COMPUTER COMMUNICATION
EC 407

Syllabus

COURSE CODE	COURSE NAME	L-T-P-C	YEAR OF INTRODUCTION
EC407	COMPUTER COMMUNICATION	3-0-0-3	2016

Prerequisite: NIL

Course objectives:

- To give the basic concepts of computer network and working of layers, protocols and interfaces in a computer network.
- To introduce the fundamental techniques used in implementing secure network communications and give them an understanding of common threats and its defences.

Module	Course content (42 hrs)	Hours	End Sem. Exam Marks
I	Introduction to computer communication: Transmission modes - serial and parallel transmission, asynchronous, synchronous, simplex, half duplex, full duplex communication. Switching: circuit switching and packet switching	2	15%

Syllabus

	Networks: Network criteria, physical structures, network models, categories of networks, Interconnection of Networks: Internetwork	2	
	Network models: Layered tasks, OSI model, Layers in OSI model, TCP/IP protocol suite.	2	
II	Physical Layer: Guided and unguided transmission media (Co-axial cable, UTP,STP, Fiber optic cable)	2	15%
	Data Link Layer: Framing, Flow control (stop and wait , sliding window flow control)	2	
	Error control, Error detection(check sum, CRC), Bit stuffing, HDLC	2	
	Media access control: Ethernet (802.3), CSMA/CD, Logical link control, Wireless LAN (802.11), CSMA/CA	2	
FIRST INTERNAL EXAM			

Syllabus

III	Network Layer Logical addressing : IPv4 & IPV6	2	15%
	Address Resolution protocols (ARP, RARP)	2	
	Subnetting, Classless Routing(CIDR), ICMP, IGMP, DHCP	3	
	Virtual LAN, Networking devices (Hubs, Bridges & Switches)	1	
IV	Routing: Routing and Forwarding, Static routing and Dynamic routing	1	15%
	Routing Algorithms: Distance vector routing algorithm, Link state routing (Dijkstra's algorithm)	2	
	Routing Protocols: Routing Information protocol (RIP), Open Shortest Path First (OSPF), Border Gateway Protocol (BGP), MPLS	3	
SECOND INTERNAL EXAM			
V	Transport Layer –UDP, TCP	1	20%
	Congestion Control & Quality of Service – Data traffic, Congestion, Congestion Control, QoS and Flow Characteristics	4	
	Application Layer – DNS, Remote Logging (Telnet), SMTP, FTP, WWW, HTTP, POP3, MIME, SNMP	3	

Syllabus

VI	Introduction to information system security, common attacks	1	20%
	Security at Application Layer (E-MAIL, PGP and S/MIME). Security at Transport Layer (SSL and TLS). Security at Network Layer (IPSec).	3	
	Defence and counter measures: Firewalls and their types. DMZ, Limitations of firewalls, Intrusion Detection Systems -Host based, Network based, and Hybrid IDSs	2	
END SEMESTER EXAM			

Question Paper Pattern

The question paper shall consist of three parts. Part A covers modules I and II, Part B covers modules III and IV, and Part C covers modules V and VI. Each part has three questions uniformly covering the two modules and each question can have maximum four subdivisions. In each part, any two questions are to be answered. Mark patterns are as per the syllabus with 90% for theory and 10% for logical/numerical problems, derivation and proof.

References

Text Books:

1. Behrouz A. Forouzan, Cryptography & Network Security , , IV Edition, Tata McGraw-Hill, 2008
2. J F Kurose and K W Ross, Computer Network A Top-down Approach Featuring the Internet, 3/e, Pearson Education, 2010

References:

1. Behrouz A Forouzan, Data Communications and Networking, 4/e, Tata McGraw-Hill, 2006.
2. Larry Peterson and Bruce S Davie: Computer Network- A System Approach, 4/e, Elsevier India, 2011.
3. S. Keshav, An Engineering Approach to Computer Networking, Pearson Education, 2005.
4. Achyut S.Godbole, Data Communication and Networking, 2e, McGraw Hill Education New Delhi, 2011

UNIT 5

UDP

UDP(User Datagram Protocol)

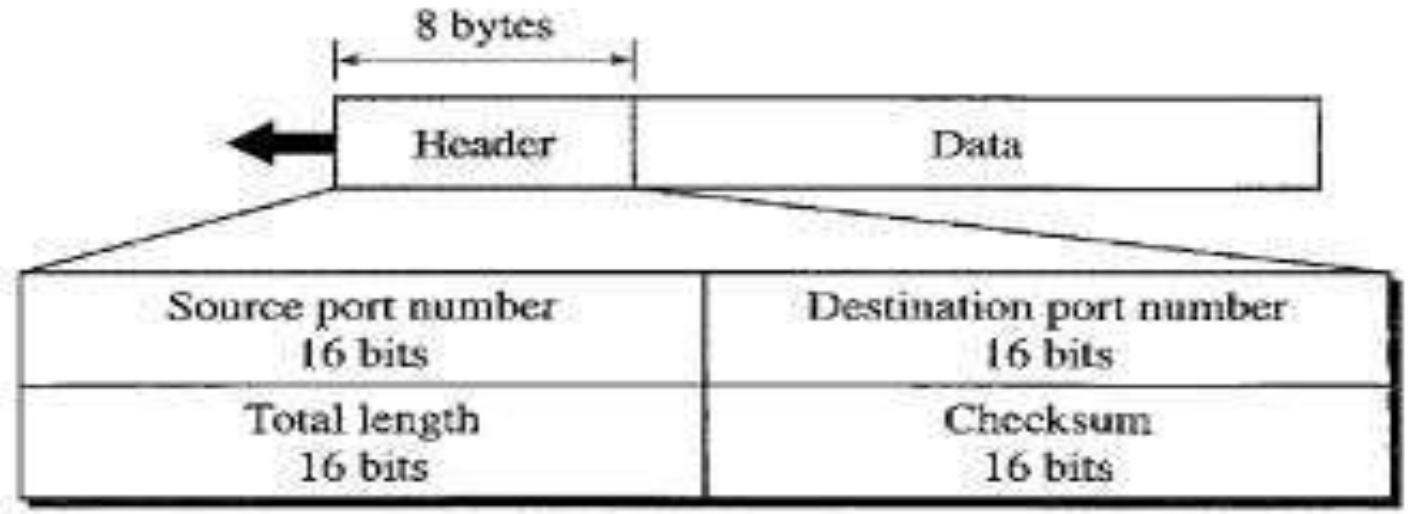
- Both UDP and TCP run on top of the Internet Protocol (IP)
- UDP enables process-to-process communication,
- UDP sends messages, called datagrams.
- **User Datagram Protocol (UDP)** is a Transport Layer protocol.
- Unlike TCP, it is **unreliable and connectionless protocol.** So, there is no need to establish connection prior to data transfer.
- For the realtime services like computer gaming, voice or video communication, live conferences; we need UDP.
- User Datagram Protocol (UDP) is more efficient in terms of both latency and bandwidth.

UDP(User Datagram Protocol)

- **Source Port :** Source Port is 2 Byte long field used to identify port number of source.
- **Destination Port :** It is 2 Byte long field, used to identify the port of destined packet.
- **Length :** Length is the length of UDP including header and the data. It is 16-bits field.
- ChecksumThe checksum field may be used for error-checking of the header and data. This field is optional in IPv4, and mandatory in IPv6. The field carries all-zeros if unused.
- No Error control or flow control is provided by UDP. Hence UDP depends on IP and ICMP for error reporting.

UDP(User Datagram Protocol)

User datagram format



UDP length = IP length – IP header's length

UDP Operation

- **Connectionless service (explanation is needed)**
- **Flow and error control : There is no mechanism for flow and error control. Process that use UDP should provide these mechanisms**

Encapsulation and Decapsulation

To send a message from one process to another, the UDP protocol encapsulates and decapsulates messages in an IP datagram.

Use of UDP

The following lists some uses of the UDP protocol:

- UDP is suitable for a process that requires simple request-response communication with little concern for flow and error control. It is not usually used for a process such as FTP that needs to send bulk data (see Chapter 26).
- UDP is suitable for a process with internal flow and error control mechanisms. For example, the Trivial File Transfer Protocol (TFTP) process includes flow and error control. It can easily use UDP.
- UDP is a suitable transport protocol for multicasting. Multicasting capability is embedded in the UDP software but not in the TCP software.
- UDP is used for management processes such as SNMP (see Chapter 28).
- UDP is used for some route updating protocols such as Routing Information Protocol (RIP) (see Chapter 22).

Transmission Control Protocol (TCP)

Topics Discussed in the Section

- ✓ Process-to-Process Communication
- ✓ Stream Delivery Service
- ✓ Full-Duplex Communication
- ✓ Multiplexing and Demultiplexing
- ✓ Connection-Oriented Service
- ✓ Reliable Service

Figure 15.1 *TCP/IP protocol suite*

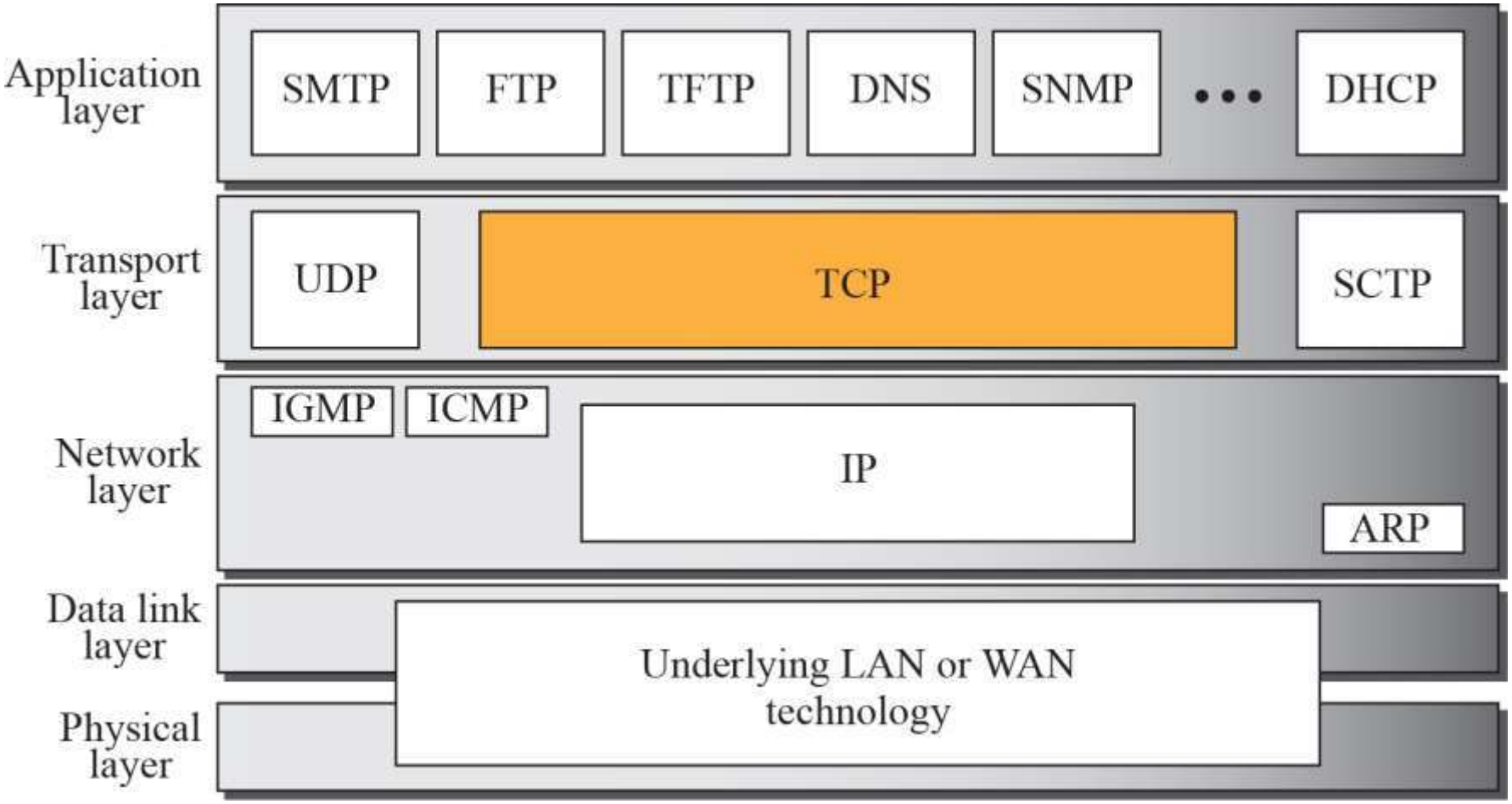


Table 15.1 *Well-known Ports used by TCP*

Port	*Protocol*	*Description*
7	Echo	Echoes a received datagram back to the sender
9	Discard	Discards any datagram that is received
11	Users	Active users
13	Daytime	Returns the date and the time
17	Quote	Returns a quote of the day
19	Chargen	Returns a string of characters
20 and 21	FTP	File Transfer Protocol (Data and Control)
23	TELNET	Terminal Network
25	SMTP	Simple Mail Transfer Protocol
53	DNS	Domain Name Server
67	BOOTP	Bootstrap Protocol
79	Finger	Finger
80	HTTP	Hypertext Transfer Protocol

Figure 15.2 *Stream delivery*

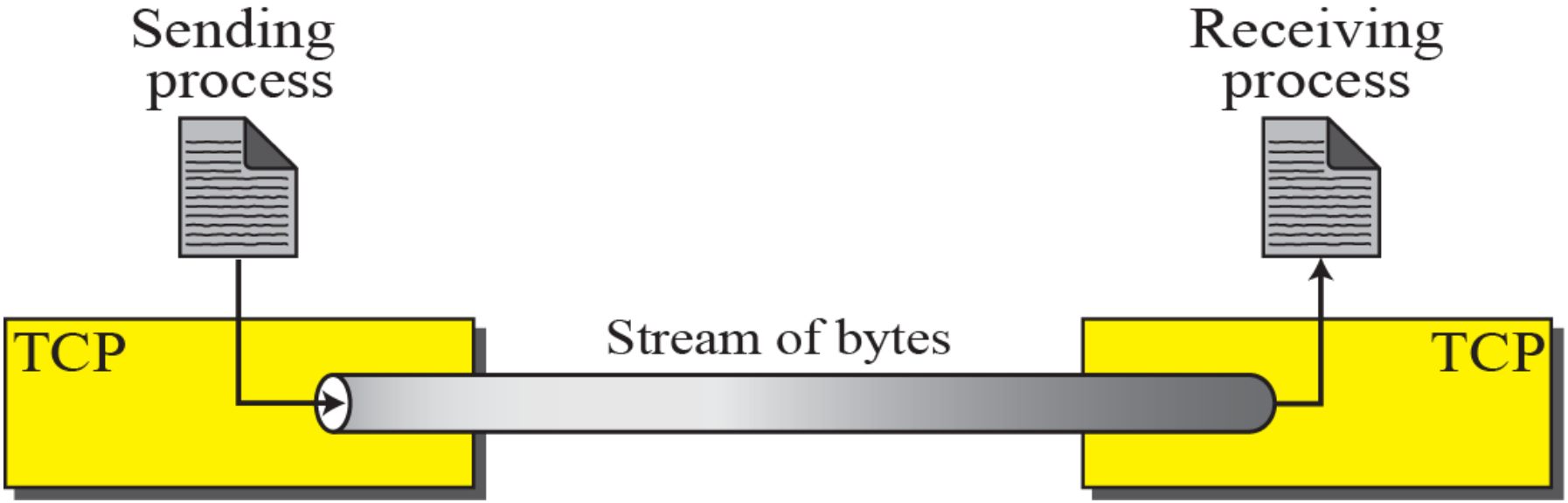


Figure 15.3 *Sending and receiving buffers*

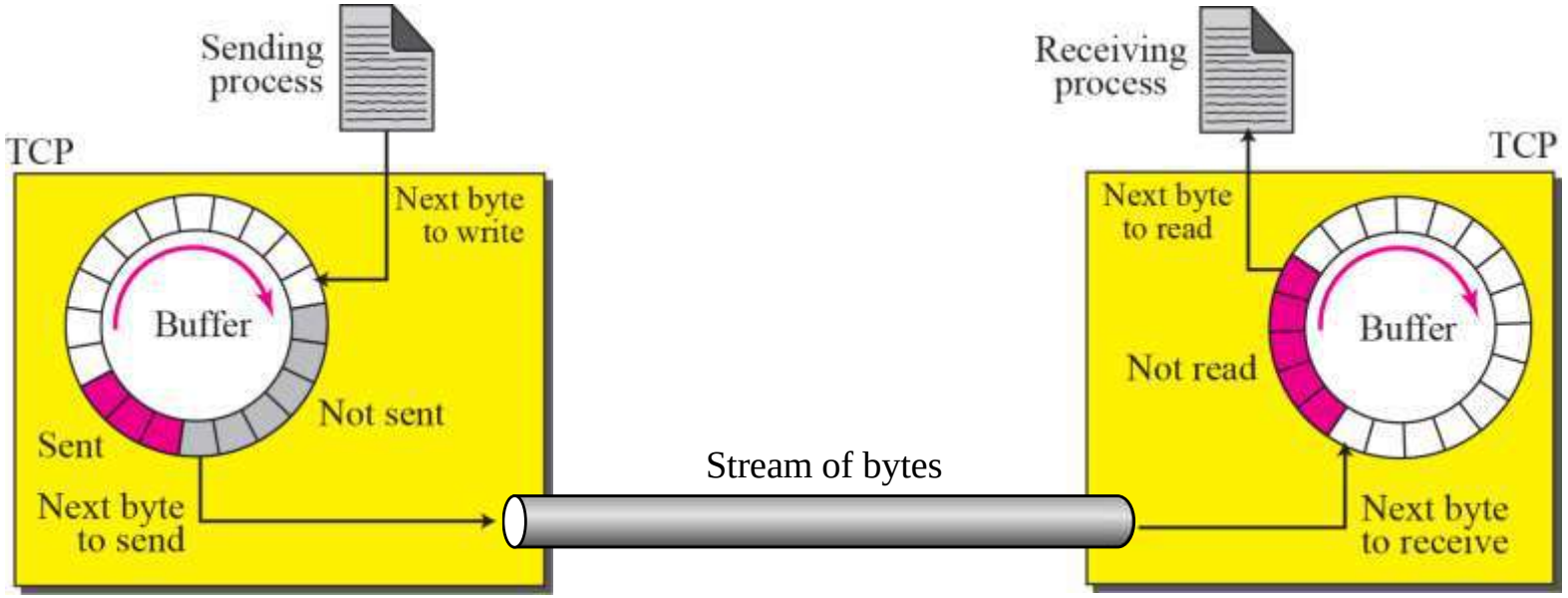
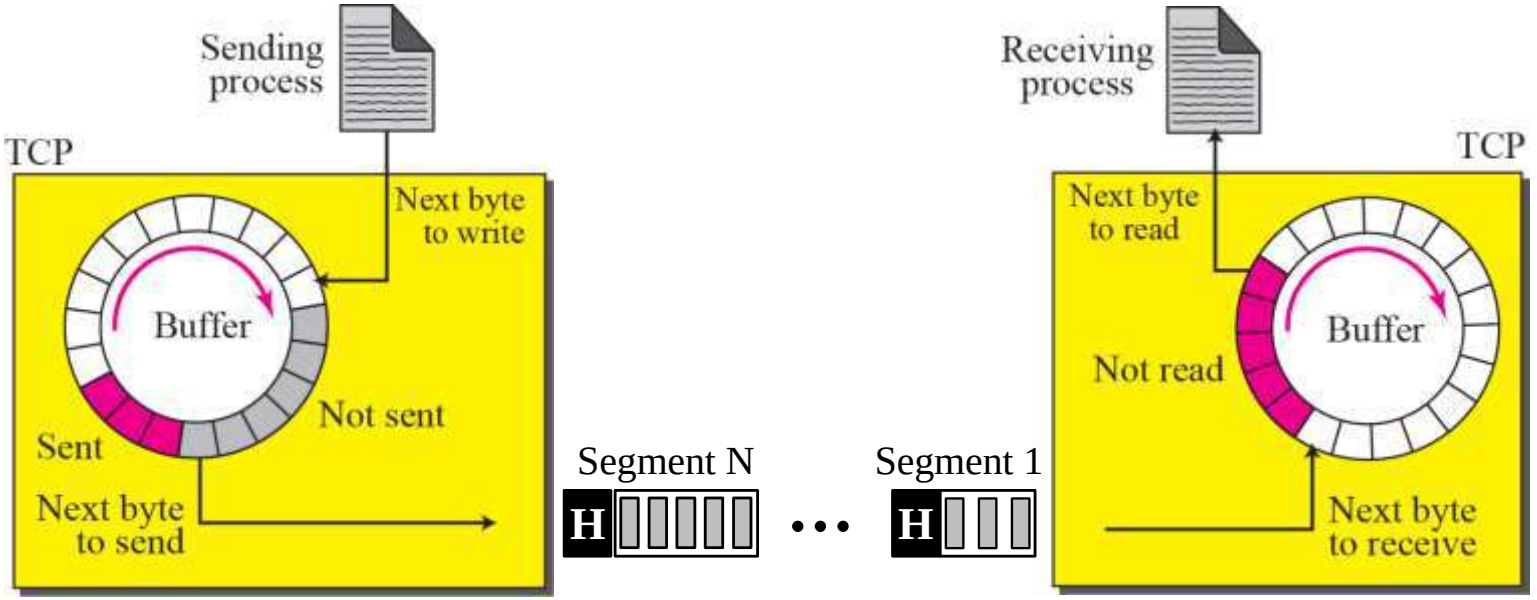


Figure 15.4 *TCP segments*



15-2 TCP FEATURES

To provide the services mentioned in the previous section, TCP has several features that are briefly summarized in this section and discussed later in detail.

Topics Discussed in the Section

- ✓ Numbering System
- ✓ Flow Control
- ✓ Error Control
- ✓ Congestion Control

The bytes of data being transferred in each connection are numbered by TCP. The numbering starts with a randomly generated number.

Note

The bytes of data being transferred in each connection are numbered by TCP.

The numbering starts with an arbitrarily generated number.

Example 15.1

Suppose a TCP connection is transferring a file of 5,000 bytes. The first byte is numbered 10,001. What are the sequence numbers for each segment if data are sent in five segments, each carrying 1,000 bytes?

Solution

The following shows the sequence number for each segment:

Segment 1	→	**Sequence Number:**	**10,001**	**Range:**	**10,001**	to	**11,000**
Segment 2	→	**Sequence Number:**	**11,001**	**Range:**	**11,001**	to	**12,000**
Segment 3	→	**Sequence Number:**	**12,001**	**Range:**	**12,001**	to	**13,000**
Segment 4	→	**Sequence Number:**	**13,001**	**Range:**	**13,001**	to	**14,000**
Segment 5	→	**Sequence Number:**	**14,001**	**Range:**	**14,001**	to	**15,000**

Note

The value in the sequence number field of a segment defines the number assigned to the first data byte contained in that segment.

Note

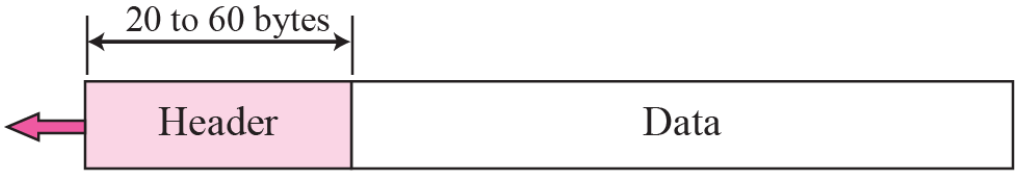
The value of the acknowledgment field in a segment defines the number of the next byte a party expects to receive.

The acknowledgment number is cumulative.

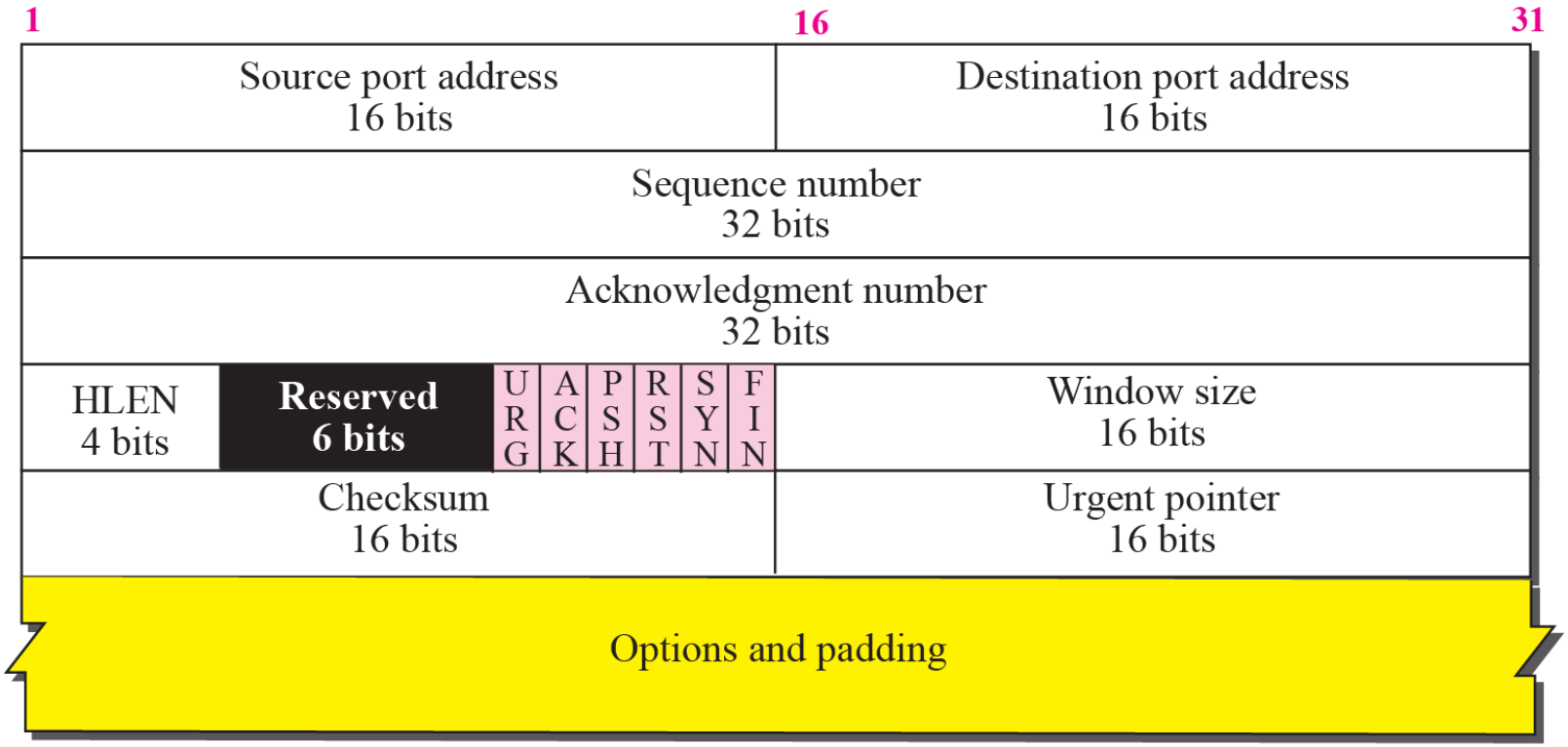
15-3 SEGMENT

A packet in TCP is called a segment.

Figure 15.5 *TCP segment format*



a. Segment



b. Header

Figure 15.5 *TCP segment format*

- **Source port address.** This is a 16-bit field that defines the port number of the application program in the host that is sending the segment. This serves the same purpose as the source port address in the UDP header.
- **Destination port address.** This is a 16-bit field that defines the port number of the application program in the host that is receiving the segment. This serves the same purpose as the destination port address in the UDP header.
- **Sequence number.** This 32-bit field defines the number assigned to the first byte of data contained in this segment. As we said before, TCP is a stream transport protocol. To ensure connectivity, each byte to be transmitted is numbered. The sequence number tells the destination which byte in this sequence comprises the first byte in the segment. During connection establishment, each party uses a random number generator to create an **initial sequence number (ISN),** which is usually different in each direction.

Figure 15.5 *TCP segment format*

- **Acknowledgment number.** This 32-bit field defines the byte number that the receiver of the segment is expecting to receive from the other party. If the receiver of the segment has successfully received byte number x from the other party, it defines $x + 1$ as the acknowledgment number. Acknowledgment and data can be piggybacked together.
- **Header length.** This 4-bit field indicates the number of 4-byte words in the TCP header. The length of the header can be between 20 and 60 bytes. Therefore, the value of this field can be between 5 ($5 \times 4 = 20$) and 15 ($15 \times 4 = 60$).
- **Reserved.** This is a 6-bit field reserved for future use.
- **Control.** This field defines 6 different control bits or flags
  One or more of these bits can be set at a time.
- **Window size.** This field defines the size of the window, in bytes, that the other party must maintain. Note that the length of this field is 16 bits, which means that the maximum size of the window is 65,535 bytes. This value is normally referred to as the receiving window (rwnd) and is determined by the receiver. The sender must obey the dictation of the receiver in this case.

Figure 15.5 *TCP segment format*

- **Checksum.** This 16-bit field contains the checksum. The calculation of the checksum for TCP follows the same procedure as the one described for UDP. However, the inclusion of the checksum in the UDP datagram is optional, whereas the inclusion of the checksum for TCP is mandatory. The same pseudoheader, serving the same purpose, is added to the segment. For the TCP pseudoheader, the value for the protocol field is 6.
- **Urgent pointer.** This 16-bit field, which is valid only if the urgent flag is set, is used when the segment contains urgent data. It defines the number that must be added to the sequence number to obtain the number of the last urgent byte in the data section of the segment. This will be discussed later in this chapter.
- **Options.** There can be up to 40 bytes of optional information in the TCP header.

Figure 15.6 *Control field*

URG: Urgent pointer is valid
ACK: Acknowledgment is valid
PSH: Request for push
RST: Reset the connection
SYN: Synchronize sequence numbers
FIN: Terminate the connection

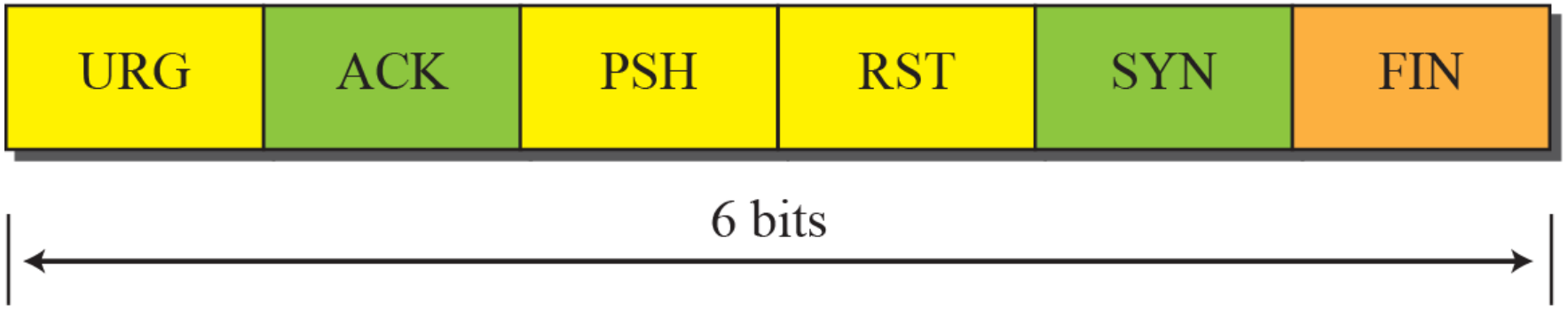
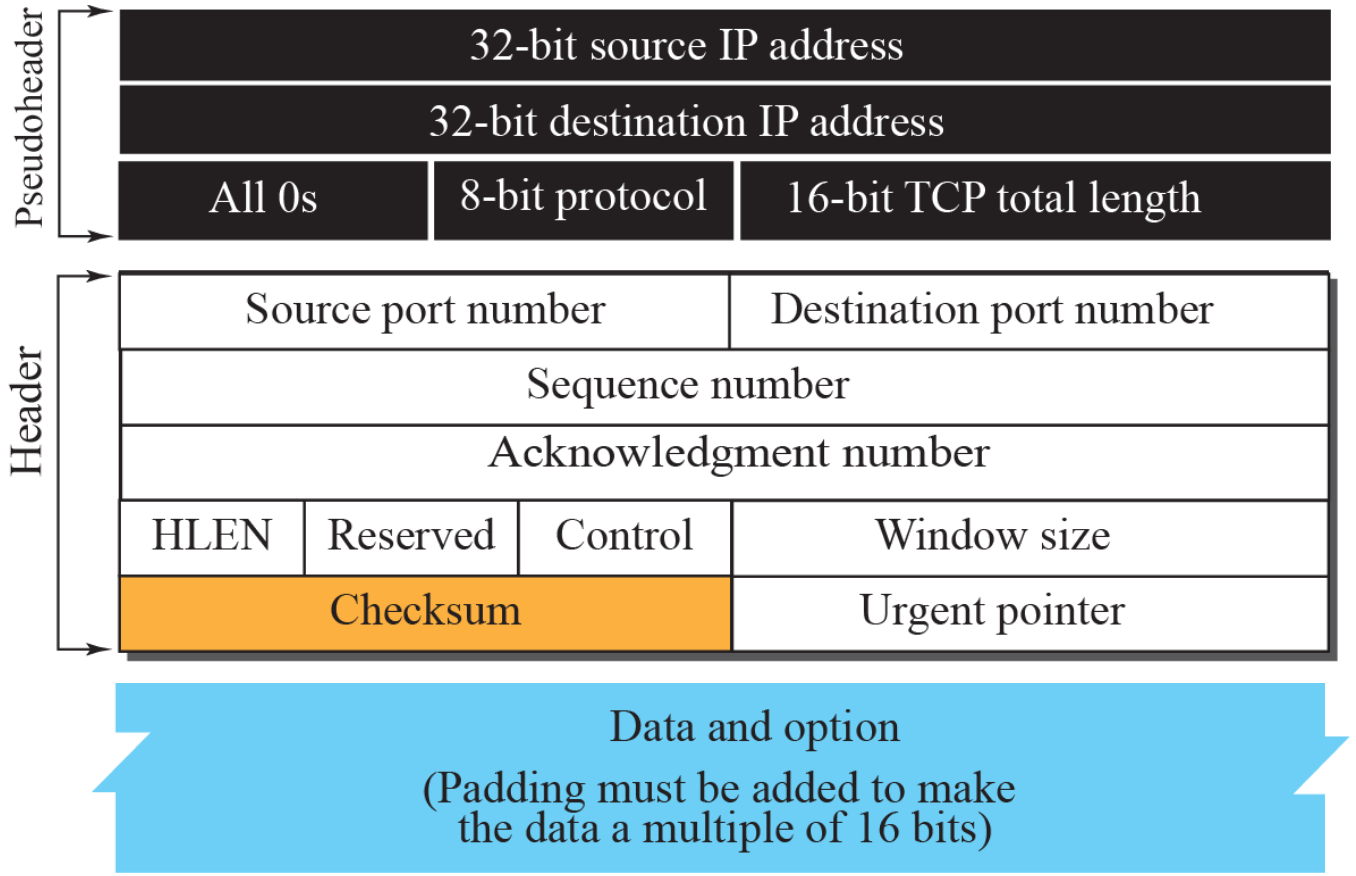


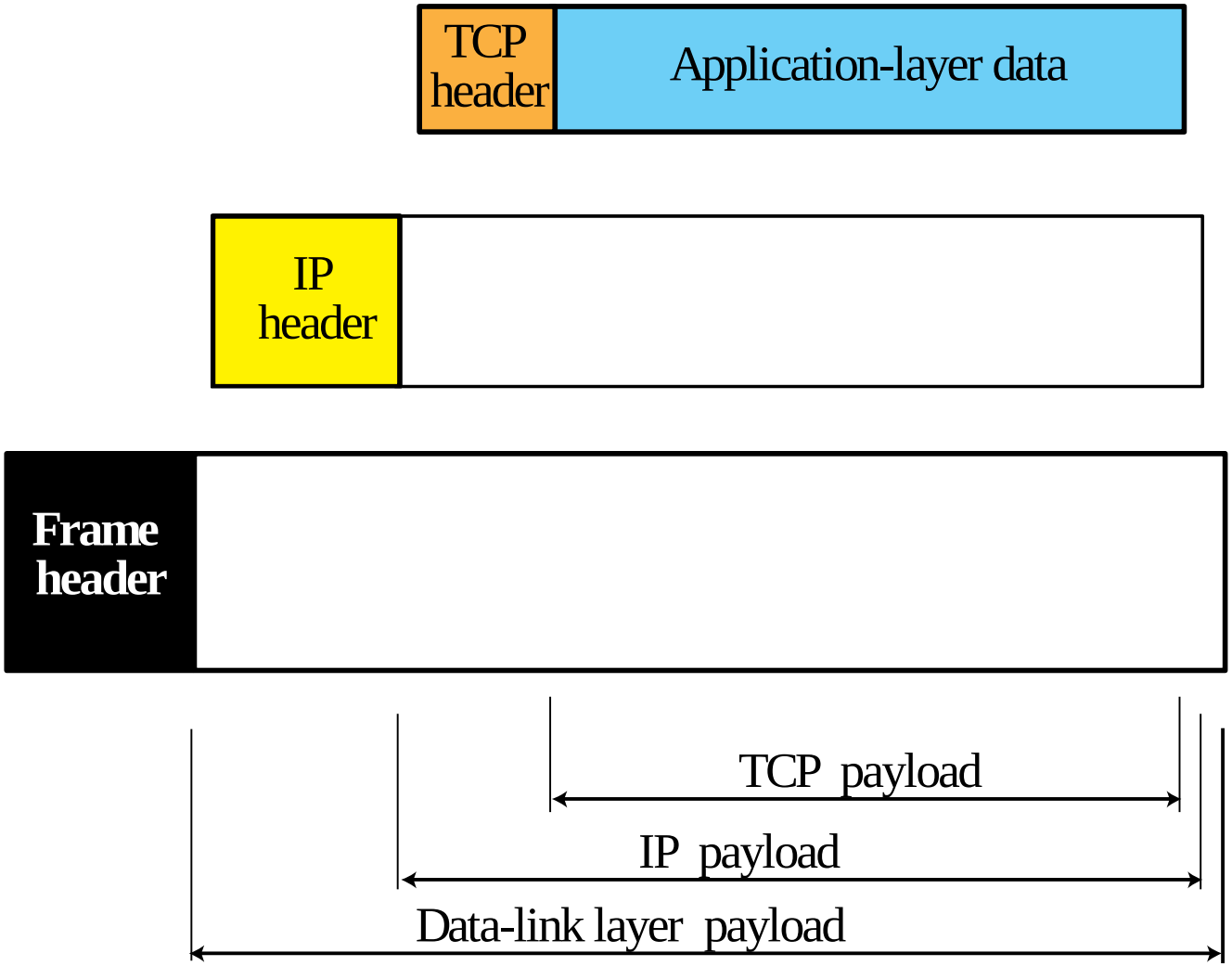
Figure 15.7 *Pseudoheader added to the TCP segment*



Note

The use of the checksum in TCP is mandatory.

Figure 15.8 *Encapsulation*



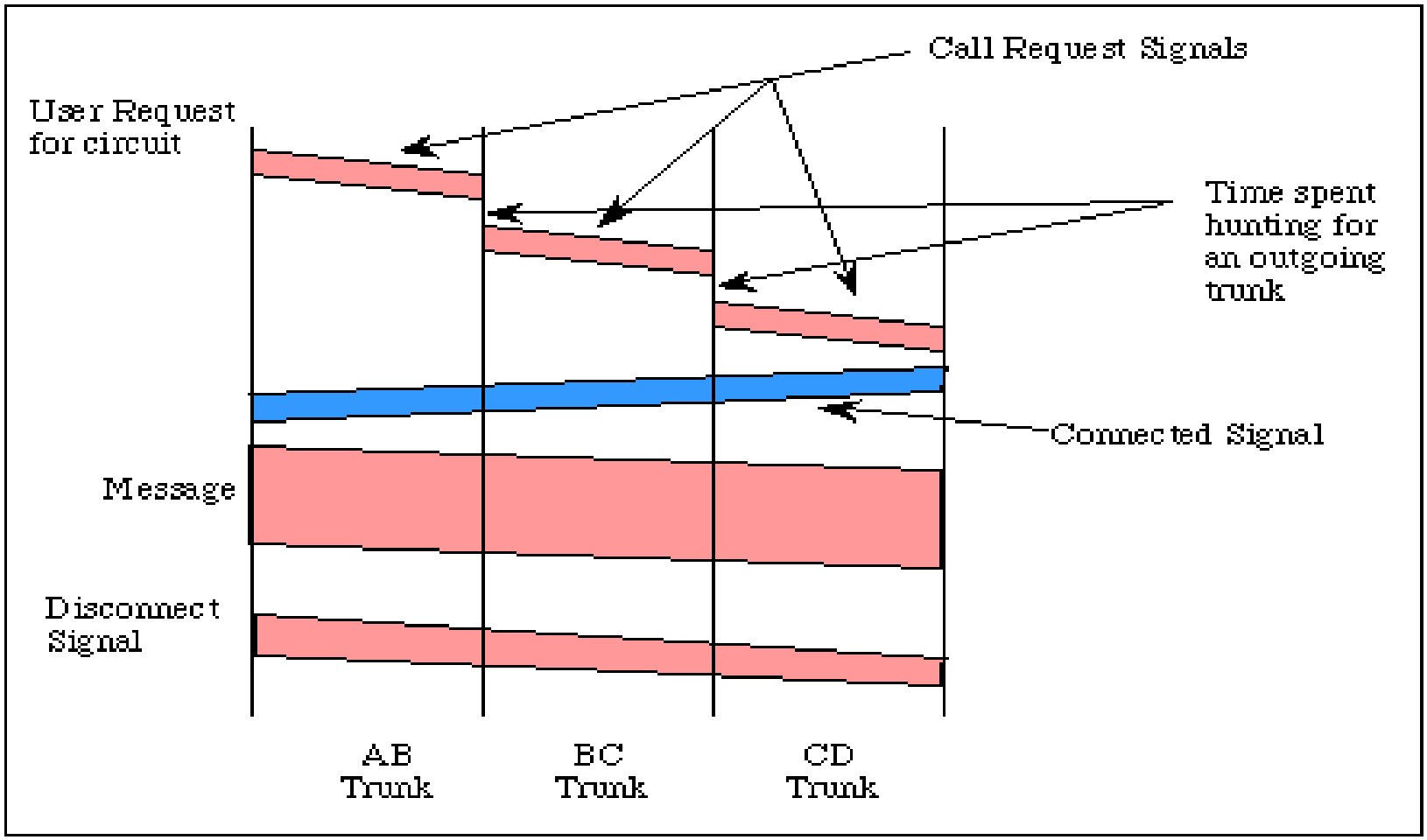
15-4 A TCP CONNECTION

TCP is connection-oriented. It establishes a virtual path between the source and destination. All of the segments belonging to a message are then sent over this virtual path. You may wonder how TCP, which uses the services of IP, a connectionless protocol, can be connection-oriented. The point is that a TCP connection is virtual, not physical. TCP operates at a higher level. TCP uses the services of IP to deliver individual segments to the receiver, but it controls the connection itself. If a segment is lost or corrupted, it is retransmitted.

Topics Discussed in the Section

- ✓ Connection Establishment
- ✓ Data Transfer
- ✓ Connection Termination
- ✓ Connection Reset

Circuit Switching



Flow control, error control and congestion control

- ✓For flow control, TCP uses Sliding window
- ✓ For error control, TCP uses Checksum
- ✓ For congestion control, slow start, congestion avoidance and congestion detection

Congestion Control and Quality of Service

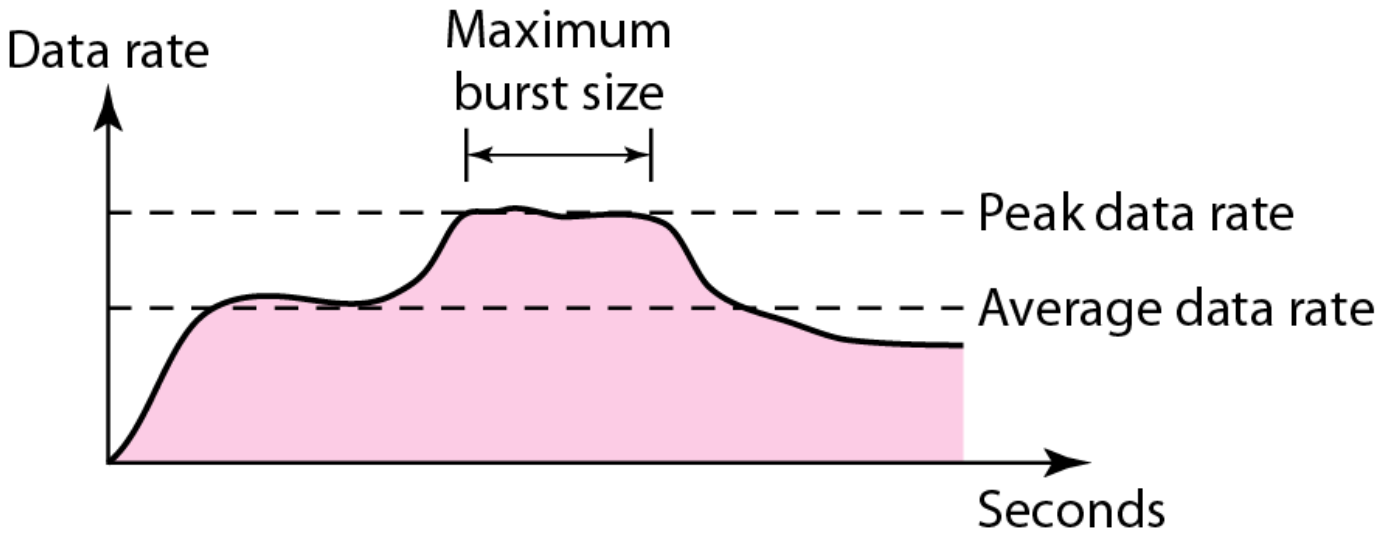
24-1 DATA TRAFFIC

The main focus of congestion control and quality of service is data traffic. In congestion control we try to avoid traffic congestion. In quality of service, we try to create an appropriate environment for the traffic. So, before talking about congestion control and quality of service, we discuss the data traffic itself.

Topics discussed in this section:

Traffic Descriptor
Traffic Profiles

Figure 24.1 *Traffic descriptors*



24-2 CONGESTION

Congestion in a network may occur if the load on the network—the number of packets sent to the network—is greater than the capacity of the network—the number of packets a network can handle. Congestion control refers to the mechanisms and techniques to control the congestion and keep the load below the capacity.

Topics discussed in this section:

Network Performance

Figure 24.3 *Queues in a router*

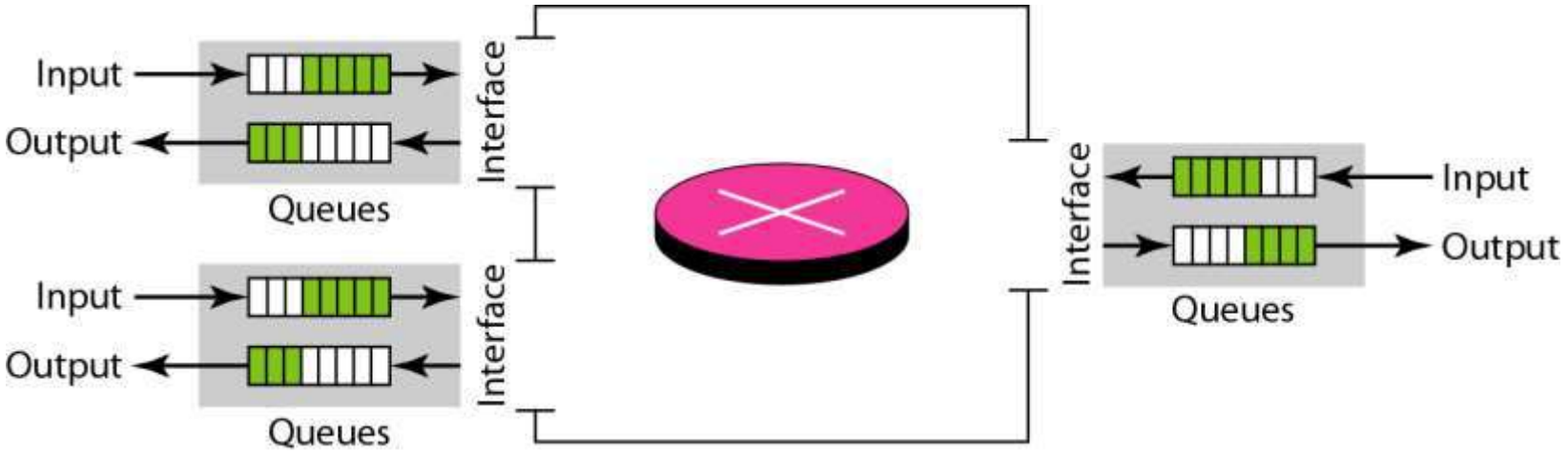
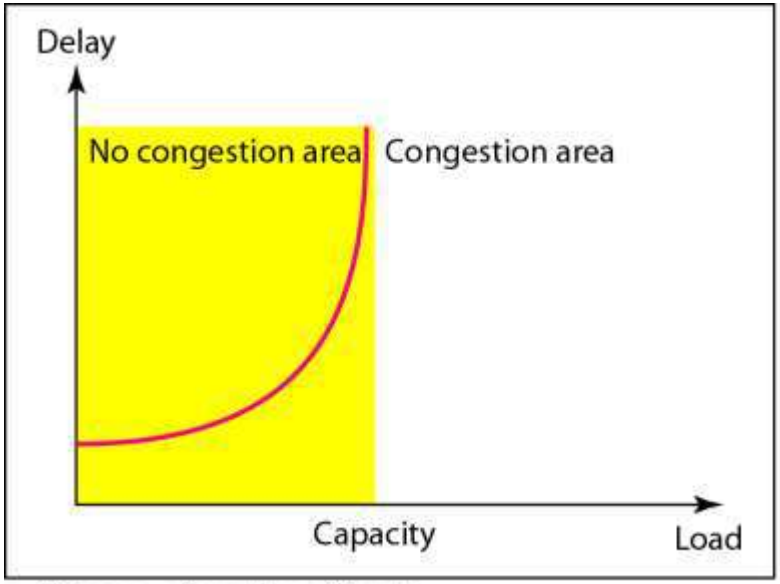
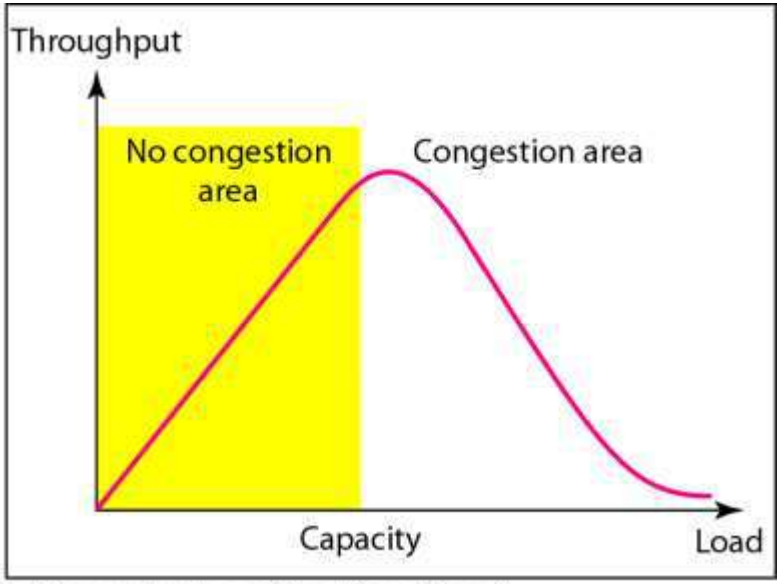


Figure *Packet delay and throughput as functions of load*



a. Delay as a function of load



b. Throughput as a function of load

24-3 CONGESTION CONTROL

Congestion control refers to techniques and mechanisms that can either prevent congestion, before it happens, or remove congestion, after it has happened. In general, we can divide congestion control mechanisms into two broad categories: open-loop congestion control (prevention) and closed-loop congestion control (removal).

Topics discussed in this section:

Open-Loop Congestion Control
Closed-Loop Congestion Control

Figure 24.5 *Congestion control categories*

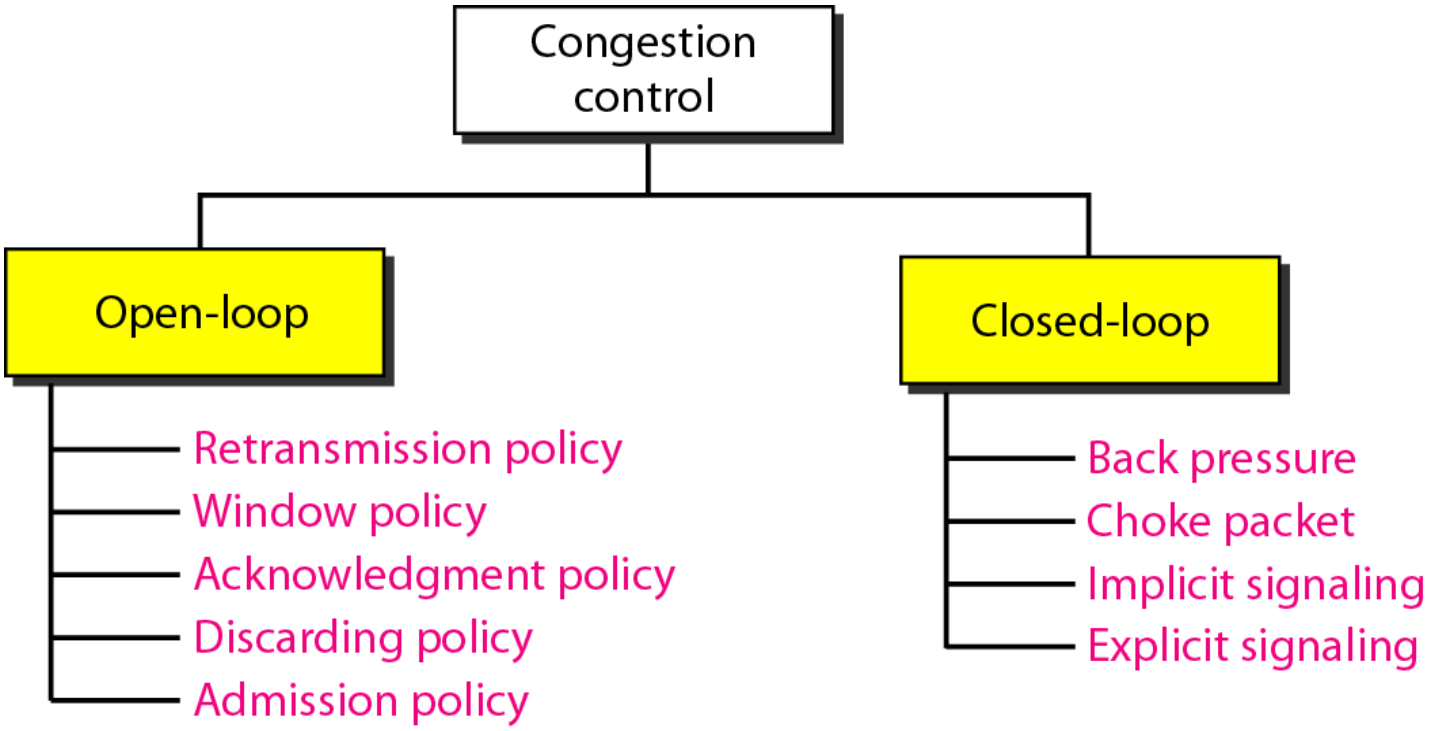


Figure 24.6 *Backpressure method for alleviating congestion*

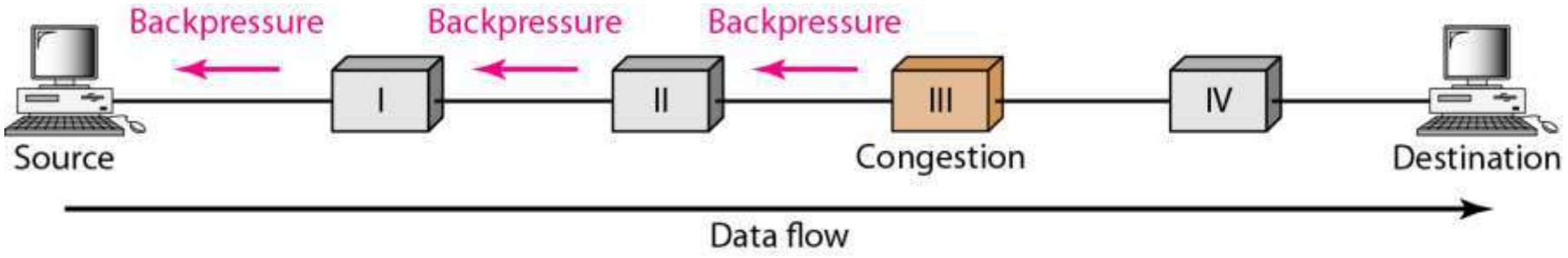
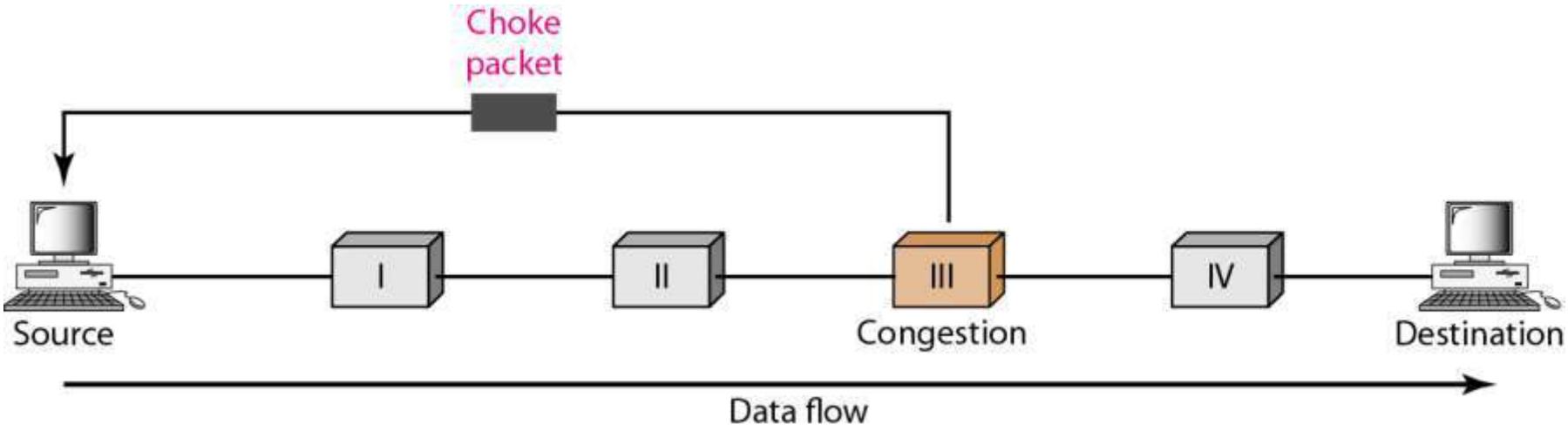


Figure 24.7 *Choke packet*



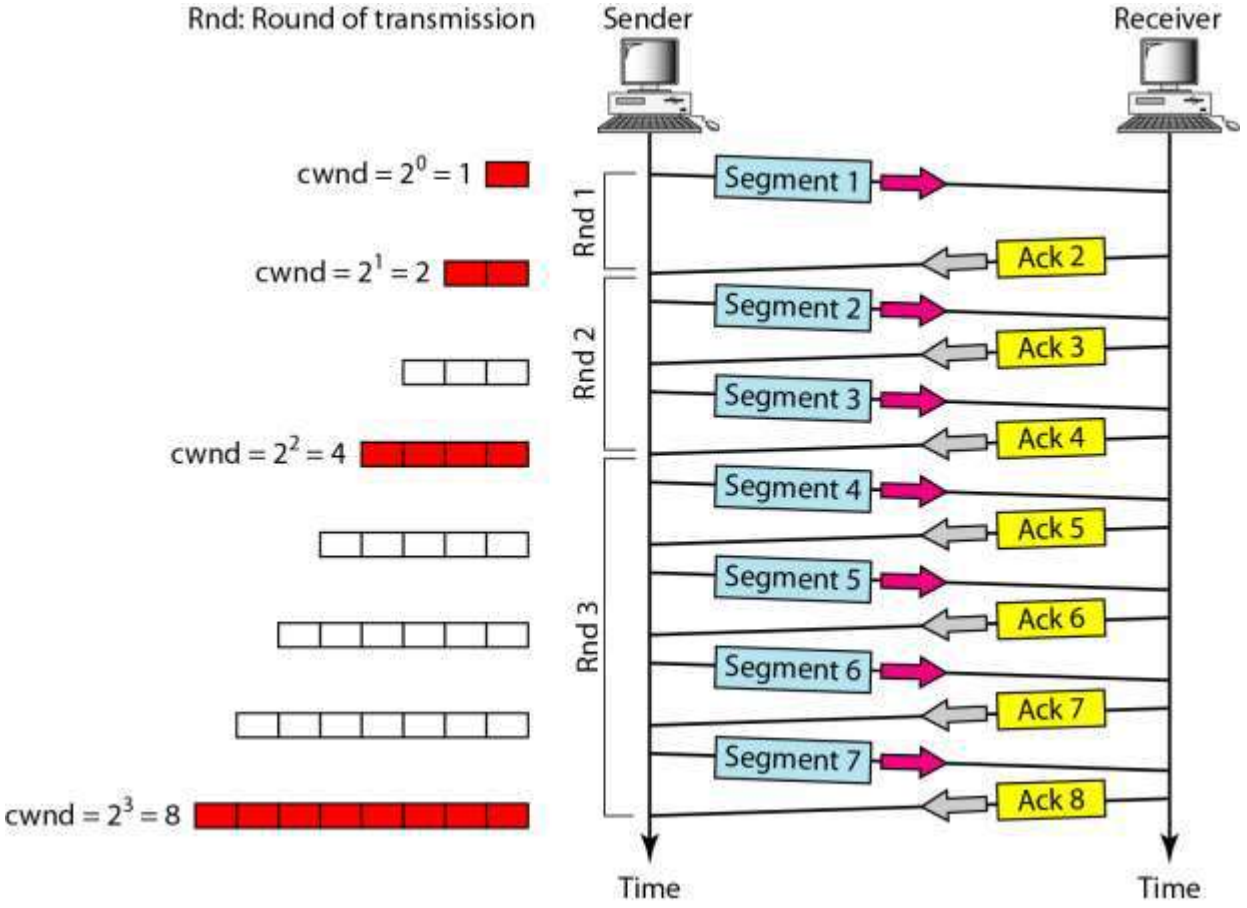
24-4 TWO EXAMPLES

To better understand the concept of congestion control, let us give two examples: one in TCP and the other in Frame Relay.

Topics discussed in this section:

Congestion Control in TCP
Congestion Control in Frame Relay

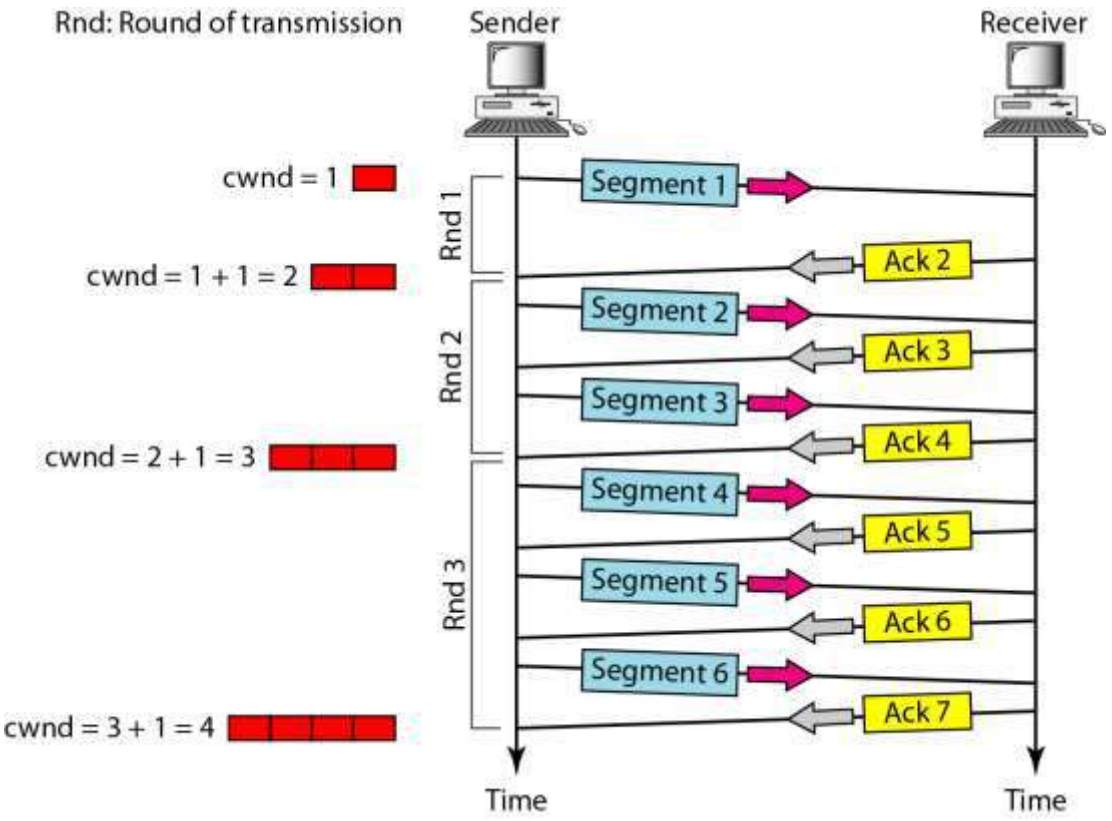
Figure 24.8 *Slow start, exponential increase*



Note

In the slow-start algorithm, the size of the congestion window increases exponentially until it reaches a threshold.

Figure 24.9 *Congestion avoidance, additive increase*



Note

In the congestion avoidance algorithm, the size of the congestion window increases additively until congestion is detected.

Note

An implementation reacts to congestion detection in one of the following ways:

- If detection is by time-out, a new slow start phase starts.
- If detection is by three ACKs, a new congestion avoidance phase starts.

Figure 24.10 *TCP congestion policy summary*

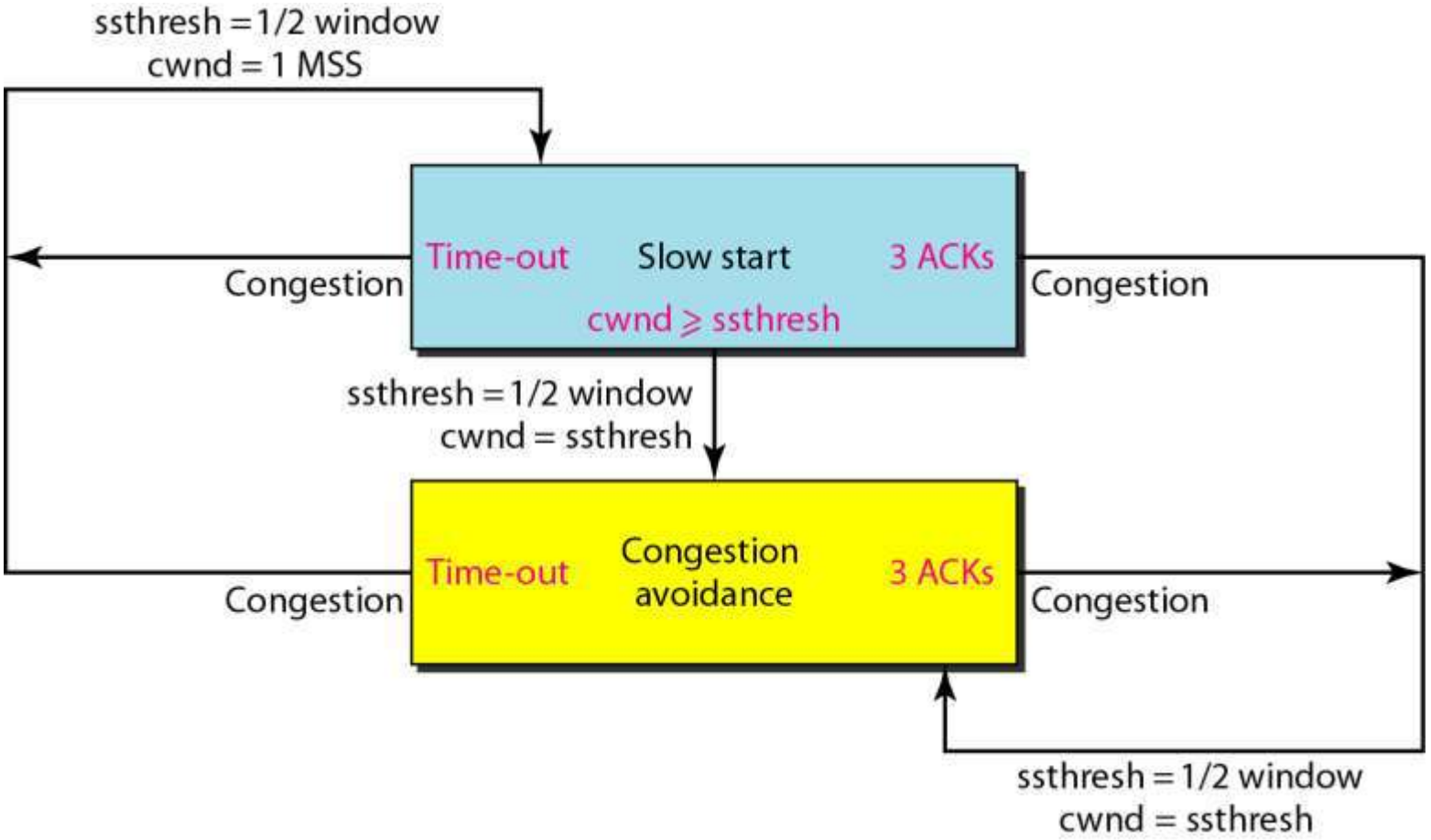


Figure 24.11 *Congestion example*

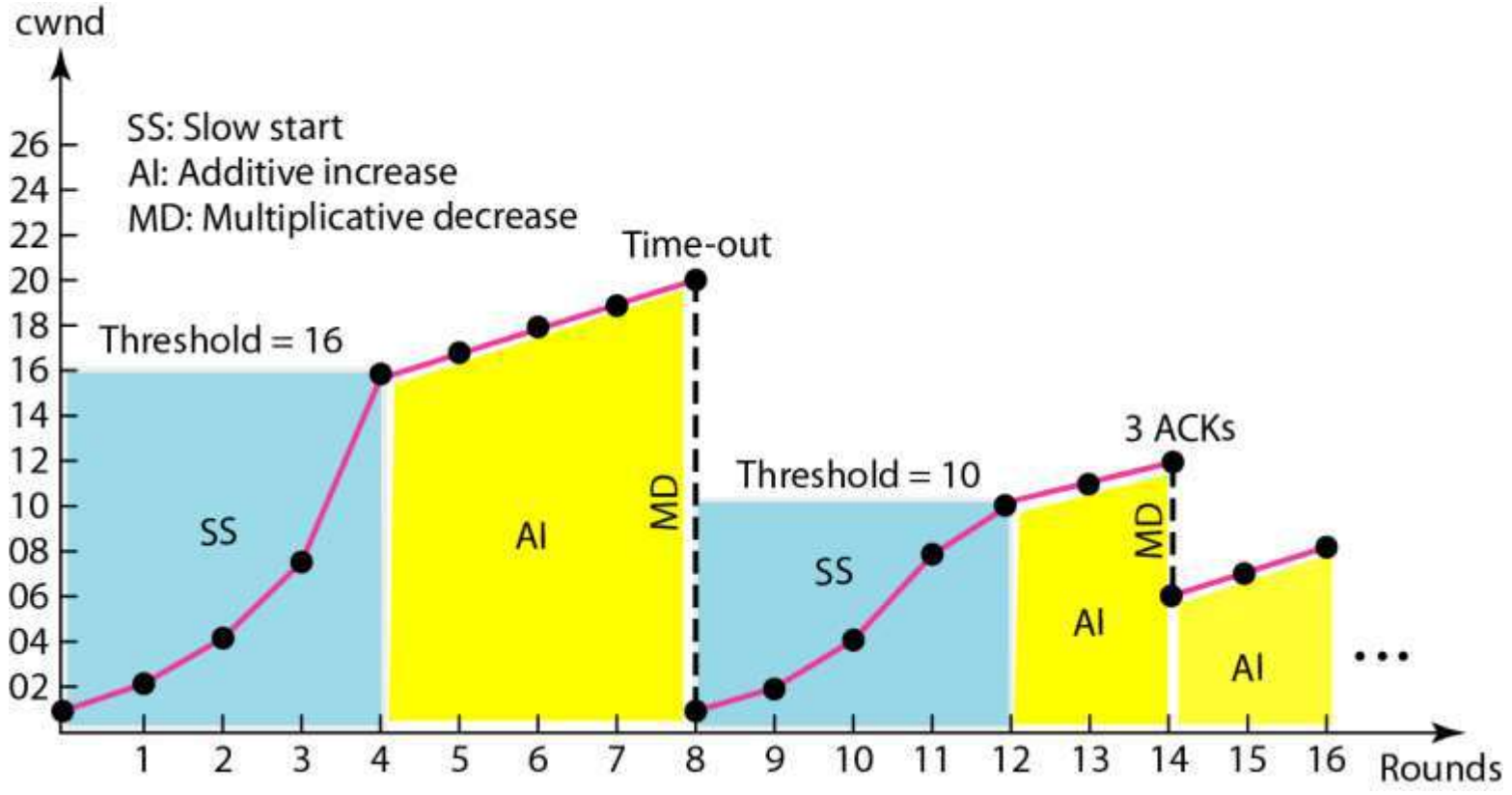


Figure 24.12 *BECN (Backward Explicit Congestion Notification bit)*

High throughput and low delay are the main goals orkf frame relay network
It allows user to to transmit bursty data

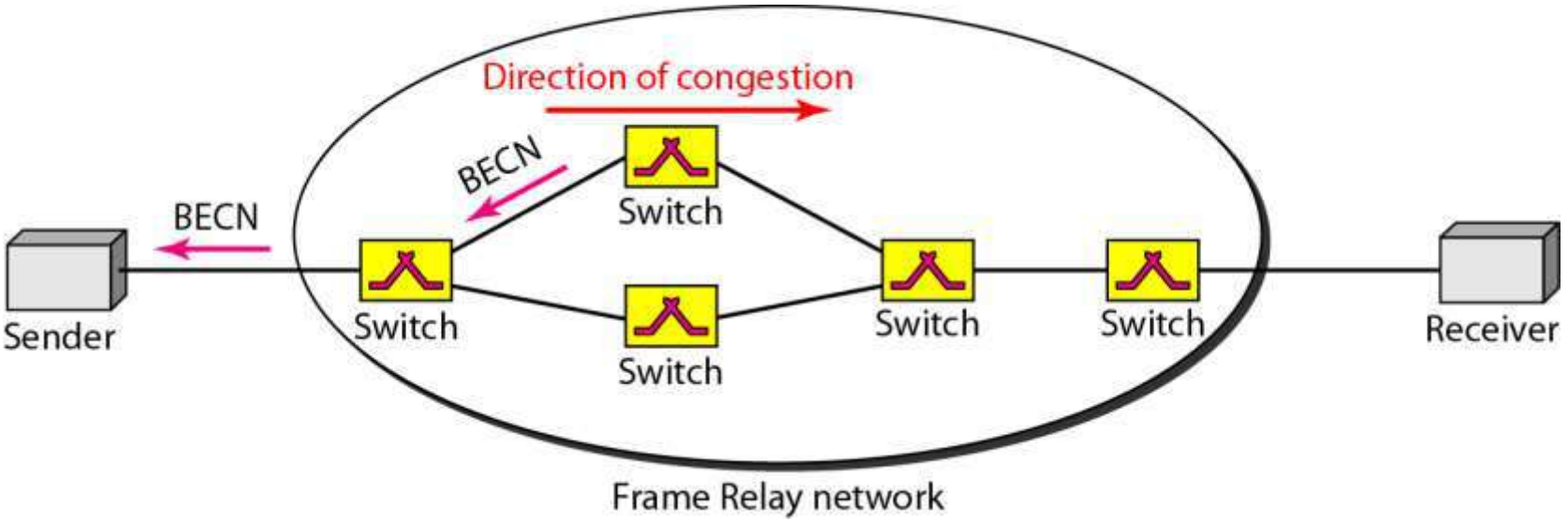


Figure 24.13 *FECN*

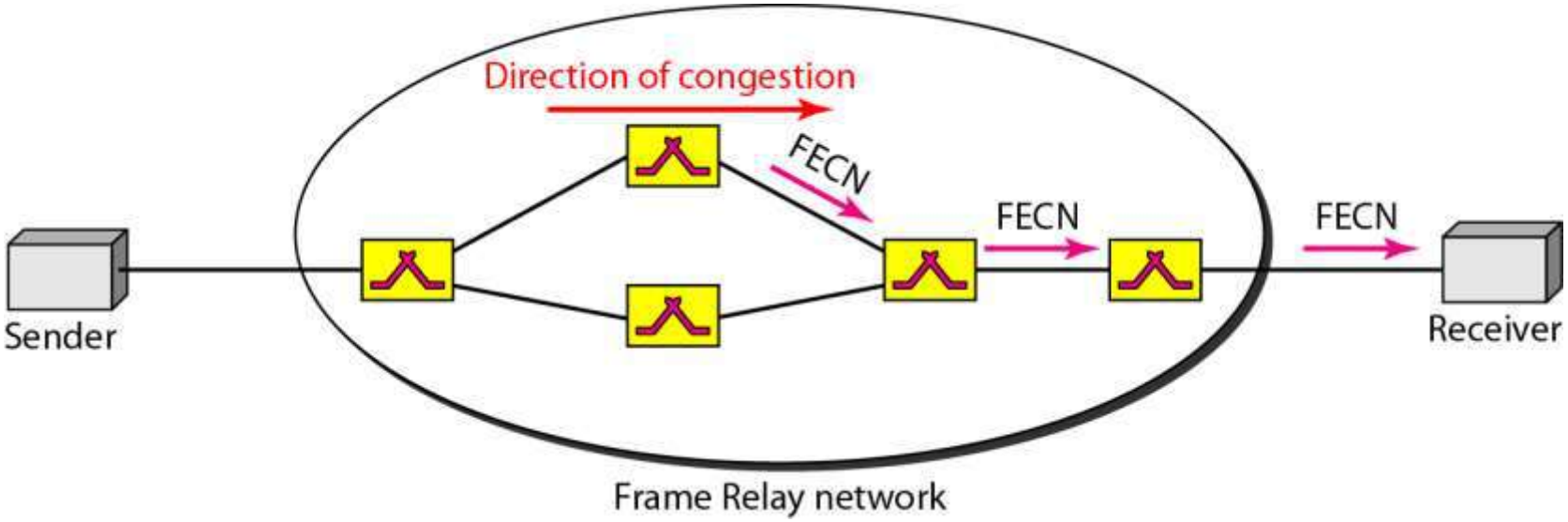
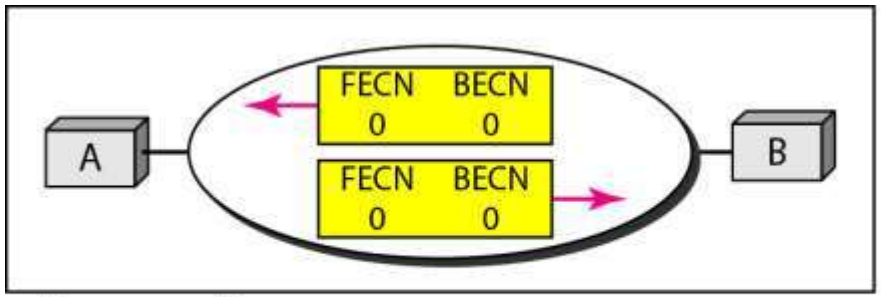
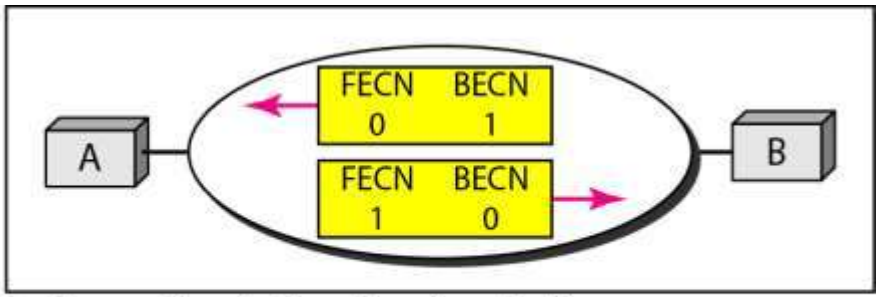


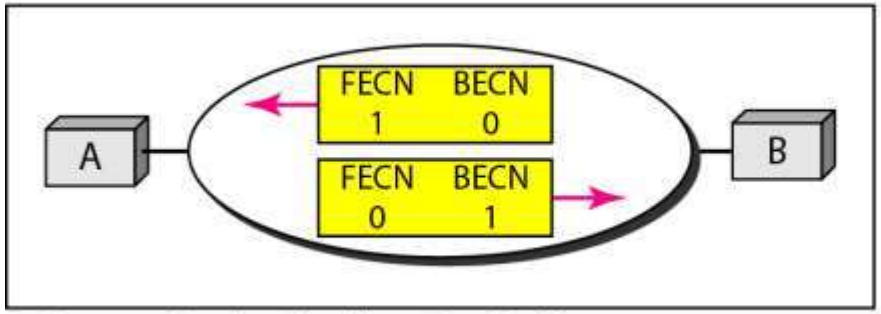
Figure 24.14 *Four cases of congestion*



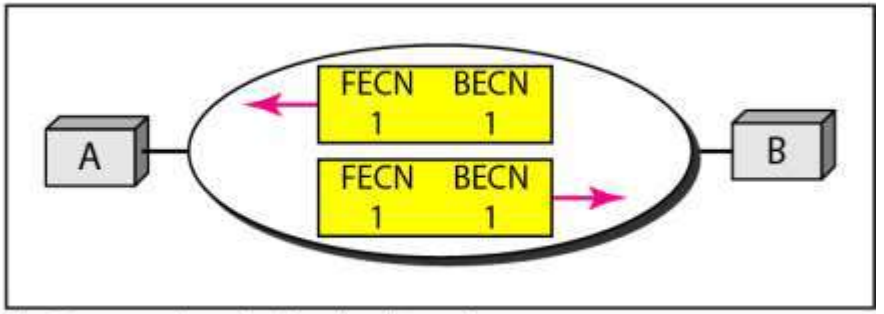
a. No congestion



b. Congestion in the direction A–B



c. Congestion in the direction B–A



d. Congestion in both directions

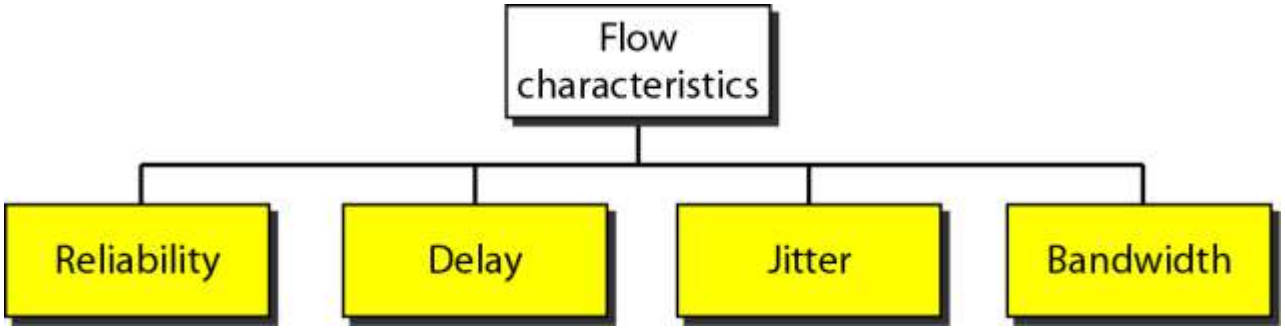
24-5 QUALITY OF SERVICE

Quality of service (QoS) is an internetworking issue that has been discussed more than defined. We can informally define quality of service as something a flow seeks to attain.

Topics discussed in this section:

Flow Characteristics
Flow Classes

Figure 24.15 *Flow characteristics*



Jitter is the variation in delay for packets belonging to the same flow. If four packets Depart at 0,1,2 , 3 and arrive at 20, 21,22,23 all have same delay.
If they arrive at 21,22,19 and 24, all have different delay: 21, 22,19 and 24
High jitter, difference in delay is large.

24-6 TECHNIQUES TO IMPROVE QoS

In this section, we discuss some techniques that can be used to improve the quality of service. We briefly discuss four common methods: scheduling, traffic shaping, admission control, and resource reservation.

Topics discussed in this section:

Scheduling
Traffic Shaping
Resource Reservation
Admission Control

Figure 24.16 *FIFO queue*

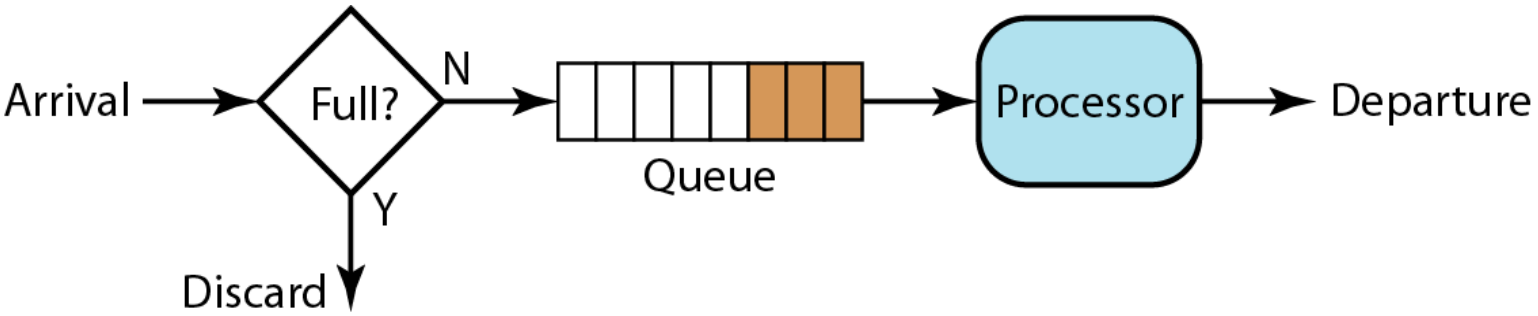
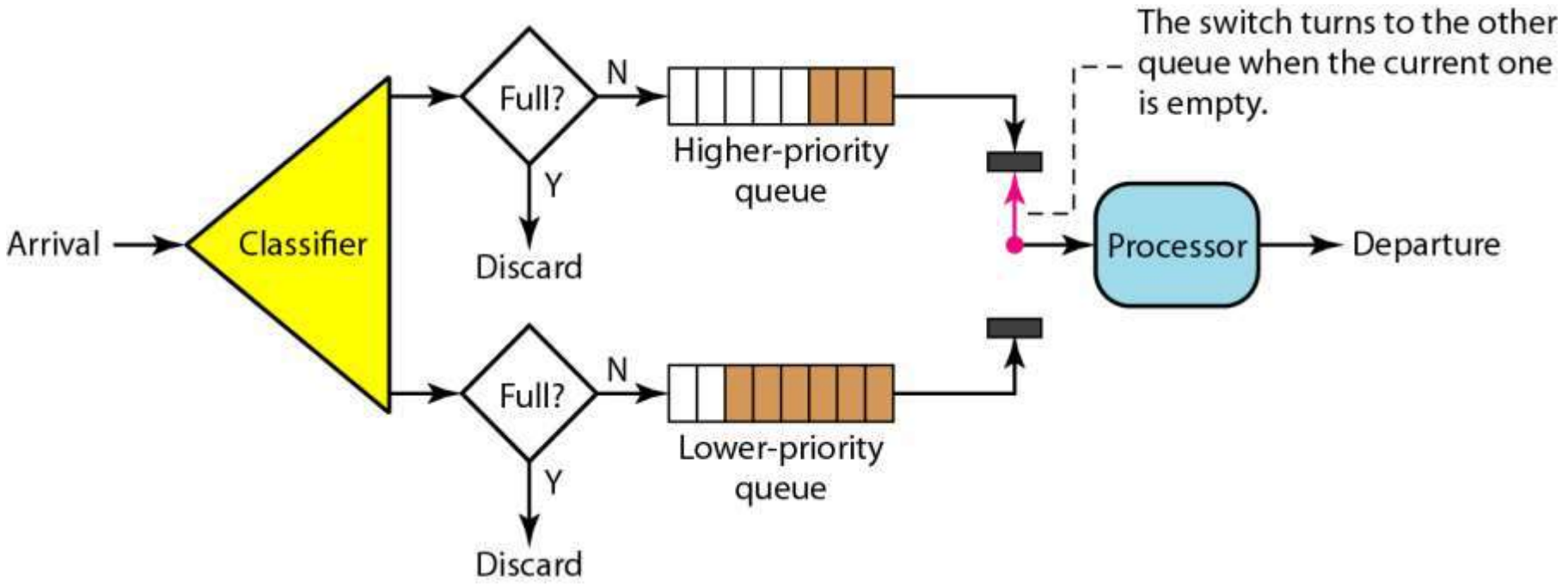


Figure 24.17 *Priority queuing*



Multimedia can reach the destination without delay
Draw is starvation. If continuous flow in the high priority, lower priority class will not be processed

24-6 TECHNIQUES TO IMPROVE QoS

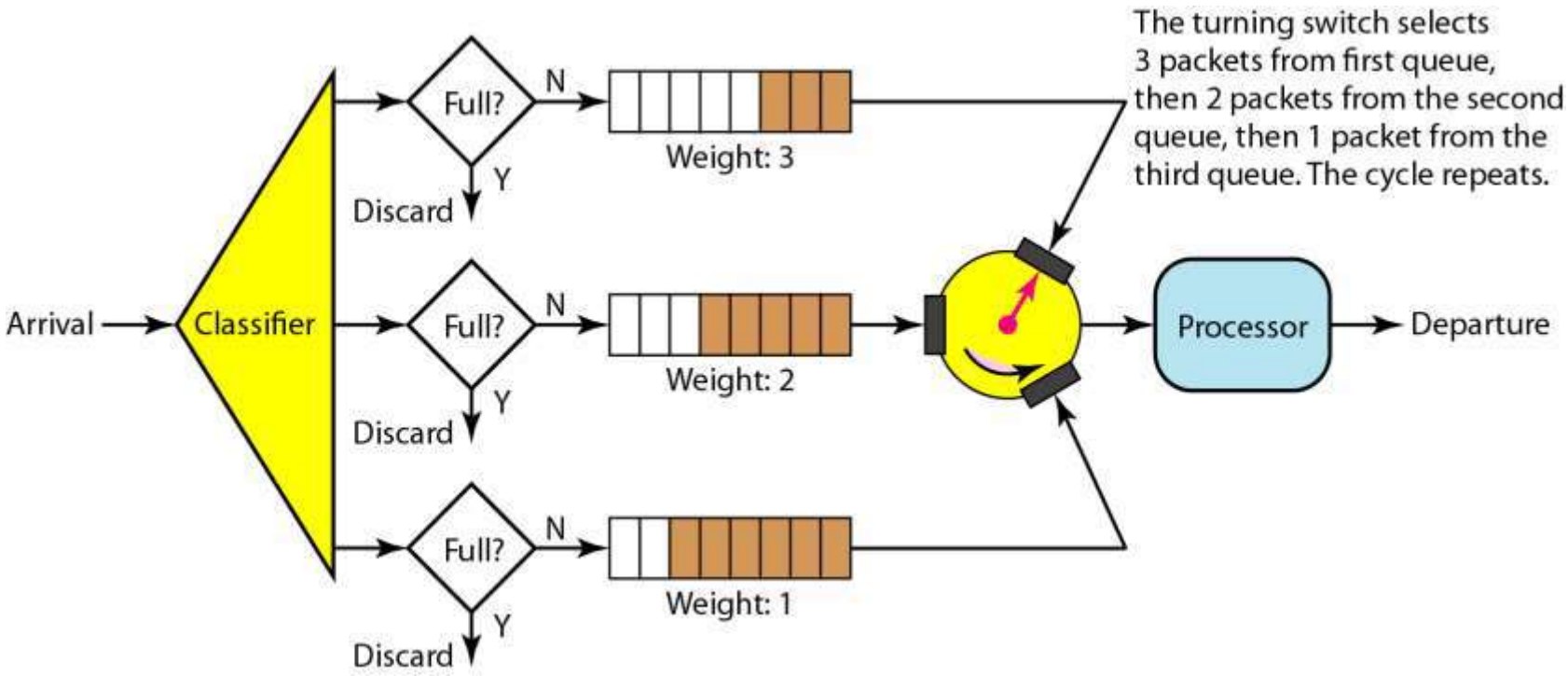
•Multimedia can reach the destination without delay.

Drawback is starvation. If continuous flow in the high priority, lower priority class will not be processed

Weighted fair queuing

If weights are 3,2,1, three packets from first priority, 2 packets from second priority and 1 packet from last priority in round-robin fashion

Figure 24.18 *Weighted fair queuing*



24-6 TECHNIQUES TO IMPROVE QoS

Traffic Shaping

Figure 24.19 *Leaky bucket*

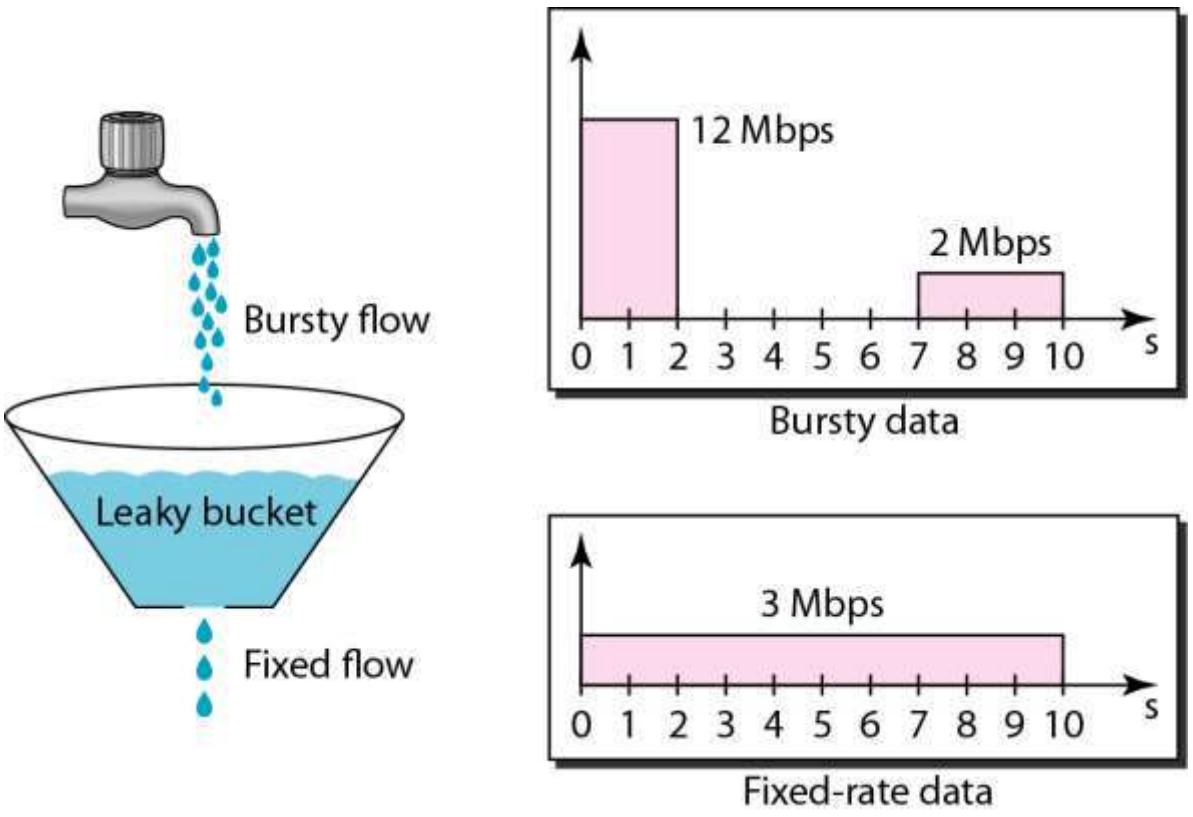
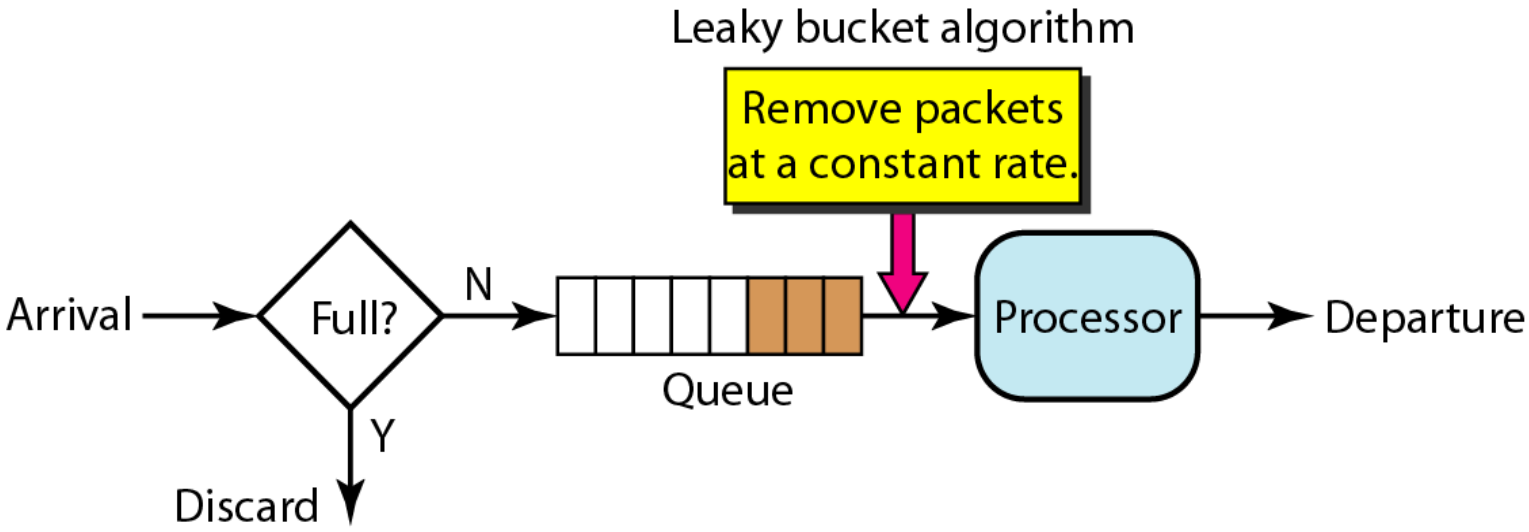


Figure 24.20 *Leaky bucket implementation*



•Initialize a counter to N at the tick of the clock

•If n is greater than the size of the packet, send the packet and decrement the counter by the packet size. Repeat the step until n is smaller than the packet size.

•Reset the counter and go to step 1

Token bucket

If n is 100, and the host is idle for 100 ticks, the bucket collects 10,000 tokens
Now the host can consume all these token in one tick with 10,000 bytes
Or the host takes 1000 ticks with 10 cells per tick

Host can send the bursty data as long as the bucket is not empty

Resources such as buffer, bandwidth, CPU time and so on are reserved beforehand.

Router accepts or rejects a flow based on predefined parameters called Flow specifications

Remote Logging, Electronic Mail, and File Transfer

Copyright © The McGraw-Hill Companies, Inc. Permission required for reproduction or display.

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.

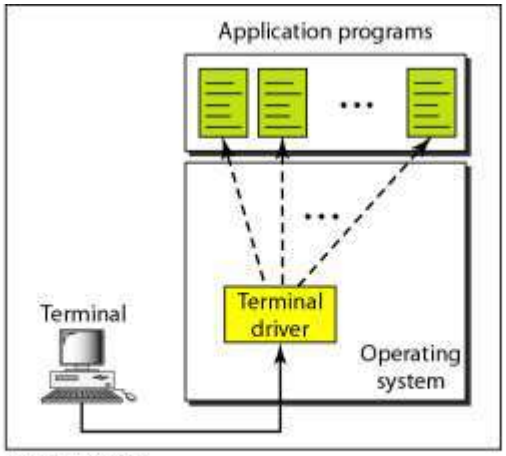
Topics discussed in this section:

TELNET

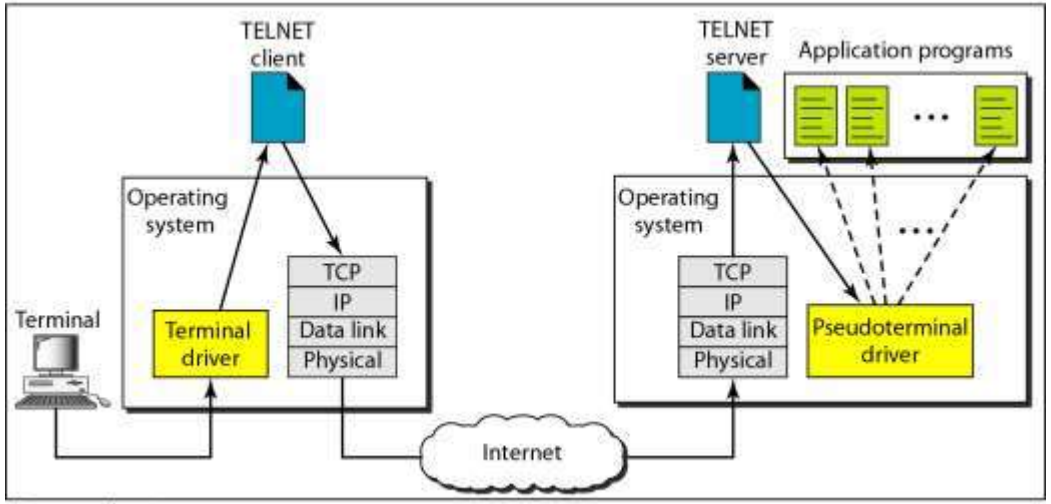
Note

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*

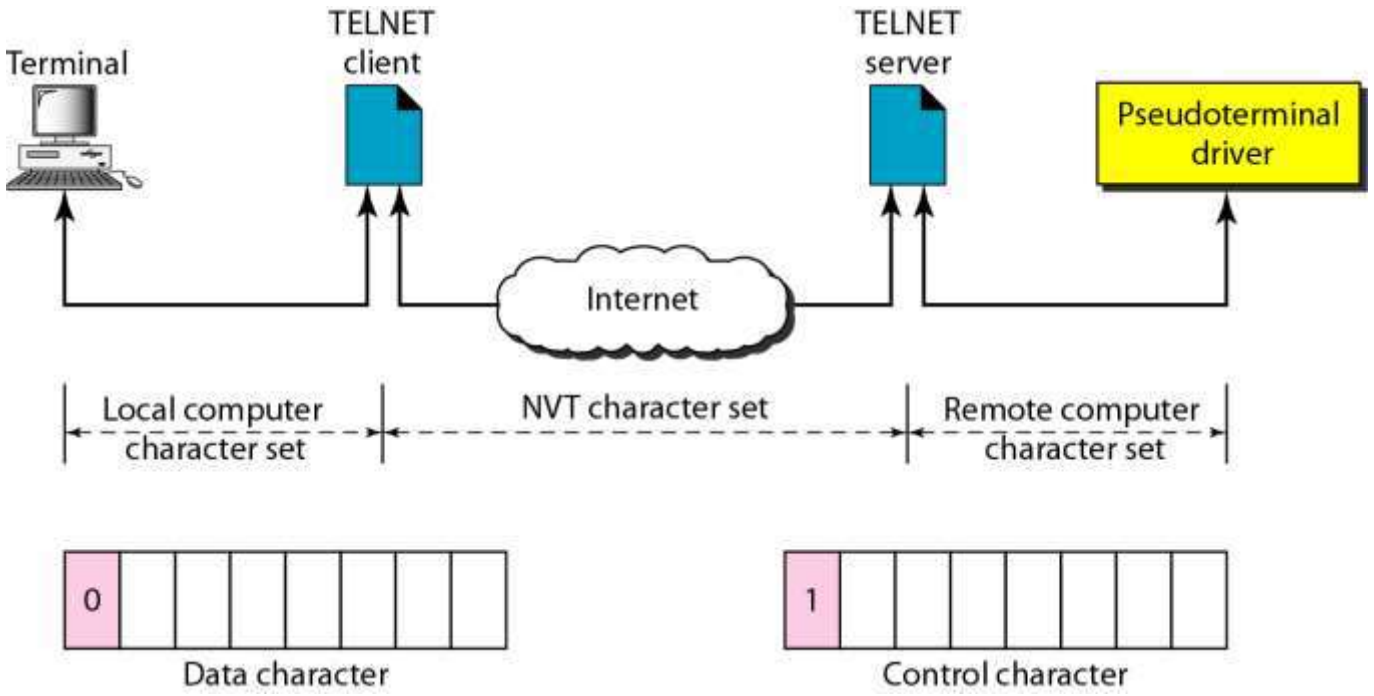


Table 26.1 *Some NVT control characters*

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

Figure 26.3 *An example of embedding*

Table 26.2 *Options*

Code	*Option*	*Meaning*
0	Binary	Interpret as 8-bit binary transmission.
1	Echo	Echo the data received on one side to the other.
3	Suppress go ahead	Suppress go-ahead signals after data.
5	Status	Request the status of TELNET.
6	Timing mark	Define the timing marks.
24	Terminal type	Set the terminal type.
32	Terminal speed	Set the terminal speed.
34	Line mode	Change to line mode.

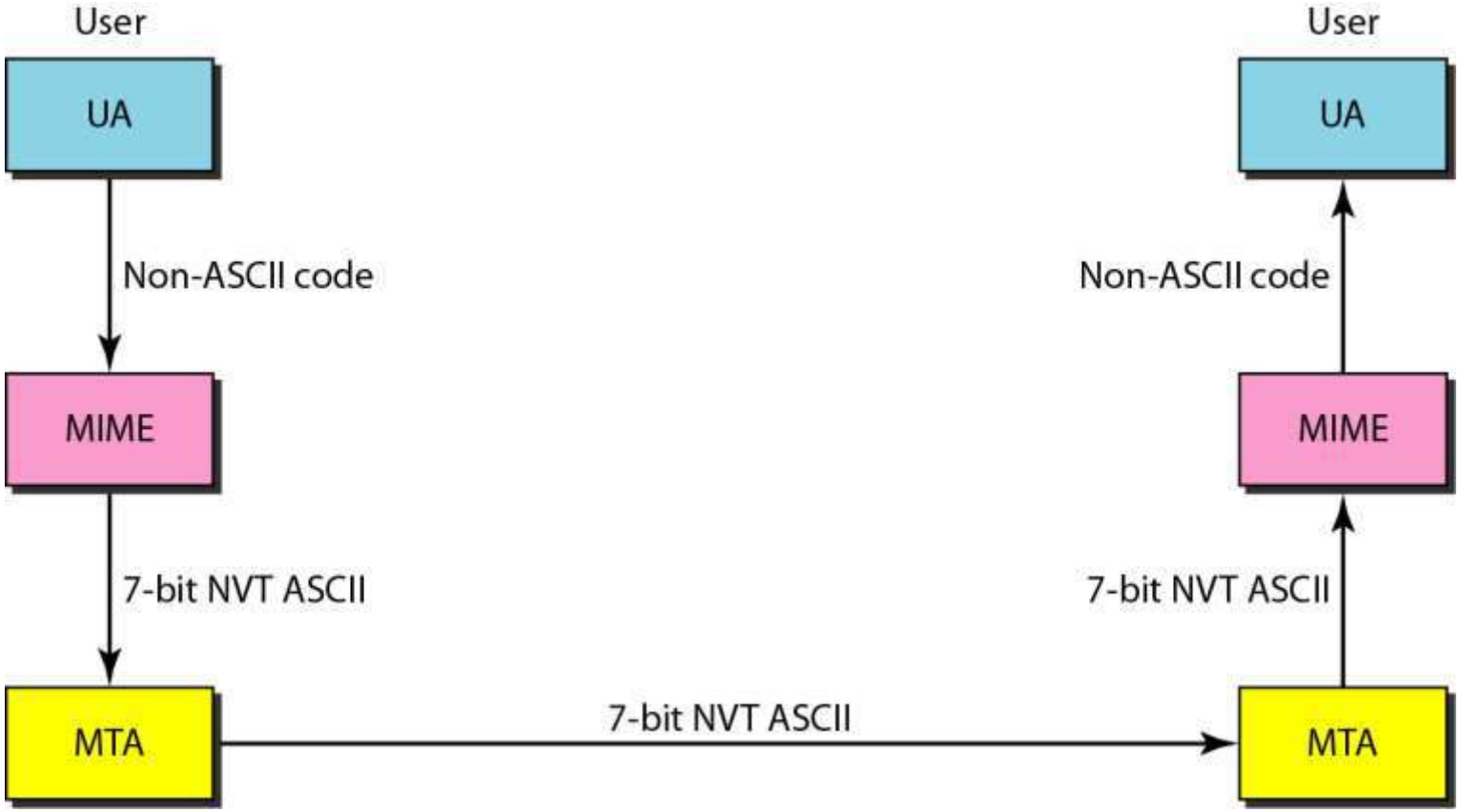
Table 26.3 *NVT character set for option negotiation*

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

MIME (Multipurpose Internet Mail Extensions)

- *Electronic mail can send message only in 7-bit ASCII format.*
- *For example, it cannot be used for languages that are not supported by 7-bit ASCII (French, German, Russian, Chinese and Japanese).*
- *Also it cannot be used for sending binary files or video or audio data.*
- *MIME is a supplementary protocol that allows non-ASCII data to be send through email.*

Figure 26.14 *MIME*



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 26.9 *Fourth scenario in electronic mail*

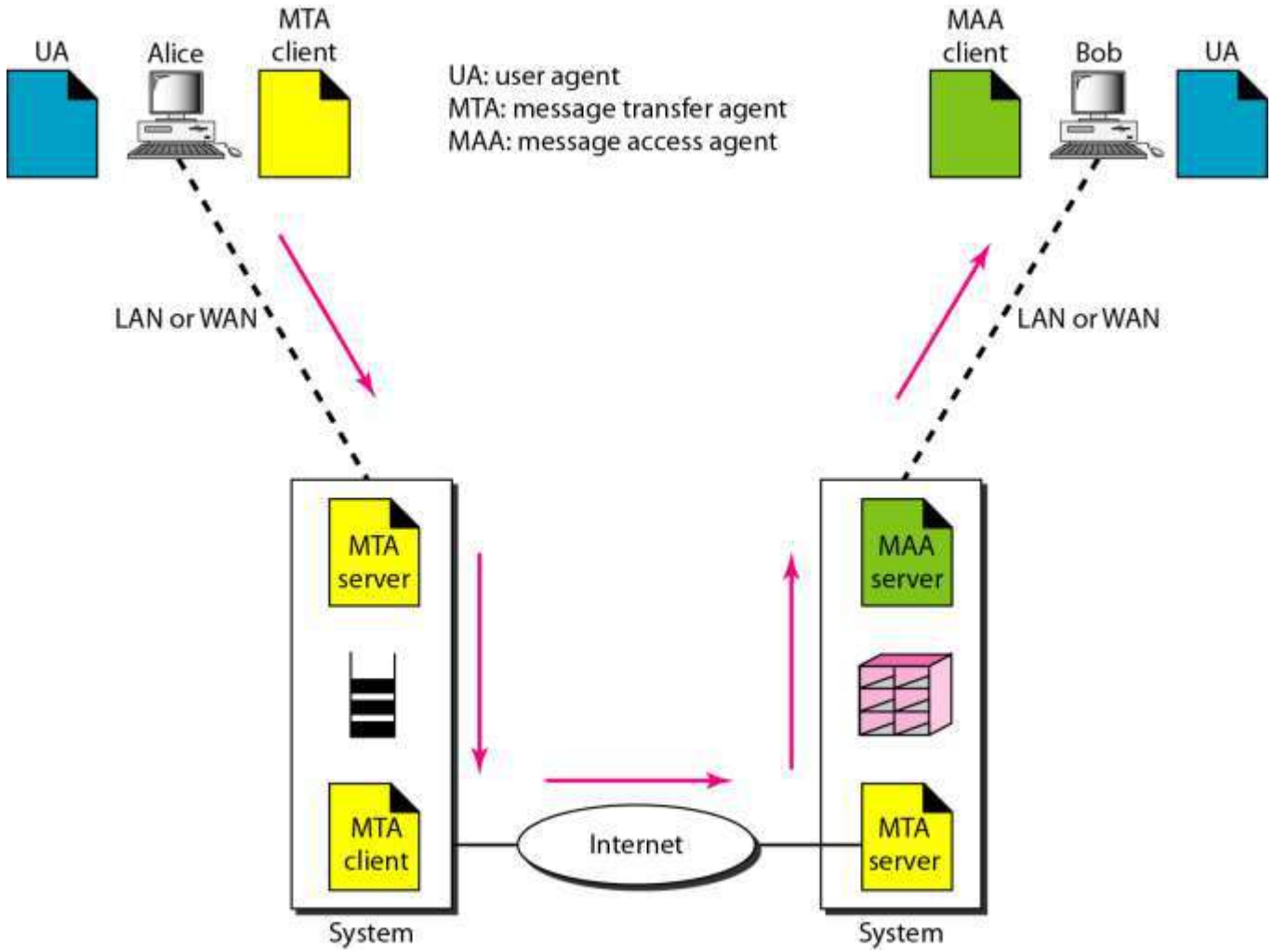


Figure 26.11 *Services of user agent*

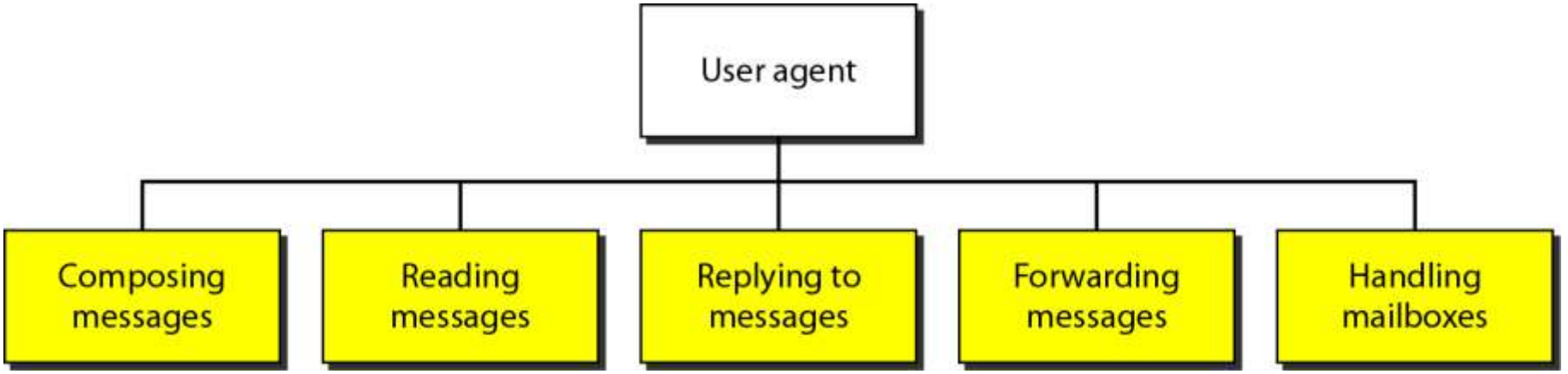


Figure 26.15 *MIME header*

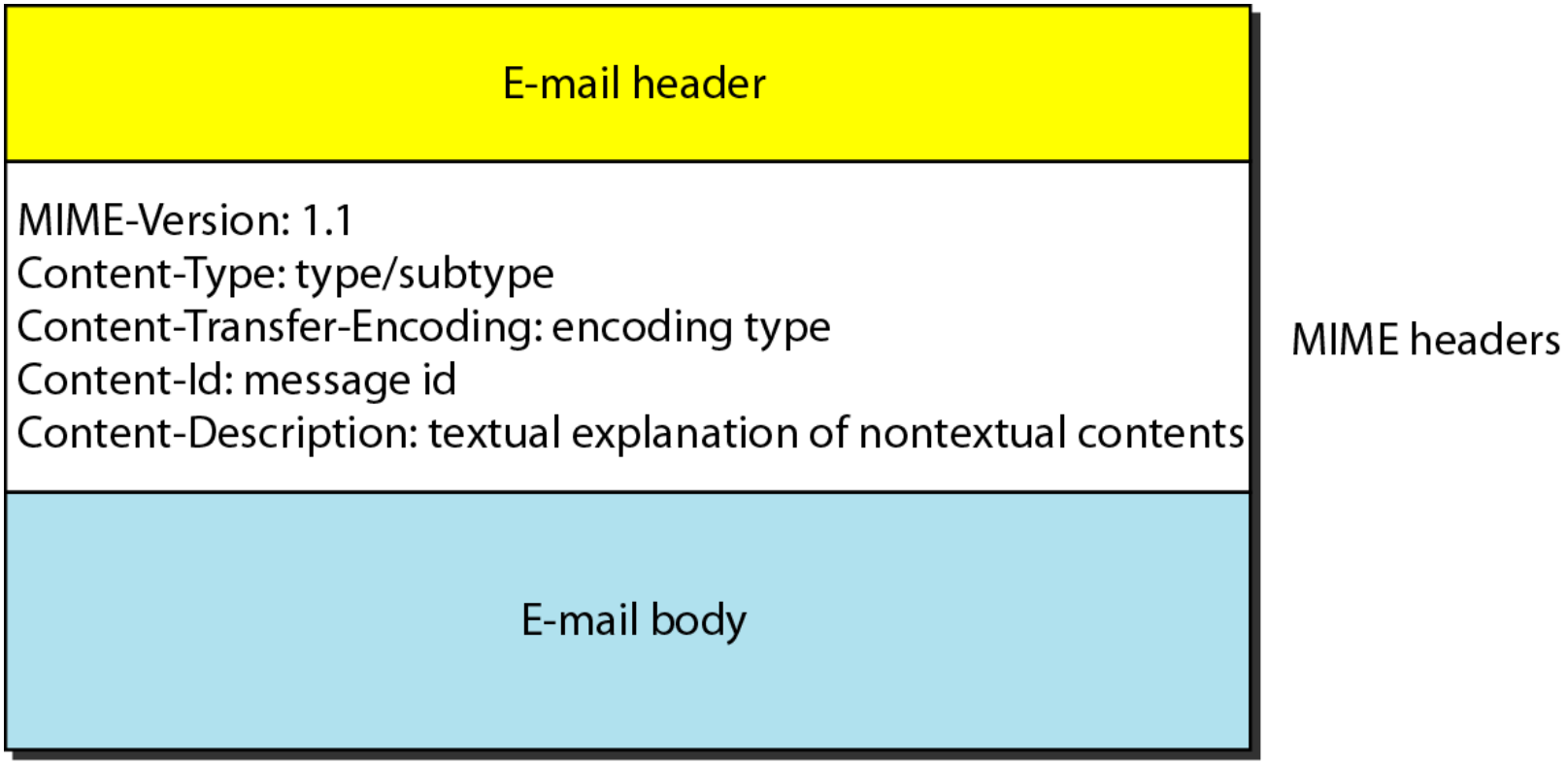


Table 26.5 *Data types and subtypes in MIME*

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)

SMTP (Simple Mail Transfer Protocol)

Message Transfer Agent

- *To send a mail, a system must have client MTA and to receive mail, system must have server MTA*
- *The formal protocol that defines the MTA client and server in the internet is called SMTP*
- *SMTP simply defines how command and responses must be sent back and forth*
- *SMTP uses command s and responses to transfer messages between an MTA client and MTA server*
- *SMTP defines 14 commands, first five are mandatory*
- *It defines 4 category of responses*
- *Left most digit of the code defines the 4 categories.*

Figure 26.16 *SMTP range*

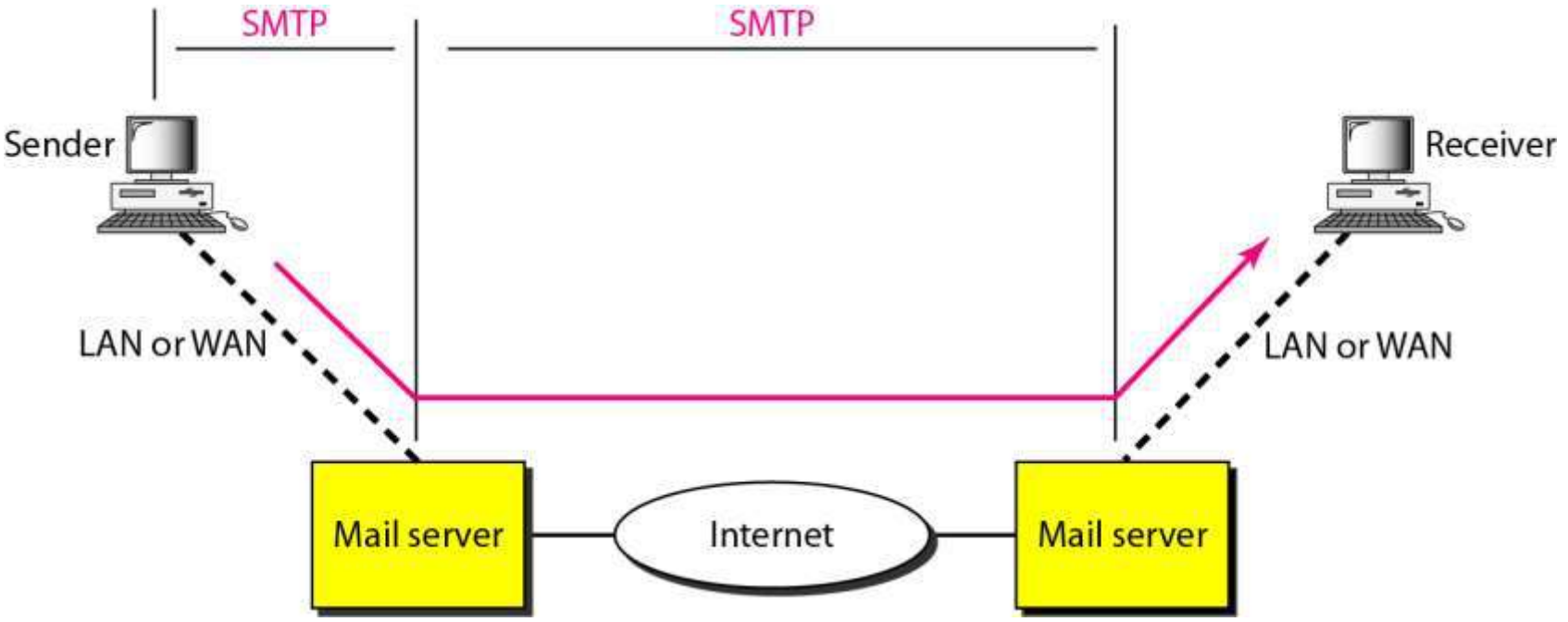
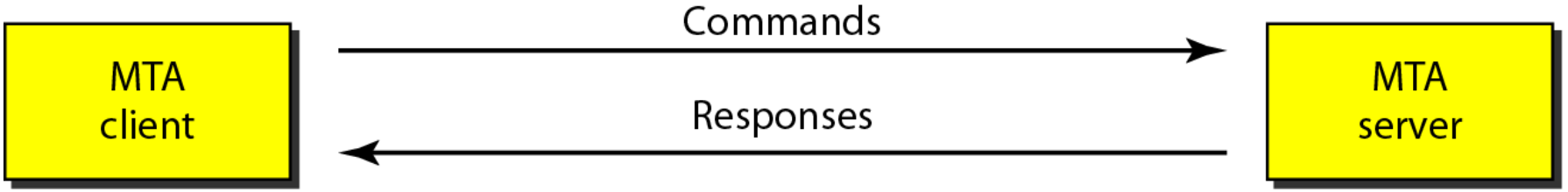


Figure 26.17 *Commands and responses*



The process of transferring a mail message occurs in three phases

1. Connection establishment
2. Mail transfer
3. Connection termination

Figure 26.18 *Command format*

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

POP3 (Post office Protocol, version 3)

Message Access Agent

❖*SMTP is used for pushing the data from client to mail server and from mail server to other mail server since the direction of bulk data is from client to server*

❖*At the other end the direction of bulk data is from Mail server to client. So third stage uses pull protocol*

- *POP 3 has two modes: Delete mode and keep mode*
- *In delete mode, message is deleted from the mail box after each retrieval . When user uses primary computer*
- *In Keep mode, messages remains in the mail box after retrieval User access the mail away from primary computer*

Figure 26.19 *POP3 and IMAP4*

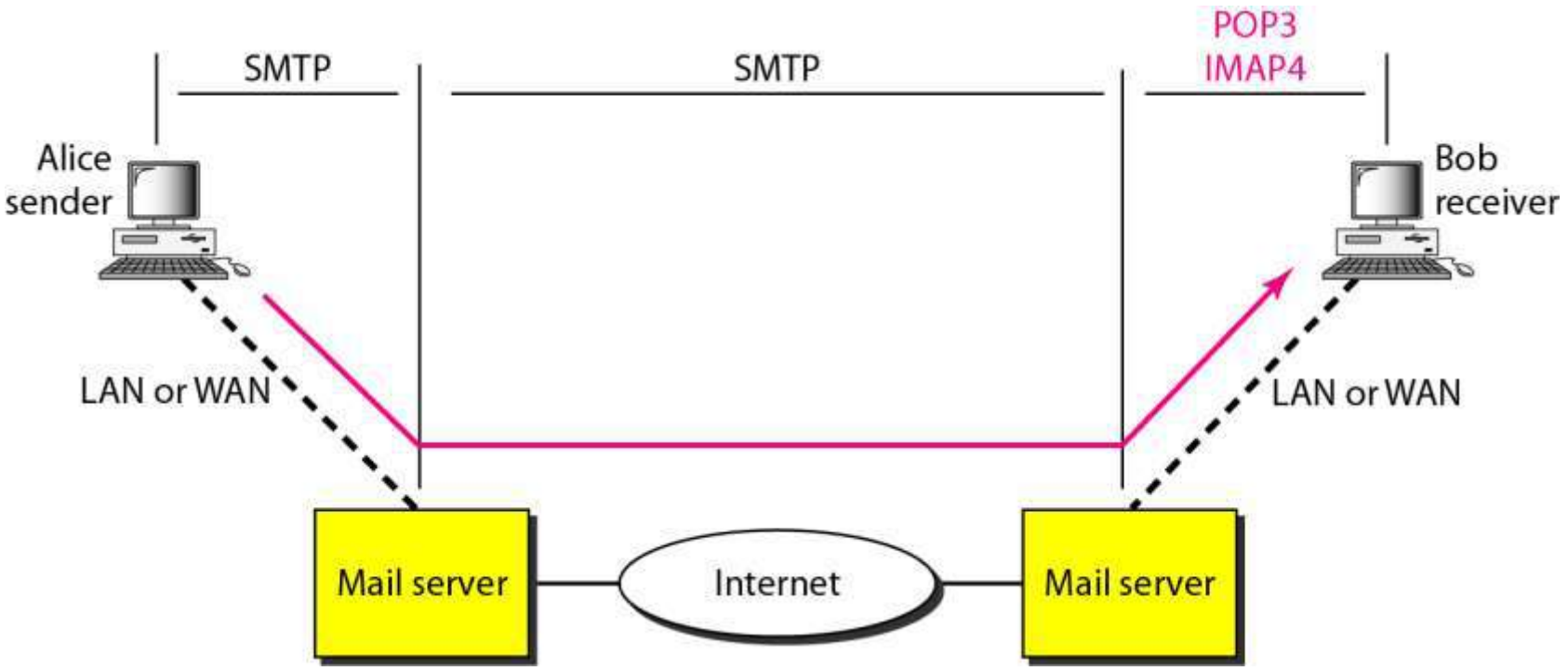


Figure 26.20 *The exchange of commands and responses in POP3*

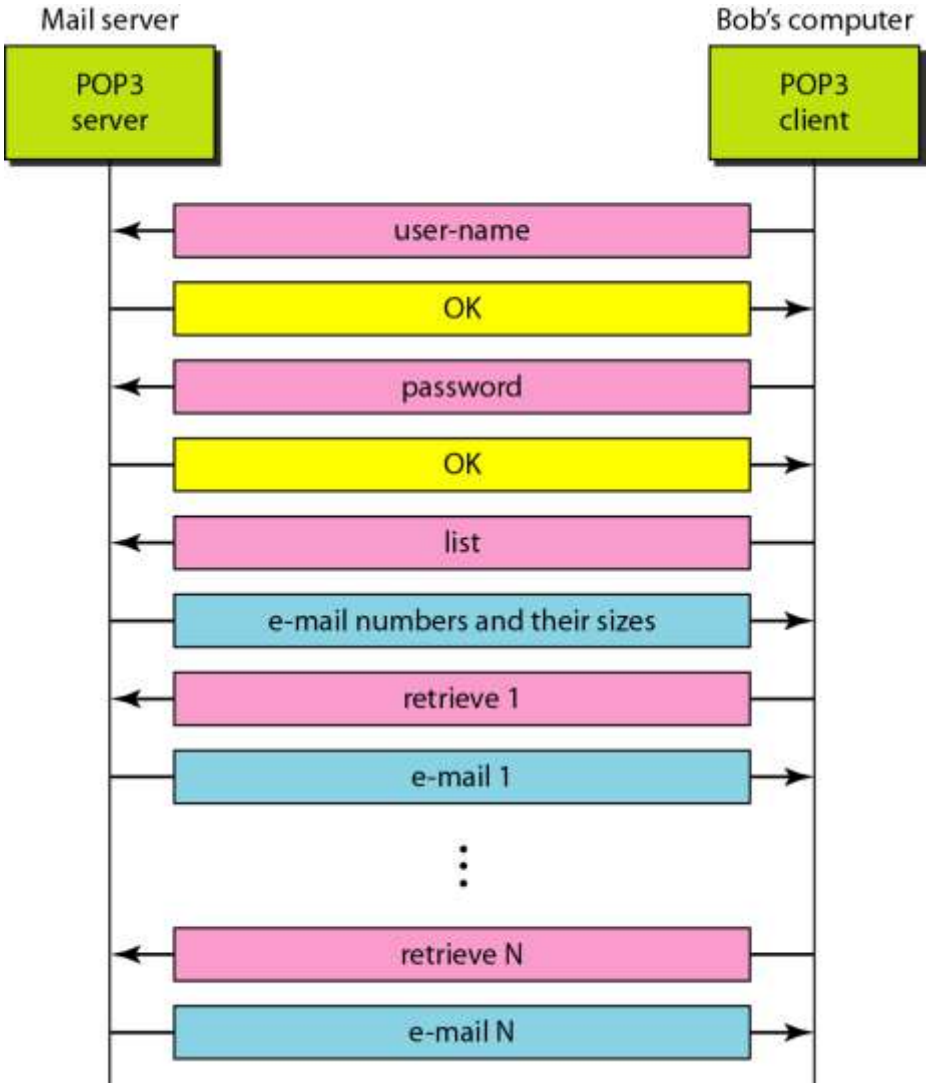
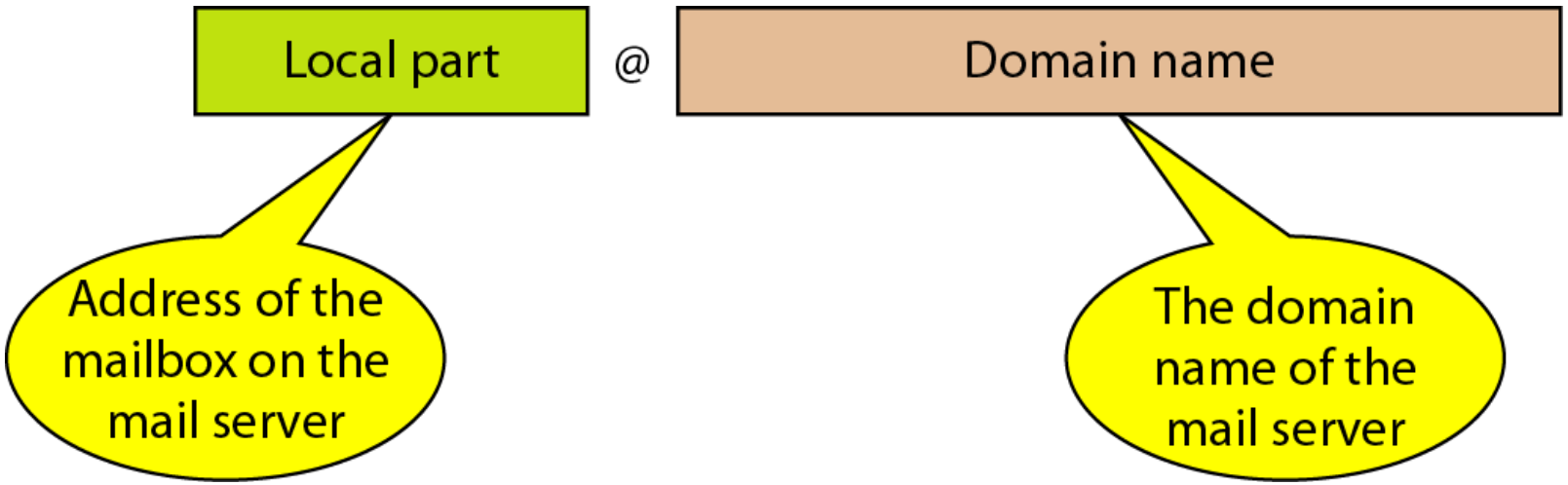


Figure 26.13 *E-mail address*



Assignment II

- How do you see for the future of networking and the Internet". Make a summary of views. What are your own reflections on the issues pointed out?
- SMS, iMessage and WhatsApp are all smart phone real-time messaging systems. After doing some information research in the Internet, for each of these systems, explain each and write a paragraph about the protocols they use.
- Try to find out the hostname and IP address of a computer you use to access the Internet. How can you do that? What is subnet mask?

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.

Topics discussed in this section:

File Transfer Protocol (FTP)
Anonymous 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*

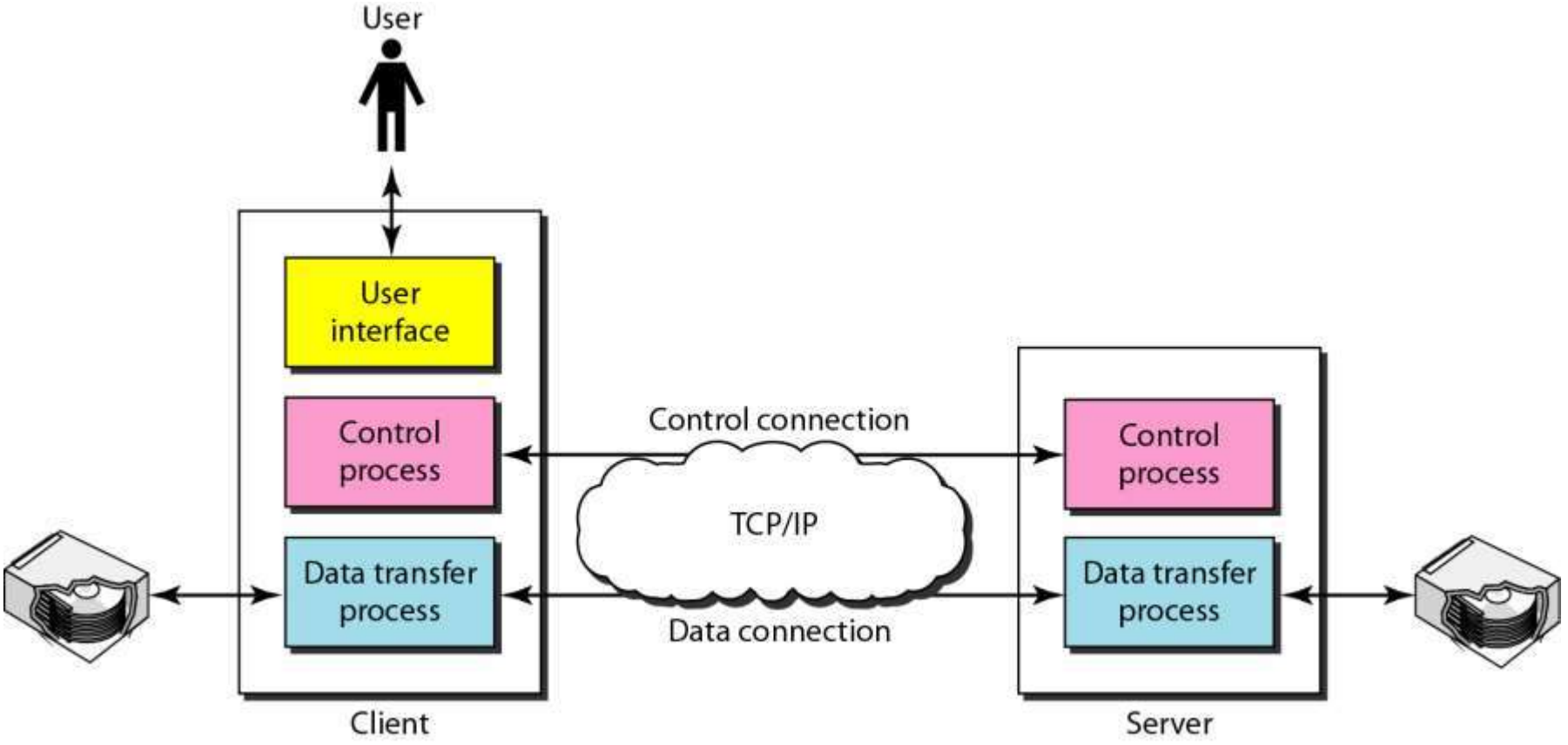


Figure 26.22 *Using the control connection*

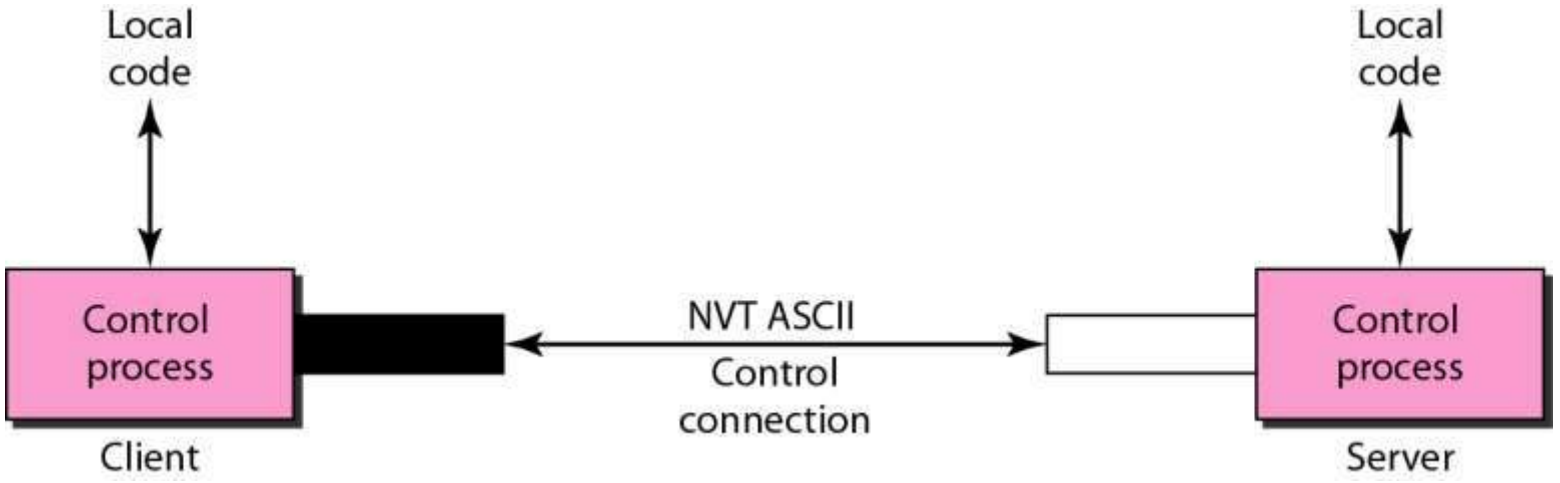
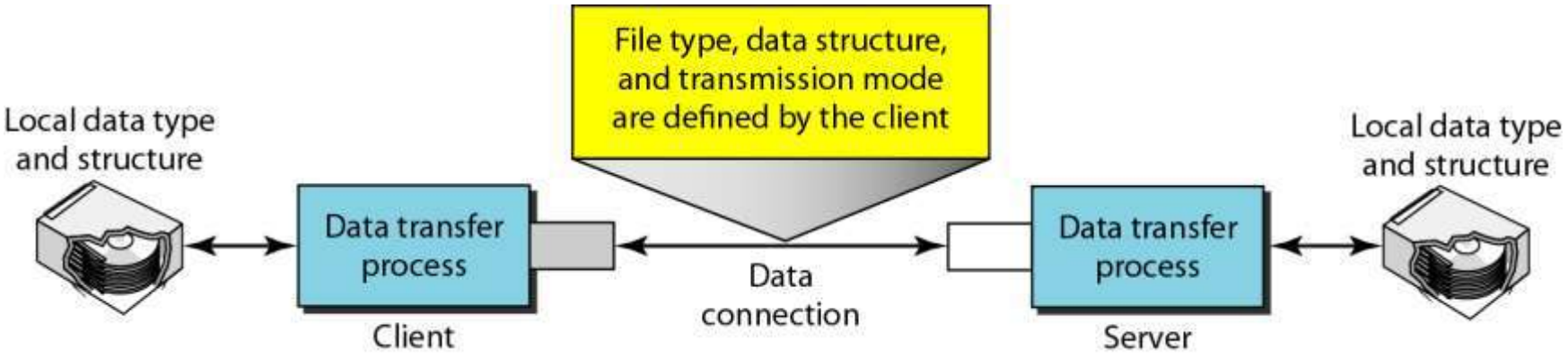


Figure 26.23 *Using the data connection*



Communication Over Data Connection

- *Wants to transfer file through the data connection*
- *File transfer occurs under the control of 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. Note that FTP treats a list of directory or file names as a file. It is sent over the data connection.

File type

❖*FTP can transfer one of the following file type across the data connection*

1. *ASCII file*
2. *EBCDIC file*
3. *Image file – No encoding is required-sending as a bit stream*

Data structure

❖*FTP can transfer use one of following strucure-interpretations*

1. *File structure-File is continuous stream of bytes*
2. *Record- File is divided into records*
3. *Page – File is divided into pages*

Mode of Transmission

❖*FTP can use one of the following mode*

1. *Stream Mode – delivered from FTP to TCP as a continuous stream of bytes*
2. *Block mode*
3. *Compressed mode-If the file is big, data is compressed*

27-1 WWW ARCHITECTURE

The WWW today is a distributed client/server service, in which a client using a browser can access a service using a server. However, the service provided is distributed over many locations called sites.

Topics discussed in this section:

Client (Browser)
Server
Uniform Resource Locator
Cookies

Figure 27.1 *Architecture of WWW*

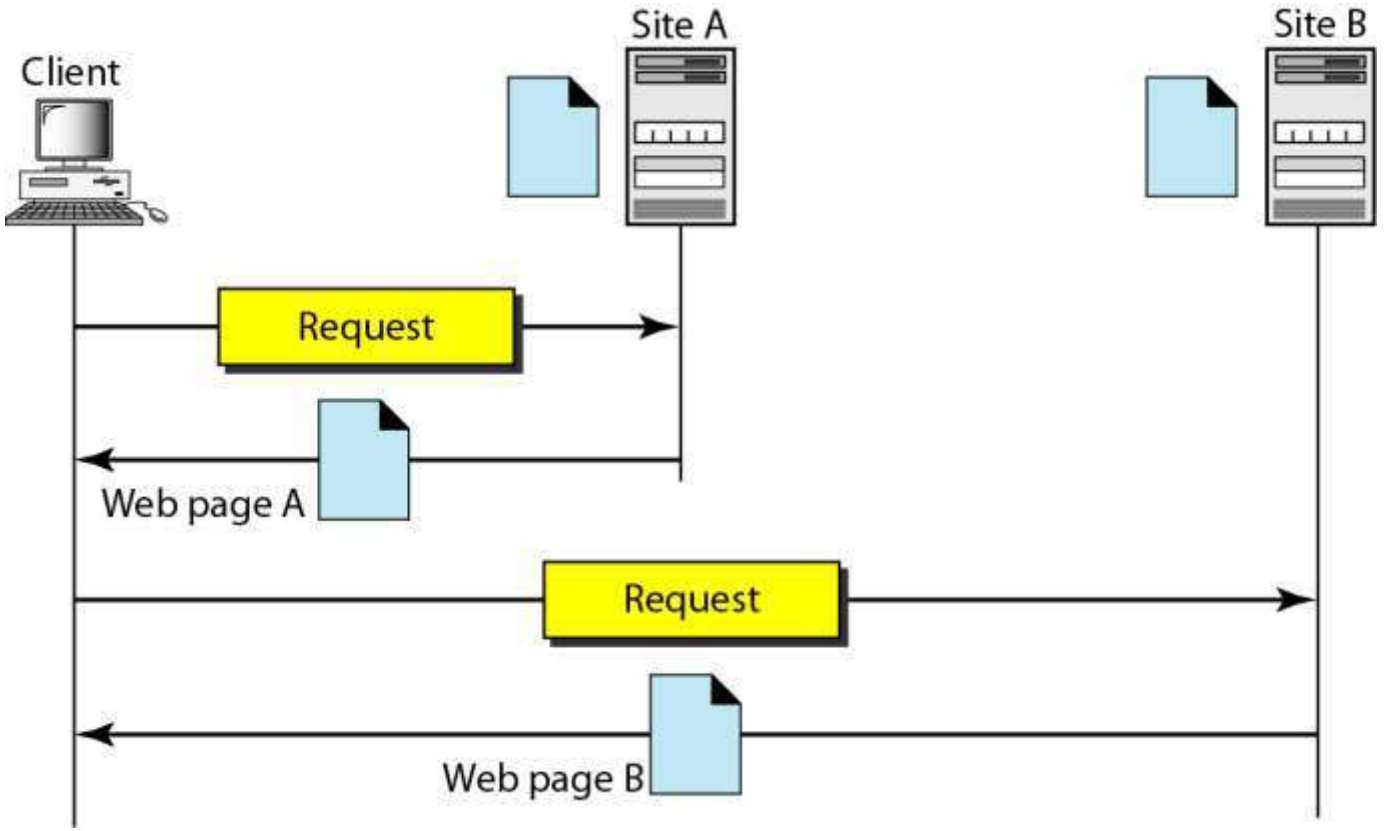
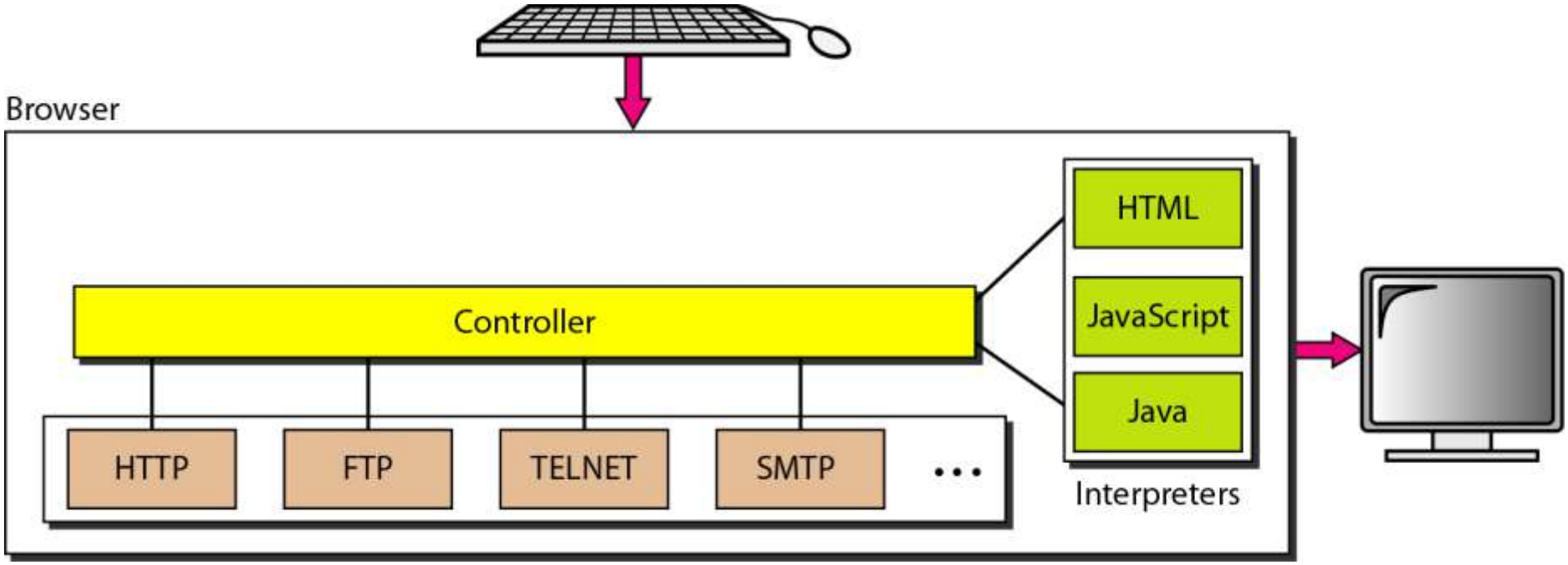


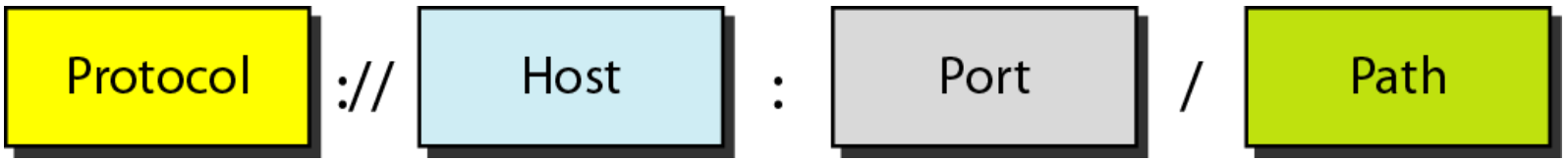
Figure 27.2 *Browser (Client)*



Controller, Client Programs and Interpreters

Figure 27.3 *URL (Universal Resource Locator)*

Protocol – Client/server pgm used to retrieve the document
Host- The computer on which information is located
Port – Port number of the server. Optional
Path – Path name of the file where the information is located



27-2 WEB DOCUMENTS

The documents in the WWW can be grouped into three broad categories: static, dynamic, and active. The category is based on the time at which the contents of the document are determined.

Topics discussed in this section:

Static Documents
Dynamic Documents
Active Documents

Figure 27.4 *Static document*

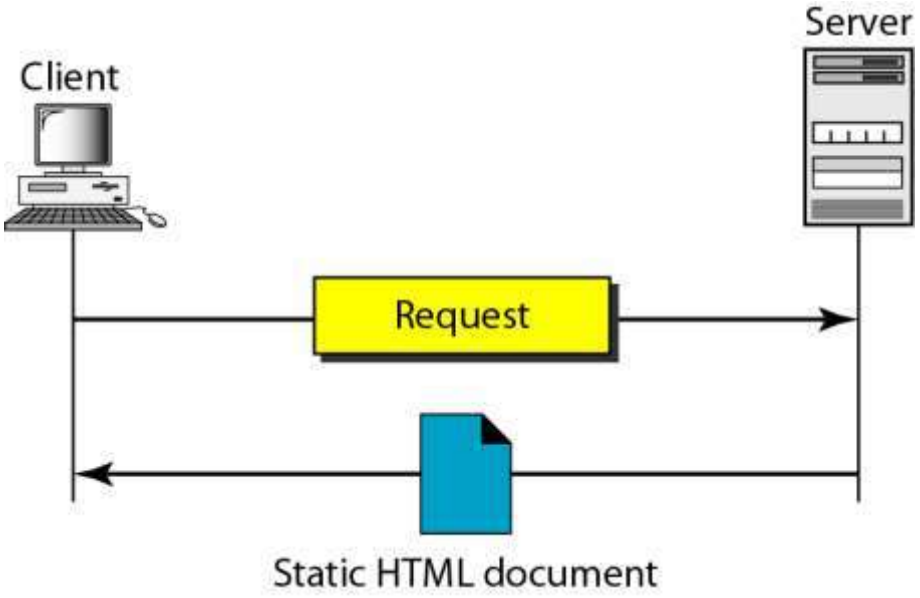


Figure 27.5 *Boldface tags*

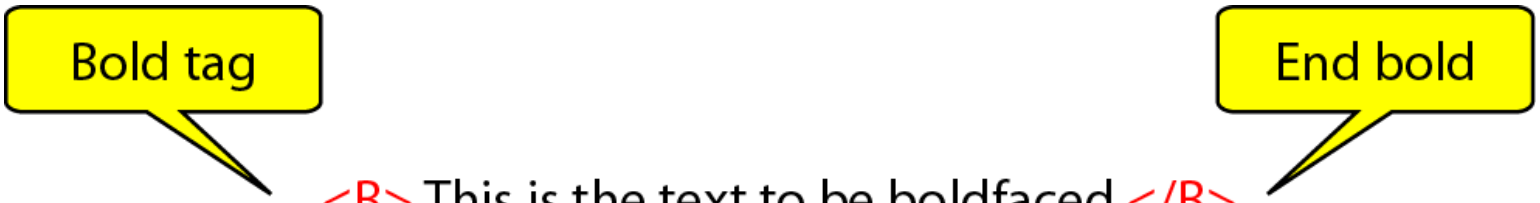


Figure 27.6 *Effect of boldface tags*

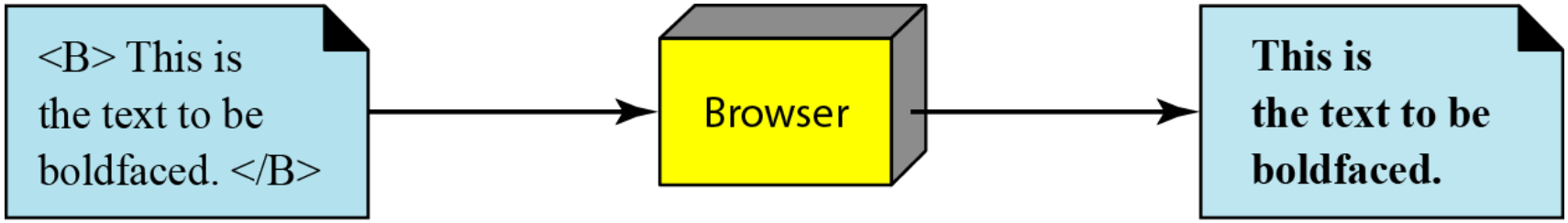


Figure 27.7 *Beginning and ending tags*

a. Beginning tag

b. Ending tag

Figure 27.8 *Dynamic document using CGI*

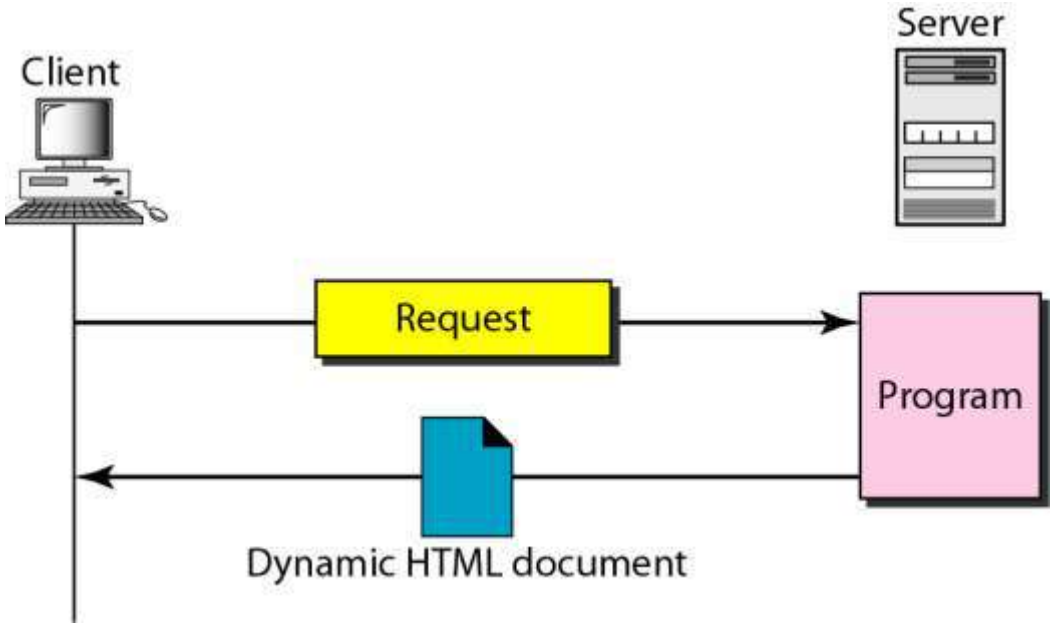
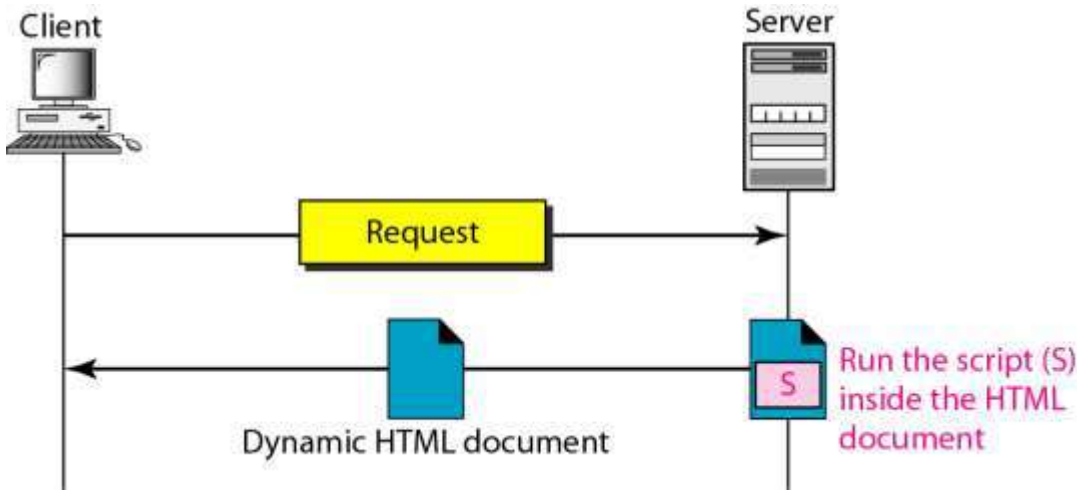


Figure 27.9 *Dynamic document using server-site script*



Note

Dynamic documents are sometimes referred to as server-site dynamic documents.

Figure 27.10 *Active document using Java applet*

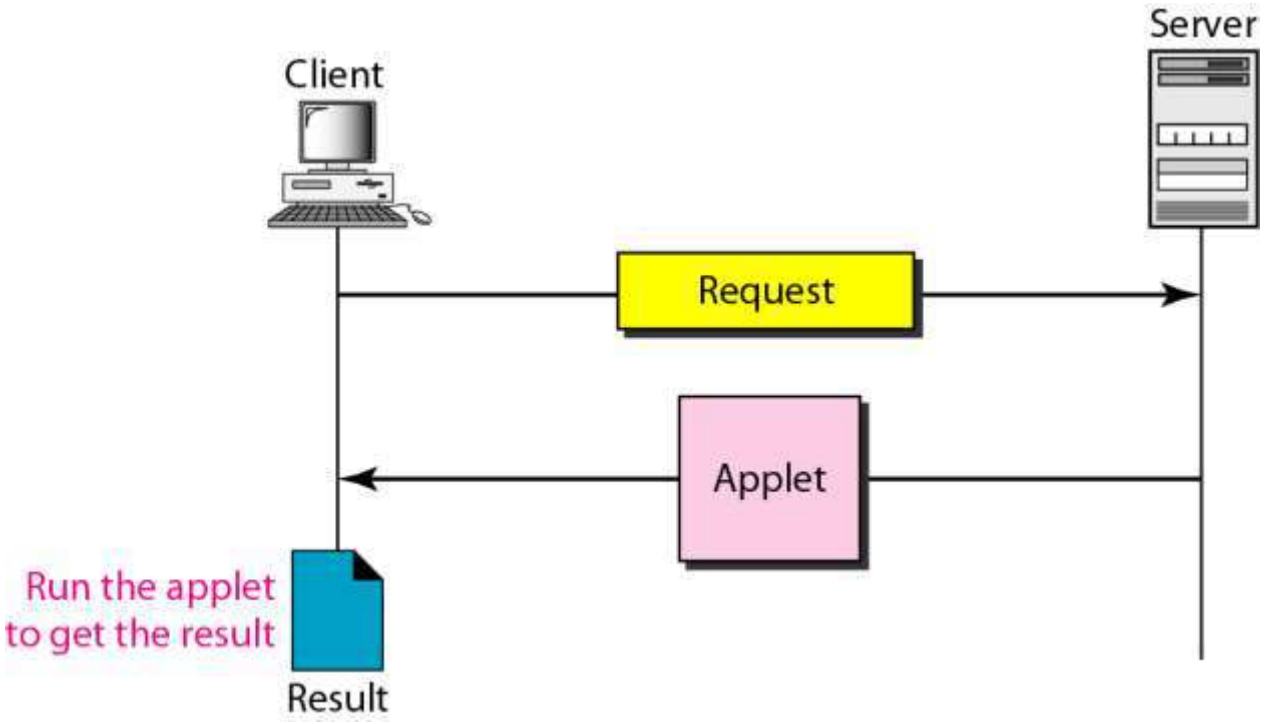
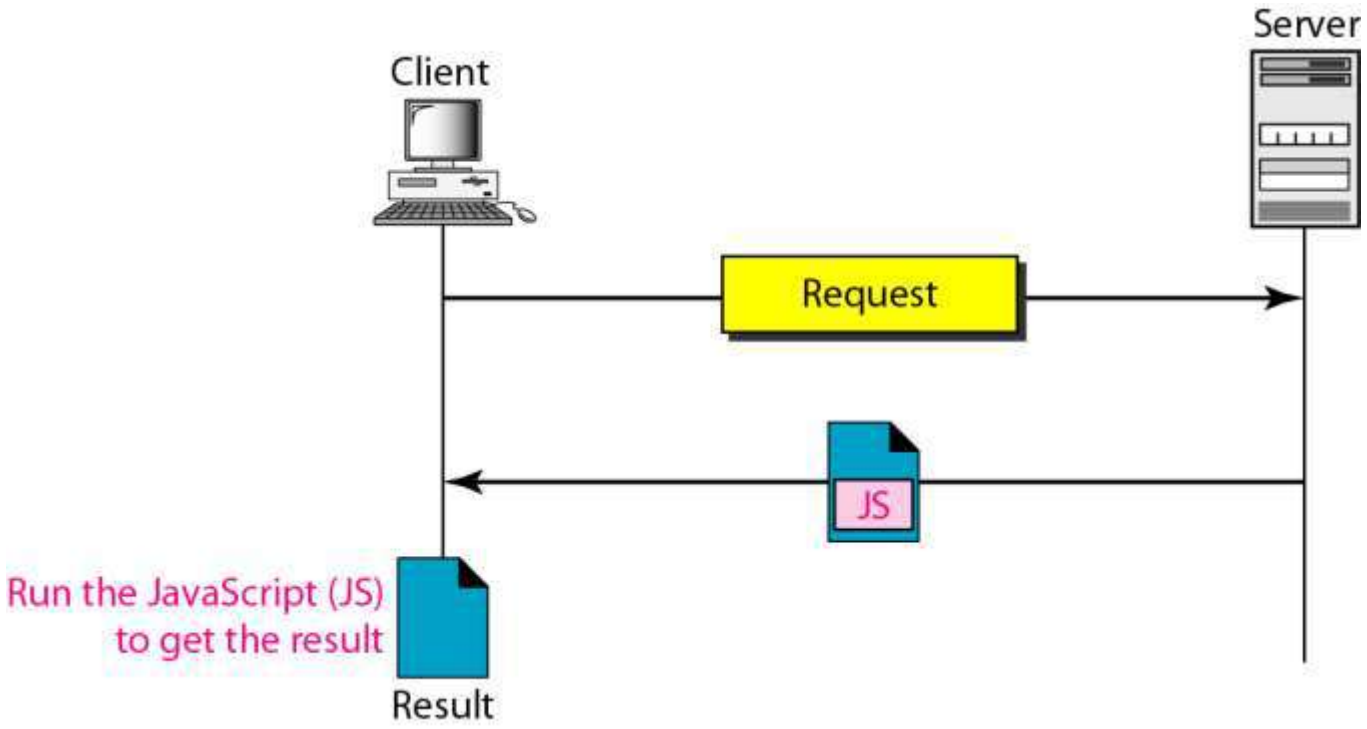


Figure 27.11 *Active document using client-site script*



Note

Active documents are sometimes referred to as client-site dynamic documents.

27-3 HTTP

The Hypertext Transfer Protocol (HTTP) is a protocol used mainly to access data on the World Wide Web. HTTP functions as a combination of FTP and SMTP.

Topics discussed in this section:

HTTP Transaction

Persistent Versus Nonpersistent Connection

Note

HTTP uses the services of TCP on well-known port 80.

Figure 27.12 *HTTP transaction*

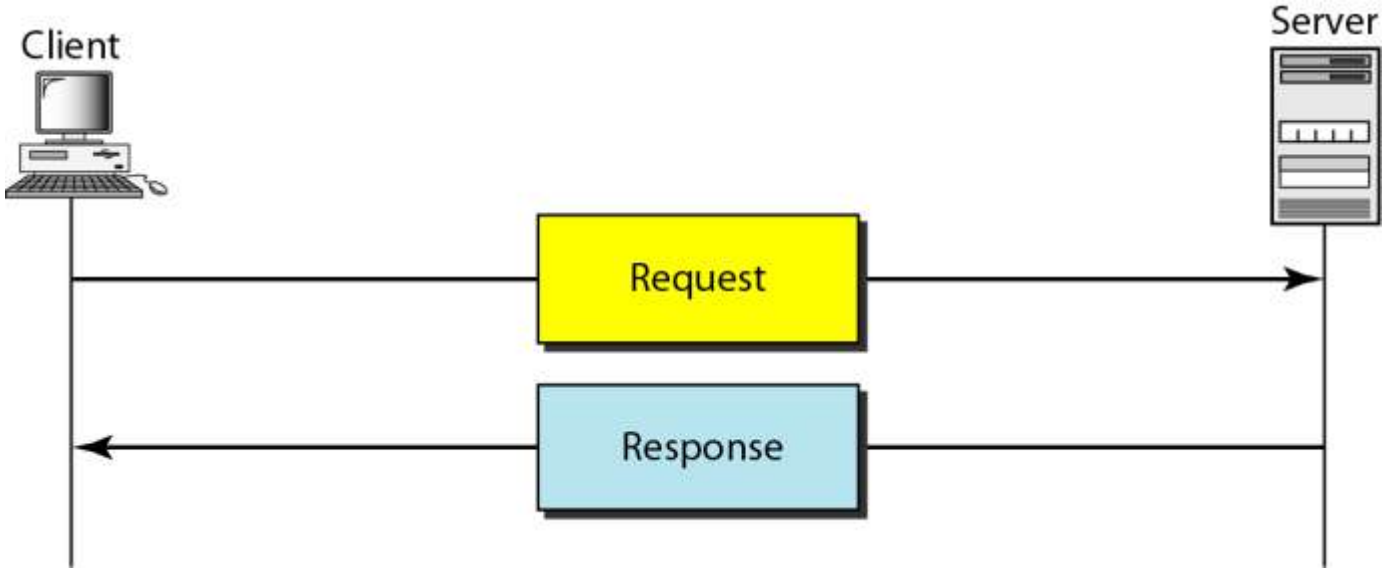


Figure 27.13 *Request and response messages*

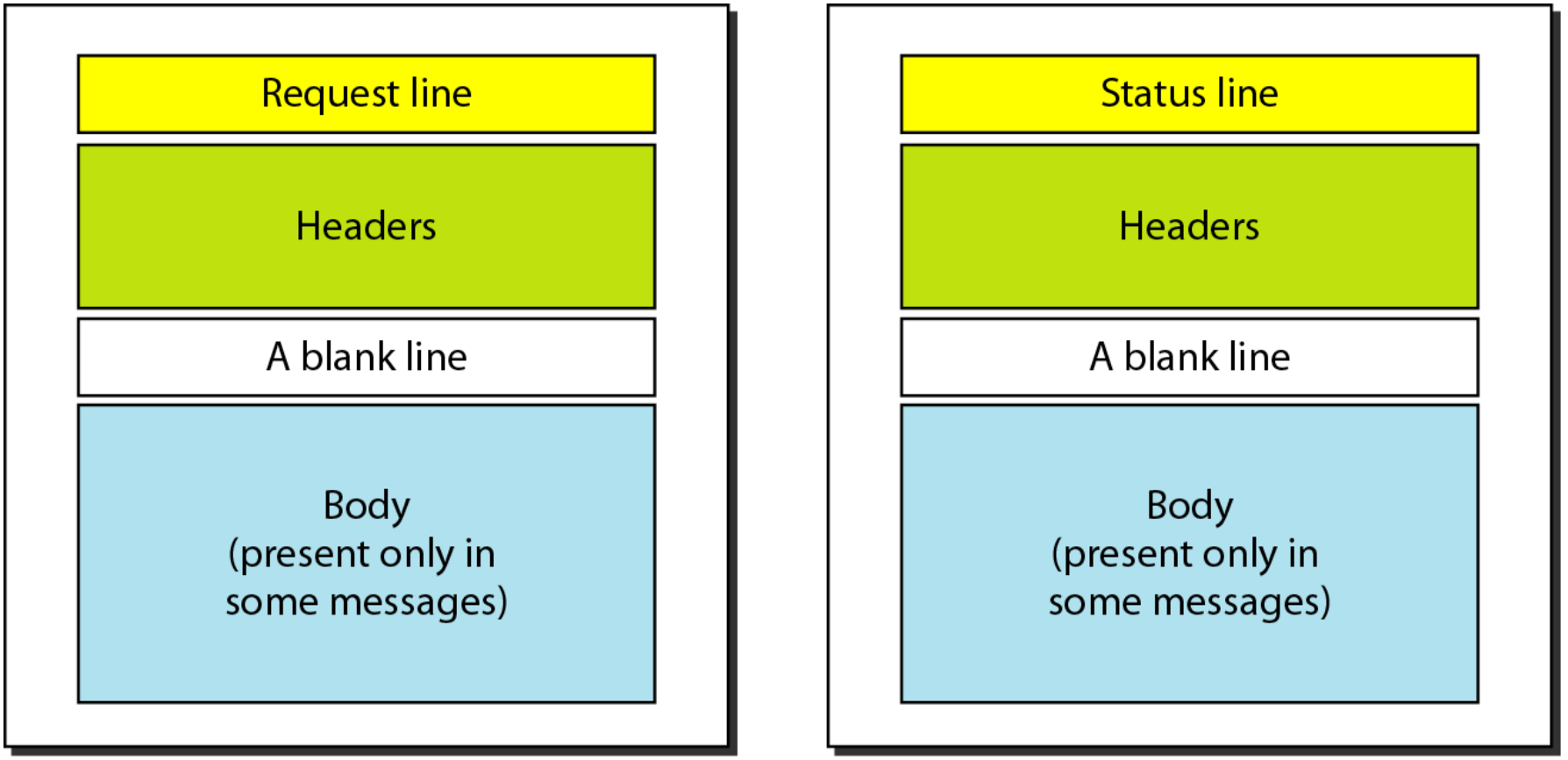


Figure 27.14 *Request and status lines*

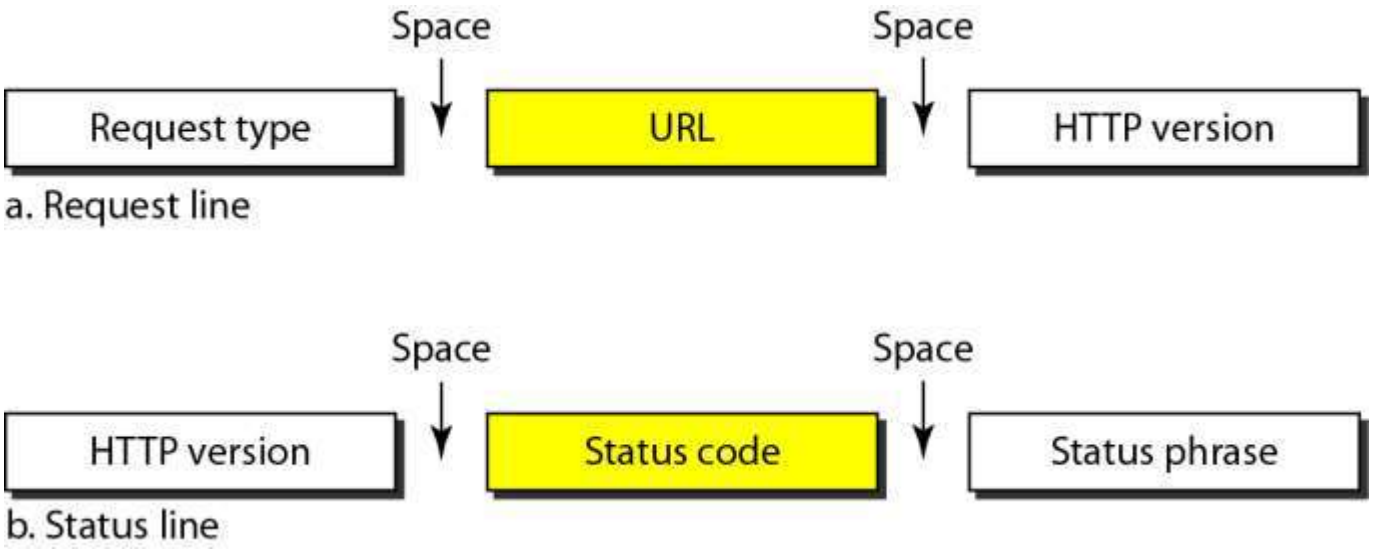


Table 27.1 *Methods(Request types)*

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

Table 27.2 *Status codes*

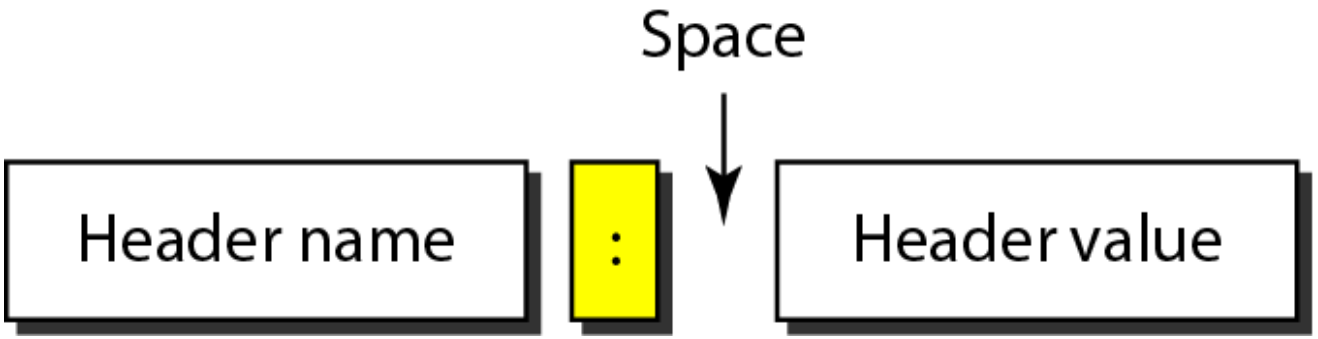
Code	*Phrase*	*Description*
	Informational	
100	Continue	The initial part of the request has been received, and the client may continue with its request.
101	Switching	The server is complying with a client request to switch protocols defined in the upgrade header.
	Success	
200	OK	The request is successful.
201	Created	A new URL is created.
202	Accepted	The request is accepted, but it is not immediately acted upon.
204	No content	There is no content in the body.

Table 27.2 *Status codes (continued)*

Code	*Phrase*	*Description*
	Redirection	
301	Moved permanently	The requested URL is no longer used by the server.
302	Moved temporarily	The requested URL has moved temporarily.
304	Not modified	The document has not been modified.
	Client Error	
400	Bad request	There is a syntax error in the request.
401	Unauthorized	The request lacks proper authorization.
403	Forbidden	Service is denied.
404	Not found	The document is not found.
405	Method not allowed	The method is not supported in this URL.
406	Not acceptable	The format requested is not acceptable.
	Server Error	
500	Internal server error	There is an error, such as a crash, at the server site.
501	Not implemented	The action requested cannot be performed.
503	Service unavailable	The service is temporarily unavailable, but may be requested in the future.

Status Phrase : It explains status code in text form

Figure 27.15 *Header format*



It gives general information about the message
Present in both request and response messages

Table 27.3 *General headers*

Header	*Description*
Cache-control	Specifies information about caching
Connection	Shows whether the connection should be closed or not
Date	Shows the current date
MIME-version	Shows the MIME version used
Upgrade	Specifies the preferred communication protocol

Table 27.4 *Request headers*

Header	*Description*
Accept	Shows the medium format the client can accept
Accept-charset	Shows the character set the client can handle
Accept-encoding	Shows the encoding scheme the client can handle
Accept-language	Shows the language the client can accept
Authorization	Shows what permissions the client has
From	Shows the e-mail address of the user
Host	Shows the host and port number of the server
If-modified-since	Sends the document if newer than specified date
If-match	Sends the document only if it matches given tag
If-non-match	Sends the document only if it does not match given tag
If-range	Sends only the portion of the document that is missing
If-unmodified-since	Sends the document if not changed since specified date
Referrer	Specifies the URL of the linked document
User-agent	Identifies the client program

It specifies clients configuration and client's preferred document format

Table 27.5 *Response headers*

Header	*Description*
Accept-range	Shows if server accepts the range requested by client
Age	Shows the age of the document
Public	Shows the supported list of methods
Retry-after	Specifies the date after which the server is available
Server	Shows the server name and version number

It specifies servers configuration and server's preferred document format

Table 27.6 *Entity headers*

Header	*Description*
Allow	Lists valid methods that can be used with a URL
Content-encoding	Specifies the encoding scheme
Content-language	Specifies the language
Content-length	Shows the length of the document
Content-range	Specifies the range of the document
Content-type	Specifies the medium type
Etag	Gives an entity tag
Expires	Gives the date and time when contents may change
Last-modified	Gives the date and time of the last change
Location	Specifies the location of the created or moved document

Gives the information about the body of the document
Body is the document to be sent or received

Persistent Versus Nonpersistent Connection

HTTP prior to version 1.1 specified a nonpersistent connection, while a persistent connection is the default in version 1.1.

Nonpersistent Connection

In a **nonpersistent connection,** one TCP connection is made for each request/response. The following lists the steps in this strategy:

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

HTTP version 1.1 specifies a persistent connection by default.

Domain Name System (DNS)

Domain Name System (DNS)

❖DNS, or the Domain Name System, translates human readable domain names (for example, www.amazon.com) to machine readable IP addresses (for example, 192.0.2.44).

❖The Internet's DNS system works much like a phone book by managing the mapping between names and numbers.

❖DNS servers translate requests for names into IP addresses, controlling which server an end user will reach when they type a domain name into their web browser.

❖These requests are called queries.

Types of DNS Service

An authoritative DNS

❖Its service provides an update mechanism that developers use to manage their public DNS names.

❖It then answers DNS queries, translating domain names into IP address so computers can communicate with each other.

❖Authoritative DNS has the final authority over a domain and is responsible for providing answers to recursive DNS servers with the IP address information.

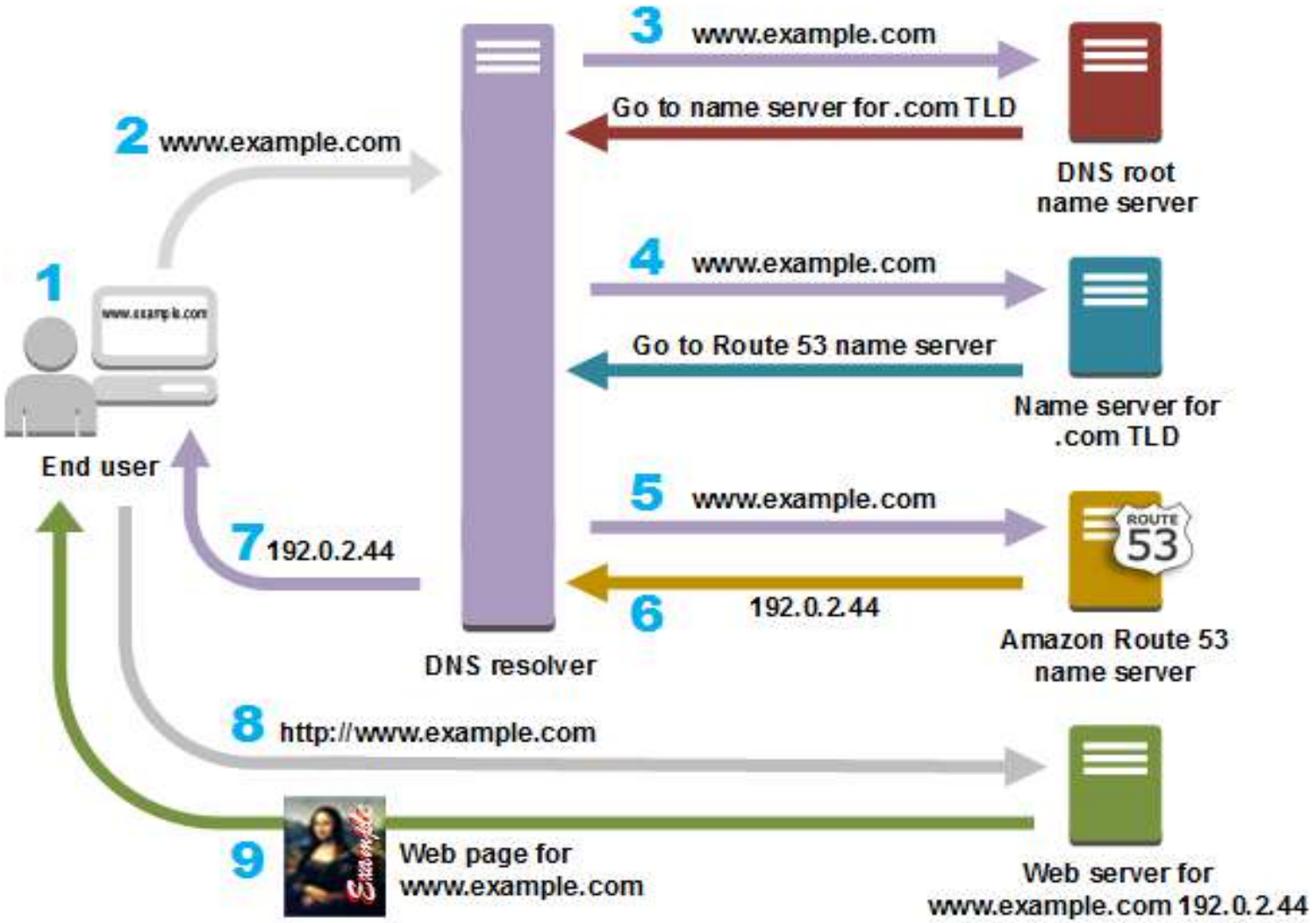
Types of DNS Service

❖Recursive DNS: Clients typically do not make queries directly to authoritative DNS services. Instead, they generally connect to another type of DNS service known a resolver, or a recursive DNS service.

If a recursive DNS has the DNS reference cached, or stored for a period of time, then it answers the DNS query by providing the source or IP information.

If not, it passes the query to one or more authoritative DNS servers to find the information.

How Does DNS Route Traffic To Your Web Application?



Domain Name System (DNS)

1. A user opens a web browser, enters www.example.com in the address bar, and presses Enter.

2. The request for www.example.com is routed to a DNS resolver, which is typically managed by the user's Internet service provider (ISP), such as a cable Internet provider, a DSL broadband provider, or a corporate network.

3. The DNS resolver for the ISP forwards the request for www.example.com to a DNS root name server.

4. The DNS resolver for the ISP forwards the request for www.example.com again, this time to one of the TLD name servers for .com domains. The name server for .com domains responds to the request with the names of the four Amazon Route 53 name servers that are associated with the example.com domain.

Domain Name System (DNS)

5. The DNS resolver for the ISP chooses an Amazon Route 53 name server and forwards the request for www.example.com to that name server.

6. The Amazon Route 53 name server looks in the example.com hosted zone for the www.example.com record, gets the associated value, such as the IP address for a web server, 192.0.2.44, and returns the IP address to the DNS resolver.

7. The DNS resolver for the ISP finally has the IP address that the user needs. The resolver returns that value to the web browser. The DNS resolver also caches (stores) the IP address for example.com for an amount of time that you specify so that it can respond more quickly the next time someone browses to example.com. For more information, see time to live (TTL).

8. The web browser sends a request for www.example.com to the IP address that it got from the DNS resolver. This is where your content is, for example, a web server running on an Amazon EC2 instance or an Amazon S3 bucket that's configured as a website endpoint.

9. The web server or other resource at 192.0.2.44 returns the web page for www.example.com to the web browser, and the web browser displays the page.