Problem 1

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

- eBay Web server uses cookie to identify users and their transaction histories.
- eBay Web server creates an entry in its back-end database that is indexed by a unique identification number when the user first browses the eBay website.
- This identification number is sent back in the HTTP response Set-cookie: header, and it will be saved to user's cookie file.
- Each time the user requests an eBay web page in the future, a cookie header line including the identification number is sent along the HTTP request.

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?

DNS resolve time $t_{DNS} = \mathrm{RTT}_1 + \mathrm{RTT}_2 + \cdots + \mathrm{RTT}_\mathrm{n}$. Once the DNS is resolved, setting up the TCP connection needs RTT_0 time and another RTT_0 to request and receive the small object. The total response time is $2 \cdot \mathrm{RTT}_0$.

Since the objects are very small, we assume that the content retrieve time is negligible. Therefore:

(a) Non-persistent HTTP with no parallel TCP connection $t_{HTTP} = 10 \cdot 2\mathrm{RTT}_0$.

Total time $t = \mathrm{RTT}_1 + \mathrm{RTT}_2 + \cdots + \mathrm{RTT}_\mathrm{n} + 22\mathrm{RTT}_0$.

(b) With 5 parallel connections, 10 small objects need 2 batches to transmit, and each batch needs 2 RTT_0. Total time $t = \mathrm{RTT}_1 + \mathrm{RTT}_2 + \cdots + \mathrm{RTT}_\mathrm{n} + 6\mathrm{RTT}_0$.

(c) With persistent HTTP:

In non-pipelining mode, there will be 1 RTT_0 per object. Thus, 10 small objects need 10 RTT_0 to transmit, the total time $t = \mathrm{RTT}_1 + \mathrm{RTT}_2 + \cdots + \mathrm{RTT}_\mathrm{n} + 12\mathrm{RTT}_0$.

(d) In pipelining mode, 1 RTT_0 is enough for all the reference objects in one round. Total time $t = \mathrm{RTT}_1 + \mathrm{RTT}_2 + \cdots + \mathrm{RTT}_\mathrm{n} + 3\mathrm{RTT}_0$.

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?

(a) Yes. His first claim is possible, as long as there are enough peers staying in the swarm for a long enough time. Bob can always receive data through optimistic unchoking by other peers.

(b) His second claim is also true. He can run a client on each host, let each client "free-ride," and combine the collected chunks from the different hosts into a single file. He can even write a small scheduling program to make the different hosts ask for different chunks of the file. This is actually a kind of Sybil attack in P2P networks.

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?

SMTP uses a line containing only a period to mark the end of a message body.
HTTP uses "Content-Length header field" to indicate the length of a message body.
No, HTTP cannot use the method used by SMTP, because HTTP message could be binary data, whereas in SMTP, the message body must be in 7-bit ASCII format.

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.

(a) Yes, we can use dig to query that Web site in the local DNS server. For example, dig www.cnn.com will return the query time for finding cnn.com. If www.cnn.com was just accessed a couple of seconds ago, an entry for www.cnn.com is cached in the local DNS cache, so the query time is 0 msec. Otherwise, the query time is large.

(b) We can examine the DNS cache periodically and take snapshots. By calculating the frequency of web server resolve messages, we can know that whichever appears most frequently in the DNS caches is the most popular server, because if more users are interested in a Web server, then DNS requests for that server are more frequently sent by users and will appear more frequently in the DNS cache.