Name: Staff Solutions

Department of Electrical Engineering and Computer Science

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

6.02 Fall 2011

Quiz III

December 19, 2011

"×" your section	Section	Time	Room	Recitation Instructor	"×" your TA
					☐ Mukul Agarwal
☐	1	10-11	36-112	Paul Ampadu	☐ Jason Cloud
☐	2	11-12	36-112	Sidhant Misra	☐ Shuo Deng
☐	3	12-1	36-112	Sidhant Misra	☐ Lyla Fischer
☐	4	1-2	36-112	Paul Ampadu	☐ Rui Li
☐	5	2-3	26-168	Karl Berggren	☐ Ruben Madrigal
☐	6	3-4	26-168	Karl Berggren	☐ Surapap R.
					☐ Xiawa Wang

There are a lucky **13 questions** (with multiple parts) and **14 pages** in this quiz booklet. Answer each question according to the instructions given. You have **2 hours** to answer the questions.

If you find a question ambiguous, please 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! **And please show your work for partial 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 use the blank sides for answers, make sure to say so!*

Please write your name CLEARLY in the space at the top of this page. NOW, please!

Two two-sided "crib sheets" allowed. No other notes, books, calculators, computers, cell phones, PDAs, information appliances, carrier pigeons carrying messages, etc.!

Do not write in the boxes below

1-2 (x/20)	3-5 (x/25)	6-7 (x/18)	8-10 (x/9)	11 (x/9)	12-13 (x/19)	Total (x/100)

I It may be Little, but it's the Law!

1. [3+7=10 points]: Carrie Coder has set up an email server for a large email provider. The email server has two modules that process messages: the *spam filter* and the *virus scanner*. As soon as a message arrives, the spam filter processes the message. After this processing, if the message is spam, the filter throws out the message. The system sends all non-spam messages immediately to the virus scanner. If the scanner determines that the email has a virus, it throws out the message. The system then stores all non-spam, non-virus messages in the inboxes of users.

Carrie runs her system for a few days and makes the following observations:

1. On average, $\lambda = 10000$ messages arrive per second.
2. On average, the spam filter has a queue size of $N_s = 5000$ messages.
3. $s = 90\%$ of all email is found to be spam; spam is discarded.
4. On average, the virus scanner has a queue size of $N_v = 300$ messages.
5. $v = 20\%$ of all non-spam email is found to have a virus; these messages are discarded.

A. On average, in 10 seconds, how many messages are placed in the inboxes?

(Explain your answer in the space below.)

Solution: Rate of messages into inboxes = 10000 * 0.1 * 0.8 = 800 messages/second, so in 10 seconds, we'll see 8000 messages entering users' inboxes.

B. What is the **average delay** between the arrival of an email message to the email server and when it is ready to be placed in the inboxes? All transfer and processing delays are negligible compared to the queueing delays. Make sure to draw a picture of the system in explaining your answer. **Derive your answer in terms of the symbols given, plugging in all the numbers only in the final step.**

(Explain your answer in the space below.)

Solution: The delay has two parts: the delay in the spam filter and the delay in the scanner. The spam filter delay is given by N_s/λ. The arrival rate to the virus scanner is $\lambda(1-s)$, so the scanner delay is $\frac{N_v}{\lambda(1-s)}$. Hence, the total average delay is

$$\frac{N_s}{\lambda} + \frac{N_v}{\lambda(1-s)}.$$

Plugging in the numbers given, we get $5000/10000 + 300/1000 = 0.8$ seconds, or 800 milliseconds.

II Hunting in (Packet) Pairs

2. [2+1+2+5=10 points]: A sender S and receiver R are connected using a link with an unknown bit rate of C bits per second and an unknown propagation delay of D seconds. At time $t = 0$, S schedules the transmission of a **pair of packets** over the link. The bits of the two packets reach R in succession, spaced by a time determined by C. Each packet has the same known size, L bits.

The last bit of the first packet reaches R at a known time $t = T_1$ seconds. The last bit of the second packet reaches R at a known time $t = T_2$ seconds. As you will find, this packet pair method allows us to estimate the unknown parameters, C and D, of the path.

A. Write an expression for T_1 in terms of L, C, and D.

Solution: $T_1 = D + L/C$.

B. At what time does the **first** bit of the **second** packet reach R? Express your answer in terms of T_1 and one or more of the other parameters given (C, D, L).

(Explain your answer in the space below.)

Solution: At time $T_1 + \frac{1}{C}$ seconds.

C. What is T_2, the time at which the **last** bit of the **second** packet reaches R? Express your answer in terms of T_1 and one or more of the other parameters given (C, D, L).

(Explain your answer in the space below.)

Solution: $T_2 = T_1 + \frac{L}{C}$.

D. Using the previous parts, or by other means, derive expressions for the bit rate C and propagation delay D, in terms of the known parameters (T_1, T_2, L).

(Explain your answer in the space below.)

Solution: We know from Part A that

$$D + L/C = T_1$$

and from Part C that

$$\frac{L}{C} = T_2 - T_1.$$

D and C are unknown. From the second equation (Part C), we can solve for C, giving us

$$C = \frac{L}{T_2 - T_1}.$$

Plugging this value of C into the first equation (Part A), we get

$$D = T_1 - \frac{L}{C} = T_1 - (T_2 - T_1) = 2T_1 - T_2.$$

III Le Big MAC

3. [8 points]: We studied TDMA, Aloha, and CSMA protocols in 6.02. In each statement below, assume that the protocols are implemented correctly.

(Circle True or False for each choice.)

A. True / False TDMA may have collisions when the size of a packet exceeds one time slot.

Solution: 3A: False. A correct TDMA implementation with larger-than-one-slot packet sizes should experience no collisions, as you may recall from PSet 8 Task 1 on TDMA.

B. True / False In Aloha with correct stabilization, two nodes have a certain probability of colliding in a time slot. If they actually collide in that slot, then they will experience a lower probability of colliding with each other when they each retry.

Solution: 3B: True. Each node reduces, usually by one-half, its probability of transmission.

C. True / False In slotted Aloha with stabilization, each node's transmission probability converges to $1/N$, where N is the number of backlogged nodes.

Solution: 3C: False. The probability of transmission at each node varies as nodes succeed and collide. There is no convergence to any specific value.

D. True / False In a network in which all nodes can hear each other, a correctly implemented CSMA protocol will have no collisions when the packet size is larger than one time slot.

Solution: 3D: False, because of reasons discussed in notes, two is that two nodes might find the medium idle at the same time and both transmit, and collide.

4. [4+4=8 points]: Carl Coder implements a simple slotted Aloha-style MAC for his room's wireless network. His room has only two backlogged nodes, A and B. Carl picks a transmission probability of $2p$ for node A and p for node B. Each packet is one time slot long and all transmissions occur at the beginning of a time slot.

A. What value of p maximizes the utilization of this network, **and** what is the maximum utilization?

(Explain your answer in the space below.)

Solution: Utilization is

$$p(1-2p) + 2p(1-p) = 3p - 4p^2$$

which is maximized for $p = \frac{3}{8}$.The maximum utilization is $\frac{9}{16}$.

B. Instead of maximizing the utilization, suppose Carl chooses p so that the throughput achieved by A is **three times** the throughput achieved by B. What is the utilization of his network now?

(Explain your answer in the space below.)

Solution: We want

$$2p(1-p) = 3p(1-2p),$$

which implies that $p = \frac{1}{4}$. The maximum utilization is $2p(1-p) + p(1-2p) = \frac{1}{2}$.

5. [4+5=9 points]: Carl Coder replaces the "send with fixed probability" MAC with one that uses a *contention window* at each node. He configures node A to use a fixed contention window of W and node B to use a fixed contention window of $2W$. Before a transmission, each node independently picks a random integer t uniformly between 1 and its contention window value, and transmits a packet t time slots from now. Each packet is one time slot long and all transmissions occur at the beginning of a time slot.

A. What is the probability that A and B will collide the next time they each transmit?

(Explain your answer in the space below.)

Solution: This is the probability that A and B both pick the same number. This can happen in W (equally likely) ways, because A has only that many possible numbers to pick. The *total* number of possibile choices for A and B is $W.2W = 2W^2$ (again, all equally likely). So the desired probability is simply the ratio of these two numbers, $W/(2W^2) = 1/(2W)$.

To see this another way, the probability of a collision is the probability of the following:
(A picks 1 and B picks 1) OR (A picks 2 and B picks 2)) OR .... OR (A picks W and B picks W)
Since the events in parentheses are mutually exclusive, the desired probability is the *sum* of the individual ones. And since A and B pick independently, the probability of an event such as (A picks k and B picks k) is the *product* of the individual probabilities. It follows that the desired probability is $W.(1/W).(1/(2W)) = 1(2W)$.

B. Suppose there is no collision at the next packet transmission. Calculate the probability that A will transmit before B? (It may be useful to apply the formula $\sum_{i=1}^{n} i = n(n+1)/2$.)

(Explain your answer in the space below.)

Solution: This is the probability that A picks a number less than the one picked by B, *given* that they don't pick the same number. The number of ways of **not** picking the same number is $2W^2 - W = W(2W-1)$.
The number of ways for A to pick a number less than the one picked by B is:
0 if B picks 1; 1 if B picks 2; 2 if B picks 3; ... ,W if B picks $W+1$; and stays at W if B picks $W+k$ for $k = 2, ..., W$. So the total number of ways is

$$(1 + 2 + \cdots + W) + W(W-1) = W(3W-1)/2 \, .$$

The desired probability is the ratio of the two numbers we have computed, i.e.,

$$[W(3W-1)/2]/[W(2W-1)] = (3W-1)/[2(2W-1)] \, .$$

Another interpretation of the problem (and actually the one we intended, though we didn't phrase it that way!) would be that it is asking for the probability A will succeed at transmitting before B at the next packet transmission. The computation here is essentially the same as before, except that the denominator is the set of *all* possible outcomes, without conditioning on there being no collision at the next packet transmission. So the denominator now is just $2W^2$ rather than $W(2W-1)$, and the desired probability is

$$[W(3W-1)/2]/(2W^2) = (3W-1)/(4W) \, .$$

IV Roto-Routing

Eager B. Eaver implements the distance-vector protocol studied in 6.02, **but on some of the nodes**, his code sets the cost and route to each advertised destination D **differently**:

> Cost to $D = \min(\text{advertised\_cost})$ heard from each neighbor.
> Route to D = link to a neighbor that advertises the minimum cost to D.

Every node in the network periodically advertises its vector of costs to the destinations it knows about to all its neighbors. All link costs are positive.

At each node, a route for destination D is **valid** if packets using that route will eventually reach D.

At each node, a route for destination D is **correct** if packets using that route will eventually reach D along some minimum-cost path.

Assume that there are no failures and that the routing protocol has converged to produce *some* route to each destination at all the nodes.

6. [3+3+2=8 points]: Circle **True** or **False** for each statement below, providing a **brief explanation** for your choice below each statement. Assume a network in which **at least two of the nodes** (and possibly all of the nodes) run Eager's modified version of the code, while the remaining nodes run the method discussed in 6.02.

A. True / False There exist networks in which some nodes will have invalid routes.

Solution: A. The most correct answer was True, given how the problem was stated. Consider the example where A, B, and C are connected in a triangle with equal weight edges. If A and C are running the new algorithm, A will, at some point in time, hear an advertisement of cost zero from B and C. Similarly, C will hear a zero cost path coming from A and B. The problem does not give any indication of how ties are broken, so it is possible that C will route packets to B through A, and A will route packets to B through C, causeing a routing loop: an invalid path. However, if ties are broken in the way described in class - where preference is given to a routing table entry that already exists, no routing loops will ever happen, because every advertisement will come from a node that has a valid (although not necessarily min-cost) path across a link that has a finite communication time (because otherwise it would not hear the advertisement.). Full credit was given to a response of false with this explanation. (This was actually the answer that we were looking for).

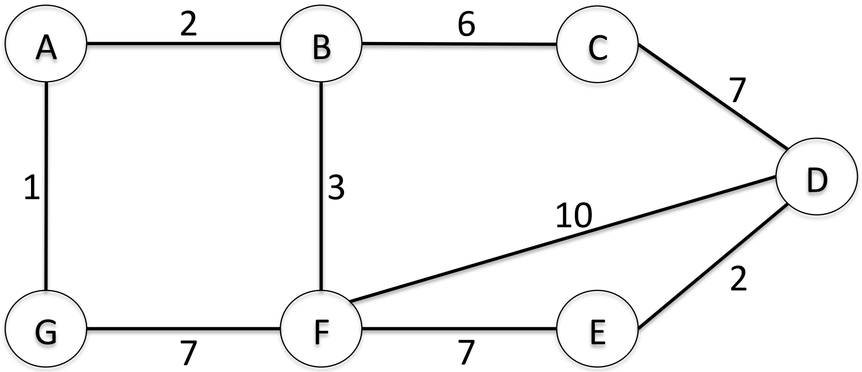
B. True / False There exist networks in which some nodes will **not** have correct routes.

Solution: B. True. For nodes to have incorrect routes, packets will eventually reach D, but not at a minimum cost. Since some nodes do not account for their own link cost, incorrect advertised costs will propagate through the network, leading to incorrect routes. There are several examples; one is a simple triangle A, B, D where the cost from A to D is 100, the cost from A to B is 1 and the cost from B to D is 1. A will pick the direct link to D as its route, but the correct route is the link to B, corresponding to the path ABD.

C. True / False There exist networks in which **all** nodes will have correct routes.

Solution: C. True. A network where there is only one path between any two nodes will have a correct routes (for example, where the network is a tree). There are other examples too; for instance, imagine a connected network where the set of nodes with the incorrect DV update have no links between each other at all. In such networks, the routes everywhere will be correct.

We can't have a 6.02 quiz without our friend Ben Bitdiddle! Help Ben answer these questions about the distance-vector protocol he runs on the network shown in the figure below. The link costs are shown near each link. Ben is interested in minimum-cost routes to destination node D.



Each node sends a distance-vector advertisement to all its neighbors at times $0, T, 2T, \ldots$. Each node integrates advertisements at times $T/2, 3T/2, 5T/2, \ldots$. You may assume that all clocks are synchronized. The time to transmit an advertisement over a link is negligible. There are no failures or packet losses.

7. [5+5=10 points]: Use the definitions of a **valid** route and a **correct** route from the previous page to answer the following questions. **You must explain your answers.**

A. At what time will **all** nodes have integrated a valid route to D into their routing tables? What node is the **last one** to integrate a valid route to D? Answer both questions.

Solution: The principle to use here is that the node whose hop-count is largest from D will be the last node to have a valid route to D, because it will take that many steps in the DV advertisement steps for that node to hear about D. In this case, that is node A, and the time required is three DV advertisement rounds plus $T/2$ to integrate, which is a total time of $2T + T/2 = 5T/2$.

B. At what time will **all** nodes have integrated a correct (minimum-cost) route to D into their routing tables? What node is the **last one** to integrate a correct route to D? Answer both questions.

Solution: The principle here is that the node with the largest number of hops in its correct (minimum-cost) path to D will be the last node to get a correct route. The time taken will depend on the number of hops in that path, because each DV advertisement round provides a minimum-cost path whose length has one more hop. In this case, that is node G, and the minimum-cost path has 5 hops, so the desired time is $4T + T/2 = 9T/2$.

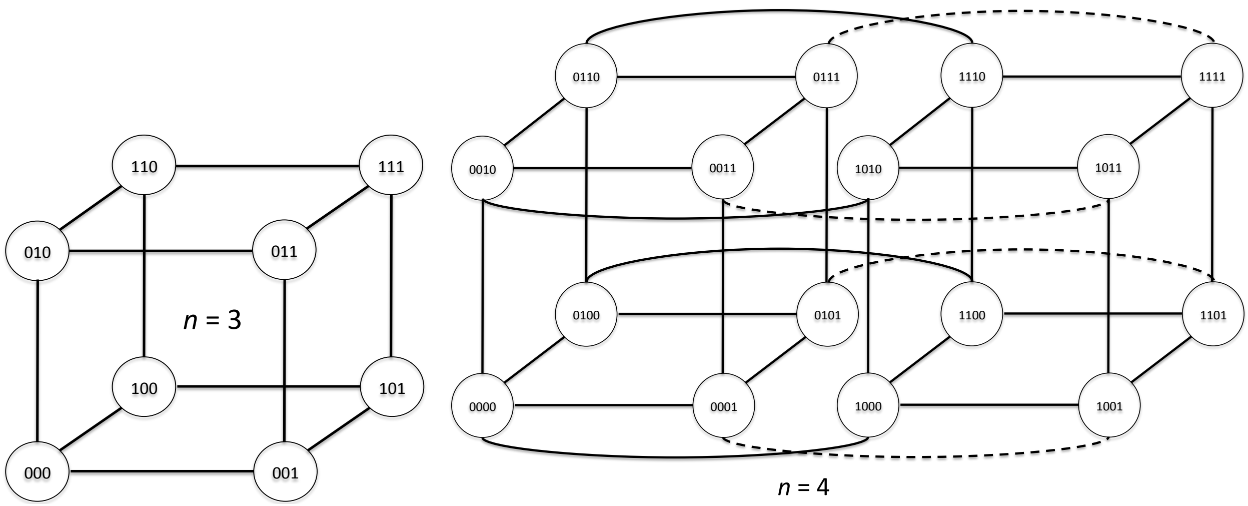
Please see the next page for the more gory details!

In more detail, the table below shows the route advertisements at each node to node D at each time:

t	A	B	C	D	E	F	G
0	∞,B; ∞,G	∞,A; ∞,C; ∞,F	∞,B; 7,D	-	2,D; ∞,F	∞,B; 10,D; ∞,E; ∞,G	∞,A; ∞,F
$T/2$	∞	∞	7,D	-	2,D	10,D	∞
T	∞,B; ∞,G	∞,A; 13,C; 13,F	∞,B; 7,D	-	2,D; 17,F	∞,B; 10,D; 9,E; ∞,G	∞,A; 17,F
$3T/2$	∞	13,C	7,D	-	2,D	9,E	17,F
$2T$	15,B; 18,G	∞,A; 13,C; 12,F	19,B; 7,D	-	2,D; 16,F	16,B; 10,D; 9,E; 24,G	∞,A; 16,F
$5T/2$	15,B	12,F	7,D	-	2,D	9,E	16,F
$3T$	14,B; 17,G	17,A; 13,C; 12,F	18,B; 7,D	-	2,D; 16,F	15,B; 10,D; 9,E; 23,G	16,A; 16,F
$7T/2$	14,B	12,F	7,D	-	2,D	9,E	16,F
$4T$	14,B; 17,G	16,A; 13,C; 12,F	18,B; 7,D	-	2,D; 16,F	15,B; 10,D; 9,E; 23,G	15,A; 16,F
$9T/2$	14,B	12,F	7,D	-	2,D	9,E	15,A
$5T$	14,B; 16,G	16,A; 13,C; 12,F	18,B; 7,D	-	2,D; 16,F	15,B; 10,D; 9,E; 22,G	15,A; 16,F
$11T/2$	14,B	12,F	7,D	-	2,D	9,E	15,A

V Hyper-Routing

The hypercube is an interesting network topology. An n-dimensional hypercube has 2^n nodes, each with a unique n-bit address. Two nodes in the hypercube are connected with a link if, and only if, their addresses have a Hamming distance of 1. The picture below shows hypercubes for $n = 3$ and 4. The solid and dashed lines are the links. We are interested in link-state routing over hypercube topologies.



8. [2 points]: Suppose $n = 4$. Each node sends a link-state advertisement (LSA) periodically, starting with sequence number 0. All link costs are equal to 5. Node 1000 discovers that its link to 1001 has failed. There are no other failures. What are the contents of the **fourth LSA** originating from node 1000?

Solution: The fourth LSA is $[origin\_addr = 1000, seqnum = 3, (0000, 5), (1100, 5), (1010, 5)]$.

9. [2+2=4 points]: Suppose $n = 4$. Three of the links at node 1000, including the link to node 1001, fail. No other failures or packet losses occur.

A. How many distinct copies of any given LSA originating from node 1000 does node 1001 receive?

Solution: 3. For any LSA originating from node 1000 that are flooding in the network, node 1001 receives **three** distinct copies from its connecting neighbors: one from each working link.

B. How many distinct copies of any given LSA originating from node 1001 does node 1000 receive?

Solution: 1. Since node 1000 only has one working link, any LSAs originating from the other nodes in the network can only come from the neighbor along that link. Hence, node 1000 just receives one copy of any given LSA originating from node 1001.

10. [3 points]: Suppose $n = 3$ and there are no failures. Each link has a **distinct**, **positive**, **integral** cost. Node 000 runs Dijkstra's algorithm (breaking ties arbitrarily) and finds that the minimum-cost path to 010 has 5 links on it. What can you say about the cost of the direct link between 000 and 010? **Explain your answer.**

Solution: The cost of the direct link between 000 and 010 must be ≥ 15. Because the minimum cost of link from 000 to 111 has 5 links on it and each of the links has a distinct integer value, the minimum possible cost for the total path is $1 + 2 + 3 + 4 + 5 = 15$. The direct link from 000 to 010 has to be at least 15 for the routing to avoid this link. The reason for "at least" is that ties are broken arbitrarily. The answer "≥ 16 with a correct explanation is also acceptable.

VI Stop, Wait, and Tell Me...

Alyssa P. Hacker sets up a wireless network in her home to enable her computer ("client") to communicate with an Access Point (AP). The client and AP communicate with each other using a **stop-and-wait** protocol.

The data packet size is 10000 bits. The total round-trip time (RTT) between the AP and client is equal to 0.2 milliseconds (that includes the time to process the packet, transmit an ACK, and process the ACK at the sender) **plus** the transmission time of the 10000 bit packet over the link.

Alyssa can configure two possible transmission bit rates for her link, with the following properties:

Bit rate	Bi-directional packet loss probability	RTT
10 Megabits/s	$1/11$	____
20 Megabits/s	$1/4$	____

11. [3+5+1=9 points]: Alyssa's goal is to select the bit rate that provides the higher throughput for a stream of packets that need to be delivered reliably between the AP and client using stop-and-wait. For both bit rates, the **retransmission timeout (RTO) is 2.4 milliseconds**.

A. What is the round-trip time (RTT) for each bit rate? Calculate it below (show your work) and enter the RTT values **in milliseconds** in the table above.

Solution:

$$RTT = 0.2 \text{ ms} + \frac{10000 \text{ bits}}{\text{Bit Rate [in Mbps]}} \cdot \frac{1000 \text{ ms}}{1 \text{ second}}$$

That works out to 1.2 ms for 10 Megabits/s and 0.7 ms for 20 Megabits/s.

B. For each bit rate, calculate the **expected time**, in milliseconds, to successfully deliver a packet and get an ACK for it. **Show your work.**

Solution: As described in Chapter 20 and multiple practice & PSet problems,

$$\text{Expected time } T = RTT + \frac{l}{1-l} RTO.$$

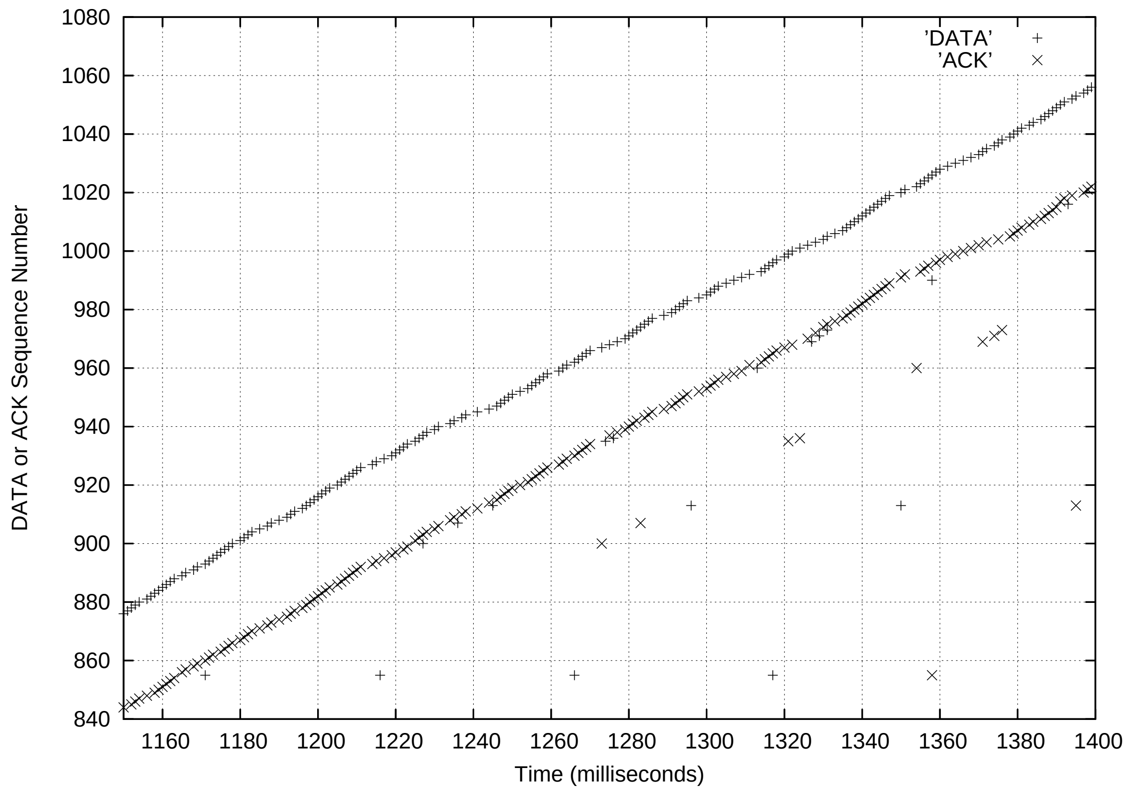
The expected times are 1.44 ms for 10 Megabits/s and 1.5 ms for 20 Megabits/s.

C. Using the above calculations, which bit rate would you choose to achieve Alyssa's goal?

Solution: We would choose the first rate, 10 Megabits/s, because it offers a better expected throughput $\frac{1}{T}$ (where T was found in part B).

VII Slip Slidin' Away

Annette Werker correctly implements the fixed-size sliding window protocol developed in 6.02. She instruments `ReliableSenderNode` to store the time at which each DATA packet is sent and the time at which each ACK is received. A snippet of the DATA and ACK traces from an experiment is shown in the picture below. Each $+$ is a DATA packet transmission, with the x-axis showing the transmission time and the y-axis showing the sequence number. Each $\times$ is an ACK reception, with the x-axis showing the ACK reception time and the y-axis showing the ACK sequence number. All DATA packets have the same size.



12. [11 points]: Answer the following questions, **providing a brief explanation** for each one.

A. Estimate any one sample round-trip time (RTT) of the connection. **Also show it on the picture.**

Solution: An RTT sample is the horizontal distance between a $+$ and the corresponding $\times$ with no intervening retransmissions. One sample RTT, for sequence number 940, is about 45 ms.

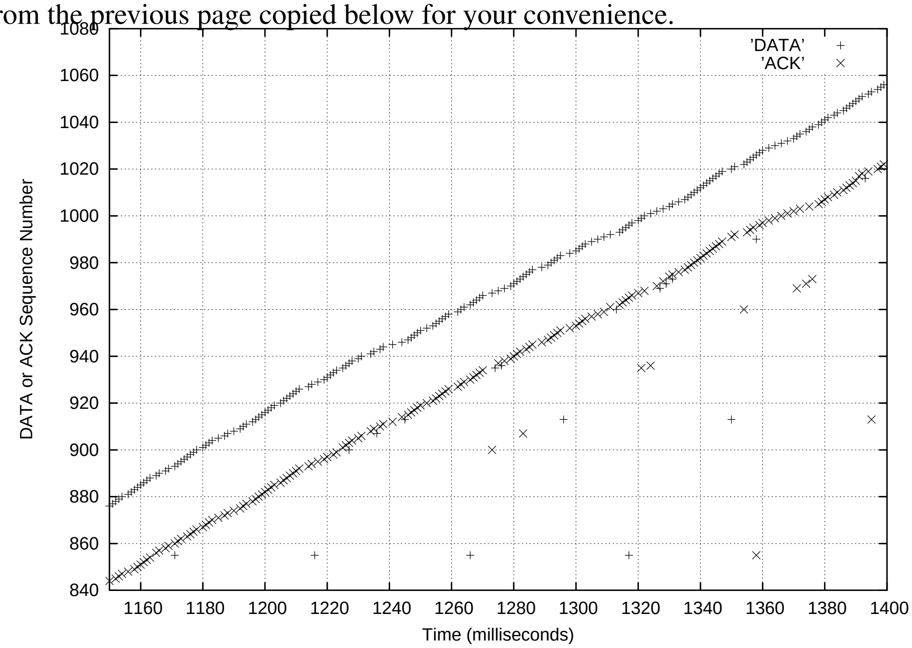
B. On the picture, circle DATA packet **re**transmissions for four different sequence numbers.

Solution: For a given sequence number, retransmissions are the second (or third, or fourth) appearance of $+$. For example, any repeated instance of sequence numbers 855, 900, 907, 913, 935, 936, and so on are all retransmissions.

C. Some DATA packets in this trace may have incurred **more than one re**transmission? **On the picture**, draw a square around one such retransmission.

Solution: Sequence number 855 is an example.

Picture from the previous page copied below for your convenience.



D. Which of these is the **best estimate** of the sender's window size? **Also show it on the picture.**

(a) 20.

(b) 28.

(c) 36.

(d) 44.

Solution: (c) 36. The window size is the vertical distance between $+$ and $\times$. For example, the window size estimated from transmission with sequence number 900) and acknowledgement (864) is $900 - 864 = 36$ packets.

E. Which of these is the **best estimate** of the throughput of the connection in packets per second? Give a one-line explanation of how you estimated it.

(a) ≈ 0.7.

(b) ≈ 500.

(c) ≈ 750.

(d) ≈ 1500.

Solution: (c) 750. Throughput is the slope of the ACK curve. This slope works out to somewhere between 720 and 750 packets per second; 750 if you do the calculation carefully.

F. Considering only sequence numbers > 880, which of these is the **best estimate** of the packet loss rate experienced by DATA packets? Give a one-line explanation of how you estimated it.

(a) $\approx 1.75\%$.

(b) $\approx 3.5\%$.

(c) $\approx 7\%$.

(d) $\approx 14\%$.

Solution: (c) 7%. Each packet loss creates a retransmission. For sequence numbers 880 and greater, there are 13 retransmissions. Hence, the loss rate is $\frac{13}{1057-880+13} \approx 7\%$.

Consider the same setup as in the previous two pages. Suppose the window size for the connection is equal to twice the bandwidth-delay product of the network path.

13. [2+3+3=8 points]: For each change to the parameters of the network path or the sender given below, explain if the connection's **throughput** (not utilization) will **increase**, **decrease**, or **remain the same**. In each statement, nothing other than what is specified in that statement changes. **Explain your answers.**

A. The packet loss rate, ℓ, decreases to $\ell/3$.

Solution: The throughput will **increase** (by a little). Since the channel is completely saturated because the window size is $2 \cdot B \cdot RTT_{min}$ and larger than the bandwidth-delay product, the utilization is $1 - l$. If l decreases, the throughput increases.

B. The minimum value of the RTT, R, increases to $1.8R$.

Solution: The resulting bandwidth-delay product increases by a factor of 1.8, but is still smaller than the window size, so the throughput will **remain the same**.

C. The window size, W, decreases to $W/3$.

Solution: The throughput will **decrease** because the window size will now be strictly smaller than the bandwidth-delay product, and the bottleneck link will no longer be saturated.

FIN

Have a great holiday season!