

Exam 1 (Midterm), Fall 1999
Computer Science 416
Computer Communication Networks
University at Albany

Prof. William A. Maniatty

October 26, 1999

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 like the lecture notes if you like).
- Set up the solution symbolically and simplify before plugging numbers in, it is easier to follow for the grader.
- You can solve problems out of order, but keep the work for each problem in one place, and mark it clearly.

the lecture notes) so:

$$\text{Attenuation}_{AC} = \text{Attenuation}_{AB} - \text{Amplification}_{BC} = 5\text{dB} - 2\text{dB} = 3\text{dB} \quad (1)$$

ii. (5 points) What is the power of the signal arriving at C ?

Using the result of Problem 2(a)i, and the definition of attenuation we apply some algebra and get.

$$\text{Attenuation} = 10 \log_{10} \frac{P_{\text{send}}}{P_{\text{receive}}} \quad (2)$$

$$\frac{\text{Attenuation}}{10} = \log_{10} \frac{P_{\text{send}}}{P_{\text{receive}}} \quad (3)$$

$$10^{\frac{\text{Attenuation}}{10}} = \frac{P_{\text{send}}}{P_{\text{receive}}} \quad (4)$$

$$P_{\text{receive}} \times 10^{\frac{\text{Attenuation}}{10}} = P_{\text{send}} \quad (5)$$

$$P_{\text{receive}} = \frac{P_{\text{send}}}{10^{\frac{\text{Attenuation}}{10}}} \quad (6)$$

$$= \frac{100\text{mW}}{10^{\frac{3\text{dB}}{10}}} \approx 50.1\text{mW} \quad (7)$$

(b) (5 points) The ascii encoding of the letter 'z' is 0x7a (hexadecimal). Draw the unipolar and bipolar encoding of this character assuming that the set of valid voltages is $\{-V, 0, +V\}$ and one byte takes T units of time to transmit.

The solution is shown in Figure 2.

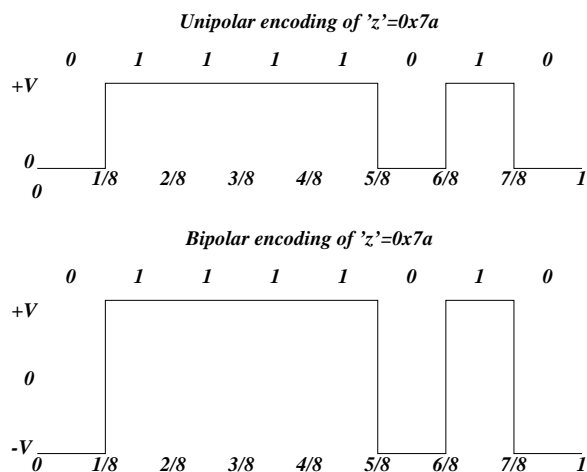


Figure 2: Solution to Problem 2b

3. Channel Performance Analysis (20 points): Suppose you are a developer and you want to upgrade a remote software installation as soon as possible, and it is now 2 p.m. The upgrade requires transferring 10 gigabytes of programs and data to your customer's machine. You can send the data over a 10 Mbps network, but you only get 10% of that bandwidth on average. Otherwise, you could transfer the data to a tape and have an overnight delivery arrive at the customer site by 9:00 A. M. tomorrow. It takes an additional hour to read the tape into the customer's machine after it arrives. You can assume you and your customer are in the same time zone. Which solution will get your customer up and running faster, the overnight delivery service, or sending the files over the wire.

In this problem we want to select the minimum of two times:

- (a) The time to transfer the data over a wire, which can be solved using:

$$\text{Wire Transfer Time} = \frac{\text{Message Size}}{\text{Effective Bandwidth}} \quad (8)$$

$$\text{Effective Bandwidth} = \frac{10 \times 10^6 \text{bits}}{\text{sec}} \times \frac{0.1 \text{ } 10^6 \text{bits}}{\text{sec}} \quad (9)$$

$$\text{Wire Transfer Time} = \frac{10 \times 10^9 \text{bytes} \times \frac{8 \text{bits}}{\text{byte}}}{\frac{10^6 \text{bits}}{\text{sec}}} = 8000 \text{sec} \approx 22.2 \text{h} \quad (10)$$

- (b) The time to transfer the data using overnight delivery can be solved:

$$\begin{aligned} \text{Mail Transfer Time} &= \text{Shipping Time} + \text{Tape Reading Time} \\ &= 20 \text{hours} + 1 \text{hour} = 21 \text{hours}. \end{aligned} \quad (12)$$

And we see that

$$\min(\text{Wire Transfer Time}, \text{Mail Transfer Time}) = \text{Mail Transfer Time} = 21 \text{hours} \quad (13)$$

So using the overnight delivery is faster!

4. ATM (10 Points): Recall that voice is sampled by telephone networks at 64 kbps.

- (a) (5 points) Assuming that the cost of putting the values in a cell, adding a header and handing the packet off to the network are negligible. Derive the packetization delay for a 48 byte cell (and show your work)!

The delay in constructing the packet is:

$$\begin{aligned} \text{Packetization Delay} &= \frac{\text{Cell Size}}{\text{Sampling Rate}} \quad (14) \\ &= \frac{48 \text{bytes} \times \frac{8 \text{bits}}{\text{byte}}}{64 \times 10^3 \frac{\text{bits}}{\text{sec}}} = 6^{-3} \text{sec} \quad (15) \end{aligned}$$

- (b) (5 points) Suppose that it took 1 msec to hand off a packet to the network. Also assume that adding a header and putting the values in the cell take negligible time. Then what would be the packetization delay if 32 byte cells were used?

In the previous problem, the handoff time was negligible, it is nonnegligible here.

$$\begin{aligned} \text{Packetization Delay} &= \frac{\text{Cell Size}}{\text{Sampling Rate}} + \text{Handoff Time} \\ &= \frac{32 \text{ bytes} \times \frac{8 \text{ bits}}{\text{byte}}}{64 \times 10^3 \frac{\text{bits}}{\text{sec}}} + 1 \text{ msec} = 5^{-3} \text{ sec} \end{aligned}$$

5. Protocol Layering (20 points):

- (a) (5 points) How is randomization used in ethernet protocols (at the data link layer), and how do token ring networks avoid randomization?

Randomization in ethernet protocols is used in CSMA/CD schemes, where collisions between packets can occur. When both stations participating in a collision back off, a small random increment must be added to prevent collisions on retransmission. Token rings avoid this, since they require a station to own the token in order to transmit.

- (b) (5 points) What services does the network layer provide?

The network layer logically concatenates host to host connections (at the data link layer) providing end-to-end connectivity. This includes services such as:

- Network Wide Addressing
- Segmentation and Reassembly of Messages
- Error Detection
- Routing Table Construction
- Packet forwarding and scheduling
- Dropping of packets (if needed)

- (c) (10 Points) Show the PPP frame containing the following C literal as the data sent using the IP protocol

"I aced 0x7e this 0x7d test!"

Where 0x7e and 0x7d are the corresponding byte values injected into the stream of text.

flag	addr	control	protocol	information	crc	flag
0x7e	0xff	0x03	0x0021	I aced \0x7d\0x5e this \0x7d\0x5d test		0x7e

6. Design principles and Queueing Theory (10 points): 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 $5\mu\text{sec}$ (on average) to process the headers and trailers of received packets at the IP layer, and $12\mu\text{sec}$ (on average) for the software drivers to process the headers for the host to host layer.

- (a) (5 points) 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?

The system is pipelined, as seen in Figure 3. Since the

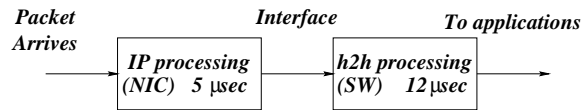


Figure 3: Pipelined System described in Problem 6a

throughput of a pipeline is equal to the throughput of its slowest stage:

$$\begin{aligned} \text{Max Packet Throughput} &= \min(\text{NIC Throughput}, \text{TCP Throughput}) \\ &= \min\left(\frac{1\text{packet}}{5\mu\text{sec}}, \frac{1\text{packet}}{12\mu\text{sec}}\right) \end{aligned} \quad (19)$$

$$= \frac{1\text{packet}}{12\mu\text{sec}} \quad (20)$$

- (b) If packets arrive every $20\mu\text{sec}$ on average (exponentially distributed), and the IP and host-to-host layers have an exponential distribution, what is the mean time to service a packet?

The system is as diagrammed in Figure 4. Note that for

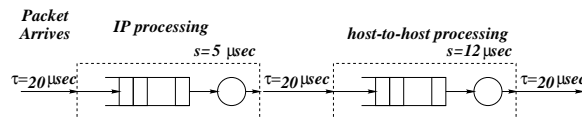


Figure 4: Queueing System described in Problem ??

stability, the interarrival times, $\tau = \frac{1}{\lambda} = 20\mu\text{sec}$ out of each stage must match. If we call t_{ip} the mean time spent in the IP processing stage, and the host-to-host layers take time

t_{h2h} :

$$t_q = t_{ip} + t_{h2h} \quad (21)$$

$$= \frac{s_{ip}}{1 - \rho_{ip}} + \frac{s_{ip}}{1 - \rho_{ip}} \quad (22)$$

$$= \frac{s_{ip}}{1 - \frac{s_{ip}}{\tau}} + \frac{s_{ip}}{1 - \frac{s_{ip}}{\tau}} \quad (23)$$

$$= \frac{5\mu\text{sec}}{1 - \frac{5\mu\text{sec}}{20\mu\text{sec}}} - \frac{12\mu\text{sec}}{1 - \frac{12\mu\text{sec}}{20\mu\text{sec}}} \quad (24)$$

$$= \frac{20}{3}\mu\text{sec} + 30\mu\text{sec} = \frac{110}{3}\mu\text{sec} \quad (25)$$

7. Network Systems programming (15 points):

- (a) (10 points) Why do many flavors of unix support scatter/gather programming using `readv` and `writev` systems calls?

Scatter/Gather programming allows us to replace multiple systems calls doing data transfer between discontinuous data locations and a contiguous data location with a single system call. This reduces the number of systems calls, which improves performance. Additionally the systems programmers can optimize across multiple systems calls (e.g. by prefetching data).

- (b) (5 points) What functionality does the `bind` system call provide?

The `bind` systems call is part of the Unix (BSD) sockets interface. `bind` is called by a server program to give the socket an externally visible name (a port number typically).