Quality of Service

Quality of service

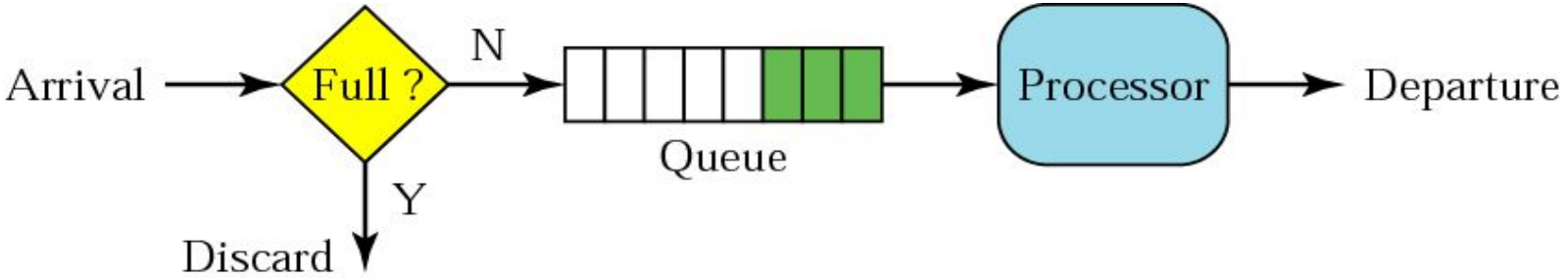
- Flow Characteristics
  - **Reliability**: lack of reliability means losing of packet and acknowledgements.
  - **Delay**: audio, vedio conferencing need less delay while file transfer and e-mail may have large delay.
  - **Bandwidth**: video conferencing require much BW as large no. bit are sent at one time.
  - **Jitter**: variation in delay.
    - If 4 pkts are sent at 0,1,2,3 time and arrive at 20,21,22,23 time. Then delay for all is same and=20. But if reach at 21,23,21,28 time, then delay is diff = 21,23,21,24

Techniques to improve QoS

- Scheduling
  - FIFO Queuing
  - Priority Queuing
  - Weighted Fair Queuing
- Traffic Shaping
  - Leaky Bucket
  - Token Bucket
- Resource Reservation
- Admission Control

Scheduling
FIFO Queuing

- Packets wait in a buffer until the node is ready to process them
- If avg rate of arrival is higher than processing rate-queue will be filled up & new pkt will be dropped.



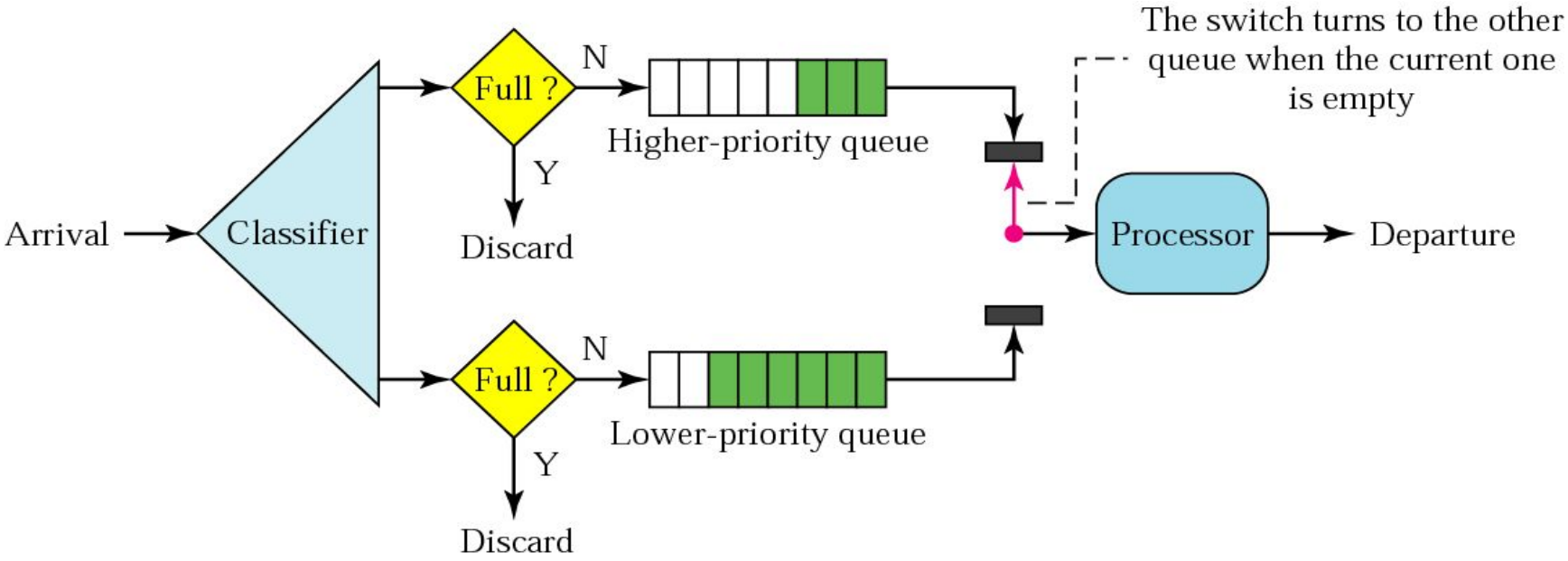
Scheduling contd..

Priority Queuing

- Packets are grouped according to some priority class.
- Each priority class has its own queue.
- Highest priority queue is processed first and lowest – in last.
- Better than FIFO.
- If higher priority packets are continuously flowing- lower priority packets will have to **STARVE.**

Scheduling contd..

Priority Queuing

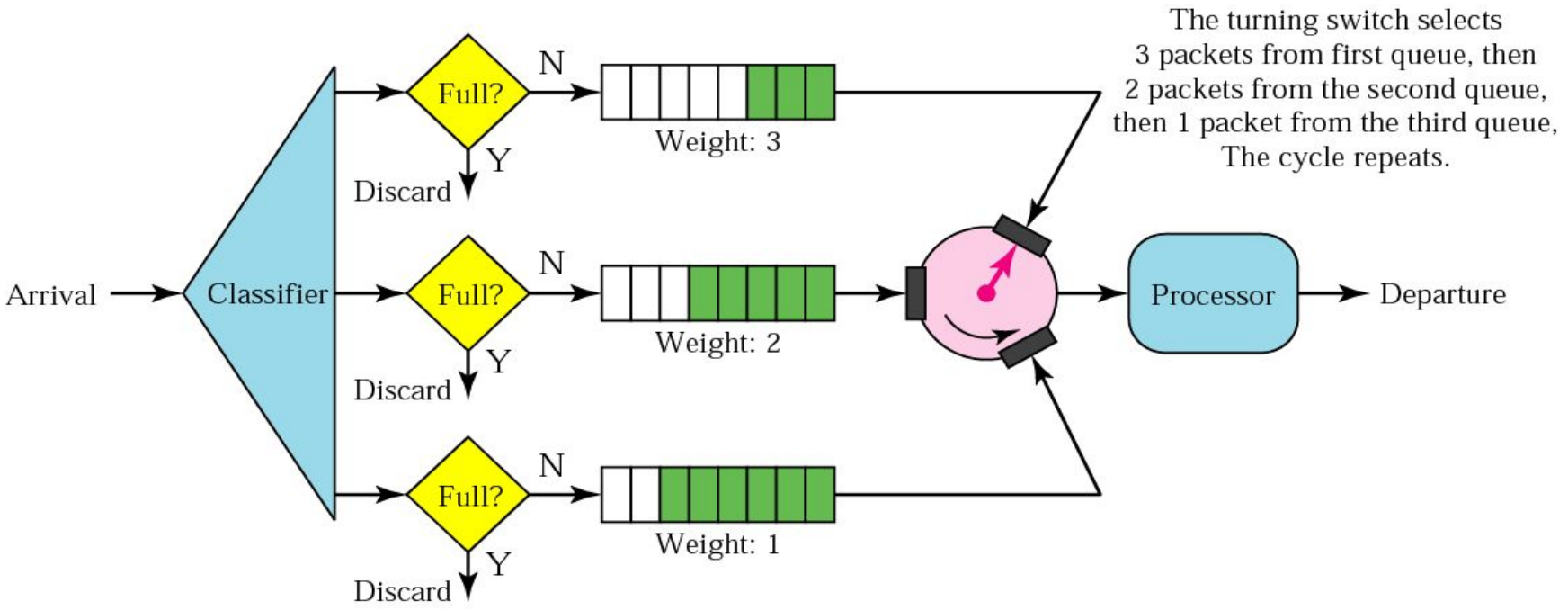


Scheduling contd..
Weighted Fair Queuing

- Queues are weighted based on priority: higher priority means higher weight.
- No. of packet selected from each queue depends on the corresponding weight.
- If no priority – all weight are considered to be having equal weight.

Scheduling contd..

Weighted Fair Queuing

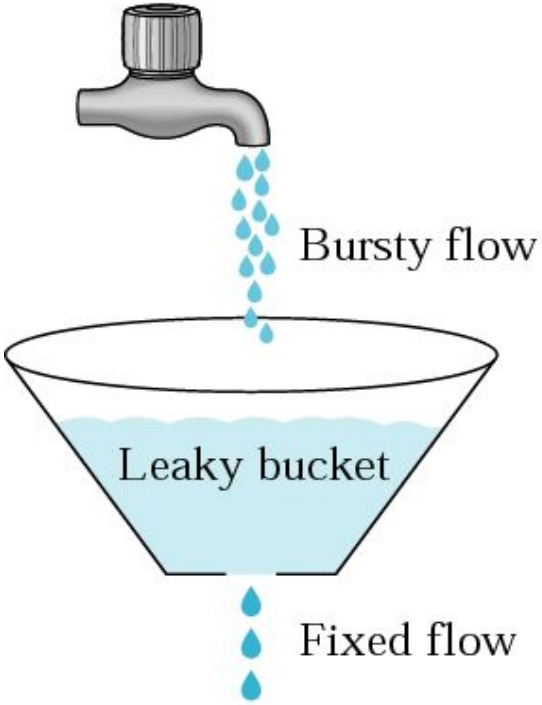


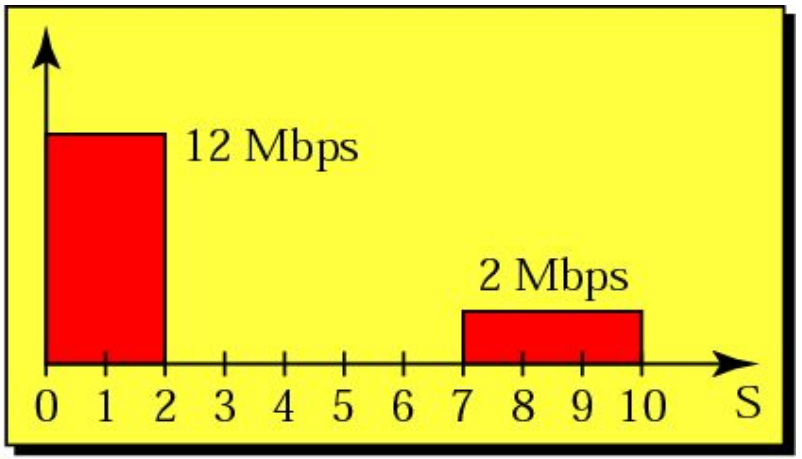
Traffic Shaping
Leaky Bucket

- If bucket has a small hole at the bottom-water will leak with constant rate.
- Input rate may vary, but output rate will remain constant.
- Similarly Bursty data is stored in the bucket and sent out with constant rate.

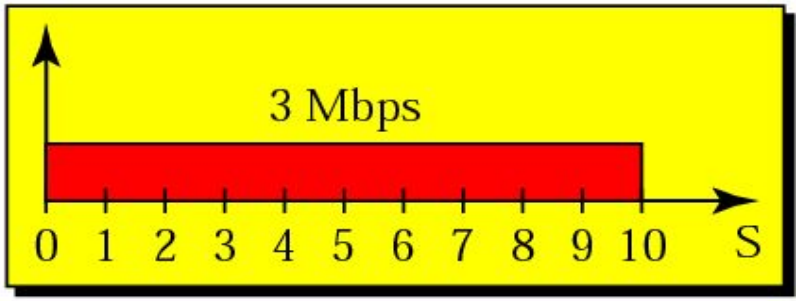
Traffic Shaping contd..

Leaky Bucket





Bursty data

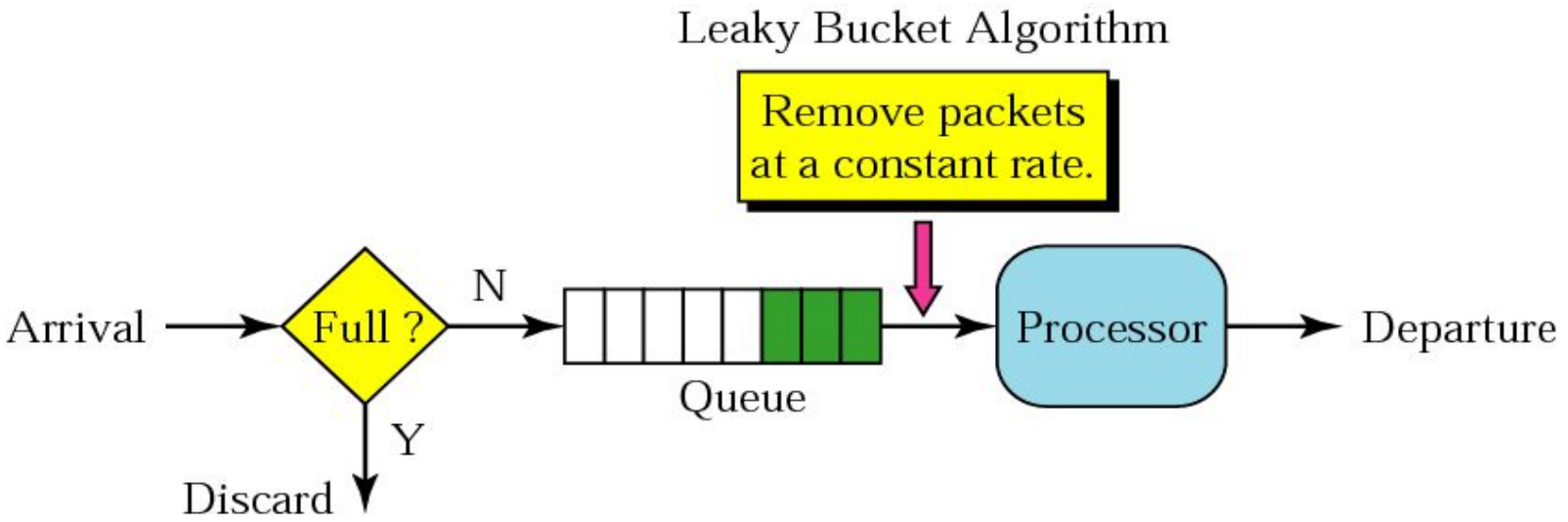


Fixed-rate data

Traffic Shaping contd..

Leaky Bucket Implementation

- A FIFO queue holds the packets.
- Shapes the bursty traffic into fixed-rate traffic by averaging the data rate. It may drop the packet-if buffer is full.



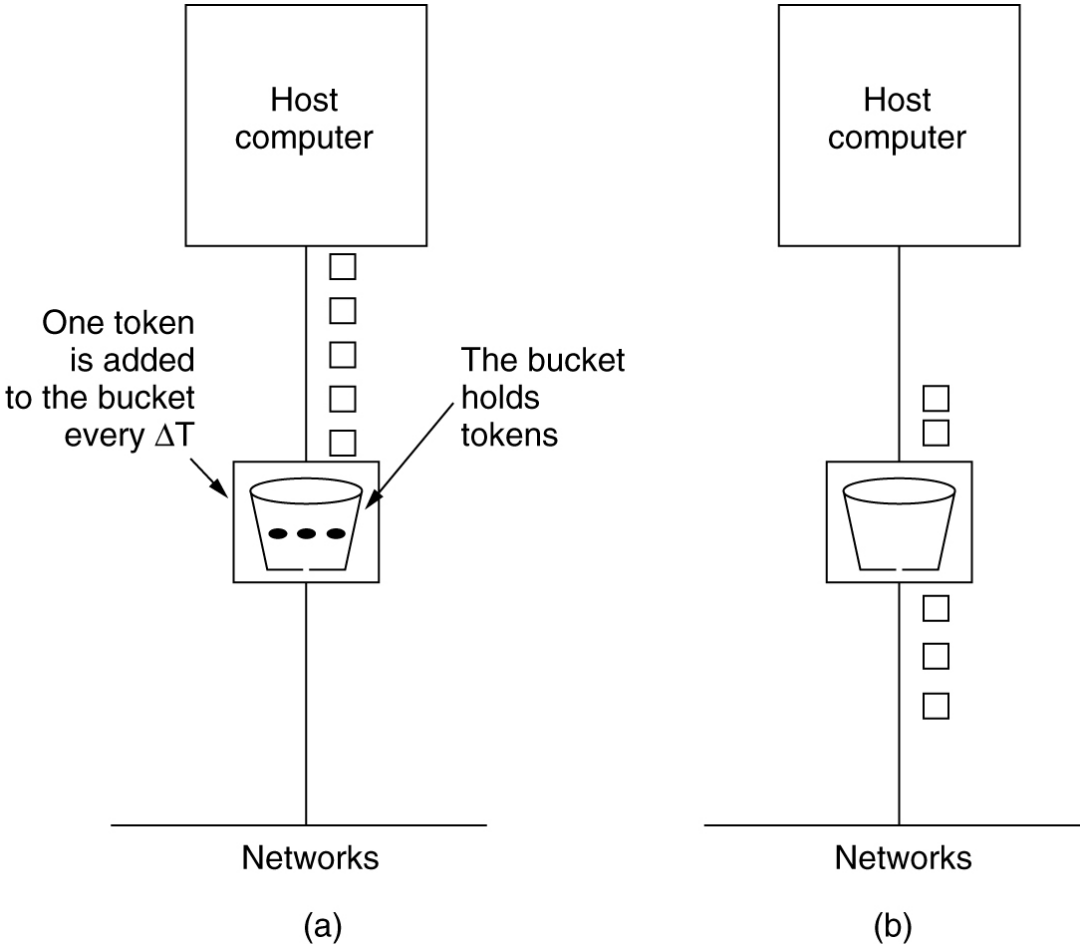
Traffic Shaping
Token Bucket

- In leaky bucket- output is constant rate. But in token bucket- it allows to speedup the output if large bursts arrive.
- Bucket holds the token, generated by a clock at the rate of one token every Δt sec.
- For a packet to be transmitted, it must capture and destroy one token.
- Implementation:
  - Initially counter is Zero.
  - Each time a token is added → the counter is incremented by 1.
  - Each time a packet is transmitted → the counter is decremented by 1.
  - When counter is Zero → the host can not send the data.

Traffic Shaping
Token Bucket

(a): Bucket holding 3 tokens, with 5 packets to be transmitted

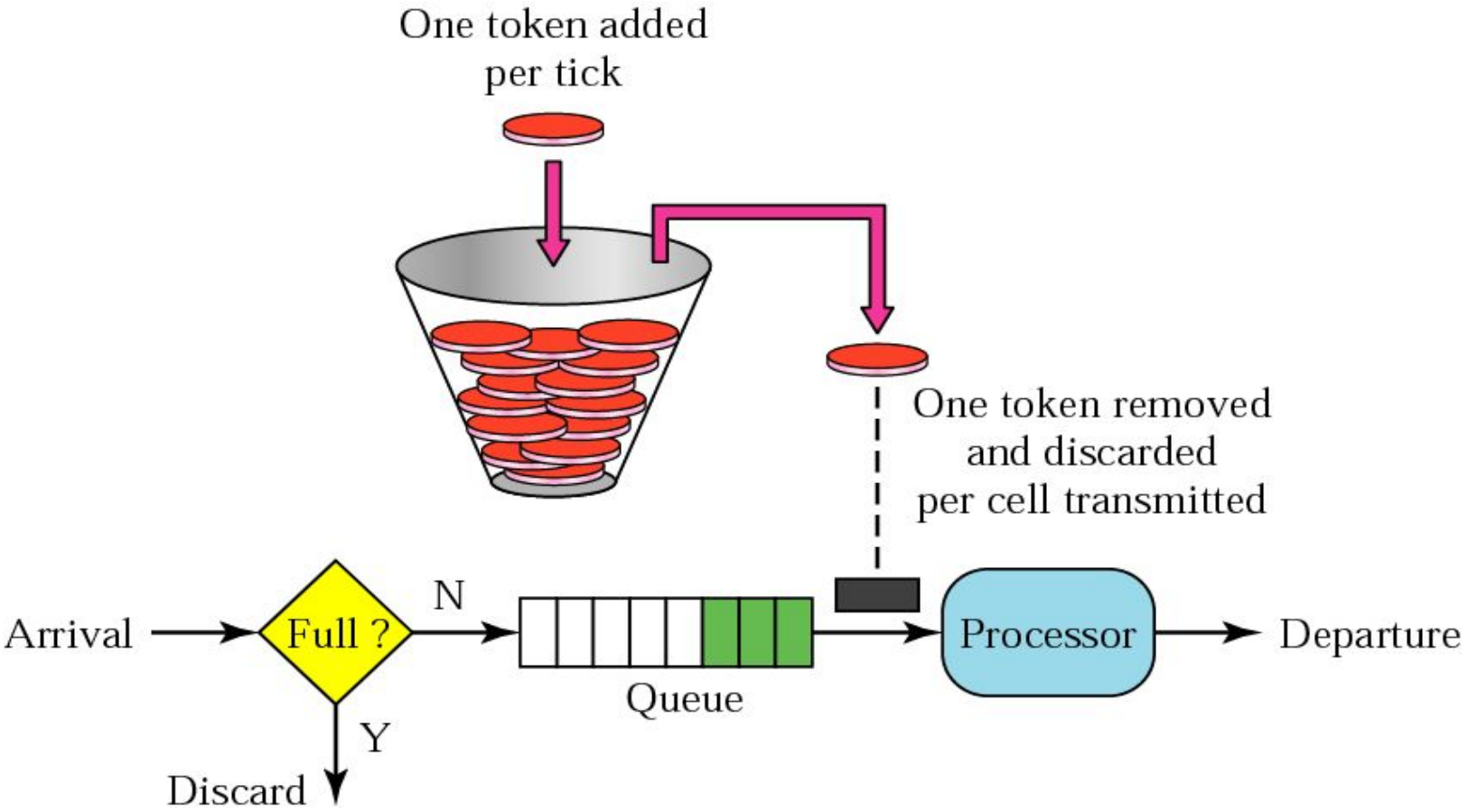
(b): 3 of the 5 packets are have gone through, but the other two are still waiting for 2 more tokens to be generated.



Leaky Bucket vs Token Bucket

- **Leaky Bucket**: if host is not sending for a sometime - its bucket becomes empty.
- **Token Bucket**: it allows the idle hosts to accumulate the credit for future in the form of token, up to the max size of the bucket i.e. 'n'.
  - By this property a host can send a burst of up to n packets at once – whenever becomes active.
- **Leaky Bucket**: Discards the packets when bucket is full.
- **Token Bucket**: Discards the tokens but never the packet.

Figure 23.18 Token bucket



Data Communications and Networking Fourth Edition

Forouzan

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.

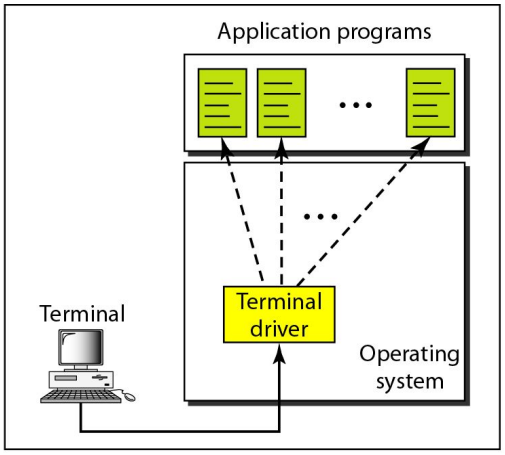
Topics discussed in this section:

TELNET (TErminaL NETwork)

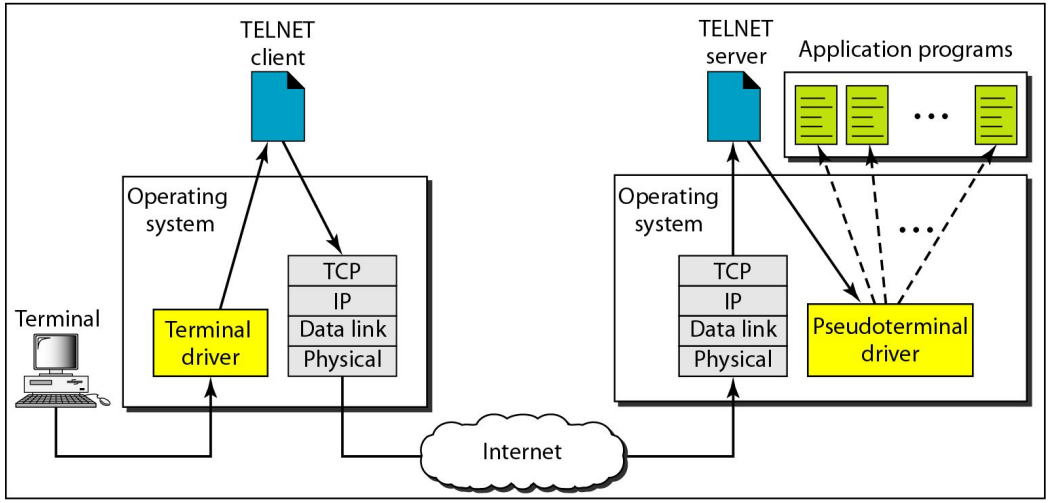
Note

TELNET is a general-purpose client/server application program.

Figure 26.1 *Local and remote log-in*

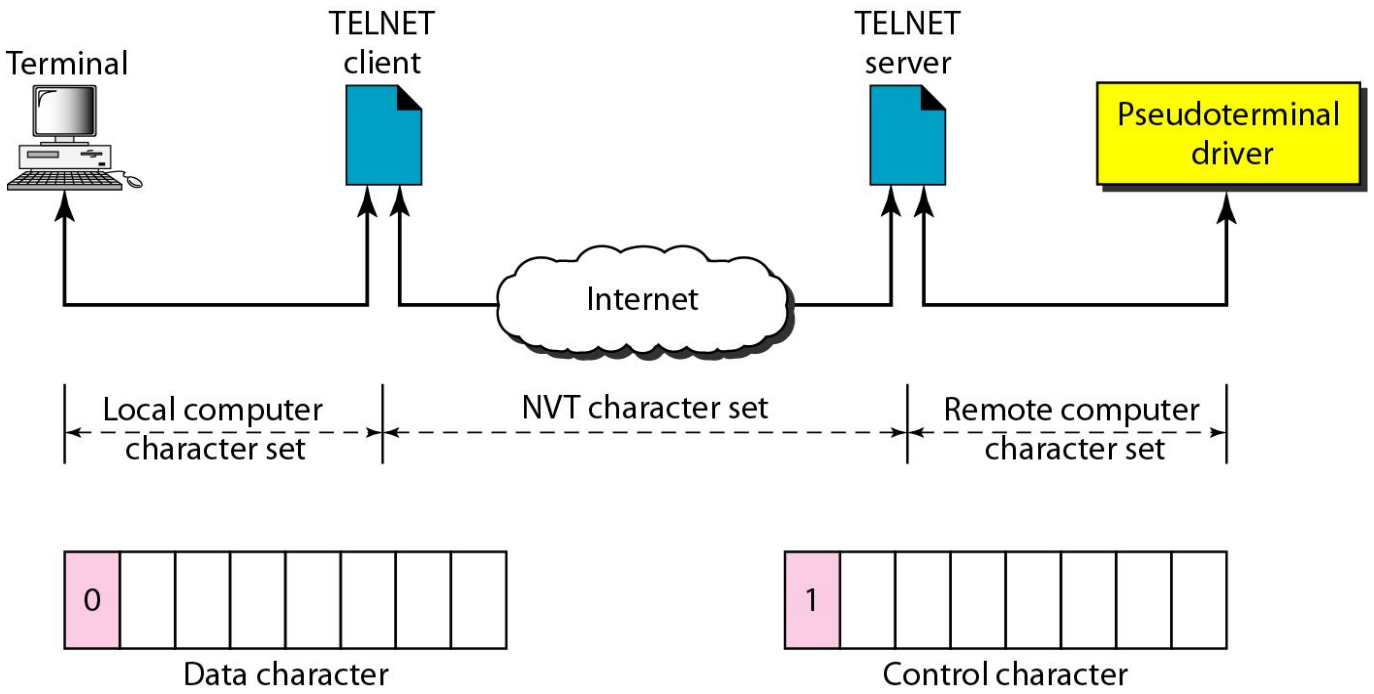


a. Local log-in



b. Remote log-in

Figure 26.2 *Concept of NVT*



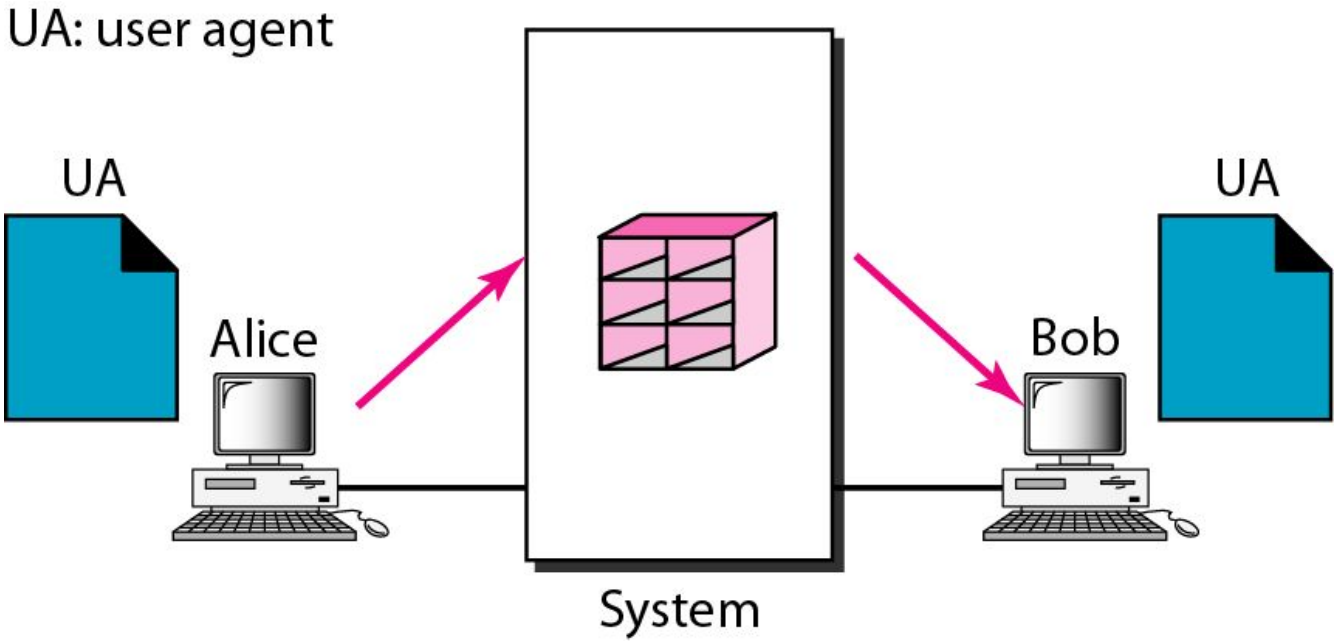
26-2 ELECTRONIC MAIL

One of the most popular Internet services is electronic mail (e-mail). The designers of the Internet probably never imagined the popularity of this application program. Its architecture consists of several components.

Topics discussed in this section:

Architecture
User Agent
Message Transfer Agent: SMTP
Message Access Agent: POP and IMAP
Web-Based Mail

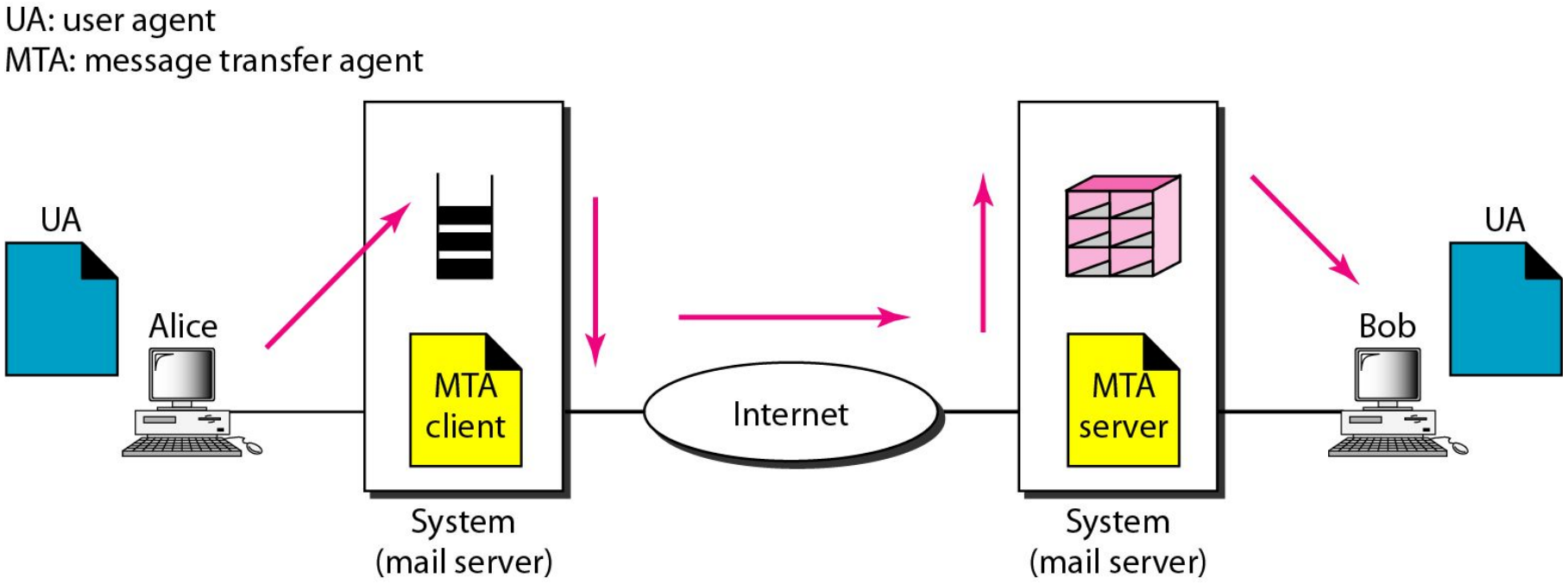
Figure 26.6 *First scenario in electronic mail*



Note

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

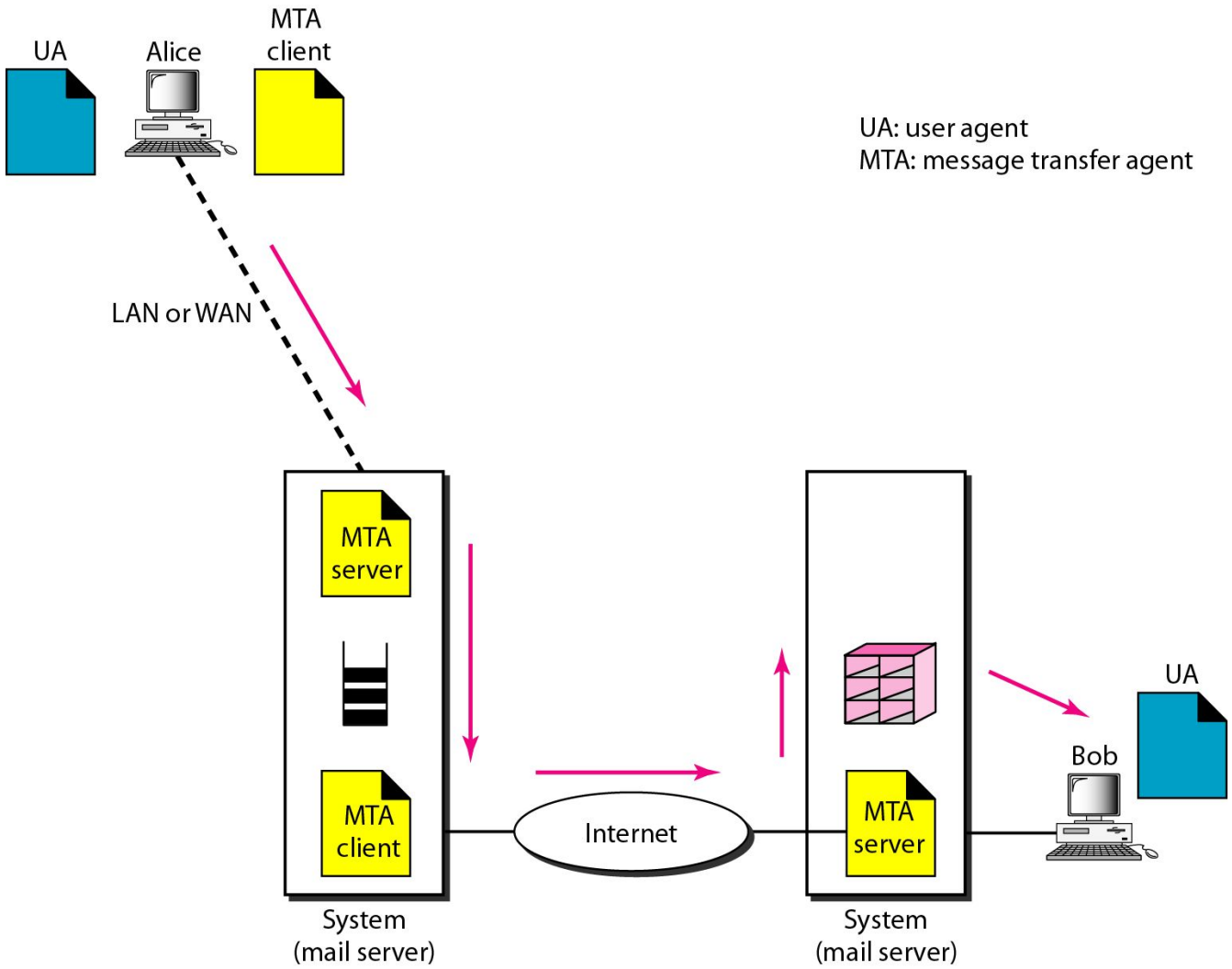
Figure 26.7 *Second scenario in electronic mail*



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*



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*

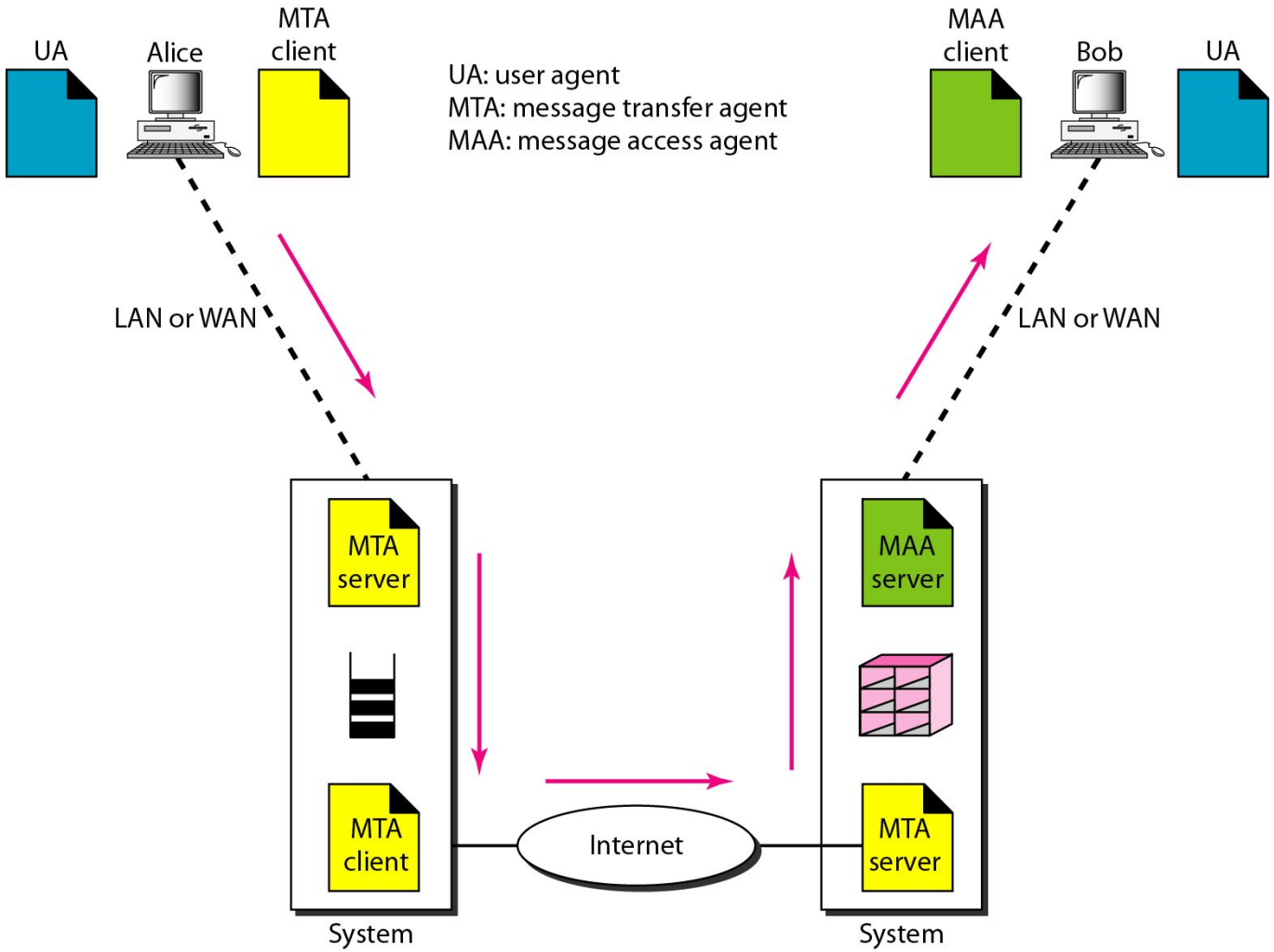
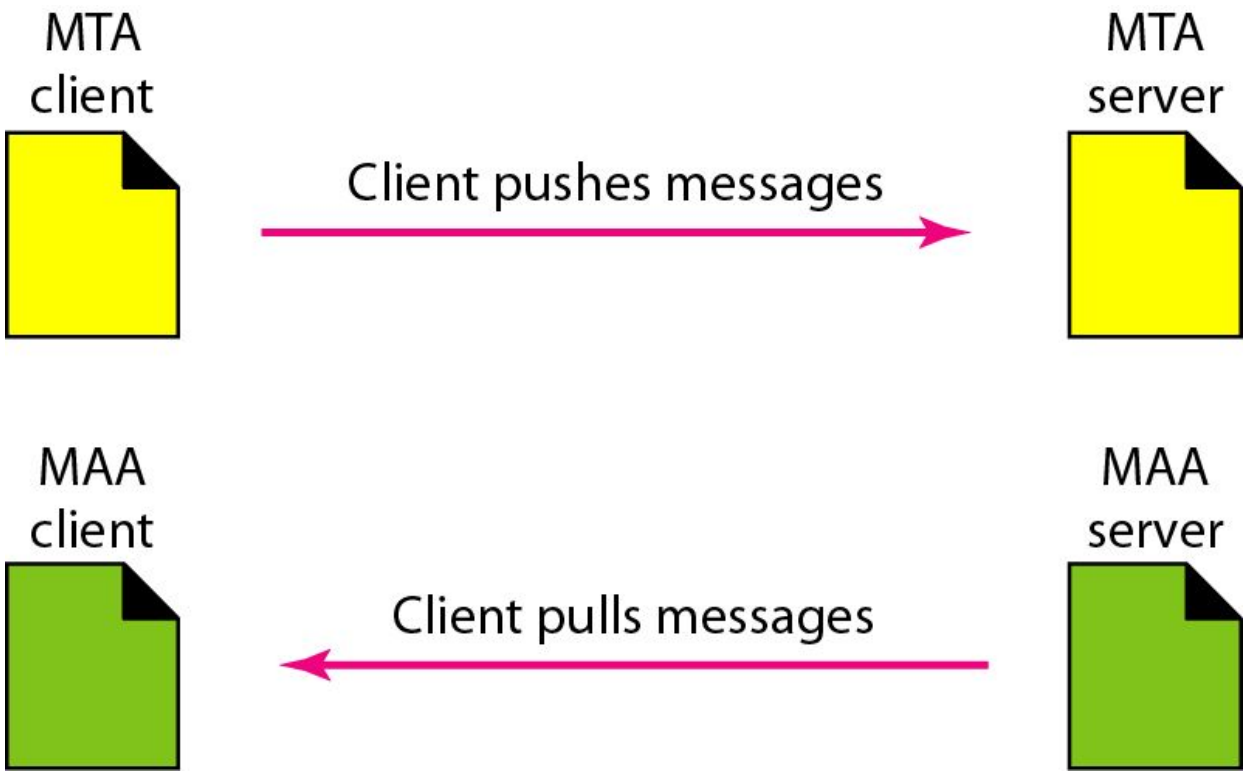


Figure 26.10 *Push versus pull in electronic email*

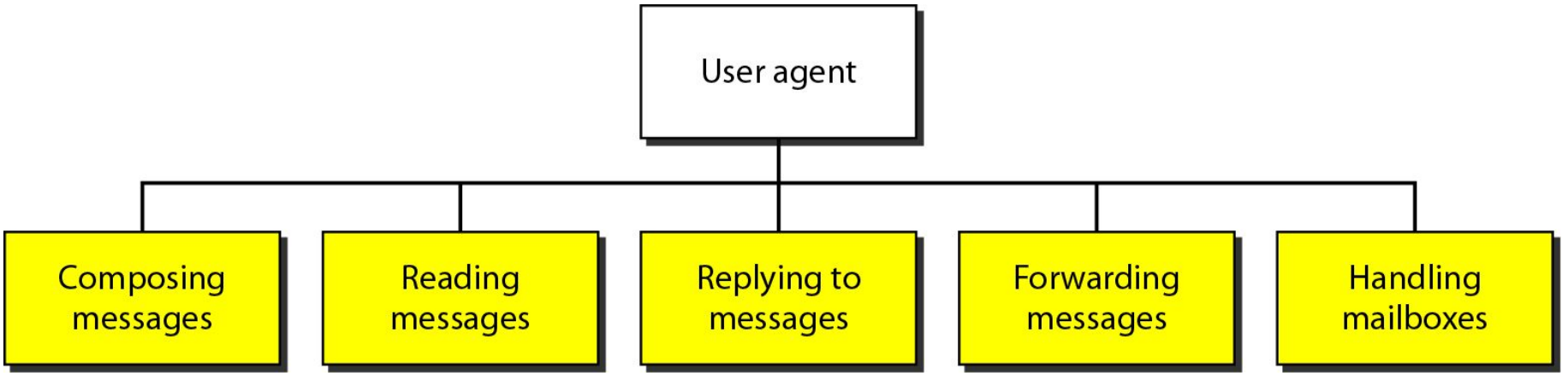


Note

When both sender and receiver are connected to the mail server via a LAN or a WAN, we need two UAs, two pairs of MTAs and a pair of MAAs.

This is the most common situation today.

Figure 26.11 *Services of user agent*



Note

Some examples of command-driven user agents are *mail*, *pine*, and *elm*.

Note

Some examples of GUI-based user agents are *Eudora*, *Outlook*, and *Netscape*.

Figure 26.12 *Format of an e-mail*

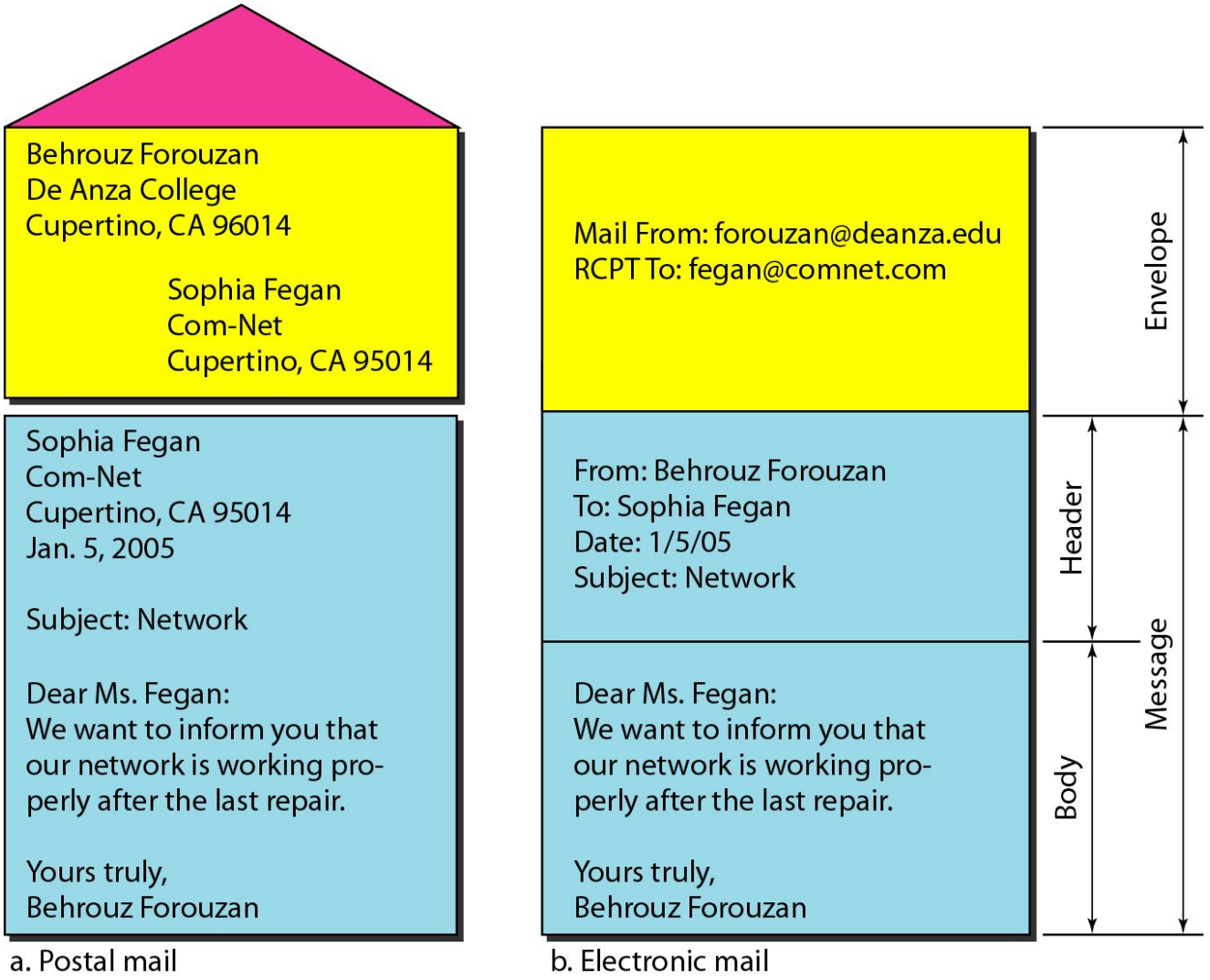


Figure 26.13 *E-mail address*

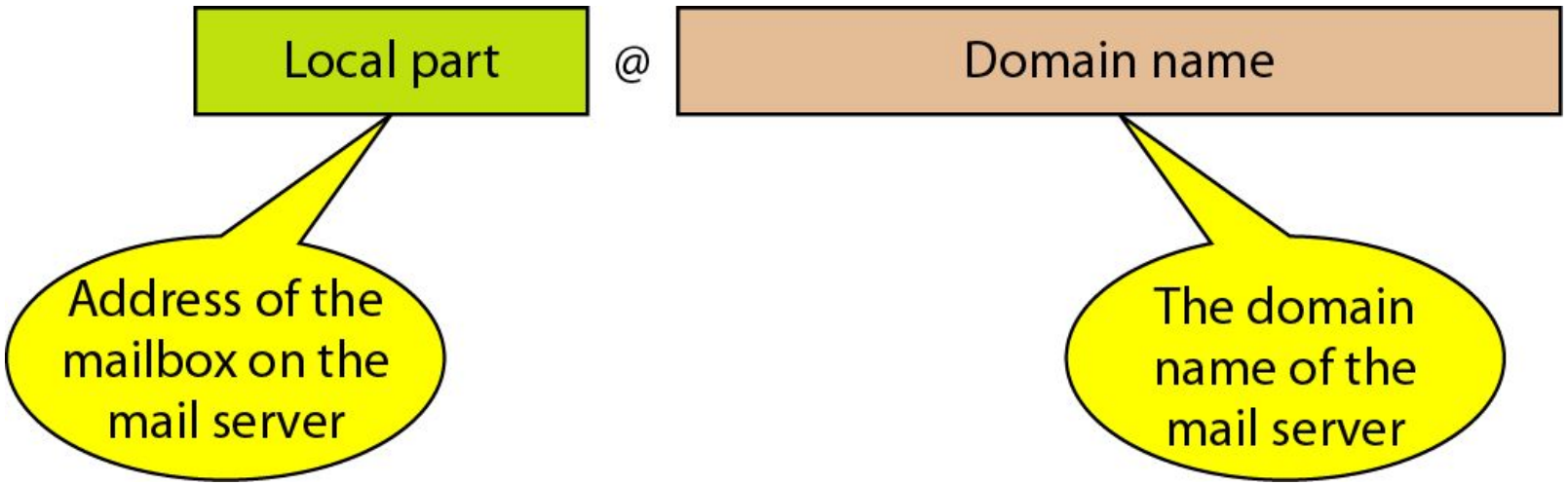


Figure 26.16 *SMTP range*

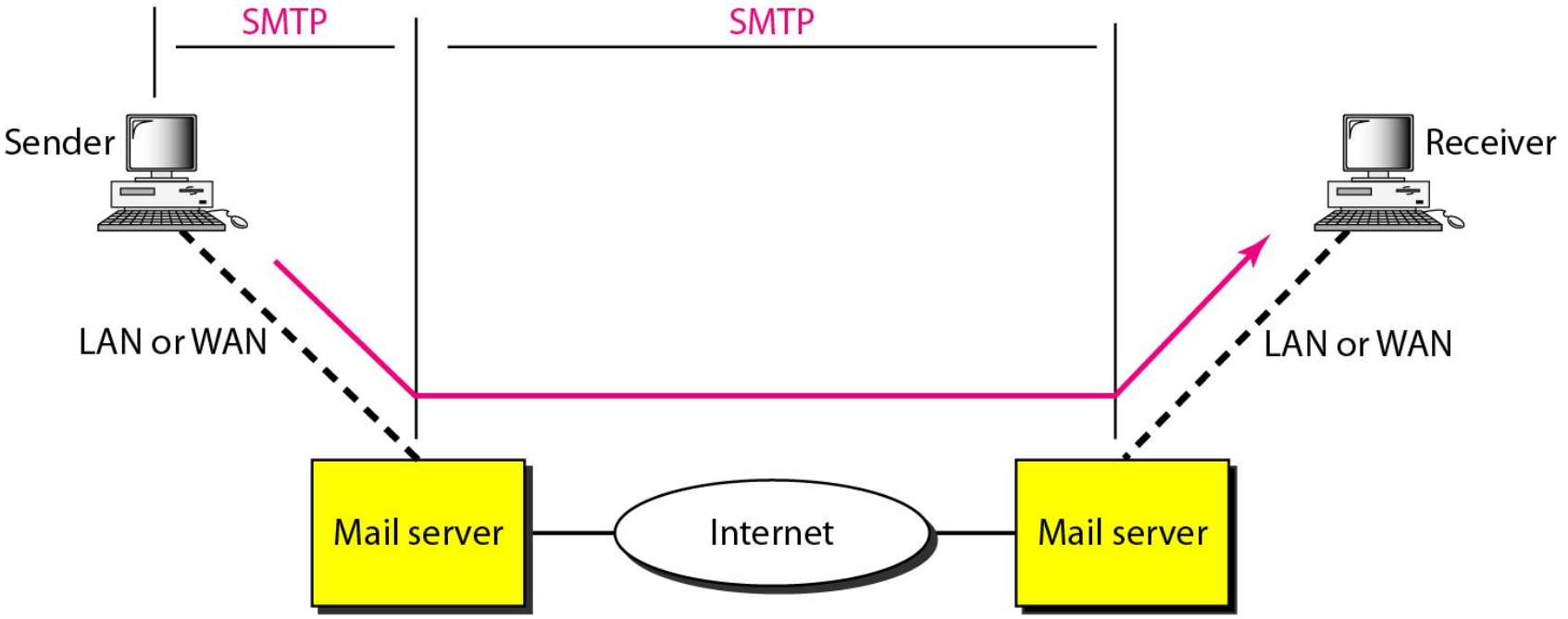


Figure 26.17 *SMTP uses Commands and responses*

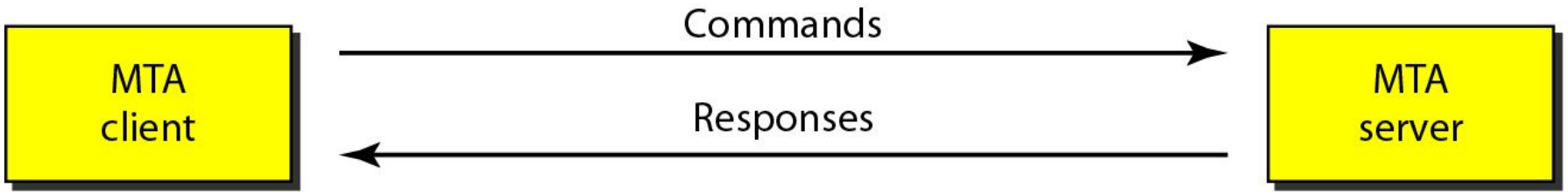


Table 26.7 *Commands*

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

Table 26.8 *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

Table 26.8 *Responses (continued)*

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

Figure 26.19 *Message Access Agents: POP3 and IMAP4*

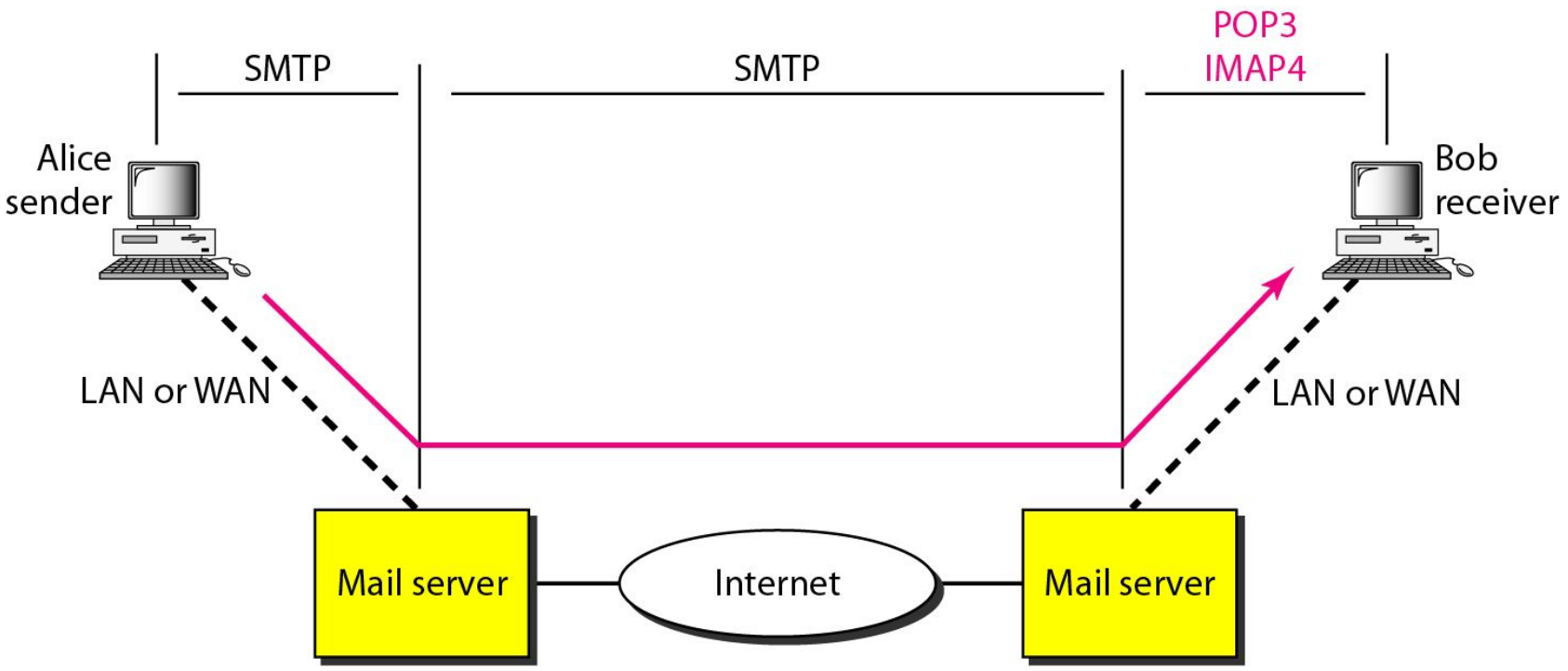


Figure 26.20 *The exchange of commands and responses in POP3*

26-3 FILE TRANSFER

Transferring files from one computer to another is one of the most common tasks expected from a networking or internetworking environment. As a matter of fact, the greatest volume of data exchange in the Internet today is due to file transfer.

File Transfer Protocol (FTP)

Note

FTP uses the services of TCP. It needs two TCP connections.

The well-known port 21 is used for the control connection and the well-known port 20 for the data connection.

Figure 26.21 *FTP*

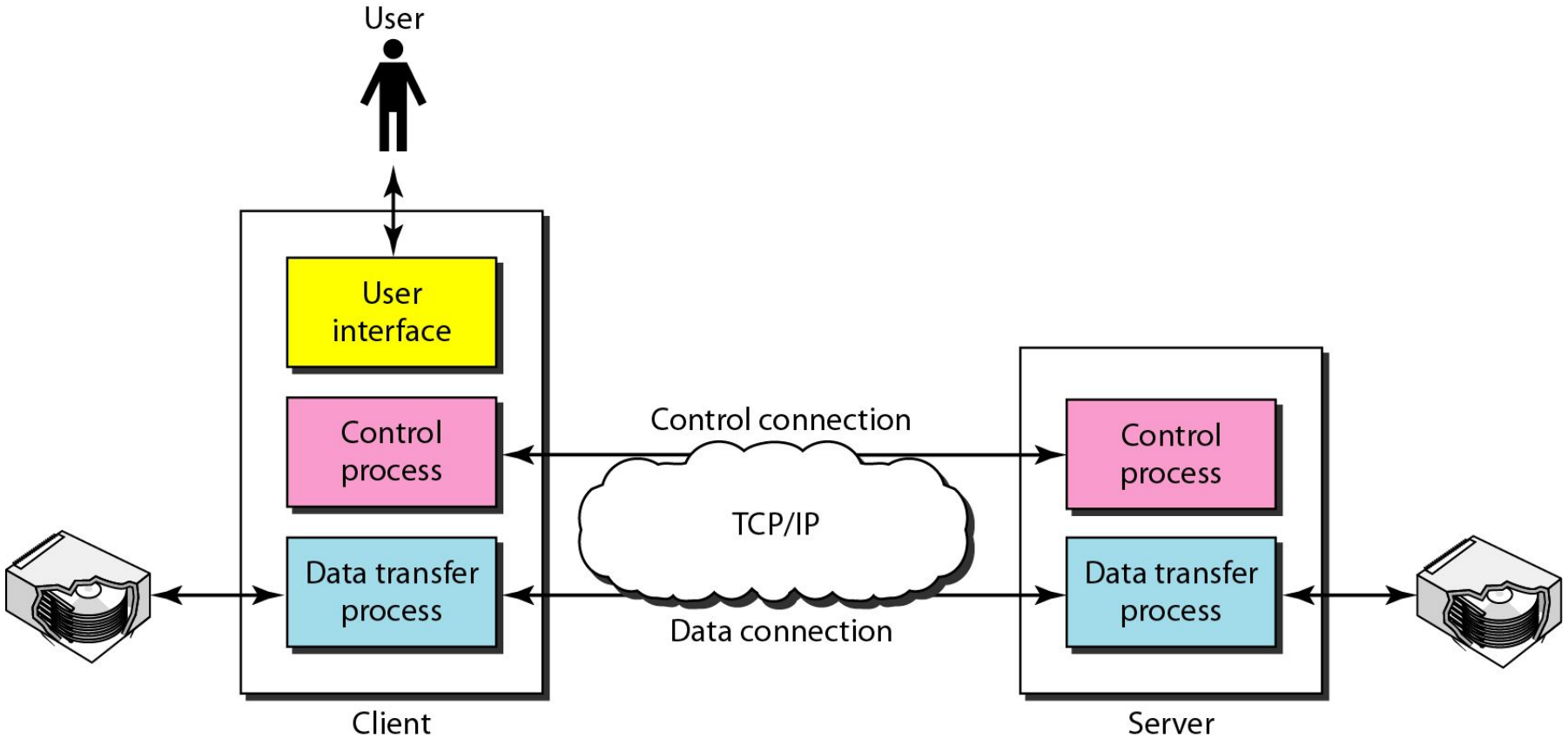


Figure 26.22 *Using the control connection*

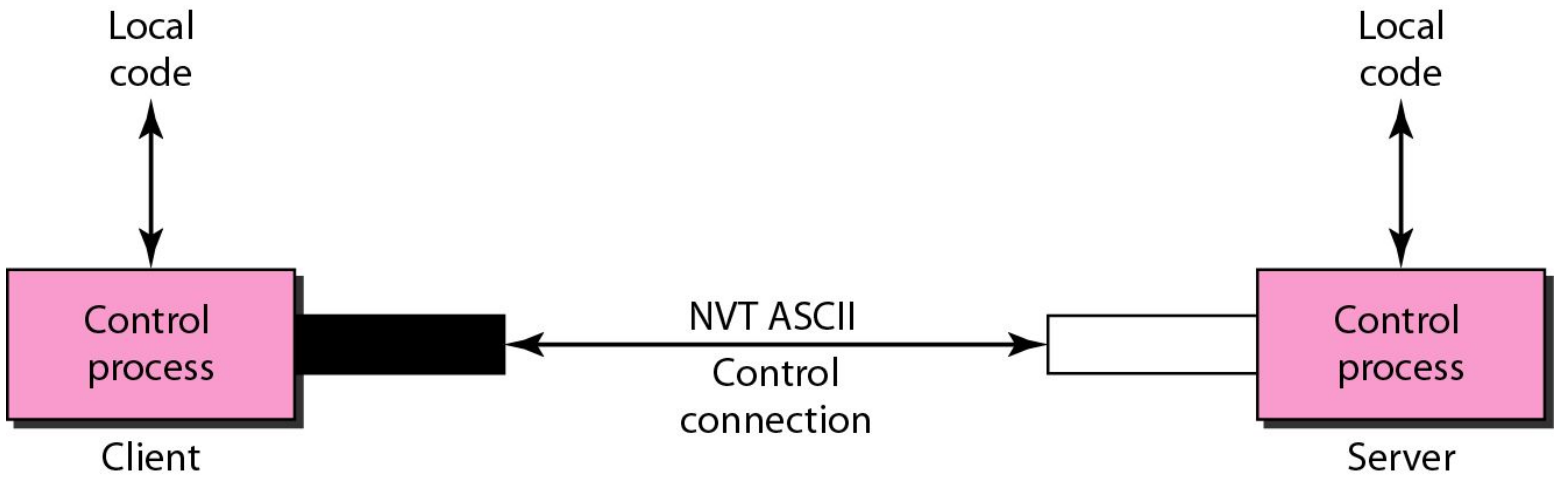


Figure 26.23 *Using the data connection*

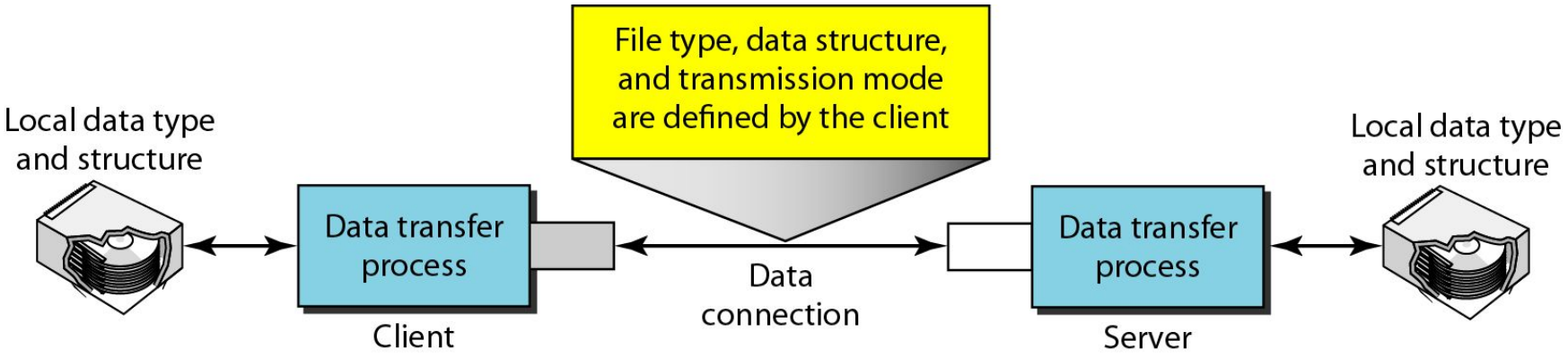


Figure
25-23

HTTP Transaction

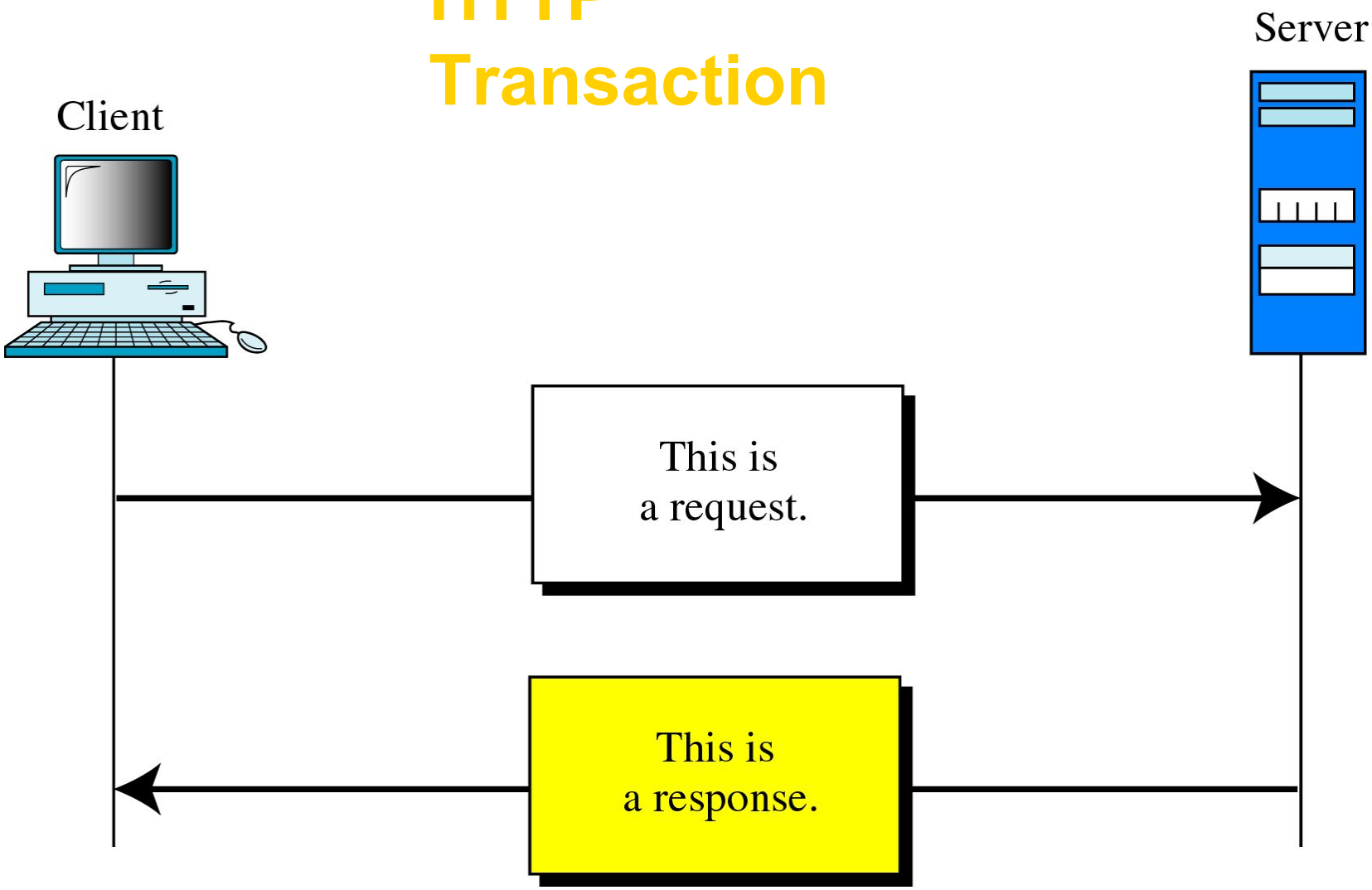


Figure
25-24

Message Categories

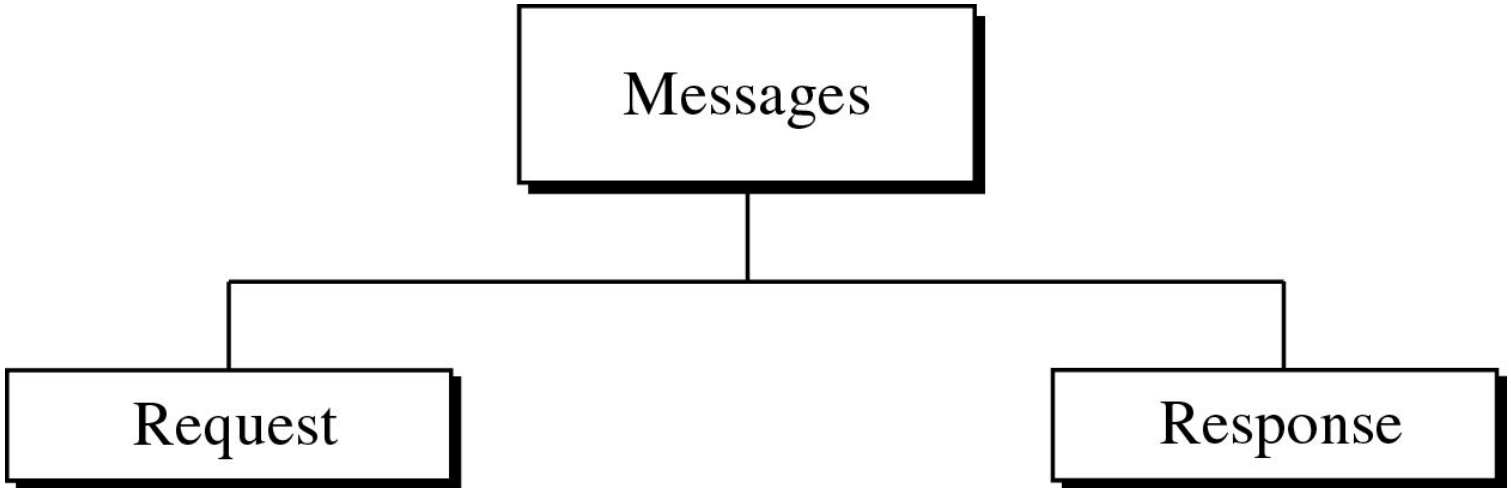
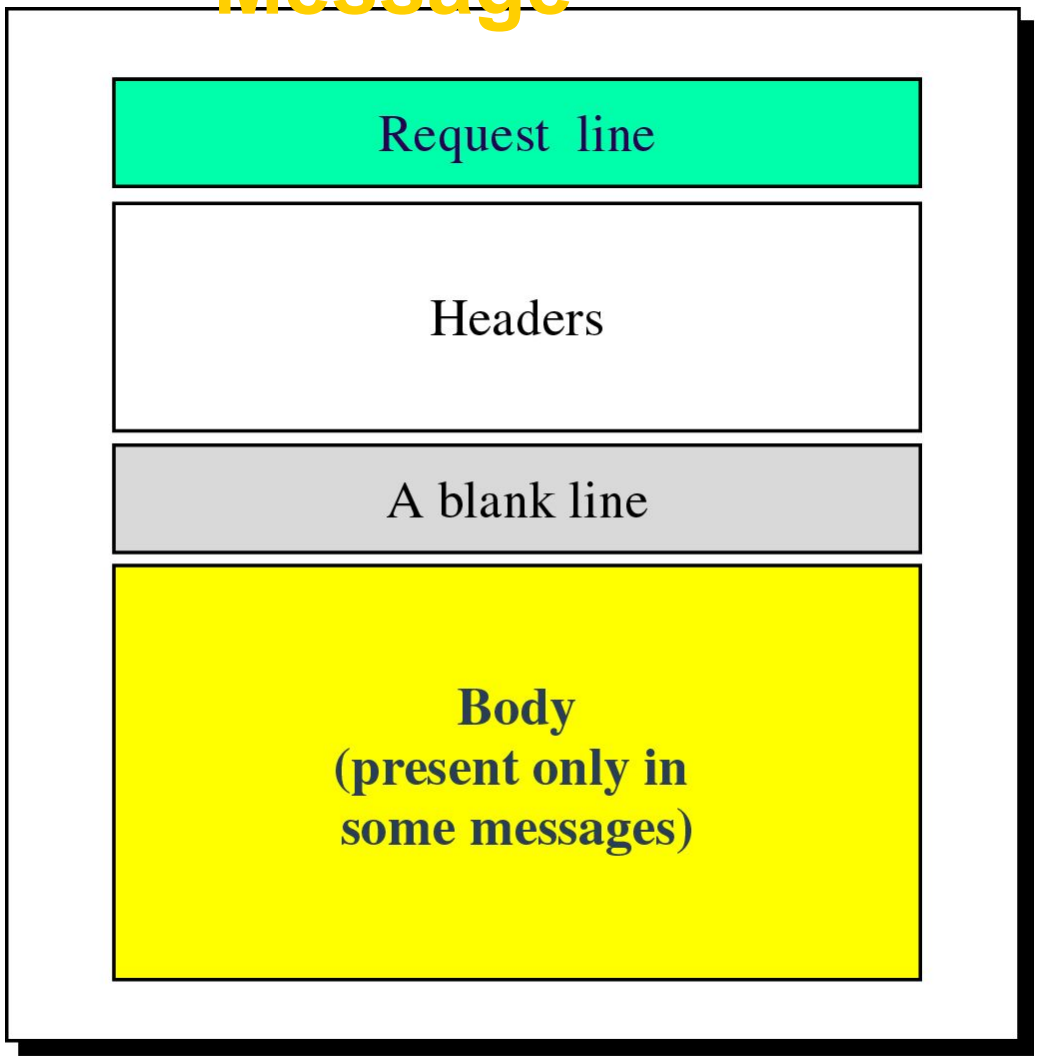


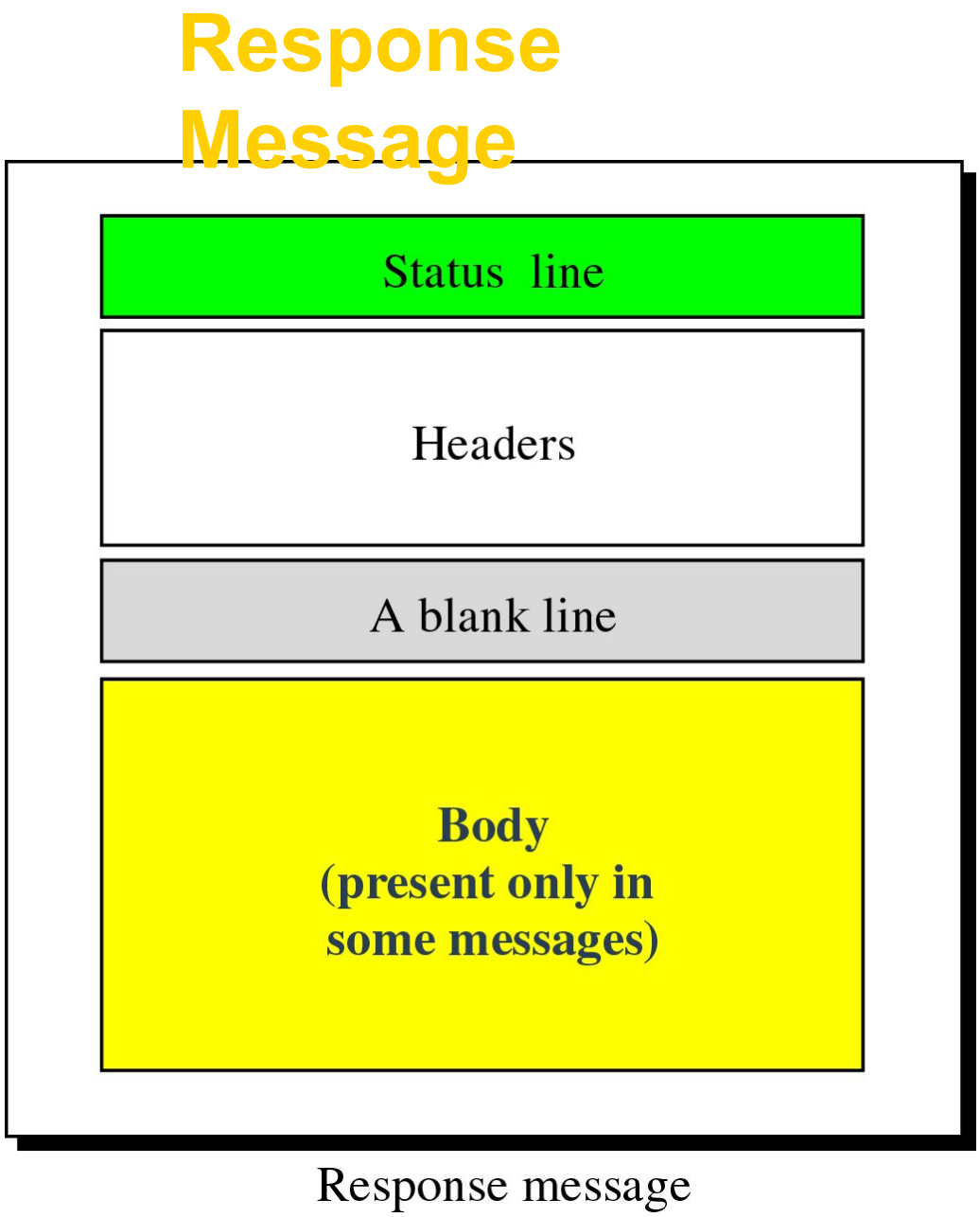
Figure
25-25

Request Message



Request message

Figure
25-26



Response message

Figure
25-27

URL

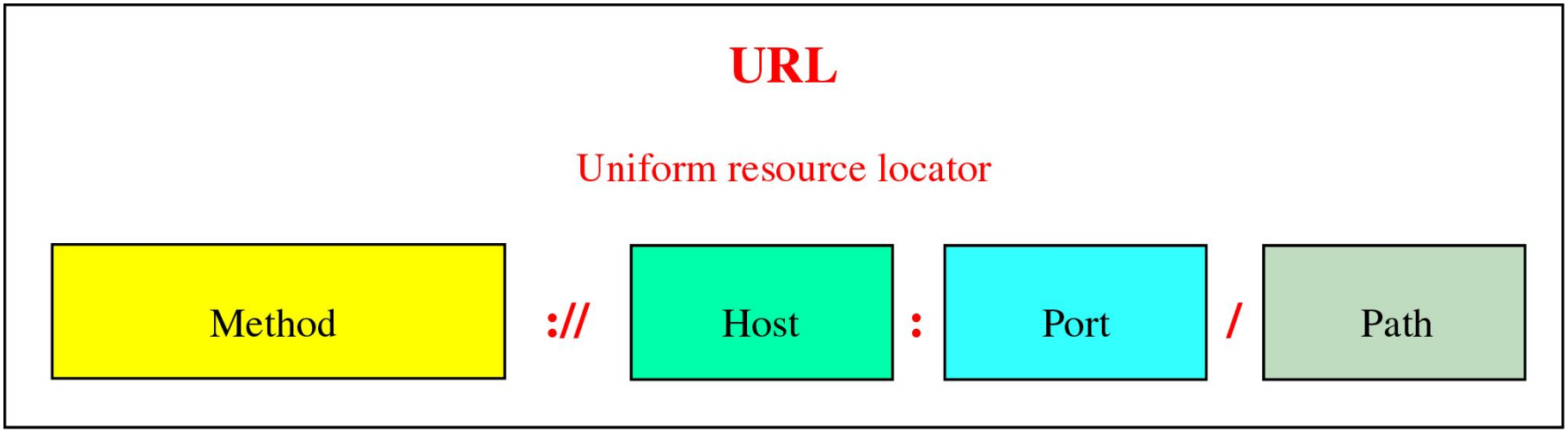


Figure
25-30

Browser Architecture

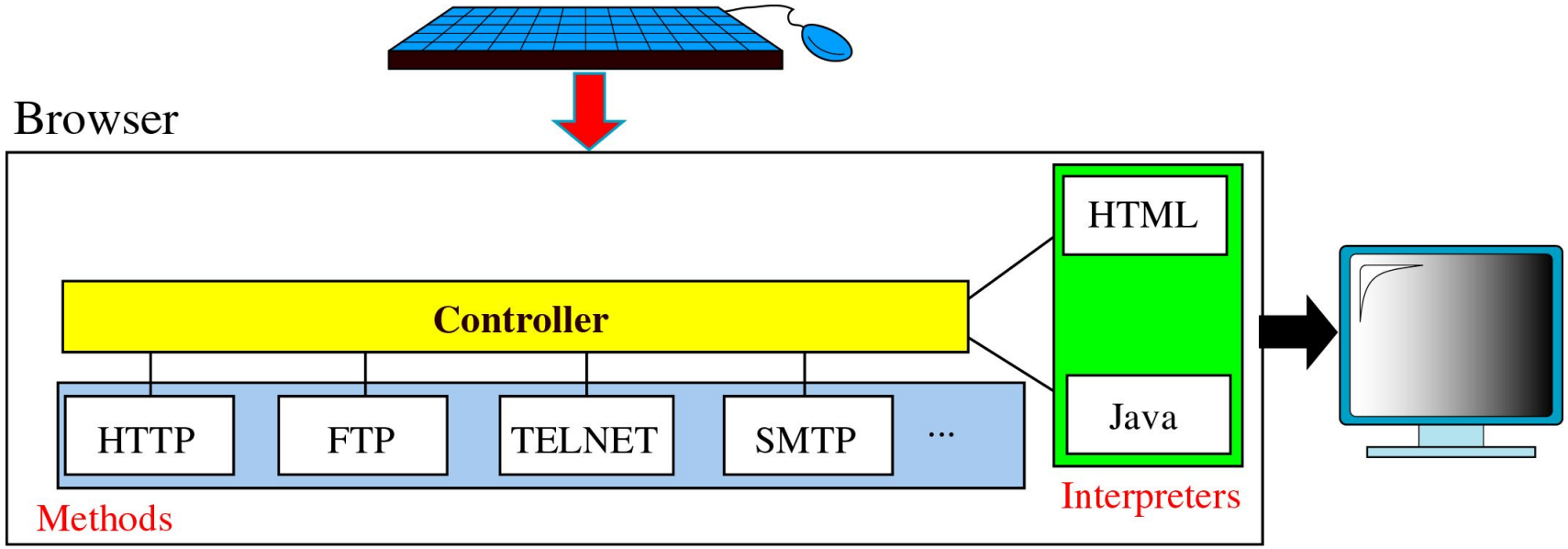


Figure
25-31

Categories of Web Documents

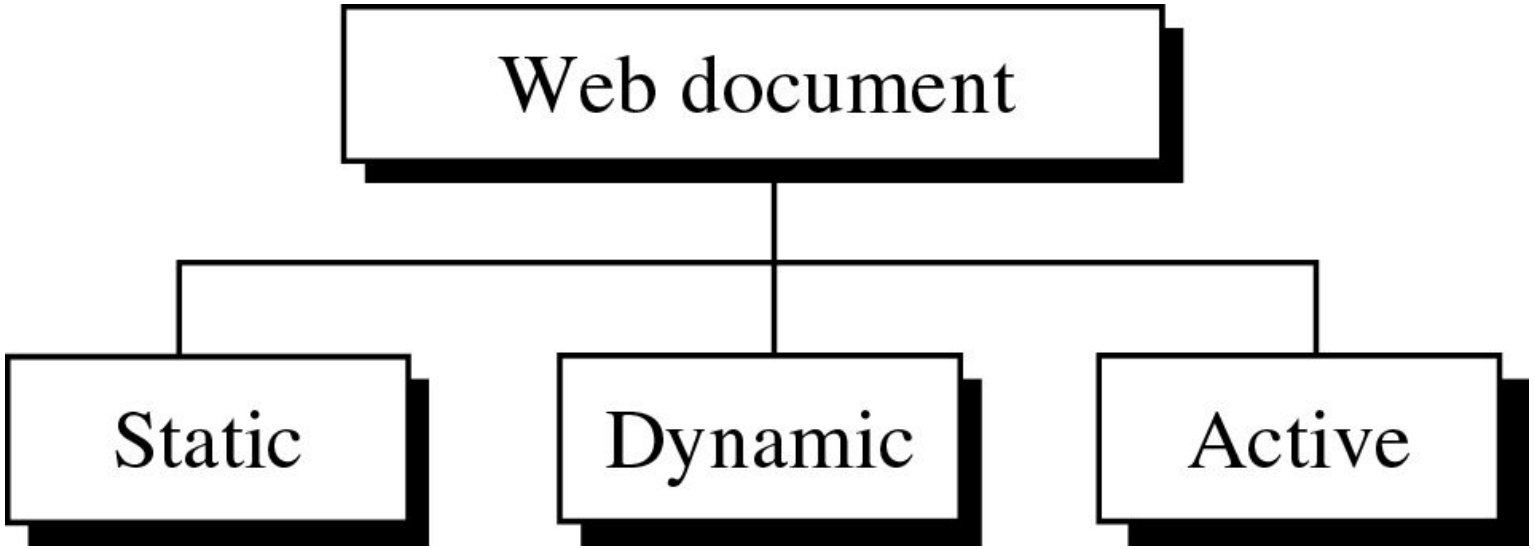
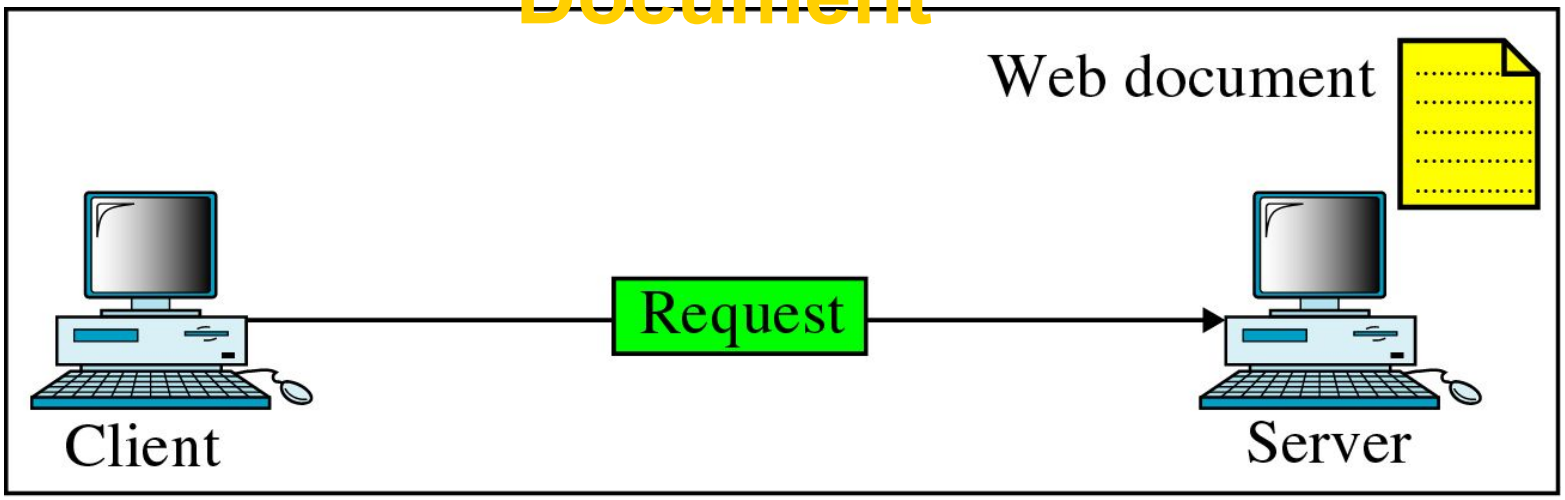
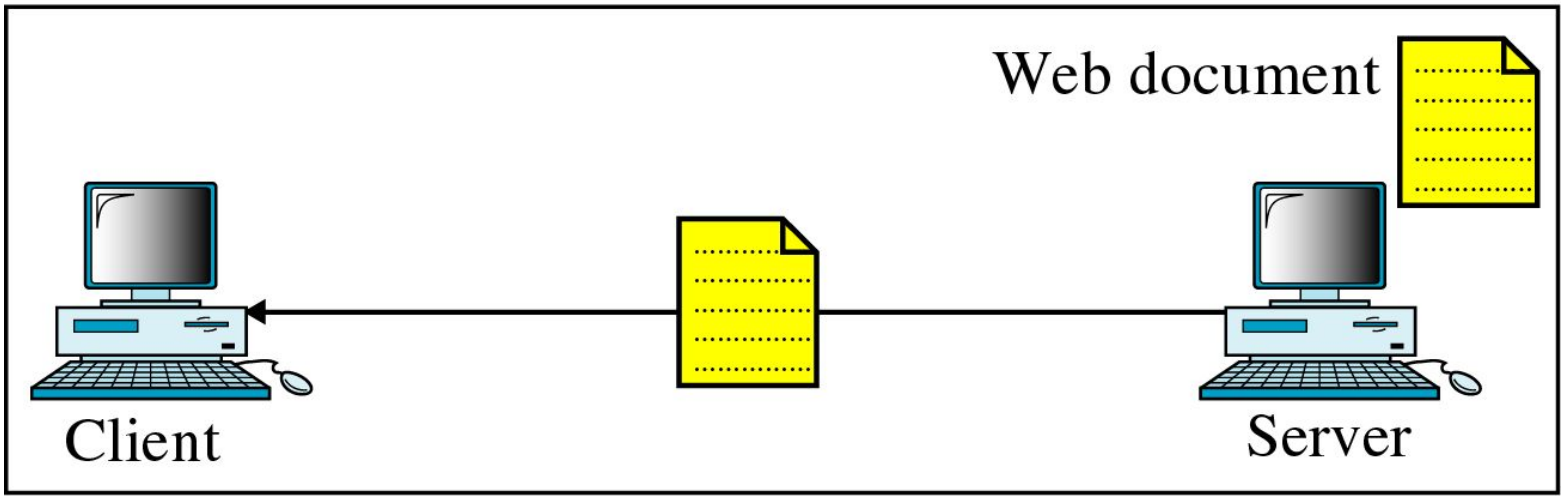


Figure 25-32

Static Document



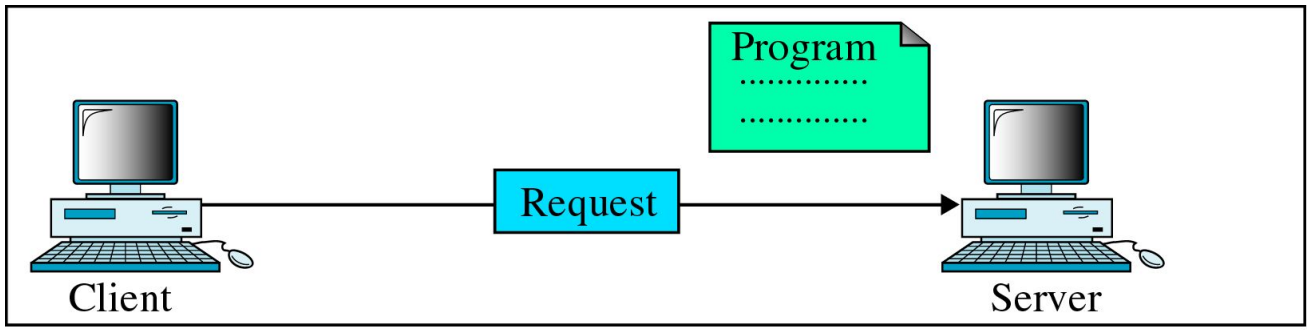
a. Request



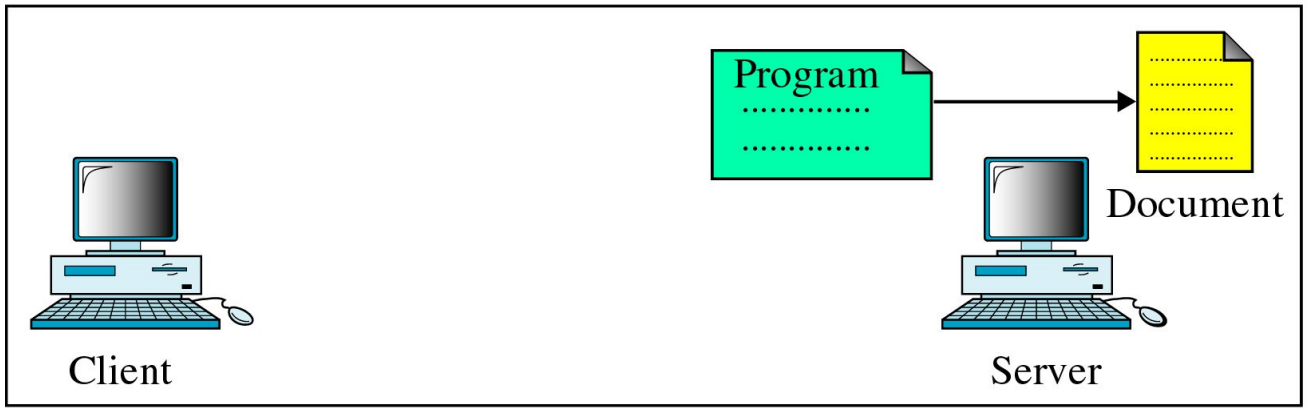
b. Response

Figure 25-36

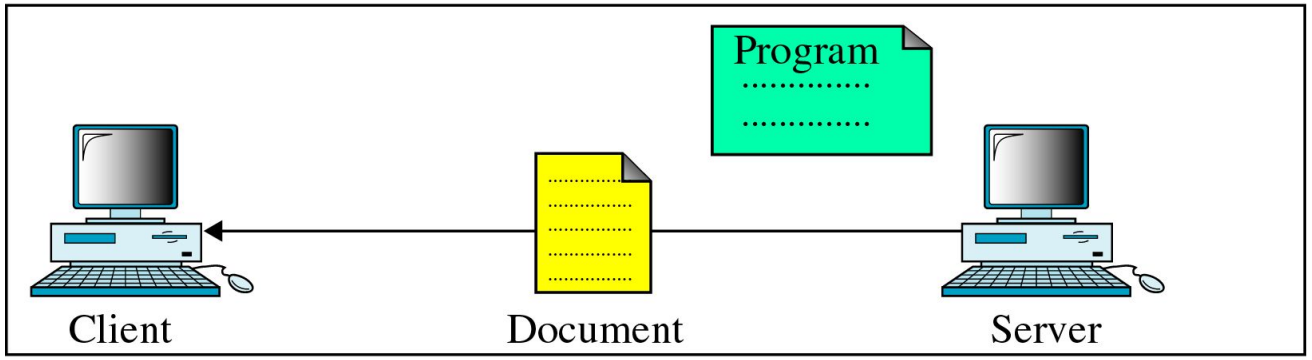
Dynamic Document



a. Request for running a program



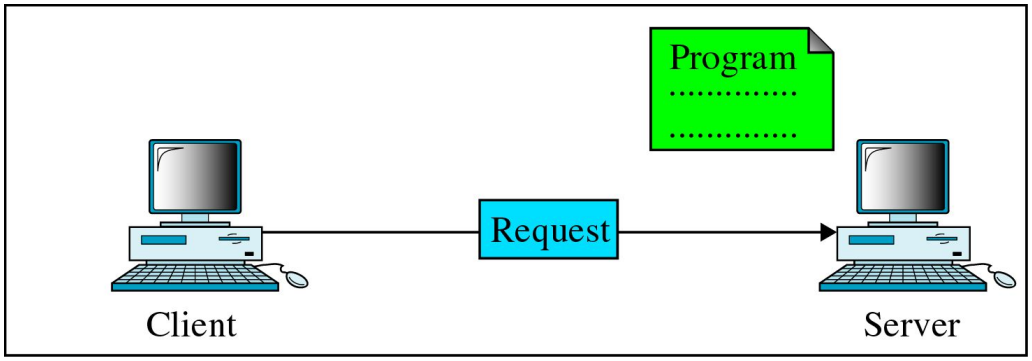
b. Running the program and creating the document



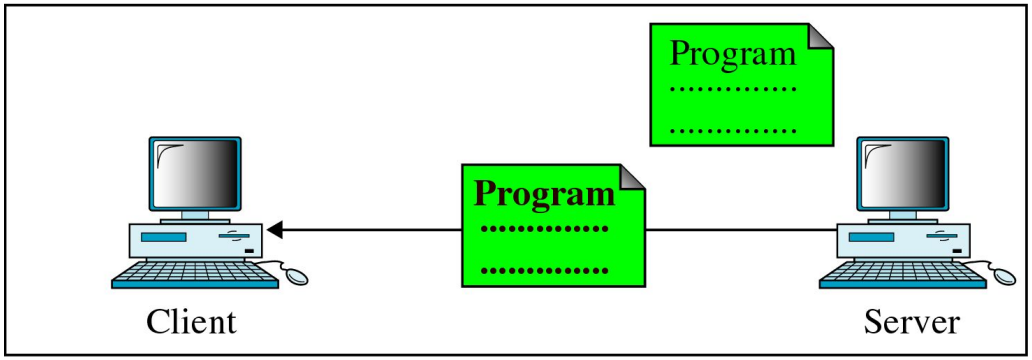
c. Response

Figure
25-37

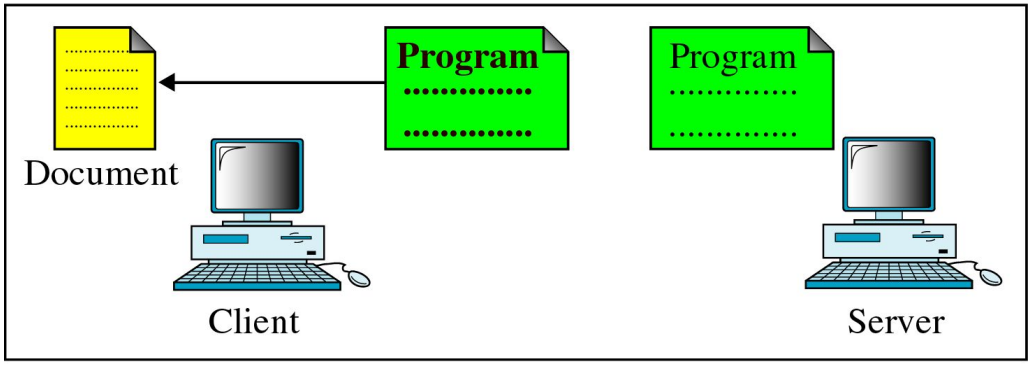
Active Document



a. Request for a copy of a program



b. Sending a copy of the program



c. Running the program and creating the document

25.1 Name Space

Flat Name Space

Hierarchical Name Space

25.2 Domain Name Space

Label

Domain Name

Figure 25.1 Domain name space

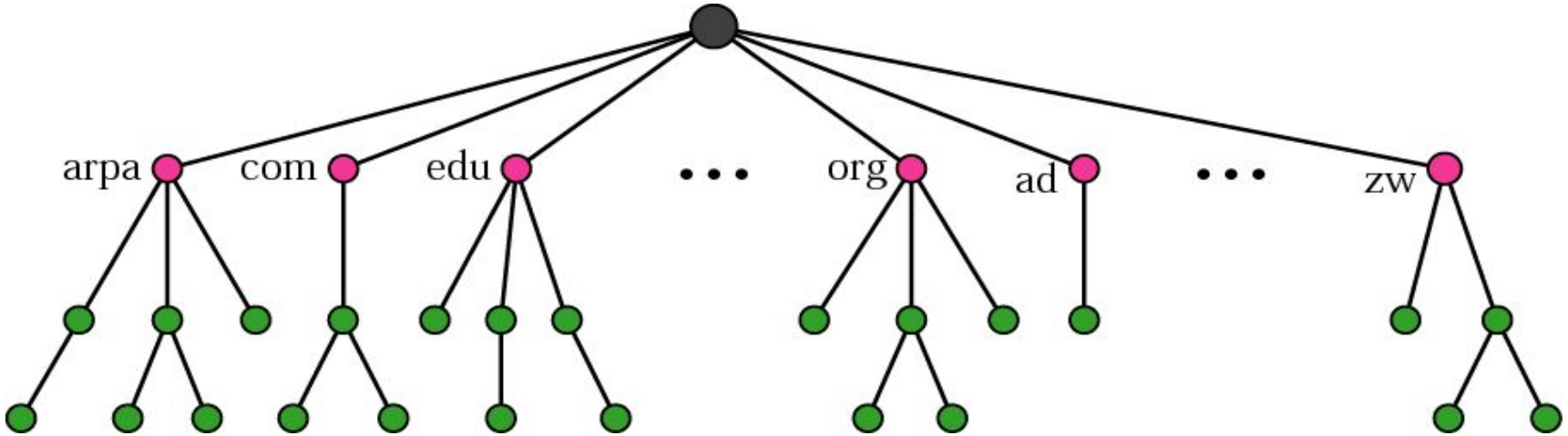


Figure 25.2 Domain names and labels

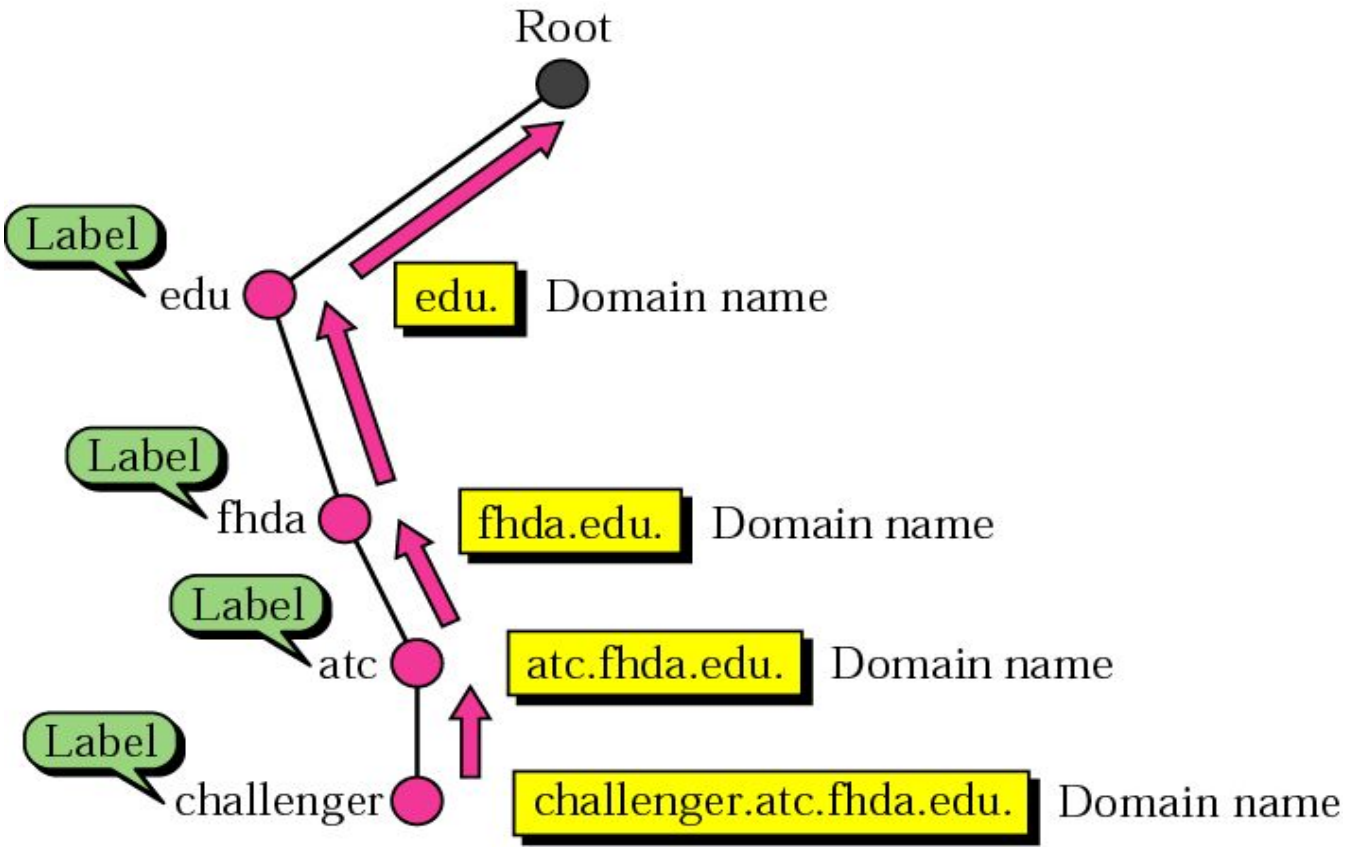


Figure 25.4 Domains

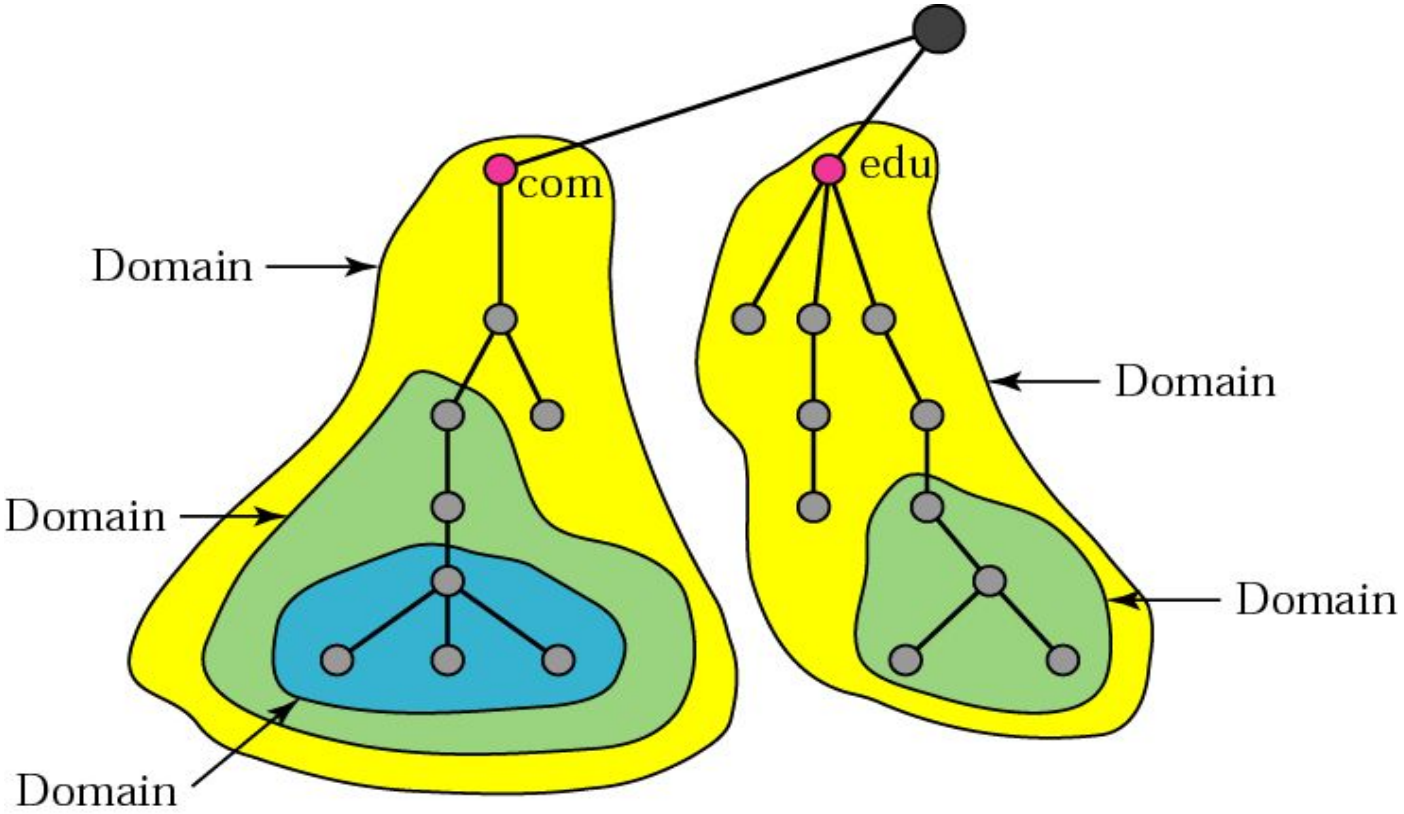


Figure 25.5 Hierarchy of name servers

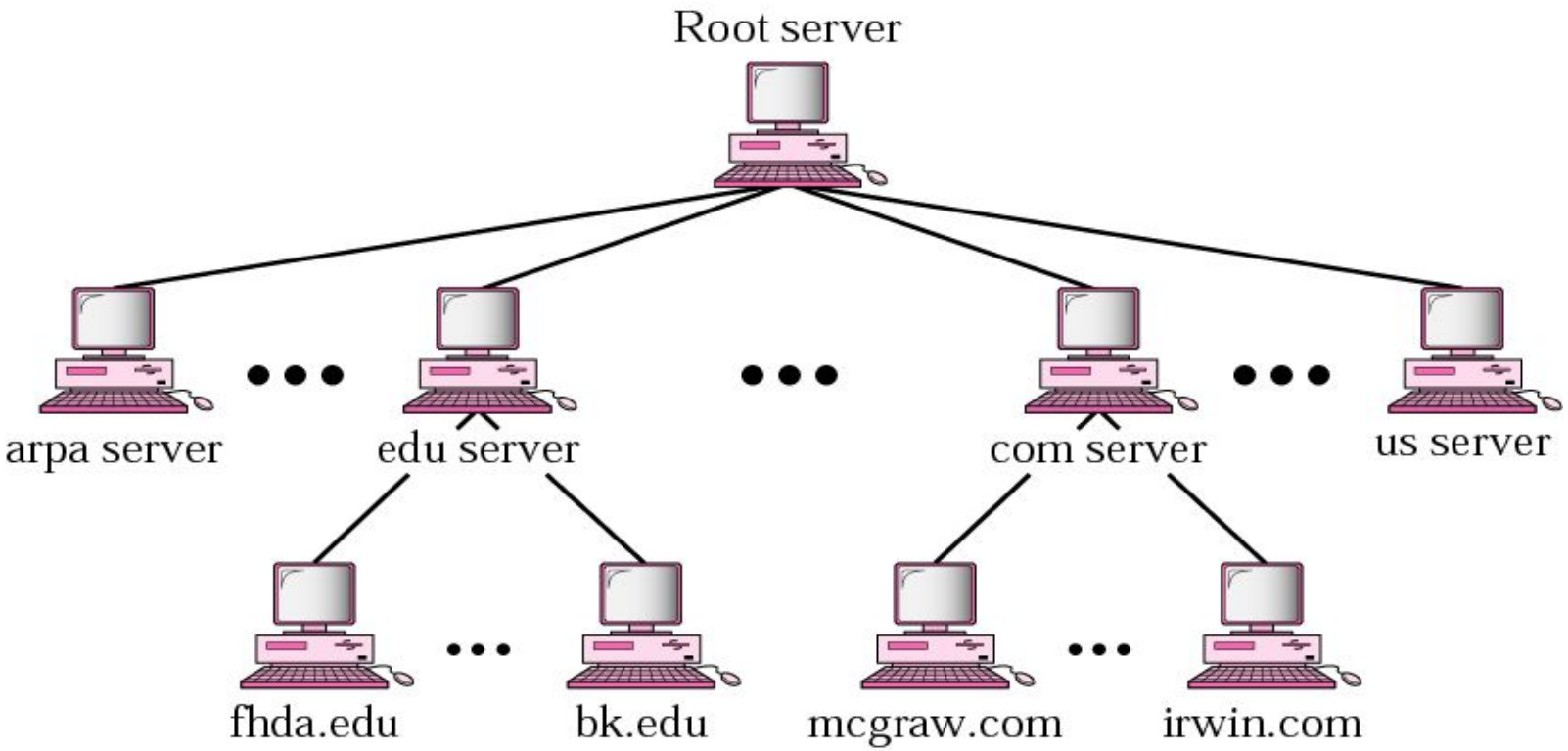
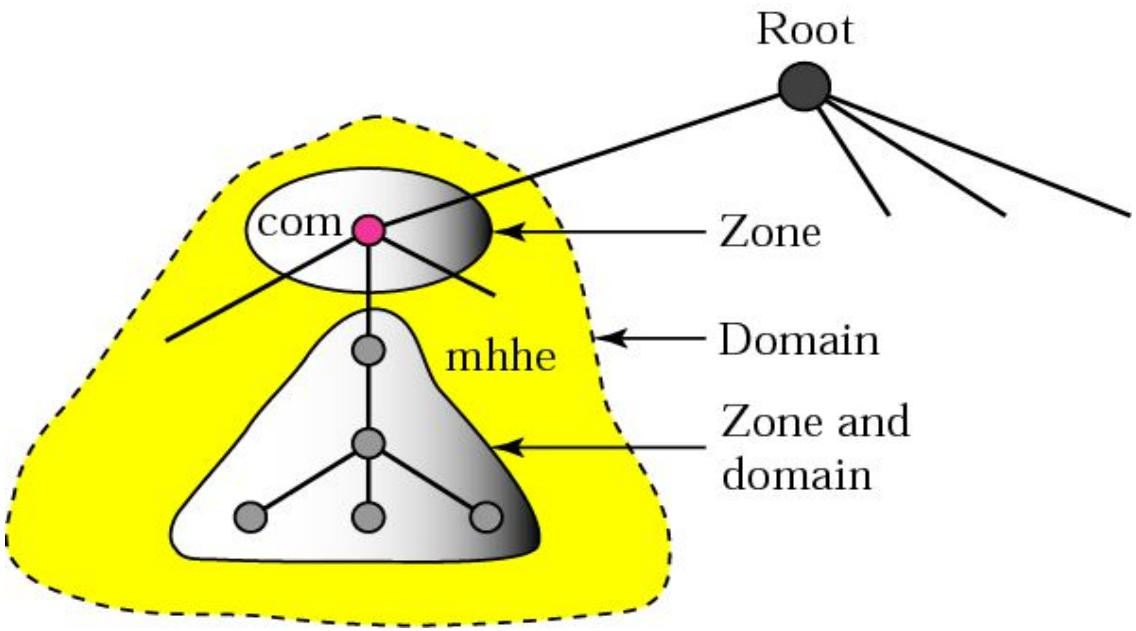


Figure 25.6 Zones and domains



Note:

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

25.4 DNS In The Internet

Generic Domain

Country Domain

Inverse Domain

Figure 25.7 DNS in the Internet

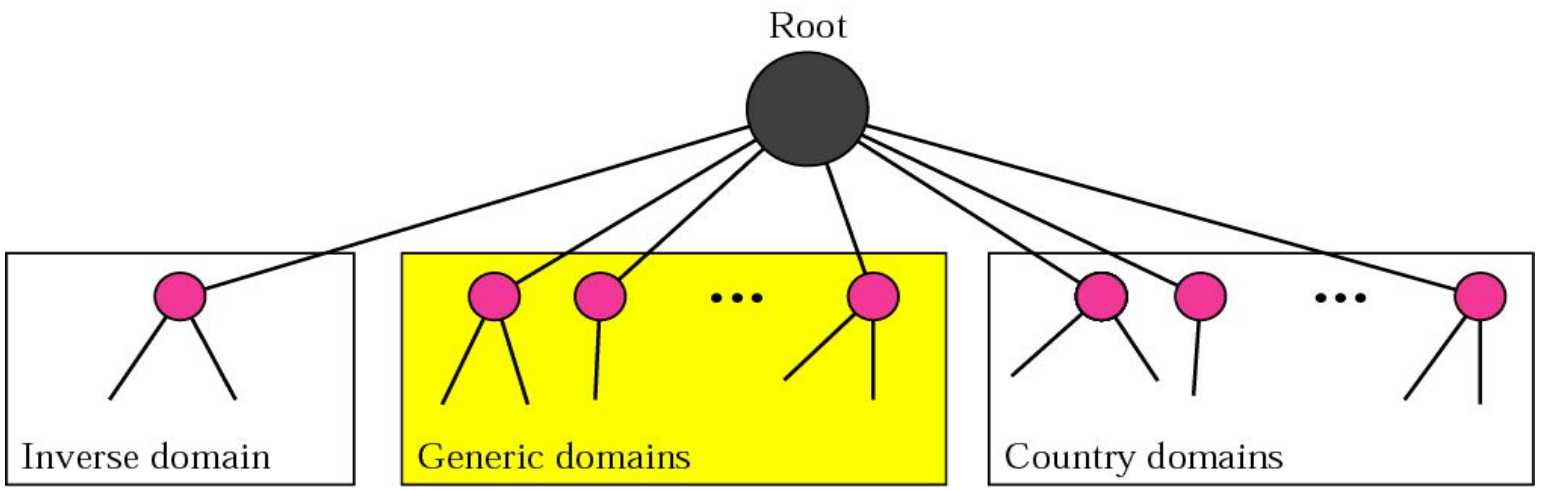


Figure 25.8 Generic domains

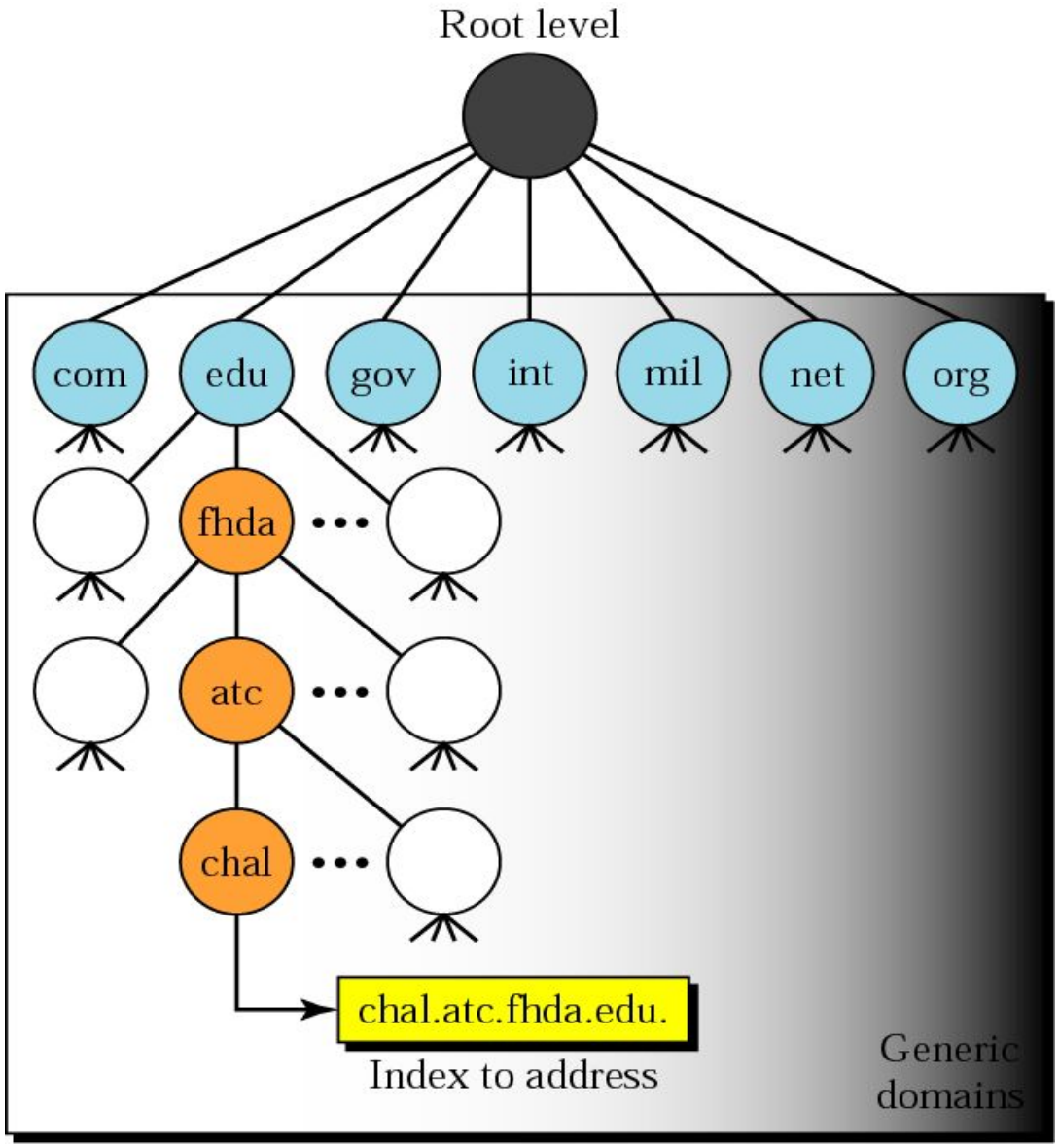


Table 25.1 Generic domain labels

Label	Description
com	Commercial organizations
edu	Educational institutions
gov	Government institutions
int	International organizations
mil	Military groups
net	Network support centers
org	Nonprofit organizations

Table 25.2 New generic domain labels

Label	Description
aero	Airlines and aerospace companies
biz	Businesses or firms (similar to com)
coop	Cooperative business organizations
info	Information service providers
museum	Museums and other nonprofit organizations
name	Personal names (individuals)
pro	Professional individual organizations

Figure 25.9 Country domains

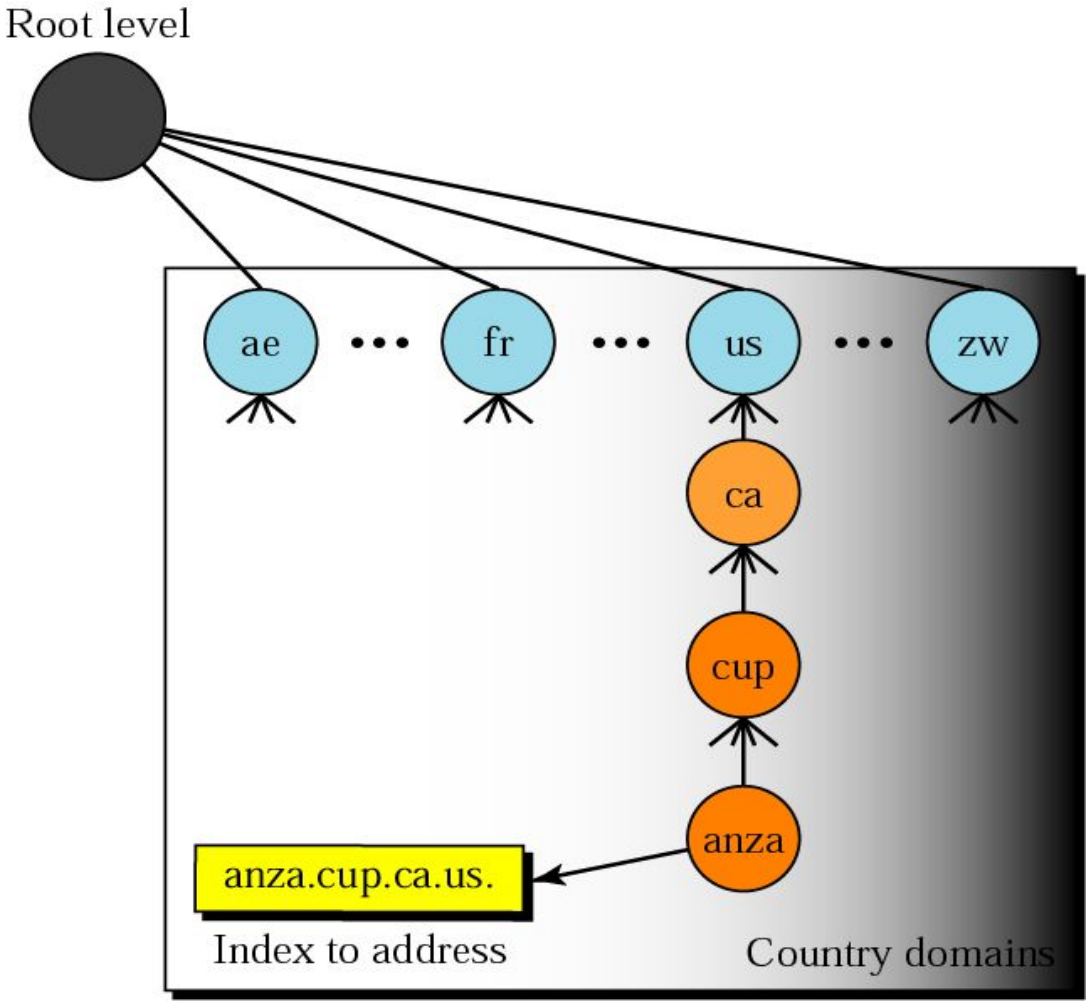
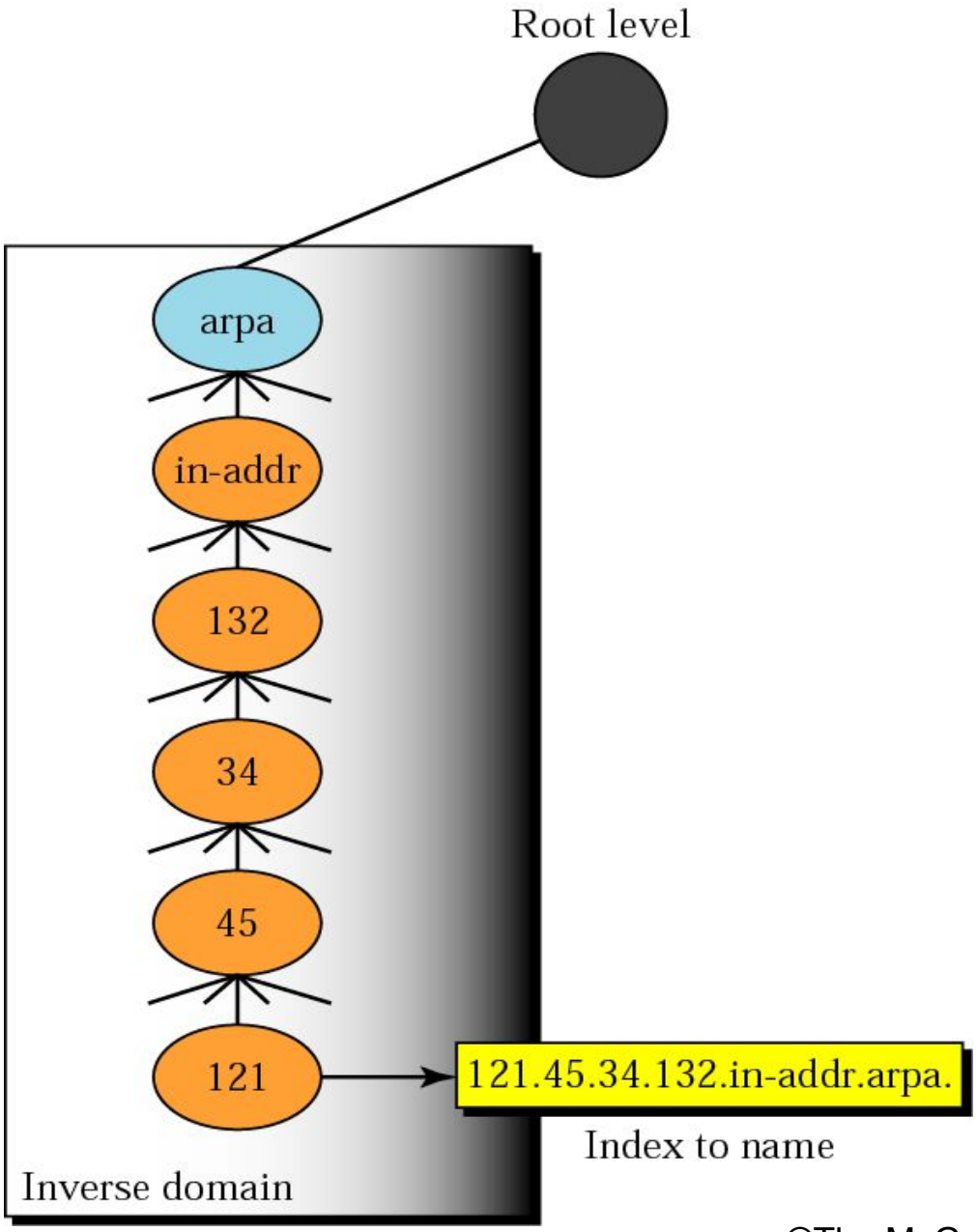


Figure 25.10 Inverse domain



25.5 Resolution

Resolver

Mapping Names to Addresses

Mapping Addresses to Names

Recursive Resolution

Iterative Resolution

Caching

Figure 25.11 Recursive resolution

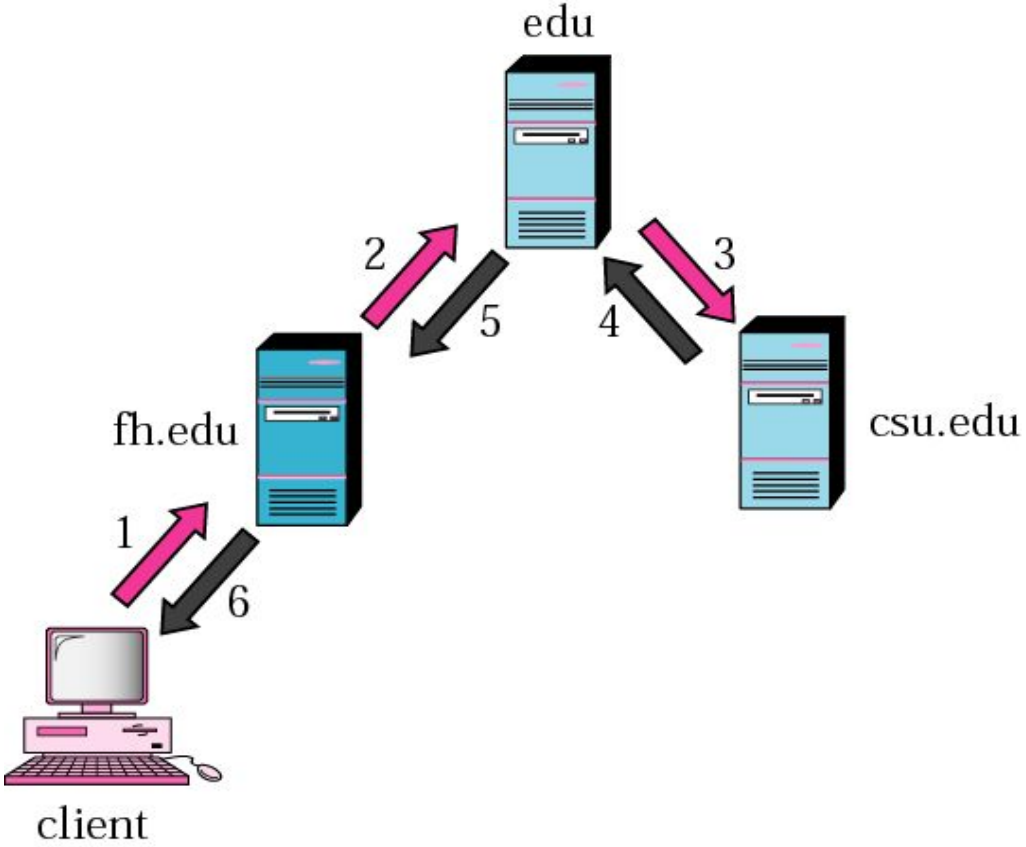


Figure 25.12 Iterative resolution

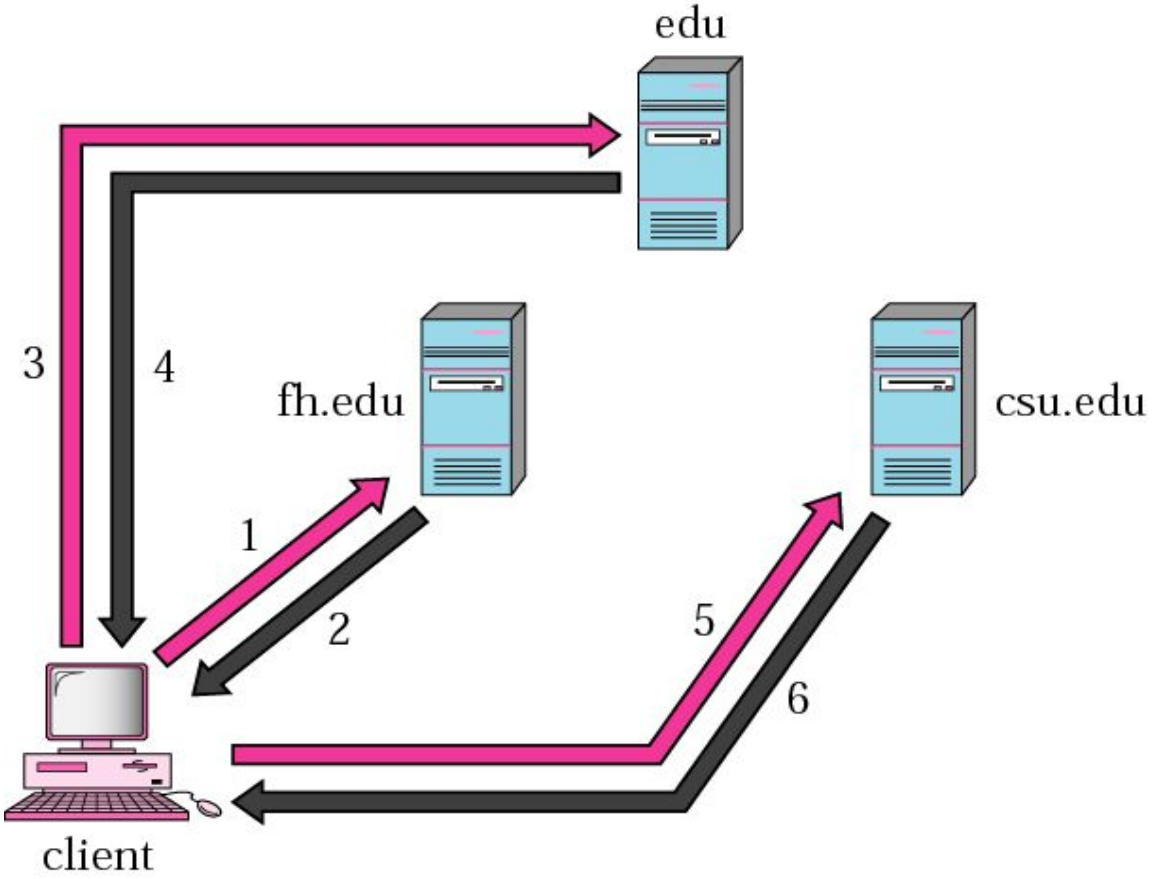


Figure 25.13 Query and response messages

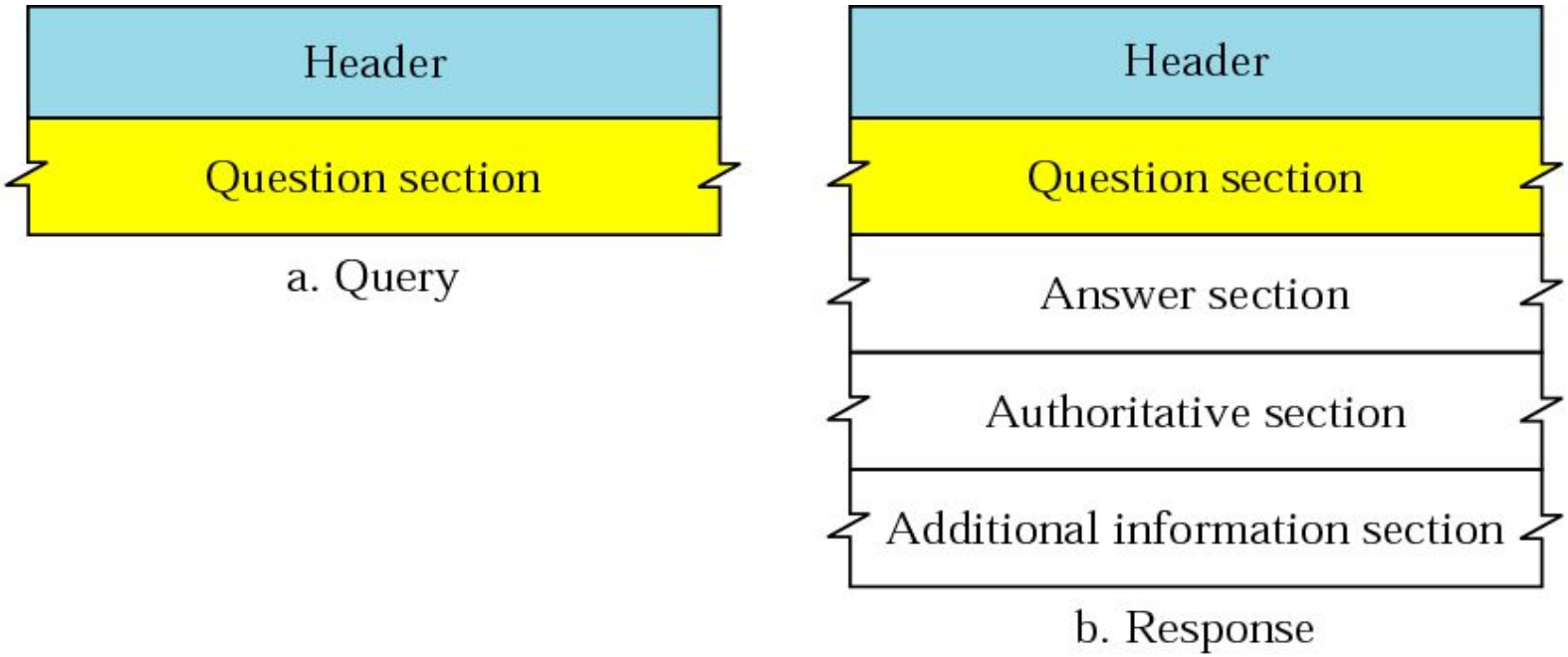
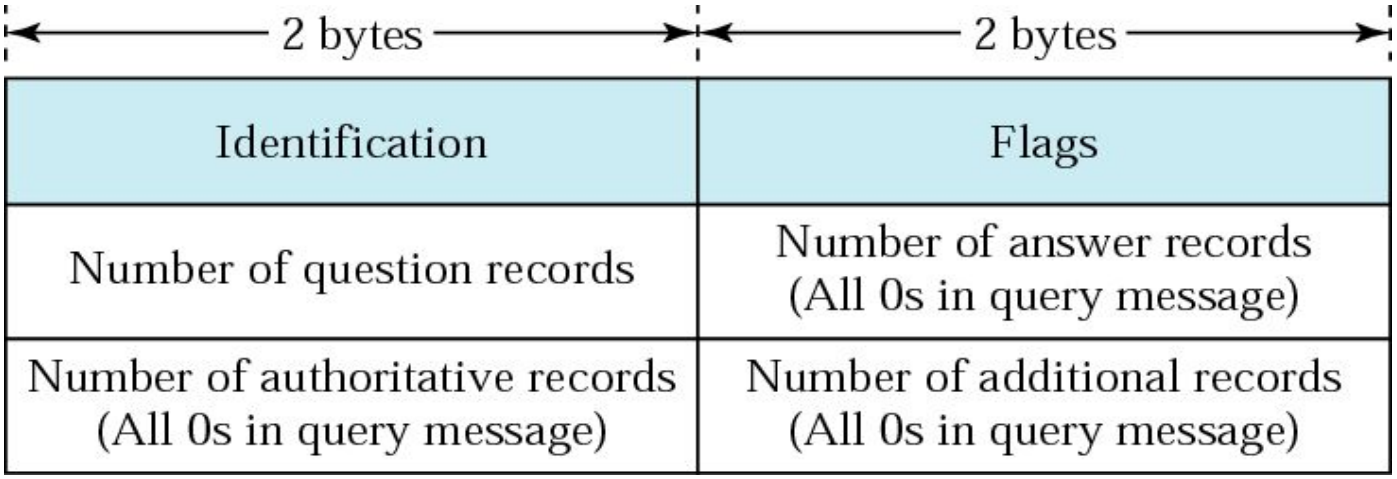


Figure 25.14 Header format



DNS can use the services of UDP or TCP, using the well-known port 53.

26.1 Electronic Mail

Sending/Receiving Mail

Addresses

User Agent

MIME

Mail Transfer Agent

Mail Access Protocols

Figure 26.1 Format of an email

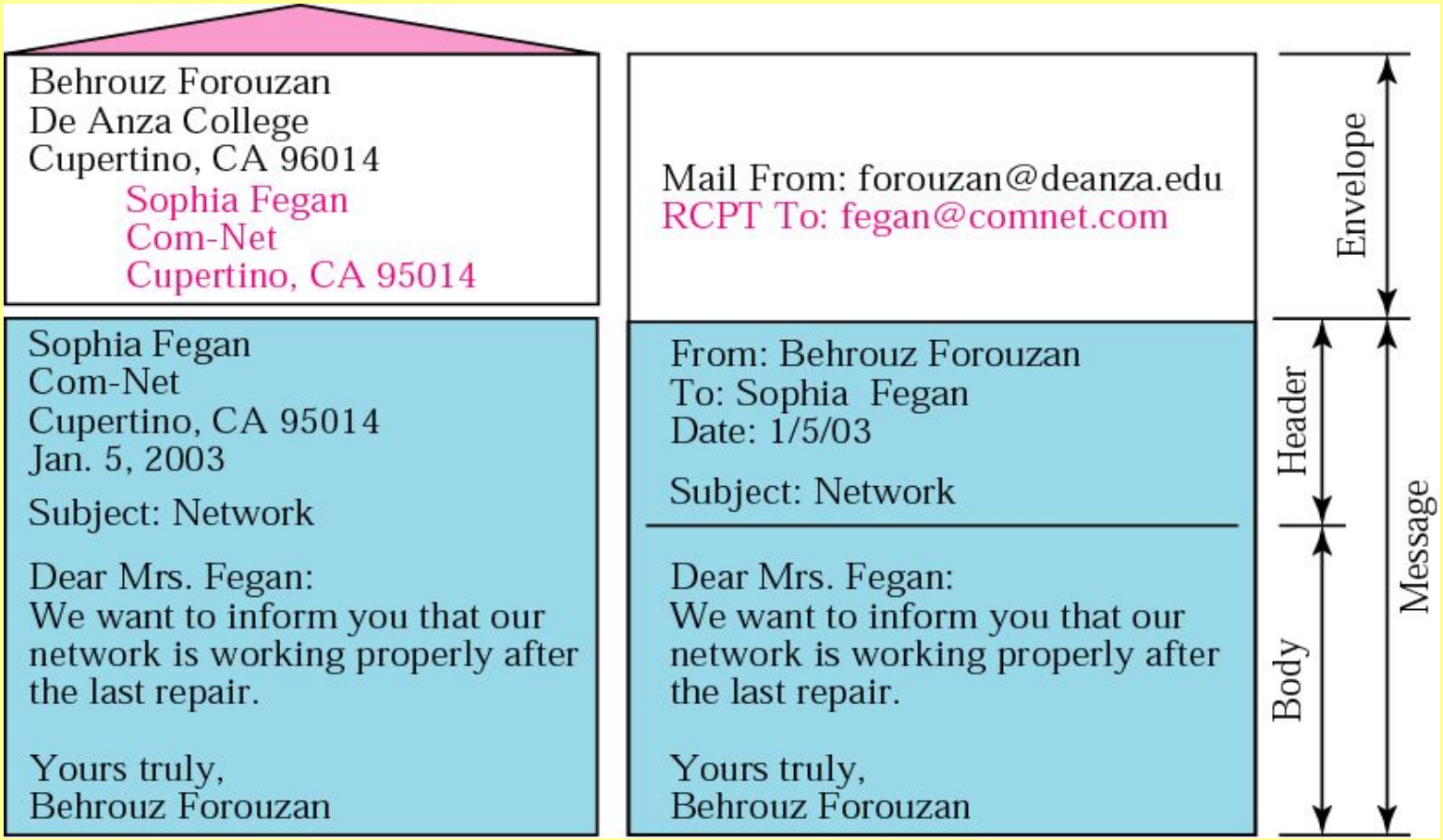


Figure 26.2 Email address

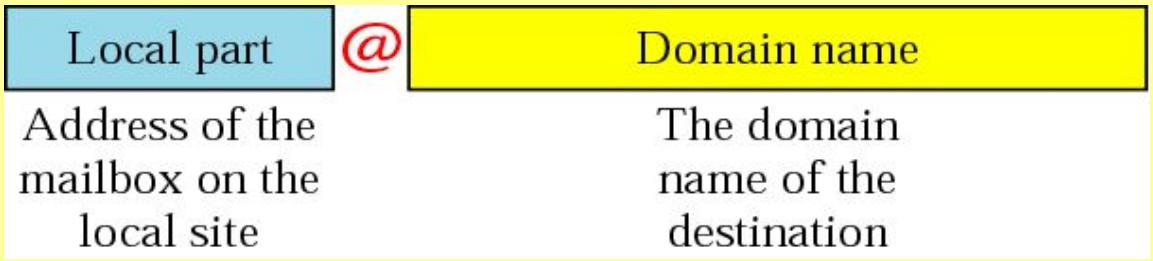


Figure 26.8 MTA client and server

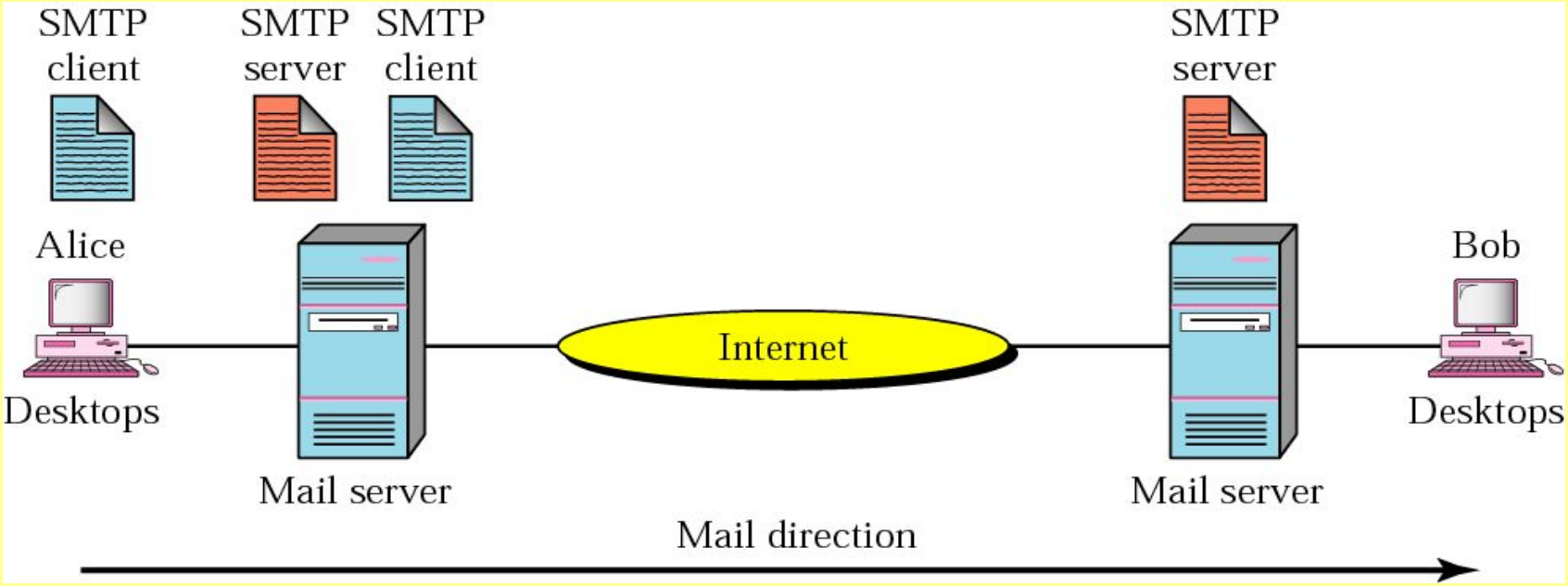


Figure 26.9 Commands and responses

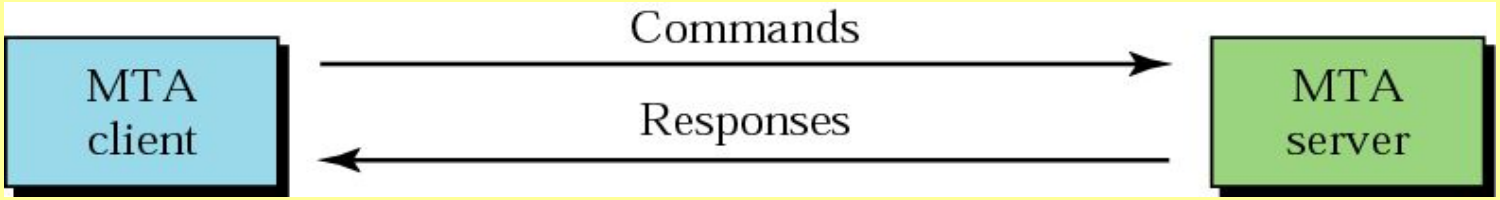


Figure 26.10 Email delivery

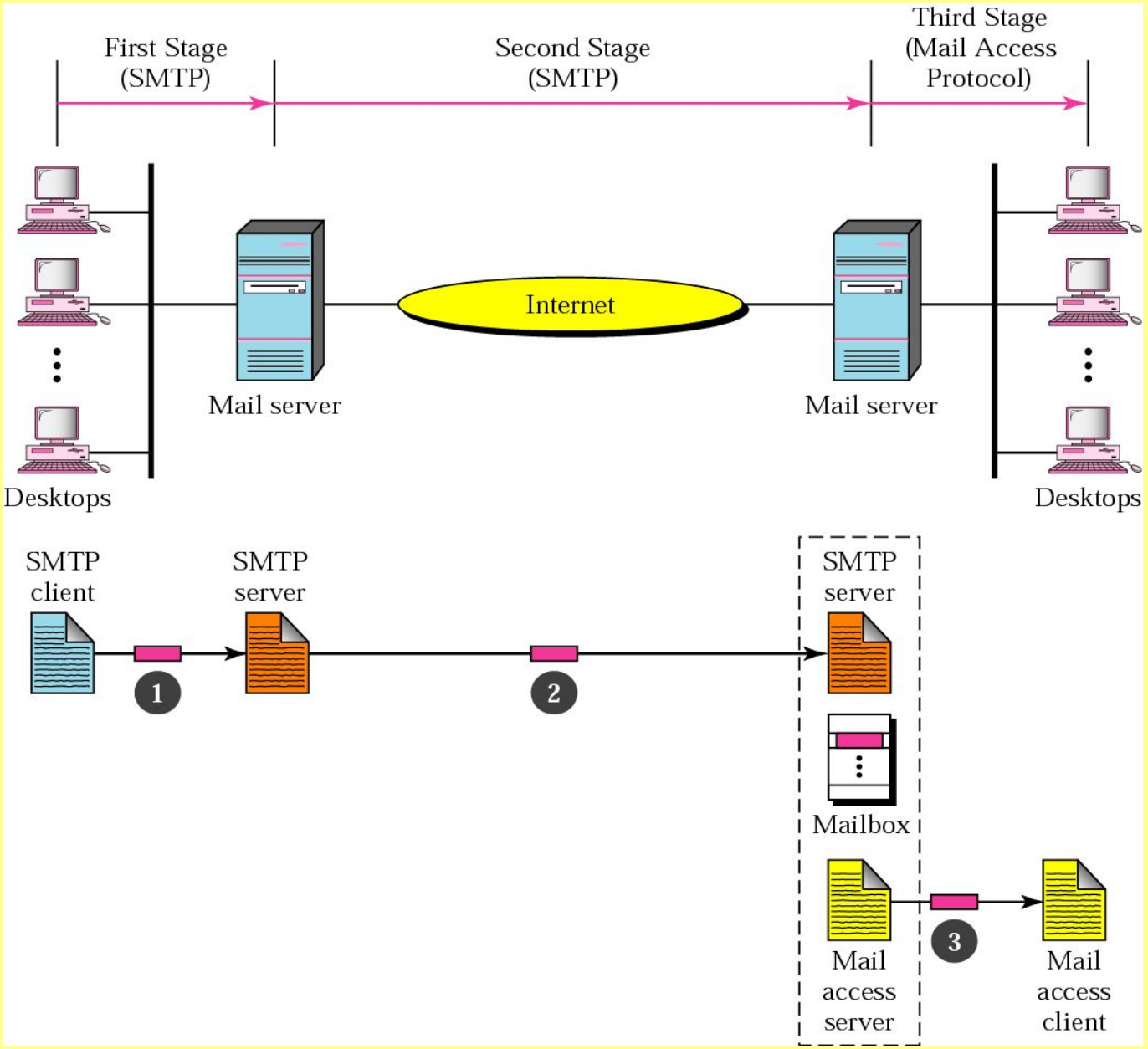
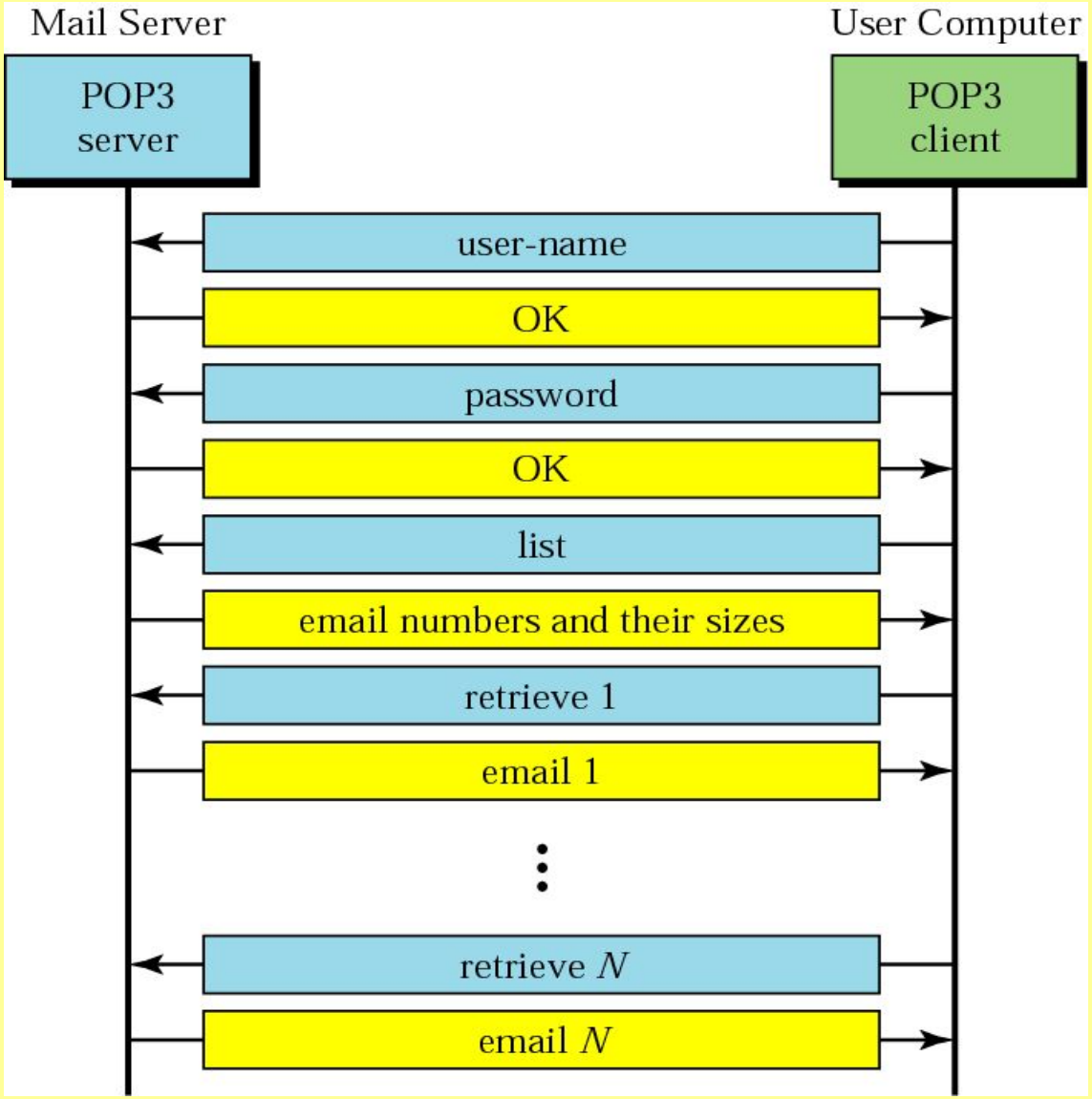


Figure 26.11 POP3



26.2 File Transfer

Connections

Communication

File Transfer

User Interface

Anonymous

Note:

FTP uses the services of TCP. It needs two TCP connections. The well-known port 21 is used for the control connection, and the well-known port 20 is used for the data connection.

Figure 26.12 FTP

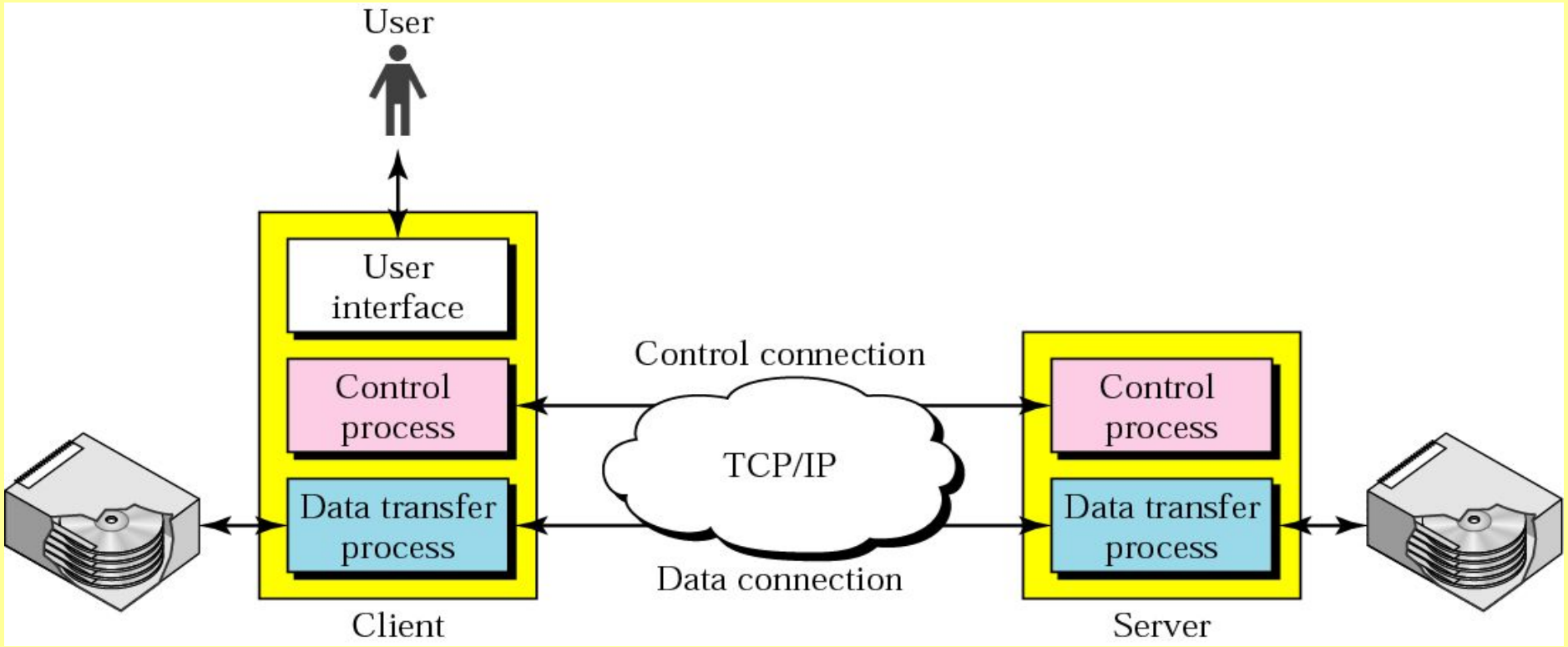


Figure 26.13 Using the control connection

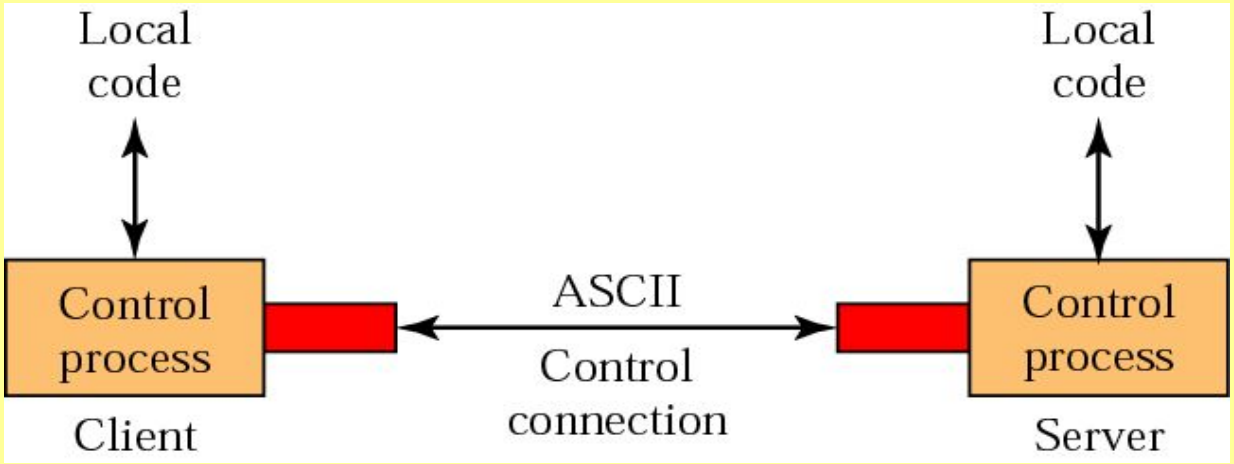


Figure 26.14 Using the data connection

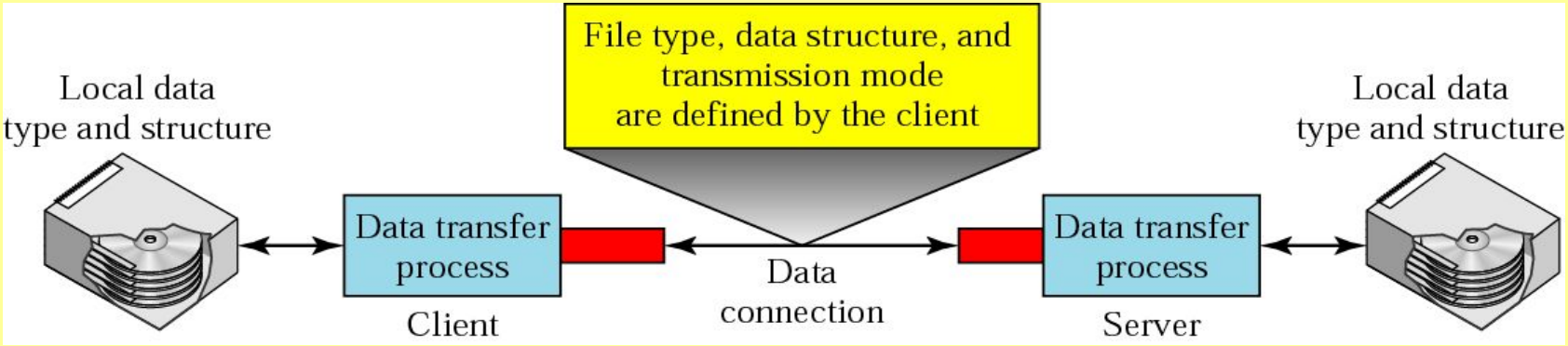
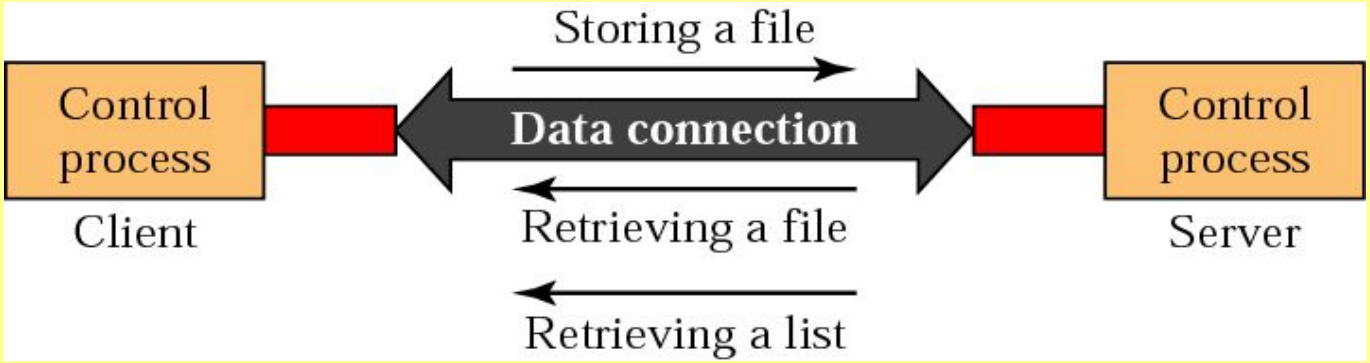


Figure 26.15 File transfer



Example 1

Figure 26.16 (next slide) shows an example of how a file is stored.

1. The control connection is created, and several control commands and responses are exchanged.
2. Data are transferred record by record.
3. A few commands and responses are exchanged to close the connection.

Figure 26.16 Example 1

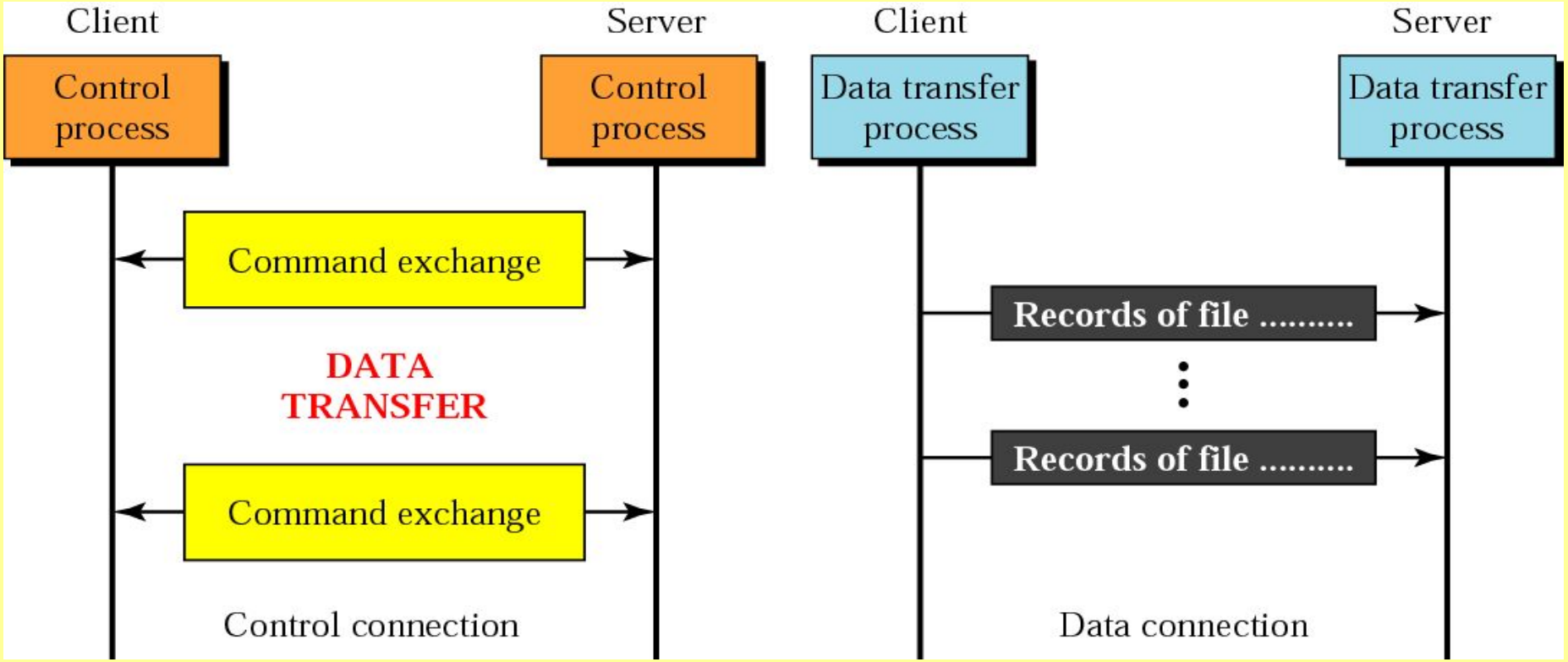


Table 26.4 List of FTP commands in UNIX

Commands
!, $, account, append, ascii, bell, binary, bye, case, cd, cdup, close, cr, delete, debug, dir, discount, form, get, glob, hash, help, lcd, ls, macdef, mdelete, mdir, mget, mkdir, mls, mode, mput, nmap, ntrans, open, prompt, proxy, sendport, put, pwd, quit, quote, recv, remotehelp, rename, reset, rmdir, runique, send, status, struct, sunique, tenex, trace, type, user, verbose,?

Example 2

We show some of the user interface commands that accomplish the same task as in Example 1. The user input is shown in boldface. As shown below, some of the commands are provided automatically by the interface. The user receives a prompt and provides only the arguments.

```
$ ftp challenger.atc.fhda.edu
Connected to challenger.atc.fhda.edu
220 Server ready
Name: forouzan
Password: xxxxxxx
ftp > ls /usr/user/report
200 OK
150 Opening ASCII mode
.....
.....
226 transfer complete
ftp > close
221 Goodbye
ftp > quit
```

Example 3

We show an example of using anonymous FTP. We connect to internic.net, where we assume there are some public data available.

```
$ ftp internic.net
Connected to internic.net
220 Server ready
Name: anonymous
331 Guest login OK, send "guest" as password
Password: guest
ftp > pwd
257 '/' is current directory
ftp > ls
200 OK
150 Opening ASCII mode
bin
...
ftp > close
221 Goodbye
ftp > quit
```