CS144
Intro to Computer Networks
Midterm Exam – Monday, October 28, 2013
CLOSED BOOK, CLOSED LAPTOP, 2 NOTE PAGES

Your Name: Catherine Lu

SUNet ID: cglu @stanford.edu

Check if you would like exam routed back via SCPD: ☐

In accordance with both the letter and the spirit of the Stanford Honor Code, I neither received nor provided any assistance on this exam.

Signature: Catherine Lu

- The exam has 14 questions totaling 100 points.
- You have 90 minutes to complete them.
- Some questions may be much harder than others.
- All questions require you to justify your answer to receive full credit, even multiple choice questions for which you circle the correct answer(s).
- Keep your answers concise. We will deduct points for a correct answer that also includes incorrect or irrelevant information.

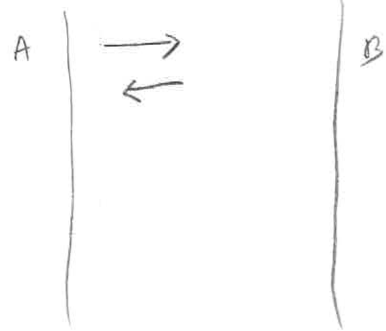
AB	**1**	5	/10
	2	5	/5
	3	0	/5
	4	0	/5
AL	**5**	5	/5
	6	5	/5
	7	5	/5
JDT	**8**	11	/15
	9	3	/5
	10	5	/5
	11	10	/10
	12	5	/5
	13	9	/10
	14	6	/10
	Total	74	/100

I Stop-n-Wait-n-Answer

1. **[10 points]:**

An application needs to send 100KB of data using a stop-and-wait reliable protocol. The protocol splits the data into segments that have a 1KB application data payload. Each segment fits in a single IP packet. The RTT is 50ms, there is no packetization delay, and no queueing delay. The protocol uses a fixed retransmission timeout of 200ms and has no retransmission limit.

(i) How long will the transmission take, in seconds, if the network does not drop, duplicate or corrupt any packets? You may assume the connection is established when you start your measurement, so there is no additional latency from connection setup and consists only of data transmissions. Your answer must be accurate to two decimal points (10ms).



25 ms one way

5.00 ✓ seconds

50 ms per 1KB

5000 ms per 100 KB

(ii) Let us now suppose that the network drops each segment with a probability of 10%, independently from segment to segment. The network drops both data and acknowledgements. What is the *expected* duration of the transmission? Show your calculations. Your answer must be accurate to two decimal points (10ms).

5

5.56 seconds

Acks can be dropped too

50 ms per 1KB if goes well (90%) → 45 ms

100 ms per 1KB $(.1)(.9)$ → 9 ms

150 ms per 1KB $(.1)^2(.9)$ → 1.35 ms

200 ms per 1KB $(.1)^3(.9)$ → 0.18 ms

250 $(.1)^4(.9)$ → 0.0225 ms

300 $(.1)^5(.9)$ → .0027 ms

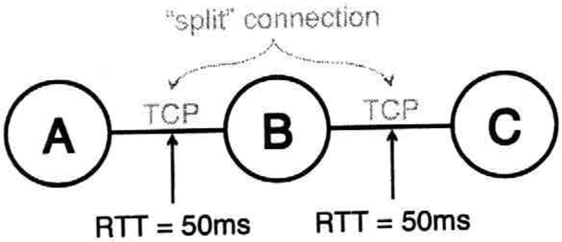
350 $(.1)^6(.9)$ → .000315

Wrong computation

55.5555 ms ~~AAA~~ $\times \frac{100}{1000}$ = 5.555... s → 5.56 ≈ 55.5555 ms on average

II Without TCP The Internet Ain't Nuthin'

There are devices and services in the Internet, such as proxy servers, that "split" TCP connections. Suppose a host A wants to open a connection to a host C. A device somewhere along the path, B, can terminate A's connection at itself, and open a connection to C. So in this case there are now two TCP connections, A to B and B to C. A thinks it's sending data to C, but B is processing the TCP segments itself and sending acknowledgments back to A, spoofed from B's IP address. Simultaneously, B opens a TCP connection to C, pretending to be A.



Suppose you have the network above, where the RTT from A to B is 50ms, the RTT from B to C is 50ms, and there is no packetization, queueing, or processing delay, such that the RTT from A to C is 100ms. The maximum segment size is 1400 bytes. A is sending an infinite stream of bytes, such that every segment is the maximum segment size. Recall that a TCP flow's throughput can be approximated as

$$\text{MSS} \cdot \sqrt{\frac{3}{2}} \cdot \frac{1}{\text{RTT}\sqrt{p}}$$

where p is the packet drop rate.

Please write out answers numerically and do not leave radicals or variables in your solutions. You may leave fractions. If you do not have a calculator, you may approximate with the following values:

$$MSS \cdot \sqrt{\tfrac{3}{2}} = 13,717 \text{ bits}$$

$$\sqrt{0.1} = 0.32$$

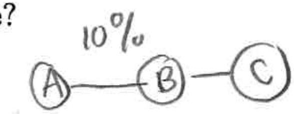
$$\sqrt{0.19} = 0.44$$

$$\sqrt{0.2} = 0.45$$

$$\sqrt{0.21} = 0.46$$

2. [5 points]:

Suppose that B does not split the TCP connection, such that packets flow directly from A to C, through B. The route between A and B drops 10% of data segments and does not drop acknowledgments, while the route between B and C does not drop any packet. What will the TCP throughput from A to C be?



434 ______ kbps

Overall, 10% packet drop as well.

$$MSS \cdot \sqrt{\frac{3}{2}} \cdot \frac{1}{RTT\sqrt{p}}$$

$$= MSS \cdot \sqrt{\frac{3}{2}} \cdot \frac{1}{.1s\sqrt{.1}}$$

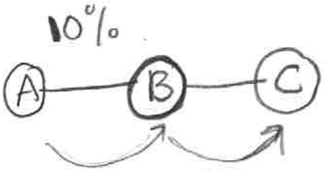
$$= (13717) \cdot \frac{1}{(.1)\sqrt{.1}}$$

$$= 433769 \text{ bps.}$$

$$= 433.77 \approx 434$$

3. [5 points]:

Suppose that B does split the connection, such that packets flow from A to B, terminate at B, then are forwarded in separate flow from B to C. The route between A and B drops 10% of data segements and drops no acknowledgments, while the route between B and C does not drop any packet. What will the throughput from A to C be?

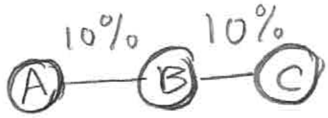


434 ______ kbps

should be the same as in #2.

4. [5 points]:

Suppose that B does split the connection, such that packets flow from A to B, terminate at B, then are forwarded in separate flow from B to C. The route between A and B drops 10% of packets, and the route between B and C also drops 10% of packets. What will the throughput from A to C be?



315 ✗ ______ kbps

$p = (.1)(.9) + (.9)(.1) + (.1)^2$

$= 0.19$

$MSS \cdot \sqrt{\frac{3}{2}} \cdot \frac{1}{RTT\sqrt{p}} =$

$= (13717) \cdot \frac{1}{.1\sqrt{.19}}$

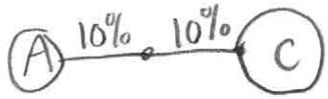
$= 314689.56$ bps.

$\rightarrow 314.68956 \approx 315$ kbps

5. [5 points]:

Finally, suppose that B does not split the connection, such that packets flow from A to C ~~B,~~ passing through but not terminating at B. The route between A and B drops 10% of data segments, and the route between B and C also drops 10% of data segments. What will the throughput from A to C be?

315 ✓ ______ kbps



same as in #4.

III FTP Is For Old People

does client send packet to FTP server?

~~Assuming no~~ yes, but separate connection

The File Transfer Protocol (FTP) is an older application protocol for transferring files. Like HTTP, it uses ASCII commands. Unlike HTTP, it uses a separate control and data channel. The protocol specification greatly predates STUN and other NAT probing/traversal approaches.

In normal operation, when a client requests a file (e.g., `RETRIEVE .cshrc`), the *FTP server* opens a TCP connection to the client to transfer the data. The client can specify the IP address and port to open a connection to with the `PORT` command. A client can alternatively tell the server to listen on a connection with the `PASSIVE` command (the server chooses the IP/port), such that the client can be the active opener.

Your client is behind a port-restricted NAT with no static mappings. The FTP server is outside the NAT and is not behind a NAT.

6. [**5 points**]:

Can your client use the `PORT` command to set up a successful file transfer?

Circle the best answer.

A Yes

B No

Briefly explain why:

The port-restricted NAT will not allow the FTP server to traverse its mappings since the client did not send out a packet to the FTP server port that is trying to open this second connection for the file transfer. Previously, the client sent the RETRIEVE command to a different port on the FTP server.

7. [5 points]:

Can your client use the `PASSIVE` command to set up a successful file transfer?

Circle the best answer.

A Yes

B No

Briefly explain why:

The client can actively open the TCP connection to the FTP server, since the server is not behind a NAT. The server will be listening for packets, and when the client sends the SYN, the server can follow-up, etc to establish the connection and then successfully transfer the file. The FTP server is, in this case, able to traverse the mappings in the NAT because the client's IP and port previously sent the SYN packet to the FTP server's IP and port.

at the specific IP/port which the client knows about and is connecting to

IV Put It All Together, Now

8. [15 points]:

You type the following URL into your web browser:

`http://gradadmissions.stanford.edu/inquiry/onlineinq.htm`

Assuming that

- your DNS resolver is 171.64.7.77,
- neither your host nor your DNS resolver have any cached DNS entries,
- DNS never needs to fail over to TCP, and
- the HTML response returns 200 OK with a web page,
- the HTML request and response each fit in a single segment, and
- the web page requires loading no additional resources,

write down the series of packet exchanges that will occur for your host to receive the web page. Include packets sent by your DNS server as well as control packets for TCP connection setup and teardown. You need not include any ARP packets, and you do not need to write down message formats. Simple descriptions such as "X sends a UDP segment to the HTML server on the HTTP port" are sufficient. In the case of the HTTP request, clearly state the path of the file requested in the GET.

My host sends a UDP segment to my DNS resolver
(DNS recursive lookup! "What is the IP address for gradadmissions.stanford.edu?"

recursive Lookup (-3)

My DNS resolver sends a UDP segment to my host
(answer to my DNS query)

My host sends a TCP segment to the HTML server on HTTP Port 80
(SYN)

HTML server sends TCP segment to my host
(SYN-ACK)

My host sends TCP segment to HTML server on HTTP Port 80
(ACK, HTML request GET for /inquiry/onlineinq.htm)

HTML server sends a TCP segment to my host
(web page and 200 OK)

HTML server sends a TCP segment to my host
(close/fin)

My host sends a TCP segment to HTML server
(fin/ack)

Last Ack? (-1)

V A Rose By Any Other Name..

The command "`dig stanford.edu A @a.edu-servers.net`" asks the machine `a.edu-servers.net` (name server for the `.edu` zone) for the IPv4 address (DNS type `A` record) of domain name `stanford.edu`. The command's output might look something like this:

```
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 31366
;; flags: qr rd; QUERY: 1, ANSWER: 0, AUTHORITY: 4, ADDITIONAL: 4
;; WARNING: recursion requested but not available

;; QUESTION SECTION:
;stanford.edu.                      IN      A

;; AUTHORITY SECTION:
stanford.edu.               172800  IN      NS      avallone.stanford.edu.
stanford.edu.               172800  IN      NS      atalante.stanford.edu.
stanford.edu.               172800  IN      NS      argus.stanford.edu.
stanford.edu.               172800  IN      NS      aerathea.stanford.edu.

;; ADDITIONAL SECTION:
avallone.stanford.edu.      172800  IN      A       171.64.7.88
atalante.stanford.edu.      172800  IN      A       171.64.7.61
argus.stanford.edu.         172800  IN      A       171.64.7.115
aerathea.stanford.edu.      172800  IN      A       152.3.104.250
```

9. [**5 points**]:

Assuming `stanford.edu` has an IP address (i.e., DNS resource record of type `A`), why is the answer section empty in this reply?

stanford.edu is an alias for another domain name, so it has a CNAME entry. Aliases don't have A records associated with them, so to get the IP address you must find the A record for the domain which stanford.edu is an alias to (i.e. do another lookup)

-2

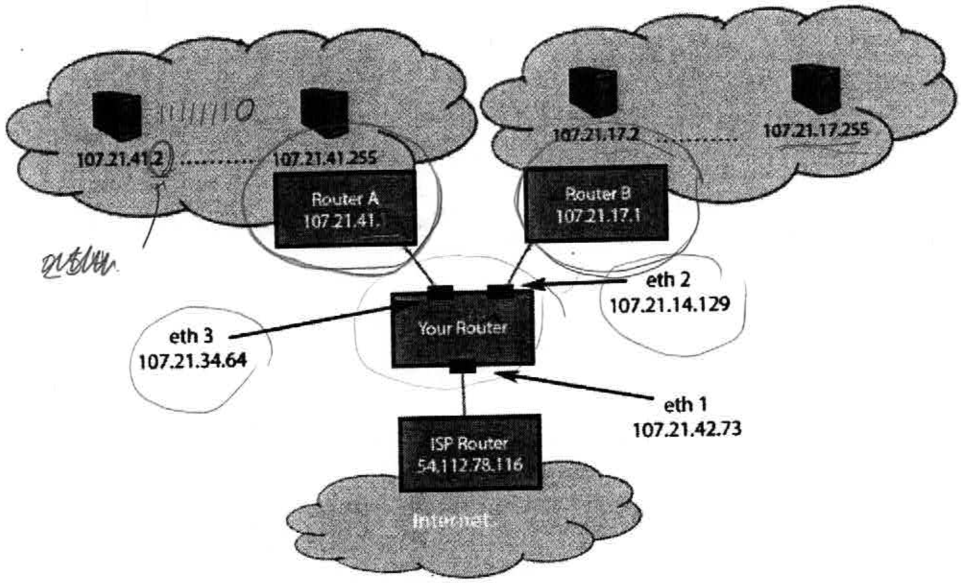
10. [5 points]:

What is the purpose of the records in the additional section—How would DNS break if replies did not contain an additional section or any of the records usually placed here?

The purpose is to give the requester the IP addresses of people to contact. The authority section only gives the names, which are not enough to contact them. without the authority section, DNS responses would not include the address to contact which means DNS is not doing its sole job (leading to a never-ending chain of DNS requests looking for IP addresses but only getting names back...)

11. [10 points]:

A web site hosted on a single server becomes extremely popular. The administrators decide to replace the server with two sets of 200 servers (i.e. 400 servers in total), each set of servers connected to the Internet via a different router (for fault tolerance). The figure shows the topology. The two routers (Router A and Router B) connect to the Internet via a router that you manage ("Your Router"). The administrator of Router A assigns 200 host IP addresses in the range `107.21.41.2–107.21.41.255`, and the administrator of Router B assigns 200 host IP addresses in the range `107.21.17.2–107.21.17.255`. You decide to manually insert routing table entries into "Your Router" to correctly route packets between the servers and the rest of the Internet.



Write down the routing table entries for "Your Router" to correctly route packets between the Internet and the four hundred servers. You need not include routes to "Your Router"'s interfaces. Use as few table entries as possible.

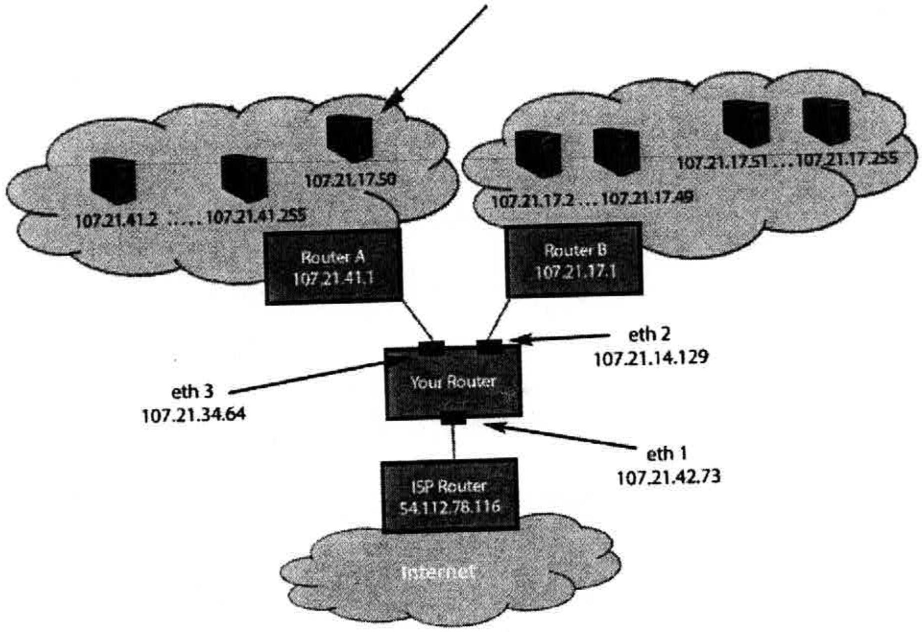
107.21.41.0 ← what to do w/ this one

talked to Gil

destination prefix (e.g. 10.0.0.0)	net mask (e.g. 255.0.0.0)	next hop (e.g. 10.0.0.1)	interface (e.g. eth0)
0.0.0.0	0.0.0.0	54.112.78.116	eth1
107.21.41.0	255.255.255.0	107.21.17.1	eth2
107.21.17.0	255.255.255.0	107.21.41.1	eth3

12. [**5 points**]:

The administrators decide to move the server with IP address `107.21.17.50` from the network behind Router B to the network behind Router A. Assuming that they change no server IP addresses (the moved server keeps the *same* IP address), how do you need to modify the routing tables of "Your Router" to correctly route packets between all servers and the Internet? You may assume the administrators will update Router A and Router B properly to support your change. The new topology looks like this:



Write down any new or modified entries you need:

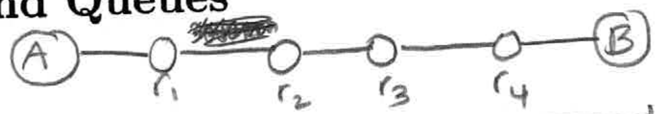
destination prefix	net mask	next hop	interface
107.21.17.50	255.255.255.255	107.21.41.1	eth3

Explain your answer:

This entry would need to be added into the routing table. It will have higher precedence than the prefix match for 107.21.17.0 because the net mask is longer (e.g. the prefix is longer)

5 links.

VI Playback Buffers and Queues



13. [10 points]:

Two endpoints in a voice-over-IP session are connected by a path of 4 routers. All links are running at 1Mb/s and the hosts are separated by 3000km. All packets are of size 1500Bytes. Assume the bit propagation speed is 2×10^8 m/s. Note that 1KB of data is 1024Bytes, but 1Mb/s is 10^6 bits/s.

(i) What is the minimum round trip time (RTT), assuming there is no queueing delay and assuming processing time at each host is negligible?

150 ✓ milliseconds

One-way

prop delay: $3000\ \cancel{km} \times \frac{\cancel{s}}{2\times10^8\ \cancel{m}} \times \frac{10^3\ \cancel{m}}{\cancel{km}} \times \frac{10^3\ ms}{\cancel{s}} = 15\ ms$

pack delay: $1500\ Bytes \times \frac{8\ bits}{Byte} \times \frac{s}{1\ Mb} \times \frac{1\ Mb}{10^6\ bits} \times \frac{10^3\ ms}{s}$

$\times$ 5 links = 60 ms.

75 ms one-way. 150 ms RTT

(ii) For this part only let us assume that one router on the path has a steady queue occupancy of 5 packets. What is the end-to-end delay (one way, not round trip) in this case?

Arrival 5.

123 milliseconds

-1

Queue means need to add additional 4 pack delay..

pack delay: $1500\ Bytes \times \frac{8\ bits}{Byte} \times \frac{s}{1Mb} \times \frac{1Mb}{10^6\ bits} \times \frac{10^3 ms}{s} \times \cancel{4}\ 5$

= 48 ms.

75 + 48 = 123

(iii) Now let us assume the maximum queue occupancy for *every* router queues is 5 packets. What is the maximum end-to-end delay?

267 milliseconds.

$48 \times 4 = 192$

$192 + 75 = 267$

(iv) For part (iii), how long should the playback buffer be at the destination voice-over-IP client be if each packet arrives successfully, without being dropped? Express your answer in bytes.

24000 bytes.

$267 - 75 = 192 \text{ ms}.$

$$\frac{1 \cancel{\text{Mb}}}{\text{s}} \times \frac{10^6 \text{ bits}}{\cancel{\text{Mb}}} \times 192 \cancel{\text{ms}} \times \frac{\text{s}}{10^3 \cancel{\text{ms}}} \times \frac{\text{byte}}{8 \cancel{\text{bits}}} = 24000$$

(v) For part (iii), how long should the playback buffer be at the destination voice-over-IP client if each packet either arrives successfully when first transmitted, or is dropped the first time and then arrives successfully on the second attempt? Assume the sender retransmits a packet only after it finds out that the packet was dropped, which is after twice the maximum end to end delay. Express your answer in bits.

768,000 bits.

267 × 2 + 267 = 801 ms

OK

801 − 75 = 726 ms.

* Same as iv except 726 instead of 192 ms.

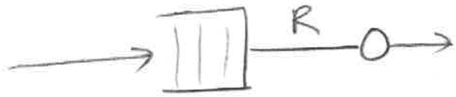
90 750 bytes → 60.5 packets.

→ 61 packets →

1500 Bytes, or 12 000 bits

* 96 000 B not 90750B b/c need to fit whole packets.

→ 768,000 bits.



14. [10 points]:

Recall that a (σ, ρ) leaky-bucket traffic regulator ensures that the number of bytes departing the regulator in any interval of time (t_1, t_2) is bounded by $B(t_1, t_2) \leq \sigma + \rho(t_2 - t_1)$, where σ and ρ are non-negative constants.

Now, consider a queue which drains at rate R. The queue has an infinite buffer size. Answer the following questions.

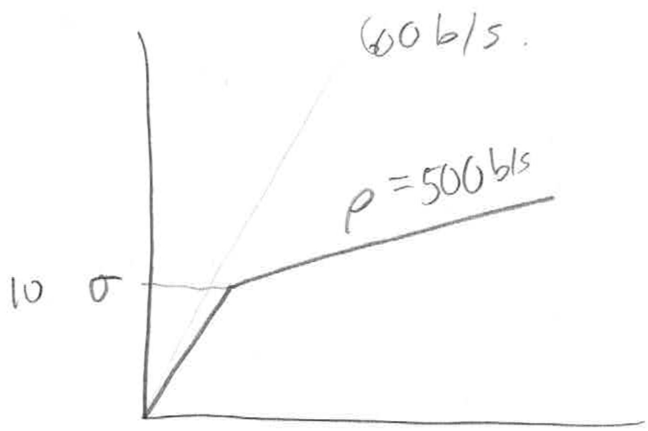
(i) Which of the following is correct?
Circle all that apply. (There may be multiple answers.)

√ **A** If the source feeding the queue is leaky-bucket constrained, then the delay of a bit through the queue is always less than 2σ, regardless of the source's average sending rate, ρ. $\sigma = 0.1$

√ **B** The delay of any bit through the queue is finite only when $\sigma < R$. ρ can be ∞

√ **C** If several leaky-bucket constrained sources (σ_i, ρ_i) feed the queue, and when combined $\sum_i \sigma_i = \sigma$ and $\sum_i \rho_i = \rho < R$, then the delay through the queue will be bounded and finite. Assume σ is finite. X no explanation −1

−2 X **D** If several leaky-bucket constrained sources (σ_i, ρ_i) feed the queue, and $\sum_i \sigma_i = 10$, $\sum_i \rho_i = 500\text{b/s}$, and $R = 600\text{b/s}$, then no bit will ever experience a delay through the queue of more than 100ms.



10 bytes. 600 b/s.

$10 \text{ bytes} \times \frac{8 \text{ bit}}{\text{byte}} \times \frac{s}{600} = 133 \text{ ms.}$

X incorrect explanation −1

$\boxed{6}$