Problem 1

How does the web server (e.g., Amazon) identify users when you do the Internet shopping? Please briefly explain how it works with HTTP protocol step by step.

- Amazon Web server uses cookie to identify users and their transaction histories.
- Amazon Web server creates an entry in its back-end database that is indexed by a unique identification number when the user first browses the Amazon 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 Amazon 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 from a web server S. The web page is a HTML file and the HTML file further contains references to 9 small JPEG files on the same server S. However, S's IP address is not cached in your local host, so a DNS lookup is required. Suppose that n DNS servers are visited by your browser before you get S's IP address. Let RTT_1, RTT_2, ..., RTT_n denote the RTTs (round-trip time) of visiting each of the n DNS server and RTT_0 denote the RTT between the local host and S. If we ignore file transmission time for DNS responses, HTML file, and JPEG files, 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) 9 Objects with non-persistent HTTP with no parallel TCP connection $9 \cdot 2\mathrm{RTT}_0$.

Total time $t = \mathrm{RTT}_1 + \mathrm{RTT}_2 + \cdots + \mathrm{RTT}_\mathrm{n} + 20\mathrm{RTT}_0$.

(b) With 5 parallel connections, 9 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 without pipelining:

In non-pipelining mode, there will be 1 RTT_0 per object. Thus, 9 small objects need 9 RTT_0 to transmit.

Total time $t = \mathrm{RTT}_1 + \mathrm{RTT}_2 + \cdots + \mathrm{RTT}_\mathrm{n} + 11\mathrm{RTT}_0$.

Considering persistent HTTP with pipelining, after TCP connection setup RTT and HTML fetching RTT, 1 more RTT_0 is enough for all the reference objects in one round.

Total time $t = \mathrm{RTT}_1 + \mathrm{RTT}_2 + \cdots + \mathrm{RTT}_\mathrm{n} + 3\mathrm{RTT}_0$.

Because of the ambiguity in the question, both 11 RTT_0 and 3 RTT_0 are acceptable.

(d) Considering pipelining mode with persistent connection, after TCP connection setup RTT and HTML fetching RTT, 1 more RTT_0 is enough for all the reference objects in one round.

Total time $t = \mathrm{RTT}_1 + \mathrm{RTT}_2 + \cdots + \mathrm{RTT}_\mathrm{n} + 3\mathrm{RTT}_0$.

Considering multiple parallel connections, after TCP connection setup RTT and HTML fetching RTT, 9 connections can be established to fetch 9 objects with 2 RTTs.

Total time $t = \mathrm{RTT}_1 + \mathrm{RTT}_2 + \cdots + \mathrm{RTT}_\mathrm{n} + 4\mathrm{RTT}_0$.

Because of the ambiguity in the question, both 3 RTT_0 and 4 RTT_0 are acceptable.

Problem 3

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 4

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.

Problem 5

Consider distributing a file of $F = 15$ Gbits to N peers. The server has an upload rate of $u_s = 30$ Mbps, and each peer has a download rate of $d_1 = 2$ Mbps and an upload rate of u. For N = 10 and 100, and $u =$ 300 Kbps and 2 Mbps, prepare a chart giving the minimum distribution time for each of the combinations of N and u for both client-server distribution and P2P distribution (i.e., there are 8 distribution time values in total). In this problem, we assume $1K = 1 \times 10^3$, $1M = 1 \times 10^6$, $1G = 1 \times 10^9$.

Please fulfill your answers into the following two charts and briefly explain how you get them.

Client-Server distribution

u\N	10	100
300Kbps	F/d_{min} =7500s	$100 \times F/u_s$ =50000s
2Mbps	F/d_{min} =7500s	$100 \times F/u_s$ =50000s

P2P distribution

u\N	10	100
300Kbps	F/d_{min} =7500s	$NF/(u_s + \sum_{i=1}^{N} u_i)$ =25000s
2Mbps	F/d_{min} =7500s	F/d_{min} =7500s

For calculating the minimum distribution time for client-server distribution, we use the following formula:
$D_{cs} = \max\{NF/u_s, F/d_{min}\}$

For P2P, we use the formula:
$D_{cs} = \max\{F/u_s, F/d_{min}, NF/(u_s + \sum_{i=1}^{N} u_i)\}$

In this problem, we have
$F = 15 \times 1000Mb$
$u_s = 30Mbps$
$d_{min} = 2Mbps$