ELECTRONIC MAIL

- Electronic mail (or e-mail) allows users to exchange messages.

Architecture:

- The sender and the receiver of the e-mail, Alice and Bob respectively, are connected via a LAN or a WAN to two mail servers. The administrator has created one mailbox for each user where the received messages are stored.
- A *mailbox* is part of a server hard drive, a special file with permission restrictions. Only the owner of the mailbox has access to it. The administrator has also created a queue(spool) to store messages waiting to be sent.
- A simple e-mail from Alice to Bob takes nine different steps, as shown in the figure. Alice and Bob use three different *agents*: a **user agent (UA),** a **message transfer agent(MTA),** and a **message access agent (MAA).**
- When Alice needs to send a message to Bob, she runs a UA program to prepare the message and send it to her mail server. The mail server at her site uses a queue (spool) to store messages waiting to be sent.
- The message ,however, needs to be sent through the Internet from Alice's site to Bob's site using an MTA. Here two message transfer agents are needed: one client and one server.
- Like most client-server programs on the Internet, the server needs to run all the time because it does not know when a client will ask for a connection. The client, on the other hand, can be triggered by the system when there is a message in the queue to be sent.
- The user agenta t the Bob site allows Bob to read the received message. Bob later uses an MAA client to retrieve the message from an MAA server running on the second server.

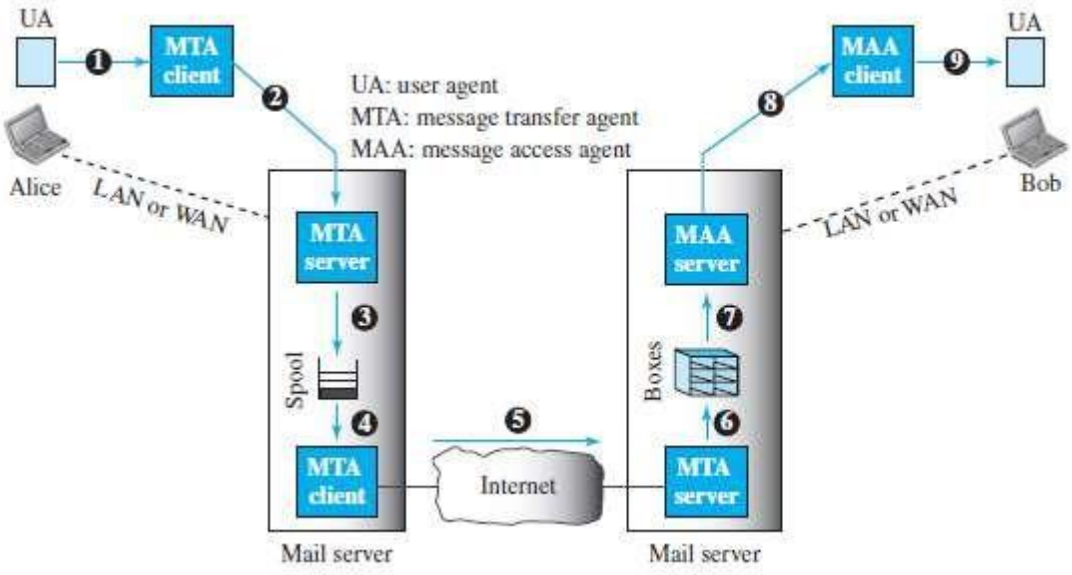


Fig: Common scenario

The electronic mail system needs two UAs, two pairs of MTAs (client and server), and a pair of MAAs(client and server).

User Agent

- The first component of an electronic mail system is the **user agent (UA).** It provides service to the user to make the process of sending and receiving a message easier.
- A user agent is a software package (program) that composes, reads, replies to, and forwards messages. It also handles local mailboxes on the user computers.
- There are two types of user agents: command-driven and GUI-based. Command driven user agents.

Sending Mail:

- To send mail, the user, through the UA, creates mail that looks very similar to postal mail. It has an *envelope* and a *message.* The envelope usually contains the sender address, the receiver address, and other information. The message contains the *header* and the *body*.

Receiving Mail

- The user agent is triggered by the user (or a timer). If a user has mail, the UA informs the user with a notice. If the user is ready to read the mail, a list is displayed in which each line contains a summary of the information about a particular message in the mailbox.

Addresses:

- **A** mail handling system must use an addressing system with unique addresses. a *local part* and a *domain name,* separated by an @ sign.
- The local part defines the name of a special file, called the user mailbox. The second part of the address is the domain name. The domain name assigned to each mail exchanger either comes from the DNS database or is a logical name.

Mailing List or Group List

- Electronic mail allows one name, an *alias,* to represent several different e-mail addresses; this is called a mailing list.

FTP

- **File Transfer Protocol (FTP)** is the standard protocol provided by TCP/IP for copying a file from one host to another.
- For example, two systems may use different file name conventions. Two systems may have different ways to represent data. Two systems may have different directory structures.
- All of these problems have been solved by FTP. FTP is a better choice to transfer large files or to transfer files using different formats.

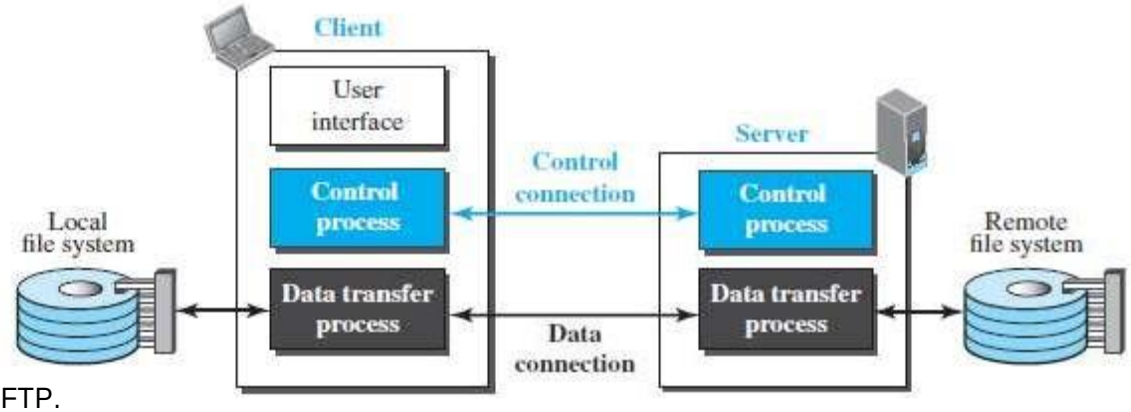


Fig: FTP.

- Figure shows the basic model of FTP.
- The client has three components: the user interface, the client control process, and the client data transfer process. The server has two components: the server control process and the server data transfer process.
- The control connection is made between the control processes. The data connection is made between the data transfer processes.

Two Connections:

- The two connections in FTP have different lifetimes. The control connection remains connected during the entire interactive FTP session.]
- The data connection is opened and then closed for each file transfer activity.
- FTP uses two well-known TCP ports: port 21 is used for the control connection, and port 20 is used for the data connection.

Control Connection:

- For control communication, FTP uses the same approach as TELNET.

- Simple method is adequate for the control connection because we send one command (or response) at a time. Each line is terminated with a two-character (carriage return and line feed) end-of-line token.
- During this control connection, commands are sent from the client to the server and responses are sent from the server to the client.

Command	Arguments	Description
ABOR		Abort the previous command
CDUP		Change to parent directory
CWD	Directory name	Change to another directory
DELE	File name	Delete a file
LIST	Directory name	List subdirectories or files
MKD	Directory name	Create a new directory
PASS	User password	Password
PASV		Server chooses a port
PORT	Port identifier	Client chooses a port
PWD		Display name of current directory
QUIT		Log out of the system
RETR	File name(s)	Retrieve files; files are transferred from server to client
RMD	Directory name	Delete a directory
RNFR	File name (old)	Identify a file to be renamed
RNTO	File name (new)	Rename the file
STOR	File name(s)	Store files; file(s) are transferred from client to server
STRU	**F**, **R**, or **P**	Define data organization (**F**: file, **R**: record, or **P**: page)
TYPE	**A**, **E**, **I**	Default file type (**A**: ASCII, **E**: EBCDIC, **I**: image)
USER	User ID	User information
MODE	**S**, **B**, or **C**	Define transmission mode (**S**: stream, **B**: block, or **C**: Compressed

Table: Some FTP commands.

- Every FTP command generates at least one response. A response has two parts: athree-digit number followed by text.

- The numeric part defines the code; the text part defines needed parameters or further explanations.

Code	Description	Code	Description
125	Data connection open	**250**	Request file action OK
150	File status OK	**331**	User name OK; password is needed
200	Command OK	**425**	Cannot open data connection
220	Service ready	**450**	File action not taken; file not available
221	Service closing	**452**	Action aborted; insufficient storage
225	Data connection open	**500**	Syntax error; unrecognized command
226	Closing data connection	**501**	Syntax error in parameters or arguments
230	User login OK	**530**	User not logged in

Table: Some response in FTP.

Data Connection:

- The data connection uses the well-known port 20 at the server site. However, the creation of a data connection is different from the control connection. The followingshows the steps:

1. The client, not the server, issues a passive open using an ephemeral port. This must be done by the client because it is the client that issues the commands for transferring files.
2. Using the PORT command the client sends this port number to the server.
3. The server receives the port number and issues an active open using the wellknown port 20 and the received ephemeral port number.

Communication over Data Connection:

- The purpose and implementation of the data connection are different from those of the control connection. We want to transfer files through the data connection.
- The client must define the type of file to be transferred, the structure of the data, and the transmission mode.
- three attributes of communication: file type, data structure, and transmission mode.

File Type:

- FTP can transfer one of the following file types across the data connection: ASCII file, EBCDIC file, or image file.

Data Structure:

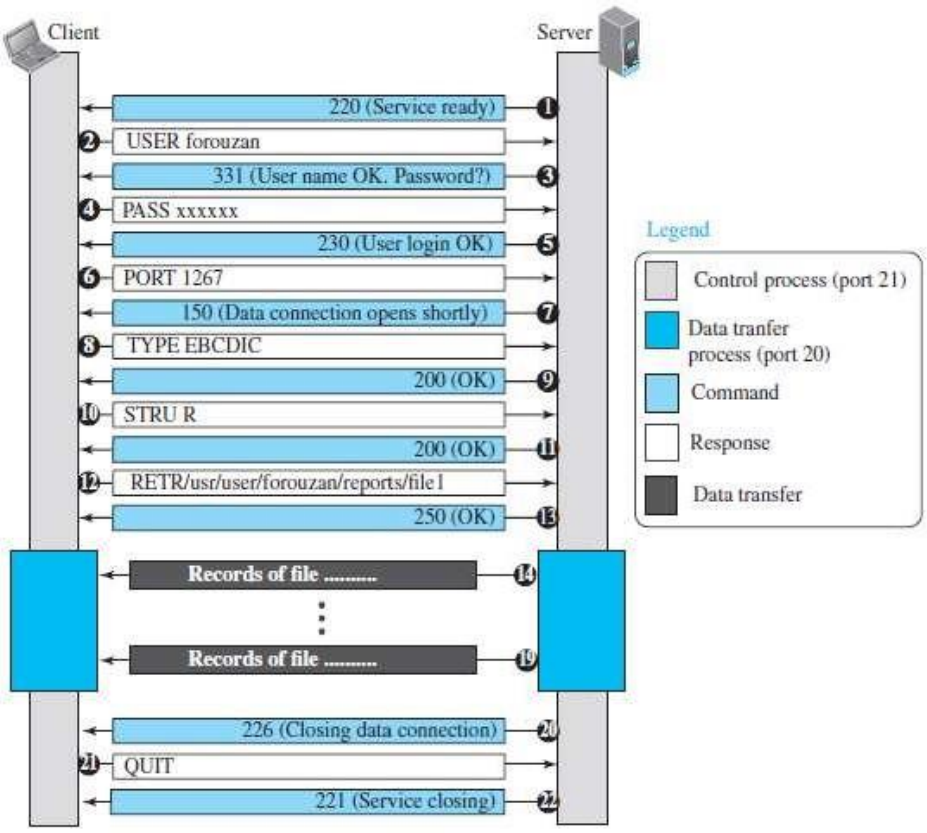
- FTP can transfer a file across the data connection using one of the following interpretations of the structure of the data: *file structure, record structure,* or *page structure*.

Transmission Mode

- FTP can transfer a file across the data connection using one of the following threetransmission modes: *stream mode, block mode,* or *compressed mode*.
- The stream modeis the default mode; data are delivered from FTP to TCP as a continuous stream ofbytes. In the block mode, data can be delivered from FTP to TCP in blocks.

File Transfer

- File transfer occurs over the data connection under the control of the commands sent over the control connection. However, we should remember that file transfer in FTP means one of three things: *retrieving a file* (server to client), *storing a file* (client to server), and *directory listing* (server to client).



```
$ ftp voyager.deanza.fhda.edu
Connected to voyager.deanza.fhda.edu.
220 (vsFTPd 1.2.1)
530 Please login with USER and PASS.
Name (voyager.deanza.fhda.edu:forouzan): forouzan
331 Please specify the password.
Password:********
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
227 Entering Passive Mode (153,18,17,11,238,169)
150 Here comes the directory listing.
drwxr-xr-x   2   3027   411   4096   Sep 24   2002   business
drwxr-xr-x   2   3027   411   4096   Sep 24   2002   personal
drwxr-xr-x   2   3027   411   4096   Sep 24   2002   school
226 Directory send OK.
ftp> quit
221 Goodbye.
```

Fig: Shows an example of using FTP for retrieving a file.

Security for FTP:

- Although FTP requires a password, the password is sent in plaintext (unencrypted), which means it can be intercepted and used by an attacker. The data transfer connection also transfers data in plaintext, which is insecure.
- To be secure, one can add a Secure Socket Layer between the FTP application layer and the TCP layer. In this case FTP is called SSL-FTP.

Hyper Text Transfer Protocol (HTTP)

- The **Hyper Text Transfer Protocol (HTTP)** is used to define how the client-server programs can be written to retrieve web pages from the Web.
- An HTTP client sends a request; an HTTP server returns a response. The server uses the port number 80; the client uses a temporary port number.

Non persistent versus Persistent Connections:
Non persistent Connections:

- In a **non persistent connection**, one TCP connection is made for each request/response.

The following lists the steps in this strategy:

1. The client opens a TCP connection and sends a request.
2. The server sends the response and closes the connection.
3. The client reads the data until it encounters an end-of-file marker; it then closes the connection.

Persistent Connections:

- HTTP version 1.1 specifies a **persistent connection** by default. In a persistent connection ,the server leaves the connection open for more requests after sending a response.
- The server can close the connection at the request of a client or if a time-out has been reached. The sender usually sends the length of the data with each response.
- However, there are some occasions when the sender does not know the length of the data. This is the case when a document is created dynamically or actively.

- In these cases, the server informs the client that the length is not known and closes the connection after sending the data so the client knows that the end of the data has been reached.
- Time and resources are saved using persistent connections. Only one set of buffers and variables needs to be set for the connection at each site.
- The round trip time for connection establishment and connection termination is saved.

Message Formats:

- The HTTP protocol defines the format of the request and response messages. The first section in the request message is called the *request line;* the first section in the response message is called the *status line*.
- The other three sections have the same names in the request and response messages.

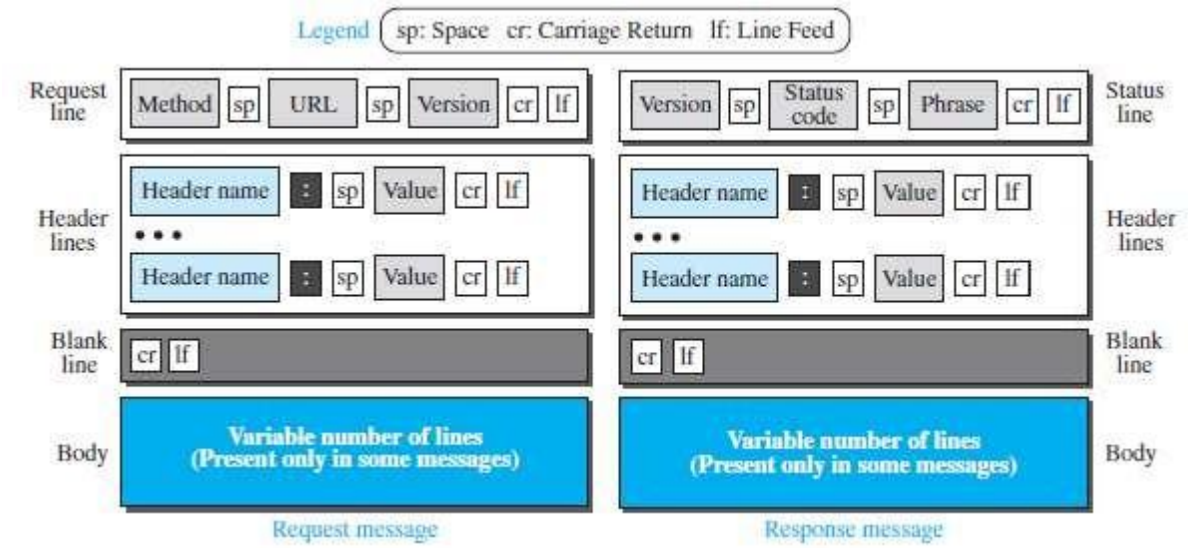


Fig: Formats of the request and response messages.

Request Message:

- The first line in a request message is called a request line. There are three fields in this line separated by one space. The fields are called *method, URL*,and *version*.The method field defines the request types.
- The second field, URL. It defines the address and name of the corresponding web page. The third field, version, gives the version of the protocol; the most current version of HTTP is 1.1.

Method	Action
GET	Requests a document from the server
HEAD	Requests information about a document but not the document itself
PUT	Sends a document from the client to the server
POST	Sends some information from the client to the server.
TRACE	Echoes the incoming request.
DELETE	Removes the web page
CONNECT	Reserved
OPTIONS	Inquires about available options.

Table: Methods.

- After the request line, we can have zero or more *request header* lines. Each header line sends additional information from the client to the server.
- The body can be present in a request message.

Header	Description
User-agent	Identifies the client program
Accept	Shows the media format the client can accept
Accept-charset	Shows the character set the client can handle
Accept-encoding	Shows the encoding scheme the client can handle
Accept-language	Shows the language the client can accept
Authorization	Shows what permissions the client has
Host	Shows the host and port number of the client
Date	Shows the current date
Upgrade	Specifies the preferred communication protocol
Cookie	Returns the cookie to the server (explained later)
If-Modified-Since	If the file is modified since a specific date

Table: Request header names.

Response Message:

- A response message consists of a status line, header lines, a blank line, and sometimes a body. The first line in a response message is called the *status line*.
- There are three fields in this line separated by spaces and terminated by a carriage return and line feed. The first field defines the version of HTTP protocol, currently 1.1.
- The status code field defines the status of the request. It consists of three digits. Whereas the codes in the 100 range are only informational, the codes in the 200 range indicate a successful request.
- The codes in the 300 range redirect the client to another URL. 400 range indicate an error at the client site. Finally, the codes in the 500 range indicate an error at the server site. the status line, we can have zero or more *response header* lines.
- Each headerline sends additional information from the server to the client.

Header	Description
Date	Shows the current date
Upgrade	Specifies the preferred communication protocol
Server	Gives information about the server
Set-Cookie	The server asks the client to save a cookie
Content-Encoding	Specifies the encoding scheme

Content-Language	Specifies the language
Content-Length	Shows the length of the document
Content-Type	Specifies the media type
Location	To ask the client to send the request to another site
Accept-Ranges	The server will accept the requested byte-ranges
Last-modified	Gives the date and time of the last change

Table: Response header names

Cookies:

Creating and Storing Cookies:

- The creation and storing of cookies depend on the implementation; however, the principle is the same.
  1. When a server receives a request from a client, it stores information about the client in a file or a string. The information may include the domain name of the client, the contents of the cookie (information the server has gathered about the client such as name, registration number, and so on), a timestamp, and other information depending on the implementation.
  2. The server includes the cookie in the response that it sends to the client.
  3. When the client receives the response, the browser stores the cookie in the cookie directory, which is sorted by the server domain name.

Using Cookies:

- When a client sends a request to a server, the browser looks in the cookie directory to see if it can find a cookie sent by that server. If found, the cookie is included in the request. When the server receives the request, it knows that this is an old client, not anew one.
  - An *electronic store* (e-commerce) can use a cookie for its client shoppers. When a client selects an item and inserts it in a cart, a cookie that contains information about the item, such as its number and unit price, is sent to the browser. If the client selects a second item, the cookie is updated with the new selection information, and so on. When the client finishes shopping and wants to check out, the last cookie is retrieved and the total charge is calculated.
  - The site that restricts access to *registered clients* only sends a cookie to the client when the client registers for the first time. For any repeated access, only those clients that send the appropriate cookie are allowed.

- A web *portal* uses the cookie in a similar way. When a user selects her favorite pages, a cookie is made and sent. If the site is accessed again, the cookie is sent to the server to show what the client is looking for.
- A cookie is also used by *advertising* agencies. An advertising agency can place banner ads on some main website that is often visited by users.

Web Caching: Proxy Servers:

- HTTP supports **proxy servers**. A proxy server is a computer that keeps copies of responses to recent requests.
- The HTTP client sends a request to the proxy server. The proxy server checks its cache. If the response is not stored in the cache, the proxy server sends the request to the corresponding server.
- Incoming responses are sent to the proxy server and stored for future requests from other clients.
- The proxy server reduces the load on the original server, decreases traffic, and improves latency. proxy server acts as both server and client.
- When it receives a request from a client for which it has a response, it acts as a server and sends the response to the client. When it receives a request from a client for which it does not have a response, it first acts as a client and sends a request to the target server.
- When the response has been received, it acts again as a server and sends the response to the client.

Proxy Server Location:

- The proxy servers are normally located at the client site.

1. A client computer can also be used as a proxy server, in a small capacity, that stores responses to requests often invoked by the client.
2. In a company, a proxy server may be installed on the computer LAN to reduce the load going out of and coming into the LAN.
3. An ISP with many customers can install a proxy server to reduce the load going out of and coming into the ISP network.

Cache Update:

- A very important question is how long a response should remain in the proxy server before being deleted and replaced. Several different strategies are used for this purpose.

- One solution is to store the list of sites whose information remains the same for a while.
- Another recommendation is to add some headers to show the last modification time of the information.

HTTP Security:

- HTTP canbe run over the Secure Socket Layer (SSL). In this case, HTTP is referred to as HTTPS.HTTPSprovidesconfidentiality,clientandserverauthentication,anddataintegrity.

POP3

- **Post Office Protocol, version 3 (POP3)** is simple but limited in functionality. The clientPOP3 software is installed on the recipient computer; the server POP3 software is installed on the mail server.
- Mail access starts with the client when the user needs to download its e-mail from the mailbox on the mail server. The client opens a connection to the server on TCP port 110. It then sends its user name and password to access the mailbox.
- The user can then list and retrieve the mail messages, one by one.
- POP3 has two modes: the *delete* mode and the *keep* mode.
- In the delete mode, the mail is deleted from the mailbox after each retrieval. In the keep mode, the mail remains in the mailbox after retrieval.
- The delete mode is normally used when the user is working at her permanent computer.

IMAP4

- Another mail access protocol is **Internet Mail Access Protocol, version 4 (IMAP4).**IMAP4 is similar to POP3, but it has more features; IMAP4 is more powerful and more complex.
- POP3 is deficient in several ways. It does not allow the user to organize her mail on the server; the user cannot have different folders on the server. In addition, POP3 does not allow the user to partially check the contents of the mail before downloading.

IMAP4 provides the following extra functions:

- A user can check the e-mail header prior to downloading.
- A user can search the contents of the e-mail for a specific string of characters priorto downloading.
- A user can partially download e-mail. This is especially useful if bandwidth is limited and the e-mail contains multimedia with high bandwidth requirements.
- A user can create, delete, or rename mailboxes on the mail server.
- A user can create a hierarchy of mailboxes in a folder for e-mail storage.

MIME:

- Electronic mail has a simple structure. **Multipurpose Internet Mail Extensions (MIME)** is a supplementary protocol that allows non-ASCII data to be sent through e-mail.

Fig: MIME.

MIME Headers:

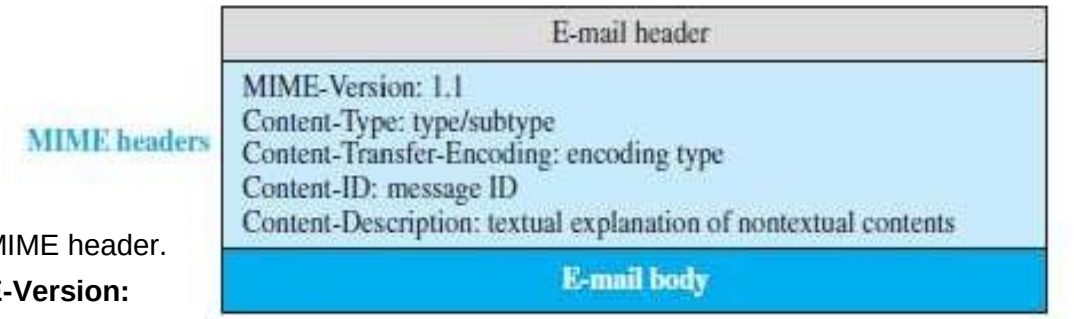


Fig: MIME header.

MIME-Version:

- This header defines the version of MIME used.

Content-Type:

- This header defines the type of data used in the body of the message. The content type and the content subtype are separated by a slash.

Table: Data types and subtypes in MIME.

Type	Subtype	Description
Text	Plain	Unformatted
	HTML	HTML format (see Appendix C)
Multipart	Mixed	Body contains ordered parts of different data types
	Parallel	Same as above, but no order
	Digest	Similar to Mixed, but the default is

		message/RFC822
	Alternative	Parts are different versions of the same message
Message	RFC822	Body is an encapsulated message
	Partial	Body is a fragment of a bigger message
	External-Body	Body is a reference to another message
Image	JPEG	Image is in JPEG format
	GIF	Image is in GIF format
Type	**Subtype**	**Description**
Video		
Audio		
Application	PostScript	Adobe PostScript
	Octet-stream	General binary data (eight-bit bytes)

Table: Data Types and subtypes in MIME.

Content-Transfer-Encoding:

- This header defines the method used to encode the messages into 0s and 1s for transport. The five types of encoding methods are listed in table.

Type	Description
7-bit	NVT ASCII characters with each line less than 1000 characters
8-bit	Non-ASCII characters with each line less than 1000 characters
Binary	Non-ASCII characters with unlimited-length lines
Base64	6-bit blocks of data encoded into 8-bit ASCII characters
Quoted-printable	Non-ASCII characters encoded as an equal sign plus an ASCII code

Table: Methods for Content- Transfer – Encoding.

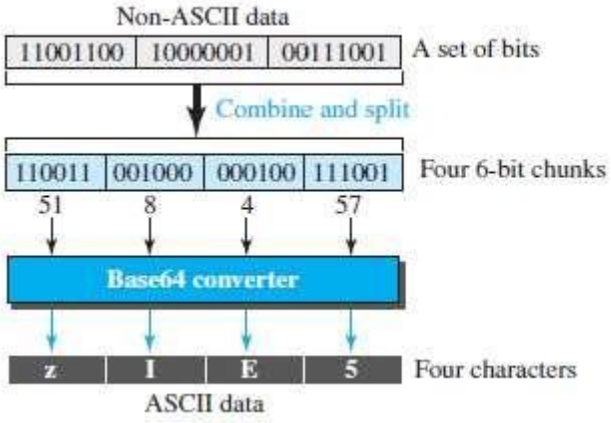


Fig: Base64 Conversion.

Content-ID:

- This header uniquely identifies the whole message in a multiple message environment.

Content-Description

- This header defines whether the body is image, audio, or video.

Web-Based Mail:

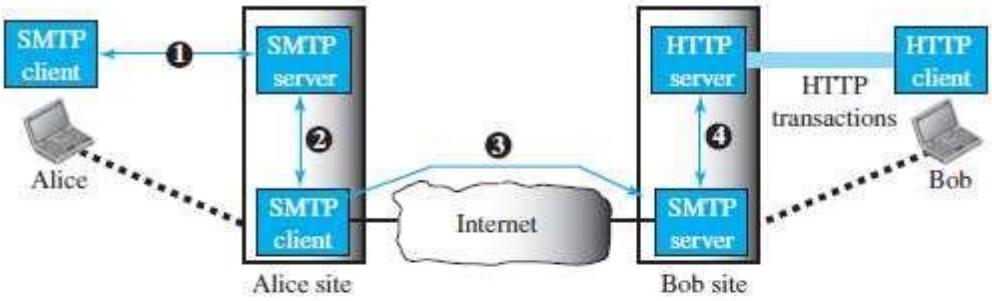
- E-mail is such a common application that some websites today provide this service to anyone who accesses the site. Three common sites are Hotmail, Yahoo, and Googlemail.

Case I

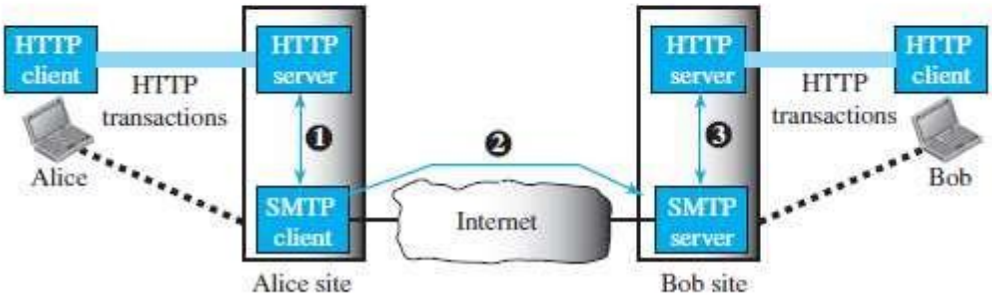
- In the first case, Alice, the sender, uses a traditional mail server; Bob, the receiver, has an account on a web-based server. Mail transfer from Alice's browser to her mail server is done through SMTP. The transfer of the message from the sending mail server to the receiving mail server is still through SMTP.
- However, the message from the receiving server (the web server) to Bob's browser is done through HTTP. In other words, instead of using POP3 or IMAP4, HTTP is norm lly used.

Case II:

- In the second case, both Alice and Bob use web servers, but not necessarily the same server. Alice sends the message to the web server using HTTP transactions. Alice sends an HTTP request message to her web server using the name and address of Bob's mailbox as the URL.
- The server at the Alice site passes the message to the SMTP client and sends it to the server at the Bob site using SMTP protocol. Bob receives the message using HTTP transactions.
- However, the message from the server at the Alice site to the server at the Bob site still takes place using SMTP protocol.



Case 1: Only receiver uses HTTP



Case 2: Both sender and receiver use HTTP

Fig: Web- based e-mail, cases I and II.

SECURE SHELL (SSH)

- **Secure Shell (SSH)** is a secure application program that can be used today for several purposes such as remote logging and file transfer, it was originally designed to replace TELNET.
- There are two versions of SSH: SSH-1 and SSH-2, which are totally incompatible. The first version, SSH-1, is now deprecated because of security flaws in it.

Components:

SSH is an application-layer protocol with three components.

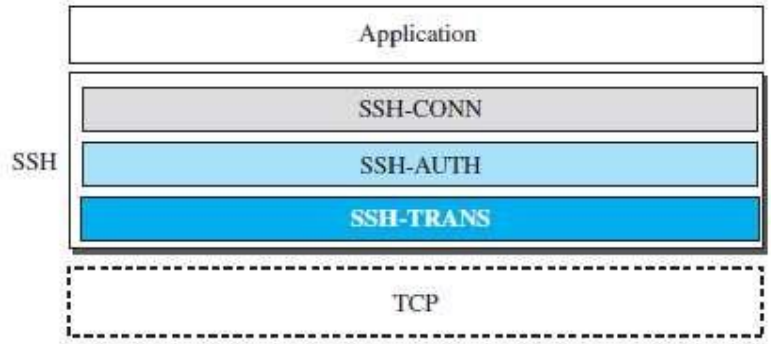


Fig: Components of SSH.

SSH Transport-Layer Protocol (SSH-TRANS)

- TCP is not a secured transport-layer protocol, SSH first uses a protocol that creates a secured channel on top of the TCP. This new layer is an independent protocol referred to as SSH-TRANS.

- services provided by this protocol:

1. Privacy or confidentiality of the message exchanged
2. Data integrity, which means that it is guaranteed that the messages exchanged between the client and server are not changed by an intruder.
3. Server authentication, which means that the client is now sure that the server is the one that it claims to be
4. Compression of the messages, which improves the efficiency of the system and makes attack more difficult.

SSH Authentication Protocol (SSH-AUTH)

- After a secure channel is established between the client and the server and the server is authenticated for the client, SSH can call another procedure that can authenticate the client for the server.
- The client authentication process in SSH is very similar to what is done in Secure Socket Layer (SSL),This layer defines a number of authentication tools.

SSH Connection Protocol (SSH-CONN)

- After the secured channel is established and both server and client are authenticated for each other, SSH can call a piece of software that implements the third protocol, SSHCONN.
- One of the services provided by the SSH-CONN protocol is multiplexing.

Applications

- Although SSH is often thought of as a replacement for TELNET, SSH is, in fact, a general-purpose protocol that provides a secure connection between a client and server.

SSH for Remote Logging

- Several free and commercial applications use SSH for remote logging. Eg. PuTTy. is a client SSH program that can be used for remote logging.

SSH for File Transfer

- One of the application programs that is built on top of SSH for file transfer is the *Secure File Transfer Program* (*sftp*). The *sftp* application program uses one of the channels provided by the SSH to transfer files. Another common application is called *Secure Copy*(*scp*).
- This application uses the same format as the UNIX copy command.

Port Forwarding

- One of the interesting services provided by the SSH protocol is **port forwarding**. We can use the secured channels available in SSH to access an application program that does not provide security services.
- So The SSH port forwarding mechanism creates a tunnel through which the messages belonging to other protocols can travel. For this reason, this mechanism is sometimes referred to as SSH *tunneling*.

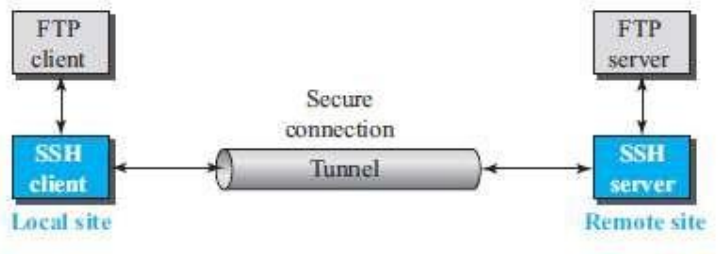


Fig: Port Forwarding.

- The FTP client can use the SSH client on the local site to make a secure connection with the SSH server on the remote site.
- Any request from the FTP client to the FTP server is carried through the tunnel provided by the SSH client and server.

- Any response from the FTP server to the FTP client is also carried through the tunnel provided by the SSH client and server.

Format of the SSH Packets:

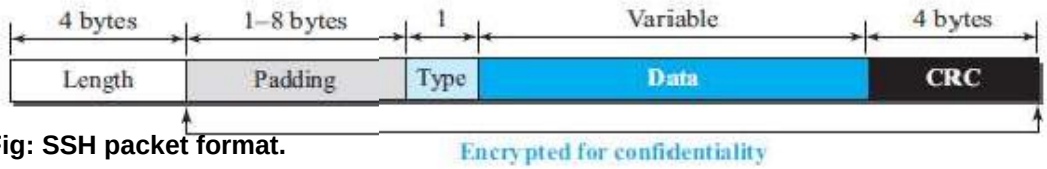


Fig: SSH packet format.

- The length field defines the length of the packet. The *cyclic redundancy check* (CRC) field is used for error detection.
- The type field designates the type of the packet used in different SSH protocols.
- The data field is the data transferred by the packet in different protocols.

SMTP

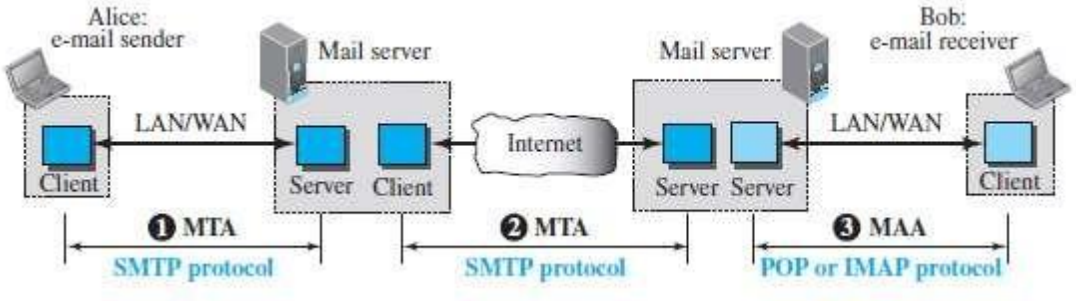


Fig: Protocols used in electronic mail.

- The formal protocol that defines the MTA client and server in the Internet is called ***Simple Mail Transfer Protocol (SMTP).*** SMTP is used two times, between the sender and the sender's mail server and between the two mail servers.

Commands and Responses:

Commands:

-
-
-
- Commands are sent from the client to the server.

Keyword	Argument(s)	Description
HELO	Sender's hostname	Identifies itself
MAIL FROM	Sender of the message	Identifies the sender of the message
RCPT TO	Intended recipient	Identifies the recipient of the message
DATA	Body of the mail	Sends the actual message
QUIT		Terminates the message
RSET		Aborts the current mail transaction
VRFY	Name of recipient	Verifies the address of the recipient
NOOP		Checks the status of the recipient
TURN		Switches the sender and the recipient
EXPN	Mailing list	Asks the recipient to expand the mailing list
HELP	Command name	Asks the recipient to send information about the command sent as the argument
SEND FROM	Intended recipient	Specifies that the mail be delivered only to the terminal of the recipient, and not to the mailbox

SMOL FROM	Intended recipient	Specifies that the mail be delivered to the terminal or the mailbox of the recipient
SMAL FROM	Intended recipient	Specifies that the mail be delivered to the terminal and the mailbox of the recipient

- Responses are sent from the server to the client. A response is a threedigitcode that may be followed by additional textual information.

Table: Responses.

Code	Description
Positive Completion Reply	
211	System status or help reply
214	Help message
220	Service ready
Code	Description
221	Service closing transmission channel
250	Request command completed
Code	Description
251	User not local; the message will be forwarded
Positive Intermediate Reply	
354	Start mail input
Transient Negative Completion Reply	
421	Service not available
450	Mailbox not available
451	Command aborted: local error
452	Command aborted; insufficient storage
Permanent Negative Completion Reply	
500	Syntax error; unrecognized command

Table: Responses (continued)

501	Syntax error in parameters or arguments

502	Command not implemented
503	Bad sequence of commands
504	Command temporarily not implemented
550	Command is not executed; mailbox unavailable
551	User not local
552	Requested action aborted; exceeded storage location
553	Requested action not taken; mailbox name not allowed
554	Transaction failed

Mail Transfer Phases

- The process of transferring a mail message occurs in three phases: connection establishment, mail transfer, and connection termination.

Connection Establishment:

- After a client has made a TCP connection to the wellknown port 25, the SMTP server starts the connection phase. This phase involves the following three steps:

1. The server sends code 220 (service ready) to tell the client that it is ready to receive mail. If the server is not ready, it sends code 421 (service not available).
2. The client sends the HELO message to identify itself, using its domain name address. This step is necessary to inform the server of the domain name of the client.
3. The server responds with code 250 (request command completed) or some other code depending on the situation.

Message Transfer:

- After connection has been established between the SMTP client and server, a single message between a sender and one or more recipients can be exchanged.
- This phase involves eight steps. Steps 3 and 4 are repeated if there is more than one recipient.

1. The client sends the MAIL FROM message to introduce the sender of the message. It includes the mail address of the sender (mailbox and the domain name). This step is needed to give the server the return mail address for returning errors and reporting messages.
2. The server responds with code 250 or some other appropriate code.
3. The client sends the RCPT TO (recipient) message, which includes the mail address of the recipient.

4. The server responds with code 250 or some other appropriate code.
5. The client sends the DATA message to initialize the message transfer.
6. The server responds with code 354 (start mail input) or some other appropriate message.
7. The client sends the contents of the message in consecutive lines. Each line is terminated by a two-character end-of-line token (carriage return and line feed). The message is terminated by a line containing just one period.
8. The server responds with code 250 (OK) or some other appropriate code.

Connection Termination:

- After the message is transferred successfully, the client terminates inthe connection. This phase involves two steps.

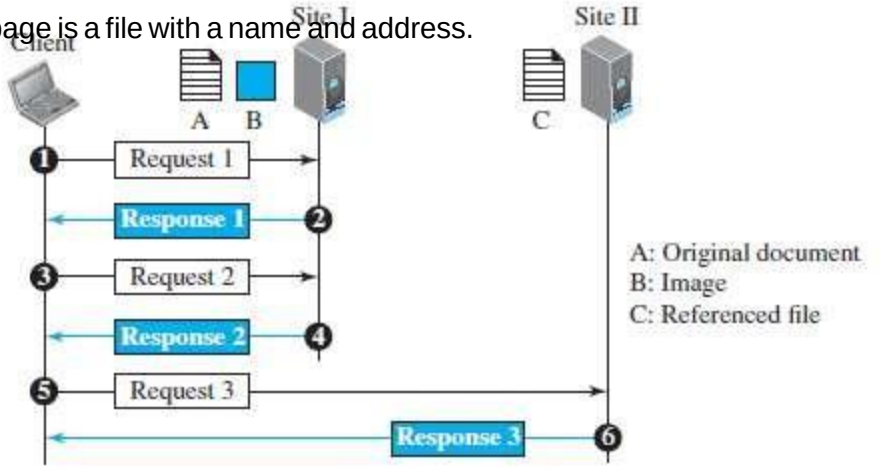
1. The client sends the QUIT command.
2. The server responds with code 221 or some other appropriate code.

WORLD WIDE WEB

World Wide Web (abbreviated WWW or Web).

Architecture

- The WWW today is a distributed client-server service, in which a client using a browser can access a service using a server. However, the service provided is distributed over many locations called *sites.*
- Each site holds one or more web pages. Each web page, however, can contain some links to other web pages in the same or other sites. A simple web page has no links to other web pages; a composite web page has one or more links to other web pages.
- Each web page is a file with a name and address.



Web Client (Browser)

- A variety of vendors offer commercial **browsers** that interpret and display a webpage, and all of them use nearly the same architecture. Each browser usually consistsof three parts: a controller, client protocols, and interpreters.

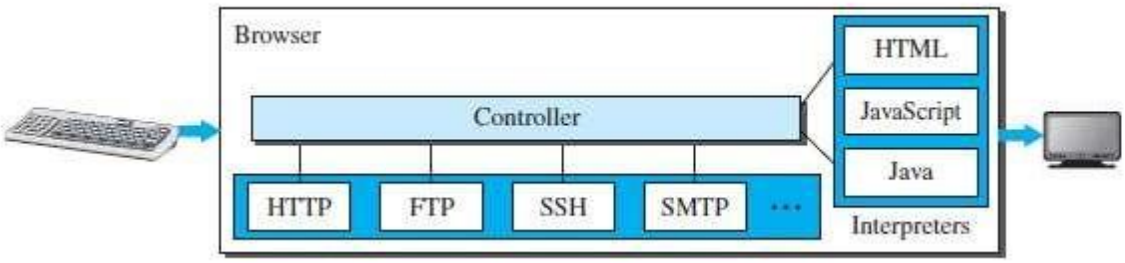


Fig: Browser

- The controller receives input from the keyboard or the mouse and uses the client programs to access the document. After the document has been accessed, the controller uses one of the interpreters to display the document on the screen.

Web Server

- The web page is stored at the server. Each time a request arrives, the corresponding document is sent to the client.
- To improve efficiency, servers normally store requested files in a cache in memory; memory is faster to access than a disk. A server can also become more efficient through multithreading or multiprocessing.
- In this case, a server can answer more than one request at a time. Some popular web servers include Apache and Microsoft Internet Information Server.

Uniform Resource Locator (URL)

- Define a web page, we need three identifiers: *host, port,* and *path*, *protocol.*

- **Protocol.** The first identifier is the abbreviation for the client-server program that we need in order to access the web page. Although most of the time the protocol is HTTP (HyperText Transfer Protocol). we can also use other protocols such as FTP (File Transfer Protocol).
- **Host.** The host identifier can be the IP address of the server or the unique name given to the server.
- **Port.** The port, a 16-bit integer, is normally predefined for the client-server application.
- **Path.** The path identifies the location and the name of the file in the underlying operating system. To combine these four pieces together, the **uniform resource locator (URL)** has been designed.

protocol://host/path	**Used most of the time**
protocol://host:port/path	**Used when port number is needed**

Static Documents:

- **Static documents** are fixed-content documents that are created and stored in a server. The client can get a copy of the document only. In other words, the contents of the file are determined when the file is created.
- Static documents are prepared using one of several languages: HyperText Markup Language (HTML), Extensible Markup Language (XML), Extensible Style Language (XSL), and Extensible Hypertext Markup Language (XHTML).

Dynamic Documents:

- A **dynamic document** is created by a web server whenever a browser requests the document. When a request arrives, the web server runs an application program or a script that creates the dynamic document.
- The server returns the result of the program or script as a response to the browser that requested the document.
- *Java Server Pages* (*JSP*),or *Active Server Pages* (*ASP*), Visual Basic language for scripting, or *ColdFusion,* Structured Query Language (SQL).

Active Documents:

- For many applications, we need a program or a script to be run at the client site. These are called **active documents.**
- When a browser requests an active document, the server sends a copy of the document ora script. The document is then run at the client (browser) site. One way to create an active document is to use *Java applets,* a program written in Java on the server.
- It is compiled and ready to be run. The document is in byte code (binary) format. Another way is to use *Java Scripts* but download and run the script at the client site.