Computer Network
Unit-VI

Application Layer

By:- Dr. D.R.Patil

• Outline

- DNS: Name Space, Resource Record and Types of Name Server.
- HTTP,
- HTTPS,
- SMTP,
- Telnet,
- FTP,
- DHCP

- **DHCP**
  - BOOTP is not a dynamic configuration protocol.
  - When a client requests its IP address, the BOOTP server consults a table that matches the physical address of the client with its IP address.
  - This implies that the binding between the physical address and the IP address of the client already exists.
  - The binding is predetermined.
  - However, what if a host moves from one physical network to another? What if a host wants a temporary IP address? BOOTP cannot handle these situations because the binding between the physical and IP addresses is static and fixed in a table until changed by the administrator.
  - **BOOTP is a static configuration protocol.**
  - The Dynamic Host Configuration Protocol (DHCP) has been devised to provide static and dynamic address allocation that can be manual or automatic.

- **Static Address Allocation**
  - In this capacity DHCP acts as BOOTP does.
  - It is backward compatible with BOOTP, which means a host running the BOOTP client can request a static address from a DHCP server.
  - A DHCP server has a database that statically binds physical addresses to IP addresses.
- **Dynamic Address Allocation**
  - DHCP has a second database with a pool of available IP addresses.
  - This second database makes DHCP dynamic.
  - When a DHCP client requests a temporary IP address, the DHCP server goes to the pool of available (unused) IP addresses and assigns an IP address for a negotiable period of time.

- When a DHCP client sends a request to a DHCP server, the server first checks its static database.
- If an entry with the requested physical address exists in the static database, the permanent IP address of the client is returned.
- On the other hand, if the entry does not exist in the static database, the server selects an IP address from the available pool, assigns the address to the client, and adds the entry to the dynamic database.
- The dynamic aspect of DHCP is needed when a host moves from network to network or is connected and disconnected from a network (as is a subscriber to a service provider).
- DHCP provides temporary IP addresses for a limited time.
- The addresses assigned from the pool are temporary addresses.
- The DHCP server issues a lease for a specific time.
- When the lease expires, the client must either stop using the IP address or renew the lease.
- The server has the option to agree or disagree with the renewal.
- If the server disagrees, the client stops using the address.

- **Manual and Automatic Configuration**
  - One major problem with the BOOTP protocol is that the table mapping the IP addresses to physical addresses needs to be manually configured.
  - This means that every time there is a change in a physical or IP address, the administrator needs to manually enter the changes.
  - DHCP, on the other hand, allows both manual and automatic configurations.
  - Static addresses are created manually; dynamic addresses are created automatically.

- **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, which, as discussed before, is a connection-oriented and reliable protocol.
- This means that, before any transaction between the client and the server can take place, a connection needs to be established between them.

- **HyperText Transfer Protocol (HTTP)**
- After the transaction, the connection should be terminated.
- The client and server, however, do not need to worry about errors in messages exchanged or loss of any message, because the TCP is reliable and will take care of this matter.

- **HyperText Transfer Protocol (HTTP)**
- **Nonpersistent versus Persistent Connections**
- As we discussed in the previous section, the hypertext concept embedded in web page documents may require several requests and responses.
- If the web pages, objects to be retrieved, are located on different servers, we do not have any other choice than to create a new TCP connection for retrieving each object.
- However, if some of the objects are located on the same server, we have two choices: to retrieve each object using a new TCP connection or to make a TCP connection and retrieve them all.

- **HyperText Transfer Protocol (HTTP)**
- **Nonpersistent versus Persistent Connections**
- The first method is referred to as a nonpersistent connection, the second as a persistent connection.
- HTTP, prior to version 1.1, specified nonpersistent connections, while persistent connections are the default in version 1.1, but it can be changed by the user.

- **HyperText Transfer Protocol (HTTP)**
- **Nonpersistent 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.
- In this strategy, if a file contains links to N different pictures in different files (all located on the same server), the connection must be opened and closed N + 1 times.
- The nonpersistent strategy imposes high overhead on the server

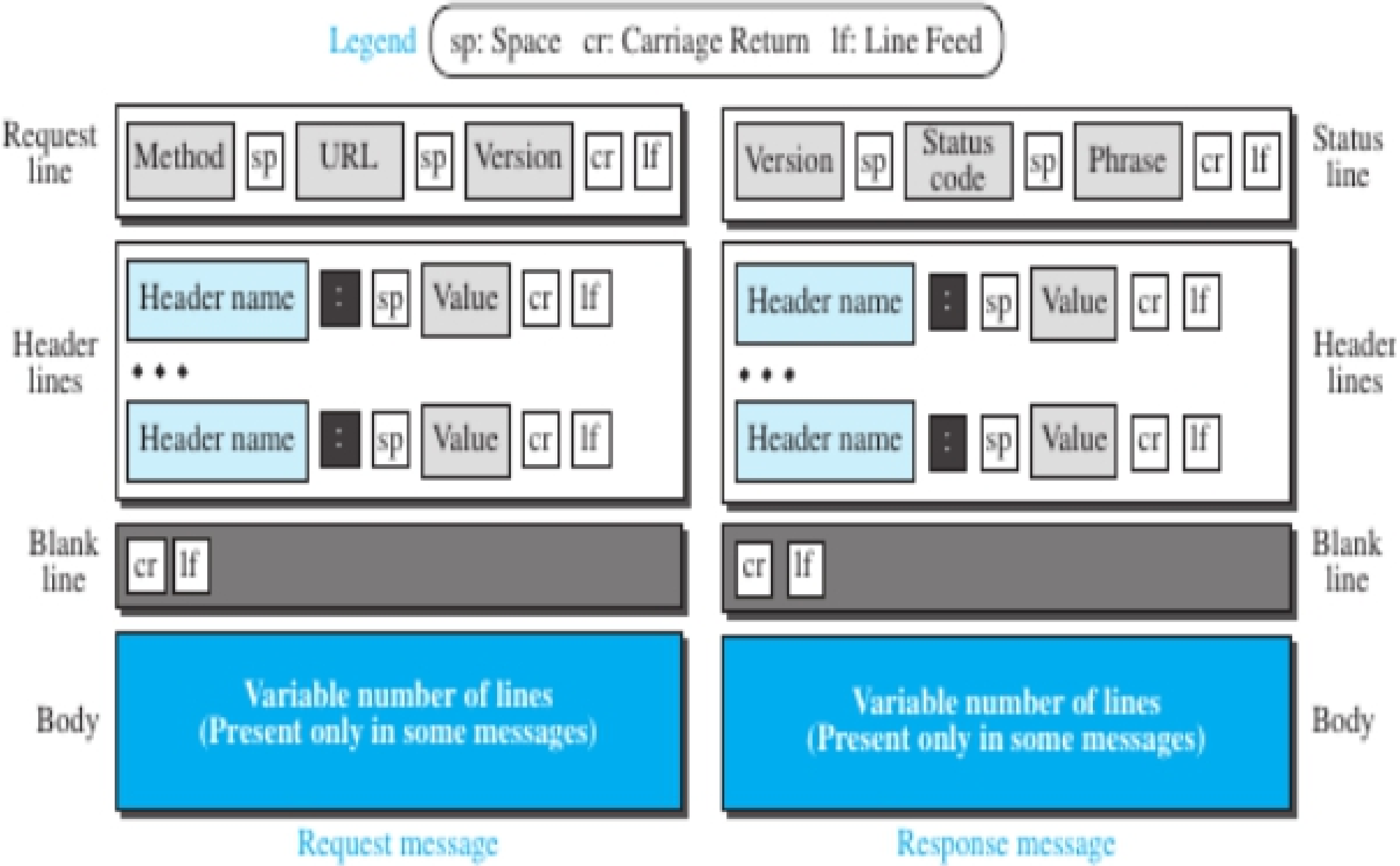
- **HyperText Transfer Protocol (HTTP)**
- **Nonpersistent Connections**
- The nonpersistent strategy imposes high overhead on the server because the server needs N + 1 different buffers each time a connection is opened.

- **HyperText Transfer Protocol (HTTP)**
- **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.

- **HyperText Transfer Protocol (HTTP)**
- **Persistent Connections**
- 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.

- **HyperText Transfer Protocol (HTTP)**
- **Message Formats**
- The HTTP protocol defines the format of the request and response messages, as shown in Figure 26.5.
- We have put the two formats next to each other for comparison.
- Each message is made of four sections.
- 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.

Figure 26.5 *Formats of the request and response messages*



- **HyperText Transfer Protocol (HTTP)**
- **Request Message**
- As we said before, 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) as shown in Figure 26.5.
- The fields are called method, URL, and version.
- The method field defines the request types.
- In version 1.1 of HTTP, several methods are defined, as shown in Table 26.1.
- Most of the time, the client uses the GET method to send a request.

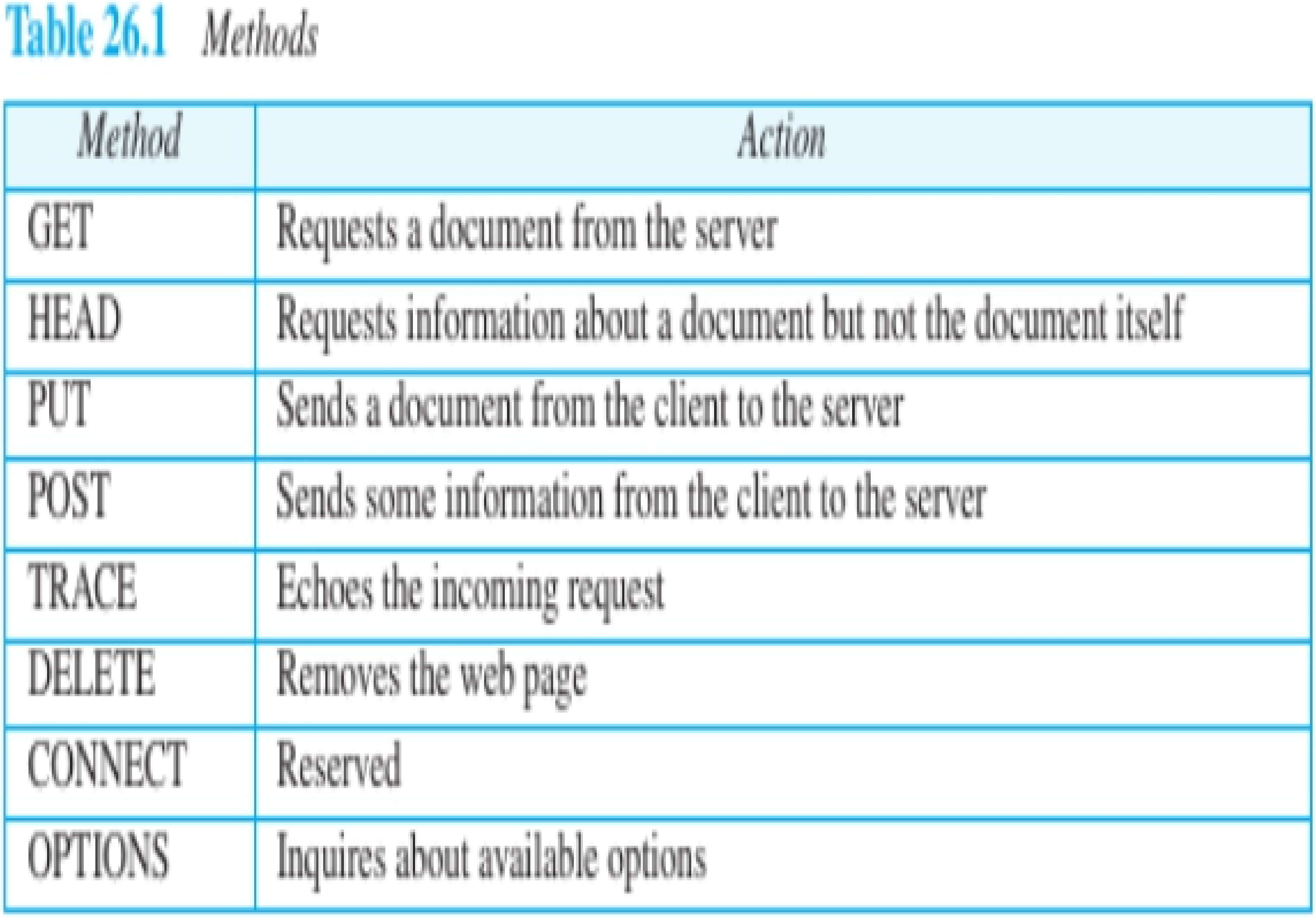


Table 26.1 *Methods*

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

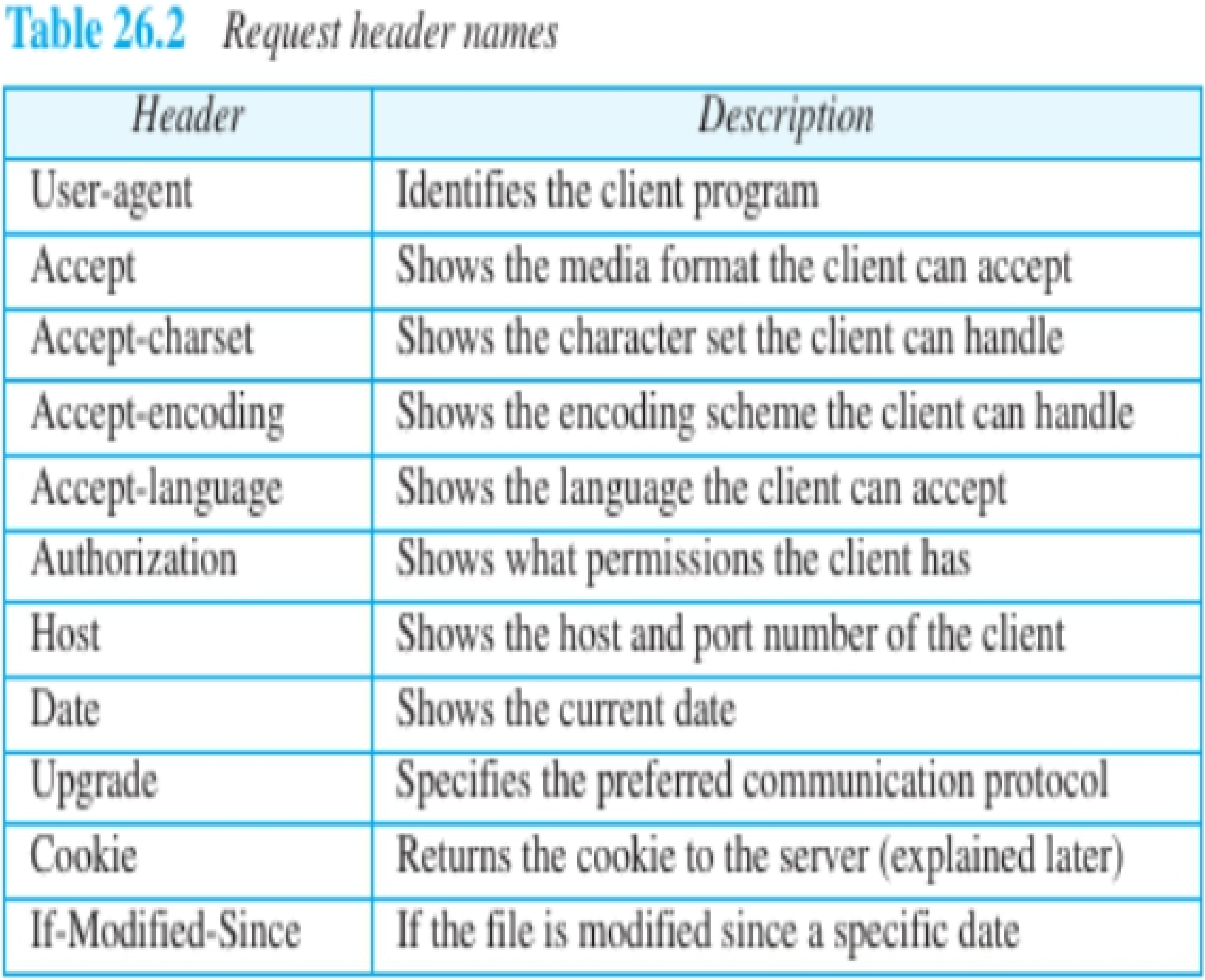


Table 26.2 *Request header names*

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

- **HyperText Transfer Protocol (HTTP)**
- **Response Message**
- The format of the response message is also shown in Figure 26.5.
- 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.

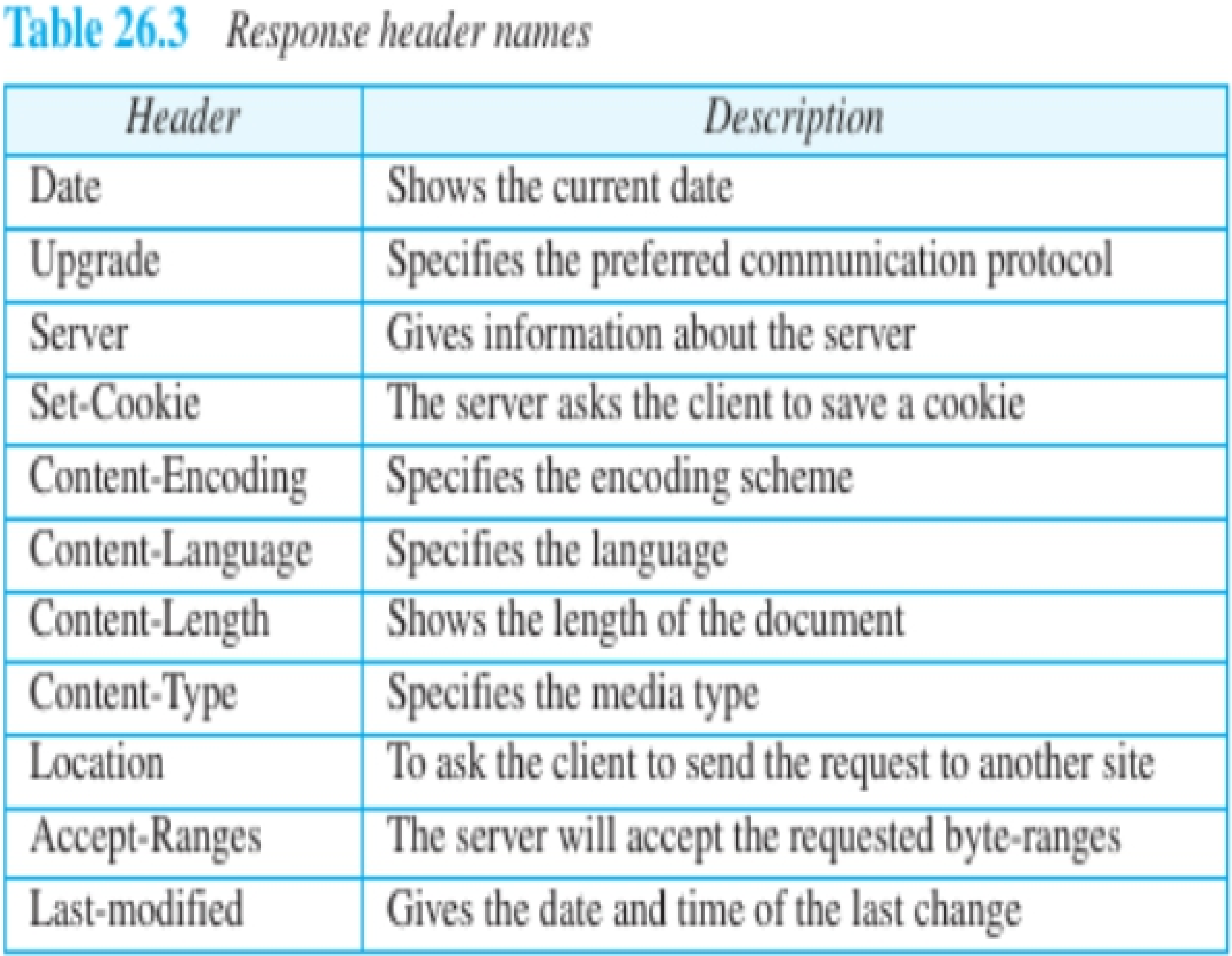
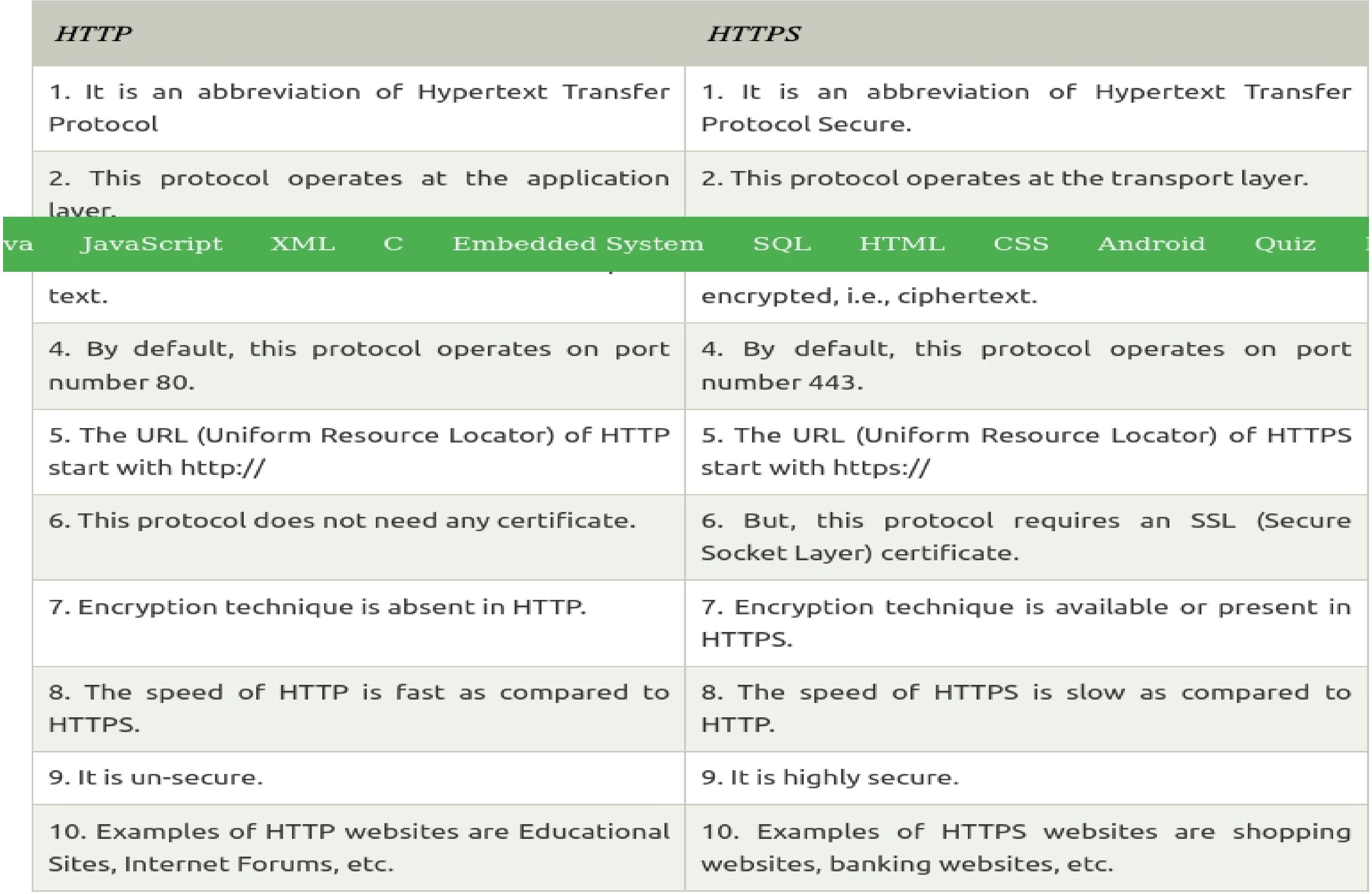


Table 26.3 *Response header names*

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

- **HTTPS**
- HTTPS is an abbreviation of Hypertext Transfer Protocol Secure.
- It is a secure extension or version of HTTP
- This protocol is mainly used for providing security to the data sent between a website and the web browser.
- It is widely used on the internet and used for secure communications.
- This protocol uses the 443 port number for communicating the data.
- This protocol is also called HTTP over SSL because the HTTPS communication protocols are encrypted using the SSL (Secure Socket Layer).
- By default, it is supported by various web browsers.

- **Difference between HTTP and HTTPS**



HTTP	*HTTPS*
1. It is an abbreviation of Hypertext Transfer Protocol	1. It is an abbreviation of Hypertext Transfer Protocol Secure.
2. This protocol operates at the application layer.	2. This protocol operates at the transport layer.
text.	encrypted, i.e., ciphertext.
4. By default, this protocol operates on port number 80.	4. By default, this protocol operates on port number 443.
5. The URL (Uniform Resource Locator) of HTTP start with http://	5. The URL (Uniform Resource Locator) of HTTPS start with https://
6. This protocol does not need any certificate.	6. But, this protocol requires an SSL (Secure Socket Layer) certificate.
7. Encryption technique is absent in HTTP.	7. Encryption technique is available or present in HTTPS.
8. The speed of HTTP is fast as compared to HTTPS.	8. The speed of HTTPS is slow as compared to HTTP.
9. It is un-secure.	9. It is highly secure.
10. Examples of HTTP websites are Educational Sites, Internet Forums, etc.	10. Examples of HTTPS websites are shopping websites, banking websites, etc.

va JavaScript XML C Embedded System SQL HTML CSS Android Quiz

- **Advantages of HTTPS**
- The main advantage of HTTPS is that it provides high security to users.
- Data and information are protected. So, it ensures data protection.
- SSL technology in HTTPS protects the data from third-party or hackers.
- And this technology builds trust for the users who are using it.
- It helps users by performing banking transactions.

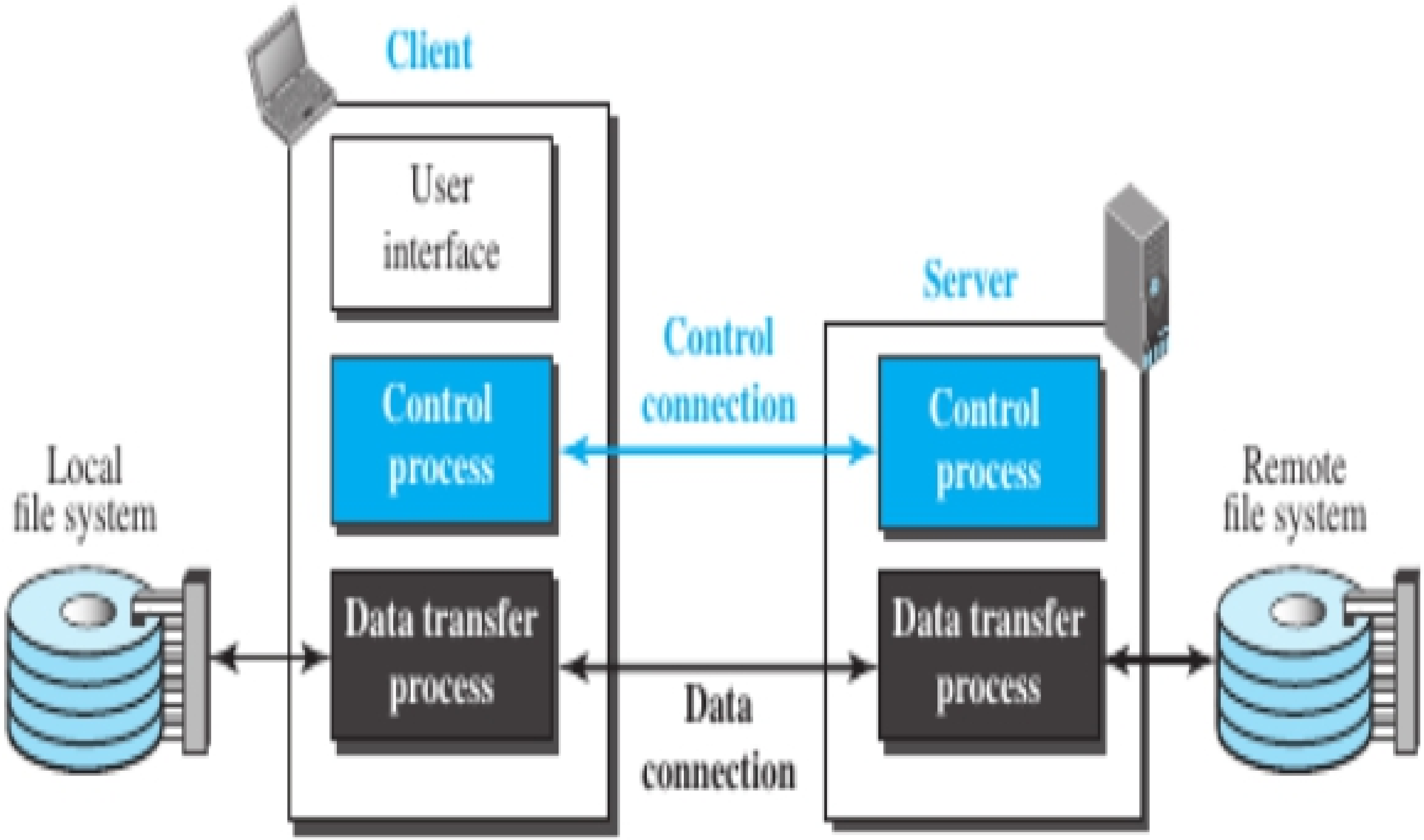
- **Disadvantages of HTTPS**
- The big disadvantage of HTTPS is that users need to purchase the SSL certificate.
- The speed of accessing the website is slow because there are various complexities in communication.
- Users need to update all their internal links.

- **File Transfer Protocol (FTP)**
- File Transfer Protocol (FTP) is the standard protocol provided by TCP/IP for copying a file from one host to another.
- Although transferring files from one system to another seems simple and straightforward, some problems must be dealt with first.
- 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 in a very simple and elegant approach.

- **File Transfer Protocol (FTP)**
- Although we can transfer files using HTTP, FTP is a better choice to transfer large files or to transfer files using different formats.
- Figure 26.10 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.

- **File Transfer Protocol (FTP)**
- Separation of commands and data transfer makes FTP more efficient.
- The control connection uses very simple rules of communication.
- We need to transfer only a line of command or a line of response at a time.
- The data connection, on the other hand, needs more complex rules due to the variety of data types transferred.

Figure 26.10 *FTP*



- **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.
- It opens each time commands that involve transferring files are used, and it closes when the file is transferred.
- In other words, when a user starts an FTP session, the control connection opens.
- While the control connection is open, the data connection can be opened and closed multiple times if several files are transferred.

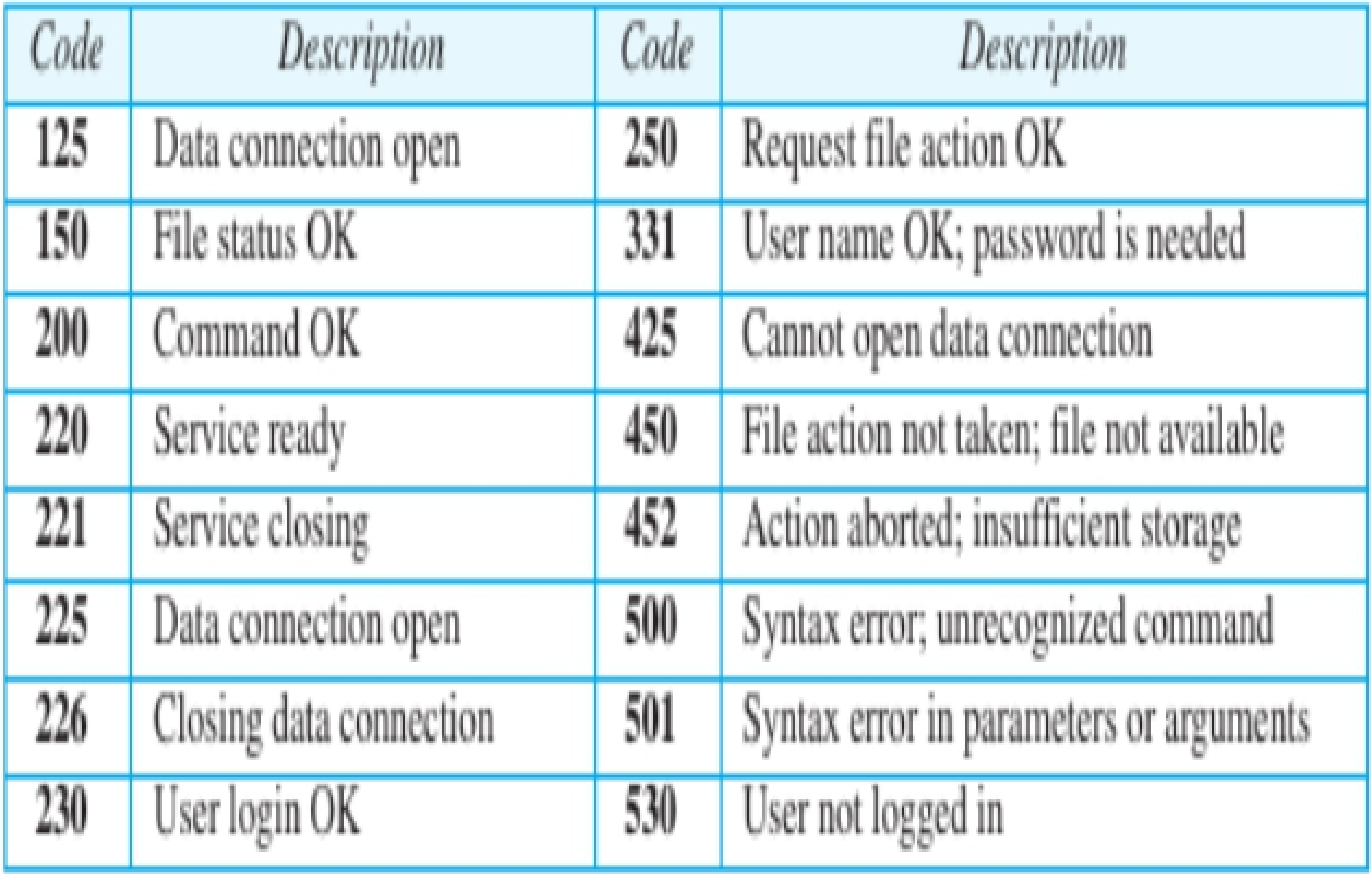
- **Two Connections**
- FTP uses two well-known TCP ports: port 21 is used for the control connection, and port 20 is used for the data connection.
- **Control Connection**
- For control communication, FTP uses the same approach as TELNET.
- It uses the NVT ASCII character set as used by TELNET. Communication is achieved through commands and responses.
- This 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.

- **Two Connections**
- During this control connection, commands are sent from the client to the server and responses are sent from the server to the client.
- Commands, which are sent from the FTP client control process, are in the form of ASCII uppercase, which may or may not be followed by an argument.

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

Table 26.5 *Some responses in FTP*

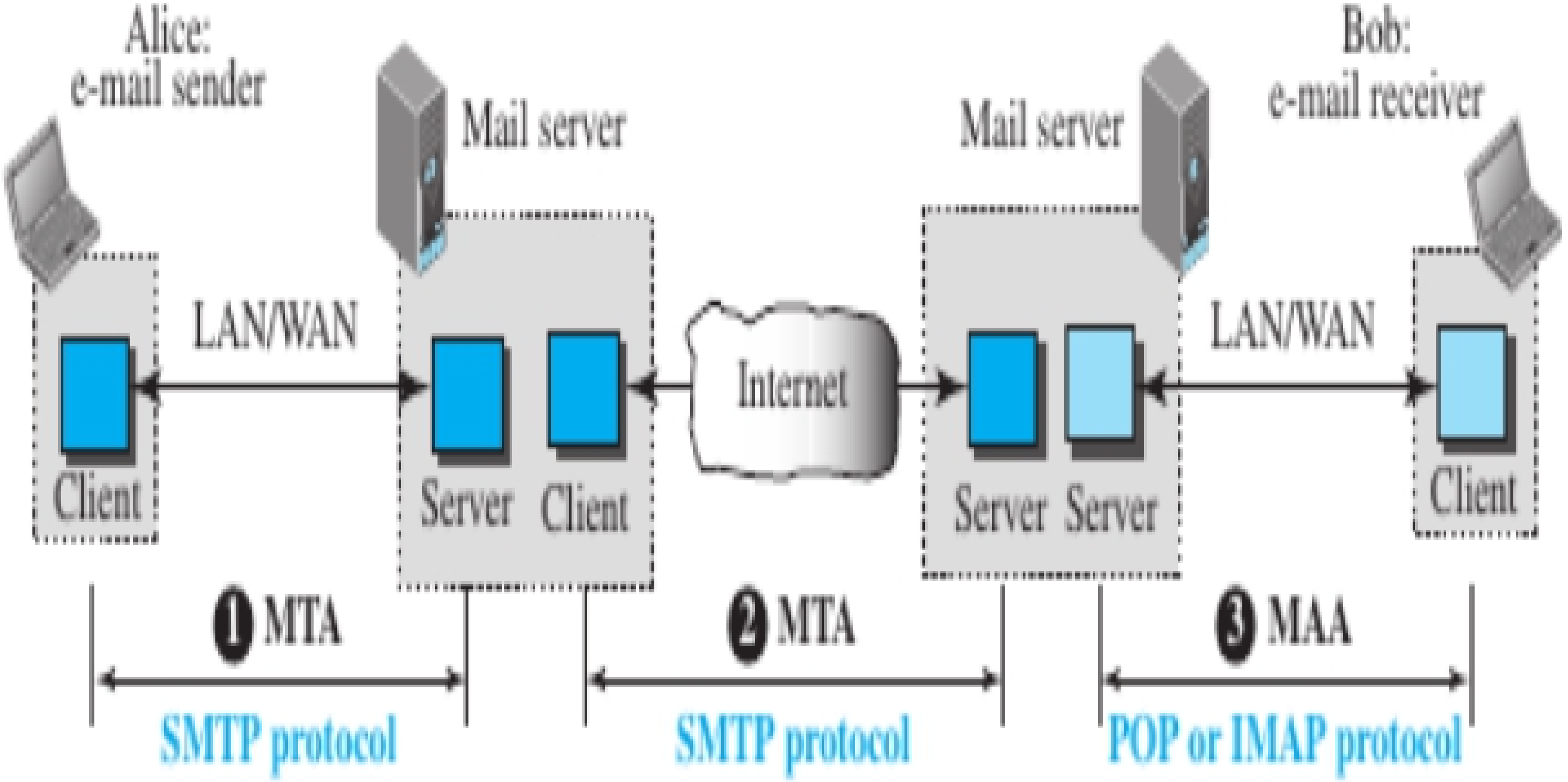


Code	*Description*	*Code*	*Description*
125	Data connection open	**250**	Request file action OK
150	File status OK	**331**	User name OK; password is needed
200	Command OK	**425**	Cannot open data connection
220	Service ready	**450**	File action not taken; file not available
221	Service closing	**452**	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

- **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:
- 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 well-known port 20 and the received ephemeral port number.

- **SMTP**
- Based on the common scenario, we can say that the e-mail is one of those applications that needs three uses of client-server paradigms to accomplish its task.
- It is important that we distinguish these three when we are dealing with e-mail.
- Figure 26.15 shows these three client-server applications.
- We refer to the first and the second as Message Transfer Agents (MTAs), the third as Message Access Agent (MAA).

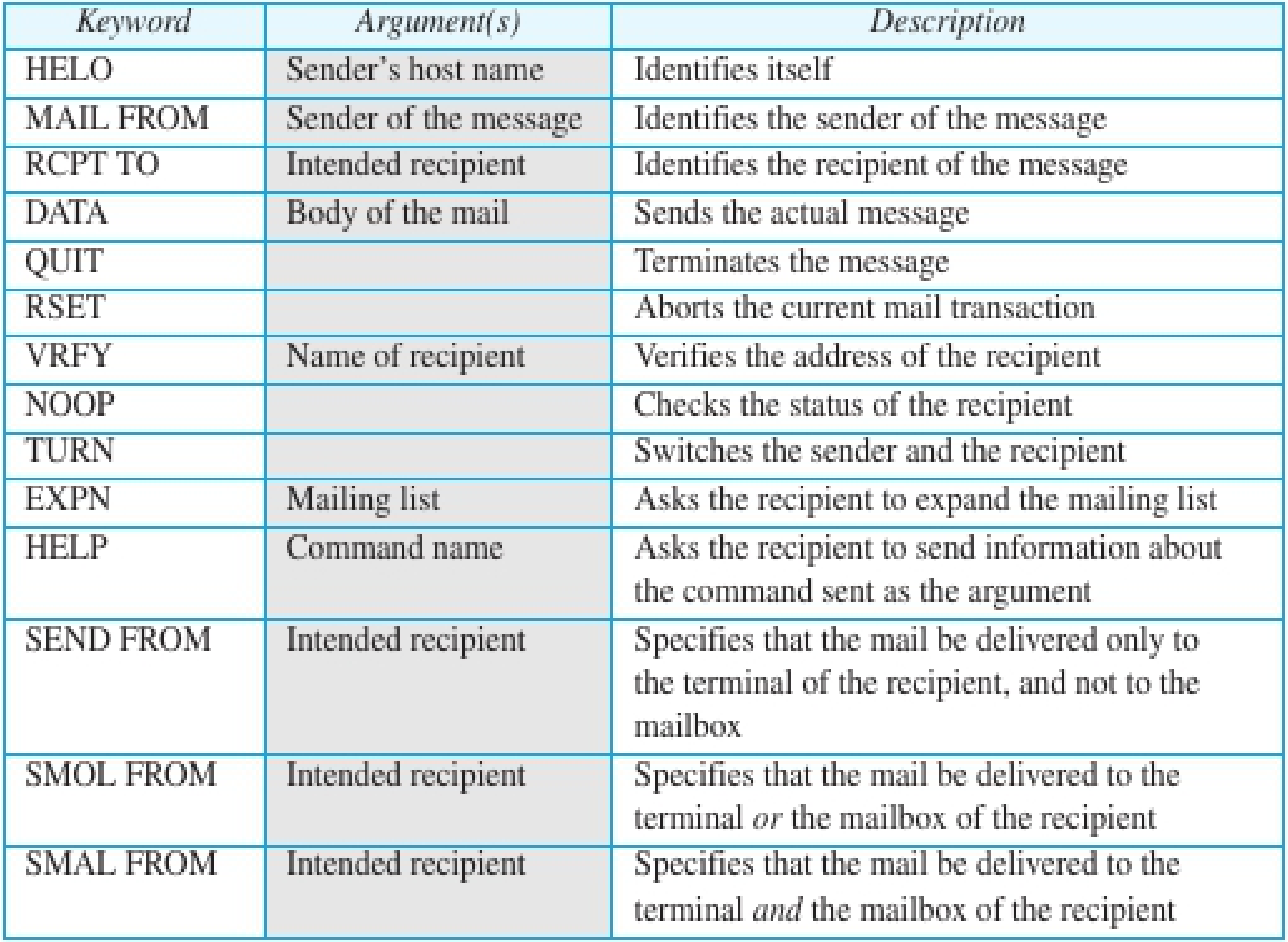
Figure 26.15 *Protocols used in electronic mail*



- **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.
- As we will see shortly, another protocol is needed between the mail server and the receiver.
- SMTP simply defines how commands and responses must be sent back and forth.

- **SMTP**
- **Commands and Responses**
- SMTP uses commands and responses to transfer messages between an MTA client and an MTA server.
- The command is from an MTA client to an MTA server; the response is from an MTA server to the MTA client.
- Each command or reply is terminated by a two-character (carriage return and line feed) end-of-line token.
- **Commands:** Commands are sent from the client to the server.
- The format of a command is shown below:
- It consists of a keyword followed by zero or more arguments. SMTP defines 14 commands, listed in Table 26.6.

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

- **SMTP**
- **Responses:** Responses are sent from the server to the client.
- A response is a three-digit code that may be followed by additional textual information.
- Table 26.7 shows the most common response types.

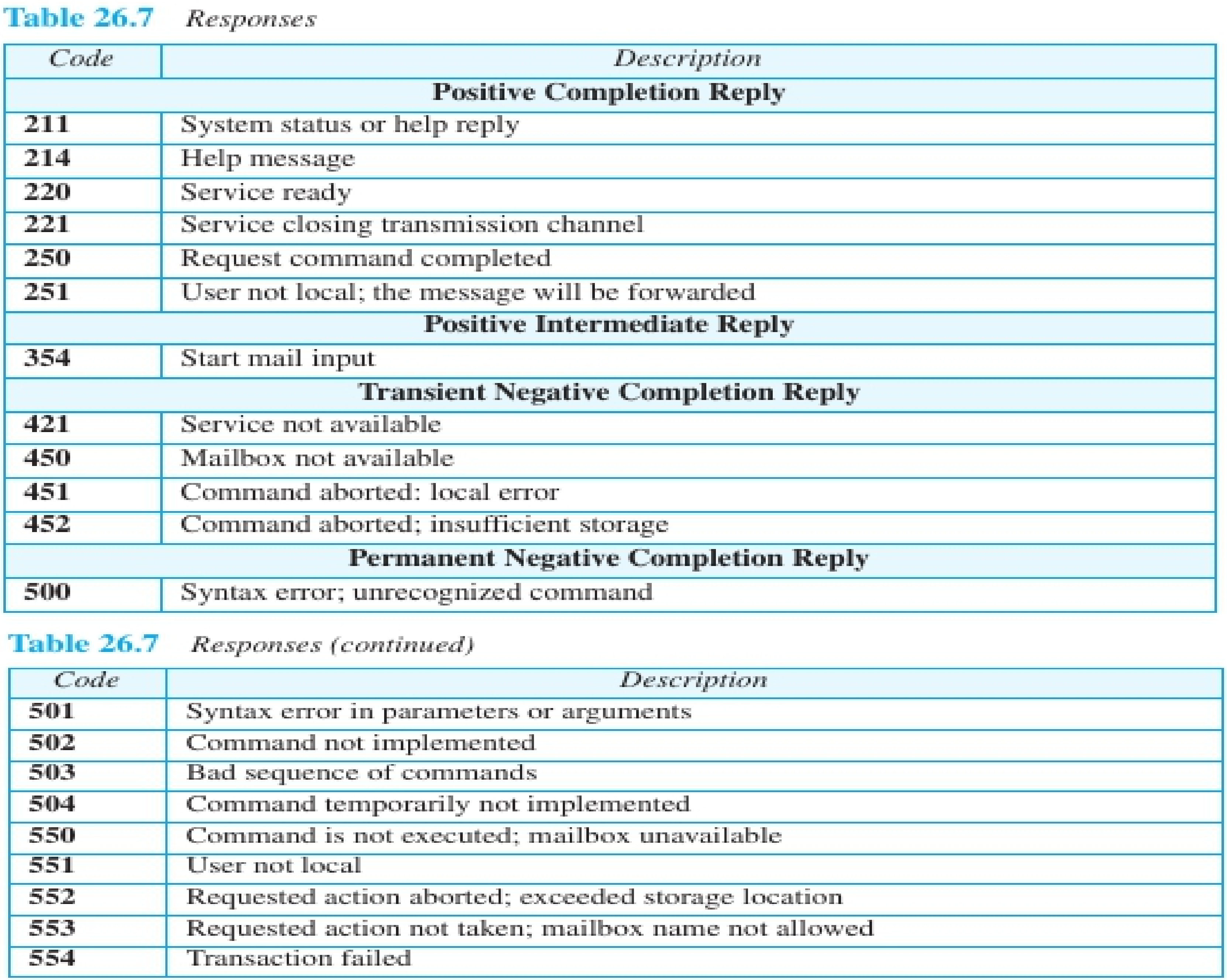


Table 26.7 *Responses*

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

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

- **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 well-known 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.

- **Mail Transfer Phases**
- 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.

- **Mail Transfer Phases**
- **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.

- **Mail Transfer Phases**
- 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.

- **Mail Transfer Phases**
- 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.

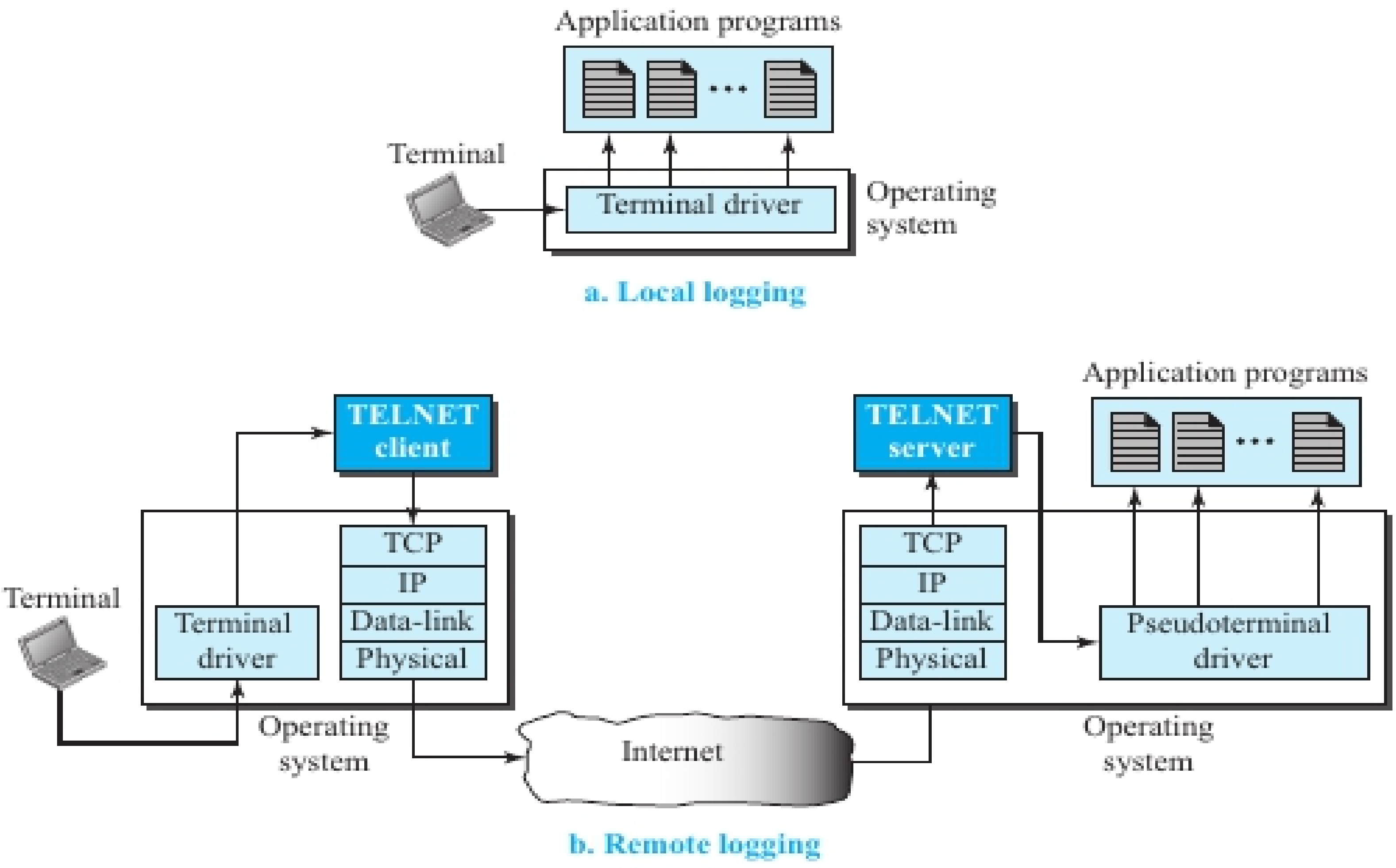
- **TELNET**
- A server program can provide a specific service to its corresponding client program.
- For example, the FTP server is designed to let the FTP client store or retrieve files on the server site.
- However, it is impossible to have a client/server pair for each type of service we need; the number of servers soon becomes intractable.
- The idea is not scalable.
- Another solution is to have a specific client/server program for a set of common scenarios, but to have some generic client/server programs that allow a user on the client site to log into the computer at the server site and use the services available there.

- **TELNET**
- For example, if a student needs to use the Java compiler program at her university lab, there is no need for a Java compiler client and a Java compiler server.
- The student can use a client logging program to log into the university server and use the compiler program at the university.
- We refer to these generic client/server pairs as remote logging applications.
- 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 (not encrypted).

- **TELNET**
- A hacker can eavesdrop and obtain the logging name and password.
- Because of this security issue, the use of TELNET has diminished in favor of another protocol, Secure Shell (SSH), Although TELNET is almost replaced by SSH, we briefly discuss TELNET here for two reasons:
- 1. The simple plaintext architecture of TELNET allows us to explain the issues and challenges related to the concept of remote logging, which is also used in SSH when it serves as a remote logging protocol.
- 2. Network administrators often use TELNET for diagnostic and debugging purposes.

- **Local versus Remote Logging**

Figure 26.23 *Local versus remote logging*



a. Local logging

b. Remote logging

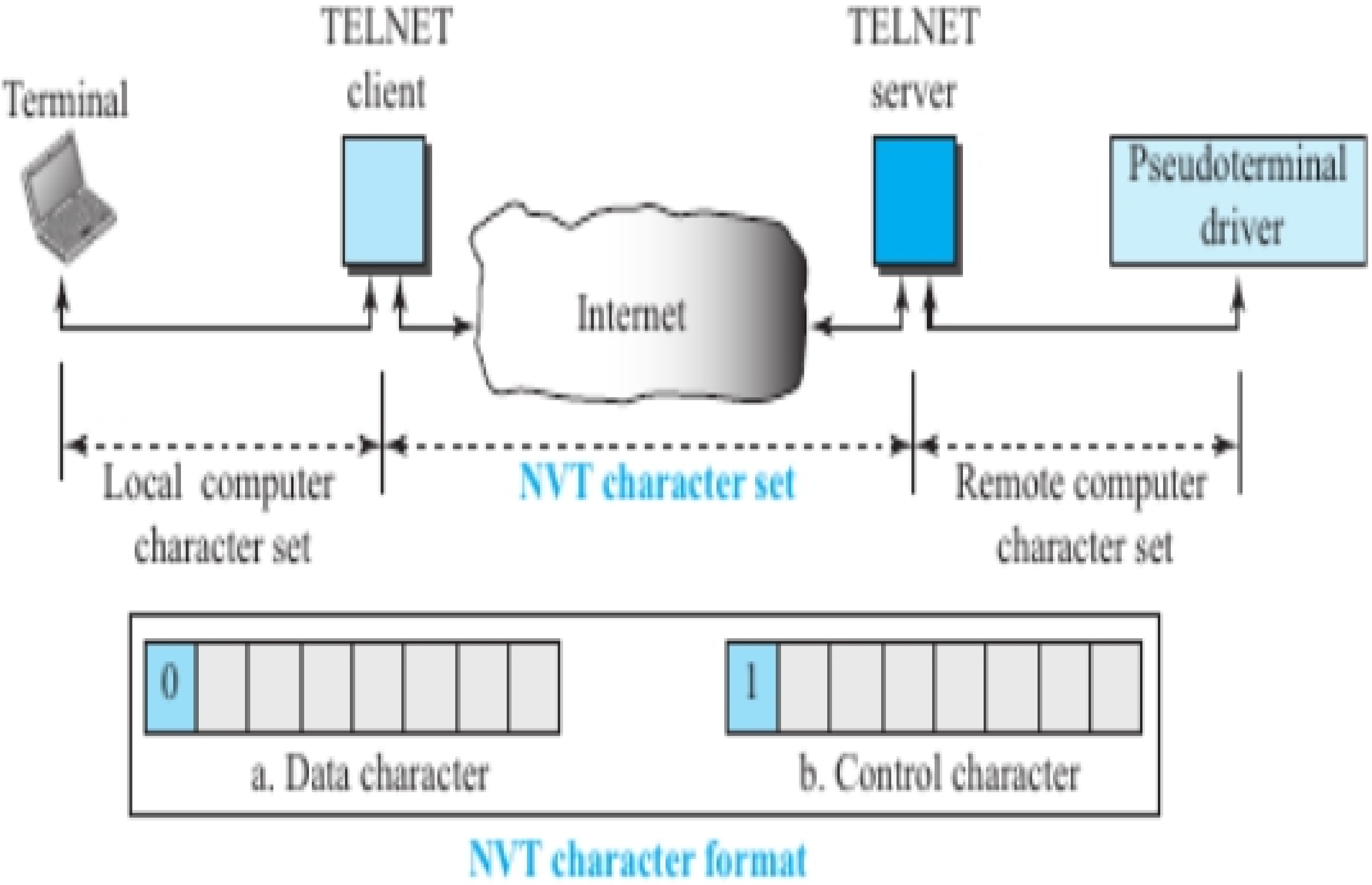
- **TELNET**
- When a user logs into a local system, it **is called local logging.**
- As a user types at a terminal or at a workstation running a terminal emulator, the keystrokes are accepted by the terminal driver.
- The terminal driver passes the characters to the operating system.
- The operating system, in turn, interprets the combination of characters and invokes the desired application program or utility.

- **TELNET**
- However, 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.

- **Network Virtual Terminal (NVT)**
- The mechanism to access a remote computer is complex.
- This is because every computer and its operating system accepts a special combination of characters as tokens.
- For example, the end-of-file token in a computer running the DOS operating system is Ctrl+z, while the UNIX operating system recognizes Ctrl+d.
- We are dealing with heterogeneous systems.
- If we want to access any remote computer in the world, we must first know what type of computer we will be connected to, and we must also install the specific terminal emulator used by that computer.

- **Network Virtual Terminal (NVT)**
- TELNET solves this problem by defining a universal interface called the Network Virtual Terminal (NVT) character set.
- Via this interface, the client TELNET translates characters (data or commands) that come from the local terminal into NVT form and delivers them to the network.
- The server TELNET, on the other hand, translates data and commands from NVT form into the form acceptable by the remote computer.
- Figure 26.24 shows the concept.

Figure 26.24 *Concept of NVT*

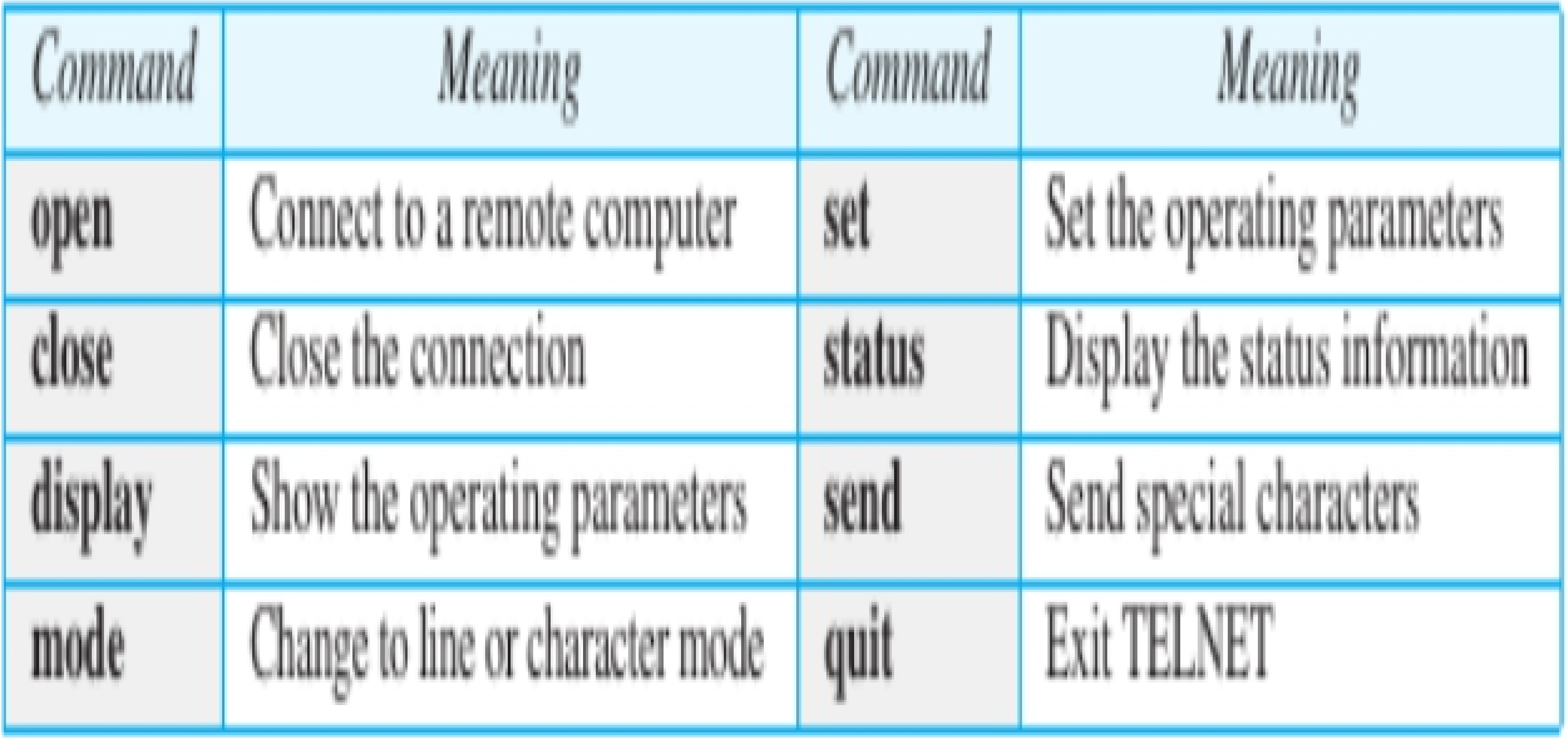


- **Network Virtual Terminal (NVT)**
- NVT uses two sets of characters, one for data and one for control.
- Both are 8-bit bytes as shown in Figure 26.24.
- For data, NVT normally uses what is called NVT ASCII.
- This is an 8-bit character set in which the seven lowest order bits are the same as US ASCII and the highest order bit is 0.
- To send control characters between computers (from client to server or vice versa), NVT uses an 8-bit character set in which the highest order bit is set to 1.

- **Network Virtual Terminal (NVT)**
- NVT uses two sets of characters, one for data and one for control.
- Both are 8-bit bytes as shown in Figure 26.24.
- For data, NVT normally uses what is called NVT ASCII.
- This is an 8-bit character set in which the seven lowest order bits are the same as US ASCII and the highest order bit is 0.
- To send control characters between computers (from client to server or vice versa), NVT uses an 8-bit character set in which the highest order bit is set to 1.

- **Options**
- TELNET lets the client and server negotiate options before or during the use of the service.
- Options are extra features available to a user with a more sophisticated terminal.
- Users with simpler terminals can use default features.
- **User Interface**
- The operating system (UNIX, for example) defines an interface with user-friendly commands.
- An example of such a set of commands can be found in Table 26.11.

Table 26.11 *Examples of interface commands*

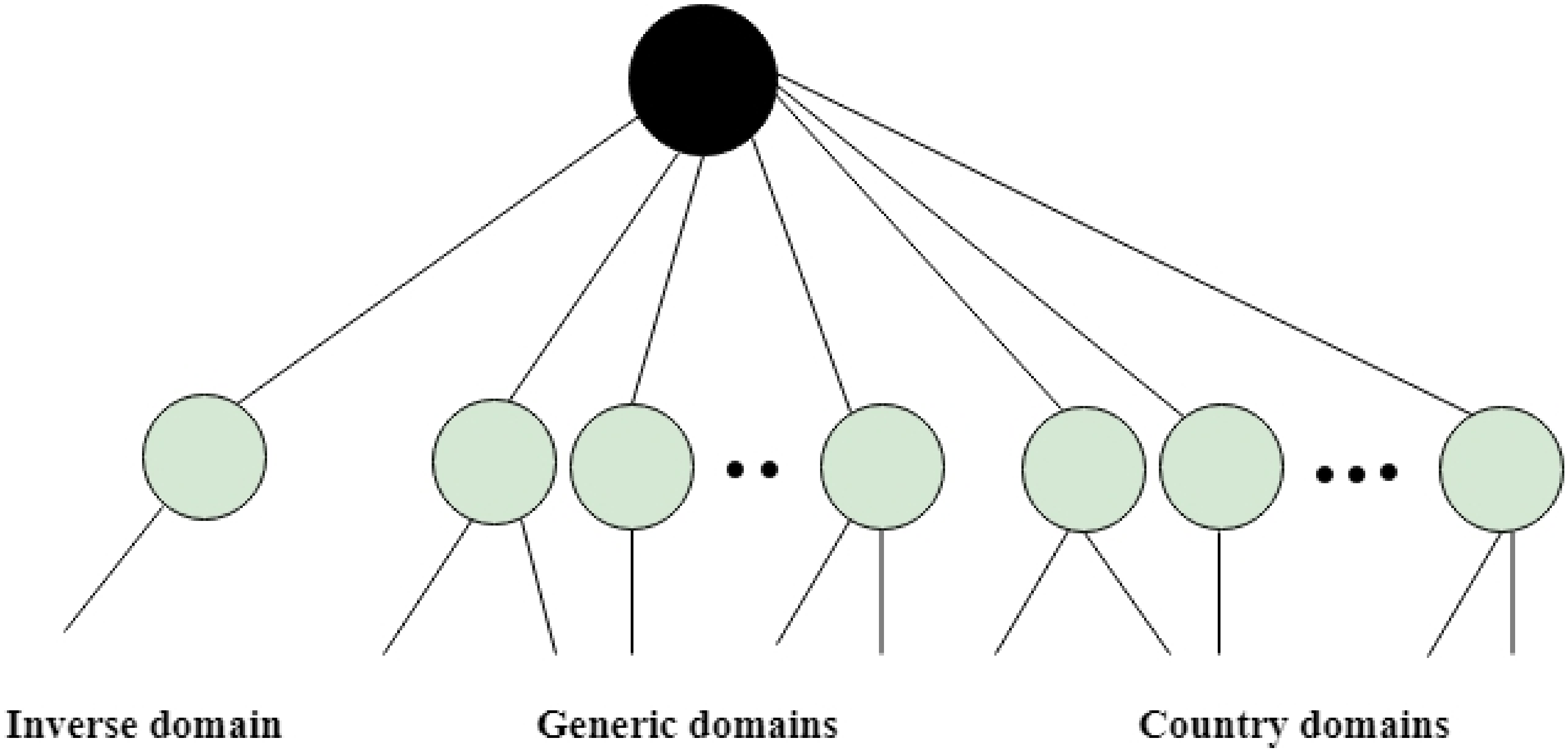


Command	*Meaning*	*Command*	*Meaning*
open	Connect to a remote computer	**set**	Set the operating parameters
close	Close the connection	**status**	Display the status information
display	Show the operating parameters	**send**	Send special characters
mode	Change to line or character mode	**quit**	Exit TELNET

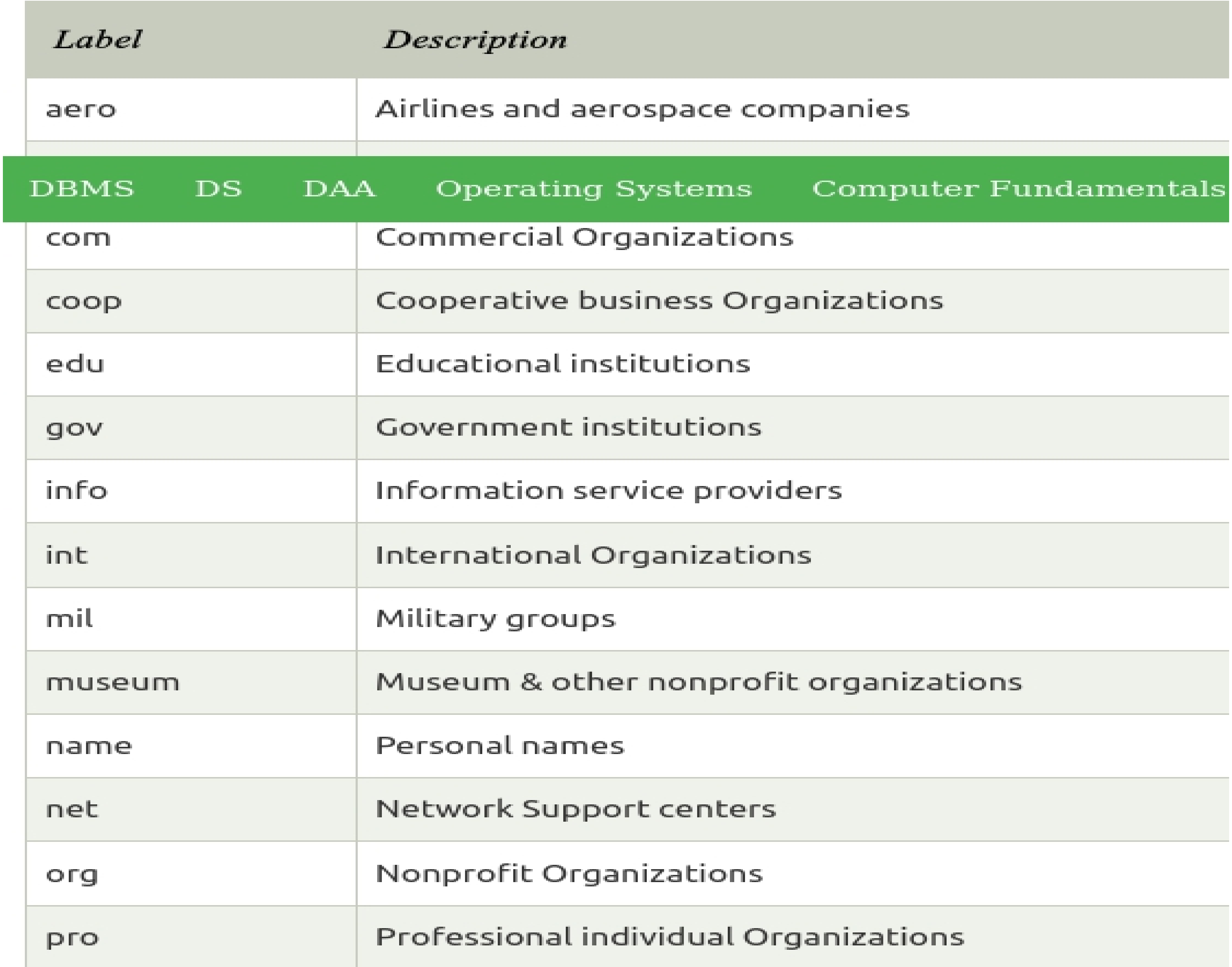
- **DNS**
- An application layer protocol defines how the application processes running on different systems, pass the messages to each other.
- DNS stands for Domain Name System.
- DNS is a directory service that provides a mapping between the name of a host on the network and its numerical address.
- DNS is required for the functioning of the internet.
- Each node in a tree has a domain name, and a full domain name is a sequence of symbols specified by dots.

- **DNS**
- DNS is a service that translates the domain name into IP addresses. This allows the users of networks to utilize user-friendly names when looking for other hosts instead of remembering the IP addresses.
- For example, suppose the FTP site at EduSoft had an IP address of 132.147.165.50, most people would reach this site by specifying ftp.EduSoft.com. Therefore, the domain name is more reliable than IP address.

- **DNS**
- DNS is a TCP/IP protocol used on different platforms.
- The domain name space is divided into three different sections: generic domains, country domains, and inverse domain.

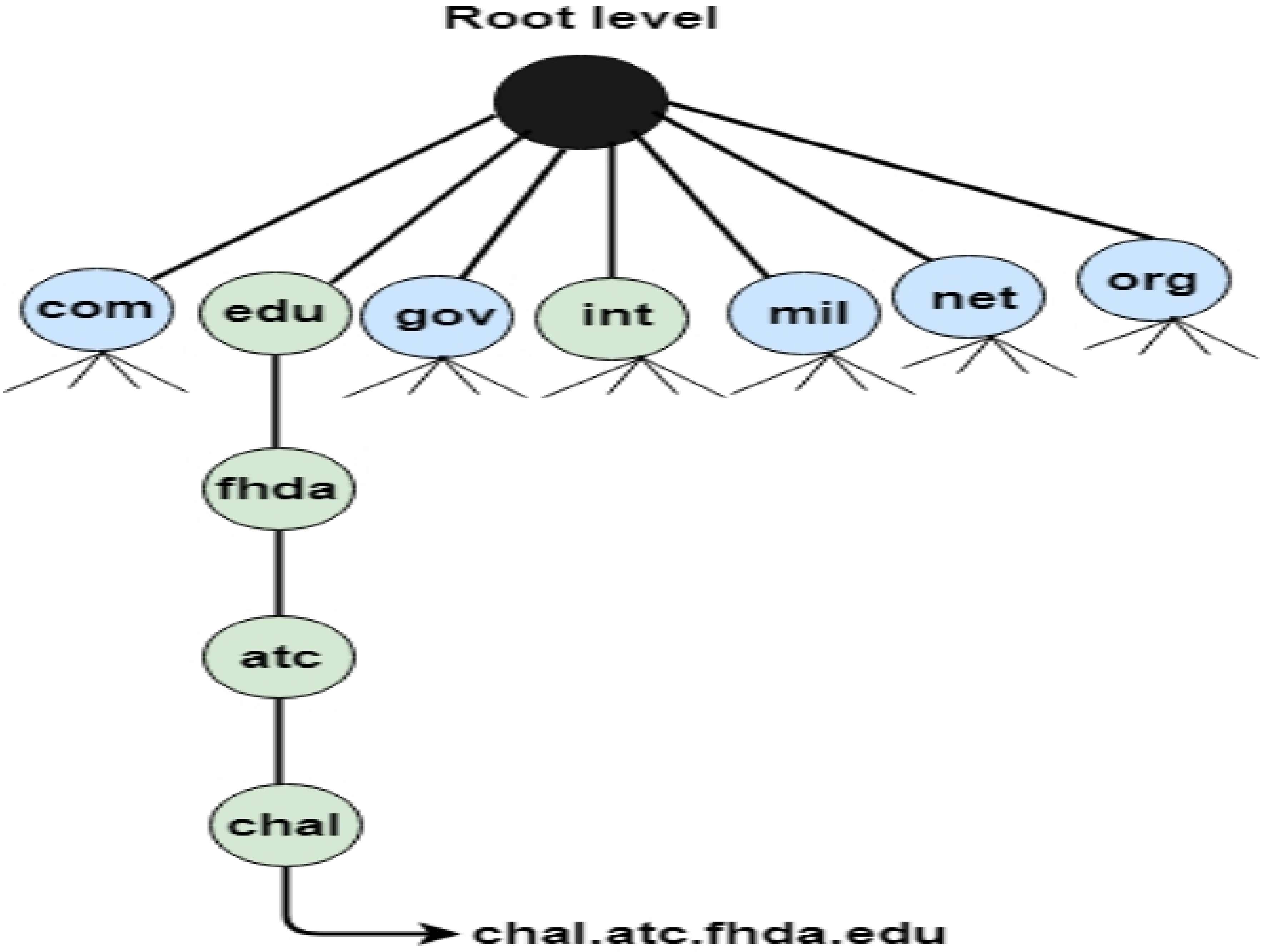


- **DNS**
- **Generic Domains**
- It defines the registered hosts according to their generic behavior.
- Each node in a tree defines the domain name, which is an index to the DNS database.
- It uses three-character labels, and these labels describe the organization type.



DBMS DS DAA Operating Systems Computer Fundamentals

Label	*Description*
aero	Airlines and aerospace companies
com	Commercial Organizations
coop	Cooperative business Organizations
edu	Educational institutions
gov	Government institutions
info	Information service providers
int	International Organizations
mil	Military groups
museum	Museum & other nonprofit organizations
name	Personal names
net	Network Support centers
org	Nonprofit Organizations
pro	Professional individual Organizations


Root level
com
edu
gov
int
mil
net
org
fhda
atc
chal
chal.atc.fhda.edu

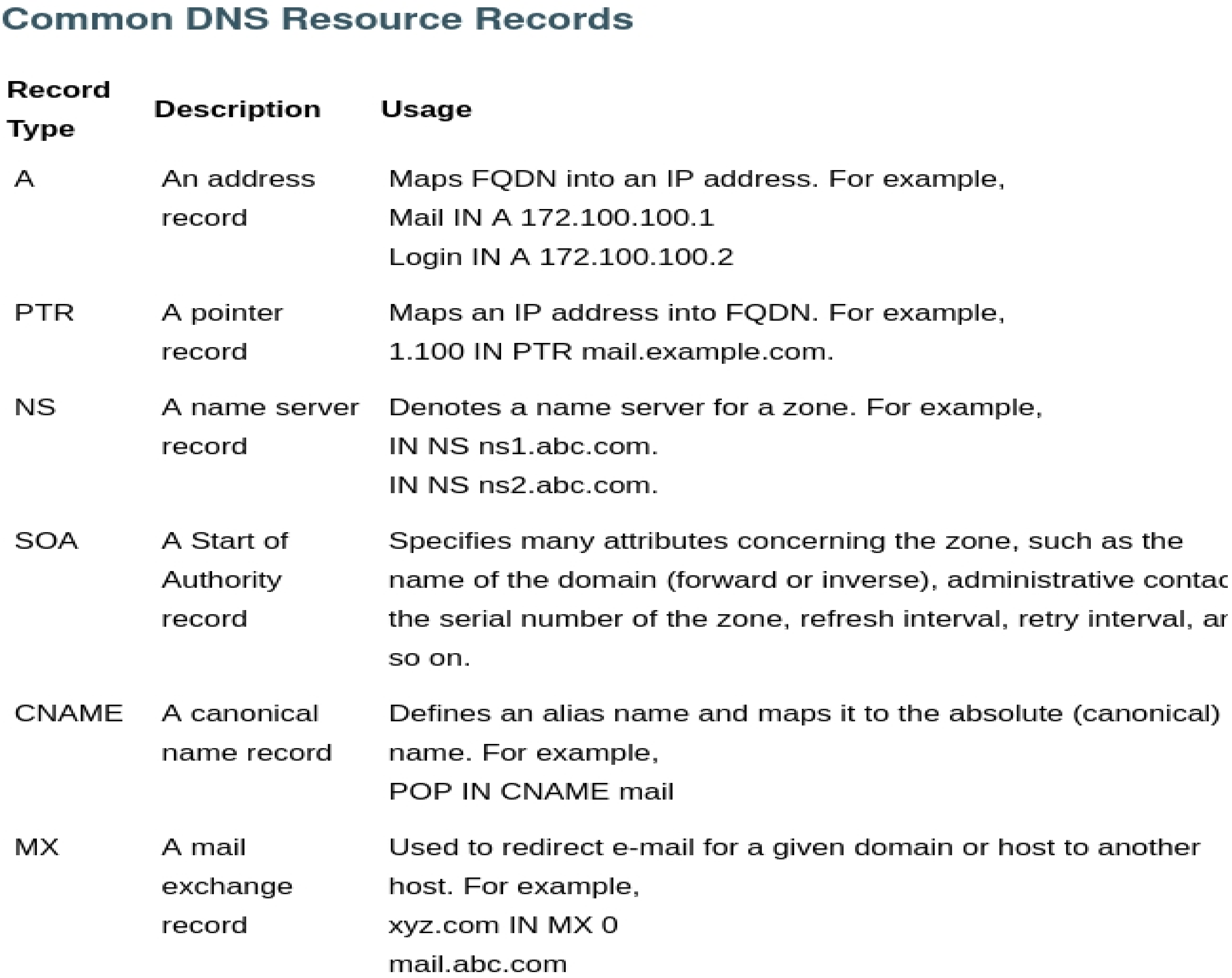
- **DNS**
- **Country Domain**
- The format of country domain is same as a generic domain, but it uses two-character country abbreviations (e.g., us for the United States) in place of three character organizational abbreviations.
- **Inverse Domain**
- The inverse domain is used for mapping an address to a name.
- When the server has received a request from the client, and the server contains the files of only authorized clients.
- To determine whether the client is on the authorized list or not, it sends a query to the DNS server and ask for mapping an address to the name.

- **Working of DNS**
- DNS is a client/server network communication protocol.
- DNS clients send requests to the. server while DNS servers send responses to the client.
- Client requests contain a name which is converted into an IP address known as a forward DNS lookups while requests containing an IP address which is converted into a name known as reverse DNS lookups.
- DNS implements a distributed database to store the name of all the hosts available on the internet.

- **Working of DNS**
- If a client like a web browser sends a request containing a hostname, then a piece of software such as DNS resolver sends a request to the DNS server to obtain the IP address of a hostname.
- If DNS server does not contain the IP address associated with a hostname, then it forwards the request to another DNS server.
- If IP address has arrived at the resolver, which in turn completes the request over the internet protocol.

- **DNS Resource Records (RR)**
- A DNS resource record (RR) contains all the information about a domain name system.
- It defines all the attributes for a domain name such as an IP address or a mail route.
- A DNS RR has six fields:
- **<NAME>, <TYPE>, <CLASS>, <TTL>, <RD Length>, and <RDATA>**
- These fields are explained in the following list:
- **Name.** This field specifies the DNS name, also known as the owner name, to which the RR belongs.
- **Type.** This field is a 2-byte value that specifies the type of the resource that is defined in the resource record. This field is necessary because a DNS name can have more than one type of

- **DNS Resource Records (RR)**
- **Class.** This defines the protocol family for the RR record. For example, IN, which stands for Internet.
- **Time To Live (TTL).** This field is the time, in seconds, for which a name server can cache an RR. A zero TTL implies that a server should not cache the RR.
- **RD Length.** This field is the RDATA field's length in octets.
- **RDATA.** This field is a resource data field and is the value to which the entity specified in the NAME field maps. It is unique for each type of RR.



Common DNS Resource Records

Record Type	Description	Usage
A	An address record	Maps FQDN into an IP address. For example, Mail IN A 172.100.100.1 Login IN A 172.100.100.2
PTR	A pointer record	Maps an IP address into FQDN. For example, 1.100 IN PTR mail.example.com.
NS	A name server record	Denotes a name server for a zone. For example, IN NS ns1.abc.com. IN NS ns2.abc.com.
SOA	A Start of Authority record	Specifies many attributes concerning the zone, such as the name of the domain (forward or inverse), administrative contac the serial number of the zone, refresh interval, retry interval, ar so on.
CNAME	A canonical name record	Defines an alias name and maps it to the absolute (canonical) name. For example, POP IN CNAME mail
MX	A mail exchange record	Used to redirect e-mail for a given domain or host to another host. For example, xyz.com IN MX 0 mail.abc.com

- **DNS:Types of Name Server**
- Name servers are the servers that make up DNS.
- They hold the records of multiple DNS types and translate a URL into an IP address.
- DNS name servers are the critical component of how DNS works, and they help direct traffic on the internet.
- There are four types of name servers that make up DNS:
  - Recursive (also known as resolver) server
  - Root name server
  - TLD name server
  - Authoritative server

- **DNS Recursive Server**
- A recursive server is usually operated by your internet service provider (ISP) or wireless carrier.
- If the website isn't cached in this server (usually by another user who has visited the website), then the query heads to a root server.
- **Root Name Server**
- The root server holds information about top-level domains (TLDs), including .com, .org, and .net.
- There are only 13 sets of root servers in the world, and they are operated by organizations like NASA and companies like Verisign.
- Once the request goes to the root server, it will respond with the TLD name server.

- **TLD Name Server**
- Once your query knows which name server to go to, it will visit a TLD name server for the information in the second-level domain (the “Bluehost” in Bluehost.com).
- The .com server will tell the request where to go to find the IP address for the website you want to reach.
- It will point to the authoritative server, the final step in the journey.

- **Authoritative Server**
- The authoritative server houses the website’s IP address for the full domain.
- Once requested, it then sends that information back to the recursive server, which sends it to your device.
- In a nutshell, your query goes back and forth between multiple servers until it has located all the information it needs to get you to that website.
- While elaborate, this process takes only a few seconds.