COMPSCI 314 S1C 05 Assignment 1

Department of Computer Science
The University of Auckland
Final version: posted 15 March 2005
Due Wednesday 23 March 2005, 4pm
in https://adb.ec.auckland.ac.nz/adb/

- A preliminary version of this assignment was web-posted on 14 March 05. In order to get full credit, you must prepare your answer script from this final version.
- This assignment will contribute 30/300 = 10% to your coursework mark, and 3% to your overall course mark.
- Total possible marks on this assignment: 30.
- To obtain full credit, your script must clearly show *how* you obtained a correct answer.
- If you require additional information in order to answer a problem, you should briefly explain why this information is necessary and why your assumptions about the "missing values" or "missing facts" are reasonable.
- You may submit your assignment either in MS Word format, or in PDF format. The latter format should be used if you are scanning a handwritten script. The filesize should not exceed 2 MB. If you have difficulty meeting these specifications, please contact a tutor **prior to Tuesday 22 March 2005**.
- Your tutors are Li Jihong (jli018@cs.auckland.ac.nz) and Li Li (lli057@ec.auckland.ac.nz).

Q1. The Cassini orbiter is now 8.5 AU (Astronomical Units) from Earth. One AU is approximately 1.5×10^{11} meters. Cassini communicates with its Earth base stations at a raw bitrate of 142 Kb/s.

Generally, this is a very long link. And end to end latency is quite long.

a. Estimate, to within ±20%, the effective user bandwidth for the Cassini-Earth datalink, under the assumption that this datalink uses the simple ARQ protocol described in the second set of lecture slides. Assume the user message is approximately one kilobyte (this is the variable *U* in the lecture-slide analysis). Explain any additional assumptions you require to make your estimate.
[3 marks]

Sample answer:

The user message is U = 1 kB (± 10%). Note that by writing (± 10%), we are indicating that we think the user message is somewhere between 900B and 1100B long.

Here we are assuming that 1 kB = 10^3 B, although if we believed that this problem was posed by a computer scientist who didn't know that the rest of the world (including data-communications specialists) use a decimalised kilobyte (sometimes written kB) rather than binarised kilobyte (sometimes written KB, or even as KiB, in which case the unit is called a "kibibyte"), we could write U = 1024 B (± 10%). Alternatively if we were *really* unsure about the intended meaning of "kilobyte" in this problem, we might "split the difference" by writing U = 1012 B (± 1.2%). However, since we're only trying to get within (± 20%) in our final answer, it's a poor use of our time to be hypercareful about ten or twenty bytes out of a thousand bytes!

In the remainder of this answer sheet, we will use both "k" and "K" to mean 10^3; for example, 1 km = 1000 m, and 1 kB = 1KB = 1000 B. We will use "M" for 10^6 and "G" for 10^9.

In COMPSCI 314, we have never seen headers and trailers longer than ten or twenty bytes, and acknowledgements are quite short, perhaps as short as a single byte. Even if the packets are encapsulated in several layers of protocol, the total messaging overheads of the ARQ protocol are bounded from below by 0 B and from above by 120 B, that is, we assume 0 = 2H + 2T + A = 120 B. This implies that 2H + 2T + A + U is some value between 900 B and 1220 B, which we may write as follows

$$2H + 2T + A + U = 1060 \text{ B} = 8480 \text{ b} (\pm 16\%)$$

Some students may assume H+T+A = 0, or say they are "ignoring" the effects of headers, trailers, and acknowledgements. This introduces a one-sided error into their estimate for 2H + 2T + A + U, but this fine distinction would be lost in the convenient 2-sided error notation we are using here in our sample answer:

$$2H + 2T + A + U = 1000 \text{ B} = 8000 \text{ b} (\pm 22\%)$$

The signalling rate is

$$R = 142 \text{ kb/s} (\pm 0.4\%)$$

Using the less precise estimate, the time to send message is

$$(2H + 2T + A + U)/R = 8000 \text{ b} / 142 \text{ kbps} = 56.3 \text{ ms} (\pm 22.4\%)$$

Using our more precise (two-sided, balanced) estimator for the headers and trailers, we would write

$$(2H + 2T + A + U)/R = 8480 \text{ b}/142 \text{ kbps} = 59.7 \text{ ms} (\pm 16.4\%)$$

Marking guideline: award 1 mark for noting 2H+2T+A is "small" in relation to U=1000B; or for noting that no careful estimation of 2H+2T+A+U ***is required because the total signalling time*** (2H+2T+A+U)/R ***is negligible in comparison to the total latency*** 2D/V***; or for making a reasonably-accurate estimate of either*** 2H + 2T + A + U ***or*** (2H + 2T + A + U)/R.

Students who do a careful error analysis, for example by calculating error bounds as in the sample answer above, should be commended for a "job well done" but not given bonus points.

The distance D is given to two signficant digits as 8.5 AU, so it must be accurate to (±0.05 AU) which is (±0.6%) The definition of an AU is also given to two significant digits: 1 AU = 1.5 x 10^{11} m (±3%).

$$D = (8.5 \text{ AU})(1.5 \times 10^{11} \text{ m}) = 1.275 \times 10^{11} \text{ m} (\pm 3.6\%)$$

Note that it is easy to compute an error estimate for the multiplication or division of two approximate values, as the relative errors are additive in such computations. In this particular case, (±3%) + (±0.6%) = (±3.6%).

The signals travel at the speed of light, c, which we treat here with two significant digits of accuracy:

$$V = c = 3.0 \times 10^8 \text{ m} (\pm 2\%)$$

End to end latency is thus

$$D/V = 4250 \text{ s} (\pm 5.6\%)$$

The round trip latency is precisely twice this value:

$$2D/V = 8500 \text{ s} (\pm 5.6\%)$$

Award 1 mark for noting that D/V = 4250s ***or that*** 2D/V = 8500 s.

The user data rate is U divided by the sum of the two times calculated above: 59.7 ms (± 16.4%) + 8500 s (±5.6%) = 8500 s (±5.6%). Note that the signalling time is truly insignificant in comparison to the latency: it has no effect on our estimate of the sum of the two times.

The user (or effective) data rate is thus

8000b (± 10%) / 8500 s (±5.6%) = 0.94 b/s (± 16%)

Note that our estimate is demonstrably with the (± 20%) error tolerance required by the problem. We are reporting the data rate only to two significant digits, because with an error of (± 16%) it would be silly (possibly even to the point of causing confusion) if we reported three or more significant digits.

Indeed, perhaps the best summary of our analysis is that this link has an effective data rate of approximately 1 b/s. Note that anyone who reads this summary statement would see that it is only accurate to its first significant digit, that is, they should understand we have estimated the effective data rate to be some value between 0.5 b/s and 1.5 b/s. Our actual estimate is somewhat more precise than this, however we have used only a simple model (and made several assumptions) when forming our estimate. So we really shouldn't be very confident that an experimental measurement of an actual link would always show an effective rate between the 0.78 b/s and 1.20 b/s outer limits of our simple estimate.

Our estimated user data rate of 1 b/s for this link is very much smaller than its raw bit rate of 142 kb/s. The reason for this huge inefficiency should be pretty obvious to any student of COMPSCI 314: the ARQ protocol is a very poor choice for any link with very high latency compared to its signalling time.

Award 1 mark for a calculation showing that the effective data rate is approximately 1 b/s.

b. Rework your previous estimate, under the revised assumption that the user message length is one megabyte. **[1 mark]**

The user message is U = 1MB = 8×10^6 b. The signalling time almost exactly 1000 times larger than in our previous estimate, because U is now 1000x larger and H+T are even less significant than before. So the signalling time (2H+2T+A+U)/R is approximately 60 s, and the user data rate is

$$(8 \times 10^6 \text{ b})/(8500 \text{ s} + 60\text{s}) = 934 \text{ b/s}$$

This estimate has a slightly larger error than the (± 16%) error of our previous estimate, because the signalling time is no longer completely negligible. However the estimated error, if we calculated it exactly, would certainly be less than (± 20%). So we can report our answer as either 930 b/s (± 20%), or as 1 kb/s, depending on how much precision we believe our audience requires.

Clearly, a larger message will boost the effective bandwidth of a link – but we might wonder whether our estimation method is appropriate for such long packet. This issue is explored in the next problem.

Award 1 mark for a calculation showing that the effective data rate is approximately 1kb/s.

c. Make some other, more reasonable, set of assumptions about the flow-control protocol and/or the packet length on the Cassini-Earth datalink. Explain your assumptions clearly. Estimate the effective user bandwidth under your new set of assumptions, and indicate why you believe your new assumptions are more reasonable than the ARQ/1KB protocol assumption of Q1a or the ARQ/1MB protocol assumption of Q1b. **[3 marks]**

From our calculations above, we might be tempted to conclude that "the more bits sent in a packet, the more efficient the link". However, real-world systems have noise; noisy systems have errors;

and the chance of a transmission error in a million-byte packet is much larger than the chance of a transmission error in a thousand-byte packet. Our estimated error rate did *not* include any allowance for the time required for a retransmission in case of an error. The time penalty per retransmission would be huge (8500 seconds), if an ARQ protocol were used for communications from Earth to Cassini. So the ARQ protocol really should be replaced by a sliding-window protocol with a large window size.

The examples seen so far in COMPSCI 314 suggest that U = 1K octets is a suitable size for one packet. Packets smaller than this are somewhat less efficient, because the 2H+2T+A "overhead costs" become noticeable for U < 1000, and they become quite significant for U < 100. Packets larger than 1 kB are also less efficient, because the more frequent transmission errors, multiplied by the higher cost of each packet retransmission, give a retransmission cost that can be quite significant (depending on the error rates).

As noted in the lecture slides and in your textbook, a transmitter should hold, in its buffers, about as much data as can be sent in a round-trip latency time. To achieve good efficiency on the Cassini-Earth link, the transmitter's buffers should thus hold (142 kb/s)(8500 s) = 1.2 Gb = 150 MB. If we use 1 kB packets, then we'll need a window size of approximately 150,000 to keep track of all the outstanding packets for the preceding 8500 seconds.

Transmission errors:* *1 mark

1 MB packet is unrealistically large:* *1 mark

Suggest use of sliding window protocol with large window size:* *1 mark

If a student caculates the 150 MB transmission buffer size, and uses this to compute an appropriate size for a sliding window (e.g. a window size of 150 K for 1 KB packets), then they should be awarded full credit (3 marks) on this problem even if they don't mention transmission errors or argue against 1 MB packets.

Q2. In December 2004, the Huygens probe was released from Cassini. The Huygens probe landed on Titan, one of Saturn's moons, in January 2005. During its descent, Huygens sent back a series of digitized images to Cassini. Cassini then relayed these images to Earth. Huygen's uplink to Cassini had a raw bitrate of only 8 kb/s, due to severe power and weight constraints on the spacecraft design.

a. When Cassini was 60,000 km from Huygens, estimate (to within ±20%) its effective bandwidth to Huygens under the assumption that it was using the ARQ protocol of the second set of lecture slides, with a packet length of approximately one kilobyte.

[2 marks]

Round trip latency = (6 x 10^7 m)/(3 x 10^8 m/s) = 0.40 s

Signalling time = (2H+2T+A+U)/R = (8480 b)/(8 kb/s) = 1.06s

Alternatively, a student may compute U/R = (8000 b)/(8 kb/s) = 1.0 s. This is accurate to within 20%, so it is quite acceptable.

User data rate = (1 kB)/(0.40 s + 1.06 s) = 0.68 kB/s = 5.5 kb/s

Alternatively, the computation may be (1 kB)/(0.4 s + 1.0 s) = 0.71 kB/s = 5.6 kb/s

Round trip latency of 0.4 s:* *1 mark

User data rate of approximately 5.5 kb/s:* *1 mark

b. Cassini sent 250 images to Huygens during its 2.5-hour descent. Compute the average number of kilobytes per image, assuming that its uplink was transmitting images continuously, and assuming that your estimate from Q2a is correct.

[1 mark]

The maximum amount of data that could be sent in 2.5 hours over a 0.7 kB/s link is

(2.5 h)(60 min/h)(60 s/min)(0.7 kB/s) = 6.3 MB

If 250 images are sent continuously over this link, then the average image size is

(6.3 MB)/(250 images) = 25 kB/image

The careful student will have noticed, after doing the next question, that Cassini actually sent 367 images. If 367 images were sent continuously over this link, then the average image size is

(6.3 MB)/(367 images) = 17 kB/image

Students will have slightly different answers here, depending on their calculations in the first part of this question. However they should be within 20% of 25 kB/image if they used the (incorrect) number of 250 images, and within 20% of 17 kB if they used the (correct) number of 367 images: 1 mark

Bonus mark for pointing out that Cassini actually sent back 367 images, according to the website named in the next part of this question: +1 mark.

Note: bonus marks can not boost a student's total mark on this assignment above the maximum possible (30 marks).

c. Compare your estimated average image size from Q2b to an average image size computed by taking a randomly-chosen sample of three images from http://esamultimedia.esa.int/docs/titanraw/index.htm. If your estimates are significantly different, give (and briefly explain) two possible reasons for this discrepancy.

[4 marks]

One randomly-chosen sample of images are:

Image *.716, size: 31 kB

Image *.174, size: 24 kB

Image *.526, size: 36 kB

On the basis of this random sample, the average size of an image is 30 kB (+/- 15 kB), where the +/- 15 kB is the 95% confidence interval on the sample mean. I used Excel's "Tools/Data Analysis/Descriptive Statistics" functions to calculate this confidence interval. Note that we cannot expect to have a low-error estimate of the average size of an image after looking at only three images!

Students will have quite different answers here, but generally their sample means should be in the 15 kB to 45 kB: 2 marks for a statistical estimation process that seems reasonably "random" (i.e. doesn't just compute an average over the first three images on first page of the website). No penalty if a student makes the mistake of sampling the "thumbnail" images displayed in the upper pages of the website: these are approximately 2 kB to 3 kB in size.

In Q2b, we estimated the "maximum possible" average image size as 25 kB, where this average is attainable if Cassini transmitted 250 images as rapidly as possible (using the analysis of Q2a) over the time available for transmission.

This 25 kB value is *not* "significantly different" (at 95% confidence) to the average mean computed by our statistical sampling, so we have no discrepancy requiring any explanation other than sampling error in our statistical estimate, at 95% confidence, of 30 kB (+/- 15 kB).

However at 90% confidence, our sample mean of 30 kB (+/- 10 kB) *is* significantly different to the average image size of 17 kB that we would calculate in the Q2b analysis, if we use the correct number of images (367) in that analysis. So it does seem that the images may be a bit (possibly quite a bit) larger than we thought possible from the analysis of Q2b.

Some of the Cassini images are quite a bit larger than 30 kB, so students whose random samples contain one or more of these images will have a sample mean that is significantly larger than the 25 kB estimate from Q2b.

Here are possible reasons why a student's sample mean might be significantly *larger* than their Q2b analysis:

1. A student may have made a mistake in the Q2b analysis, or in the Q2a analysis on which it was based.
2. A student may have chosen, at random, an image that is much larger than average, causing their sample mean to be much larger than the actual mean size of the 367 images.
3. The assumptions in the Q2a analysis may be incorrect, in ways that would cause an incorrectly-low effective bandwidth figure to be computed.
    a. For example, if the average distance from Cassini to Huygens was significantly lower than 60000 km, then the ARQ protocol would be more efficient than predicted in the analysis of Q2a which assumed a distance of 60000 km.
    b. Another possibility is that a sliding-window protocol was used by Cassini, in which case the Q2a analysis (which assumed the less efficient ARQ) would give a lowball estimate.
4. The assumptions in the Q2b analysis may be incorrect, in ways that would cause an incorrectly-low average image size to be computed. For example it is conceivable that the images were transmitted in a different format to the JPG versions on the website (even though these were called "raw"), and if this format is significantly more efficient than the JPG file, then the JPG-format files on the website would be significantly larger than what was actually transmitted.

Here are possible reasons why a student's sample mean might be significantly *smaller* than their Q2b analysis:

1. A student may have made a mistake in the Q2b analysis, or in the Q2a analysis on which it was based.
2. A student may have chosen, at random, an image that is much smaller than average, causing their sample mean to be much smaller than the actual mean size of the 367 images.
3. The assumptions in the Q2a analysis may be incorrect, in ways that would cause an incorrectly-high effective bandwidth figure to be computed.
    a. For example, if the average distance from Cassini to Huygens was significantly greater than 60000 km, then the ARQ protocol would be even less efficient than predicted in the analysis of Q2a which assumed a distance of 60000 km.
    b. Another possibility is that the data communications protocols actually used by Cassini had overheads that were significantly higher than

assumed. In particular, the error-correction protocols may have required either a lot of retransmissions, or large signalling overheads, or both.

4. The assumptions in the Q2b analysis may be incorrect, in ways that would cause an incorrectly-high average image size to be computed.
   a. We know of one such error already: the actual number of images was 367, not 250.
   b. Also, it is conceivable that the images were transmitted in a different format to the JPG versions on the website (even though these were called "raw"), and if this format is significantly less efficient than the JPG file, then the JPG-format files on the website would be significantly larger than what was actually transmitted.
   c. Huygens might not have been sending photos continuously during its descent, perhaps because it was busy sending other data over the link, or because the link was "down" for part of the 2.5 hour time of its descent.

Award 2 marks for any understandable discussion that explains two reasons why a statistical estimate differs significantly from the analysis of Q2b; or for any understandable indication that the statistical estimate does* not *differ significantly. No penalty if a student does not compute a confidence interval, however students should be penalised 1 marks (so their maximum score on this problem is 3 marks) if they assert that a difference of less than 20% is "significant".

Q3. At pp. 299-304, your 2nd Edition textbook analyses the effective bandwidth of a protocol it calls "Stop and Wait". The same material appears at pp. 356-360 in the 3rd Edition. The textbook analysis is quite similar to the effective bandwidth analysis of the ARQ protocol that appears at slides 17-22 of the second set of lecture slides for this course.

a. Draw a diagram, in the style of lecture slide 4 of Set 3, for the Stop and Wait (SAW) protocol of the textbook. Your diagram should have the following elements:

- A transmitting station TX (on the left)
- A receiving station RX (on the right)
- A transmitted data message DATA
- A positive acknowledgement message ACK
- A negative acknowledgement message NAK
- A clear indication of what should happen when a valid message (DATA, ACK, or NAK) is received by the TX or RX.
- A clear indication of what should happen when an invalid message is received by the TX or the RX. **[3 marks]**

Diagram clearly shows the TX retransmission after the RX receives invalid message and sends NAK: 1 mark

Diagram clearly shows the TX sending next message after the RX receives valid message and sends ACK: 1 mark

Diagram clearly shows the TX doing something reasonable when it receives an invalid input. 1 mark Note that the textbook's description of SAW doesn't specify what is supposed to happen in this case, so a variety of answers are possible. The only reasonable actions a TX might make when it receives an invalid input are listed below.

1. ***The TX might resend the previous message; this will not cause any trouble if the messages carry a serial number, so that the RX will know to throw away a duplicated message (in the case that the RX's ACK signal is damaged in transmission).***
2. ***Alternatively, as shown in the sample answer above, the TX might send an error message to a higher-level protocol.***
3. ***Full credit should also be given to any student who notes that the SAW protocol does not define what the TX should do in this case.***

A student should get a minimum of 1 mark on this problem if their diagram is recognisably a "protocol diagram" with a TX sending information to an RX, with the RX deciding what to do next on the basis of what it receives from the TX, and with the TX deciding what to do next on the basis of what it receives from the RX.

b. The SAW protocol of the textbook is essentially the same as the ARQ protocol of the second set of lecture slides. However the variables are differently named and, in some cases, differently defined.

Some variables in the ARQ analysis correspond directly to a variable in the SAW analysis. Some variables in the ARQ analysis may correspond to a sum (or difference, or other algebraic combination) of variables in the SAW analysis. There may also be some SAW variables that do **not** appear in the ARQ analysis; or some ARQ variables that do **not** appear in the SAW analysis.

Your answer script for Q3b should present this information in tabular form, with three columns.

- The first column in your table should give a descriptive name to each parameter in the effective bandwidth analysis of either ARQ or SAW. For example, this column might contain the name "raw bit rate".
- The second column should indicate the ARQ variable (or algebraic combination of ARQ variables) for each parameter named in the first column. If this parameter is *not* evaluated in the ARQ analysis of the second set of lecture slides, then write "not modeled" in this position. For example, you might write "*R*" in the second column of your table, in the row for the parameter you named "raw bit rate".
- The third column should give the value of each parameter in the SAW analysis. For example, you might write "R" in the third column of your table, at the row labeled "raw bit rate", because this is the name of the variable used in the textbook's SAW analysis for this parameter.
- Here are the first two rows of your table:

Descriptive Name	ARQ Variable or Formula	SAW Variable or Formula
Raw bit rate	***R***	***R***

[3 marks]

In our sample answer below, we indicate the units of measurement for each variable in parentheses after the variable name. This is helpful and relevant information, but is not required.

Descriptive Name	***ARQ Variable or Formula***	***SAW Variable or Formula***
Raw bit rate	***R (b/s)***	***R (Mb/s)***
User data size	***U (B)***	***N (b)***

Frame size	***U+H+T (B)***	***F (b)***
Header size	***H (B)***	***(not modeled, however the envelope size F-N in the SAW analysis has the same value as H+T in the ARQ analysis)***
Trailer size	***T (B)***	***(not modeled)***
Acknowledgement Size	***A (B)***	***A (b)***
Distance between TX and RX	***D (m)***	***D (m)***
Signal transmission velocity	***V (m/s)***	***S (m/μs)***
Time to create one frame	***(not modeled)***	***T (μs)***
Total time	***8(2H+2T+A+U)/R + 2D/V (s)***	***(F+A)/R + 2(T+D/S) (μs)***
Effective data rate	***U/(8(2H+2T+A+U)/R + 2D/V) b/s***	***N/((F+A)/R + 2(T+D/S)) Mb/s***

Grading is by "spot-check":

1 mark for a row on "Frame Size" (or with a similar name) with U+H+T for ARQ and F for SAW.

1 mark for a row on "Signal Transmission Velocity" (or with a similar name) with V for ARQ and S for SAW.

1 mark for a row on "Time to create one frame" (or with a similar name) with "not modeled" for ARQ and T for SAW.

Graders have discretion to award a student full credit if their table fails one of these "spot checks" but seems quite complete and accurate in other respects.

c. Interpret the information in your table of Q3b by answering the following question. Which of the two analyses (SAW or ARQ) takes into consideration *more* of the reasons why the effective bandwidth of a datalink can be significantly less than its raw bitrate? Explain your answer briefly, with reference to your table.

[2 marks]

The time for creating a frame (T) is ignored in the ARQ formula. So the ARQ formula will overestimate the effective bandwidth of any link in which the frame-creation time is significant. For example, in the SAW analysis of the textbook, the frame creation time is T = 1 μs; if we calculate the effective bandwidth under the assumption that T = 0 μs we would predict a slightly larger bandwidth: maybe 7% larger. In cases where T were very large relative to link latency and signalling time, then the effective bandwidth could be greatly affected, and the SAW analysis would be much more accurate than the ARQ analysis.

The ARQ formula distinguishes between headers (H) and trailers (T), whereas the SAW formula is only sensitive to the sum of these two variables. However this doesn't make the SAW prediction less accurate than the ARQ prediction, because there should be exactly as many headers as trailers in any network analysis.

Award 2 marks for any understandable discussion that mentions the effect of T in the SAW analysis, and its absence in the ARQ analysis.

d. Name, and briefly describe, one *additional* reason (not modeled in either the ARQ or SAW analysis!) why the effective bandwidth of a datalink might be significantly less than its raw bitrate. **[1 mark]**

Award 1 mark for a ny one of the following reasons.

1. Time for frame retransmission. This effect can reduce the user data rate significantly.
2. Time for user interaction. Users do not send messages continuously, in most real applications, if only because they take some time to "think" about what to send next.
3. Time required to run other protocols on the link, e.g. for a select/poll in the TD800.

Q4. What octets would be transmitted in a Burroughs TD800 packet, containing the 5-character data message `Hello`, which is sent from a CPU to the data terminal at address #3? Write your characters in hex with parity, following the style of the sample TD800 packet shown in lecture slide 6 of set 3. **[2 marks]**

Using even parity check (as in the lecture slides):

Character	SOH	3	STX	H	e	l	l	o	ETX	BCC
Hex value	01	33	02	48	65	6c	6c	6f	03	71
with parity	81	***33***	82	48	65	6c	6c	6f	03	***71***

Spot-check marking: award 1 mark to students who write "33" in the second byte, and 1 mark to students who write "71" in the last byte.

Using an odd parity check (***no penalty if a student makes this assumption***):

Character	SOH	3	STX	H	e	l	l	o	ETX	BCC
Hex value	01	33	02	48	65	6c	6c	6f	03	71
with parity	01	***b3***	02	c8	e5	ec	ec	ef	83	***f1***

Q5. Slide 4 of lecture set #3 does not clearly indicate what the polled terminal in a TD800 system should do, if it receives a corrupted packet from the CPU instead of the expected ACK or NAK. Consider the following three possibilities:

a. the polled terminal might send an EOT in response,

b. the polled terminal might resend its data message, or

c. the polled terminal might not send anything in response.

Which of these three possibilities is the most appropriate choice for the TD800 protocol? Explain your reasoning by briefly describing a problematic situation that could arise in each of the other two possibilities. **[3 marks]**

The polled terminal might send an EOT in response:

This will work if the damaged acknowledgement was an "ACK" from the CPU. However, it will lead to an error if the damaged acknowledgement was a "NAK" from the CPU, as in this case the CPU will not get the required retransmission of data from the terminal. This will result in lost user data, so it is an inappropriate protocol.

The polled terminal might resend its data message:

This will lead to an error (duplicated user data) if the CPU had actually sent an "ACK", so it is inappropriate.

The polled terminal might not send anything in response:

This should (eventually) cause the CPU to time out. The CPU could then safely ask the terminal to resend the message, and the CPU would "know" that the resent message is a duplicate of whatever it last received (whether it was received correctly or incorrectly).

From the above analysis, we can see the third choice is only appropriate way to extend the TD800 protocol of slide 4, to handle the case of a damaged ACK or NAK.

Qualitative marking: 3 marks for an answer that shows clear reasoning and doesn't contain any "obvious" mistakess. 2 marks to any answer that shows a student understands that

1. ***a communication protocol is incorrect if it allows messages to be lost or duplicated, and***
2. ***that a party in a communication protocol cannot "guess" or somehow "know" what the other party really means; instead a party must base all of its decisions on the information it receives from the link or has stored in its memory.***

1 mark to any answer that shows understanding of just one of the points noted above.

Q6. A Burroughs TD800 CPU might be upgraded with an Ethernet network card, to replace the serial I/O card that handled its original 38,400 bps multi-dropped datalink. The new Ethernet card would require special-purpose "device driver" software. The driver software accepts 7-bit ASCII characters from the Burroughs TD800 link-layer protocol software. These 7-bit characters should be transmitted over the Ethernet to an appropriately-configured Burroughs TD800 terminal (which would also require an Ethernet card and special-purpose driver). The CPU's Ethernet device driver would also have to handle the back-channel: whenever this card receives one or more 7-bit ASCII characters from an Ethernet-equipped TD800 terminal, the CPU's device driver should pass these characters on to the (original, unchanged) Burroughs TD800 link-layer protocol software. After these modifications, the standard Ethernet protocol can be used on the Physical and MAC layers of a Burroughs TD800 system.

We could use a special 8-byte SNAP header to identify this special TD800 traffic on a 10 Mbps Ethernet link. If each Ethernet packet carries a single character of the TD800 protocol, would you expect the upgraded TD800 system to be significantly faster or slower than the original TD800 system? Explain. **[2 marks]**

The minimum size of an Ethernet packet is 7+1+6+6+2+8+38+4 = 72B, if it follows the structure given below (taken from the lecture slides):

| Preamble (7B) | Start Delimiter (1B) | Dest Addr (6B) | Src Addr (6B) |

Length (2B) | SNAP (8B) | Data (38B) | FCS (4B) |

In the application outlined above, each Ethernet packet will contain exactly one character (7 bits) from the TD800 protocol.

On a 10Mbps Ethernet, we might hope to transmit 72 B packets at the rate of (10 Mb/s)/((72 B/packet)*(8 b/B)) = 17000 packets/s.

This compares favourably with the maximum rate of TD800 protocol characters that can be sent on a 38400 b/s asynchronous serial link, because on an asychronous link each character will take 10 (or perhaps 11) bit-times (including start bit, stop bit and parity): (38400 b/s)/(10b/packet) = 3840 packet/s.

So the new system might run more than four times faster than the original system, if it actually sends 17000 packets/s rather than 3840 packets/s, because 17000/3840 = 4.4.

However, we cannot be certain of seeing such a large speedup. For all we know, the speed of the Burroughs system is not actually limited by the speed of its serial link; quite possibly, it spends most of its time waiting on its disk drives, users, CPU, or main-memory circuits. Also our assumptions about the Ethernet performance are suspect, as the Burroughs CPU is very slow by modern standards. It might require many milliseconds to prepare an Ethernet packet for transmission, or to extract a TD800 character from an Ethernet packet it has received. So it seems unlikely that it could drive the Ethernet at 17000 packets/second, and even 3840 packets/second might be unacheivable – so the new system might actually be slower than the original one, because of the increased CPU time required to handle Ethernet packets through an Ethernet controller, as compared to the CPU time required to handle characters through a simple UART connected to a 38400 b/s serial line.

***1 mark** to any answer that shows the student understands how the Ethernet protocol could encapsulate the TD800 protocol. This would best be done on a character-by-character basis, rather than by trying to send entire TD800 messages in a single Ethernet packet, however no penalty should be given to students who try to do something this complicated.*

***1 mark** to any answer that calculates a data rate (approx 17000 chars/sec) for the Ethernet-encapsulated TD800 protocol, and compares this with the data rate (approx 3840 chars/sec) for the original system.*
