Final — Dec. 16, 2006

EE122: Introduction to Communication Networks

Fall 2006

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 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. Seven of these are worth 20 points each, and one (problem #7) is worth 40 points. ***The entire test is worth 160 points****, so you can miss up to 20 points across various problems and still get a perfect score.*

The test is 180 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	/ 20	/ 20	/ 20	/ 20	/ 20	/ **40**	/ 20	

1. **Router messages and feedback.** (Total: 20 points)

   (a) When routers generate ICMP messages, to where do they send them? Along with the ICMP header at the beginning, what additional contextual information do routers include in the messages? (5 pts)

   (b) Are ICMP messages delivered reliably? If so, briefly explain the mechanism. If not, give a reason why not. (5 pts)

   (c) Name a circumstance under which an end-host (not a router) will *send* an ICMP message. (5 pts)

   (d) Briefly describe how the `traceroute` tool works (i.e., what does it do in order to identify the routers that make up an Internet path). (5 pts)

2. **Attacks.** (Total: 20 points)

   Suppose we could deploy a mechanism that would ensure IP source addresses correspond to the actual sender of a packet (i.e., it's impossible to "spoof" source addresses). For each of the following threats, explain whether (and briefly why) the mechanism would: *(i)* completely eliminate the threat, *(ii)* eliminate some instances of the threat, but not all of them, or *(iii)* have no impact on the threat. (Each is worth 4 pts.)

   (a) Buffer overflow attacks

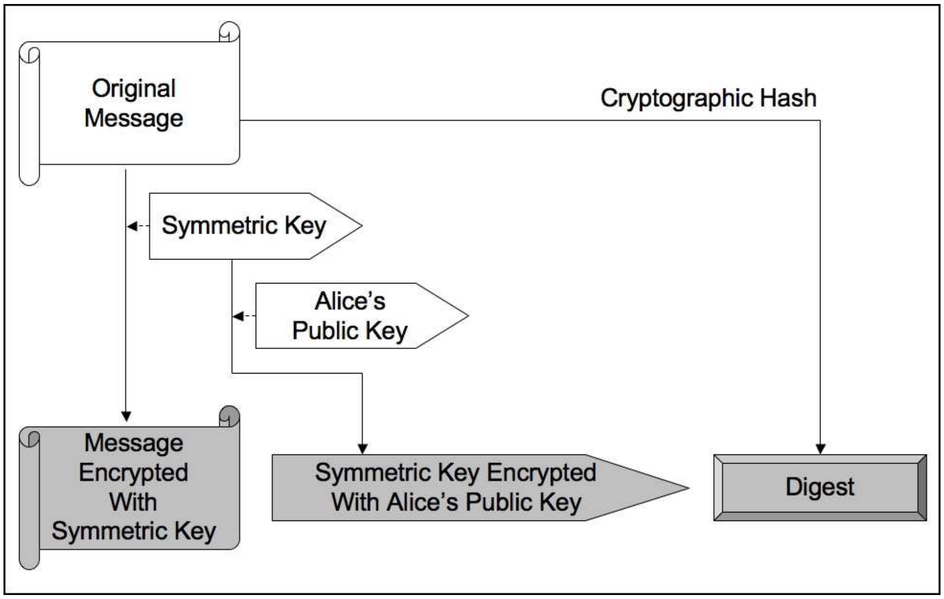
   (b) TCP SYN flooding

   (c) TCP "ack splitting" to open up the congestion window quickly

   (d) Reflector DDOS attacks

   (e) DNS cache poisoning

3. **Securing communication with cryptography.** (Total: 20 points)


Original Message
Cryptographic Hash
Symmetric Key
Alice's Public Key
Message Encrypted With Symmetric Key
Symmetric Key Encrypted With Alice's Public Key
Digest

Alice wishes to send a message securely to Bob. She wants to ensure that Eve cannot read her message or generate a fake message that Eve can trick Bob into falsely believing came from Alice. In addition, Alice's message is large, so she needs to encrypt it using a fast algorithm.

Assume that Alice's and Bob's public keys are well known, and that Alice uses the operations shown in the above figure, after which she sends to Bob the three grey items at the bottom of the figure: her original message encrypted with a symmetric key, a ciphertext version of that key encrypted with Alice's public key, and a hash generated using a function such as SHA-1 as a digest of the original message. Eve can see all three of these.

Alice's intention is that Bob will recover the symmetric key and use it to decrypt the message. He can then independently compute the SHA-1 hash of the message to verify that it matches the digest in order to be assured that the message has not been modified, and so that he can later demonstrate to others that Alice definitely sent the message (non-repudiation).

(a) Identify two different errors Alice has made in her use of cryptographic methods, and describe how to correct each of these. (12 pts) *(space for answer to Problem 3(a), con't)*

(b) Assuming that these errors are corrected, but that Eve figures out how to "break" SHA-1 such that she can generate hash collisions, explain an attack that Eve can then conduct. (8 pts)

4. **QoS.** (Total: 20 points)

(a) Suppose the capacity C of a link is 18. Assume that 4 sources—S1, S2, S3, and S4—are trying to send over the link at rates of r1=2, r2=4, r3=5, and r4=8, respectively. What is the max-min fairness allocation? (8 pts)

(b) For each of the following, annotate it with "IS" if it applies to Integrated Services (IntServ), "DS" if it applies to Differentiated Services (DiffServ), and "BE" if it applies to Best Effort. (A given statement can apply to more than just one type of service.)

i. The service is provided end-to-end (3 pts):

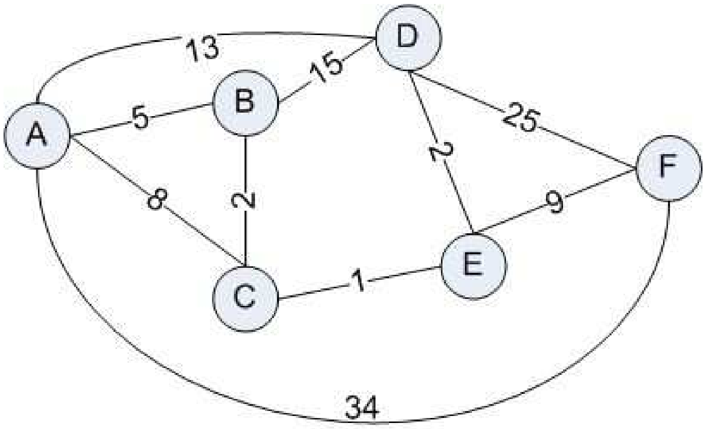
ii. Among the three, requires the most state in routers (3 pts):

iii. Is widely available in the Internet today (3 pts):

iv. Provides isolation and guarantees among aggregated flows but not individual connections (3 pts):

5. **Routing.** (Total: 20 points)

(a) Consider the following network with nodes A through F:



The number on top of each link represents the cost of the link. (The weight of the link between E and F is 9.) Node A runs Djikstra's algorithm to compute the shortest paths to all other nodes in the network. The following table captures the algorithm state at each step. Fill in the values for each step of the algorithm. (10 pts)

Step	S	D(B),p(B)	D(C),p(C)	D(D),p(D)	D(E),p(E)	D(F),p(F)
0 (Initialization)						
1						
2						
3						
4						
5						

The notation used in the above table is identical to that used in the lecture:

S : Set of nodes whose least cost path is definitively known

D(v) : Current value of cost of path from source to v

p(v) : Predecessor node along path from source to v, that is next to v

(b) In the above example, all the link costs are positive. Explain why Djikstra's algorithm does not work if some of the link costs are negative. (5 pts)

(c) Name two problems that would arise if all routing in the Internet was done using Link-State and Djikstra's algorithm. Does Distance-Vector routing suffer from these? How about Path-Vector? (5 pts)

6. **TCP mechanisms.** (Total: 20 points) You decide to modify the TCP stack on your desktop so you experience better performance (higher throughput when either sending, receiving, or both). Note that you only get to change your desktop's TCP—you can't change that of the other endpoint with which you'll be communicating.

   (a) Suppose your stack originally supports both timeout-driven retransmission and fast retransmission. Among the following, **circle** which one would gain you the greatest benefit to your performance:

      i. disable timeout retransmissions (instead only retransmit using fast retransmission)

      ii. disable exponential backoff of timeouts

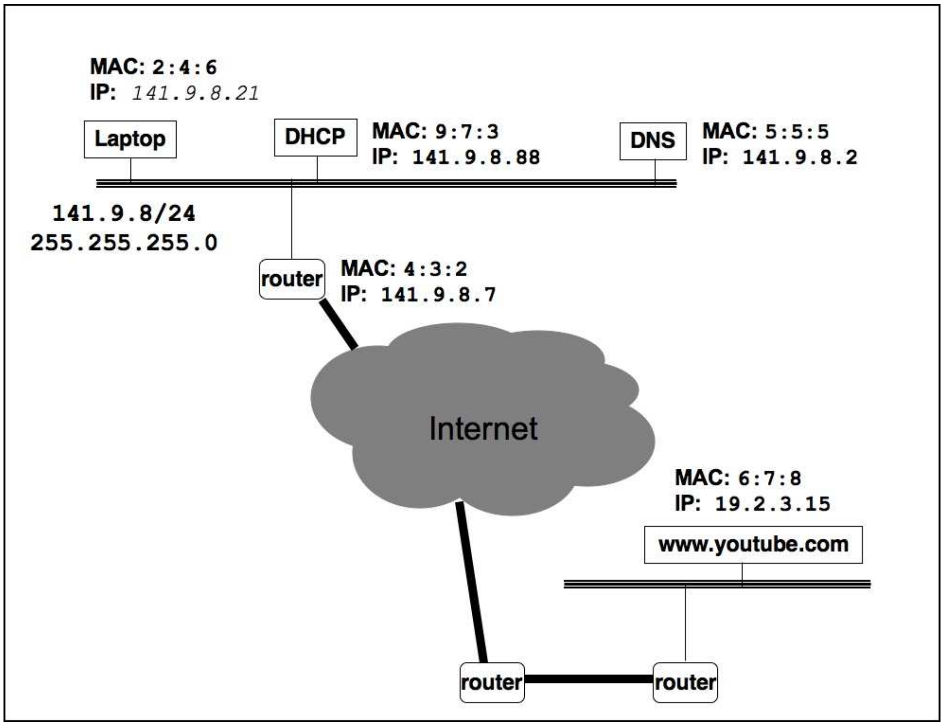
      iii. disable fast retransmission (instead only retransmit using timeouts)

      iv. disable RTT estimation/adaptation (when computing the retransmission timeout, just base it on the initial value RTT was set to)

   and **explain** why it would offer an improvement (8 pts):

   (b) What would happen if everyone's TCP stack in the Internet did this? (6 pts)

   (c) If you could pick something *else* to modify about your TCP stack so that you experience better performance, what would it be, and why? (6 pts)



7. **Putting it all together.** (Total: **40 points**)

   In the above figure (which is repeated on a later page for convenience), you connect your laptop via Ethernet to a local area network. Hosts on the LAN have IP addresses out of the 141.9.8/24 block, with a corresponding netmask of 255.255.255.0. For each host, the diagram shows its MAC address and its IP address. (We use shortened MAC addresses of just 3 digits to make them easier to write down.) Your laptop's MAC address is `2:4:6`. It initially does *not* have an IP address, though once it acquires one, it will be 141.9.8.21.

   You type http://www.youtube.com into your browser's URL line and hit return. Back comes the YouTube home page.

   Describe *all* of the messages (IP packets, and link-layer Ethernet frames for messages that are not transmitted using IP) sent and received by your laptop. Number the packets in the order they are sent or received, and for each give:

   - Source MAC and IP addresses, if any.
   - Destination MAC and IP address, if any.
   - Transport protocol and flag bits (e.g., TCP SYN), if any.
   - Description of the message carried in the packet.
   - For data packets, the sender's CWND (in terms of MSS-sized packets), as applicable.

For example, here is how we might describe the tail end of an SMTP session during which your laptop sends mail to a server running on the same machine as the DNS server in the diagram (your laptop has already acquired an IP address and both hosts know the MAC address of the other):

```
1. laptop -> DNS
     src MAC 2:4:6, IP 141.9.8.21 ; dst MAC 5:5:5, IP 141.9.8.2
     TCP data, "QUIT" + ack of prev. msg, CWND=6
2. DNS -> laptop
     src 5:5:5, 141.9.8.2 ; dst 2:4:6, 141.9.8.21
     TCP data, "221 closing" + ack of prev. msg, CWND=5
3. laptop -> DNS (same addresses)
     TCP FIN + ack of prev. message, beginning of close handshake
4. DNS -> laptop (same)
     TCP FIN + ack of FIN, both sides closed
5. laptop -> DNS (same)
     TCP ACK of FIN, termination handshake complete
```

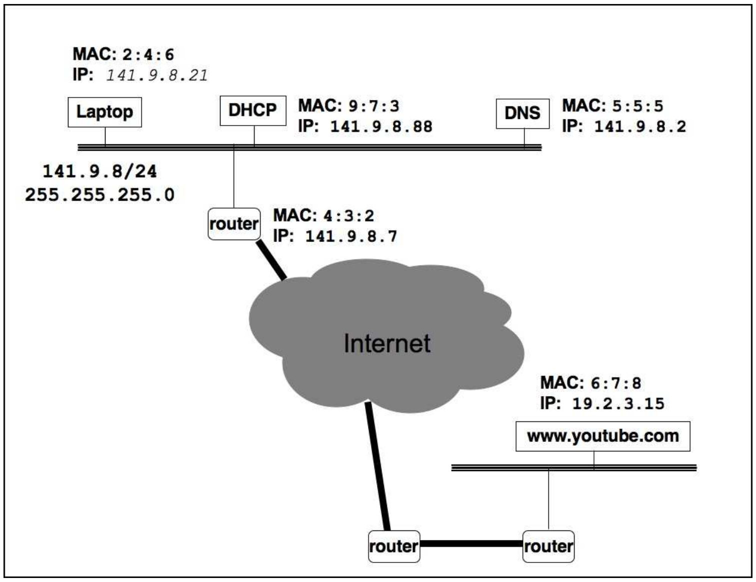
In addition to needing to acquire an IP address, assume:

- No packets are lost.
- Your laptop's DNS cache is empty.
- Your laptop's ARP cache is empty.
- For any other host your laptop communicates with, its caches are already populated with any necessary information. (For example, any request you make to the DNS server can be answered without the server making any further requests.)
- The HTTP request fits in a single packet, but the reply requires four packets. You can describe the request as just "HTTP GET" (no need to describe details or headers) and the replies as "HTTP REPLY 1" . . . "HTTP REPLY 4".
- All TCPs use delayed acknowledgments (ack-every-other) and standard congestion control (including the MSS*MSS/CWND version of Congestion Avoidance, as applicable).
- DNS requests are made using UDP.

The scoring for this problem is in terms of the following elements:

- Correct TCP connection establishment (4 pts).
- Correct TCP connection termination (2 pts).
- Correct name resolution (4 pts).
- Correct MAC addresses used (6 pts).
- Correct discovery of MAC addresses (4 pts).
- Correct IP addresses used when applicable (4 pts).
- Correct TCP data transfer, including acking and congestion control (6 pts).
- Correct host configuration messages (6 pts).
- Correct sequence of operations among all of these (4 pts).

(Problem 7, con't)



(Problem 7, con’t)

(Problem 7, con’t)

8. (20 pts) What different or alternative packets are generated in the previous problem in the following situations:[1]

   (a) The browser crashes just before the last data packet sent by www.youtube.com arrives. (5 pts)

   (b) The URL being fetched is instead https://www.youtube.com. No need to give TCP-level details, just sketch the additional higher-layer interactions. (5 pts)

(con't)

[1] These hypotheticals are *not* cumulative. Each is a single variation on the original problem.

(c) The router at 141.9.8.7 is also a NAT box, though the subnet continues to use the public address block 141.9.8/24. (5 pts)

(d) You connect your laptop to the LAN using 802.11 (any version) rather than Ethernet. You need only state which different or additional link-layer messages will be sent, and in what order, for the first packet transmitted by the laptop. (5 pts)