University of California at Berkeley
College of Engineering
Department of Electrical Engineering and Computer Science

EECS 122 I. Stoica
Fall 2008

FIRST MIDTERM EXAMINATION
Monday, 13 October 2008

INSTRUCTIONS—READ THEM NOW! This examination is CLOSED BOOK/CLOSED NOTES. There is no need for calculations, and so you will not require a calculator, Palm Pilot, laptop computer, or other calculation aid. 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. The examination has been designed for 80 minutes/80 points (1 point = 1 minute, so pace yourself accordingly). All work should 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 grade the question based on them. If necessary, of course, you may 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.

We will post solutions to the examination as soon as possible, and will grade the examination as soon as practical, usually within a week. Requests for regrades should be submitted IN WRITING, explaining why you believe your answer was incorrectly graded, within ONE WEEK of the return of the examination in class. We try to be fair, and do realize that mistakes can be made during the regarding process. However, we are not sympathetic to arguments of the form "I got half the problem right, why did I get a quarter of the points?"

_______________________________________ SID: _____________________________
(Signature)

_______________________________________ Discussion Section (Day/Time): ________
(Name—Please Print!)

QUESTION	POINTS ASSIGNED	POINTS OBTAINED
1	10	
2	10	
3	10	
4	20	
5	20	
6	10	
7 (bonus)	5	
TOTAL	80 (+5)	

Question 1. *Miscellaneous (10 points)*
For each of the following statements, indicate whether the statement is True or False, and provide a very short explanation of your selection (2 points each).

a. End-to-end packet delay of cut-through routers is smaller than of store & forward routers. T F
Rationale:

True. With cut-though routers a packet starts being transmitted as soon as its header is processed; with store & forward routers a packet is first received in its entirety before being forwarded.

b. According to end-to-end argument reliability must be implemented at networking layer. T F
Rationale:

False. Since reliability at the networking layer cannot ensure the application end-to-end reliability, and since it may hurt the latency of applications that do not require reliability (e.g., voice over IP), according to the end-to-end argument the networking layer shouldn't provide reliability.

c. Pipelined requests reduce the response time in HTTP. T F
Rationale:

True. Pipelining allows the client to send a new request before receiving the reply for the previous requests.

d. CIDR allocates IP addresses less efficiently than Classful Addressing. T F
Rationale:

False. CIDR enables finer grain address allocation, thus improving address allocation efficiency over Classful Adressing.

e. In the OSI model, the transport layer can *directly* invoke (use) the data link layer. T F
Rationale:

False. In the OSI model a layer can only use the service provided by the layer below it. In this case, the transport layer can only use the service provided by the networking layer.

Question 2. *Packet transmission (10 points)*
Two nodes, A and B, communicate through a store & forward network. Node A is connected to the network by a 10Mbps link, while node B is connected by a 5Mbps link. Node A sends two back-to-back packets of 1000bits each. The difference between the arrival times of the two packets at B is 1ms. What is the smallest capacity of a link along the path between A and B?

Note: Assume that there are no other packets in the network except the ones sent by A, and ignore the packet processing time. Assume both packets follow the same path, and they are not reorderd. The arrival time of a packet at a node is defined as the time when the last bit of the packet has arrived at that node.

Solution: Since packets are sent back-to-back, the difference between the arrival times of the packets at B represents the transmission time of the second packet on the slowest link in the path. Thus, the capacity of the slowest link is 1000bits/1ms = 1Mbps.

Question 3. *Routing (10 points)*
An IP router (which does longest prefix match to route packets) has the following forwarding table, mapping prefixes to outgoing interfaces:

128.2.0.0/24 → interface 1
128.2.1.0/24 → interface 2
128.2.2.0/24 → interface 2
128.2.3.0/24 → interface 3

But space in the forwarding table might be scarce. Suppose, for example, my IP router only has space for 3 prefixes in its forwarding table. Produce a forwarding table with only 3 entries which is equivalent to the one above -- that is, each IP address will still be mapped to same outgoing interface as in the above 4-entry table. Assume the router uses the longest prefix matching to match incoming packets to interfaces.

______________ / __ → interface ___

______________ / __ → interface ___

______________ / __ → interface ___

Solution:

128.2.0.0/22 ---> interface 2
128.2.0.0/24 ---> interface 1
128.2.3.0/24 ---> interface 3

Question 4. *Queueing (20 points)*
Consider an infinite queue that can send data at 10Kbps. Assume the following arrival traffic:

- During every *odd* second the queue receives an 1000bit packet every 50ms
- During every *even* second the queue receives no data.

Assume an interval *I* of 10sec starting with an odd second (i.e., a second in which the queue receives data). At the beginning of interval *I* the queue is empty.

(a) (5 points) What is the maximum queue size during interval *I*?
(b) (5 points) What is the average time (delay) a packet spends in the queue during interval *I*?

Now assume that during odd seconds the queue receives 1000bit packets every 25ms (instead of every 50ms), and during even seconds it still receives no data. For this traffic patterns answer the same questions:

(c) (5 points) What is the maximum queue size during interval *I*?
(d) (5 points) What is the average time (delay) a packet spends in the queue during interval *I*?

Solution:
(a) 10packets. There are 20packets arriving during the 1st second and 10 packets sent at the end of that second. Thus at the end of 1st second there are 10packets in the queue. All the 10packets will be sent at the end of 2nd second (since no new packets are received). Thus, at the end of 2nd second the queue size is 0. After that the process repeats.

(Note: The following alternate answers: 11packets, 10Kb, and 11Kb all received maximum points. The 11 packets and 11Kbps assume that at a time when a packet is received and another one is sent out, the received packet is already in the queue as the other packet is sent out.)

(b)

- 1st packet arrives at time 0 and starts being transmitted immediately, at time 0. → delay 0
- 2nd packet arrives at 0.05s and starts being transmitted at 0.1s (after the first packet) → delay 0.05s
- 3rd packet arrives at 0.1s and starts being transmitted at 0.2s (after first two packets) → delay 0.1s
- 4th packet arrives at 0.15s and starts being transmitted at 0.3s (after first three packets) → delay 0.15s
- …
- kth packet arrives at (k-1)*0.05s and starts being transmitted at (k-1)*0.1s → delay (k-1)*0.05s

This process repeats every 2 seconds.

Thus, the average delay of the first 20 packets is (0 + 1 + 2 + … + 19)*0.05s/20 = 10*19*0.05s/20 = 0.475s

Alternate solution that approximates the average delay: We use Little's theorem: avg_delay = avg_number_of_packets/arrival_rate. During an odd second the number of packets in the queue increases *linearly* from 0 to 10 and during the next second it decreases from 10 to 0. This means that the average number of packets in the queue is 5. Over an odd and an even second the average arrival rate is (20+0/)2 = 10. Then, according to Little's theorem avg_delay = avg_number_of_packets/arrival_rate = 5/10 = 0.5sec.

(Note: The following answers also received maximum points: 0.575s and 0.5s.)

(c) 110packets. In this case the queue is never empty. During the first 9 seconds of interval *I* there are 5s*(1s/25ms) = 200packets received and 90packets sent out. Thus, at the end of the 9th second there are 110packets in the queue.

(Note: The following answers also received the maximum number of points: 111packets, 110Kb, and 111Kb.)

(d)

Packets received during 1st second

- 1st packet arrives at time 0 and starts being transmitted immediately, at time 0. → delay 0
- 2nd packet arrives at 0.025s and starts being transmitted at 0.1s (after the first packet) → delay 0.075s
- 3rd packet arrives at 0.05s and starts being transmitted at 0.2s (after first two packets) → delay 0.15s
- 4th packet arrives at 0.075s and starts being transmitted at 0.3s (after first three packets) → delay 0.225s
- …
- kth packet arrives at (k-1)*0.025s and starts being transmitted at (k-1)*0.1s → delay (k-1)*0.075s

The average delay of the packetsin the first two seconds is (0 + 1 + … + 39)*0.075s/40 = 20*39*0.075s/40 = 1.4625s

Packets received during 3rd second: note that at the beginning of the 3rd second there are still 20packets in the queue

- 1st packet arrives at time 0 and starts being transmitted immediately, at time 2s → delay 2
- 2nd packet arrives at 0.025s and starts being transmitted at 2+0.1s → delay 2+0.075s
- 3rd packet arrives at 0.05s and starts being transmitted at 2+0.2s → delay 2+0.15s
- 4th packet arrives at 0.075s and starts being transmitted at 2+0.3s → delay 2+0.225s
- …
- kth packet arrives at (k-1)*0.025s and starts being transmitted at 2+(k-1)*0.1s → delay 2+(k-1)*0.075s

The average delay of the packetsin the first two seconds is (0 + 1 + … + 39)*0.075s/40 = 20*39*0.075s/40 = 3.4625s

…

Packets received during 9th second: note that at the beginning of the 9th second there are still 80packets in the queue

- 1st packet arrives at time 0 and starts being transmitted immediately, at time 8s → delay 2
- 2nd packet arrives at 0.025s and starts being transmitted at 8+0.1s → delay 8+0.075s
- 3rd packet arrives at 0.05s and starts being transmitted at 8+0.2s → delay 8+0.15s
- 4th packet arrives at 0.075s and starts being transmitted at 8+0.3s → delay 8+0.225s
- …
- kth packet arrives at (k-1)*0.025s and starts being transmitted at 8+(k-1)*0.1s → delay 8+(k-1)*0.075s

The average delay of the packetsin the first two seconds is (0 + 1 + … + 39)*0.075s/40 = 20*39*0.075s/40 = 9.4625s

Thus, the average delay over 10 seconds is: (1.4625+3.4625+5.4625+7.4625+9.4625)/5 = 5.4625

Alternate solution that approximates the average delay: The average arrival rate is 40packets/2sec = 20packets/sec. During the 1st sec the number of packets in the queue increases linearly from 0 to 30, thus the average number of packets in the queue in the 1st sec is 15. During 2nd second the queue decreases linearly from 30 to 20, thus the average number of packets in the queue is 25, and the average number of packets in the queue over the first two seconds is 20. During 3rd and 4th seconds the process repeats with the difference that there are 20 packets in the queue at the beginning of the 3rd second. Thus, the average number of packets in the queue during the 3rd and 4th seconds is 20+20 = 40. Similarly, the average number of packets during the 5th and 6th seconds is 40+20 = 60, during the 7th and 8th seconds 60+20=80, and during the 9th and 10th seconds is 80+20=100. Thus the average number of packets over the entire interval *I* is (20+40+60+80+100)/5 = 60. According to the Little's theorem avg_delay = avg_number_of_packets/arrival_rate = 60/10 = 6sec. (Note: In general the average number of packets over the interval defined by the 2*k-1 and 2*k seconds is k*20, where k>= 1.)

Question 5. *Flow control (20 points)*
Sender S communicates with receiver R using a flow control protocol with a window of 3 packets. This means that S can send at most 3 unacknowledged packets at a time. Each packet has a sequence number starting from 1. R always acknowledges a packet by sending back to A the sequence number of that packet (i.e., when R receives a packet with sequence number 2 it sends an acknowledgement (ack) containing 2 to S.)

Ignore packet transmission times, and assume that neither the packets nor the ack are reordered in the network.

Let RTT denote the round-trip time between S and R. S uses two mechanisms to retransmit a packet:

- "timeout": S retransmits a packet if it has not received an ack for it within T seconds after sending the packet, where T > RTT.
- "out-of-order ack": S retransmits an unacknowledged packet *p* when it receives an ack with a sequence number higher than *p*. For example, if packet 3 hasn't been acknowledged, and S receives ack 4, then S assumes that R hasn't received packet 3 and retransmits it immediately.

Assume S wants to transfer a file that spawns exactly 8 packets to R as fast as possible. During the transfer *at most one* packet (or ack) is lost. For all questions below express your answer in terms of T and RTT.

(a) (7 points) What is the minimum time it could take to transfer the file? The file transfer time is the time between the moment S sends the first packet and the moment it receives the last ack.

(b) (7 points) What is the maximum possible file transfer time assuming S uses only the "timeout" retransmission mechanism? Please give a scenario which achieves the maximum transfer time. This scenario should indicate which packet (or ack) is dropped, if any.

(c) (6 points) Repeat question (b) but now assume that S uses both the "timeout" and "out-of-order ack" retransmission mechanisms.

Solution:

(a) 3*RTT: No packet or ack is lost.
(b) 3*RTT + T: A packet in the last window is lost.
(c) Same as (b), if the last packet is lost.

Question 6. *Media Access Protocol (10 points)*
Three users X, Y and Z use a shared link to connect to the Internet. Only one of X, Y or Z can use the link at a given time. The link has a capacity of 1Mbps. There are two possible strategies for accessing the shared link:

- TDMA: equal slots of 0.1 seconds.
- "Taking turns": adds a latency of 0.05 seconds before taking the turn. The user can then use the link for as long as it has data to send. A user requests the link only when it has data to send.

In each of the following two cases, which strategy would you pick and why?

(a) (5 points) X, Y and Z send a 40KBytes file every 1sec.
(b) (5 points) X sends 80KBytes files every 1sec, while Y and Z send 10KBytes files every 1sec.

Solution:
(a) TDMA. Why: Each of the users generate a load of 40KB/s = 0.32Mbps, which can be fully transmitted given the share of 0.33Mbps available per user when partitioning the channel with TDMA. Taking turns on the other hand does not offer enough capacity for all the files to be transmitted: 3*0.32 + 3*0.05 = 1.11s > 1s, and would incur extra overhead.
(b) Taking Turns
Why: First, by using TDMA, X does not have enough capacity to transmit, 80KB/s = 0.640Mbps > 0.33Mbps. Second, with TDMA, Y and Z waste 3 out of 4 slots. On the other hand, when taking turns, there is enough capacity to transmit all the data: 0.64+0.05+0.08+0.05+0.08+0.05=0.95s.

Question 7. *DNS (bonus question: 5 points)*
Agent McSnooperson of the FBI agent walks into Café Strada, which has wireless Internet and runs its own local DNS server. The agent wants to figure out whether anyone in the café has recently accessed www.illicitinfo.uk, before Agent McSnooperson arrived. How can the agent do this using standard computer networking tools, and without breaking into any computers or servers?

Solution:
Bellow are three possible solutions. All solutions use the fact that if the site has been accessed recently, it will be in the local DNS server's cache. Here are three ways to determine whether the name is in the cache.

1. Time nslookup www.illicitinfo.uk. If it's in the cache, this will complete faster than if it isn't.

2. Use dig to query www.illicitinfo.uk at the local DNS server. If the returned TTL is large (about equal to the value when querying the domain from a DNS server which doesn't have it cached), then it wasn't cached. If the returned TTL is smaller, then it was cached.

3. Use dig, with recursion explicitly turned off, to query www.illicitinfo.uk at the local DNS server. If the server returns a mapping to an IP address (Type A record for www.illicitinfo.uk), then it was in the cache. Otherwise, it wasn't.