R.M.K GROUP OF ENGINEERING INSTITUTIONS

R.M.K
GROUP OF
INSTITUTIONS

R.M.K
GROUP OF INSTITUTIONS

R.M.K ENGINEERING COLLEGE

R.M.D ENGINEERING COLLEGE

R.M.K.C.E.T

அறிவே ஆக்கம்

R.M.K
GROUP OF
INSTITUTIONS

Please read this disclaimer before proceeding:

This document is confidential and intended solely for the educational purpose of RMK Group of Educational Institutions. If you have received this document through email in error, please notify the system manager. This document contains proprietary information and is intended only to the respective group / learning community as intended. If you are not the addressee you should not disseminate, distribute or copy through e-mail. Please notify the sender immediately by e-mail if you have received this document by mistake and delete this document from your system. If you are not the intended recipient you are notified that disclosing, copying, distributing or taking any action in reliance on the contents of this information is strictly prohibited.

DIGITAL NOTES ON 20CS501

COMPUTER NETWORKS

Department	**: Computer Science and Engineering**
Batch/Year	**: 2020-2024/III**
Created by	**: Ms.Srijayanthi,AP/ADS** **Ms. K.RAMYA DEVI,AP/CSE** **Mr. KINGSLEY,AP/CSE**
Date	**: 10.08.2022**

Table of Contents

COURSE OBJECTIVES

- To understand the protocol layering and physical level communication.
- To analyze the performance of a network.
- To understand the various components required to build different networks.
- To learn the functions of network layer and the various routing protocols.
- To familiarize the functions and protocols of the Transport layer.

PREREQUISITE

- IT8201 INFORMATION TECHNOLOGY ESSENTIALS
- EC8394 ANALOG AND DIGITAL COMMUNICATION

SYLLABUS

20CS501 COMPUTER NETWORKS 3 0 0 3

UNIT I INTRODUCTION AND PHYSICAL LAYER 9

Data Communications – Network Types – Protocol Layering – Network Models (OSI, TCP/IP) Networking Devices: Hubs, Bridges, Switches – Performance Metrics – Transmission media - Guided media -Unguided media- Switching-Circuit Switching - Packet Switching.

UNIT II DATA-LINK LAYER & MEDIA ACCESS 11

Introduction – Link-Layer Addressing- Error Detection and Correction - DLC Services – Data Link Layer Protocols – HDLC – PPP - Wired LANs: Ethernet - Wireless LANs – Introduction – IEEE 802.11, Bluetooth

UNIT III NETWORK LAYER 9

Network Layer Services – Packet switching – Performance – IPV4 Addresses – Forwarding of IP Packets - Network Layer Protocols: IP, ICMP v4 – Unicast Routing Algorithms – Protocols – Multicasting Basics – IPV6 Addressing – IPV6 Protocol.

UNIT IV TRANSPORT LAYER 8

Introduction – Transport Layer Protocols – Services – Port Numbers – User Datagram Protocol –Transmission Control Protocol – SCTP.

UNIT V APPLICATION LAYER 8

Application layer-WWW and HTTP – FTP – Email –Telnet –SSH – DNS – SNMP.

Course outcomes

Course Code	Course Outcome Statement	Cognitive/Affective Level of the Course Outcome	Expected Level of Attainment
Course Outcome Statements in Cognitive Domain			
C302.1	Explain the basic layers and its functions, and transmission media in computer networks	Understand K2	60%
C302.2	Examine the performance of different types of networks	Analyse K4	60%
C302.3	Inspect the functionalities of data link and media access control protocols	Analyse K4	60%
C302.4	Examine different routing algorithms	Analyse K4	60%
C302.5	Identify appropriate protocol to be used at the transport layer	Apply K3	60%
C302.6	Explain the working of various application layer protocols.	Understand K2	60%
Course Outcome Statements in Affective domain			
C302.7	Attend the classes regularly	Respond (A2)	95%
C302.8	Submit the Assignments regularly.	Respond (A2)	95%
C302.9	Participation in Seminar/Quiz/ Group Discussion/ Collaborative learning and content beyond syllabus	Valuing (A3)	95%

CO- PO/PSO Mapping

Overall Correlation Matrix of the Course as per Anna University Curriculum

Course Code	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
C302	3	1	2									

Correlation Matrix of the Course Outcomes to Programme Outcomes and Programme Specific Outcomes Including Course Enrichment Activities

Course Outcomes (COs)		Programme Outcomes (POs), Programme Specific Outcomes (PSOs)														
		PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12	PSO1	PSO2	PSO3
		K3	K4	K5	K5	K3 /K5	A2	A3	A3	A3	A3	A3	A2	K3	K3	K3
C302 .1	K2	2	1											2	2	2
C302 .2	K4	3	3	2	2									3	3	3
C302 .3	K4	3	3	2	2									3	3	3
C302 .4	K4	3	2	2	2									3	3	3
C302 .5	K3	3	2	1	1									3	3	3
C302 .6	K2	2	1											2	2	2
C302 .7	A2												3			
C302 .8	A2								2	2	2		3			
C302 .9	A3						3	3		3	3		3			
C302		3	3	2	2		1	1	1	3	3		3	3	3	3

LECTURE PLAN

UNIT – V

S No	Topics	No of periods	Proposed date	Actual Lecture Date	pertaining CO	Taxonomy level	Mode of delivery
1	WWW	1	9.11.2022		CO5	K1	Chalk & Talk
2	HTTP	1	10.11.2022		CO5	K1	ICT Tools
3	FTP	1	11.11.2022		CO5	K2	ICT Tools
4	Email	1	12.11.2022		CO5	K2	ICT Tools
5	Telnet	1	16.11.2022		CO5	K2	ICT Tools
6	SSH	1	17.11.2022		CO5	K3	Chalk & Talk
7	DNS	1	18.11.2022		CO5	K2	Chalk & Talk
8	SNMP	1	23.11.2022		CO5	K2	Chalk & Talk
9	(Content Beyond Syllabus)	1	24.11.2022		CO5	K2	ICT Tools

UNIT V APPLICATION LAYER

WWW and HTTP – FTP – Email –Telnet –SSH – DNS – SNMP.

5.1 World Wide Web [WWW]

- The idea of the Web was first proposed by Tim Berners-Lee in 1989 at CERN, the European Organization for Nuclear Research, to allow several researchers at different locations throughout Europe to access each others' researches.

- The commercial Web started in the early 1990s.

- The Web today is a repository of information in which the documents, called web pages, are distributed all over the world and related documents are linked together.

- The popularity and growth of the Web can be related to two terms in the above statement: distributed and linked.

Distributed

- Distribution allows the growth of the Web.

- Each web server in the world can add a new web page to the repository and announce it to all Internet users without overloading a few servers.

Linked

- Linking allows one web page to refer to another web page stored in another server somewhere else in the world.

- The linking of web pages was achieved using a concept called hypertext, which was introduced many years before the advent of the Internet.

- The term hypertext, coined to mean linked text documents, has been changed to hypermedia, to show that a web page can be a text document, an image, an audio file, or a video file.

Uses

- The purpose of the Web has gone beyond the simple retrieving of linked documents.

- The Web is used to provide electronic shopping and gaming.

- The Web is used to listen to radio programs or view television programs.

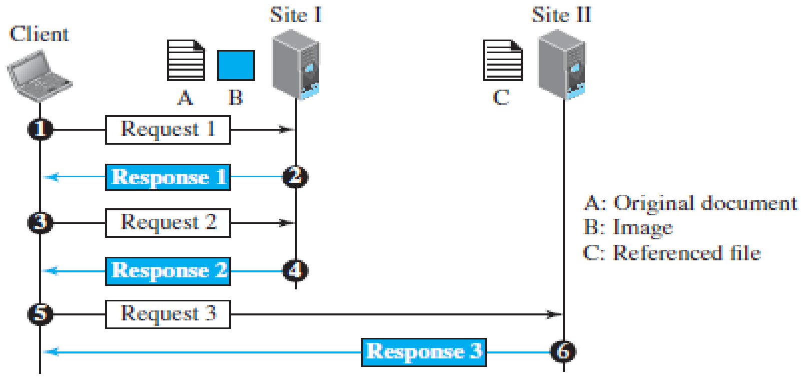
ARCHITECTURE

- 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 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.
- Each web page is a file with a name and address.

Example

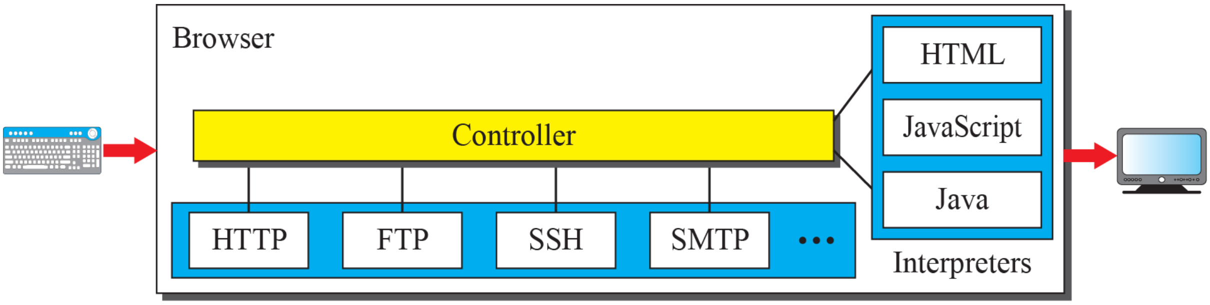
- Assume we need to retrieve a scientific document that contains one reference to another text file and one reference to a large image. Figure shows the situation.
- The main document and the image are stored in two separate files (file A and file B) in the same site; the referenced text file (file C) is stored in another site. Since we are dealing with three different files, we need three transactions if we want to see the whole document.

Figure 26.1 *Example 26.1*



1. 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:
- **1. Controller 2. Client Protocols 3. Interpreters**



Controller

- 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.

Client Protocols

- The client protocol can be one of the protocols, such as HTTP or FTP.

Interpreters

- The interpreter can be HTML, Java, or JavaScript, depending on the type of document.
- Some commercial browsers include Internet Explorer, Netscape Navigator, and Firefox.

2. 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.

3. Uniform Resource Locator (URL)

- A web page, as a file, needs to have a unique identifier to distinguish it from other web pages.
- To define a web page, we need **Four identifiers:**
- 1.Protocol 2.Host 3. Port 4.Path
- The first is used to fetch the web page; the last three that defines the destination object (web page).

Protocol

The first identifier is used to access the web page. Although most of the time the protocol is HTTP (HyperText Transfer Protocol), is used, 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. IP addresses can be defined in dotted decimal notation, (such as 64.23.56.17); the name is normally the domain name that uniquely defines the host.

Port

The port, a 16-bit integer, is normally predefined for the client-server application. For example, if the HTTP protocol is used for accessing the web page, the well-known port number is 80. However, if a different port is used, the number can be explicitly given.

Path

The path identifies the location and the name of the file in the underlying operating system. The format of this identifier normally depends on the operating system. In UNIX, a path is a set of directory names followed by the file name, all separated by a slash.

Example: ***/top/next/last/myfile***

4. Web Documents

The documents in the WWW can be grouped into three broad categories:

1. Static 2. Dynamic 3. Active

1. 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.
- 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).

2. 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.
- Because a fresh document is created for each request, the contents of a dynamic document may vary from one request to another.
- A very simple example of a dynamic document is the retrieval of the time and date from a server.

Example Languages: Common Gateway Interface (CGI), Java Server Pages (JSP), Active Server Pages (ASP) and Structured Query Language (SQL) database in the HTML document.

3. Active Documents

- For many applications, we need a program or a script to be run at the client site. These are called ***active documents*.**
- For example, suppose we want to run a program that creates animated graphics on the screen or a program that interacts with the user.
- When a browser requests an active document, the server sends a copy of the document or a script.
- The document is then run at the client (browser) site.

Languages: 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 ***JavaScript's*** but download and run the script at the client site.

Reference Video

https://www.youtube.com/watch?v=_mNOXDbXr9c

2. 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, 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.
- After the transaction, the connection should be terminated.

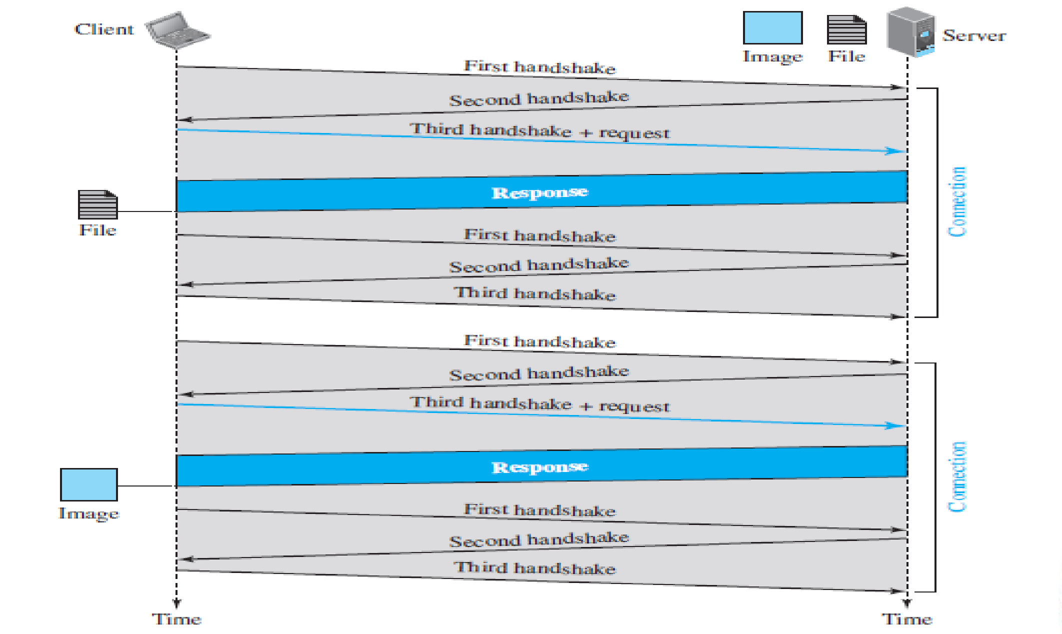
Nonpersistent Connections

- HTTP, prior to version 1.1, specified 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, the connection must be opened and closed N + 1 times.
- The nonpersistent strategy imposes high overhead on the server because the server needs N + 1 different buffers each time a connection is opened.

Figure 26.3 *Example 26.3*

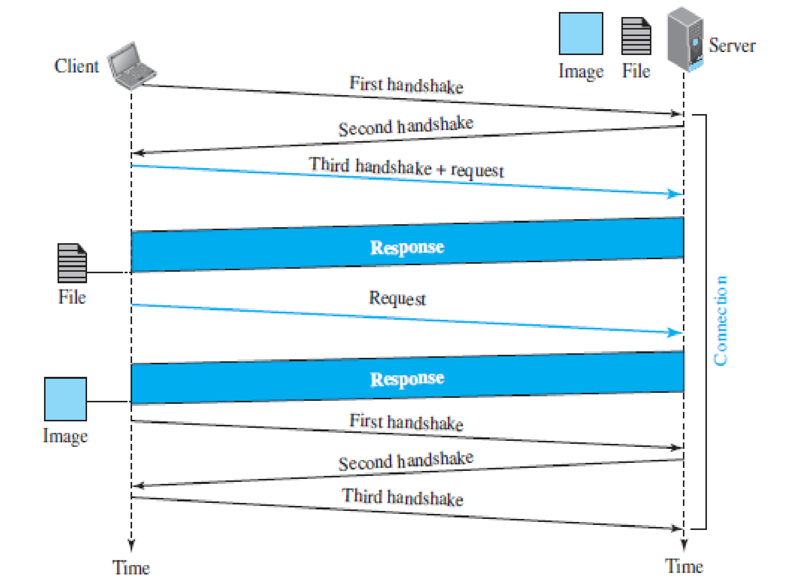


- For each connection, TCP requires at least three handshake messages to establish the connection, but the request can be sent with the third one.
- After the connection is established, the object can be transferred.
- After receiving an object, another three handshake messages are needed to terminate the connection.
- This means that the client and server are involved in two connection establishments and two connection terminations.
- For each connection the client and server need to allocate extra resources such as buffers and variables. This is another burden on both sites, but especially on the server site.

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.

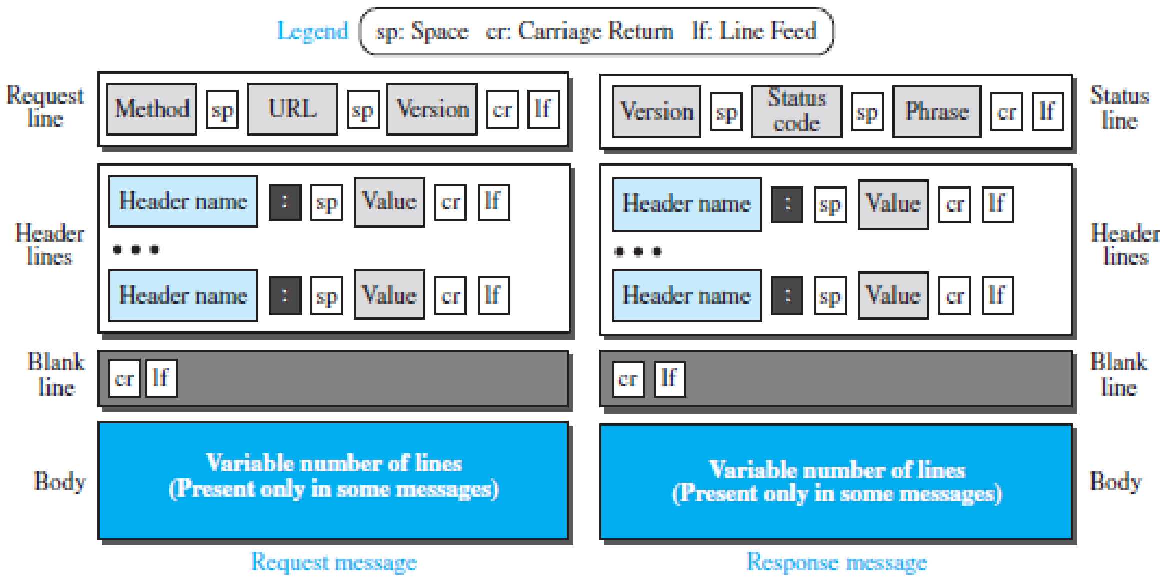
Figure 26.4 *Example 26.4*



Message Formats

- The HTTP protocol defines the format of the request and response messages, as shown in Figure.

Figure 26.5 *Formats of the request and response messages*



- 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.
- However, the similarities between these sections are only in the names; they may have different contents.

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) as shown in above Figure.
- 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.

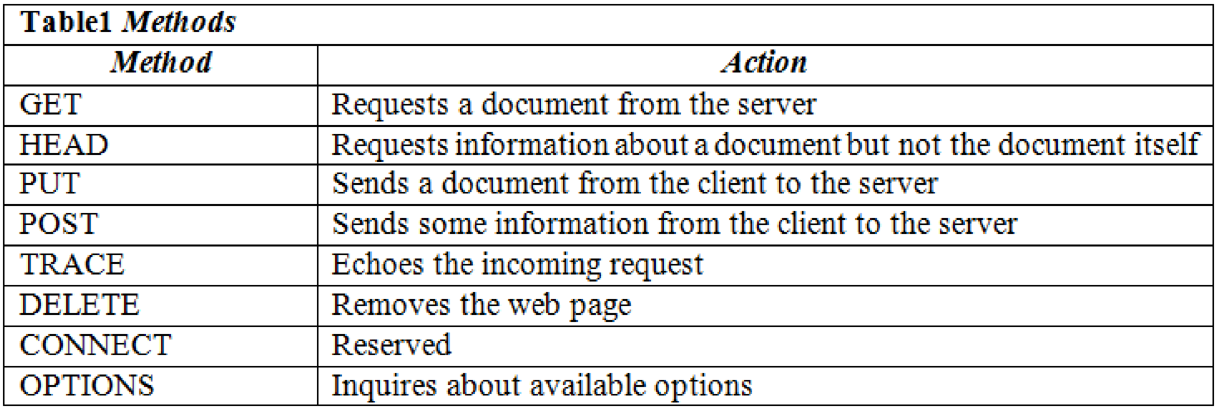


Table1 ***Methods***

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

- The second field, URL, 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.
- 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.
- Each header line has a header name, a colon, a space, and a header value.
- Table shows some header names commonly used in a request.

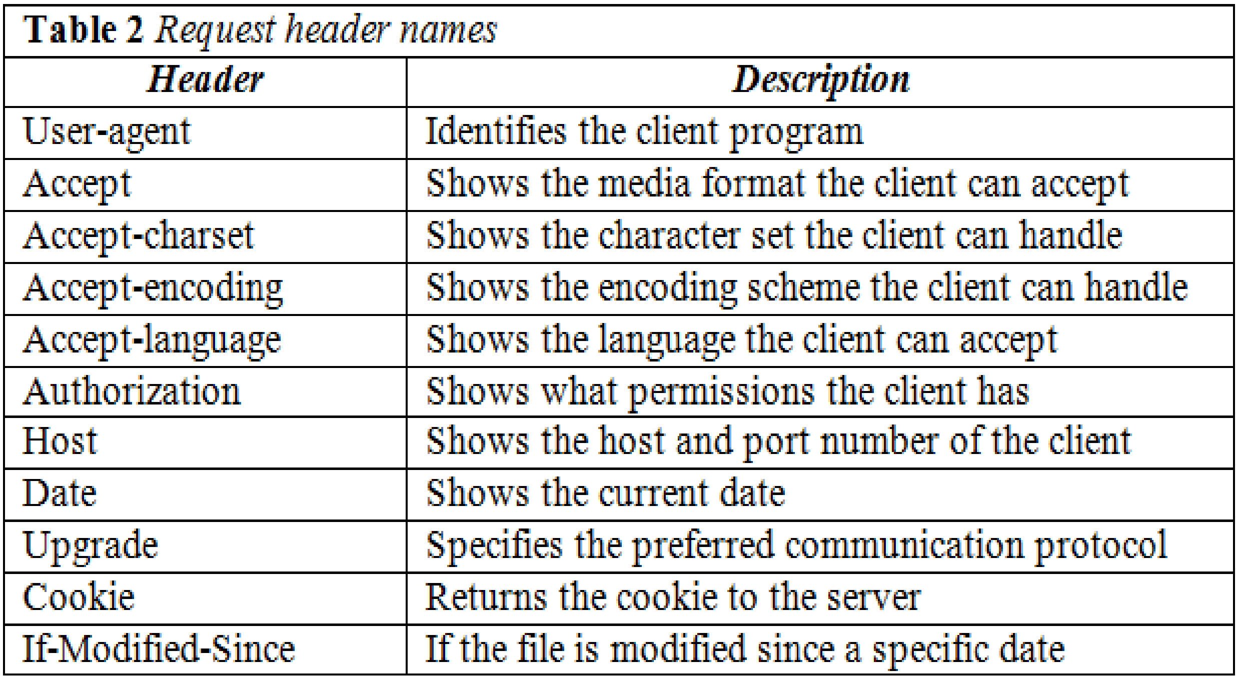


Table 2 *Request header names*

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
If-Modified-Since	If the file is modified since a specific date

Response Message

- The format of the response message is also shown in above Figure.
- 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, and the codes in the 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 phrase explains the status code in text form.
- 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.
- For example, the sender can send extra information about the document.
- Each header line has a header name, a colon, a space, and a header value.
- Table shows some header names commonly used in a response message.

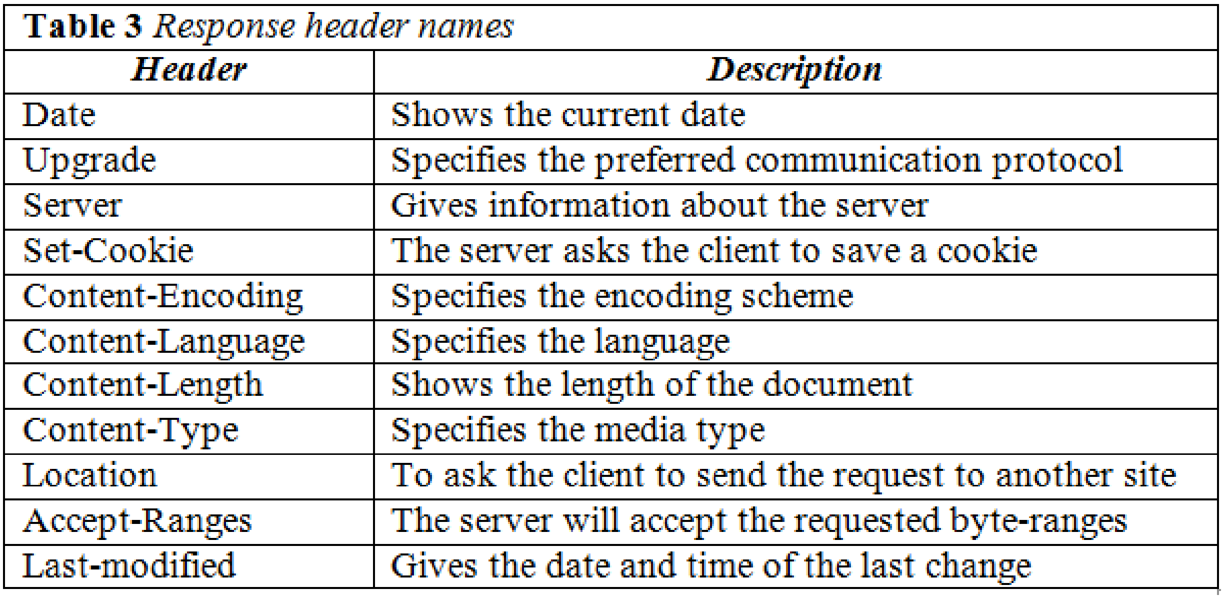


Table 3 *Response header names*

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

Caching

- Caching has many benefits. From the client's perspective, a page that can be retrieved from a nearby cache can be displayed much more quickly than if it has to be fetched from across the world.
- From the server's perspective, having a cache intercept and satisfy a request reduces the load on the server. Caching can be implemented in many different places.
- For example, a user's browser can cache recently accessed pages, and simply display the cached copy if the user visits the same page again.
- There is a set of ―cache directives‖ that must be obeyed by all caching mechanisms along the request/response chain.
- These directives specify whether or not a document can be cached, how long it can be cached, how fresh a document must be, and so on.

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.
- However, to use the proxy server, the client must be configured to access the proxy instead of the target server.

Reference Video

https://www.youtube.com/watch?v=QghbZkks3Dw

5.3 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. Although we can transfer files using HTTP, FTP is a better choice to transfer large files or to transfer files using different formats.

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.
- 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.

Two Connections

- The two connections in FTP have different lifetimes.
- When a user starts an FTP session, the control connection opens.
- The data connection is opened and then closed for each file transfer activity.
- While the control connection is open, the data connection can be opened and closed multiple times if several files are transferred.
- 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.
- 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. 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.
- Some of the most common commands are shown in Table.

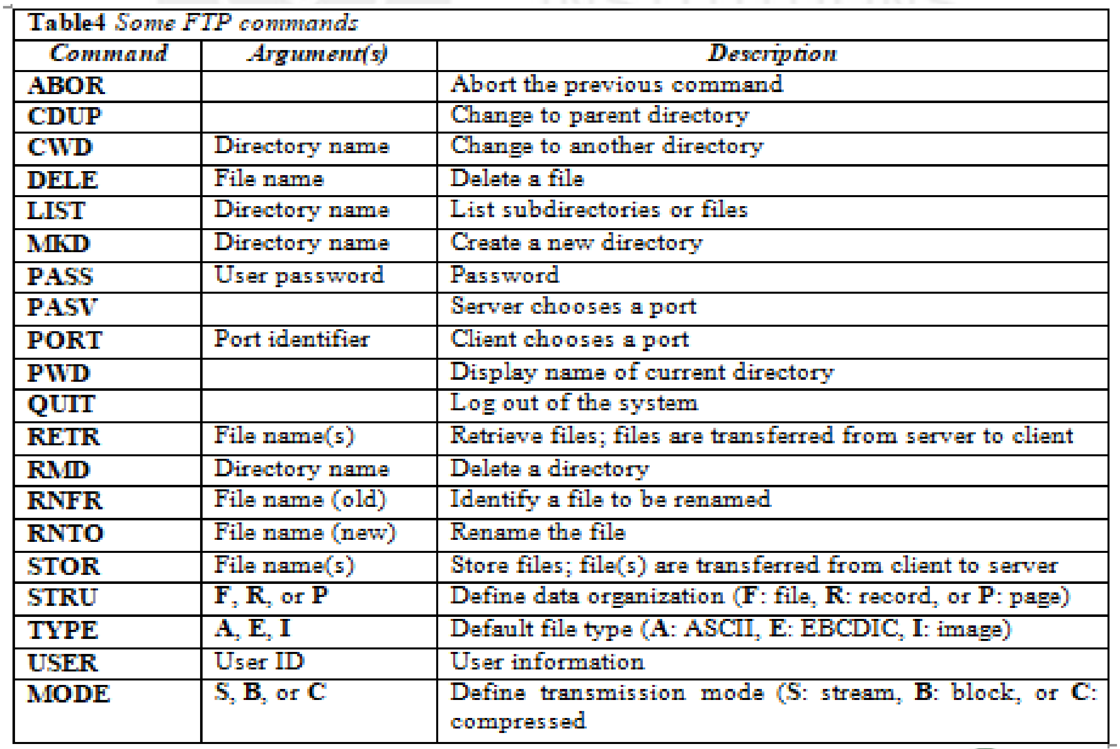


Table4 *Some FTP commands*

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

Response:

- Every FTP command generates at least one response.
- A response has two parts: a three-digit number followed by text.
- The numeric part defines the code; the text part defines needed parameters or further explanations.
  - ❖ The first digit defines the status of the command.
  - ❖ The second digit defines the area in which the status applies.
  - ❖ The third digit provides additional information.

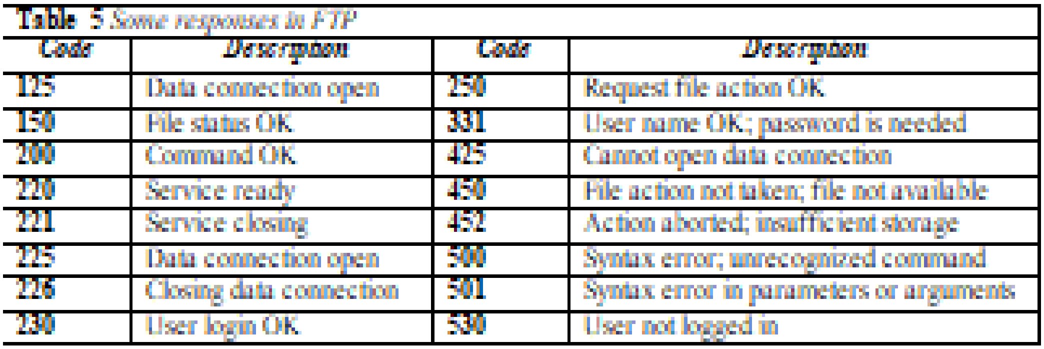


Table 5 *Some responses in FTP*

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 creation of a data connection is different from the control connection. The following shows the steps:

1. The client issues a passive open using well known port. 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.

Communication over Data Connection

- The client must define the type of file to be transferred, the structure of the data, and the transmission mode.
- Before sending the file through the data connection, we prepare for transmission through the control connection.
- The heterogeneity problem is resolved by defining three attributes of communication:
- 1. File type 2. Data structure 3.Transmission mode.

1. File Type

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

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).

2. Data Structure

- FTP can transfer a file across the data connection using one of the following interpretations of the structure of the data:
- 1. File structure 2. Record structure 3. Page structure.
- The **file structure** format (used by default) has no structure. It is a continuous stream of bytes.
- In the **record structure**, the file is divided into records. This can be used only with text files.
- In the **page structure** the file is divided into pages, with each page having a page number and a page header. The pages can be stored and accessed randomly or sequentially.

3. Transmission Mode

- FTP can transfer a file across the data connection using one of the following three transmission modes:
- 1. Stream mode 2.Block mode or compressed mode.
- The **stream mode** is the default mode; data are delivered from FTP to TCP as a continuous stream of bytes.
- In the **block mode** data can be delivered from FTP to TCP in blocks. In this case, each block is preceded by a 3-byte header.
- The first byte is called the block descriptor; the next two bytes define the size of the block in bytes.

Reference Video

https://www.youtube.com/watch?v=7v3GDgvdWO4

4. TELNET

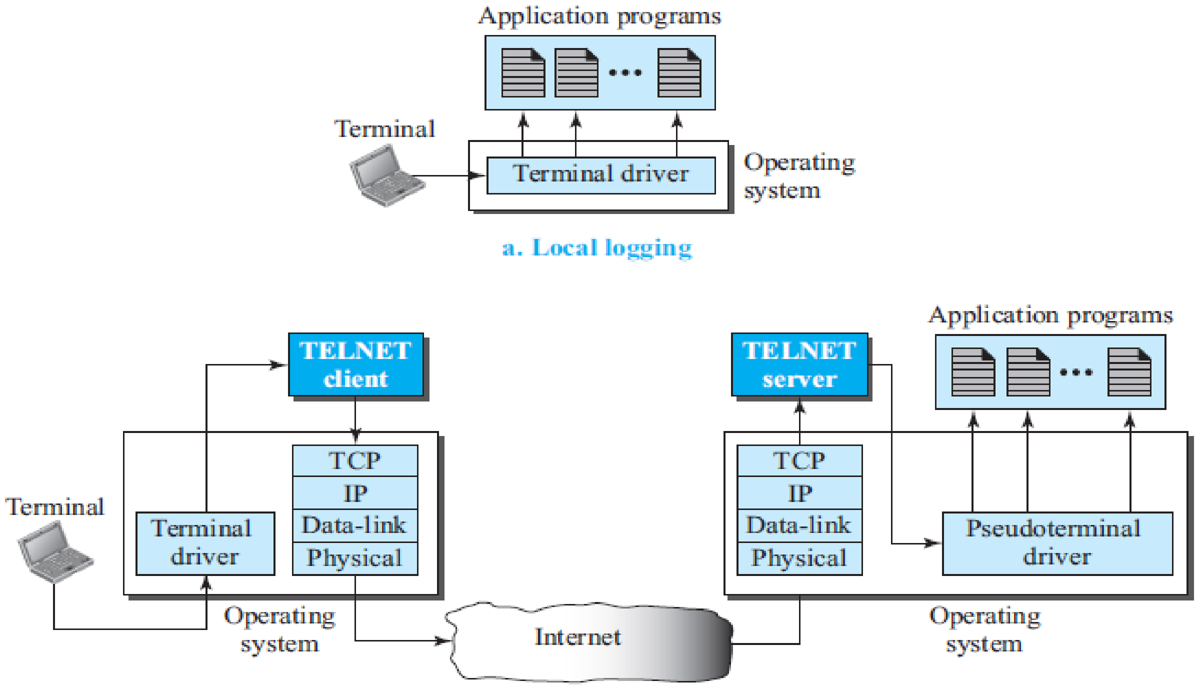
- One of the original remote logging protocols is **TELNET,** which is an abbreviation for TErminaL NETwork.
- Although TELNET requires a logging name and password, it is vulnerable to hacking because it sends all data including the password in plaintext (not encrypted).
- A hacker can eavesdrop and 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).

TELNET here for two reasons:

- 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 Logging

- 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.



a. Local logging

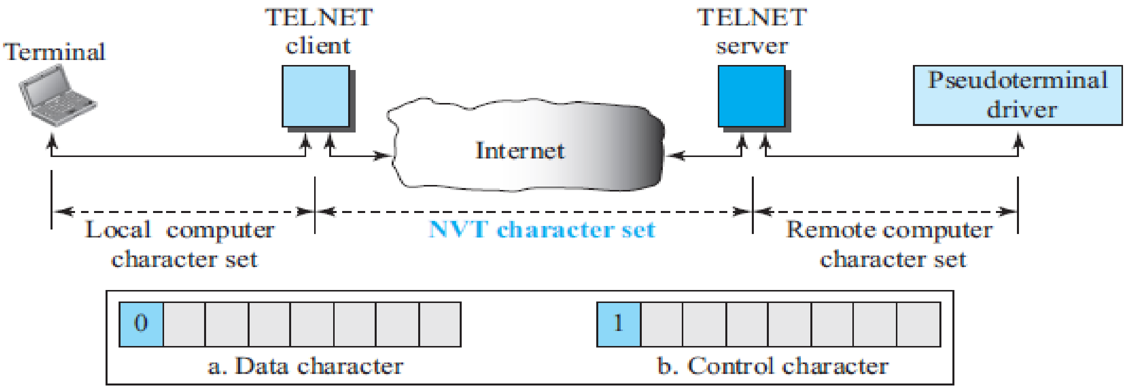
b. Remote logging

Remote Logging

- However, when a user wants to access an application program or utility located on a remote machine called 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.
- 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 ***pseudo terminal driver***, which pretends that the characters are coming from a terminal.
- The operating system then passes the characters to the appropriate application program.

Network Virtual Terminal (NVT)

- The mechanism to access a remote computer is complex. This is because every computer
  and its operating system accept 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.
- 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 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.
- NVT uses two sets of characters, one for data and one for control. Both are 8-bit bytes as shown in Figure.



NVT character format

- 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.

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.

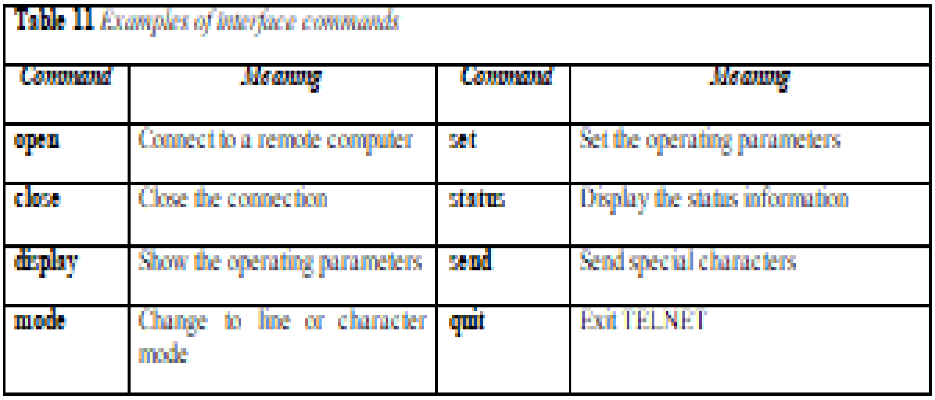


Table 11 *Examples of interface commands*

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

Reference Video

https://www.youtube.com/watch?v=tZop-zjYkrU

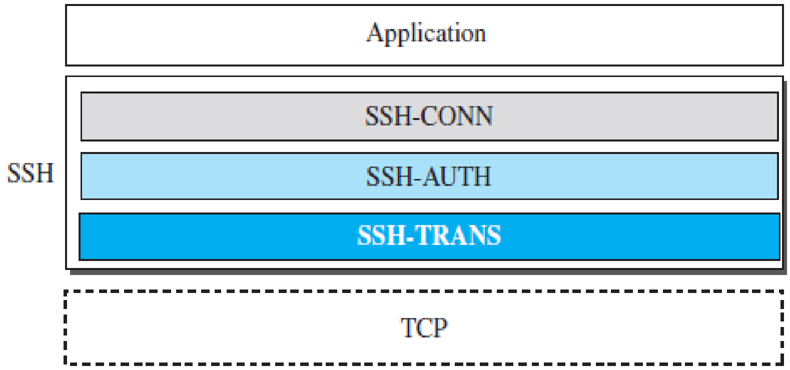
5.5 SECURE SHELL (SSH)

- Although 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. In this section, we discuss only SSH-2.

Components

- SSH is an application-layer protocol with three components, as shown in Figure.

Components of SSH



1. 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.
- When the procedure implementing this protocol is called, the client and server first use the TCP protocol to establish an insecure connection.
- Then they exchange several security parameters to establish a secure channel on top of the TCP.

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

2. 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 similar to the ones used in SSL. 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.

3. 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 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.
- Among them, we can mention PuTTy, by Simon Tatham, which is a client SSH program that can be used for remote logging.
- Another application program is Tectia, which can be used on several platforms.

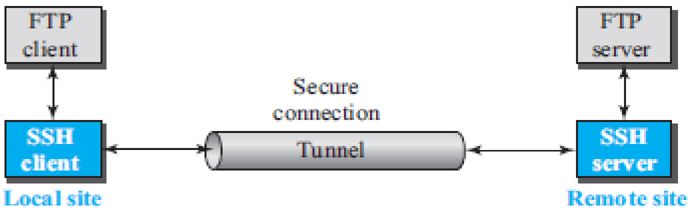
SSH for File Transfer

- 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, cp, to copy files.

Port Forwarding

- One of the interesting services provided by the SSH protocol is port forwarding.
- 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.
- Figure shows the Concept of port forwarding for securing the FTP application.

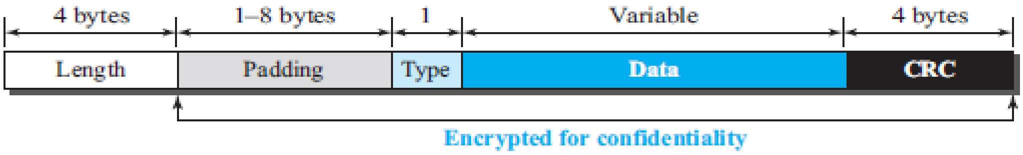
Figure 26.26 *Port forwarding*



- 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

Figure 26.27 *SSH packet format*



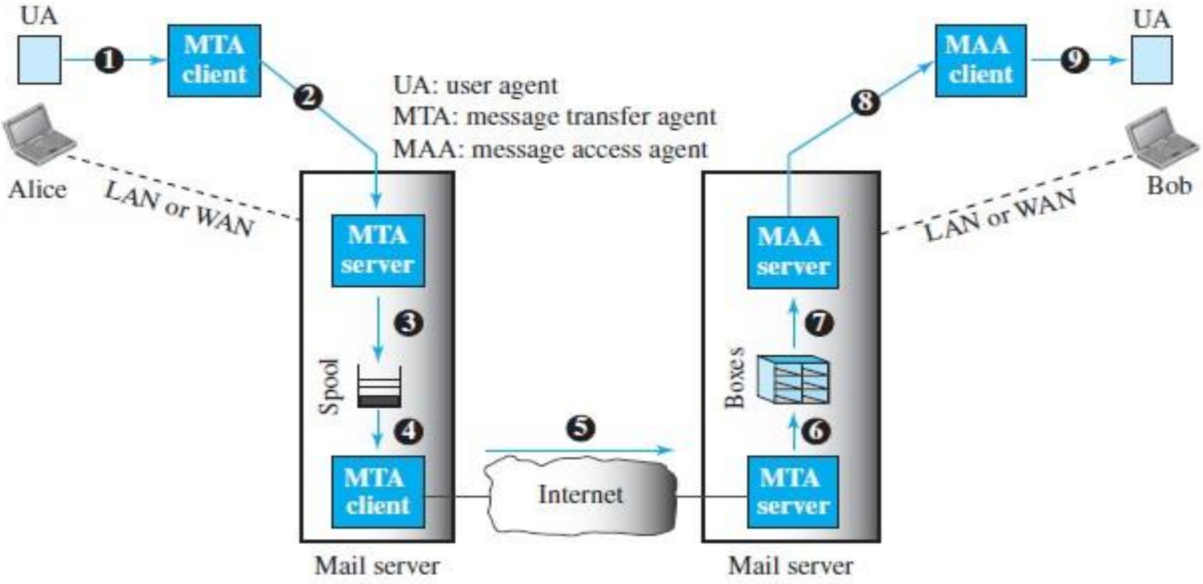
- The length field defines the length of the packet but does not include the padding.
- One to eight bytes of padding is added to the packet to make the attack on the security provision more difficult.
- The cyclic redundancy check (CRC) field is used for error detection.
- The type field designates the type of the packet used in different SSH protocols.
- The data field is the data transferred by the packet in different protocols.

6. ELECTRONIC MAIL

- Electronic mail (or E-mail) allows users to exchange messages.
- First, E-mail is considered a one-way transaction.
- The idea of client/server programming should be implemented in another way: using some intermediate computers (servers).
- The users run only client programs when they want and the intermediate servers apply the client/server paradigm.

1. Architecture

- To explain the architecture of e-mail, we give a common scenario, as shown in Figure.
- In the common scenario, 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.
- 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 User-1 to User-2 takes nine different steps, as shown in the figure.
- User-1[Alice] and User-2[Bob] use three different agents: **a user agent (UA), a message transfer agent (MTA), and a message access agent (MAA).**

1. When Alice needs to send a message to Bob, she runs a UA program to prepare the message and send it to her MTA Client.
2. The message is transferred from MTA client to her mail server.
3. The mail server at her site uses a queue (spool) to store messages waiting to be sent.
4. Later from spool the message is transferred to MTA client.
5. Here two message transfer agents are needed: one client and one server.
6. 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 can be triggered by the system when there is a message in the queue to be sent.
7. The user agent at the Bob site allows Bob to read the received message.
8. Bob later uses an MAA client to retrieve the message from an MAA server running on the second server.
9. This is because an MTA client-server program is a push program: the client pushes the message to the 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 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 belong to the early days of electronic mail.
  - A command-driven user agent normally accepts a one character command from the keyboard to perform its task.
  - Some examples of command driven user agents are mail, pine, and elm.
- Modern user agents are GUI-based.
  - They contain graphical user interface (GUI) components that allow the user to interact with the software by using both the keyboard and the mouse.
  - They have graphical components such as icons, menu bars, and windows that make the services easy to access.
  - Some examples of GUI-based user agents are Eudora and Outlook**.**

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.
- The header of the message defines the sender, the receiver, the subject of the message, and some other information.
- The body of the message contains the actual information to be read by the recipient.

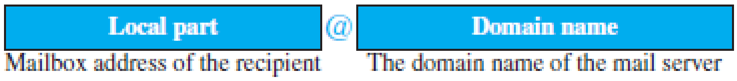
Receiving Mail

- 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.
- The summary usually includes the sender mail address, the subject, and the time the mail was sent or received. The user can select any of the messages and display its contents on the screen.

Addresses

- To deliver mail, a mail handling system must use an addressing system with unique addresses.
- In the Internet, the address consists of two parts: a local part and a domain name, separated by an @ sign.

Figure 26.14 *E-mail address*


Local part
@
Domain name
Mailbox address of the recipient
The domain name of the mail server

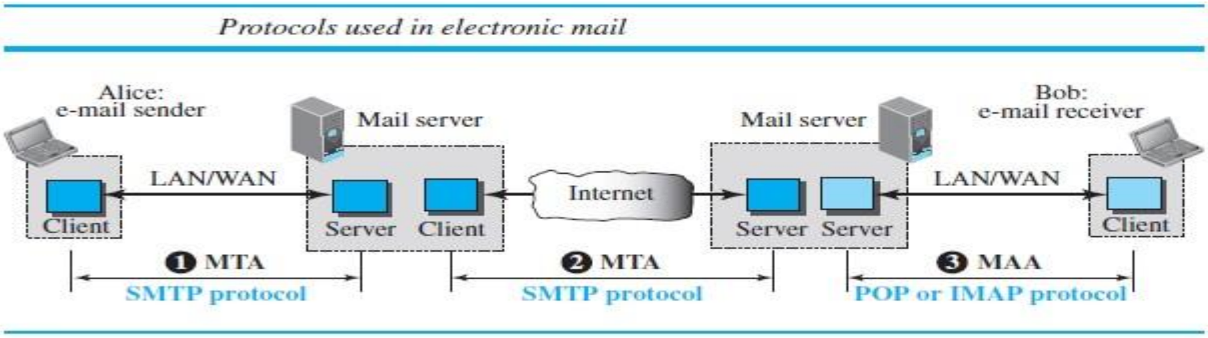
- The local part defines the name of a special file, called the user mailbox, where all the mail received for a user is stored for retrieval by the message access agent.
- The second part of the address is the domain name.
- An organization usually selects one or more hosts to receive and send e-mail; they are sometimes called mail servers or exchangers.

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.
- Every time a message is to be sent, the system checks the recipient's name against the alias database; if there is a mailing list for the defined alias, separate messages, one for each entry in the list, must be prepared and handed to the MTA.

5.6.2 SMTP

- We refer to the first and the second as Message Transfer Agents (MTAs), the third as Message Access Agent (MAA).
- 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***
- 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.
- It consists of a keyword followed by zero or more arguments. SMTP defines 14 commands, listed in Table.

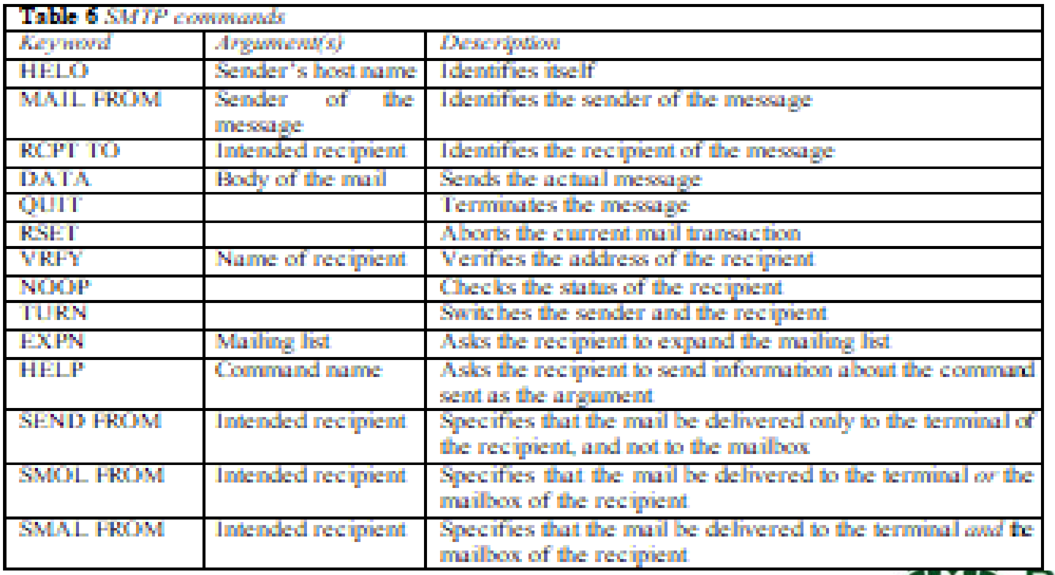


Table 6 *SMTP commands*

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

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 shows the most common response types.

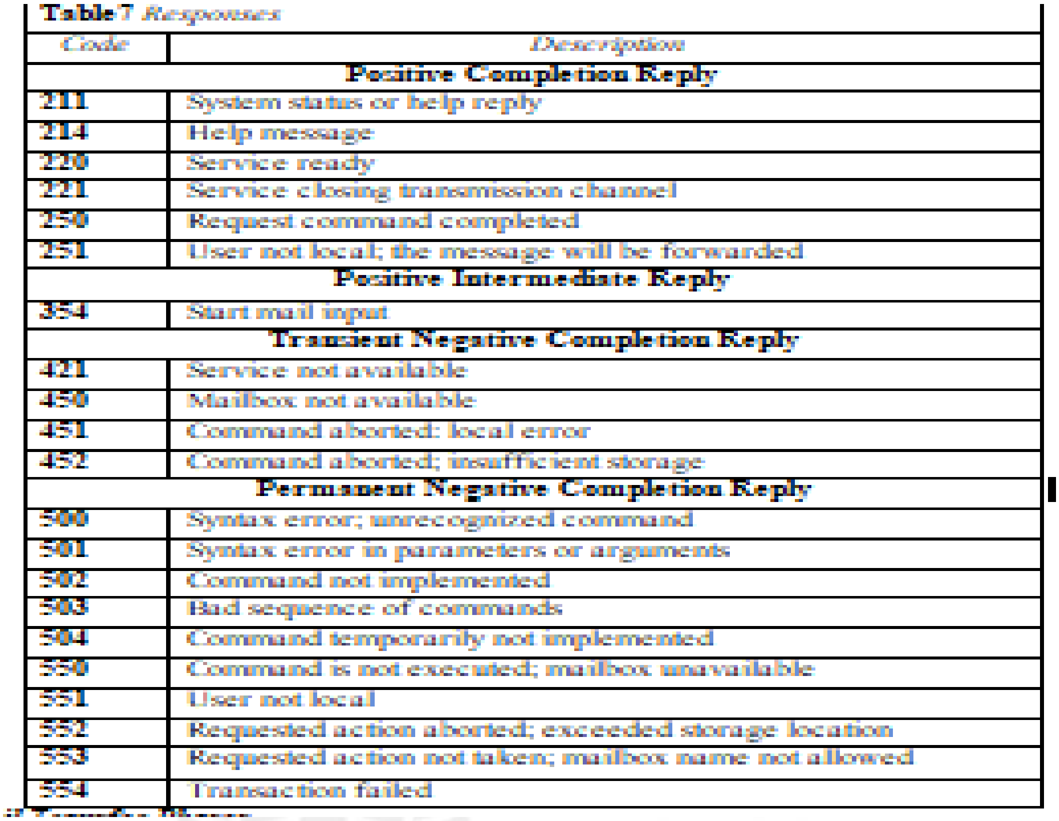


Table 7 *Responses*

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
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:
- 1. Connection establishment 2.Mail transfer 3.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. 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

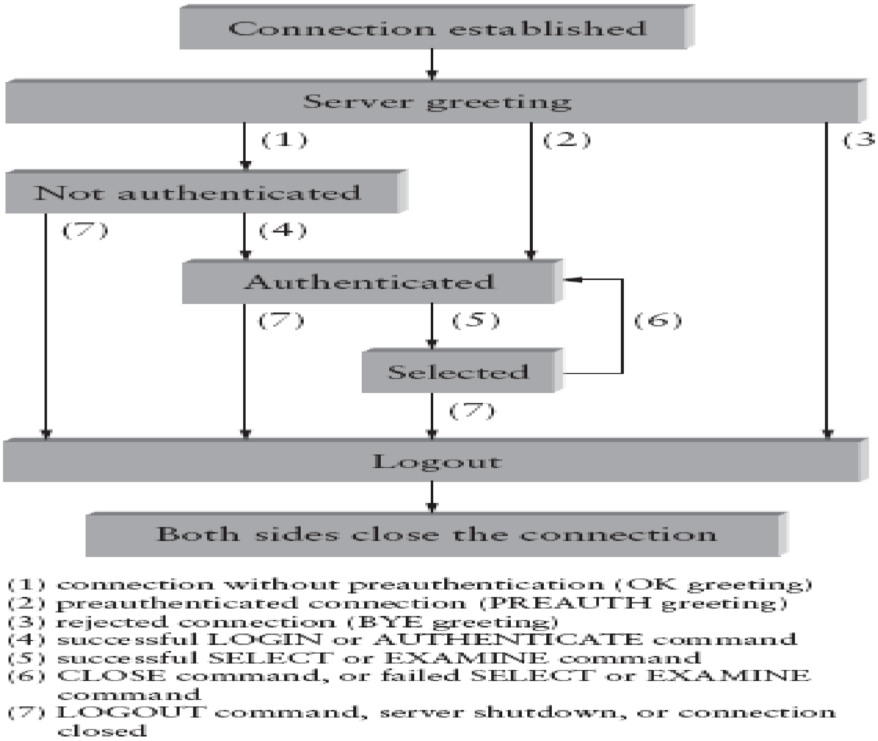
After the message is transferred successfully, the client terminates the connection. This phase involves two steps.

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

5.6.3 IMAP

- It is a client/server protocol running over TCP, where the client (running on the user's desktop machine) issues commands in the form of <CRLF>-terminated ASCII text lines and the mail server (running on the machine that maintains the user's mailbox) responds in kind.
- The exchange begins with the client authenticating him- or her, and identifying the mailbox he or she wants to access.

This can be represented by the simple state transition diagram shown in figure.


Connection established
Server greeting
(1)
(2)
(3)
Not authenticated
(7)
(4)
Authenticated
(7)
(5)
(6)
Selected
(7)
Logout
Both sides close the connection
(1) connection without preauthentication (OK greeting)
(2) preauthenticated connection (PREAUTH greeting)
(3) rejected connection (BYE greeting)
(4) successful LOGIN or AUTHENTICATE command
(5) successful SELECT or EXAMINE command
(6) CLOSE command, or failed SELECT or EXAMINE command
(7) LOGOUT command, server shutdown, or connection closed

IMAP state transition diagram.

5.6.4 POP3 (Post Office Protocol)

- POP is simple and limited in functionality. Current version is POP3. The earlier version (POP2) required SMTP to send a message. POP3 can be used with or without SMTP.
- POP client is installed on the recipient computer and POP server on the mail server.
- Client opens a connection to the server using TCP on port 110.
- Client sends username and password to access mailbox and to retrieve messages. The user can retrieve the mail message one by one.
- It is intended to permit a workstation to dynamically access a mail drop to a server.

Mail Delivery

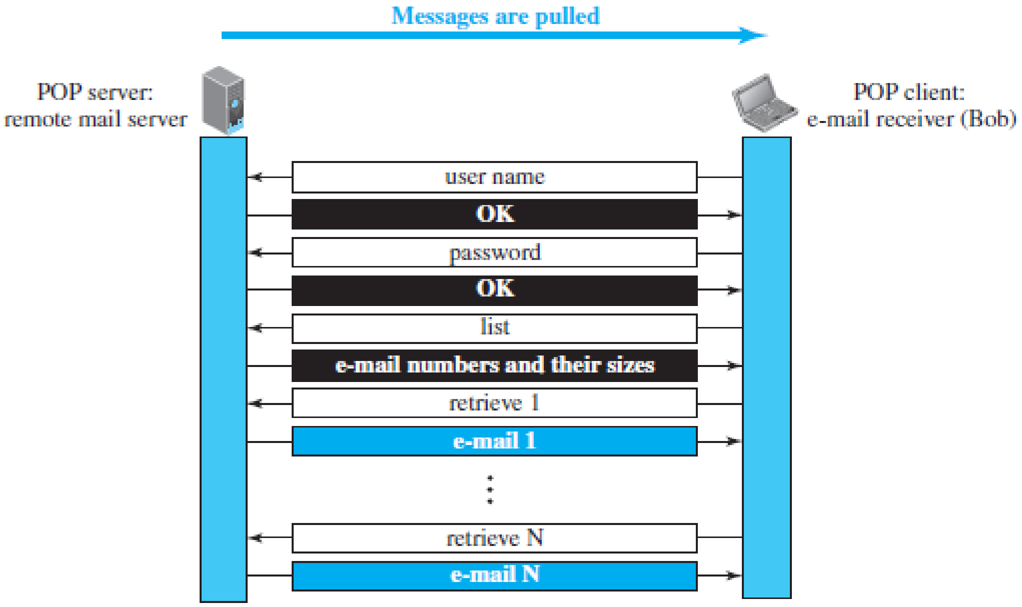
- There are 3 stages for a mail delivery between the sending user agent and receiving user agent.
- Mail is sent from the user agent to the local server.
- The mail goes from local server to remote server(SMTP server)
- User agent gets the mail from the remote server using protocols like POP3.

POP3 Modes

- Delete mode: In delete mode, mail is *deleted* from the mailbox after retrieval.
- Keep mode: In keep mode, mail after reading is *kept* in mailbox for later retrieval.

POP3 Commands

- USER <username> - <username> is your mailbox's username
- PASS <password> - <password> is your mailbox's password
- QUIT - to log off POP3 converstion
- STAT - displays the number of messages currently in the mailbox and the size in bytes
- LIST [msg] -To get a summary of messages where each message number is shown with the size in bytes next to it
- RETR msg - To retrieve a particular message
- DELE msg - To delete a messageNOOP – No operationRSET - reset the session to its initial state using the RSET command



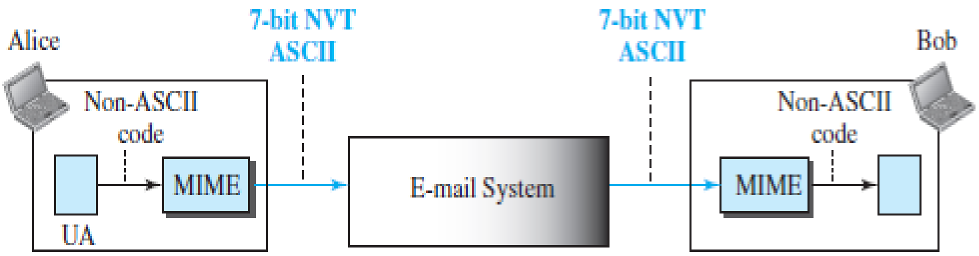
List the advantages of IMAP over POP.

- IMAP is more powerful and more complex than POP.
- User can *check* the e-mail header prior to downloading.
- User can *search* e-mail for a specific string of characters prior to downloading.
- User can download *partially,* very useful in case of limited bandwidth.
- User can create, delete, or rename *mailboxes* on the mail server.

5.6.5 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.
- MIME transforms non-ASCII data at the sender site to NVT ASCII data and delivers it to the client MTA to be sent through the Internet.
- The message at the receiving site is transformed back to the original data.
- We can think of MIME as a set of software functions that transforms non-ASCII data to ASCII data and vice versa, as shown in Figure.
- It can send messages only in NVT 7-bit ASCII format. Also, it cannot be used to send binary files or video or audio data.

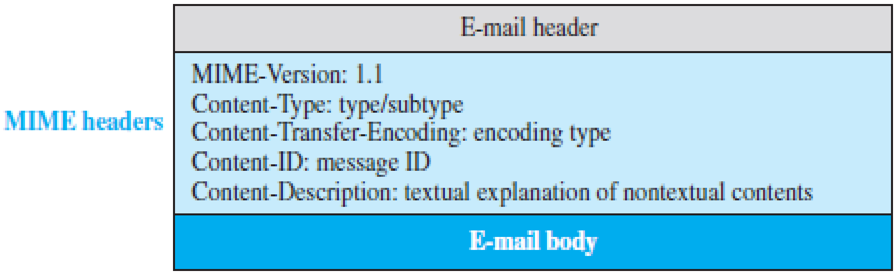
MIME



MIME Headers

- MIME defines five headers, as shown in figure, which can be added to the original e-mail header section to define the transformation parameters:

MIME header



E-mail header

MIME-Version: 1.1
Content-Type: type/subtype
Content-Transfer-Encoding: encoding type
Content-ID: message ID
Content-Description: textual explanation of nontextual contents

E-mail body

MIME-Version

- This header defines the version of MIME used. The current version is 1.1.

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.
- Depending on the subtype, the header may contain other parameters.
- MIME allows seven different types of data, listed in Table.

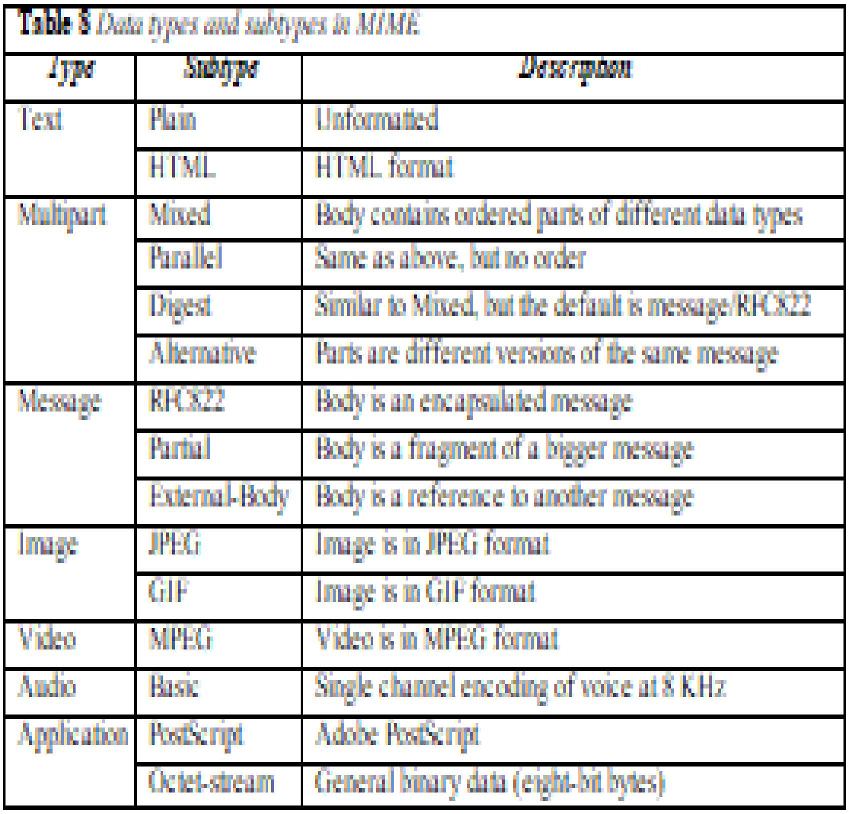


Table 8 *Data types and subtypes in MIME*

Type	*Subtype*	*Description*
Text	Plain	Unformatted
	HTML	HTML format
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
Video	MPEG	Video is in MPEG format
Audio	Basic	Single channel encoding of voice at 8 KHz
Application	PostScript	Adobe PostScript
	Octet-stream	General binary data (eight-bit bytes)

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.

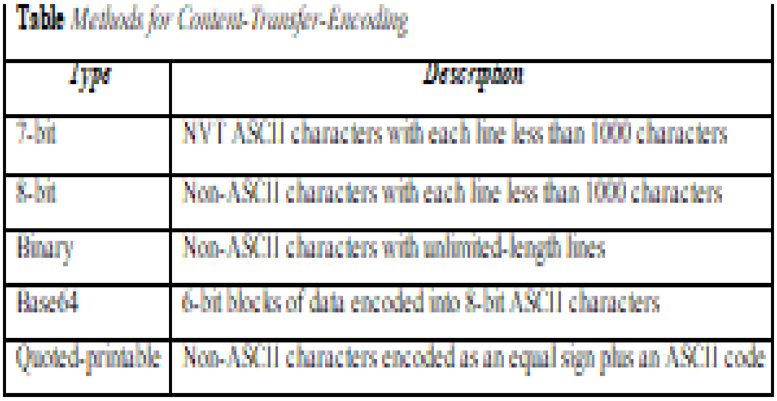


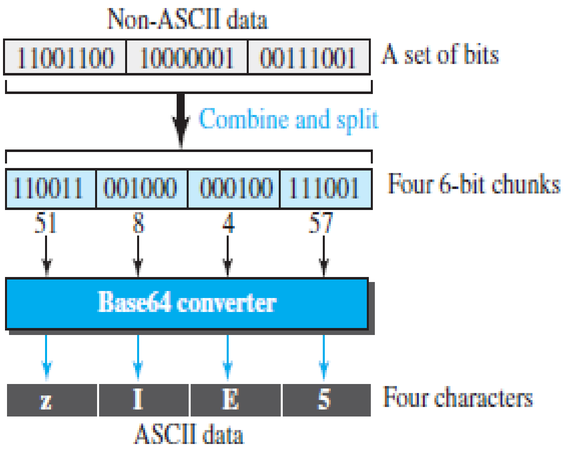
Table *Methods for Content-Transfer-Encoding*

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

5.6.6 BASE 64 ENCODING SCHEME

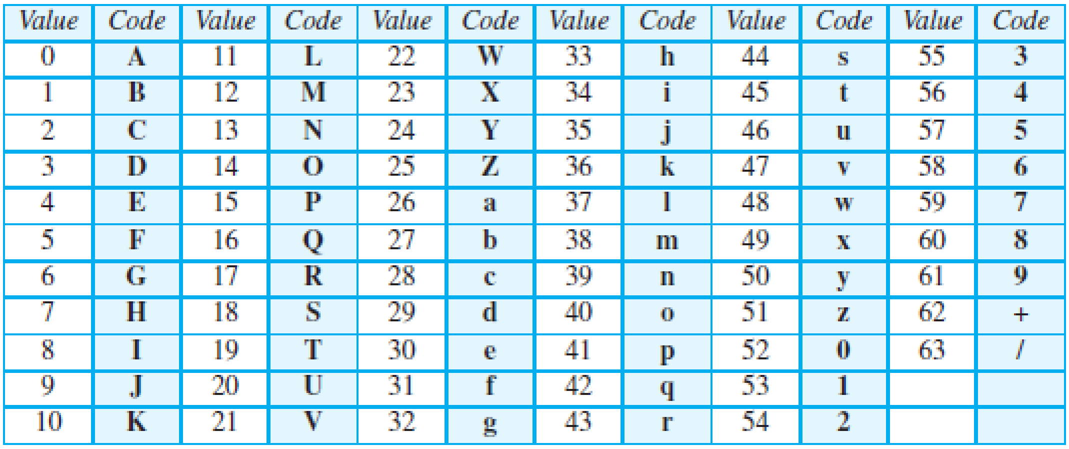
- The idea is to map every three bytes of the original binary data into four ASCII characters.
- This is done by grouping the binary data into 24-bit units.
- Breaking each such unit into four 6-bit pieces.
- Each 6-bit piece maps onto one of 64 valid ASCII characters.
- The first 64 valid ASCII characters are the 52 upper- and lowercase letters, the 10 digits 0 through 9, and the special characters + and /.

Figure 26.20 *Base64 conversion*



- Each 6-bit section is then converted into an ASCII character according to Table.

Table 26.10 *Base64 converting table*



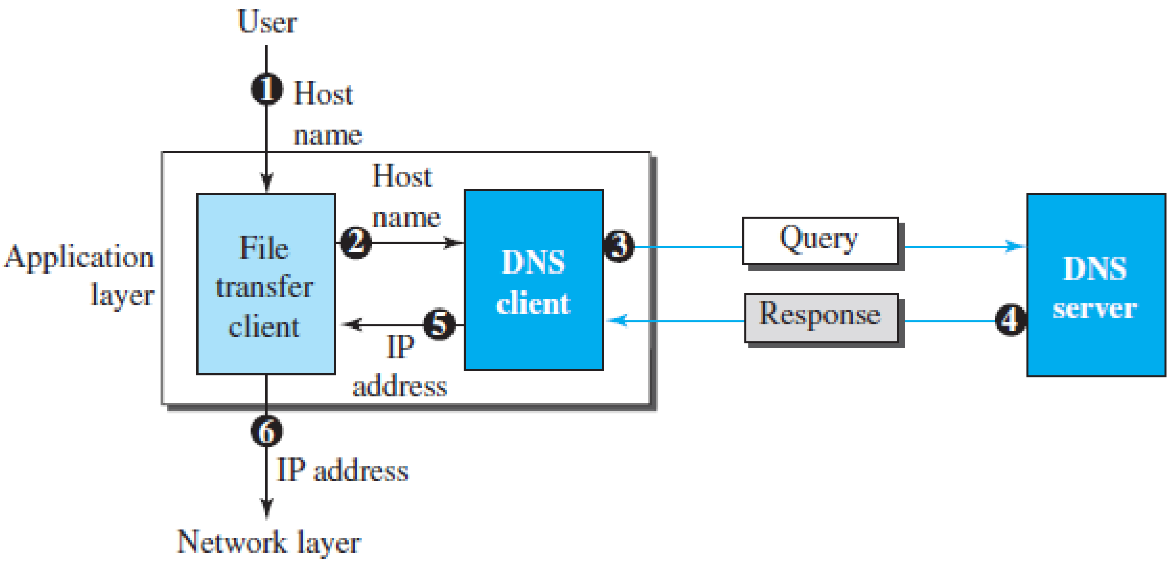
Value	Code	Value	Code	Value	Code	Value	Code	Value	Code	Value	Code
0	A	11	L	22	W	33	h	44	s	55	3
1	B	12	M	23	X	34	i	45	t	56	4
2	C	13	N	24	Y	35	j	46	u	57	5
3	D	14	O	25	Z	36	k	47	v	58	6
4	E	15	P	26	a	37	l	48	w	59	7
5	F	16	Q	27	b	38	m	49	x	60	8
6	G	17	R	28	c	39	n	50	y	61	9
7	H	18	S	29	d	40	o	51	z	62	+
8	I	19	T	30	e	41	p	52	0	63	/
9	J	20	U	31	f	42	q	53	1		
10	K	21	V	32	g	43	r	54	2		

Reference Video

https://www.youtube.com/watch?v=DvwVSdghxr0

5.7 DOMAIN NAME SYSTEM (DNS)

- Domain Name System Protocol is used to query name servers and send the responses.
- The Internet needs to have a directory system that can map a name to an address.
- The first is for mapping the name to an IP address; the second is for transferring files.



The following six steps map the host name to an IP address:

- 1. The user passes the host name to the file transfer client.
- 2. The file transfer client passes the host name to the DNS client.
- 3.Each computer, after being booted, knows the address of one DNS server. The DNS client sends a message to a DNS server with a query that gives the file transfer server name using the known IP address of the DNS server.
- 4. The DNS server responds with the IP address of the desired file transfer server.
- 5. The DNS server passes the IP address to the file transfer client.
- 6.The file transfer client now uses the received IP address to access the file transfer server.

Name Space

- The names must be unique because the addresses are unique.
- A name space that maps each address to a unique name can be organized in two ways:

  1. Flat 2. Hierarchical.

Flat name space

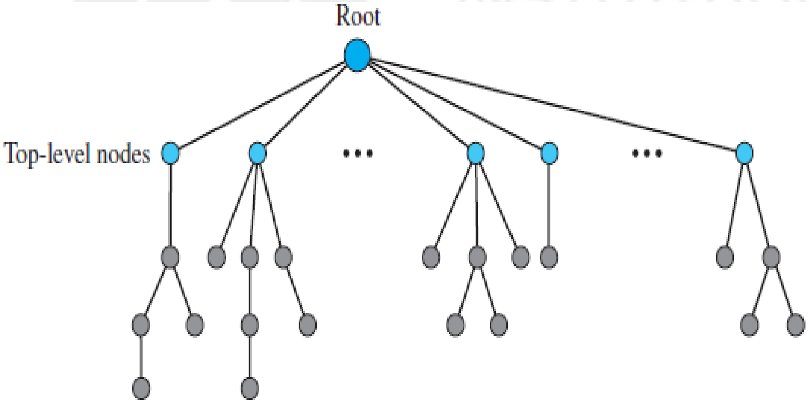
- A flat name is assigned to an address.
- A name in this space is a sequence of characters without structure.
- The names may or may not have a common section; if they do, it has no meaning.
- The main disadvantage of a flat name space is that it cannot be used in a large system such as the Internet because it must be centrally controlled to avoid ambiguity and duplication.

Hierarchical name space

- Each name is made of several parts.
- The first part can define the nature of the organization, the second part can define the name of an organization and the third part can define departments in the organization, and so on.
- A central authority can assign the part of the name that defines the nature of the organization and the name of the organization.

Domain Name Space

- To have a hierarchical name space, a domain name space was designed.
- In this design the names are defined in an inverted-tree structure with the root at the top.
- The tree can have only 128 levels: level 0 (root) to level 127

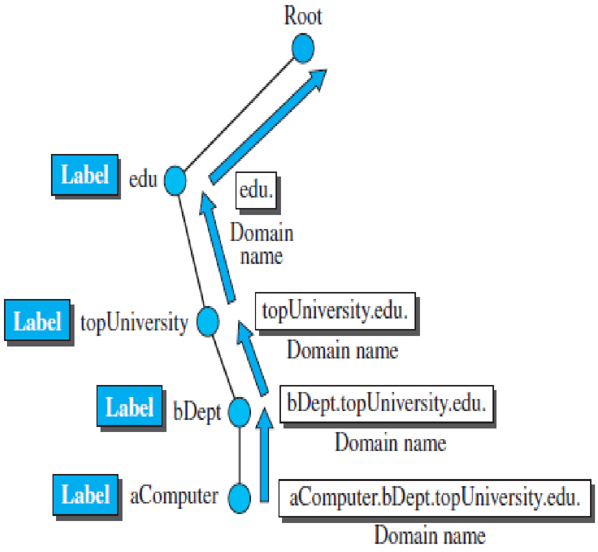


Label

- Each node in the tree has a label, which is a string with a maximum of 63 characters.
- The root label is a null string (empty string). DNS requires that children of a node (nodes that branch from the same node) have different labels, which guarantees the uniqueness of the domain names.

Domain Name

- Each node in the tree has a domain name. A full domain name is a sequence of labels separated by dots (.)
- The domain names are always read from the node up to the root. The last label is the label of the root (null).
- This means that a full domain name always ends in a null label, which means the last character is a dot because the null string is nothing.

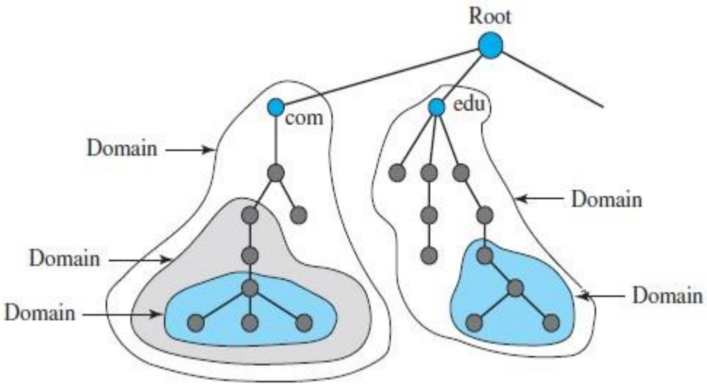


FQDN: If a label is terminated by a null string, it is called a **fully qualified domain name (FQDN)**. The name must end with a null label, but because null means nothing, the label ends with a dot.

PQDN: If a label is not terminated by a null string, it is called a **partially qualified domain name (PQDN).** A PQDN starts from a node, but it does not reach the root.

Domain

- A **domain** is a subtree of the domain name space.
- The name of the domain is the name of the node at the top of the subtree.

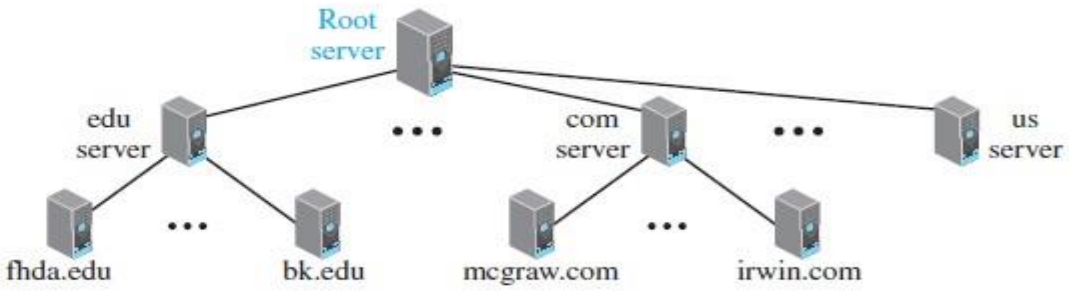


Distribution of Name Space

- The information contained in the domain name space must be stored.
- However, it is very inefficient and also not reliable to have just one computer store such a huge amount of information.
- It is inefficient because responding to requests from all over the world places a heavy load on the system.
- It is not reliable because any failure makes the data inaccessible.

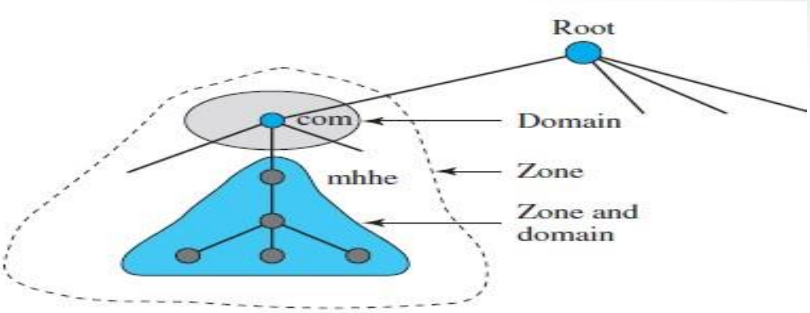
Hierarchy of Name Servers

- The solution to these problems is to distribute the information among many computers called ***DNS servers.***
- One way to do this is to divide the whole space into many domains based on the first level.
- In other words, we let the root stand alone and create as many domains (subtrees) as there are first-level nodes.
- Because a domain created this way could be very large, DNS allows domains to be divided further into smaller domains (subdomains).
- Each server can be responsible (authoritative) for either a large or small domain.
- In other words, we have a hierarchy of servers in the same way that we have a hierarchy of names.



Zone

- The hierarchy is partitioned into subtrees called **zones.** What a server is responsible for or has authority over is called a ***zone***.
- For example, the following figure shows how the hierarchy given in the above figure might be divided into zones.
- Each zone can be thought of as corresponding to some administrative authority that is responsible for that portion of the hierarchy.
- The information contained in each zone is implemented in two or more name servers.
- Each name server, in turn, is a program that can be accessed over the Internet.



Root Server

- A **root server** is a server whose zone consists of the whole tree.
- A root server usually does not store any information about domains but delegates its authority to other servers, keeping references to those servers.
- There are several root servers, each covering the whole domain name space.
- The root servers are distributed all around the world.

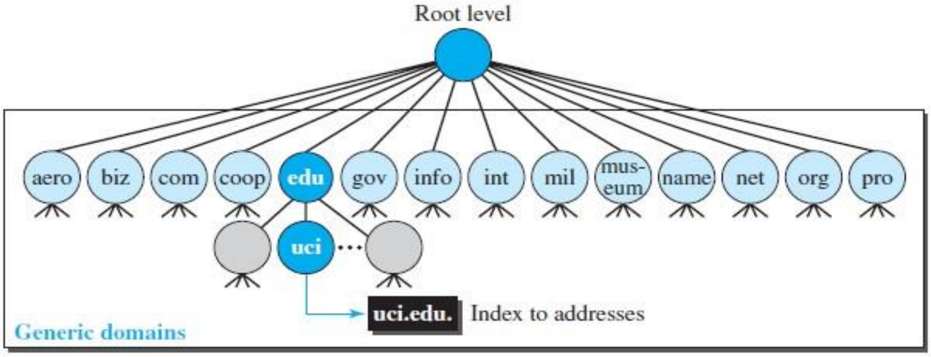
Primary and Secondary Servers

- DNS defines two types of servers: **primary and secondary**.
- A ***primary server*** is a server that stores a file about the zone for which it is an authority.
- It is responsible for creating, maintaining, and updating the zone file. It stores the zone file on a local disk.
- A ***secondary server*** is a server that transfers the complete information about a zone from another server (primary or secondary) and stores the file on its local disk. The secondary server neither creates nor updates the zone files.
- If updating is required, it must be done by the primary server, which sends the updated version to the secondary.

DNS in the Internet

- DNS is a protocol that can be used in different platforms. In the Internet, the domain name space (tree) was originally divided into three different sections: **Generic domains, Country domains, and the Inverse domains.**

1. ***Generic Domains:*** The **generic domains** define registered hosts according to their generic behavior. Each node in the tree defines a domain, which is an index to the domain name space database.

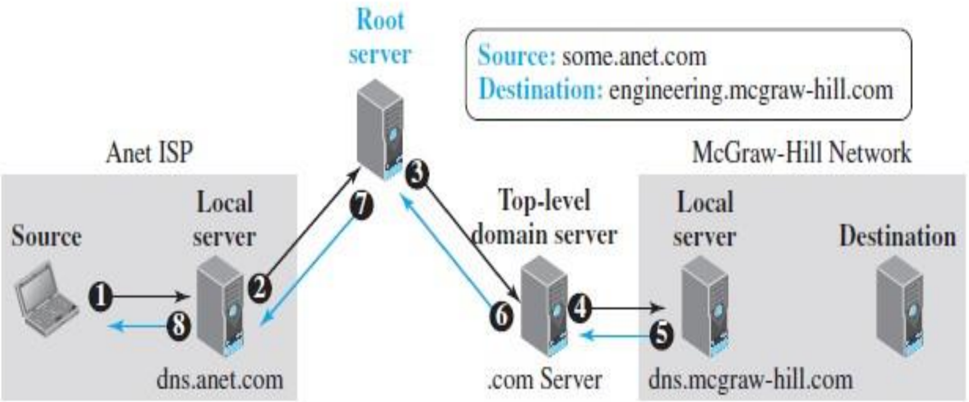


2. ***Country Domains:*** The **country domains** section uses two-character country abbreviations. Second labels can be organizational, or they can be more specific national designations.

3. Inverse Domain: The **inverse domain** is used to map an address to a name.

Name Resolution

- Mapping a name to an address is called name-**address resolution.**
- DNS is designed as a client-server application.
- A host that needs to map an address to a name or a name to an address calls a DNS client called a **resolver.**
- The resolver accesses the closest DNS server with a mapping request.
- If the server has the information, it satisfies the resolver; otherwise, it either refers the resolver to other servers or asks other servers to provide the information.
- After the resolver receives the mapping, it interprets the response to see if it is a real resolution or an error, and finally delivers the result to the process that requested it.
- A resolution can be either recursive or iterative.



Recursive Resolution

- We assume that an application program running on a host named **some.anet.com** needs to find the IP address of another host named **engineering.mcgraw-hill.com** to send a message to.
- The source host is connected to the Anet ISP; the destination host is connected to the McGraw-Hill network.
- The application program on the source host calls the DNS resolver (client) to find the IP address of the destination host.

- The resolver, which does not know this address, sends the query to the local DNS server (for example, dns.anet.com) running at the Anet ISP site **(event 1).**

- We assume that this server does not know the IP address of the destination host either.

- It sends the query to a root DNS server, whose IP address is supposed to be known to this local DNS server **(event 2).**

- Root servers do not normally keep the mapping between names and IP addresses, but a root server should at least know about one server at each top level domain (in this case, a server responsible for com domain).

- The query is sent to this top-level-domain server **(event 3).** We assume that this server does not know the name-address mapping of this specific destination, but it knows the IP address of the local DNS server in the McGraw-Hill company (for example, dns.mcgraw-hill.com). The query is sent to this server **(event 4)** which knows the IP address of the destination host. The IP address is now sent back to the top-level DNS server **(event 5)** then back to the root server **(event 6)** then back to the ISP DNS server, which may cache it for the future queries **(event 7)** and finally back to the source host **(event 8).**

- ***Iterative Resolution***

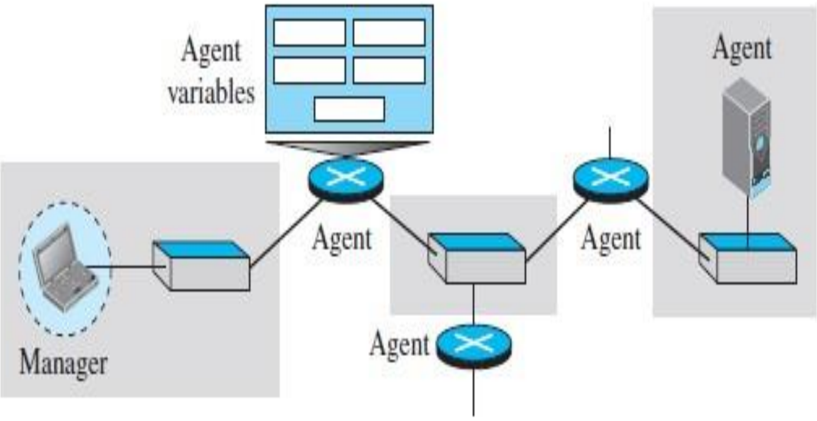
- In **iterative resolution,** each server that does not know the mapping sends the IP address of the next server back to the one that requested it.

Reference Video

https://www.youtube.com/watch?v=JkEYOt08-rU

5.8 SNMP

- Simple Network Management Protocol (SNMP) is an application layer protocol that monitors and manages routers, distributed over a network.



- SNMP is a framework for managing devices in an internet using the TCP/IP protocol suite. It provides a set of fundamental operations for monitoring and maintaining an internet.
- SNMP uses the concept of manager and agent. That is, a manager, usually a host, controls and monitors a set of agents; usually routers or servers.
- SNMP is an application-level protocol in which a few manager stations control a set of agents.

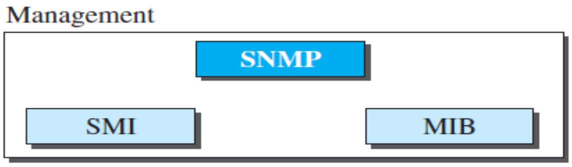
Managers and Agents

- A management station, called a **manager,** is a host that runs the SNMP client program.
- A managed station, called an **agent,** is a router (or a host) that runs the SNMP server program. Management is achieved through simple interaction between a manager and an agent.
- The agent keeps performance information in a database.
- The manager has access to the values in the database.
- SNMP uses services of UDP on two well-known ports: 161 (agent) and 162 (manager).
- **SNMP** is essentially a **specialized request/reply protocol** that supports two kinds of **request messages**: **GET and SET**.
- **GET** is used to **retrieve a piece of state from some node.**
- **SET** is used to **store a new piece of state in some node**.
- SNMP also supports a third operation—**GET-NEXT.**
- SNMP uses the concept of manager and agent.
  - Manager is a host that runs SNMP client program (GUI)
  - Agent is a router that runs SNMP server program.

Management Components

- SNMP is supported by two protocols:
  - Structure of Management Information (SMI)
  - Management Information Base (MIB).

Figure 27.3 *Components of network management on the Internet*



Role of SNMP

- SNMP has some very specific roles in network management.
- It defines the format of the packet to be sent from a manager to an agent and vice versa.
- It also interprets the result and creates statistics (often with the help of other management software).
- The packets exchanged contain the object (variable) names and their status (values).
- SNMP is responsible for reading and changing these values.

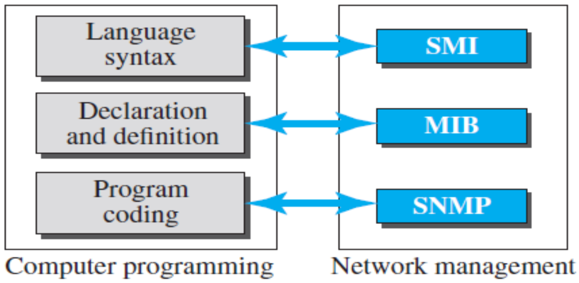
Role of SMI

- To use SNMP, we need rules for naming objects. This is particularly important because the objects in SNMP form a hierarchical structure (an object may have a parent object and some child objects).
- The sender may be a powerful computer in which an integer is stored as 8-byte data; the receiver may be a small computer that stores an integer as 4-byte data.
- SMI is a protocol that defines these rules.
- However, we must understand that SMI only defines the rules; it does not define how many objects are managed in an entity or which object uses which type.
- SMI is a collection of general rules to name objects and to list their types.
- The association of an object with the type is not done by SMI.

Role of MIB

- For each entity to be managed, this protocol must define the number of objects, name them according to the rules defined by SMI, and associate a type to each named object.
- This protocol is MIB.
- MIB creates a set of objects defined for each entity in a manner similar to that of a database (mostly metadata in a database, names and types without values).

Figure 27.4 *Comparing computer programming and network management*



- **SNMP** defines the format of packets exchanged between a manager and an agent. It reads and changes the status of objects (values of variables) in SNMP packets.
- **SMI** defines the general rules for naming objects, defining object types (including range and length), and showing how to encode objects and values.
- **MIB** creates a collection of named objects, their types, and their relationships to each other in an entity to be managed.

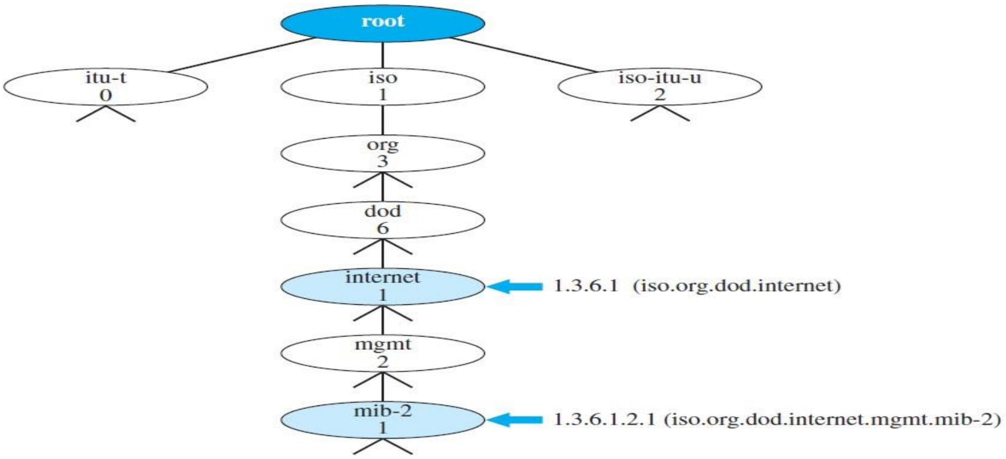
5.8.1 SMI

- The Structure of Management Information, version 2 (SMIv2) is a component for network management. SMI is a guideline for SNMP.
- It emphasizes **three attributes** to handle an object: 1. Name 2. Data type 3.Encoding method. Its functions are:
- To name objects.
- To define the type of data that can be stored in an object.
- To show how to encode data for transmission over the network.

Name

- SMI requires that each managed object (such as a router, a variable in a router, a value, etc.) have a unique name.
- To name objects globally, SMI uses an **object identifier,** which is a hierarchical identifier based on a tree structure.
- The tree structure starts with an unnamed root.
- Each object can be defined using a sequence of integers separated by dots.
- The tree structure can also define an object using a sequence of textual names separated by dots.
- The integer-dot representation is used in SNMP.
- The name-dot notation is used by people.

Figure 27.6 *Object identifier in SMI*


root
itu-t
0
iso
1
iso-itu-u
2
org
3
dod
6
internet
1
1.3.6.1 (iso.org.dod.internet)
mgmt
2
mib-2
1
1.3.6.1.2.1 (iso.org.dod.internet.mgmt.mib-2)

Type

- The second attribute of an object is the type of data stored in it.
- To define the data type, SMI uses **Abstract Syntax Notation One (ASN.1)** definitions and adds some new definitions.
- SMI has two broad categories of data types: simple and structured.
- The **simple data types** are atomic data types.
- By combining simple and structured data types, we can make new **structured data types.**
- SMI defines two **structured data types:** sequence and sequence of.

❑ ***Sequence.*** A sequence data type is a combination of simple data types, not necessarily of the same type. It is analogous to the concept of a struct or a record used in programming languages such as C.

❑ ***Sequence of.*** A sequence of data type is a combination of simple data types all of the same type or a combination of sequence data types all of the same type. It is analogous to the concept of an array used in programming languages such as C.

Encoding Method

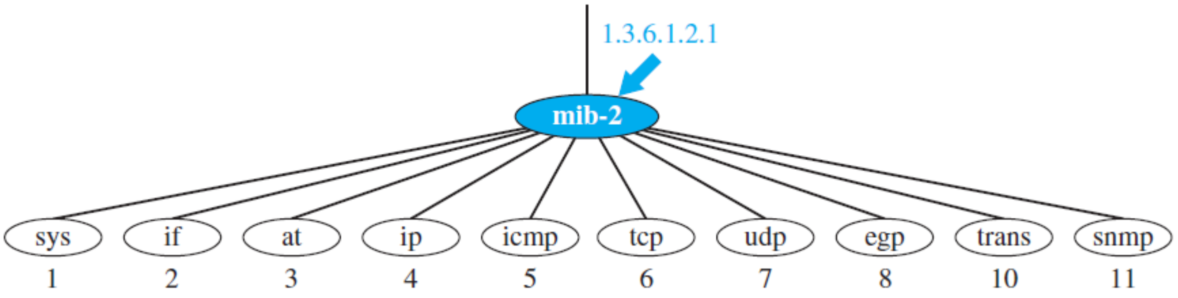
- SMI uses another standard, Basic Encoding Rules (BER), to encode data to be transmitted over the network.
- BER specifies that each piece of data be encoded in triplet format: tag, length, and value (TLV).
- The tag is a 1-byte field that defines the type of data.
- The length field is 1 or more bytes. If it is 1 byte, the most significant bit must be 0.
- The other 7 bits define the length of the data. If it is more than 1 byte, the most significant bit of the first byte must be 1.
- The other 7 bits of the first byte specify the number of bytes needed to define the length.
- The value field codes the value of the data according to the rules defined in BER.

5.8.2 MIB

- Each agent has its own MIB, which is a collection of objects to be managed.
- The current version of MIB, called MIB-II, organizes variables into 10 different groups.

- The following is a brief description of some of the objects:
- **sys** This object (system) defines general information about the node (system), such as the name, location, and lifetime.
- **if** This object (interface) defines information about all of the interfaces of the node including interface number, physical address, and IP address.
- **at** This object (address translation) defines the information about the ARP table.
- **ip** This object defines information related to IP, such as the routing table and the IP address.
- **icmp** This object defines information related to ICMP, such as the number of packets sent and received and total errors created.
- **tcp** This object defines general information related to TCP, such as the connection table, time-out value, number of ports, and number of packets sent and received.
- **udp** This object defines general information related to UDP, such as the number of ports and number of packets sent and received.
- **egp** These objects are related to the operation of EGP.
- **trans** These objects are related to the specific method of transmission (future use).
- **snmp** This object defines general information related to SNMP itself.

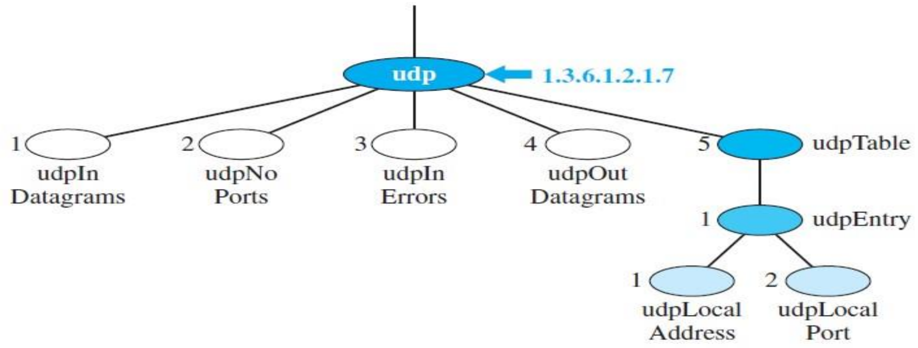
Figure 27.13 *Some mib-2 groups*



Accessing MIB Variables

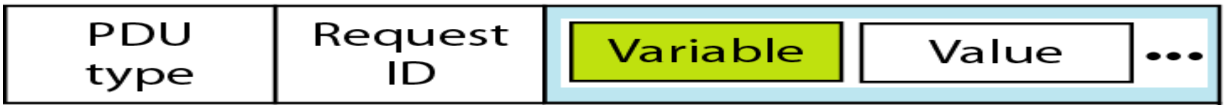
- To show how to access different variables, we use the UDP group as an example.
- There are four simple variables in the UDP group and one sequence of (table of) records.
- Figure shows the variables and the table. We will show how to access each entity

Figure 27.14 *udp group*



Protocol Data Unit (PDU)

- SNMP is request/reply protocol that supports various operations using PDUs:
- GET used by manager to retrieve value of agent variable.
- GET-NEXT used by manager to retrieve next entries in an agent's table.
- SET used by manager to set value of an agent's variable.
- RESPONSE sent from an agent to manager in response to GET/GET-NEXT that contains value of variables.
- TRAP sent from agent to the manager to report an event such as reboot.
- When administrator selects a piece of information, manager puts identifier for the MIB variable and sends request message to the agent.
- Agent maps the identifier, retrieves value of the variable, and sends encoded value back to the manager.



PDU type | Request ID | Variable | Value | •••

Reference Video

https://www.youtube.com/watch?v=fuxb37aiJ4Y

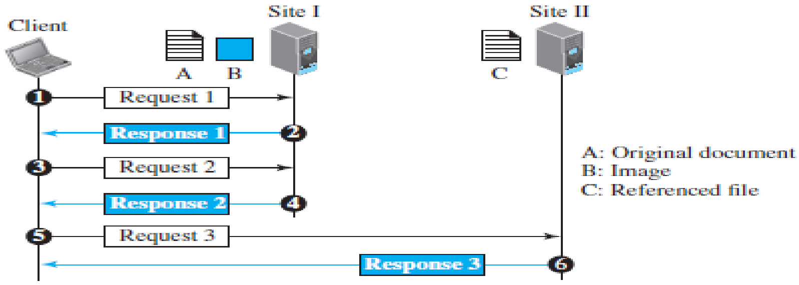
QUIZ TIME

https://quizizz.com/admin/quiz/5f475efbfafecf001bf6722c/application-layer

ASSIGNMENT – I

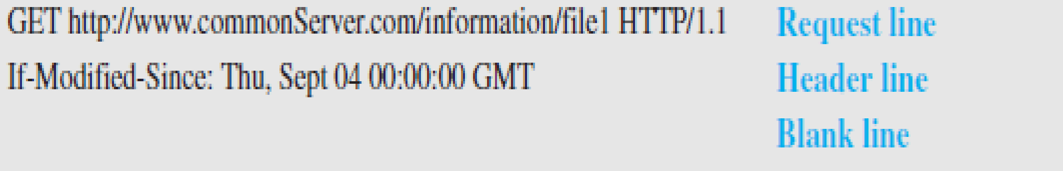
1. Assume we need to retrieve a scientific document that contains one reference to another text file and one reference to a large image. Figure 26.1 shows the situation. The main document and the image are stored in two separate files (file A and file B) in the same site; the referenced text file (file C) is stored in another site. Since we are dealing with three different files, we need three transactions if we want to see the whole document.

Figure 26.1 *Example 26.1*



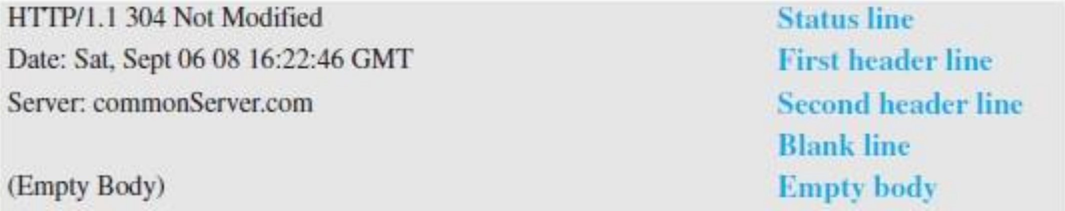
- The first transaction (request/response) retrieves a copy of the main document (file A), which has references (pointers) to the second and third files.
- When a copy of the main document is retrieved and browsed, the user can click on the reference to the image to invoke the second transaction and retrieve a copy of the image (file B).

2. The following shows how a client imposes the modification data and time condition on a request.



GET http://www.commonServer.com/information/file1 HTTP/1.1	Request line
If-Modified-Since: Thu, Sept 04 00:00:00 GMT	Header line
	Blank line

The status line in the response shows the file was not modified after the defined point in time. The body of the response message is also empty.



HTTP/1.1 304 Not Modified	Status line
Date: Sat, Sept 06 08 16:22:46 GMT	First header line
Server: commonServer.com	Second header line
	Blank line
(Empty Body)	Empty body

UNIT V QUESTION BANK

PART –A

1.Differentiate application programs and application protocols. (Nov/Dec 2013)

API describes all the valid messages that one program can accept. It says nothing about the proper ordering of these messages

Protocols sit on top of APIs. A protocol describes the valid sequence of messages that flow between the APIs of multiple parties to accomplish some higher-level task.

2.What is WWW and SMTP? *(Nov 10,15)(May 15)*

World Wide Web is an internet application that allows user to view pages and move from one web page to another. It helps to store and share data across varied distances. The TCP/IP protocol that supports electronic mail on the Internet is called Simple Mail Transfer (SMTP). It is a system for sending messages to other computer users based on e-mail addresses.

3.What is a URL, web browser and rlogin?(*May 2016*)

Uniform Resource Locator is a string identifier that identifies a page on the World Wide Web.

Web browser is a software program that interprets and displays the contents of HTML web pages.

Remote login or rlogin is used to login into remote system and access its contents.

4.What are the four main properties of HTTP?

Global Uniform Resource Identifier.

Request-response exchange.

Statelessness.

Resource metadata.

5.What are the four groups of HTTP Headers? What are the two methods of HTTP?

(May 15) (Nov 15)

The four groups of HTTP headers are

General headers,Entity Headers,Request Headers,Response Headers.

Two methods of HTTP are

GetMethod( ) ,PostMethod( )

6.What are the advantages of allowing persistent TCP connections in HTTP? (May/June 2013)

HTTP requests and responses can be pipelined on a connection. Pipelining allows a client to make multiple requests without waiting for each response, allowing a single TCP connection to be used much more efficiently, with much lower elapsed time.

Network congestion is reduced by reducing the number of packets caused by TCP opens, and by allowing TCP sufficient time to determine the congestion state of the network.

7. What are the uses of HTTP? (May 2018)

The Hypertext Transfer Protocol (HTTP) is an application protocol for distributed, collaborative, hypermedia information systems. HTTP is the foundation of data communication for the World Wide Web. Hypertext is structured text that uses logical links (hyperlinks) between nodes containing text.

8. State the usage of conditional get in HTTP. (May 2017)

The conditional GET method is intended to reduce unnecessary network usage by allowing cached entities to be refreshed without requiring multiple requests or transferring data already held by the client. The semantics of the GET method change to a "partial GET" if the request message includes a Range header field.

9. Describe why HTTP is defined as a stateless protocol.

Maintaining state across request – Response connections significantly increases the initial interactions in a connections since the identity of each party needs to be established and any saved state much be retrieved. HTTP is therefore stateless to ensure that internet is scalable since state is not contained in a HTTP request / response pairs by default.

10. What are the transmission modes of FTP?

- Stream mode: Default mode and data is delivered from FTP to TCP as a continuous stream of data.
- Block mode: Data is delivered from FTP to TCP in terms of blocks. Each data block follows the three byte header.
- Compressed mode: File is compressed before transmitting if size is big. Run length encoding method is used for compression.

11. What are the TCP connections needed in FTP?

- FTP establishes two connections between the hosts. One connection is used for data transfer, the other for control information.
- The control connection uses very simple rules of communication. The data connection needs more complex rules due to the variety of data types transferred.

12. What if TFTP?

- Trivial file transfer protocol is designed for transferring bootstrap and configuration files. It is so simple and can fit into ROM of a disc less memory. TFTP does reading and writing of files.
- Reading means copying files from server site to client site and writing in FTP means copying a file from client site to server site.

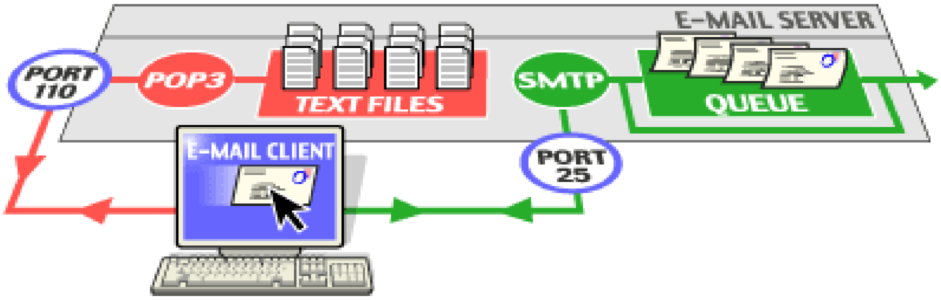
13. Define SMTP. (May/June 2015, Nov 2012, May/June 2012, Nov 2010)

- The TCP/IP protocol supports electronic mail on the Internet is called Simple Mail Transfer (SMTP). It is a system for sending messages to other computer users based on e-mail addresses. SMTP provides mail exchange between users on the same or different computers.

14. Write the services of UA.

- Composing messages
- Reading messages
- Replying to messages
- Forwarding message
- Handling mailboxes

15. Draw the scenario of E-Mail.



16. State the difference between SMTP and MIME. (Nov/Dec 2014)

SMTP	MIME
Simple Mail Transfer (SMTP) is a system for sending messages (electronic mail) to other computer users through Internet based on e-mail addresses. SMTP provides mail exchange between users on the same or different computers.	Multipurpose Internet Mail Extensions (MIME) is a supplementary protocol that allows non-ASCII data to be sent through e-mail. MIME transforms non-ASCII data at the sender site to NVT ASCII data and delivers them to the client to be sent through the Internet. The message at the receiving side is transformed back to the original data.

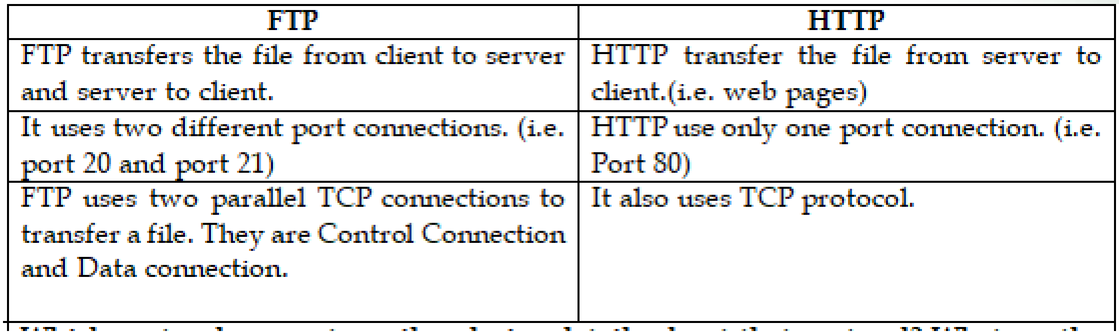
17. Why is an application such as POP needed for electronic messaging? *(May 12)*

- Workstations interact with the SMTP host, which receives the mail on behalf of every host in the organization, to retrieve messages by using a client-server protocol such as Post Office Protocol.
- Although POP3 is used to download messages from the server, the SMTP client still needed on the desktop to forward messages from the workstation user to its SMTP mail server.

18. What is the use of MIME Extension?

- **Multipurpose Internet Mail Extensions (**MIME) is a supplementary protocol that allows non-ASCII data to be sent through SMTP. MIME transforms non-ASCII data at the sender site to NVT ASCII data and deliverers it to the client SMTP to be sent through the Internet. MIME converts binary files, executed files into text files. Then only it can be transmitted using SMTP

19. Compare the HTTP and FTP.



FTP	HTTP
FTP transfers the file from client to server and server to client.	HTTP transfer the file from server to client.(i.e. web pages)
It uses two different port connections. (i.e. port 20 and port 21)	HTTP use only one port connection. (i.e. Port 80)
FTP uses two parallel TCP connections to transfer a file. They are Control Connection and Data connection.	It also uses TCP protocol.

20.Which protocol support email and give details about that protocol? What are the basic functions of e-mail?

- **SMTP** is a standard protocol for transferring mails using TCP/IP
- SMTP standardization for message character is 7 bit ASCII
- SMTP adds log info to the start (i.e.) path of the message.
- Basic functions of e-mail: composition, Transfer, Reporting, Displaying, and Disposition.

21. What is POP3?

- POP3 (Post Office Protocol 3) is the most recent version of a standard protocol for receiving e-mail. POP3 is a client/server protocol in which e-mail is received and held for you by your Internet server. POP and IMAP deal with the receiving of e-mail and are not to be confused with the Simple Mail Transfer Protocol (SMTP).

22. List the function of POP3. (May/June 2011)

- POP3 (Post Office Protocol 3) is the most recent version of a standard protocol for receiving e-mail.
- POP3 is a client/server protocol in which e-mail is received and held for you by your Internet server.
- POP and IMAP deal with the receiving of e-mail and are not to be confused with the Simple Mail Transfer Protocol (SMTP),a protocol for transferring e-mail across the Internet.

23. What is IMAP?

- Internet Message Access Protocol (IMAP) is a standard protocol for accessing e-mail from your local server. IMAP is a client/server protocol in which e-mail is received and held for you by your Internet server. MAP can be thought of as a remote file server. POP3 can be thought of as a "store-and-forward" service.

24. What are the applications of Telnet? (May/June 2011)

- Configuring network devices
- Participating in online communities
- Remote Login

25. What is Telnet? (May/June 2014, Nov 2011)

- A Telnet is a Transmission Control Protocol (TCP). Connection used to transmit data with interspersed Telnet Control Information. The Telnet Protocol is built upon three main ideas:
- The concept of a network virtual terminal
- The principle of negotiated options
- A symmetric view of terminals and processes.
- Telnet is the standard TCP/IP protocol for virtual terminal service.

26. What is DNS. (May 2018)

The Domain Name System (DNS) is the phonebook of the Internet. Humans access information online through domain names, like nytimes.com or espn.com. Web browsers interact through Internet Protocol (IP) addresses. DNS translates domain names to IP addresses so browsers can load Internet resources.

27. Present the information contained in DNS resource record. (May 2017)

A resource record, commonly referred to as an RR, is the unit of information entry in DNS zone files; RRs are the basic building blocks of host-name and IP information and are used to resolve all DNS queries. Resource records come in a fairly wide variety of types in order to provide extended name-resolution services.

28.Why do we need a Domain Name System? What role does the DNS Resolver play in the DNS system? *(Nov 12)*

Domain Name System can map a name to an address and conversely an address to name. The Domain Name System converts domain names into IP numbers. IP numbers uniquely identify hosts on the Internet: however they are difficult to remember. We therefore need a memorable way of identifying hosts. A DNS Resolver is responsible for making requests of the local DNS server in behalf of clients. A DNS Resolver must know the IP address of at least one DNS server. It uses this address to start the DNS Lookup process.

29. List the two types of DNS message.*(May 16)*

- Query
- Response
- **Query message** – consists of the header and question records.
- **Response message** – consists of header, question record, authoritative record and additional record.

30. Define zone.

Zone is defined as a contiguous part which is a server responsible for or has authority over the entire tree (a collection of one or more sub domain within a domain).

31. Why name services are sometimes called as middleware? (Nov/Dec 2012)

- Name services are sometimes called middleware because they fill a gap between applications and the underlying network.

32. Discuss the three main division of the domain name space. *(May 12,16)*

- Domain name space is divided into three different sections: generic domains, country domains & inverse domain.
- Generic domain: Define registered hosts according to their generic behavior, uses generic suffixes.
- Country domain: Uses two characters to identify a country as the last suffix.
- Inverse domain: Finds the domain name given the IP address.

33. What DNS cache issues are involved in changing the IP address of a web server host name? (Nov/Dec 2013)

- An obsolete entry can be a serious problem, since you might get served the wrong page if you contact the "old" owner of a given name. This problem might be minimized by providing a mechanism for sending "DNS update" messages to inform hosts that their entries have gone bad.

34. Present the information contained in a DNS resource record. *(May 17)*

- Different types of resource records can be used to provide DNS-based data about computers on a TCP/IP network. This section describes the following resource records:SOA, NS, A, PTR, CNAME, MX, SRV.

35. Define Name Resolution.

- To improve reliability, some of the name servers can be located outside the zone. The process of looking up a name and finding an address is called name resolution.

36. Define SNMP. *(May 12)*

- **Simple Network Management Protocol** (**SNMP**) is an "Internet-standard protocol for managing devices on IP networks". Devices that typically support SNMP include routers, switches, servers, workstations, printers, & modem. It is used mostly in network management systems to monitor network-attached devices for conditions that warrant administrative attention.

37. State the purpose of SNMP. (Nov/Dec 2011)

- This means we need a protocol that allows us to read, and possibly write, various pieces of state information on different network nodes. The most widely used protocol for this purpose is the Simple Network Management Protocol (SNMP).

38. What is MIB?

- Each agent has its own management information base (MIB), which is a collection of all objects. A management station performs the monitoring function by retrieving the value of MIB objects.

UNIT – II

PART-B & PART – C

1. Explain WWW. ***(Nov 12)***
2. Tabulate the various HTTP request operations. Draw the state transition diagram. (or)Explain in detail about HTTP operation. ***(May 2019, May 2018, May 2017)***
3. Explain about HTTP. Give their uses, state strengths and weaknesses. ***(Nov 10,13)***
4. Explain in detail about FTP. ***(Nov 12, 13), (May 13,***19).
5. Write short notes on (i) Email (ii) HTTP. ***(May/June 2016, Nov/Dec 2015)***
6. Discuss how the Simple Mail Transfer Protocol (SMTP) is useful in electronic mail.
7. (***May 12,15,17)(Nov 13,15)***
8. Illustrate the sequence of events and the respective protocols involved while accessing a web page from a machine when it is connected with internet for the first time. ***(May 2017)***
9. Explain the final delivery of email to the end user using POP3 and IMAP. Or Illustrate the role of POP3 in Electronic mail Applications. (May 2019, May 2018, May 2017, Nov/Dec 2016, May/June 2016, May/June 2015)
10. Discuss about MIME, IMAP and POP3. ***(May 15,17)***
11. Illustrate the features of TELNET. What is the need for network virtual terminal?

(May 13)

12. Describe about Secure Shell (SSH).
13. Explain the role of a DNS on a computer network, including its involvement in the process of a user accessing a web page. ***(May 2017, Nov/Dec 2015, Nov/Dec 2016)***
14. Explain SNMP messages in detail. ***(May 2017, Nov/Dec 2016, May/June 2014)***

SUPPORTIVE ONLINE COURSES

S No	Course provider	Course title	Link
1	Coursera	The Bits and Bytes of Computer Networking	https://www.coursera.org/learn/computer-networking?action=enroll
2	Coursera	Computer Communications specialization	https://www.coursera.org/specializations/computer-communications
3	Udemy	Computer networks for beginners	https://www.udemy.com/course/computer-networks-for-beginners-it-networking-fundamentals/
4	Udemy	Computer Network Cabling: Ethernet Wiring Infrastructure	https://www.udemy.com/course/network-cabling/
5	Udemy	Tech Basics: Cables & Connectors	https://www.udemy.com/course/tech101-cables-and-connectors/

REAL TIME APPLICATIONS IN DAY TO DAY LIFE AND TO INDUSTRY

1. SSH Connection to Server

https://www.youtube.com/watch?v=1LwNQ35w2MA

CONTENT BEYOND THE SYLLABUS

Introduction to Wireless Application Protocol

https://www.youtube.com/watch?v=r3-ooXTUBzY

Assessment Schedule

- **Tentative schedule for the Assessment During 2022-2023 odd semester**

S.NO	Name of the Assessment	Start Date	End Date	Portion
1	IAT 1	16.09.2022	22.09.2022	UNIT 1 & 2
2	IAT 2	02.11.2022	08.11.2022	UNIT 3 & 4
3	REVISION	26.11.2022	29.11.2022	UNIT 5 , 1 & 2
4	MODEL	01.12.2022	10.12.2022	ALL 5 UNITS

Prescribed Text Books & Reference Books

TEXT BOOK

Data Communications and Networking, Behrouz A. Forouzan, McGraw Hill Education, 5th Ed., 2017.

REFERENCES

1. Computer Networking- A Top Down Approach, James F. Kurose, University of Massachusetts and Amherst Keith Ross, 8th Edition, 2021.
2. Computer Networks, Andrew S. Tanenbaum, Sixth Edition, Pearson, 2021.
3. Data Communications and Computer Networks, P.C. Gupta, Prentice-Hall of India, 2006.
4. Computer Networks: A Systems Approach , L. L. Peterson and B. S. Davie, Morgan Kaufmann, 3rd ed., 2003.

Thank you

Disclaimer:

This document is confidential and intended solely for the educational purpose of RMK Group of Educational Institutions. If you have received this document through email in error, please notify the system manager. This document contains proprietary information and is intended only to the respective group / learning community as intended. If you are not the addressee you should not disseminate, distribute or copy through e-mail. Please notify the sender immediately by e-mail if you have received this document by mistake and delete this document from your system. If you are not the intended recipient you are notified that disclosing, copying, distributing or taking any action in reliance on the contents of this information is strictly prohibited.