EECS489 Computer Networks, Midterm (Fall 2005)
SOLUTIONS

Instructions: You are allowed to use one sheet of notes (front and back). This exam is closed book, no computers are allowed. You can use a calculator. Read the entire exam through before you begin working. Work on those problems you find easiest first. Read each question carefully, and note all that is required of you. Please keep your answers clear and concise, and state all of your assumptions carefully.
Please write out the details of how you reach your final answer in order to get partial credit.

You are to abide by the University of Michigan/Engineering honor code. Please sign below to signify that you have kept the honor code pledge. Honor code pledge: I have neither given nor received aid on this exam.

Name: ________________________________

Signature: ________________________________

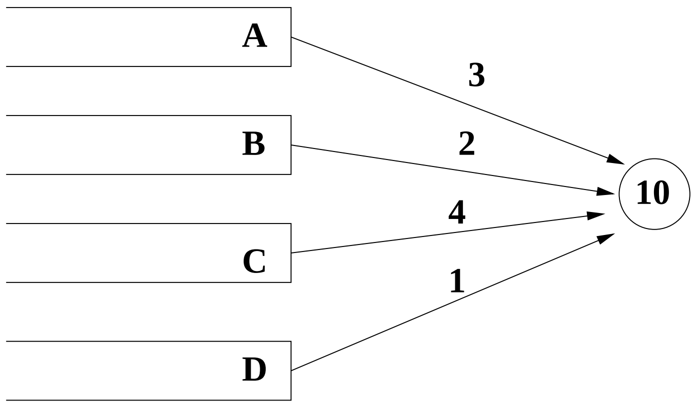
Uniqname: ________________________________

Problem 1: Router Queue Management

[4 pts] (1) Explain why fair queuing is not adopted in today's Internet. What is the most prevalent queuing discipline today and what disadvantage does it have? Given that there is no guarantee in queuing delay, what can applications do?

Answer: Fair queuing requires keeping per flow state and therefore is too expensive in core Internet routers. Drop-tail is the most prevalent queuing discipline. The disadvantage is that there is no fairness. Applications can use adaptive application-layer mechanisms such as multiple connections, redundant coding for the data, applicatin layer routing.

[10 pts] (2) Consider a system of four queues being serviced according to a WFQ (Weighted Fair Queuing) scheduling policy. The weights given to the four queues (A, B, C, D) are 3, 2, 4, and 1 respectively. They are being serviced by a processor at the rate of 10 Mbps.



The table below gives a list of different input traffic rates (in Mbps) at the four input queues. Fill in the resultant output rates for each of the four queues. [Each row 2 points]

Comments: Many students didn't get teh third row correctly. Here is how to do this: Only C is sending less than its share, so C gets what it sends – 2. The rest needs to be divided according to the weight of 3:2:1 (for A:B:D). So, A gets 3 out of 6, or half of what's left, which is 4. B gets 2 out of 6, or one third of what's left (8), which is $8/3$ or 2.67 approximately. Finally, D gets 1/6 of 8, which is about 1.33.

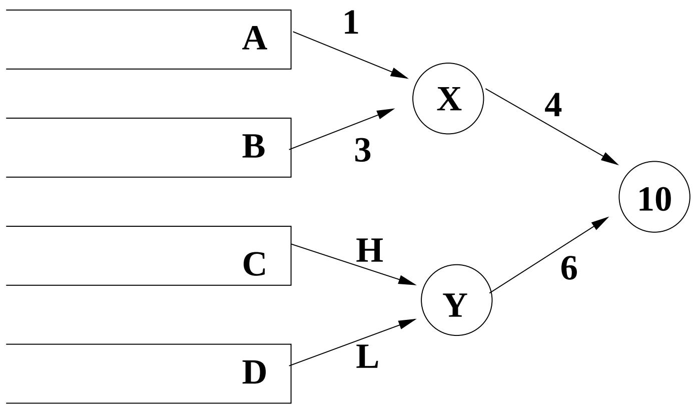
INPUT RATES				OUTPUT RATES			
A	B	C	D	A	B	C	D
1	1	1	1	1	1	1	1
10	10	10	10	3	2	4	1
6	6	2	2	4	2.67	2	1.33
8	0	0	8	7.5	0	0	2.5
1	5	3	5	1	4	3	2

[6 pts] (3) This time we still have the same four queues, but the processors are split in two levels and are based on priority queuing, WFQ, or Time Division Multiplexing (TDM).

Each of the two second level processors X and Y have a weight of 4 and 6 respectively. The policy implemented between these two processors is that of TDM, i.e., each of the processors gets a fixed share of the bandwidth.

Processor X implements a WFQ scheduling between its two queues A and B. A has a weight of 1 and B has a weight of 3.

Processor Y implements a strict priority between its two queues C and D. Queue C has strictly higher priority than queue D.



The table below gives a list of different input traffic rates (in Mbps) at the four input queues. Fill in the resultant output rates for each of the four queues. [Each row 2 points]

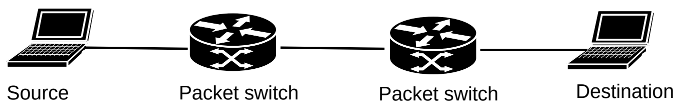
INPUT RATES				OUTPUT RATES			
A	B	C	D	A	B	C	D
6	6	2	2	1	3	2	2
2	6	2	6	1	3	2	4
6	2	6	2	2	2	6	0

Comments: It is important to realize that because of TDM, some of the bandwidths may be wasted if the output rate of X is less than 4 or if the output rate of Y is less than 6.

Problem 2: Packet switching – 20 points

(a) [2 points] Describe one advantage and one disadvantage circuit-switched networks have over a packet-switched networks.
Answer: advantage: can guarantee a certain amount of end-to-end bandwidth for the duration of the circuit. Most packet-switched networks today cannot make any end-to-end gurantees for bandwidth. Disadvantage: especially for short-lived flows, the set up overhead of the circuit can consume relatively high overhead. Also, the reserved bandwidth if not used is wasted.
(b) [3 points] Consider sending a packet from a sending host to a receiving host over a fixed route. List the delay components in the end-to-end delay computation. Which of these delays are constant and which are variable?
Answer: The delay components are processing delays, transmission delays, propagation delays, and queung delays. All of them are fixed for a fixed-size packet except queuing delay.

(c) In modern packet-switched networks, the source host segments long, application-layer messages (for example, an image or a music file) into smaller packets and sends the packets into the network. The receiver then reassembles the packets back into the original message. We refer to this process as *message segmentation*. The figure below illustrates a switched network. Consider a message that is $7.5 * 10^6$ bits long to be sent from the source to the destination. (Assume header size is negligible relative to the entire message size). Suppose each link is 1.5 Mbps. Focus on transmission delays only and assume all other delay components are negligible.



(1) [5 points] Consider sending the message from source to the destination *without* message segmentation. How long does it take to move the message from the source host to the first packet switch? Keep in mind that each switch uses store-and-forward packet switching. What is the total time to move the message from source to the destination host?
Answer: $7.5 * 10^6 / 1.5 * 10^6 = 5$ sec.
Total delay = 5 sec $\times$ 3 hops = 15 sec.

(2) [5 points] Now suppose that the message is segmented into 5000 packets, with each packet being 1500 bits long. How long does it take to move the first packet from source to the first switch?
When the first packet is being sent from the first switch to the second switch, the second packet is being sent from the source host to the first switch. At what time will the second packet be fully received at the first switch?

Answer: $1500/1.5 * 10^6 = 1$ msec.
Time at which second packet is received at the first switch is the time at which first packet is received at the second switch: 2×1 msec = 2 msec.

(3) [5 points] How long does it take to move the file from source host to destination host when message segmentation is used?
Answer: Time at which the first packet is reeived at the destination host is 3 msec. After this, every 1 msec one packet will be received; thus time at which the last or 5000th packet is received is 3msec + 49999 * 1msec = 5.002 sec.

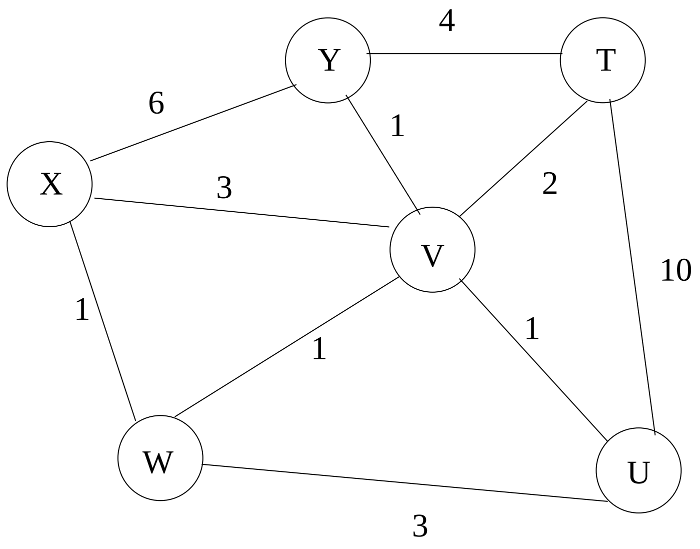
Compare this result with our answer in part (1) and give at least one advantage and disadvantage of message segmentation.
Answer: Using segmentation, delay is significantly less than the delay in (1) – without segmentation. There are at least two drawbacks with segmentation: packets ahve to put in sequence at the destination. Message segmentation rseults in many smaller packets. The overall number of bytes transmitted will be higher. The advantage is that because of store-and-forward, the overall delay in transferring the message is lower due to parallelism.

Problem 3: Internet Routing – 20 points

Part One - Intradomain Routing [10 points]
(a) [2 points] Link state protocols are used for intradomain routing. Give two reasons why it is not used for interdomain routing, or routing for the entire Internet.

Answer: (1) It will not scale. (2) privacy concerns, networks don't want to share topology information.

(b) [4 points] Given the following artificial network topology, please fill the routing table for node T, both the distance and the nexthop used for forwarding.



Destination	NextHop	Distance
X	V	4
Y	V	3
U	V	3
V	V	2
W	V	3

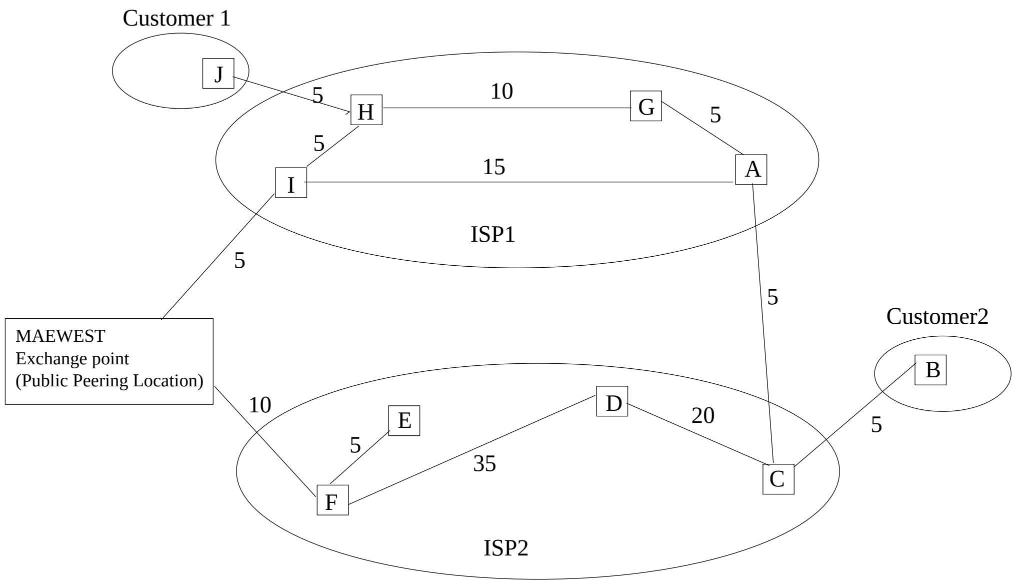
(c) [4 points] Assume node V needs to be brought down for maintenance (e.g., upgrade its software). One common way to deal with this is to increase the cost of the links associated with the node to ∞, then node V is brought down. Update the table below (for node T) when all the links of V have its cost changed. Is this graph still connected? (*Assume the network has reached convergence, i.e., a stable routing.*)

Destination	NextHop	Distance
X	Y	10
Y	Y	4
U	U	10
V	N/A	∞
W	Y	11

Yes, the graph is still connected ignoring node V.

Part Two - Interdomain Routing [10 points]

Consider the following topology. The cost metric of a link denotes the one-way propagation delay on the link in msec (assuming the delays are symmetric). The two ISPs ISP1 and ISP2 are peers. CIDR is used for addressing and BGP is used for inter-domain routing.



(a) [5 points] Assume that both ISPs always try to enforce hot-potato routing above all other routing policies. What is the one-way propagation delay between Customer1 and Customer2? *(Hint: routing may not be symmetric.)*

Answer: Hot potato routing means that traffic is handed off to the next hop network as soon as possible. From Customer1 to Customer2: J-H-I-F-D-C-B: 5+5+5+10+35+20+5 = 85 msec.
From Customer2 to Customer1: B-C-A-G-H-J: 5+5+5+10+5=30 msec.
Note, routing here is asymmetric.

(b) [5 points] Assume Customer2 decides to increase its reliability to the internet by purchasing a second connection from ISP1. Customer2 is now also a customer of ISP1. Customer2's new connection to ISP1 is to router A. The cost metric for A-B link is 5. Assume that Customer2 gives equal preference to both of its connections to the Internet. With this new link, what is the one way propagation delay between Customer1 and Customer?

Answer: Now that Customer2 is multihomed to ISP1 and ISP2, both Customer2 and Customer1 are customers of ISP1. For routing between customers of the same ISP, there is no need to traverse a different network (i.e. ISP2), because IGP learned routes are preferred over EBGP learned routes.
From Customer1 to Customer2: J-H-G-A-B = 25 msec
From Customer2 to Customer1: B-A-G-H-J = 25 msec
Note, routing here is symmetric.

Problem 4: The Web Server Project – 20 points

Part One - 489TP Client and Server [12 points]

The following two code snippets are from a client and server who are running the 489TP protocol. 489TP is used for the client to get its grade on a test.

```
//// Common structure shared by client and server
struct gradeStruct {
    short percentGrade;
    short classRank;
    char letterGrade[4];
};

//// 489TP Server, s is a SOCKET just accepted from the client
char recvbuf[0x100];
rval = recv(s, recvbuf, 0x100, 0);
if(rval == -1) exit(1);

struct gradeStruct yourGrade;
if(!strncmp(recvbuf,”HELLO GET GRADE\r\n\r\n”)) {
    yourGrade.percentGrade = 80;
    yourGrade.classRank = 10;
    strcpy(yourGrade.letterGrade,”B+”);
    rval = send(s, &yourGrade, sizeof(yourGrade), 0);
}
else {
    rval = send(s, “489TP BAD REQUEST”, 12, 0);
}
if(rval == -1) exit(1);

//// 489TP Client, s is a SOCKET just connected to the server
rval = send(s, ”HELLO GET GRADE\r\n\r\n”, 19, 0);
if(rval == -1) exit(1);

struct gradeStruct myGrade;
rval = recv(s, &myGrade, sizeof(myGrade), 0);
if(rval == -1) exit(1);

printf(“Your grade is %d percent -- %s. Your class rank is %d”,
myGrade.percentGrade, myGrade.letterGrade, myGrade.classRank);
```

(a) [4 points] Assuming that the client and server are supposed to exit in the case of a communication error, and that setup and teardown of the connection is performed outside of the code snippet provided, list what is wrong with the implementation of the protocol listed above and why.

Answer:
Both the client and the server do not send or receive in a loop. (1 POINT) This could cause messages to be cut off. (1 POINT) Also, the server does not use htons() on the integer values, and the client does not use ntohs().(1 POINT) This could cause the integer values to be different if the client and server are running on a big-endian and little-endian architecture or vice-versa. (1 POINT)
(b) [4 points] Under what circumstances will the client and server still work properly despite any problems with the above implementation?

Answer:
The client and server will still work properly under two conditions. If the messages are not fragmented and the send and receive are able to send and receive everything in one call, then there will be no problem with messages getting cut off. (2 POINTS) Secondly, if the two hosts are running on the same architecture, then the integer values will be the same for both the client and server. (2 POINTS)
(c) [4 points] For each problem identified in part (a), indicate the first line number on which it occurs, and provide a correct replacement. *(Hint: if there is a similar problem with sending and receiving, then you only have to provide correct code for the first occurrence of one, not both)*
Answer:

```
Not receiving in a loop -- first seen on line 10.
Proper replacement: (3 POINTS)
int bufsize = 0x100;
int recved = 0;
```

```
recvbuf[0] = '\0';
while(!strstr(recvbuf, ''\r\n\r\n'' && recved < bufsize){
rval = recv(s, recvbuf + recved, bufsize - recved, 0);
if(rval == -1) exit(1)
recved += rval;
recvbuf[recved] = '\0';
}
```

Not using htons – first seen on line 15. Proper replacement: (1 POINT) yourGrade.percentGrade = htons(80);

Part Two - Threads vs. Select - [8 points]

For this problem, assume each question has an *and why?* attached to the end of it. You will not receive credit for just writing “threaded” or “select”.

(a) [2 points] For a protocol where a large number of clients stay idle for very long periods of time, which type of server will have more CPU overhead, a threaded server or a select server?

Answer:
The select server will have more CPU overhead because of all the polling required for a large number of idle connections.
(b) [2 points] Using the same assumptions from part (a), if the amount of state kept for each connection is very small, which will have more memory overhead?

Answer:
The thread server will have more memory overhead because it will allocate space of all the thread stacks. A select server would only have to allocate memory for connection state.

(c) [4 points] Your boss tells you that you need to implement a server to do complex image processing for a large number of clients. Each session will require 20 megabytes of state, an average of ten minutes of processing time, and no state will be shared between sessions. The processing API calls each take about an average of one minute to complete, do not have a callback mechanism, and cannot be changed. Furthermore, you only have one week to implement the server, by yourself. Which server model would you use? Give at least two reasons based on the above constraints why this is a better choice.

Answer:
A threaded server would probably be better here for two reasons. The first is that the image processing operations take a very long time and are blocking. If implemented on a select server, nothing would get serviced during the processing operations, which would be very bad. (2 POINTS) Another good reason to use a threaded server is that you do not have very much time. Using a multi-threaded server is almost always easier and more intuitive, especially if there is no shared state between sessions (no synchronization required).

Problem 5: Transport Layer: TCP and UDP – 20 points

Part One: TCP vs. UDP [8 points]
(a) [3 points] List three services provided by TCP that are not provided by UDP.

Answer:
Reliability, In-order delivery, Congestion control and others (NOT port numbering, since UDP provides that as well) (1 POINT EACH UP TO 3 CORRECT RESPONSES)
(b) [5 points] Indicate whether you think TCP or UDP would be better-suited for each of the following applications **and** briefly explain why. State any assumptions that you are making for each application.

- Streaming video client/server

  Answer:
  UDP – reliability is not needed, but a guaranteed transfer rate is. UDP will give a higher-quality transmission if there are limited network resources.

- Multiplayer online first-person shooting game

  Answer:
  Probably UDP – reliability is needed here, but so is low latency. A case could be made for both, but many games use UDP in order to achieve the low latency.

- IRC (chat) client/server

  Answer:
  Definitely TCP – reliability, in-order delivery are needed, and bandwidth/latency guarantees are not.

- Internet telephony voice channel

  Answer:
  UDP – Same as answer to streaming video: reliability is not needed, but a guaranteed transfer rate is. UDP will give a higher-quality transmission if there are limited network resources.

- A protocol designed to synchronize the clocks of computers over a network, what protocol should be used for packets exchanged to identify time differences?

  Answer:
  Most likely UDP: designed particularly to resist the effects of variable latency. (NTP uses UDP.) TCP will incur additional overhead and becomes harder to synchronize the clock.

Part Two: TCP Slow Start [12 points]

There are two nodes on a network C and S. C is the client and S is the server. C wants to connect to S and send a message that is 35 kilobytes long using TCP. Assume that a single packet can hold up to 2 kilobytes of data and the headers are negligibly small. Processing time at both ends of the connection is negligible, but the latency in between node S and node C is 5 ms. The link transmission rate is 10 Megabits per second. Assume that control packets (SYN, ACK, etc) are very small and can be sent and received instantaneously. *Also assume that the connection starts off in the slow start stage and that there is no packet loss.*

(a) [1 points] At what time will C begin sending the message to the server S?

Answer:
2 * Trip Time (SYN has to be sent to server and SYN/ACK has to be received) = 2 * 5 ms = 10 ms
(b) [2 points] At what time will S send an ACK packet in response to the first data packet sent by C?

Answer:
3 * Trip time + 2 kilobytes / ((10/8) * 1024 kilobytes / second) = 3 * 5 ms + 1.56 ms =16.56ms
(c) [2 points] How many windows will be needed for the client to send the entire message?

Answer:
2 + 4 + 8 + 16 + 32> 35 = 5 windows. 4 windows would not be enough to transmit the entire message
(d) [3 points] After the end of which window will there no longer be any delay time? (Delay time is time where no packets are being sent from client to server)

Answer:
(# of packets-1) * transmission time per packet > RTT N * 1.56ms > 2 * 5 ms
N = 8 (The fourth window) will be the first one after which there is no delay
(e) [4 points] At what time will the server finish receiving the message from the client? *(Hint: it may help to draw a timing diagram)*

Answer:
P = transfer time of one packet = 1.56 ms
RTT + (start of first window)P + RTT + (start of second window)P + RTT + (start of third window)P + RTT + (start of fourth window)P*8 + (start of fourth window)P*3 + RTT/2 = RTT * 4.5 + P * 14 = 45 ms + 21.8 ms = 66.8 ms

Extra Credit: 5 points

(a) [2 points] TCP does not work well for wireless networks. Explain why and how would you improve it?
Answer: In TCP, a packet loss detected through timeout or 3 duplicate ACKs is assumed to be caused by network congestion. However, in wireless networks, packet loss can be due to collision, signal attenuation, signal interference. Thus, to improve wireless transport, the transport protocol may need more information from the link layer to decide when to adjust window size.

(b) [2 points] Assume a fraction of routers are compromised on the Internet. Propose a way to identify these compromised routers and describe how to ensure that the Internet can still deliver packets correctly?
Answer: Assuming these compromised routers can do the following things to the packets: alter the packet content, drop them, delay them. For each such operation, there are ways to detect them. For example, error coding can help detect packet modification. To ensure that packets are still correctly delivered, one can use application layer routing or overlay routing to bypass compromised routers.

(c) [1 point] Describe a recent news (not mentioned in class) that is relevant to the networking material we have learned in class so far.