Computer Networks HW 2

Shane Cincotta

March 31, 2020

1 Chapter 2, Review Question 6

I would use UDP. I would choose UDP for a variety of reasons including: TCP requires a minimum of 2 roundtrip times, while UDP only requires one. This then ensures that UDP requires less time to complete the transaction between client and server.

2 Chapter 2, Review Question 7

A smart traffic controller requires no data loss and is also highly time-sensitive. In a smart traffic control system, a person controls the flow of traffic based off signals from sensors. If any loss of data or delay in time occurs, there will be more accidents. Thus is is highly time-sensitive and requires no data loss.

3 Chapter 2, Review Question 12

When a customer visits the website, the site will create a cookie for the user. This cookie is stored on the user's host machine and is managed by the browser, this allows the user to not have to explicitily enter their username and password everytime they vist the site. The cookie is present in the cookie header, its generated by the server. If the client tries to access the site again, a cookie will be checked for. If a cookie exists, the client will receive a file with a cookie header, which contains login details for that site.

4 Chapter 2, Review Question 13

Web caching is defined as a temporary storage of documents to reduce bandwidth usage. A web cache keeps copies of requested objects of the client in storage. An object is cached or stored the first time a user visits the page and the next time a user requests the same page, a cache will serve the copy, which helps keep the origin server from getting overloaded. It's like keeping a small offline copy of web objects. The cache has limited space however, so determining which objects are cached is very important, as those objects will load the fastest. One method to determine which objects get placed in the cache is called LRU (least recently used). This essentially means that if a new item is being requested to be put into the cache, but the cache memory is full, the cache will replace the object at the least recently used memory address with the new object. Thus it's impossible to fit EVERY object in the cache, as the cache has limited space.

5 Chapter 2, Review Question 14

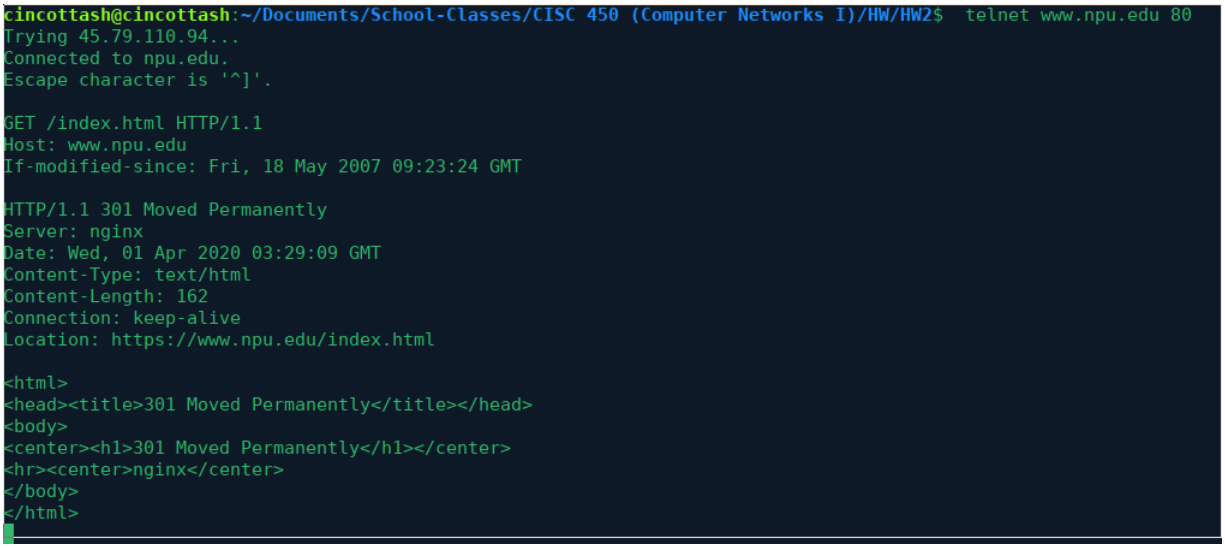

Figure 1:

6 Chapter 2, Review Question 15

SMS can use one or more protocols, some of them include: TPC, HTTP, HTTPS, FPT.

Some popular messaging apps and their protocols include : Vibe-TCP, Skype-VoIP, Facebook Messenger-XMPP, Yahoo Messenger-TCP/HTTP.

Thus some messaging apps can use the same protocols, while some use different protocols.

7 Chapter 2, Review Question 20

Figure 2:

Figure 2 shows the header from an email I received from a .edu email address. You can see the IP address of the host from which the message was sent.

The IP address of the sender's host is only included in the Received: hedaer line if the mail is sent from an SMTP environment. Gmail uses a non-SMTP environment, thus the IP address of the senders host is not included in the Received field.

8 Chapter 2, Review Question 26

The reason is because TCP is a connection-oriented protocol. This entails that TCP has a welcoming socket, in addition to the client's socket that is created during a connection. UDP is a connection-less protocol and as such only has one socket. This socket

is used to send a receive data from the client. Thus for a TCP server to hold n connections, n+1 sockets are needed. N is the number of clients connecting, plus 1 more for the welcoming socket.

9 Chapter 2, Review Question 27

TCP is a connection-oriented protocol, this means that a connection between the server and client must be established before data can be transferred. A connection request must first be accepted by a socket on the TCP server. Thus in order for this listening socket to exist and accept connections, the server must start before the client connects. UDP is a connection-less protocol, this entails that a connection does not need to be established before attempting to send data. UDP also permits packets to be lost, that is, if a sent packet fails to reach it's endpoint, it will not be resent.

10 Chapter 2, Problem 1

10.1 a

False, In HTTP, the client must send one request per object.

10.2 b

True, because the exist on the same domain.

10.3 c

False, a TCP segment can only carry multiple HTTP request message with a persistent connection.

10.4 d

False, the Date: header indicates the time and date when the object in the response was created and sent by the server.

10.5 e

False, the server may send an empty body along with a status code such as 400, 404 if the requested file is unable to be found.

11 Chapter 2, Problem 3

In the transport layer, UDP is needed for DNS, and TCP is needed for HTTP.

In the application layer, DNS and HTTP are needed.

12 Chapter 2, Problem 5

12.1 a

The server was able to find the document. The reply was provided at Tue, 07 Mar 2008 12:39:45GMT.

I found the answer in the red highlighted section of figure 3.

Figure 3:

12.2 b

The document was last modified Sat, 10 Dec 2005 18:27:46 GMT.

I found the answer in the green highlighted section of figure 3.

12.3 c

3874 bytes are in the doucment being returned.

I found the answer in the blue highlighted section of figure 3.

12.4 d

The first 5 bytes are <!doc. THe server agreed to a persistent connection.

I found the answer in the pink highlighted section of figure 3.

13 Chapter 2, Problem 6

13.1 a

According to the the HTTP/1.1 specification (RFC 2616), a client, server or both can send the close of connection signal. To close the connection, the connection header field is sent Connection token close.

13.2 b

The HTTP/1.1 specification (RFC 2616) does not provide encryption services.

13.3 c

According to section 8.14 of RFC-2616, the number of connections that a client have have simultaneously with a server or proxy is 2.

13.4 d

Yes this is possible. The transport connection can be closed from either the client or server side at any time. The server may initiate a signal to close a connection after that connection has been idle for some amount of time. At the same time, the client can send a request to the server (via the same connection), but the client is unaware of the asynchronous closing of the connection.

14 Chapter 2, Problem 7

The amount of time that elapses from when a user clicks the link until the object is received by the client is: Time for DNS lookup + Time to establish TPC connection + Time to send request/receive object.

$= 2RTT_0 + (RTT_1 + RTT_2 + RTT_n)$

15 Chapter 2, Problem 8

15.1 a

THe time to receive an object by the client is: Time to establish TCP connection + time to send the request/receive object + transmission time of object.

$= 2RTT_0 = 16RTT_0$ (multiplied by 8 for the 8 objects).

Thus the total time taken for DNS lookup + time to establish TCP connection + time to send request/receive object + time to receive 8 objects.

$= 18RTT_0 + (RTT_1 + RTT_2 + RTT_n)$

15.2 b

The time to establish a TCP connection + time to request/receive an object is $2RTT_0$.

Thus the total time that elapses from when a user clicks the link until the file is receive by the client is time taken for DNS + time to establish TCP connection + time to send request/receive reply + time to receive 8 objects.

$= 4RTT_0 + (RTT_1 + RTT_2 + RTT_n)$

15.3 c

Because of persistence, the 8 objects can be received in one RTT.
Thus the total time taken for DNS lookup + time to establish TCP connection + time to send request/receive object + time to receive 8 objects.

$= 3RTT_0 + (RTT_1 + RTT_2 + RTT_n)$

16 Chapter 2, Problem 9

16.1 a

The average time $= \frac{L}{R} = \frac{850kbits}{15Mbits/sec} = 0.0567$ seconds.

The average access delay $= \frac{0.0567}{1-(16*0.0567)} = 0.61099$ seconds.

The total average time is determined with the following formula:

Total Average Time = Average access delay + average internet delay = $0.061099 + 3 = 3.61099$ seconds.

16.2 b

With a miss ratio of 0.4, the average access delay becomes $\frac{0.0567}{1-(0.4)*0.9072} = 0.089$ seconds.

Thus the average response time is $0.089 + 3 = 3.089$ seconds.

Thus the total response time is $0.4 * 3.089sec = 1.24$ seconds.

17 Chapter 2, Problem 20

To do this, one would simply need to look at the requests in the cache to different web servers at different points in time. After collection enough data (essentially snapshots of what is in the cache at what time), one could then could make a graph of the frequency distribution of web servers recorded by the cache of the DNS server. The servers with the highest frequency are the most popular web servers among the users of the department.