Problem 1

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 1 in this box

If user accesses from same device,
the first time they visit the site, inital HTTP request creates unique ID and entry of that ID in backend DB for the web server.

This unique ID is the cookie saved by browser. Since eBay has ID in their DB, when you access site again, will access your cookie file and give the usual HTTP response msg.

Cookies help website learn about you. Can access for payment, shopping cart, reccamandations...

Problem 2

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 (round-trip time) of 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 ten very small objects on the same server. 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 2 in this box

a). $RTT_1 + \dots + RTT_n + 2RTT_0 + 2 \cdot 10\, RTT_0$

$22\, RTT_0 + RTT_1 \dots + RTT_N$

b). $RTT_1 + \dots RTT_N + 2RTT_0 + 2 \cdot 2\, RTT_0$

$6RTT_0 + RTT_1 \dots + RTT_N$

c). $RTT_1 + RTT_N + 2RTT_0 + 10RTT_0$

$12\cancel{2}\, RTT_0 + RTT_1 + \dots RTT_N$

d). $RTT_1 + \dots RTT_n + 2RTT_0 + 1\, RTT_0$

$3RTT_0 + RTT_1 + \dots RTT_N$.

CS 118 Spring 2019 : Homework 2

Problem 3

Suppose Bob joins a BitTorrent torrent, but he does not want to upload any data to any other peers (so called free-riding).

(a) Bob claims that he can receive a complete copy of the file that is shared by the swarm. Is Bob's claim possible? Why or why not?

(b) Bob further claims that he can further make his "free-riding" more efficient by using a collection of multiple computers (with distinct IP addresses) in the computer lab in his department. How can he do that?

Write your solution to Problem 3 in this box

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 single line contained by period to mark end.

HTTP uses content-length header to mark end of msg.

They cannot use same method as SMTP is in 7-bit ASCII and HTTP is binary.

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) - No cannot because ordinary user does not have Permissions.
- Local DNS Server special because it acts as a proxy to send queries to DNS servers above it in hierarchy.
- A host computer sends query to local DNS which sends to main DNS server.

b) - The local DNS server saves DNS replies.
- The local saves the response locally then sends the query back to host. If this cache is not cleared, taking periodical snapshots of DNS caches in the load servers. The webserver that appears most frequently is the most popular.