MODULE 1-PART 2

- Client-server paradigm
- Peer-to-peer paradigm.
- Application Layer Protocols:
  - ✓Web, HTTP, FTP, SMTP, POP3, and DNS.

Application Layer:

- Providing Services

- Highest layer in the suite.

- The protocols in application layer do not provide services to any other protocol in the suite; they only receive services from the protocols in the transport layer.

- This means that protocols can be removed from this layer easily. New protocols can be also added to this layer as long as the new protocol can use the service provided by one of the transport-layer protocols.

- Application layer is the only layer that provides services to the Internet user, the flexibility of the application layer, allows new application protocols to be easily added to the Internet, which has been occurring during the lifetime of the Internet.

Standard and Nonstandard Protocols

- <u>Standard Application-Layer Protocols</u>:
- Application-layer protocols that have been standardized and documented by the Internet authority, and we are using them in our daily interaction with the Internet.
- Each standard protocol is a pair of computer programs that interact with the user and the transport layer to provide a specific service to the user.
- The study of these protocols enables a network manager to easily solve the problems that may occur when using these protocols.

- Nonstandard Application-Layer Protocols [Eg: Skype]
- A programmer can create a nonstandard application-layer program if she can write two programs that provide service to the user by interacting with the transport layer.
- The creation of a nonstandard (proprietary) protocol that does not even need the approval of the Internet authorities if privately used, has made the Internet so popular in the world.
- A private company can create a new customized application protocol to communicate with all of its offices around the world using the service provided by the first four layers of the TCP/IP protocol suite without using any of the standard application programs.

To use the Internet we need two application programs to interact with each other.

- Both application programs be able to request services and provide services, or should the application programs just do one or the other.

<u>Two paradigms:</u>

i. Client-server paradigm

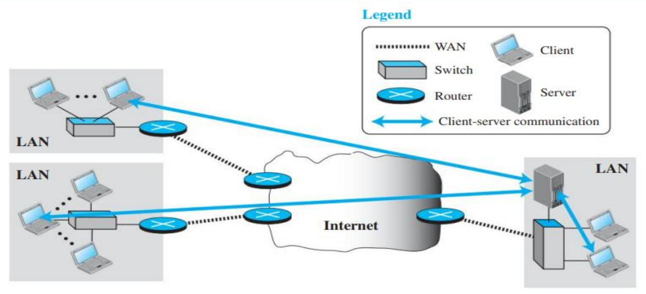
ii. Peer-to-peer paradigm

- Mixed paradigm

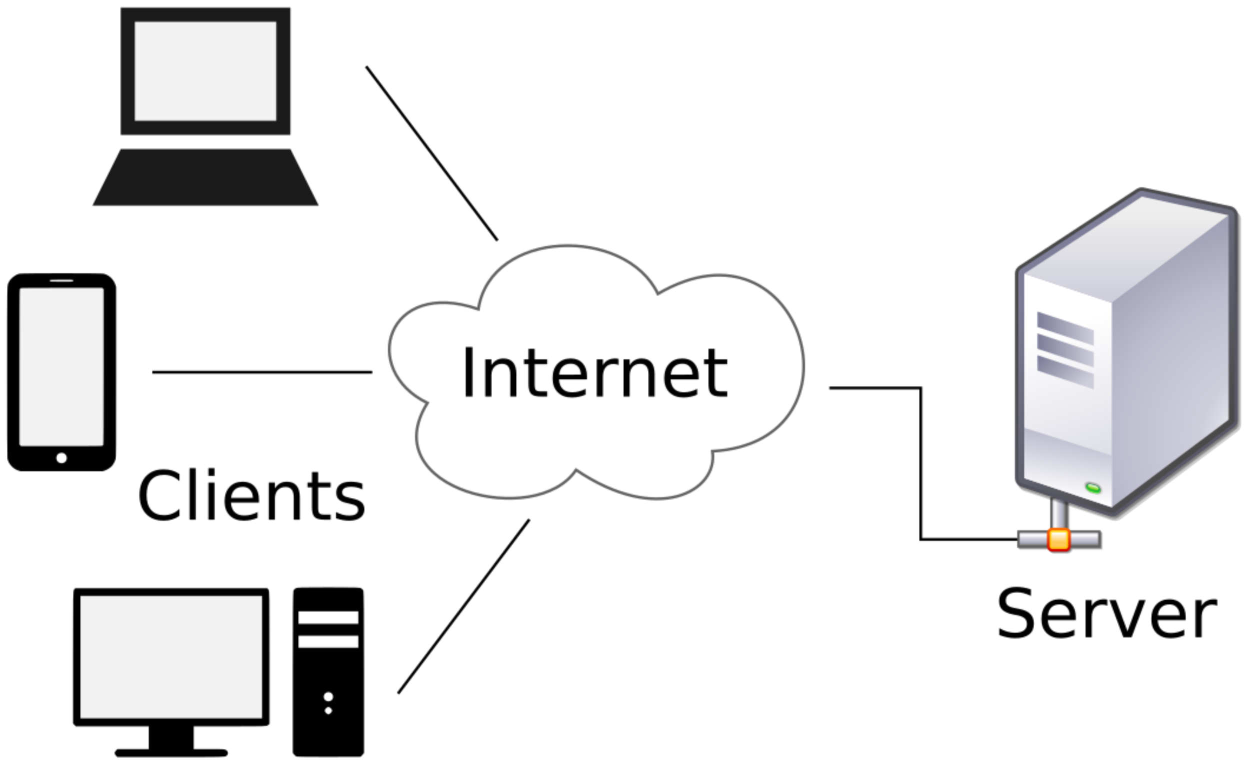
Traditional Paradigm: Client-Server

- In this paradigm, the service provider is an application program, called the server process; it runs continuously, waiting for another application program, called the client process, to make a connection through the Internet and ask for service.
- There are normally some server processes that can provide a specific type of service, but there are many clients that request service from any of these server processes.
- The Server process must be running all the time;
- the client process is started when the client needs to receive service.
- The communication in the client-server paradigm is between two application programs, the role of each program is totally different, we cannot run a client program as a server program or vice versa

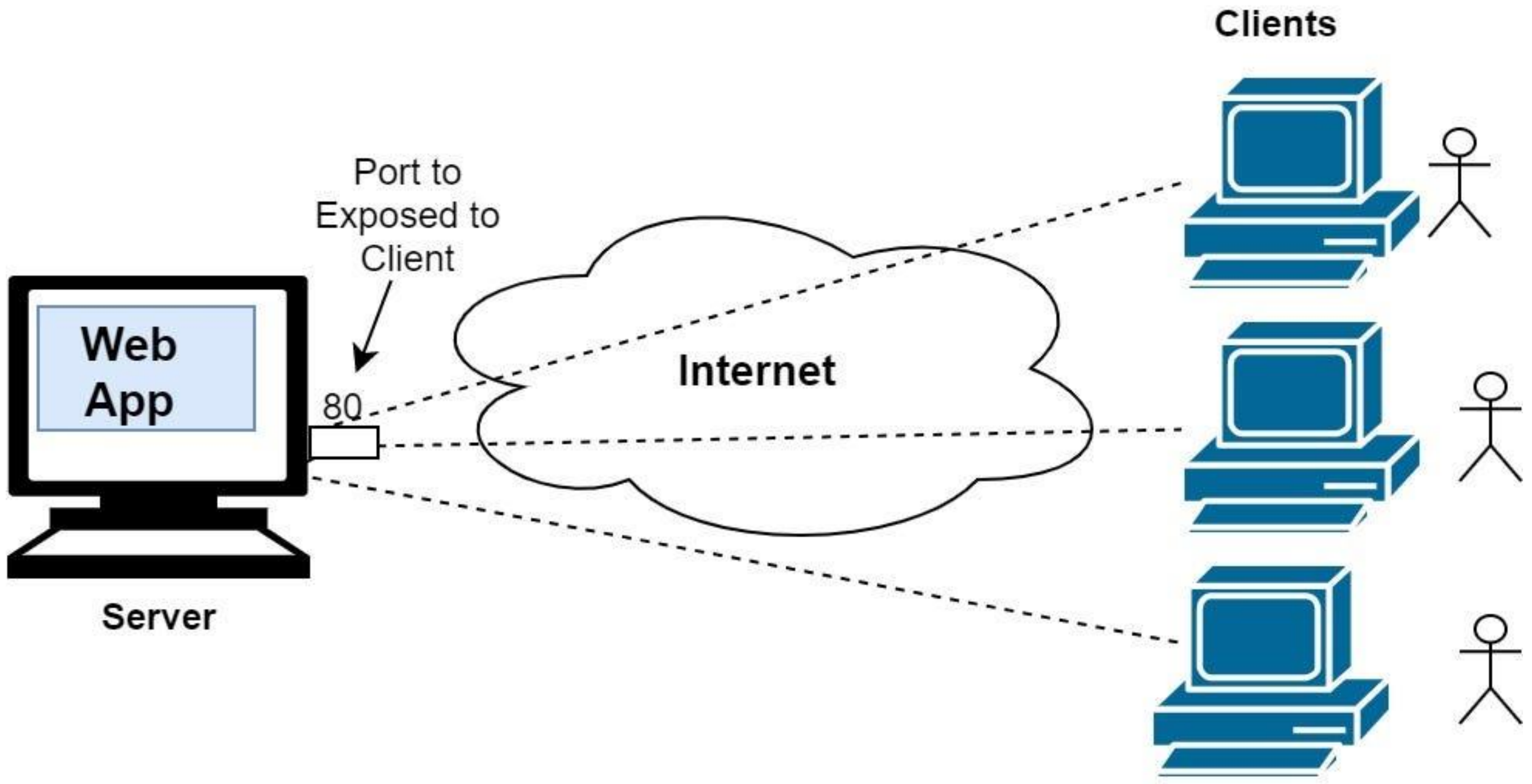
- One problem with this paradigm is that the concentration of the communication load is on the shoulder of the server, which means the server should be a powerful computer.
- Even a powerful computer may become overwhelmed if a large number of clients try to connect to the server at the same time.
- Another problem is that there should be a service provider willing to accept the cost and create a powerful server for a specific service, which means the service must always return some type of income for the server in order to encourage such an arrangement.
- Eg:- World Wide Web (WWW), Hyper Text Transfer Protocol (HTTP), file transfer protocol (FTP), secure shell (SSH), e-mail.



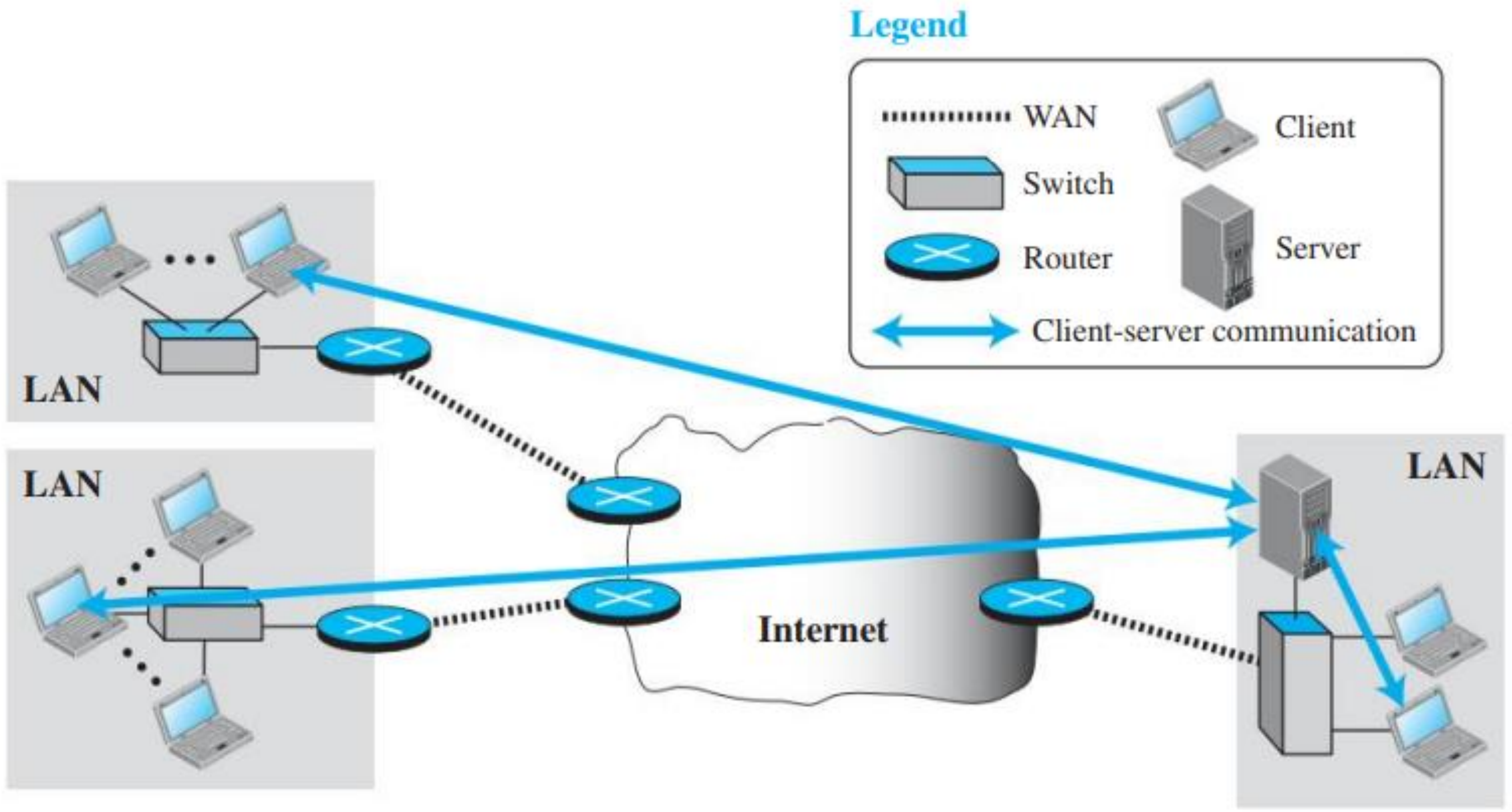
Client-Server Architecture



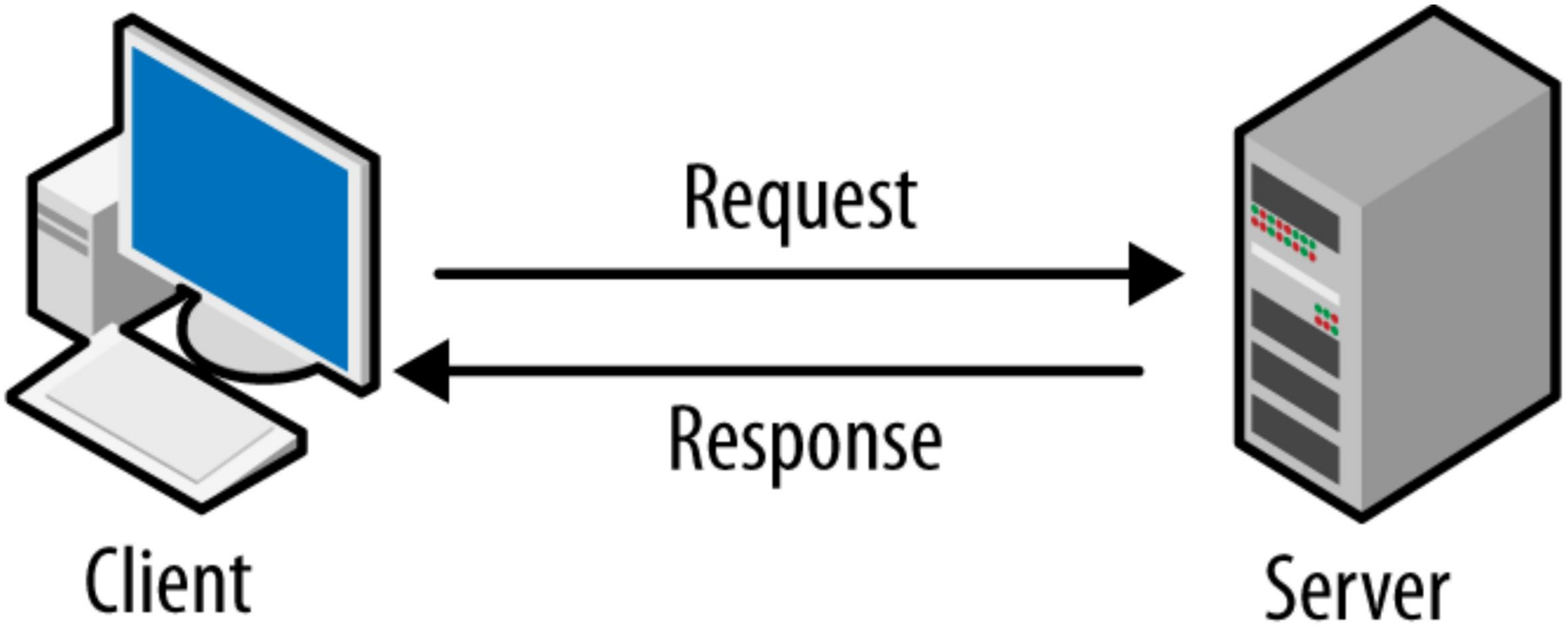
Client-Server Architecture



Client-Server Architecture



Client-Server Architecture



Peer-to-peer paradigm (P2P paradigm):

- In this paradigm, there is no need for a server process to be running all the time and waiting for the client processes to connect. The responsibility is shared between peers.

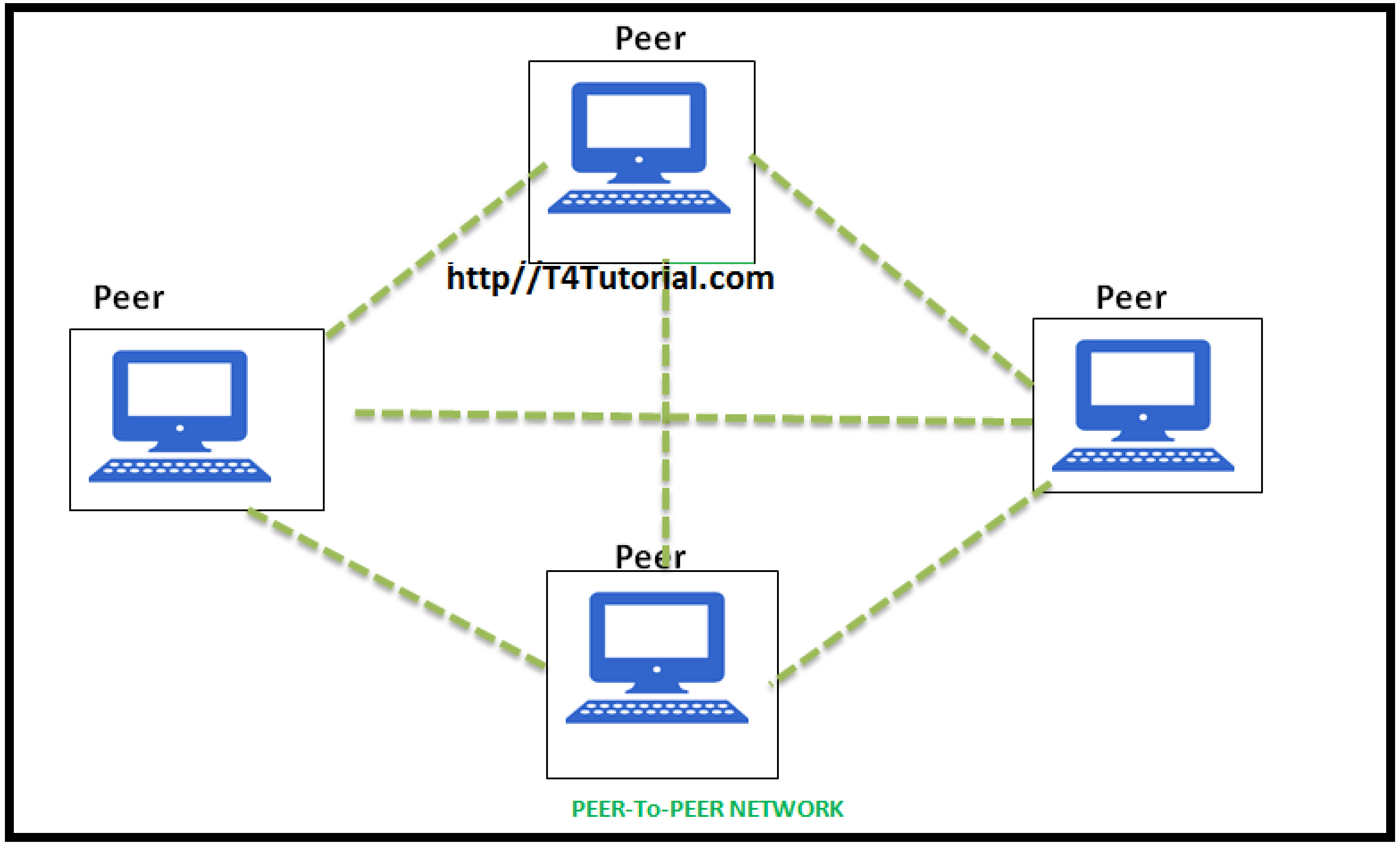
- A computer connected to the Internet can provide service at one time and receive service at another time. A computer can even provide and receive services at the same time.

- Example: Internet telephony.
  - Communication by phone is indeed a peer-to-peer activity; no party needs to be running forever waiting for the other party to call.

➢• Another area in which the peer-to-peer paradigm can be used is when some computers connected to the Internet have something to share with each other.

- For example: File transfer
  - if an Internet user has a file available to share with other Internet users, there is no need for the file holder to become a server and run a server process all the time waiting for other users to connect and retrieve the file.

Peer-to-peer paradigm (P2P paradigm):



Peer-to-peer paradigm (P2P paradigm):

- Scalable and cost-effective in eliminating the need for expensive servers to be running and maintained all the time.

- Challenges

• Security: it is more difficult to create secure communication between distributed services than between those controlled by some dedicated servers.

• Applicability: it appears that not all applications can use this new paradigm. For example, not many Internet users are ready to become involved, if one day the Web can be implemented as a peer-to-peer service.

• Examples: BitTorrent, Skype, IPTV, and Internet telephony

Mixed Paradigm

- An application may choose to use a mixture of the two paradigms by combining the advantages of both.

- For example, a light-load client-server communication can be used to find the address of the peer that can offer a service.

When the address of the peer is found, the actual service can be received from the peer by using the peer-to peer paradigm.

Advantages of Client Server Architecture:

- **Data backup is easy and cost effective** as there is no need to manage the backup on each computer.
- **Performance is better** as the response time is greatly improves because the server is more powerful computer than the other computers in the network.
- **Security is better** as unauthorized access are denied by server computer and all the data goes through the server.
- **Scalability is not an issue** in this Architecture as large number of computers can be connected with server.

Disadvantages of Client Server Architecture :

- In case of **server failure entire network is down.**
- **Server maintenance cost is high** as the server is the main component in this Architecture
- **Cost is high as the server** needs more resources to handle that many client requests and to be able to hold large amount of data.

Advantages of a Peer to Peer Architecture

- Less costly as there is no central server that has to take the backup.
- In case of a computer failure all other computers in the network are not affected and they will continue to work as same as before the failure.
- Installation of peer to peer architecture is quite easy as each computer manages itself.

Disadvantages of a Peer to Peer Architecture

- Each computer has to take the backup rather than a central computer and the security measures are to be taken by all the computers separately.
- Scalability is a issue in a peer to Peer Architecture as connecting each computer to every computer is a headache on a very large network.

Standard Client- Server Applications

- ✓**HTTP and the World Wide Web**
- ✓**FTP**
- ✓**Electronic Mail**
- ✓ **Telnet**
- ✓**Secure Shell(SSH)**
- ✓**Domain Name System(DNS)**

World Wide Web

World Wide Web (WWW, Web)

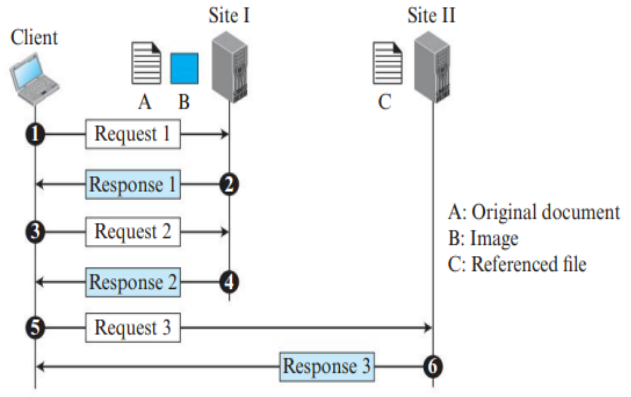
- **Tim Berners-Lee in 1989 at CERN, the European Organization for Nuclear Research, to allow several researchers at different locations throughout Europe to access each others' researches.**
- **The commercial Web started in the early 1990s.**
- **The Web today is a repository of information in which the documents, called Web pages, are distributed all over the world and related documents are linked together.**

Distributed and Linked:

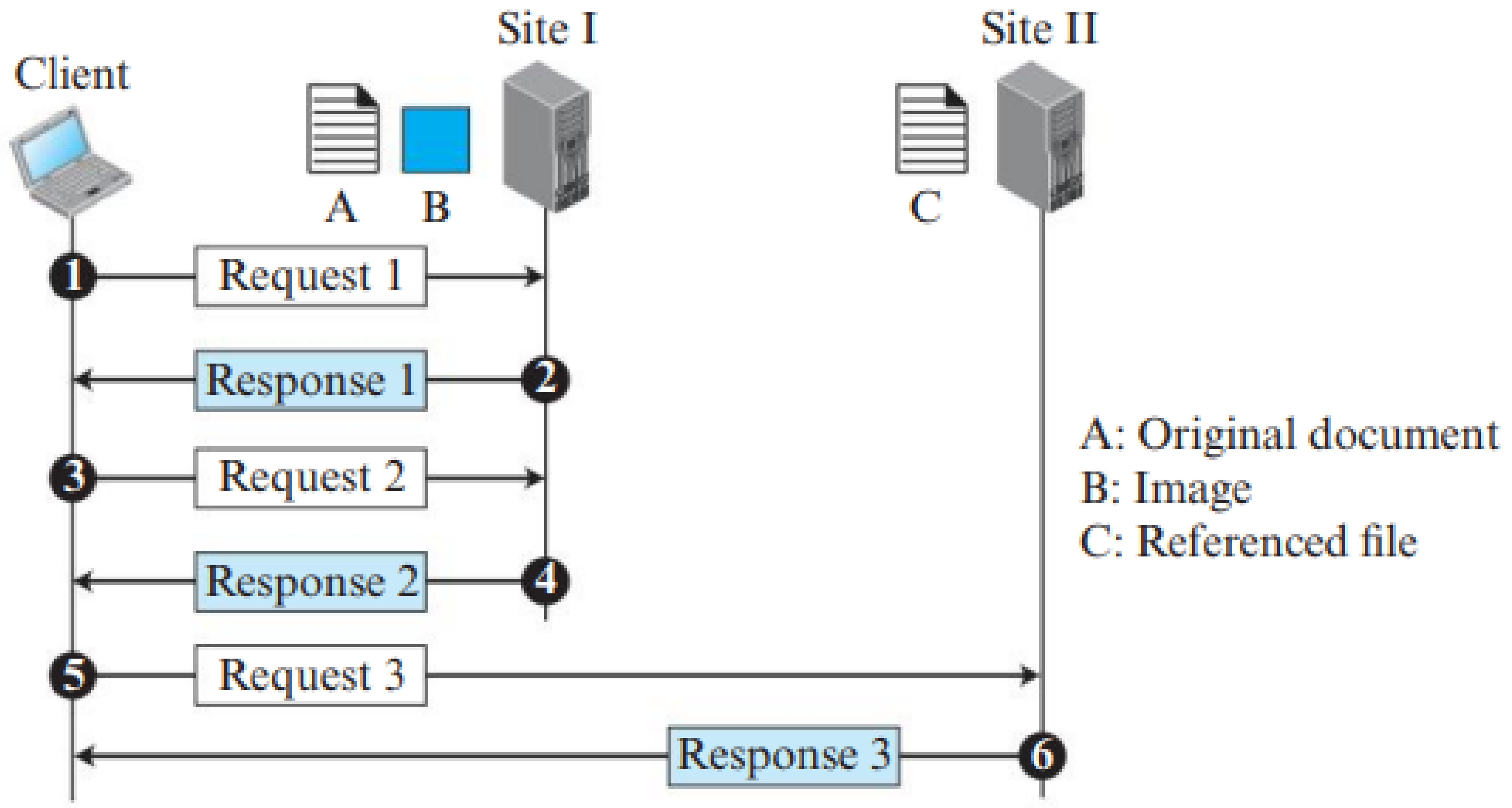
- Distribution allows the growth of the Web. Each web server in the world can add a new web page to the repository and announce it to all Internet users without overloading a few servers.
- Linking allows one web page to refer to another web page stored in another server somewhere else in the world. The linking of web pages was achieved using a concept called hypertext.
- The Web implemented it electronically, to allow the linked document to be retrieved when the link was clicked by the user.
- Hypertext: Linked Text Documents;
- Hypermedia: to show that a web page can be a text document, an image, an audio file, or a video file.
- Web is used to provide electronic shopping and gaming, listen to radio programs or view television programs .

WWW Architecture

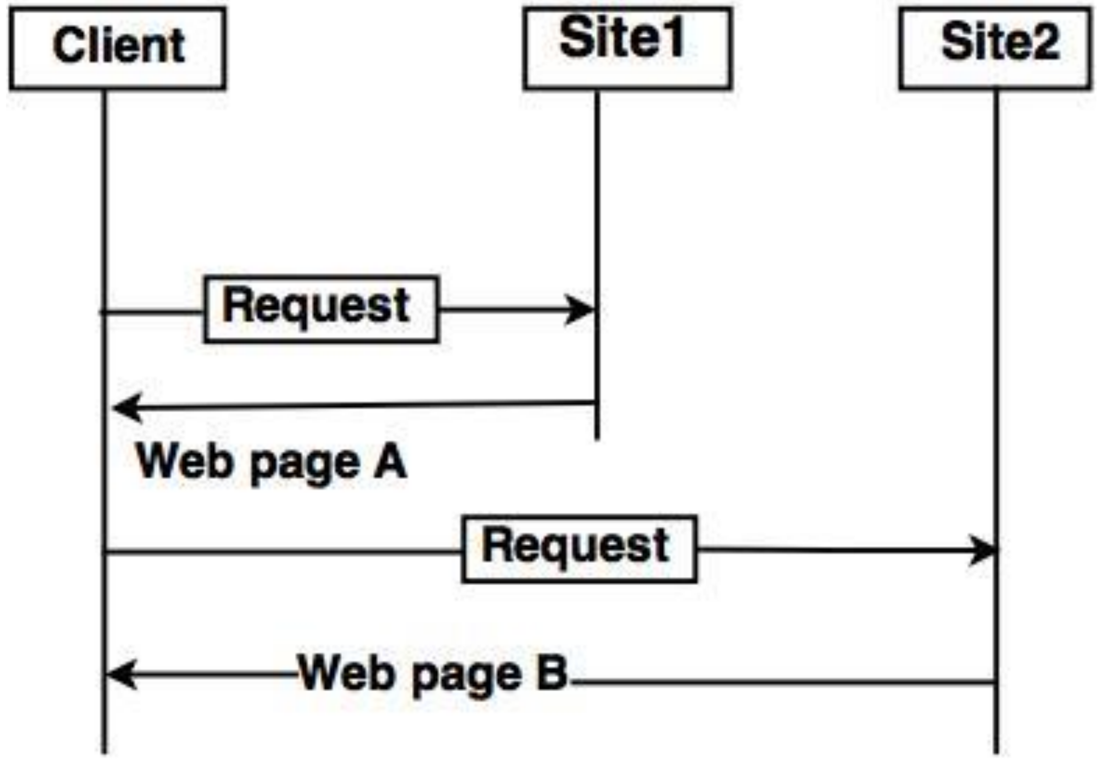
- WWW is a distributed client -server service, in which a client using a browser can access a service using a server, the service provided is distributed over many locations called sites .
- Each site holds one or more documents, referred to as web pages . Each web page, can contain some links to other web pages in the same or other sites .
- 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 .



WWW Architecture

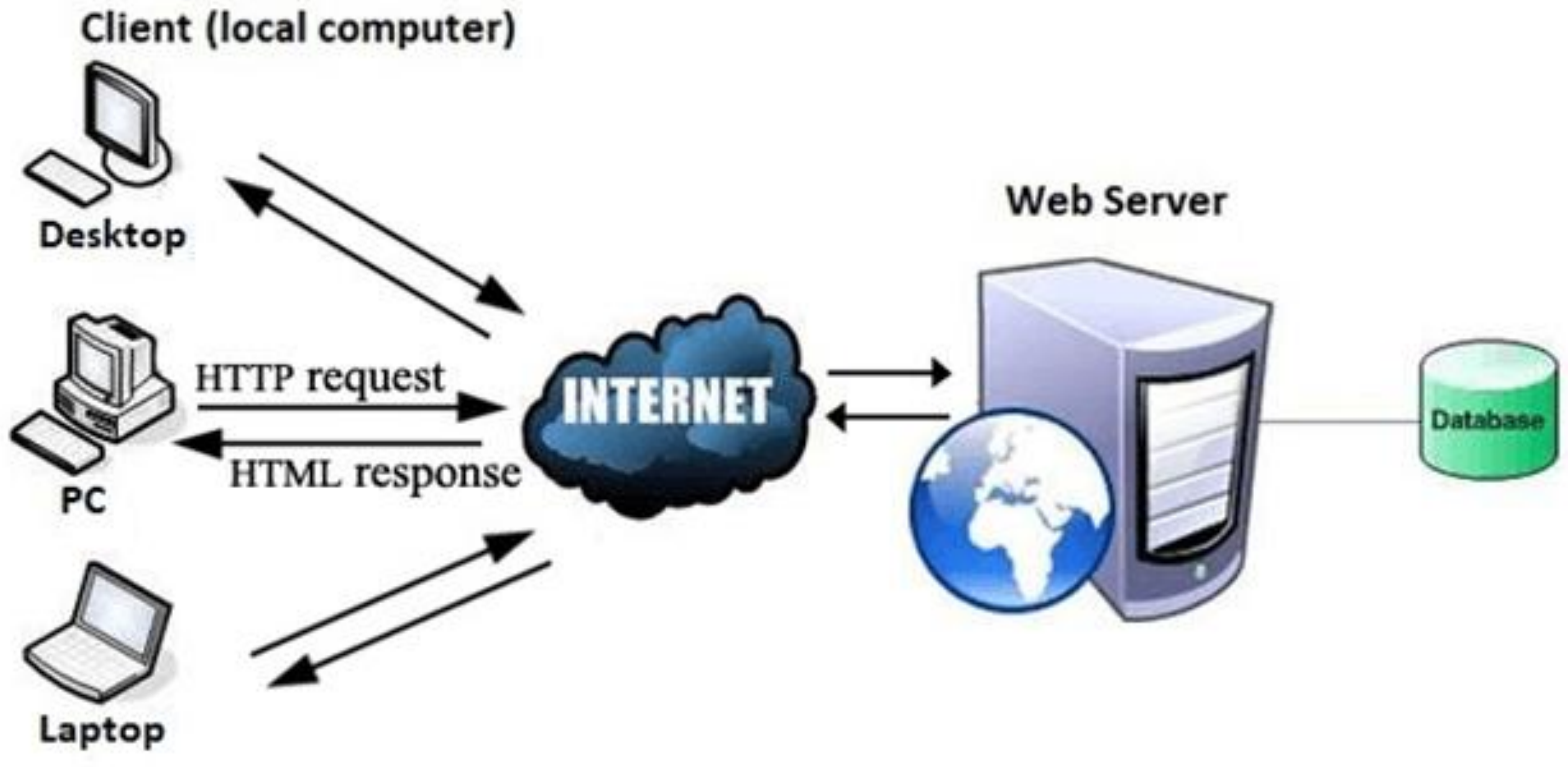


WWW Architecture



Architecture of WWW

WWW Architecture

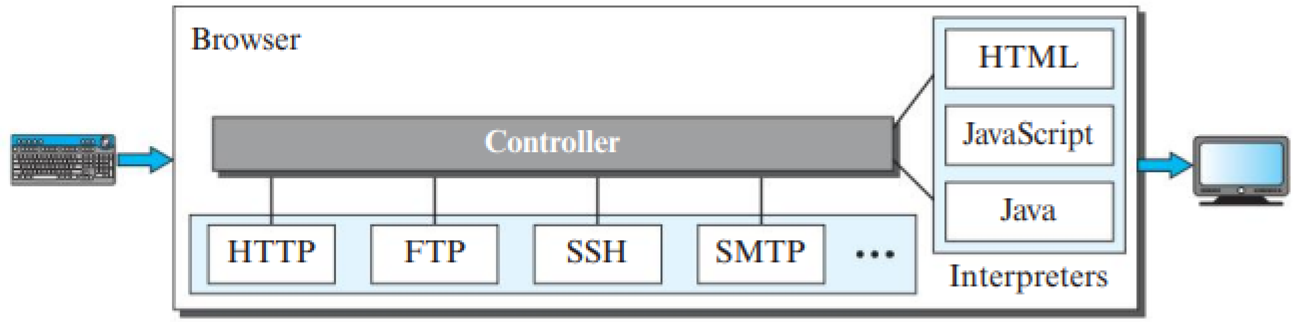


Web Client (Browser) :

- A variety of vendors offer commercial browsers that interpret and display a web page, and all of them use nearly the same architecture.
- Browser usually consists of three parts:
  i. A controller
  ii. Client protocols
  iii. Interpreters.
- 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.
- The client protocol → can be HTTP or FTP.
- The Interpreter → can be HTML, Java, or JavaScript, depending on the type of document.
- Some commercial browsers include Internet Explorer, Netscape Navigator, and Firefox.

World Wide Web (WWW)

Figure 2.9 *Browser*



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 disk.
- A server can also become more efficient through multithreading or multiprocessing. In this case, a server can answer more than one request at a time.
- Some popular web servers include Apache and Microsoft Internet Information Server.

Uniform Resource Locator (URL):

- 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 three identifiers: host, port, and path.
- Before defining the web page, we need to tell the browser what client-server application we want to use, which is called the protocol.
- We need four identifiers to define the web page.
- The first is the type of vehicle to be used to fetch the web page;
- the last three make up the combination that defines the destination object (web page).
- Protocol: The first identifier is the abbreviation for the client-server program that we need in order to access the web page.
  - Eg: HTTP (HyperText Transfer Protocol),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 notations(64.23.56.17); the name is normally the domain name that uniquely defines the host, such as forouzan.com

- **Port:** The port, a 16-bit integer, is normally predefined for the client-server application. If HTTP protocol is used for accessing the web page, the well-known port number is 80.

- **Path:** The path identifies the location and the name of the file in the underlying operating system. The format of the identifier 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.
  - Eg: /top/next/last/myfile is a path that uniquely defines a file named myfile, stored in the directory last, which itself is part of the directory next, which itself is under the directory top.
  - ✓ *To combine these four pieces together, the uniform resource locator (URL) has been designed; it uses three different separators between the four pieces as shown below:*
  - protocol://host/path Used most of the time
  - protocol://host:port/path Used when port number is needed

Web Documents ------!!! (H.w)

- The documents in the WWW can be grouped into three broad categories:
  i. static,
  ii. dynamic, and
  iii. active.

Hypertext Transfer Protocol (HTTP)

- The HyperText Transfer Protocol (HTTP) is a protocol that 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;
  - Port number 8080 is usually used for web servers
- The client uses a temporary port number.
- HTTP uses the services of TCP, which is a connection-oriented and reliable protocol.

Nonpersistent versus Persistent Connections

- **Nonpersistent Connection**

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

Steps:

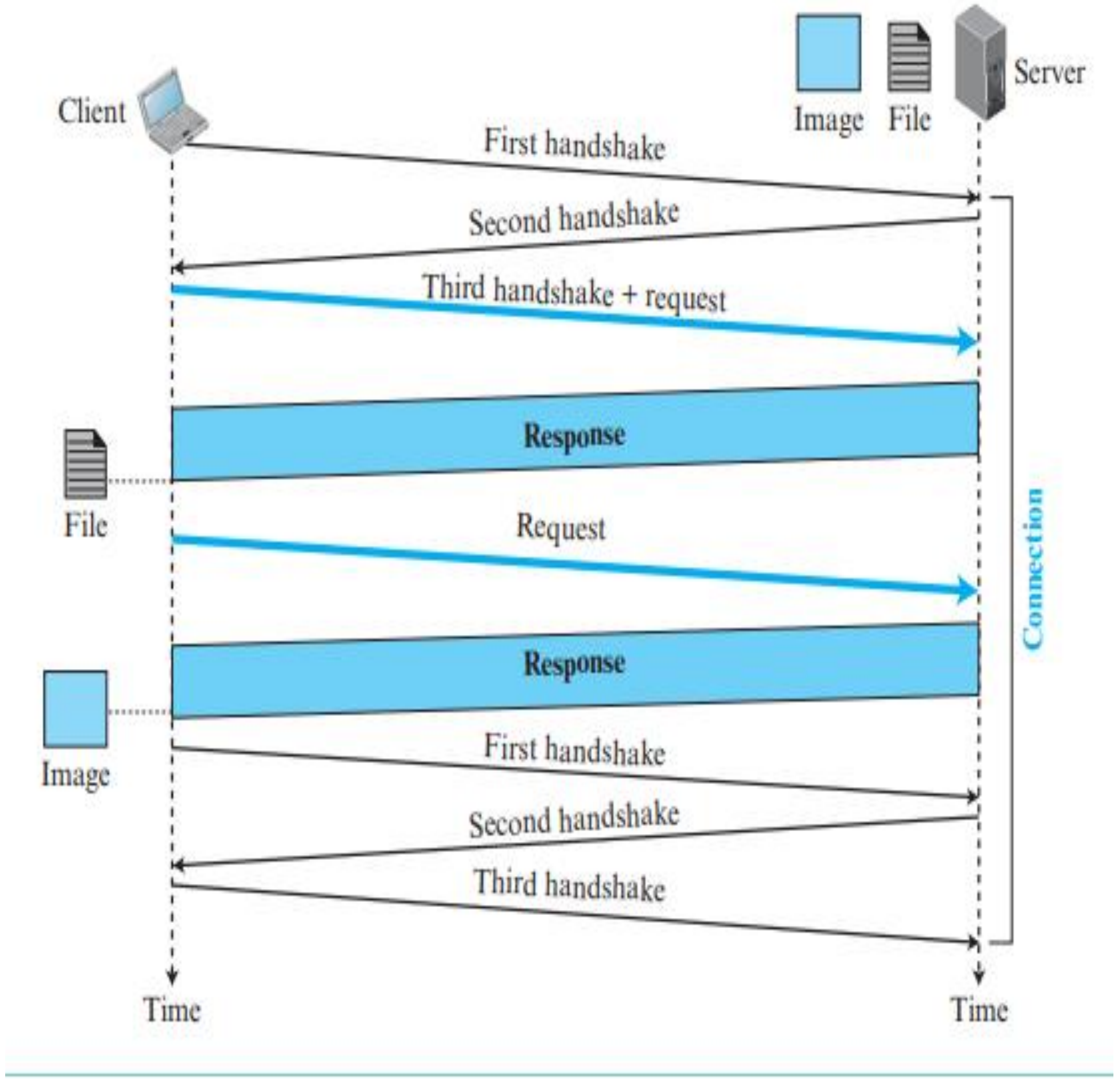
1. **The client opens a TCP connection and sends a request.**
2. **The server sends the response and closes the connection.**
3. **The client reads the data until it encounters an end-of-file marker; it then closes the connection.**

- **The nonpersistent strategy imposes high overhead on the server because the server needs N + 1 different buffers each time a connection is opened.**

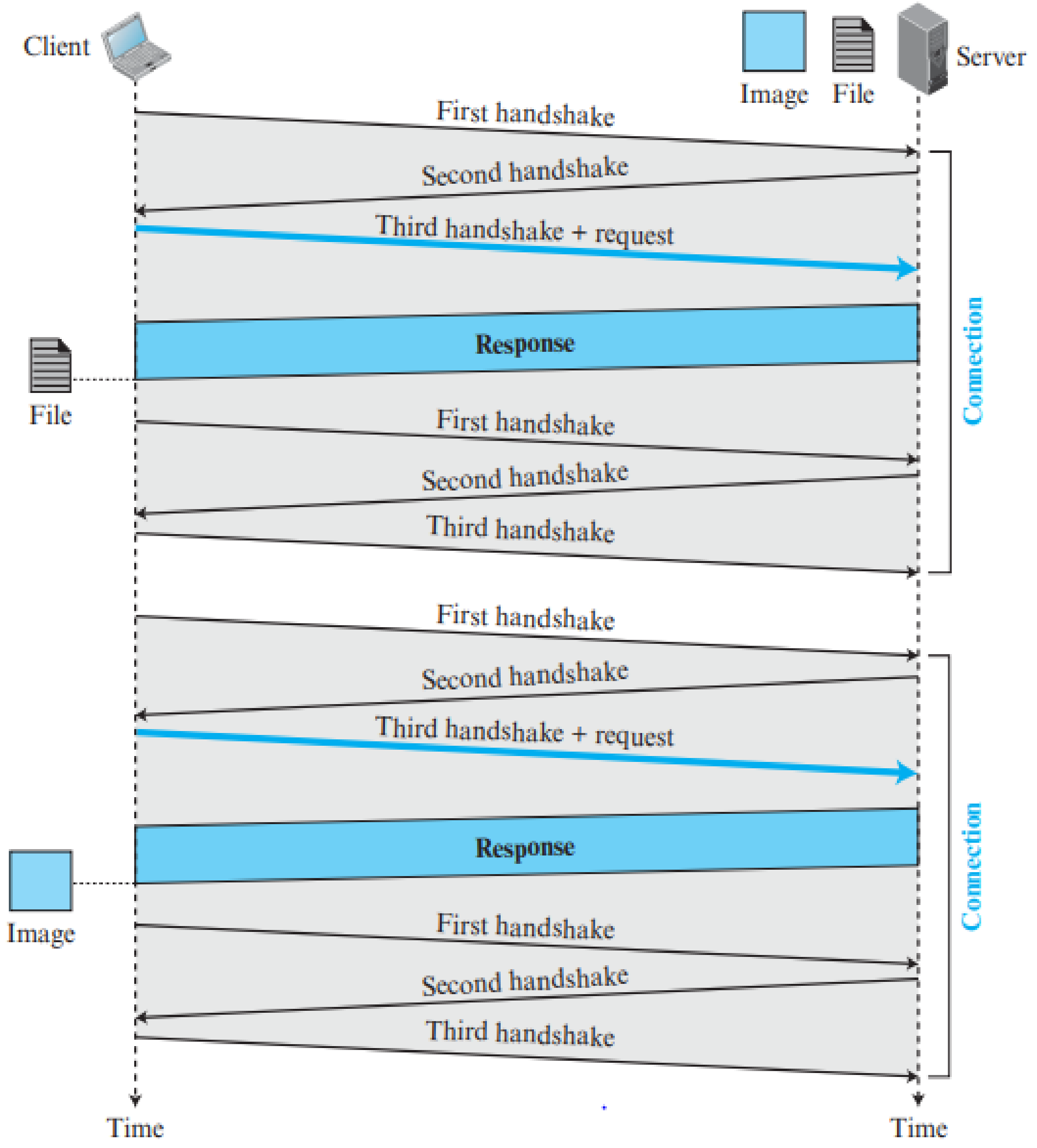
Persistent Connections

- 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.

HTTP



Persistent connection

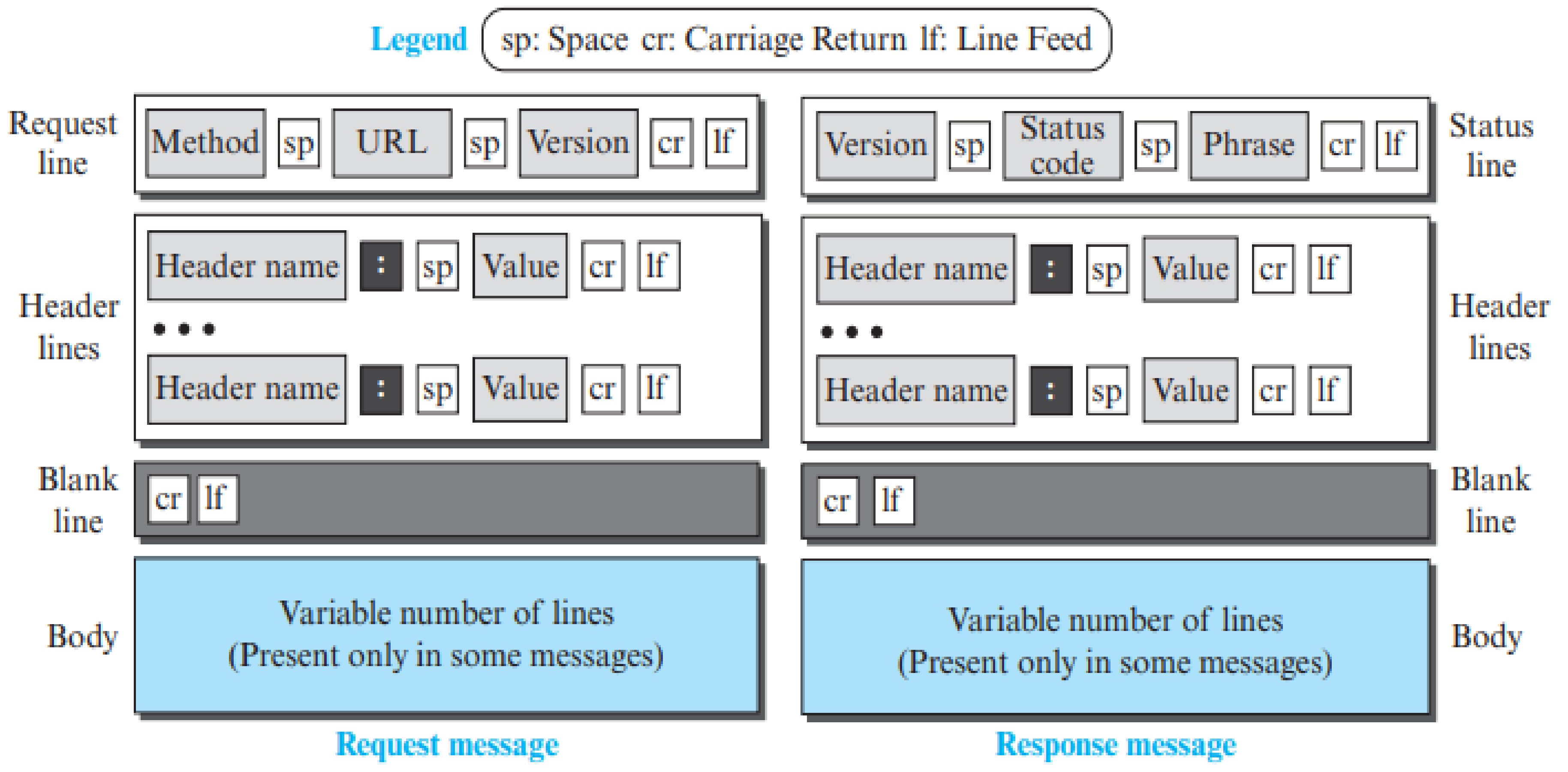


Nonpersistent connection

HTTP

Message Formats: **The HTTP protocol defines the format of the request and response messages.**

Figure 2.12 *Formats of the request and response messages*



Message Formats:

The HTTP protocol defines the format of the request and response messages.

Four sections .

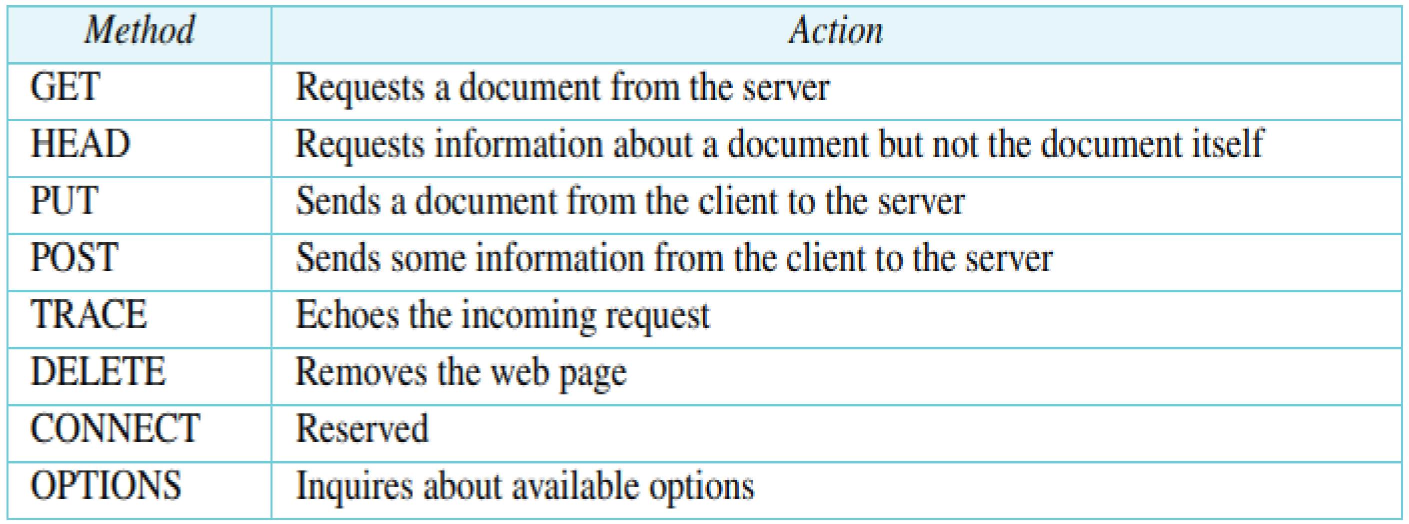
- The first section in the request message is called the request line and in the response message is called the status line .
- Three sections have the same names in the request and response messages, Header lines, Blank lines, Body .

Request Message

- Request line: Three fields in this line separated by one space and terminated by two characters (carriage return and line feed).
- The fields are called method, URL, and version.
- The method field defines the request types. Most of the time, the client uses the GET method to send a request, the body of the message is empty.
- HEAD method is used when the client needs only some information about the web page from the server, such as the last time it was modified. It can also be used to test the validity of a URL. The response message in this case has only the header section; the body section is empty.
- PUT method is the inverse of the GET method; it allows the client to post a new web page on the server.
- POST method is similar to the PUT method, but it is used to send some information to the server to be added to the web page or to modify the web page.

- **TRACE method** is used for debugging; the client asks the server to echo back the request to check whether the server is getting the requests.
- **DELETE method** allows the client to delete a web page on the server if the client has permission to do so.
- **CONNECT method** was originally made as a reserve method; it may be used by proxy servers, as discussed later.
- **OPTIONS method** allows the client to ask about the properties of a web page.

Table 2.1 *Methods*



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

URL: **defines the address and name of the corresponding web page.**

Version: **gives the version of the protocol, current version of HTTP is 1.1.**

Header lines: **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. The value field defines the values associated with each header name.**

Body: **present in a request message, it contains the comment to be sent or the file to be published on the website when the method is PUT or POST.**

Table 2.2 *Request Header Names*

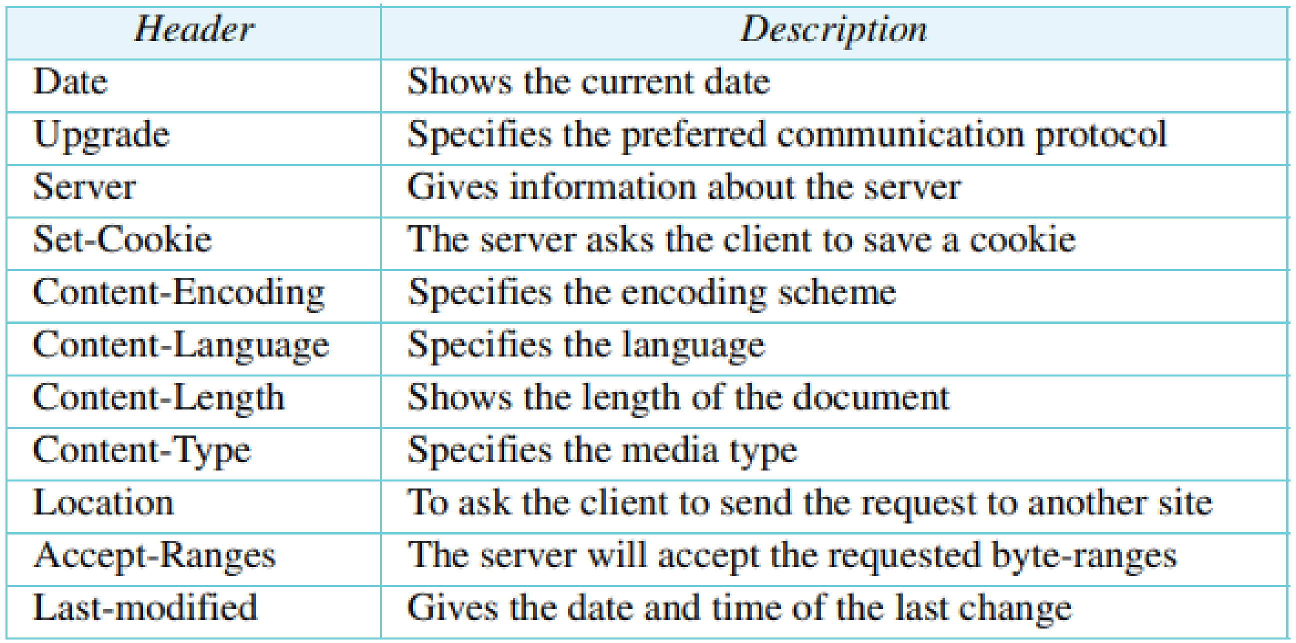
Header	*Description*
User-agent	Identifies the client program
Accept	Shows the media format the client can accept
Accept-charset	Shows the character set the client can handle
Accept-encoding	Shows the encoding scheme the client can handle
Accept-language	Shows the language the client can accept
Authorization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

Response Message

- A response message consists of a status line, header lines, a blank line, and sometimes a body.
- Status line: 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.
- The status code field → status of the request.
- It consists of three digits.
- 100 range is only informational
- 200 range is a successful request.
- 300 range redirect the client to another URL.
- 400 range indicate an error at the client site.
- 500 range indicate an error at the server site.
- The status phrase explains the status code in text form

- **Header Lines:** **zero or more response header lines. Each header line sends additional information from the server to the client.**
- **Each header line has a header name, a colon, a space, and a header value.**
- **Body: contains the document to be sent from the server to the client.**
- **The body is present unless the response is an error message.**

Table 2.3 *Response Header Names*

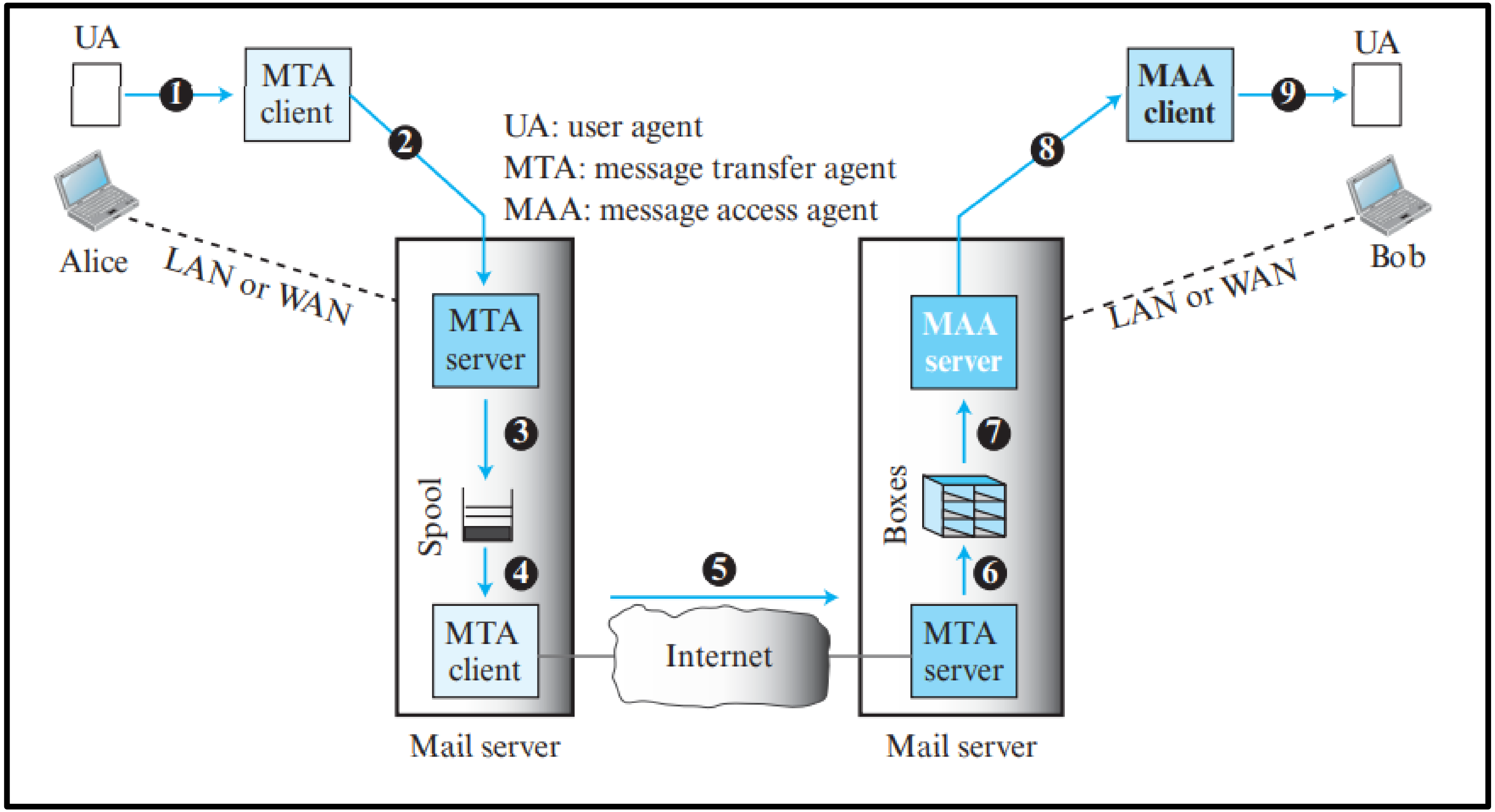


Header	*Description*
Date	Shows the current date
Upgrade	Specifies the preferred communication protocol
Server	Gives information about the server
Set-Cookie	The server asks the client to save a cookie
Content-Encoding	Specifies the encoding scheme
Content-Language	Specifies the language
Content-Length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

Electronic Mail (E-mail)

- 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.
- E-mail is considered a one-way transaction. When Alice sends an e-mail to Bob, she may expect a response, but this is not a mandate. Bob may or may not respond. If he does respond, it is another one-way transaction.
- It is neither feasible nor logical for Bob to run a server program and wait until someone sends an e-mail to him. Bob may turn off his computer when he is not using it.
- Client/server programming should be implemented using some intermediate computers (servers).
- The users run only client programs when they want and the intermediate servers apply the client/server paradigm.

Architecture of e-mail (a common scenario)



E-mail

- A simple e-mail from Alice to Bob takes Nine Different Steps, as shown in the figure.
- Alice and Bob use three different agents: a User Agent (UA), a Mail Transfer Agent (MTA), and a Message Access Agent (MAA).
- When Alice needs to send a message to Bob, she runs a UA program to prepare the message and send it to her mail server.
- The mail server at her site uses a queue (spool) to store messages waiting to be sent.
- The message, however, needs to be sent through the Internet from Alice’s site to Bob’s site using an MTA.
- Here Two Message Transfer Agents are needed: One Client and One Server.
- Like most client-server programs on the Internet, the server needs to run all the time because it does not know when a client will ask for a connection.

E-mail

- The client, on the other hand, can be triggered by the system when there is a message in the queue to be sent.
- The user agent at the Bob site allows Bob to read the received message.
- Bob later uses an MAA client to retrieve the message from an MAA server running on the second server.

E-mail

- The client, can be triggered by the system when there is a message in the queue to be sent. The user agent at the Bob site allows Bob to read the received message.
- Bob later uses an MAA client to retrieve the message from an MAA server running on the second server.
- Bob cannot bypass the mail server and use the MTA server directly. To use the MTA server directly, Bob would need to run the MTA server all the time because he does not know when a message will arrive.
- This implies that Bob must keep his computer on all the time if he is connected to his system through a LAN. If he is connected through a WAN, he must keep the connection up all the time. Neither of these situations is feasible today.
- Bob needs another pair of client-server programs: Message Access Programs.
  - This is because an MTA client-server program is a push program: the client pushes the message to the server.
  - Bob needs a pull program. The client needs to pull the message from the server.

E-mail

User Agent

- First component of an electronic mail system is the user agent (UA).
- It provides service to the user to make the process of sending and receiving a message easier.
- A user agent is a software package (program) that composes, reads, replies to, and forwards messages.
- It also handles local mailboxes on the user computers.
- Two types of user agents:
  - command-driven and
  - GUI-based.
- Command-driven user agents belong to the early days of electronic mail. They are still present as the underlying user agents. 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. Examples of GUI based user agents are Eudora and Outlook.

E-mail

Sending Mail

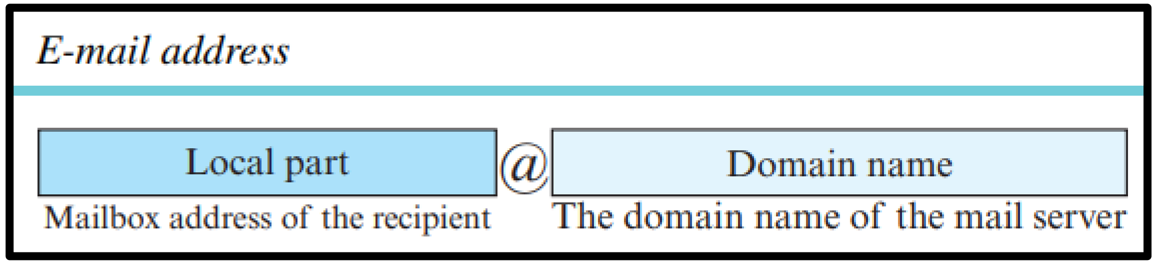
- To send mail, the user, through the UA, creates mail that looks very similar to postal mail. It has an envelope and a message. The envelope usually contains the sender address, the receiver address, and other information. The message contains the header and the body.
- The header of the message defines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

E-mail

- ***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:**
    1. **a local part** and
    2. **a domain name**, separated by an **@ sign.**


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

- The local part → name of a special file, called the user mailbox.
- Domain name → assigned to each mail exchanger, either comes from the DNS database or is a logical name (for example, the name of the organization).
- Eg: principal@amaljyothi.ac.in

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.

SMTP PROTOCOL

- SMTP (OUTGOING/PUSH PROTOCOL)
- With email, the incoming and outgoing mail is handled differently.
- The Simple Mail Transfer Protocol (SMTP), which is an Email Delivery Protocol.
- It's used to send mail over the internet.
- SMTP contains information regarding the transmission details of an email message and is specifically used for outgoing mail.
- A Mail Transfer Agent (MTA) is a server program that uses SMTP to deliver emails.
- There are two types of MTA:
  1) A client-based MTA which involves installing software to access emails (such as Outlook)
  2) A web-based MTA which is accessed through a web browser (GMail for example).
- SMTP relies on the TCP port 25. When an email is sent, Port 25 is typically used to route the message.

SMTP

- The SMTP has all the necessary information about the recipient to send the message from its server on to the email recipients MTA server.
- The next step involves transferring the message between mail servers.
- The SMTP has done its job of routing the message to the destination server.
- The message is now in the recipient’s mail server (MTA).
- The receiving server will store the message and make it available to the recipient who can access it via web, POP, or IMAP protocols.
- To retrieve email, supporting protocol must be used.
- There are two main protocols, POP3 and IMAP.

SMTP

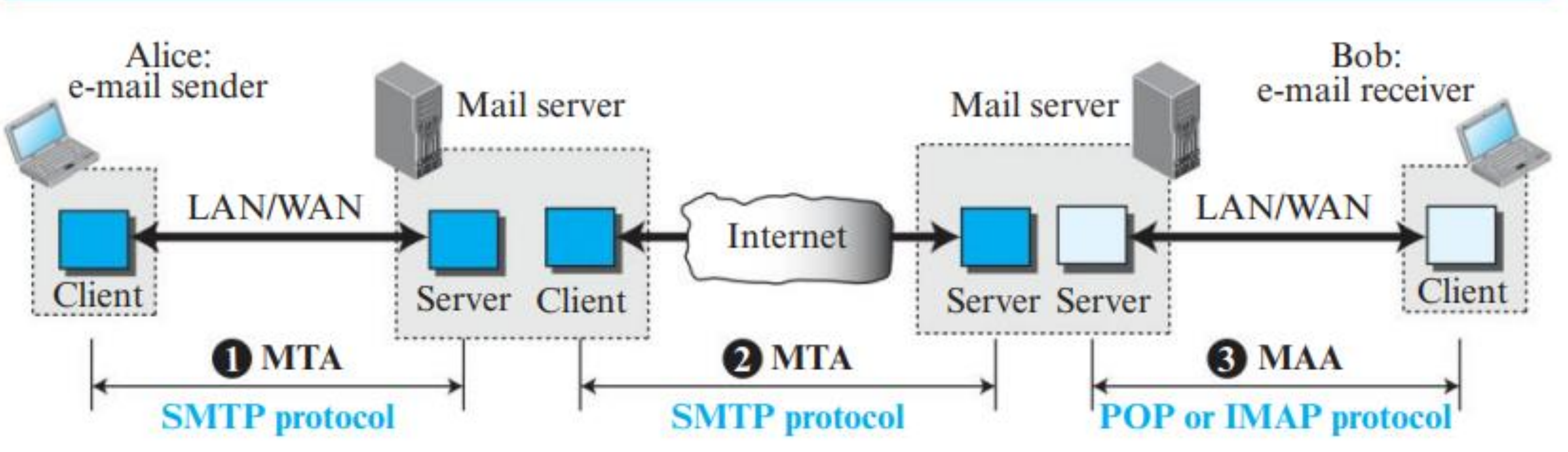


Figure 2.22 *Protocols used in electronic mail*

- **The formal protocol that defines the MTA client and server in the Internet is called Simple Mail Transfer Protocol (SMTP).**
- **SMTP is used two times, between the sender and the sender's mail server and between the two mail servers.**
- **SMTP simply defines how commands and responses must be sent back and forth.**

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.**

SMTP Commands

The following table describes some of the SMTP commands:

HELLO

Senders Host Name. Identifies itself .This command initiates the SMTP conversation.

MAIL FROM

This indicates the sender's address.

RCPT TO

It identifies the recipient of the mail. In order to deliver similar message to multiple users this command can be repeated multiple times.

SIZE

This command let the server know the size of attached message in bytes.

DATA

The DATA command signifies that a stream of data will follow. Here stream of data refers to the body of the message.

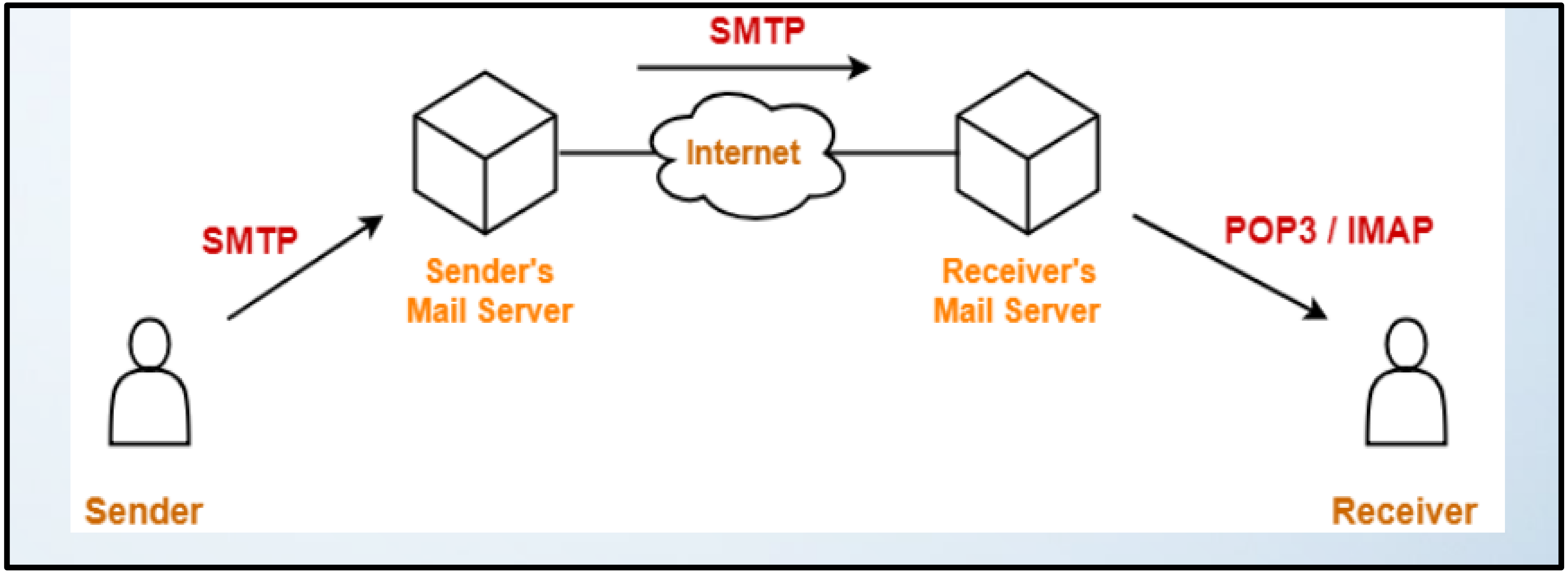
Table 2.6 *SMTP Commands*

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

POP (INCOMING/PULL PROTOCOL)

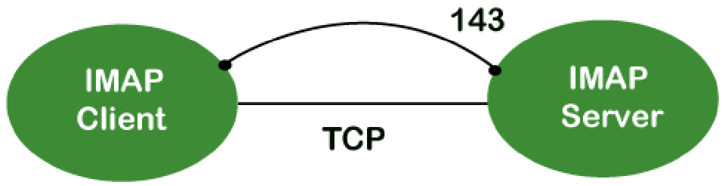
- **POP** stands for Post Office Protocol.
- This piece of software is used for retrieving email.
- Just like a visit to a post office you can drop in, pick up your mail and leave.
- POP3 gives an email user access to their emails stored in their user account on that server.
- You don’t need to stay online for the emails to come through.

SMTP and POP


SMTP
Internet
SMTP
Sender's Mail Server
Receiver's Mail Server
POP3 / IMAP
Sender
Receiver

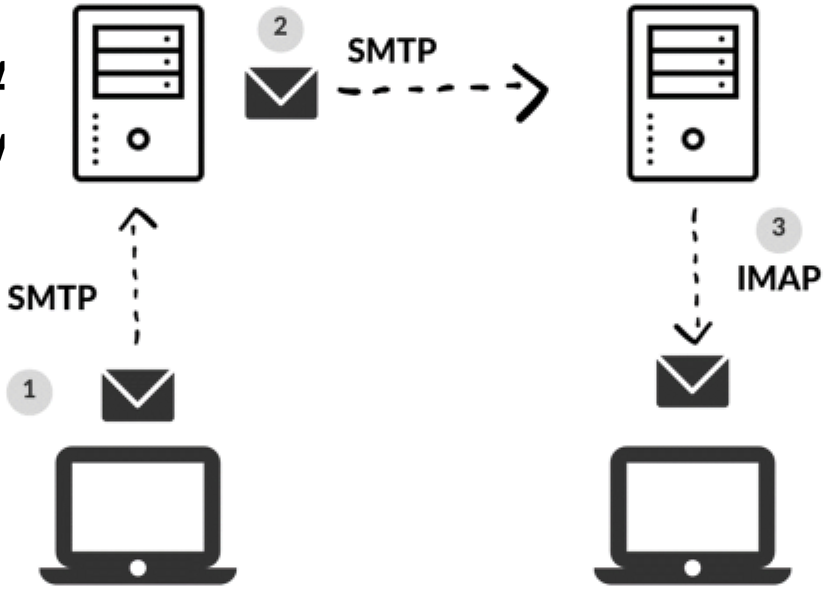
IMAP Protocol

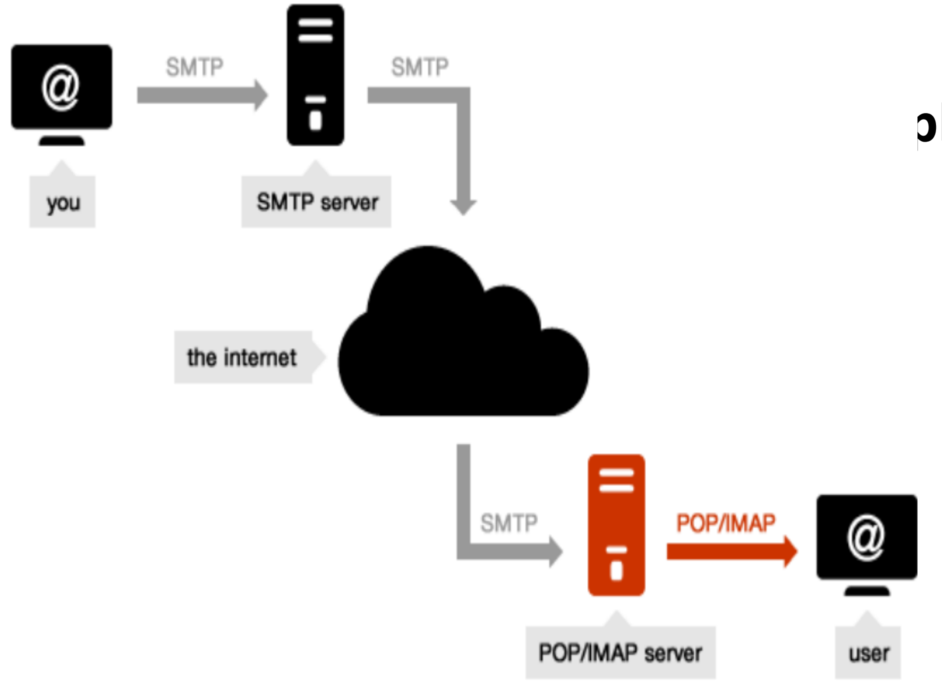
- IMAP stands for Internet Message Access Protocol.
- It is an application layer protocol which is used to receive the emails from the mail server.
- It is the most commonly used protocols like POP3 for retrieving the emails.
- It also follows the client/server model.
- On one side, we have an IMAP client, which is a process running on a computer.
- On the other side, we have an IMAP server, which is also a process running on another computer.
- Both computers are connected through a network.
-



IMAP (INCOMING/PULL PROTOCOL)

- It also follows the client/server model. On one side, we have an IMAP client, which is a process running on a computer. On the other side, we have an IMAP server, which is also a process running on another computer. Both computers are connected through a network.
- IMAP is an mail protocol used to access a mailbox on a remote server from a local email client
- IM vide more c ple dev





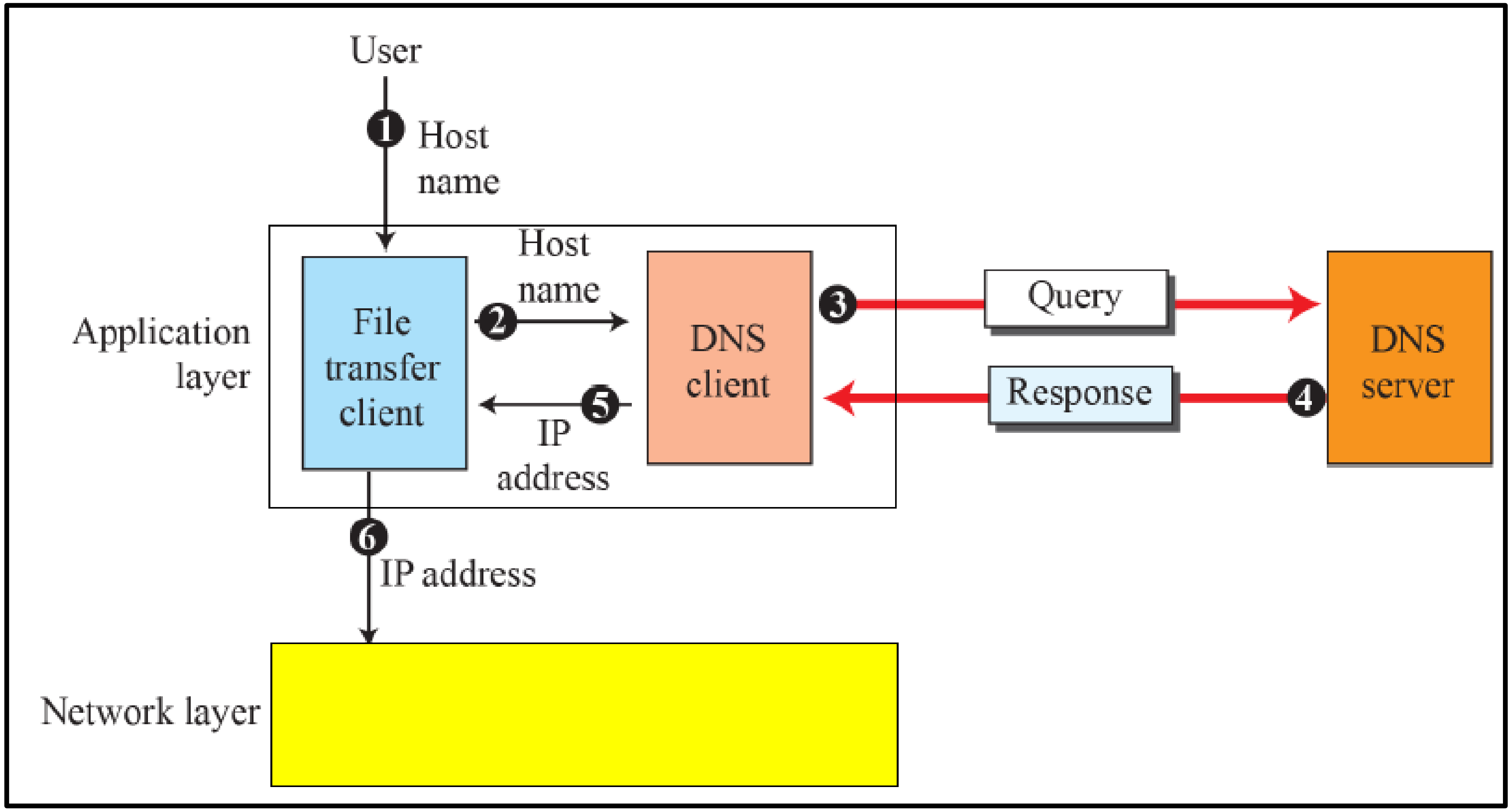
DNS (Domain Name System)

- It is a hierarchical decentralized naming system for computers, services, or other resources connected to the internet or a private network.
- It translates domain names, which are human-readable addresses like "example.com," into numerical IP addresses that computers use to identify each other on the network.
- When a user enters a domain name into their web browser or other network application, the DNS resolver (typically provided by the user's internet service provider or another third party) queries DNS servers to find the corresponding IP address for that domain name.

DNS (Domain Name System)

- The Domain Name System (DNS) is a directory lookup service that provides a mapping between the name of a host on the Internet and its numerical address.
- To identify an entity TCP/IP protocols use the IP address which uniquely identifies connection of host to the internet.
- People prefer to use names instead of numerical address.
- A domain name can be any combination of letters and numbers, and it can be used in combination of the various domain name extensions, such as .com, .net and more.

Purpose of DNS



DNS (Domain Name System)

- Internet Corporation for Assigned Names and Numbers (ICANN) manages the domain names system.
- It is a non-profit organization that creates and implements the policies for domain names.
- ICANN gives permission to companies called Domain Name Registrars for selling domain names.

Namespace:

- Names must be **unique** because addresses are unique.
- A namespace that **map each address** to **unique name** can be organized in two ways:
  1. Flat Namespace
  2. Hierarchical Namespace

✓ Flat Namespace- A name in this space is a **sequence of characters without structure.**

- The names may or may not have common section
- It cannot be used in large system such as internet.

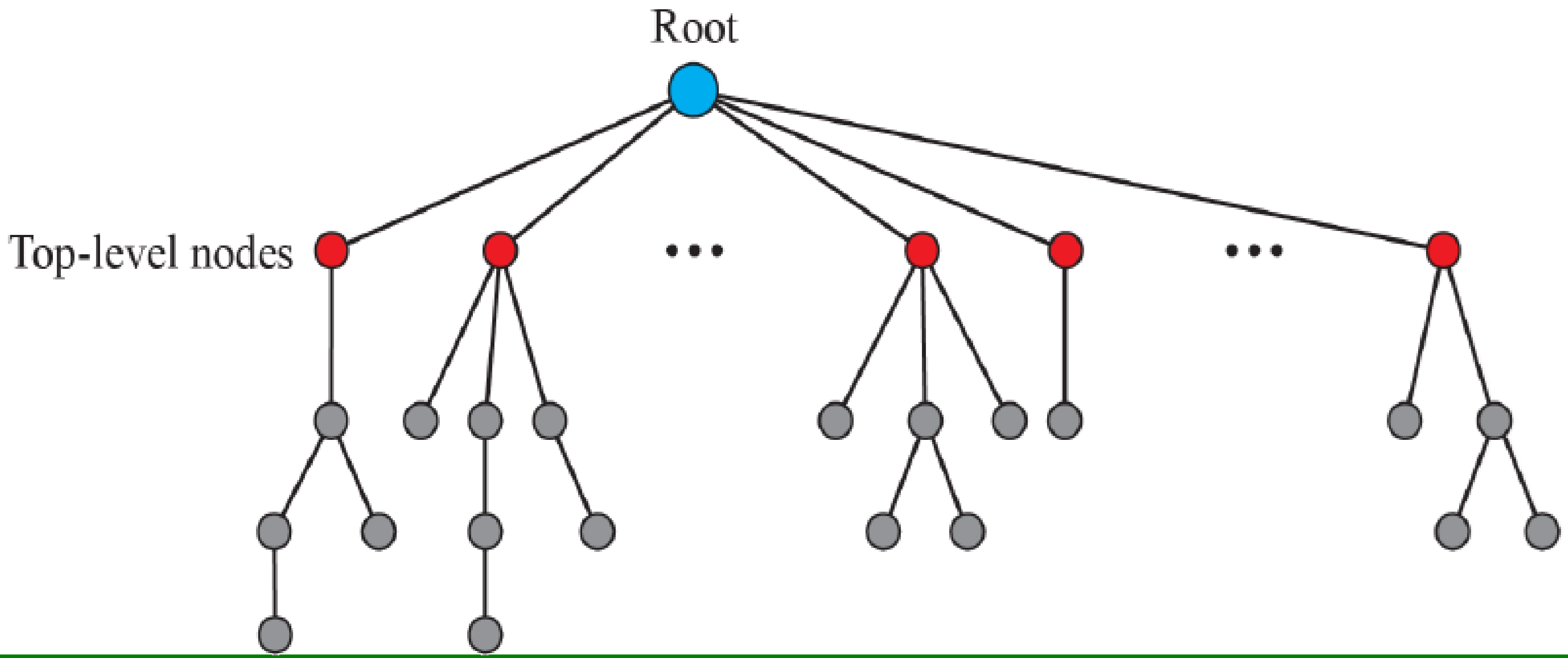
Namespace:

- ❑ Hierarchical Namespace - In Hierarchical Namespace each name is made up of several parts .

- ➢ The first part can define the nature of organization.
- ➢ Second part define the name of organization.
- ➢ Third part can define the department of an organization.
- ➢ In this case the authority to assign and control the namespace can be decentralized.

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;



DNS (Domain Name System)

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.

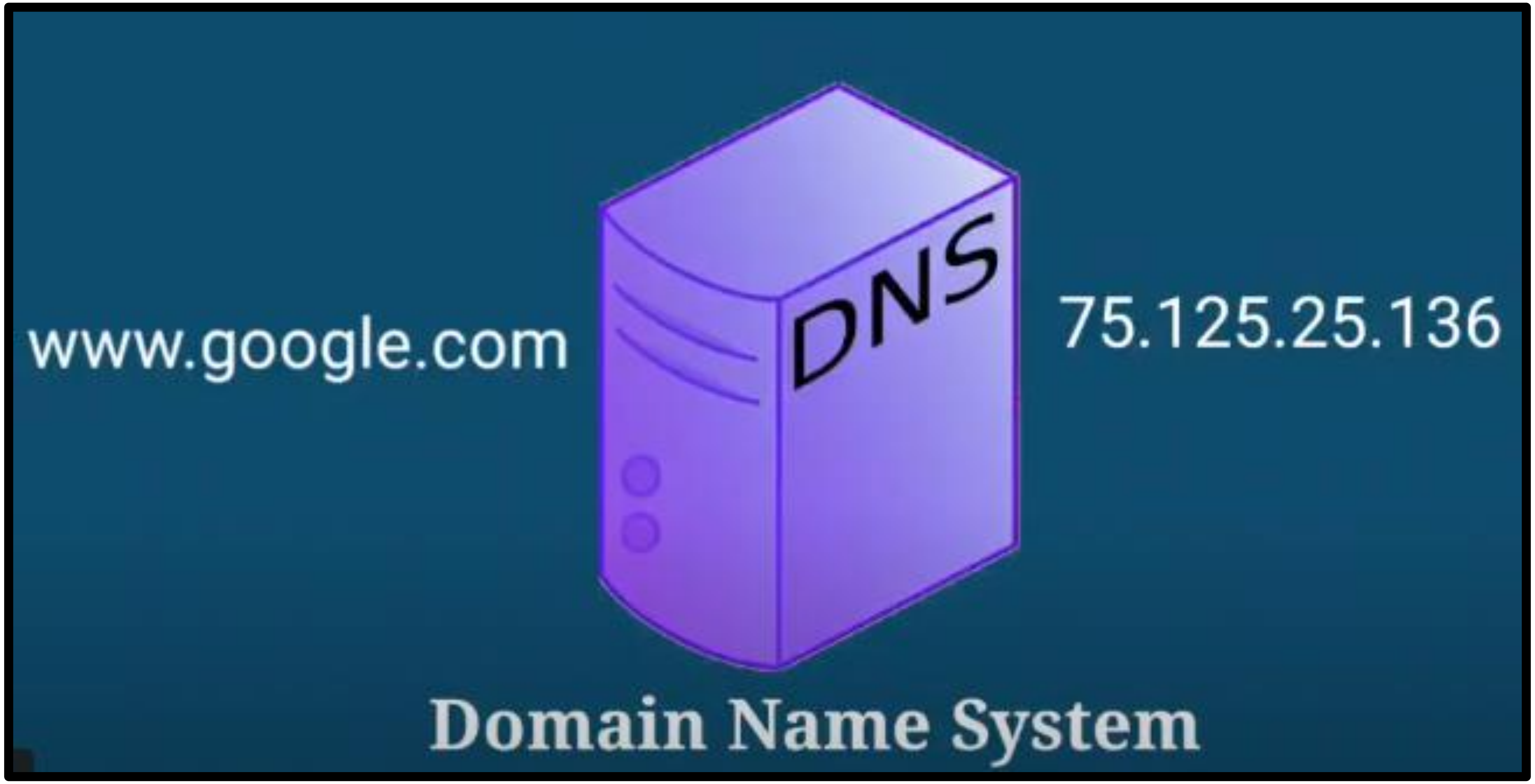
Domain Name: (cntd.)

- 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.
- Example: challenger.atc.fhda.edu.
- 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.
- Eg atc.fhda.edu

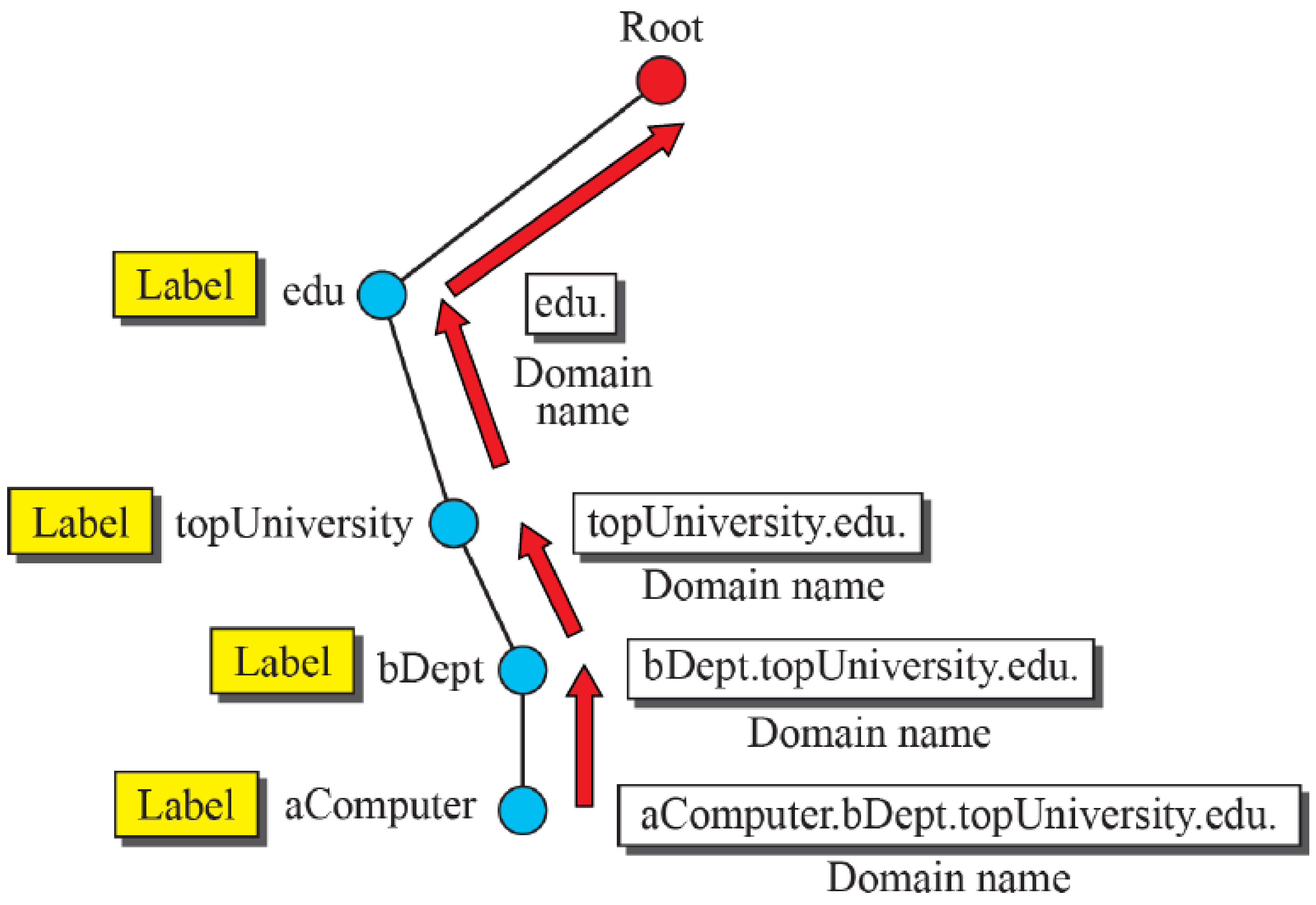
Domain

- 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.
- Note that a domain may itself be divided into domains.

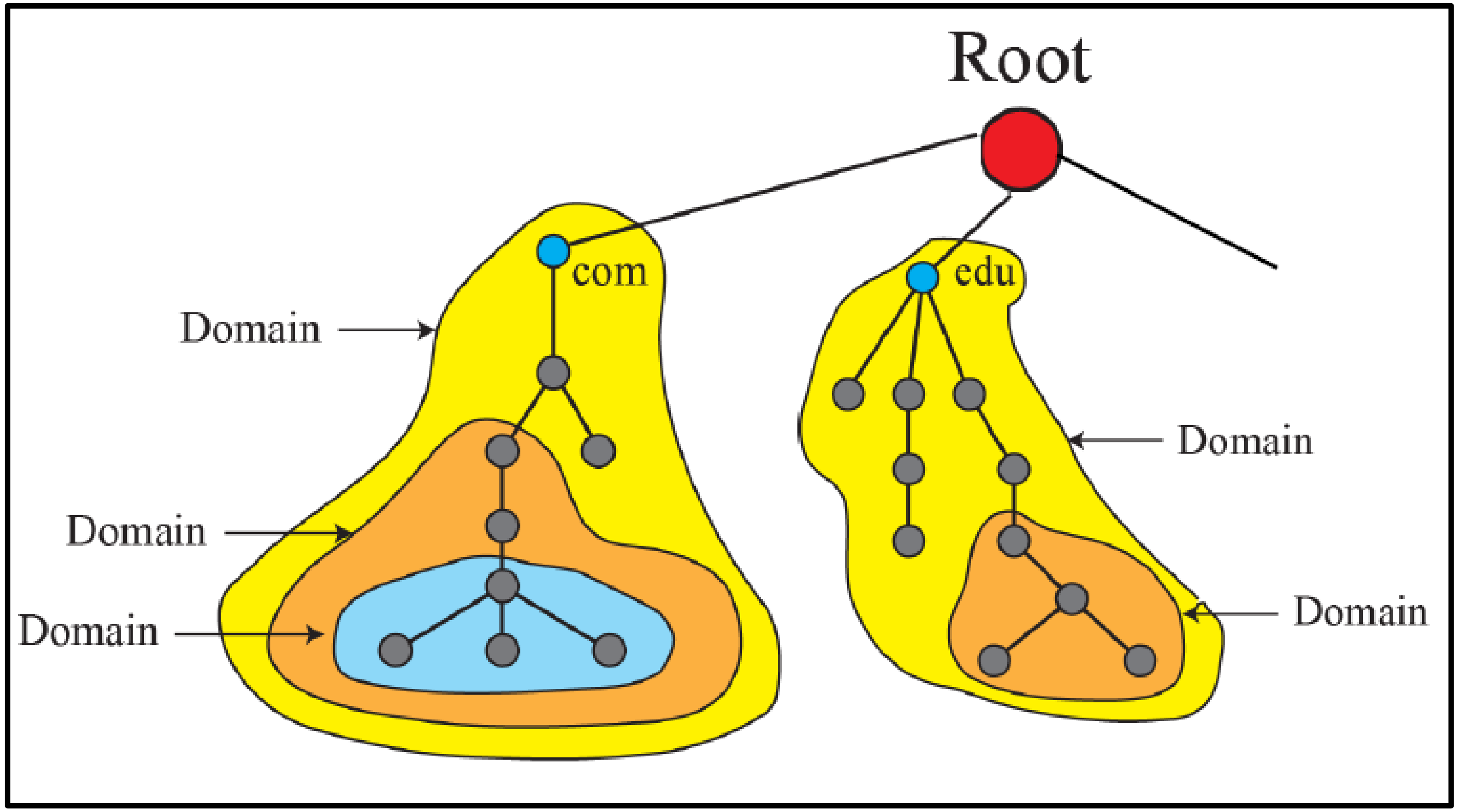
DNS (Domain Name System)



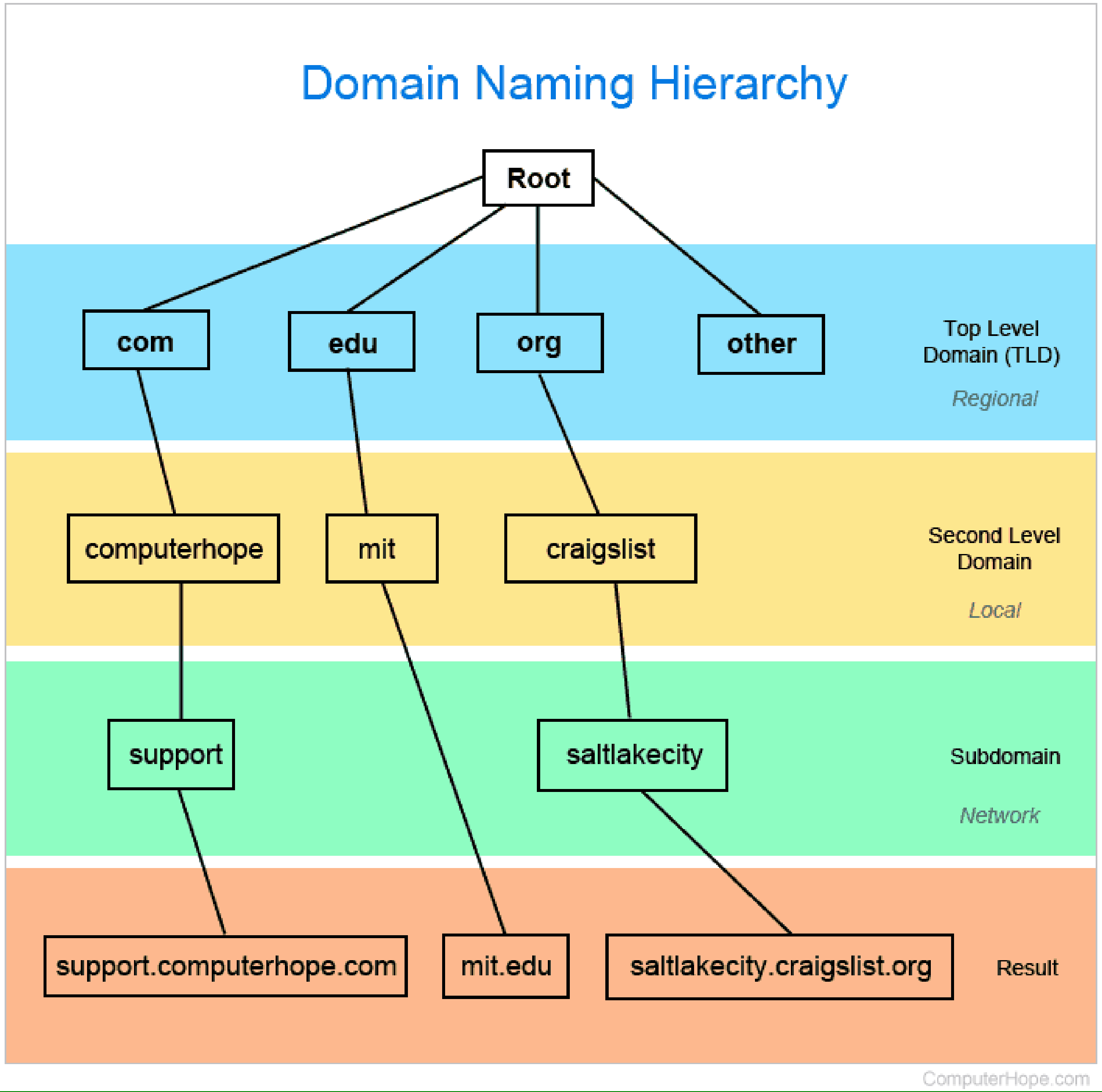
Domain Names and Labels



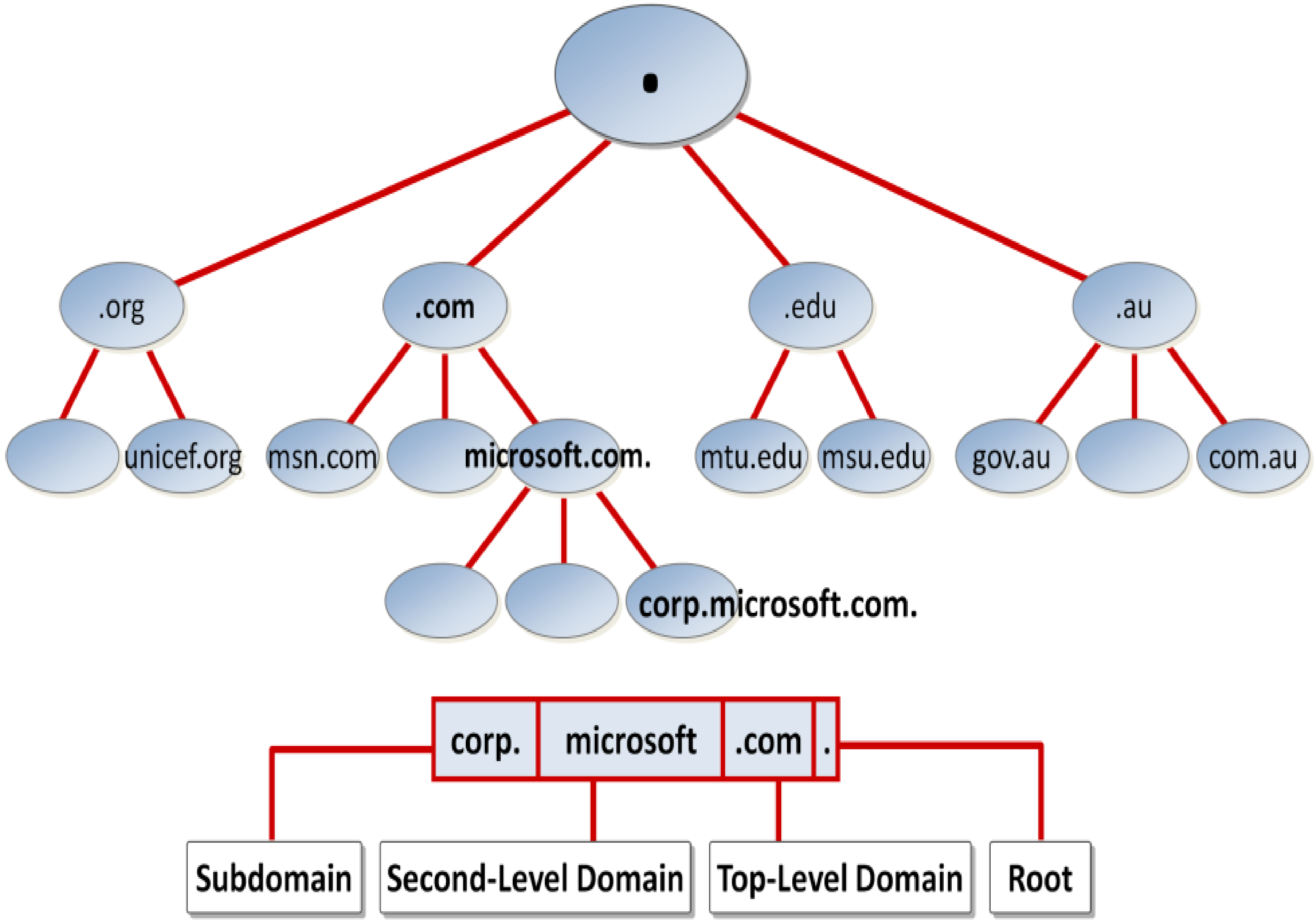
Domains



DNS (Domain Name System)

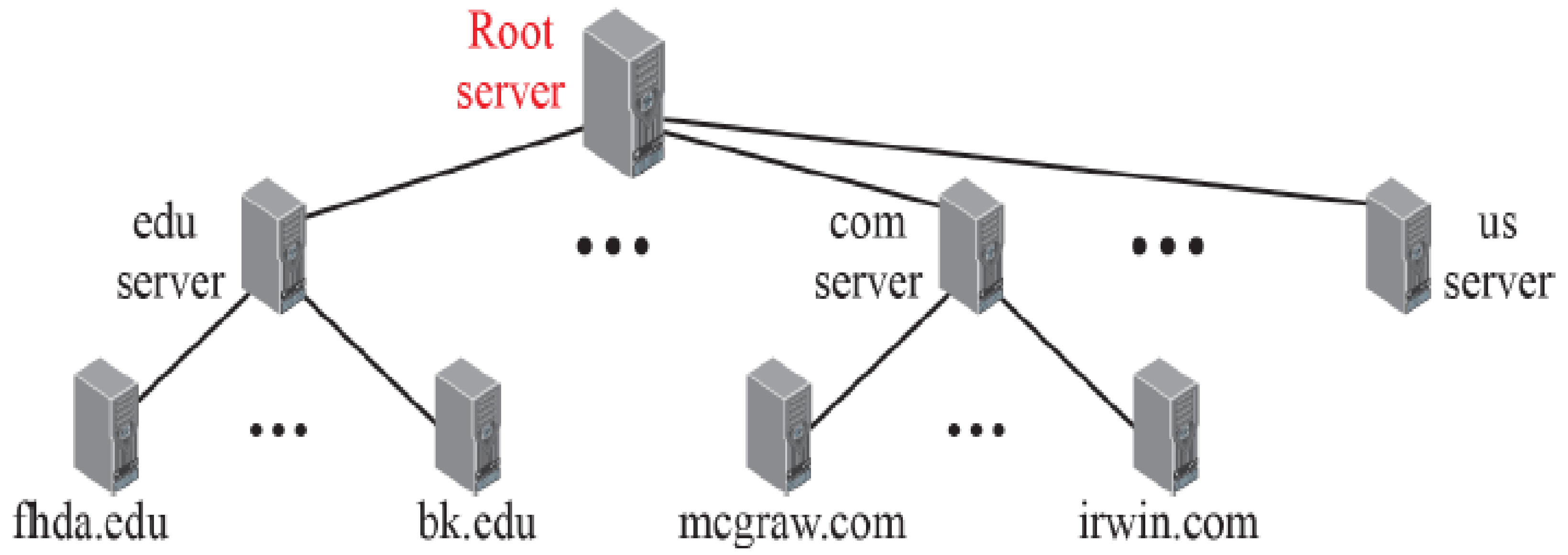


DNS (Domain Name System)

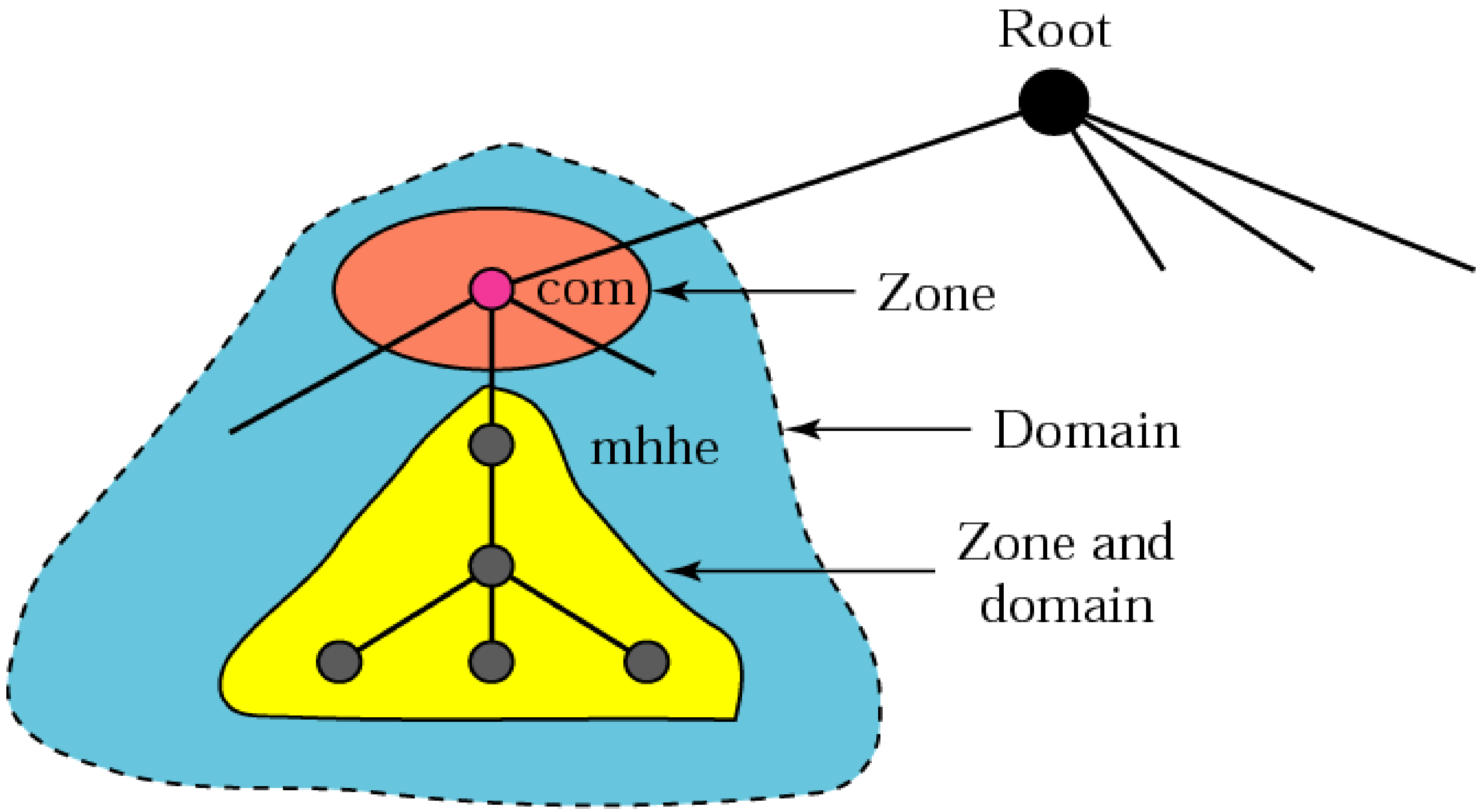


Distribution of Namespace

- The **information** contained in the domain name space is distributed among **many computers** called **DNS servers.**
- **Fig: Hierarchy of nameservers.**



Zones and domains



Primary server and Secondary server

- A primary server loads all information from the disk file;
- secondary server loads all information from the primary server.
- When the secondary downloads information from the primary, it is called zone transfer.

DNS in the Internet

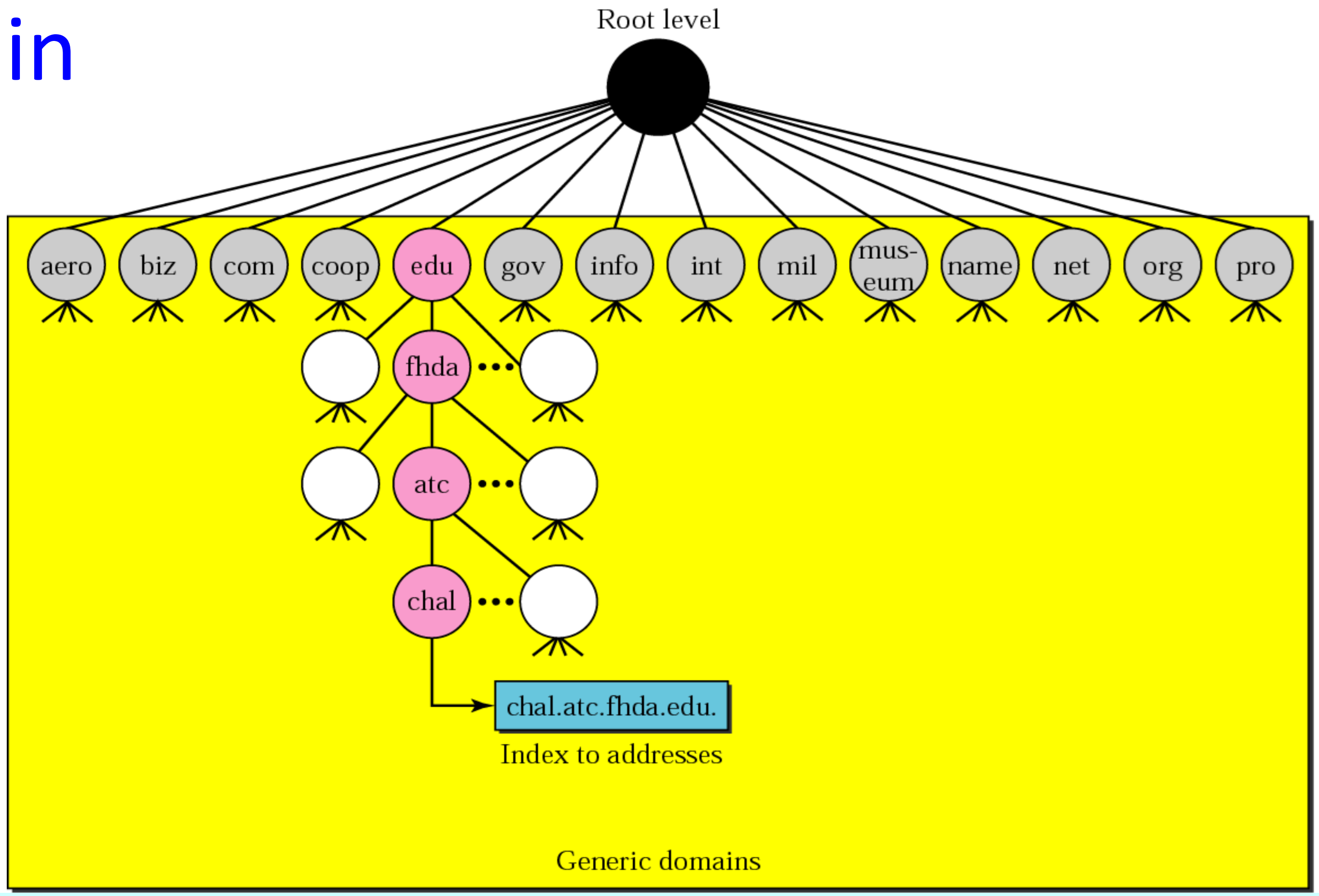
- **The domain name space (tree) is divided into three different sections:**

i. Generic domains

ii. Country domains &

iii. Inverse domain.

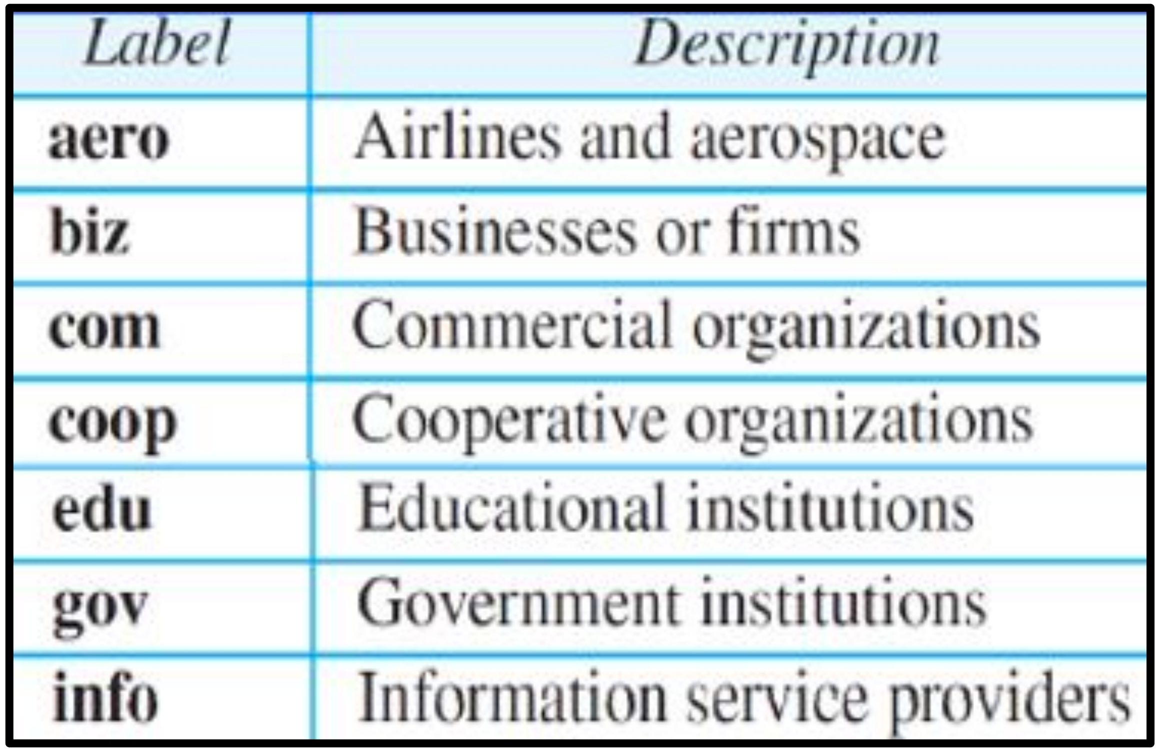
Generic Domain



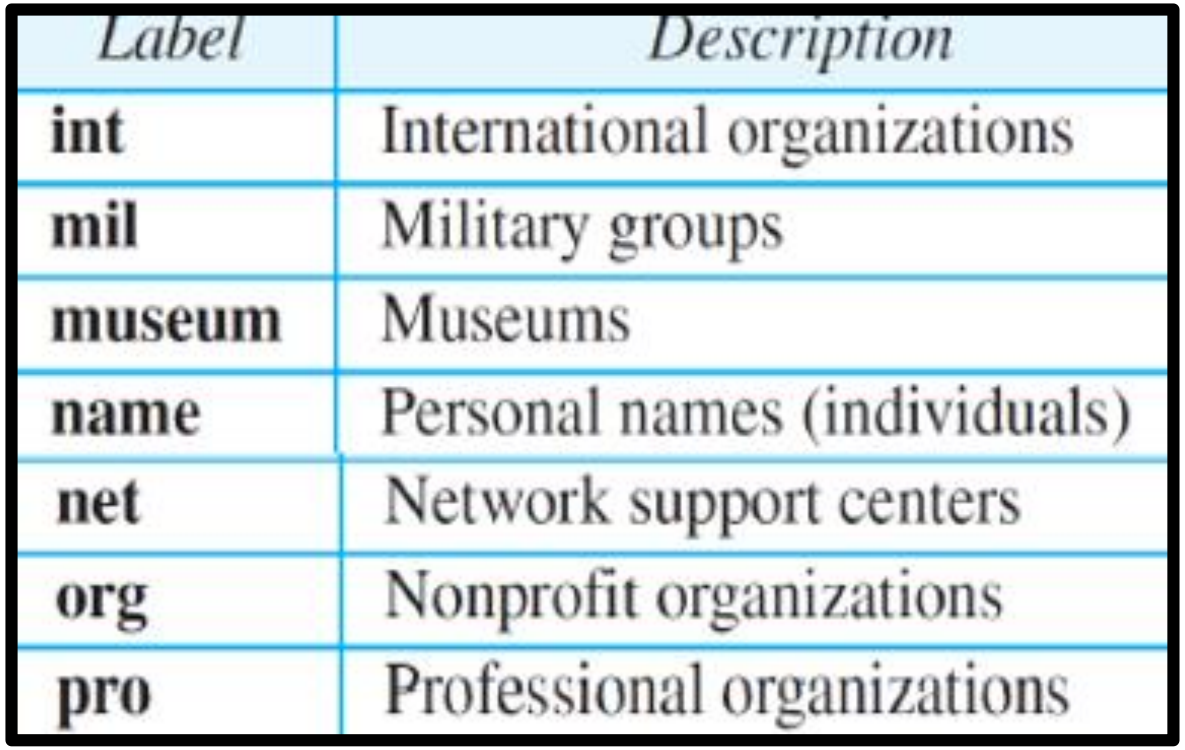
Generic Domains

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.

Generic Domain Labels



Label	*Description*
aero	Airlines and aerospace
biz	Businesses or firms
com	Commercial organizations
coop	Cooperative organizations
edu	Educational institutions
gov	Government institutions
info	Information service providers

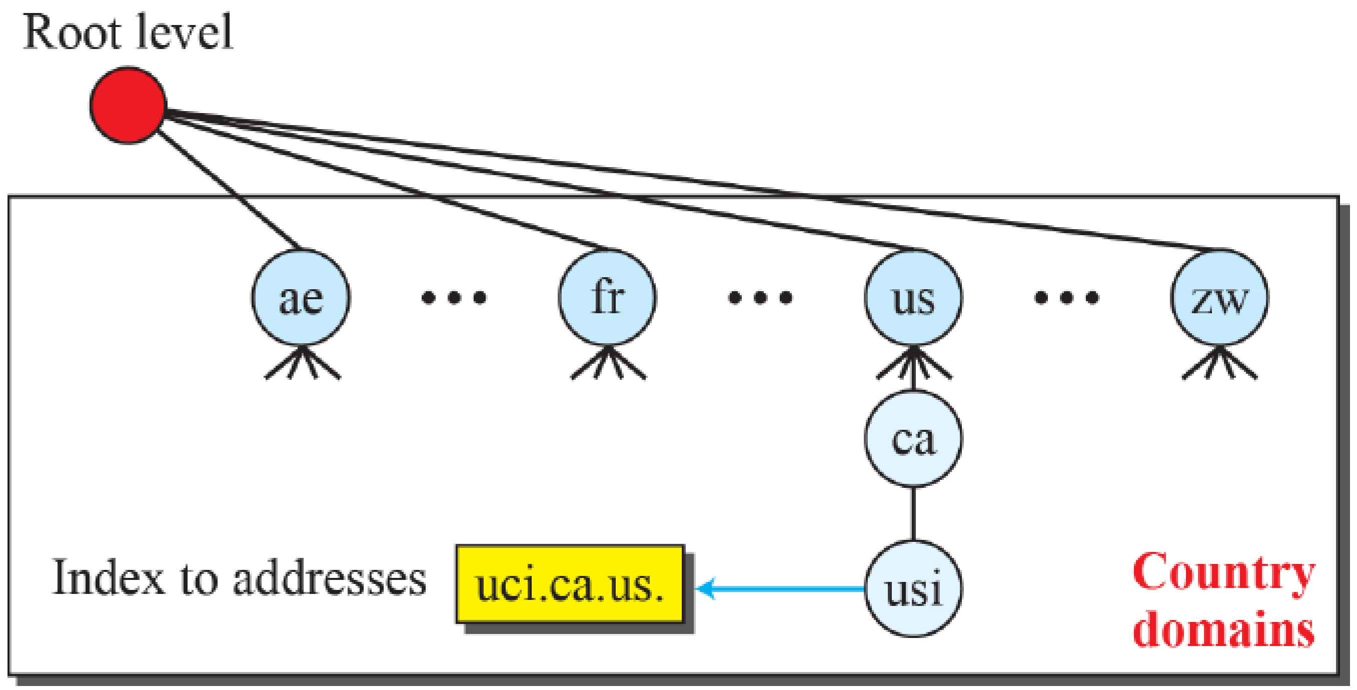


Label	*Description*
int	International organizations
mil	Military groups
museum	Museums
name	Personal names (individuals)
net	Network support centers
org	Nonprofit organizations
pro	Professional organizations

Country Domain

Country Domains:

- The country domains section uses two-character country abbreviations.
- (e.g., us → United States).
- Second labels can be organizational, or they can be more specific, national designations.



TLD	Country
.uk	United Kingdom
.fr	France
.ch	Switzerland
.in	India

Resolution

- Mapping name to an address or address to a name is called name-address resolution.
- Resolver
- A host that needs to map an address to a name or name to an address calls a DNS client called resolver.
- DNS Client is a client machine configured to send name resolution queries to a DNS server.
- The resolver accesses the closest DNS server with a mapping request.

Resolution

- Mapping Names to Addresses
- Mapping Addresses to Names
- Recursive Resolution
- Iterative Resolution
- Caching

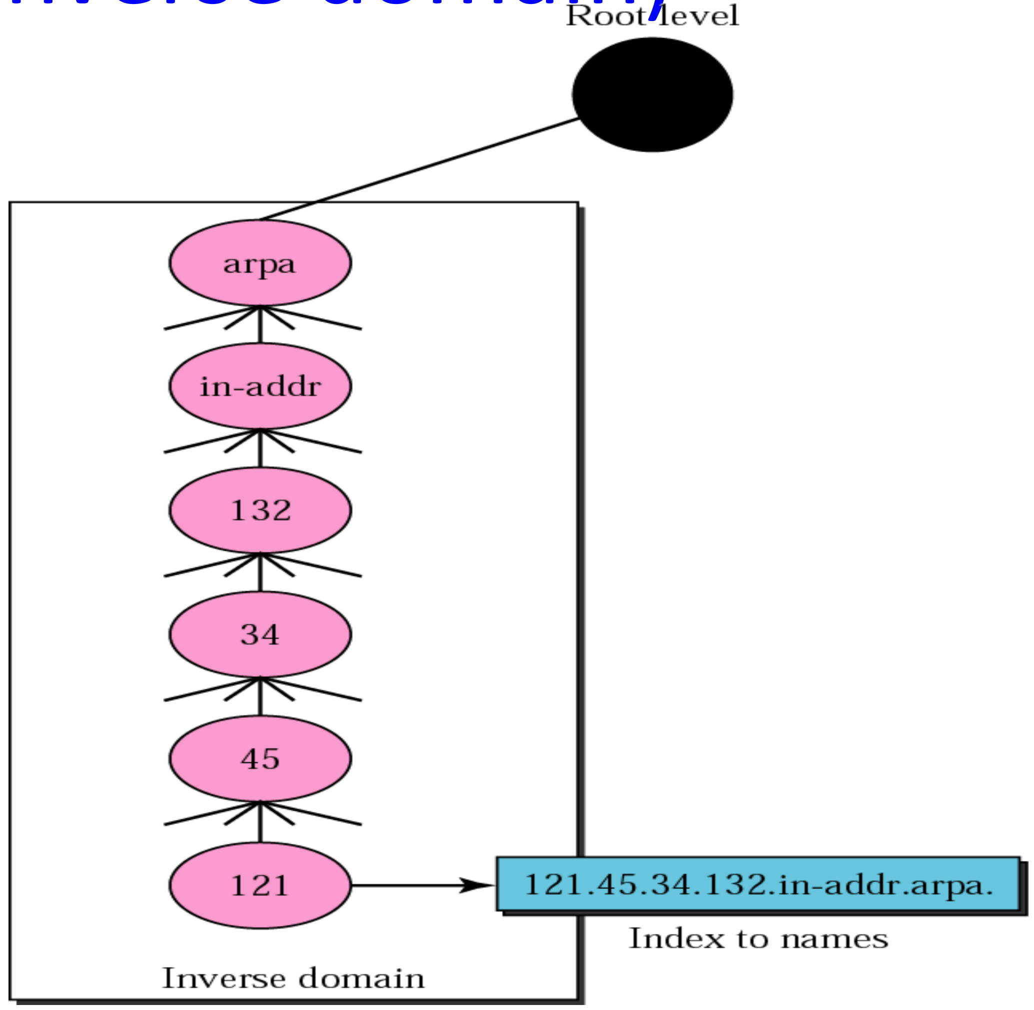
Mapping Names to Addresses

- **The resolver gives a domain name to the server and ask for the corresponding address.**
- **The server checks the generic domains or the country domains to find the mapping.**
- **If the domain name is from the generic domain the resolver receives a domain name such as for eg: "chal.atc.fhda.edu".**
- **The query is sent by the resolver to the local DNS server for resolution.**
- **If a local server cannot resolve the query ,it either refers the resolver to other servers or ask other servers directly.**
- **If the domain name is from the country domain section, the resolver receives domain name as for eg: "ch.fhda.cu.us".**

Mapping Addresses to Names

- **A client** can **send IP address to a server** to be mapped **to a domain name**.
- Here DNS uses **inverse domain.**
- IP address is reversed and two labels in-addr and arpa are appended to create a domain acceptable by the inverse domain section.
- If the resolver receives the ip address 132.34.45.121,the resolver first inverts the address and adds the labels before sending.
- The **domain name sent** is “**121.45.34.132 .in-addr.arpa”** which is received by the local DNS and resolved.

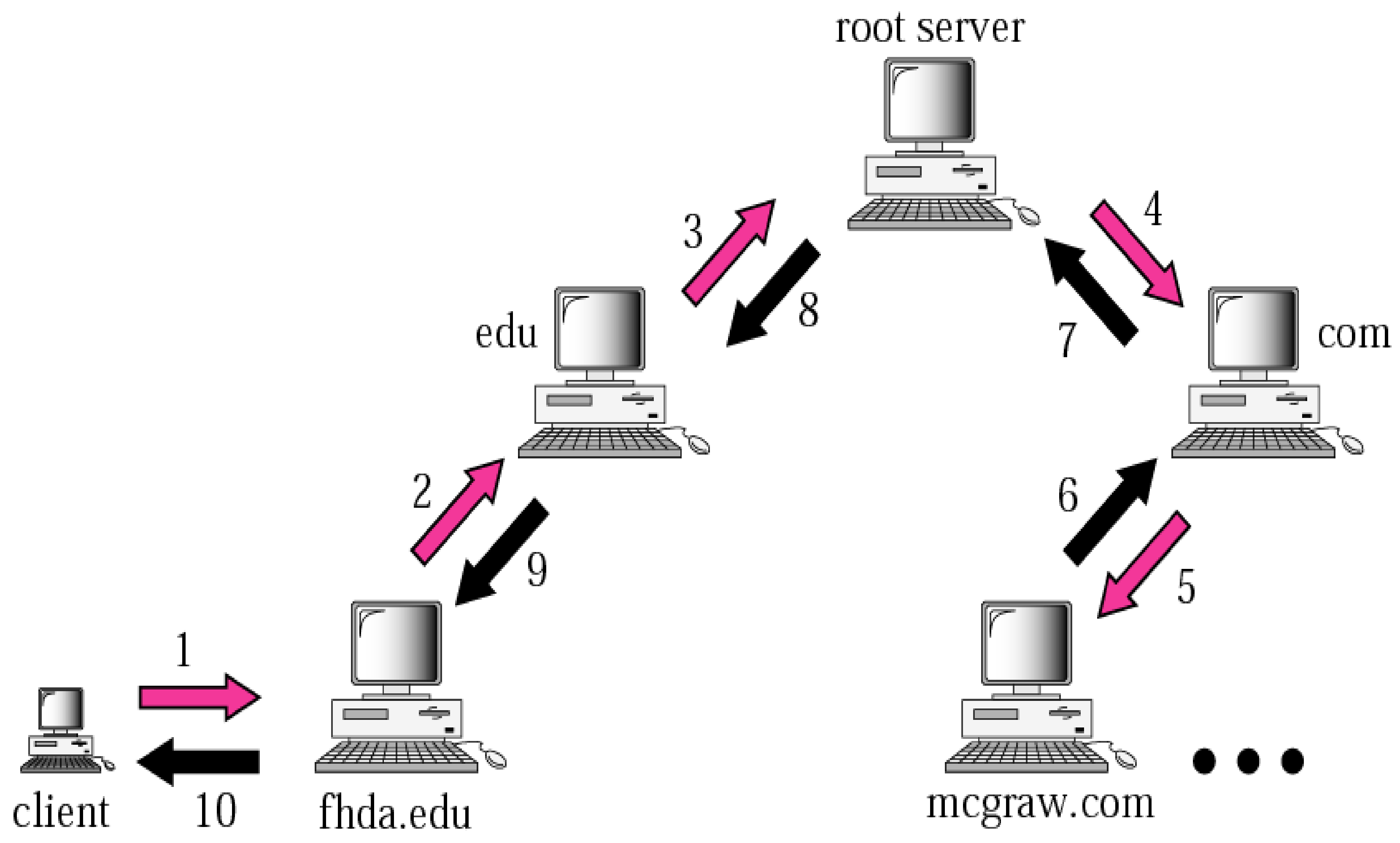
Mapping Addresses to Names (Inverse domain)



Recursive Resolution

- The client(resolver) can ask for recursive answer from a name server.
- If the server is the authority for the domain name ,it check its database and respond.
- If the server is not the authority it sends the request to the parent server and wait for the response.
- If parent is the authority it responds, otherwise it sends the query to another server(parent usually).
- When the query is finally resolved, the responds travel back until it finally reaches the requesting client.
- This is called Recursive Resolution.

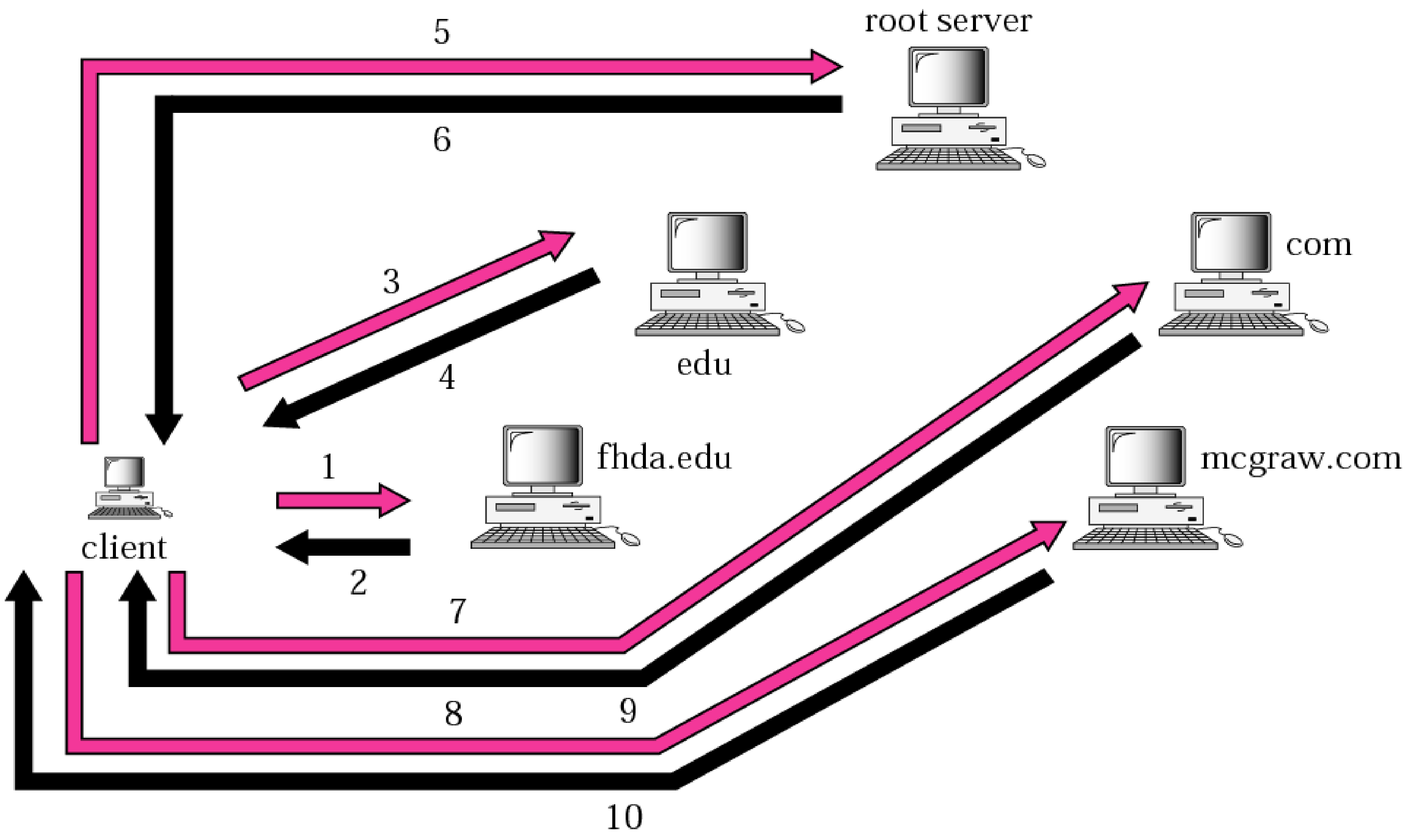
Recursive Resolution



Iterative Resolution

- If the server is an authority for the name, it sends the answer.
- If it is not,it returns to the client, the IP address of the server that it thinks can resolve the query.
- The client is responsible for repeating the query to this second server.
- If the newly addressed server can resolve the problem it answers the query with IP address otherwise it returns the IP address of a new server to the client.
- Now the client must repeat the query to the third server.
- This process is called iterative resolution because the **client repeats the same query to multiple servers.**

Iterative Resolution



Caching

- When a server asks for a mapping from another server and receives the response, it stores the information in its cache memory before sending it to the client.
- If the same or another client ask for the same mapping, it can check its cache memory and solve the problem.
- Thus reduce the search time and increase efficiency.

How can we Register a Domain Name

- Getting a domain name involves registering the name you want with an organization called ICANN through a domain name registrar.
- For example, if you choose a name like "example.com", you will have to go to a registrar, pay a registration fee that costs around Rs.700 for that name.
- That will give you the right to the name for a year, and you will have to renew it annually for (usually) the same amount per annum.

✓ Domain.com
✓ GoDaddy
✓ Bluehost
✓ Namecheap

File Transfer Protocol(FTP)

- File Transfer Protocol(FTP) is an application layer protocol which moves files between local and remote file systems.

- To transfer a file, 2 TCP connections are used by FTP in parallel

- Control connection : For sending control information like user identification, password, commands to change the remote directory, commands to retrieve and store files, etc., FTP makes use of control connection.
- Data connection: For sending the actual file, FTP makes use of data connection.

File Transfer Protocol(FTP)

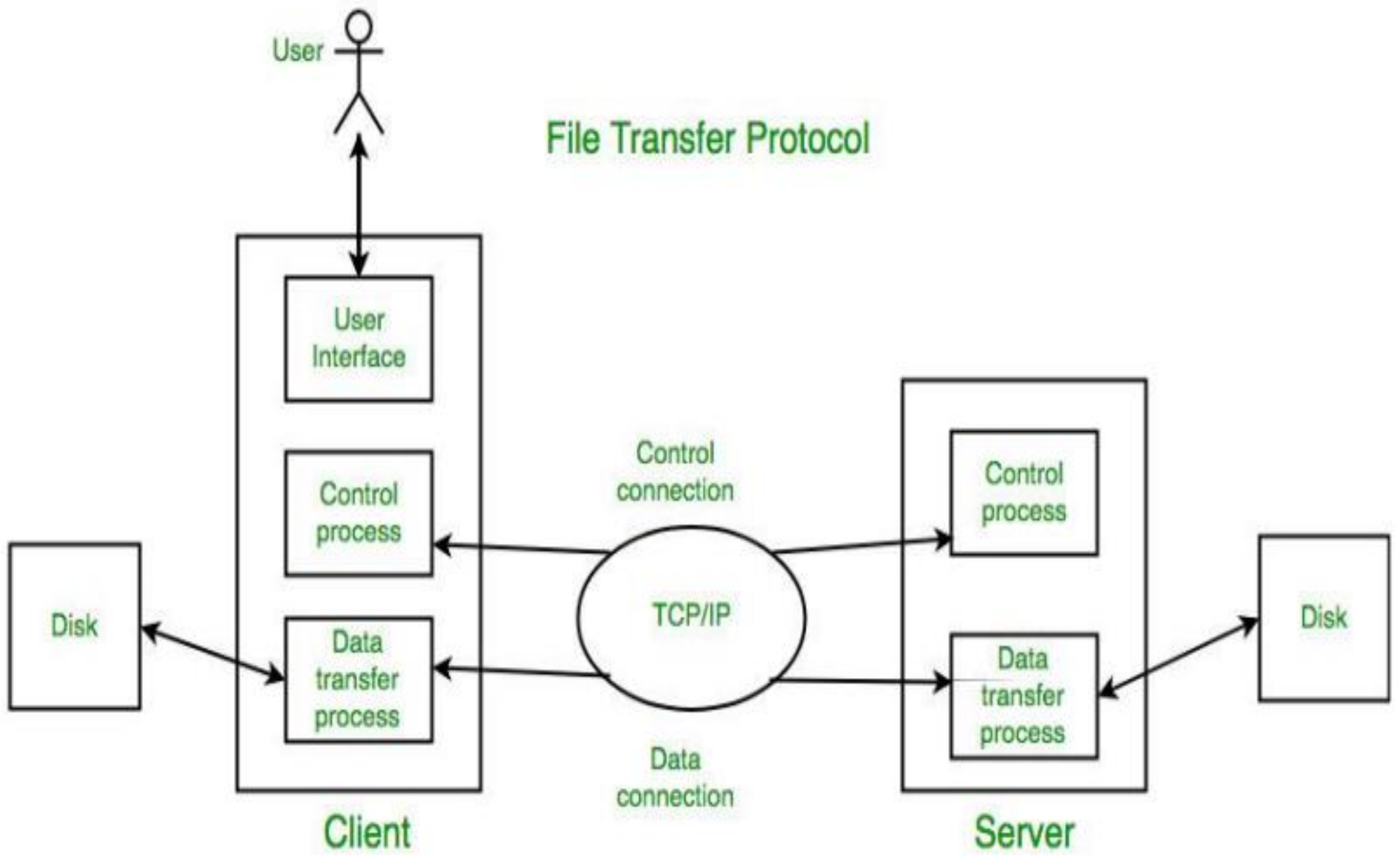
FTP Session

- When a FTP session is started between a client and a server, the client initiates a control TCP connection with the server side. The client sends control information over this.

- When the server receives this, it initiates a data connection to the client side. Only one file can be sent over one data connection. But the control connection remains active throughout the user session.

- The client requests the server for a file. When the server receives a request for a file, it opens a TCP connection for the client and transfers the file. After the transfer is complete, the server closes the connection. For a second file, client requests again and the server reopens a new TCP connection.

File Transfer Protocol(FTP)



Assignment-1:

i. Compare OSI & TCP/IP Models.

ii. Write Notes on P2P(Peer To Peer) File Sharing.

iii. Certifications / badges acquired based on computer Networks.

P2P(Peer To Peer) File Sharing

- When one peer makes a request, it is possible that multiple peers have the copy of that requested object.
- Now the problem is how to get the IP addresses of all those peers.
- This is decided by the underlying architecture supported by the P2P systems.
- By means of one of these methods, the client peer can get to know about all the peers which have the requested object/file and the file transfer takes place directly between these two peers.

Three such Architectures exist:

i. Centralized Directory
ii. Query Flooding
iii. Exploiting Heterogeneity

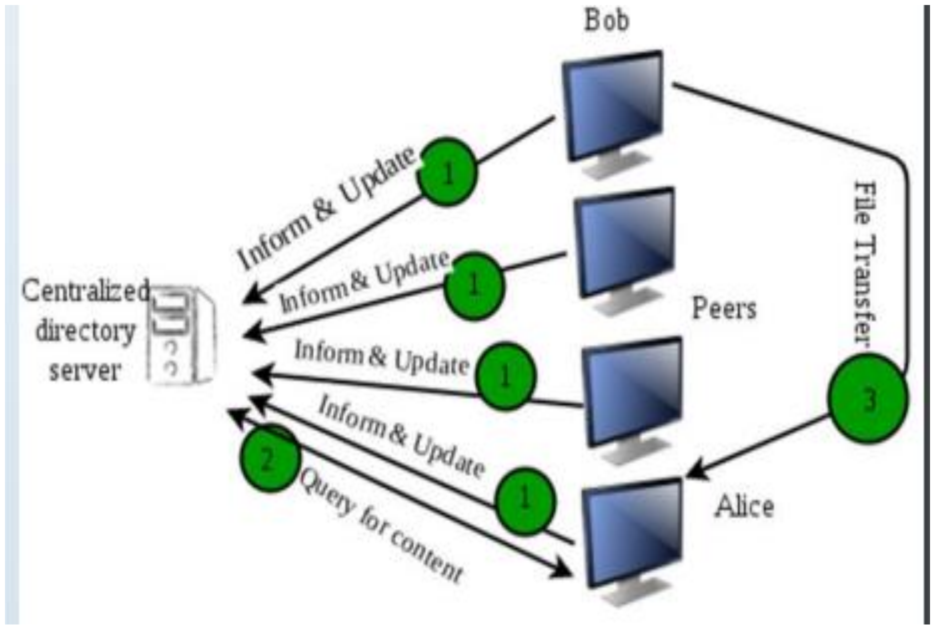
1. Centralized Directory

- It is somewhat similar to client server architecture in the sense that it maintains a huge central server to provide directory service.

- All the peers inform this central server of their IP address and the files they are making available for sharing.

- The server queries the peers at regular intervals to make sure if the peers are still connected or not.

- So basically this server maintains a huge database regarding which file is present at which IP addresses



Working

- Now whenever a requesting peer comes in, it sends its query to the server.
- Since the server has all the information of its peers, so it returns the IP addresses of all the peers having the requested file to the peer.
- Now the file transfer takes place between these two peers.
- The first system which made use of this method was Napster, for the purpose of Mp3 distribution.

- The major problem with such an architecture is that there is a single point of failure. If the server crashes, the whole P2P network crashes.

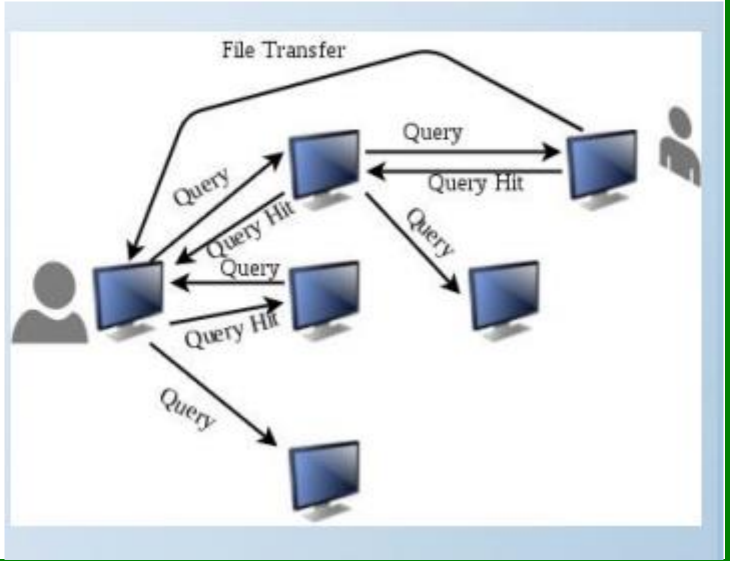
- Also, since all of the processing is to be done by a single server so a huge amount of database has to be maintained and regularly updated.

2. Query Flooding

- Unlike the centralized approach, this method makes use of distributed systems.

In this, the peers are supposed to be connected into an overlay network. It means if a connection/path exists from one peer to other, it is a part of this overlay network.

- In this overlay network, peers are called as nodes and the connection between peers is called an edge between the nodes, thus resulting in a graph-like structure.



Working

• Now when one peer requests for some file, this request is sent to all its neighboring nodes i.e. to all nodes which are connected to this node. If those nodes don’t have the required file, they pass on the query to their neighbors and so on. This is called as query flooding.

• When the peer with requested file is found (referred to as query hit), the query flooding stops and it sends back the file name and file size to the client, thus following the reverse path.

• If there are multiple query hits, the client selects from one of these peers. • Gnutella was the first decentralized peer to peer network.

• This method also has some disadvantages like, the query has to be sent to all the neighboring peers unless a match is found. This increases traffic in the network.

3. Exploiting heterogeneity

• This P2P architecture makes use of both the above discussed systems.

• It resembles a distributed system like Gnutella because there is no central server for query processing.

• But unlike Gnutella, it does not treat all its peers equally. The peers with higher bandwidth and network connectivity are at a higher priority and are called as group leaders/super nodes. The rest of the peers are assigned to these super nodes.

• These super nodes are interconnected and the peers under these super nodes inform their respective leaders about their connectivity, IP address and the files available for sharing.

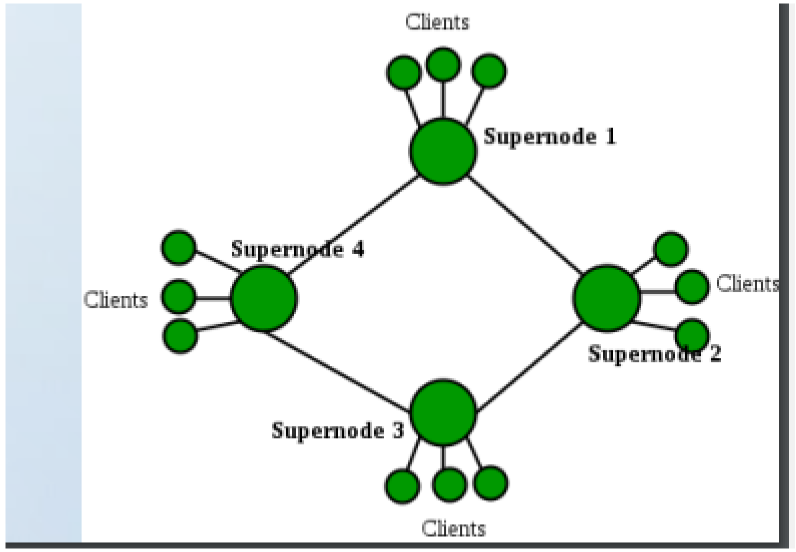
• KaZaA technology is such an example which makes use of Napster and Gnutella both. • Thus, the individual group leaders along with their child peers form a Napster-like structure. These group leaders then interconnect among themselves to resemble a Gnutella-like structure.

Working

- This structure can process the queries in two ways.

- The first one is that the super nodes could contact other super nodes and merge their databases with its own database. Thus, this super node now has information of a large number of peers.

- Another approach is that when a query comes in, it is forwarded to the neighboring super nodes until a match is found, just like in Gnutella. Thus query flooding exists but with limited scope as each super node has many child peers. Hence, such a system exploits the heterogeneity of the peers by designating some of them as group leaders/super nodes and others as their child peers.



The file-transfer load is distributed between the computers exchanging files, but file searches and transfers from your computer to others can cause bottlenecks.

- Some people download files and immediately disconnect without allowing others to obtain files from their system, which is called leeching. This limits the number of computers the software can search for the requested file.
- These are some Peer to Peer (P2P) File Sharing Programs and Applications
- uTorrent
- BitTorrent
- qBittorrent
- Transmission
- Soulseek