UNIT V

APPLICATION LAYER

- Application Layer:
- Domain Name System (DNS),
- EMAIL,
- File Transfer Protocol (FTP),
- WWW,
- HTTP,
- SNMP.

- The application layer is responsible for providing services to the user.
- The application layer enables the user, whether human or software, to access the network.
- It provides user interfaces and support for services such as

✓ electronic mail,

✓ file access and transfer,

✓ access to system resources,

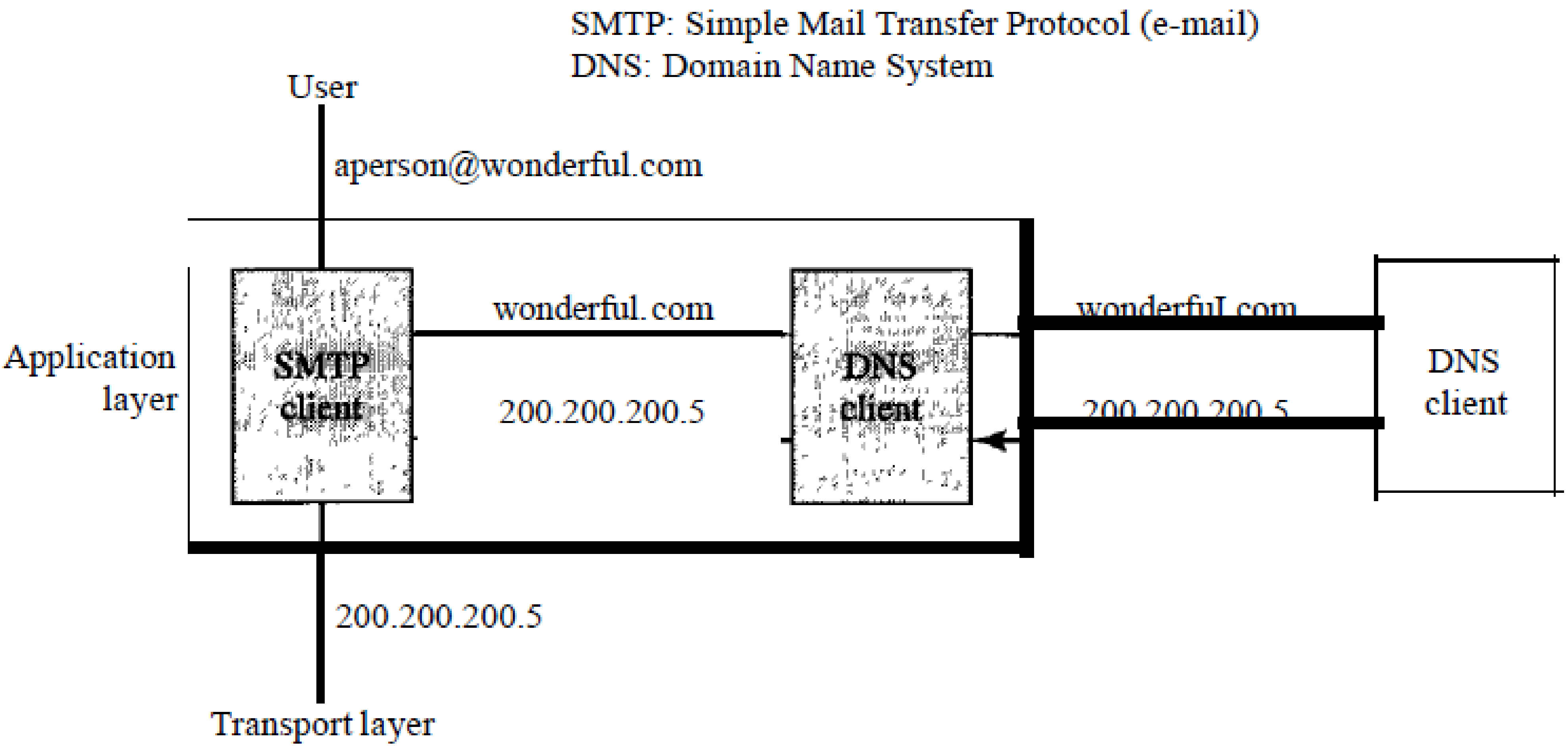
✓surfing the world wide web, and network management.

Domain Name System (DNS)

- The client/server programs can be divided into two categories:
- those that can be directly used by the user, such as e-mail, and
- those that support other application programs.
- The Domain Name System (DNS) is a supporting program that

  is used by other programs such as e-mail.

Example of using the DNS service



SMTP: Simple Mail Transfer Protocol (e-mail)
DNS: Domain Name System

- A user of an e-mail program may know the e-mail address of the recipient; however, the IP protocol needs the IP address.
- The DNS client program sends a request to a DNS server to map the e-mail address to the corresponding IP address.

- When the Internet was small, mapping was done by using a host file. The host file had only two columns: name and address.
- Every host could store the host file on its disk and update it periodically from a master host file.
- When a program or a user wanted to map a name to an address, the host consulted the host file and found the mapping.

- Today, however, it is impossible to have one single host file to relate every address with a name and vice versa.
- The host file would be too large to store in every host.
- In addition, it would be impossible to update all the host files every time there was a change.

- One solution would be to store the entire host file in a single computer and allow access to this centralized information to every computer that needs mapping.
- But we know that this would create a huge amount of traffic on the Internet.
- Another solution, the one used today, is to divide this huge amount of information into smaller parts and store each part on a different computer.
- In this method, the host that needs mapping can contact the closest computer holding the needed information.

NAME SPACE

- the names assigned to machines must be carefully selected from a name space with complete control over the binding between the names and IP addresses.
- In other words, 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:
- flat or hierarchical.

Flat Name Space

- In a flat name space, a 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.

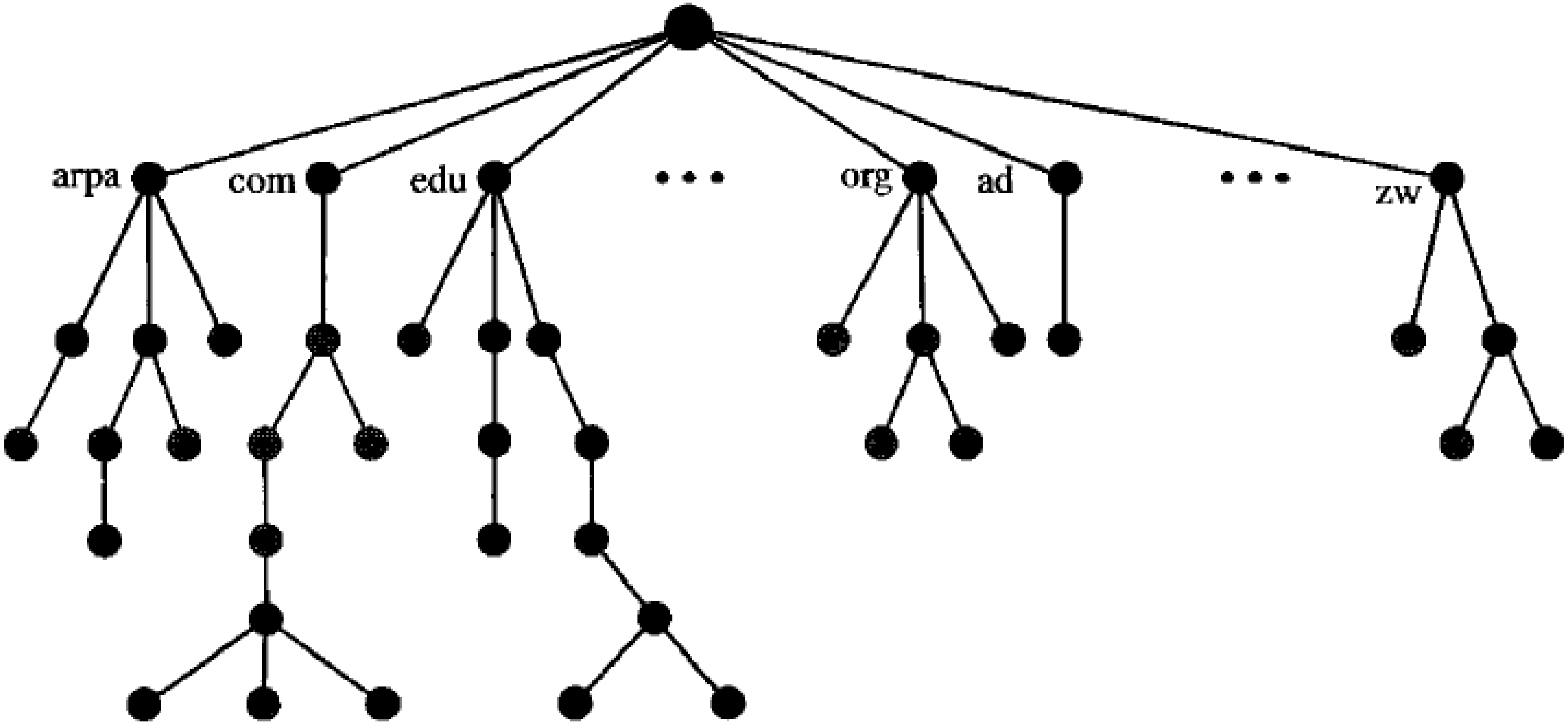
- In a hierarchical name space, each name is made of several parts. The first part can define the nature of the organization, the second part can define the name of an organization, the third part can define departments in the organization, and so on.
- In this case, the authority to assign and control the name spaces can be decentralized.
- A central authority can assign the part of the name that defines the nature of the organization and the name of the organization.

- For example, assume two colleges and a company call one of their computers *challenger.*
- The first college is given a name by the central authority such as *jhda.edu,* the second college is given the name *berkeley.edu,* and the company is given the name *smart. com.*
- When these organizations add the name *challenger* to the name they have already been given, the end result is three distinguishable
- names: *challenger.jhda.edu,*

  challenger.berkeley.edu, and *challenger.smart.com.*

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

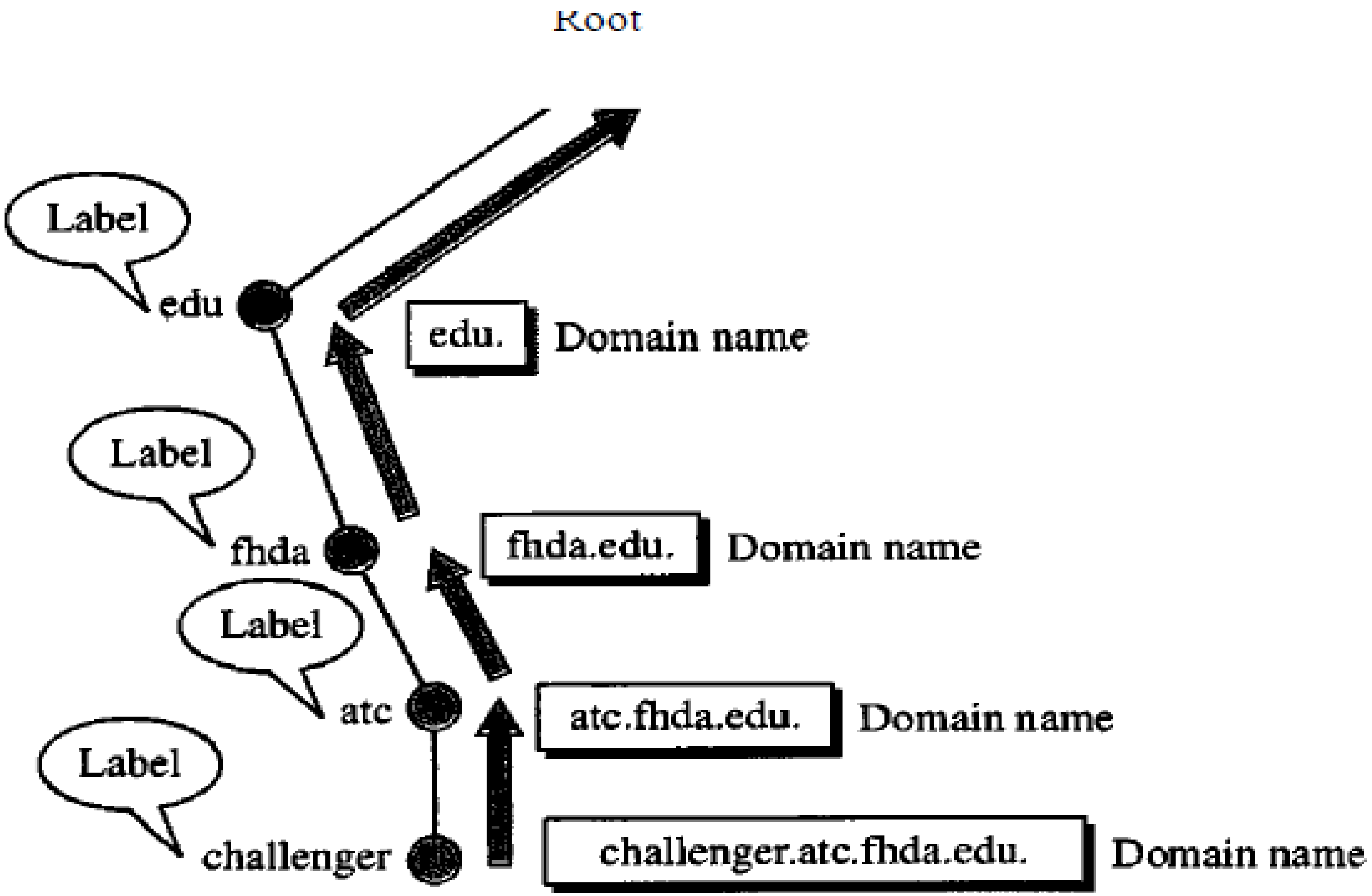

arpa
com
edu
• • •
org
ad
• • •
zw

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.


Root
Label
edu
edu.
Domain name
Label
fhda
fhda.edu.
Domain name
Label
atc
atc.fhda.edu.
Domain name
Label
challenger
challenger.atc.fhda.edu.
Domain name

Fully Qualified Domain Name

- If a label is terminated by a null string, it is called a fully qualified domain name (FQDN).
- An FQDN is a domain name that contains the full name of a host. It contains all labels, from the most specific to the most general, that uniquely define the name of the host.
- For example, the domain name challenger.ate.tbda.edu.
- The FQDN of a computer named *challenger* installed at the Advanced Technology Center (ATC) at De Anza College.
- A DNS server can only match an FQDN to an address.
- Note that the name must end with a null label, but because null means nothing, the label ends with a dot (.).

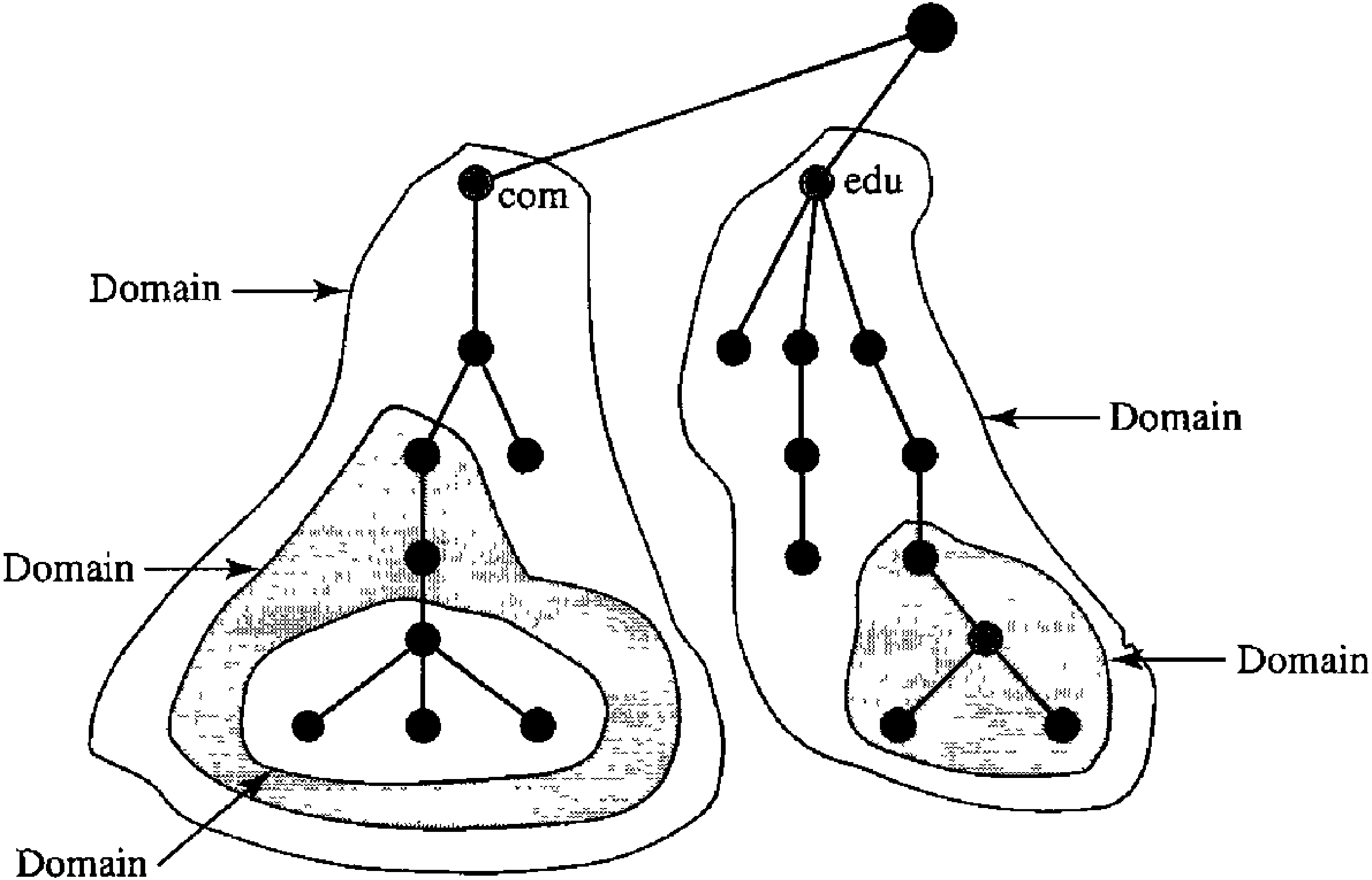
Partially Qualified Domain Name

- 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.
- It is used when the name to be resolved belongs to the same site as the client.

- Here the resolver can supply the missing part, called the suffix, to create an FQDN.
- For example, if a user at the *jhda.edu.* site wants to get the IP address of the challenger computer, he or she can define the partial name challenger
- The DNS client adds the suffix *atc.jhda.edu.* before passing the address to the DNS server.

Domain

- A **domain** is a subtree of the domain name space.
- The name of the domain is the domain name of the node at the top of the subtree. Figure shows some domains.
- Note that a domain may itself be divided into domains


com
edu
Domain
Domain
Domain
Domain
Domain

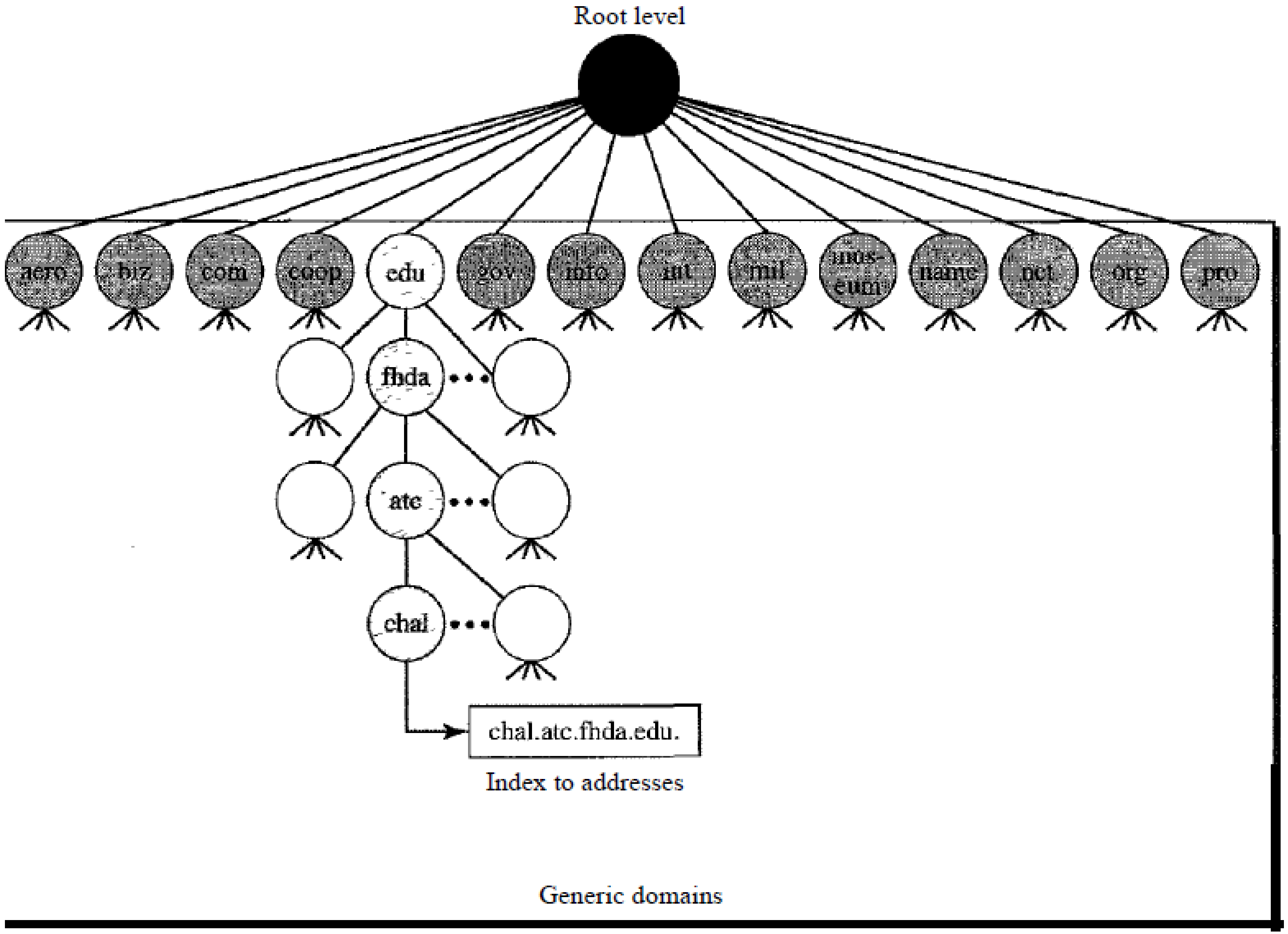
DNS IN THE INTERNET

- DNS is a protocol that can be used in different platforms. In the Internet, the domain name space (tree) is divided into three different sections:

-- generic domains, country domains, and the inverse domain

Generic Domains

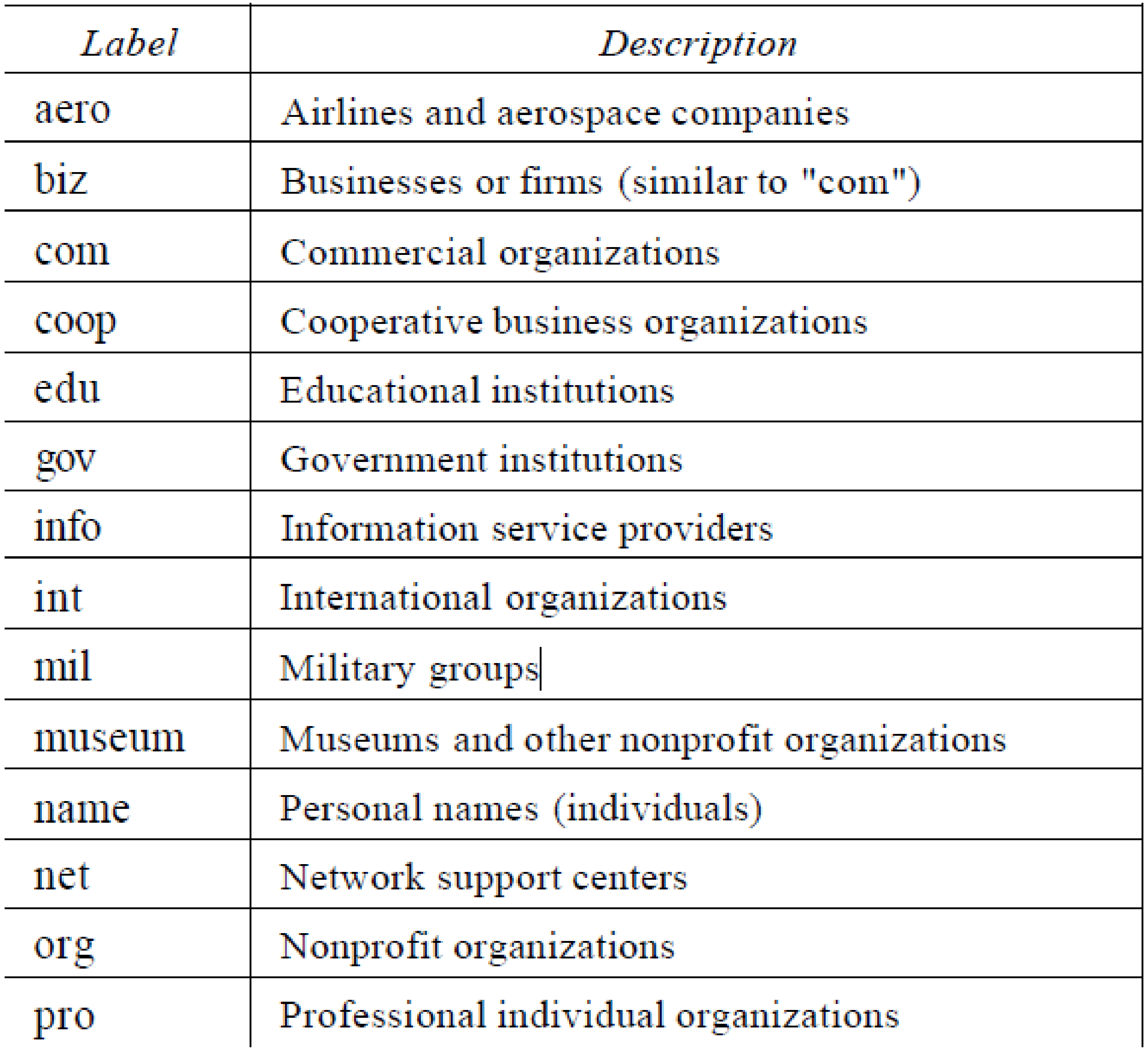
- The **generic domains** define registered hosts according to their generic behavior.
- Each node in the tree defines a domain, which is an index to the domain name space database


Root level
aero
biz
com
coop
edu
gov
info
int
mil
mus-
eum
name
net
org
pro
fhda
atc
chal
chal.atc.fhda.edu.
Index to addresses
Generic domains

- we see that the first level in the generic domains section allows 14 possible labels.

Country Domains

- The country domains section uses two-character country abbreviations (e.g., us for United States).
- Second labels can be organizational, or they can be more specific, national designations.
- The United States, for example, uses state abbreviations as a subdivision of us.



Label	*Description*
aero	Airlines and aerospace companies
biz	Businesses or firms (similar to "com")
com	Commercial organizations
coop	Cooperative business organizations
edu	Educational institutions
gov	Government institutions
info	Information service providers
int	International organizations
mil	Military groups
museum	Museums and other nonprofit organizations
name	Personal names (individuals)
net	Network support centers
org	Nonprofit organizations
pro	Professional individual organizations

Inverse Domain

- The inverse domain is used to map an address to a name.
- This may happen, for example, when a server has received a request from a client to do a task.
- Although the server has a file that contains a list of authorized clients, only the IP address of the client (extracted from the received IP packet) is listed.
- The server asks its resolver to send a query to the DNS server to map an address to a name to determine if the client is on the authorized list.

- 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.

Mapping Names to Addresses

- Most of the time, the resolver gives a domain name to the server and asks for the corresponding address.
- **In** this case, the server checks the generic domains or the country domains to find the mapping.
- If the domain name is from the generic domains section, the resolver receives a domain name such as *"chal.atc.jhda.edu."*. The query is sent by the resolver to the local DNS server for resolution. **If** the local server cannot resolve the query, it either refers the resolver to other servers or asks other servers directly.

Mapping Addresses to Names

- A client can send an **IP** address to a server to be mapped to a domain name. As mentioned before, this is called a PTR query.
- To answer queries of this kind, DNS uses the inverse domain.
- However, in the request, the **IP** address is reversed and the two labels *in-addr* and *arpa* are appended to create a domain acceptable by the inverse domain section.
- For example, if the resolver receives the IF address 132.34.45.121, the resolver first inverts the address and then adds the two labels before sending.
- The domain name sent is *"121.45.34.132.in-addr.arpa."* which is received by the local DNS and resolved.

The 8 steps in a DNS lookup:

1. A user types ‘example.com’ into a web browser and the query travels into the Internet and is received by a DNS recursive resolver.
2. The resolver then queries a DNS root name server (.).
3. The root server then responds to the resolver with the address of a Top Level Domain (TLD) DNS server (such as .com or .net), which stores the information for its domains. When searching for example.com, our request is pointed toward the .com TLD.
4. The resolver then makes a request to the .com TLD.

5. The TLD server then responds with the IP address of the domain’s name server, example.com.
6. Lastly, the recursive resolver sends a query to the domain’s name server.
7. The IP address for example.com is then returned to the resolver from the name server.
8. The DNS resolver then responds to the web browser with the IP address of the domain requested initially.

Once the 8 steps of the DNS lookup have returned the IP address for example.com, the browser is able to make the request for the web page:

9. The browser makes a HTTP request to the IP address.
10. The server at that IP returns the webpage to be rendered in the browser (step 10).

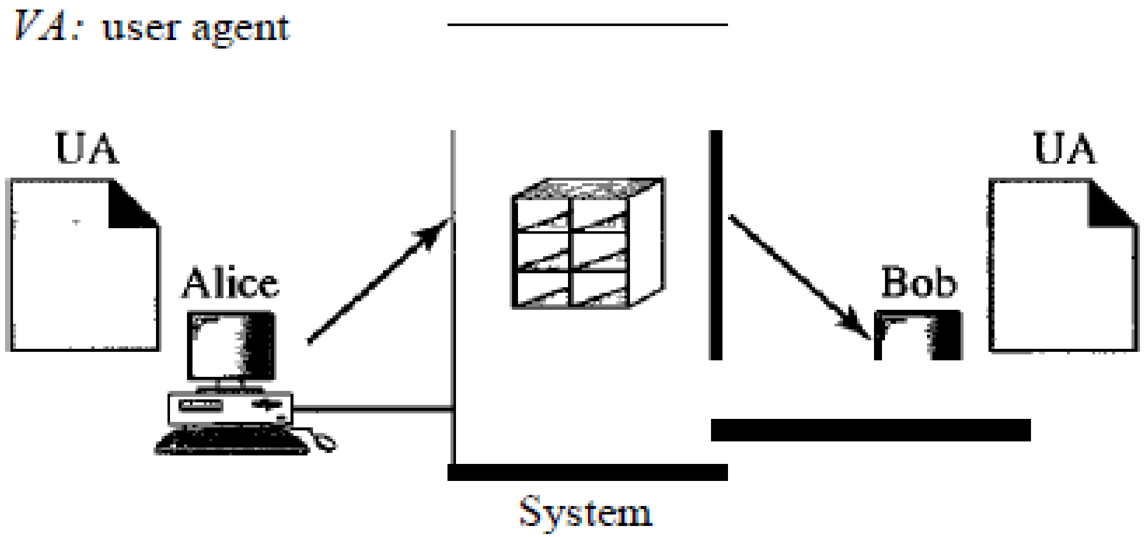
ELECTRONIC MAIL(E MAIL)

- One of the most popular Internet services is electronic mail (e-mail).

Architecture:

First Scenario

- the sender and the receiver of the e-mail are users (or application programs) on the same system; they are directly connected to a shared system.
- The administrator has created one mailbox for each user where the received messages are stored.
- A *mailbox* is part of a local hard drive, a special file with permission restrictions.

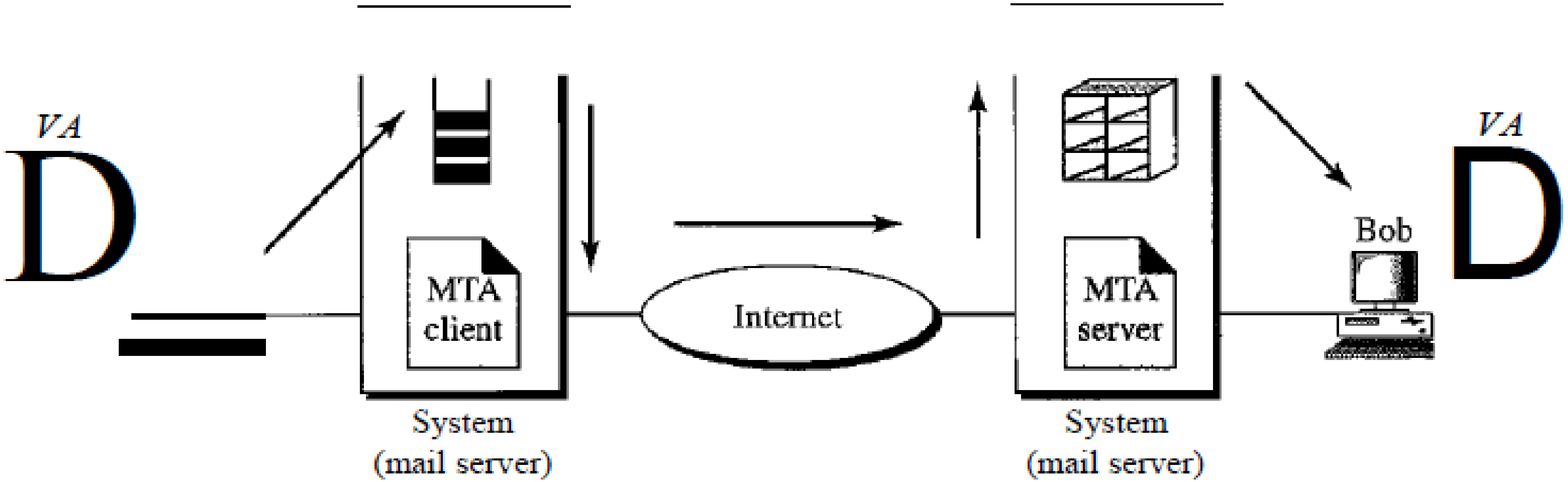


When the sender and the receiver of an e-mail are on the same system, we need only two user agents.

Second Scenario

- the sender and the receiver of the e-mail are users (or application programs) on two different systems. The message needs to be sent over the Internet.
- Here we need user agents (VAs) and message transfer agents

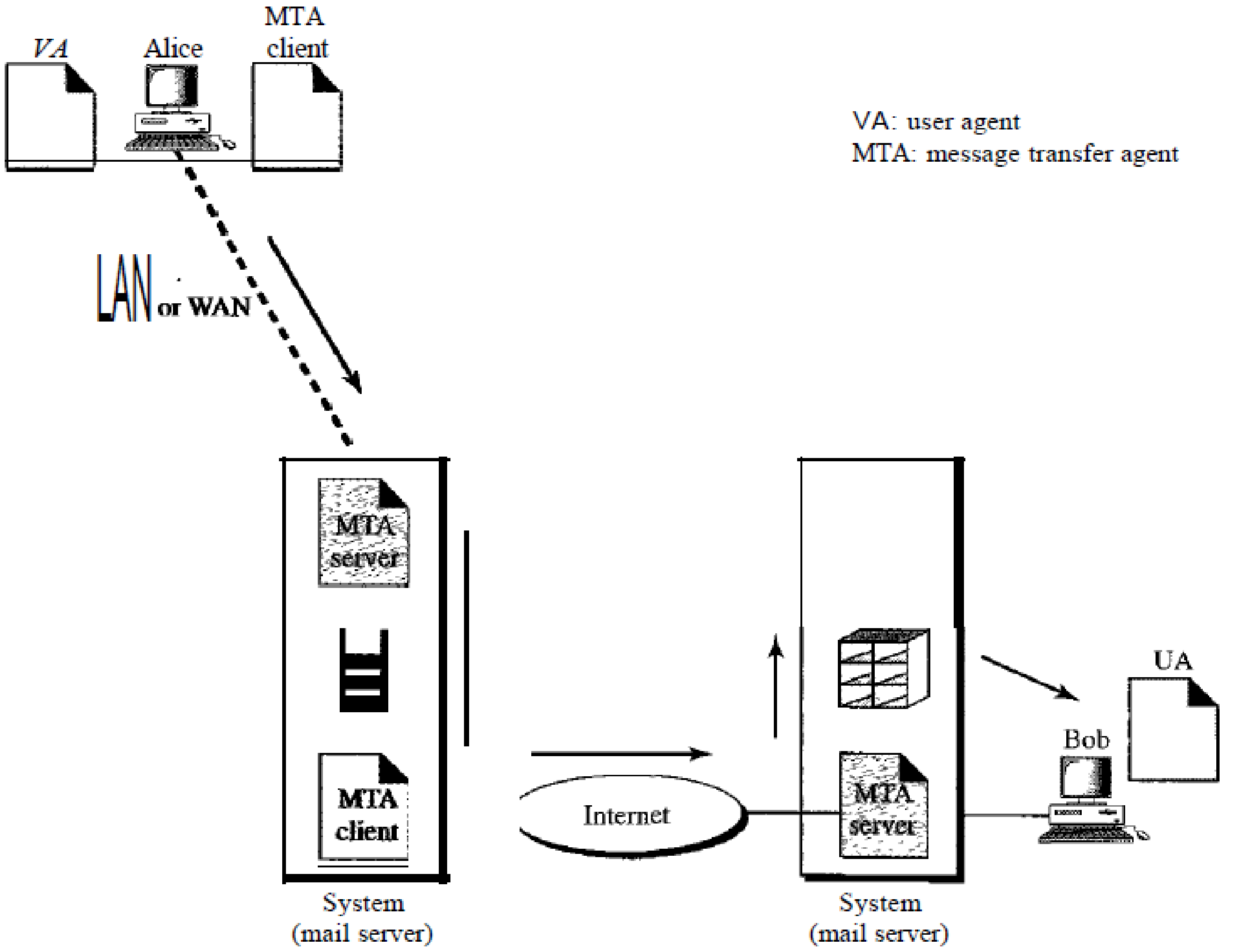
VA: user agent
MTA: message transfer agent



When the sender and the receiver of an e-mail are on different systems, we need two VAs and a pair of MTAs (client and server).

Third Scenario

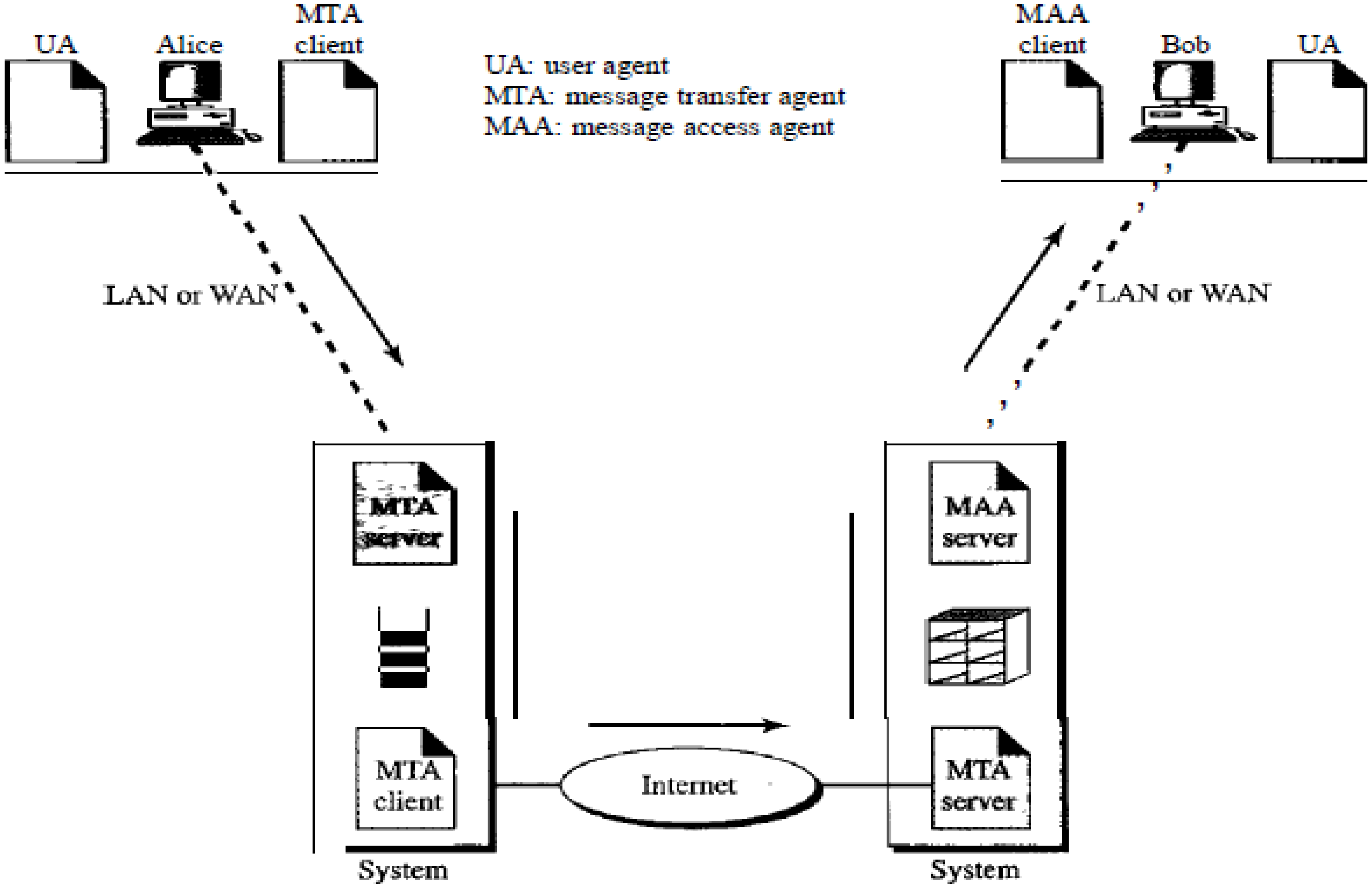
- Bob, as in the second scenario, is directly connected to his system.
- Alice, however, is separated from her system. Either Alice is connected to the system via a point-to-point WAN, such as a dial-up modem, a DSL, or a cable modem; or she is connected to a LAN in an organization that uses one mail server for handling e-mails-all users need to send their messages to this mail server.



When the sender is connected to the mail server via a LAN or a WAN, we need two *VAs* and two paIrs of MTAs (clIent and server).

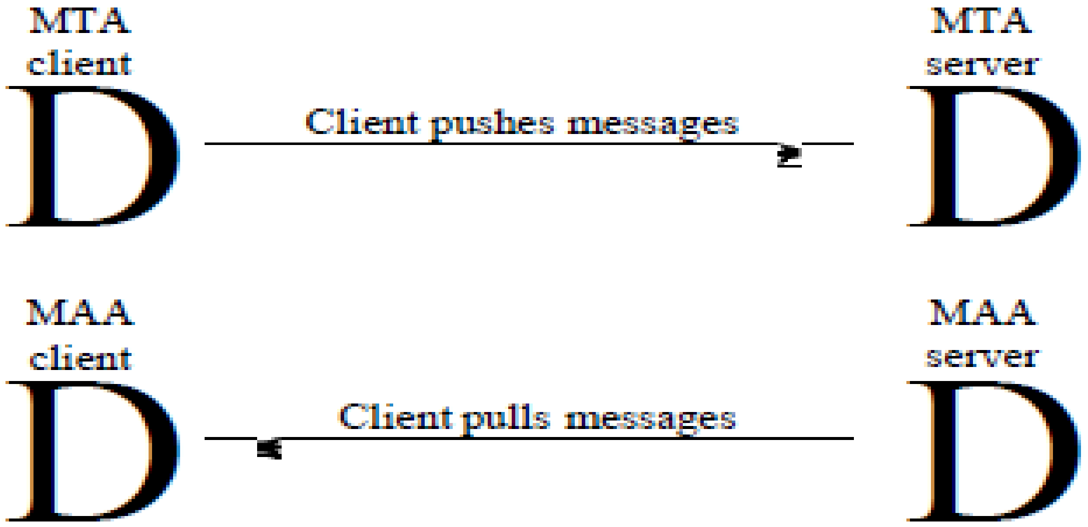
Fourth Scenario

- Bob is also connected to his mail server by a WAN or a LAN. After the message has arrived at Bob's mail server, Bob needs to retrieve it. Here, we need another set of client/server agents, which we call message access agents (MAAs).


UA
Alice
MTA client
UA: user agent
MTA: message transfer agent
MAA: message access agent
MAA client
Bob
UA
LAN or WAN
LAN or WAN
MTA server
MTA client
System
Internet
MAA server
MTA server
System

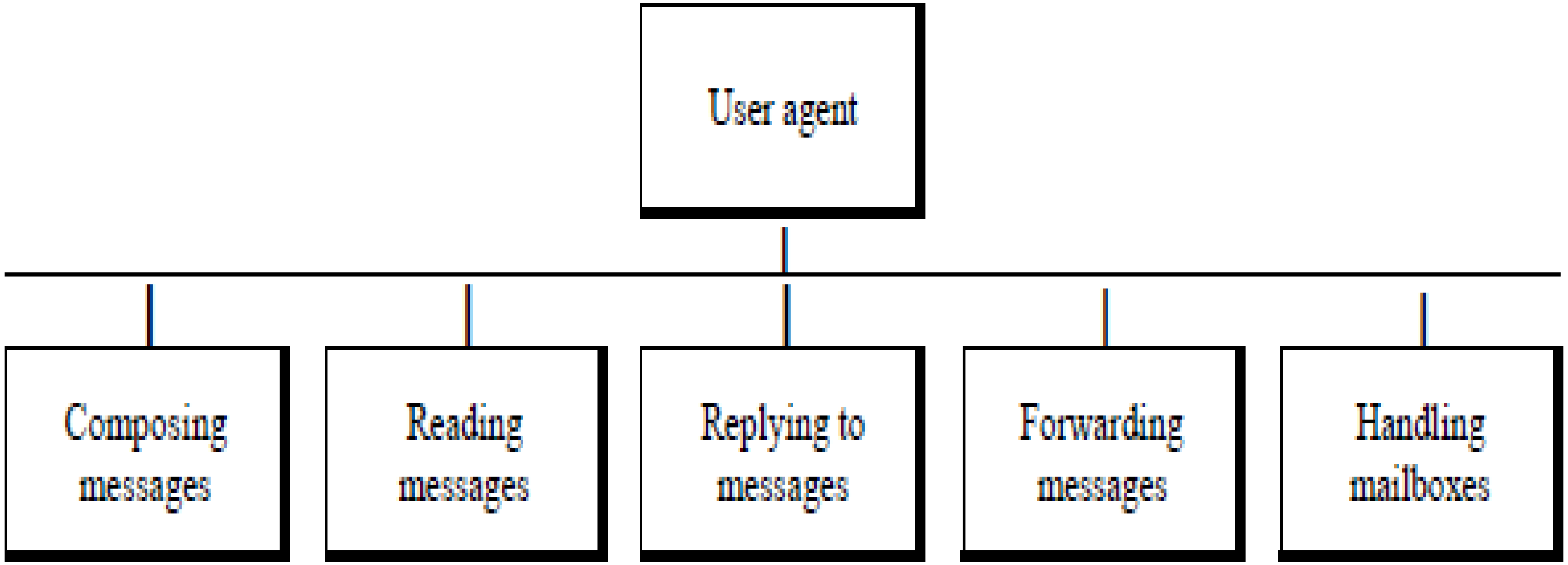
- Note that Bob needs another pair of client/server programs: message
- access programs. This is so because an MTA client/server program is a *push* program: the client pushes the message to the server.
- Bob needs a *pull* program. The client needs to pull the message from the server.

Push versus pull in electronic email



User Agent

- The first component of an electronic mail system is the user agent *(VA).*
- 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 mailboxes.


User agent
Composing messages
Reading messages
Replying to messages
Forwarding messages
Handling mailboxes

Composing Messages:

- A user agent helps the user compose the e-mail message to be sent out.
- Most user agents provide a template on the screen to be filled in by the user.
- Some even have a built-in editor that can do spell checking, grammar checking, and other tasks expected from a sophisticated word processor.
- A user, of course, could alternatively use his or her favorite text editor or word processor to create the message and import it, or cut and paste it, into the user agent template.

Reading Messages:

- The second duty of the user agent is to read the incoming messages.
- When a user invokes a user agent, it first checks the mail in the incoming mailbox.
- Most user agents show a one-line summary of each received mail. Each e-mail contains the following fields.

1. A number field.
2. A flag field that shows the status of the mail such as new, already read but not replied to, or read and replied to.
3. The size of the message.
4. The sender.
5. *The optional subject field.*

Replying to Messages:

- After reading a message, a user can use the user agent to reply to a message.
- A user agent usually allows the user to reply to the original sender or to reply to all recipients of the message.
- The reply message may contain the original message (for quick reference) and the new message.

Forwarding Messages:

- *Replying is defined as sending a message to the sender or* recipients of the copy.
- *Forwarding is defined as sending the message to a third party.*
- *A* user agent allows the receiver to forward the message, with or without extra comments, to a third party.

Handling Mailboxes:

- A user agent normally creates two mailboxes: an inbox and an outbox.
- Each box is a file with a special format that can be handled by the user agent.
- The inbox keeps all the received e-mails until they are deleted by the user.
- The outbox keeps all the sent e-mails until the user deletes them

User Agent Types

There are two types of user agents:

1. Command-driven and

2. GUI-based.

Command-Driven :

- Command-driven user agents belong to the early days of electronic mail.
- They are still present as the underlying user agents in servers. A command-driven user agent normally accepts a one-character command from the keyboard to perform its task.
- For example, a user can type the character r, at the command prompt, to reply to the sender of the message, or type the character R to reply to the sender and all recipients.
- Some examples of command-driven user agents are *mail, pine, and elm.*

GUI-Based :

- Modem 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, Microsoft's Outlook, and Netscape.

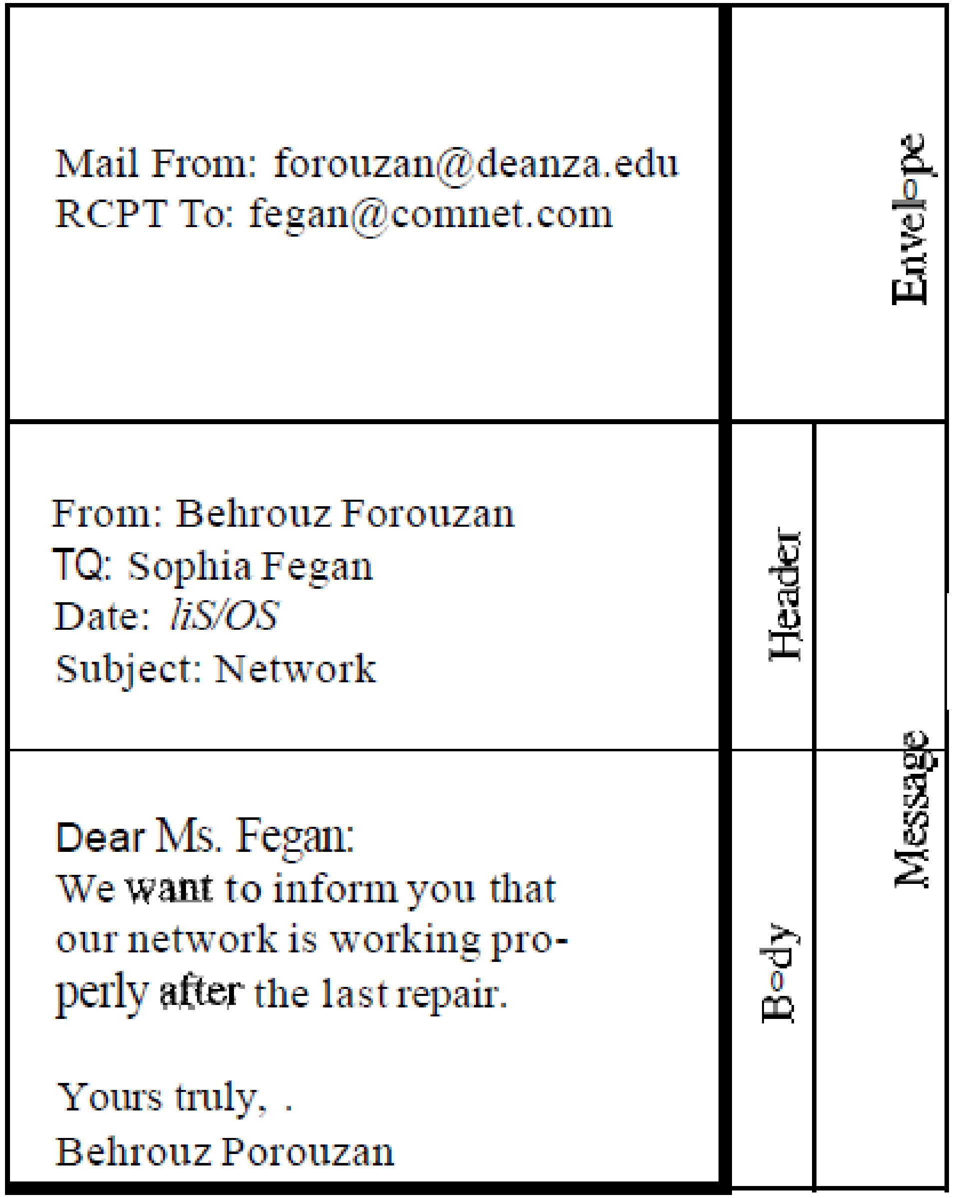
Sending Mail

- To send mail, the user, through the UA, creates mail that looks very similar to postal mail.
- It has an *envelope and a message.*

Envelope: The envelope usually contains the sender and the receiver addresses.

Message: 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.



Mail From: forouzan@deanza.edu
RCPT To: fegan@comnet.com

From: Behrouz Forouzan
TQ: Sophia Fegan
Date: *liS/OS*
Subject: Network

Dear Ms. Fegan:
We want to inform you that our network is working properly after the last repair.

Yours truly, .
Behrouz Porouzan

b. Electromc mall

- 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 *VA 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

Local Part :

- 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.

Domain Name :

- The second part of the address is the domain name.
- An organization usually selects one or more hosts to receive and send e-mail; the hosts are sometimes called *mail servers or exchangers.*
- *The 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).

Mailing 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.
- If there is no mailing list for the alias, the name itself is the receiving address and a single message is delivered to the mail transfer entity.

MIME

- Electronic mail can send messages only in 7-bit ASCII format and it has some limitations.
- For example, it cannot be used for languages that are not supported by 7-bit ASCII characters (such as French, German, Hebrew, Russian, Chinese, and Japanese).
- Also, it cannot be used to send binary files or video or audio data.

- 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 ASCII data and delivers them to the client.
- MIME as a set of software functions that transforms non-ASCII

data (stream of bits) to ASCII data and vice versa.

Message Transfer Agent: SMTP

- The actual mail transfer is done through message transfer agents.
- To send mail, a system must have the client MTA, and to receive mail, a system must have a server MTA.
- The formal protocol that defines the MTA client and server in the Internet is called the 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.

FILE TRANSFER

- Transferring files from one computer to another is one of the most common tasks expected from a networking or internetworking environment.
- one popular protocol involved in transferring files is

  File Transfer Protocol *(FTP).*
- File Transfer Protocol (FTP) is the standard mechanism 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.

Problems

- two systems may use

✓different file name conventions.

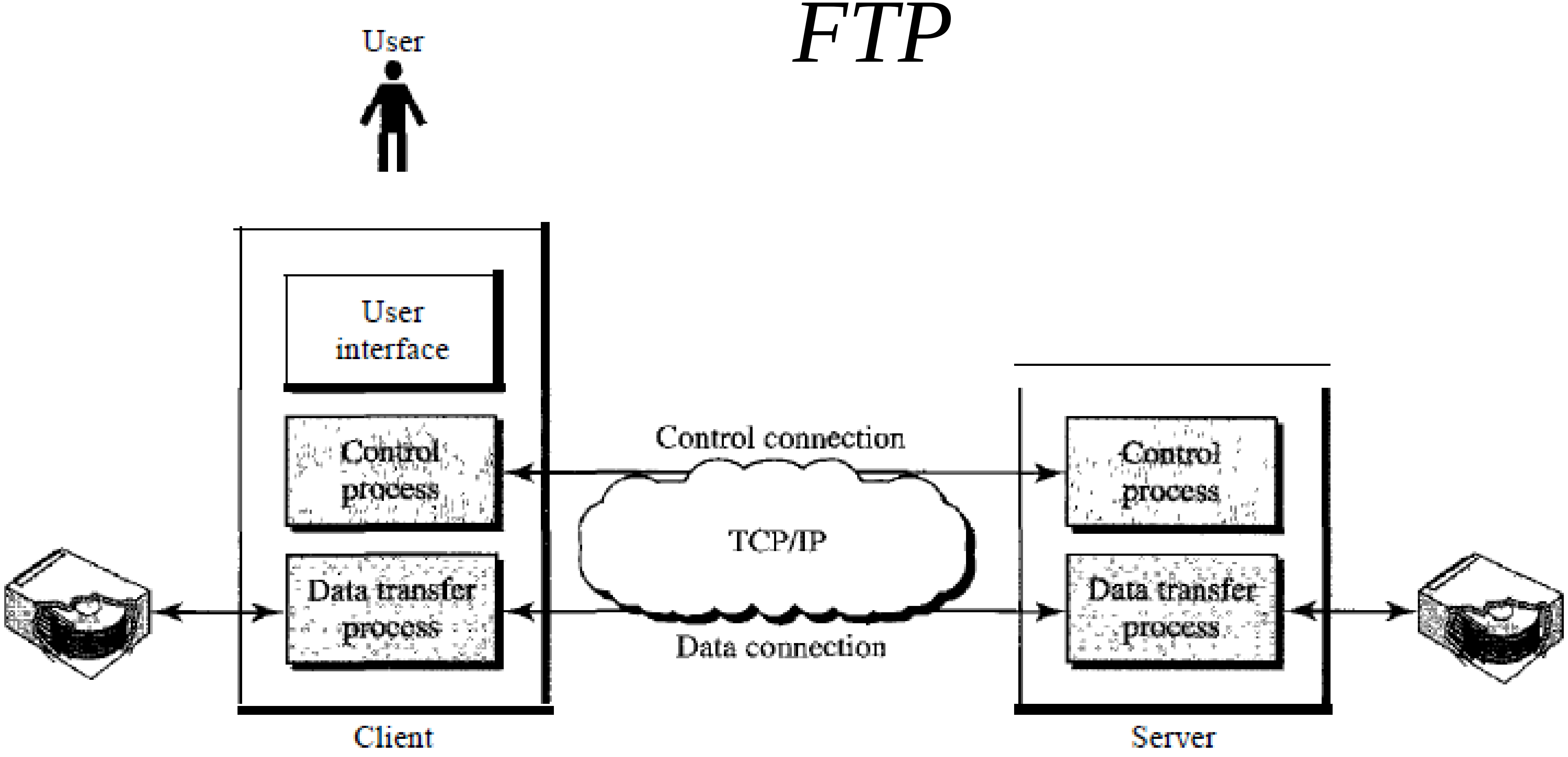
✓different ways to represent text and data.

✓Different directory structures.

- All these problems have been solved by FTP in a very simple and elegant approach.

- FTP establishes two connections between the hosts.
- One connection is used for data transfer, the other for control
- information (commands and responses).
- Separation of commands and data transfer makes FTP more efficient.
- FTP uses the services of TCP. It needs two TCP connections.
- The port 21 is used for the control connection port 20 for the data connection.

FTP


User
User interface
Control process
Data transfer process
Control connection
TCP/IP
Data connection
Control process
Data transfer process
Client
Server

- The client has three components:

✓User interface,

✓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

- The control connection is made between the control processes.
- The data connection is made between the data transfer processes.
- The control connection remains connected during the entire interactive FTP session.
- The data connection is opened and then closed for each file transferred.
- It opens each time commands that involve transferring files are used, and it closes when the file is transferred.

- FTP uses the same approach as SMTP to communicate across the control connection.
- It uses the 7-bit ASCII character set.
- 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 command or response is only one short line, so we need not worry about file format or file structure.
- Each line is terminated with a two-character (carriage return and line feed) end-of-line token.

Communication over Data Connection

- We want to transfer files through the data connection.
- File transfer occurs over the data connection under the control of the commands sent over the control connection.
- File transfer in FTP means one of three things:

1. A file is to be copied from the server to the client.

   This is called *retrieving aft/e.*

   It is done under the supervision of the RETR command,

2. A file is to be copied from the client to the server.

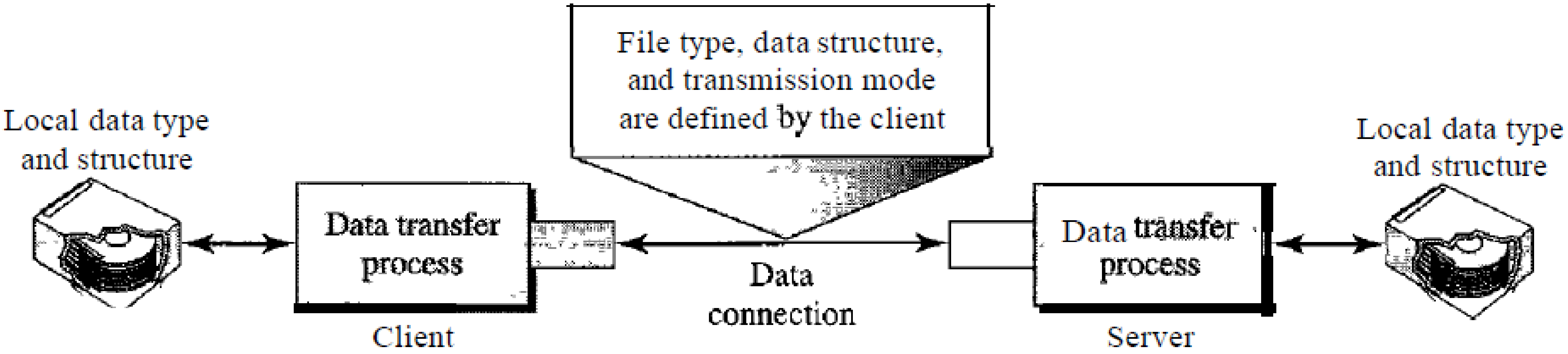
This is called *storing aft/e.*

It is done under the supervision of the STOR command.

3. A list of directory or file names is to be sent from the server to the client.

This is done under the supervision of the LIST command

- 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: file type, data structure, and transmission mode


File type, data structure, and transmission mode are defined by the client
Local data type and structure
Data transfer process
Client
Data connection
Data transfer process
Server
Local data type and structure

- FTP can transfer one of the following file types across the data connection: an ASCII file, EBCDIC file, or image file.
- The ASCII file is the default format for transferring text files.
- The sender transforms the file from its own representation into ASCII characters, and the receiver transforms the ASCII characters to its own representation.

- If one or both ends of the connection use EBCDIC encoding (the file format used by IBM), the file can be transferred using EBCDIC encoding.
- The image file is the default format for transferring binary files.
- The file is sent as continuous streams of bits without any interpretation or encoding.

- FTP can transfer a file across the data connection by using one of the the structure of the data:

✓ file structure,

✓record structure,

✓page structure.

- In the file structure format, the file 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.

Transmission Mode

- FTP can transfer a file across the data connection by using one of the three transmission modes:

✓stream mode,

✓block mode and

✓Compressed mode.

- IN stream mode Data are delivered from FTP to TCP as a continuous stream of bytes.
- TCP is responsible for chopping data into segments of appropriate size.
- If the data are simply a stream of bytes (file structure), no end-of-file is needed.
- End-of-file in this case is the closing of the data connection by the sender.

- The data are divided into records in record structure, each record will have a 1-byte end-of- record (EOR) character and the end of the file will have a I-byte end- of-file (EOF) character.
- In 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 2 bytes define the size of the block in bytes.*

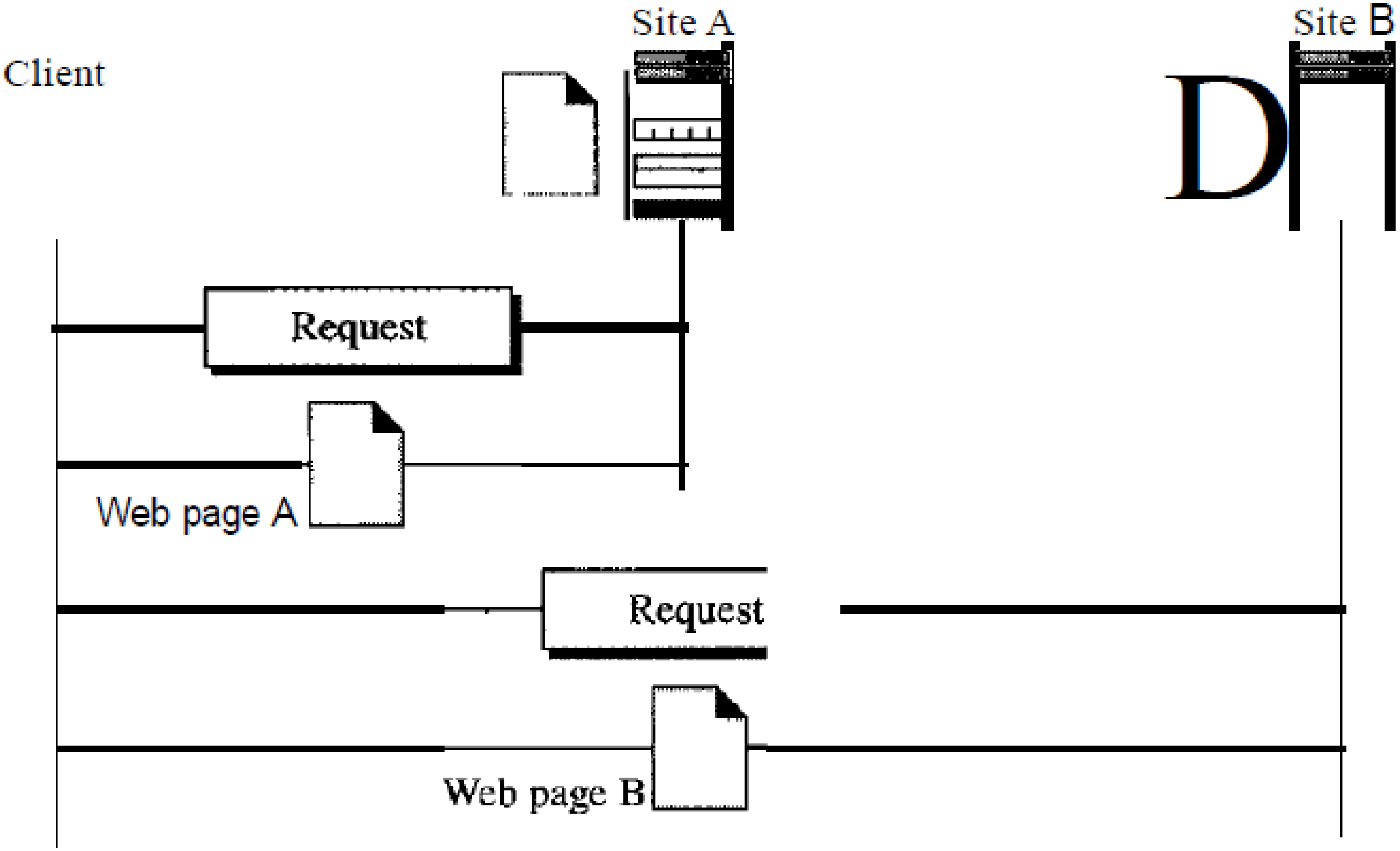
- In the compressed mode, if the file is big, the data can be compressed.
- The compression method normally used is run-length encoding.

WORLD WIDE WEB (WWW)

- The **World Wide Web (WWW)** is a repository of information linked together from points all over the world.
- The WWW has a unique combination of flexibility, portability, and user-friendly features that distinguish it from other services provided by the Internet.
- The WWW project was initiated by CERN (European Laboratory for Particle Physics)

- 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 documents, referred to as *Web pages.*
- *Each Web page can* contain a link to other pages in the same site or at other sites.
- The pages can be retrieved and viewed by using browsers.

Architecture ofWWW



- The client needs to see some information that it knows belongs to site A.
- It sends a request through its browser, a program that is designed to fetch Web documents.
- The request, among other information, includes the address of the site and the Web page, called the URL.
- The server at site A finds the document and sends it to the client.

- When the user views the document, finds some references to other documents, including a Web page at site B.
- The reference has the URL for the new site.
- The user is also interested in seeing this document.
- The client sends another request to the new site, and the new page is retrieved.

Client (Browser)

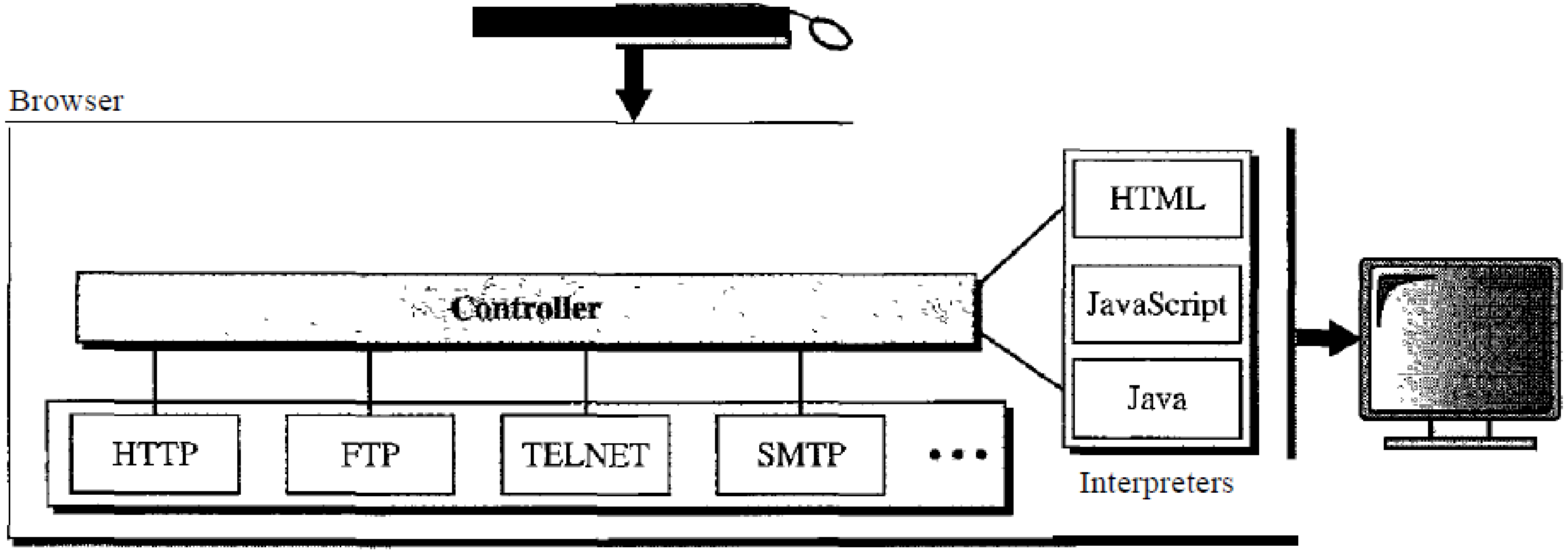
- A variety of vendors offer commercial browsers that interpret and display a Web document, and all use nearly the same architecture.
- Each browser usually consists of three parts:

-- a controller,

client protocol, and

interpreters.

- The controller receives input from the keyboard or the mouse and uses the client programs to access the document.


Browser
Controller
HTTP
FTP
TELNET
SMTP
• • •
HTML
JavaScript
Java
Interpreters

- 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 one of the protocols, such as FTP or HTTP.
- The interpreter can be HTML, Java, or JavaScript, depending on the type of document.

Server

- The Web page is stored at the server.
- Each time a client request arrives, the corresponding document is sent to the client.
- To improve efficiency, servers normally store requested files in a cache in memory.
- A server can answer more than one request at a time through multithreading or multiprocessing.

Uniform Resource Locator

- A client that wants to access a Web page needs the address.
- To facilitate the access of documents distributed throughout the world, HTTP uses locators.
- The uniform resource locator (URL) is a standard for specifying any kind of information on the Internet.
- The URL defines four things:

-protocol, host computer, port, and path

The *protocol*

- *client/server program used to retrieve the document. (*FTP or HTTP.)

The host

- the computer on which the information is located, although the name of the computer can be an alias.

The URL

- can optionally contain the port number of the server.
- If the *port is* included, it is inserted between the host and the path, and it is separated from the host by a colon.

The URL

- can optionally contain the port number of the server.
- If the *port is* included, it is inserted between the host and the path, and it is separated from the host by a colon.

Path

- the pathname of the file where the information is located.
- the path can itself contain slashes.