Saad Saiyed | 200415258

Assignment 2

2. (10 points) 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 n DNS servers are visited before your host receives the IP address from DNS; the successive visits incur an RTT of RTT1, ... , RTTn. Further suppose that the Web page associated with the link contains exactly one object, consisting of a small amount of HTML text. Let RTT0 denote the RTT between the local host and the server containing the object. Assuming zero transmission time of the object, how much time elapses from when the client clicks on the link until the client receives the object? (10 points)

Ans: Considering the given information we can say that,

Total time taken to retrieve IP is $RTT_1 + \dots + RTT_n$

After receiving IP address we must develop the connection, in this case it will be TCP connection which will take RTT_0 and after that we will receive the object upon the request which will consume another RTT_0.

So total time elapsed from the client's click to receiving the object is;

$RTT_0 + RTT_0 + RTT_1 + \dots + RTT_n$

$= 2\,RTT_0 + RTT_1 + \dots + RTT_n$

3. (15 points) Referring to Question 2, suppose the HTML file references eight very small objects on the same server. Neglecting transmission times, how much time elapses with

a. Non-persistent HTTP with no parallel TCP connections?

Ans a: With non-persistent HTTP request with no parallel TCP connection the eight files will consume [(8 * 2 RTT_0) + 2 RTT_0 + RTT_1 + ... + RTT_n] time. Here we have additional eight times two RTT_0 which is the time to get the eight individual objects one by one. The rest of the time is to get the IP address and to produce HTTP request. The final result will be;

$18\ RTT_0 + RTT_1 + \ldots + RTT_n$

b. Non-persistent HTTP with the browser configured for 5 parallel Connections?

Ans b: The 5 parallel connection will consume 2 RTT_0 and TCP connection will consume 2 RTT_0. This will result in;

$2\ RTT_0 + 2\ RTT_0 + 2\ RTT_0 + RTT_1 + \ldots + RTT_n$

$= 6\ RTT_0 + RTT_1 + \ldots\ RTT_n$

c. Persistent HTTP?

Ans c: For persistent HTTP connection requests there is only one thing different which is receiving eight objects. They will be received with no additional requests. This will result in one RTT_0 for receiving objects and rest for connection building;

$RTT_0 + 2\ RTT_0 + RTT_1 + \ldots + RTT_n$

$= 3\ RTT_0 + RTT_1 + \ldots\ RTT_n$

4. (15 points) Consider a short, 10-meter link, over which a sender can transmit at a rate of 150 bits/sec in both directions. Suppose that packets containing data are 100,000 bits long, and packets containing only control (e.g., ACK or handshaking) are 200 bits long. Assume that N parallel connections each get 1/N of the link bandwidth. Now consider the

HTTP protocol, and suppose that each downloaded object is 100 Kbits long, and that the initial downloaded object contains 10 referenced objects from the same sender. Would parallel downloads via parallel instances of non-persistent HTTP make sense in this case? Now consider persistent HTTP. Do you expect significant gains over the non-persistent case? Justify and explain your answer.

Ans: We know that,

Length of connection = 10 meters, Transmission rate = 150 bits/sec, Size of data = 100,000 bits, Control size = 200 bits, No. of parallel Connection = N, Capacity of each downloadable object = 100 Kbits, Initial reference object size = 10.

Considering Tρ as one way propagation delay,

For parallel downloads using non-persistent connection. The download capacity for 10 connections over 150 bits/sec will be 15 bits per SECOND. So the total delay will be;

(200/150 + Tρ + 200/150 +Tρ + 200/150 + Tρ + 100000/150 + Tρ) + (200/15 + Tρ +200/15 +Tρ + 200/15 + Tρ + 100000/15 + Tρ)

= ((200 + 200 + 200 + 10000)/150 + 4Tρ) + ((200 + 200 + 200 + 10000)/15 + 4Tρ)

= 7377 + 8Tρ sec

For parallel downloads using persistent connection, total delay will be;

(200/150 + Tρ + 200/150 + Tρ + 200/150 + Tρ + 100000/150 + Tρ) + 10*(200/150 + Tρ + 100000/150 + Tρ)

= 7351 + 24Tρ sec

Now generally we assume the light speed for the calculation so

Tρ = 10/(300*(10^6)), where 300*(10^6) is the speed of light.

~ 0.03 micro sec

This proves that propagation delay is very less than transmission delay. So we get a slight increase in persistent connection over non persistent connection.

5. (15 points) Consider distributing a file of F bits to N peers using a client-server architecture. Assume a fluid model where the server can simultaneously transmit to multiple peers, transmitting to each peer at different rates, as long as the combined rate does not exceed us.

a. **Suppose that us/N ≤ dmin. Specify a distribution scheme that has a distribution time of NF/us.**

Ans: Suppose that the client receives data in parallel fashion and the Transmission rate is us/N. Now considering us/N <= dmin where minimum download rate i.e. the rate of transmission from server to client will not increase from dmin, the maximum rate at which server can transmit the data is us/N which will be the minimum rate at which client receives the data.

So the total time to receive the data by client will be F/(us/N).

Therefore to transfer all the files from server to all the clients, the distributed time will be *NF/us*

b. **Suppose that us/N ≥ dmin. Specify a distribution scheme that has a distribution time of F/dmin.**

Ans: Considering us/N >= dmin i.e. the rate of transmission from server to client will never be less than dmin.

us/N >= dmin

=> us >= N * dmin,

From the above expression we can say that the minimum download rate is less than transmission rate.

So client can receive data at the rate dmin i.e. so client will receive a file at rate dmin.

Therefore total distributed time to transfer all the files will be *F / dmin.*

c. **Conclude that the minimum distribution time is in general given by max{NF/us, F/dmin}.**

Ans: We know that distribution time is defined by,

Dcs >= max{NF/us, F/dmin}

Case 1: Considering, us/N <= dmin i.e. the rate of transmission from server to client will not increase from dmin

∴ Dcs >= NF/us

But we already proved this case in part a. that Dcs <= NF / us

∴ *Dcs = NF/us*

Case 2: Considering, us/N >= dmin,

∴ Dcs >= F/dmin

But we already proved in part b. that Dcs <= F/dmin

∴ *Dcs = F/dmin*

Now combining two cases we can say that *Dcs = max{NF/us, F/dmin}*.

6. (10 points) In this problem, we use the useful dig tool available on Unix and Linux hosts to explore the hierarchy of DNS servers. Recall that in Figure 2.19, a DNS server higher in the DNS hierarchy delegates a DNS query to a DNS server lower in the hierarchy, by sending back to the DNS client the name of that lower-level DNS server. First read the man page for dig, and then answer the following questions.

a. **Starting with a root DNS server (from one of the root servers [a-m].rootservers.net), initiate a sequence of queries for the IP address for your**

department's Web server by using dig. Show the list of the names of DNS servers in the delegation chain in answering your query.

Ans: If we use A.ROOT-SERVERS.NET as the root server name. If we use dig operation on university of regina's official website with following command

dig A.ROOT-SERVERS.NET A uregina.ca

We will get following dns nameservers in the delegation chain. (Performed on putty)

fw.uregina.ca.

ns1.d-zone.ca.

ns2.d-zone.ca.

b. **Repeat part a) for several popular Web sites, such as google.com, yahoo.com, or amazon.com.?**

Ans: Command: *dig A.ROOT-SERVERS.NET A google.com*

ns1.google.com

ns2.google.com

ns.3google.com

ns4.google.com

7. (10 points) Suppose Client A initiates a Telnet session with Server S. At about the same time, Client B also initiates a Telnet session with Server S. Provide possible source and destination port numbers for:

a. **The segments sent from A to S.**

Ans: The port numbers do not have to be the same as this but these are used just for example. We will say that Client A has 467 port and Client B has 513 and Server S has 20 port.

So for A to S: Source port is 467 and Destination port is 20.

b. **The segments sent from B to S.**

Ans: For B to S: Source port is 513 and Destination port is 20.

c. **The segments sent from S to A.**

Ans: For S to A: Source port is 20 and Destination port is 467.

d. **The segments sent from S to B.**

Ans: For S to B: Source port is 20 and Destination port is 513.

e. **If A and B are different hosts, is it possible that the source port number in the segments from A to S is the same as that from B to S?**

Ans: If both client A and client B are sharing the same source port on *different* hosts, yes the segments will reach the destination as the instruction also has the IP address of the host.

f. **how about if they are the same host?**

Ans: No, as the segment will reach only the first client it will find as they share the same name and address.