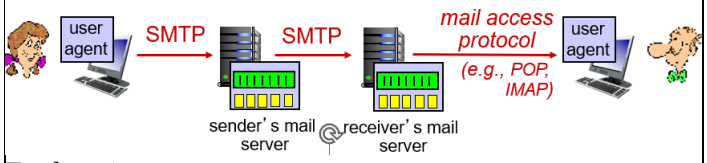
Scheme of Evaluation Internal Assessment Test 1 – November 2022

Sub:	Computer Networks and Security							Code:	18CS52
Date:	04/11/2022	Duration:	90mins	Max Marks:	50	Sem:	V	Branch:	ISE

Note: **Answer Any five full questions.**

Question #		Description	Marks Distribution		Max Marks
1	a.	Define computer Network. Illustrate the two different architectures available in network and explain? a) Computer networking refers to interconnected computing devices that can exchange data and share resources with each other ❖ client-server ❖ peer-to-peer (P2P) Diagram with explanation	5M 5M 1M 4M	10M	10M
	b)	Illustrate how DNS server will interact to various DNS servers hierarchically b) Iterated Query explanation with diagram Recursive Query explanation with diagram	5M		
2		Illustrate about stateless protocol request and response format. Request message header format with explanation Response message header format with explanation	5M 5M	10M	10M
3		Describe the P2P file distribution. Diagram with Explanation	10 M	10M	10M

4	a)	Suppose Alice, with a Web based E-mail account (such as hotmail or Gmail), and sends a message to Bob, who accesses his mail from his mail server using POP3. Discuss how the message gets from Alice's host to Bob's Host. (Be sure to list the series of application layers protocols that are used to move the message between the two hosts.)  Explanation	5M	5M	10M
	b)	Compare HTTP and SMTP 5 differences each	5M	5M	
5	a)	Illustrate the connectionless protocol segment header with diagram and error checking method with an example. UDP header Explanation Checksum example	2M 3M 3M	8M	10M
	b)	Define Multiplexing and Demultiplexing The job of delivering the data in a transport-layer segment to the correct socket is called demultiplexing job of gathering data chunks at the source host from different sockets, encapsulating each data chunk with header information to create segments, and passing the segments to the network layer is called multiplexing	2M	2M	
6		Build a Reliable Data Transfer Protocol for satisfying following conditions • Stop and wait protocol. • Packet contains sequence number, data and checksum. • Need to use both ACK and NACK. • Need to handling duplicates Transition diagram with explanation	2M 2M 3M 3M	10M	10M

6	a)	What are the steps involved between client and server in order to fetch 10 JPEG images which are residing in the same server by using non-persistent HTTP connection, the URL for the base HTML file is http://www.xyz.edu/department/base.index **Explanation** **Steps**	1M 5M	6M	10M
6	b)	Discuss FTP and its commands and replies. **Commands** **Replies**	3M 2M	4M	

Scheme Of Evaluation Internal Assessment Test 1 – NOV 2022

Sub:	Computer Networks and Security							Code:	18CS52
Date:	04/11/2022	Duration:	90mins	Max Marks:	50	Sem:	V	Branch:	ISE

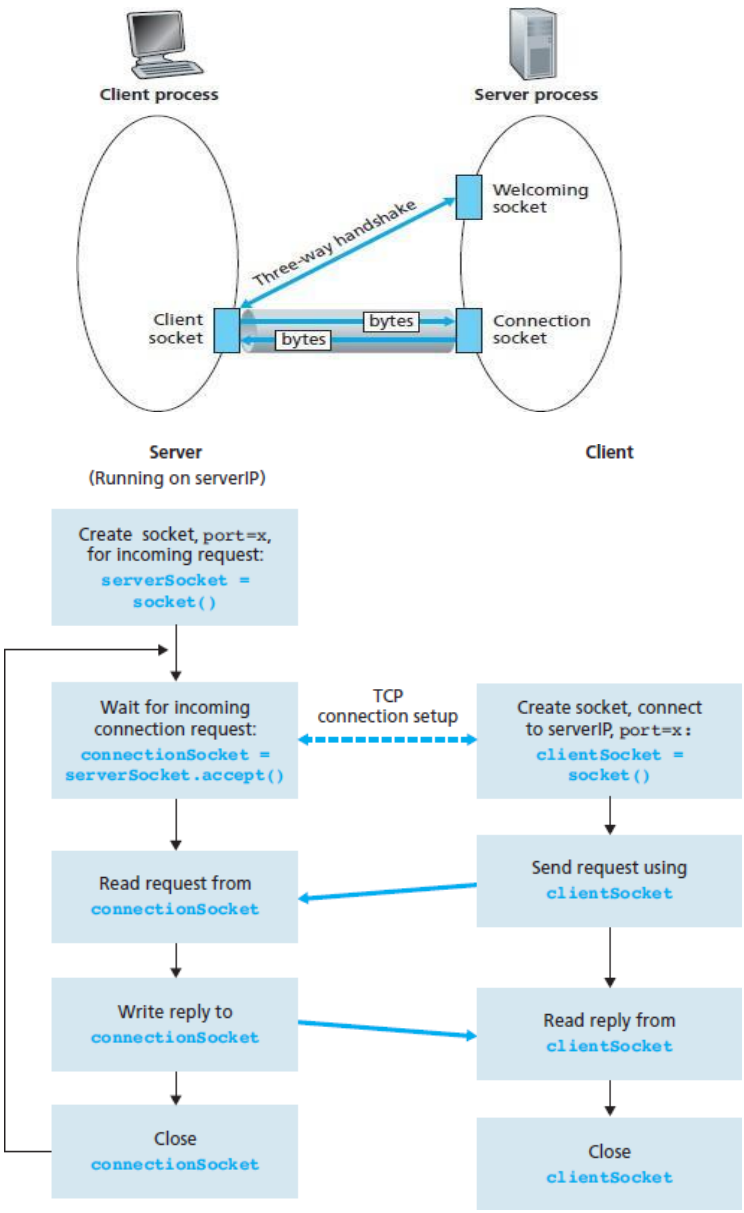
Note: **Answer Any full five questions**

Q. 1 Design network application using socket programming for satisfying following conditions?

- **reliable protocol.**
- **transmission of data in the form of stream of bytes.**

each segment contains Source IP, Destination IP, Source Port and Destination Port.

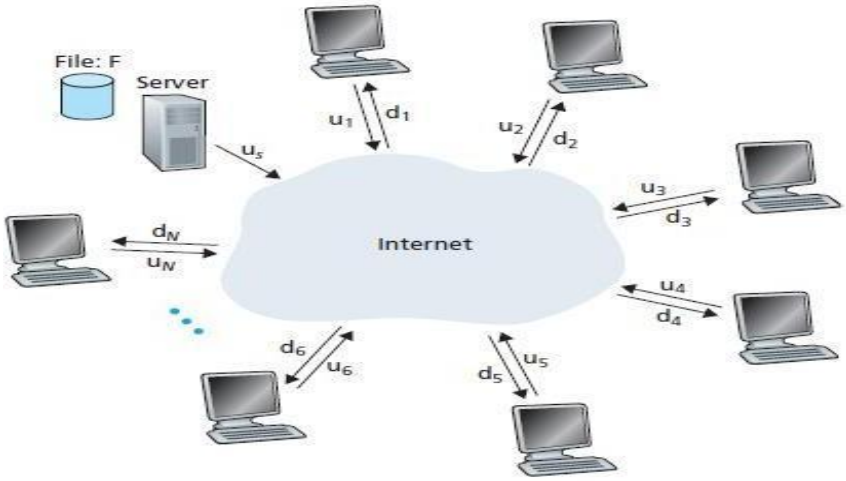
- Unlike UDP, TCP is a connection-oriented protocol. This means that before the client and server can start to send data to each other, they first need to handshake and establish a TCP connection.
- One end of the TCP connection is attached to the client socket and the other end is attached to a server socket.
- When creating the TCP connection, we associate with it the client socket address (IP address and port number) and the server socket address (IP address and port number). With the TCP connection established, when one side wants to send data to the other side, it just drops the data into the TCP connection via its socket. This is different from UDP, for which the server must attach a destination address to the packet before dropping it into the socket.
- During the three-way handshake, the client process knocks on the welcoming door of the server process. When the server "hears" the knocking, it creates a new door— more precisely, a new socket that is dedicated to that particular client.



Q.2 Describe the P2P file distribution.

- In P2P file distribution, each peer can redistribute any portion of the file it has received to any other peers, thereby assisting the server in the distribution process.
- The most popular P2P file distribution protocol is BitTorrent.
- As shown in below Figure the server and the peers are connected to the Internet with access links. Denote the upload rate of the server's access link by u_s, the upload rate of the ith peer's access link by u_i, and the download rate of the ith peer's access link by d_i. Also denote the size of the file to be distributed (in bits) by F and the number of peers that want to obtain a copy of the file by N.

- The **distribution time** is the time it takes to get a copy of the file to all N peers.
-



In the client-server architecture, none of the peers aids in distributing the file. We make the following observations:

- The server must transmit one copy of the file to each of the N peers. Thus the server must transmit NF bits. Since the server's upload rate is us, the time to distribute the file must be at least NF/us.
- Let dmin denote the download rate of the peer with the lowest download rate, that is, dmin = min{d1,dp,...,dN}. The peer with the lowest download rate cannot obtain all F bits of the file in less than F/dmin seconds. Thus the minimum distribution time is at least F/dmin.

Putting these two observations together, we obtain

$$D_{cs} \geq \max\left\{\frac{NF}{u_s}, \frac{F}{d_{min}}\right\}.$$

In the P2P architecture we make the following observations:

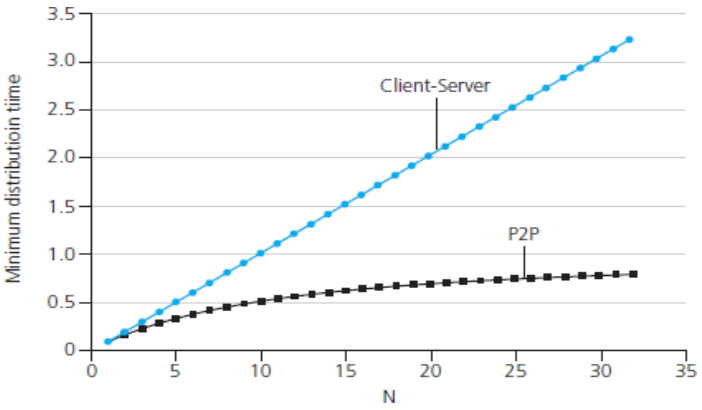
- At the beginning of the distribution, only the server has the file. To get this file into the community of peers, the server must send each bit of the file at least once into its access link. Thus, the minimum distribution time is at least F/us.
- As with the client-server architecture, the peer with the lowest download rate cannot obtain all F bits of the file in less than F/dmin seconds. Thus the minimum distribution time is at least F/dmin.

- Finally, observe that the total upload capacity of the system as a whole is equal to the upload rate of the server plus the upload rates of each of the individual peers, that is, utotal = us + u1 + … + uN. The system must deliver (upload) F bits to each of the N peers, thus delivering a total of NF bits. This cannot be done at a rate faster than utotal. Thus, the minimum distribution time is also at least NF/(us + u1 + … + uN).

Putting these three observations together, we obtain the minimum distribution time for P2P, denoted by D_{P2P}.

$$D_{P2P} \geq \max\left\{\frac{F}{u_s}, \frac{F}{d_{min}}, \frac{NF}{u_s + \sum_{i=1}^{N} u_i}\right\}$$

Below Figure compares the minimum distribution time for the client-server and P2P architectures assuming that all peers have the same upload rate u.



Q.3 a) Illustrate how user1 can send mail to user2 and how user2 receives the mail by using SMTP.

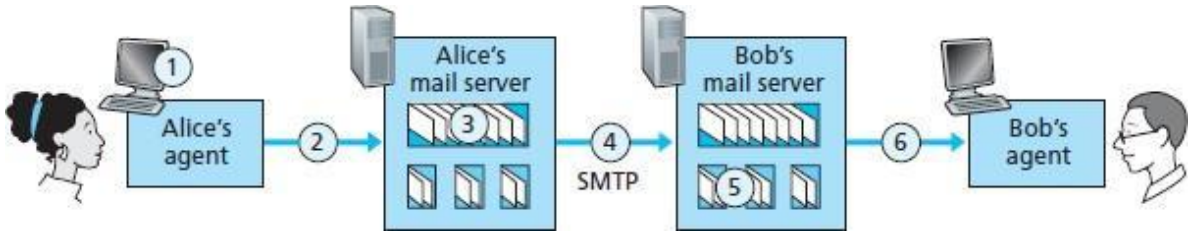
SMTP transfers messages from senders' mail servers to the recipients' mail servers. It restricts the body (not just the headers) of all mail messages to simple 7-bit ASCII.

Suppose Alice wants to send Bob a simple ASCII message.

1. Alice invokes her user agent for e-mail, provides Bob's e-mail address (for example, bob@someschool.edu), composes a message, and instructs the user agent to send the message.
2. Alice's user agent sends the message to her mail server, where it is placed in a message queue.
3. The client side of SMTP, running on Alice's mail server, sees the message in the message queue. It opens a TCP connection to an SMTP server, running on Bob's mail server.
4. After some initial SMTP handshaking, the SMTP client sends Alice's message into the TCP connection.

5. At Bob's mail server, the server side of SMTP receives the message. Bob's mail server then places the message in Bob's mailbox.

6. Bob invokes his user agent to read the message at his convenience.



An example transcript of messages exchanged between an SMTP client (C) and an SMTP server (S).

S: 220
hamburger.edu C:
HELO crepes.fr
S: 250 Hello crepes.fr, pleased to meet
you C: MAIL FROM:
<alice@crepes.fr>
S: 250 alice@crepes.fr ... Sender
ok C: RCPT TO:
<bob@hamburger.edu>
S: 250 bob@hamburger.edu ... Recipient
ok C: DATA
S: 354 Enter mail, end with "." on a line by
itself C: Do you like ketchup?
C: How about
pickles? C: .

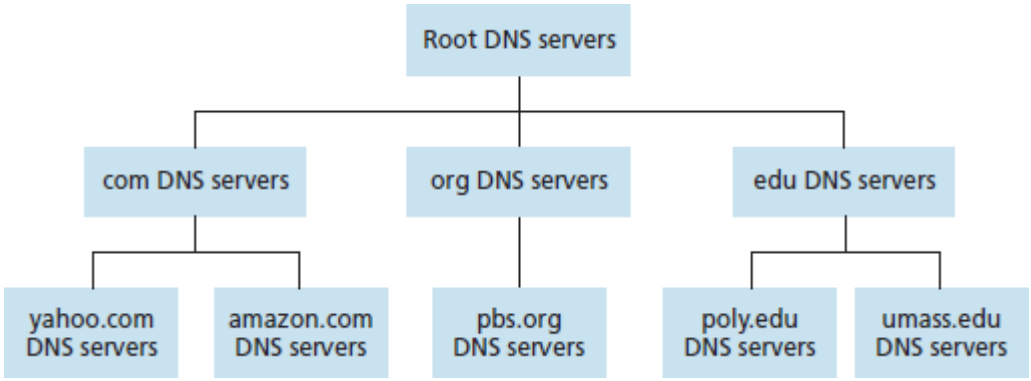
Q.3 b) Explain how DNS server will interact to various DNS servers hierarchically.

- In order to deal with the issue of scale, the DNS uses a large number of servers, organized in a hierarchical fashion and distributed around the world.
- There are three classes of DNS servers—root DNS servers, top-level domain (TLD) DNS servers, and authoritative DNS servers—organized in a hierarchy.

- **Root DNS servers.** In the Internet there are 13 root DNS servers (labeled A through M), most of which are located in North America.

Although we have referred to each of the 13 root DNS servers as if it were a single server, each "server" is actually a network of replicated servers, for both security and reliability purposes. All together, there are 247 root servers.

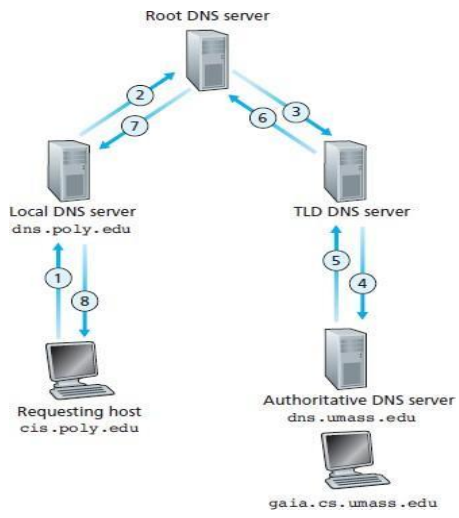
- **Top-level domain (TLD) servers:** These servers are responsible for top-level domains such as com, org, net, edu, and gov, and all of the country top-level domains such as in,uk, fr, ca.
- **Authoritative DNS servers:** Every organization with publicly accessible hosts on the Internet must provide publicly accessible DNS records that map the names of those hosts to IP addresses. An organization's authoritative DNS server houses these DNS records.
- There is another important type of DNS server called the **local DNS server**. A local DNS server does not strictly belong to the hierarchy of servers but is nevertheless central to DNS



architecture. Each ISP—such as a university, an academic department, an employee's company, or a residential ISP—has a local DNS server.

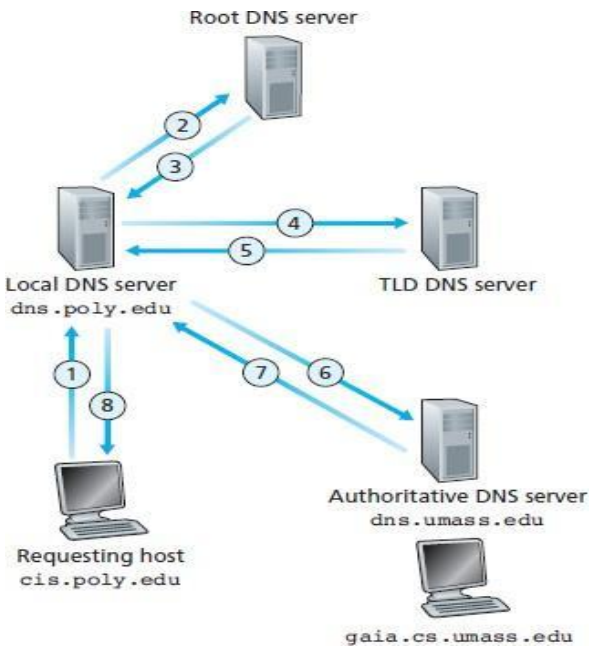
Two type of Interaction:

1) Recursive Queries:



Here DNS query is sent to local DNS server then to root server, then to TLD server and finally to authoritative DNS server. DNS response arrives in the reverse order.

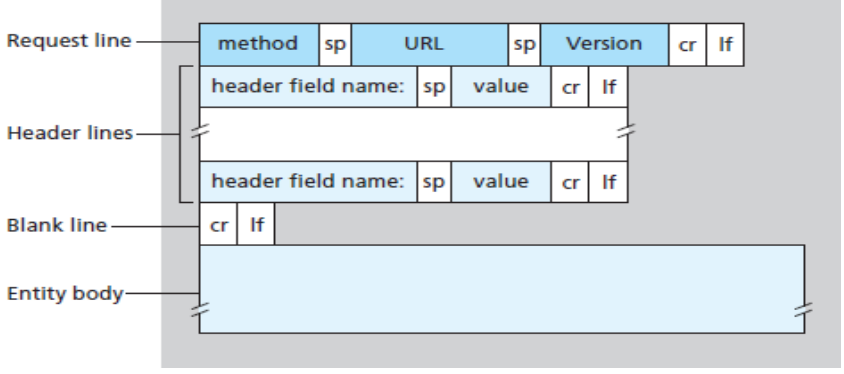
2) Iterative Queries:



Here DNS query will be sent to Local DNS server, then to root server. Root server sends the IP address of TLD server. Now local DNS server sends query to TLD DNS server. TLD DNS server sends the IP address of authoritative DNS server to local DNS server. Now Local DNS server sends query to authoritative DNS server. Authoritative DNS server sends the IP address of host to local DNS server. Local DNS server sends it to the host.

4. Explain about stateless protocol request and response format.

HTTP Request



Message:

Where sp – space, cr – carriage return and lf – line feed.

Method:

There are five HTTP methods:

- **GET:** The GET method is used when the browser requests an object, with the requested object identified in the URL field.
- **POST:** With a POST message, the user is still requesting a Web page from the server, but the specific contents of the Web page depend on what the user entered into the form fields. If the value of the method field is POST, then the entity body contains what the user entered into the form fields.
- **PUT:** The PUT method is also used by applications that need to upload objects to Web servers.
- **HEAD:** Used to retrieve header information. It is used for debugging purpose.
- **DELETE:** The DELETE method allows a user, or an application, to delete an object on a Web server.

URL: Specifies URL of the requested object

Version: This field represents HTTP version, usually HTTP/1.1

Header line:

Ex:

Host: www.someschool.edu

Connection: close

User-agent:

Mozilla/5.0 Accept-

language: fr

The header line **Host:www.someschool.edu** specifies the host on which the object resides.

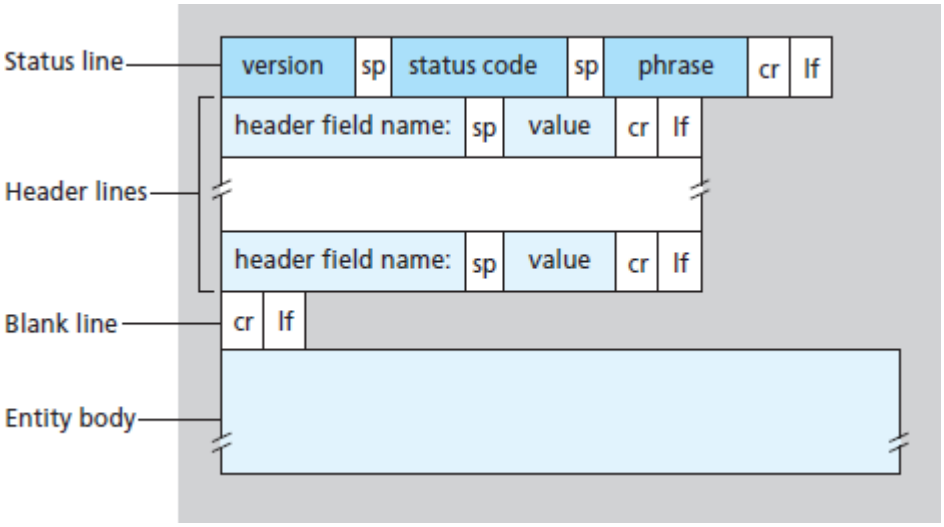
By including the **Connection:close** header line, the browser is telling the server that it doesn't want to bother with persistent connections; it wants the server to close the connection after sending the requested object.

The **User-agent:** header line specifies the user agent, that is, the browser type that is making the

request to the server. Here the user agent is Mozilla/5.0, a Firefox browser.

The **Accept-language:** header indicates that the user prefers to receive a French version of the object, if such an object exists on the server; otherwise, the server should send its default version.

HTTP Response Message



Ex:

```
HTTP/1.1 200 OK
Connection: close
Date: Tue, 09 Aug 2011 15:44:04 GMT
Server: Apache/2.2.3 (CentOS)
Last-Modified: Tue, 09 Aug 2011 15:11:03 GMT
Content-Length: 6821
Content-Type: text/html

(data data data data data ...)
```

The **status line** has three fields: the protocol version field, a status code, and a corresponding status message.

Version is HTTP/1.1

The status code and associated phrase indicate the result of the request. Some common status codes and associated phrases include:

• 200 OK: Request succeeded and the information is returned in the response.

• 301 Moved Permanently: Requested object has been permanently moved; the new URL is specified in Location: header of the response message. The client software will automatically retrieve the new URL.

• 400 Bad Request: This is a generic error code indicating that the request could not be understood by the server.

• 404 Not Found: The requested document does not exist on this server.

• 505 HTTP Version Not Supported: The requested HTTP protocol version is not supported by the server.

Header fields:

- The server uses the **Connection: close** header line to tell the client that it is going to close the TCP connection after sending the message.
- The **Date:** header line indicates the time and date when the HTTP response was created and sent by the server.
- The **Server:** header line indicates that the message was generated by an Apache Web server; it is analogous to the User-agent: header line in the HTTP request message.
- The **Last-Modified:** header line indicates the time and date when the object was created or last modified.
- The **Content-Length:** header line indicates the number of bytes in the object being sent.

The **Content-Type:** header line indicates that the object in the entity body is HTML text.

5. a) Consider an e-commerce site that wants to keep a purchase record for each of its customers. Describe how this can be done with cookies.

Suppose a user, who always accesses the Web using Internet Explorer from her home PC, contacts Amazon.com for the first time. Let us suppose that in the past he has already visited the eBay site. When the request comes into the Amazon Web server, the server creates a unique identification number and creates an entry in its back-end database that is indexed by the identification number. The Amazon Web server then responds to Susan's browser, including in the HTTP response a Set-cookie: header, which contains the identification number.

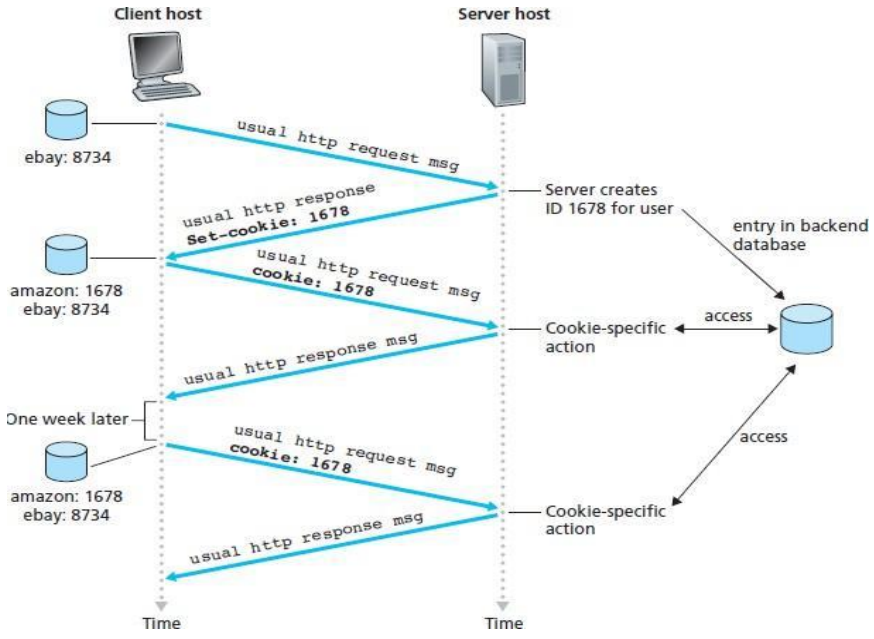
For example, the header line might be:

Set-cookie: 1678

When users browser receives the HTTP response message, it sees the Set-cookie: header. The browser then appends a line to the special cookie file that it manages. This line includes the hostname of the server and the identification number in the Set-cookie: header.

As user continues to browse the Amazon site, each time he requests a Web page, his browser consults his cookie file, extracts his identification number for this site, and puts a cookie header line that includes the identification number in the HTTP request. Specifically, each of his HTTP requests to the Amazon server includes the header line:

Cookie: 1678



5. b) Suppose Alice, with a Web based E-mail account (such as hotmail or Gmail), and sends a message to Bob, who accesses his mail from his mail server using POP3. Discuss how the message gets from Alice's host to Bob's Host. (Be sure to list the series of application layers protocols that are used to move the message between the two hosts.)

When an e-mail message is sent from one person to another, a header containing peripheral information precedes the body of the message.

The header lines and the body of the message are separated by a blank line.

Every header must have a From: header line and a To: header line; a header may include a Subject: header line as well as other optional header lines.

A typical message header looks like this:

From: alice@crepes.fr
To: bob@hamburger.edu
Subject: Searching for the meaning of life.

SMTP protocol delivers the mail to the mail server. To fetch the mail from mail server receiver used mail access protocols.

There are currently a number of popular mail access protocols, including Post Office Protocol—Version 3 (POP3), Internet Mail Access Protocol (IMAP), and HTTP.

POP3

- POP3 is an extremely simple mail access protocol.
- POP3 begins when the user agent (the client) opens a TCP connection to the mail server (the server) on port 110.
- With the TCP connection established, POP3 progresses through three phases: authorization, transaction, and update.
- During the **authorization phase**, the user agent sends a username and a password to authenticate the user.
- During the **transaction phase**, the user agent retrieves messages; also during this phase, the user agent can mark messages for deletion, remove deletion marks, and obtain mail statistics.
- The update phase occurs after the client has issued the quit command, ending the POP3 session; at this time, the mail server deletes the messages that were marked for deletion.
- In a POP3 transaction, the user agent issues commands, and the server responds to each command with a reply. There are two possible responses: +OK used by the server to indicate that the previous command was fine; and -ERR, used by the server to indicate that something was wrong with the previous command.
- The authorization phase has two principal commands: user <username> and pass <password>.

```
user bob
+OK
pass hungry
+OK user successfully logged on
```

- A user agent using POP3 can often be configured (by the user) to "**download and delete**" or to "**download and keep**."
- In the download-and-delete mode, the user agent will issue the list, retr, and dele commands.

Ex:

```
C:
list
```

C: retr 1
S: (blah
blah ... S:
.....
S blah)
S: .
C: dele 1
C: retr 2
S: (blah blah ...
S: .....
S blah)

- A problem with this download-and-delete mode is that the recipient cannot access mail messages from multiple machines.

In the download-and keep mode, the user agent leaves the messages on the mail server after downloading them. In this case, user can reread messages from different machines.

6 a) What are the steps involved between client and server in order to fetch 10 JPEG images which are residing in the same server by using non-persistent HTTP connection, the URL for the base HTML file is http://www.xyz.edu/department/base.index

Let's suppose the page consists of a base HTML file and 10 JPEG images, and that all 11 of these objects reside on the same server.

Further suppose the URL for the base HTML file is

http://www.xyz.edu/department/base.index Here is what

happens:

1. The HTTP client process initiates a TCP connection to the server www.xyz.edu on port number 80, which is the default port number for HTTP. Associated with the TCP connection, there will be a socket at the client and a socket at the server.
2. The HTTP client sends an HTTP request message to the server via its socket. The request message includes the path name /department/base.index.
3. The HTTP server process receives the request message via its socket, retrieves the object

/department/base.index from its storage (RAM or disk), encapsulates the object in an HTTP response message, and sends the response message to the client via its socket.

4. The HTTP server process tells TCP to close the TCP connection.

5. The HTTP client receives the response message. The TCP connection terminates. The message indicates that the encapsulated object is an HTML file. The client extracts the file from the response message, examines the HTML file, and finds references to the 10 JPEG objects.

The first four steps are then repeated for each of the referenced JPEG objects.

6 b) Discuss FTP and its commands and replies.

Some of the more common commands are given below:

- USER username: Used to send the user identification to the server.
- PASS password: Used to send the user password to the server.
- LIST: Used to ask the server to send back a list of all the files in the current remote directory. The list of files is sent over a (new and non-persistent) data connection rather than the control TCP connection.
- RETR filename: Used to retrieve (that is, get) a file from the current directory of the remote host. This command causes the remote host to initiate a data connection and to send the requested file over the data connection.
- STOR filename: Used to store (that is, put) a file into the current directory of the remote host.

Each command is followed by a reply, sent from server to client. The replies are three-digit numbers, with an optional message following the number.

- 331 Username OK, password required
- 125 Data connection already open; transfer starting
- 425 Can’t open data connection
- 452 Error writing file