Midterm — Oct. 12, 2007

EE122: Introduction to Communication Networks

Fall 2007

Prof. Paxson
Department of Electrical Engineering and Computer Sciences
College of Engineering
University of California, Berkeley

*INSTRUCTIONS: This examination is **CLOSED BOOK/CLOSED NOTES**. There is no need for detailed numeric calculations, and no calculation or informational aids (PDAs, laptops, cell phones, etc.) are allowed—please put them away. You may use one 8.5" by 11" double-sided crib sheet, as densely packed with notes, formulas, and diagrams as you wish.*

There are 8 problems, each worth either 15 or 20 points. The entire test is worth 150 points.

The test is 80 minutes long. All work needs to be done on the attached pages.

In general, if something is unclear, write down your assumptions as part of your answer. If your assumptions are reasonable, we will endeavor to factor them in when grading the question. If necessary, raise your hand and a TA or the instructor will come to you. Please try not to disturb the students taking the examination around you.

Name (please print clearly):

SID:

Signature:

Please circle the last two letters of your ee122- login:

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z

Problem	1	2	3	4	5	6	7	8	Total
Score	/ 20	/ 15	/ 20	/ 20	/ 20	/ 15	/ 20	/ 20	

1. **Alphabet soup.**

   For each of the concepts given below, list which of the following acronyms apply:

   > 802.3, ARQ, CDN, CIDR, CNAME, CRC, CSMA, CSMA/CD, DF, FDMA, FTP, HTTP, ICANN, IMAP, IPv4, IPv6, LAN, MIME, MTU, MX, Mbps, NIC, NRZ, NRZI, NS, PTR, RFC, RIR, RTT, SMTP, SONET, TCP, TDMA, TLD, TTL, UDP, URI

   Not all acronyms are used. Unless otherwise stated, there is one acronym per concept. (1 point per item)

   (a) Used to provide a web site's content from a large number of distributed servers:

   (b) A way of interpreting IP addresses as having an initial (variable-length) network prefix, plus the remaining bits identifying a host within that network:

   (c) An Internet standards document:

   (d) The IEEE standardization of Ethernet:

   (e) 2 types of layer-4 transport protocols:

   (f) A DNS resource record used to identify email servers associated with a domain:

   (g) An application-layer protocol that uses an additional connection between the same client and server for each data object transferred:

   (h) A term referring to a host's network adaptor:

   (i) 4 types of DNS resource records:

   (j) The largest sized packet that can be sent across a link (or a network path) without requiring fragmentation:

   (k) The style of MAC (Media Access Control) protocol used by Ethernet:

   (l) A scheme for computing a checksum value over a block of data in order to detect bit errors:

   (m) A standardized scheme for encoding different types of email (and Web) content:

   (n) An entity responsible for allocating Internet address blocks for a large region:

   (o) The time it takes to send a packet to a destination and hear a response back from it:

   (p) The main protocol used to transmit email:

   (q) The term for primary DNS zones such as **.com** or **.uk** :

   (r) A counter in the IP header that is decreased at each hop; if it reaches 0, the packet is discarded (also refers to how long to keep a DNS response in a local cache):

(s) A class of mechanisms used for transport protocols that achieve reliability by retransmitting missing data. Particularly used in reference to simple reliable schemes such as Stop-and-Wait:

(t) A way of sharing a link's capacity among a group of senders in which each sender is assigned its own frequency to use when transmitting, which it can use to transmit whenever it pleases:

2. **MAC Protocols.**

For each of the following types of media access control protocols, place an X if the given attribute applies to it. (15 points)

Attribute	Ethernet	Slotted Aloha	Token Passing	TDMA	FDMA
Uses Collision Detection					
Uses Carrier Sense					
Uses Exponential Backoff					
A single node can potentially use close to all of the capacity					
Nodes have to wait for it to be their turn to send					
Operates efficiently when many nodes all have data to send					
Vulnerable to failure of a single node					
Uses randomness to avoid synchronization					

3. **Multiple Data Transfers.**

Using a Web browser, you visit the web site for `www.hamburger.com`. The base HTML page for the main page `www.hamburger.com` is 30,000 bits. Once the base HTML page is fetched, it contains URL references for the following embedded images:

`http://www.hamburger.com/burger_banner.jpg`	15,000 bits
`http://www.hamburger.com/lettuce.jpg`	5,000 bits
`http://www.hamburger.com/mmm_bacon.jpg`	10,000 bits
`http://www.hamburger.com/veggie.jpg`	10,000 bits
`http://www.hamburger.com/disclaimer.txt`	5,000 bits
`http://www.hamburger.com/royale_with_cheese.jpg`	35,000 bits

Your Web browser uses the HTTP protocol to download the base page and the embedded objects. Make the following assumptions:

- At most 10,000 bits of data fits into a single packet. You can ignore the overhead of any headers or framing.
- You must first download the entire base page before you can start fetching the embedded images.
- HTTP requests are 1,000 bits in size.
- Any new connection to a machine requires a connection-establishment handshake.

  For this problem, you do not need to worry about closing connections, and you can ignore the delay introduced in acknowledging the final data packet sent by the server to your browser.
- All senders use windows of 20,000 bits.
- No packets are lost.

(a) For the initial transfer of the home page, how many RTTs are required, and what occurs during each of them? (5 points)

(b) How quickly (in terms of RTTs) can your browser download the base page for `www.hamburger.com` and all embedded objects if the browser uses:

i. One connection per item, with up to 4 concurrent connections. (5 points)

ii. A single persistent, non-pipelined connection. (5 points)

iii. A single pipelined connection. (5 points)

4. **Performance of reliable data transfer.**

Consider two nodes, A and B. Suppose the network path from A to B has a bandwidth of 5 KB/s (5,000 bytes per second) and a propagation time of 120 msec. The path in the reverse direction, from B to A, has a bandwidth of 10 KB/s and a propagation time of 80 msec.

Let data packets have a size (including all headers) of 500 bytes and acknowledgment packets a size of 100 bytes.

(a) Give a numeric expression for the throughput A can achieve in transmitting to B using Stop-and-Wait. You can treat a 500-byte data packet as transferring 500 bytes of useful data (that is, ignore that it's a bit less due to the headers). (5 points)

(b) Give a numeric expression for the size of the window, in terms of number of data packets, that A must use in order to transfer its data as fast as possible, if A instead uses Sliding Window. (5 points)

(c) What is the maximum rate A can achieve? (5 points)

(d) If the bandwidth of the path from B to A drops to 100 bytes/sec, can A still achieve this rate? If so, roughly how much smaller or bigger is the new window size? If not, what is the new limit on the rate A can achieve? (5 points)

5. **Encoding.**

Consider a link which has two levels, **hi** or **lo**. We wish to transmit the bit sequence **1110** across this link. Assume that the bit we sent most recently was **0**, and when we finished transmitting it the link was at the **lo** level.

(a) For the following patterns of signals, indicate whether they correspond to NRZ, NRZI, Manchester, or 4-bit/5-bit encoding. (2 points each)

i. **hi**, **lo**, **hi**, **hi**, **hi**.

ii. **hi**, **hi**, **hi**, **lo**.

iii. **hi**, **lo**, **hi**, **lo**, **hi**, **lo**, **lo**, **hi**.

iv. **hi**, **lo**, **hi**, **hi**.

(b) Consider an alternate encoding scheme that represents a **1** bit using a single transition and a **0** bit using two transitions.

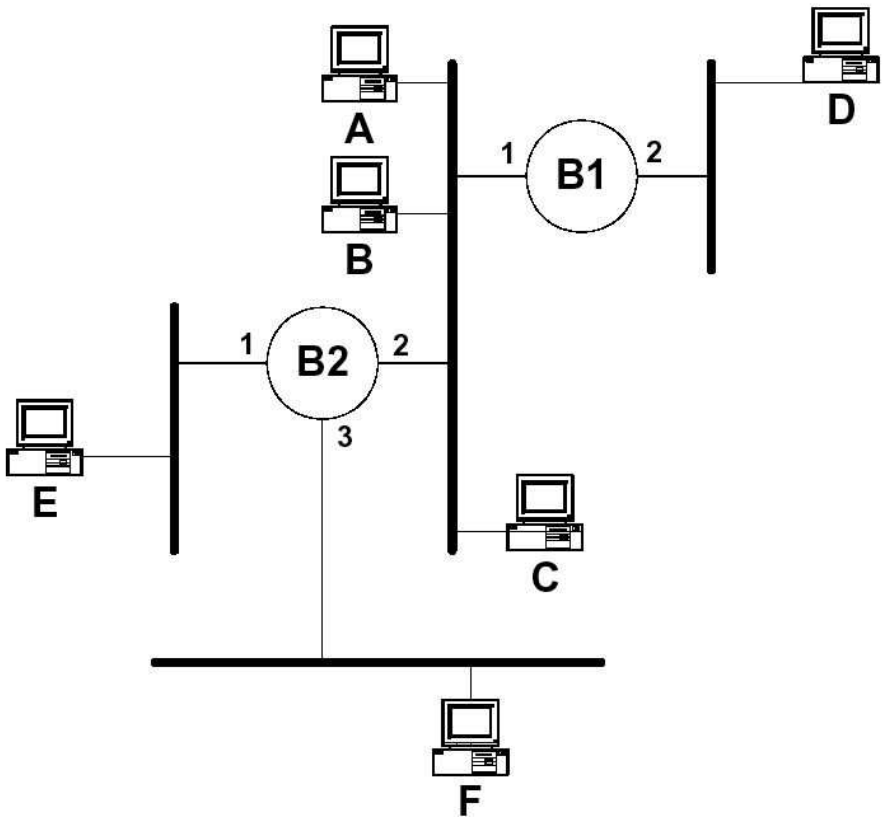
i. Write down the **hi**/**lo** representation of **1110**. (4 points)

ii. What advantage does this scheme offer over NRZI? (4 points)

iii. What disadvantage does this scheme have compared to NRZI? (4 points)

6. **Bridges / Switches.**

Consider the network of learning bridges (switches) shown in the following figure:



Show the forwarding table in each of the bridges after the following transmissions (which occur in the given order), assuming each table starts out empty:

(a) C sends to A

(b) F sends to E

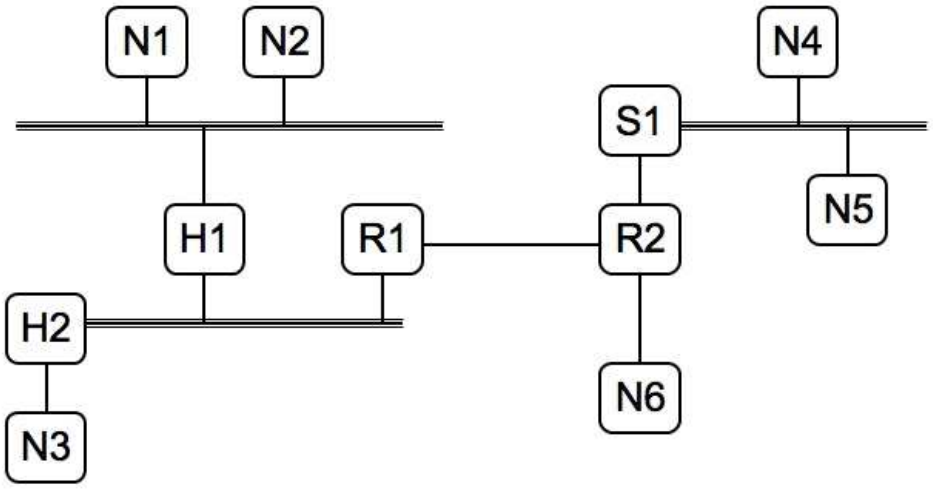
(c) E sends to F

(d) D sends to B

Give a table for each bridge, each with two columns: destination and port number, showing how the bridge would forward traffic. (15 points)

7. **Error detection.**

Consider the following topology, where a node labeled with **N** denotes an end system, **H** denotes a hub, **S** denotes a switch and **R** denotes a router:



All links are Ethernet.

Suppose **N1** uses TCP to send a 1 KB message to **N5**. The TCP connection has already been established, so this message is sent in a single packet.

(a) When the frame holding the packet arrives at **N5**: (8 points total)

i. Does it have the same Ethernet checksum as the frame holding the packet had when **N1** sent it? If not, why not?

ii. Does it have the same IP checksum as the original did? If not, why not?

iii. Does it have the same TCP checksum as the original did? If not, why not?

(b) Suppose when **H1** processes the frame it introduces a single bit error. At which nodes (i.e., any of the end systems, hubs, switches, or routers) will the errored packet appear? Here, "appear" means the frame arrives at the node's adapter, whether or not the adapter will then accept the frame. (3 points)

(c) Suppose instead that after **N1**'s kernel constructs the TCP header, but before it constructs the IP header, a single bit in the 1 KB message gets flipped in error. At which nodes will the errored packet appear? (3 points)

(d) Suppose instead that after **N1**'s kernel constructs both the TCP and IP headers, but before it constructs the Ethernet header and trailer, a single bit in the IP header gets flipped in error. At which nodes will the errored packet appear? (3 points)

(e) For this last case (**N1**'s kernel processing flips a single bit in the IP header), suppose that in addition when **H1** processes the frame it also flips a single bit in the IP header. Under what circumstances, if any, can the packet arrive at **N5**? (3 points)

8. **Framing.** In networking, being able to determine the beginning and ending of a message is termed *framing*. Framing issues come up at both lower layers of the networking stack and higher layers.

   (a) One style of framing is to precede a message with a *count* of its length. (4 points)

      i. When using counts for link-layer framing, what problem can arise?

      ii. When using this style to send application-layer messages over TCP, does this problem still arise? Why or why not?

   (b) Another style of framing is to use a *sentinel* value. (6 points)

      i. When using sentinels for link-layer framing, what problem can arise?

      ii. Briefly describe a solution for this problem.

      iii. When using this style to send application-layer messages over TCP, does this problem still arise? Why or why not?

   (c) Consider internetwork-layer datagrams sent using IP. (6 points)

      i. What style of framing does IP use for the datagrams it transmits?

ii. What solution, if any, does it use to address the problem you identified above that can arise for this style?

iii. How does a host that receives an IP datagram know what type of transport protocol information is inside the datagram?

(d) Application-layer protocols are free to use a wide range of framing techniques. (4 points)

i. What sort of framing does SMTP use to figure out where an email message being transferred ends?

ii. How does FTP indicate when it has finished transferring a file?

iii. What sort of framing does HTTP use for header information in requests and responses?

iv. Give an example of a type of framing HTTP uses for figuring out where an *item* being transferred ends. (Here an "item" refers to the object returned in response to a `GET` request, rather than the headers returned by the server.)