Name: ***These R. Solutions***

Department of Electrical Engineering and Computer Science

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

6.02 Fall 2010

Quiz III

December 16, 2010

"×" your section	Section	Time	Room	Recitation Instructor
☐	1	10-11	36-112	Tania Khanna
☐	2	11-12	36-112	Tania Khanna
☐	3	12-1	36-112	George Verghese
☐	4	1-2	36-112	George Verghese
☐	5	2-3	26-168	Alexandre Megretski
☐	6	3-4	26-168	Alexandre Megretski

There are **22 questions** (some with multiple parts) and **19 pages** in this quiz booklet. Answer each question according to the instructions given. You have **3 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!

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

Do not write in the boxes below

1-4 (x/13)	5-9 (x/20)	10-11 (x/20)	12-14 (x/14)	15-17 (x/13)	18-22 (x/20)	Total (x/100)

I Switching Places

1. [4 points]: Annette Werker has developed a new switch. In this switch, 10% of the packets are processed on the "slow path", which incurs an average delay of 1 millisecond. All the other packets are processed on the "fast path", incurring an average delay of 0.1 milliseconds. Annette observes the switch over a period of time and finds that the average number of packets in it is 19. What is the average rate, in packets per second, at which the switch processes packets?

(Explain your answer in the space below.)

Solution: Rate $= 19000/(0.1 \cdot 1 + .9 \cdot 0.1) = 19000/(0.19) = 100000$ packets/s.

2. [3 points]: Alyssa P. Hacker designs a switch for a *circuit-switched network* to send data on a 1 Megabit/s link using *time division multiplexing* (TDM). The switch supports a maximum of 20 different simultaneous conversations on the link, and any given sender transmits data in frames of size 2000 bits. Over a period of time, Alyssa finds that the average number of conversations simultaneously using the link is 10. The switch forwards a data frame sent **by a given sender** every δ seconds according to TDM. Determine the value of δ.

(Explain your answer in the space below.)

Solution: 2000 bits / 1 Megabit/s is 2 milliseconds of time. 20 conversations in the TDM schedule implies that $\delta = 20 \cdot 2 = 40 milliseconds$.

3. [3 points]: Louis Reasoner implements the link-state routing protocol discussed in 6.02 on a best-effort network with a non-zero packet loss rate. In an attempt to save bandwidth, instead of sending link-state advertisements periodically, each node sends an advertisement *only if* one of its links fails or when the cost of one of its links changes. The rest of the protocol remains unchanged. Will Louis' implementation always converge to produce correct routing tables on all the nodes?

(Explain your answer in the space below.)

Solution: No, because the LSA could be lost on all of the links connected to some one node (or more than one node), causing that node to not necessarily have correct routes.

In fact, as one student pointed out in her answer, this protocol doesn't converge even if packets are not lost. Consider a network where the failure of one link disconnects the network into two connected components, each with multiple nodes and links. Suppose the cost of one or more of the links in some component changes. When the network heals because the failed link recovers, the previously disconnected component will not have the correct link costs for one or more links in the other component. However, its routes will still be correct because all paths to the other component go via the failed link. However, if we were to add another link between the two components at this time, the routing would never converge correctly.

Other answers that people wrote included "doesn't converge when new nodes are added" or "doesn't converge when new links are added". Those answers aren't as elegant, though we gave them points; the reason is that the addition of new nodes or links is essentially the same as changing link costs from non-existent to some value. Similarly for removing nodes – that's not an adequate answer, though we gave credit for it.

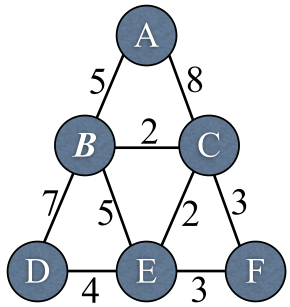
4. [3 points]: Consider a network implementing minimum-cost routing using the distance-vector protocol. A node, S, has k neighbors, numbered 1 through k, with link cost c_i to neighbor i (all links have symmetric costs). Initially, S has no route for destination D. Then, S hears advertisements for D from each neighbor, with neighbor i advertising a cost of p_i. The node integrates these k advertisements. What is the cost for destination D in S's routing table after the integration?

(Explain your answer in the space below.)

Solution: This question asks for the update rule in the Bellman-Ford integration step. The cost in S's routing table for D should be set to $\min_i\{c_i + p_i\}$.

II Get Shorty

Consider the network shown in the picture below. Each node implements Dijkstra's shortest paths algorithm using the link costs shown in the picture.



5. [4 points]: Initially, node **B**'s routing table contains only one entry, for itself. When **B** runs Dijkstra's algorithm, in what order are nodes added to the routing table? **List all possible answers.**

Solution: **B,C,E,A,F,D** and **B,C,E,F,A,D**.

6. [6 points]: Now suppose the link cost for one of the links changes but all costs remain non-negative. For each change in link cost listed below, **state whether it is possible for the *route* at node B** (i.e., the link used by **B**) for any destination to change, **and if so, name the destination(s) whose routes may change**.

A. The cost of link(A, C) increases: Solution: **No effect.** The edge AC is not in any shortest path.

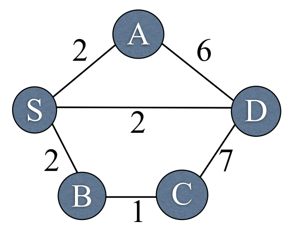
B. The cost of link(A, C) decreases: Solution: **Can affect route to A**. If $cost_{AC} \leq 3$, then we can start using this edge to go to A instead of the edge BA.

C. The cost of link(B, C) increases: Solution: **Can affect route to C,F,E**. If $cost_{BC} \geq 7$, then we can use BE-EC to go to C instead of BC. If $cost_{BC} \geq 5$, then we can use BE-EF to go to F. If $cost_{BC} \geq 3$, can use BE to go to B.

D. The cost of link(B, C) decreases: Solution: **Can affect route to D**. If $cost_{BC} \leq 1$, then we can use BC-CE-ED to go to D instead of BD.

III Don't Lie to Me

Alyssa P. Hacker implements the 6.02 distance-vector protocol on the network shown below. Each node has its own local clock, which may not be synchronized with any other node's clock. Each node sends its distance-vector advertisement every 100 seconds. When a node receives an advertisement, it immediately integrates it. The time to send a message on a link and to integrate advertisements is negligible. No advertisements are lost. There is no HELLO protocol in this network.



7. [5 points]: At time 0, all the nodes **except** D are up and running. At time 10 seconds, node D turns on and immediately sends a route advertisement for itself to all its neighbors. What is the *minimum time* at which each of the other nodes is **guaranteed** to have a correct routing table entry corresponding to a minimum-cost path to reach D? Justify your answers.

Node S: Solution: 10 seconds.

Node A: Solution: 110 seconds.

Node B: Solution: 110 seconds.

Node C: Solution: 210 seconds.

Solution: At time t = 10, D advertises to S, A, and C. They integrate this advertisement into their routing tables, so that cost(S, D) = 2, cost(A, D) = 2, cost(C, D) = 7. Note that only S's route is correct. In the worst case, we wait 100s for the next round of advertisements. So at time t = 110, S, A, and C all advertise about D, and everyone integrates. Now cost(A, D) = 4 (via S), cost(B, D) = 4 (via S), and cost(C, D) = 7 still. A and B's routes are correct; C's is not. Finally, after 100 more seconds, another round of advertisements is sent. In particular, C hears about B's route to D, and updates cost(C, D) = 5 (via B).

8. [2 points]: If every node sends packets to destination D, and to no other destination, which link would carry the most traffic?

Solution: $S \rightarrow D$. Every node's best route to D is via S.

Alyssa is unhappy that one of the links in the network carries a large amount of traffic when all the nodes are sending packets to D. She decides to overcome this limitation with Alyssa's Vector Protocol (AVP). In AVP, S *lies*, advertising a "path cost" for destination D that is *different* from the sum of the link costs along the path used to reach D. All the other nodes implement the standard distance-vector protocol, not AVP.

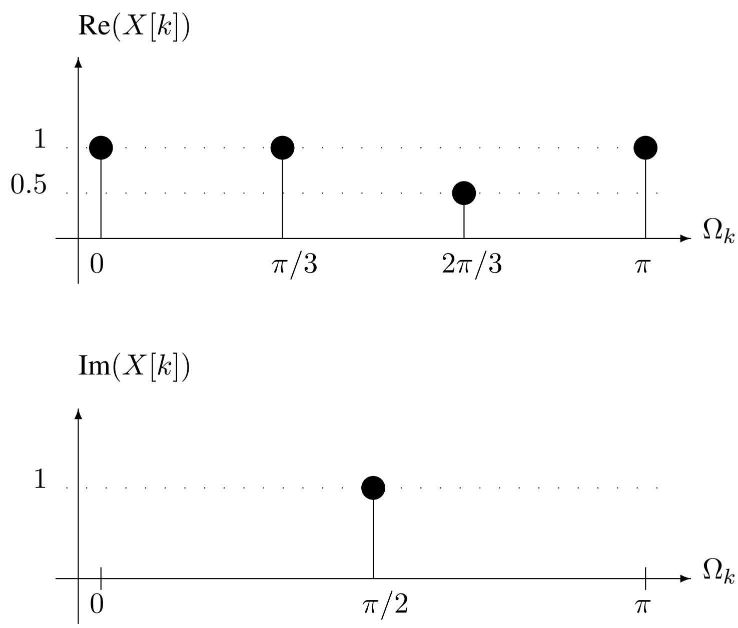
9. [3 points]: What is the *smallest* numerical value of the cost that S should advertise for D along each of its links, to **guarantee** that only its own traffic for D uses its direct link to D? Assume that all advertised costs are integers; if two path costs are equal, one can't be sure which path will be taken.

(Explain your answer in the space below.)

Solution: 7. S needs to advertise a high enough cost such that everyone's path to D via S will no longer be the best path. In particular, since B's cost to D *without* going through S is the highest (8), S must advertise a cost so that linkcost(B, S) + advertisedcost(S, D) > 8. Hence, S advertises a cost of 7.

IV Real and Imaginary Friends

10. [8 points]: The figure below shows the real and imaginary parts of all *non-zero* Fourier series coefficients $X[k]$ of a real periodic discrete-time signal $x[n]$, for frequencies $\Omega_k \in [0, \pi]$. Here $\Omega_k = k(2\pi/N)$ for some fixed even integer N, as in all our analysis of the discrete-time Fourier series (DTFS), but the plots below only show the range $0 \le k \le N/2$. (If you wish to be reminded of the specific formulas associated with the DTFS, see the next page.)



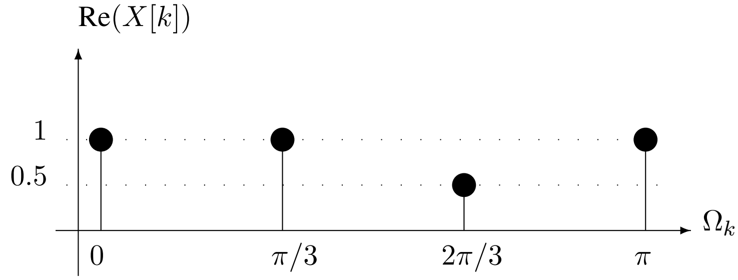
A. Find all *non-zero* Fourier series coefficients of $x[n]$ at Ω_k in the interval $[-\pi, 0)$, i.e., for $-(N/2) \le k < 0$. Give your answer in terms of careful and fully labeled plots of the real and imaginary parts of $X[k]$ **in the space below** (following the style of the figure above).

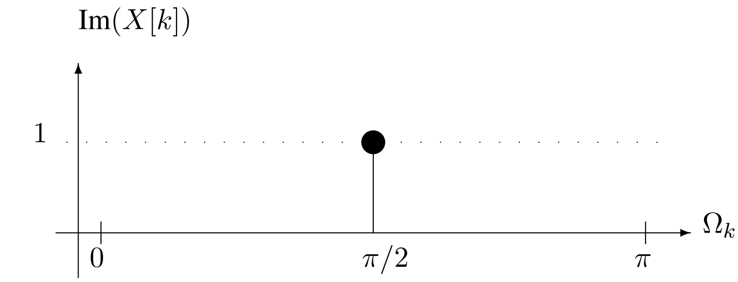
Solution: On the real graph: 1 at $\Omega = -\pi$, 0.5 at $\Omega = -2\pi/3$, 1 at $\Omega = -\pi/3$.
On the imaginary graph: -1 at $\Omega = -\pi/2$. This is based on the observation that for a real-valued signal, the Fourier coefficients must be symmetric, i.e. even for the real part, odd for the imaginary part.

B. Find the period of $x[n]$, i.e., the smallest integer T for which $x[n + T] = x[n]$, for all n.

Solution: Even though we do not knbow N, we can deduce that $T = 12$. This is because the least common multiple of $6 = 2\pi/\frac{\pi}{3}$, $4 = 2\pi/\frac{\pi}{2}$, $3 = 2\pi/\frac{2\pi}{3}$ and $2 = 2\pi/\pi$ is 12.

Picture from the previous page repeated for convenience.





C. For the frequencies $\Omega_k \in [0, \pi]$, find all non-zero Fourier series coefficients of the signal $x[n-6]$ obtained by delaying $x[n]$ by 6 samples.

Solution: When a complex exponent $Ae^{j\Omega n}$ passes through an LTI filter, it gets multiplied by $H(e^{j\Omega})$, where $H(\cdot)$ is the frequency response of the filter. Since "delay by 6 samples" is an LTI filter with frequency response $H(e^{j\Omega}) = e^{-j6\Omega}$, the Fourier coefficients at $\Omega = 0$, $\Omega = \pi/3$, $\Omega = \pi/2$, $\Omega = 2\pi/3$, and $\Omega = \pi$ will be multiplied by $e^{-j6\cdot 0} = 1$, $e^{-j6\cdot\pi/3} = 1$, $e^{-j6\cdot\pi/2} = -1$, $e^{-j6\cdot 2\pi/3} = 1$, and $e^{-j6\cdot\pi} = 1$ respectively, the resulting Fourier coefficients will be:

- On the real graph: 1 at $\Omega = 0$, $\Omega = \pi/3$, $\Omega = 2\pi/3$, and $\Omega = \pi$.
- On the imaginary graph: -1 at $\Omega = \pi/2$.

For reference, we remind you that a signal $x[n]$ over the time interval $[0, N-1]$ (where N is even, and $N/2$ is denoted by K) can be written as a DTFS:

$$x[n] = \sum_{k=-K}^{K-1} X[k]\, e^{j\Omega_k n} ,$$

where $\Omega_k = k(2\pi/N)$, and the Fourier coefficients $X[k]$ are given by

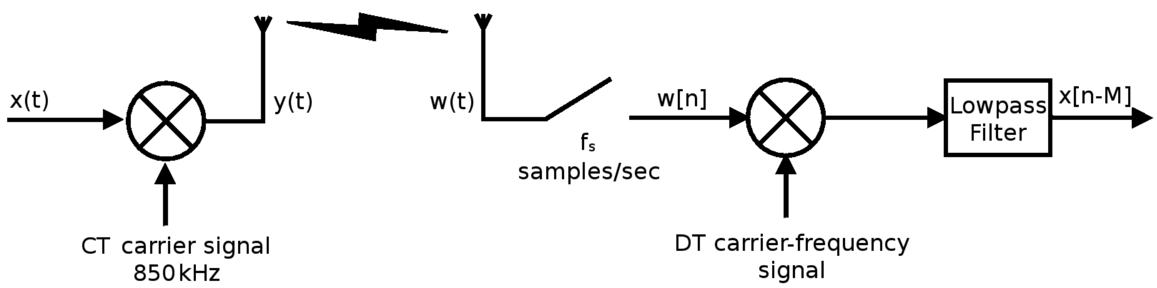
$$X[k] = \frac{1}{N} \sum_{n=0}^{N-1} x[n]\, e^{-j\Omega_k n} .$$

V All We Hear Is... One AM Radio Station

The Boston sports radio station WEEI AM ("amplitude modulation") broadcasts on a carrier frequency of 850 kHz, so its continuous-time (CT) carrier signal can be taken to be $\cos(2\pi \times 850 \times 10^3 t)$, where t is measured in seconds. Denote the CT audio signal that's modulated onto this carrier by $x(t)$, so that the CT signal transmitted by the radio station is

$$y(t) = x(t)\cos(2\pi \times 850 \times 10^3 t) \ , \tag{1}$$

as indicated schematically on the left side of the figure below.



We use the symbols $y[n]$ and $x[n]$ to denote the discrete-time (DT) signals that would have been obtained by respectively sampling $y(t)$ and $x(t)$ in Eq.(1) at f_s samples/sec; more specifically, the signals are sampled at the discrete time instants $t = n(1/f_s)$. Thus

$$y[n] = x[n]\cos(\Omega_c n) \tag{2}$$

for an appropriately chosen value of the angular frequency Ω_c. Assume that $x[n]$ is periodic with some period N, and that $f_s = 2 \times 10^6$ samples/sec.

11. [12 points]: Answer the following questions, explaining your answers in the space provided.

A. Determine the value of Ω_c in Eq.(2), restricting your answer to a value in the range $[-\pi, \pi]$. (You can assume in what follows that the period N of $x[n]$ is such that $\Omega_c = 2k_c\pi/N$ for some integer k_c; this is a detail, and needn't concern you unduly.)

Solution: Putting $t = n/f_s$ in Eq.(1), we get

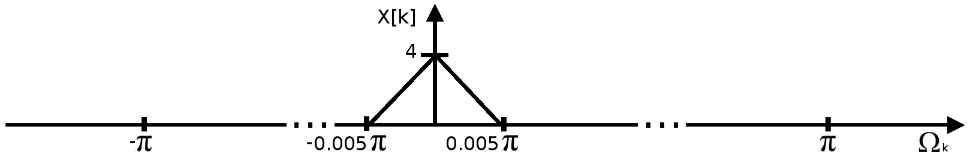
$$\begin{aligned} y[n] &= y(n/f_s) = x(n/f_s)\cos(2\pi \times 850 \times 10^3 n/f_s) \\ &= x[n]\cos(2\pi \times 850 \times 10^3 n/f_s) \ , \end{aligned} \tag{3}$$

and Ω_c is just the factor that multiplies n in the argument of the cosine, so

$$\Omega_c = 2\pi \times 850 \times 10^3/f_s = 2\pi \times 850 \times 10^3/(2 \times 10^6) = \mathbf{0.85}\pi \ .$$

(2 points)

B. Suppose the Fourier series coefficients $X[k]$ of the DT signal $x[n]$ in Eq.(2) are *purely real*, and are as shown in the figure below, plotted as a function of $\Omega_k = 2k\pi/N$. (Note that the figure is not drawn to scale. Also, the different values of Ω_k are so close to each other that we have just interpolated adjacent values of $X[k]$ with a straight line, rather than showing you a discrete "stem" plot.) Observe that the Fourier series coefficients are non-zero for frequencies Ω_k in the interval $[-.005\pi, .005\pi]$, and 0 at all other Ω_k in the interval $[-\pi, \pi]$.



Draw a carefully labeled sketch below (though not necessarily to scale) to show the Fourier series coefficients of the DT modulated signal $y[n]$. However, rather than labeling your horizontal axis with the Ω_k, as we have done above, you should **label the axis with the appropriate frequency f_k in Hz**.

Solution: The spectrum $Y[k]$ of $y[n]$ obtained by cosine modulation of $x[n]$ is given by

$$Y[k] = \frac{1}{2}\Big\{X[k - k_c] + X[k + k_c]\Big\} \ ,$$

and is purely real. The index k_c was introduced earlier by writing $\Omega_c = 2k_c\pi/N$.

The corresponding plot as a function of Ω_k consists of two half-height (i.e., height $4/2 = 2$) copies of the triangular shape given in the plot of $X[k]$, one of these copies being centered at Ω_c and the other at $-\Omega_c$.

The frequency f_c corresponding to Ω_c is 850kHz again, so on the frequency axis f_k (which extends from $-f_s/2$ to $f_s/2$, i.e., covers the interval $[-1, 1]$ MHz) the two half-height triangles are centered at ± 850 kHz. The spectrum (i.e., the set of nonzero Fourier series coefficients) is restricted to the intervals $[-855, -845]$ kHz and $[+845, +855]$ kHz.

(3 points)

Assume now that the receiver detects the CT signal $w(t) = 10^{-3}y(t - t_0)$, where $t_0 = 3 \times 10^{-6}$ sec, and that it samples this signal at f_s samples/sec, thereby obtaining the DT signal

$$w[n] = 10^{-3}y[n - M] = 10^{-3}x[n - M]\cos\Big(\Omega_c(n - M)\Big) \qquad (4)$$

for an appropriately chosen integer M.

C. Determine the value of M in Eq.(4).

Solution: The delay $t_0 = 3 \times 10^{-6}$ sec, at a sampling rate of 2×10^6 samples/sec, corresponds to

$$M = 3 \times 10^{-6} \times 2 \times 10^6 = 6 \ \text{samples.}$$

(2 points)

D. Noting your answer from part **B**, determine for precisely which intervals of the frequency axis the Fourier series coefficients of the signal $y[n-M]$ in Eq.(4) are non-zero. You **need not find the actual coefficients**, only the frequency range over which these coefficients will be non-zero. Also **state whether or not the Fourier coefficients will be real**. Explain your answer.

Solution: Delaying $y[n]$ by M steps to get $y[n-M]$ has the following effect on the spectrum of $y[n]$: The kth Fourier coefficient $Y[k]$ gets multiplied by the frequency response associated with the delay operation, evaluated at the frequency Ω_k. The new Fourier coefficients are thus

$$e^{-j\Omega_k M} Y[k] = e^{-j6\Omega_k} Y[k] \ .$$

The spectrum of the delayed signal is therefore non-zero (respectively zero) precisely where the spectrum of the original signal is non-zero (respectively zero), i.e., inside (respectively outside) the frequency bands $[-855, -845]$kHz and $[+845, +855]$kHz. Though the original spectrum $Y[k]$ was real, the multiplication by the complex number $e^{-j6\Omega_k}$ causes the spectrum of the delayed signal to have both real and imaginary parts.

(3 points)

E. The demodulation step to obtain the DT signal $x[n-M]$ from the received signal $w[n]$ now involves multiplying $w[n]$ by a DT carrier-frequency signal, followed by appropriate low-pass filtering (with the gain of the low-pass filter in its passband being chosen to scale the signal to whatever amplitude is desired). Which one of the following six DT carrier-frequency signals would you choose to multiply the received signal by? Circle your choice and give a brief explanation.

(a) $\cos\Big(\Omega_c n\Big)$.

(b) $\cos\Big(\Omega_c(n-M)\Big)$.

(c) $\cos\Big(\Omega_c(n+M)\Big)$.

(d) $\sin\Big(\Omega_c n\Big)$.

(e) $\sin\Big(\Omega_c(n-M)\Big)$.

(f) $\sin\Big(\Omega_c(n+M)\Big)$.

Solution: One gets the maximum signal through the low-pass filter by using a demodulating signal that is maximally in-phase with the carrier signal associated with what one is trying to demodulate. The latter carrier signal, as Eq.(3) shows, is $\cos\Big(\Omega_c(n-M)\Big)$. So the best signal for demodulation is the one in (b), namely $\cos\Big(\Omega_c(n-M)\Big)$.

VI Why So Slow?

A sender A and a receiver B communicate using the stop-and-wait protocol studied in 6.02. There are n links on the path between A and B, each with a data rate of R bits per second. The size of a data packet is S bits and the size of an ACK is K bits. Each link has a physical distance of D meters and the speed of signal propagation over each link is c meters per second. The total processing time experienced by a data packet *and* its ACK is T_p seconds. ACKs traverse the same links as data packets, except in the opposite direction on each link (the propagation time and data rate are the same in both directions of a link). There is no queueing delay in this network. Each link has a packet loss probability of p, with packets being lost independently.

12. [6 points]: What are the following four quantities in terms of the given parameters?

A. Transmission time for a data packet *on one link* between A and B: ________________________.

Solution: $\frac{S}{R}$. Each data packet has size S bits, and the speed of the link is R bits per second.

B. Propagation time for a data packet across n links between A and B: ________________________.

Solution: $\frac{nD}{C}$. Total distance to be travelled is nD since each link has length D meters, and there are n such links. The propagation speed is c meters/second.

C. Round-trip time (RTT) between A and B? ____.
(The RTT is defined as the elapsed time between the start of transmission of a data packet and the completion of receipt of the ACK sent in response to the data packet's reception by the receiver.)
Solution: $\frac{nS}{R} + \frac{nK}{R} + \frac{2nD}{c} + T_p$. We need to consider the following times:

- Transmit data across n links: $\frac{nS}{R}$ using result from 12A.
- Transmit ACK across n links: $\frac{nK}{R}$ also using result from 12A
- Propagate data across n links and ACKS across n links: $\frac{2nD}{C}$
- Total time to process the data and the ACK: T_p

D. Probability that a data packet sent by A will reach B = ____.
Solution: $(1 - p)^n$. Probability of loss in a link is p, so probability of no loss in a link is $1 - p$. Since link losses are independent, probability of no loss in n links is $(1-p)^n$. No loss in n links means the data gets from A to B successfuly.

VII Stop Wait... Do Tell Me

Ben Bitdiddle gets rid of the timestamps from the packet header in the 6.02 stop-and-wait transport protocol running over a best-effort network. The network may lose or reorder packets, but it never duplicates a packet. In the protocol, the receiver sends an ACK for each data packet it receives, echoing the sequence number of the packet that was just received.

The sender uses the following method to estimate the round-trip time (RTT) of the connection:

1. When the sender transmits a packet with sequence number k, it stores the time on its machine at which the packet was sent, t_k. If the transmission is a retransmission of sequence number k, then t_k is updated.
2. When the sender gets an ACK for packet k, if it has not already gotten an ACK for k so far, it observes the current time on its machine, a_k, and measures the RTT sample as $a_k - t_k$.

If the ACK received by the sender at time a_k was sent by the receiver in response to a data packet sent at time t_k, then the RTT sample $a_k - t_k$ is said to be correct. Otherwise, it is incorrect.

13. [5 points]: Circle **True** or **False** for the following statements.

A. True / False If the sender never retransmits a data packet during a data transfer, then all the RTT samples produced by Ben's method are correct.

Solution: True. If there are no retransmissions ever made, t_k gets set once and never updated, and the ACK for k can be unambiguously associated with the corresponding packet transmission, and the RTT sample will be correct.

B. True / False If data and ACK packets are never reordered in the network, then all the RTT samples produced by Ben's method are correct.

Solution: False. If the sender retransmits a packet, it can no longer unambiguously associate a packet's ACK reception with a particular transmission or retransmission of a packet with the same sequence number.

C. True / False If the sender makes no spurious retransmissions during a data transfer (i.e., it only retransmits a data packet if all previous transmissions of data packets with the same sequence number did in fact get dropped before reaching the receiver), then all the RTT samples produced by Ben's method are correct.

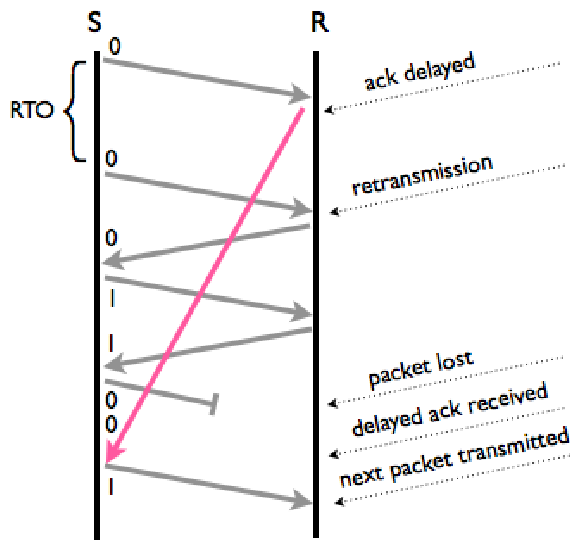
Solution: True. Given that there are no spurious retransmissions, *at most* one packet with a given sequence number, k can reach the receiver, and the sender can get at most one ACK for k. If the sender gets an ACK for k, that ACK *must* correspond to the last packet transmission of that sequence number, k. The RTT samples in this case will be produced correctly.

Opt E. Miser implements the 6.02 stop-and-wait reliable transport protocol with one modification: being stingy, he replaces the sequence number field with a 1-bit field, deciding to reuse sequence numbers across data packets. The first data packet has sequence number 1, the second has number 0, the third has number 1, the fourth has number 0, and so on. Whenever the receiver gets a packet with sequence number $s (= 0 \text{ or } 1)$, it sends an ACK to the sender echoing s. The receiver delivers a data packet to the application if, and only if, its sequence number is different from the last one delivered, and upon delivery, updates the last sequence number delivered.

14. [3 points]: He runs this protocol over a best-effort network that can lose packets (with probability less than 1) or reorder them, and whose delays may be variable. Does the modified protocol always provide correct reliable, in-order delivery of a stream of packets?

(Explain your answer in the space below.)

Solution: No. For example, see the picture below.



VIII Slip Sliding (Windows) Away

Consider a reliable transport connection using the 6.02 sliding window protocol on a network path whose RTT in the absence of queueing is $\text{RTT}_{\text{min}} = 0.1$ seconds. The connection's bottleneck link has a rate of $C = 100$ packets per second, and the queue in front of the bottleneck link has space for $Q = 20$ packets.

15. [6 points]: Assume that the sender uses a sliding window protocol with fixed window size. There is no other connection on the path.

A. If the size of the window is 8 packets, then what is the throughput of the connection?

Solution: The bandwidth-delay product of the connection is 10 packets (bottleneck rate times the minimum RTT). With a window size of 8, queues will not yet have built up, so the throughput is 80 packets/second.

B. If the size of the window is 16 packets, then what is the throughput of the connection?

Solution: Queues will have built up; the bottleneck link is now saturated, so the throughput is 100 packets/second.

C. What is the smallest window size for which the connection's RTT exceeds RTT_{min}?

Solution: 11 packets. The bandwidth-delay product is 10 packets. It's probably reasonable to accept an answer of 10 packets too.

IX Start Me Up

TCP, the standard reliable transport protocol used on the Internet, uses a sliding window. Unlike the protocol studied in 6.02, however, the size of the TCP window is *variable*. The sender changes the size of the window as ACKs arrive from the receiver; it does not know the best window size to use a priori.

TCP uses a scheme called **slow start** at the beginning of a new connection. Slow start has three rules, R1, R2, and R3, listed on the next page for convenience (TCP uses some other rules too, which we will ignore).

In the following rules for slow start, the sender's current window size is W and the last in-order ACK received by the sender is A. The first packet sent has sequence number 1.

R1. Initially, set $W \leftarrow 1$ and $A \leftarrow 0$.
R2. If an ACK arrives for packet $A+1$, then set $W \leftarrow W+1$, and set $A \leftarrow A+1$.
R3. When the sender retransmits a packet after a timeout, then set $W \leftarrow 1$.

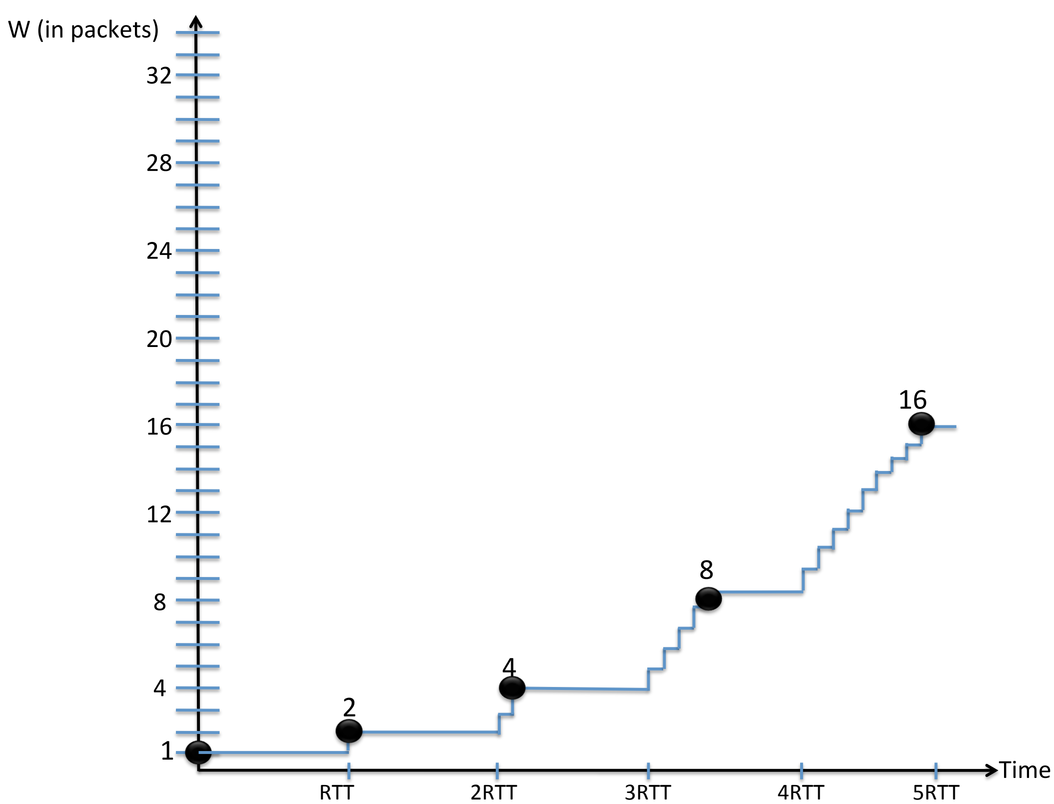
Assume that all the other mechanisms are the same as the 6.02 sliding window protocol. Data packets may be lost because packet queues overflow, but assume that packets are not reordered by the network.

We run slow start on a network with $\text{RTT}_{\text{min}} = 0.1$ seconds, bottleneck link rate = 100 packets per second, and bottleneck queue = 20 packets.

16. [2 points]: What is the smallest value of W at which the bottleneck queue overflows?

Solution: The smallest W for which the queue overflows is 10 + 20 + 1 = 31 packets. The 10 is because that's the bandwidth-delay product; the 20 is the maximum size of the queue. And we need one more packet to cause an overflow.

17. [5 points]: Sketch W as a function of time on the graph below for the first 5 RTTs of a connection. The X-axis marks time in terms of multiples of the connection's RTT. (*Hint: Non-linear!*)



Solution: The growth in the window size at the beginning of a connection doing "slow start" is exponential! After 1 RTT, the window goes from 1 to 2. At that time the sender sends two packets; those ACKs arrive about 1 RTT after the packets were sent, making the window equal to 4 after the second of the ACKs arrives. Then, after the next RTT, 4 more ACKs arrive, doubling the window from 4 to 8 after the last of those 4 ACKs arrive. Eventually, the window size exceeds 30 and at some point in time in the next RTT after that, a packet drop occurs, causing the window to drop to 1 when the timeout happens. In this problem, the timeout does not happen in the first 5 RTTs, and hence isn't shown. Note that once queues start to grow—i.e., when the window exceeds 10 packets, the RTT isn't constant. I.e., the x-axis of the picture drawn is not to scale.

X Information and Cmprssn

18. [3 points]: After careful data collection, Alyssa P. Hacker observes that the probability of "HIGH" or "LOW" traffic on Storrow Drive is given by the following table:

	HIGH traffic level	LOW traffic level
If the Red Sox are playing	P(HIGH traffic) = 0.999	P(LOW traffic) = 0.001
If the Red Sox are not playing	P(HIGH traffic) = 0.25	P(LOW traffic) = 0.75

A. If it is known that the Red Sox are playing, then how much information in bits is conveyed by the statement that the traffic level is LOW. Give your answer as a mathematical expression.

Solution: $\log_2(1/0.001) = \log_2 1000$ bits.

B. Suppose it is known that the Red Sox are **not** playing. What is the entropy of the corresponding probability distribution of traffic? Give your answer as a mathematical expression.

Solution: $0.25 \log_2(1/0.25) + 0.75 \log_2(1/0.75) = 0.5 + 0.75 \log_2(4/3) = 2.5 - \log_2 3$.

19. [3 points]: X is an unknown 4-bit binary number picked uniformly at random from the set of all possible 4-bit numbers. You are given another 4-bit binary number, Y, and told that the Hamming distance between X (the unknown number) and Y (the number you know) is *two*. How many bits of information about X have you been given?

Solution: $\log_2(16/6)$.

20. [3 points]: Consider a Huffman code over four symbols, A, B, C, and D. For each of the following encodings, circle **True** if it is a valid Huffman encoding, and **False** otherwise. Give a brief explanation next to each one.

A. True / False $A : 0, B : 11, C : 101, D : 100$. Solution: True.

B. True / False $A : 1, B : 01, C : 00, D : 010$. Solution: False; not prefix-free because the code for B is a prefix of the code for D.

C. True / False $A : 00, B : 01, C : 110, D : 111$ Solution: False; a shorter code would have C and D encoded in 2 bits, as 10 and 11 (or vice versa), and that would be a Huffman code for the same symbol probabilities, not the one given.

21. [4 points]: Huffman is given four symbols, A, B, C, and D. The probability of symbol A occurring is p_A, symbol B is p_B, symbol C is p_C, and symbol D is p_D, with $p_A \geq p_B \geq p_C \geq p_D$. Write down a single condition (equation or inequality) that is both necessary and sufficient to guarantee that, when Huffman constructs the code bearing his name over these symbols, each symbol will be encoded using exactly two bits. Explain your answer.

Solution: $p_C + p_D > p_A$. The reason is that only two trees are possible. For the tree where every symbol has two bits, we need to have $p_C + p_D + p_B > p_B + p_A$, which gives us the desired answer. We need a strict inequality because otherwise there is a possibility of a tie, and the unbalanced tree may be chosed because it has the same expected number of bits.

22. [7 points]: Consider the LZW compression and decompression algorithms as described in 6.02. Assume that the scheme has an initial table with code words 0 through 255 corresponding to the 8-bit ASCII characters; character "a" is 97 and "b" is 98. The receiver gets the following sequence of code words, each of which is 10 bits long:

97 97 98 98 257 256

A. What was the original message sent by the sender?

Solution: aabbabaa

B. By how many bits is the compressed message shorter than the original message (each character in the original message is 8 bits long)?

Solution: 64 - 10*6 = 4 bits.

C. What is the first string of length 3 added to the compression table? (If there's no such string, your answer should be "None".)

Solution: "aba".

FIN

Have a great winter break!