# Computer Network

**Lecture taken by**

**Dharmendra Kumar**
**(Associate Professor)**
**United College of Engineering and Research, Prayagraj**

# UNIT-5

# Domain Name System(DNS)

# Domain Name System(DNS)

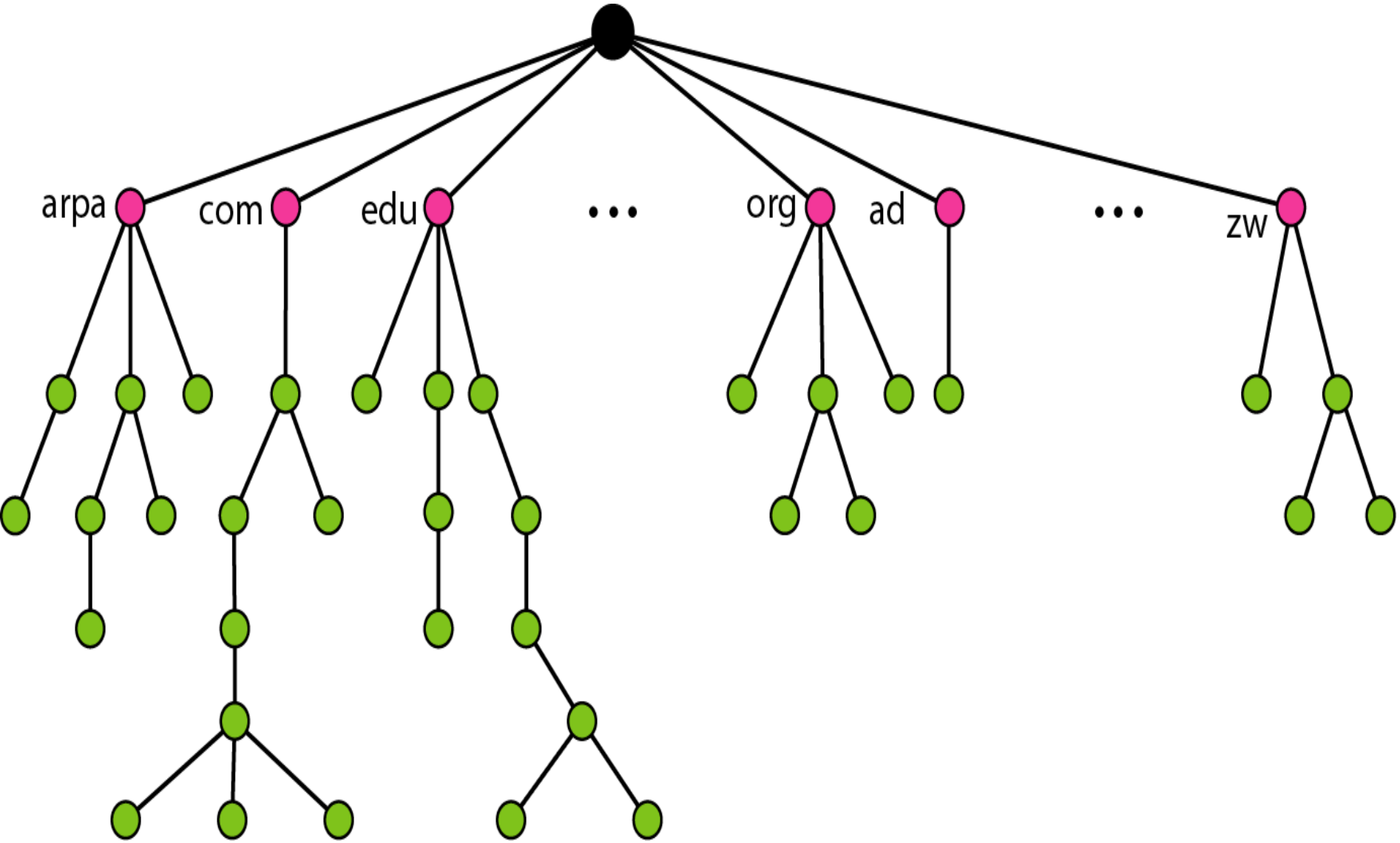
- Domain Name System is a client/server based application layer protocol.
- It translates a domain name (eg. nec.edu.np) into an IP address (eg. 202.37.94.177).
- The DNS has a distributed database that resides on multiple machines on the Internet.

# Domain Name System(DNS)

**DOMAIN NAME SPACE**

- Domain name space is designed in the form of hierarchical name space.
- 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.
- It is shown in the following figure:-

# Domain Name System(DNS)



# Domain Name System(DNS)

**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).

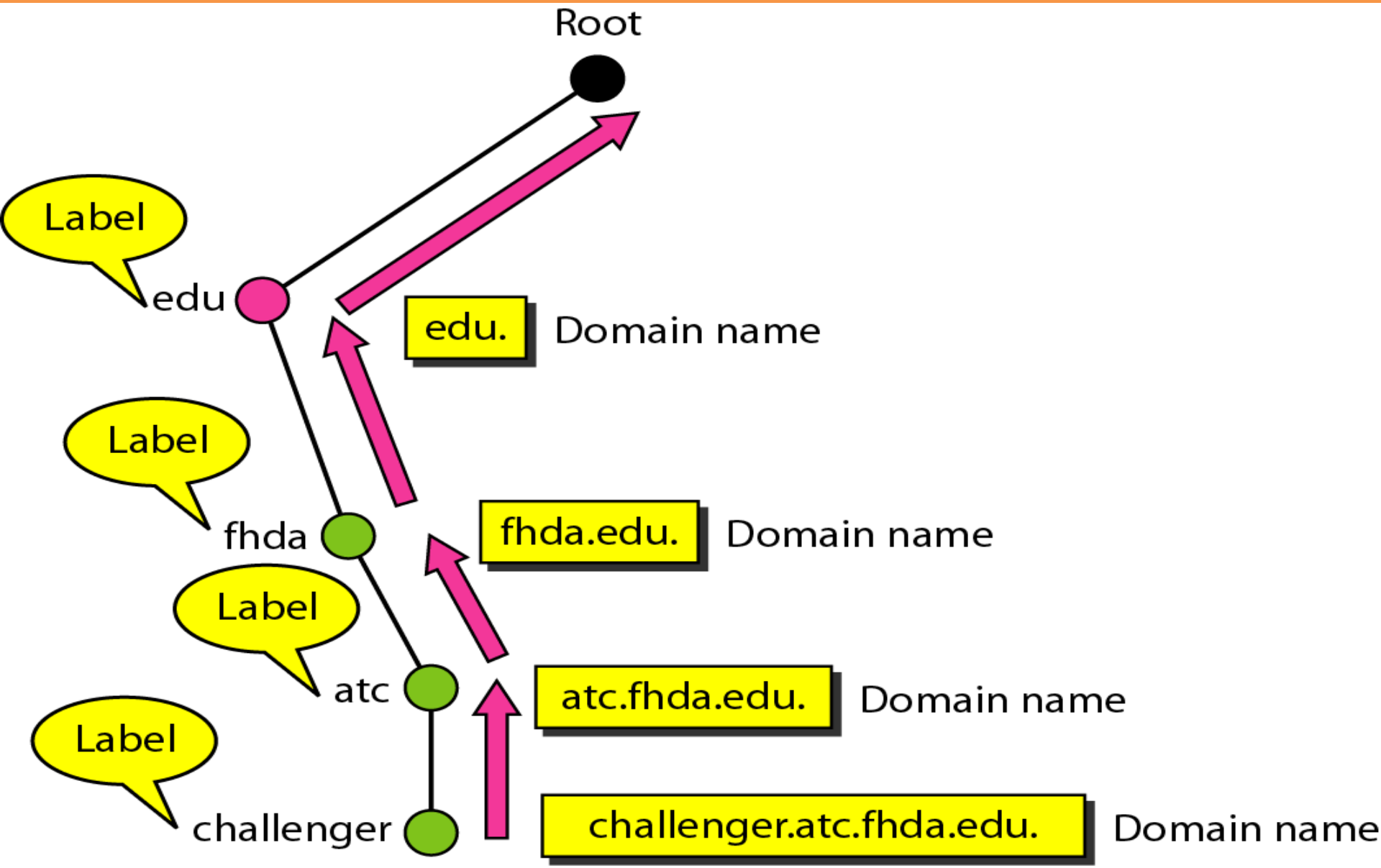
- DNS requires that children of a node (nodes that branch from the same node) have different labels, which guarantees the uniqueness of the domain names.

# Domain Name System(DNS)

**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.
- Following figure shows some domain names.

# Domain Name System(DNS)



# Domain Name System(DNS)

**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.

**Example:** challenger.ate.tbda.edu.

# Domain Name System(DNS)

**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 Name System(DNS)

FQDN

challenger.atc.fhda.edu.
cs.hmme.com.
www.funny.int.

PQDN

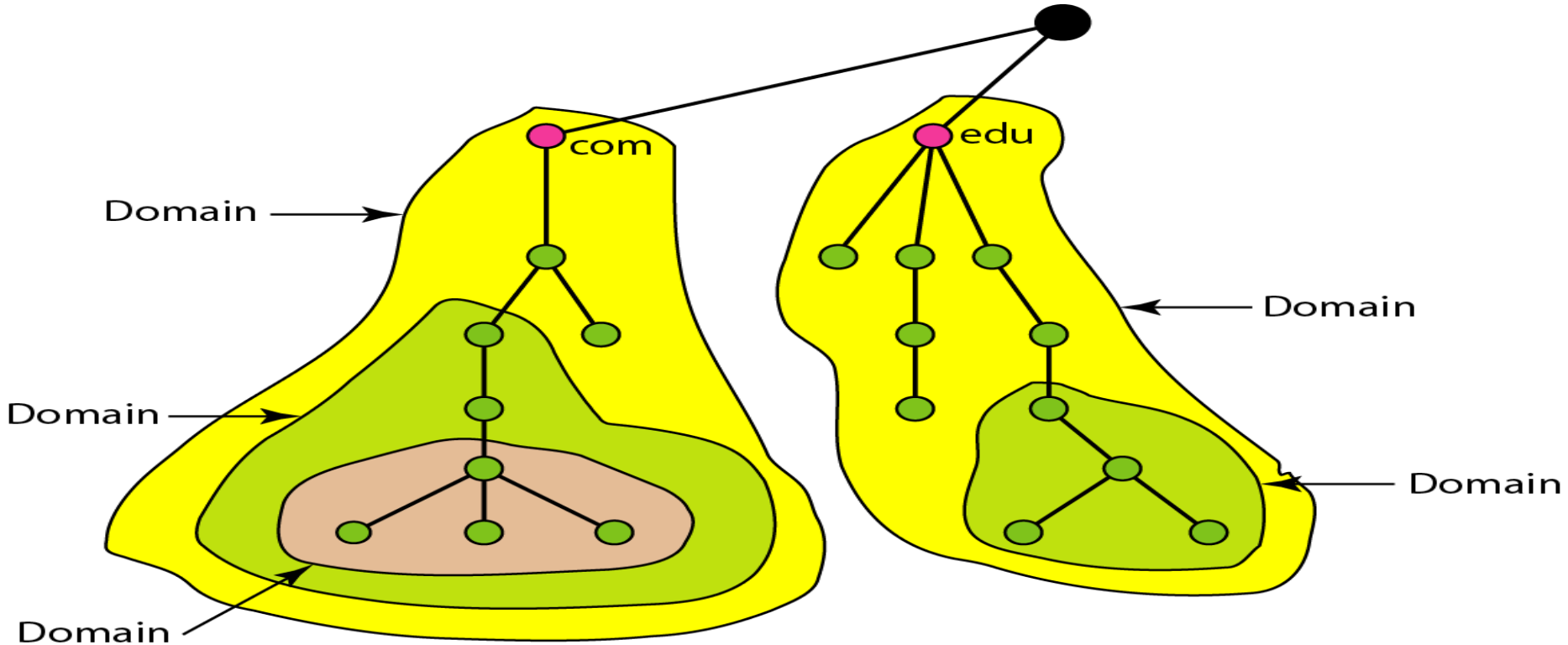
challenger.atc.fhda.edu
cs.hmme
www

# Domain Name System(DNS)

## Domain

A domain is a sub-tree of the domain name space. The name of the domain is the domain name of the node at the top of the sub-tree. Following figure shows some domains.

**Note:** A domain may itself be divided into domains.



# Domain Name System(DNS)

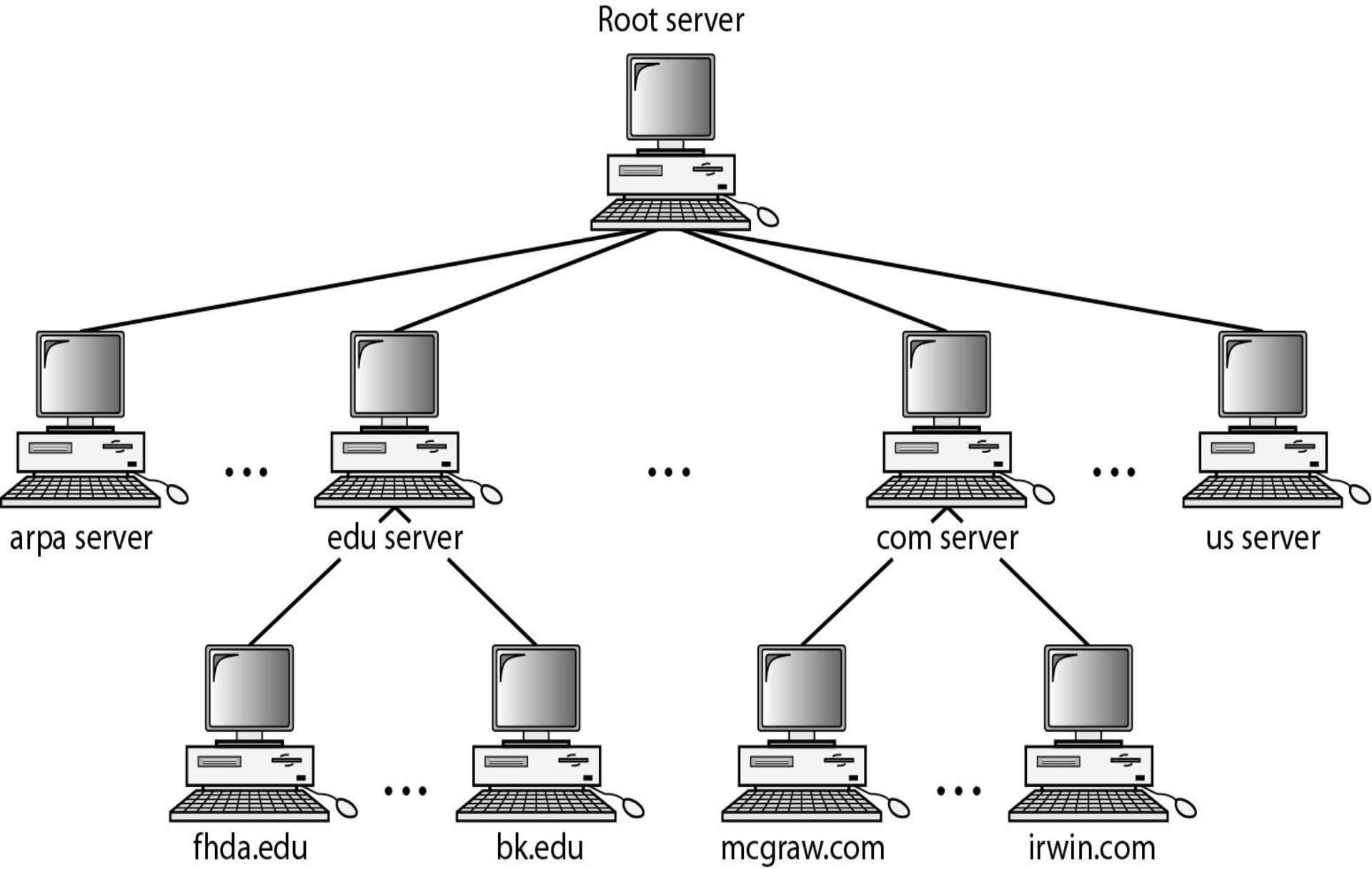
**DISTRIBUTION OF NAME SPACE**

- It is very inefficient and also unreliable to have just one computer store all domain name space.
- It is inefficient because responding to requests from all over the world places a heavy load on the system.
- It is unreliable because any failure makes the data inaccessible.

**Hierarchy of Name Servers**

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

# Domain Name System(DNS)



# Domain Name System(DNS)

**Zone**

- Since the complete domain name hierarchy can not be stored on a single server, so it is divided among many servers. What a server is responsible for or has authority over is called a zone.

- We can define a zone as a contiguous part of the entire tree.

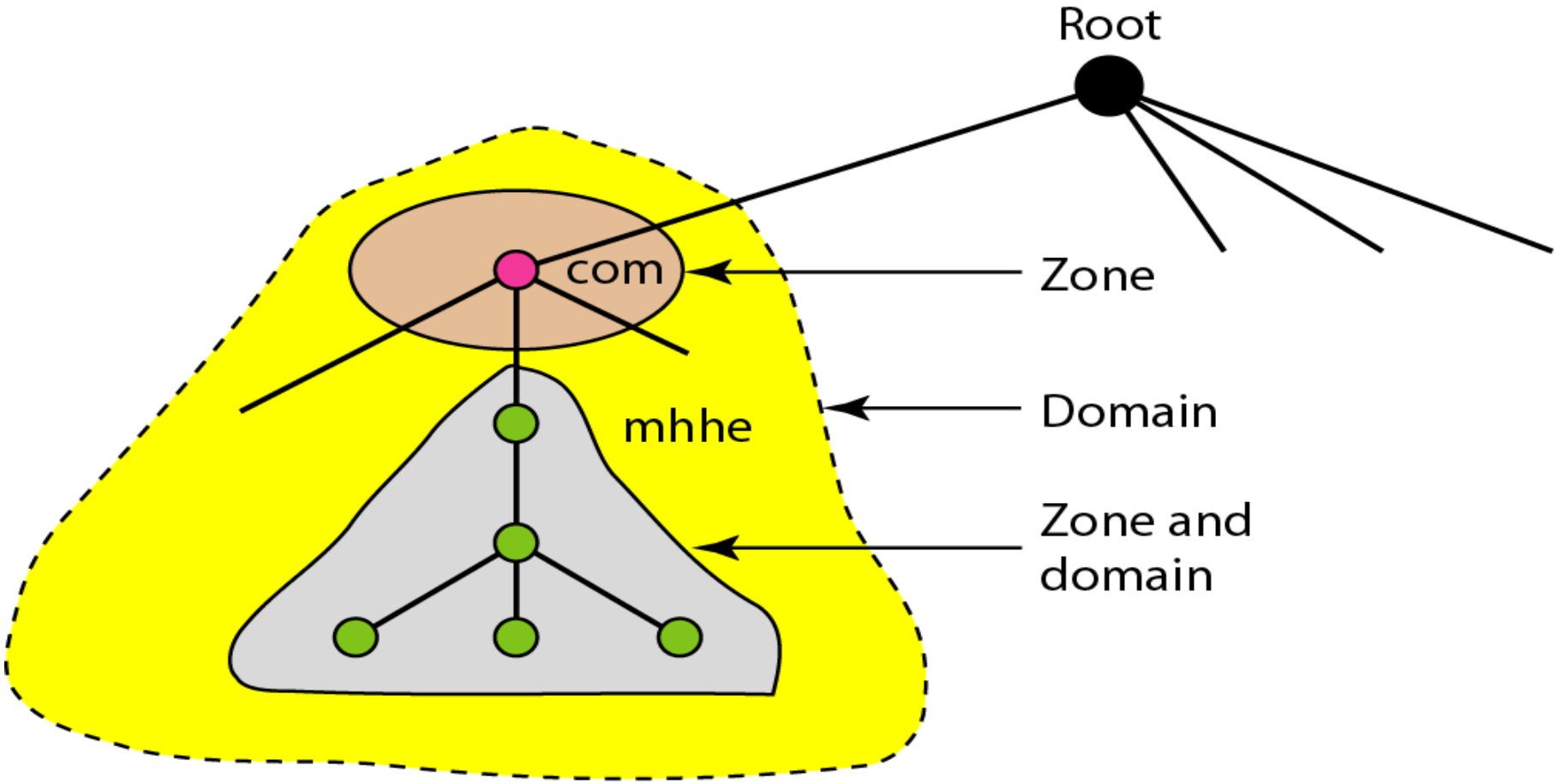
- If a server accepts responsibility for a domain and does not divide the domain into smaller domains, the domain and the zone refer to the same thing.

- The server makes a database called a zone file and keeps all the information for every node under that domain.

- If a server divides its domain into subdomains and delegates part of its authority to other servers, then domain and zone refer to different things.

# Domain Name System(DNS)

- The information about the nodes in the subdomains is stored in the servers at the lower levels, with the original server keeping some sort of reference to these lower-level servers.



# Domain Name System(DNS)

**Root Server**

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

# Domain Name System(DNS)

**Primary and Secondary Servers**

- DNS defines two types of servers: **primary and secondary**. A primary server is a server that stores a file about the zone for which it is an authority. It is responsible for creating, maintaining, and updating the zone file. It stores the zone file on a local disk.

- A secondary server is a server that transfers the complete information about a zone from another server (primary or secondary) and stores the file on its local disk. The secondary server neither creates nor updates the zone files. If updating is required, it must be done by the primary server, which sends the updated version to the secondary.

- The primary and secondary servers are both authoritative for the zones they serve.

# Domain Name System(DNS)

**Note:** A server can be a primary server for a specific zone and a secondary server for another zone.

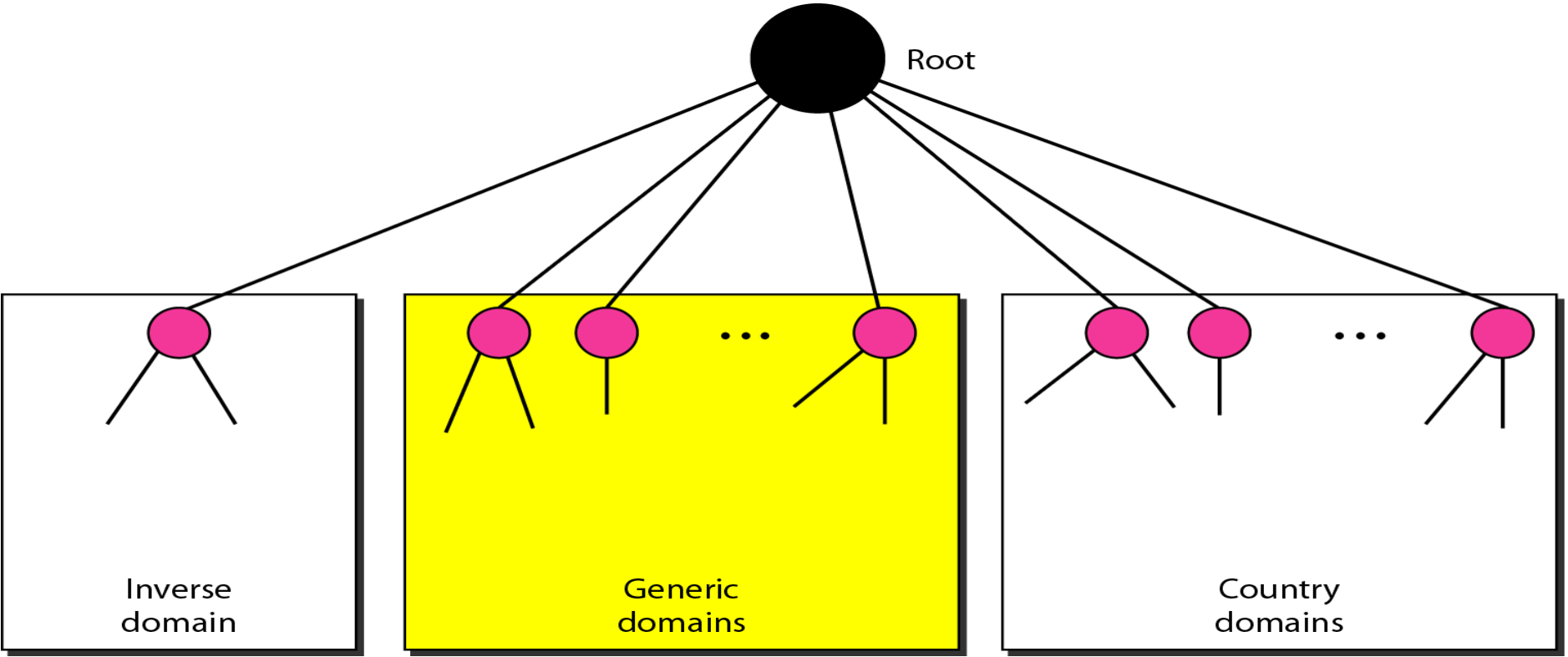
- A primary server loads all information from the disk file; the secondary server loads all information from the primary server.

- When the secondary downloads information from the primary, it is called **zone transfer**.

# Domain Name System(DNS)

## DNS IN THE INTERNET

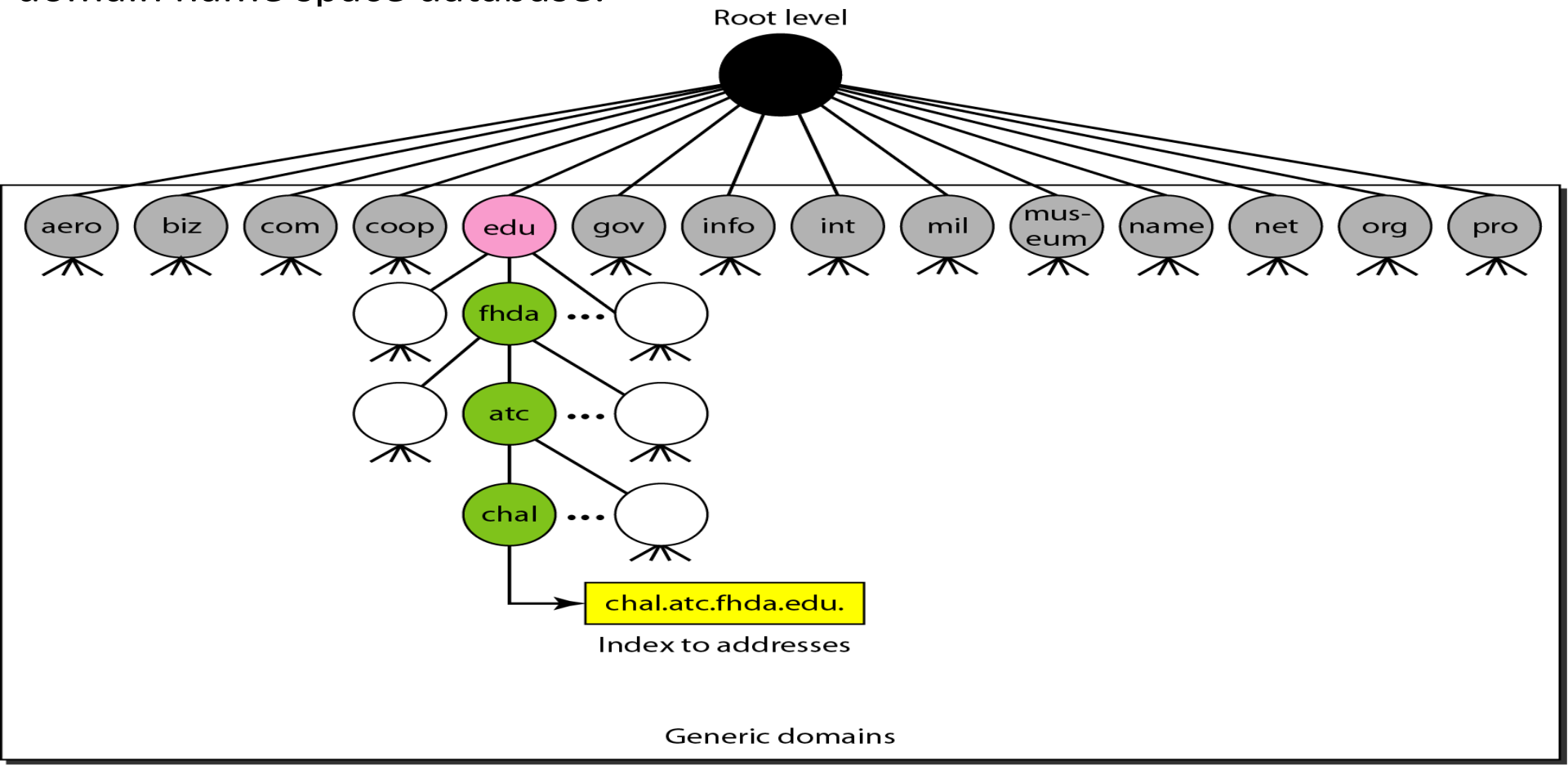
- DNS is a protocol that can be used in different platforms.
- In the Internet, the domain name space (tree) is divided into three different sections: generic domains, country domains, and the inverse domain.



# Domain Name System(DNS)

## 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 domains

# Domain Name System(DNS)

- In the tree, the first level in the generic domains section allows 14 possible labels. These labels describe the organization types.

| *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 |

# Domain Name System(DNS)

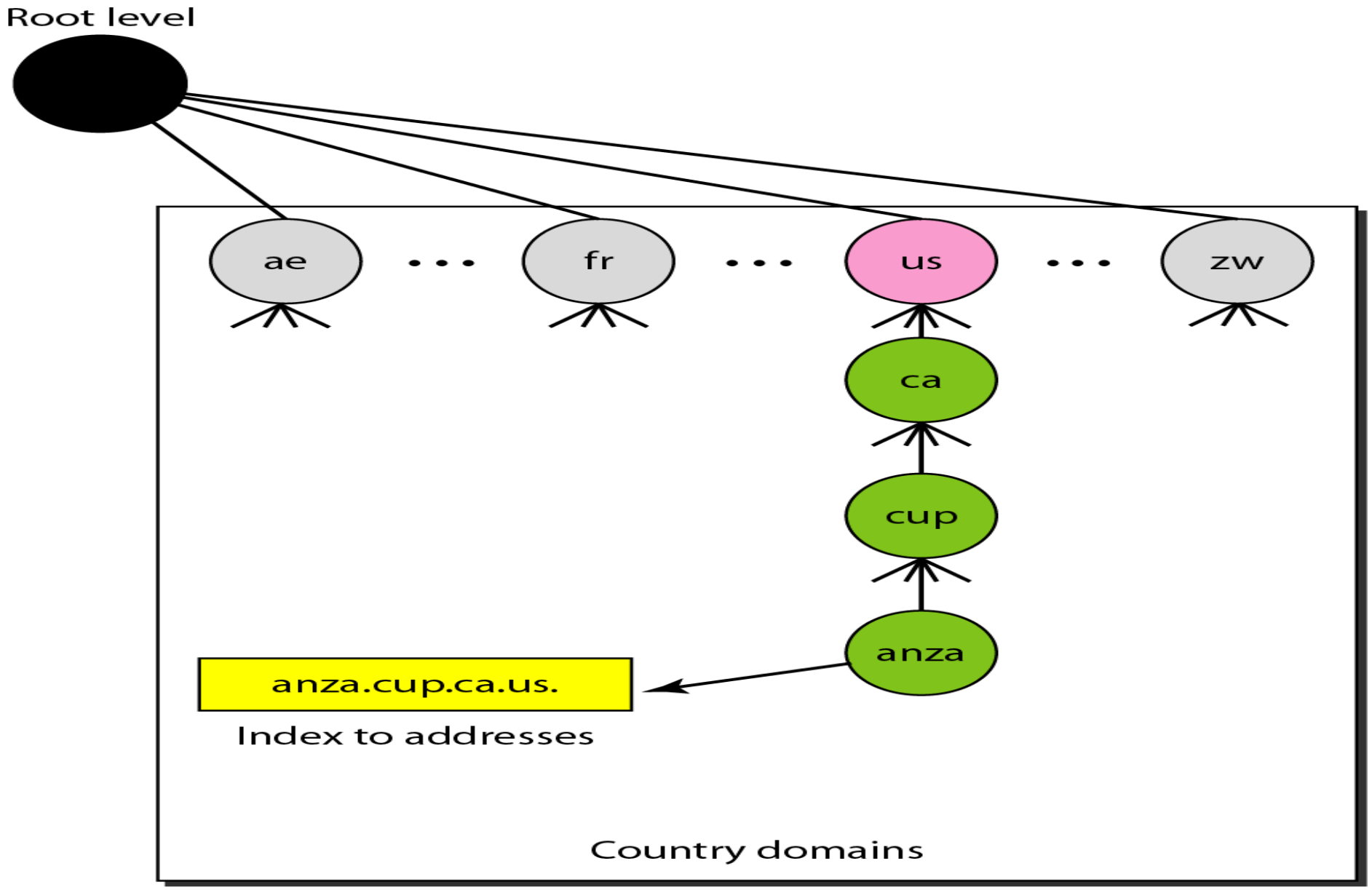
**Country Domains**

- 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.

- Following figure shows the country domains section.

- The address **anza.cup.ca.us** can be translated to De Anza College in Cupertino, California, in the United States.

# Domain Name System(DNS)

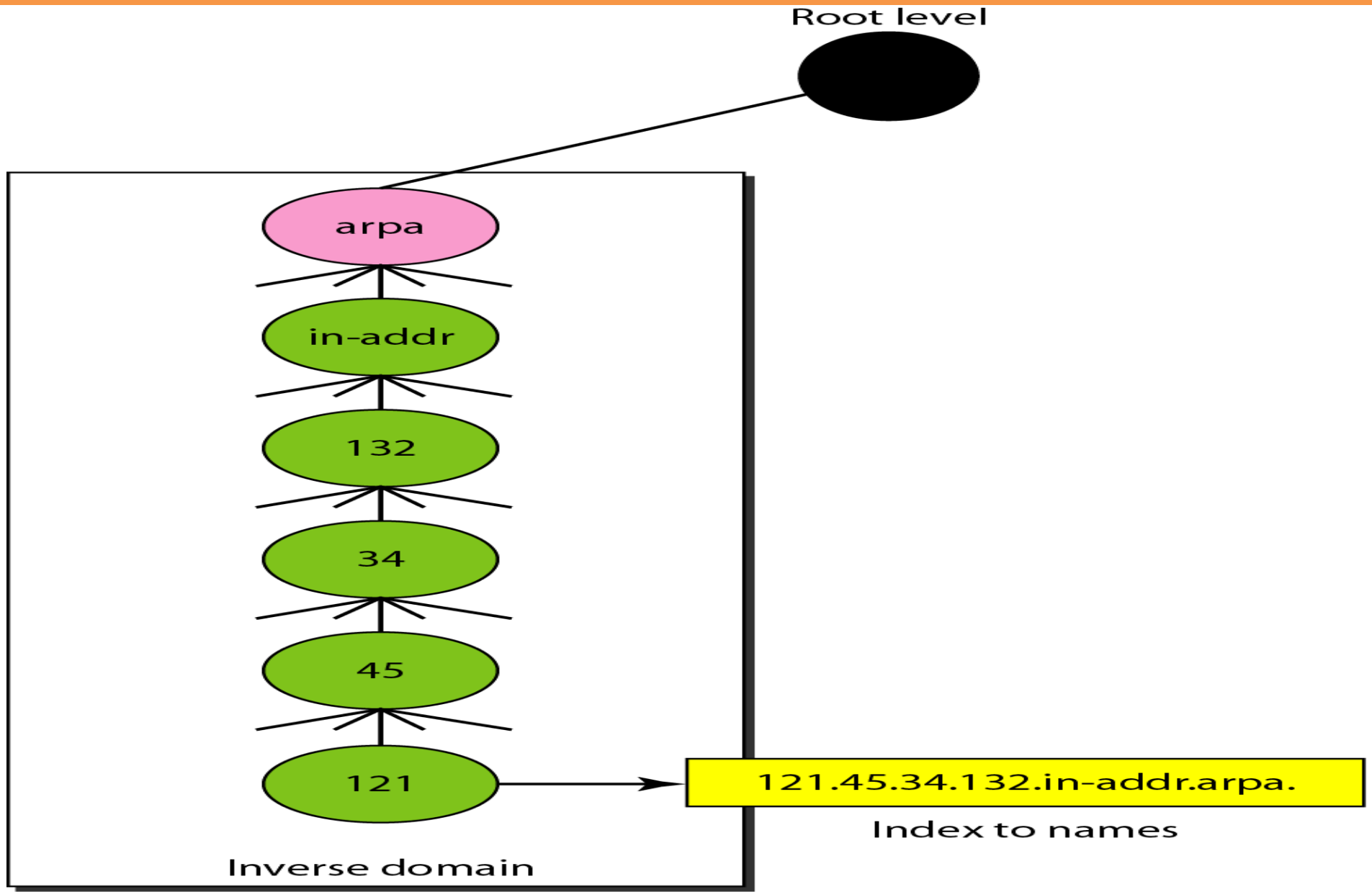


# Domain Name System(DNS)

**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.
- 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. This type of query is called an inverse or pointer (PTR) query.
- To handle a pointer query, the inverse domain is added to the domain name space with the first-level node called arpa. The second level is also one single node named in-addr (for inverse address). The rest of the domain defines IP addresses.

# Domain Name System(DNS)



# Domain Name System(DNS)

**RESOLUTION**

Mapping a name to an address or an address to a name is called name-address resolution.

**Resolver**

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

# Domain Name System(DNS)

**Mapping Names to Addresses**

- 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.
- If the domain name is from the country domains section, the resolver receives a domain name such as "ch.jhda.cu.ca.us.". The procedure is the same.

# Domain Name System(DNS)

**Mapping Addresses to Names**

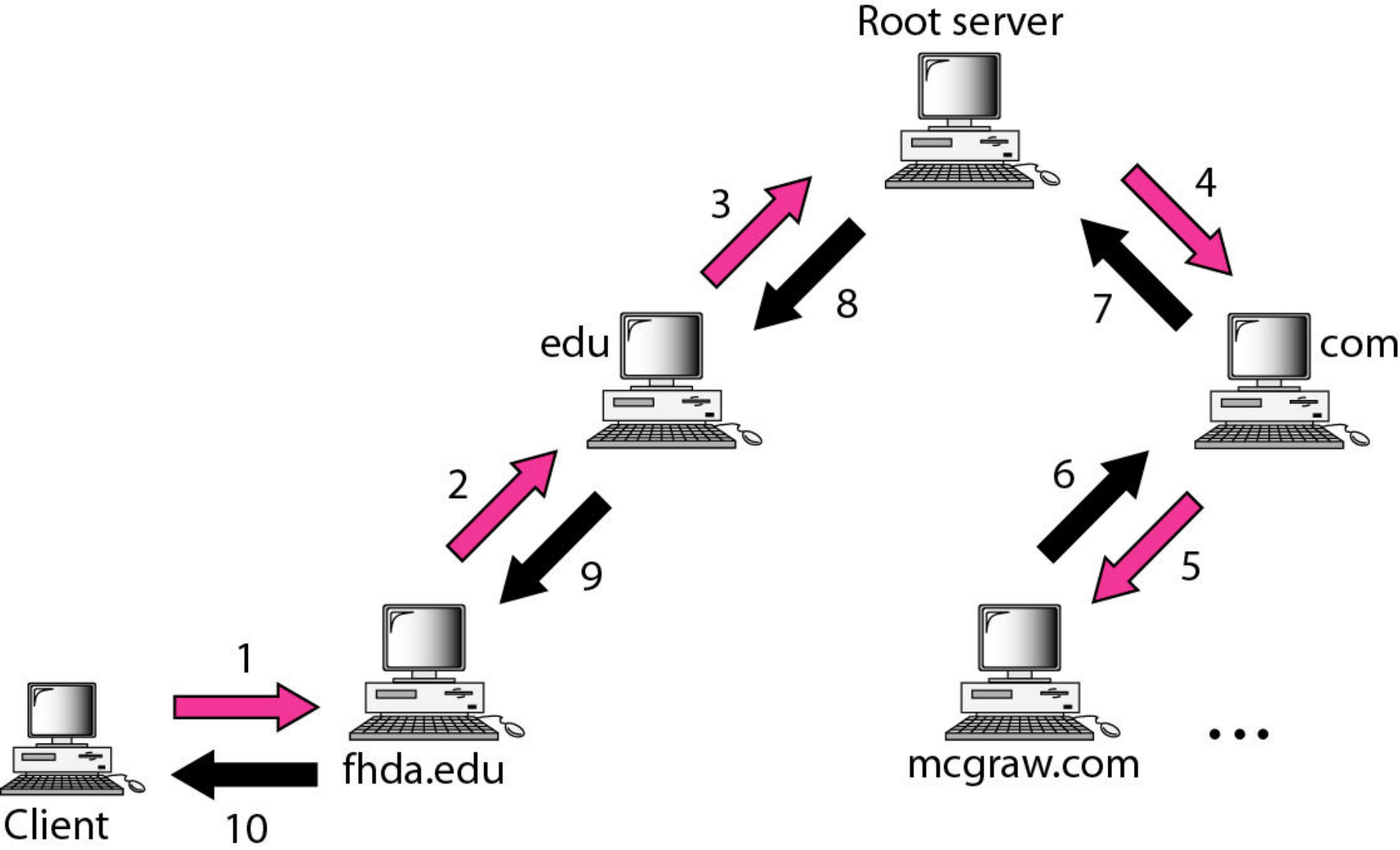
- A client send an IP address to a server to be mapped to a domain name. This is called a PTR query.
- To answer queries of this kind, DNS uses the inverse domain. However, in the request, the IP address is reversed and the two labels in-addr and arpa are appended to create a domain acceptable by the inverse domain section.
- For example, if the resolver receives the IP 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.

# Domain Name System(DNS)

**Recursive Resolution**

- The client (resolver) can ask for a recursive answer from a name server. This means that the resolver expects the server to supply the final answer.
- If the server is the authority for the domain name, it checks its database and responds.
- If the server is not the authority, it sends the request to another server (the parent usually) and waits for the response. If the parent is the authority, it responds; otherwise, it sends the query to yet another server. When the query is finally resolved, the response travels back until it finally reaches the requesting client. This is called recursive resolution and is shown in the following figure:-

# Domain Name System(DNS)

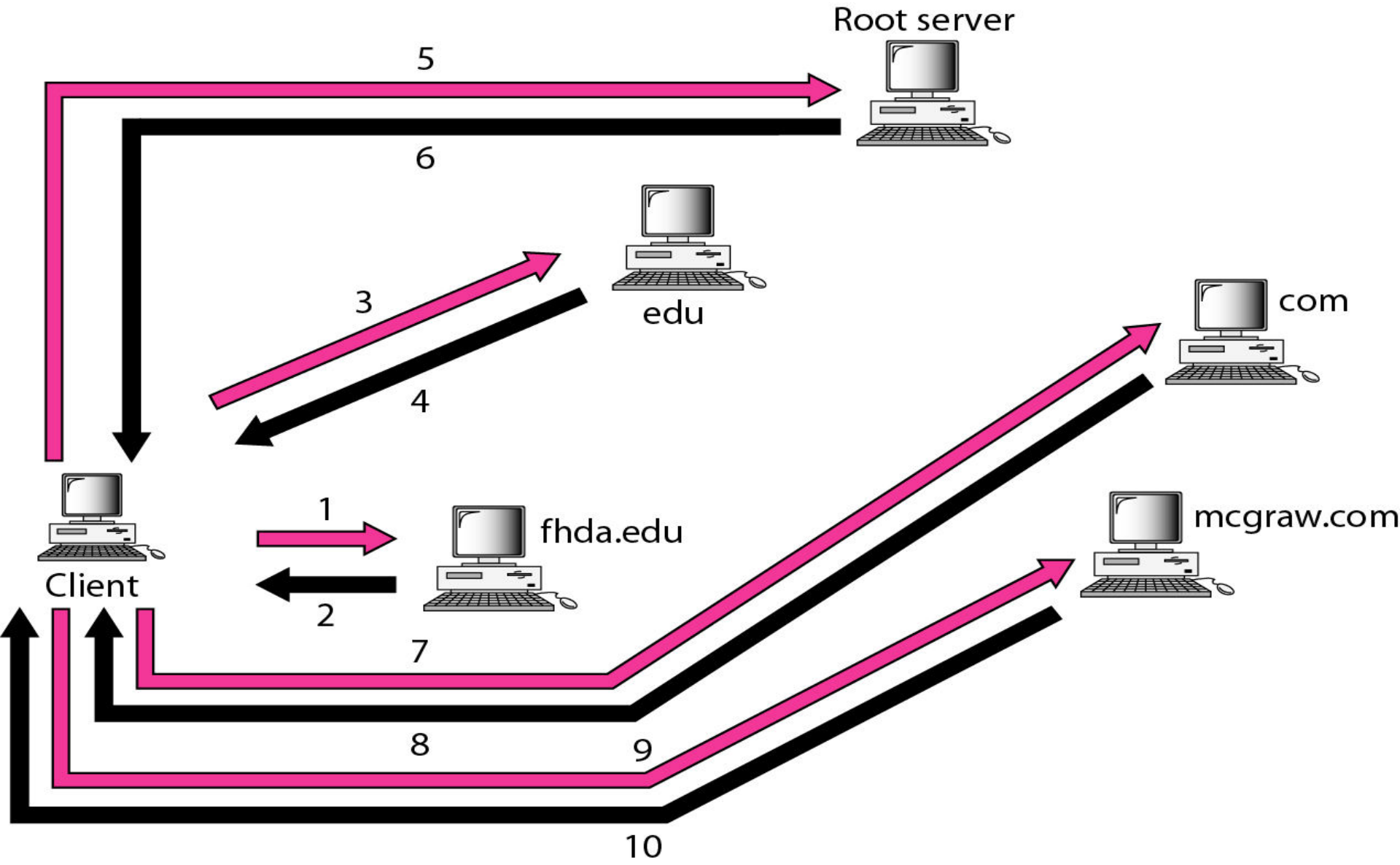


# Domain Name System(DNS)

**Iterative Resolution**

- In this approach, 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 the 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.
- In following figure, the client queries four servers before it gets an answer from the mcgraw.com server.

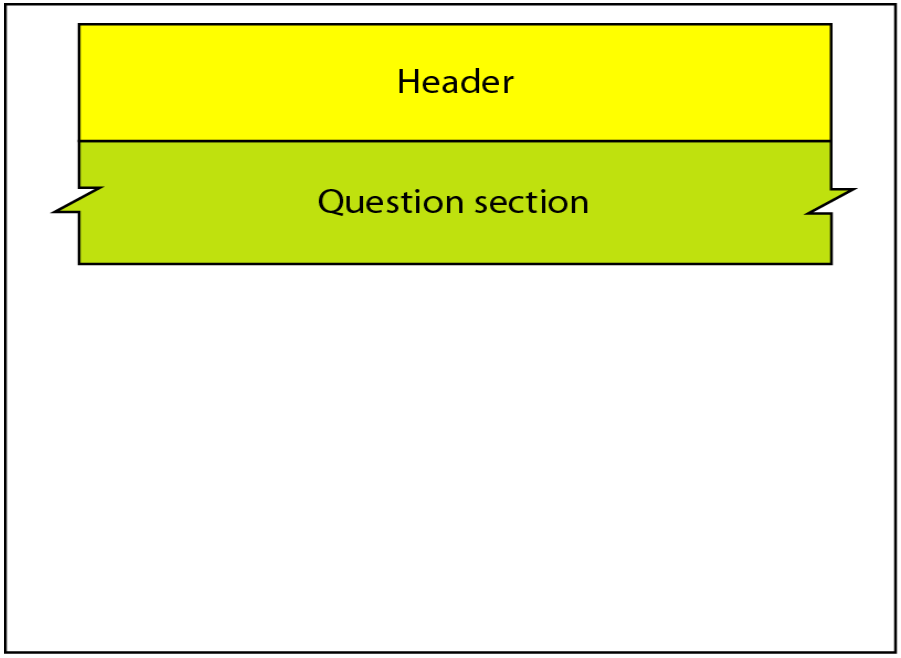
# Domain Name System(DNS)



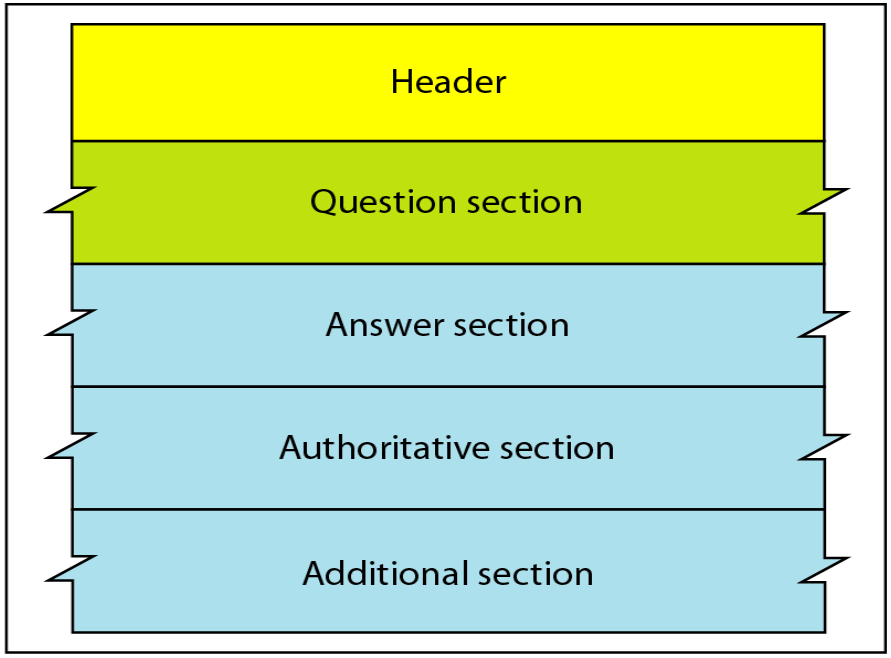
# Domain Name System(DNS)

## DNS MESSAGES

- DNS has two types of messages: **query and response**.
- Both types have the same format. The query message consists of a header and question records; the response message consists of a header, question records, answer records, authoritative records, and additional records.




a. Query




b. Response

# Domain Name System(DNS)

**Header**

Both query and response messages have the same header format with some fields set to zero for the query messages. The header is 12 bytes, and its format is shown in the following figure:-

| Identification | Flags |
| --- | --- |
| Number of question records | Number of answer records (all 0s in query message) |
| Number of authoritative records (all 0s in query message) | Number of additional records (all 0s in query message) |

# Domain Name System(DNS)

- The **identification subfield** is used by the client to match the response with the query. The client uses a different identification number each time it sends a query. The server duplicates this number in the corresponding response.

- The **flags subfield** is a collection of subfields that define the type of the message, the type of answer requested, the type of desired resolution (recursive or iterative), and so on.

- The **number of question records subfield** contains the number of queries in the question section of the message.

# Domain Name System(DNS)

- The **number of answer records subfield** contains the number of answer records in the answer section of the response message. Its value is zero in the query message.

- The **number of authoritative records subfield** contains the number of authoritative records in the authoritative section of a response message. Its value is zero in the query message.

- Finally, the **number of additional records subfield** contains the number additional records in the additional section of a response message. Its value is zero in the query message.

# Domain Name System(DNS)

**Question Section**

This is a section consisting of one or more question records. It is present on both query and response messages.

**Answer Section**

This is a section consisting of one or more resource records. It is present only on response messages. This section includes the answer from the server to the client (resolver).

**Authoritative Section**

This is a section consisting of one or more resource records. It is present only on response messages. This section gives information (domain name) about one or more authoritative servers for the query.

# Domain Name System(DNS)

**Additional Information Section**

This is a section consisting of one or more resource records. It is present only on response messages. This section provides additional information that may help the resolver.

For example, a server may give the domain name of an authoritative server to the resolver in the authoritative section, and include the IP address of the same authoritative server in the additional information section.

# Domain Name System(DNS)

**TYPES OF RECORDS**

Two types of records are used in DNS.

**Question Record**

- The question records are used in the question section of the query and response messages.
- A question record is used by the client to get information from a server. This contains the domain name.

**Resource Record**

- The resource records are used in the answer, authoritative, and additional information sections of the response message.
- Each domain name (each node on the tree) is associated with a record called the resource record. The server database consists of resource records. Resource records are also what is returned by the server to the client.

# Remote Logging, Electronic Mail, and File Transfer

# REMOTE LOGGING

- Remote Login is a process in which user can login into remote site i.e. computer use services that are available on the remote computer.
- After logging on, a user can use the services available on the remote computer and transfer the results back to the local computer.

**TELNET**

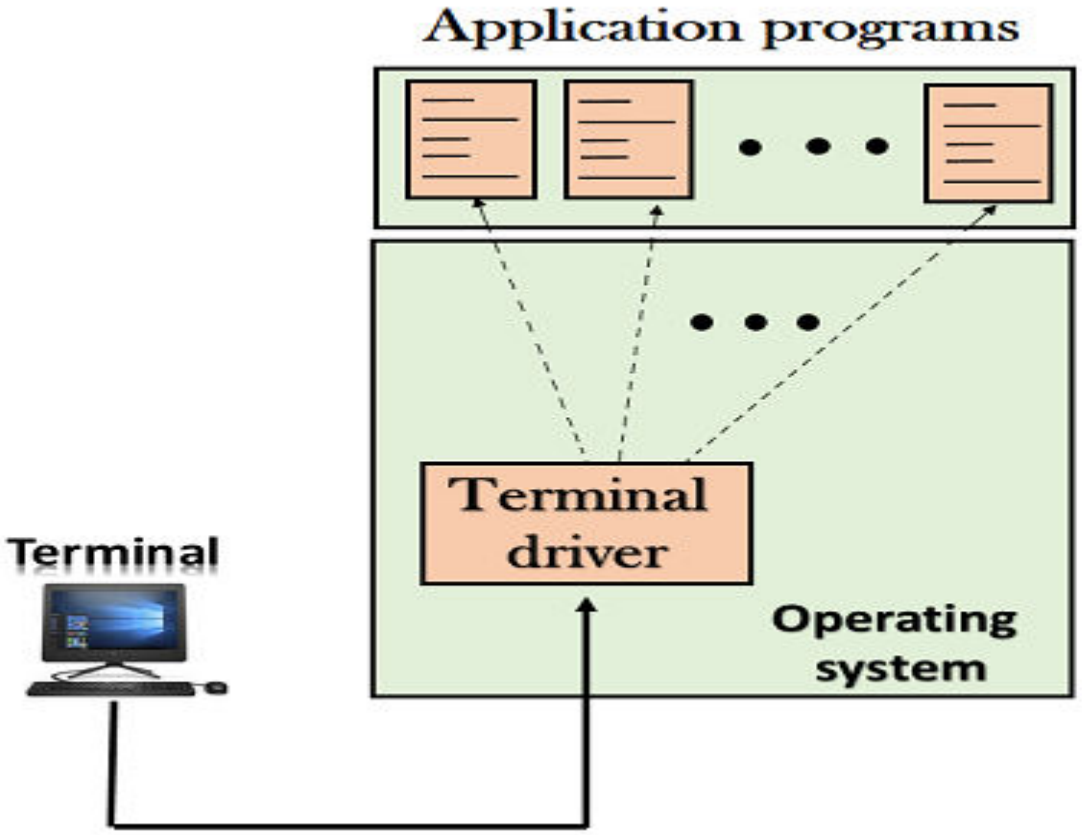
- It is a client/server application program.
- TELNET is an abbreviation for TErminaL NETwork.
- It is the standard TCP/IP protocol for virtual terminal service as proposed by the International Organization for Standards (ISO).
- TELNET enables the establishment of a connection to a remote system in such a way that the local terminal appears to be a terminal at the remote system.

# TELNET

## Types of login

There are two types of login: local login and remote login.

## Local login

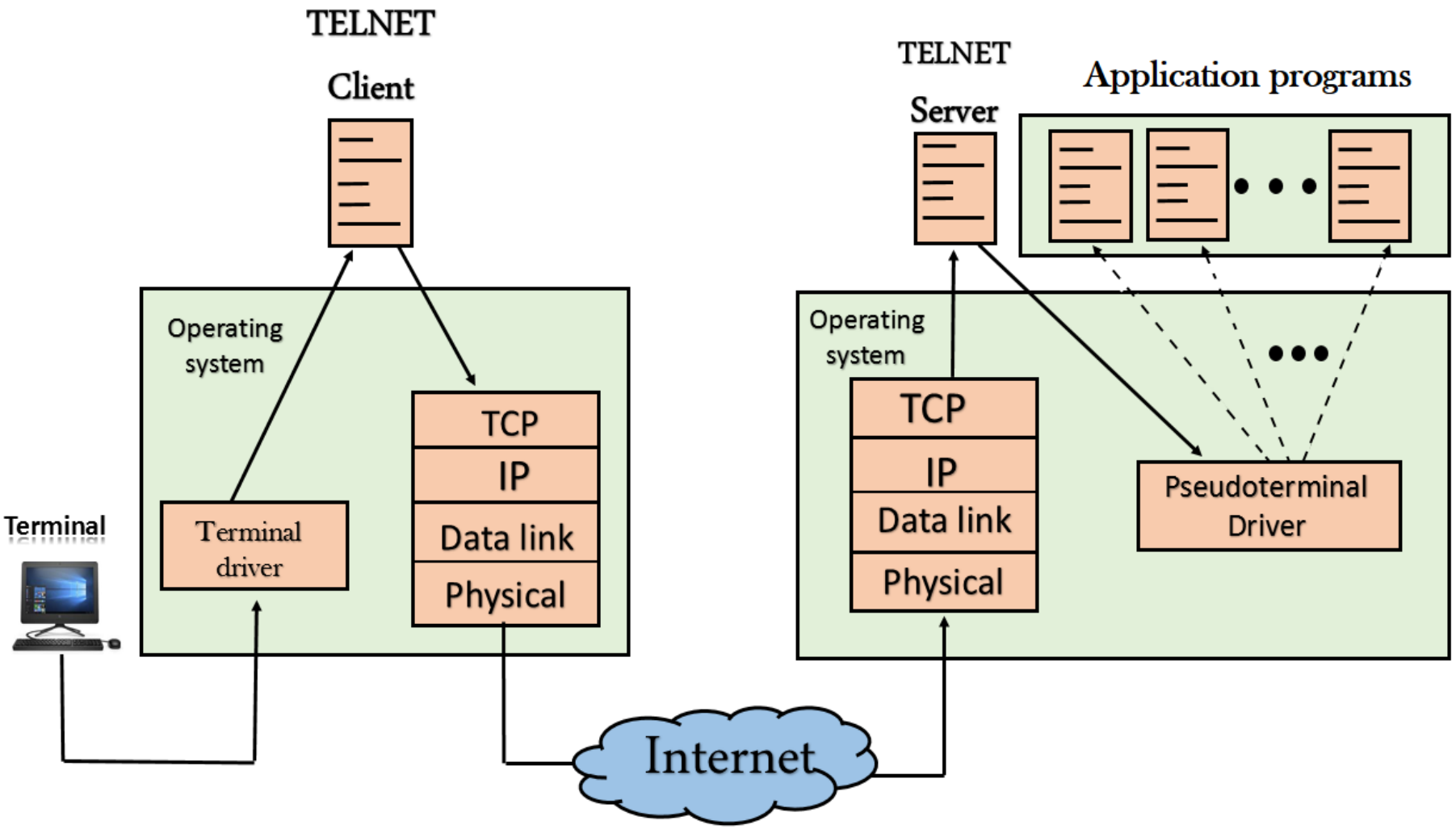


# Local Login(continue)

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

# Remote login

## Remote login



# Remote login(continue)

- When a user wants to access an application program or utility located on a remote machine, he performs remote login.
- Here, the TELNET client and server programs come into use.
- The user sends the keystrokes to the terminal driver, where the local operating system accepts the characters but does not interpret them. The characters are sent to the TELNET client, which transforms the characters to a universal character set called **network virtual terminal** (NVT) characters and delivers them to the local TCP/IP protocol stack.

# Remote login(continue)

- The commands or text, in NVT form, travel through the Internet and arrive at the TCP/IP stack at the remote machine.
- Here, the characters are delivered to the operating system and passed to the TELNET server, which changes the characters to the corresponding characters understandable by the remote computer. However, the characters cannot be passed directly to the operating system because the remote operating system is not designed to receive characters from a TELNET server: It is designed to receive characters from a terminal driver.
- The solution is to add a piece of software called a pseudoterminal driver which pretends that the characters are coming from a terminal. The operating system then passes the characters to the appropriate application program.

# Network Virtual Terminal(NVT)

- The mechanism to access a remote computer is complex. This is so because every computer and its operating system accept a special combination of characters as tokens.

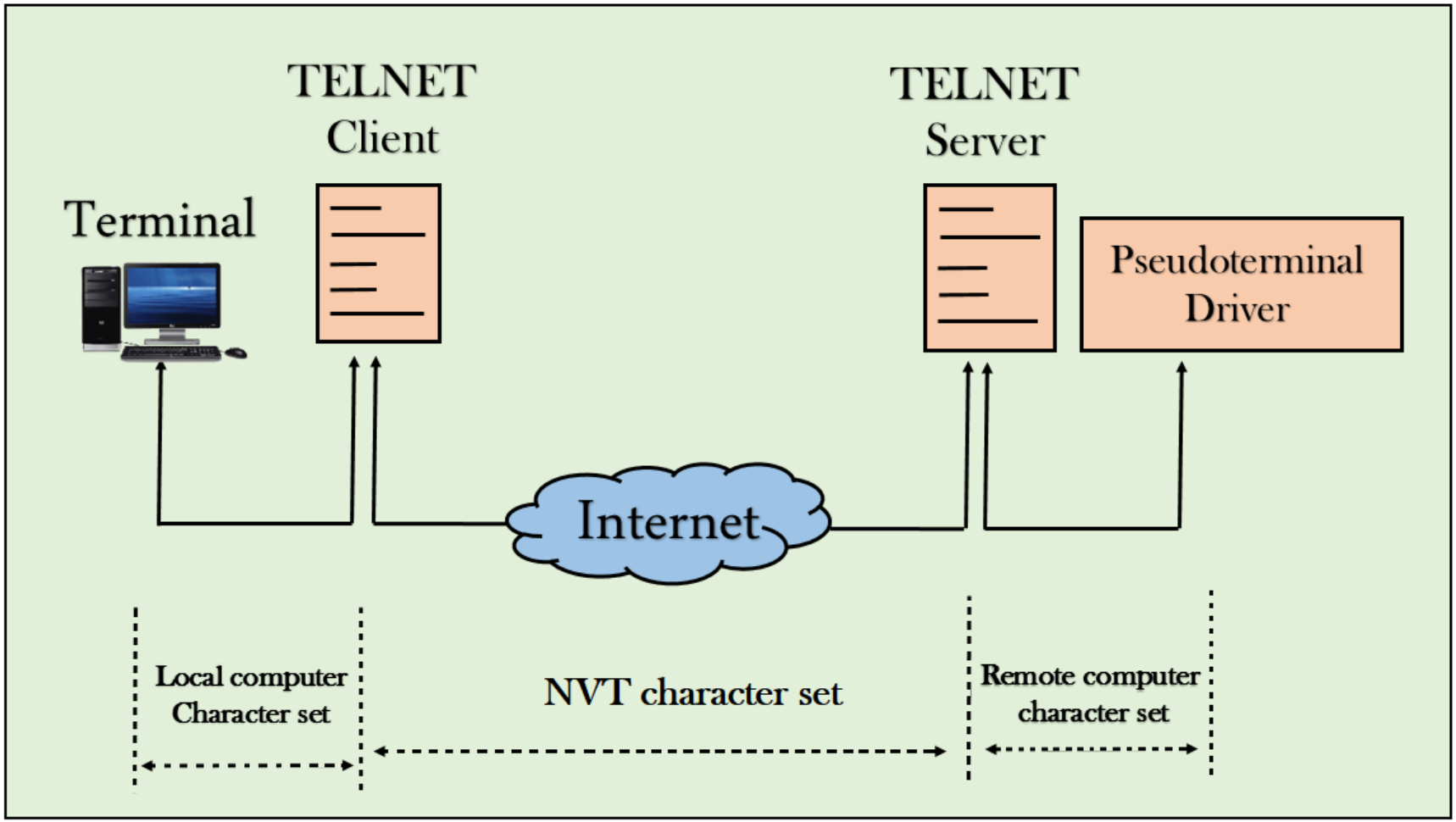
- For example, the end-of-file token in a computer running the DOS operating system is Ctrl+z, while the UNIX operating system recognizes Ctrl+d.

- TELNET solves this problem by defining a universal interface called the network virtual terminal (NVT) character set.

- Via this interface, the client TELNET translates characters (data or commands) that come from the local terminal into NVT form and delivers them to the network. The server TELNET, on the other hand, translates data and commands from NVT form into the form acceptable by the remote computer.

# Network Virtual Terminal(NVT)

It is illustrated in the following figure:-



# Network Virtual Terminal(NVT)

**NVT Character Set**

- NVT uses two sets of characters, one for data and the other for control. Both are 8-bit bytes.
- For data, NVT is an 8-bit character set in which the 7 lowest-order bits are the same as ASCII and the highest-order bit is 0.
- To send control characters between computers (from client to server or vice versa), NVT uses an 8-bit character set in which the highest-order bit is set to 1.

# Telnet

- TELNET uses only one TCP connection. The server uses the well-known port 23, and the client uses an ephemeral port.
- The same connection is used for sending both data and control characters.
- TELNET accomplishes this by embedding the control characters in the data stream. However, to distinguish data from control characters, each sequence of control characters is preceded by a special control character called ***interpret as control*** **(IAC).**

# ELECTRONIC MAIL

- E-mail is defined as the transmission of messages on the Internet.
- It is one of the most commonly used features over communications networks that may contain text, files, images, or other attachments.
- Email messages are conveyed through email servers; it uses multiple protocols within the TCP/IP suite.
- For example, SMTP is a protocol, stands for simple mail transfer protocol and used to send messages whereas other protocols IMAP or POP are used to retrieve messages from a mail server.

# ELECTRONIC MAIL

## 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.

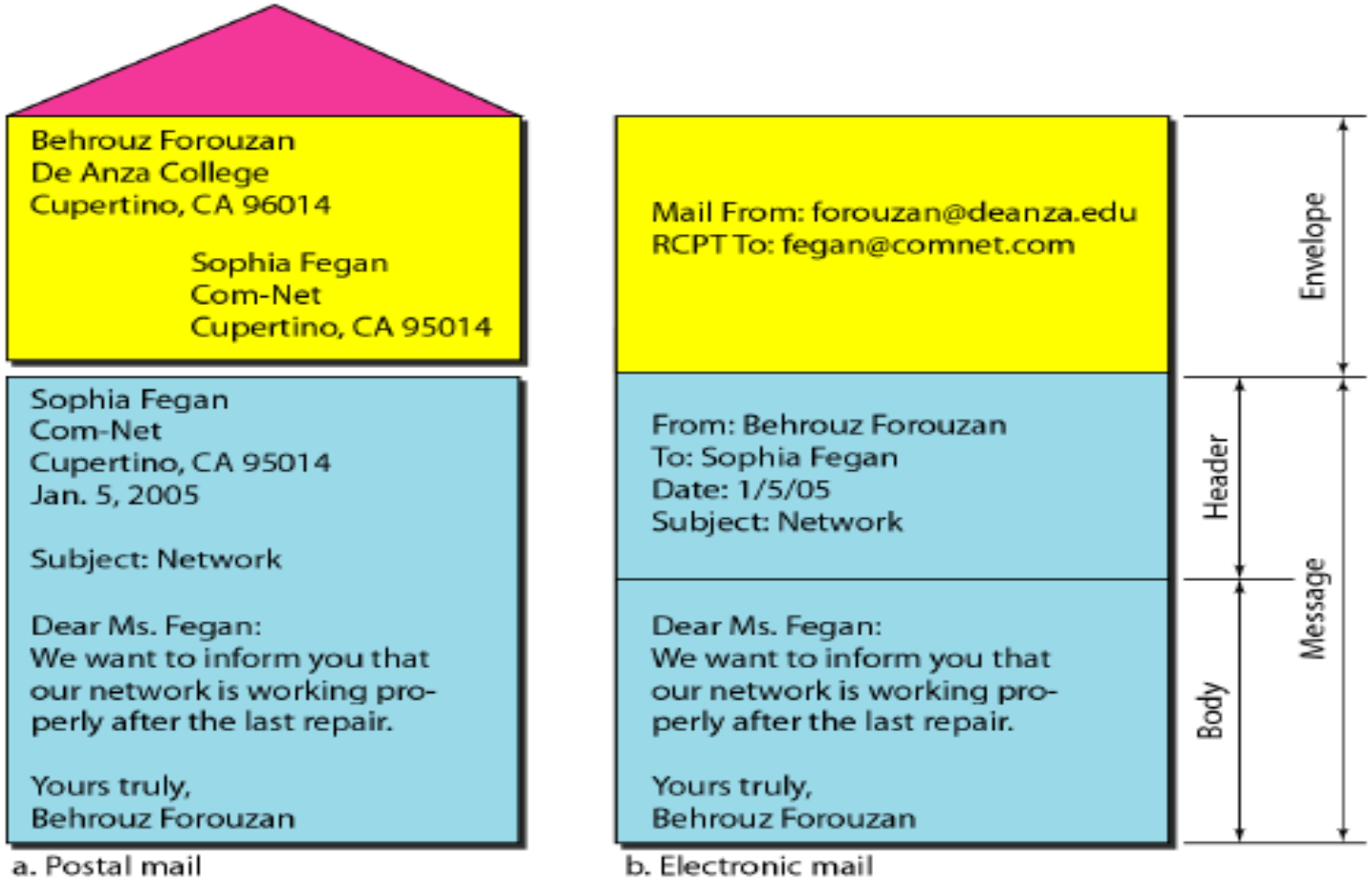
## User Agent Types

There are two types of user agents: **command-driven** and **GUI-based**

# ELECTRONIC 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.



# ELECTRONIC MAIL

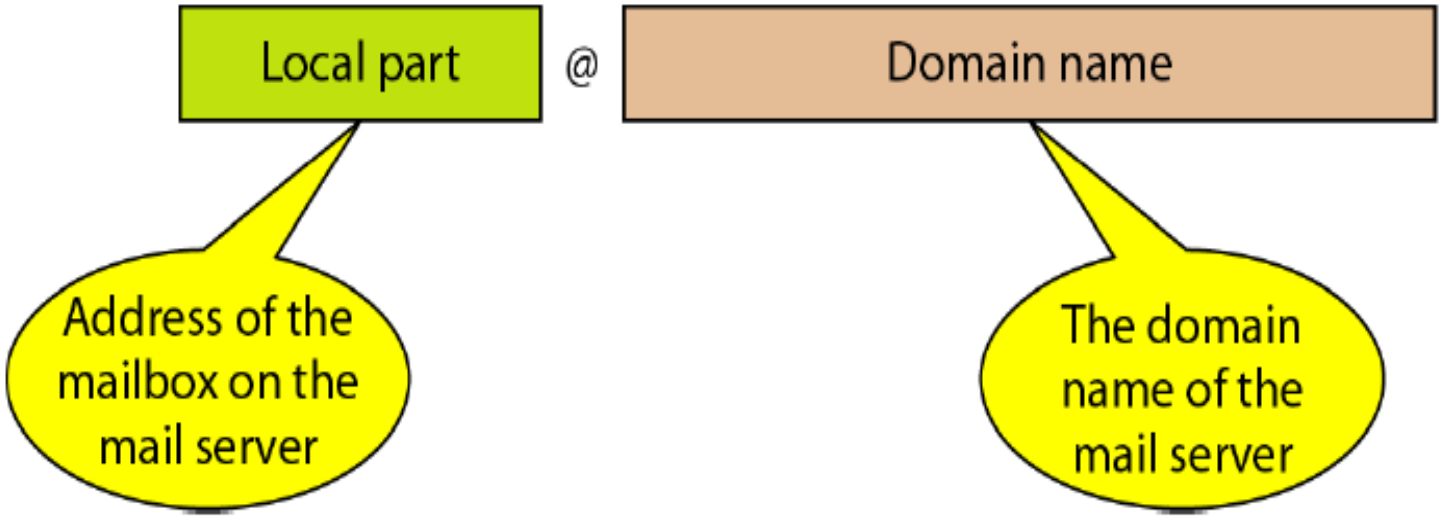
## Receiving Mail

- 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.

# ELECTRONIC 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: a local part and a domain name, separated by an @ sign.

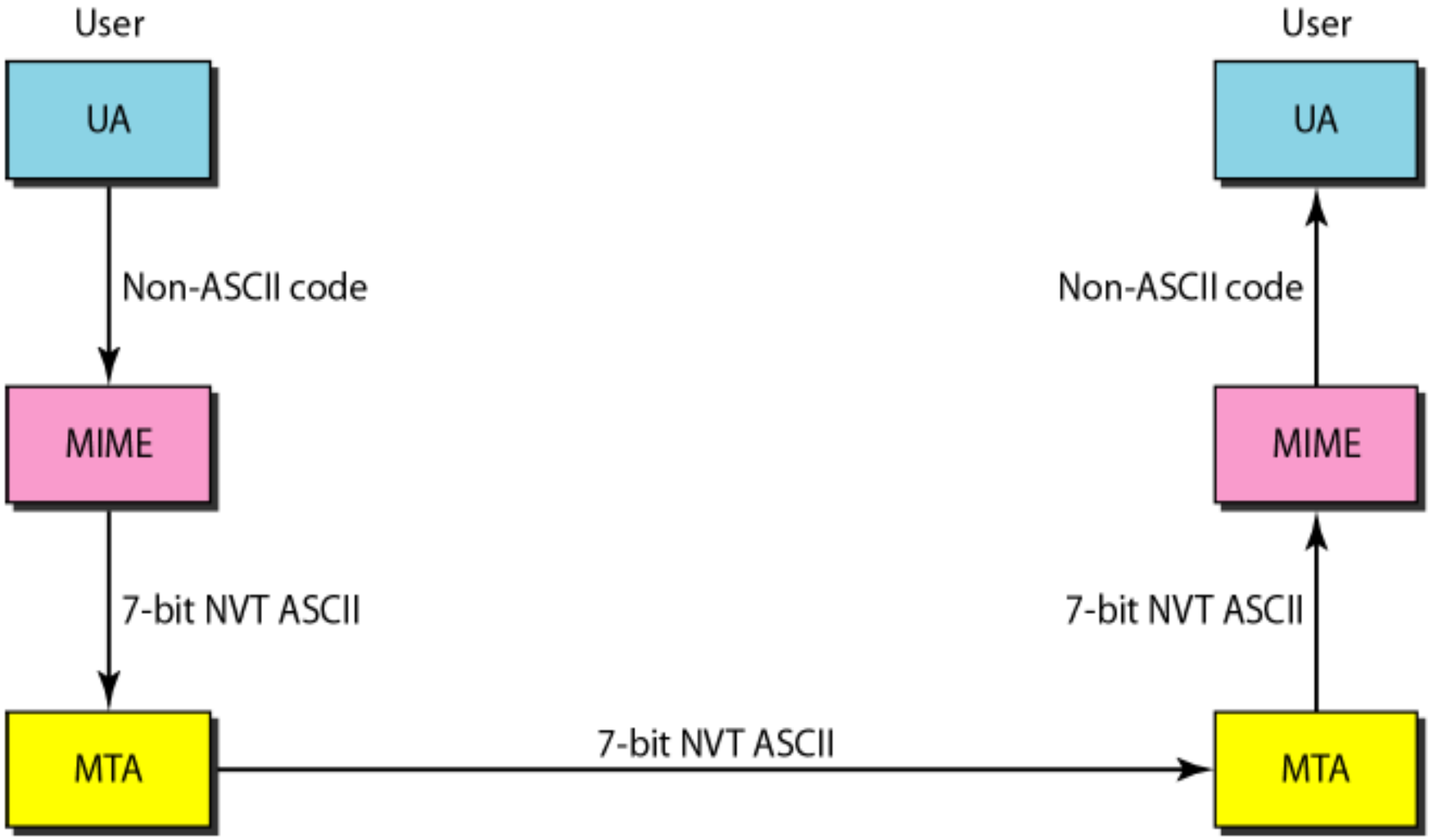


# MIME

- Multipurpose Internet Mail Extensions (MIME) is a supplementary protocol that allows non-ASCII data to be sent through e-mail.

- MIME transforms non-ASCII data at the sender site to NVT ASCII data and delivers them to the client MTA to be sent through the Internet. The message at the receiving side is transformed back to the original data.

# MIME


User
UA
Non-ASCII code
MIME
7-bit NVT ASCII
MTA
7-bit NVT ASCII
User
UA
Non-ASCII code
MIME
7-bit NVT ASCII
MTA

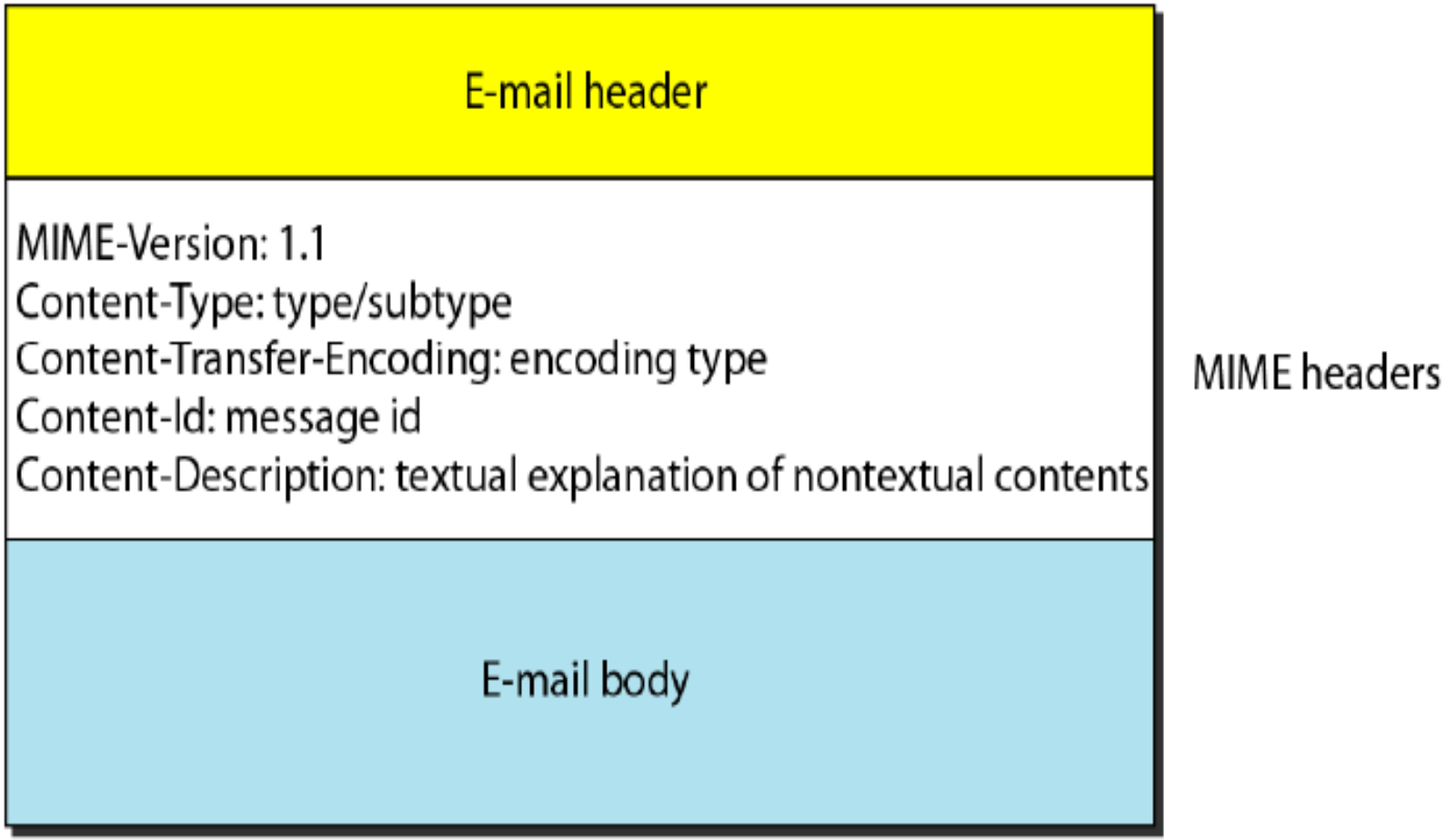
# MIME

MIME defines five headers that can be added to the original e-mail header section to define the transformation parameters:

1. MIME-Version
2. Content-Type
3. Content-Transfer-Encoding
4. Content-Id
5. Content-Description

# MIME

Following figure shows the MIME headers:-



# MIME

**MIME-Version**

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

**Content-Type**

This header defines the type of data used in the body of the message. The content type and the content subtype are separated by a slash. Depending on the subtype, the header may contain other parameters.

**Content-Type: <type/subtype; parameters> .**

MIME allows seven different types of data.

# MIME

## Content-type

| *Type* | *Subtype* | *Description* |
|---|---|---|
| Text | Plain | Unformatted |
| | HTML | HTML format (see Chapter 27) |
| Multipart | Mixed | Body contains ordered parts of different data types |
| | Parallel | Same as above, but no order |
| | Digest | Similar to mixed subtypes, but the default is message/ RFC822 |
| | Alternative | Parts are different versions of the same message |
| Message | RFC822 | Body is an encapsulated message |
| | Partial | Body is a fragment of a bigger message |
| | External-Body | Body is a reference to another message |
| Image | JPEG | Image is in JPEG format |
| | GIF | Image is in GIF format |
| Video | MPEG | Video is in MPEG format |
| Audio | Basic | Single-channel encoding of voice at 8 kHz |
| Application | PostScript | Adobe PostScript |
| | Octet-stream | General binary data (8-bit bytes) |

# MIME

## Content-Transfer-Encoding

This header defines the method used to encode the messages into 0's and 1's for transport:

**Content-Transfer-Encoding: <type>**

The five types of encoding methods are listed in the following table:-

| *Type* | *Description* |
|---|---|
| 7-bit | NVT ASCII characters and short lines |
| 8-bit | Non-ASCII characters and short lines |
| Binary | Non-ASCII characters with unlimited-length lines |
| Base-64 | 6-bit blocks of data encoded into 8-bit ASCII characters |
| Quoted-printable | Non-ASCII characters encoded as an equals sign followed by an ASCII code |

# MIME

**Content-Id**

This header uniquely identifies the whole message in a multiple-message environment.

**Content-Id: id=<content-id>**

**Content-Description**

This header defines whether the body is image, audio, or video.

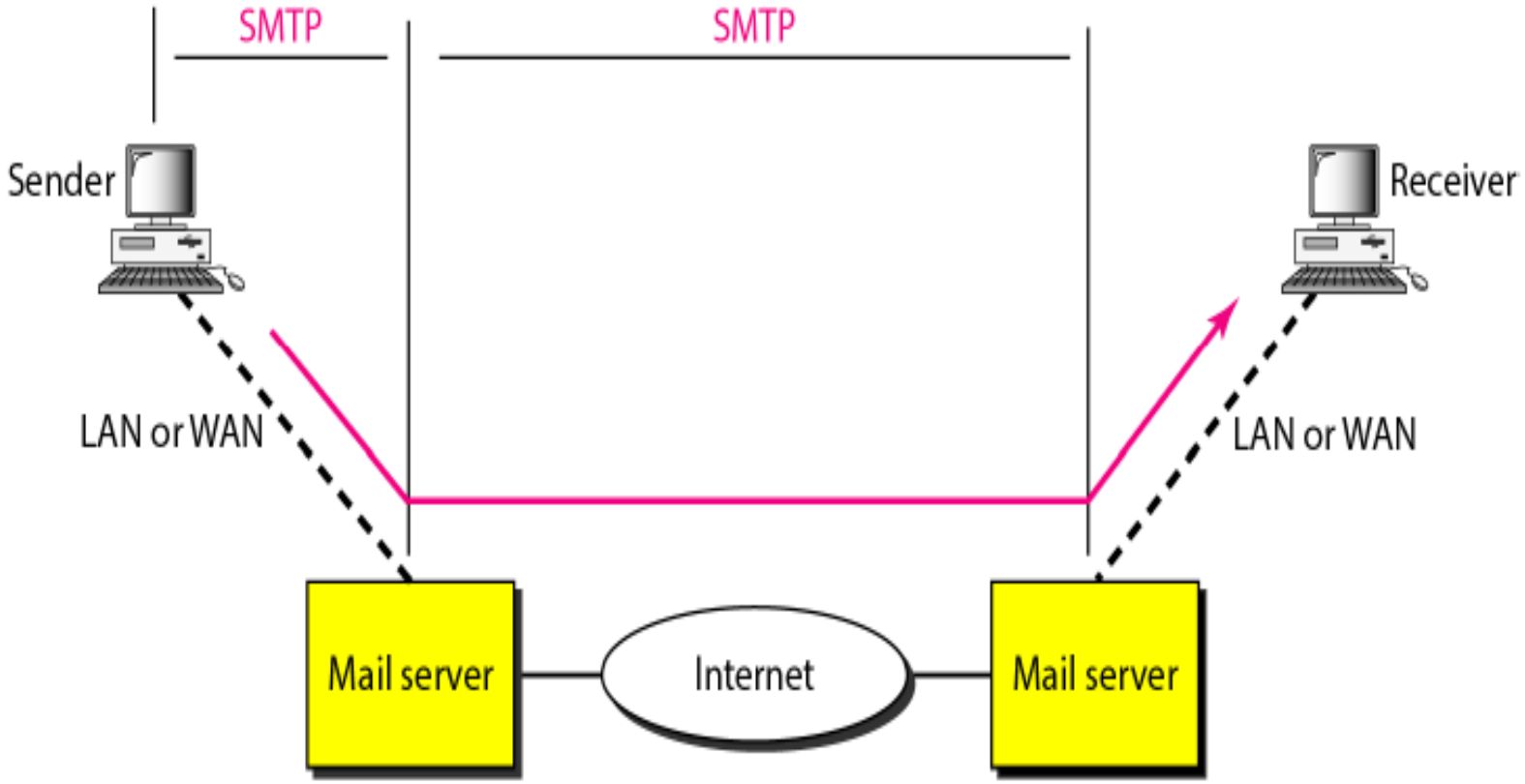
**Content-Description: <description>**

# Message Transfer Agent: SMTP

- SMTP is a message transfer agent.
- 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 protocol that defines the MTA client and server in the Internet is called the Simple Mail Transfer Protocol (SMTP).

# SMTP

Following figure shows the range of the SMTP protocol

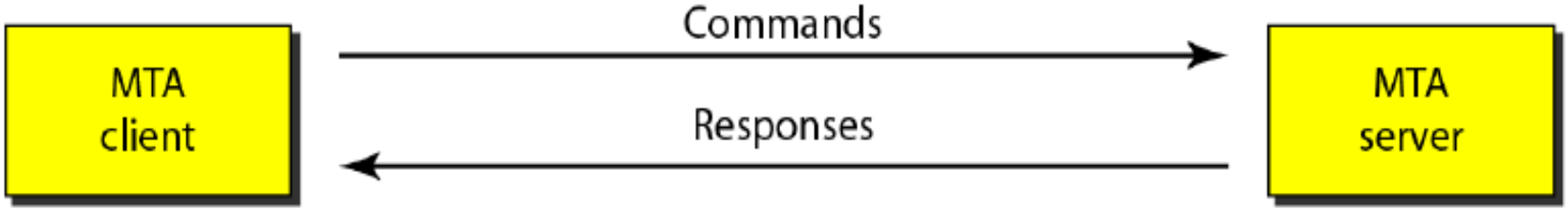


# SMTP

- SMTP is used two times, between the sender and the sender's mail server and between the two mail servers.

**Commands and Responses**

SMTP uses commands and responses to transfer messages between an MTA client and an MTA server.




- Each command or reply is terminated by a two-character (carriage return and line feed) end-of-line token.

# SMTP

## Commands

Commands are sent from the client to the server. The format of a command is shown in following figure:-

Keyword: argument(s)

- It consists of a keyword followed by zero or more arguments.
- SMTP defines 14 commands. The first five are mandatory; every implementation must support these five commands. The next three are often used and highly recommended. The last six are seldom used.

# SMTP

## Commands

| *Keyword* | *Argument(s)* |
|---|---|
| HELO | Sender's host name |
| MAIL FROM | Sender of the message |
| RCPT TO | Intended recipient of the message |
| DATA | Body of the mail |
| QUIT | |
| RSET | |
| VRFY | Name of recipient to be verified |
| NOOP | |
| TURN | |
| EXPN | Mailing list to be expanded |
| HELP | Command name |
| SEND FROM | Intended recipient of the message |
| SMOL FROM | Intended recipient of the message |
| SMAL FROM | Intended recipient of the message |

# SMTP

## Responses

Responses are sent from the server to the client. A response is a three digit code that may be followed by additional textual information. Following table lists some of the responses:-

| *Code* | *Description* |
|---|---|
| **Positive Completion Reply** | |
| **211** | System status or help reply |
| **214** | Help message |
| **220** | Service ready |
| **221** | Service closing transmission channel |
| **250** | Request command completed |
| **251** | User not local; the message will be forwarded |
| **Positive Intermediate Reply** | |
| **354** | Start mail input |
| **Transient Negative Completion Reply** | |
| **421** | Service not available |
| **450** | Mailbox not available |
| **451** | Command aborted: local error |
| **452** | Command aborted: insufficient storage |

# SMTP

## Responses(continue)

| *Code* | *Description* |
|---|---|
| **Permanent Negative Completion Reply** | |
| **500** | Syntax error; unrecognized command |
| **501** | Syntax error in parameters or arguments |
| **502** | Command not implemented |
| **503** | Bad sequence of commands |
| **504** | Command temporarily not implemented |
| **550** | Command is not executed; mailbox unavailable |
| **551** | User not local |
| **552** | Requested action aborted; exceeded storage location |
| **553** | Requested action not taken; mailbox name not allowed |
| **554** | Transaction failed |

- As the table shows, responses are divided into four categories. The leftmost digit of the code (2, 3, 4, and 5) defines the category.

# SMTP

## Mail Transfer Phases

The process of transferring a mail message occurs in three phases: connection establishment, mail transfer, and connection termination.

## Example

Let us see how we can directly use SMTP to send an e-mail and simulate the commands and responses we described in this section. We use TELNET to log into port 25 (well known port for SMTP). We then use the commands directly to send an e-mail. In this example, forouzanb@adelphia.net is sending an e-mail to himself. The first few lines show TELNET trying to connect to the Adelphia mail server.

After connection, we can type the SMTP commands and then receive the responses, as shown below. We have shown the commands in black and the responses in color. Note that we have added, for clarification, some comment lines, designated by the "=" signs. These lines are not part of the e-mail procedure.

# SMTP

```
$ telnet mail.adelphia.net 25
Trying 68.168.78.100 . . .
Connected to mail.adelphia.net (68.168.78.100).
```

```
================== Connection Establishment ================
  220 mta13.adelphia.net SMTP server ready Fri, 6 Aug 2004 . . .
HELO mail.adelphia.net
  250 mta13.adelphia.net
```

# SMTP

```
==================== Mail Transfer ================
MAIL FROM: forouzanb@adelphia.net
  250 Sender <forouzanb@adelphia.net> Ok
RCPT TO: forouzanb@adelphia.net
  250 Recipient <forouzanb@adelphia.net> Ok
DATA
  354 Ok Send data ending with <CRLF>.<CRLF>
From: Forouzan
TO: Forouzan

This is a test message
to show SMTP in action.
.
```

```
==================== Connection Termination ===============
    250 Message received: adelphia.net@mail.adelphia.net
QUIT
    221 mta13.adelphia.net SMTP server closing connection
Connection closed by foreign host.
```

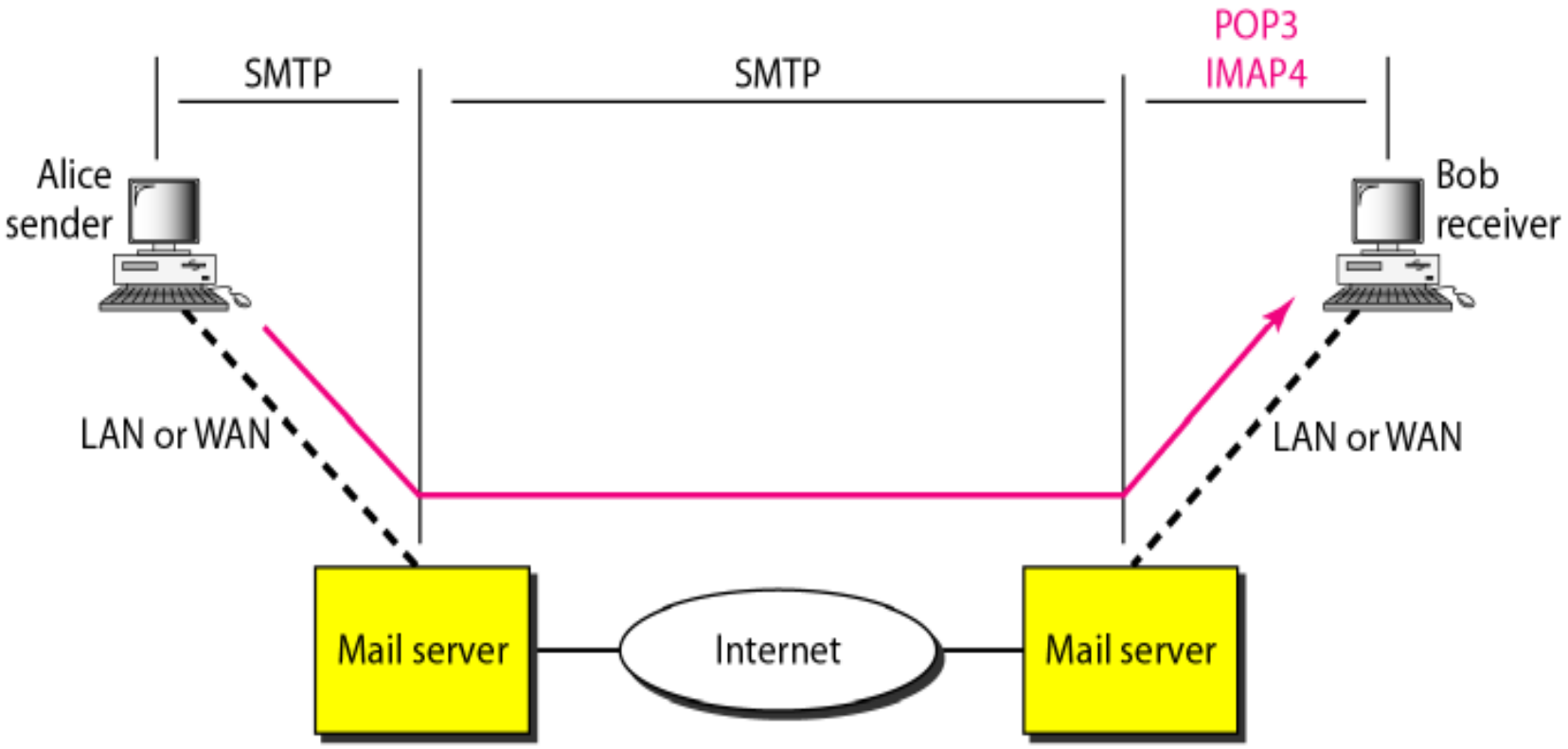
# Message Access Agent: POP and IMAP

- The first and the second stages of mail delivery use SMTP. However, SMTP is not involved in the third stage because SMTP is a push protocol; it pushes the message from the client to the server. In other words, the direction of the bulk: data (messages) is from the client to the server.

- On the other hand, the third stage needs a pull protocol; the client must pull messages from the server. The direction of the bulk data is from the server to the client. The third stage uses a message access agent.

# Message Access Agent: POP and IMAP

Currently two message access protocols are available: Post Office Protocol, version 3 (POP3) and Internet Mail Access Protocol, version 4 (IMAP4). Following figure shows the position of these two protocols in the most common situation.



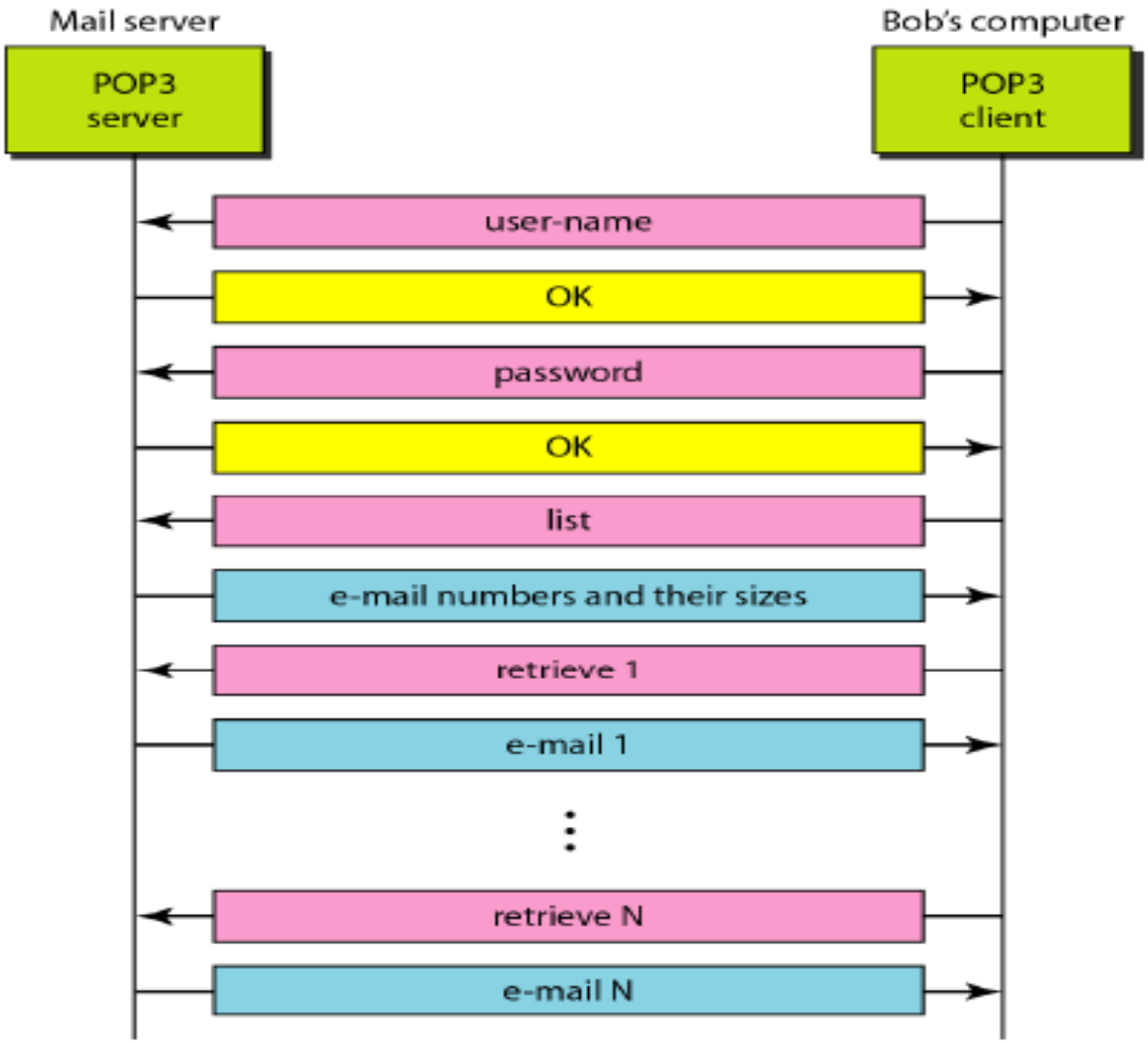
# POP3

- Post Office Protocol, version 3 (POP3) is simple and limited in functionality. The client POP3 software is installed on the recipient computer; the server POP3 software is installed on the mail server.

- Mail access starts with the client when the user needs to download e-mail from the mailbox on the mail server. The client opens a connection to the server on TCP port 110. It then sends its user name and password to access the mailbox. The user can then list and retrieve the mail messages, one by one.

# POP3

Following figure shows an example of downloading using POP3



# POP3

- POP3 has two modes: the delete mode and the keep mode.
- In the delete mode, the mail is deleted from the mailbox after each retrieval.
- In the keep mode, the mail remains in the mailbox after retrieval.
- The delete mode is normally used when the user is working at her permanent computer and can save and organize the received mail after reading or replying.
- The keep mode is normally used when the user accesses her mail away from her primary computer (e.g., a laptop). The mail is read but kept in the system for later retrieval and organizing.

# POP3

**Deficiencies:**

- It does not allow the user to organize her mail on the server.
- The user cannot have different folders on the server.
- In addition, POP3 does not allow the user to partially check the contents of the mail before downloading.

# IMAP4

- Another mail access protocol is Internet Mail Access Protocol, version 4 (IMAP4). IMAP4 is similar to POP3, but it has more features. IMAP4 is more powerful and more complex.

- IMAP4 provides the following extra functions:
  - A user can check the e-mail header prior to downloading.
  - A user can search the contents of the e-mail for a specific string of characters prior to downloading.

# IMAP4

- A user can partially download e-mail. This is especially useful if bandwidth is limited and the e-mail contains multimedia with high bandwidth requirements.
- A user can create, delete, or rename mailboxes on the mail server.
- A user can create a hierarchy of mailboxes in a folder for e-mail storage.

# Web-Based Mail

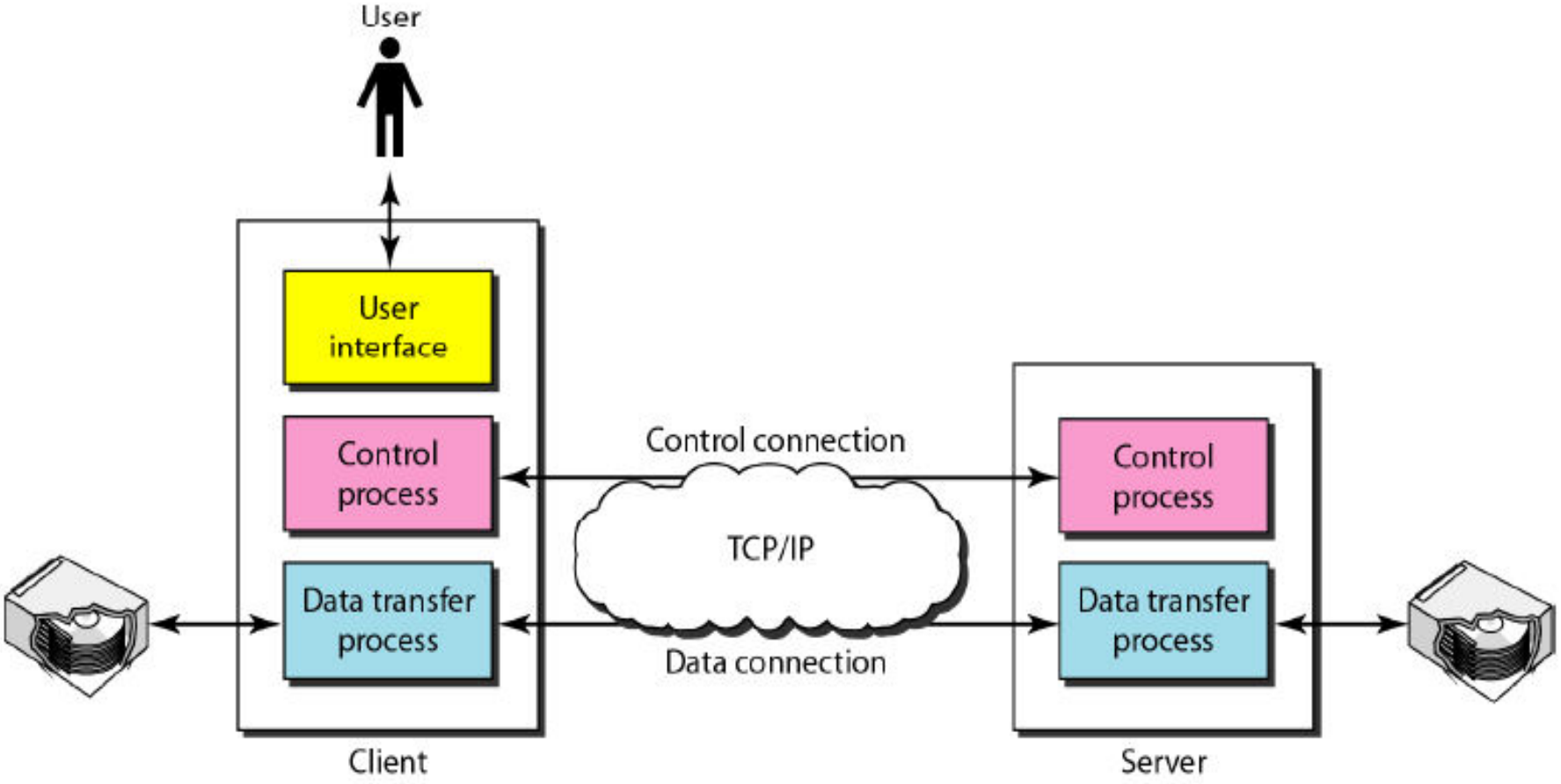
- E-mail is such a common application that some websites today provide this service to anyone who accesses the site.
- Suppose the mail is transferred from Alice to Bob.
  - Mail transfer from Alice's browser to her mail server is done through HTTP.
  - The transfer of the message from the sending mail server to the receiving mail server is still through SMTP.
  - Finally, the message from the receiving server (the Web server) to Bob's browser is done through HTTP.
- In the last phase, instead of POP3 or IMAP4, HTTP is normally used.

# 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.
- FTP differs from other client/server applications in that it 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.
- The control connection uses very simple rules of communication. We need to transfer only a line of command or a line of response at a time.
- The data connection, on the other hand, needs more complex rules due to the variety of data types transferred.
- However, the difference in complexity is at the FTP level, not TCP.
- For TCP, both connections are treated the same.

# File Transfer Protocol (FTP)

- FTP uses two well-known TCP ports: Port **21** is used for the control connection, and port **20** is used for the data connection.
- Following figure shows the basic model of FTP:-



# File Transfer Protocol (FTP)

- 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 connection is made between the control processes. The data connection is made between the data transfer processes.

# File Transfer Protocol (FTP)

- 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.
- In other words, when a user starts an FTP session, the control connection opens. While the control connection is open, the data connection can be opened and closed multiple times if several files are transferred.

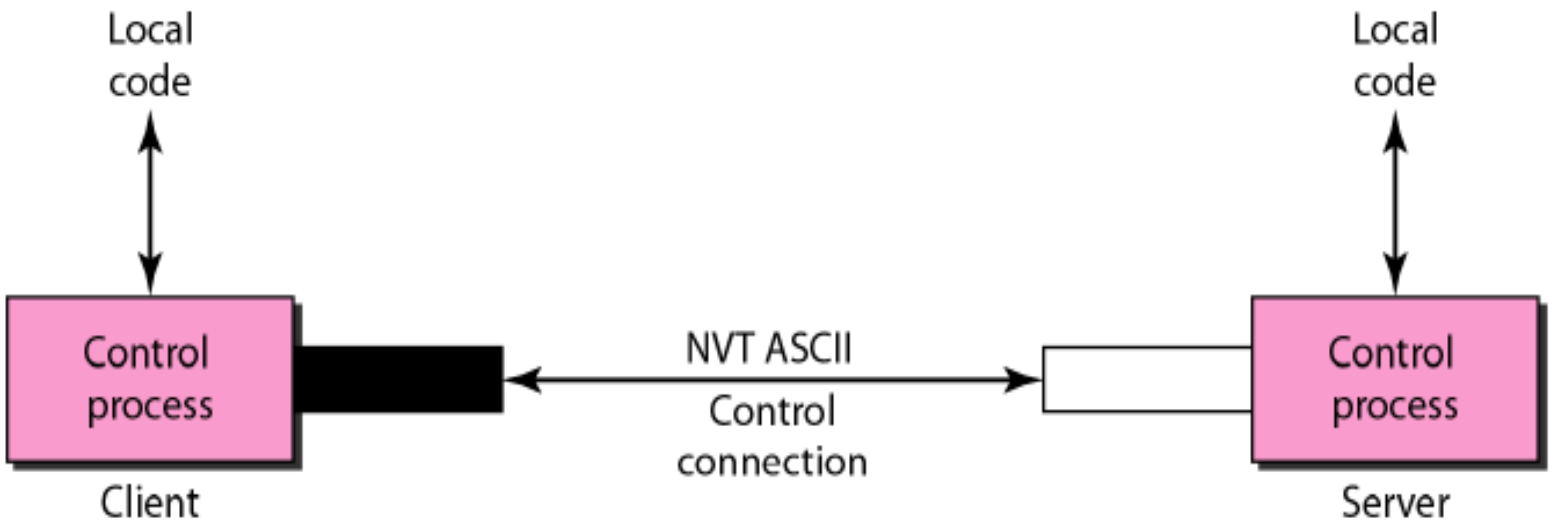
# File Transfer Protocol (FTP)

## Communication over Control Connection

- 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.

# File Transfer Protocol (FTP)

- 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.



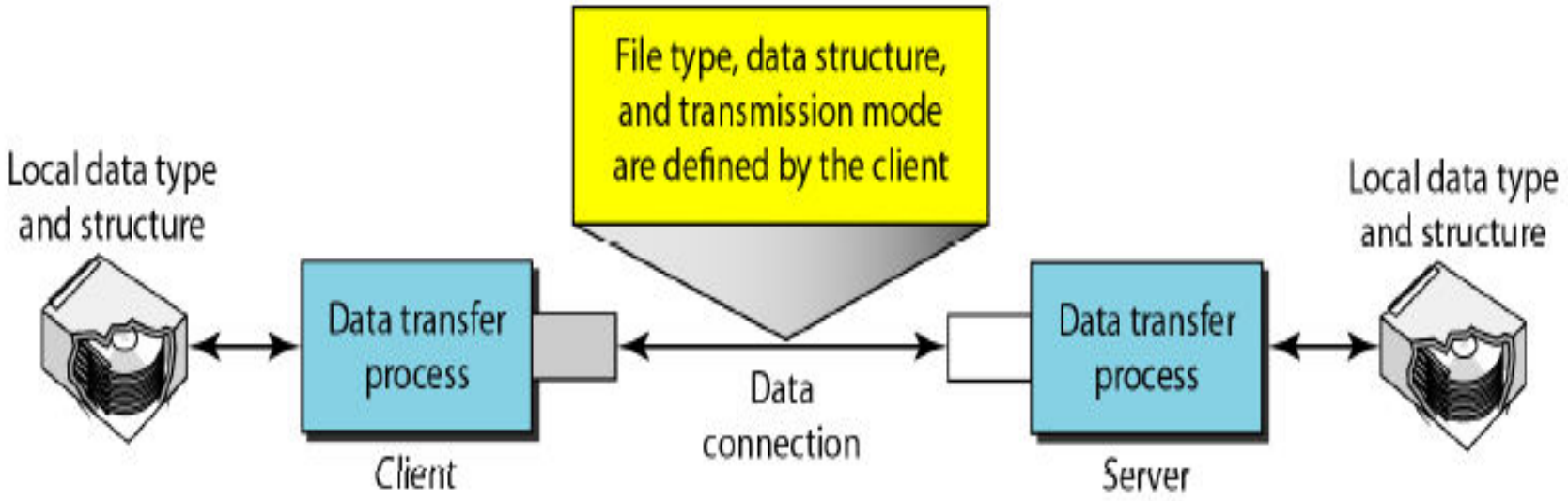
# File Transfer Protocol (FTP)

## Communication over 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:
  - A file is to be copied from the server to the client. This is called retrieving a file. It is done under the supervision of the **RETR** command.
  - A file is to be copied from the client to the server. This is called storing a file. It is done under the supervision of the **STOR** command.
  - 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.

# File Transfer Protocol (FTP)

- The client must define the type of file to be transferred, the structure of the data, and the transmission mode.



# File Transfer Protocol (FTP)

**File Type**

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. Each character is encoded using 7-bit ASCII. 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. This is mostly used to transfer binary files such as compiled programs.

# File Transfer Protocol (FTP)

## Data Structure

FTP can transfer a file across the data connection by using one of the following interpretations about the structure of the data: file structure, record structure, and 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. The pages can be stored and accessed randomly or sequentially.

# File Transfer Protocol (FTP)

## Transmission Mode

FTP can transfer a file across the data connection by using one of the following three transmission modes: stream mode, block mode, and compressed mode.

- The **stream mode** is the default 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. If the data are divided into records (record structure), each record will have a 1-byte end-of-record (EOR) character and the end of the file will have a 1-byte end-of-file (EOF) character.

# File Transfer Protocol (FTP)

## Transmission Mode(continue)

- 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. In this method, consecutive appearances of a data unit are replaced by one occurrence and the number of repetitions.

# File Transfer Protocol (FTP)

**Example**

The following shows an actual FTP session for retrieving a list of items in a directory. The colored lines show the responses from the server control connection; the black lines show the commands sent by the client. The lines in white with a black background show data transfer.

# File Transfer Protocol (FTP)

```
$ ftp voyager.deanza.fhda.edu
Connected to voyager.deanza.fhda.edu.
220 (vsFTPd 1.2.1)
530 Please login with USER and PASS.
Name (voyager.deanza.fhda.edu:forouzan): forouzan
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls reports
227 Entering Passive Mode (153,18,17,11,238,169)
150 Here comes the directory listing.
drwxr-xr-x    2 3027     411          4096 Sep 24  2002 business
drwxr-xr-x    2 3027     411          4096 Sep 24  2002 personal
drwxr-xr-x    2 3027     411          4096 Sep 24  2002 school
226 Directory send OK.
ftp> quit
221 Goodbye.
```

# File Transfer Protocol (FTP)

1. After the control connection is created, the FIP server sends the 220 (service ready) response on the control connection.

2. The client sends its name.

3. The server responds with 331 (user name is OK, password is required).

4. The client sends the password (not shown).

5. The server responds with 230 (user log-in is OK).

6. The client sends the list command (ls reports) to find the list of files on the directory named reports.

# File Transfer Protocol (FTP)

7. Now, the server responds with 150 and opens the data connection.

8. The server then sends the list of the files or directories (as a file) on the data connection. When the whole list (file) is sent, the server responds with 226 (closing data connection) over the control connection.

9. The client now has two choices. It can use the QUIT command to request the closing of the control connection, or it can send another command to start another activity (and eventually open another data connection). In our example, the client sends a QUIT command.

10. After receiving the QUIT command, the server responds with 221 (service closing) and then closes the control connection.

# File Transfer Protocol (FTP)

# AKTU Examination Questions

1. Mention the use of HTTP.
2. List out few email gateways.
3. Elaborate about TELNET and its working procedure.
4. How does FTP work? Differentiate between passive and active FTP.
5. Explain the SNMP protocols in detail.
6. How is TFTP different from FTP?
7. What three functions can SNMP perform to manage network devices?
8. How is the BOOTP different from DHCP?
9. What is the purpose of the Domain Name System? Discuss the three main divisions of the domain name space.
10. Write short notes on any two: (i) SMTP (ii) TELNET (iii) HTTP

# AKTU Examination Questions

11. Elaborate about TELNET and its working procedure.
12. Write short notes on any two of the following:
    i. DNS in the internet
    ii. Voice Over IP
    iii. File Transfer Protocol
13. Explain the SNMP protocols in detail.
14. What do you mean by DNS?
15. The symbols & their frequencies are given below

| **Symbol** | A | B | C | D | E | F | G | H |
|---|---|---|---|---|---|---|---|---|
| **Frequency** | 20 | 28 | 16 | 15 | 15 | 10 | 4 | 2 |

Construct Huffman codes.

# AKTU Examination Questions

16. Encrypt “EXTRANETPLANETSOURCE” using a transposition cipher with the following key: 3 5 2 1 4

17. Explain the following:

(i) Telnet

(ii) FTP

(iii) SNMP

(iv) HTTP

(v) MIME

18. How does DNS perform data name resolution? What are the different types of name servers? Mention the DNS message format for query and reply messages.

# AKTU Examination Questions

19. Write short notes on any three of the following:

(i) DNS in the Internet

(ii) Voice Over IP

(iii) SNMP

(iv) Electronic mail

(v) File Transfer Protocol