Exam 1 (Midterm), Fall 2000
Computer Science 516
Computer Communication Networks
University at Albany

Prof. William A. Maniatty

October 26, 2000

1 Some Hints

Your professor suggests the following preparation strategies:

- This exam is open book and open notes (your own books and notes that is!). Calculators are permitted, networked devices are not.
- Write neat clean answers, since if the grader cannot understand you on the real exam, it will go badly for you.
- Show your work, if you are guessing the grader will not give much credit (even if you get lucky and guess right).
- Define your notation (you can use tables and/or diagrams).
- Set up the solution symbolically and simplify before plugging numbers in, it is easier to follow for the grader (and will get you most of the points).
- You can solve problems out of order, but keep the work for each problem in one place, and mark it clearly.

2 The Problems

1. Protocol Layering (15 %):

   (a) (5 %) What layer of TCP/IP is responsible for network wide addressing.

      The Internet Protocol Layer provides addressing (IP addresses).

   (b) (5 %) When is segmentation and reassembly used in protocol implementation?

      Segmentation and reassembly are used in packet switched systems to allow higher layers with large packet (message) sizes to be supported using lower layers with a small packet size. Large packets sent from a higher layer are first broken into smaller messages by the supporting lower layer (segmentation). The receiver's lower layer reassembles the fragments to regenerate the original packet and which is passed to the higher layer.

   (c) (5 %) On a Unix system (using the BSD TCP/IP sockets API) running many servers (say FTP, HTTP and mail) what mechanism is used to provide a unique service access point for each server?

      That is the port numbering (used in the bind system call). Each server gets a unique set of ports, so the combination of the host's network address and the port number of each server distinguish the server which the client accesses via the connect system call.

2. Network Systems programming (20 %): Assume a BSD style sockets programming environment.

   (a) (5 %) What functionality does the accept system call provide?

      The accept system call establishes a connection oriented link to the client using streaming sockets. The server must be listening to the socket bound to a port prior to accepting the connection. Once accepted, the result of the accept call is a socket suitable for reading and writing.

   (b) (10 %) How does FTP's passive mode work? How is the port number selected in a Unix environment?

      Recall that FTP uses both a control and a data connection. The data connection is established when a file retrieve or put command is invoked. What passive mode does is allow the server to specify an ephemeral address/port number combination for the data port. This is useful for security in say firewalled environments. The ephemeral port number can be obtained by specifiying a wild card on the port number to bind.

(c) (5 %) How can a program with a single thread of control wait for (and respond to) asynchronous events on multiple sockets?

The program should use the select (or poll) system call. The select system call takes arrays of sockets as its inputs, and as a result manipulates bitmapped sets indicating which file descriptors have pending activity.

3. The Physical Layer and Information Theory (20 %):

(a) (5 %) Downstream Cable Modems use QAM (quadrature amplitude modulation). Which parts of the wave are modulated by QAM?

Quadrature amplitude modulation encodes information by modulating the phase and amplitude of the wave.

(b) (10 %) Consider the distribution of grades in all 10^6 freshmen CS students students as summarized in Table 1. Suppose that the professor conducting the sample wanted to compress and transmit the entire list of grades using Huffman Coding. The grades are originally stored as a list of ASCII characters. If the dictionary generated by the Huffman code is 256 bytes, how much space would a Huffman encoded version take, and how long is the transmission time on a 50 Kbps link? (10 %)?

Grade	Number of Students (in thousands)
A	100
B	150
C	230
D	175
E	220
W	125

Table 1: Grades List for Problem 3b

The first step is to derive the Huffman coding by constructing the tree shown in Figure 1. All characters are encoded using either 2 or 3 bits, so the compression factor is then $\frac{2\times0.45+3\times0.55}{8} = 0.31875$. So the space required is:

$$\begin{aligned} \text{Compressed Data Size} &= 10^6\text{bytes} \times 0.31875 = 318750\text{bytes} \\ \text{Compressed Size} &= \text{Dictionary Size} + \text{Compressed Data Size} \\ &= 256 + 318750\text{bytes} = 319006\text{bytes} \end{aligned}$$

The transmission time at a bandwidth of 50 Kbps is:

$$\begin{aligned} \text{Transmission time} &= \frac{\text{data size}}{\text{bandwidth}} \\ &= \frac{319006\text{bytes}}{\frac{50000\text{bits}}{\text{sec}}} \times \frac{8\text{bits}}{\text{byte}} = 51.04\text{sec.} \end{aligned}$$

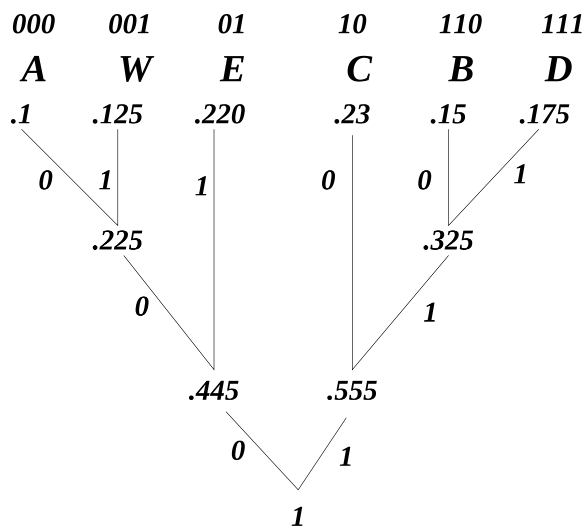

Figure 1: Huffman Coding Tree for Problem 3b

(c) (5 %) Consider the CRC which has the generating polynomial: $G(x) = x^5 + x^4 + x + 1$.Give the CRC for the message 10010110_2.

We do this by augmenting the input with 5 (the degree of the generating polynomial) zeros on the right and dividing in the bit string using modulo 2 arithmetic. The remainder by definition is the CRC.

```
               11101010
        ---------------
110011 | 1001011000000
         110011|||||||
         ------V||||||
          101101||||||
          110011||||||
          ------V|||||
           111100|||||
           110011|||||
           ------VV|||
             111100|||
             110011|||
             ------VV|
               111100|
               110011|
               ------V
                 11110 <- Remainder and CRC by definition
```

4. Multiplexing and Telephony (25 %)

(a) (10 Points) Consider a concentrator, as shown in Figure 2. Suppose

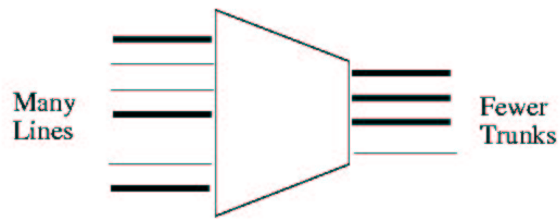

Figure 2: Multiplexing and ATM system from problem 4a

that the offered load is 9 Erlangs. What is the minimum number of output trunks required to handle that load with a blocking probability of 1 % ? Show your work.

Recall the ungraded take home exercise given in the lecture (and in the lecture notes!). In Leon-Garcia and Widjaja, the text suggests that 16 (or 17) trunks should be sufficient for that level of blocking (as does the problem in the lecture notes). So the trick is to compute the probability of blocking given a 9 Erlang offered load for both 16 and 17 trunks and to observe whether the blocking probability of 1% is in the interval of the 15 output trunk case and the 16 output trunk case. The Erlang B formula for blocking probability is:

$$P_b = \frac{\frac{a^c}{c!}}{\sum_{k=0}^{c} \frac{a^k}{k!}}$$

where $a = 9$ Erlangs is the offerred load and c is the number of output trunks. Note that since the denominator lacks a closed form solution for the sum, it would take a bit of work to compute (although a computer can do this kind of stuff real fast). We see that when $c = 16, P_b = 0.011$, which is too large and when $c = 17, P_b = 0.0058$ which is small enough, so 17 trunks are needed.

(b) (15 Points) Consider an inverse multiplexer is used for a packet switched system. If we have two gigabit fiber link inputs with a sustained 70% utilization per link (on average).

i. (5 %) How many 100 Mbps copper output links will we need if we are willing to have 85% utilization of the output links?

We solve this using dimensional analysis as follows:

$$\begin{aligned}
\text{Num. Outputs} &= \left\lceil \frac{\text{Input Bandwidth} \times \text{Input Utilization} \times \text{Num Inputs}}{\text{Output Link Bandwidth} \times \text{Output Link Utilization}} \right\rceil \\
&= \left\lceil \frac{10^9 \frac{\text{bits}}{\text{sec}} \times 2\text{links} \times 0.7}{100 \times 10^6 \frac{\text{bits}}{\text{sec}} \times 0.85} \right\rceil = \lceil 16.47 \rceil \text{links} = 17\text{links}
\end{aligned}$$

ii. (10 %) What percentage of the input must be lost by the inverse multiplexer designed in problem 4(b)i if both input links saturate (go to 100 % utilization)?

Again we need to use dimensional analysis:

$$\begin{aligned}
\text{Percent Input lost} &= \frac{\text{Rate of Input Loss}}{\text{Input Arrival Rate}} \times 100 \\
&= \frac{\text{Input Arrival Rate} - \text{Output Departure Rate}}{\text{Input Arrival Rate}} \times 100\% \\
&= \frac{(2.0 - 1.7) \times 10^9 \frac{\text{bits}}{\text{sec.}}}{2.0 \times 10^9 \frac{\text{bits}}{\text{sec.}}} \times 100\% \\
&= 15\%
\end{aligned}$$

5. Design principles and Queueing Theory (20 %): Suppose that a network interface card (NIC) implements the IP layer of the TCP/IP protocol in hardware (using a dedicated processor). Assume that when a packet is received it takes 3μsec (on average) to process the headers and trailers of received packets at the IP layer, and 6μsec (on average) for the software drivers to process the headers for the host to host layer.

   (a) (10 %) What is the maximum throughput of this system for received data if both the NIC hardware and host-to-host software can process packets in constant time? What would be the minimum response time of this system?

   Note that the system is deterministic, and has both maximum throughput and minimum response time when arrivals are also deterministic (since this avoids queueing which slows the progress of jobs through the system). As seen in Figure 3, this system is a pipelined system, so the maximum throughput is the throughput of the slowest stage, that is: max throughput $= \frac{1\text{job}}{\max(3,6)\mu\text{sec}} = \frac{1\text{job}}{6\mu\text{sec}}$. Now let's consider minimizing response time. The response time of a pipelined systems is the sum of the response time of each of its stages. The minimum response time per stage is the service time, so we get min response time $= 3\mu\text{sec} + 6\mu\text{sec} = 9\mu\text{sec}$

   (b) (10 %) Suppose that the mean interarrival time of packets into the system is 9μsec (on average), and that interarrival times and service times (both IP and host-to-host layers) are exponentially distributed. What is the expected response time and throughput of this system?

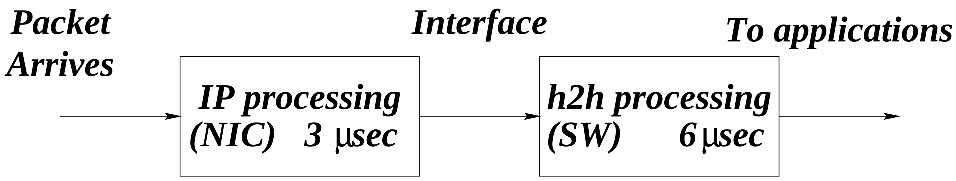

Figure 3: The Pipelined System in Problem 5a

This time the system is still pipelined, however the arrivals and service times are now bursty (due to the exponential distribution) and we have queueing as per $M/M/1$ systems, as seen in Figure 4. The time spent in the IP stage is:

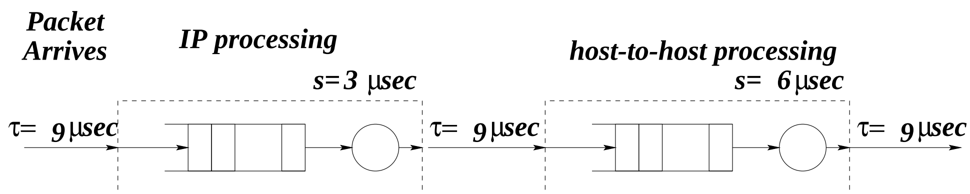

Figure 4: The System In Problem 5b

$$
\begin{aligned}
t_{ip} &= \frac{s_{ip}}{1-\rho_{ip}} \\
&= \frac{3\mu\text{sec}}{1-\frac{1}{9\mu\text{sec}}\times 3\mu\text{sec}} \\
&= 4.5\mu\text{sec.} \\
t_{htoh} &= \frac{s_{htoh}}{1-\rho_{htoh}} \\
&= \frac{s_{htoh}}{1-\lambda s_{htoh}} \\
&= \frac{6\mu\text{sec}}{1-\frac{1}{9\mu\text{sec.}}\times 6\mu\text{sec.}} \\
&= 18\mu\text{sec.} \\
t_{total} &= t_{ip}+t_{htoh} = (4.5+18)\mu\text{sec.} = 22.5\mu\text{sec} \\
\text{throughput} &= \lambda = \frac{1}{\tau} = \frac{1\text{job}}{9\mu\text{sec.}}
\end{aligned}
$$