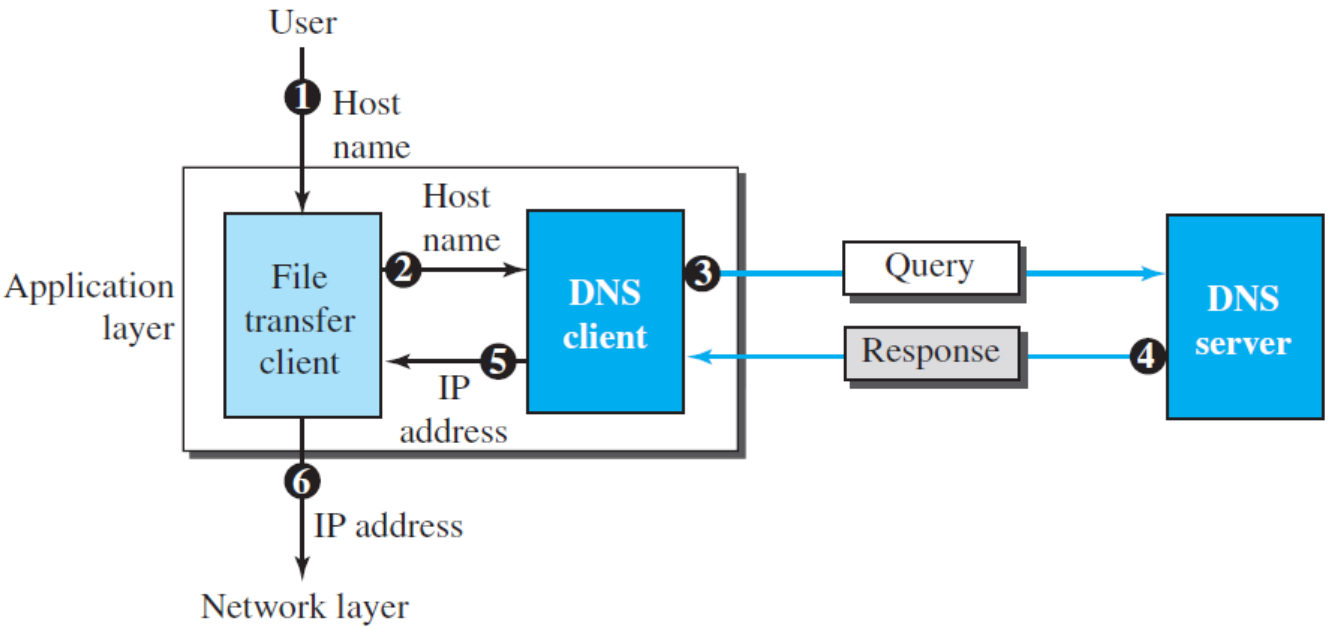
Module 5
Application Layer

Domain Name System

Figure 25.1 *Example of using the DNS service*

Figure 26.28 *Purpose of DNS*



The Domain Name System (DNS) used in the Internet

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

1. The user passes the host name to the file transfer client.

2. The file transfer client passes the host name to the DNS client.

3. Each computer, after being booted, knows the address of one DNS server. The
DNS client sends a message to a DNS server with a query that gives the file transfer
server name using the known IP address of the DNS server.

4. The DNS server responds with the IP address of the desired file transfer server.

5. The DNS server passes the IP address to the file transfer client.

6. The file transfer client now uses the received IP address to access the file transfer
server.

25-1 NAME SPACE

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

Topics discussed in this section:

Flat Name Space

Hierarchical Name Space

Flat Name Space (like giving everyone just a first name)

- In a *flat name space*, each device or resource is given one simple name, like “Alex”. There is no structure in the name—just a word.
- Problem:
  If the system is big (like the entire Internet), many people will want the same name.
  To avoid duplicates, one central boss must keep track of all names.
  This becomes difficult and inefficient.

Hierarchical Name Space (like first name + last name + department)

- In a *hierarchical name space*, a name has several parts, like:
  department → organization → category
- This is similar to a full name such as Alex Johnson from Sales Department.
- Because the name has layers, it becomes unique even if the first part is the same.
- How it works:
  - A central authority assigns top-level parts (like .com, .edu, or company names such as first.com and second.com).
  - Each company can then create its own sub-names inside its space without worrying about conflicts.
  - For example:
    - Company A is called first.com
    - Company B is called second.com
    - Both companies name one computer caesar
  - Thanks to hierarchy, the full names become:
    - caesar.first.com
    - caesar.second.com

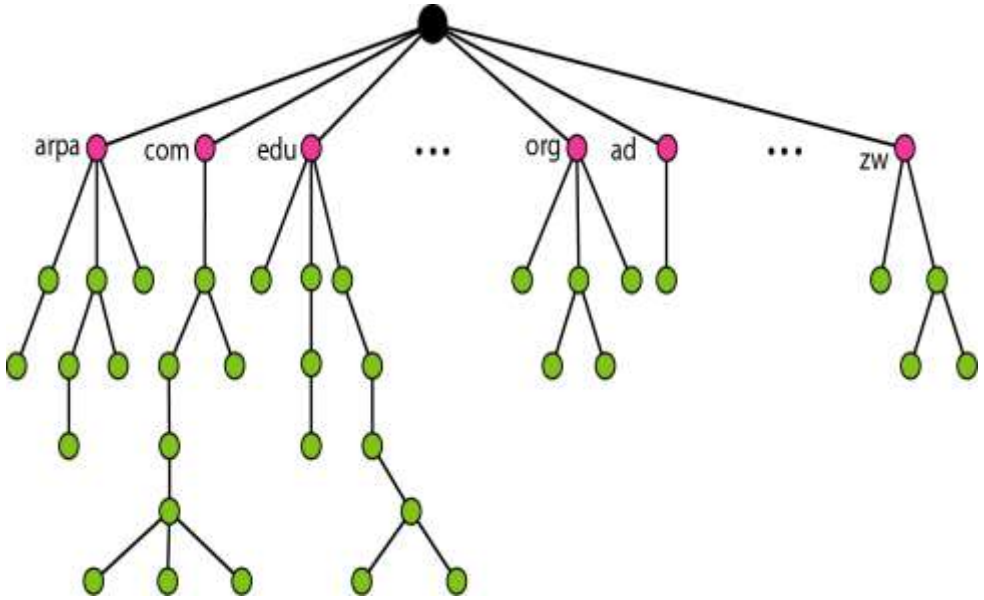
25-2 DOMAIN NAME SPACE

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.

Topics discussed in this section:

Label
Domain Name
Domain

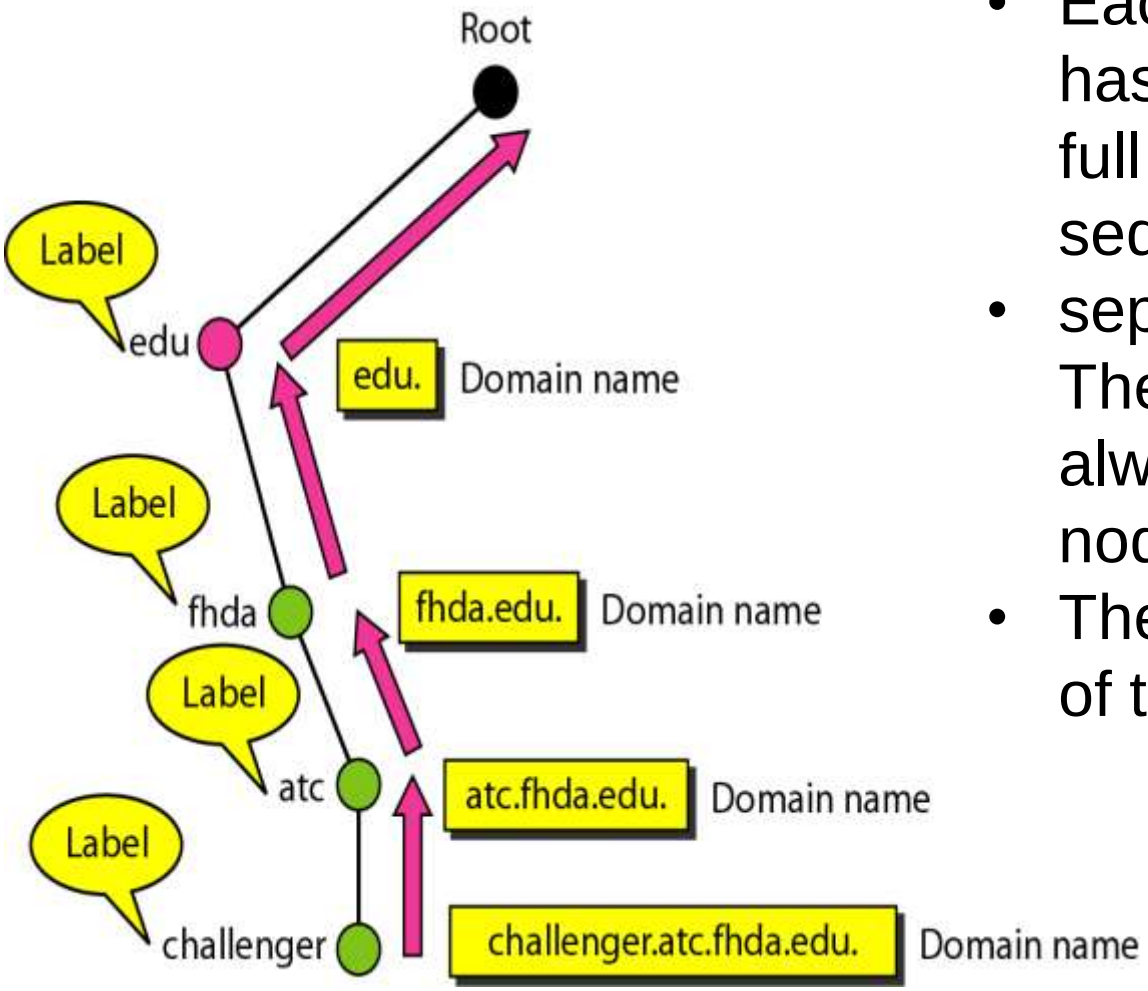
Figure 25.2 *Domain name space*



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).

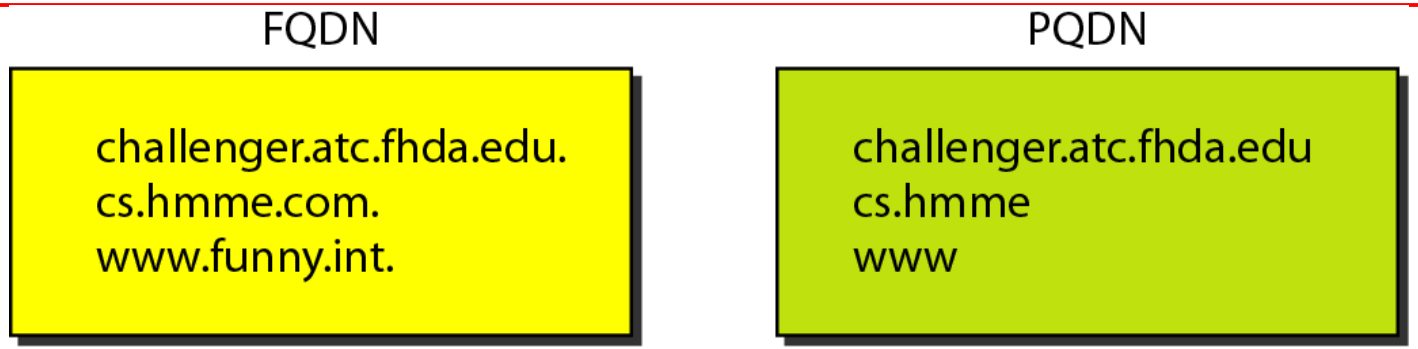
Figure 25.3 *Domain names and labels*



Domain Name

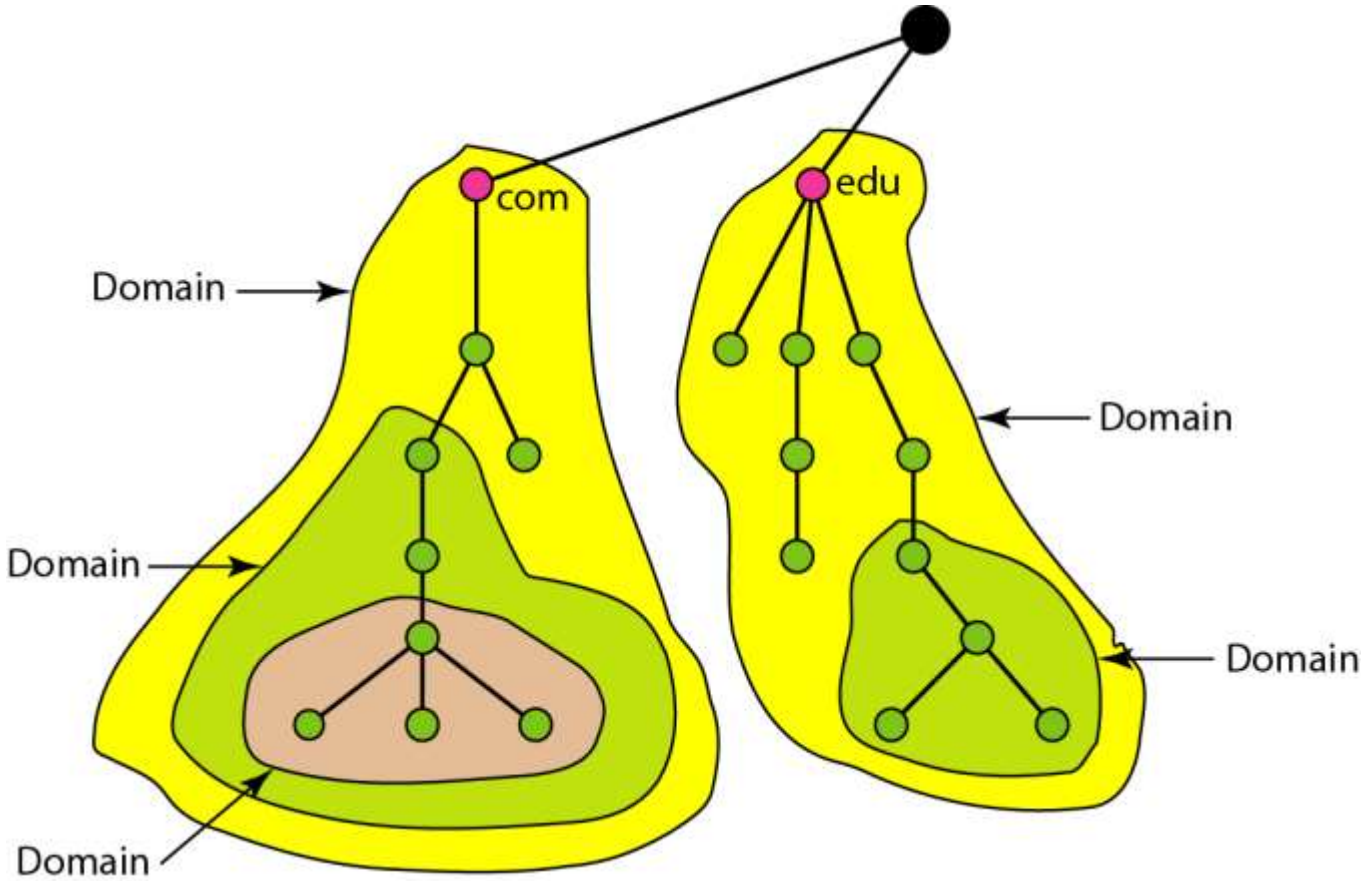
- Each node in the tree has a domain name. A full **domain name** is a sequence of labels
- separated by dots (**.**). The domain names are always read from the node up to the root.
- The last label is the label of the root (null).

Figure 25.4 *FQDN and PQDN*



- 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.
- 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.

Figure 25.5 *Domains*



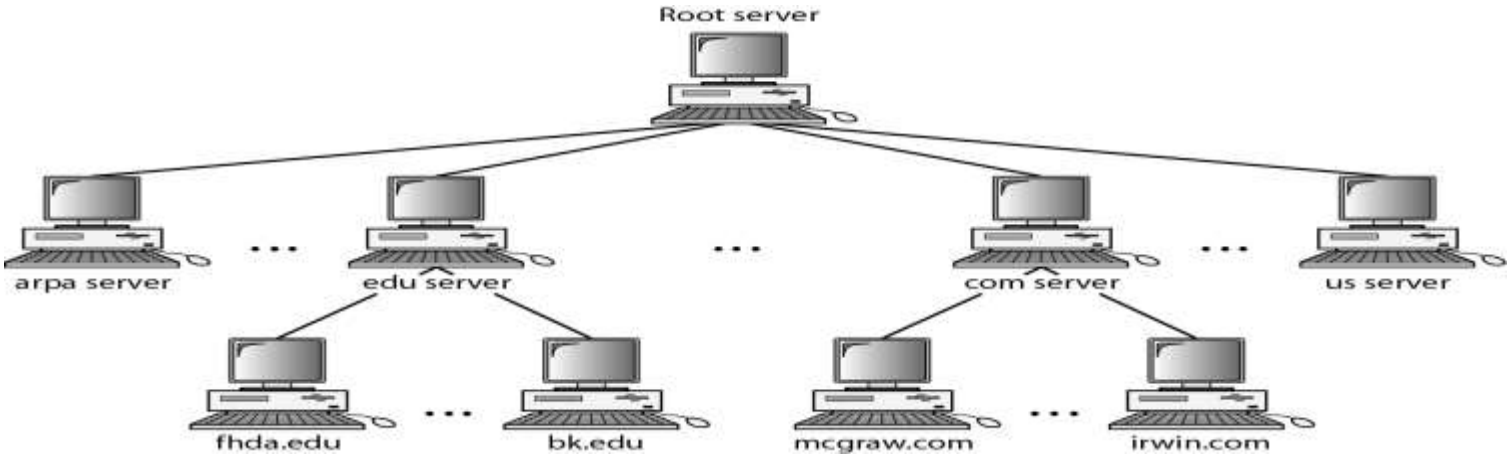
25-3 DISTRIBUTION OF NAME SPACE

The information contained in the domain name space must be stored. However, it is very inefficient and also unreliable to have just one computer store such a huge amount of information. In this section, we discuss the distribution of the domain name space.

Topics discussed in this section:

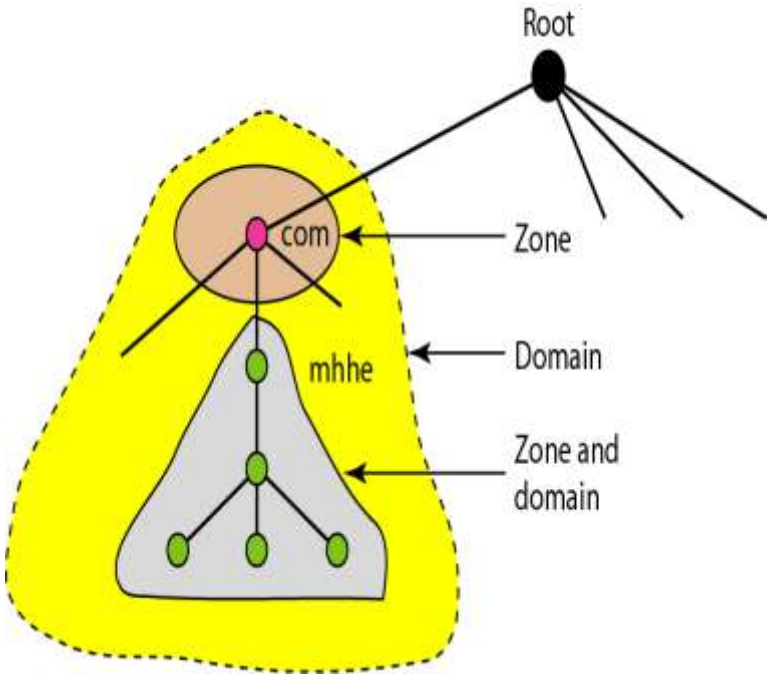
Hierarchy of Name Servers
Zone
Root Server
Primary and Secondary Servers

Figure 25.6 *Hierarchy of name servers*



- To address scalability and management issues in naming systems, the Domain Name System (DNS) distributes naming information across multiple computers known as DNS servers.
- This distribution is achieved through a hierarchical structure.The DNS namespace is organized as a tree, with the root at the top level. Directly under the root are the top-level domains (TLDs) such as .com, .org, .edu, and country codes.
- a domain at this level may still contain a vast number of names, DNS allows further subdivision into subdomains. For example, within example.com, subdomains such as sales.example.com or research.example.com may be created.
- This process of subdivision can continue to multiple levels, forming a hierarchical structure of domains and subdomains.

Figure 25.7 *Zones and domains*



Zone:

- **Because the entire DNS hierarchy is too large for one server, it is split among many servers.**
- **Each server is responsible for a zone, which is a continuous part of the DNS tree.**
- **If a server manages a domain and does not divide it into subdomains, then the domain = zone.**
- **If the server does divide its domain into subdomains and gives control of those subdomains to other servers, then the domain ≠ zone.**
- **The main server keeps only basic references, while the lower-level servers store the detailed information.**

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

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.

25-4 DNS IN THE INTERNET

In the Internet, the domain name space (tree) is divided into three different sections: generic domains, country domains, and the inverse domain.

Topics discussed in this section:

Generic Domains
Country Domains
Inverse Domain

Figure 25.8 *DNS IN THE INTERNET*

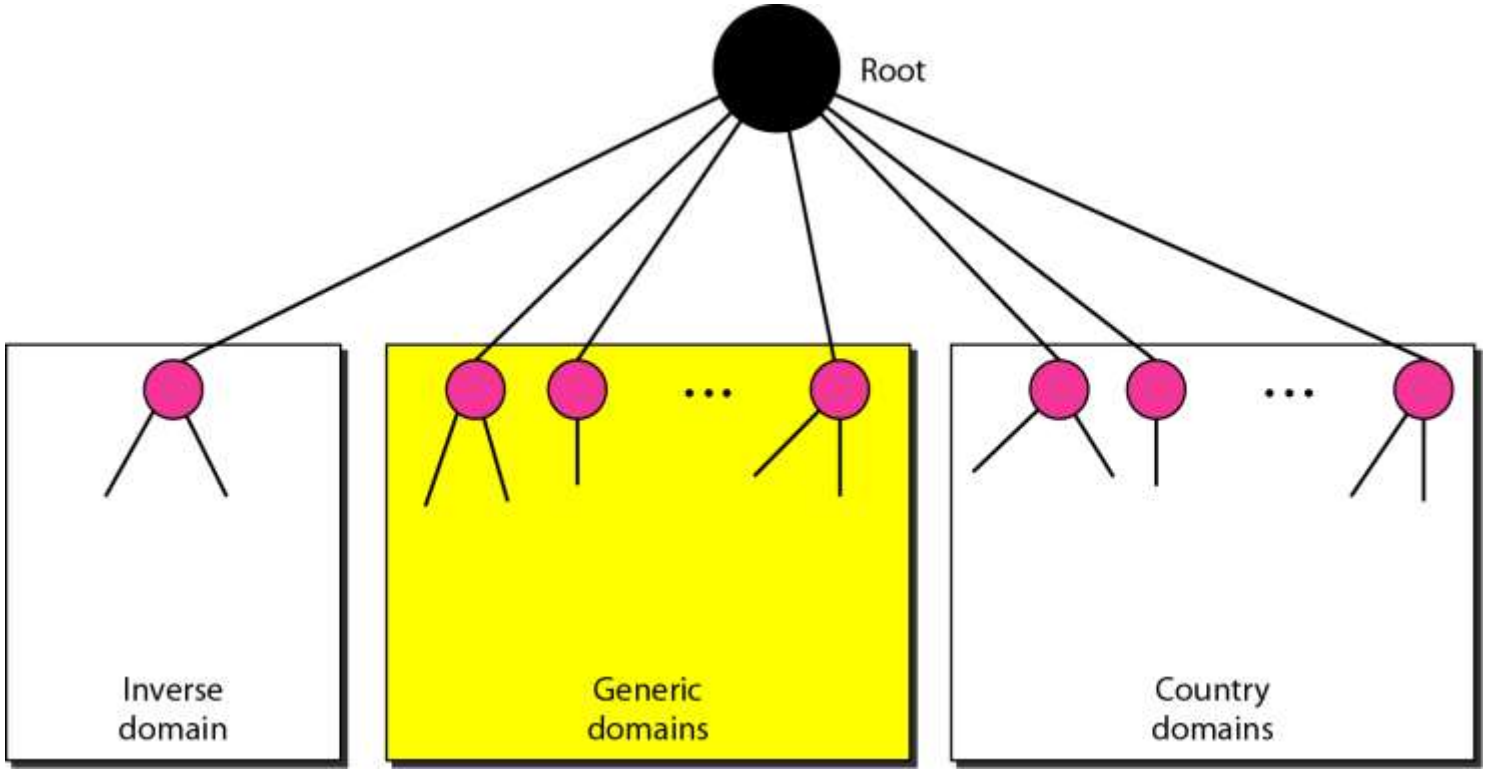
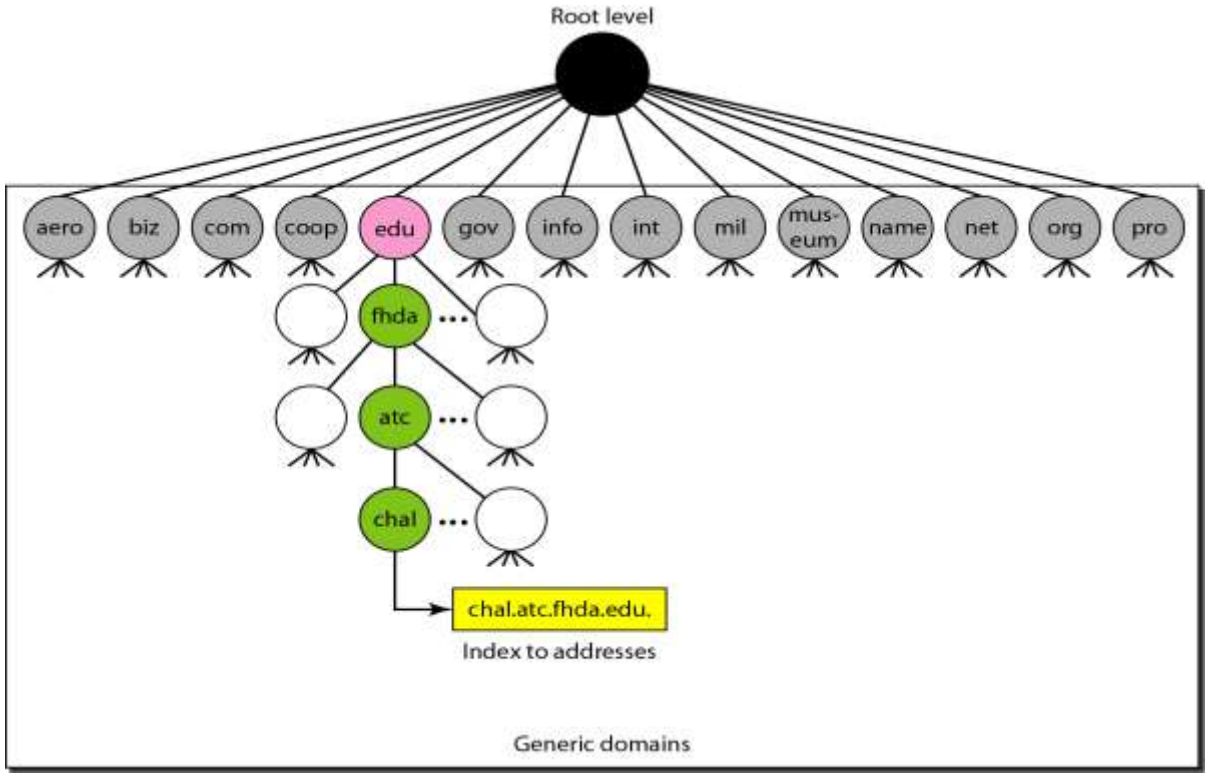


Figure 25.9 *Generic domains*

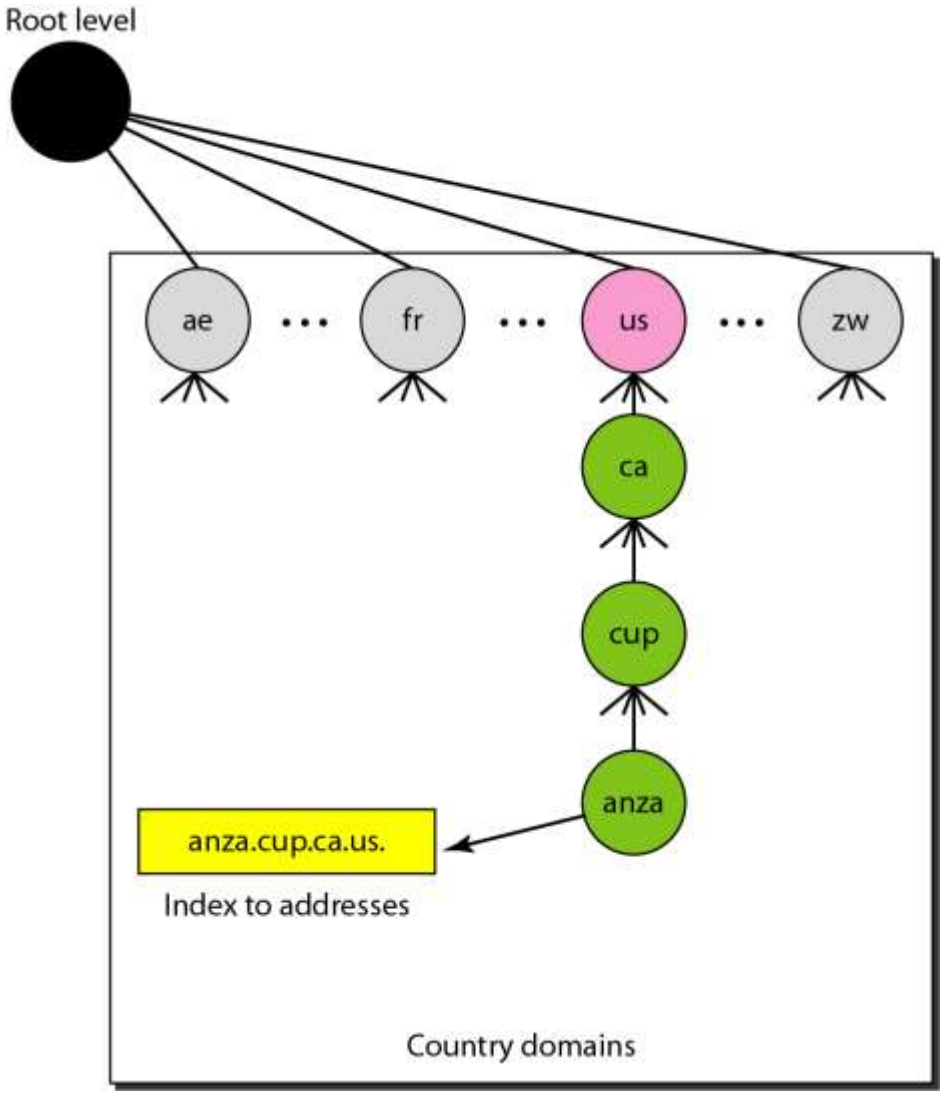


- The **generic domains** define registered hosts according to their generic behavior.
- Each node in the tree defines a domain, which is an index to the domain name space database

Table 25.1 *Generic domain labels*

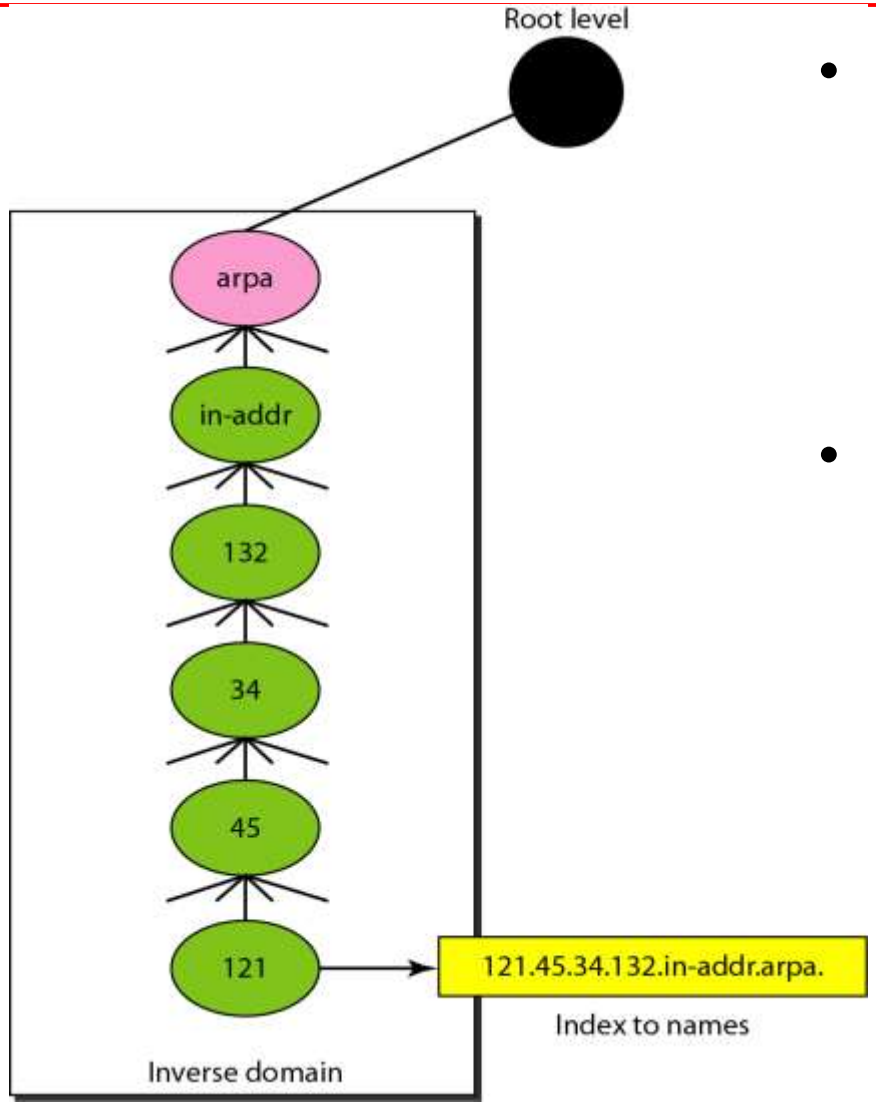
Label	*Description*
aero	Airlines and aerospace companies
biz	Businesses or firms (similar to "com")
com	Commercial organizations
coop	Cooperative business organizations
edu	Educational institutions
gov	Government institutions
info	Information service providers
int	International organizations
mil	Military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

Figure 25.10 *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

Figure 25.11 *Inverse domain*



- **The inverse domain is a part of DNS used to map an IP address back to its corresponding domain name.**
- **It does this by reversing the IP address and appending in-addr.arpa for reverse lookup.**

25-5 RESOLUTION

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

Topics discussed in this section:

Resolver
Mapping Names to Addresses
Mapping Addresses to Names
Recursive Resolution
Iterative resolution

Resolver

- Mapping a name to an address is called *name-address resolution.*
- DNS is designed as a client-server application. A host that needs to map an address to a name or a name to an address calls a DNS client called a ***resolver*.**
- The resolver accesses the closest DNS server with a mapping request. If the server has the information, it satisfies the resolver;
- otherwise, it either refers the resolver to other servers or asks other servers to provide the information

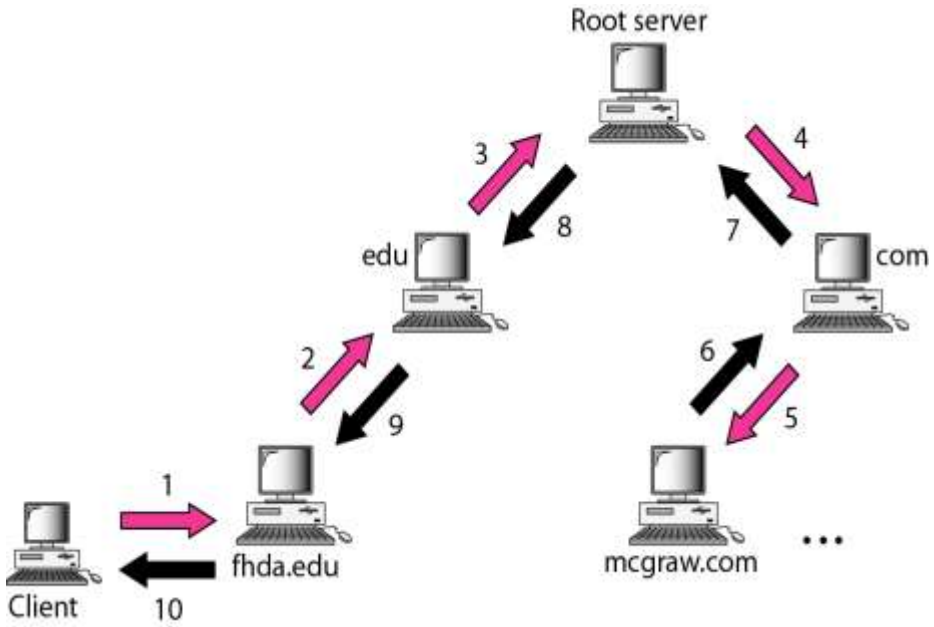
Mapping Names to Addresses

- 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.

Mapping Addresses to Names

- A client can send an **IP** address to a server to be mapped to a domain name.
- 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

Figure 25.12 *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.

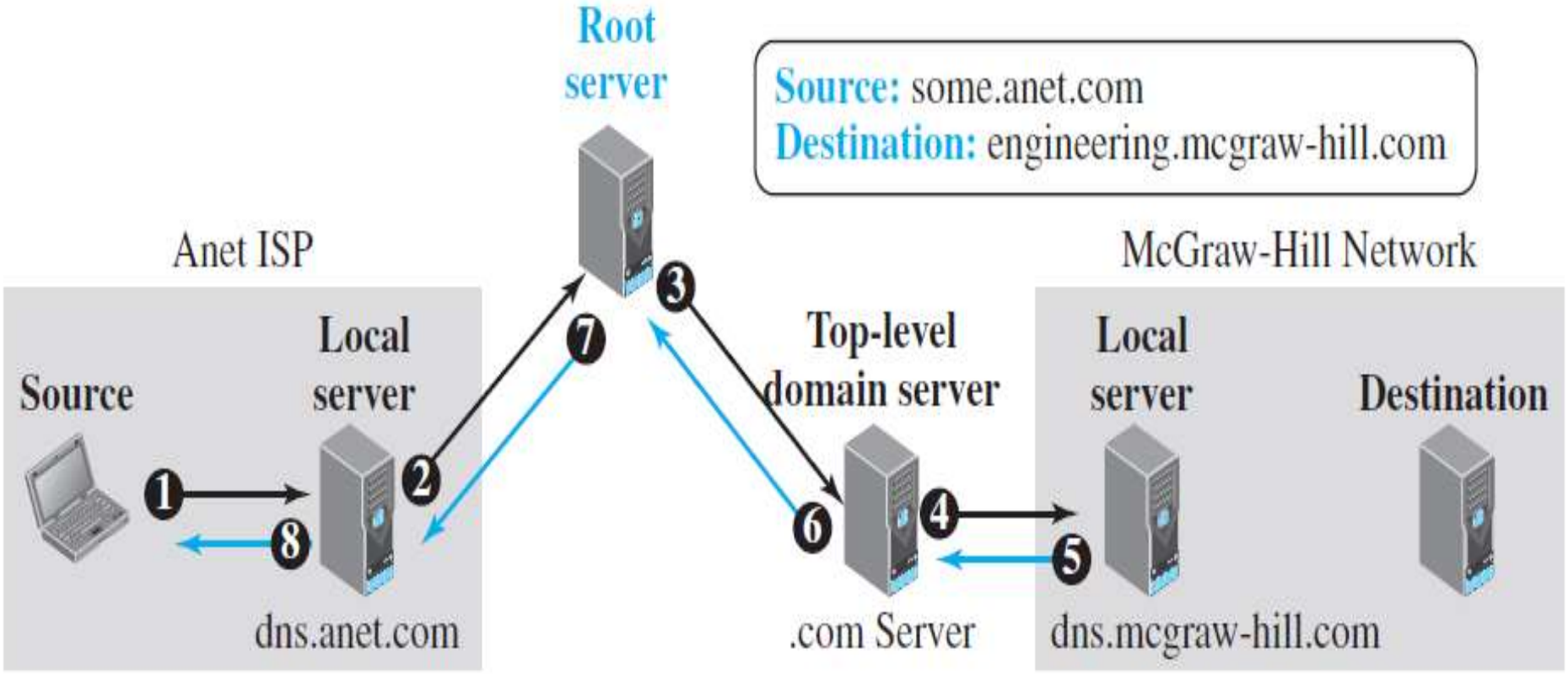
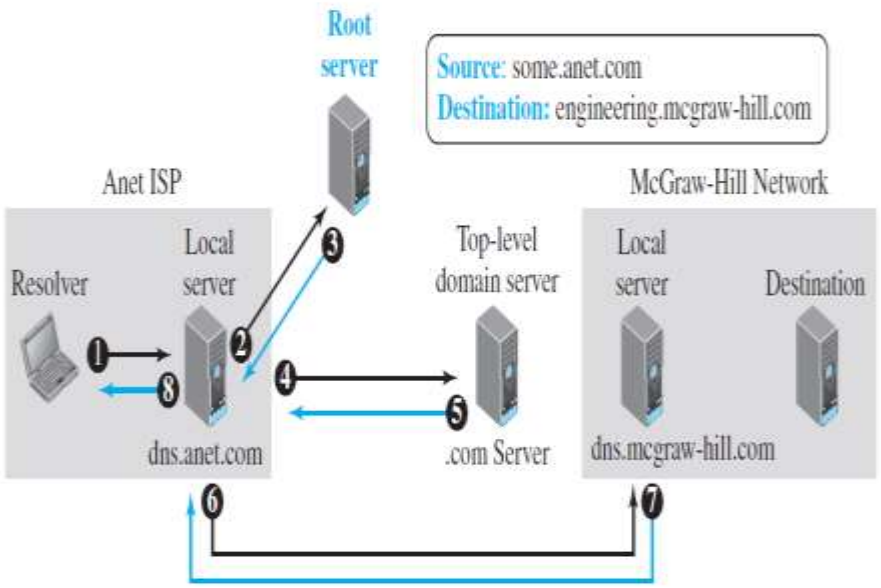

Root
server
Source: some.anet.com
Destination: engineering.mcgraw-hill.com
Anet ISP
McGraw-Hill Network
Source
Local
server
Top-level
domain server
Local
server
Destination
1
2
3
4
5
6
7
8
dns.anet.com
.com Server
dns.mcgraw-hill.com

Figure 25.13 *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 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.

25-6 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.

Topics discussed in this section:

Header

Figure 25.14 *Query and response messages*

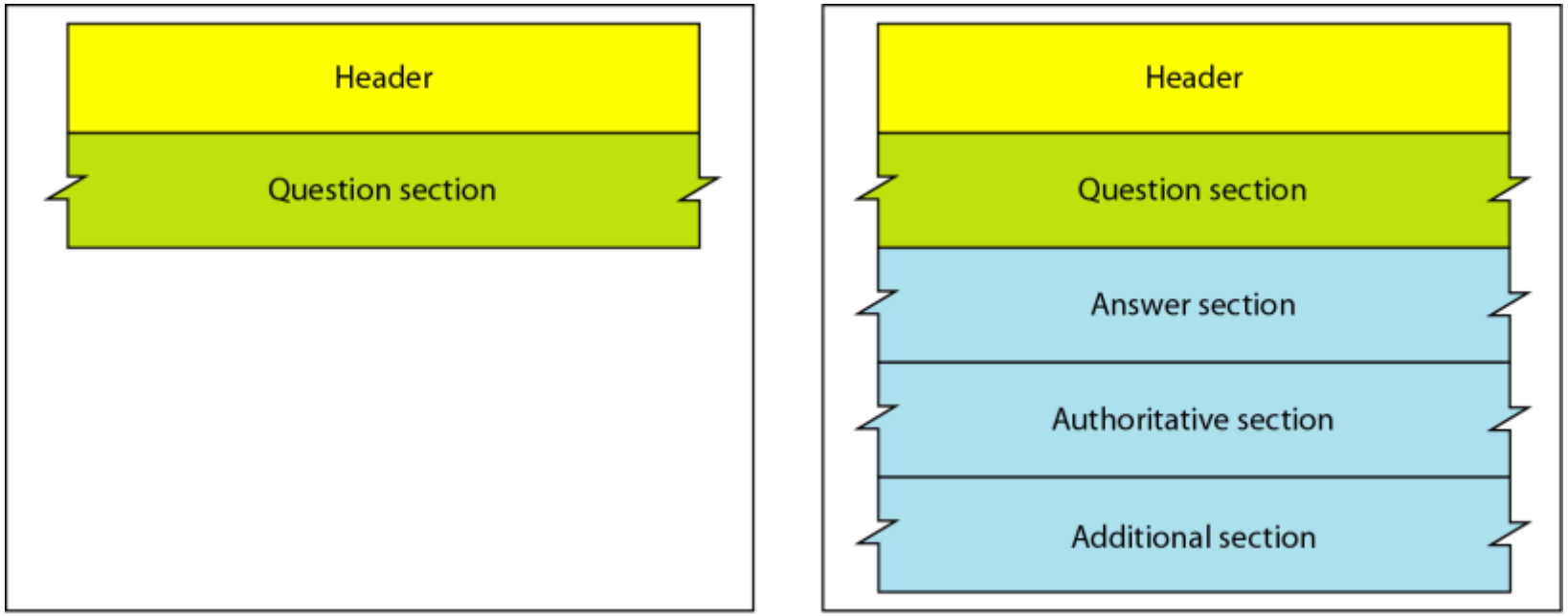
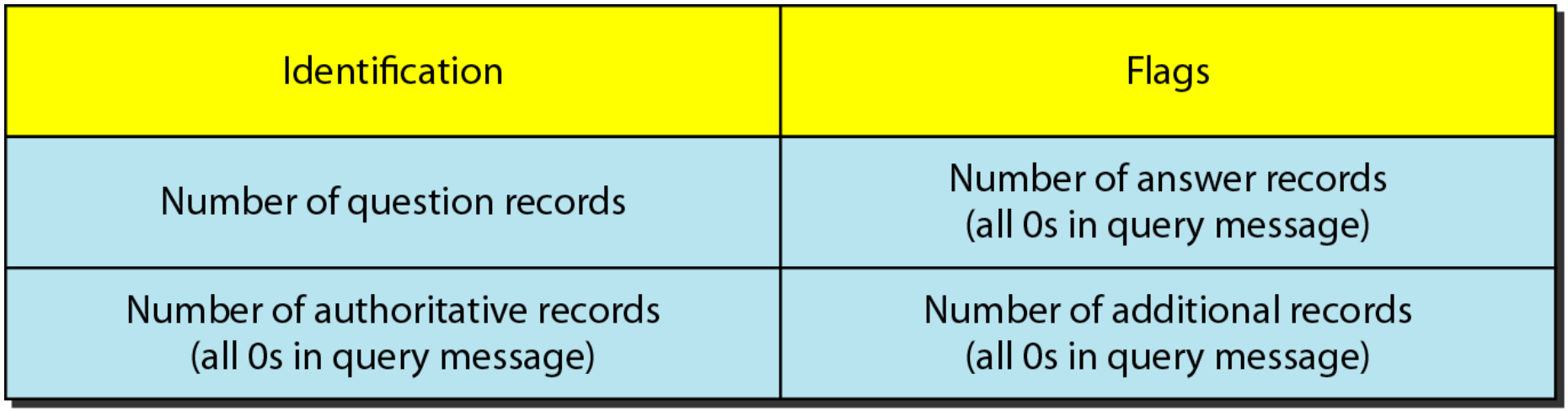
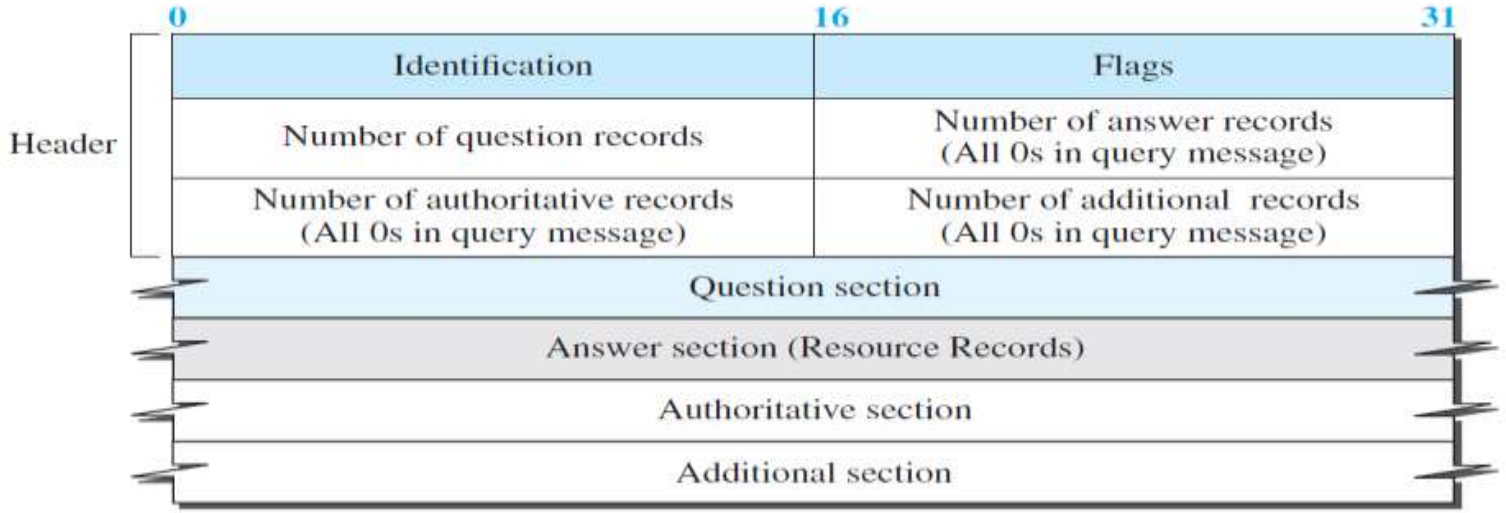


Figure 25.15 *Header format*





- **The header contains six 16-bit fields**
- **Identification field contains A random ID chosen by the client. Returned by the server in the response to match the query and reply**
- **Question Section Contains, QNAME: Domain name being queried**
- **Answer Section Contains resource records (RRs) that answer the query**
- **Authoritative Section, Contains the authoritative name servers for the domain**
- **Additional Section Contains additional helpful RR information**
- **Flags contains several sub-fields:**

  QR (1 bit): 0 = query, 1 = response
  Opcode (4 bits): usually 0 (standard query)
  AA (1 bit): authoritative answer
  TC (1 bit): truncated
  RD (1 bit): recursion desired
  RA (1 bit): recursion available
  Z (3 bits): reserved, must be 0
  RCODE (4 bits): response code

25-7 TYPES OF RECORDS

As we saw in Section 25.6, two types of records are used in DNS. The question records are used in the question section of the query and response messages. The resource records are used in the answer, authoritative, and additional information sections of the response message.

Topics discussed in this section:

Question Record

Resource Record

1. Question Record (in DNS Query)

- **A Question Record (QR) is the entry in the *Question Section* of a DNS message.**
- **It describes what the client is asking for.**
- **A Question Record contains only 3 fields:**

  1.QNAME – The domain name being queried
  Example: www.example.com
  2.QTYPE – The type of record requested
  Examples: A (IPv4 address), AAAA (IPv6), MX (mail server)
  3.QCLASS – The class of the query
  Usually: IN (Internet)

2. Resource Record (RR)

- **A Resource Record appears in the Answer, Authority, or Additional sections of a DNS message.**
- **It contains the actual data returned in a DNS response.**
- **A Resource Record has 5 fields:**

  NAME – Domain name
  TYPE – Type of data (A, AAAA, MX,)
  CLASS – Normally IN
  TTL – Time To Live (how long the data can be cached)
  RDLENGTH – Length of the data
  RDATA – The actual data

Chapter 26

Remote Logging, Electronic Mail, and File Transfer

26-1 REMOTE LOGGING

It would be impossible to write a specific client/server program for each demand. The better solution is a general-purpose client/server program that lets a user access any application program on a remote computer.

Topics discussed in this section:

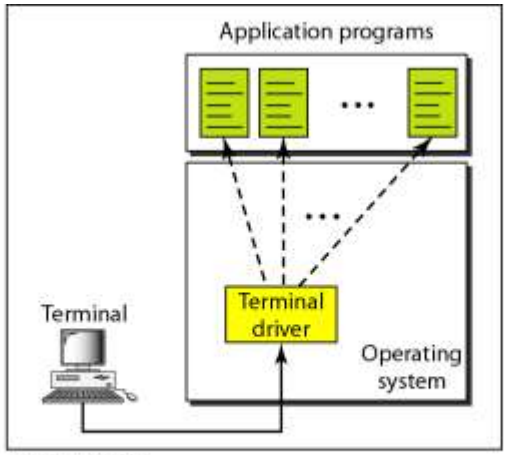
TELNET

- One of the original remote logging protocols is **TELNET,** which is an abbreviation for *TErminaL NETwork.*
- Although TELNET requires a logging name and password, it is vulnerable to hacking because it sends all data including the password in plaintext (not encrypted)
- When a user logs into a local system, it is called ***local logging***. As a user types at a terminal or at a workstation running a terminal emulator, the keystrokes are accepted by the terminal driver. The terminal driver passes the characters to the operating system. The operating system, in turn, interprets the combination of characters and invokes the desired application program or utility

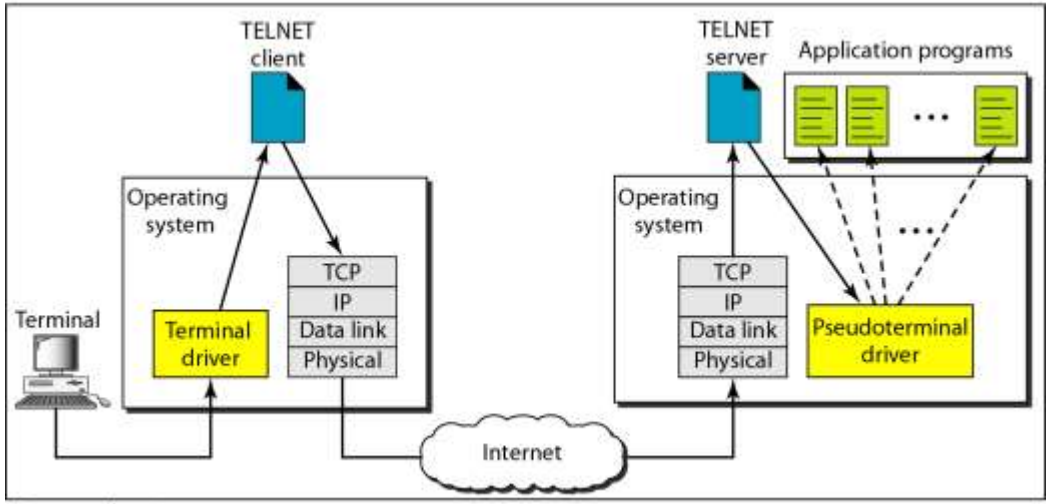
Remote Log-in

- The user sends the keystrokes to the terminal driver where the local operating system accepts the characters but does not interpret them.
- The characters are sent to the TELNET client, which transforms the characters into a universal character set called *Network Virtual Terminal* (NVT) characters (discussed below) and delivers them to the local TCP/IP stack.
- The commands or text, in NVT form, travel through the Internet and arrive at the TCP/IP stack at the remote machine.
- Here the characters are delivered to the operating system and passed to the TELNET server, which changes the characters to the corresponding characters understandable by the remote computer.
- However, the characters cannot be passed directly to the operating system because the remote operating system is not designed to receive characters from a TELNET server; it is designed to receive characters from a terminal driver
- 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

Figure 26.1 *Local and remote log-in*



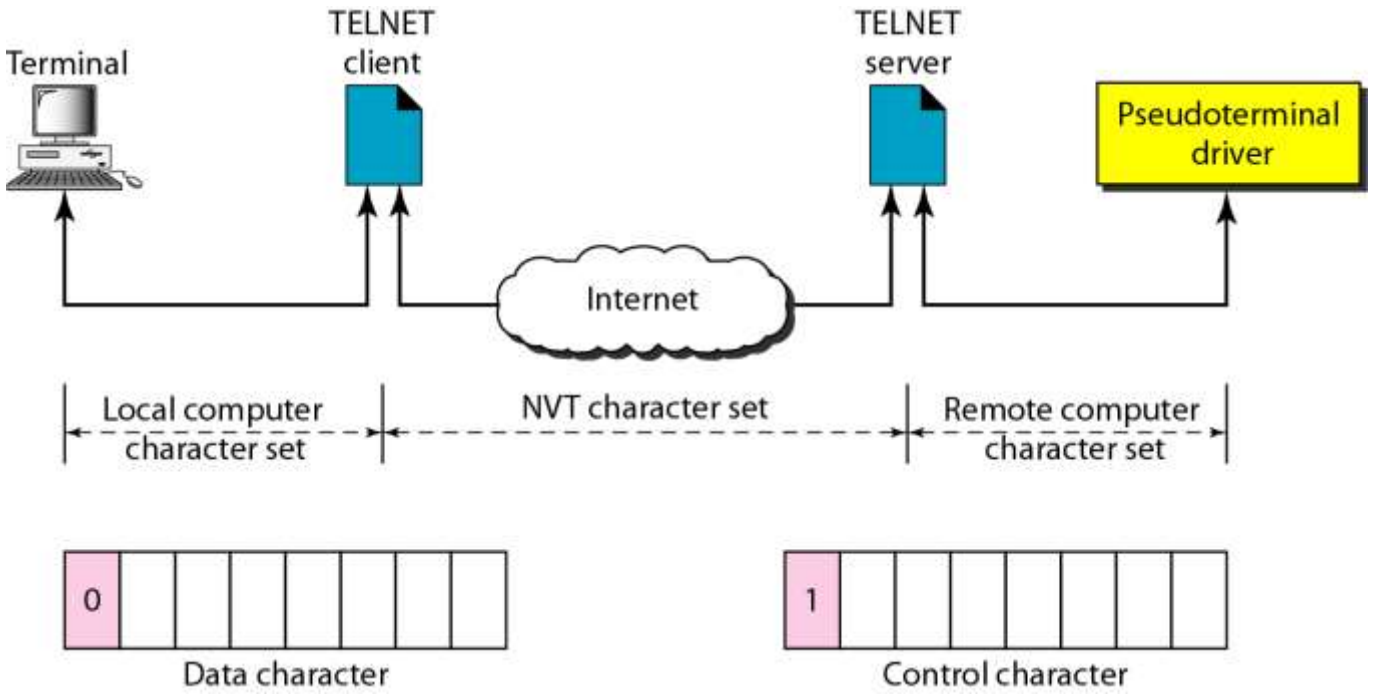
a. Local log-in



b. Remote log-in

Network Virtual Terminal (NVT)

Figure 26.2 *Concept of NVT*



- NVT uses two sets of characters, one for data and one for control.
- Both are 8-bit bytes as shown in Figure 26.24.
- For data, NVT normally uses what is called *NVT ASCII*. This is an 8-bit character set in which the seven lowest order bits are the same as US ASCII and the highest order bit is 0.
- To send control characters between computers (from client to server or vice versa), NVT uses an 8-bit character set in which the highest order bit is set to 1.

Table 26.1 *Some NVT control characters*

Character	*Decimal*	*Binary*	*Meaning*
EOF	236	11101100	End of file
EOR	239	11101111	End of record
SE	240	11110000	Suboption end
NOP	241	11110001	No operation
DM	242	11110010	Data mark
BRK	243	11110011	Break
IP	244	11110100	Interrupt process
AO	245	11110101	Abort output
AYT	246	11110110	Are you there?
EC	247	11110111	Erase character
EL	248	11111000	Erase line
GA	249	11111001	Go ahead
SB	250	11111010	Suboption begin
WILL	251	11111011	Agreement to enable option
WONT	252	11111100	Refusal to enable option
DO	253	11111101	Approval to option request
DONT	254	11111110	Denial of option request
IAC	255	11111111	Interpret (the next character) as control

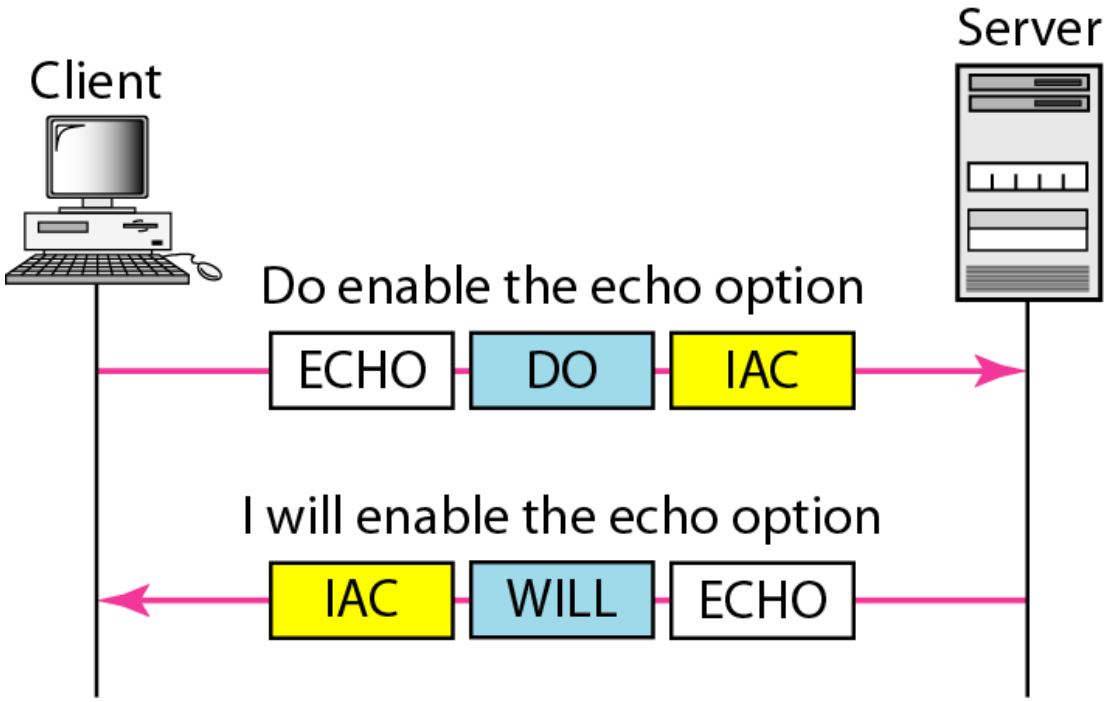
Table 26.3 *NVT character set for option negotiation*

Character	*Decimal*	*Binary*	*Meaning*
WILL	251	11111011	1. Offering to enable 2. Accepting a request to enable
WONT	252	11111100	1. Rejecting a request to enable 2. Offering to disable 3. Accepting a request to disable
DO	253	11111101	1. Approving an offer to enable 2. Requesting to enable
DONT	254	11111110	1. Disapproving an offer to enable 2. Approving an offer to disable 3. Requesting to disable

Example 26.1

Figure 26.4 shows an example of option negotiation. In this example, the client wants the server to echo each character sent to the server. The echo option is enabled by the server because it is the server that sends the characters back to the user terminal. Therefore, the client should request from the server the enabling of the option using DO. The request consists of three characters: IAC, DO, and ECHO. The server accepts the request and enables the option. It informs the client by sending the three-character approval: IAC, WILL, and ECHO.

Figure 26.4 *Example 26.1: Echo option*



26-2 ELECTRONIC MAIL

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

Topics discussed in this section:

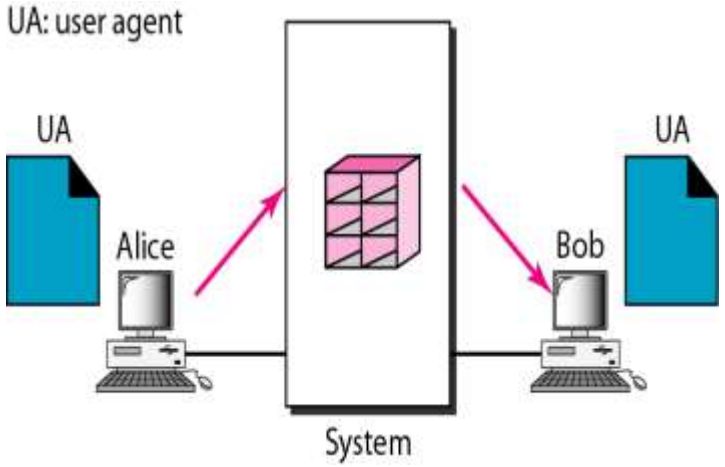
Architecture- ***First scenario , second scenario, third scenario , fourth scenario***

User Agent

Message Transfer Agent: SMTP

Message Access Agent: POP and IMAP

Figure 26.6 *First scenario in electronic mail*

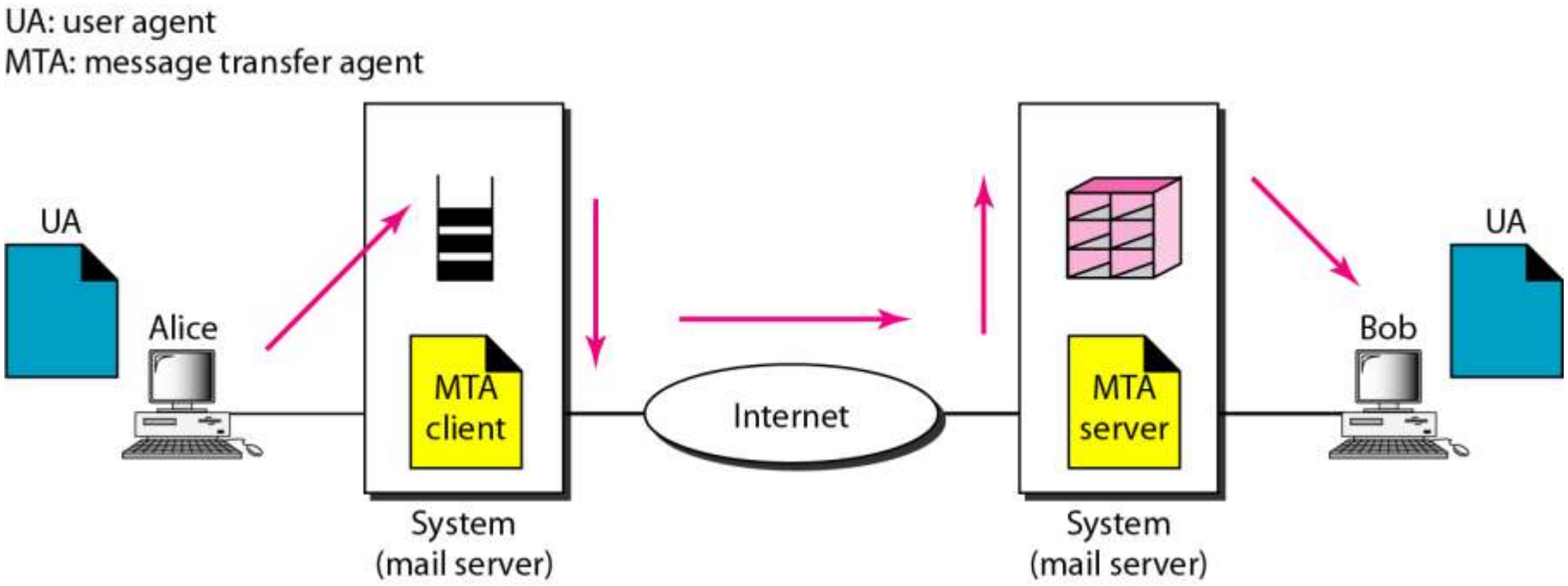


1. When both the sender and receiver are on the same computer system, they share a common storage area.
2. Each user (like Alice and Bob) has a separate mailbox, which is just a protected file on the system.
3. Only the owner of a mailbox is allowed to read or access it.
4. Alice uses a user agent (email program) to write her message and send it directly into Bob's mailbox.
5. The message includes the sender and receiver mailbox names so the system knows where to store it.
6. Bob later opens his user agent to read the message from his mailbox whenever he wants.

Note

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

Figure 26.7 *Second scenario in electronic mail*

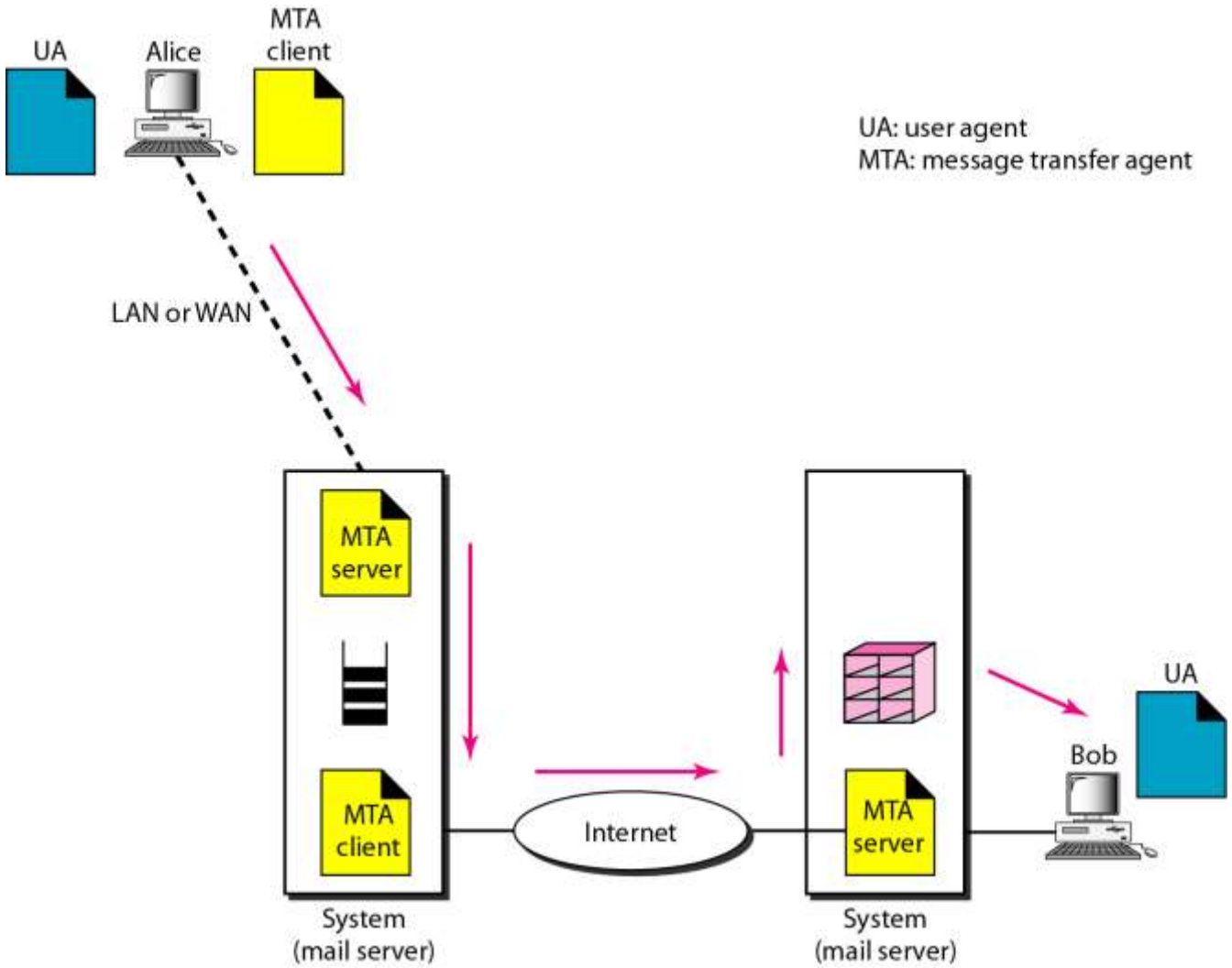


1. When the sender and receiver are on different systems, the e-mail must travel across the Internet.
2. Alice uses her user agent (UA) to write the message, just like in the first scenario.
3. Her computer sends the message to a mail server, which has a special program called the Message Transfer Agent (MTA client).
4. The MTA client sends the message over the Internet to Bob's mail server.
5. Bob's mail server has an MTA server, which receives the message and stores it in Bob's mailbox.
6. Bob uses his own user agent to open and read the message from his mailbox.

Note

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

Figure 26.8 *Third scenario in electronic mail*

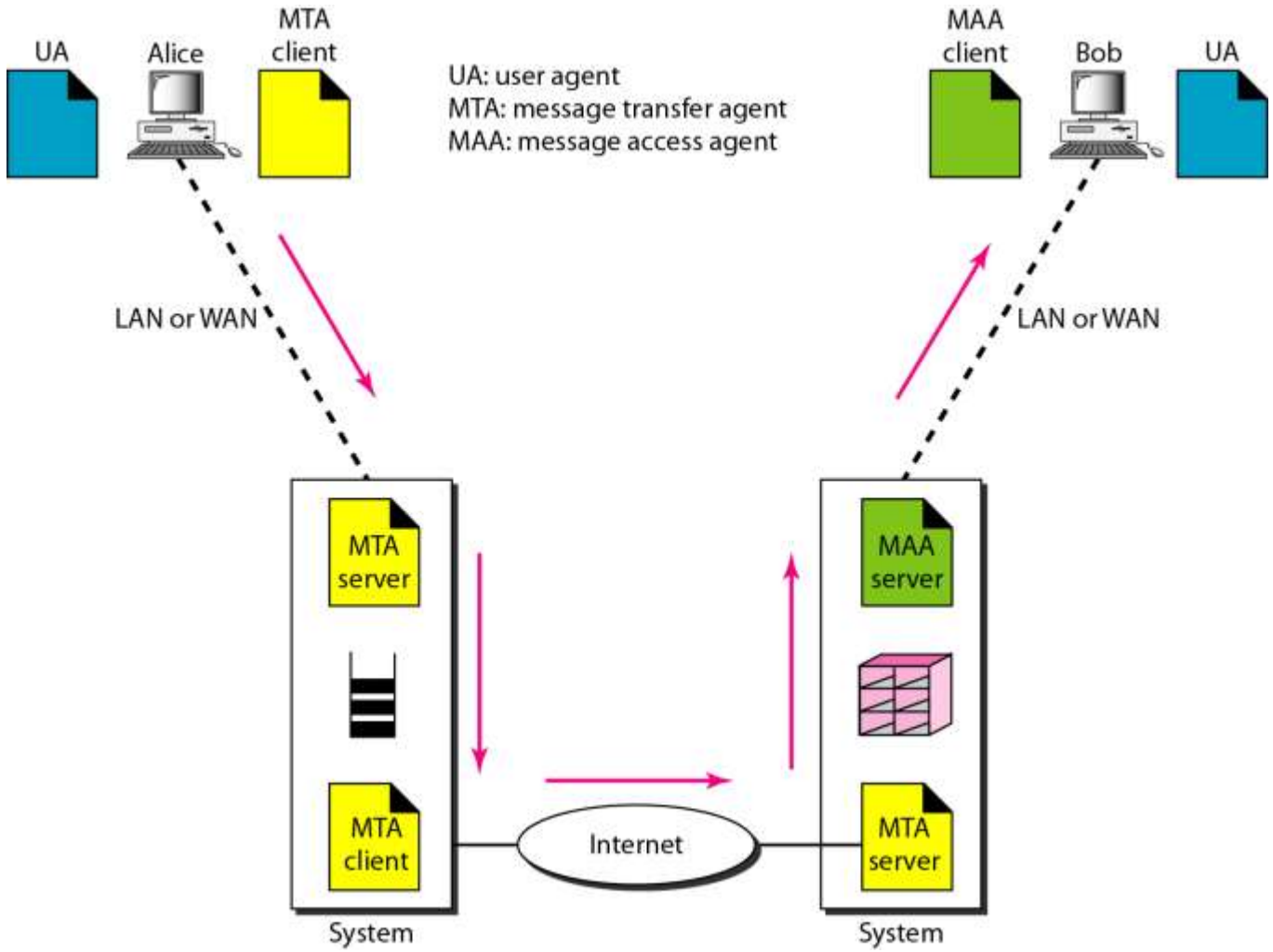


1. **Bob is normally connected to his mail server just like in the second scenario, so he receives messages in the usual way through his server.**
2. **Alice is not directly connected to her mail server; she connects from a distance.**
3. **She may use a point-to-point WAN connection like dial-up, DSL, or cable modem, or she may use a LAN in her organization.**
4. **Because of this, Alice must first send her message to the mail server she is connected to, not directly over the Internet.**
5. **The mail server then uses its MTA client to send Alice's message over the Internet to Bob's mail server.**
6. **Bob's mail server stores the message in Bob's mailbox, and Bob later uses his user agent (UA) to read it.**

Note

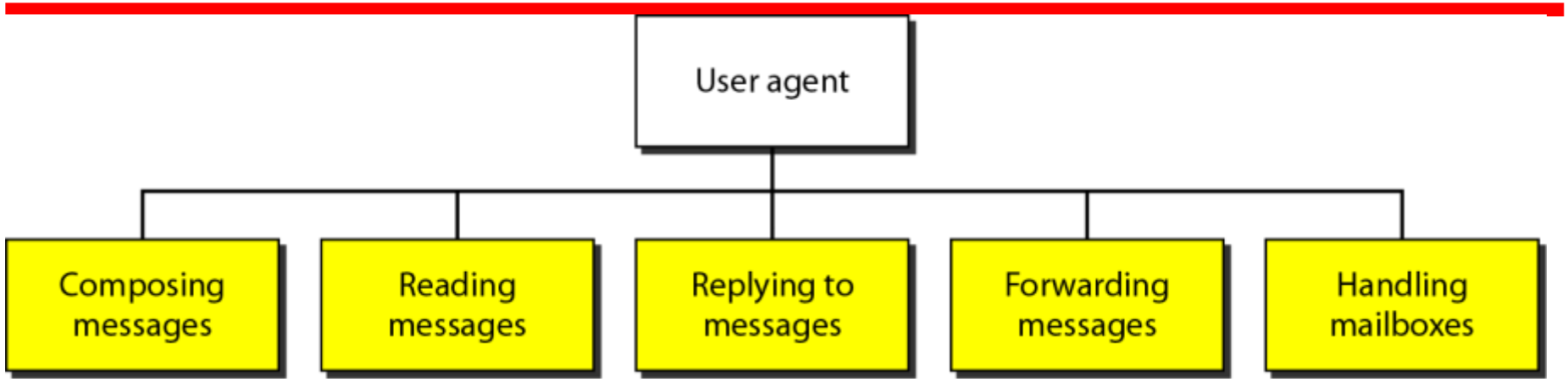
When the sender is connected to the mail server via a LAN or a WAN, we need two UAs and two pairs of MTAs (client and server).

Figure 26.9 *Fourth scenario in electronic mail*



1. In this scenario, both Alice and Bob are connected to their mail servers through either a LAN or a WAN.
2. Alice sends her message using her user agent (UA), and her mail server delivers it to Bob's mail server using message transfer agents (MTAs).
3. The message reaches Bob's mail server and is stored in Bob's mailbox on that server.
4. Bob is not directly connected to the mailbox; he must download or access the message from his server.
5. To do this, Bob uses a message access agent (MAA) client, such as POP3 or IMAP4.
6. The MAA client communicates with the MAA server running on Bob's mail server to retrieve and display Bob's messages.

Figure 26.11 *Services of user agent*



- Composing Messages A user agent helps the user compose the e-mail message to be sent out.
- Reading Messages The second duty of the user agent is to read the incoming messages.
- Replying to Messages After reading a message, a user can use the user agent to reply to a 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

User Agent Types

- There are two types of user agents: command-driven and GUI-based.
- Command-Driven Command-driven user agents belong to the early days of electronic mail.
- 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
- Some examples of GUI·based user agents are *Eudora, 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*

Figure 26.12 ***Format of an e-mail***

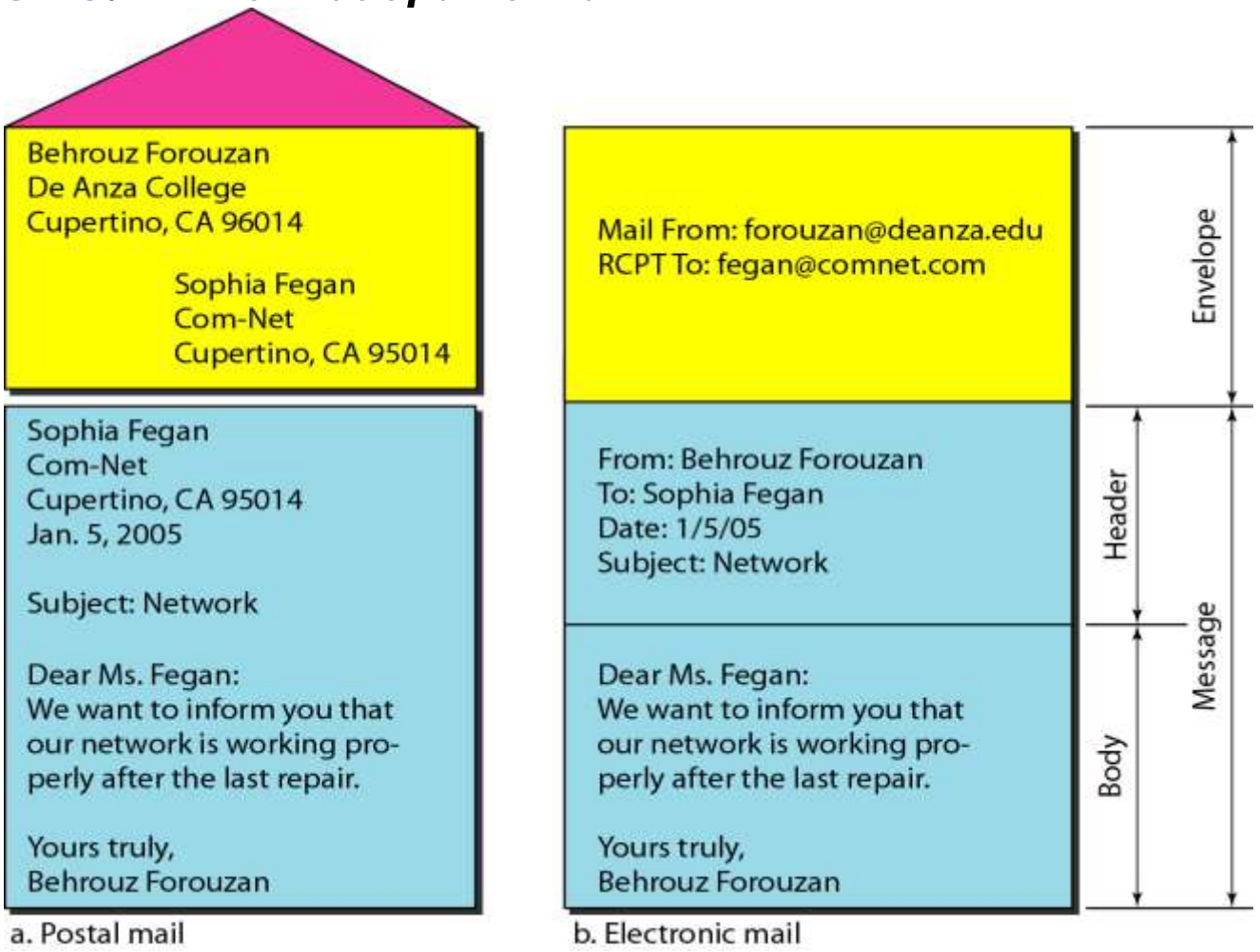
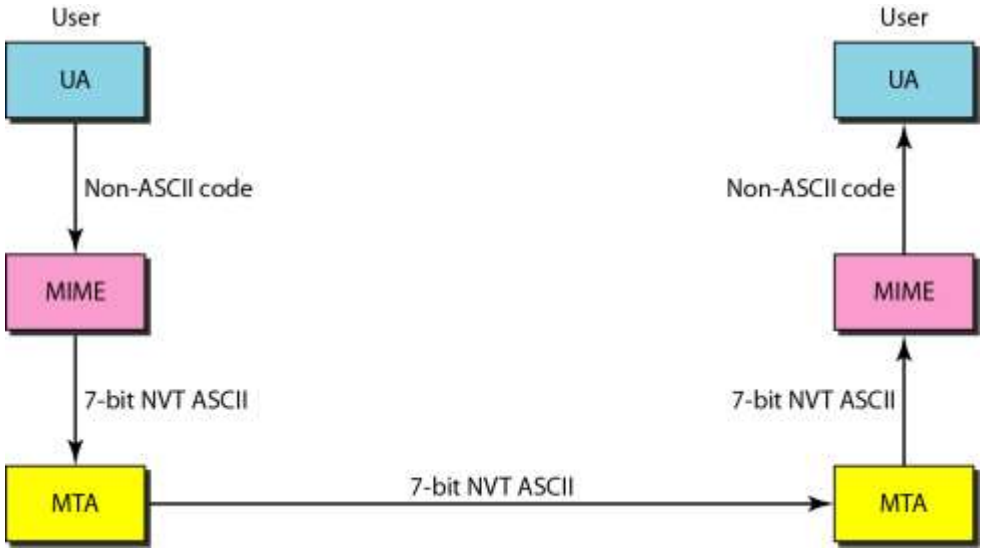


Figure 26.13 *E-mail address*

MIME Multipurpose Internet Mail Extensions



- Electronic mail has a simple structure. It can send messages only in NVT 7-bit ASCII format.
- 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

Figure 26.15 *MIME header*

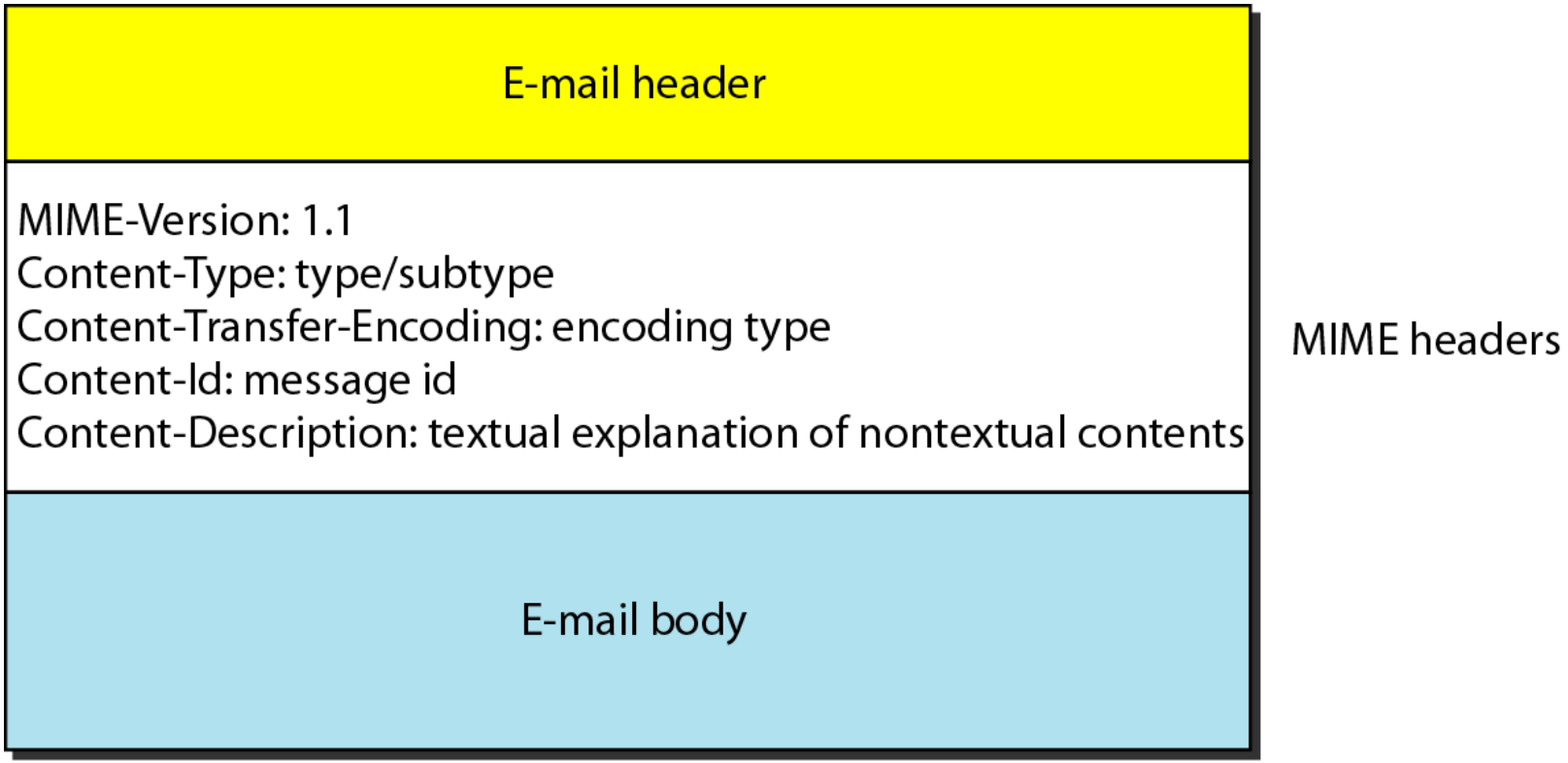


Table 26.5 *Data types and subtypes in MIME*

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)

Table 26.6 *Content-transfer-encoding*

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

3 important mail transfer phases in electronic mail using Simple Mail Transfer Protocol (SMTP)

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 (the 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 on the next slide. Note that we have added, for clarification, some comment lines, designated by the "=" signs. These lines are not part of the e-mail procedure.

Phase 1

```
$ 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
```

Phase 2

```
==================       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.
.
```

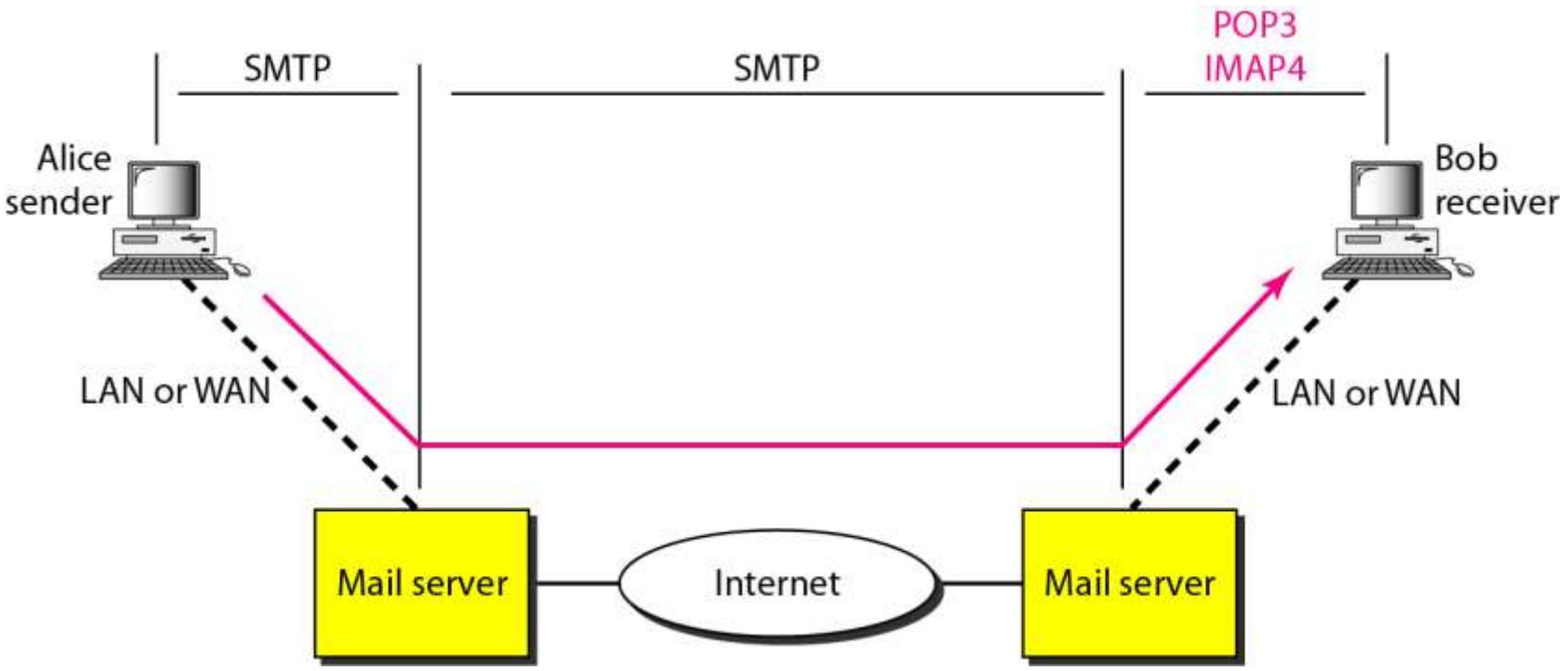
Phase 3

```
========== 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
- 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
- Currently two message access protocols are available: Post Office Protocol, version 3 (POP3) and Internet Mail Access Protocol, version 4 (IMAP4)

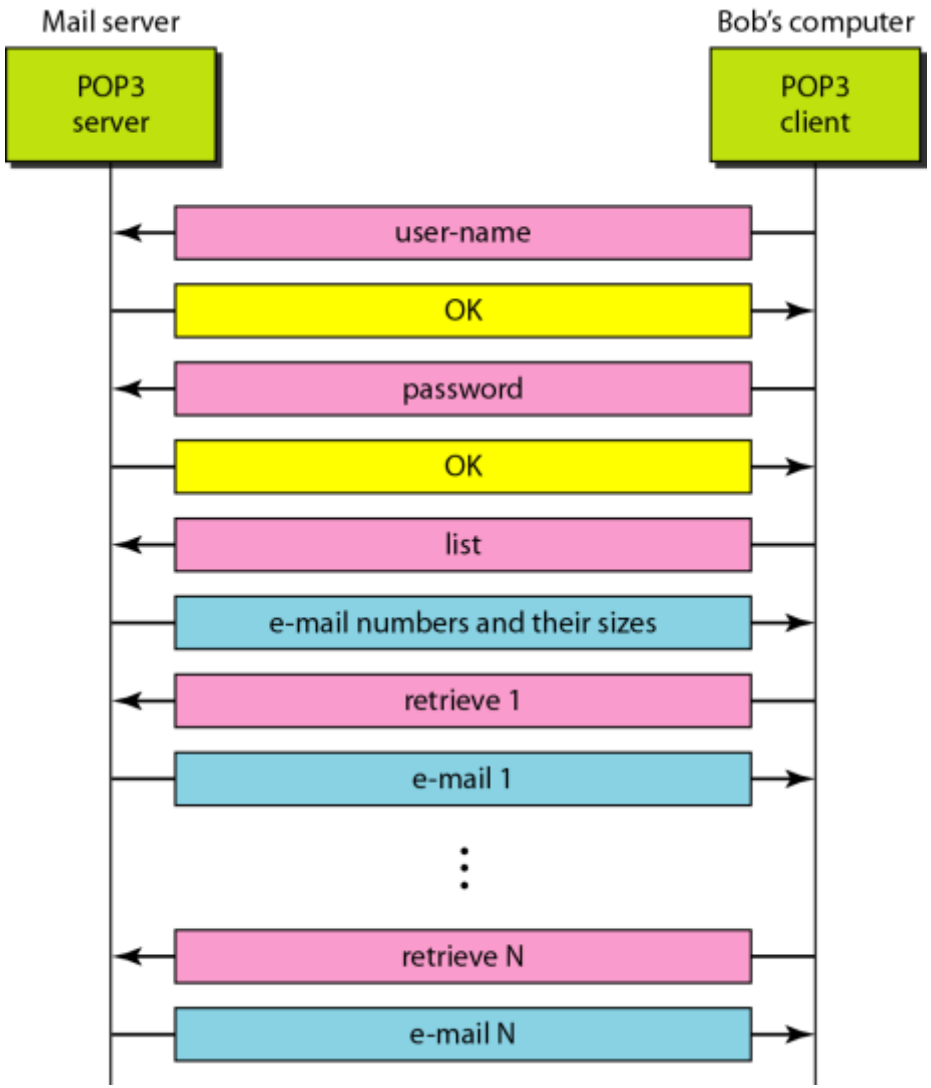
Figure 26.19 *POP3 and IMAP4*



POP3

- POP3 is a simple e-mail access protocol that allows a user to download e-mail from their mail server.
- The POP3 client program runs on the user's computer, and the POP3 server program runs on the mail server.
- When the user wants to check e-mail, the POP3 client connects to the server using TCP port 110.
- The user must send a username and password so the server can give access to the mailbox.
- The user can then list and download the messages stored on the server, usually one at a time.
- POP3 works in two modes:
  - Delete mode → messages are removed from the server after being downloaded.
  - Keep mode → messages stay on the server even after downloading.

Figure 26.20 *The exchange of commands and responses in POP3*



Internet Mail Access Protocol, version 4 (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.
- A user can partially download e-mail.
- 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.