Standard Client-Server Protocols

WORLD WIDE WEB

- **World Wide Web:**
- 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.
- In other words, a web page can be **simple or composite**. A simple web page has no links to other web pages; a composite web page has one or more links to other web pages.
- **Web Client (Browser):**
- A variety of vendors offer commercial browsers that interpret and display a web page, and all of them use nearly the same architecture. Each browser usually consists of three parts: **a controller, client protocols, and interpreters.**
- The **controller** receives input from the keyboard or the mouse.
- **Client programs** are used to access the document. (HTTP,FTP)
- The controller uses one of the **interpreters** to display the document on the screen. (HTML,JavaScript)

Continue…

- **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.
- **Uniform Resource Locator (URL):**
- To define a web page, we need three identifiers: **host, port, and path.** However, before defining the web page, we need to tell the browser what client-server application we want to use, which is called the **protocol**.
- E.g. **protocol://host:port/path**

Continue…

- **Web Documents:**
- The documents in the WWW can be grouped into three broad categories: **static, dynamic, and active.**
- **Static Documents: Static documents** are fixed-content documents that are created and stored in a server. 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. E.g. JSP,ASP etc.
- **Active Documents:** For many applications, we need a program or a script to be run at the client site. These are called **active documents. E.g.** Java applets, JavaScript.

HyperText Transfer Protocol (HTTP)

- The HyperText 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.
- HTTP uses the services of TCP.
- This means that, before any transaction between the client and the server can take place, a connection needs to be established between them. After the transaction, the connection should be terminated.
- **Nonpersistent versus Persistent Connections:**
- In a **nonpersistent** 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.

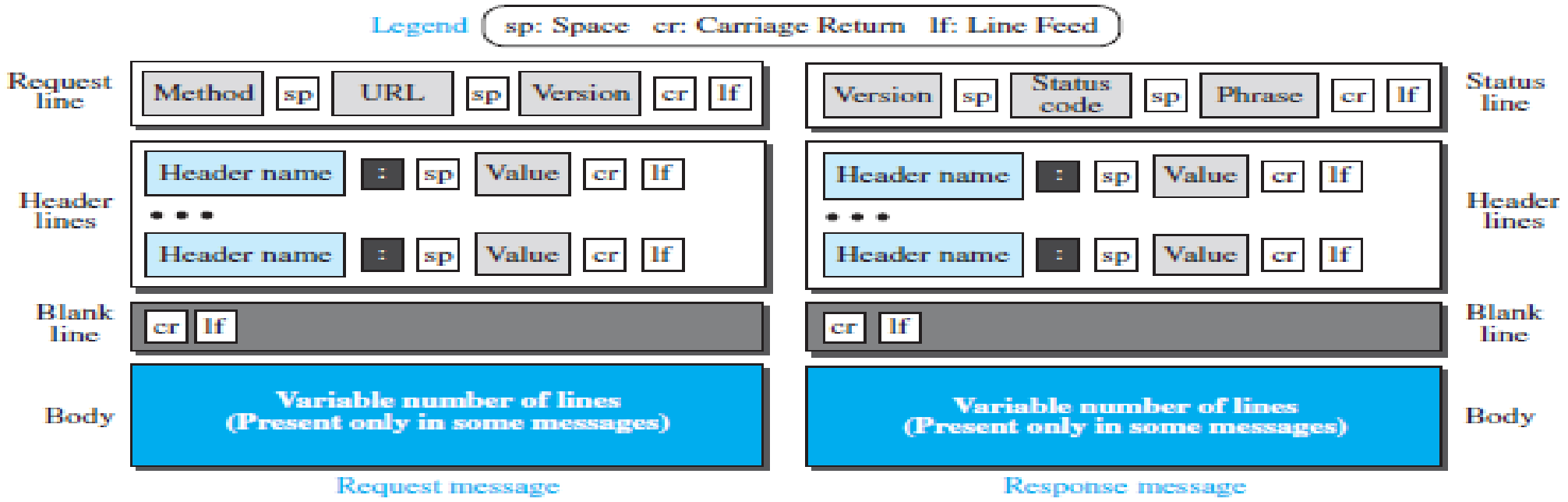
Continue…

- 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.
- Time and resources are saved using persistent connections.
- The round trip time for connection establishment and connection termination is saved.

Continue…

- **Message Formats:**

Figure 26.5 *Formats of the request and response messages*



Continue…

- **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 and terminated by two characters (carriage return and line feed). The fields are called **method, URL, and version**.

Table 26.1 *Methods*

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

Continue…

- 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.

Table 26.2 *Request header names*

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

Continue…

- **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. They are **HTTP version, status code, status text.**
- After the status line, we can have zero or more response header lines. Each header line sends additional information from the server to the client.

Continue…

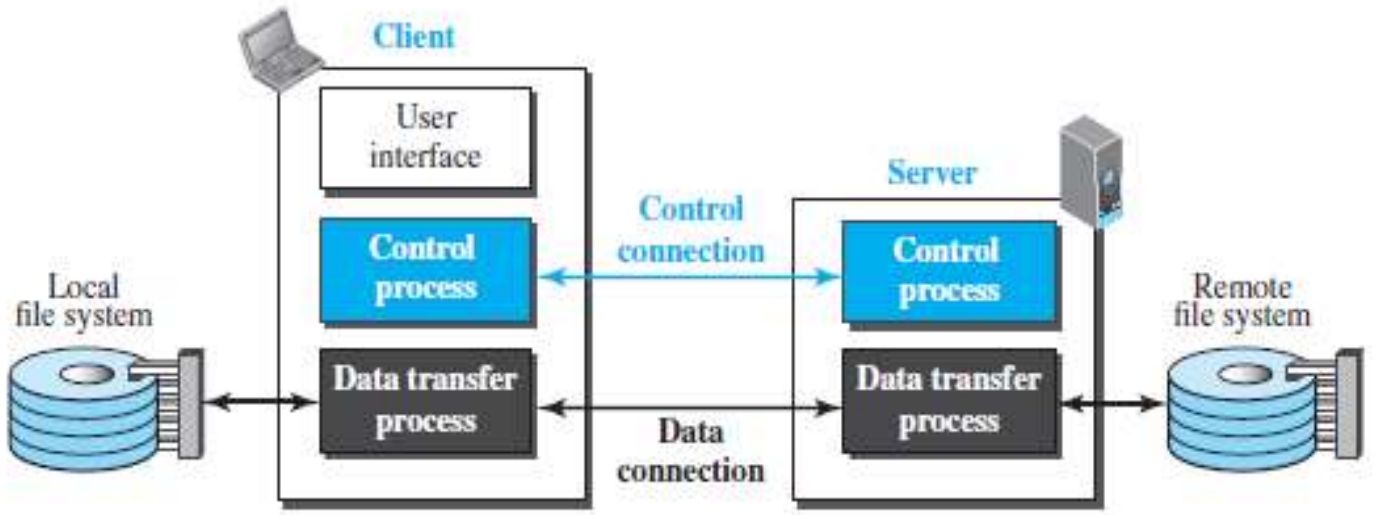
Table 26.3 *Response header names*

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

FTP

- **File Transfer Protocol (FTP)** is the standard protocol provided by TCP/IP for copying a file from one host to another.

Figure 26.10 *FTP*



Continue…

- 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.

Continue…

Table 26.4 *Some FTP commands*

Command	*Argument(s)*	*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

Continue…

Table 26.5 *Some responses in FTP*

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

Continue…

- **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 following shows the steps:

I. 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.

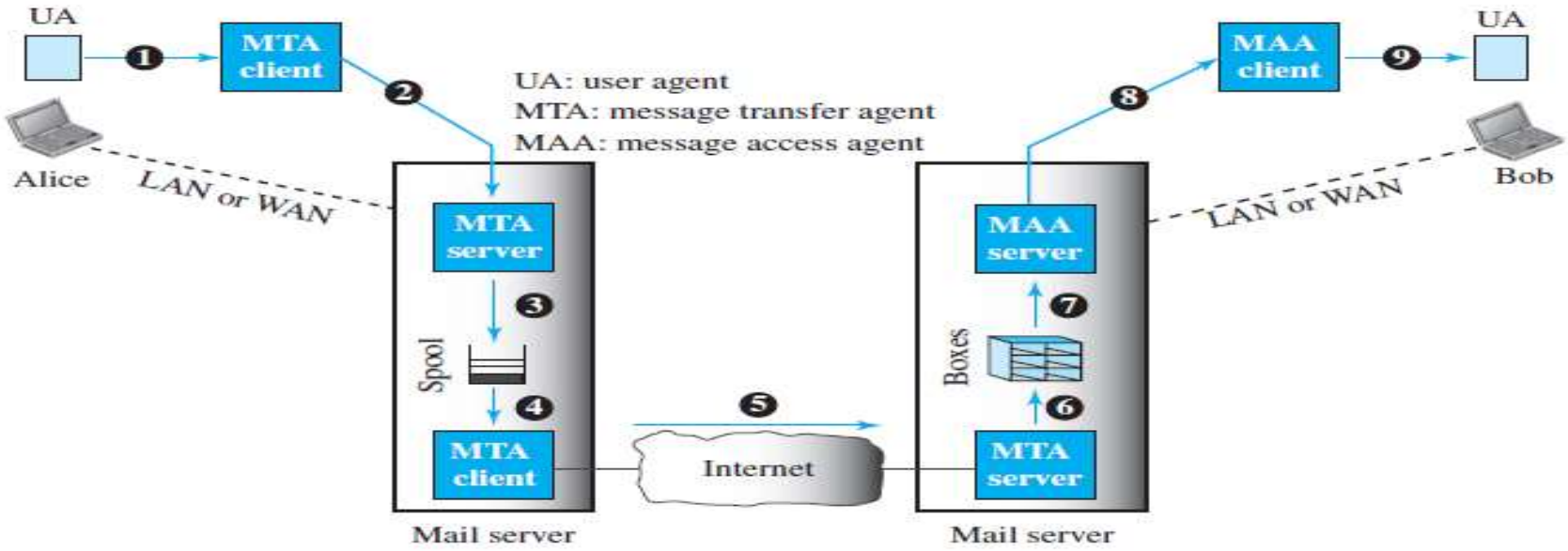
II. Using the PORT command the client sends this port number to the server.

III. The server receives the port number and issues an active open using the wellknown port 20 and the received ephemeral port number.

ELECTRONIC MAIL

- Electronic mail (or e-mail) allows users to exchange messages.
- E-mail is considered a one-way transaction.
- **Architecture:**

Figure 26.12 *Common scenario*



Continue…

- The sender and the receiver of the e-mail 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.
- 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.
- The user agent at 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.

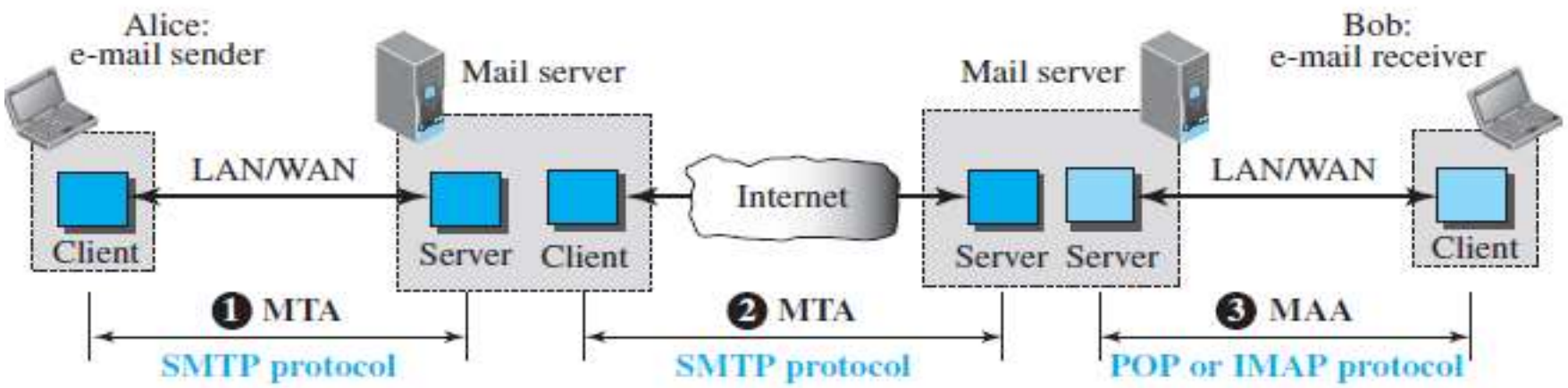
Continue…

- **User Agent:**
- 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.
- **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.
- **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.
- **Addresses:**
- To deliver mail, a mail handling system must use an addressing system with unique addresses.

Continue…

- **Message Transfer Agent: SMTP**
- 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.

Figure 26.15 *Protocols used in electronic mail*



Continue…

- **Commands and Respor**

Table 26.6 *SMTP commands*

Keyword	*Argument(s)*	*Description*
HELO	Sender's host name	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

Continue…

- **Commands and Respo**

Table 26.7 *Responses*

Code	*Description*
	Positive Completion Reply
211	System status or help reply
214	Help message
220	Service ready
221	Service closing transmission channel
250	Request command completed
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

Continue…

- **Commands and Responses:**

Table 26.7 *Responses (continued)*

Code	*Description*
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

Continue…

- **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.

Continue…

- **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.

Continue…

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 the 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.

Message Access Agent: POP and IMAP

- The first and second stages of mail delivery use SMTP. However, SMTP is not involved in the third stage because SMTP is a push protocol; it pushes the message from the client to the server.
- On the other hand, the third stage needs a pull protocol; the client must pull messages from the server.
- The third stage uses a message access agent.
- Currently two message access protocols are available: Post Office Protocol, version 3 (POP3) and Internet Mail Access Protocol, version 4 (IMAP4).
- **POP3:**
- Post Office Protocol, version 3 (POP3) is simple but limited in functionality.
- The client POP3 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.

Continue…

- 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.
- **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.
- IMAP4 provides the following extra functions:

I. A user can check the e-mail header prior to downloading.

II. A user can search the contents of the e-mail for a specific string of characters prior to downloading.

III. 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.

IV. A user can create, delete, or rename mailboxes on the mail server.

V. A user can create a hierarchy of mailboxes in a folder for e-mail storage.

TELNET

- One of the original remote logging protocols is TELNET, which is an abbreviation for **TErminaL NETwork**. Although TELNET requires a logging name and password, it is vulnerable to hacking because it sends all data including the password in plaintext.
- When a user logs into a local system, it is called **local logging**. As a user types at a terminal or at a workstation running a terminal emulator, the keystrokes are accepted by the terminal driver. The terminal driver passes the characters to the operating system. The operating system, in turn, interprets the combination of characters and invokes the desired application program or utility.
- when a user wants to access an application program or utility located on a remote machine, she performs remote logging. Here the TELNET client and server programs come into use. The user sends the keystrokes to the terminal driver where the local operating system accepts the characters but does not interpret them. The characters are sent to the TELNET client, which transforms the characters into a universal character set called Network Virtual Terminal (NVT) characters and delivers them to the local TCP/IP stack.
- The commands or text, in NVT form, travel through the Internet and arrive at the TCP/IP stack at the remote machine. Here the characters are delivered to the operating system and passed to the TELNET server, which changes the characters to the corresponding characters understandable by the remote computer.

Continue…

- However, the characters cannot be passed directly to the operating system because the remote operating system is not designed to receive characters from a TELNET server; it is designed to receive characters from a terminal driver. The solution is to add a piece of software called a **pseudoterminal driver**, which pretends that the characters are coming from a terminal. The operating system then passes the characters to the appropriate application program.

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.
- **Components:**
- SSH is an application-layer protocol with three components:
- **SSH Transport-Layer Protocol (SSH-TRANS):**
- Since 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:**

I. Privacy or confidentiality of the message exchanged

II. Data integrity, which means that it is guaranteed that the messages exchanged between the client and server are not changed by an intruder

Continue…

III. Server authentication, which means that the client is now sure that the server is the one that it claims to be

IV. 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.
- Authentication starts with the client, which sends a request message to the server.
- The request includes the user name, server name, the method of authentication, and the required data. The server responds with either a success message, which confirms that the client is authenticated, or a failed message, which means that the process needs to be repeated with a new request message.
- **SSH Connection Protocol (SSH-CONN):**

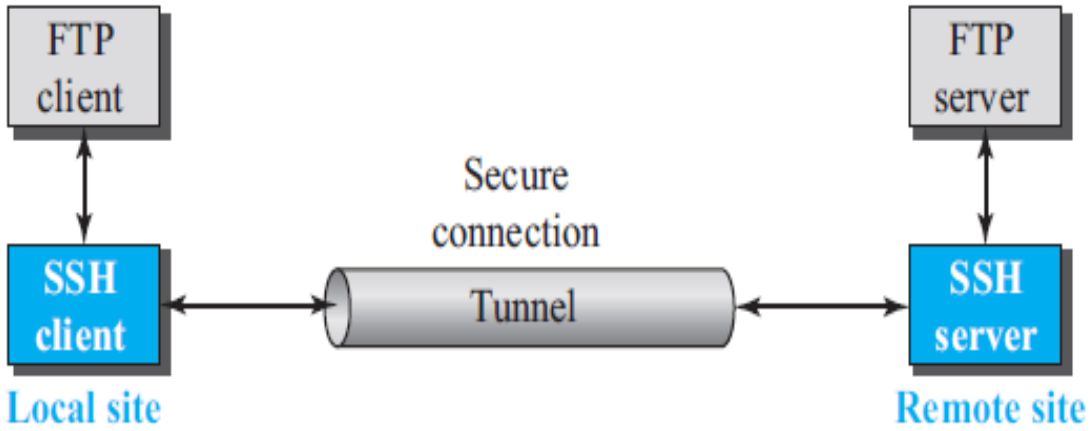
Continue…

- **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.
- SSH-CONN takes the secure channel established by the two previous protocols and lets the client create multiple logical channels over it. Each channel can be used for a different purpose, such as remote logging, file transfer, and so on.
- **Applications:**
- **SSH for Remote Logging:** Several free and commercial applications use SSH for remote logging. E.g. Putty, Tectia.
- **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).** Another common application is called **Secure Copy (scp).** This application uses the same format as the UNIX copy command, cp, to copy files.

Continue…

- **Port Forwarding:** The SSH port forwarding mechanism creates a tunnel through which the messages belonging to other protocols can travel. It is also called as **SSH tunneling.**

Figure 26.26 *Port forwarding*



Continue…

Figure 26.25 *Components of SSH*