NAME:

Login name:

Computer Science 461
Midterm Exam
March 14, 2012
10:00-10:50am

This test has seven (7) questions, each worth ten points. Put your name on ***every page***, and write out and sign the Honor Code pledge before turning in the test. You should spend roughly seven minutes per question, but ideally five minutes on each of the first five questions, leaving yourself 10-12 minutes for each of the last two questions.

``I pledge my honor that I have not violated the Honor Code during this examination.''

Question (points)	*Score*
1 (10 pts)	
2 (10 pts)	
3 (10 pts)	
4 (10 pts)	
5 (10 pts)	
6 (10 pts)	
7 (10 pts)	
Total:	

QUESTION 1: *Waiting to Go* *(10 points)*

A home-network router has an upload bandwidth of 1 million bits per second (i.e., 125,000 bytes per second) to the Internet, and a 100,000-byte first-in first-out buffer for packets awaiting transmission. Packets have a maximum transmission unit (MTU) size of 1000 bytes.

1(a) If the router's buffer is completely full of *TCP acknowledgment* packets, how many packets are queued in the buffer? Assume the TCP segment has no payload, and the packet headers do not contain any IPv4 or TCP options. (*3 points*)

Ignoring the link-layer header and any minimum size restriction on the link-layer frame, a TCP ACK packet is 40 bytes long, consisting of a 20-byte IP header and a 20-byte TCP header. A 100,000-byte buffer can hold 100000/40 or 2500 ACK packets.

1(b) Suppose the buffer is completely full. How long does it take the router to transmit all of the bytes in the buffer? (*3 points*)

Sending 100,000 bytes at 125,000 bytes/sec requires 100000/125000 or 0.8 seconds, or 800 msec.

1(c) Suppose the router supports two first-in first-out queues, one for interactive applications (like Voice over IP) and the other for all remaining traffic, with static priority for the queue handling interactive traffic. If a VoIP packet arrives when the queue for interactive applications is empty, what is the maximum time before the router starts transmitting the VoIP packet? (Assume that the router does *not* preempt any ongoing packet transmission.) (*4 points*)

At worst, the VoIP packets arrives just as the router starts transmitting a full-sized (i.e., 1000-byte) non-VoIP packet, forcing the VoIP packet to wait for 1000/125000 or 0.008 seconds, or 8 msec.

QUESTION 2: *A Good, Firm Handshake* *(10 points)*

Host A initiates a TCP connection with host B by sending a SYN packet, and B responds with a SYN-ACK packet. Upon receiving the SYN-ACK packet, A responds with an ACK packet. For each question, please check one box

2a) What is the earliest time that A can start sending data to B? (*3 points*)

_____ Immediately after sending the SYN packet to B

__X__ Immediately after receiving the SYN-ACK packet from B

_____ Immediately after receiving data from B

Once the SYN-ACK arrives, A knows that B has A's initial sequence number and is ready to accept data packets from A.

2b) What is the earliest time that B can start sending data to A? (*3 points*)

_____ Immediately after receiving the SYN packet from A

_____ Immediately after sending the SYN-ACK packet to A

__X__ Immediately after receiving the ACK packet from A

_____ Immediately after receiving data from A

Once the ACK arrives, B knows that A has B's initial sequence number and is ready to accept data packets from B.

2c) Suppose the ACK packet from A gets lost. What ultimately triggers A to retransmit the lost ACK packet? Assume no other packets get lost and no data packets are sent. (*4 points*)

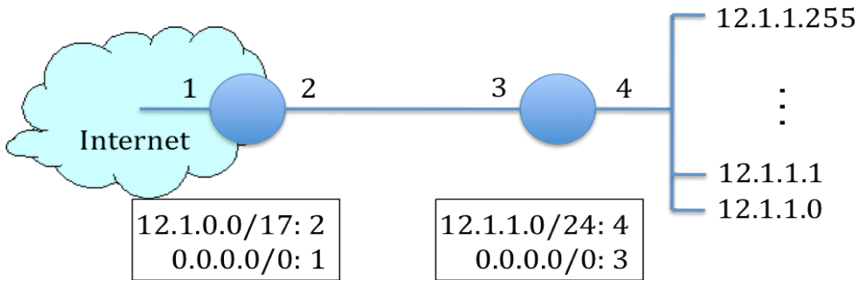
_____ A retransmission timeout at A

__X__ A retransmission timeout at B, triggering a retransmission of the SYN-ACK packet

ACK packets are not retransmitted, so A does not experience a retransmission timeout. If B does not receive an ACK for the SYN-ACK, then B will retransmit the (presumed lost) SYN-ACK after the retransmission timeout expires.

QUESTION 3: *Defaulting* *(10 points)*

A small university campus is assigned a large address block 12.1.0.0/17, but is only using a portion of these addresses (in 12.1.1.0/24) to number its computers. The campus uses a single Internet Service Provider (ISP) to reach the rest of the Internet. This picture shows the forwarding tables on the ISP's router (on the left) and the campus edge router (on the right):



For example, the ISP forwards all packets with destination addresses in 12.1.0.0/17 out link #2 toward the campus edge router. Both routers include a default forwarding entry 0.0.0.0/0 that can match any destination IP address.

3a) How many IP addresses does the campus "own" in its 12.1.0.0/17 block? You can represent your answer as a power of two. (*2 points*)

The 17-bit mask leaves 32-17, or 15 bits to identify the addresses within the block. As such, the block contains 2^{15} or 32,768 addresses.

3b) What are the smallest and largest IP addresses that the campus "owns", whether or not the campus is currently using the address? (*2 points*)

12.1.0.0-12.1.127.255

3c) Suppose the ISP router receives a packet with destination IP address 12.1.1.1? What path does this packet follow? (*2 points*)

The packet flows over the path 1→ 2 → 3 → 4.

3d) Suppose the ISP router receives a packet with destination IP address 12.1.20.1? What path does this packet follow? (*2 points*)

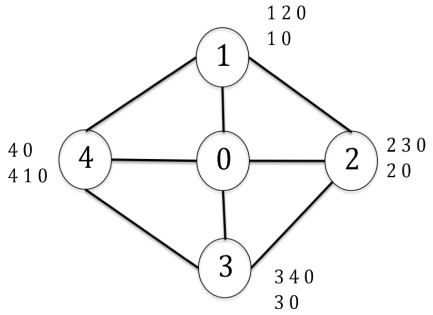
The packet flows over the path 1→ 2 → 3 → 2 → 3 …

3e) What ultimately happens to a packet with destination IP address 12.1.20.1? Where does it go? (*2 points*)

The looping packet is discarded once its IP TTL (Time-To-Live) expires.

QUESTION 4: *Stability* *(10 points)*

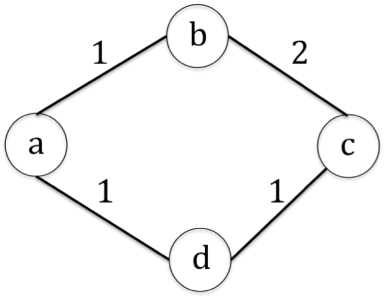
4a) The Stable Paths Problem (SPP) is an abstract model of BGP routing, where each node has a ranked list of "permitted paths" (where the first path is preferred over the second path, and so on). In a solution of an SPP instance, each node selects the highest-ranked path consistent with its neighbors' choices. Consider the SPP below, where nodes 1, 2, 3, and 4 each want to select a path to destination node 0. Does this SPP instance have a unique, stable solution? If yes, please indicate the path that each node selects in the stable solution. If not, give an example of an oscillation. Use the space to the right of the figure for your answer. (*4 points*)



Stable solution:

1: 1 2 0
2: 2 0
3: 3 4 0
4: 4 0

4b) Consider the network below that runs a link-state routing protocol that computes shortest paths as a sum of link weights. The number on each link is the weight of the link in each direction (e.g., links b-c and c-b both have weight 2). Suppose nodes a, b, and d send packets to destination node c. If link d-c (and c-d) *fails*, which of nodes a, b, and d could conceivably see their packets stuck in a temporary forwarding loop? Which ones would not? (*3 points*)



a and d can loop, but not b

4c) Using the same figure as in question 4b), suppose the network operator is taking link d-c (and c-d) down for *planned maintenance*. The network operator wants to temporarily set the weight of the link to a higher value, to coax nodes to move away from paths that use the link without creating any blackholes or loops. What is the (minimum) sequence of weight settings for the link that ensures no packets destined to node c experience a blackhole or loop? (*3 points*)

1. Change the cost of the d-c link to 3, causing node a to shift to the upper path
2. Change the cost of the d-c link to 5 or higher, causing node d to shift to the upper path
3. Shutdown the d-c link, since nobody is using it

QUESTION 5: *Sockets* *(10 points)*

This question concerns the relationship between socket calls at the application layer and the behavior of the transport layer. Consider a reliable stream (i.e., TCP) socket between application processes A and B running on two different hosts.

5a) Suppose process A calls send() to send 10,000 bytes of data, and the operating system begins sending the data as a sequence of packets destined to B. If process A calls close() on the socket *before* the operating system finishes transmitting the data, will the remaining data be sent to B? Why or why not? (*3 points*)

Yes, the Operating System continues transmitted the remaining data to B, to ensure the ordered, reliable delivery of the byte stream.

5b) Suppose process A calls close() *after* the operating system has transmitted all of the outstanding data. What kind of packet does the operating system send in response to A's call to close()? (*2 points*)

TCP FIN packet

5c) Explain why B may need to perform *multiple* calls to recv() to receive all 10,000 bytes of data from A. (*3 points*)

The data may not reach B in its entirety at the same time, since it is split into multiple packets. These packets may experience different delays due to congestion, retransmission and receive-window constraints.

5d) Suppose B's operating system has learned that A has closed its end of the connection. Once process B has read all of the data from A, what does B's *next* call to recv() return? (*2 points*)

A 0, indicating the end of the stream.

QUESTION 6: *Location, Location, Location* *(10 points)*

An IP address relates to a host's location in the Internet topology, but is not easily translated to a precise geographic location. This question concerns mapping an IP address to a location.

6a) Give two examples of how knowing a client's rough geographic location can improve an Internet service. (*4 points*)

- **Customizing the content to the user's location (e.g., showing weather, advertisements, traffic reports, etc. related to the user's presumed location)**
- **Blacking out access to sites disallowed in certain geographic regions (e.g., due to local laws, commercial restrictions on showing sporting events in the local area, etc.)**
- **Directing clients to the closest Web server offering a given service**

6b) Propose a technique to infer the geographic location associated with an IP address, without cooperation from the end user. Explain how your technique works, and what assumptions it makes. (*4 points*)

- **Performing a traceroute to the client address, and looking for geographically-meaningful names late in the traceroute path (e.g., "yale-univer.car2.stamford1.level3.net")**
- **Performing pings to measure latency to reach the client from multiple vantage points with known geographic locations, and triangulating to infer the client's location**
- **Altering the design of the routing system to include geographic information (though this requires substantial control over the entire Internet, making this approach unrealistic in practice)**
- **Performing a reverse DNS lookup on the client IP address (though most client's don't have DNS names, and these names often don't reveal location information), or performing a "whois" query to find the address of the organization that owns the IP address**

6c) Give an example where your technique will *not* produce accurate results. (*2 points*)

- **If the client lies at the end of a high-latency link (e.g., a satellite)**
- **If routers near the client do not participate in traceroute (e.g., do not send ICMP responses)**
- **If the client's machine does not respond to ping requests**
- **If the organization in whois covers a very large geographic region (e.g., the address of corporate headquarters may not reveal the client's location all that accurately)**

QUESTION 7: *In Name Only* *(10 points)*

Some WiFi hotspots offer wireless access for a fee. When a user connects to the access point, a DHCP server assigns them an IP address, network mask, gateway router, and local DNS server. When the user tries to access a Web site, the hotspot directs the HTTP request to a "pay wall" where the user can enter credit card information, and only then can the user access the requested Web site. These hotpots typically allow *DNS queries* through the pay wall, though, since the user's Web browser needs to use DNS to map domain names to IP addresses.

7a) Suppose, before the user pays, the hotspot only allows the user to send and receive UDP packets with source or destination port 53 (the port number used for DNS). Assume the hotspot allows *any* such traffic to travel to and from the user, independent of the destination IP address. Briefly describe the design of a system that would allow the user to access arbitrary sites on the Internet via the hotspot *without paying*. The user may need to deploy parts of the system ahead of time. Include a picture. (*5 points*)

The user can run a proxy that listens on UDP port 53. The client can encapsulate an IP packet (say, containing a TCP segment) inside UDP packets sent to the proxy, and the proxy can send the traffic to arbitrary Internet services at the client's request. The responses can be encapsulated and sent back to the client using UDP packets with port 53. To ensure reliability, the client and the proxy can acknowledge received UDP packets and retransmit unacknowledged packets.

7b) Now suppose that the hotspot only allows DNS queries (i.e., correctly formed DNS requests sent to UDP port 53) destined to the IP address of its own local DNS server, and DNS responses from that server. Design a new solution that allows the user to access arbitrary Internet sites without paying. Include a picture. Note that you should *not* use this technique in real life to evade paying for Internet service! (*5 points*)

The user can run a proxy that acts as an authoritative DNS server for a domain the user has registered (say, jrex.com). To communicate with the proxy, the client sends a DNS request to the hotspot's local DNS server, with the contents of the message embedded in the domain name (e.g., here-is-my-message.jrex.com). This will trigger the local DNS server to direct the query to the jrex.com authoritative server, which can encode a message in the DNS response (e.g., using the four bytes of the IP address to encode four characters, or using DNS TXT records to encode more information. As in the previous question, the client and the proxy need to acknowledge and retransmit the UDP packets to ensure reliable delivery. The domain names could even embed sequence and acknowledgment information (e.g., here-is-my-message-X.jrex.com, where X is a number). This class of solutions is known as "DNS tunneling."

Several students offered a different answer that was also valid. The idea is to hijack the hotspot's local DNS server by sending a forged ARP response for the DNS server's IP address, learned by the client in DHCP bootstrapping. Now the client can send UDP port 53 packets, and follow the approach in part 7a of the question. Note that this approach may be easier for the hotspot administrators to detect, since the local DNS server would suddenly be carrying no traffic. (That said, even the solution above is relatively easy to detect if the hotspot administrators are monitoring DNS query/response loads on a per-client basis.)

Other students offered another answer that was also valid. The client could snoop on DHCP, ARP, or flooded packets to learn the IP addresses of other users who have (presumably) paid for service. The client can then send a forged ARP response for another client's IP address, and be able to receive packets sent to that address. The client can start sending packets from that address, too, allowing free communication. (This might be detected by the other user, and in fact that user may sometimes send traffic, causing the switches to shift traffic back to the old client. As such, the crafty client would need to repeatedly hijack the original client.)