EDAYATHANGUDY G.S.PILLAY ARTS & SCIENCE COLLEGE
Affiliated To Bharathidasan University, Tiruchirappalli
(NAAC Accredited With 'A' Grade)
NAGAPATTINAM-611002

UNIT V
Computer Networks

Introduction to Application Layer: Introduction, Client Server Programming, WWW and HTTP,FTP, e-mail, TELNET, Secure Shell, Domain Name System, SNMP.

HTTP

The Hyper Text Transfer Protocol (HTTP) is used to define how the client-server programs can be written to retrieve web pages from the Web.

An HTTP client sends a request; an HTTP server returns a response. The server uses the port number 80; the client uses a temporary port number.

Non persistent versus Persistent Connections

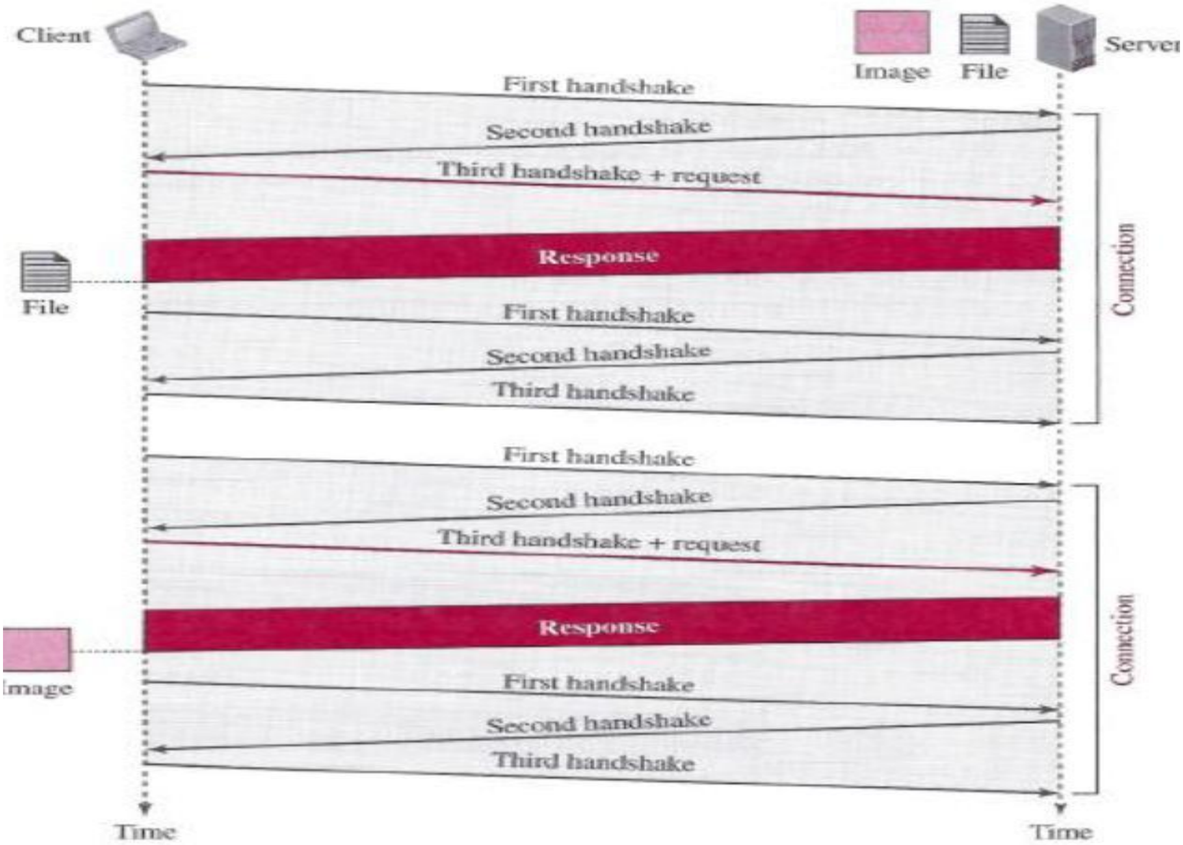
If the web pages, objects to be retrieved, are located on different servers, we do not have any other choice than to create a new TCP connection for retrieving each object. If some of the objects are located on the same server, we have two choices: to retrieve each object using a new TCP connection

Non persistent Connections

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

The following lists the steps in this strategy:

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

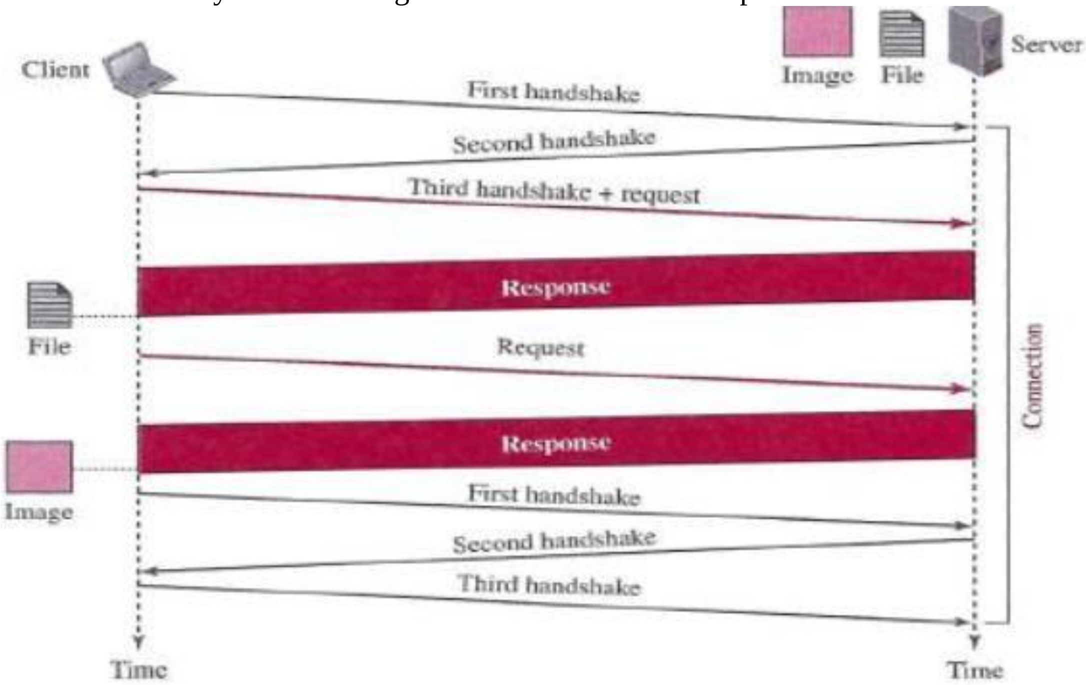

Client
Image
File
Server
First handshake
Second handshake
Third handshake + request
Response
File
First handshake
Second handshake
Third handshake
Connection
First handshake
Second handshake
Third handshake + request
Response
Image
First handshake
Second handshake
Third handshake
Connection
Time
Time

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.

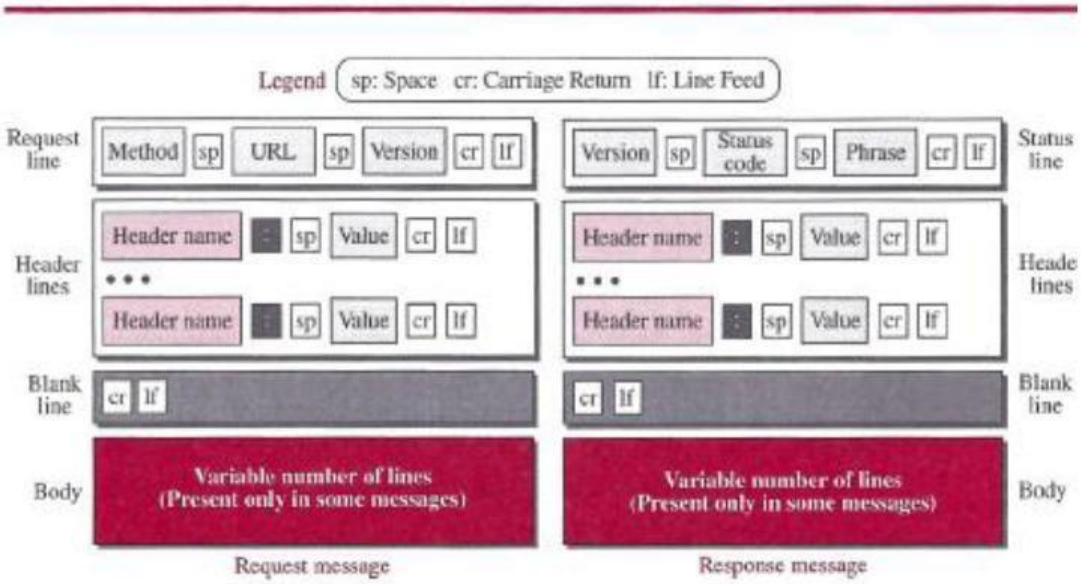


Message Formats

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.

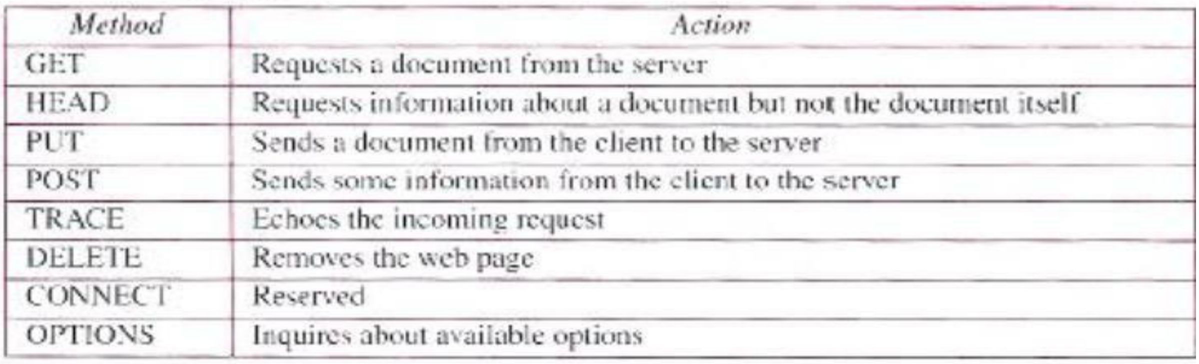
Formats of the request and response messages



Request Message

There are three fields in this line separated by one space and terminated by two characters (carriage return and line feed)

Methods



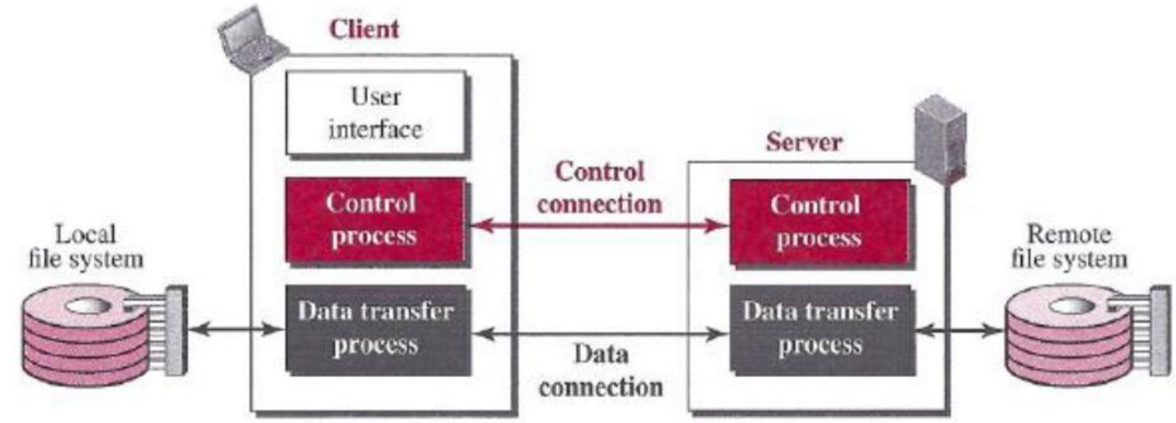
Method	*Action*
GET	Requests a document from the server
HEAD	Requests information about a document but not the document itself
PUT	Sends a document from the client to the server
POST	Sends some information from the client to the server
TRACE	Echoes the incoming request
DELETE	Removes the web page
CONNECT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

FTP

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

Two systems may have different directory structures. All of these problems have been solved by FTP in a very simple and elegant approach.

FTP



Two Connections

- The control connection remains connected during the entire interactive FTP session.
- When a user starts an FTP session, the control connection opens. While the control connection is open, the data connection can be opened and closed multiple times if several files are transferred.
- FTP uses two well-known TCP ports: port 21 is used for the control connection, and port 20 is used for the data connection.

ELECTRONIC MAIL

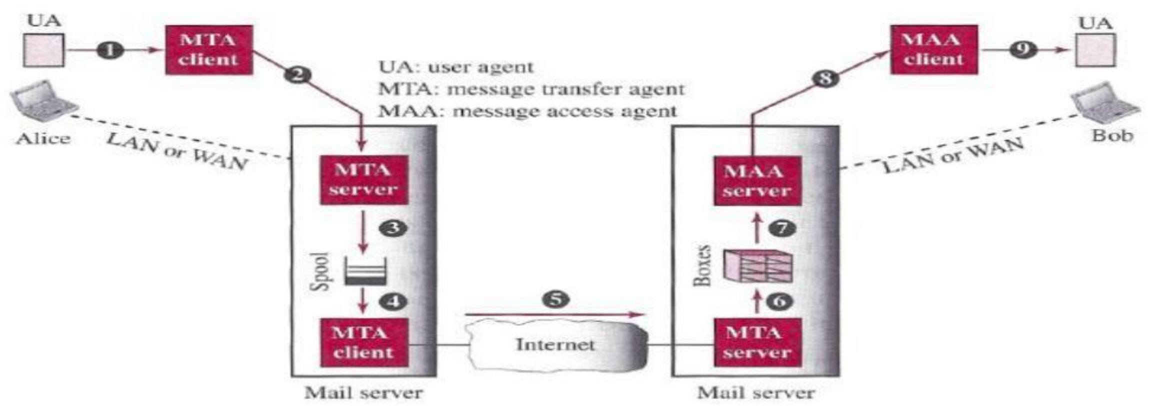
Electronic mail (or e-mail) allows users to exchange messages.

- In an application such as HTTP or FTP, the server program is running all the time, waiting for a
- request from a client.
- When the request arrives, the server provides the service. There is a request and there is a response.

- In the case of electronic mail, the situation is different. First, e-mail is considered a one-way transaction
- The users run only client programs when they want and the intermediate servers apply the client/server paradigm.

Architecture

Common scenario



- 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 use three different *agents:* a user agent (UA), a message transfer agent (MTA), and a message access agent (MAA).

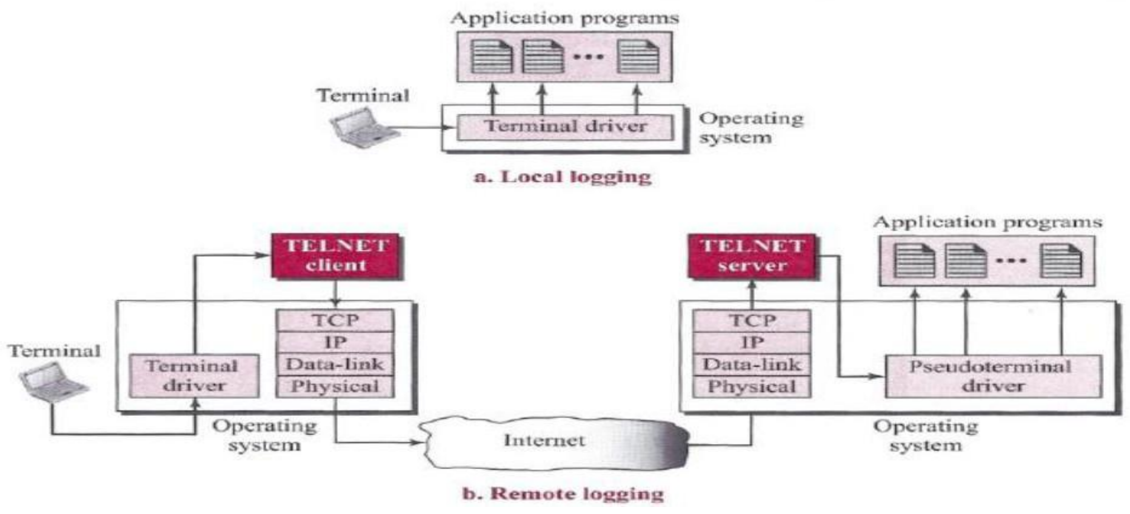
TELNET

One of the original remote logging protocols is **TELNET,** which is an abbreviation for *Terminal Network.*

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),

Local versus Remote Logging

Local versus remote logging



a. Local logging

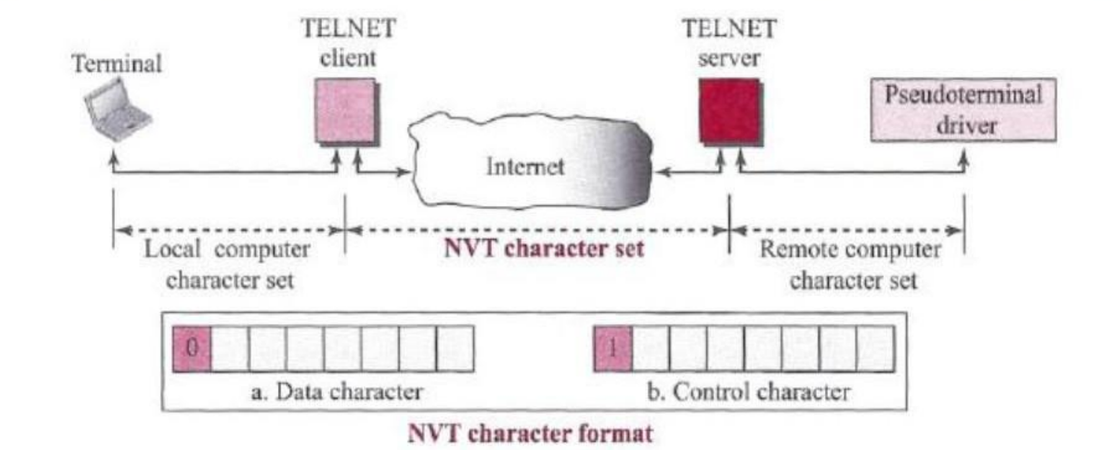
b. Remote logging

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. When a user wants to access an application program or utility located on a remote machine, she performs *remote logging.*

The characters are sent to the TELNET client, which transforms the characters into a universal character set called *Network Virtual Terminal* (NVT) characters

Network Virtual Terminal (NVT)

Concept of 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 .We is dealing with heterogeneous systems. If we want to access any remote computer in the world TELNET solves this problem by defining a universal interface called the *Network Virtual Terminal (NVT)* character set.

The server TELNET, on the other hand, translates data and commands from NVT form into the form acceptable by the remote computer.

SECURE SHELL (SSH)

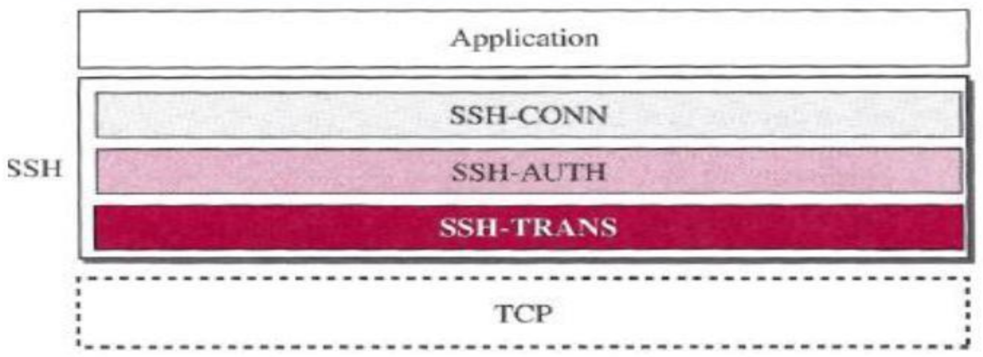
Secure Shell (SSH) is a secure application program that can be used today for several purposes such as remote logging and file transfer; it was originally designed to replace TELNET.

There are two versions of SSH: SSH-l and SSH-2

Components

SSH Transport-Layer Protocol (SSH-TRANS)

Components of SSH



SSH Transport-Layer Protocol (SSH-TRANS)

SSH first uses a protocol that creates secured channel on top of the TCP.

When the procedure implementing this protocol is called, the client and server first use the TCP protocol to establish an insecure connection.

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)
- 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

SSH Connection Protocol (SSH-CONN)

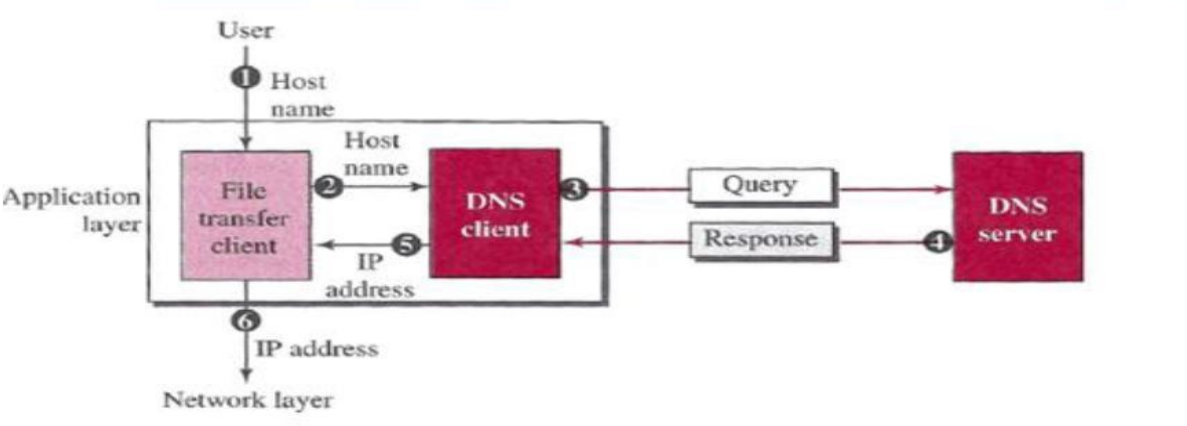
One of the services provided by the SSH-CONN protocol is multiplexing. SSH-CONN takes the secure channel established by the two previous protocols and lets the client create multiple logical channels over it.

Each channel can be used for a different purpose, such as remote logging, file transfer, and so on

DOMAIN NAME SYSTEM (DNS)

The host that needs mapping can contact the closest computer holding the needed information. This method is used by the Domain Name System (DNS).

Purpose of DNS



A user wants to use a file transfer client to access the corresponding file transfer server running on a remote host.

The user knows only the file transfer server name, such as *afilesource.com.*

Name Space

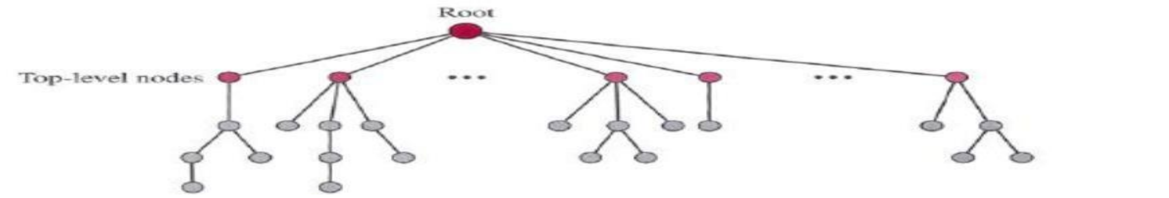
A **name** space that maps each address to a unique name can be organized in two ways: flat or hierarchical.

In a *flat name space,* a name is assigned to an address. A name in this space is a sequence of characters without structure.

In a *hierarchical name space,* each name is made of several parts.

Domain Name Space

Domain name space



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.

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).

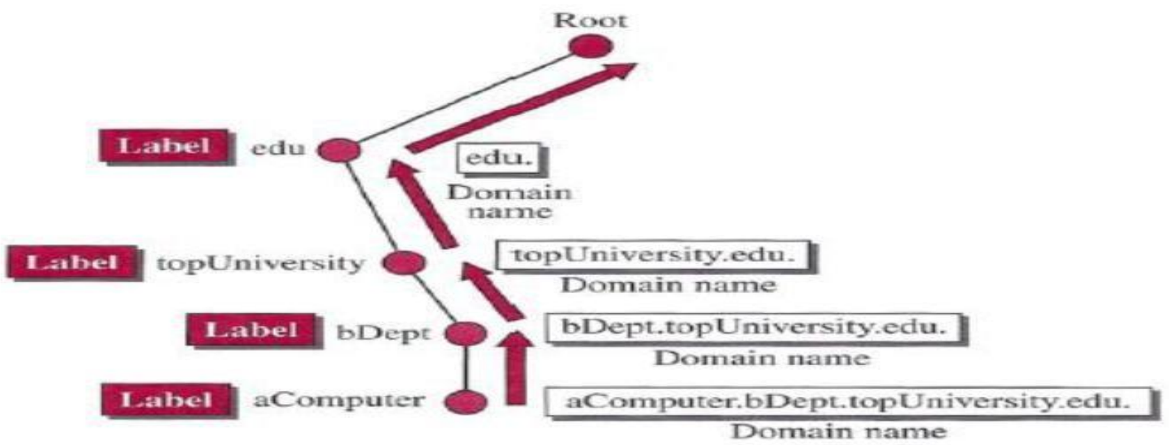
Domain Name

If a label is terminated by a null string, it is called a fully qualified domain name (FQDN).
If a label is not terminated by a null string, it is called a partially qualified domain name PQDN).

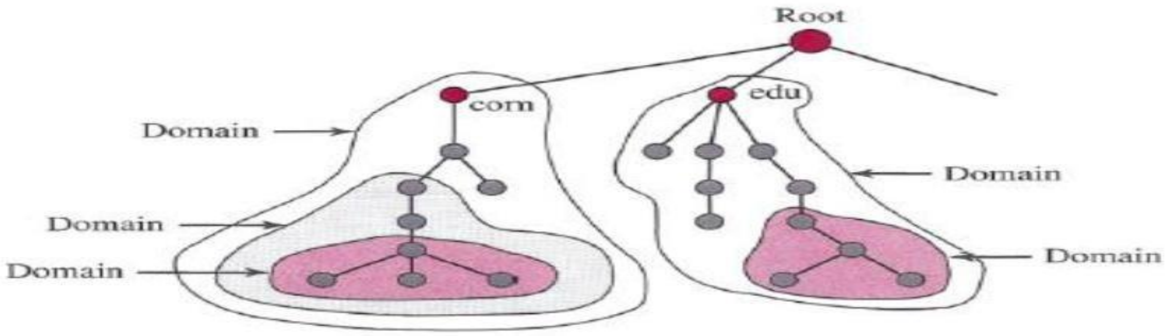
Domain

A domain is a sub tree of the domain name space. The name of the domain is the name of the node at the top of the sub tree.

Domain names and labels



Domains



Distribution of Name Space

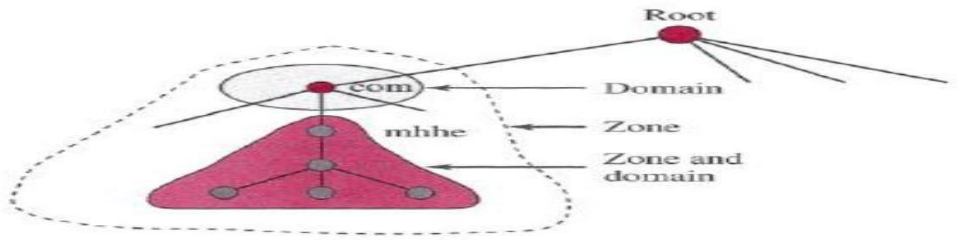
The information contained in the domain name space must be stored.

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.

Zone

Since the complete domain name hierarchy cannot be stored on a single server, it is divided among many servers. What a server is responsible for or has authority over is called a *zone*. The server makes a database called a *zone file* and keeps all the information for every node under that domain.

Zone


Root
com
mhhe
Domain
Zone
Zone and
domain