by observing that that economies of scale have resulted in Ethernet switches being substantially cheaper than IP routers. They then argue that both 100-Mb and Gigabit Ethernet can be run over fiber, so long-distance links are not an issue, and that modern switches use full-duplex links which are collision-free so that there are no issues related to CSMA/CD, link idle time due to backoff, etc. With this design, all traffic within Florida will cross Ethernet switches, but never an IP router.

Describe the two most serious flaws you see in this design. A well-written paragraph should suffice for each flaw; do not write more than one page for each or mention more than two flaws.

1. This proposal would turn the entire state of Florida into a single broadcast domain. When, for example, a machine in Tallahassee ARPs to find the Ethernet MAC address of the departmental file server, the Ethernet broadcast packet (addressed to FF:FF:FF:FF:FF:FF) will be transmitted to every state-government machine in the state (even on Key West!). Probably most network bandwidth would be wasted on bothering totally unrelated computers with ARP queries.
2. Closely related to the first issue is a serious problem with **non-broadcast** packets. Ethernet switches would be essentially unable to learn which MAC addresses were served by which ports, since most switches don't have forwarding tables with hundreds of thousands (or millions) of entries. While a switch would arguably receive frequent updates from nearby stations, there would also be a steady flood of packets from distant hosts which would cause entries to be forced out of the forwarding table almost as soon as they were inserted.
3. The Ethernet spanning-tree algorithm, designed to work well in a LAN environment, where redundant links are often not necessary, deals with loops by temporarily disabling (ignoring) some links in the network. A statewide corporate/governmental network, on the other hand, should have a non-trivial amount of redundancy, which would be disabled and hence wasted in financial terms.

The best answer we can think of would be to mention either of the first two points and the third: one learning problem, and one spanning-tree problem.

6. [20 points] IP vs. Ethernet

During their respective convergence processes, both IP and Ethernet can exhibit forwarding loops.

(a) [5 points] Briefly explain what an IP forwarding loop "looks like"-how it comes into existence and provide an example.

One simple example of IP forwarding loop is the following: A and B are two neighboring nodes, A's forwarding table has the entry <X, B> and B's forwarding table has the entry <X, A>. Here <X,A> means that for a packet with destination address X, forwards to the interface that connects to node A.

Forwarding loop forms because events (such as link failures) reach different nodes at different times. During the period between event happens and the distributed routing computations finally converge, nodes may have inconsistent view of the network.

There are many examples, details omitted here.

(b) [5 points] Briefly explain what an Ethernet forwarding loop "looks like"-how it comes into existence, and provide an example.

Before the spanning tree algorithm converges, unblocked interfaces of some bridges may form a loop. A broadcast packet (e.g. a packet with a destination address that has not been learned by bridges in the loop) will circulate among the bridges.

(c) [5 points] Assume a packet encounters an IP forwarding loop. How long will it circulate?

It will stop circulation when its TTL goes to zero.

(d) [5 points] Assume a packet encounters an Ethernet forwarding loop. How long will it circulate?

It will stop circulation when the spanning tree algorithm converges and the forwarding loop disappears. The number of replicated packets in this case can be quite high.