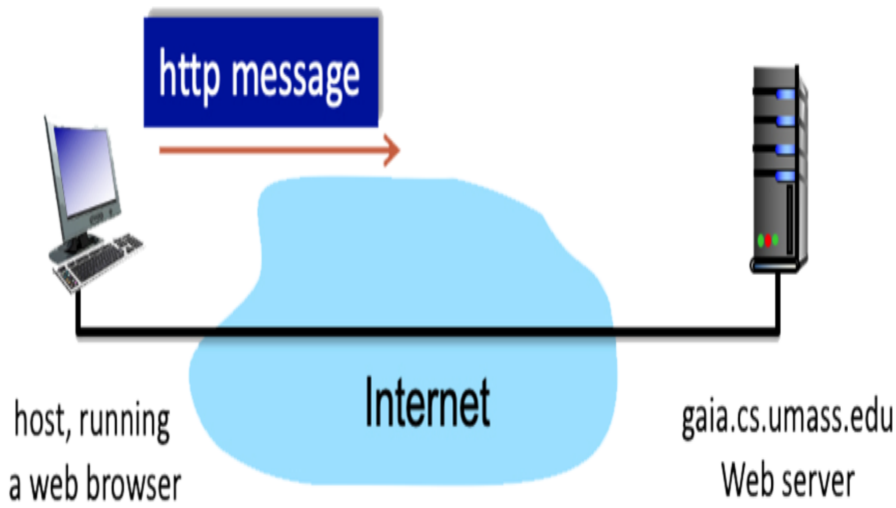
THE HTTP GET MESSAGE

Consider the figure below, where a client is sending an HTTP GET message to a web server, gaia.cs.umass.edu



Suppose the client-to-server HTTP GET message is the following:

GET /kurose_ross_sandbox/interactive/quotation2.htm HTTP/1.1

Host: gaia.cs.umass.edu

Accept: text/plain, text/html, image/png, image/gif, audio/mpeg, audio/vnf.wave, video/wmv, video/mp4,

Accept-Language: en-us, en-gb;q=0.3, en;q=0.7, fr, fr-ch, da, de, fi, ar

If-Modified-Since: Fri, 10 Sep 2021 08:33:40 -0700

User Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:10.0.2) Gecko/20100101 Firefox/10.0.2

QUESTION LIST

1. What is the name of the file that is being retrieved in this GET message?

2. What version of HTTP is the client running?

3. True or False: The client will accept html files

4. True or False: The client will accept jpeg images

5. What is the client's preferred version of English?

6. What is the client's least preferred version of English?

7. True or False: The client will accept the German language

8. True or False: The client already has a cached copy of the file

SOLUTION

1. The name of the file is quotation2.htm.

2. The client is running on HTTP/1.1

3. True. In the 'Accept' field the client includes 'text/html' files.

4. False. The client does NOT include 'image/jpeg' in its 'Accept' field.

5. The client's preferred version of English is American English. Any language without a defined q value has a default value of 1

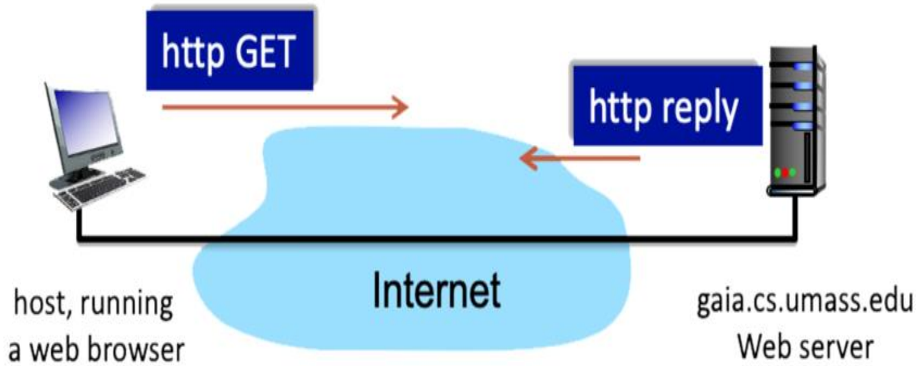
6. The client's least preferred version of English is British English because it has the lowest q value.

7. True. The client does include German in its 'Accepted-Language' field.

8. True. The client has a cached copy of the file that was updated on: Fri, 10 Sep 2021 08:33:40 -0700

THE HTTP RESPONSE MESSAGE

Consider the figure below, where the server is sending a HTTP RESPONSE message back the client.



Suppose the server-to-client HTTP RESPONSE message is the following:

HTTP/1.0 200 OK
Date: Fri, 10 Sep 2021 15:32:47 +0000
Server: Apache/2.2.3 (CentOS)
Last-Modified: Fri, 10 Sep 2021 16:04:07 +0000
ETag:17dc6-a5c-bf716880.
Content-Length: 855
Connection: Close
Content-type: image/html

QUESTION LIST

1. Is the response message using HTTP 1.0 or HTTP 1.1?

2. Was the server able to send the document successfully? Yes or No

3. How big is the document in bytes?

4. Is the connection persistent or nonpersistent?

5. What is the type of file being sent by the server in response?

6. What is the name of the server and its version? Write your answer as server/x.y.z

7. Will the ETag change if the resource content at this particular resource location changes? Yes or No

SOLUTION

1. The response is using HTTP/1.0

2. Since the response code is 200 OK, the document was received successfully.

3. The document is 855 bytes.

4. The connection is nonpersistent.

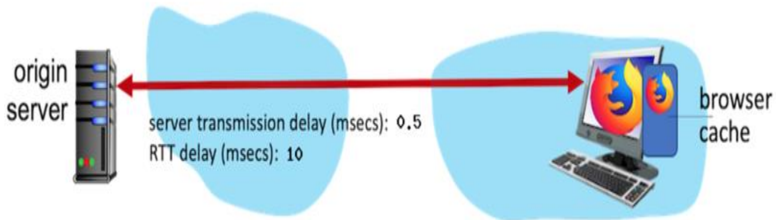
5. The file type the server is sending is image/html.

6. The name and version of the server is Apache/2.2.3

7. Yes. The Etag is a string that uniquely identifies a resource. If a resource is updated, the Etag will change.

BROWSER CACHING

Consider an HTTP server and client as shown in the figure below. Suppose that the RTT delay between the client and server is 10 msecs; the time a server needs to transmit an object into its outgoing link is 0.5 msecs; and any other HTTP message not containing an object has a negligible (zero) transmission time. Suppose the client again makes 70 requests, one after the other, waiting for a reply to a request before sending the next request.



Assume the client is using HTTP 1.1 and the IF-MODIFIED-SINCE header line. Assume 30% of the objects requested have NOT changed since the client downloaded them (before these 70 downloads are performed)

QUESTION LIST

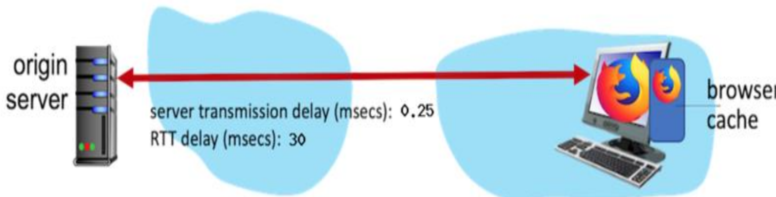
1. How much time elapses (in milliseconds) between the client transmitting the first request, and the completion of the last request?

SOLUTION

1. (RTT * NUM_PACKETS) + (NUM_PACKETS * (PERCENT__NOT_CACHED / 100) * TRANS_DELAY) = (10 * 70) + (70 * ((100-30) / 100) * 0.5) = 724.5 ms

BROWSER CACHING

Consider an HTTP server and client as shown in the figure below. Suppose that the RTT delay between the client and server is 30 msecs; the time a server needs to transmit an object into its outgoing link is 0.25 msecs; and any other HTTP message not containing an object has a negligible (zero) transmission time. Suppose the client again makes 80 requests, one after the other, waiting for a reply to a request before sending the next request.



Assume the client is using HTTP 1.1 and the IF-MODIFIED-SINCE header line. Assume 50% of the objects requested have NOT changed since the client downloaded them (before these 80 downloads are performed)

QUESTION LIST

1. How much time elapses (in milliseconds) between the client transmitting the first request, and the completion of the last request?

SOLUTION

1. (RTT * NUM_PACKETS) + (NUM_PACKETS * (PERCENT__NOT_CACHED / 100) * TRANS_DELAY) = (30 * 80) + (80 * ((100-50) / 100) * 0.25) = 2410 ms

ELECTRONIC MAIL AND SMTP

Look at the scenario below, where Alice sends an email to Bob.

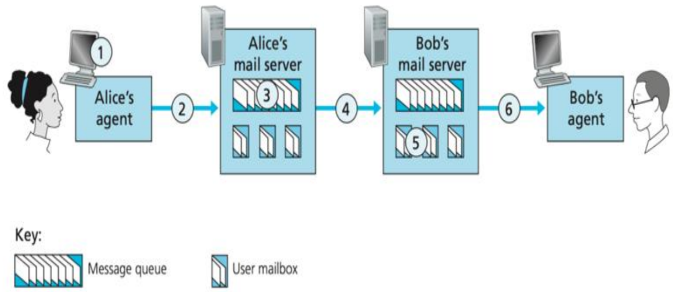


Figure 2.15 ♦ Alice sends a message to Bob

For the questions below, assume both Bob's and Alice's user agents use the POP3 protocol.

QUESTION LIST

1. At point 2 in the diagram, what protocol is being used?

2. At point 4 in the diagram, what protocol is being used?

3. At point 6 in the diagram, what protocol is being used?

4. Does SMTP use TCP or UDP?

5. Is SMTP a 'push' or 'pull' protocol?

6. Is POP3 a 'push' or 'pull' protocol?

7. What port does SMTP use?

8. What port does POP3 use?

SOLUTION

1. At point 2 in the diagram, the SMTP protocol is used.

2. At point 4 in the diagram, the SMTP protocol is used.

3. At point 6 in the diagram, the POP3 protocol is used.

4. SMTP uses the TCP protocol.

5. SMTP is a 'push' protocol

6. POP3 is a 'pull' protocol

7. SMTP uses port 25

8. POP3 uses port 110

DNS - BASICS

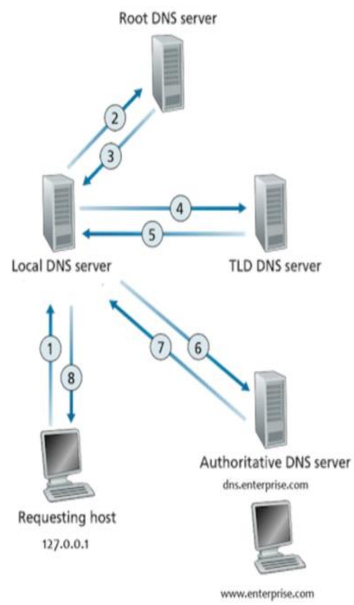
Imagine that you are trying to visit www.enterprise.com, but you don't remember the IP address the web-server is running on.

Assume the following records are on the TLD DNS server:

- (www.enterprise.com, dns.enterprise.com, NS)
- (dns.enterprise.com, 146.54.126.238, A)

Assume the following records are on the enterprise.com DNS server:

- (www.enterprise.com, east4.enterprise.com, CNAME)
- (east4.enterprise.com, 142.81.17.206, A)
- (www.enterprise.com, mail.enterprise.com, MX)
- (mail.enterprise.com, 247.29.244.211, A)



Assume your local DNS server only has the TLD DNS server cached.

QUESTION LIST

1. What transport protocol(s) does DNS use: TCP, UDP, or Both?

2. What well-known port does DNS use?

3. How many types of Resource Records (RR) are there?

4. Can you send multiple DNS questions and get multiple RR answers in one message? Answer with Yes or No

5. To which DNS server does a host send their requests to? Answer with the full name

6. Which type of DNS server holds a company's DNS records? Answer with the full name

7. In the example given in the problem, what is the address of the DNS server for enterprise.com?

8. When you make the request for www.enterprise.com, your local DNS requests the IP on your behalf. When it contacts the TLD server, how many answers (RR) are returned?

9. In the previous question, there were two responses, one was a NS record and the other an A record. What was the content of the A record? Answer with the format: "name, value"

10. Assume that the enterprise.com website is actually hosted on east4.enterprise.com, what type of record is needed for this?

11. Now imagine we are trying to send an email to admin@enterprise.com, and their mail server has the address mail.enterprise.com. What type of record will we receive?

12. In that MX record, what are the contents? Answer with the format: "name, value"

13. Does your local DNS server take advantage of caching similar to web requests? Answer with Yes or No

SOLUTION

1. DNS generally uses UDP, but in some cases (such as zone transfer) it will use TCP, so the answer is: Both.

2. DNS uses well-known port 53.

3. There are 4 types of RR's: A, CNAME, NS, and MX.

4. Yes, there can be multiple 'questions' and 'answers' in a single DNS request.

5. The host first contacts the Local DNS server, which acts on behalf of the host.

6. The company's Authoritative DNS server is where their RR are stored.

7. The Authoritative DNS server for www.enterprise.com is dns.enterprise.com

8. There are 2 records returned; a NS record, and an A record for the DNS server.

9. The A record has contents: (dns.enterprise.com, 146.54.126.238)

10. In this case, a CNAME record is needed.

11. An MX record will be returned.

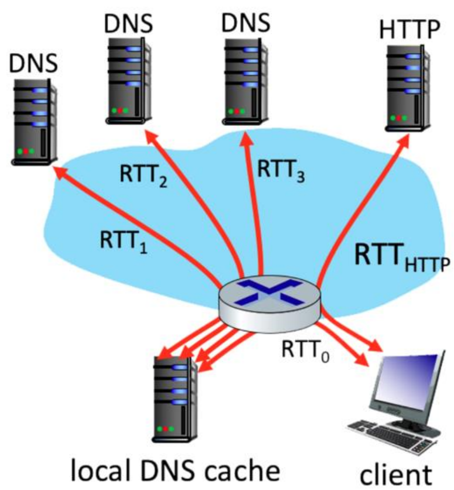
12. The MX record has contents: (mail.enterprise.com, 247.29.244.211)

13. Yes, DNS servers (especially your Local DNS server) cache records for faster retrieval.

DNS AND HTTP DELAYS

Before doing this question, 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 the DNS (in particular the text surrounding Figure 2.19).

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 four 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 RTT_0 = 4 msecs. The second, third and fourth DNS servers contacte have RTTs of 25, 15, and 13 msecs, respectively. 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 RTT_{HTTP} = 26 mse



QUESTION LIST

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

2. Now suppose the HTML object references 5 very small objects on the same server. Neglecting transmission times, how much time (in msec) elapses from when the client clicks on the link until the base object and all 5 additional objects are received from web server at the client, assuming non-persistent HTTP and no parallel TCP connections?

3. Suppose the HTML object references 5 very small objects on the same server, but assume that the client is configured to support a maximum of 5 parallel TCP connections, with non-persistent HTTP.

SOLUTION

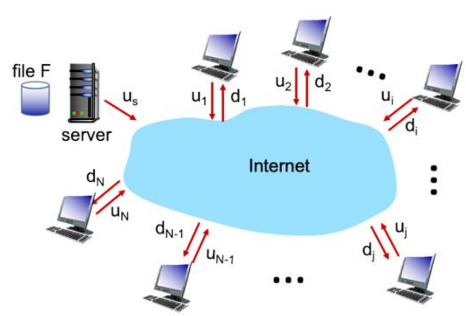
1. The time from when the Web request is made in the browser until the page is displayed in the browser is: $RTT_0 + RTT_1 + RTT_2 + RTT_3 + 2{*}RTT_{HTTP}$ = 4 + 25 + 15 + 13 + 2*26 = 109 msecs. Note that 2 RTT_{HTTP} are needed to fetch the HTML object - one RTT_{HTTP} to establish the TCP connection, and then one RTT_{HTTP} to perform the HTTP GET/response over that TCP connection.

2. The time from when the Web request is made in the browser until the page is displayed in the browser is: $RTT_0 + RTT_1 + RTT_2 + RTT_3 + 2{*}RTT_{HTTP} + 2{*}5{*}RTT_{HTTP}$ = 4 + 25 + 15 + 13 + 2*26 + 2*5*26 = 369 msecs. Note that two RTT_{HTTP} delays are needed to fetch the base HTML object - one RTT_{HTTP} to establish the TCP connection, and one RTT_{HTTP} to send the HTTP request, and receive the HTTP reply. Then, serially, for *each* of the 5 embedded objects, a delay of $2{*}RTT_{HTTP}$ is needed - one RTT_{HTTP} to establish the TCP connection and then one RTT_{HTTP} to perform the HTTP GET/response over that TCP connection.

3. Since there are only 5 objects, there's a delay of 57 msec for the DNS query, two RTT_{HTTP} for the base page, and $2{*}RTT_{HTTP}$ for the objects since the requests for these can be run in parallel. The total is 57 + 52 + 52 = 161 msec. As in 2 above, 2 RTT_{HTTP} are needed to fetch the base HTML object - one RTT_{HTTP} to establish the TCP connection, and one RTT_{HTTP} to send the HTTP request and receive the HTTP reply containing the base HTML object. Once the base object is received at the client, the 5 HTTP GETS for the embedded objects can proceed in parallel. Each (in parallel) requires two RTT_{HTTP} delays - one RTT_{HTTP} to set up the TCP connection, and one RTT_{HTTP} to perform the HTTP GET/response for an embedded object.

A COMPARISON OF CLIENT-SERVER AND P2P FILE DISTRIBUTION DELAYS

In this problem, you'll compare the time needed to distribute a file that is initially located at a server to clients via either client-server download or peer-to-peer download. Before beginning, you might want to first review Section 2.5 and the discussion surrounding Figure 2.22 in the text.



The problem is to distribute a file of size F = 5 Gbits to each of these 6 peers. Suppose the server has an upload rate of u = 85 Mbps.

The 6 peers have upload rates of: u_1 = 17 Mbps, u_2 = 16 Mbps, u_3 = 15 Mbps, u_4 = 16 Mbps, u_5 = 29 Mbps, and u_6 = 19 Mbps

The 6 peers have download rates of: d_1 = 34 Mbps, d_2 = 21 Mbps, d_3 = 12 Mbps, d_4 = 17 Mbps, d_5 = 12 Mbps, and d_6 = 40 Mbps

QUESTION LIST

1. What is the minimum time needed to distribute this file from the central server to the 6 peers using the client-server model?

2. For the previous question, what is the root cause of this specific minimum time? Answer as 's' or 'ci' where 'i' is the client's number

3. What is the minimum time needed to distribute this file using peer-to-peer download?

4. For question 3, what is the root case of this specific minimum time: the server (s), client (c), or the combined upload of the clients and the server (cu)

SOLUTION

1. The minimum time needed to distribute the file = max of: $N*F / U_S$ and F / d_{min} = 416.67 seconds.

2. The root cause of the minimum time was c3.

3. The minimum time needed to distribute the file = max of: F / U_S, F / d_{min}, and $N * F$ / sum of u_i for all i + u_S = 416.67 seconds.

4. The root cause of the minimum time was c.