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

Your Name: Answers

SUNet ID: root @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: ____________________

- 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.

1	/10
2	/5
3	/5
4	/5
5	/5
6	/5
7	/5
8	/15
9	/5
10	/5
11	/10
12	/5
13	/10
14	/10
Total	/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).

Answer: 5 seconds

*Total number of frames needed = 100KB / 1KB = 100. Since this is stop-and-wait, the time needed = 100 * 50 = 5000 ms = 5s.*

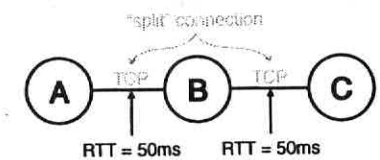
(ii) Let us now suppose that the network drops each segment with a probability of 10%, independently from segment to segment. The network drops both data and acknowledgements. What is the *expected* duration of the transmission? Show your calculations. Your answer must be accurate to two decimal points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

Answer: 7.22 seconds

The time it takes to successfully transmit a packet is a random variable $X = 200 \times Y + 50$, where Y is the number of trials before the first successful transmission. Here, Y is geometrically distributed with success probability $p = \frac{9}{10}$. Hence, the expected time it takes to transmit one packet is $\mathbb{E}[X] = 200 \quad \mathbb{E}[Y] + 50 = 200\left(\frac{1}{p} - 1\right) + 50 = 72.2222$ms. Multiply this by 100, and you get the answer.

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:

$MSS \cdot \sqrt{\frac{3}{2}} = 13,717$ 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?

428-434 kbps

Answer:

$RTT\sqrt{p} = .032$, $\sqrt{\frac{3}{2}}MSS = 13,717$, *so* $\frac{1.2\cdot(1400\cdot 8)}{0.032} = 428,656$, *or 429 kbps. 434 kbps is a more accurate calculation using more significant digits.*

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?

856-888 kbps

Answer:

The connection between B and C is limited by the connection between A and B, since it has a lower packet drop rate. The rate between A and B would be the same as above, except that the RTT is halved from 100ms to 50ms. So its throughput is double than the throughput in the unsplit case. More precisely, $RTT\sqrt{p} = .016$, $\sqrt{\frac{3}{2}}MSS = 13,717$, *so* $\frac{1.22\cdot(1400\cdot 8)}{0.01583/2} = 857,313$, *or 857 kbps.*

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?

___*856-888*___ kbps

Answer:

It is the same as above. The path between B and C has the same throughput as the path between A and B.

5. [5 points]:

Finally, suppose that B does not split the connection, such that packets flow from A to 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?

___*312-315*___ kbps

Answer:

The losses between A and B and B and C affect one another, such that the end-to-end drop rate is 19%. The success rate is 0.9^2*, or 81%. so* $p = 19\%$*.* $RTT\sqrt{p} = 0.044$*.* $\sqrt{\frac{3}{2}}MSS = 13,717$*. so* $\frac{13,717}{0.044} = 311,750$*, or 312kbps. 315kbps is a more accurate calculation using more significant digits.*

III FTP Is For Old People

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:

Answer:

The NAT does not have a mapping from an external IP/port to the internal IP/Port: the FTP server's TCP connection will be refused by the NAT.

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:

Answer:

The NAT does not restrict the user from connecting to the FTP server.

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.

Answer:

- *Client sends a DNS A request for gradadmissions.stanford.edu to 171.64.7.77, using UDP port 53.*
- *Resolver on 171.64.7.77 iterates from root server (for edu) to TLD server (for stanford) to stanford's server (for gradadmissions), finally getting the address and sending it back to the client, using UDP port 57.*
- *Client sends TCP SYN packet to port 80, dest IP address returned by resolver*
- *Server sends SYN-ACK, client completes three way handshake with ACK, sends GET request as TCP data. The GET request is for* `/inquiry/onlineinq.htm`*.*
- *Server sends back web page. If the client requests a persistent connection, the server keeps the socket open until the timeout. Otherwise, the server closes its sending end of the connection, sending a FIN packet.*

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?

Answer:

`a.edu-servers.net` does not know the answer, only Stanford's servers do. This reply tells the client to re-send its query to argus or one of the other servers.

10. [5 points]:

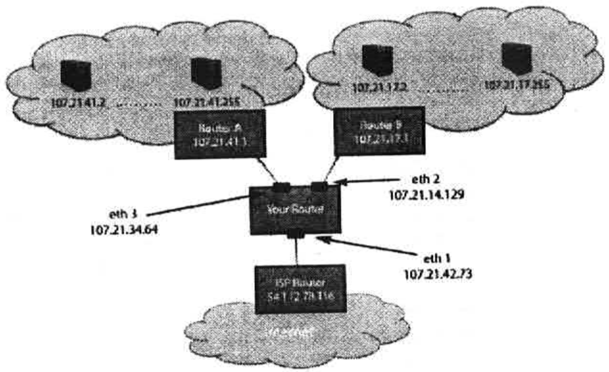
What is the purpose of the records in the additional section—How would DNS break if replies did not contain an additional section or any of the records usually placed here?

Answer:

These are glue records. Without glue records, clients would not be able to find the address of `argus.stanford.edu` or the other three NS records without contacting one of those four machines, which they cannot do without the machine's address. Thus, the parent zone must supply the IP addresses of Stanford's name servers or there would be no way for clients to contact those servers.

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.

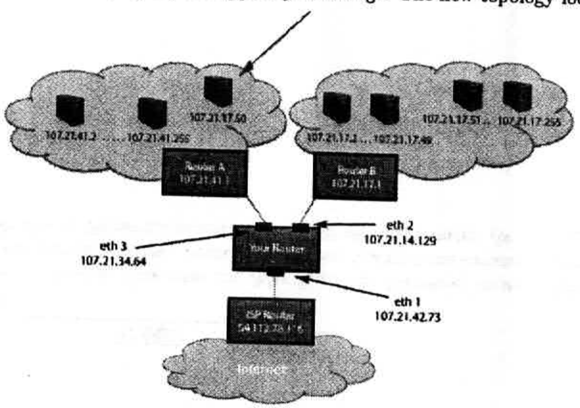


Write down the routing table entries for "Your Router" to correctly route packets between the Internet and the four hundred servers. You need not include routes to "Your Router"'s interfaces. Use as few table entries as possible.

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)
107.21.41.0	255.255.255.0	107.21.41.1	eth1
107.21.17.0	255.255.255.0	107.21.17.1	eth2
0.0.0.0	0.0.0.0	54.112.78.116	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*	*eth1*

Explain your answer:

Answer:

Since routers use longest prefix match, a /32 routing entry for just that one destination IP address will re-route traffic to Router A without disrupting traffic to any other servers behind B.

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?

Answer: *150* milliseconds

*RTT = 2 * (prop + packetization * (5 hops)). The packetization delay = 1500B/1e6 = 12ms per hop. For 5 hops = 60ms. The propagation delay = 3000km / 2e8 m/s = 15ms. Therefore, the RTT is 2 * 75 ms = 150 ms*

(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?

Answer: *135* milliseconds

*Average queueing delay = 5 packets / (1 packet) * 12 ms = 60 ms. So, total end-to-end delay is 60+75ms = 135 ms.*

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

315 milliseconds.

Answer:
*Maximum queueing delay is 4 * 60 = 240ms. So, the maximum e2e delay is 240+75ms = 315ms.*

(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.

30,000 bytes.

Answer:
The min e2e delay is 75ms, max e2e delay is 315ms. Therefore, we need 315 - 75 = 240ms of buffer. Put another way, the playback buffer needs to absorb the variation caused by the queueing delay. At 1Mb/s, 240ms corresponds to 240,000 bits, 30,000bytes, (or 29.3Kbytes).

(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.

870,000 bits.

Answer:
The min e2e delay is still 75ms. The maximum time a packet takes to arrive is $2 \times 315ms$ (to send and find out it failed) + 315ms (to resend successfully). Recall that a playback buffer should never run empty inorder to provide a pleasant playback experience. Therefore, it should absorb the variation, and therefore, we need at most $3 \times 315 - 75 = 870ms$ worth of buffering. At 1Mb/s, this corresponds to 870,000 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 $B(t_1, t_2) \le \sigma + \rho(t_2 - t_1)$, 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.

B The delay of any bit through the queue is finite only when $\sigma < R$.

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.

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.

Answer:

(c) is correct because the arrival rate is less than the departure rate, so the queue will not grow. σ is finite, so the maximum occupancy of the queue remains bounded. (d) is correct because the maximum delay through the queue is bounded by $\sigma/R = 10/600 < 100ms$.

Figure 1: Cartoon based off research from Stanford by student Katherine N. Dektar, prof. Balaji Prabhakar (EE/CS), and prof. Deborah M. Gordon (biologist).