Kyle Romero
204 747 283

Problem 1

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 or RTT_1, RTT_2, ..., RTT_n. Further suppose that the Web page associated with the link has a small amount of HTML text. Let RTT_0 denote the RTT between the local host and the server containing the HTML file. Assume zero transmission time. Suppose the HTML file references 11 very small objects on the same server. Neglect transmission times, how much time elapses from when the client clicks on the link until the client receives all objects with:

(a) Non-persistent HTTP with no parallel TCP connections?

(b) Non-persistent HTTP with the browser configured for 5 parallel connections?

(c) Persistent HTTP with no parallel TCP connections?

(d) Persistent HTTP with the browser configured for arbitrarily many parallel connections?

Write your solution to Problem 1 in this box

a) Elapsed time to get IP address: $RTT_1 + \dots + RTT_n$
To bootstrap HTTP connection and get HTML file: $2RTT_0$
To bootstrap HTTP connection and get 11 objects: $2RTT_0 \cdot 11$
In total, $RTT_1 + \dots + RTT_n + 2RTT_0 + 24RTT_0 = \boxed{RTT_1 + \dots + RTT_n + 24RTT_0}$

b) Again for IP address: $RTT_1 + \dots + RTT_n$
To bootstrap the HTTP connection and get HTML file: $2RTT_0$
To bootstrap the HTTP connection and get objects (w/ 5 parallel connections):
$\lceil \frac{11}{5} \rceil (2RTT_0) = 6RTT_0$
Elapsed time $= \boxed{RTT_1 + \dots + RTT_n + 8RTT_0}$

c) To get IP address: $RTT_1 + \dots + RTT_n$
To bootstrap the HTTP connection and get HTML file: $2RTT_0$
To get all 11 objects: $RTT_0 \cdot 11$
In total, $\boxed{\text{Elapsed time} = RTT_1 + \dots + RTT_n + 13RTT_0}$

d) To get IP address: $RTT_1 + \dots + RTT_n$
To bootstrap the HTTP connection and get HTML file: $2RTT_0$
To get all 11 objects: RTT_0
In total, $\boxed{\text{Elapsed time} = RTT_1 + \dots + RTT_n + 3RTT_0}$

Problem 2

How does the web server (e.g., eBay) identify users when you do the Internet shopping? Briefly explain how it works.

Write your solution to Problem 2 in this box

When a user enters an online shopping site a cookie is generated which is used to keep track of users.

When a user contacts an online shopping site for the first time, a unique ID number is generated and stored in a back-end database. When the HTTP response message is sent back to a user's browser, this ID number, or cookie, is contained in the HTTP response header.

Now, when this user browses the site the cookie is sent from client to server with each of their HTTP requests.

Online shopping sites can use this cookie for several purposes. Including, providing persistent sessions, shopping carts, or to provide shopping recommendations based on a specific user's search history.

Problem 3

A Web browser running on the client host is requesting a webpage from the server. We make the following assumptions:

- TCP window is large once the TCP handshake is complete (i.e. ignore flow control). TCP header size is h bits, and the maximum payload size is p bits.
- The bandwidth is b bps, and the propagation delay is d seconds.
- Ignore DNS related delays, and ignore the payload in three-way handshake packets, ACK packets, and HTTP request packets. In other words, those packets consist of header only.
- The client requests a webpage consisting of an HTML file that indexes 5 binary files on the same server. Each of the file is $2p$ bits long. In other words, each of the file can be sent in exactly 2 TCP packets. Piggybacking is used whenever possible.
- Each HTTP request is sent in one TCP packet.

Please answer the following questions:

(a) Suppose pipelining of HTTP requests is allowed and no parallel TCP connections are used. Calculate the minimal time it takes the browser to receive all the files.

(b) Suppose the non-persistent, non-pipelining mode with parallel TCP connections is used, repeat the calculation.

(c) Which mode gives the smaller latency? Briefly justify your answer.

$S = \frac{h}{b} + d$ $\quad\quad R = 2\left(\frac{p+h}{b} + d\right)$

① for bootstrap + initial file

$3\left(\frac{h}{b} + d\right) + 2\left(\frac{p+h}{b} + d\right)$

② for 5 files

$\frac{h}{b} + d + 10\left(\frac{p+h}{b} + d\right)$

$\frac{h}{b} + d + \frac{10p + 10h}{b} + 10d$

$11d + \frac{10p + 11h}{b} + \frac{3h}{b} + 3d + \frac{2p + 2h}{b} + 2d$

$= 16d + \frac{12p + 16h}{b}$

Write your solution to Problem 3 in this box

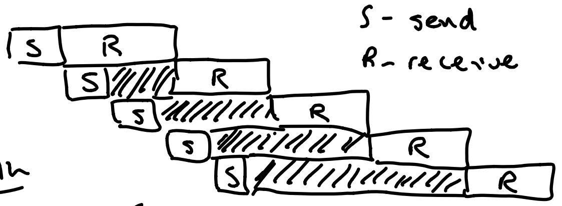
a) This method can be split into two parts. ① bootstrapping the TCP connection / sending the initial HTML file, and ② sending the 5 binary files

① $\underset{\substack{\uparrow \\ \text{3-way handshake}}}{3}\left(\frac{h}{b}+d\right) + \underset{\substack{\nwarrow \\ \text{2 packets}}}{2}\left(\frac{p+h}{b}+d\right)$

② $\frac{h}{b}+d + (5)(2)\left(\frac{p+h}{b}+d\right)$

$= \frac{h}{b} + 11d + \frac{10p+10h}{b} = 11d + \frac{10p+11h}{b}$

Timing of pipelined requests



S - send
R - receive

S + 5R = total delay for ②

Total time: ① + ② $= 3\left(\frac{h}{b}+d\right) + 2\left(\frac{p+h}{b}\right)+d + 11d + \frac{10p+11h}{b}$

$= \boxed{16d + \frac{12p+16h}{b}}$

b) Again two sections, ① bootstrapping + sending HTML file and ② bootstrapping and sending 5 binary files on 5 parallel TCP connections:

① $3\left(\frac{h}{b}+d\right) + 2\left(\frac{p+h}{b}+d\right)$

② $3\left(\frac{h}{b}+d\right) + 2\left(\frac{p+h}{b}+d\right)$

Total time $= \frac{6h}{b} + 6d + \frac{4p+4h}{6} + 4d$

$\boxed{= 10d + \frac{4p+10h}{b}}$

c) The $\boxed{\text{parallel connections are faster}}$,

$$\frac{10d + \frac{4p+10h}{b}}{16d + \frac{12p+16h}{b}} = \frac{\frac{10db+4p+10h}{b}}{\frac{16db+12p+16h}{b}} = \frac{10db+4p+10h}{16db+12p+16h}$$

Assuming d, b, p, and h are nonegative, this # is less than 1, meaning delay is less for parallel

Problem 4

How does SMTP marks the end of a message body? How about HTTP? Can HTTP use the same method as SMTP to mark the end of the message body?

Write your solution to Problem 4 in this box

SMTP uses a line consisting of a single period to mark the end of the message body.

HTTP uses a content length field in the header to determine when the end of a message is reached.

HTTP cannot use the same method as SMTP to mark the end of the message body. This is because HTTP can contain any arbitrary sequence of bits for any type of file. SMTP must contain only 7-bit ASCII, therefore we can assign a specific ASCII sequence that signifies the end of a message body.

Problem 5

Suppose your department has a local DNS server for all computers in the department.

(a) Suppose you are an ordinary user (i.e., not a network/system administrator). Can you determine if an external Web site was likely accessed from a computer in your department a couple of seconds ago? Explain.

(b) Now suppose you are a system administrator and can access the caches in the local DNS servers of your department. Can you propose a way to roughly determine the Web servers (outside your department) that are most popular among the users in your department? Explain.

Write your solution to Problem 5 in this box

a) Yes, you should be able to tell if an external web site was likely accessed from a computer in your department a couple seconds ago.

If the page was accessed a few seconds ago, then there should be a very small delay, basically 0msec, to look up the IP address of a given domain name. Otherwise, the delay should be large because it did not find the domain name to IP address translation on the local DNS server. This kind of query can typically be timed using the 'dig` command.

b) In order to determine the most popular external web servers amongst users in my department, I could periodically note the contents of my DNS caches on the local DNS server.

The most popular sites will be present most often because they receive the most requests and are cached most of the time.