HW#2

P4. Consider the following string of ASCII characters that were captured by Wireshark when the browser sent an HTTP GET message (i.e., this is the actual content of an HTTP GET message). The characters <*cr*><*lf*>are carriage return and line-feed characters (that is, the italized character string <*cr*> in the text below represents the single carriage-return character that was contained at that point in the HTTP header). Answer the following questions, indicating where in the HTTP GET message below you find the answer.

GET /cs453/index.html HTTP/1.1 *<cr><lf>*Host: gaia.cs.umass.edu *<cr><lf>*User-Agent: Mozilla/5.0 (Windows;U; Windows NT 5.1; en-US; rv:1.7.2) Gecko/20040804 Netscape/7.2(ax) *<cr><lf>*Accept:ext/xml,application/xml,application/xhtml+xml,text/html;q=0.9,text/plain;q=0.8,image/png,*/*;q=0.5 *<cr><lf>*Accept-Language:en-us,en;q=0.5 *<cr><lf>*AcceptEncoding: zip,deflate<cr><lf>Accept-Charset: ISO -8859-1,utf-8;q=0.7,*;q=0.7Keep-Alive:300 *<cr><lf>* Connection:keep-alive *<cr><lf><cr><lf>*

a. What is the URL of the document requested by the browser?

b. What version of HTTP is the browser running?

c. Does the browser request a non-persistent or a persistent connection?

d. What is the IP address of the host on which the browser is running?

e. What type of browser initiates this message? Why is the browser type needed in an HTTP request message?

Solution:

a. The document requested was http://gaia.cs.umass.edu/cs453/index.html. The Host: field indicates the server's name and the parameter for GET method, /cs453/index.html corresponds to the file name.

b. The browser is running HTTP version 1.1, as indicated as the last parameter of GET command.

c. The browser is requesting a persistent connection, as indicated by the Connection: keep-alive. Otherwise Connection: close would be there.

d. This is a trick question. This information is not contained in an HTTP message anywhere. So there is no way to tell this from looking at the exchange of HTTP messages alone. One would need information from the IP datagrams (that carried the TCP segment that carried the HTTP GET request) to answer this question.

e. The type of browser is Mozilla/5.0. The browser type information is needed by the server to send different versions of the same object to different types of browsers.

P7. Assume that the RTT between a client and the local DNS server is RTT_l, while the RTT

between the local DNS server and other DNS servers is RTT_r. Assume that no DNS server performs caching.

a. What is the total response time for the scenario illustrated in Figure 2.19?

b. What is the total response time for the scenario illustrated in Figure 2.20?

c. Assume now that the DNS record for the requested name is cached at the local DNS server. What is the total response time for the two scenarios?

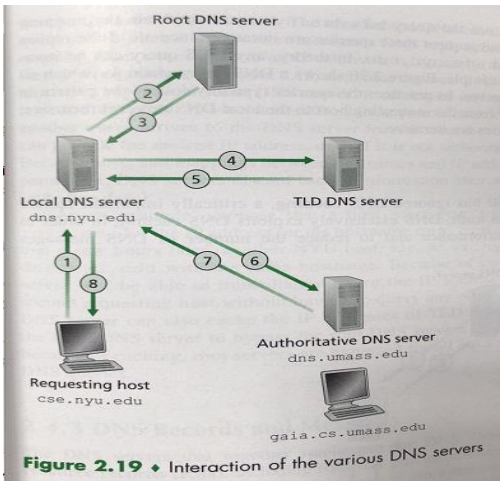

Figure 2.19 ♦ Interaction of the various DNS servers

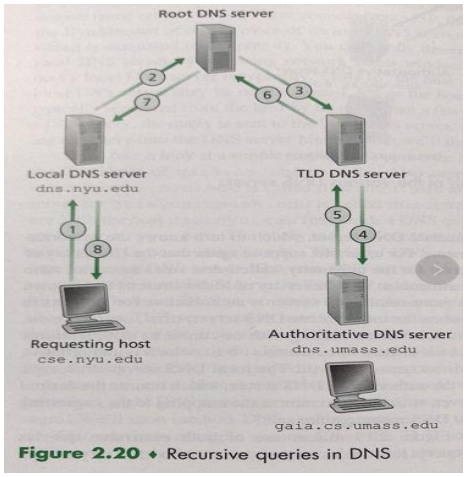

Figure 2.20 ♦ Recursive queries in DNS

Solution:

a. What we need to do for this problem is to count the number of RTTs in Figure 2.19 before the query response to come back. Therefore the answer is $RTT_l + 3RTT_r$.

b. This problem can be solved in the same way as problem (a) in the above so that the answer is $RTT_l + 3RTT_r$. Here the round trip time between any pairs of authoritative DNS server is assumed to be RTT_r.

c. The answer is RTT_l in both cases since the local DNS server will respond with one in its cache.

P10. Assume you request a webpage consisting of one document and five images. The document size is 1 kbyte, all images have the same size of 50 kbytes, the download rate is 1Mbps, and the RTT is 100ms. How long does it take to obtain the whole webpage under the following conditions? (Assume no DNS name query is needed and the impact of the request line and the headers in the HTTP messages is negligible).

a. Non-persistent HTTP with serial connections.

b. Non-persistent HTTP with two parallel connections.

c. Non-persistent HTTP with six parallel connections.

d. Persistent HTTP with one connection.

Solution:

a. Under non-persistent HTTP with serial connections, each object takes two RTTs and the transmission delay of the object for its retrieval. Since we have six objects to transfer, the total time to complete the delivery of this page is 2 * RTT + transmission delay of a document + 5 (2*RTT + transmission delay of an image), leading to 12*RTT + 251(=1kbyte+50kbyte) * 8/1Mbps = 3.2 seconds.

b. Under non-persistent HTTP with two parallel connections, we can get two obj ects over two parallel connections at one time. So, the total time is 3*(2RTT+ 50kbytes/1Mbps) = 6*RTT+150kbytes/1Mbps = 1.8 seconds. Here two GET co mmands are sent in parallel and the physical link speed is assumed to be gre ater than 2*1Mbps so that the transmission delays of two objects are overlapp ed.

c. Under non-persistent HTTP with six parallel connections, the six objects can be fetched at the same time through six connections. So the total time is 2*RTT + 50Kbyte / 1Mbps = 0.6 seconds. Here the physical link speed is assumed to be greater than 6*1Mbps. When the physical link speed is the same as the download rate 1Mbps, the total delay is 2 * RTT + (1kbyte + 50kbytes * 5)*8/1Mbps.

d. Under persistent HTTP with one connection, the total time is 2*RTT+1kbte/1Mps 5*50kbytes/1Mbs = 2*RTT + 251kbytes / 1Mbps = 2.2 seconds. Here six GET commands are sent back-to-back after a TCP connection is open and the six objects also come back consecutively.

P19. 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.

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

Solution:

a) The following delegation chain is used for gaia.cs.umass.edu, a.root-servers.net, E.GTLD-SERVERS.NET, ns1.umass.edu(authoritative).

First command: dig +norecurse @a.root-servers.net any gaia.cs.umass.edu

;; AUTHORITY SECTION:

edu. 172800 IN NS E.GTLD-SERVERS.NET.

edu.	172800	IN	NS	A.GTLD-SERVERS.NET.
edu.	172800	IN	NS	G3.NSTLD.COM.
edu.	172800	IN	NS	D.GTLD-SERVERS.NET.
edu.	172800	IN	NS	H3.NSTLD.COM.
edu.	172800	IN	NS	L3.NSTLD.COM.
edu.	172800	IN	NS	M3.NSTLD.COM.
edu.	172800	IN	NS	C.GTLD-SERVERS.NET.

Among all returned edu DNS servers, we send a query to the first one.

dig +norecurse @E.GTLD-SERVERS.NET any gaia.cs.umass.edu

umass.edu.	172800	IN	NS	ns1.umass.edu.
umass.edu.	172800	IN	NS	ns2.umass.edu.
umass.edu.	172800	IN	NS	ns3.umass.edu.

Among all three returned authoritative DNS servers, we send a query to the first one.

dig +norecurse @ns1.umass.edu any gaia.cs.umass.edu

gaia.cs.umass.edu.	21600	IN	A	128.119.245.12

b) The answer for google.com could be: a.root-servers.net, E.GTLD-SERVERS.NET, and ns1.google.com(authoritative).

P23. 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 u_s.

a. Suppose that $u_s/N \leq d_{min}$. Specify a distribution scheme that has a distribution time of NF/u_s.

b. Suppose that $u_s/N \geq d_{min}$. Specify a distribution scheme that has a distribution time of F/d_{min}.

c. Conclude that the minimum distribution time is in general given by $\max\{NF/u_s , F/d_{min}\}$.

Solution:

a. Consider a distribution scheme in which the server sends the file to each client, in

parallel, at a rate of a rate of u_s/N. Note that this rate is less than each of the client's download rate, since by assumption $u_s/N \leq d_{min}$. Thus each client can also receive at rate u_s/N. Since each client receives at rate u_s/N, the time for each client to receive the entire file is $F/(u_s/N) = NF/u_s$. Since all the clients receive the file in NF/u_s, the overall distribution time is also NF/u_s.

b. Consider a distribution scheme in which the server sends the file to each client, in parallel, at a rate of d_{min}. Note that the aggregate rate, $N d_{min}$, is less than the server's link rate u_s, since by assumption $u_s/N \geq d_{min}$. Since each client receives at rate d_{min}, the time for each client to receive the entire file is F/d_{min}. Since all the clients receive the file in this time, the overall distribution time is also F/d_{min}.

c. From (a) and (b) in the above, we have $D_{CS} = NF/u_s = max\ \{NF/u_s, F/d_{min}\}$ when $u_s/N \leq d_{min}$. (Equation 1) and $D_{CS} = F/d_{min} = max\ \{NF/u_s, F/d_{min}\}$ when $u_s/N \geq d_{min}$ (Equation 2). Therefore in general $D_{CS} = max\ \{NF/u_s, F/d_{min}\}$.