Department of Electrical Engineering and Computer Science

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

6.02 Fall 2009

Quiz III

There are **24 questions** and **13 pages** in this quiz booklet. Answer each question according to the instructions given. You have **120 minutes** to answer the questions.

If you find a question ambiguous, be sure to write down any assumptions you make. **Please be neat and legible.** If we can't understand your answer, we can't give you credit!

Use the empty sides of this booklet if you need scratch space. You may also use them for answers, although you shouldn't need to. *If you do use the blank sides for answers, make sure to clearly say so!*

Before you start, please write your name CLEARLY in the space below.

One two-sided "crib sheet" and calculator allowed. No other notes, books, computers, cell phones, PDAs, information appliances, carrier pigeons carrying answer slips, etc.!

Do not write in the boxes below

1-5 (x/24)	6-11 (x/29)	12-17 (x/27)	18-24(x/20)	Total (x/100)

Name: SOLUTIONS

I MAC Protocols

Alyssa P. Hacker is setting up an 8-node broadcast network in her apartment building in which all nodes can hear each other. Nodes send packets of the same size. If packet collisions occur, both packets are corrupted and lost; no other packet losses occur. All nodes generate equal load on average.

1. [4 points]: Alyssa observes a utilization of 0.5. Which of the following are **consistent** with the observed utilization?

(Circle True or False for each choice.)

A. True / False Four nodes are backlogged on average, **and** the network is using Slotted Aloha with stabilization, **and** the fairness is close to 1.

False. With four backlogged nodes and fairness close to 1, the probability of of transmission is 1/4. That would put the utilization at $4 \cdot 1/4 \cdot (1 - 1/4)^3 < 0.5$.

B. True / False Four nodes are backlogged on average, **and** the network is using TDMA, **and** the fairness is close to 1.

True. TDMA gives each node an equal share of the network, but 4 nodes have nothing to send, giving a utilization of 0.5.

2 points each.

Now suppose Alyssa's 8-node network runs the Carrier Sense Multiple Access (CSMA) MAC protocol. The maximum data rate of the network is 10 Megabits/s. Including retries, each node sends traffic according to some unknown random process at an average rate of 1 Megabit/s per node. Alyssa measures the network's utilization and finds that it is 0.75. No packets get dropped in the network except due to collisions, and each node's average queue size is 5 packets.

2. [4 points]: What fraction of packets sent by the nodes (including retries) experience a collision?

(Explain your answer in the space below.)

Solution. The offered load presented to the network is 8 Megabits/s in aggregate. The throughput of the protocol is $0.75 \cdot 10 = 7.5$ Megabits/s. The packet collision rate is therefore equal to $1 - 7.5/8 = 1/16 = 6.25\%$.

As mentioned earlier, the average number of packets in a node's queue is 5. Each packet is 10000 bits long.

3. [4 points]: What is the average queueing delay, in milliseconds, experienced by a packet before it is sent over the medium?

(Explain your answer in the space below.)

Solution. Apply Little's law. The rate at which the network processes packets is $10^6/10^4 = 100$ packets/s. The average queueing delay is therefore equal to $5/100$ seconds, or 50 milliseconds.

4. [6 points]: In the Aloha stabilization protocols we studied, when a node experiences a collision, it decreases its transmission probability, but sets a lower bound, p_{min}. When it transmits successfully, it increases its transmission probability, but sets an upper bound, p_{max}.

(Explain your answers in the space below.)

A. Why would we set a lower bound on p_{min} that is not too close to 0?

Solution. To avoid starvation where some nodes are denied access to the medium for long periods of time.

B. Why would we set p_{max} to be significantly smaller than 1?

Solution. To avoid the capture effect, in which a successful node hogs the medium for multiple time slots even when other nodes are backlogged.

C. Let N be the average number of backlogged nodes. What happens if we set $p_{\text{min}} >> 1/N$?

Solution. The rate of collisions will be high and the utilization close to 0.

5. [6 points]: You have two computers, A and B, sharing a wireless network in your room. The network runs the slotted Aloha protocol with equal-sized packets. You want B to get twice the throughput over the wireless network as A whenever both nodes are backlogged. You configure A to send packets with probability p. What should you set the transmission probability of B to, in order to achieve your throughput goal?

(Explain your answer in the space below.)

Let B's transmission probability be p_b. The utilization is equal to $p(1-p_b)+(1-p)p_b$. A's throughput is the first term, $p(1-p_b)$, and B's throughput is the second term, $p_b(1-p)$. Using the fact that we want B to get twice the throughput of A, and solving for p_b in terms of p, we find that $p_b = \frac{2p}{1+p}$.

II Reliable Data Transport

In the reliable transport protocols we studied, the receiver sends an acknowledgment (ACK) saying "I got k" whenever it receives a packet with sequence number k. Ben Bitdiddle invents a different method using **cumulative ACKs**: whenever the receiver gets a packet, whether in order or not, it sends an ACK saying "I got every packet up to and including ℓ", where ℓ is the **highest, in-order** packet received so far.

The definition of the window is the same as before: a window size of W means that the maximum number of unacknowledged packets is W. Every time the sender gets an ACK, it may transmit one or more packets, within the constraint of the window size. It also implements a timeout mechanism to retransmit packets that it believes are lost using the algorithm from class and Lab 10.

Network assumptions. The protocol runs over a best-effort network, but *no packet or ACK is duplicated at the network or link layers*.

6. [$5 + 2 =$ 7 points]: The sender sends a stream of new packets according to the sliding window protocol, and in response gets the following cumulative ACKs from the receiver:

1 2 3 4 4 4 4 4 4 4

A. Now, suppose that the sender times out and retransmits the first unacknowledged packet. When the receiver gets that retransmitted packet, what can you say about the ACK, a, that it sends? **(Circle the BEST answer)**

(a) $a = 5$.
(b) $a \geq 5$.
(c) $5 \leq a \leq 11$.
(d) $a = 11$.
(e) $a \leq 11$.

Solution. (b), $a \geq 5$. The reason is that packet 6 could also be lost. It might be tempting to pick (c), but that would be incorrect because the receiver might have received packets $6, 7, \ldots, 11, 12, \ldots$, and the receiver's ACKs for $12, \ldots$ lost, yielding the sequence of ACKs. Now, when the receiver gets 5, the cumulative ACK it sends would be larger than 11. For this reason, choices (d) and (e) are also wrong.

B. Is it possible for the given sequence of cumulative ACKs to have arrived at the sender even when no packets were lost en route to the receiver when they were sent? **(Explain your answer in the space below.)**

Solution. Yes, 5 may be reordered in the network and experiencing a long delay.

7. [4 points]: A little bit into the data transfer, the sender observes the following sequence of cumulative ACKs sent from the receiver:

21 22 23 25 28

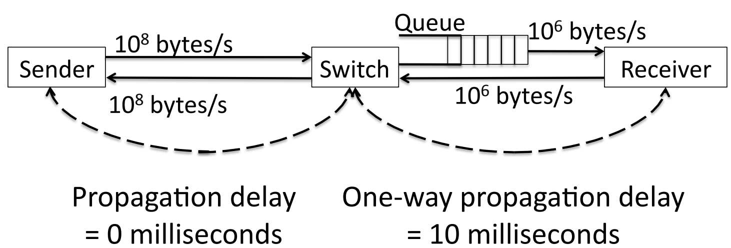
The window size is 8 packets. What packet(s) should the sender transmit upon receiving each of the above ACKs, if it wants to maximize the number of unacknowledged packets?

On getting ACK # → Send ??		On getting ACK # → Send ??	
21	→ 29	22	→ 30
23	→ 31	25	→ 32, 33
28	→ 34, 35, 36		

8. [2 points]: Give one example of a situation where the cumulative ACK protocol gets higher throughput than the sliding window protocol described in class and Lab 10.

Solution. When ACKs are lost, but data packets aren't, the cumulative ACK protocol provides higher throughput than the one we studied in class because the sender may not need to retransmit the packet presumed missing.

Ben decides to use the sliding window transport protocol we studied in 6.02 and implemented in Lab 10 on the network below. The receiver sends **end-to-end ACKs** to the sender. The switch in the middle simply forwards packets in best-effort fashion.



Max queue size = 30 packets
Packet size = 1000 bytes
ACK size = 40 bytes
Initial sender window size = 10 packets

9. [5 points]: The sender's window size is 10 packets. At what approximate rate (in packets per second) will the protocol deliver a multi-gigabyte file from the sender to the receiver? Assume that there is no other traffic in the network and packets can only be lost because the queues overflow.

(Circle the BEST answer)

A. Between 900 and 1000.

B. Between 450 and 500.

C. Between 225 and 250.

D. Depends on the timeout value used.

Explain your answer in the space below for full credit.

Solution. B. The RTT, which is the time taken for a single packet to reach the receiver is about 20 milliseconds, plus the transmission time, which is about 1 millisecond (1000 bytes at a rate of 1 Megabyte/s). Hence, the throughput is 10 packets / 21 milliseconds = 476 packets per second. If one ignored the transmission time, which is perfectly fine given the set of choices, one would estimate the throughput to be about 500 packets per second.

10. [9 points]: You would like to double the throughput of this sliding window transport protocol running on the network shown on the previous page. To do so, you can apply **one** of the following techniques alone:

A. Double the window size.

B. Halve the propagation time of the links.

C. Double the speed of the link between the Switch and Receiver.

For each of the following sender window sizes, list which of the above techniques, **if any, can approximately double the throughput**. If no technique does the job, say "None". There might be more than one answer for each window size, in which case you should list them all for full credit. Note that each technique works in isolation. Use the space below each window size to explain your answer.

1. $W = 10$: A and B.

   When $W = 10$, the throughput is about 476 packets/s. If we double the window size, throughput would double to 952 packet/s. If we reduce the propagation time of the links, throughput would roughly double as well. The new throughput would still be smaller than the bottleneck capacity of 1000 packets/s.

2. $W = 50$: C.

   When $W = 50$, throughput is already 1000 packets/s. At this stage, doubling the window or halving the RTT does not increase the throughput. If we double the speed of the link between the Switch and Receiver, the bottleneck becomes 2000 packets/s. A window size of 50 packets over an RTT of 20 or 21 milliseconds has a throughput of more than 2000 packet/s. Hence, doubling the bottleneck link speed will double the throughput when $W = 50$ packets.

   With a queue size of 30 packets and a window size of 50, the initial window of packets sent back-to-back would indeed cause the queue to overflow. However, that doesn't cause the throughput to drop in the steady state, so for a long data transfer, the throughput will be as described above.

3. $W = 30$: None.

   When $W = 30$, throughput is already 1000 packets/s. Now, if we double the window or halve the RTT, the throughput won't change. An interesting situation occurs when we double the link speed, because the bottleneck link would now be capable of delivering 2000 packets/s. But our window size is 30 and RTT about 20 milliseconds, giving a throughput of about 1500 packets/s (or if we use 21 milliseconds, we get 1428 packet/s). That's an improve of about 50%, far from the doubling we wanted. None of the techniques work.

11. [2 points]: The sender uses an exponentially weighted moving average (EWMA) filter to estimate the smoothed round trip time, srtt, every time it gets an ACK with an RTT sample r.

srtt $\rightarrow \alpha \cdot r + (1 - \alpha) \cdot$ srtt

We would like every packet in a window to contribute a weight of at least 1% to the srtt calculation. As the window size increases, should α increase, decrease, or remain the same, to achieve this goal? (You should be able to answer this question without writing any equations.)

(Explain your answer in the space below.)

Solution. α should decrease. If α is small, the EWMA has more history, and as α becomes larger, the contribution reduces. One can write out a mathematical expression, but it isn't necessary to answer the question.

III Routing and the Network Layer

12. [4 points]: We studied a few principles for designing networks in 6.02.

A. State one significant difference between a circuit-switched and a packet-switched network.

(Explain your answer in the space below.)

In a packet-switched network, packets carry information in the header that tells the switches about the destination. Circuit-switched networks don't carry any destination information in the data frames.

The abstraction provided by a circuit-switched network is that of a dedicated link of a fixed rate; a packet-switched network provides no such guarantee to the communicating end points.

B. Why does topological addressing enable large networks to be built?

(Explain your answer in the space below.)

It reduces the size of the routing tables and the amount of information that must be exchanged in the routing protocol.

13. [8 points]: Eager B. Eaver implements distance vector routing in his network in which the links all have arbitrary positive costs. In addition, there are at least two paths between any two nodes in the network. One node, u, has an erroneous implementation of the integration step: it takes the advertised costs from each neighbor and picks the route corresponding to the minimum advertised cost to each destination as its route to that destination, **without adding the link cost to the neighbor**. It breaks any ties arbitrarily. All the other nodes are implemented correctly.

Let's use the term "correct route" to mean the route that corresponds to the minimum-cost path. Which of the following statements are true of Eager's network?

A. True / False Only u may have incorrect routes to any other node.

B. True / False Only u and u's neighbors may have incorrect routes to any other node.

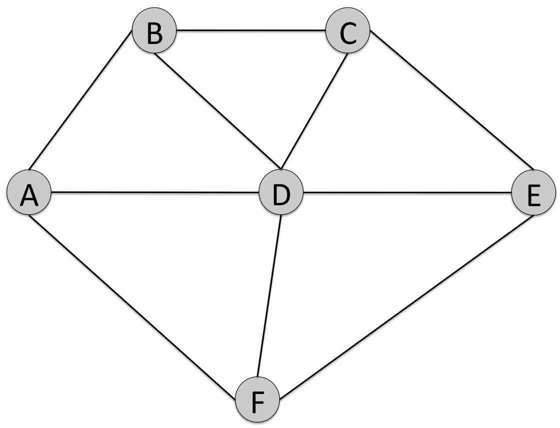
C. True / False In some topologies, all nodes may have correct routes.

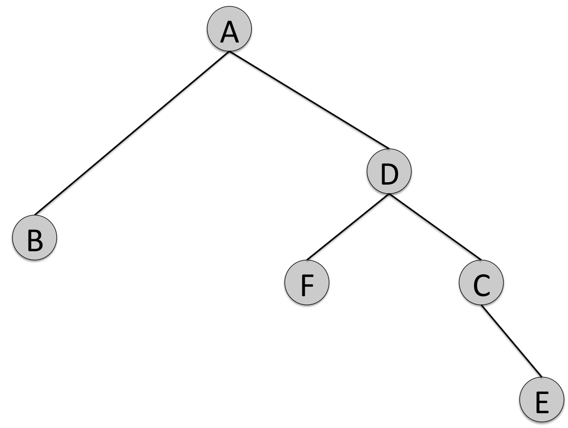
D. True / False Even if no HELLO or advertisements packets are lost and no link or node failures occur, a routing loop may occur.

Solution. A and B are false, C is true, and D is false. A is false because u could propagate an incorrect cost to its neighbors causing the neighbor to have an incorrect route. In fact, u's neighbors could do the same. C is correct; a simple example is where the network is a tree, where there is exactly one path between any two nodes.

D is false; no routing loops can occur under the stated condition. We can reason by contradiction. Consider the shortest path from any node s to any other node t running the flawed routing protocol. If the path does not traverse u, no node on that path can have a loop because distance vector routing without any packet loss or failures is loop-free. Now consider the nodes for which the computed paths go through u; all these nodes are correctly implemented except for u, which means the paths between u and each of them is loop-free. Moreover, the path to u is itself loop-free because u picks one of its neighbors with *smaller* cost, and there is no possibility of a loop.

Alyssa P. Hacker is trying to reverse engineer the trees produced by running Dijkstra's shortest paths algorithm at the nodes in the network shown in the picture **on the left, below**. She doesn't know the link costs, but knows that they are all positive. All link costs are symmetric (the same in both directions). She also knows that there is exactly one minimum-cost path between any pair of nodes in this network.





She discovers that the routing tree computed by Dijkstra's algorithm at node **A** looks like the picture **on the right, above**. Note that the exact order in which the nodes get added in Dijkstra's algorithm is not obvious from this picture.

14. [4 points]: Which of A's links has the highest cost? If there could be more than one, tell us what they are.

(Explain your answer in the space below.)

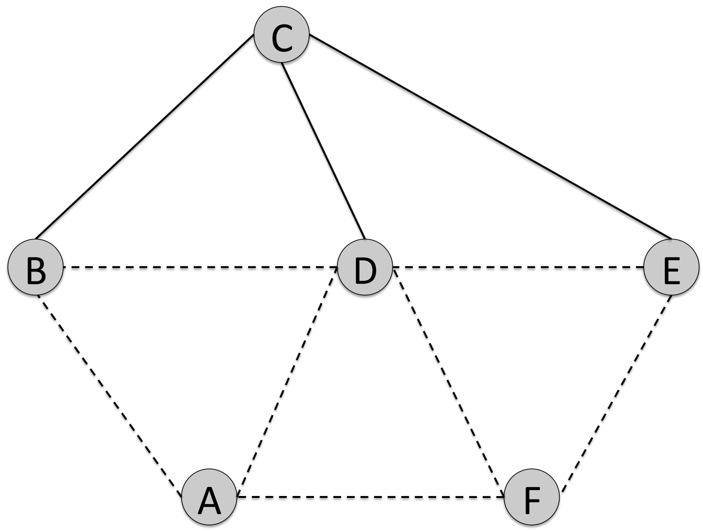
Solution. AF or AB could be the highest cost link; AD clearly has lower cost than AF.

15. [3 points]: Which of A's links has the lowest cost? If there could be more than one, tell us what they are.

(Explain your answer in the space below.)

Solution. Either AB or AD could be the lowest cost link; AF clearly has a higher cost than AD.

Alyssa now inspects node **C**, and finds that it looks like the picture below. She is sure that the bold (not dashed) links belong to the shortest path tree from node **C**, but is not sure of the dashed links.



16. [3 points]: List all the dotted links that are **guaranteed to be** on the routing tree at node **C**.
(Explain your answer in the space below.)

Solution. AD is guaranteed to be on the routing tree because AD is on the shortest path tree from node A. No other dotted link is guaranteed to be on a shortest path from C.

17. [5 points]: List all the dotted links that are **guaranteed not to be** (i.e., surely not) on the routing tree at node **C**.
(Explain your answer in the space below.)

Solution. BD, BA, AF, DE.

IV Information and Huffman Coding

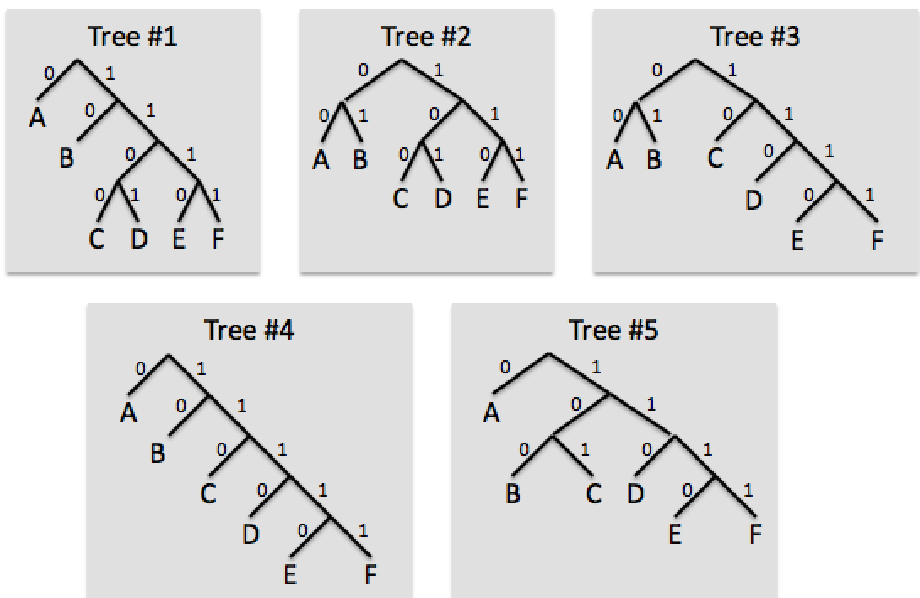
18. [2 points]: I randomly select a letter from the 26-letter alphabet and tell you that my letter is **not** X, Y, or Z. How much information have I told you about my letter? Give a number or an expression.

Solution. Before, $N = 26$, after, $N = 23$. Information received = $\log_2(26/23)$ bits.

19. [2 points]: You are trying to guess a card picked at random from a standard 52-card deck. Sam tells you the card is a spade; Nora tells you it's not an ace; Rita tells you it's a seven. What is the total amount of information about the card given by Sam, Nora, and Rita? Give a number or an expression.

Solution. Observe that you now know the card exactly (it's a 7 of spades). So you've gone down from 52 choices to 1, and have received $\log_2(52/1)$ bits of information.

Consider the five 6-leaf binary trees shown below, each of which diagrams a particular Huffman code for message sequences composed from six symbols A, B, C, D, E, F. Each symbol has an associated probability $p(A), p(B), p(C), p(D), p(E), p(F)$.



20. [3 points]: Which tree or trees are consistent with a Huffman code where $p(A) > 0.5$?

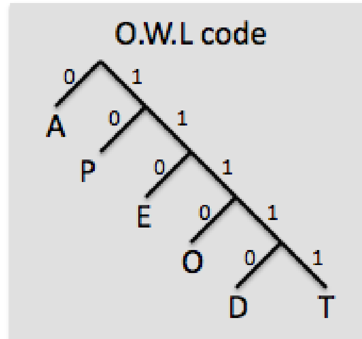
Solution. Trees 1, 4, 5.

21. [3 points]: Which tree or trees are consistent with a Huffman code where all six of the probabilities are equal?

Solution. Tree 2.

The Hogwarts Registrar encodes the results of the O.W.L.s using the variable-length code shown below, next to the table of showing the probability that a student will receive a particular grade.

Grade	p(Grade)
O – outstanding	0.10
E – exceeds expectations	0.15
A – acceptable	0.40
P – poor	0.21
D – dreadful	0.09
T – troll	0.05



22. [2 points]: Give an expression for the amount of information received when learning that a particular grade is passing (i.e., one of O, E, or A)?

(Explain your answer in the space below.)

Solution. $\log_2 \frac{1}{0.1+0.15+0.4} = \log_2(1/0.65)$.

23. [$1+1+2=$ 4 points]: The Registrar is encoding a message containing 1000 O.W.L. grades.

(Explain your answers in the space below.)

A. What is the length of the longest encoded message that might be produced?

Solution. Worst case is all D's and T's, so $1000 \times 5 = 5000$ bits.

B. What is the length of the shortest encoded message that might be produced?

Solution. All A's, so 1000 bits.

C. Give an expression for the expected length of the encoded message.

Solution. $1000(0.4(1) + 0.21(2) + 0.15(3) + 0.1(4) + 0.14(5)) = 2370$ bits.

24. [4 points]: Assuming that grades are encoded one-at-a-time, is the encoding shown above optimal (i.e., no other encoding would produce encoded messages with a shorter expected length)? If yes, explain why. If no, give the binary tree diagram for an optimal variable-length encoding.

(Explain your answer in the space below.)