COSC 4377 – Networking - Kevin B Long

# interlocking-uh-m-186.eps

Homework #3

Due 11:59am, Saturday, 2 March 2019

Multiple submissions accepted.

Name:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Peoplesoft ID: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. (20 **bonus** pts) What follows is an **experimental problem**. If you can get it installed and working and make it to the end, it’s worth 20 extra homework points. The due date for this question is the Monday after Spring Break.

Install **mailcatcher** on your machine by following the one line of installation instructions at <https://mailcatcher.me/>. Basically you need to type “gem install mailcatcher” or perhaps “sudo gem install mailcatcher” on a machine. I did this on a Mac and on Windows 7.1.

Once installed, in the same command prompt window, run mailcatcher. You should see something like this:

KEVILONG-M-70EP:~ kevinlong$ mailcatcher

Starting MailCatcher

==> smtp://127.0.0.1:1025

==> http://127.0.0.1:1080

\*\*\* MailCatcher runs as a daemon by default. Go to the web interface to quit.

KEVILONG-M-70EP:~ kevinlong$

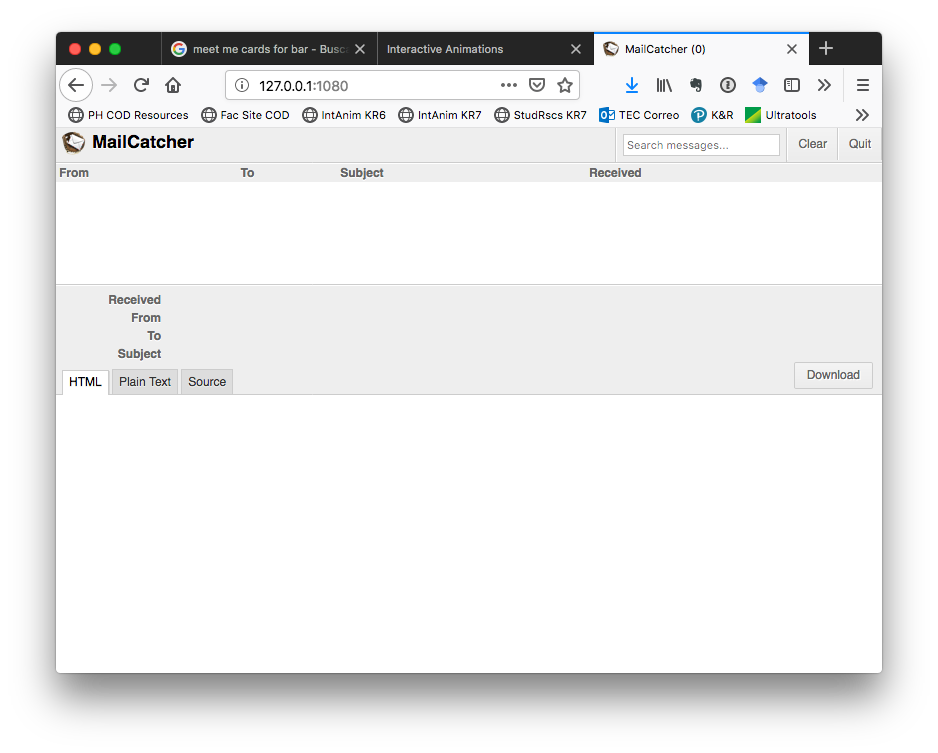
You are now running mailcatcher in the background and will forever, until you either kill it, reboot, or open your web browser and point to the HTTP interface using the URL shown and stop it.

For Windows, you’ll need to install GEM. On Macs it seems to be installed already. However, that may be just from my past work on my Mac. This stackoverflow article gives some advice on how to proceed.

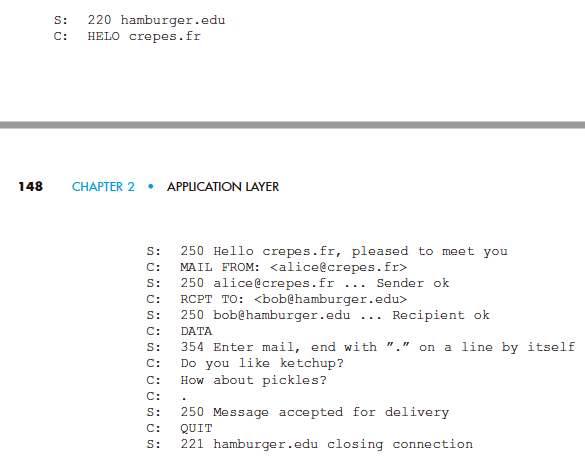
<https://stackoverflow.com/questions/18908708/installing-ruby-gem-in-windows?utm_medium=organic&utm_source=google_rich_qa&utm_campaign=google_rich_qa>

What have you done? You’ve just started an email server, an SMTP server to be specific. It will accept connections from any email program you have on your computer, and will immediately display that email in a web browser for you to verify that you successfully transmitted an email to an SMTP server. But unlike a real SMTP server, it won’t try and send it anywhere; it’s purely for display. So you can pretend to send an email containing whatever you want to whomever you want at whatever fake address you like.

Open up a browser and go to <http://127.0.0.1:1080> (127.0.0.1 is your machine, and 1080 is the port number to which the mailcatcher server has a web server interface listening. You should see something like this:



With this browser window open (and constantly monitoring the mailcatcher server you’re running), we want to try and connect using SMTP to the server and see if we can deliver a properly-formatted email message. We will follow the SMTP script in our textbook that had Alice sending an email message to be delivered to Bob. In the 6th edition it’s on page 123, in the 7th it’s on page 147. It looks like this:



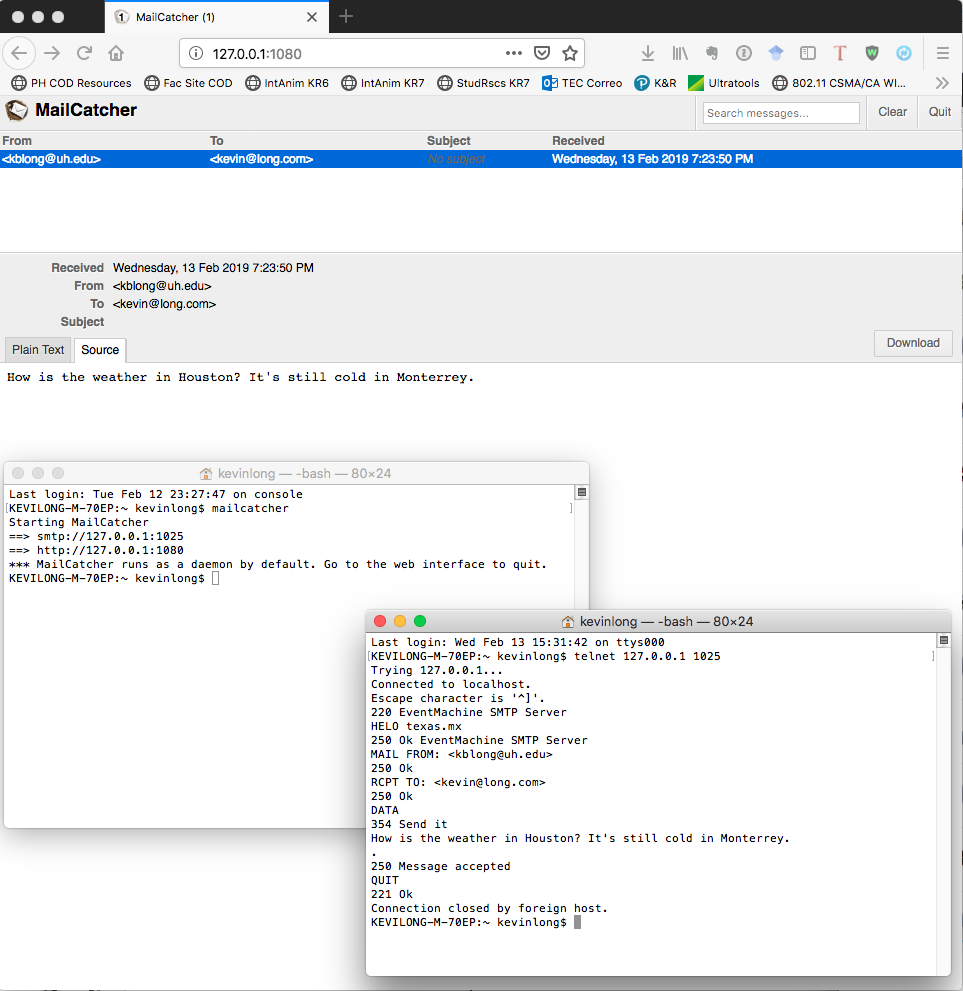
Open a second command window, and telnet to the program you just created by saying “telnet 127.0.0.1 1025” (1025 is the port number the SMTP mailcatcher server is monitoring). Follow the sequence of commands from the book, but substitute in values that personalize the email for you. Send an email from your email account to a friend, and in the body of the message include whatever you want, but include your name and peoplesoft ID # so we know it’s you. The server will not forward it anywhere else; it will die on your machine. Once you’re done sending, return to the mailcatcher web page you opened and refresh to see the email you just sent. If you don’t see one, you’ve got a problem; it should be instantaneous. Assuming you can see it, select the email, and then click on the “Source” tab to display its contents.

Organize best you can three windows on your screen:

* The telnet command prompt window you used to compose the email
* The telnet window you used to install and run mailcatcher
* The browser showing mailcatcher and your email

<<<Paste it here>>>

Here’s a snapshot I took of my screen. It’s in the homework folder too, if you want to zoom in on it. Remember, you must make this customized to you – use your email addresses, create your own short message body. If it matches the book or anyone else, you’ll get no credit for the problem.



**From chapter 2:**

1. (4 pts) Referring to Figure 2.4, we see that none of the applications listed in the book’s examples require that all of the data arrives AND that it arrives immediately (because it’s time-sensitive). Can you think of an example of an application that DOES require both characteristics? A network application that requires no data loss and that is also highly time-sensitive? Be specific as to the type of program and the specific type of business or industry that would run it and for what specific purpose.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. (16 pts) From the HTTP Delay Estimation animations at <https://media.pearsoncmg.com/aw/ecs_kurose_compnetwork_7/cw/content/interactiveanimations/http-delay-estimation/index.html>, play around and answer the following questions:
2. (4 pts) When is the total delay in RTTs of a non-persistent connection equal (or approaching equality) to that of a non-persistent connection with parallel connections? Mark all that apply ⌧.
3. ☐ Never
4. ☐ Always
5. ☐ When the number of objects equals the number of parallel connections
6. ☐ When the number of objects equals one
7. ☐ When the number of parallel connections equals one
8. ☐ When there are no objects on the page
9. ☐ When the number of objects is extremely large (approaches ∞)
10. ☐ When the per-object transmission delay is 0
11. ☐ When the per-object transmission delay is 1
12. (4 pts) Is there a variable you are allowed change in this animation tool that would make the total delay in RTTs for persistent connections without pipelining equal that of persistent connections with pipelining? What variable is it and what value would it need to have?

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. (8 pts) Calculate the total number of RTTs required to send 8 objects two different ways: first with non-persistent connections with parallel connnections and second with persistent connections with pipelining. Assume 1.00 RTT transmission delay for the base HTML page and the 8 objects, and assume you can never send more than 4 objects at a time (whether via parallel connections or pipelining).

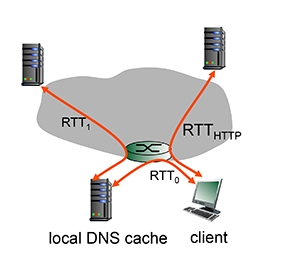
Non-persistent:\_\_\_\_\_\_\_\_\_

Persistent:\_\_\_\_\_\_\_\_\_\_\_\_\_

You must show your work for credit.

1. (20 pts, 4x5) From the DNS and HTTP delays interactive exercise from the text at <http://gaia.cs.umass.edu/kurose_ross/interactive/DNS_HTTP_delay.php>, solve the following problem. The web site will step you through similar randomly-generated problems for practice. *Before doing this problem, you might want to review sections 2.2.1 and 2.2.2 on HTTP (in particular the text surrounding Figure 2.7) and the operation of DNS (in particular the text surrounding Figure 2.19).*

Consider the following network:



Suppose you open a web page on your browser. The IP address for the associated URL is not cached in your computer’s own DNS cache, so a DNS lookup is necessary to obtain the IP address.

Suppose within your Web browser you click on a link to obtain a Web page. The IP address for the associated URL is not cached in your local host, so a DNS lookup is necessary to obtain the IP address. Suppose that two DNS servers are visited before your host receives the IP address from DNS. The first DNS server visited is the local DNS cache, with an RTT delay of RTT0 = 1 msecs. The second DNS server contacted has an RTT of 5 msecs. Initially, let's suppose that the Web page associated with the link contains exactly one object, consisting of a small amount of HTML text. Suppose the RTT between the local host and the Web server containing the object is RTTHTTP = 54 msecs.

1. (4 pts) Assuming zero transmission time for the HTML object, how much time elapses from when the client clicks on the link until the client receives the object?

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. (4 pts) Now suppose the HTML object references 3 very small objects on the same web server. Neglecting transmission times, how much time elapses from when the client clicks on the link until the base object and all 3 additional objects are received from web server at the client, assuming non-persistent HTTP and no parallel TCP connections?
2. (4 pts) Repeat 2. above but assume that the client is configured to support a maximum of 5 parallel TCP connections, with non-persistent HTTP.
3. (4 pts) Repeat 2. above but assume that the client is configured to support a maximum of 5 parallel TCP connections, with persistent HTTP.
4. (2 pts) What type of transport connection is DNS using in this animation?

☐ TCP or ☐⌧UDP?

1. (4 pts) If you had to switch transport protocols to get around a network administrator who chose to block all incoming connections to the local DNS server that did not initiate with a handshake, by how many msecs if any would this change our answers? Be sure to indicate if this would change your total time up or down by marking “+” or “-“. Justify your answer.

☐ + ☐ - \_\_\_\_\_\_\_\_\_ msecs

Justification: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. (6 pts) Recall that TCP can be enhanced with TLS/SSL to provide process-to-process security services, including encryption.
   1. (1 pt) Do these security protocols like TLS and SSL operate at the transport layer or the application layer?

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

* 1. (5 pts) If the application developer wants to use TCP but wants packets to be encrypted with TLS, what does the developer have to do?

☐ Call the encryption-capable version of TCP: TCP/TLS

☐ Activate the transport layer security, which is what TLS stands for, which will handle encryption and decryption at the transport layer.

☐ Implement security at the application layer, negating the need for TCP, and allowing us to move to UDP for the transport layer

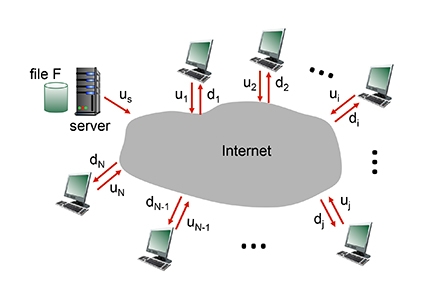
☐ Insert an encryption mini-layer between the transport and IP layer

☐ Run TLS as an app-layer service, and pass it your messages from the browser to be encrypted before deliver to TCP

1. (12 pts, 3x4) Go to the CS vs P2P interactive exercise at:

<http://gaia.cs.umass.edu/kurose_ross/interactive/CS_vs_P2P_download.php>

And practice until ready to solve the following problem, providing the answers below.



The problem is to distribute a file of size F = 5 Gbits to each of these 9 peers. Suppose the server has an upload rate of us = 100 Mbps, and that the 9 peers have upload and download rates in bits per second as shown in the table below. u1 is the upload speed for the 1st peer, and d1 is the download speed for the 1st peer. Thus, u1’s upload speed is 15Mbps and its download speed is 40Mbps.

|  |  |  |  |
| --- | --- | --- | --- |
| u1 | 1.50E+07 | d1 | 4.00E+07 |
| u2 | 2.20E+07 | d2 | 3.90E+07 |
| u3 | 3.00E+07 | d3 | 4.50E+07 |
| u4 | 2.85E+07 | d4 | 5.00E+07 |
| u5 | 1.90E+07 | d5 | 4.60E+07 |
| u6 | 1.60E+09 | d6 | 5.30E+07 |
| u7 | 1.50E+07 | d7 | 4.90E+07 |
| u8 | 2.60E+07 | d8 | 3.50E+07 |
| u9 | 1.70E+07 | d9 | 3.80E+07 |

Answer the following questions:

1. (2 pts) What is the minimum time needed to distribute this file from the central server to the 9 peers using the client-server model? (Hint: see equation 2.1 in the text).

\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. (2 pts) For question 1, what is the root case of this specific minimum time: the server upload rate, or a specific client's download rate (and if so, which client?)? Explain your answer.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. (2 pts) What is the minimum time needed to distribute this file using peer-to-peer download? (Hint: see equation 2.2 in the text).

\_\_\_\_\_\_\_\_\_\_\_\_\_\_

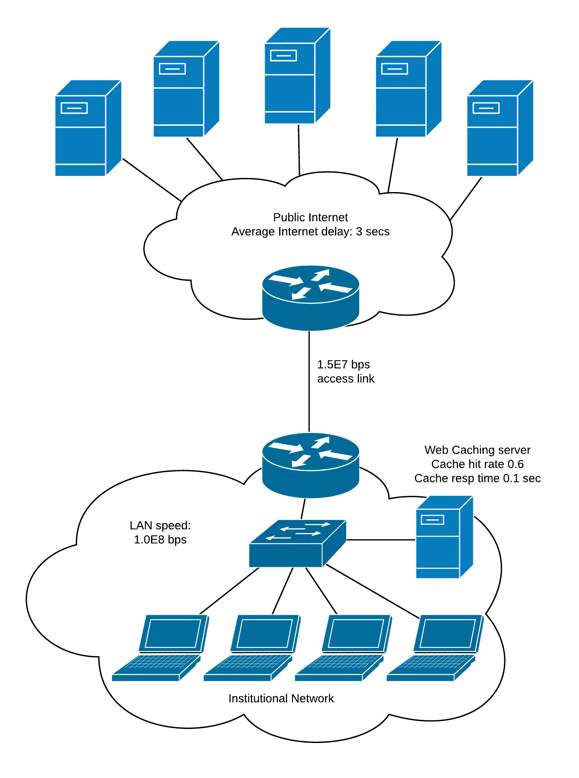
1. (2 pts) For question 3, what is the root case of this specific minimum time: the server upload rate, a specific client's download rate (amd of so, which client?), or the sum of the server and peer upload rates? Explain your answer.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. (2 pts) If you increased the server’s connection speed to 1 Gbps, what would the new minimum times be?

Client-Server:\_\_\_\_\_\_\_\_\_\_\_\_ Peer-to-Peer:\_\_\_\_\_\_\_\_\_\_\_\_\_

1. (24 pts) Consider Figure 2.12, for which there is an institutional network connected to the Internet:



There are two routers shown, an institutional router, and one at the ISP.

A few definitions:

* Objects are either HTML base pages or additional objects on those pages. All of them must be retrieved from remote “origin” web servers on the Internet, shown across the top of the diagram.
* Average Internet delay refers to the time required to return with a response from a web server from the point where the request has arrived at the ISP’s router (the top one) over the access link. Section 2.2.5 might help make this less confusing.

Our network exhibits the following characteristics:

Average HTTP requests per second: 16 (the “arrival rate”)

Average HTTP object size: 850,000 bits

Average Internet delay: 3 secs

Access link speed: 1.5E7 bps

LAN speed: 1.0E8 bps

Router processing delay: 0 ms

Router queueing delay: 0 ms

We will ignore the Web Caching Server in the diagram for now.

1. (4 pts) Model the total average response time. This will include the time to send the object to from the laptop to the institutional router, across the access link, the Internet delay, back across the access link and the LAN to the laptop.

First calculate how long it takes to send one object. That’s the same L/R formula we’ve seen before. If your link was 100,000 bits per second and object was 50,000 bits, then the time to send one object would be 50,000/100,000 = 0.5 seconds. We will assign that to the variable Δ. We’ll let ß represent the number of arriving object requests per second, also called the “arrival rate”.

Δ for Access Link: \_\_\_\_\_\_\_\_\_\_\_\_\_\_sec

Δ for LAN: \_\_\_\_\_\_\_\_\_\_\_\_\_\_secs

ß: \_\_\_\_\_\_\_\_\_\_\_\_\_\_

Answer: 16

1. (4 pts) Calculate Δß, the traffic intensity. It’s also the % utilization of the link. Hopefully the values are less than 1, since Δß =1 is a fully-saturated link with infinite delay.

Δß for Access Link: \_\_\_\_\_\_\_\_\_\_\_\_\_\_

Δß for LAN: \_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. (4 pts) Finally, calculate the average network delay for the access link and the LAN. Use the formula Δ / (1 – Δß). Here’s why this makes sense: if you show up with an object that needs Δ time, and the network is very busy already, then the denominator will be very small, increasing the result, and your wait time will be large. So if your network was 90% busy (Δß = .9), then a 500Kb object would require Δ / (1 – Δß) = 5 seconds. If the link was 99% busy, 50 sec. Conversely, 20% busy: .625 seconds.

Give your answer in scientific notation, with 2 places past the decimal point.

Avg delay for Access Link: \_\_\_\_\_\_\_\_\_\_\_\_\_\_sec

Avg delay for LAN: \_\_\_\_\_\_\_\_\_\_\_\_\_\_sec

Please copy your answers into the table below; we will use a table like this on our exam, so I want you to get familiar with it.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Network | Time to send 1 obj | Arrival rate | Traffic Intensity | Link delay |
| LAN |  |  |  |  |
| Access Link |  |  |  |  |

1. (4 pts) Now that we have all the parts, we calculate the total average response time:

Average LAN delay + average access link delay + average Internet delay + average access link delay + average LAN delay.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_sec

1. (4 pts) If you increase the request rate from the browsers to 18 per second, what will happen to the network? Explain how traffic intensity and link delay values are significant to your answer.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. (6 pts, ½ pt ea for table, 2 for total) We return back to the original scenario but add our web caching server to the analysis. With web caching servers, the router is reprogrammed to redirect 100% of incoming and outgoing HTTP requests to the web caching server. It intercepts everything coming and going. It adds a processing delay to all LAN to Internet outgoing object requests.;pl. But the delay is only in one direction. Unless requested otherwise (e.g. with a NOCACHE label), the server will make a copy of incoming objects in order to provide copies locally to users.

We find that our cache can provide object requests 60% of the time.

To summarize:

Cache hit rate: 60%

Cache object delay: 0.2 ms

Find the new total response time. If you apply the miss rate (1 – hit rate) to the access link’s arrival rate, it will cascade through to a new link delay.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Network | Time to send 1 obj | Arrival rate | Traffic Intensity | Link delay |
| LAN |  |  |  |  |
| Access Link |  |  |  |  |

Total average response time with cache: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_sec

1. We discussed content-delivery networks at length in class.
   1. (8 pts) There are two key server placement philosophies in play with many CDN networks. What are they? You may need to refer to section 2.6.3 of the 7th edition of the textbook.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ and \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

* 1. (8 pts) In the book there is a reference to an article that provided our book’s authors this information. Find the article, and cite the example company mentioned in the article as the primary example of each type.

Company 1:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Company 2:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_