



Data Communication Networks

(EC 307)

Dr. Raghavendra Pal

Syllabus

Course Outcomes (COs):

At the end of the course the students will be able to:

CO1	Describe the basic knowledge of data communication methods, centralized/distributed networking architectures, OSI reference model, networking issues, protocols
CO2	Illustrate the suitable network protocols at various layers in computer networks along with the constraints
CO3	Apply the protocols and techniques in developing the standard networks using standard tools or software overcoming the constraints
CO4	Analyze the performance of various techniques and protocols in a given network topology, case study and problem solving as per given data.
CO5	Design the codes for the given protocols using appropriate tools

Syllabus:

- **DATA COMMUNICATION AND NETWORKING OVERVIEW** **(06 Hours)**
A Communication Model, Data Communication, Networking Concept, Topology And Transmission Media, Subnet, Concept of Client and Server, An Example Configuration, The Need For Protocol Architecture, Protocol Architecture and peer processes, OSI Reference Model, The TCP/IP Protocol Stack.

- **DATA LINK CONTROL** **(05 Hours)**
Medium Access Control (MAC) And Logical Link Control (LLC) Sublayer Issues, Flow Control, Error Control, Access Control, Sliding Window Protocol, Polling, High-Level Data Link Control (HDLC), PPP, Performance Issues.

Syllabus

- **LOCAL AREA NETWORKS — OVERVIEW** (05 Hours)
LAN Protocol Architecture, Bridges, Emergence of High Speed LANs, Ethernet, Wireless LAN Technology (Wi-Fi) Protocols.
- **ROUTING AND CONGESTION CONTROL** (06 Hours)
Logical Addresses, Circuit-Switching and Packet Switching Networks, Classful Addressing, Classless Addressing (CIDR), Subnetting, Supernetting, Network Address Translation, Routing In Packet-Switching Networks, Broadcasting, Multicasting, Flooding, Routing Algorithms, Effects Of Congestion, Congestion Control In Packet-Switching Networks. IP address classes, Ad-Hoc network Routing constraints. Mobile IP and its architecture
- **INTERNETWORK PROTOCOLS** (05 Hours)
Basic Protocol Functions, Principles Of Internetworking, Fragmentation Concept, Connectionless Internetworking, Gateway And Routers, The Internet with IPv4 and IPv6 packet formats, ARP, RARP, DHCP, ICMP, IGMP.
- **TRANSPORT PROTOCOLS** (04 Hours)
Protocol Overview Of Various Protocols - TCP, UDP And SCTP Protocols - Their Top
- **NETWORK SECURITY** (04 Hours)
Security Requirement And Attacks, Cryptography, Classical Ciphers, Modern Ciphers, Confidentiality With Encryption, Message Authentication And Hash Functions, Public-Key Encryption And Digital Signatures
- **DISTRIBUTED APPLICATIONS** (07 Hours)
Network Virtual Terminal (TELNET), File Transfer Protocol (FTP), Electronic Mail - SMTP And MIME, Hyper Transfer Protocol (HTTP), Network Management - SNMP, Domain Name Server (DNS), URL, WWW.



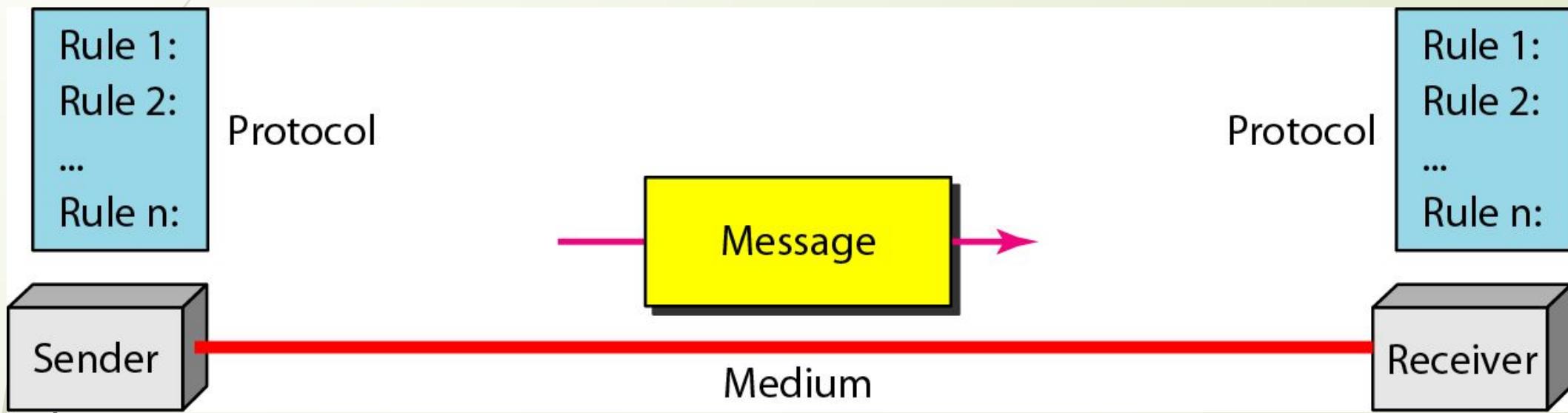
Chapter 1

Introduction

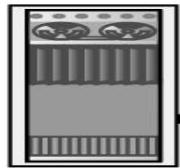
Data Communications

- The term telecommunication means communication at a distance. The word data refers to information presented in whatever form is agreed upon by the parties creating and using the data. Data communications are the exchange of data between two devices via some form of transmission medium such as a wire cable.

Five components of Data Communication



Data Flow types



Mainframe

Direction of data



Monitor

a. Simplex



Direction of data at time 1



Direction of data at time 2

b. Half-duplex



Direction of data all the time



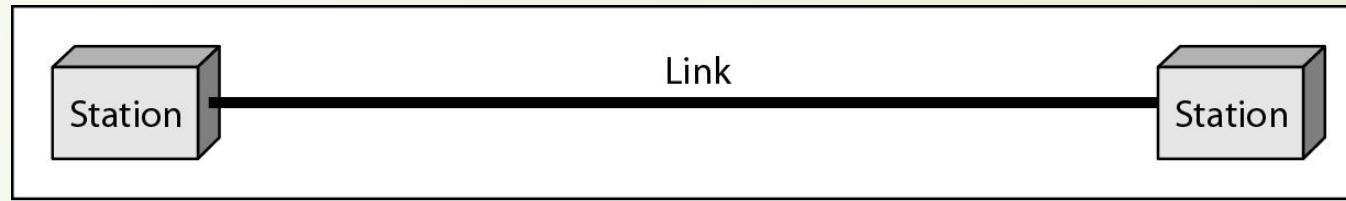
c. Full-duplex



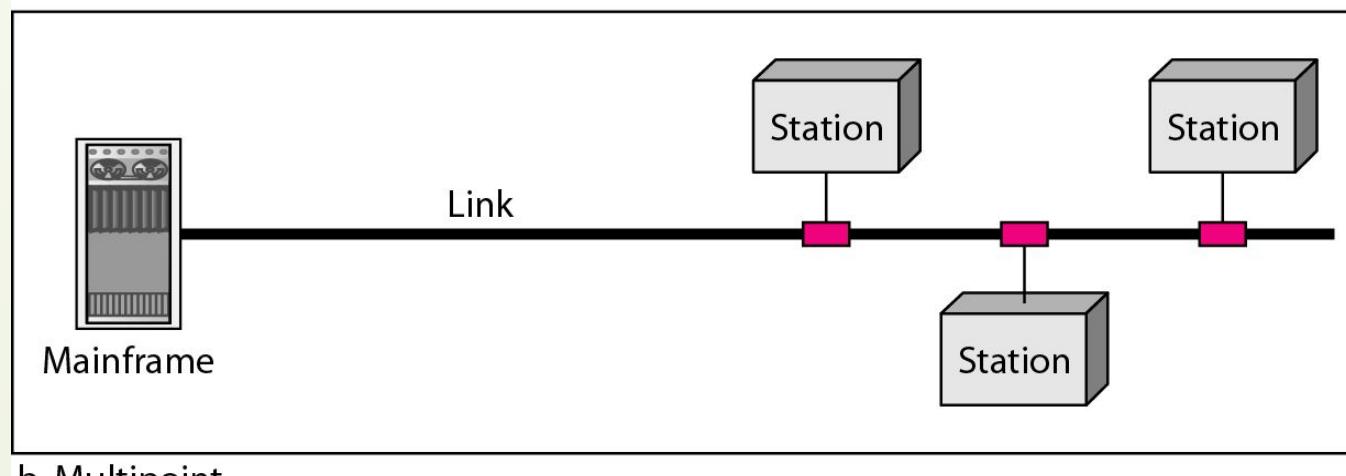
NETWORKS

- A network is a set of devices (often referred to as nodes) connected by communication links. A node can be a computer, printer, or any other device capable of sending and/or receiving data generated by other nodes on the network.

Types of connections: point-to-point and multipoint

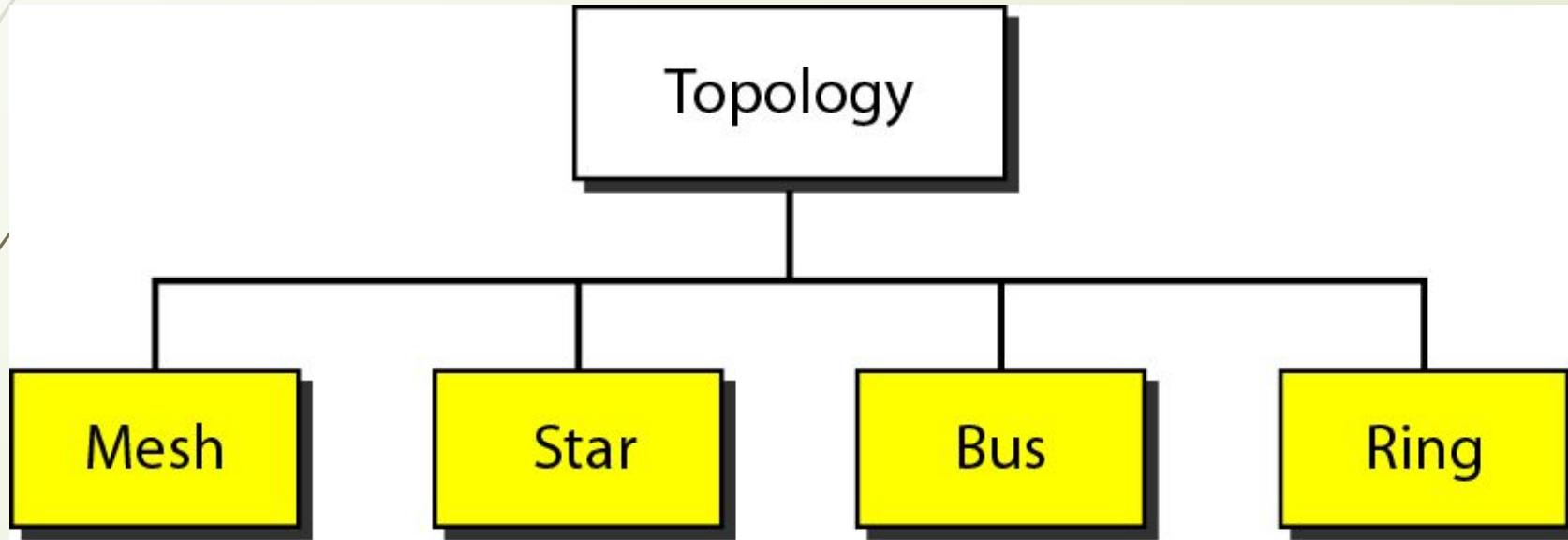


a. Point-to-point

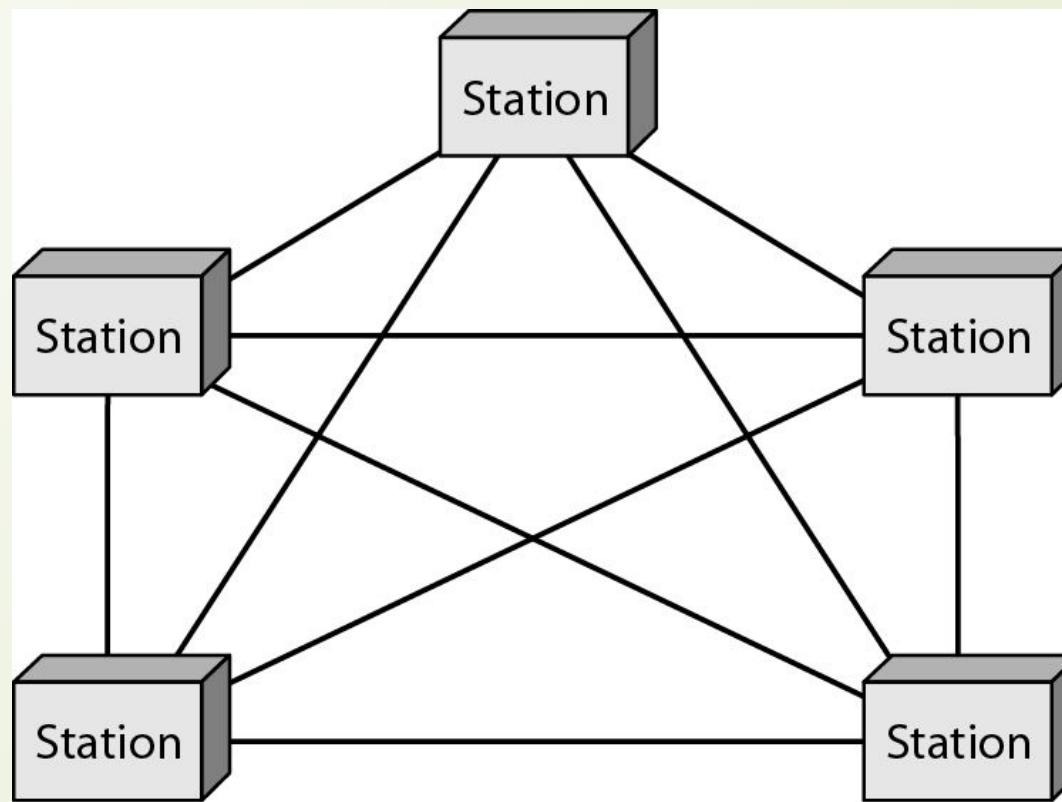


b. Multipoint

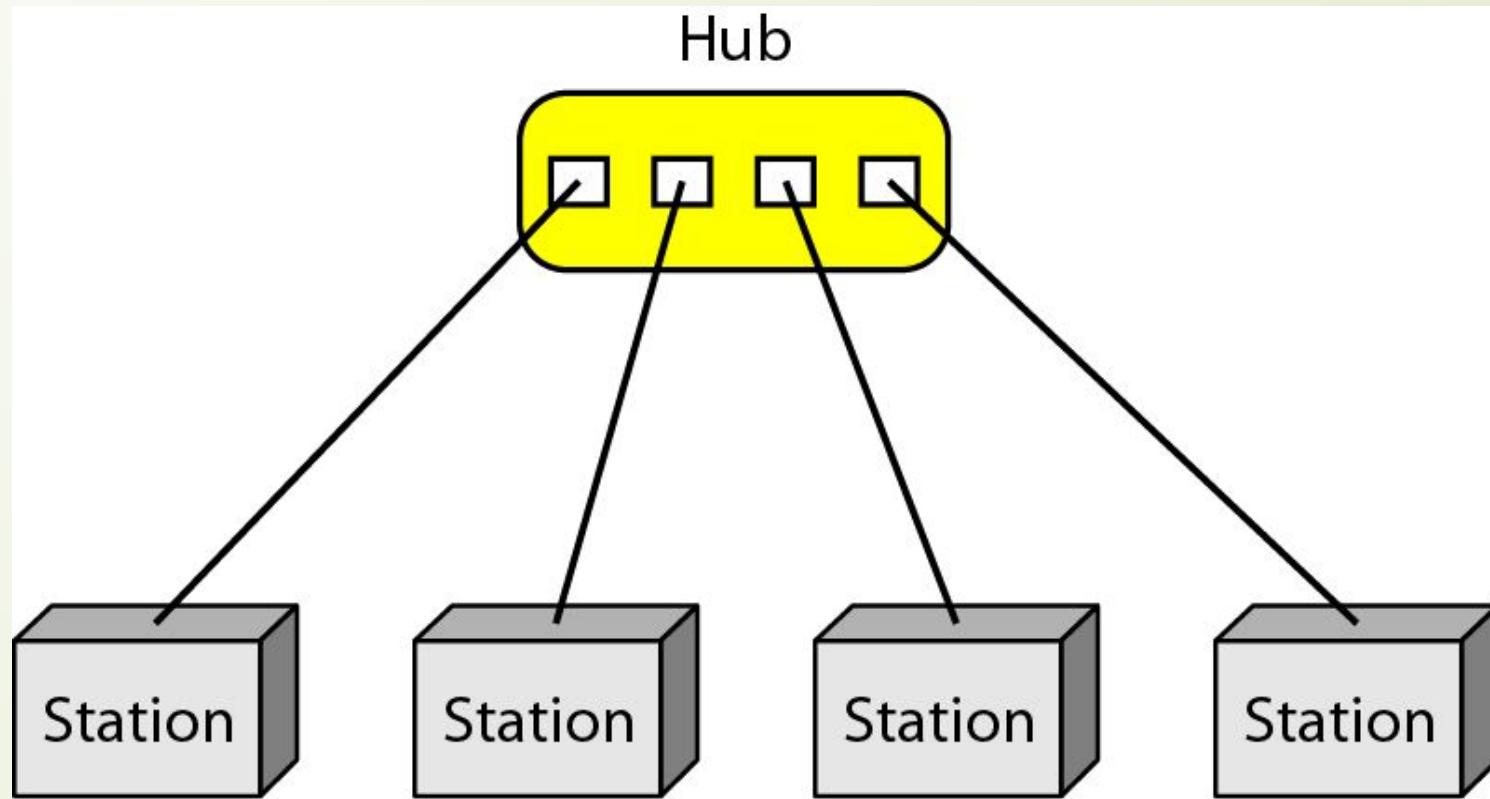
Categories of topology



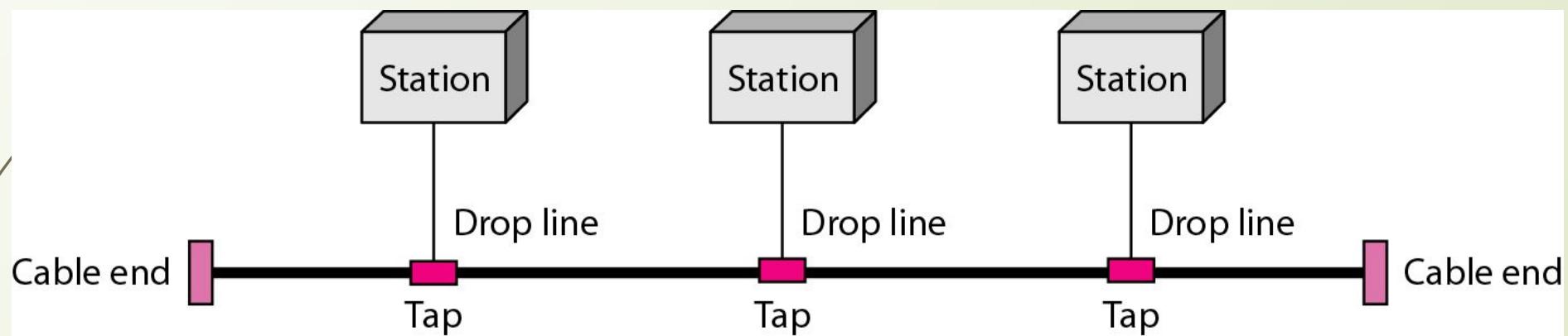
A fully connected mesh topology (five devices)



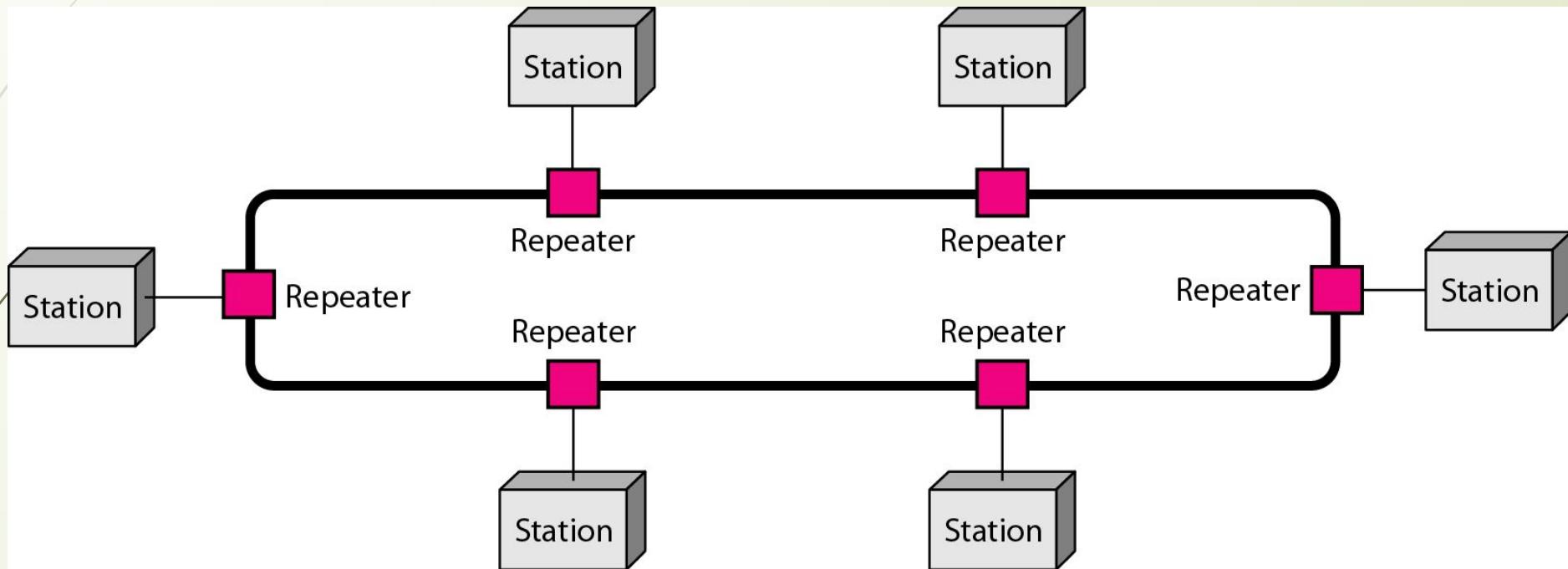
A star topology connecting four stations



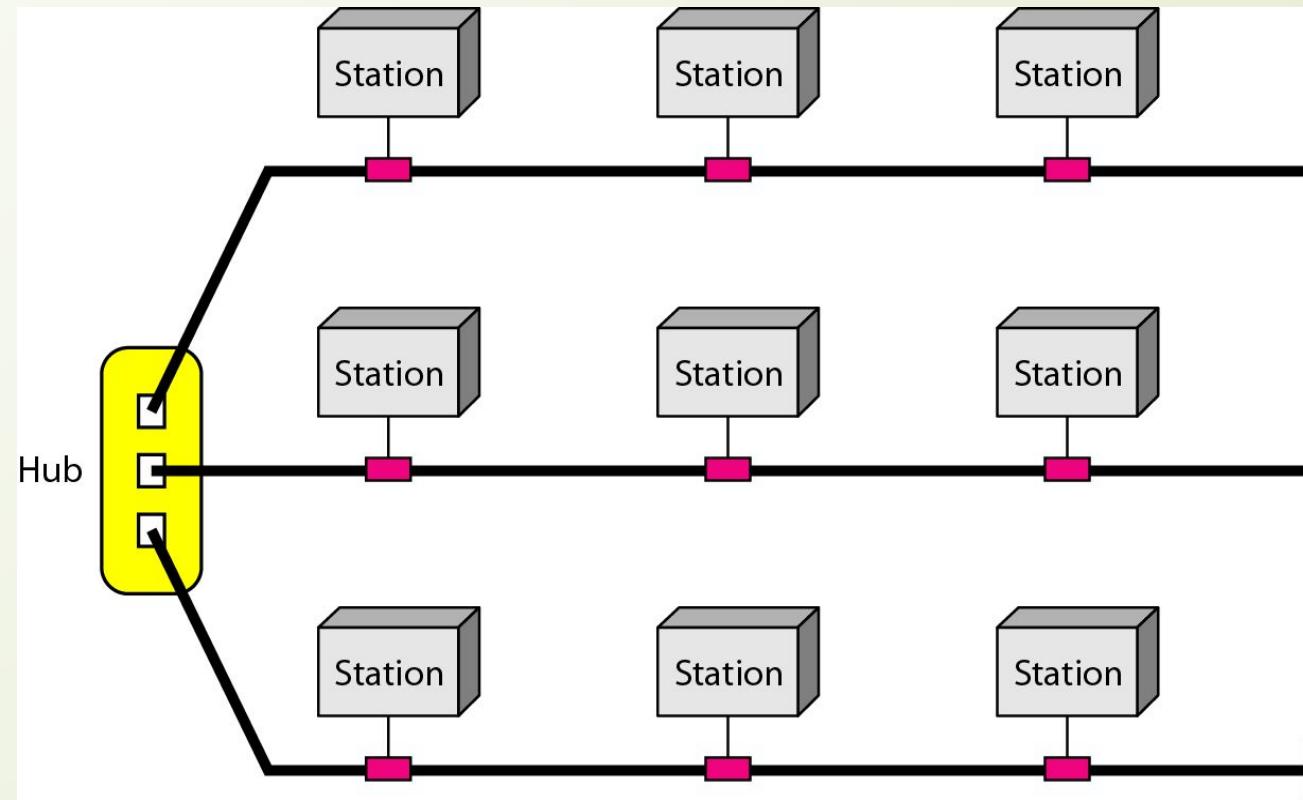
A bus topology connecting three stations



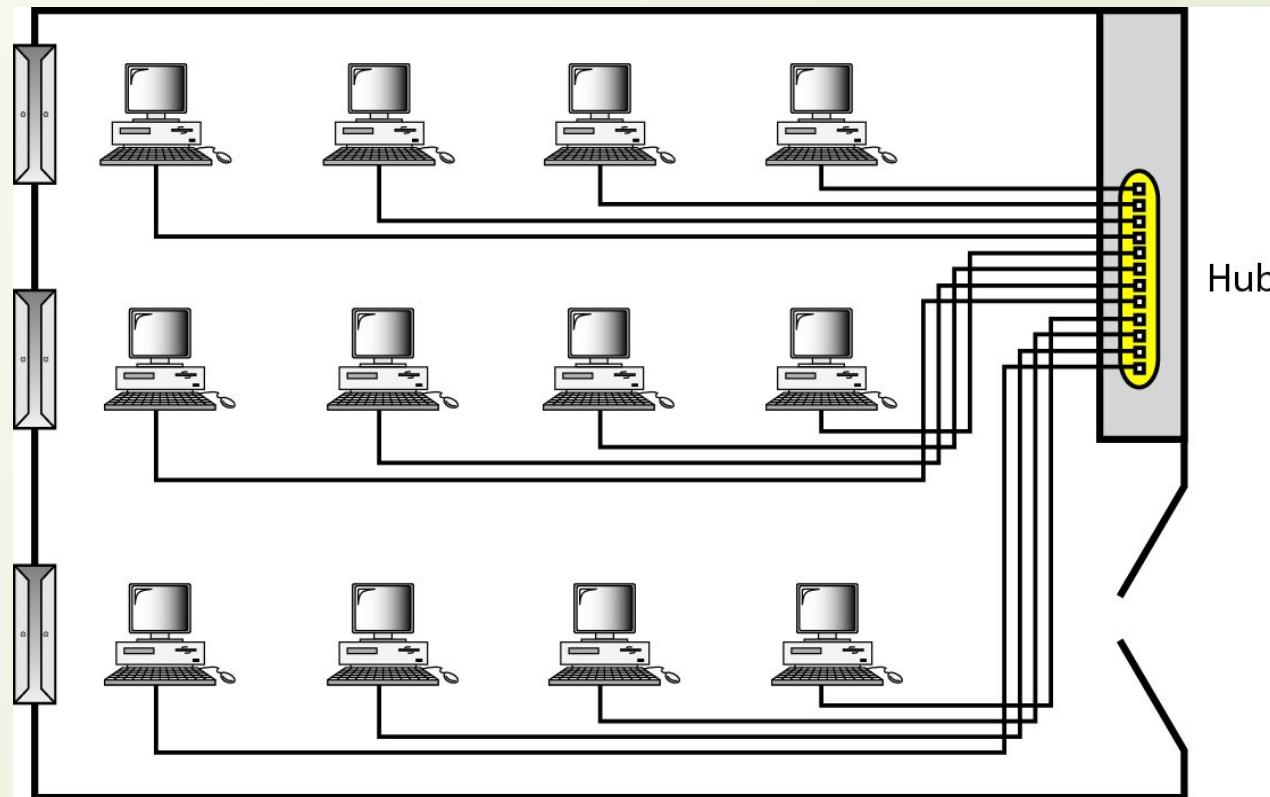
A ring topology connecting six stations



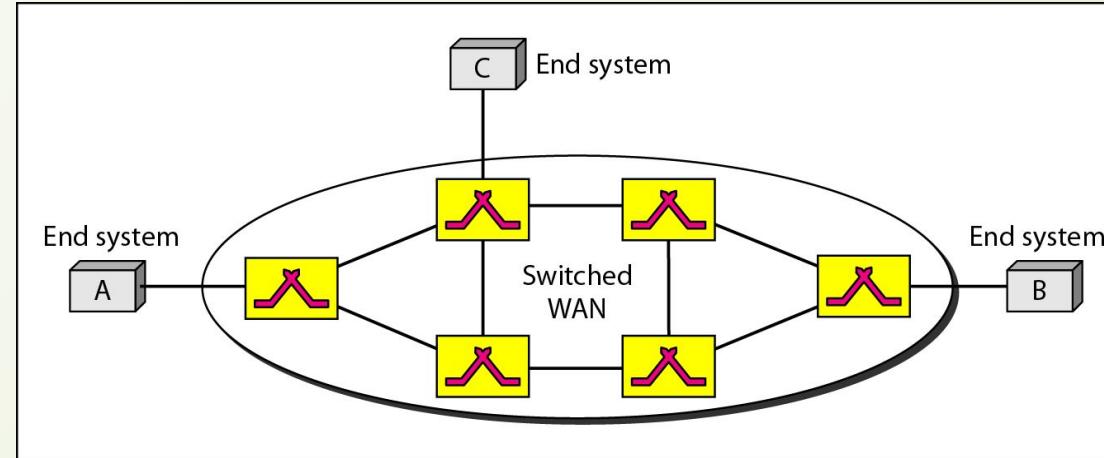
A hybrid topology: a star backbone with three bus networks



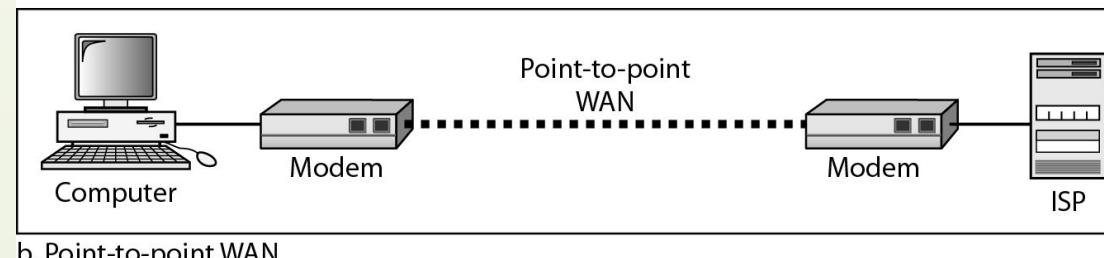
An isolated LAN connecting 12 computers to a hub in a closet



WANs: a switched WAN and a point-to-point WAN

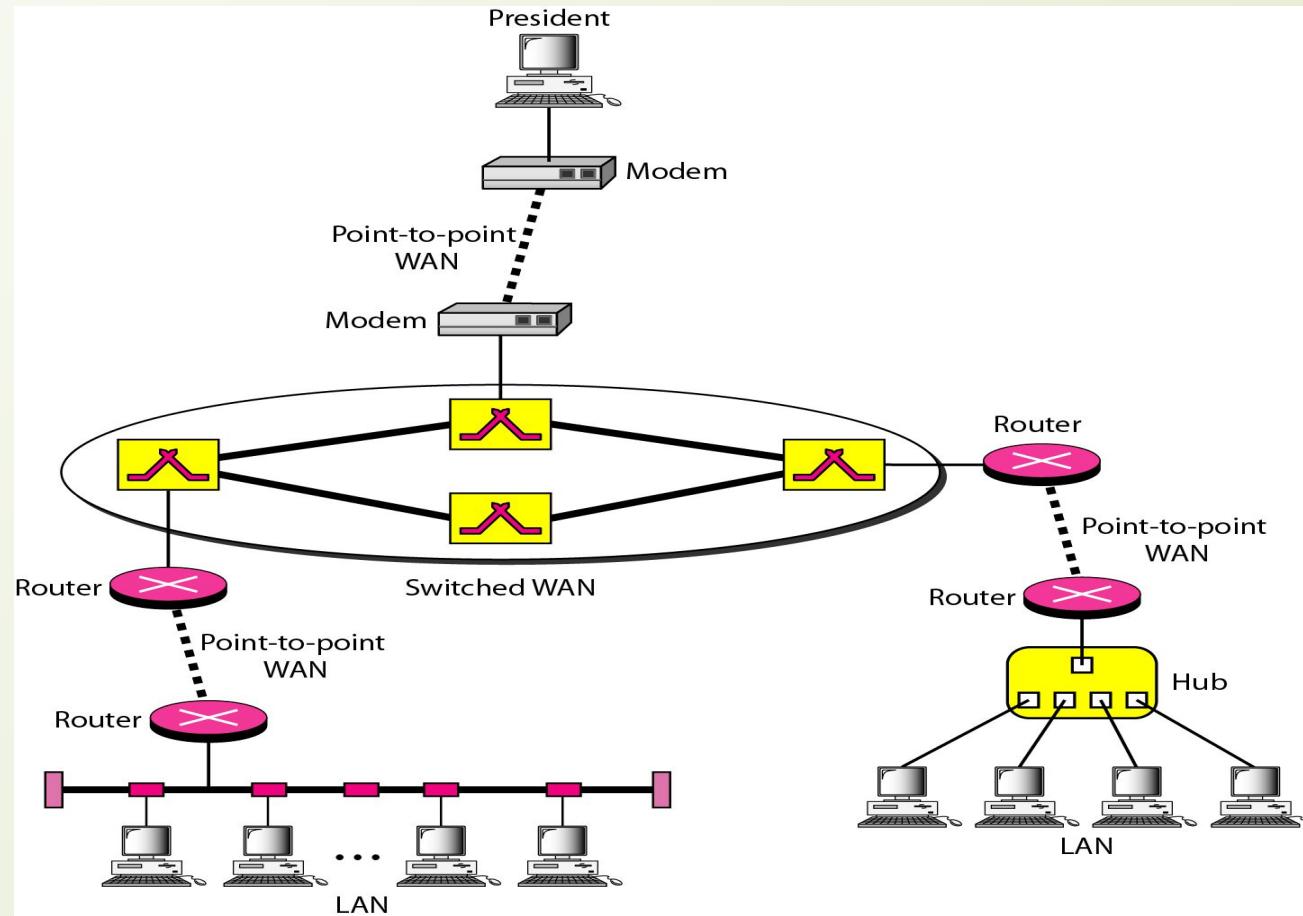


a. Switched WAN



b. Point-to-point WAN

A heterogeneous network made of four WANs and two LANs



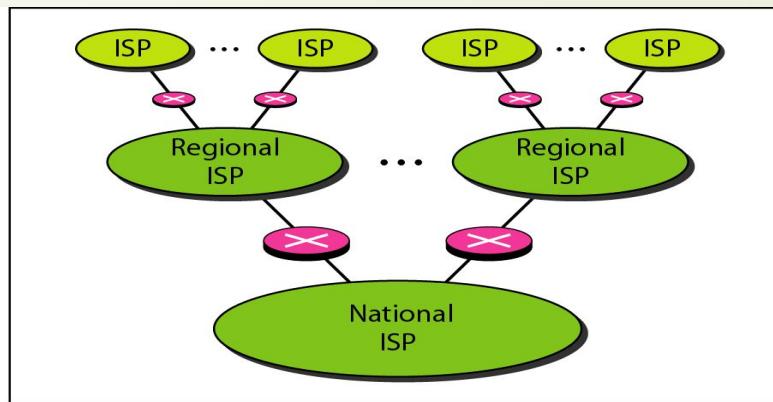


THE INTERNET

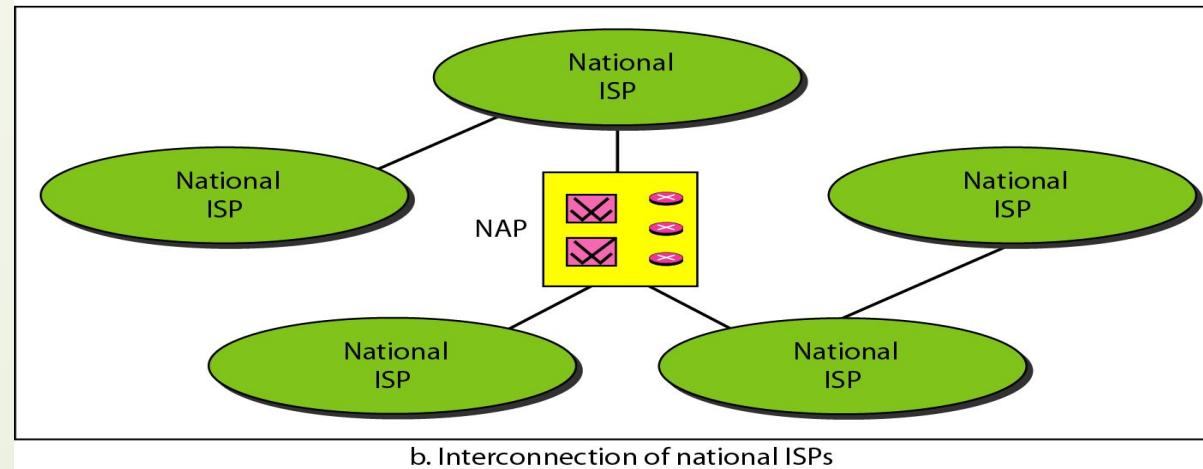
□ *The Internet has revolutionized many aspects of our daily lives. It has affected the way we do business as well as the way we spend our leisure time. The Internet is a communication system that has brought a wealth of information to our fingertips and organized it for our use.*



Hierarchical organization of the Internet



a. Structure of a national ISP



b. Interconnection of national ISPs



PROTOCOLS AND STANDARDS

□ In this section, we define two widely used terms: *protocols* and *standards*. First, we define protocol, which is synonymous with rule. Then we discuss standards, which are agreed-upon rules.

Topics discussed in this section:

Protocols

Standards

Standards Organizations

Internet Standards



Chapter 2

Network Models



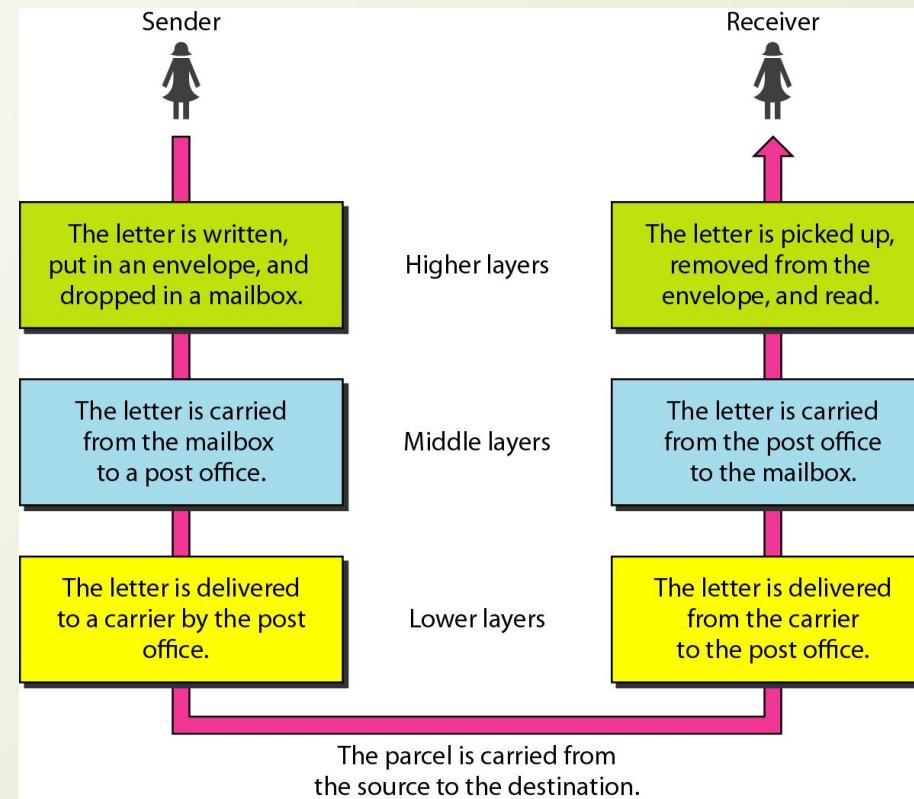
LAYERED TASKS

- We use the concept of *layers* in our daily life. As an example, let us consider two friends who communicate through postal mail. The process of sending a letter to a friend would be complex if there were no services available from the post office.

Topics discussed in this section:

Sender, Receiver, and Carrier
Hierarchy

Tasks involved in sending a letter



THE OSI MODEL

- Established in 1947, the International Standards Organization (ISO) is a multinational body dedicated to worldwide agreement on international standards. An ISO standard that covers all aspects of network communications is the Open Systems Interconnection (OSI) model. It was first introduced in the late 1970s.

Topics discussed in this section:

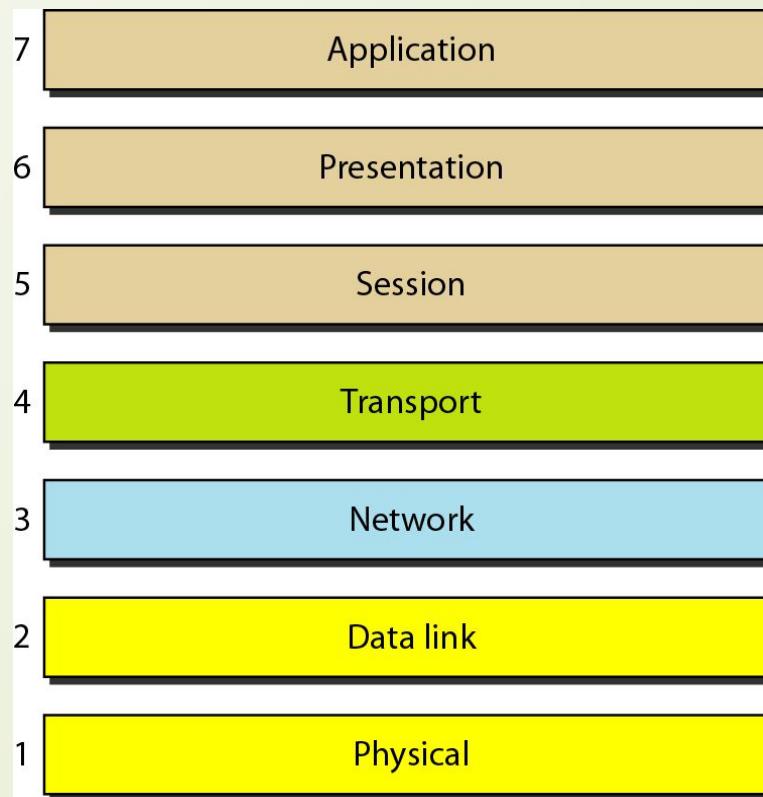
Layered Architecture
Peer-to-Peer Processes
Encapsulation



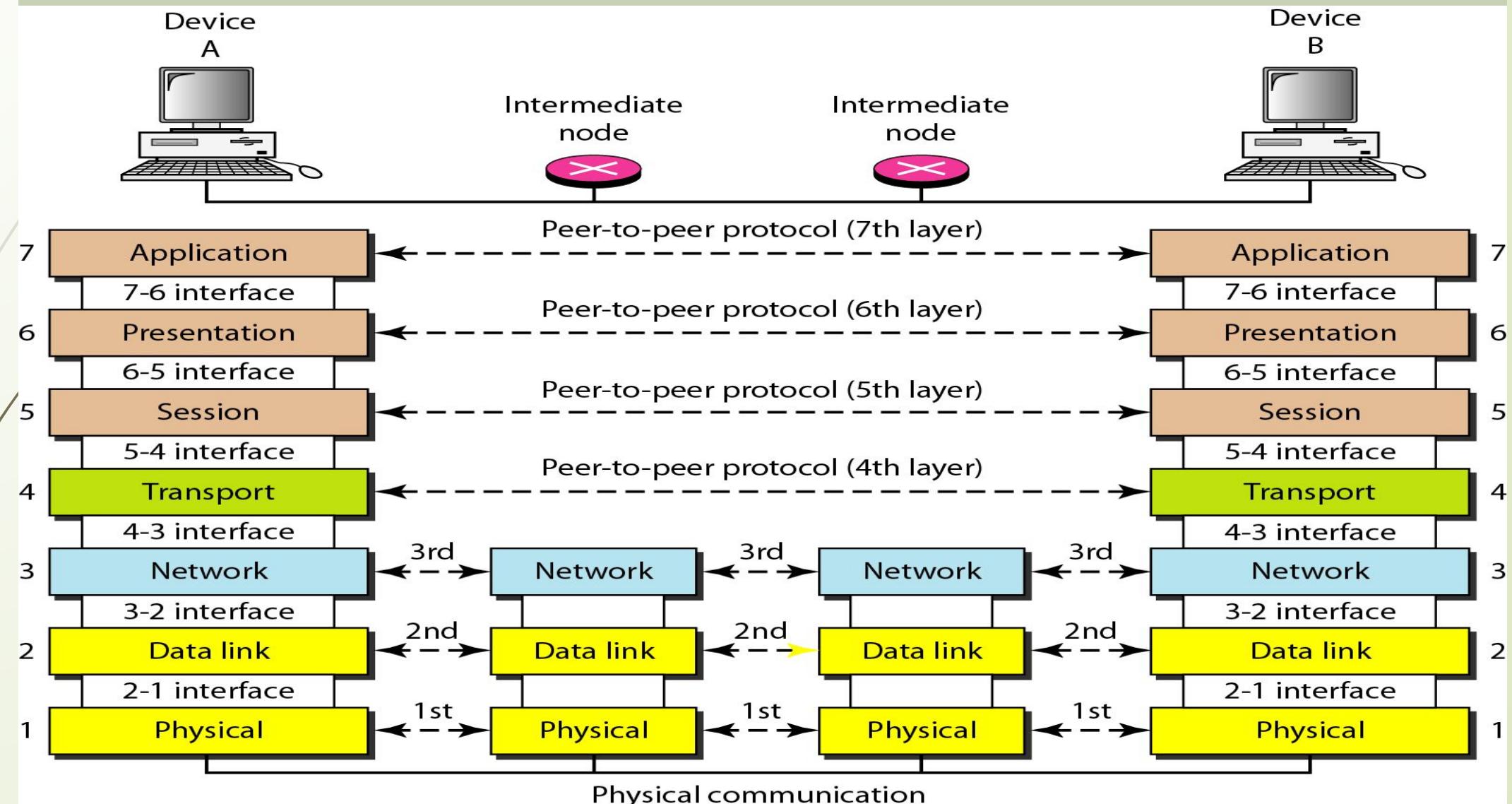
Note

ISO is the organization.
OSI is the model.

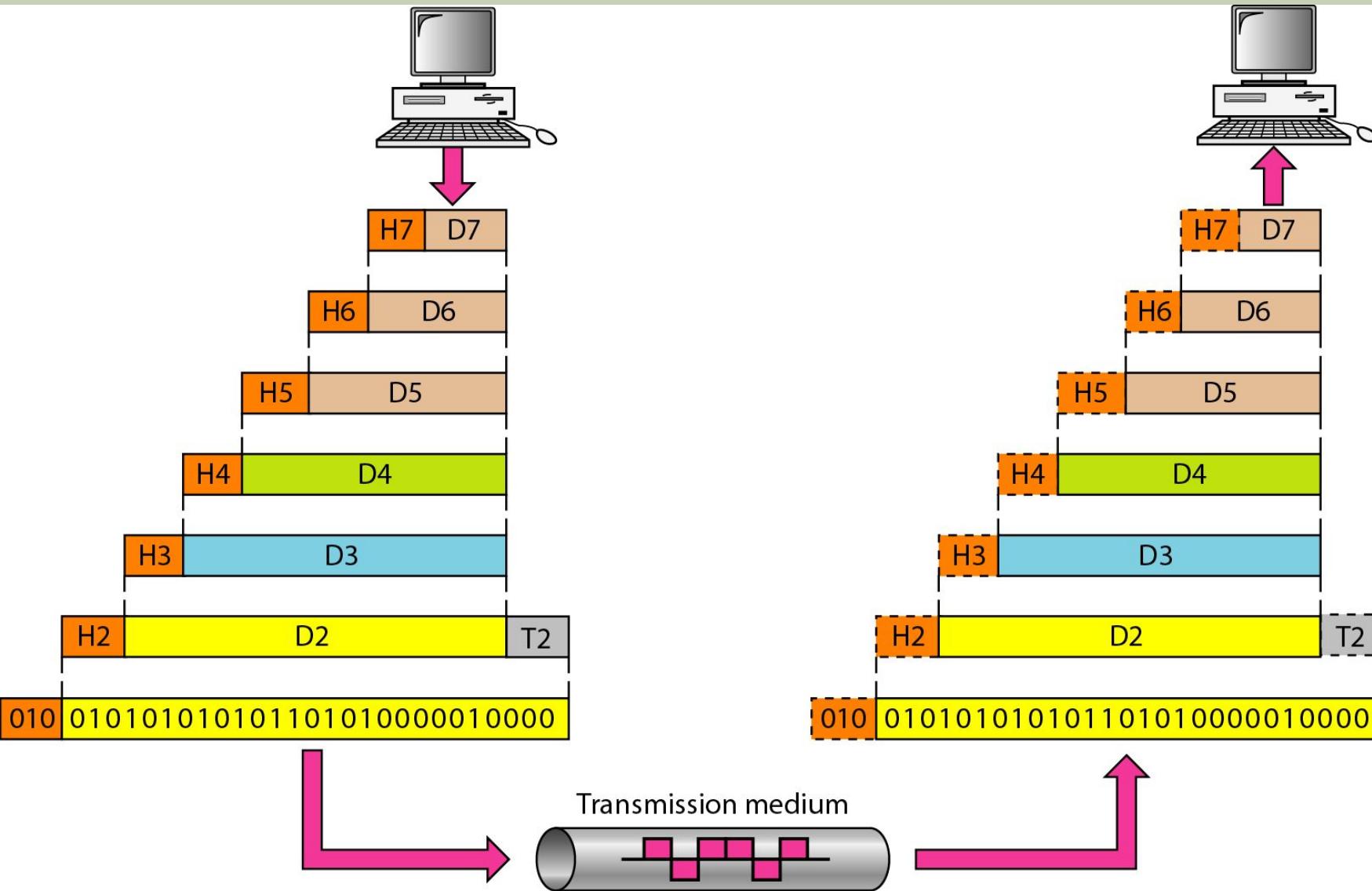
Seven layers of the OSI model



The interaction between layers in the OSI model



An exchange using the OSI model





LAYERS IN THE OSI MODEL

□ *In this section we briefly describe the functions of each layer in the OSI model.*

Topics discussed in this section:

Physical Layer

Data Link Layer

Network Layer

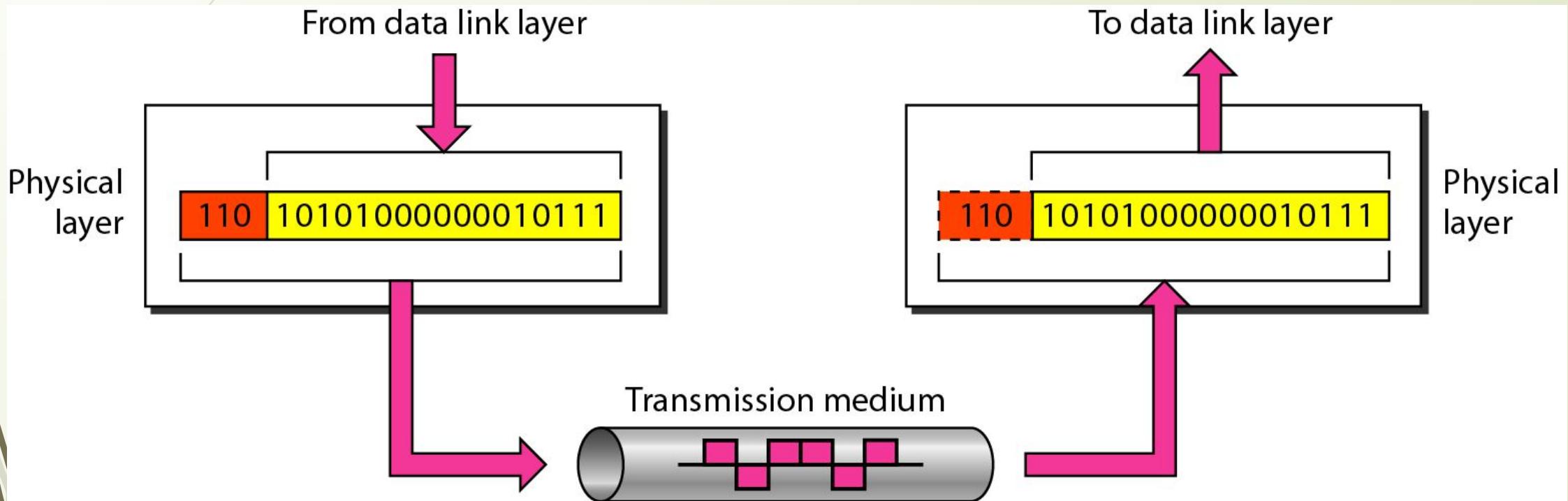
Transport Layer

Session Layer

Presentation Layer

Application Layer

Physical layer



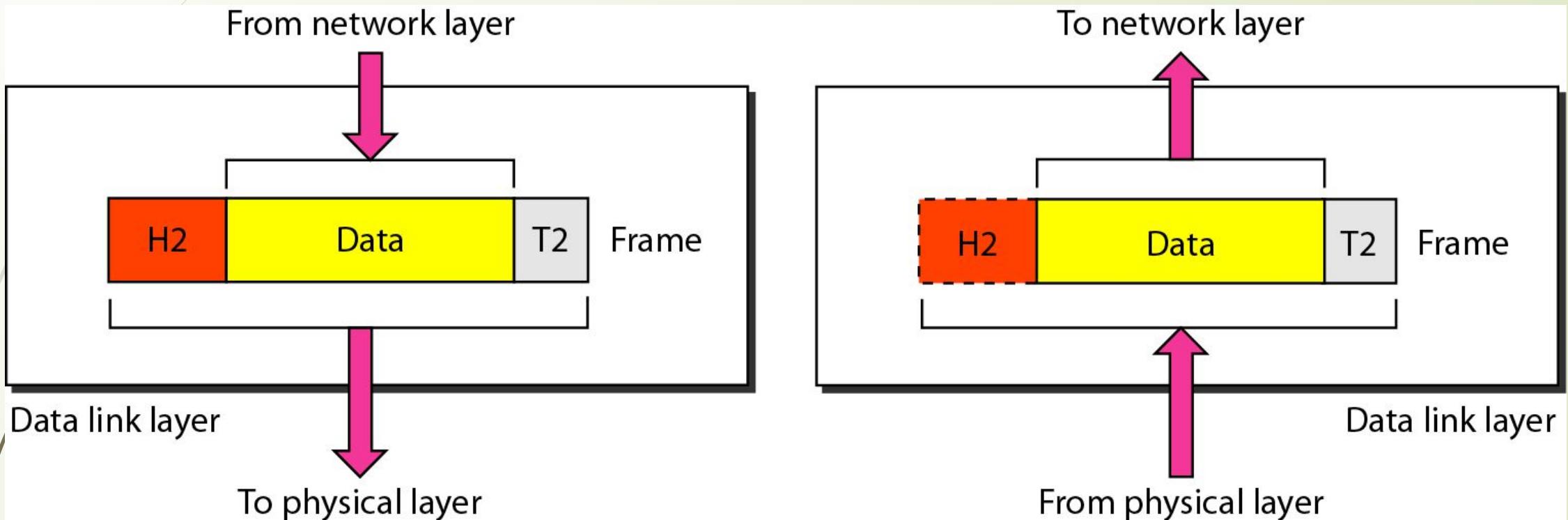


Note



The physical layer is responsible for movements of individual bits from one hop (node) to the next.

Data link layer

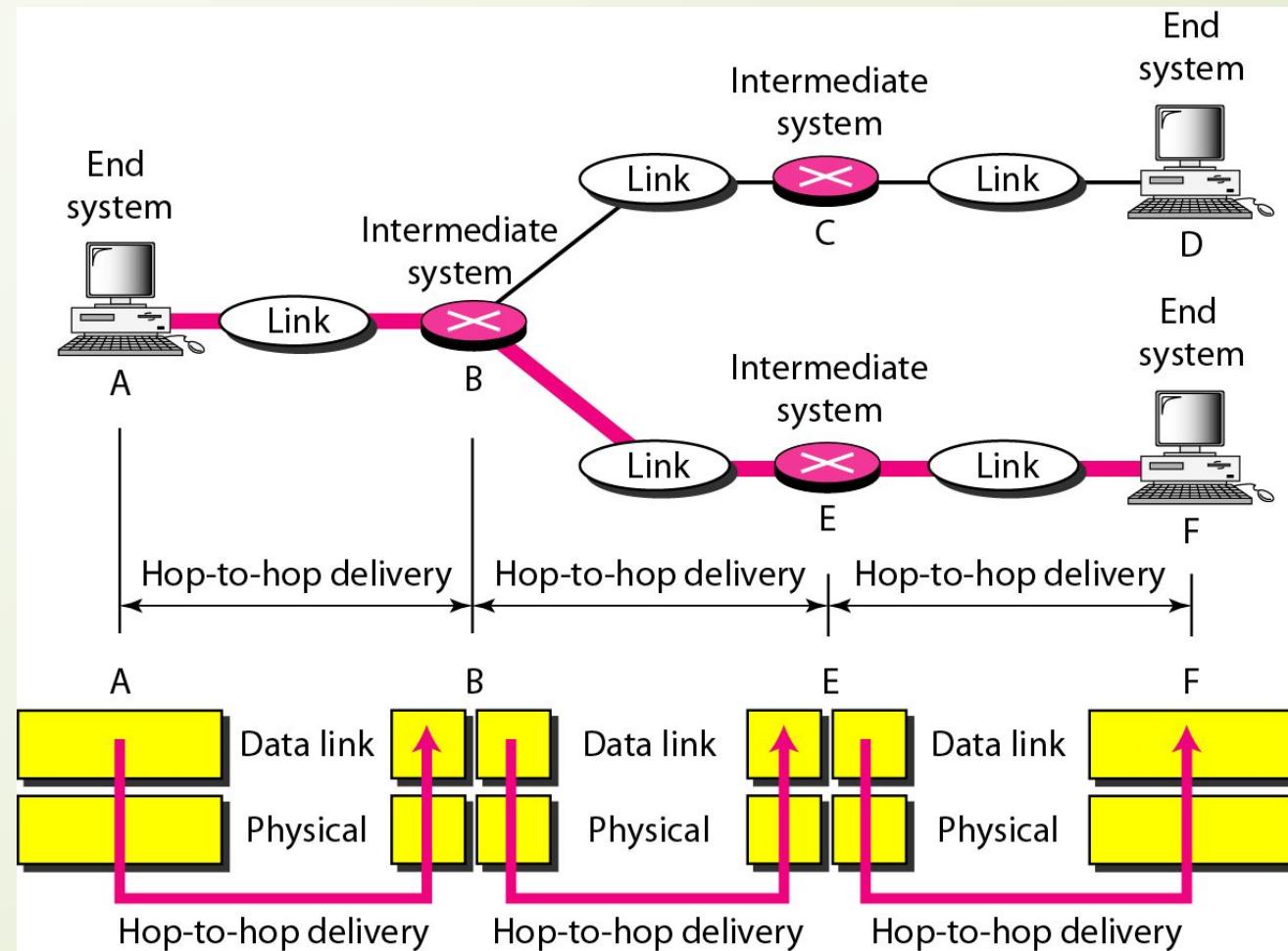




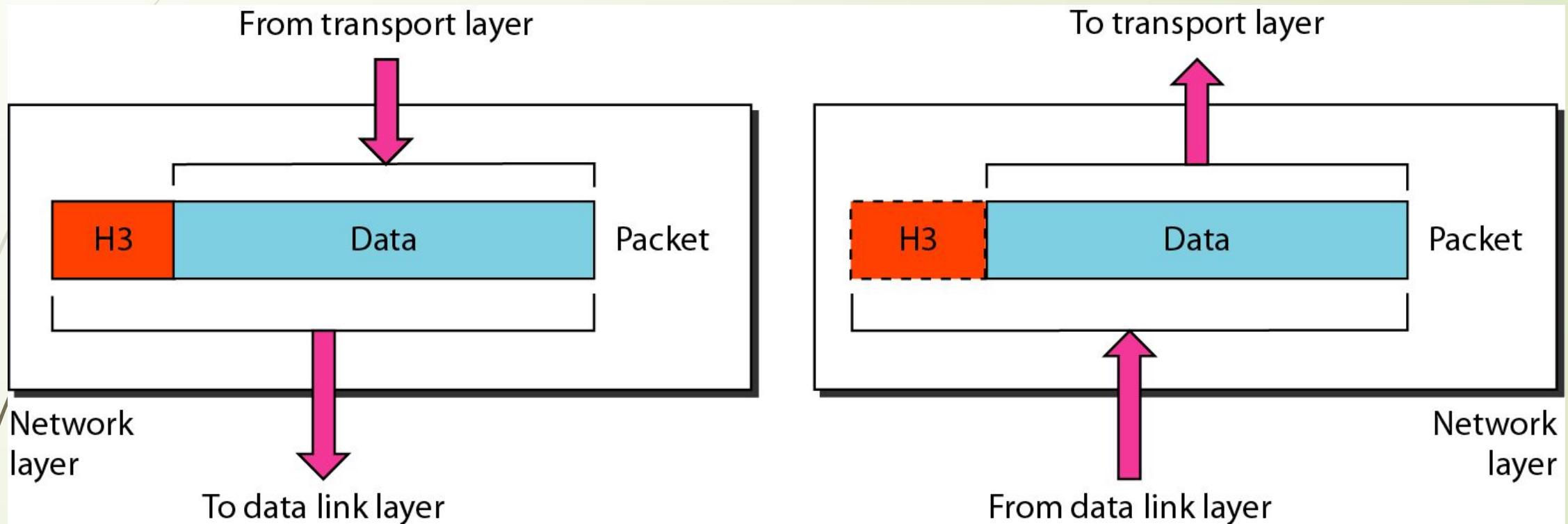
Note

- The data link layer is responsible for moving frames from one hop (node) to the next.

Hop-to-hop delivery



Network layer

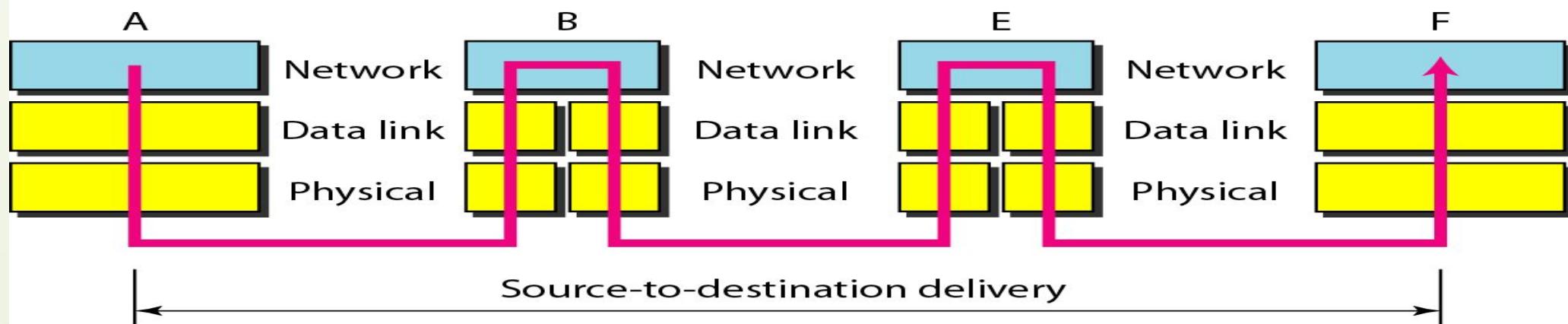
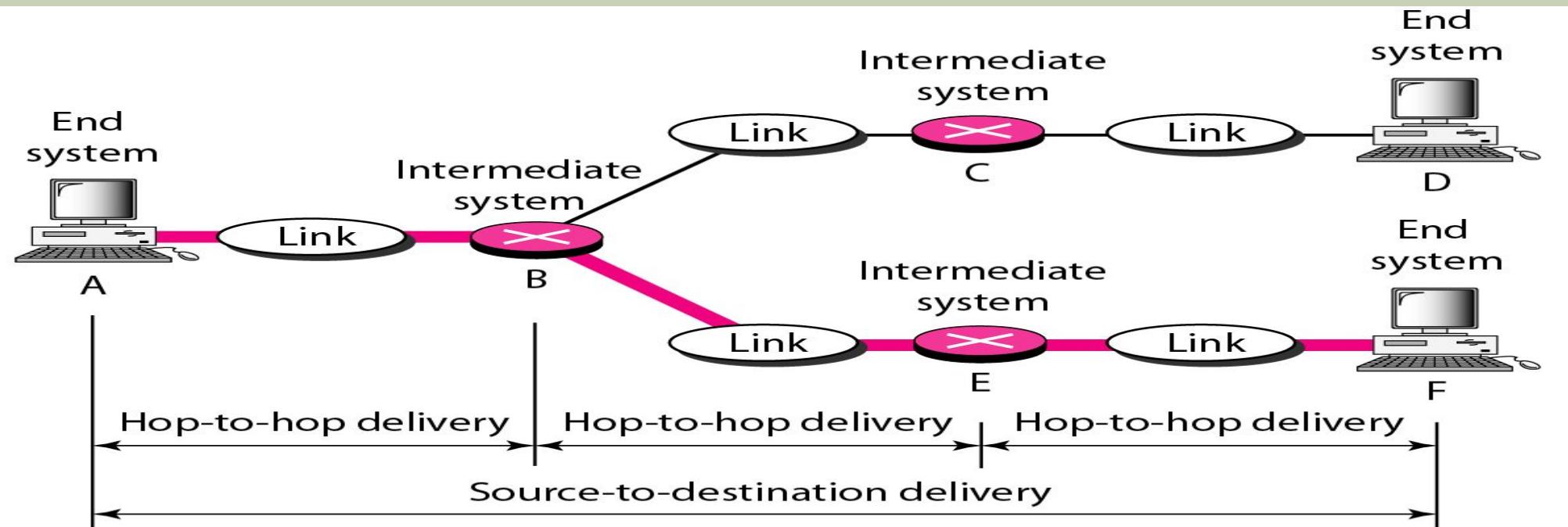




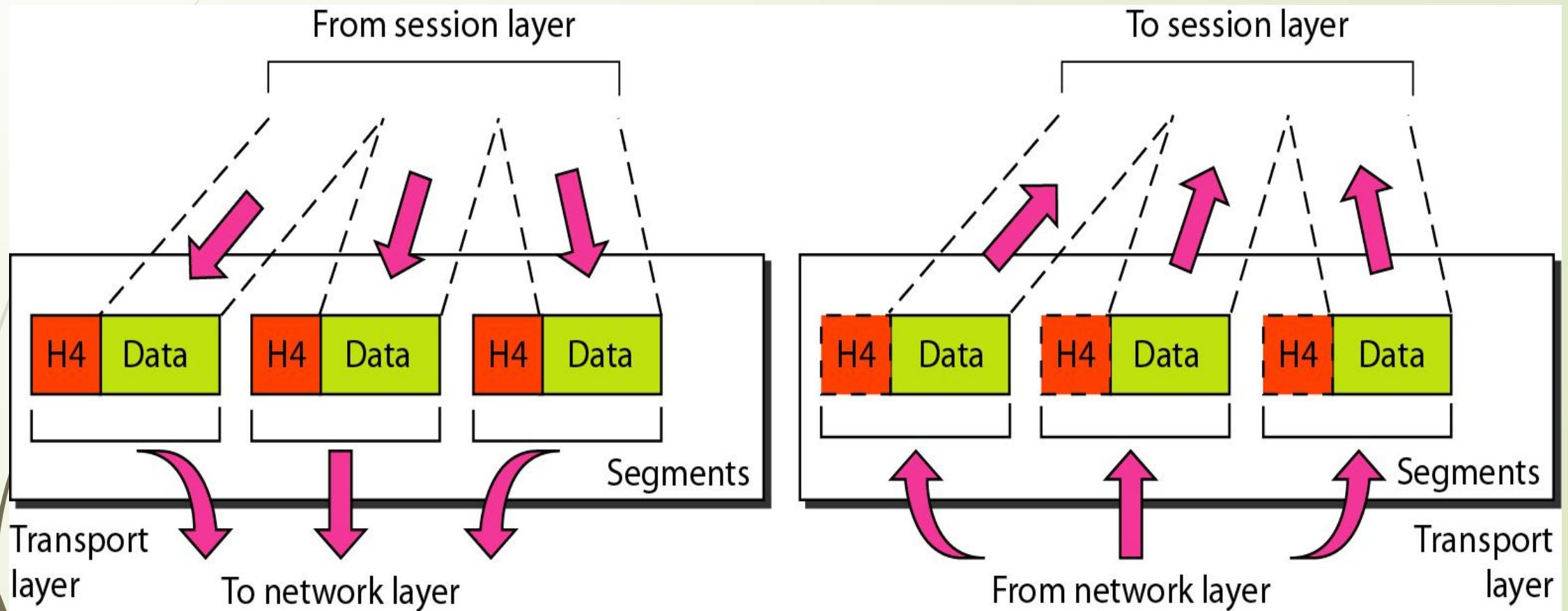
Note

The network layer is responsible for the delivery of a message from one device to another.

Source-to-destination delivery



Transport layer

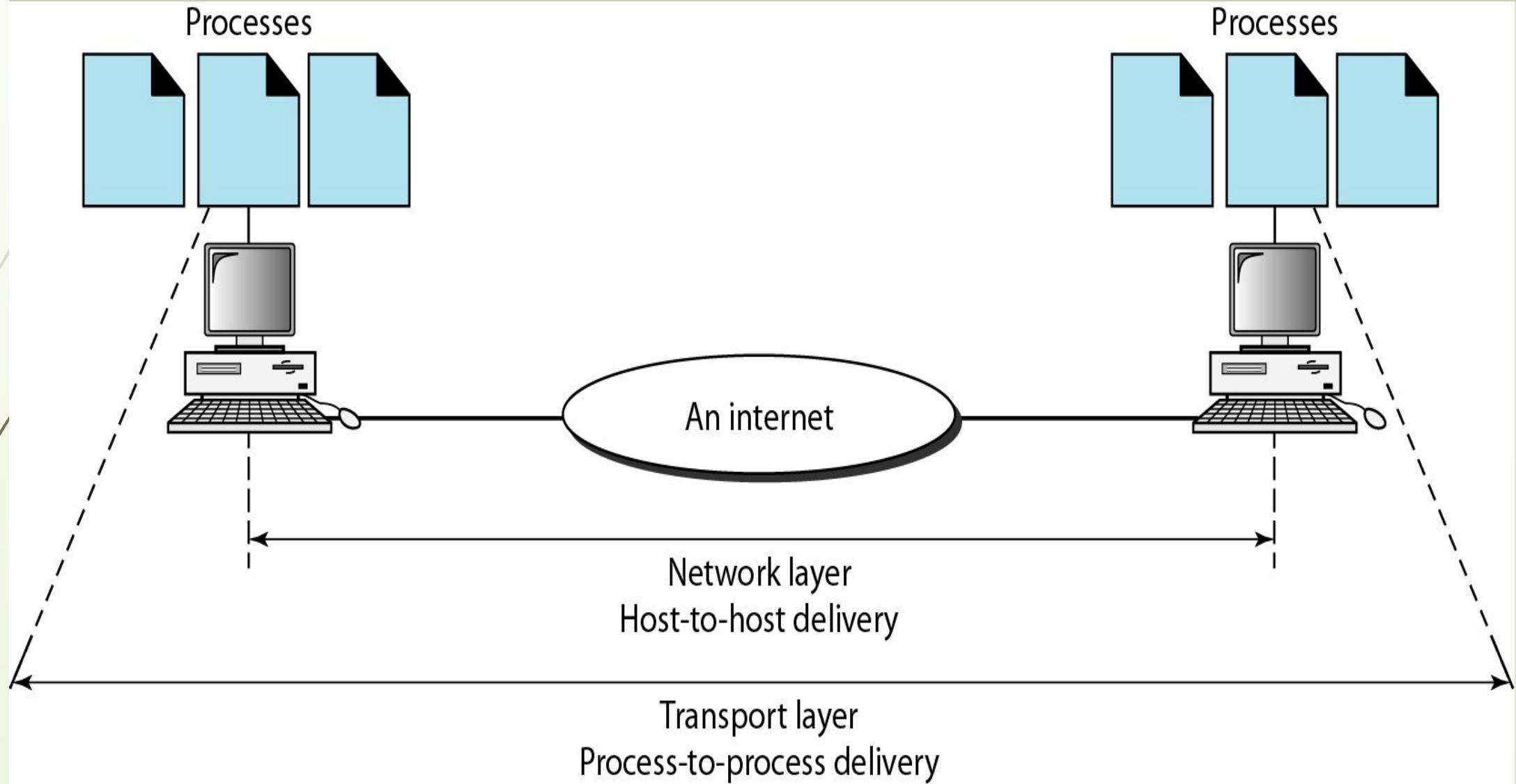




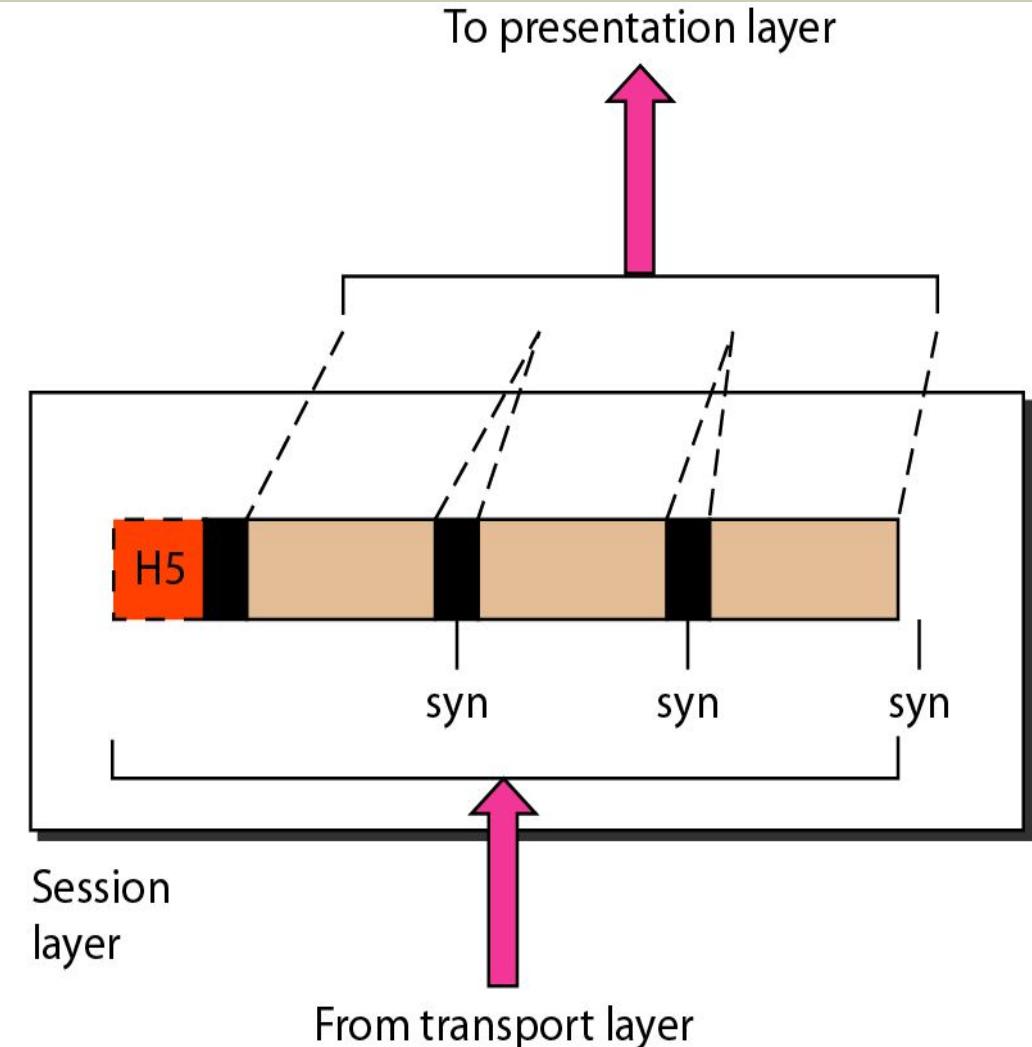
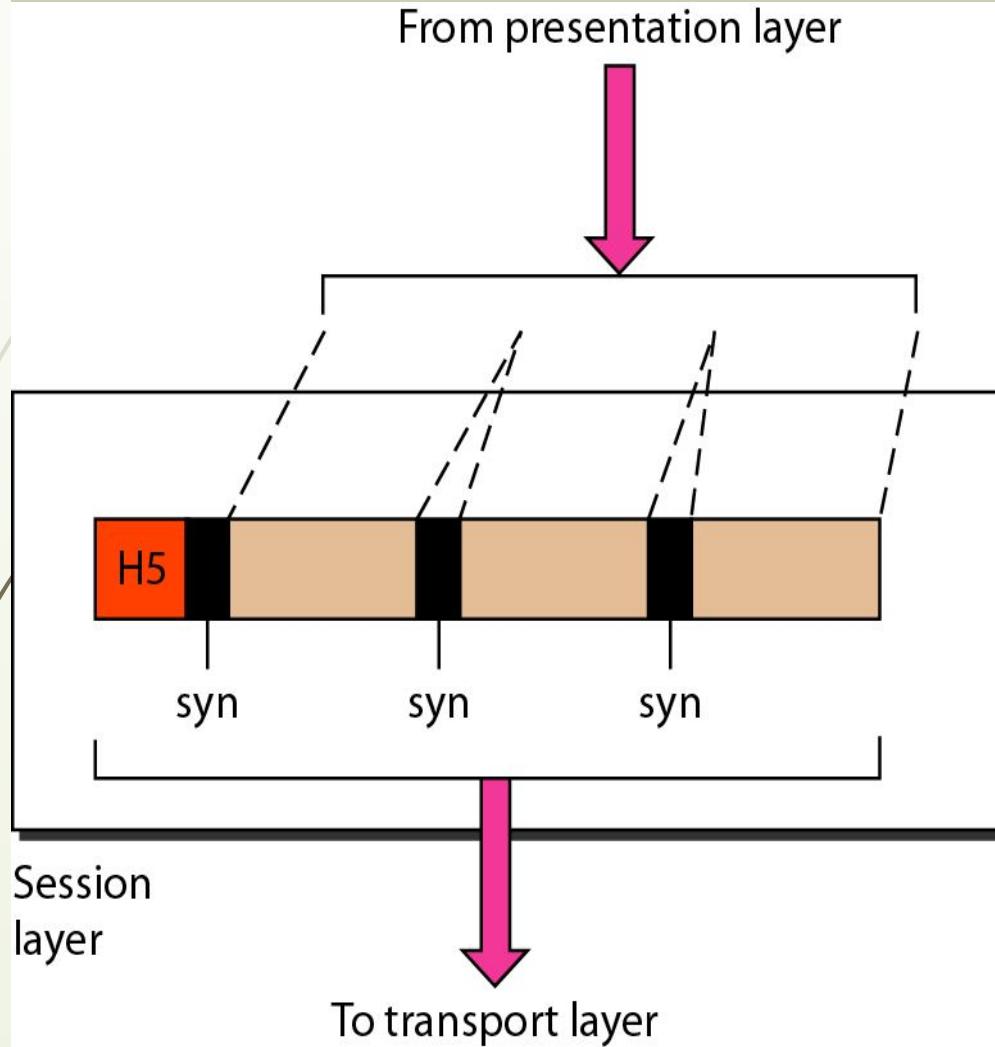
Note

The transport layer is responsible for the delivery of a message from one process to another.

Reliable process-to-process delivery of a message



Session layer

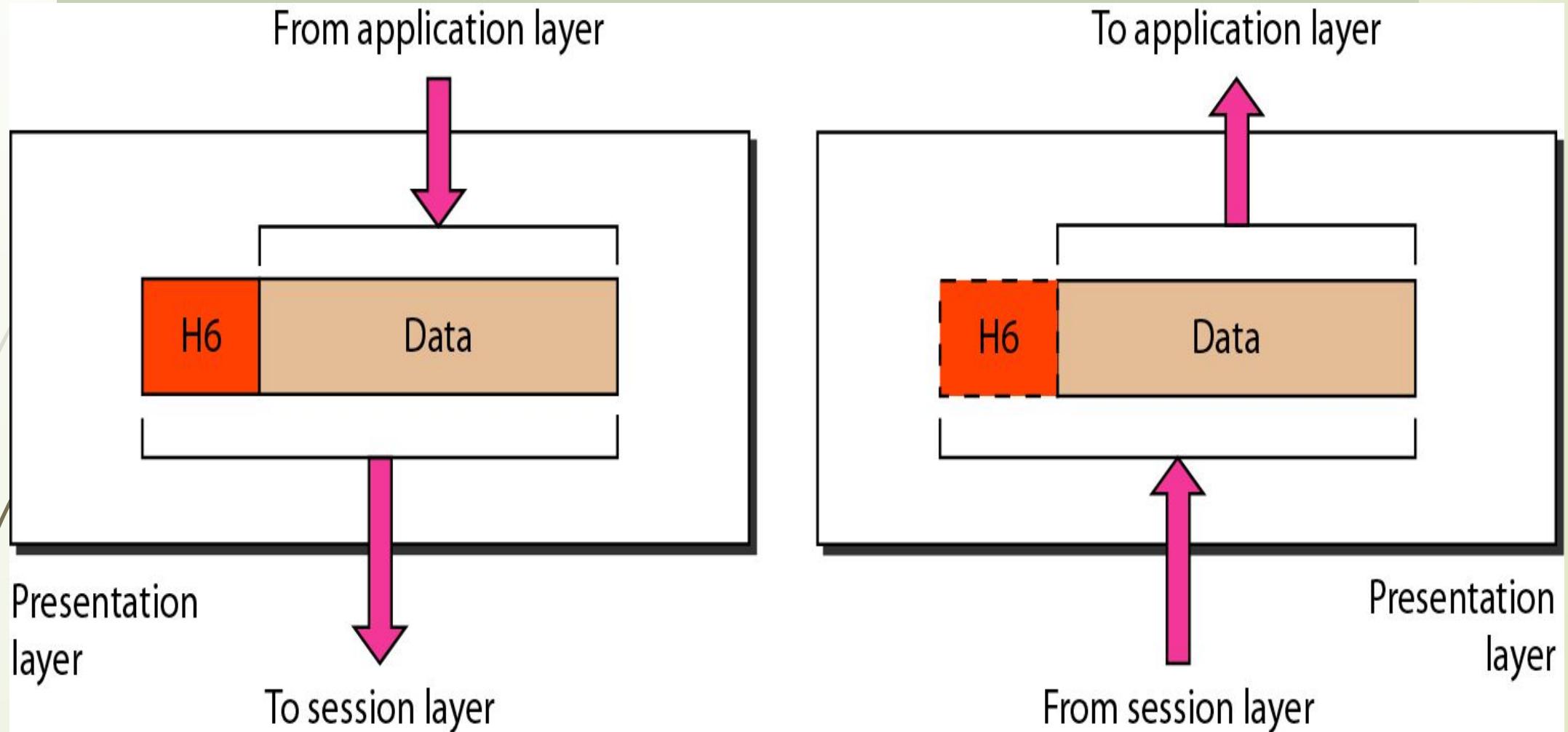




Note

- The session layer is responsible for dialog control and synchronization.

Presentation layer

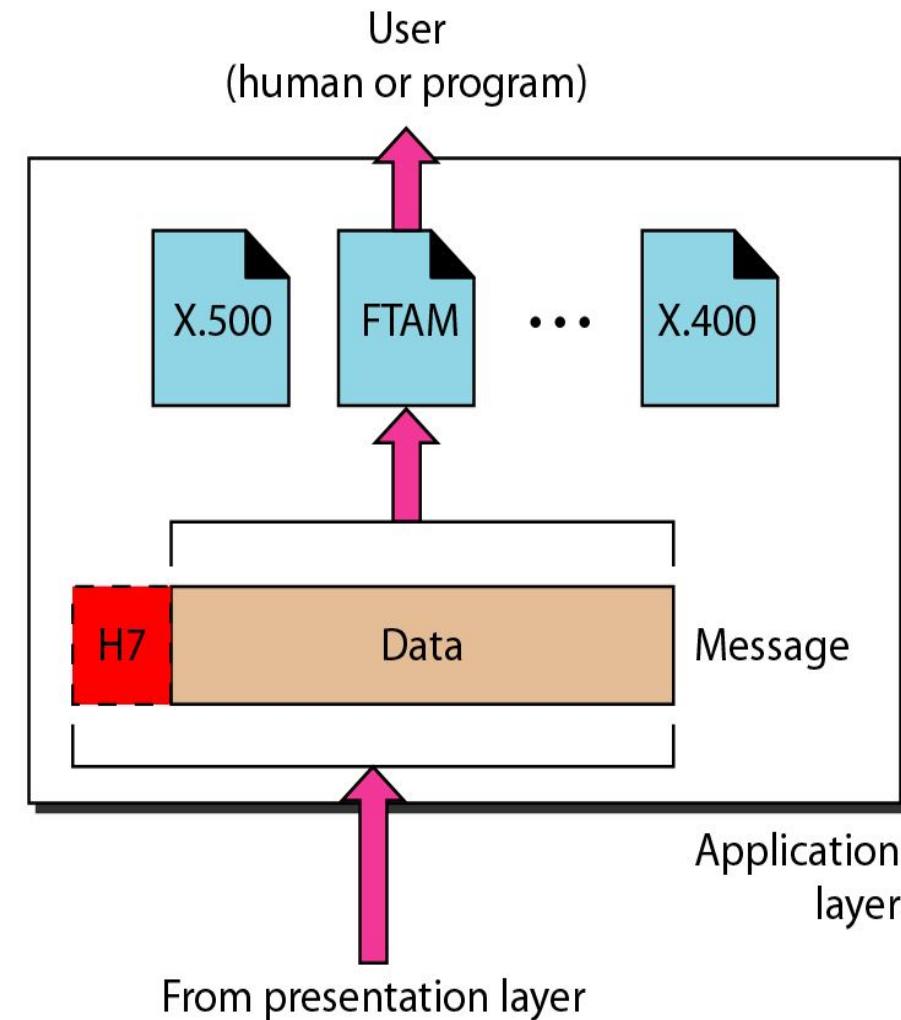
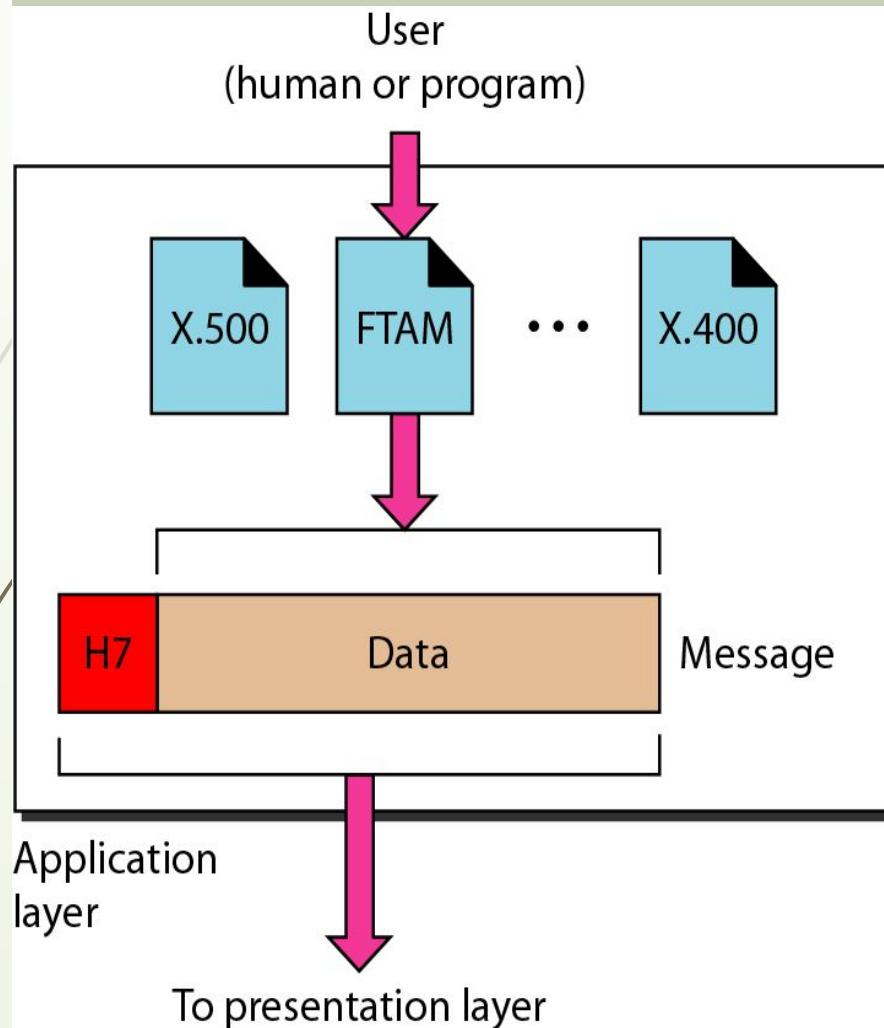




Note

- The presentation layer is responsible for translation, compression, and encryption.

Application layer

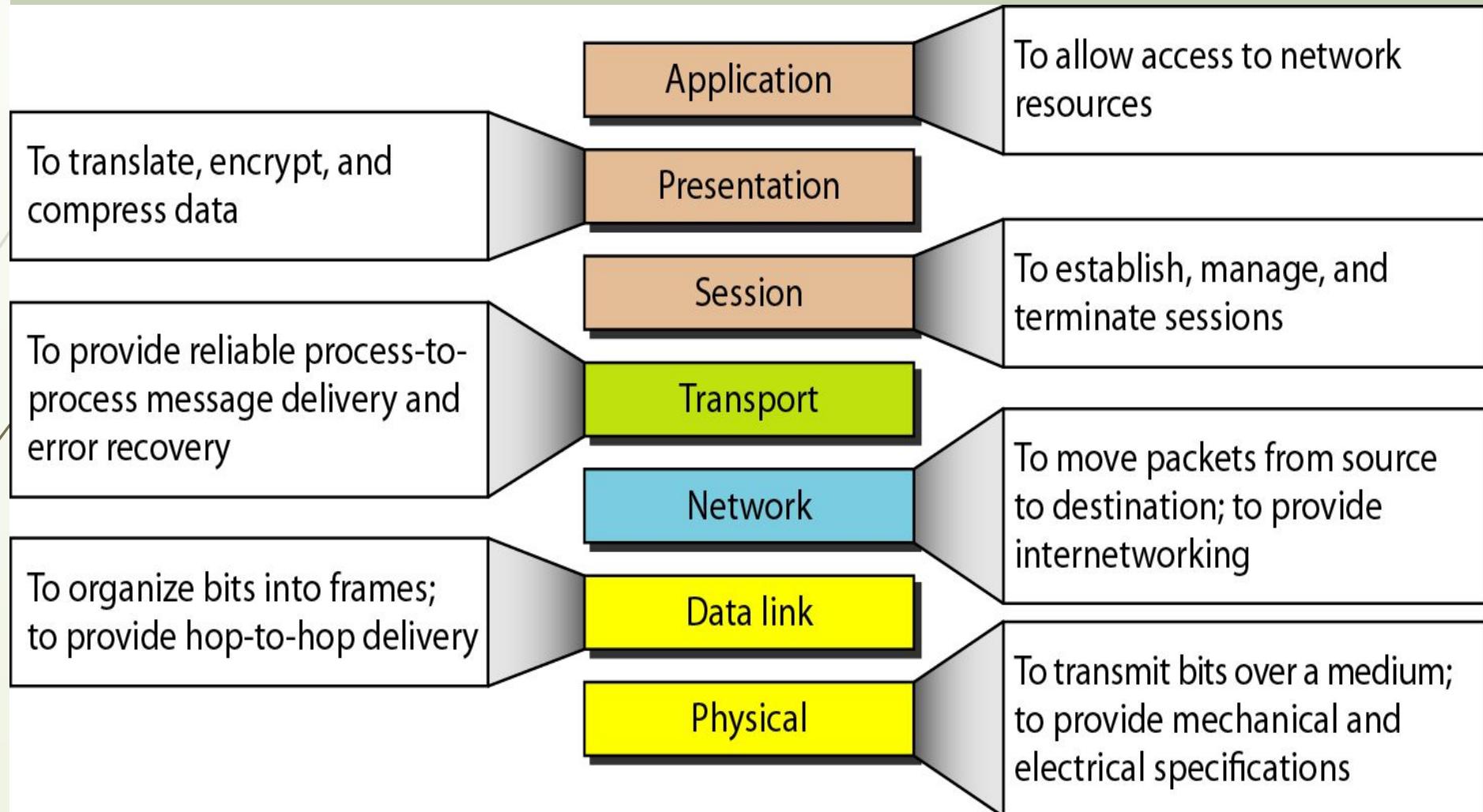




Note

- The application layer is responsible for providing services to the user.

Summary of layers

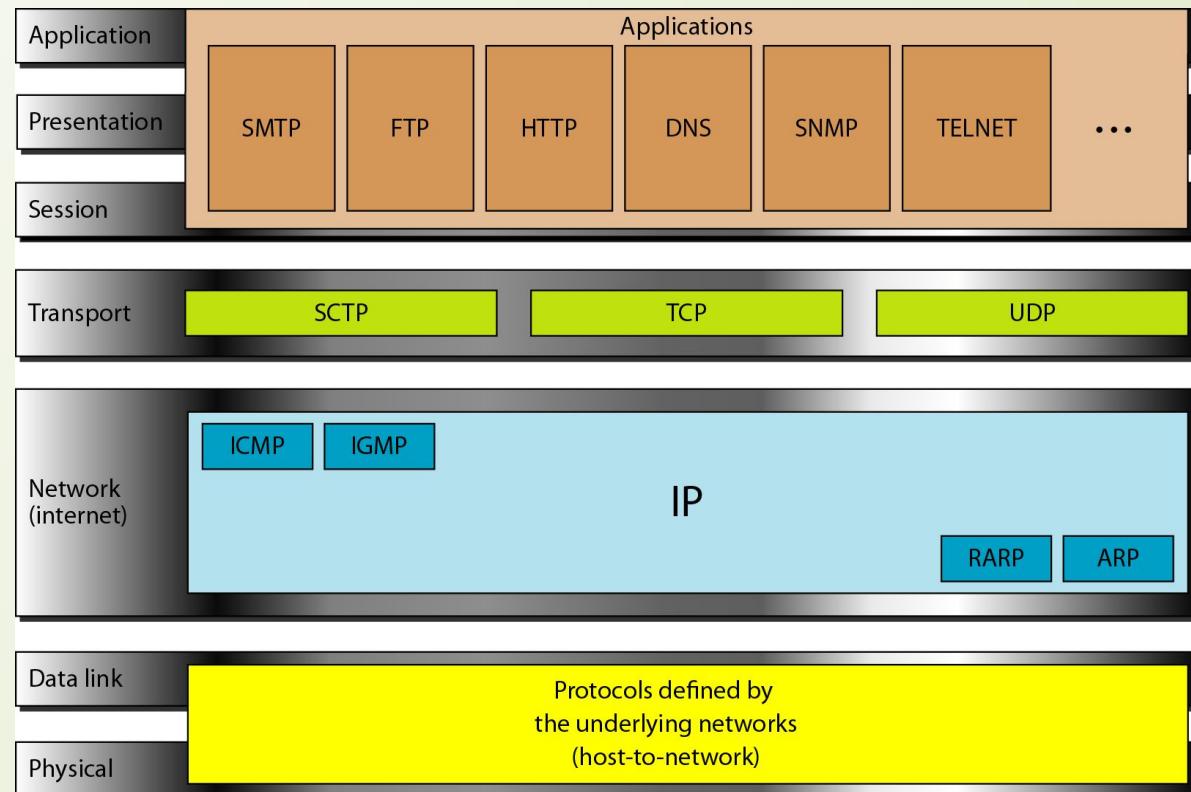




TCP/IP PROTOCOL SUITE

- The layers in the **TCP/IP protocol suite** do not exactly match those in the OSI model. The original TCP/IP protocol suite was defined as having four layers: **host-to-network**, **internet**, **transport**, and **application**. However, when TCP/IP is compared to OSI, we can say that the TCP/IP protocol suite is made of five layers: **physical**, **data link**, **network**, **transport**, and **application**.

TCP/IP and OSI model





ADDRESSING

□ *Four levels of addresses are used in an internet employing the TCP/IP protocols: physical, logical, port, and specific.*

Topics discussed in this section:

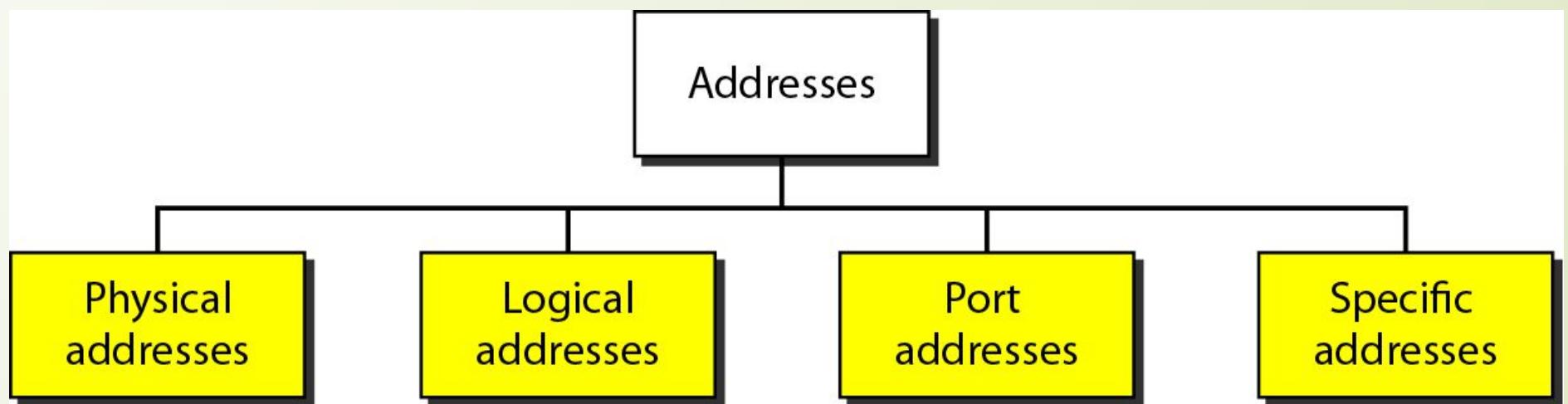
Physical Addresses

Logical Addresses

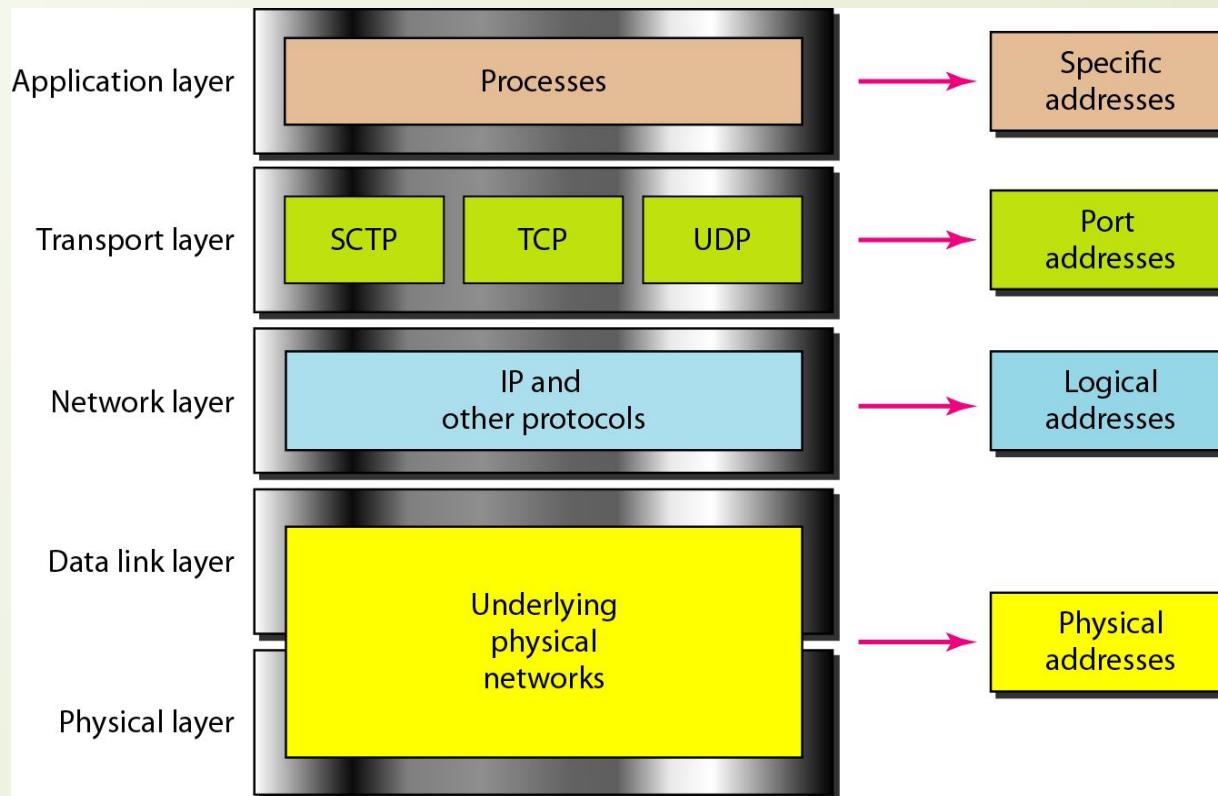
Port Addresses

Specific Addresses

Addresses in TCP/IP

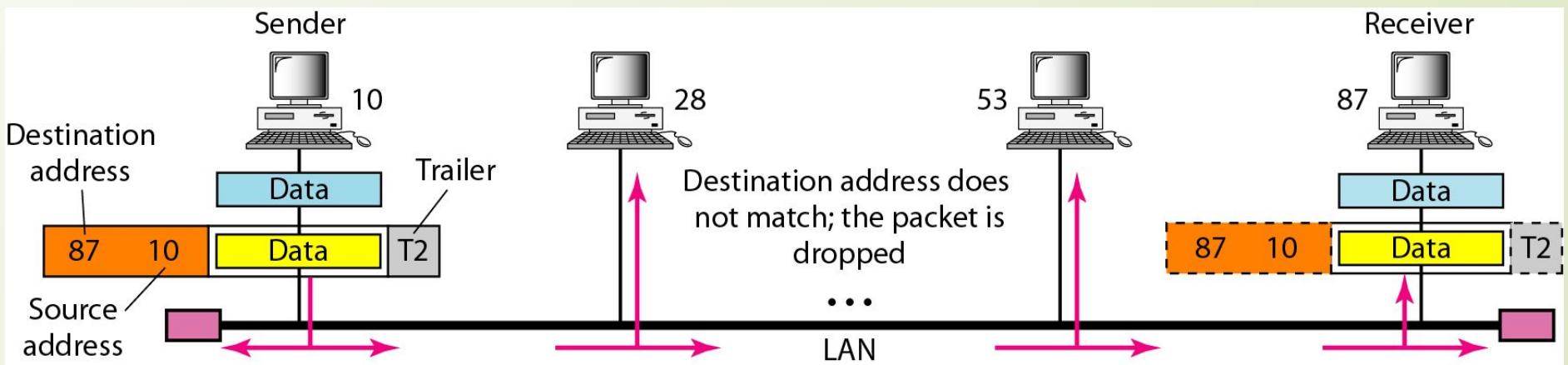


Relationship of layers and addresses in TCP/IP



Example 2.1: Physical Address

- In this figure, a node with physical address 10 sends a frame to a node with physical address 87. The two nodes are connected by a link (bus topology LAN). As the figure shows, the computer with physical address 10 is the sender, and the computer with physical address 87 is the receiver.



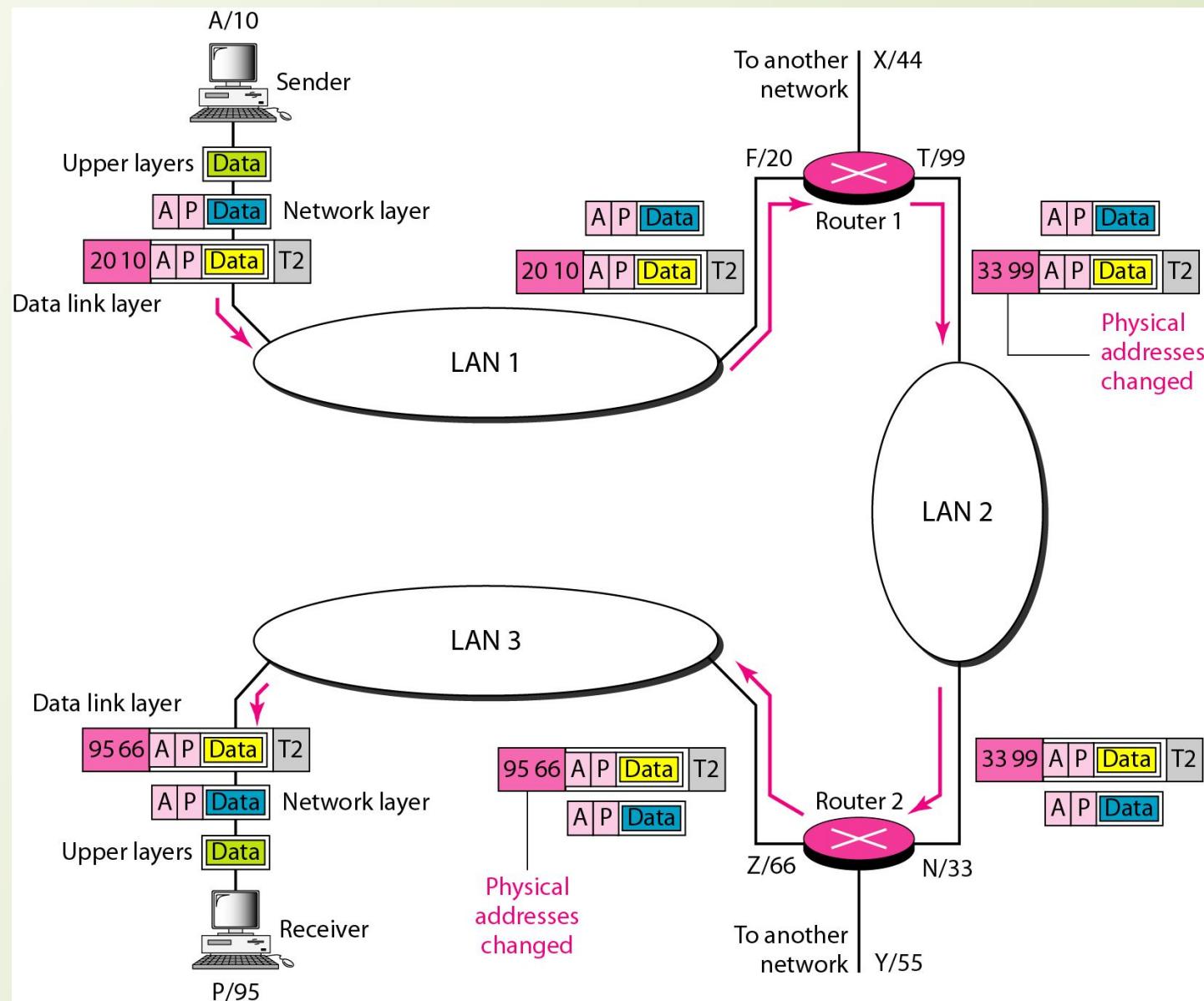
- 
- Most local-area networks use a **48-bit** (6-byte) physical address written as 12 hexadecimal digits; every byte (2 hexadecimal digits) is separated by a colon, as shown below:

07:01:02:01:2C:4B

A 6-byte (12 hexadecimal digits) physical address.

IP addresses

Figure in the next slide shows a part of an internet with two routers connecting three LANs. Each device (computer or router) has a pair of addresses (logical and physical) for each connection. In this case, each computer is connected to only one link and therefore has only one pair of addresses. Each router, however, is connected to three networks (only two are shown in the figure). So each router has three pairs of addresses, one for each connection.



Port addresses

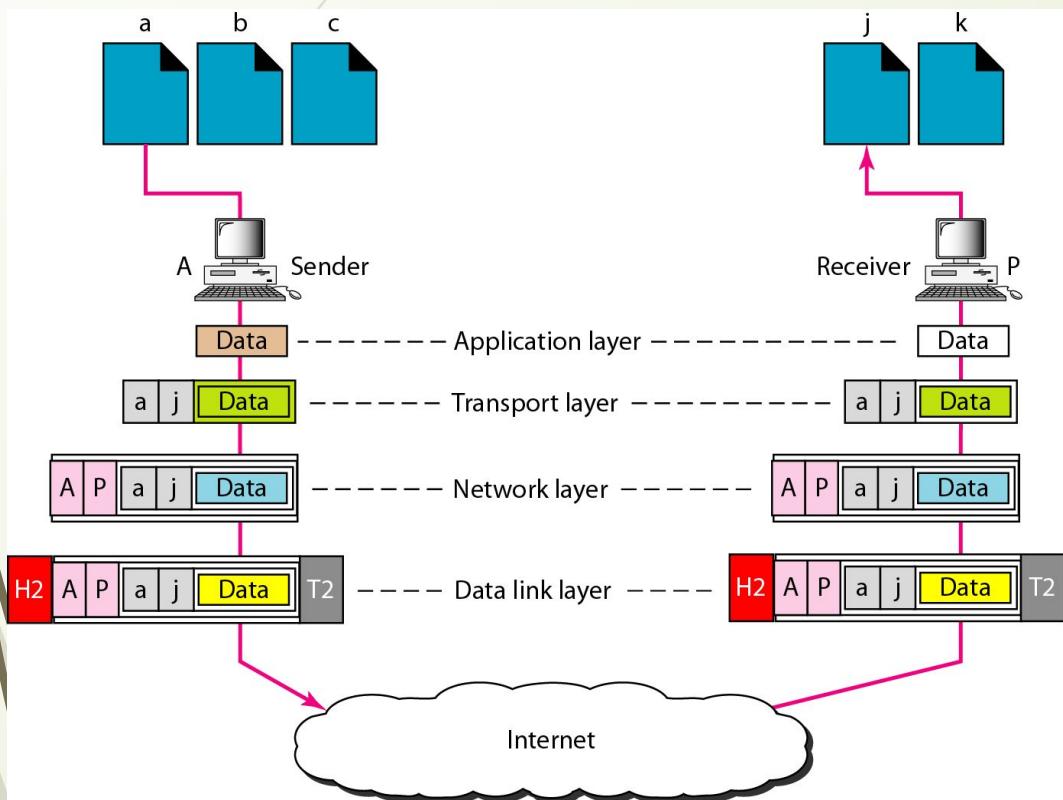
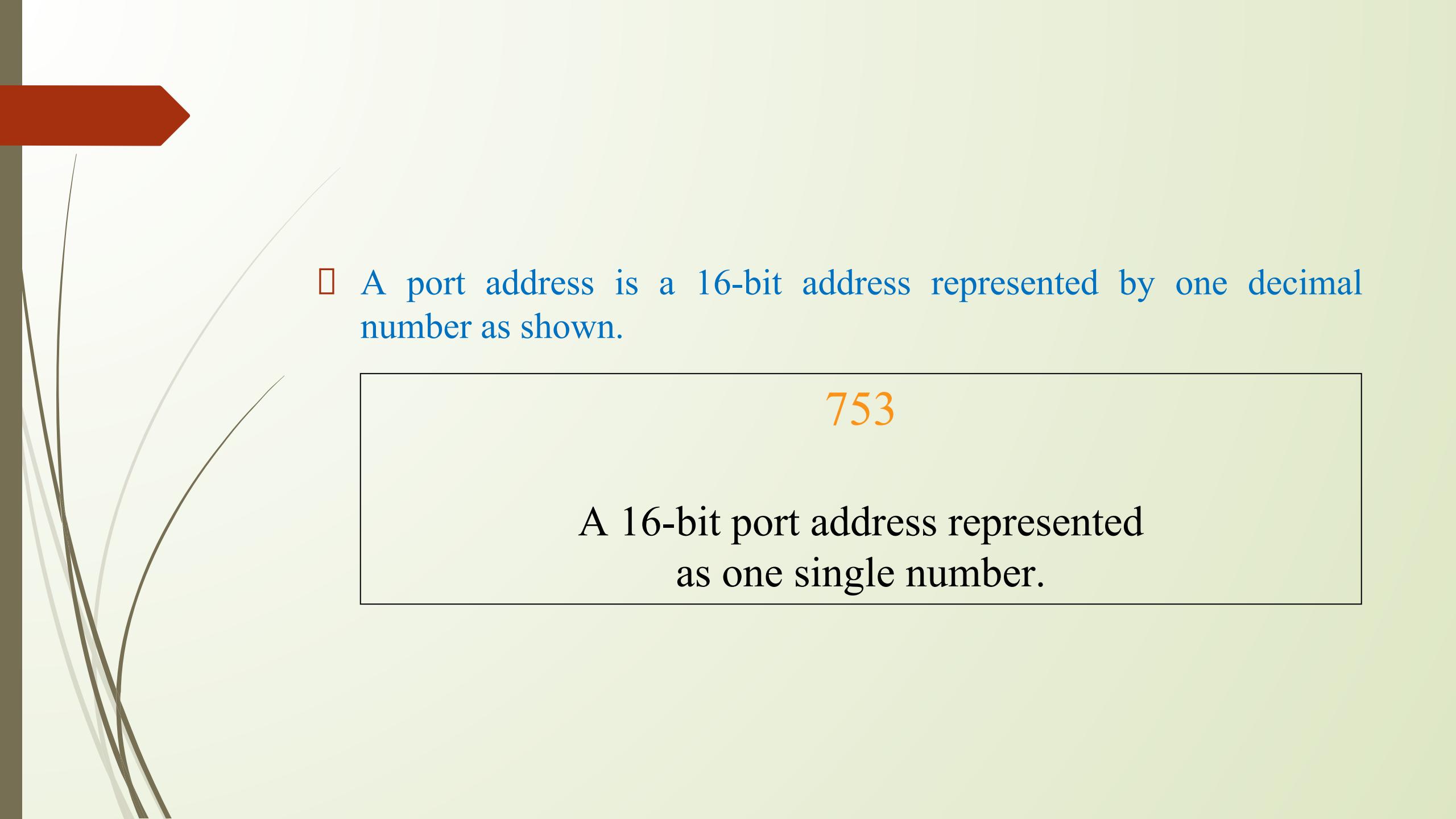


Figure shows a part of an internet with two routers connecting three LANs. Each device (computer or router) has a pair of addresses (logical and physical) for each connection. In this case, each computer is connected to only one link and therefore has only one pair of addresses. Each router, however, is connected to three networks (only two are shown in the figure). So each router has three pairs of addresses, one for each connection.



Note

- The physical addresses will change from hop to hop, but the logical addresses usually remain the same.

- 
- A port address is a 16-bit address represented by one decimal number as shown.

753

A 16-bit port address represented
as one single number.



Note

- The physical addresses change from hop to hop, but the logical and port addresses usually remain the same.



Example 2.6

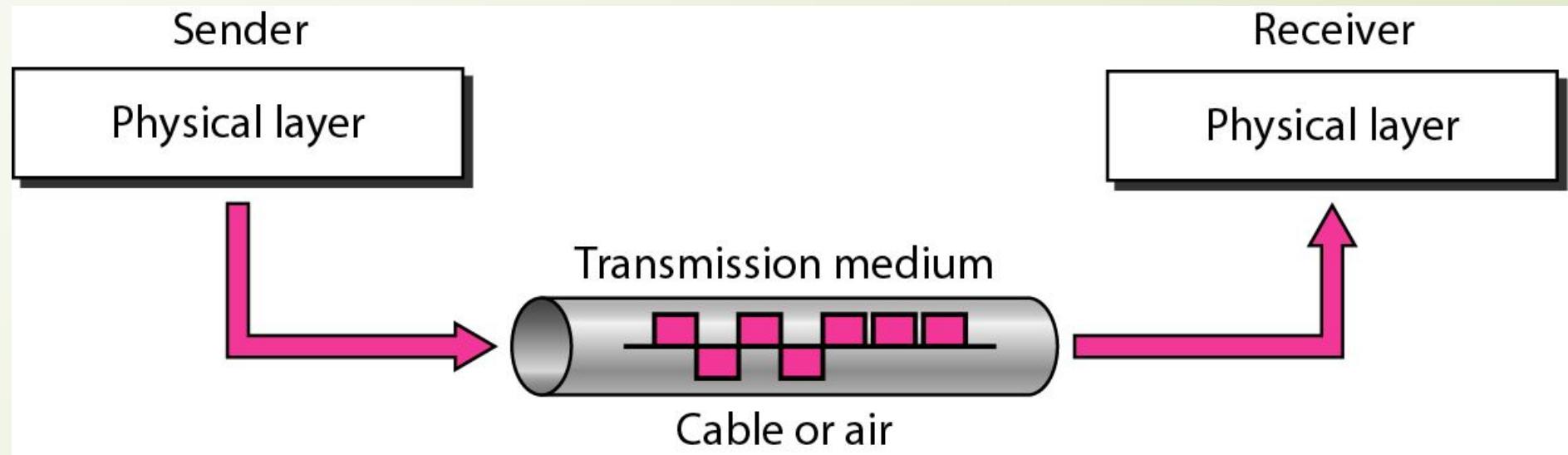
- Specific Addresses:
- These are Application-specific addresses. For instance, Uniform Resource Identifiers (URIs) or Uniform Resource Locators (URLs) that pinpoint specific resources on the web.



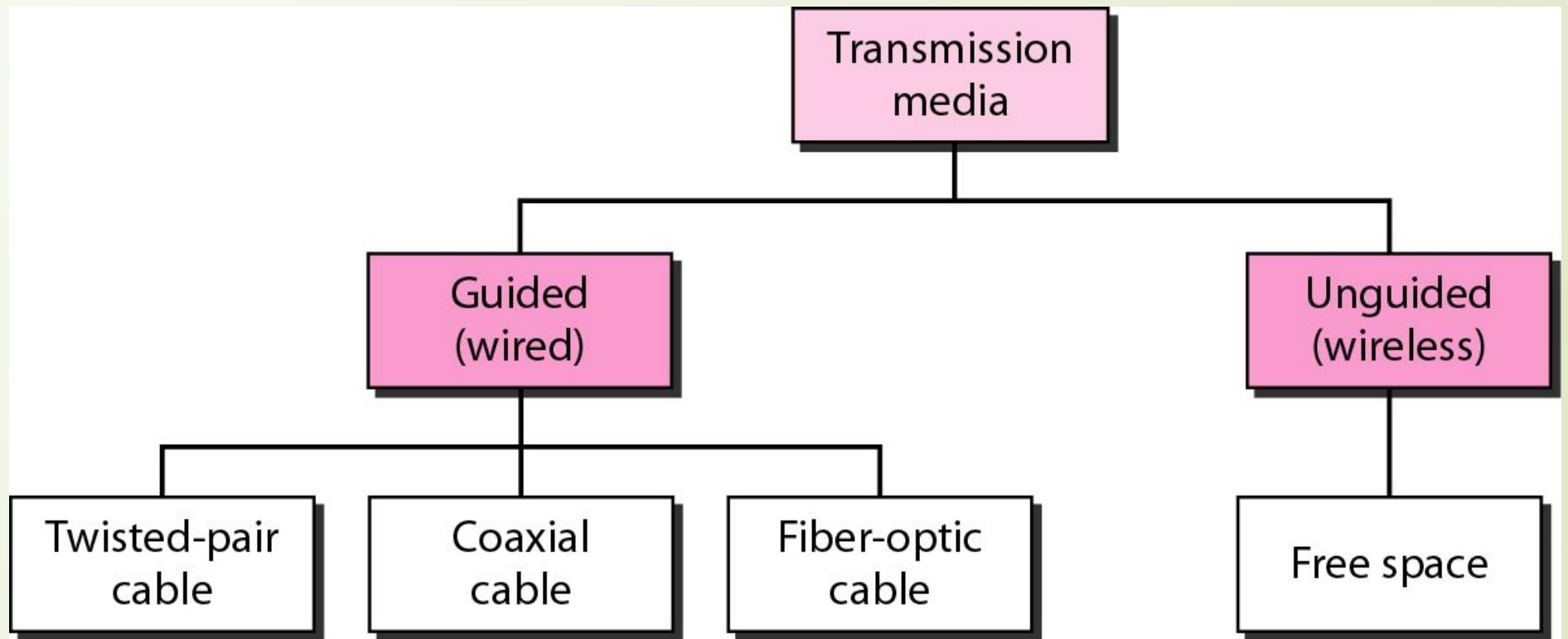
Chapter 3

Transmission Media

Transmission medium and physical layer



Classes of transmission media



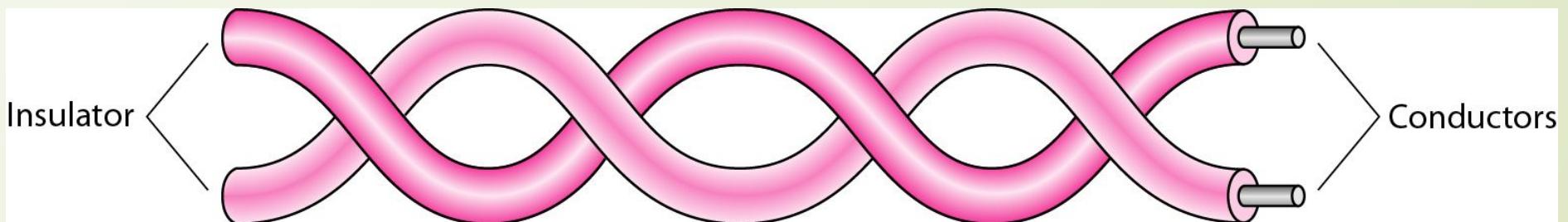


GUIDED MEDIA

- Guided media, which are those that provide a conduit from one device to another, include twisted-pair cable, coaxial cable, and fiber-optic cable.
- Topics discussed in this section:
 - Twisted-Pair Cable
 - Coaxial Cable
 - Fiber-Optic Cable

Twisted-pair cable

Twisted pair cable is a type of wiring in which two conductors of a single circuit are twisted together for the purposes of improving electromagnetic compatibility. The wires are twisted to cancel out electromagnetic interference (EMI) from external sources and crosstalk from adjacent pairs.





Types of Twisted-pair cable

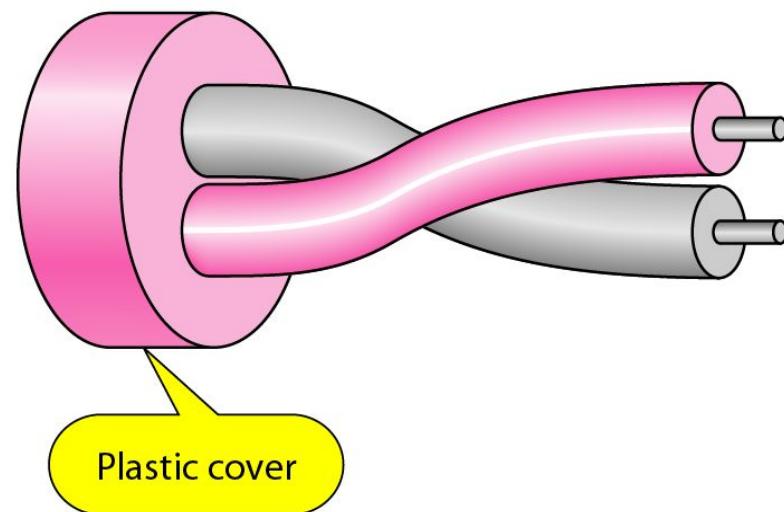
- **Unshielded Twisted Pair (UTP):**

- UTP cables are the most common type of twisted pair cabling.
- They lack shielding, which makes them more flexible and easier to install.
- Commonly used in telecommunication and networking.
- Examples include Category 5e (Cat 5e), Category 6 (Cat 6), and Category 6a (Cat 6a) cables.

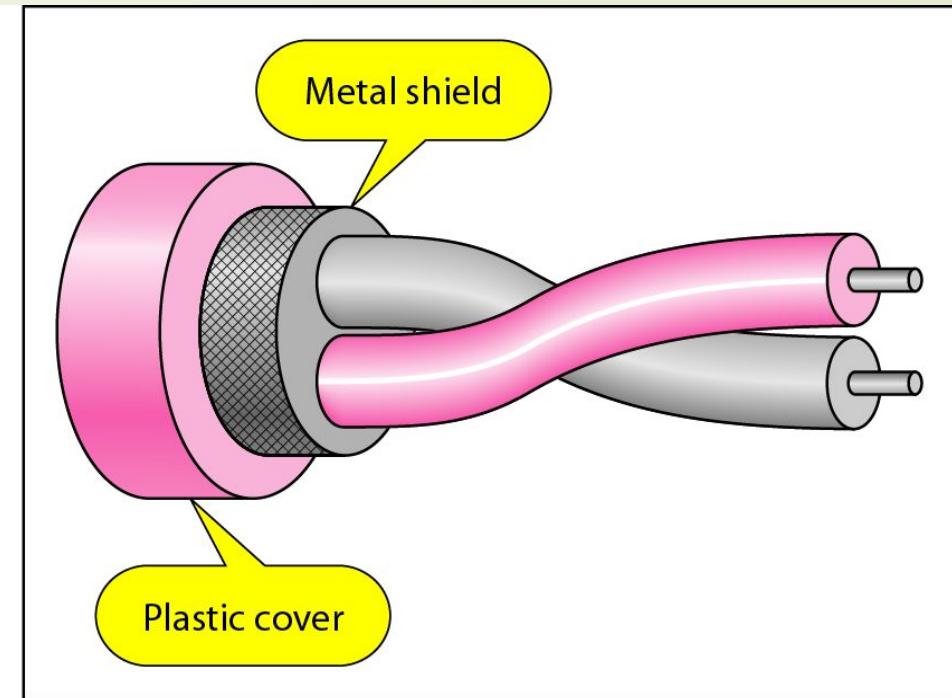
- **Shielded Twisted Pair (STP):**

- STP cables have a shielding layer that protects the twisted pairs from external EMI.
- The shielding can be in the form of a foil or braided mesh.
- They are used in environments with high interference and for high-speed networks.
- Examples include Category 7 (Cat 7) cables.

UTP and STP cables



a. UTP



b. STP

Categories of unshielded twisted-pair cables

Category	Specification	Data Rate (Mbps)	Use
1	Unshielded twisted-pair used in telephone	< 0.1	Telephone
2	Unshielded twisted-pair originally used in T-lines	2	T-1 lines
3	Improved CAT 2 used in LANs	10	LANs
4	Improved CAT 3 used in Token Ring networks	20	LANs
5	Cable wire is normally 24 AWG with a jacket and outside sheath	100	LANs
5E	An extension to category 5 that includes extra features to minimize the crosstalk and electromagnetic interference	125	LANs
6	A new category with matched components coming from the same manufacturer. The cable must be tested at a 200-Mbps data rate.	200	LANs
7	Sometimes called SSTP (shielded screen twisted-pair). Each pair is individually wrapped in a helical metallic foil followed by a metallic foil shield in addition to the outside sheath. The shield decreases the effect of crosstalk and increases the data rate.	600	LANs

Working Principle of Twisted Pair Cable

- **Twisting:**

- The twisting of wires helps in reducing electromagnetic interference (EMI) and crosstalk between pairs.
- When two wires are twisted, any external EMI affects both wires equally, and since the signals are out of phase, the interference cancels out.

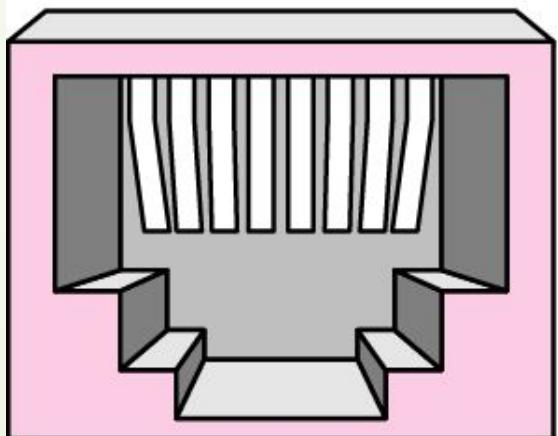
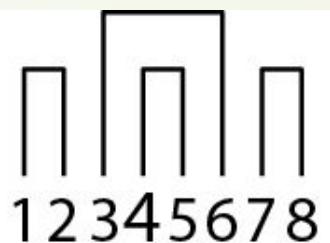
- **Transmission of Data:**

- Data is transmitted over the twisted pairs as electrical signals.
- Differential signaling is often used, where two wires carry the same signal but with opposite polarity. This helps in further reducing noise and improving signal integrity.

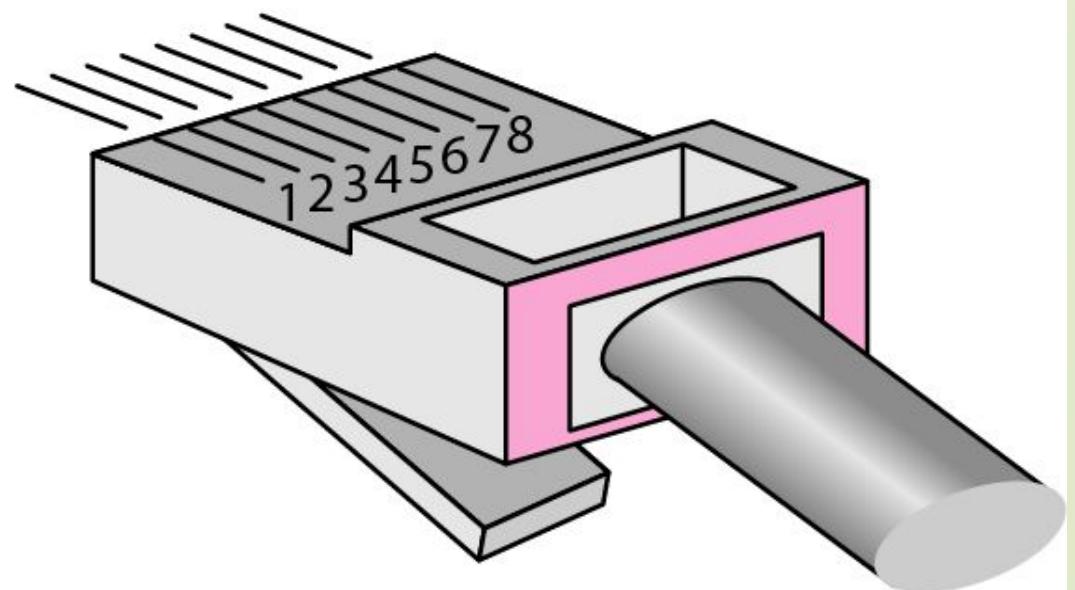
- **Connectors:**

- Twisted pair cables typically use RJ45 connectors, especially in networking applications.

UTP connector



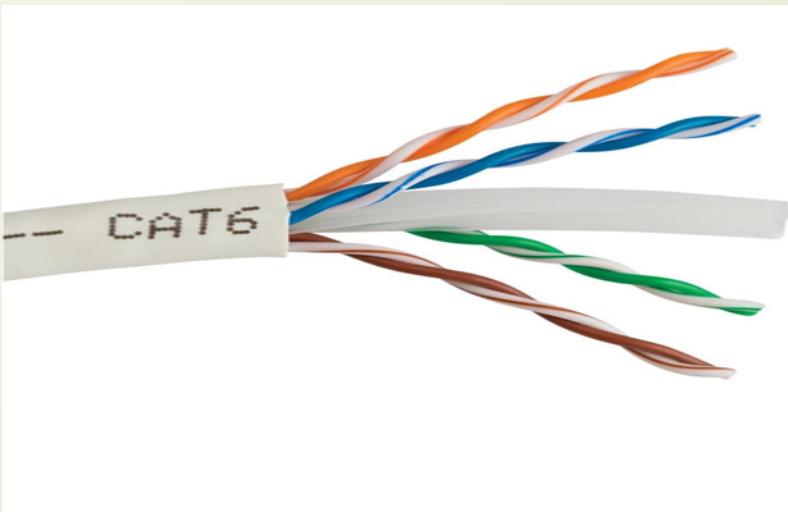
RJ-45 Female



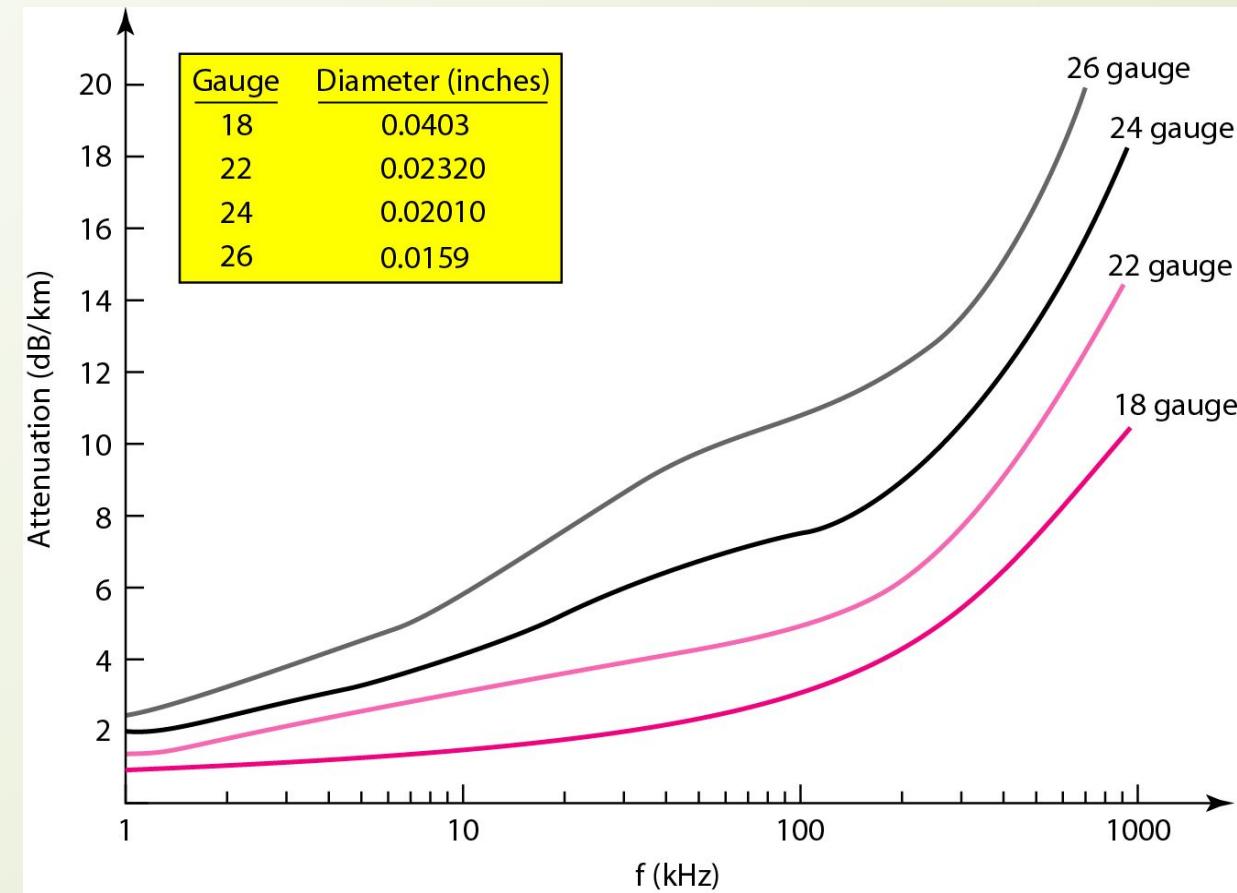
RJ-45 Male

Uses of Twisted Pair Cables

- It's important to clarify that typically the term "twisted pair cable" refers to cables with multiple twisted pairs. For example, Cat 5e, Cat 6, and Cat 7 cables typically contain four twisted pairs (8 wires total).
- The most important use of the Twisted Pair Cables is Standard Ethernet Cable (Cat 5e, Cat 6, Cat 7).

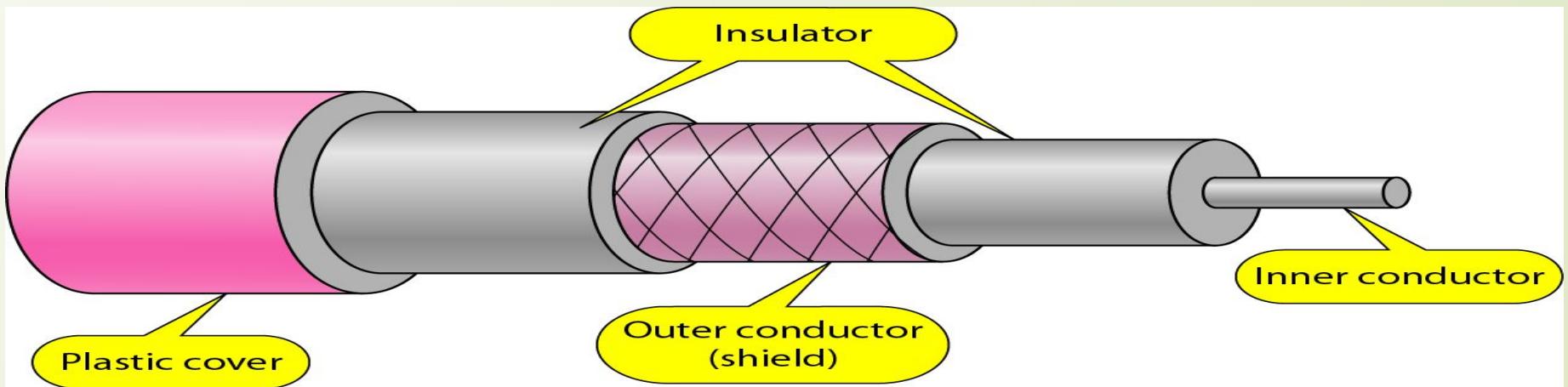


UTP performance



Coaxial cable

- Coaxial cable, or coax, is a type of electrical cable consisting of a central conductor, an insulating layer, a metallic shield, and an outer insulating layer.
- It is used for transmitting cable television signals, internet connections, and other data communications.





Structure of Coaxial Cable

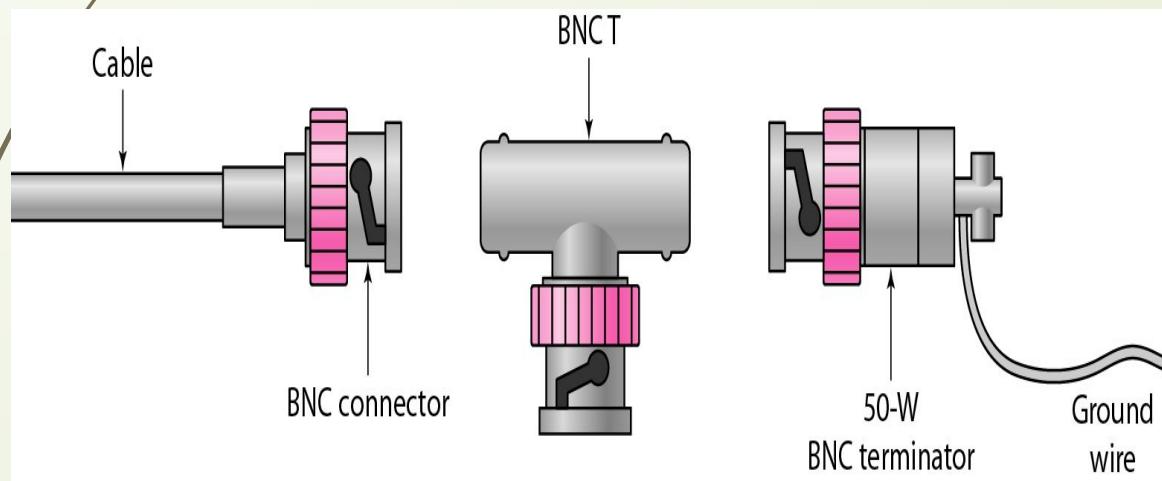
- **Central Conductor:** Usually made of copper or aluminum, it carries the electrical signal.
- **Dielectric Insulator:** Surrounds the central conductor, made of a non-conductive material (like plastic), it insulates the conductor.
- **Metallic Shield:** A braided or solid metallic layer (often made of copper or aluminum) that protects the signal from external electromagnetic interference (EMI).
- **Outer Insulator:** The outermost layer made of plastic or rubber, it protects the cable from physical damage.

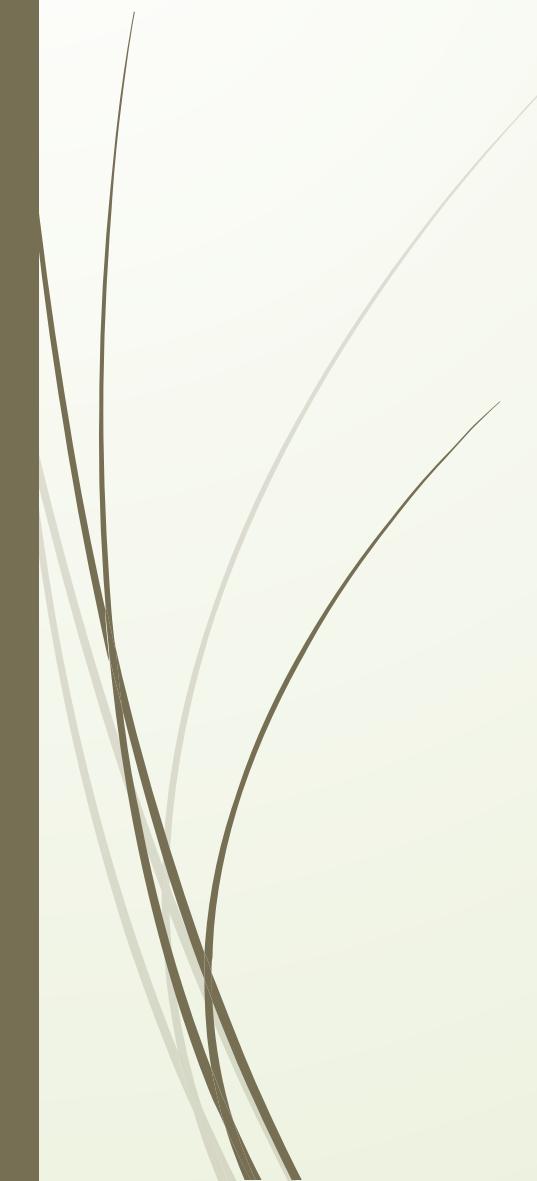
Categories of coaxial cables

<i>Category</i>	<i>Impedance</i>	<i>Use</i>
RG-59	75 Ω	Cable TV
RG-58	50 Ω	Thin Ethernet
RG-11	50 Ω	Thick Ethernet

BNC connectors

A BNC (Bayonet Neill-Concelman) connector is a type of coaxial RF connector used for terminating coaxial cables. It is widely used in various applications such as video and audio transmission, RF and microwave systems, and test equipment. The BNC connector is named after its inventors, Paul Neill and Carl Concelman.





Structure and Features of BNC Connector

1. Bayonet Locking Mechanism:

- The BNC connector features a bayonet locking mechanism, which ensures a secure and reliable connection. The connector has a rotating ring with two small lugs that fit into matching slots on the mating connector.
- To connect, you align the lugs with the slots, push the connector in, and then twist the ring to lock it in place.

2. Center Pin:

- The center pin of the BNC connector makes contact with the central conductor of the coaxial cable, carrying the signal.

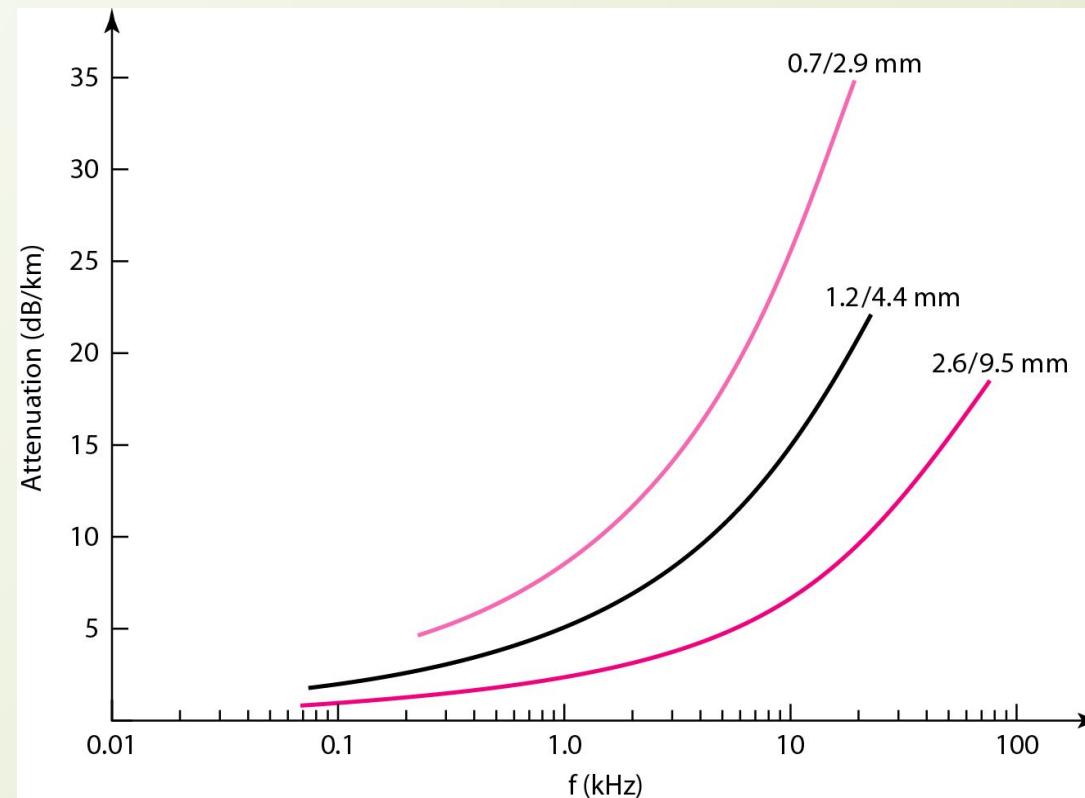
3. Outer Shield:

- The outer metal shell of the BNC connector contacts the outer shield of the coaxial cable, providing grounding and shielding from electromagnetic interference (EMI).

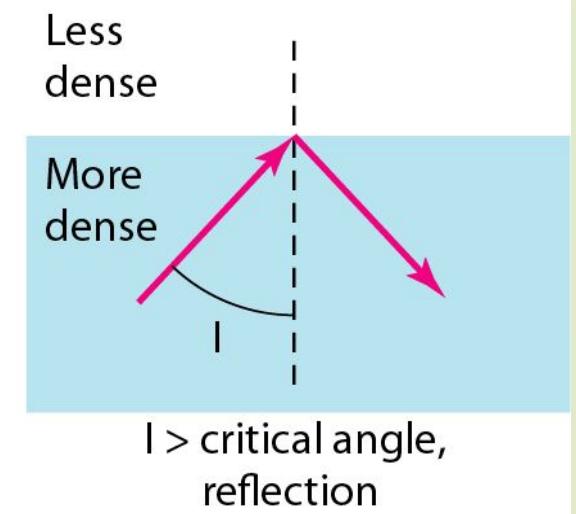
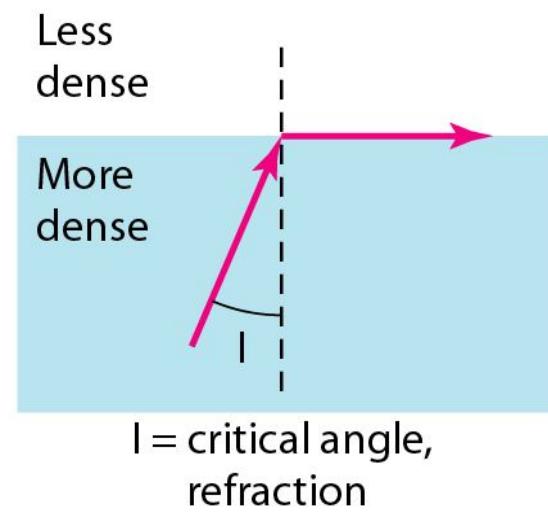
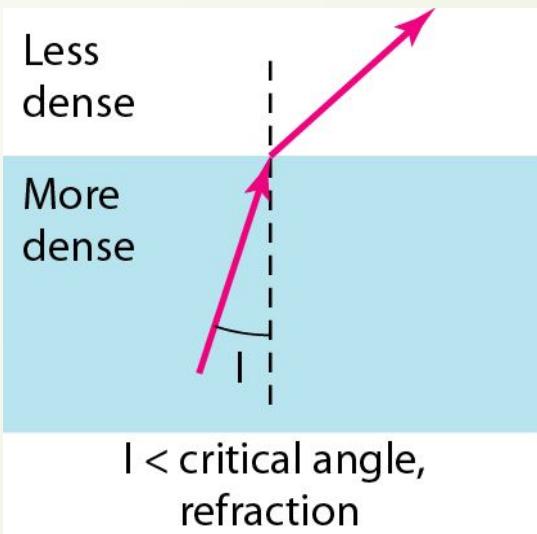
4. Insulating Dielectric:

- An insulating material separates the center pin and the outer shield within the connector, maintaining the characteristic impedance of the coaxial cable.

Coaxial cable performance

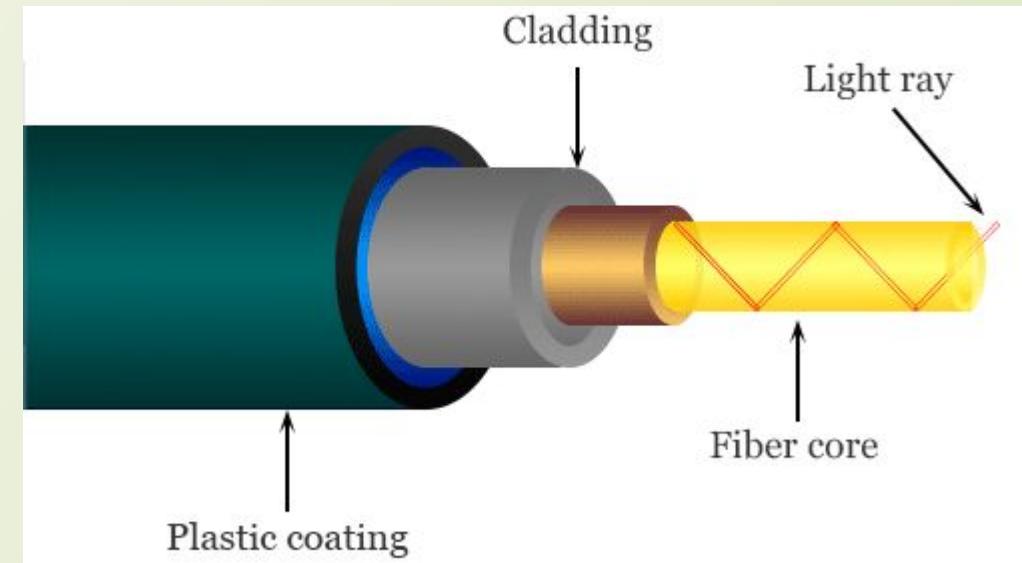
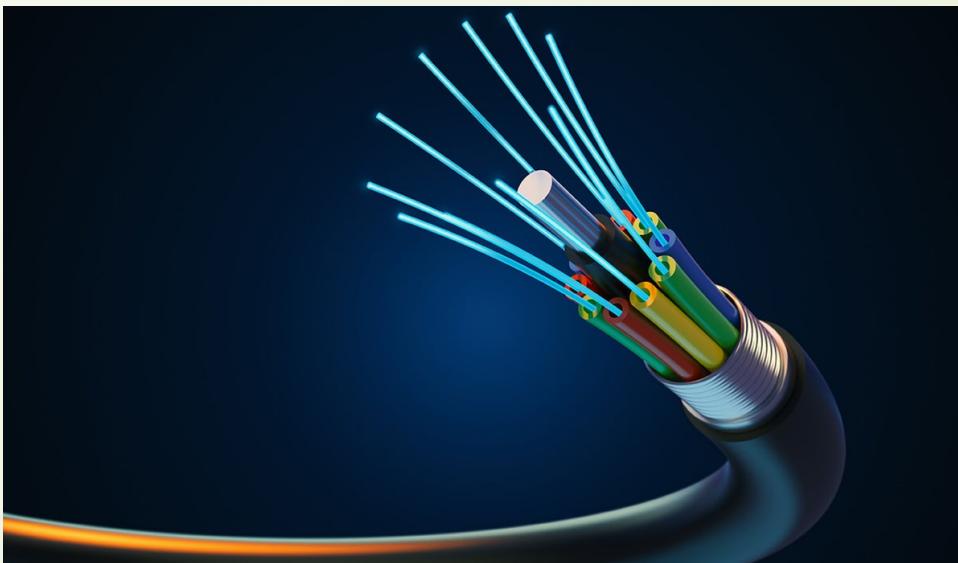


Bending of light ray



Optical Fiber

- Optical fiber is a flexible, transparent fiber made of high-quality glass (silica) or plastic, slightly thicker than a human hair, that functions as a waveguide or light pipe to transmit light between the two ends of the fiber. It is primarily used in telecommunications and networking, where it enables long-distance, high-speed data transmission. Optical fibers operate on the principle of total internal reflection, allowing light to be guided through the core of the fiber with minimal loss. This makes them highly effective for transmitting large amounts of data over long distances with high fidelity and low attenuation.



Structure of Optical Fiber Cable

Core:

- The core is the central part of the optical fiber where light is transmitted.
- It is made of high-purity glass or plastic.
- The core is the medium through which the light signals travel. The diameter of the core determines the mode of transmission (single-mode or multi-mode).

Cladding:

- Surrounding the core is the cladding, which is also made of glass or plastic but with a lower refractive index than the core.
- The cladding reflects the light back into the core through total internal reflection, ensuring the light signals remain within the core without escaping, which allows the light to travel long distances.

Coating/Buffer Coating:

- The coating is a layer of plastic that surrounds the cladding.
- The primary function is to protect the core and cladding from physical damage and moisture. It also adds mechanical strength to the fiber.

Structure of Optical Fiber Cable

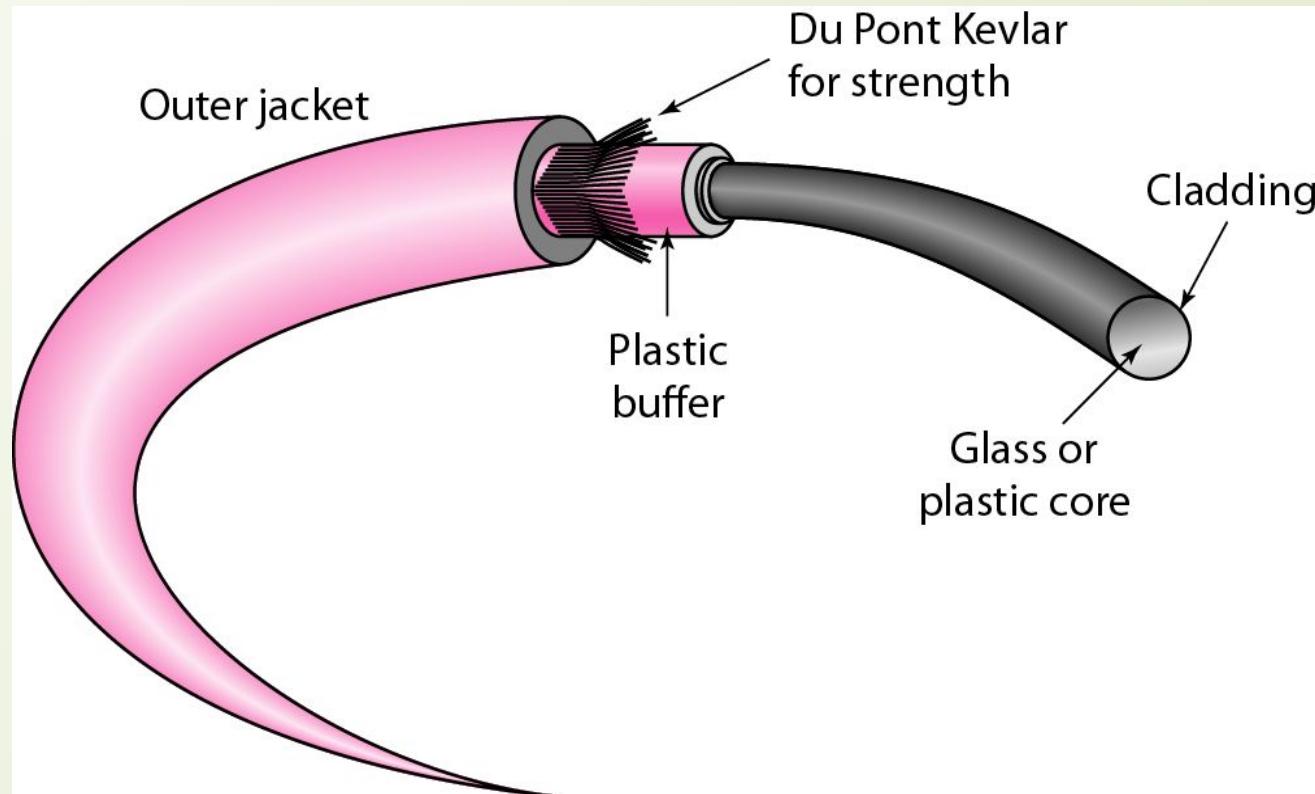
Strengthening Fibers:

- These are made of materials like aramid yarn (e.g., Kevlar).
- They provide additional strength to the optical fiber cable, preventing it from breaking during handling, installation, and operation.

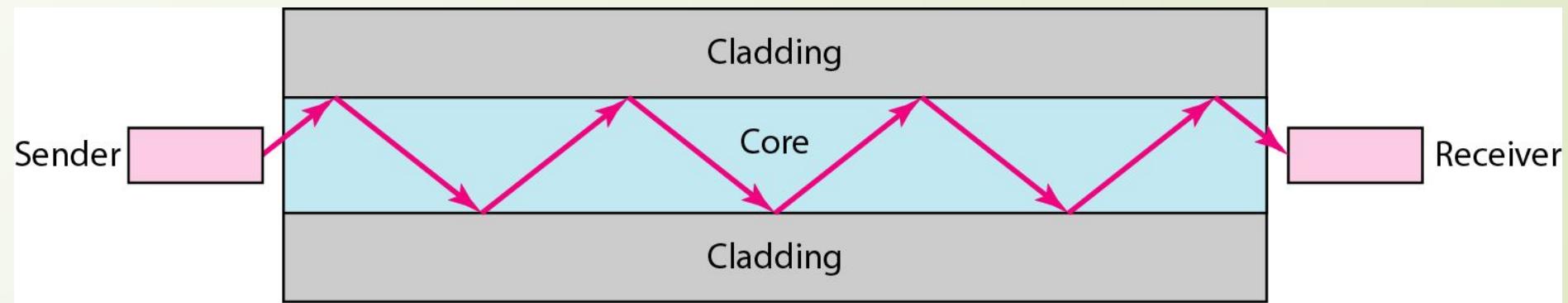
Outer Jacket:

- The outermost layer of the optical fiber cable, usually made of durable materials like PVC (polyvinyl chloride) or polyurethane.
- The outer jacket protects the entire assembly from environmental factors such as moisture, chemicals, and physical abrasion.

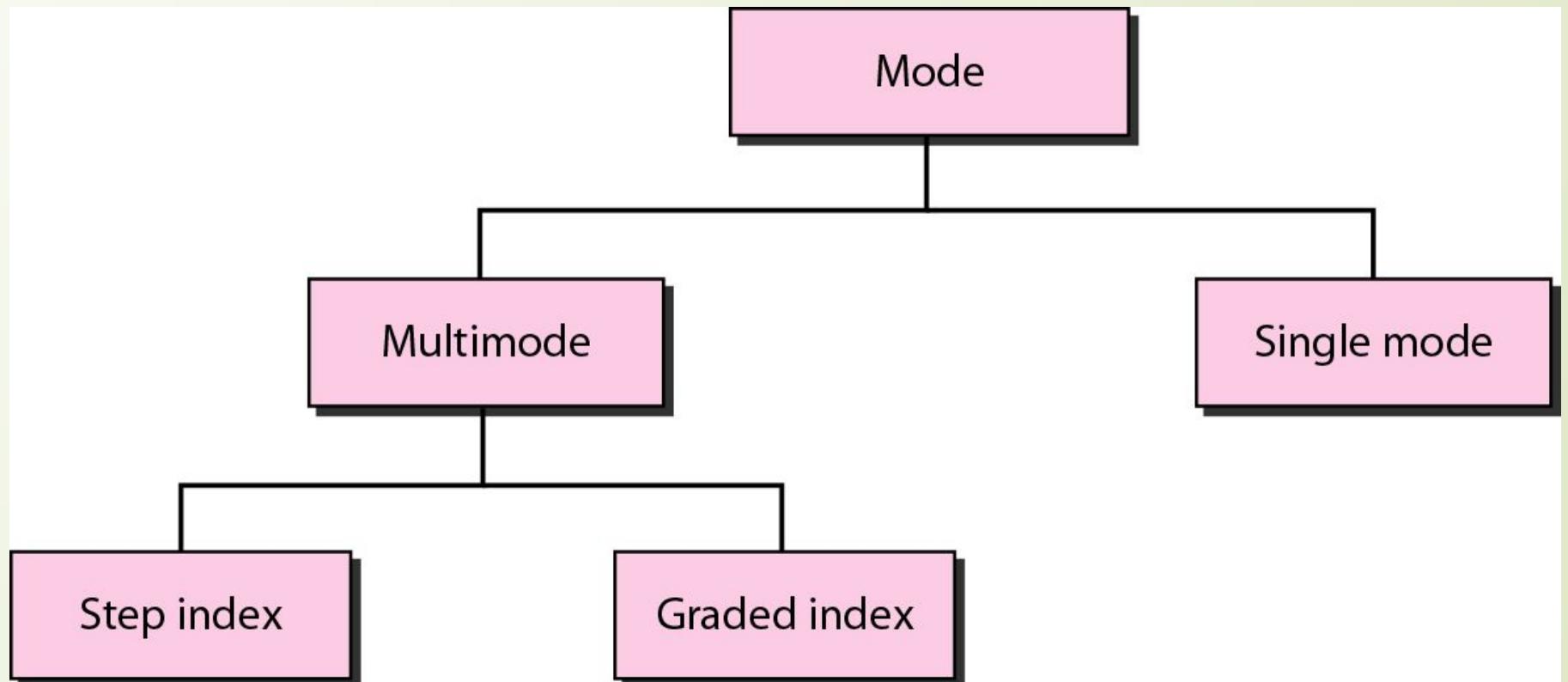
Fiber construction



Optical fiber



Propagation modes



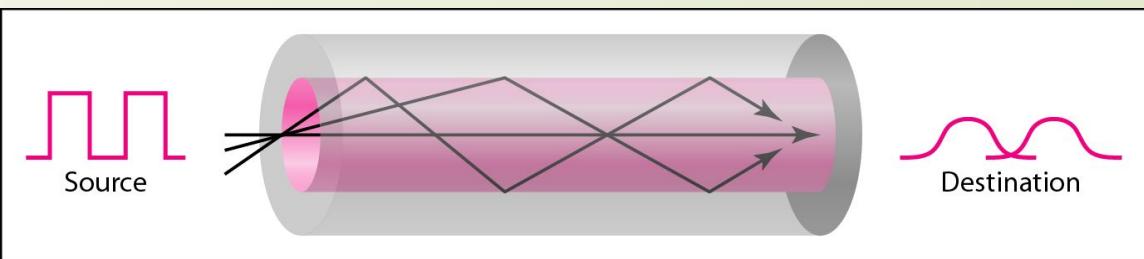
Modes

Single-Mode Fiber (SMF):

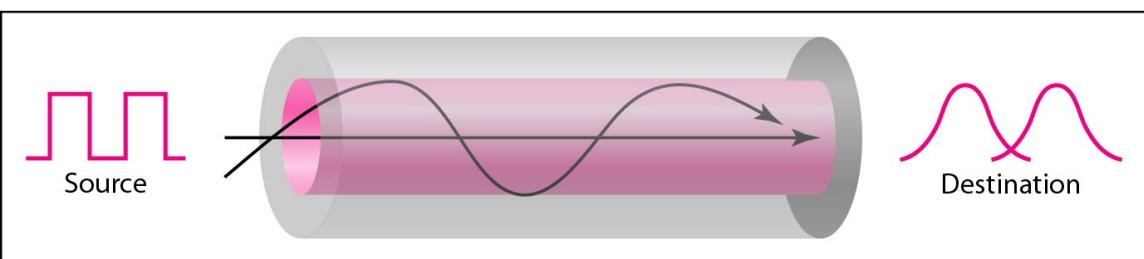
Definition: A type of optical fiber designed to carry light directly down the fiber with minimal loss.

Multi-Mode Fiber (MMF):

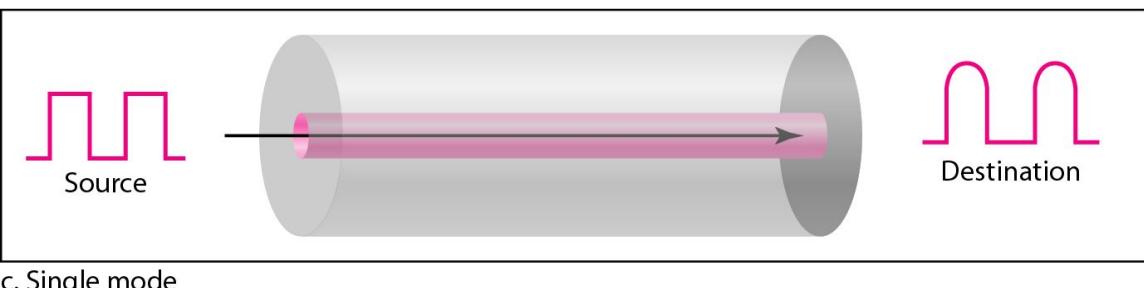
Definition: A type of optical fiber that can carry multiple light rays or modes simultaneously.



a. Multimode, step index



b. Multimode, graded index



c. Single mode

Modes

Parameter	Single-Mode Fiber (SMF)	Multi-Mode Fiber (MMF)
Core Diameter	8-10 micrometers	50-62.5 micrometers
Cladding Diameter	125 micrometers	125 micrometers
Light Source	Laser	LED or Laser
Wavelengths Used	1310 nm, 1550 nm	850 nm, 1300 nm
Bandwidth	Higher, virtually unlimited	Lower, up to several hundred MHz/km
Modal Dispersion	Minimal	Higher
Attenuation	Lower (0.4 dB/km at 1310 nm)	Higher (3.5 dB/km at 850 nm)
Cost	Higher	Lower
Installation Complexity	More complex	Easier

Fiber types

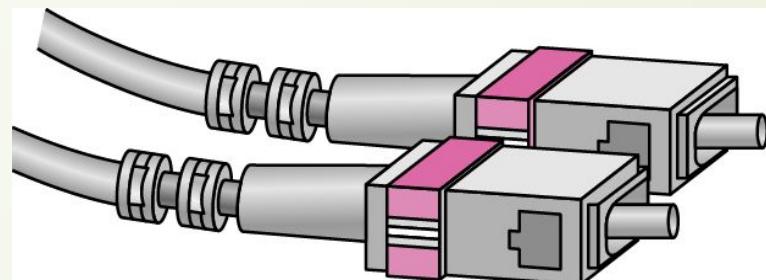
Type	Core (μm)	Cladding (μm)	Mode
50/125	50.0	125	Multimode, graded index
62.5/125	62.5	125	Multimode, graded index
100/125	100.0	125	Multimode, graded index
7/125	7.0	125	Single mode

Step Index fiber and Graded Index Fiber

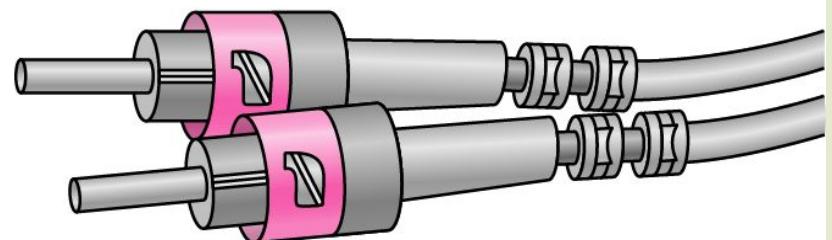
- Step Index Fiber and Graded Index Fiber are two types of optical fibers that differ in the way their core's refractive index is structured.

Parameter	Step Index Fiber	Graded Index Fiber
Core Refractive Index	Uniform	Gradually decreasing from center to edge
Cladding Refractive Index	Uniform, lower than core	Uniform, lower than the outer core
Index Profile	Abrupt step at core-cladding boundary	Gradual, parabolic-like profile
Light Path	Sharp reflections within the core	Curved paths due to continuous refraction
Modal Dispersion	Higher in multi-mode fibers	Lower due to more uniform travel times
Bandwidth	Lower in multi-mode fibers	Higher due to reduced dispersion
Manufacturing Complexity	Simpler and cheaper	More complex and expensive
Applications	Single-mode for long-distance, high-bandwidth	Multi-mode for medium distance, high-bandwidth

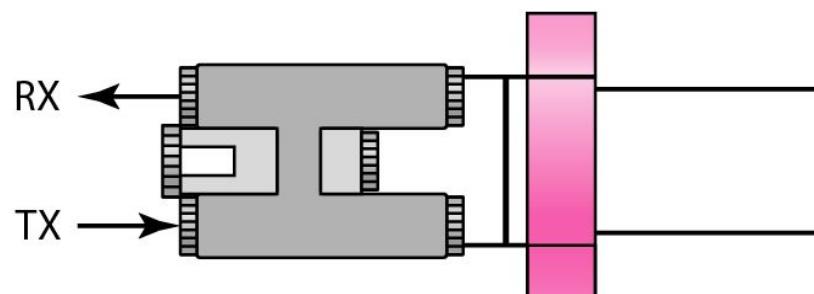
Fiber-optic cable connectors



SC connector

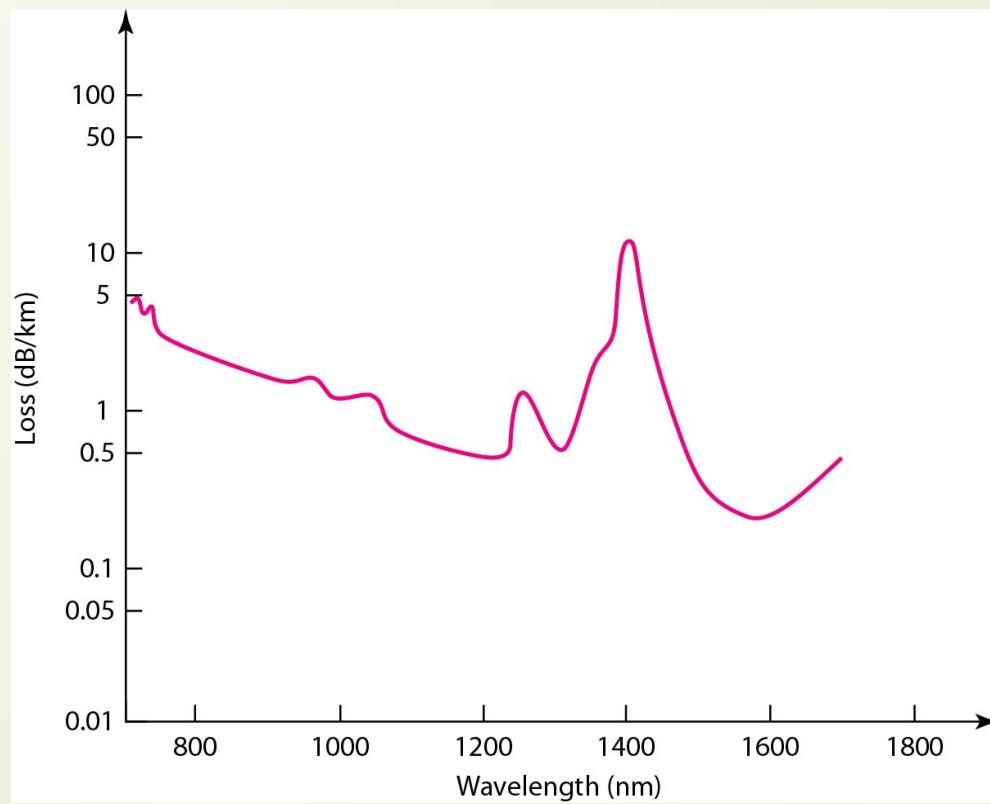


ST connector



MT-RJ connector

Optical fiber performance

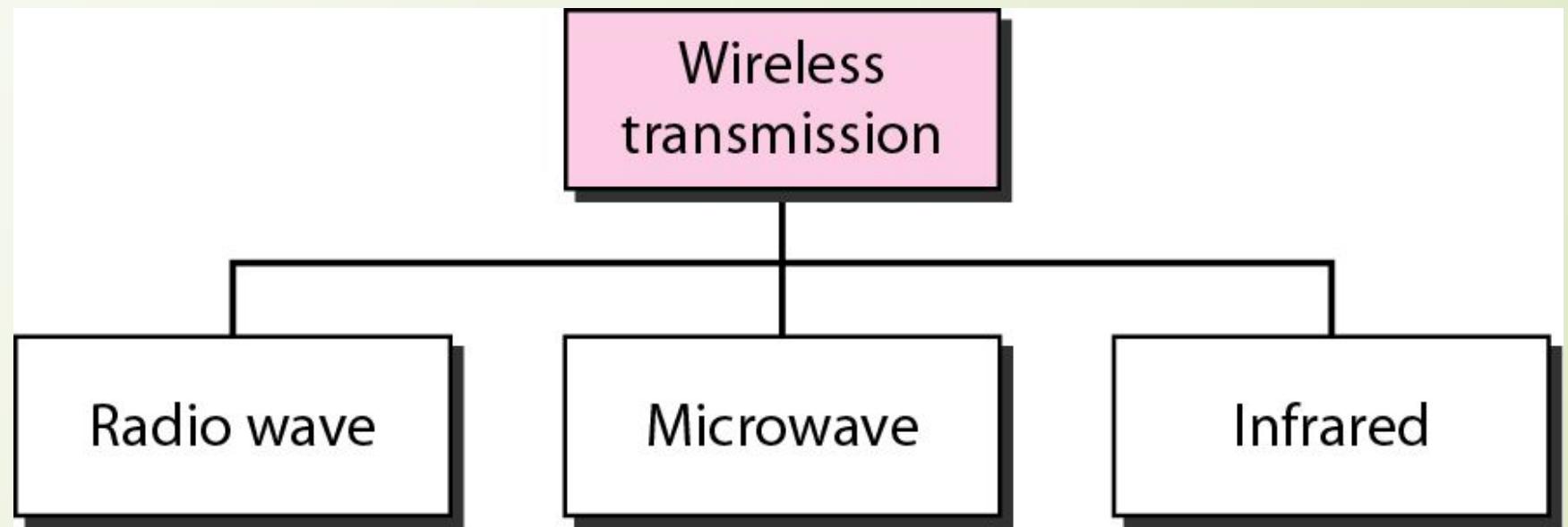




UNGUIDED MEDIA: WIRELESS

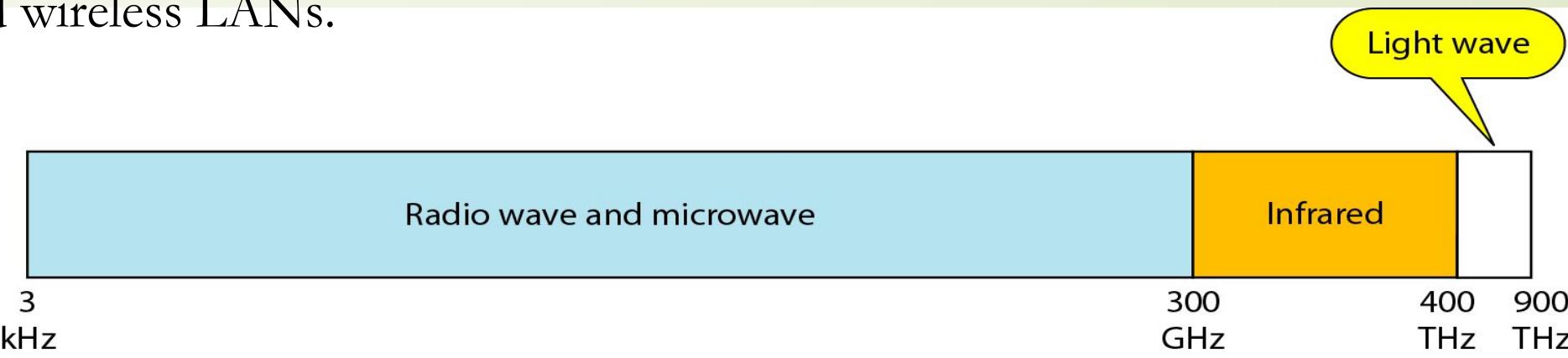
- Unguided media transport electromagnetic waves without using a physical conductor. This type of communication is often referred to as wireless communication.

Wireless transmission waves



Electromagnetic spectrum for wireless communication

- **Radio Waves:** Typically range from 3 Hz to 300 GHz. This broad range encompasses many different types of waves, including microwaves. Radio waves are used for multicast communications, such as radio and television, and paging systems.
- **Microwaves:** A subset of radio waves, typically ranging from 300 MHz (0.3 GHz) to 300 GHz. This range includes the 6 GHz frequency. Microwaves are used for unicast communication such as cellular telephones, satellite networks, and wireless LANs.



Electromagnetic spectrum for wireless communication

Infrared (IR) waves are a type of electromagnetic radiation with wavelengths longer than visible light but shorter than microwaves. They are typically categorized into three main regions based on their wavelength: near-infrared, mid-infrared, and far-infrared. Infrared signals can be used for short-range communication in a closed area using line-of-sight propagation.

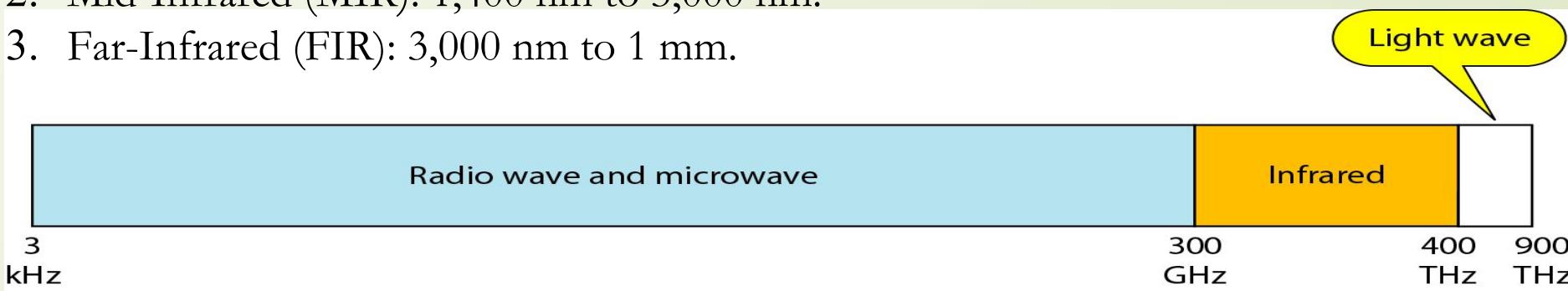
Characteristics

1. Wavelength and Frequency:

1. Wavelength: Ranges from approximately 700 nanometers (nm) to 1 millimeter (mm).
2. Frequency: Ranges from about 300 GHz to 430 THz.

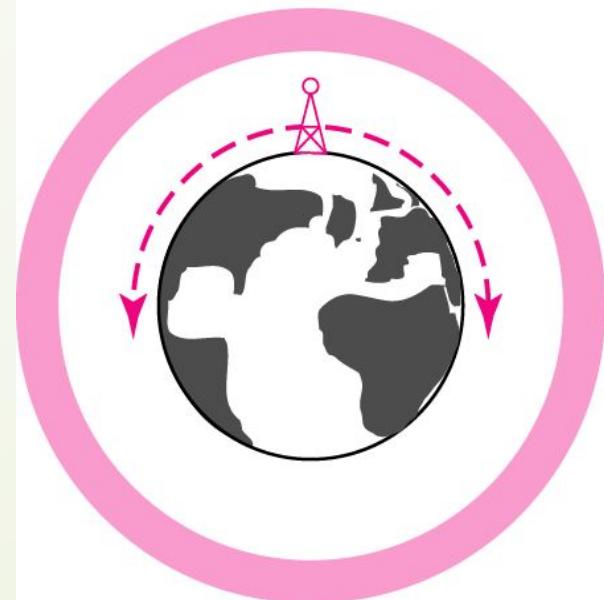
2. Sub-Divisions:

1. Near-Infrared (NIR): 700 nm to 1,400 nm.
2. Mid-Infrared (MIR): 1,400 nm to 3,000 nm.
3. Far-Infrared (FIR): 3,000 nm to 1 mm.



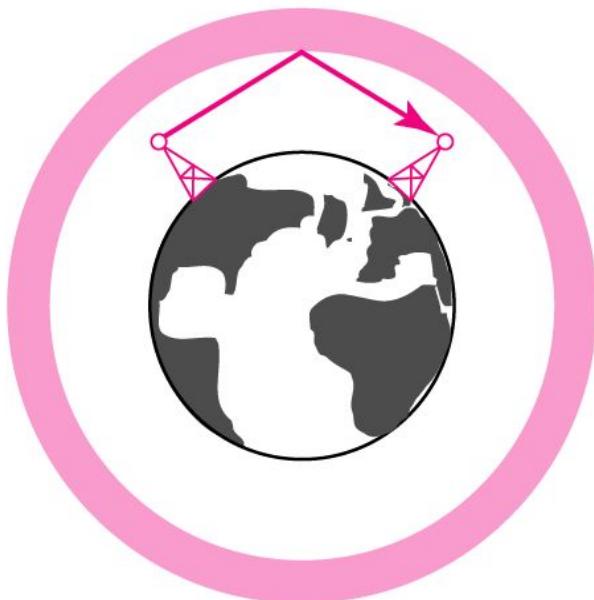
Propagation methods

Ionosphere



Ground propagation
(below 2 MHz)

Ionosphere



Sky propagation
(2–30 MHz)

Ionosphere

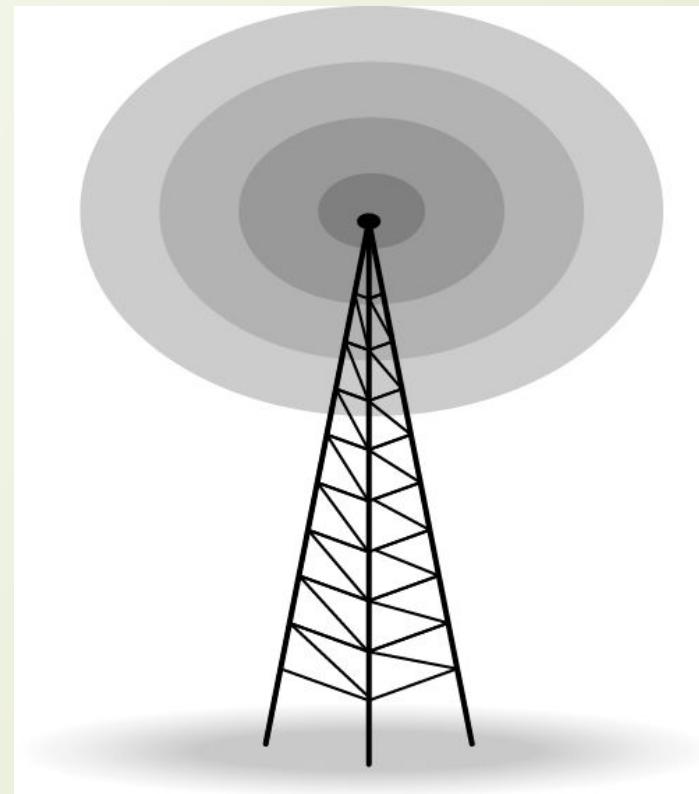


Line-of-sight propagation
(above 30 MHz)

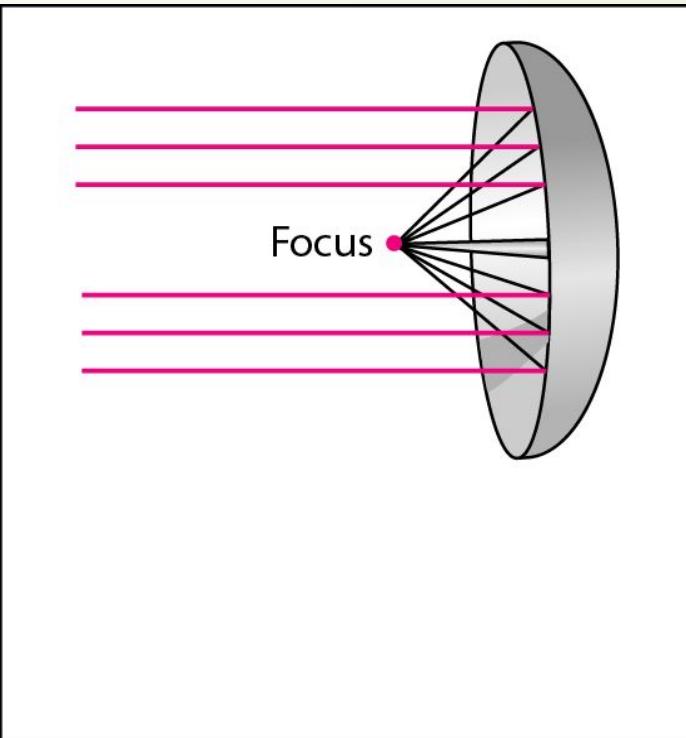
Bands

<i>Band</i>	<i>Range</i>	<i>Propagation</i>	<i>Application</i>
VLF (very low frequency)	3–30 kHz	Ground	Long-range radio navigation
LF (low frequency)	30–300 kHz	Ground	Radio beacons and navigational locators
MF (middle frequency)	300 kHz–3 MHz	Sky	AM radio
HF (high frequency)	3–30 MHz	Sky	Citizens band (CB), ship/aircraft communication
VHF (very high frequency)	30–300 MHz	Sky and line-of-sight	VHF TV, FM radio
UHF (ultrahigh frequency)	300 MHz–3 GHz	Line-of-sight	UHF TV, cellular phones, paging, satellite
SHF (superhigh frequency)	3–30 GHz	Line-of-sight	Satellite communication
EHF (extremely high frequency)	30–300 GHz	Line-of-sight	Radar, satellite

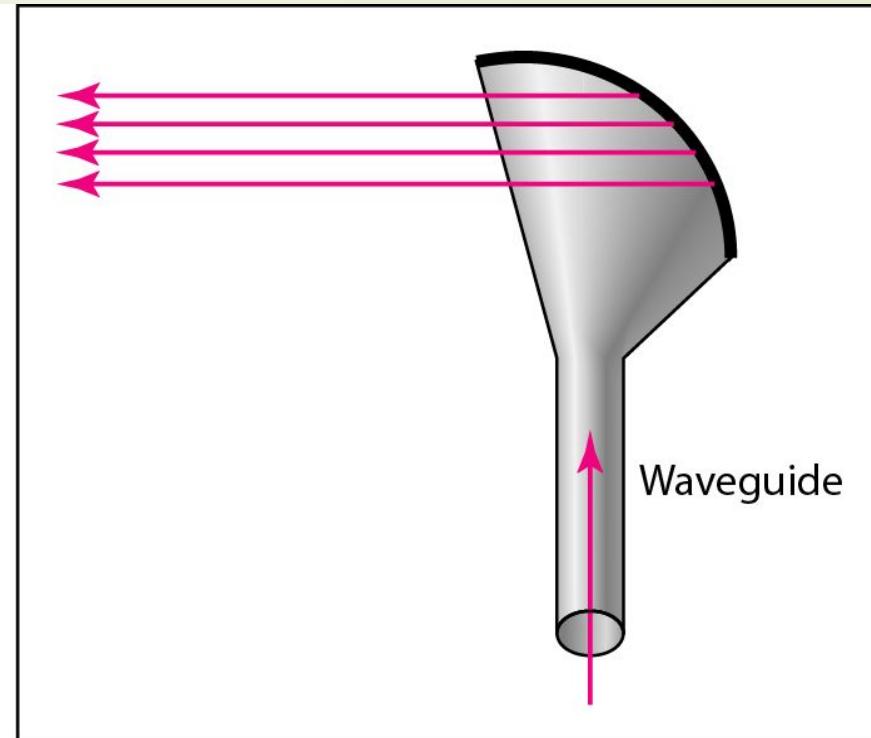
Omnidirectional antenna



Unidirectional antennas



a. Dish antenna



b. Horn antenna



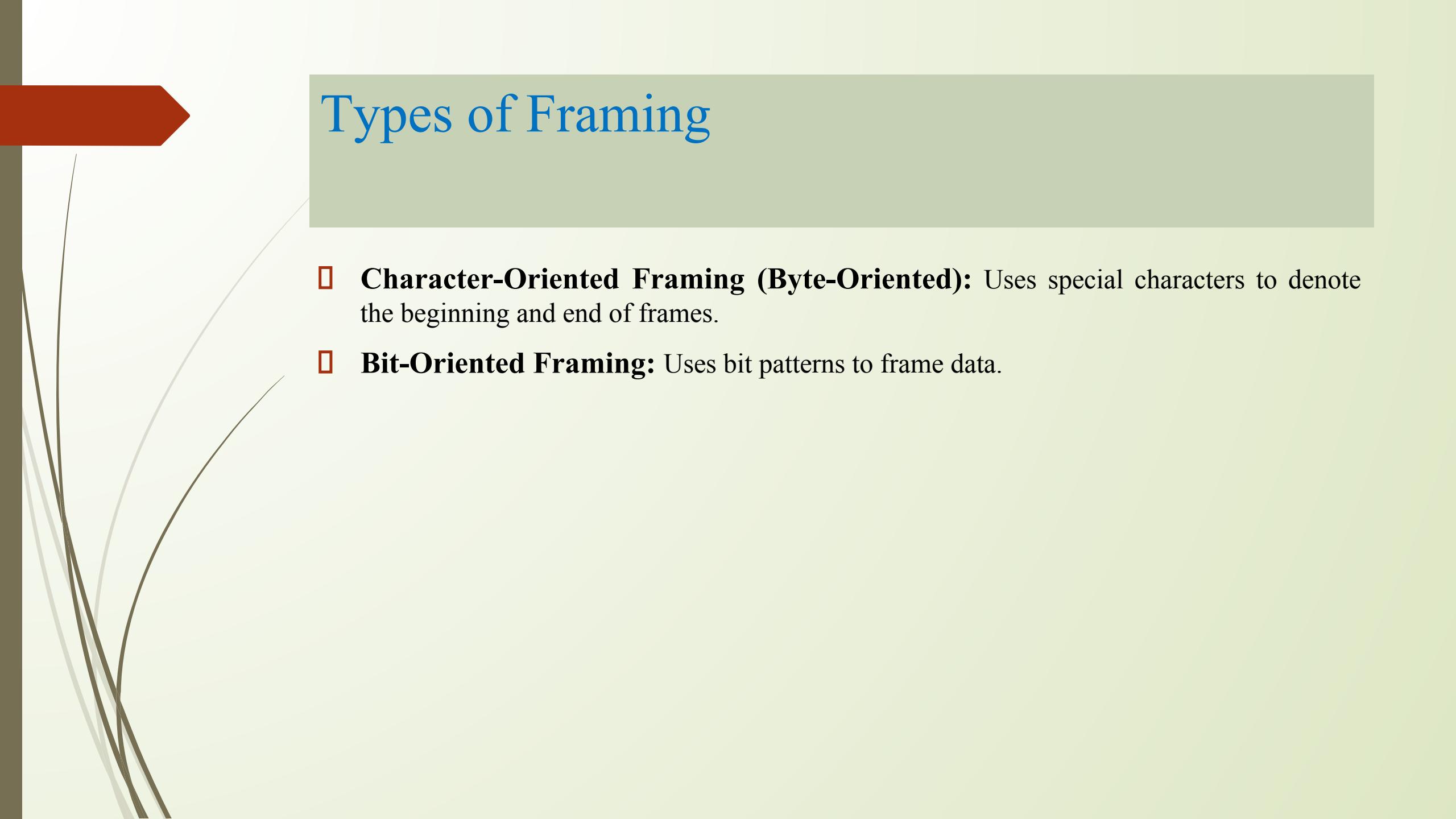
Chapter 4

Data Link Control



FRAMING

- Framing is the process of breaking down data into manageable units called frames for efficient and reliable data transmission over a network.
- Frames contain not only the raw data but also control information like headers, trailers, and checksums to ensure proper delivery.
- The data link layer needs to pack bits into frames, so that each frame is distinguishable from another. Our postal system practices a type of framing. The simple act of inserting a letter into an envelope separates one piece of information from another; the envelope serves as the delimiter.



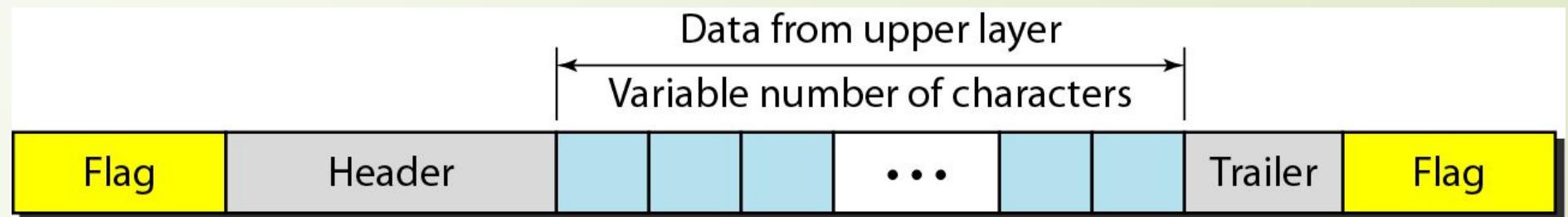
Types of Framing

- **Character-Oriented Framing (Byte-Oriented):** Uses special characters to denote the beginning and end of frames.
- **Bit-Oriented Framing:** Uses bit patterns to frame data.

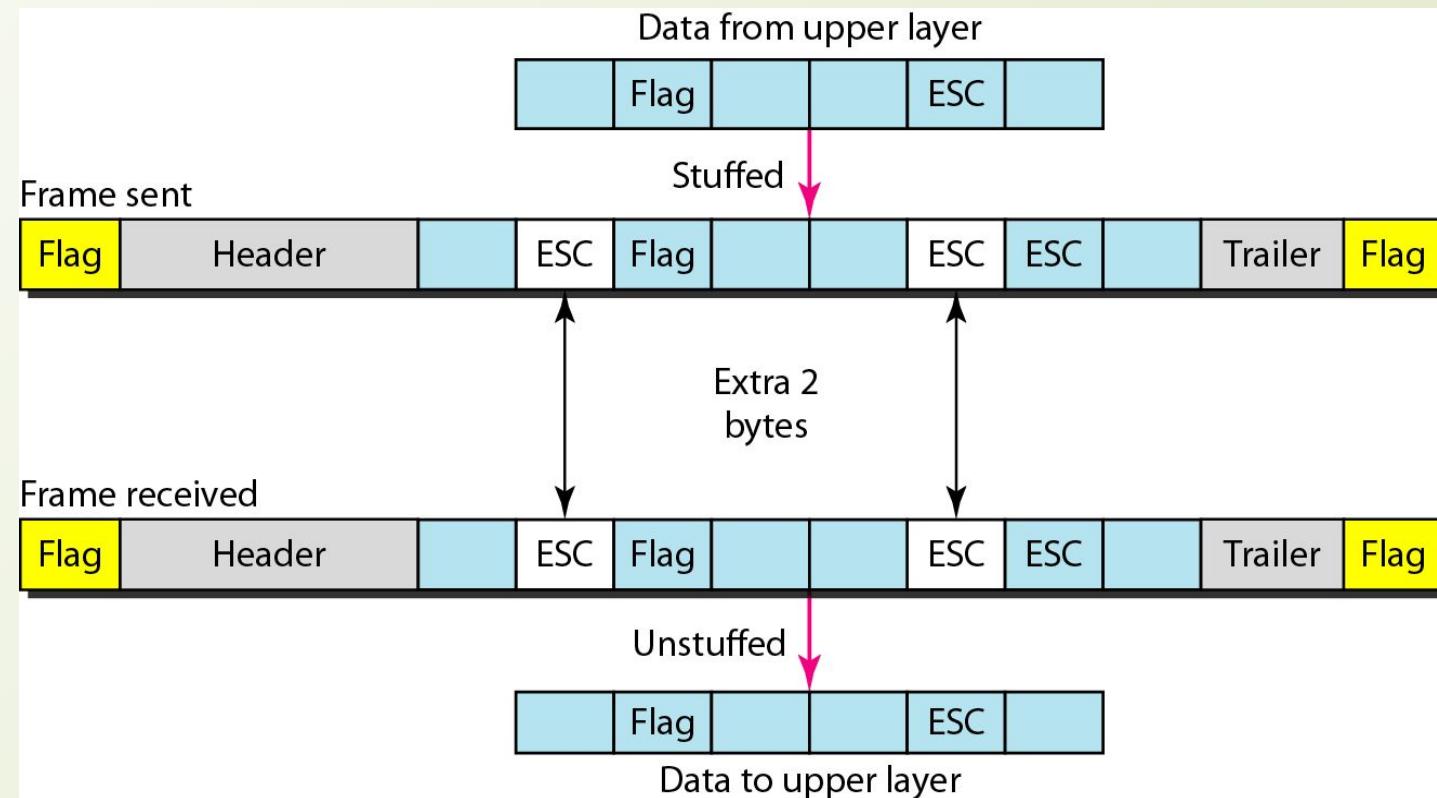
Byte Stuffing

- A technique used in data transmission to differentiate between data bytes and control bytes (such as frame delimiters). Byte stuffing ensures special control characters (e.g., frame delimiters) do not appear in the actual data being transmitted.

A frame in a character-oriented protocol



Byte stuffing and unstuffing



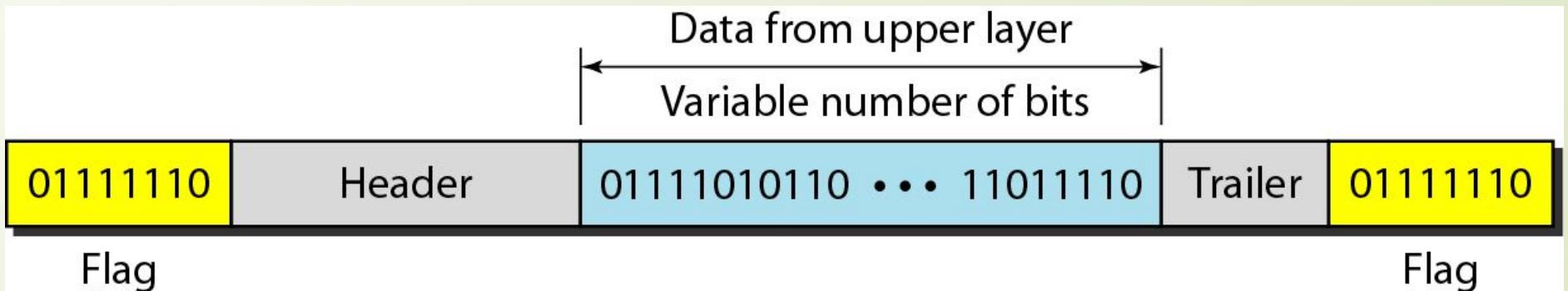


Note

- Byte stuffing is the process of adding 1 extra byte whenever there is a flag or escape character in the text.

A frame in a bit-oriented protocol

Bit stuffing is a technique used in data transmission protocols to ensure that special patterns of bits, which might be mistaken for control information, do not appear in the data being transmitted.

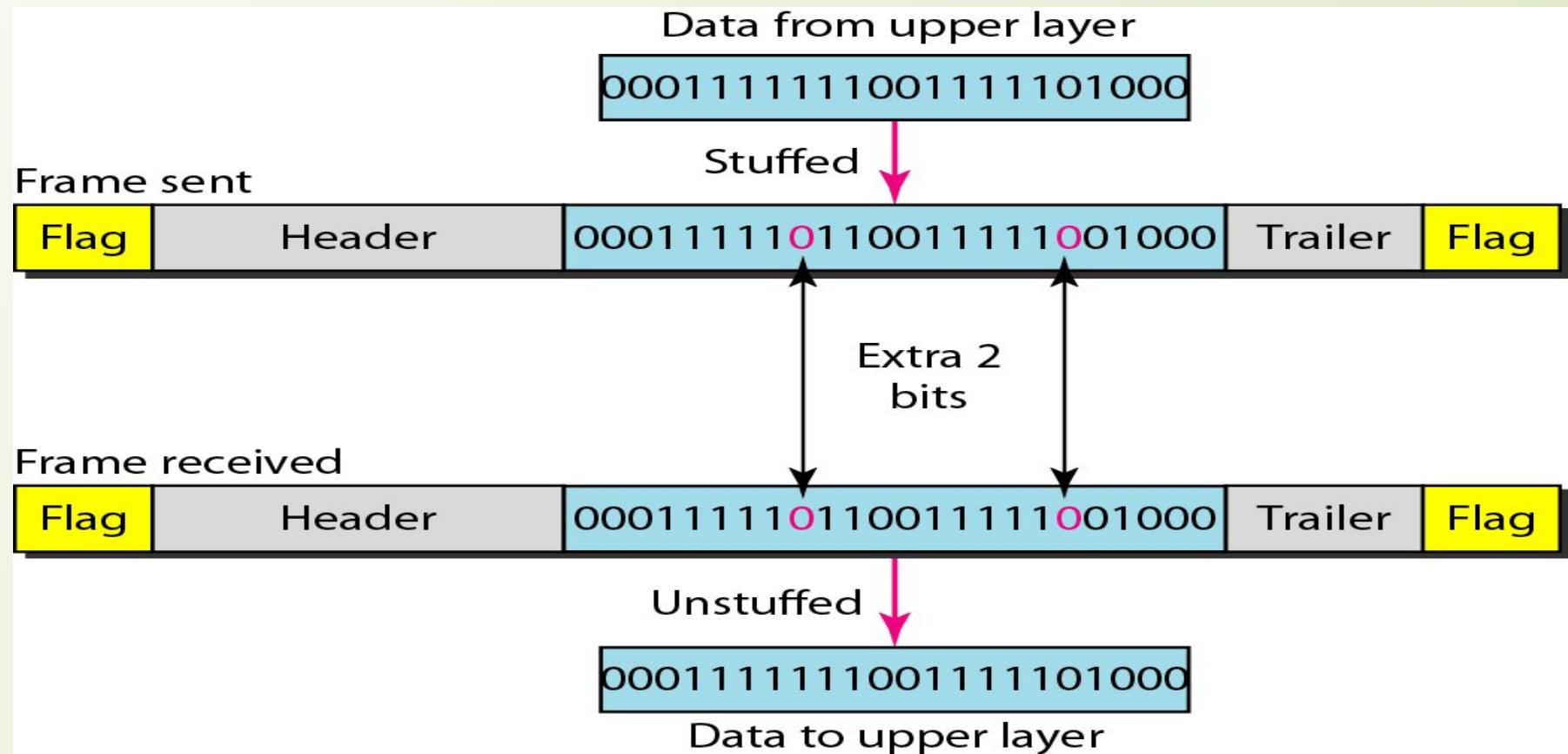




Note

- Bit stuffing is the process of adding one extra 0 whenever five consecutive 1s follow a 0 in the data, so that the receiver does not mistake
- the pattern 0111110 is for a flag.

Bit stuffing and unstuffing



Additional Use of bit stuffing

□ Clock Synchronization:

- In synchronous communication systems, long sequences of the same bit (e.g., all 0s or all 1s) can cause issues with clock recovery.
- Bit stuffing breaks up long sequences, providing transitions that help the receiver maintain synchronization with the transmitter's clock.

Point to remember

- Both long sequences of 0s and 1s can cause issues because they lack transitions (edges), leading to potential synchronization problems.
- Why Bit Stuffing Focuses only on Consecutive 1s?
 - Many bit-oriented protocols, like HDLC, are designed with specific frame delimiters that contain sequences of 1s (e.g., 01111110 in HDLC). To ensure these delimiters do not appear in the data, bit stuffing focuses on sequences of 1s.
 - Inserting a 0 after five consecutive 1s prevents these sequences from being mistaken for frame delimiters.
- In cases where long sequences of 0s might be an issue for clock recovery, different strategies such as Physical Layer Encoding (Manchester Encoding and 4B/5B Encoding) and Scrambling is used.



FLOW AND ERROR CONTROL

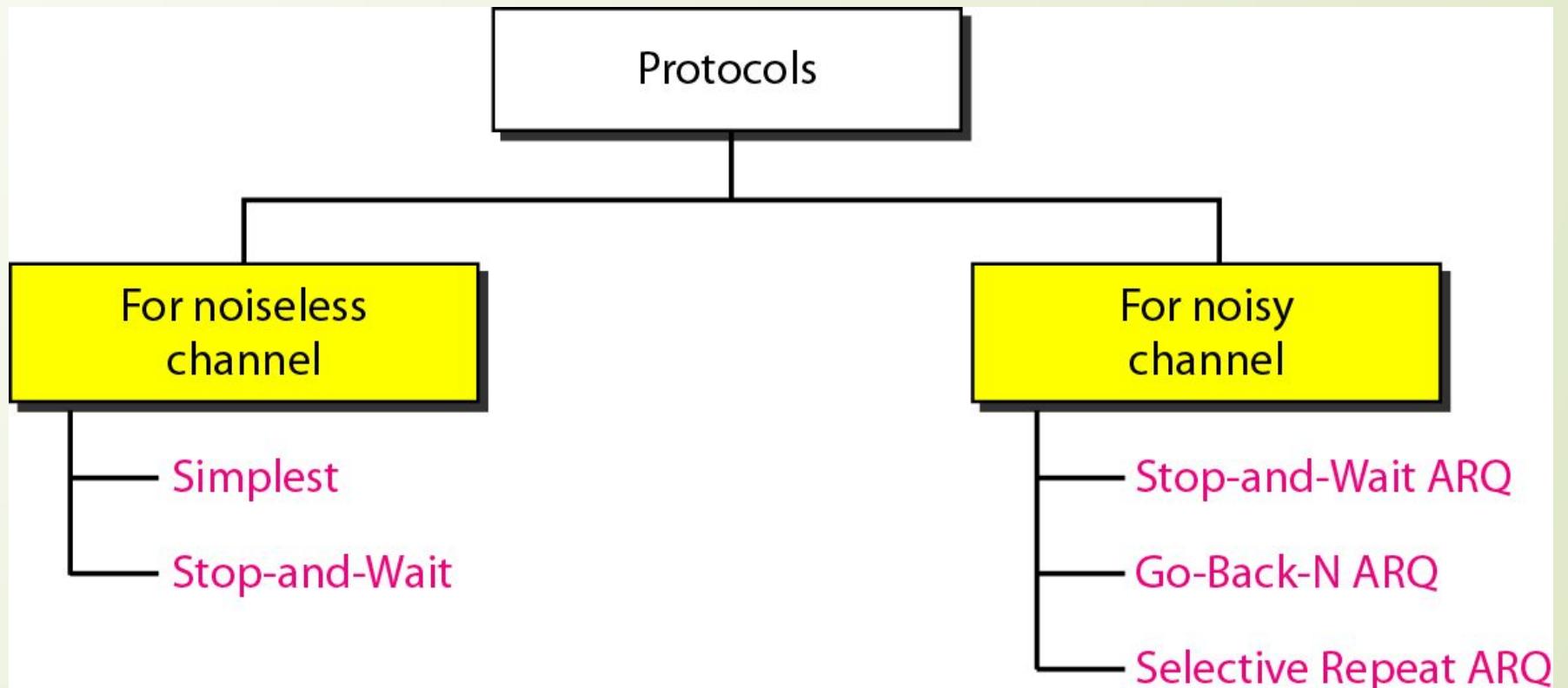
- The most important responsibilities of the data link layer are flow control and error control. Collectively, these functions are known as data link control.
- Flow control refers to a set of procedures used to restrict the amount of data that the sender can send before waiting for acknowledgment.
- Error control in the data link layer is based on automatic repeat request, which is the retransmission of data.



PROTOCOLS

- Now let us see how the data link layer can combine framing, flow control, and error control to achieve the delivery of data from one node to another. The protocols are normally implemented in software by using one of the common programming languages. To make our discussions language-free, we have written in pseudocode a version of each protocol that concentrates mostly on the procedure instead of delving into the details of language rules.

Taxonomy of protocols discussed in this chapter





NOISELESS CHANNELS

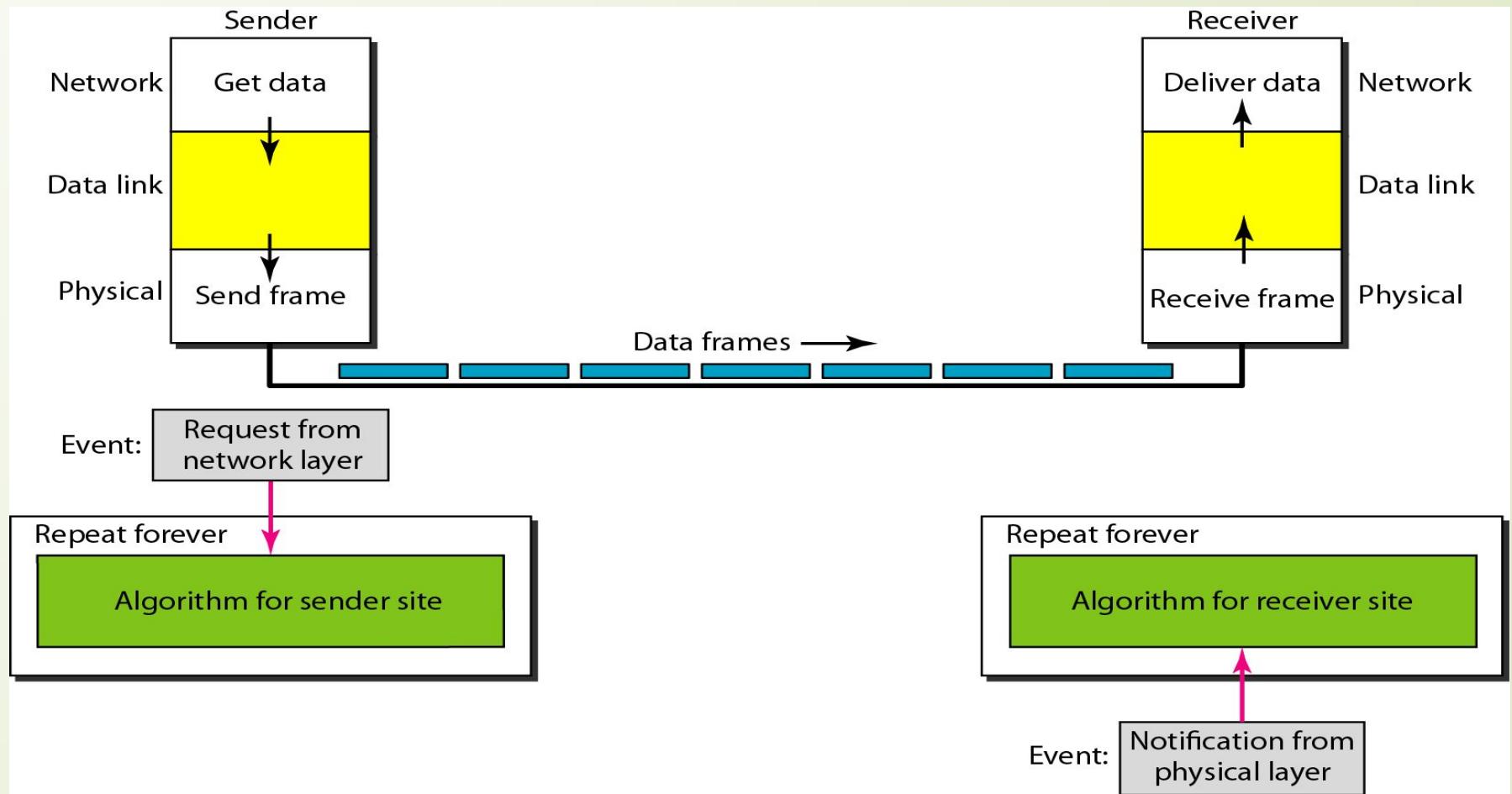
- Let us first assume we have an ideal channel in which no frames are lost, duplicated, or corrupted. We introduce two protocols for this type of channel.

Topics discussed in this section:

Simplest Protocol

Stop-and-Wait Protocol

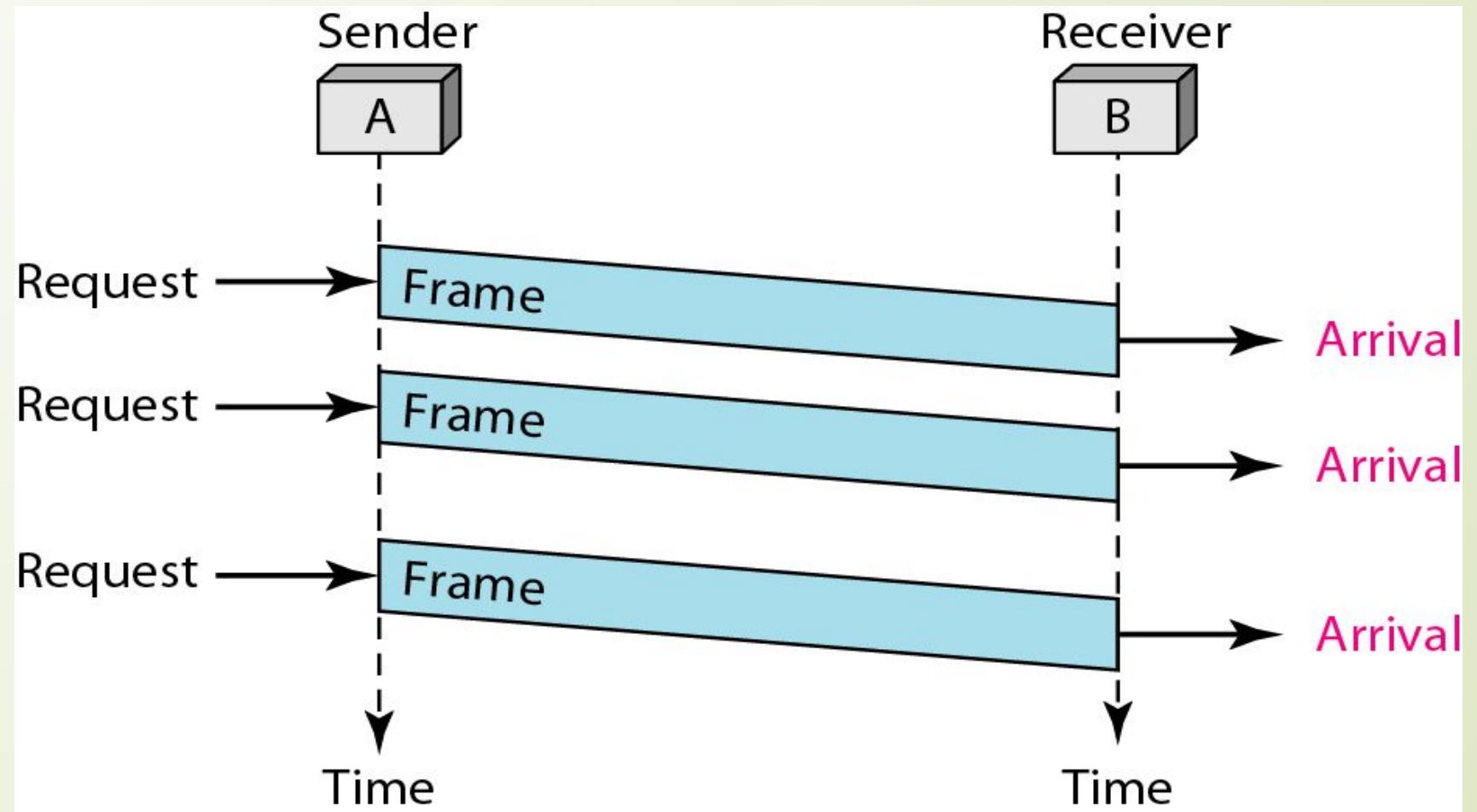
The design of the simplest protocol with no flow or error control



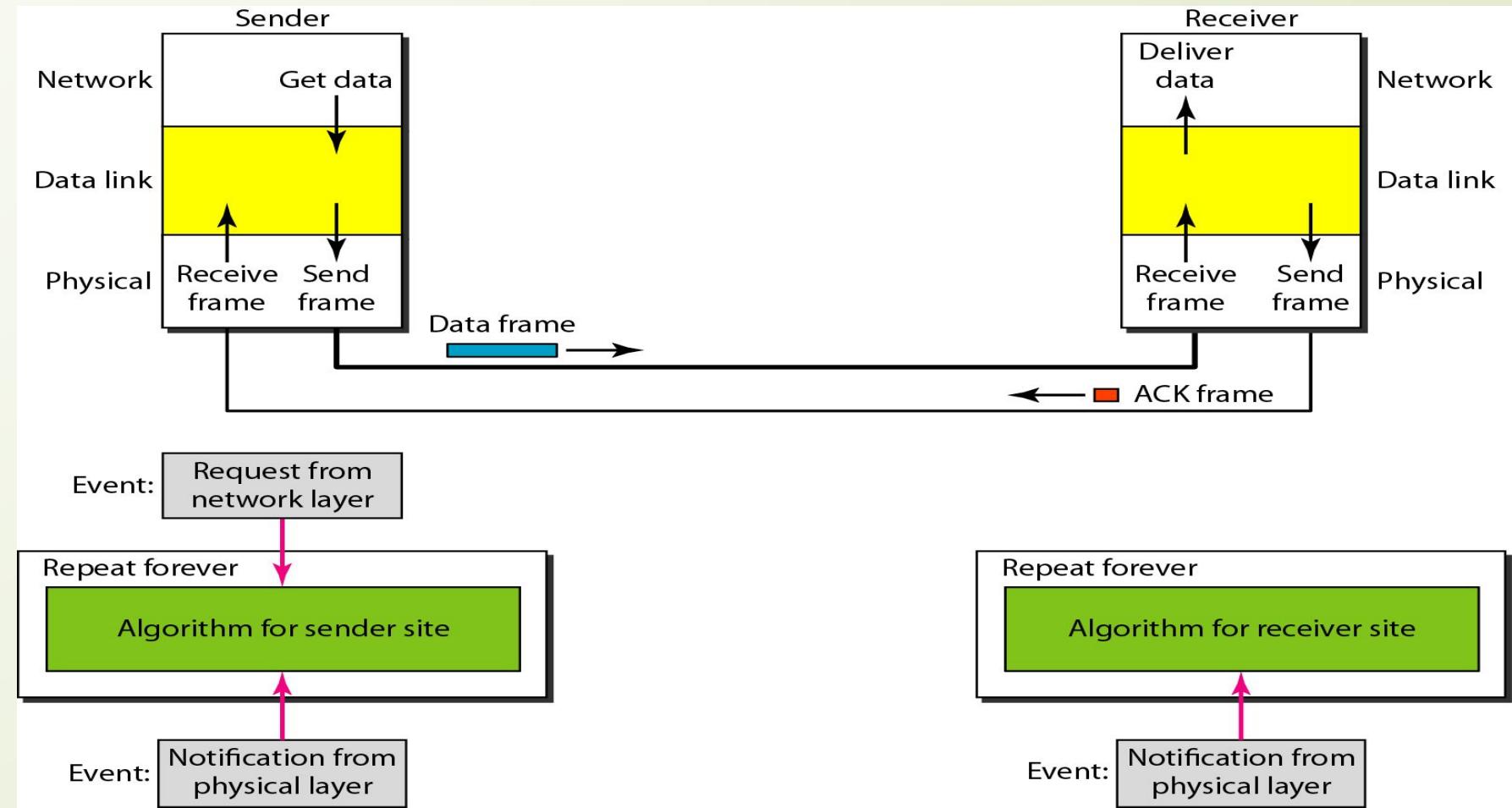
Example

- Figure 4.1 shows an example of communication using this protocol. It is very simple. The sender sends a sequence of frames without even thinking about the receiver. To send three frames, three events occur at the sender site and three events at the receiver site. Note that the data frames are shown by tilted boxes; the height of the box defines the transmission time difference between the first bit and the last bit in the frame.

Figure 4.1 Flow diagram for Example



Design of Stop-and-Wait Protocol

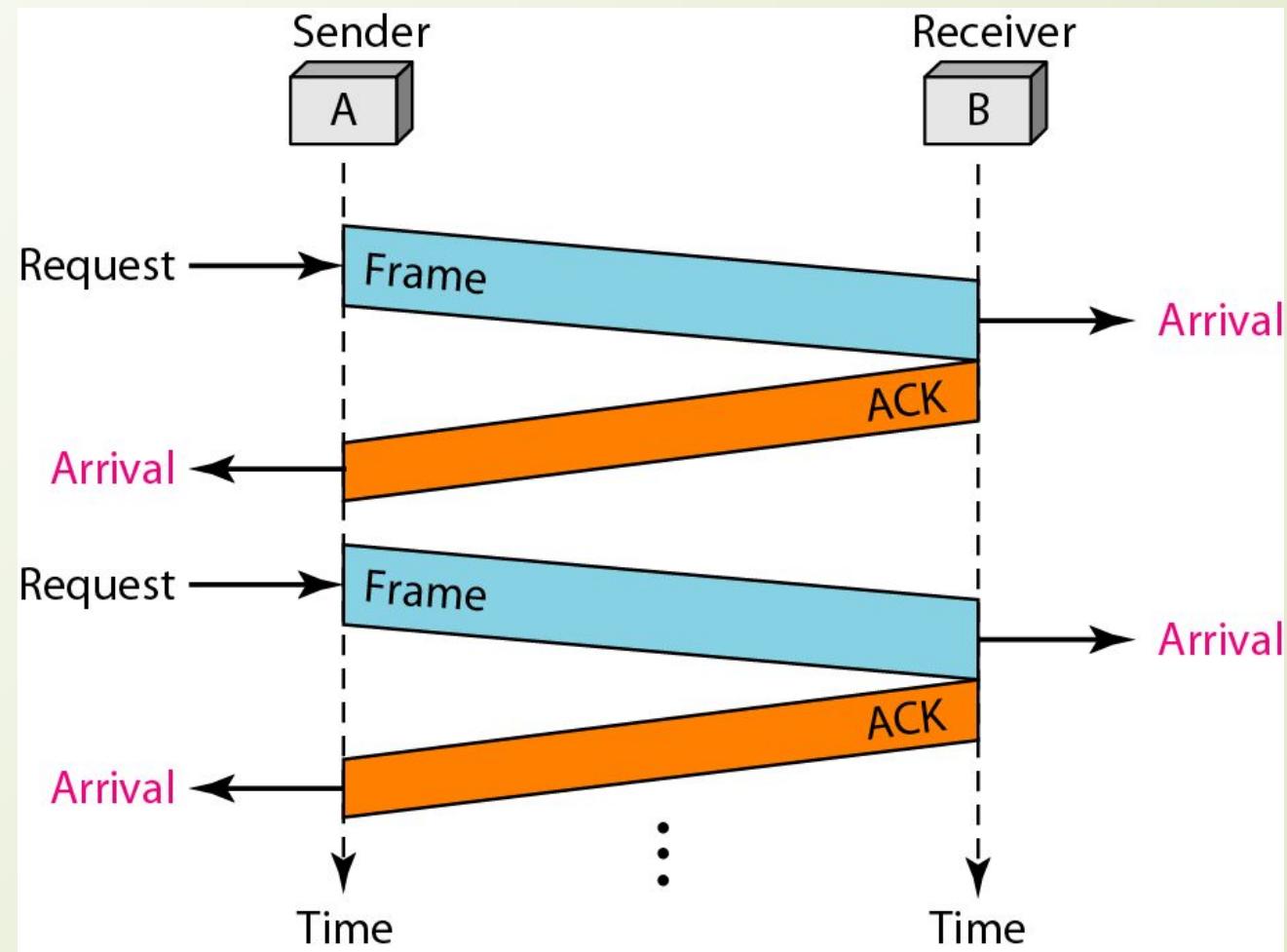




Example

- Figure 4.2 shows an example of communication using this protocol. It is still very simple. The sender sends one frame and waits for feedback from the receiver. When the ACK arrives, the sender sends the next frame. Note that sending two frames in the protocol involves the sender in four events and the receiver in two events.

Figure 4.2 Flow diagram for Example





NOISY CHANNELS

- Although the Stop-and-Wait Protocol gives us an idea of how to add flow control to its predecessor, noiseless channels are nonexistent. We discuss three protocols in this section that use error control.

Topics discussed in this section:

- Stop-and-Wait Automatic Repeat Request
- Go-Back-N Automatic Repeat Request
- Selective Repeat Automatic Repeat Request



Note

- Error correction in Stop-and-Wait ARQ is done by keeping a copy of the sent frame and retransmitting of the frame when the timer expires.



Note

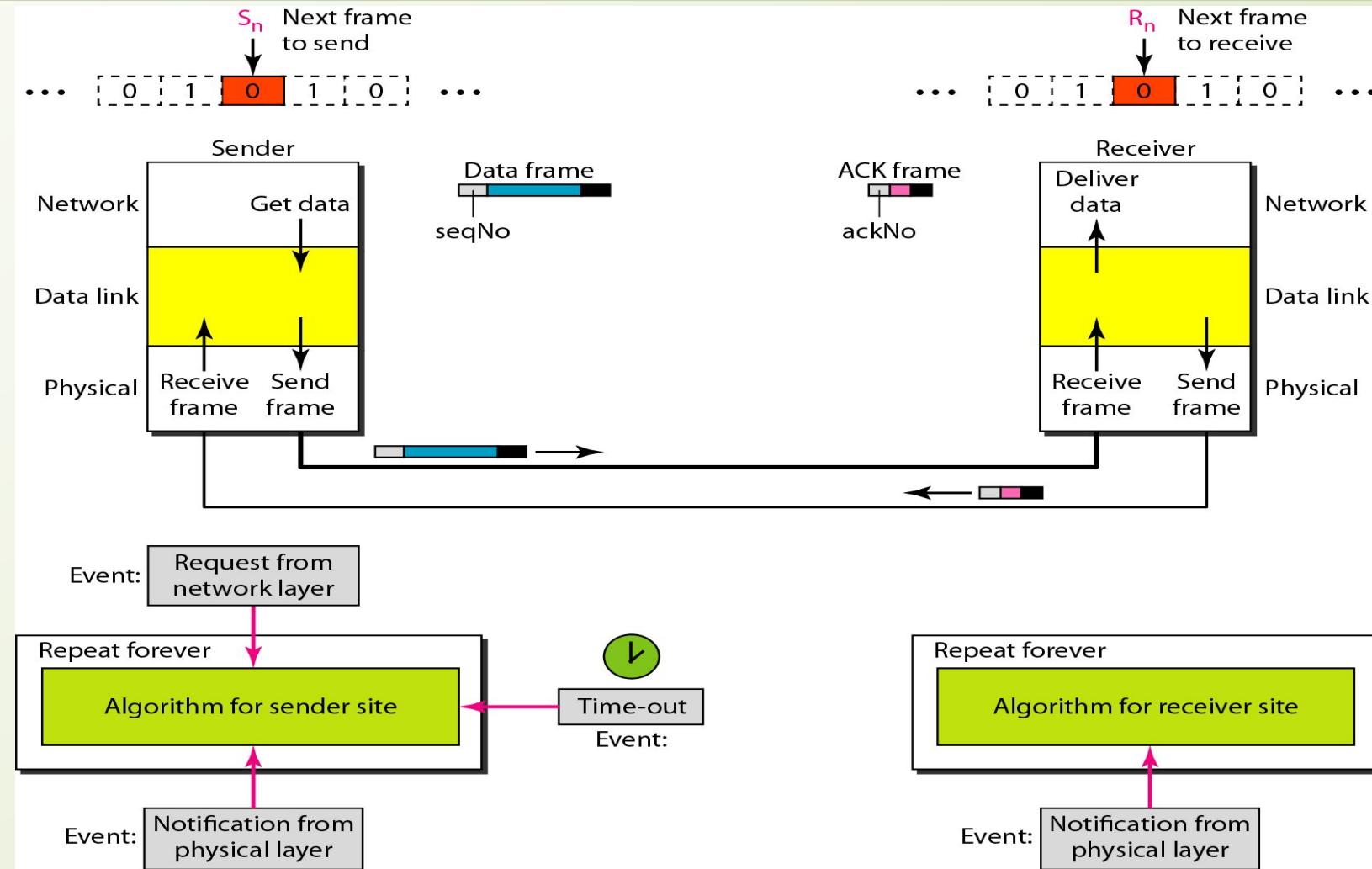
- In Stop-and-Wait ARQ, we use sequence numbers to number the frames.
- The sequence numbers are based on modulo-2 arithmetic.



Note

- In Stop-and-Wait ARQ, the acknowledgment number always announces in modulo-2 arithmetic the sequence number of the next frame expected.

Figure 4.3 Design of the Stop-and-Wait ARQ Protocol

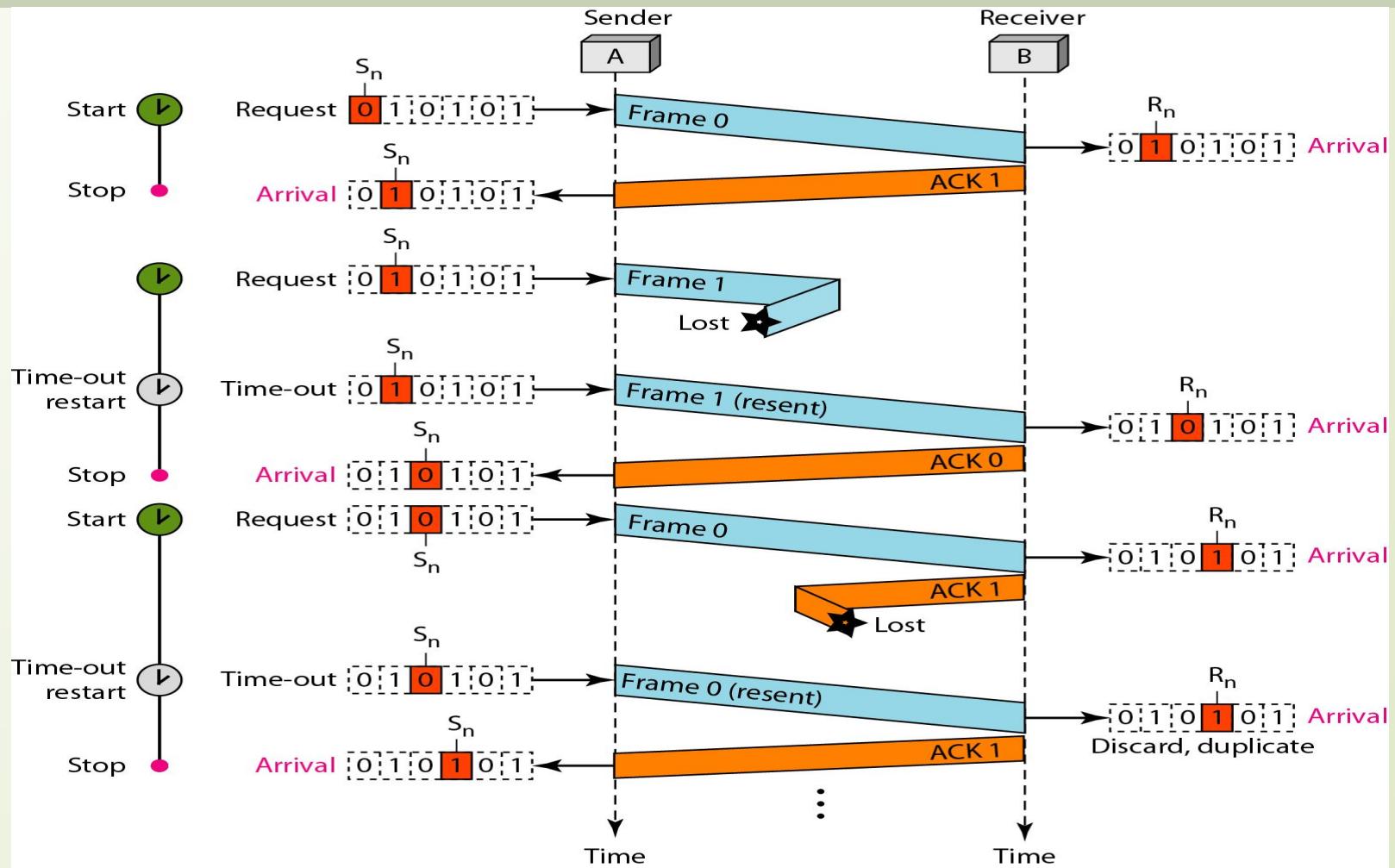


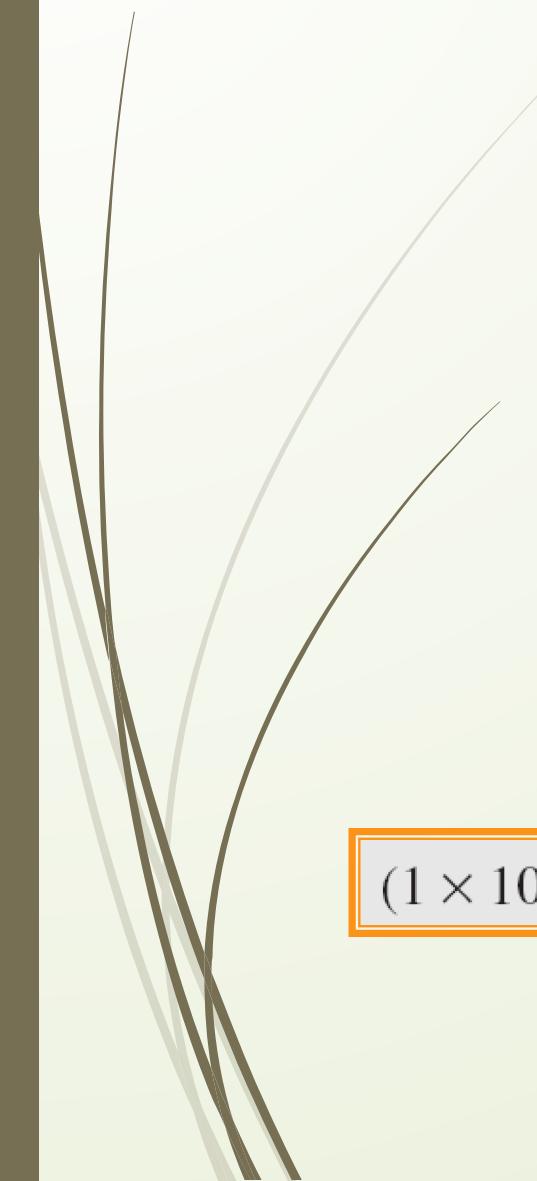


Example

- Figure 4.3 shows an example of **Stop-and-Wait ARQ**. Frame 0 is sent and acknowledged. Frame 1 is lost and resent after the time-out. The resent frame 1 is acknowledged and the timer stops. Frame 0 is sent and acknowledged, but the acknowledgment is lost. The sender has no idea if the frame or the acknowledgment is lost, so after the time-out, it resends frame 0, which is acknowledged.

Figure 4.3 Flow diagram

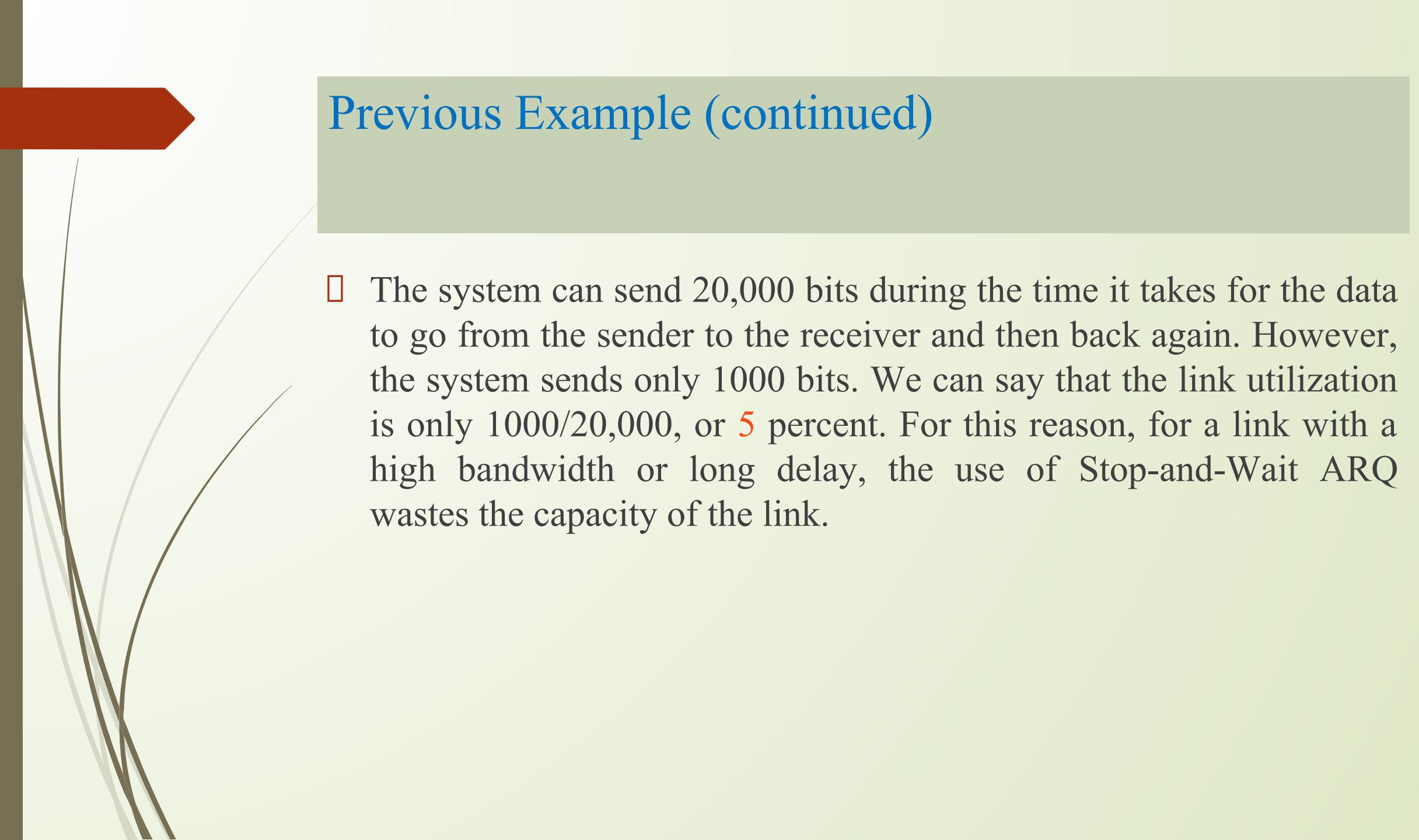




Example

- Assume that, in a Stop-and-Wait ARQ system, the bandwidth of the line is 1 Mbps, and 1 bit takes 20 ms to make a round trip. What is the bandwidth-delay product? If the system data frames are 1000 bits in length, what is the utilization percentage of the link?
- Solution
- The bandwidth-delay product is

$$(1 \times 10^6) \times (20 \times 10^{-3}) = 20,000 \text{ bits}$$



Previous Example (continued)

- The system can send 20,000 bits during the time it takes for the data to go from the sender to the receiver and then back again. However, the system sends only 1000 bits. We can say that the link utilization is only $1000/20,000$, or 5 percent. For this reason, for a link with a high bandwidth or long delay, the use of Stop-and-Wait ARQ wastes the capacity of the link.



Example

- What is the utilization percentage of the link in Previous Example if we have a protocol that can send up to 15 frames before stopping and worrying about the acknowledgments?
- Solution
- The bandwidth-delay product is still 20,000 bits. The system can send up to 15 frames or 15,000 bits during a round trip. This means the utilization is $15,000/20,000$, or 75 percent. Of course, if there are damaged frames, the utilization percentage is much less because frames have to be resent.

Go-back-N ARQ Protocol

- It is a sliding window protocol for reliable data transmission.
- It increases efficiency by allowing multiple frames to be sent before needing an acknowledgment.
- It is an improvement over Stop-and-Wait ARQ.

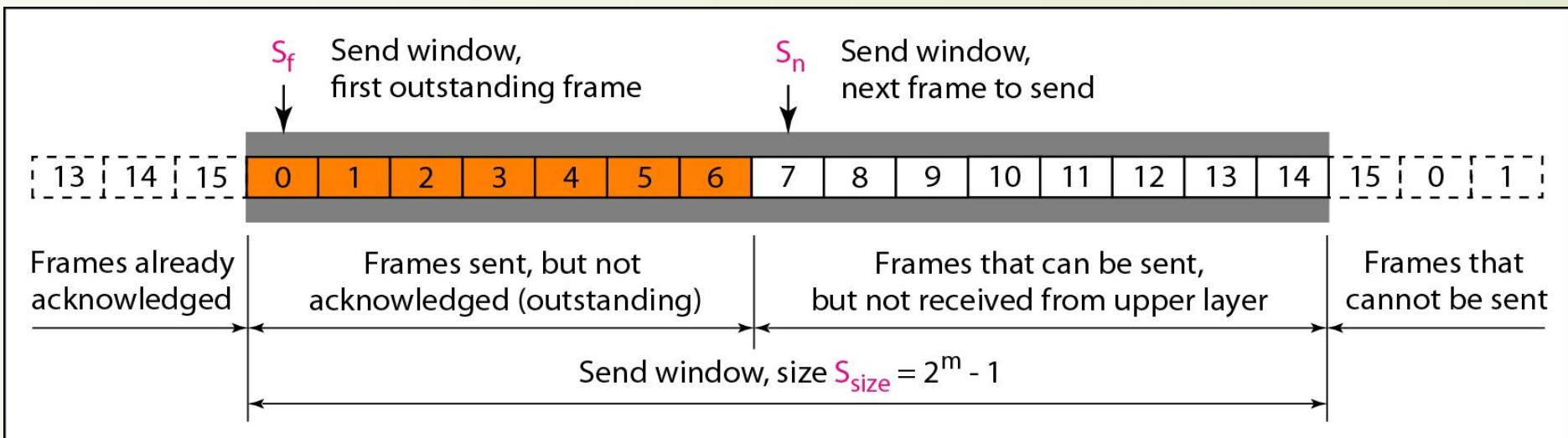


Note

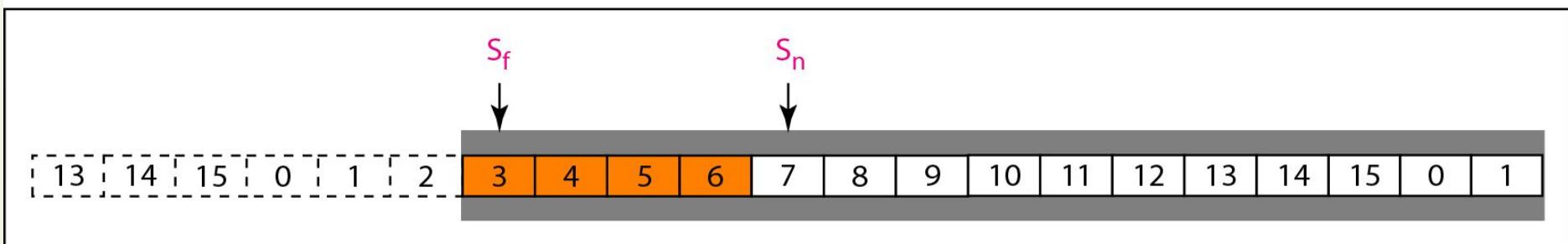
- In the Go-Back-N Protocol, the sequence numbers are modulo 2^m ,
- where m is the size of the sequence number field in bits.

The term **modulo 2^m** refers to a mathematical operation in which a number is divided by 2^m (where m is an integer), and the remainder of this division is the result of the operation.

Send window for Go-Back-N ARQ



a. Send window before sliding



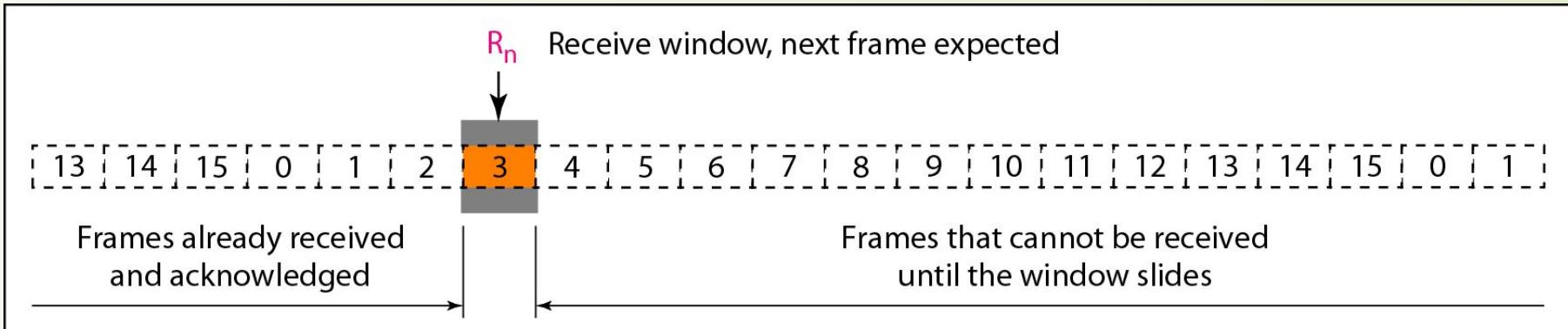
b. Send window after sliding



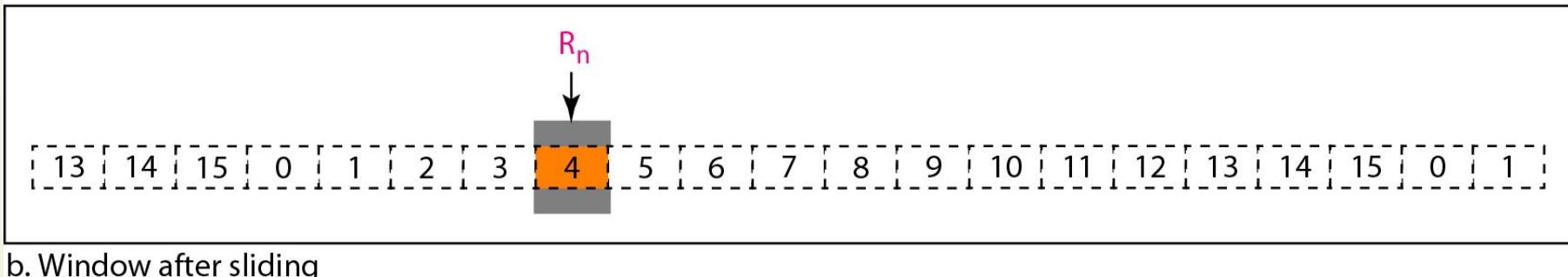
Note

- The send window is an abstract concept defining an imaginary box of size $2^m - 1$ with three variables: S_f , S_n , and S_{size} .
- The send window can slide one or more slots when a valid acknowledgment arrives.

Receive window for Go-Back-N ARQ



a. Receive window



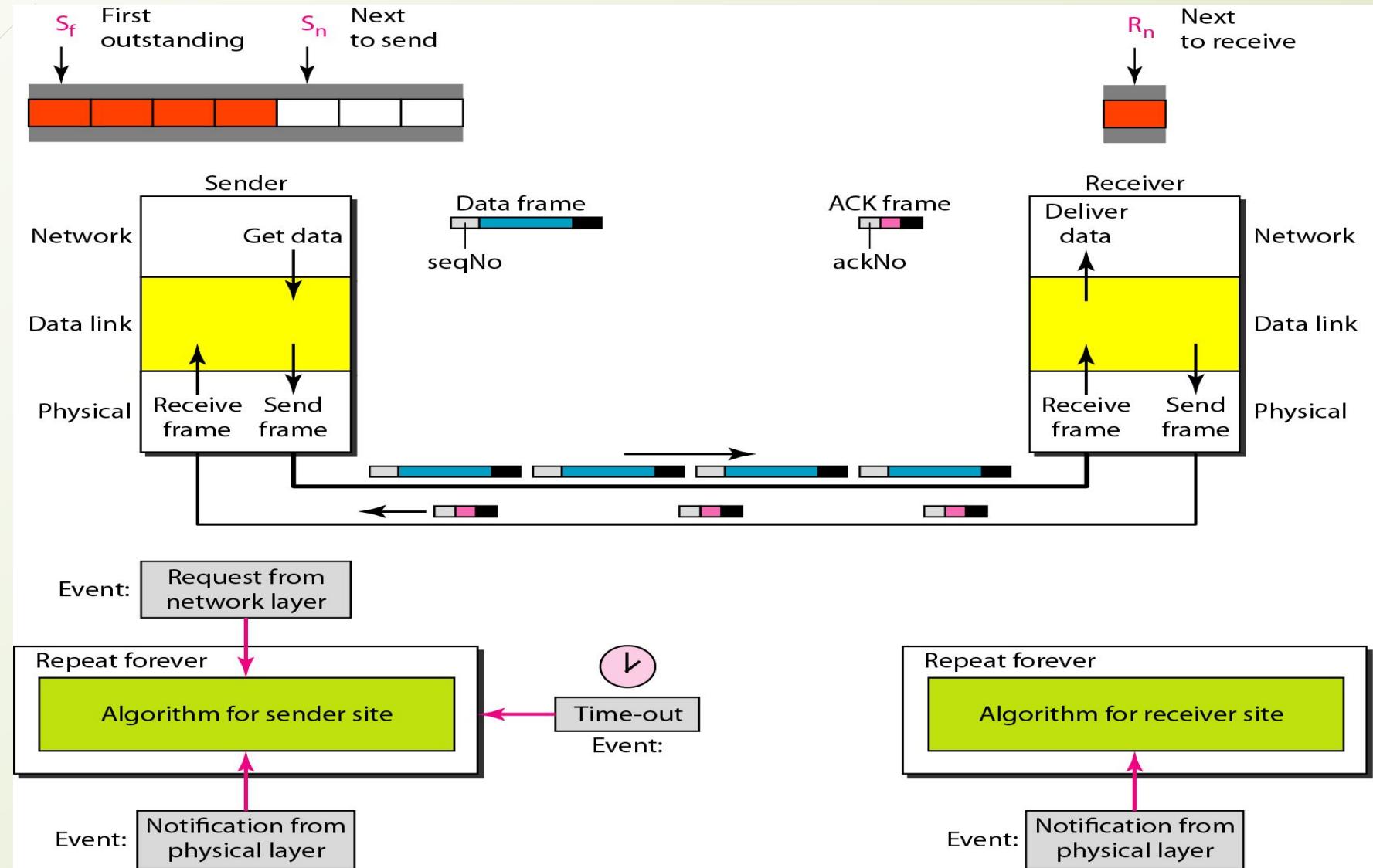
b. Window after sliding



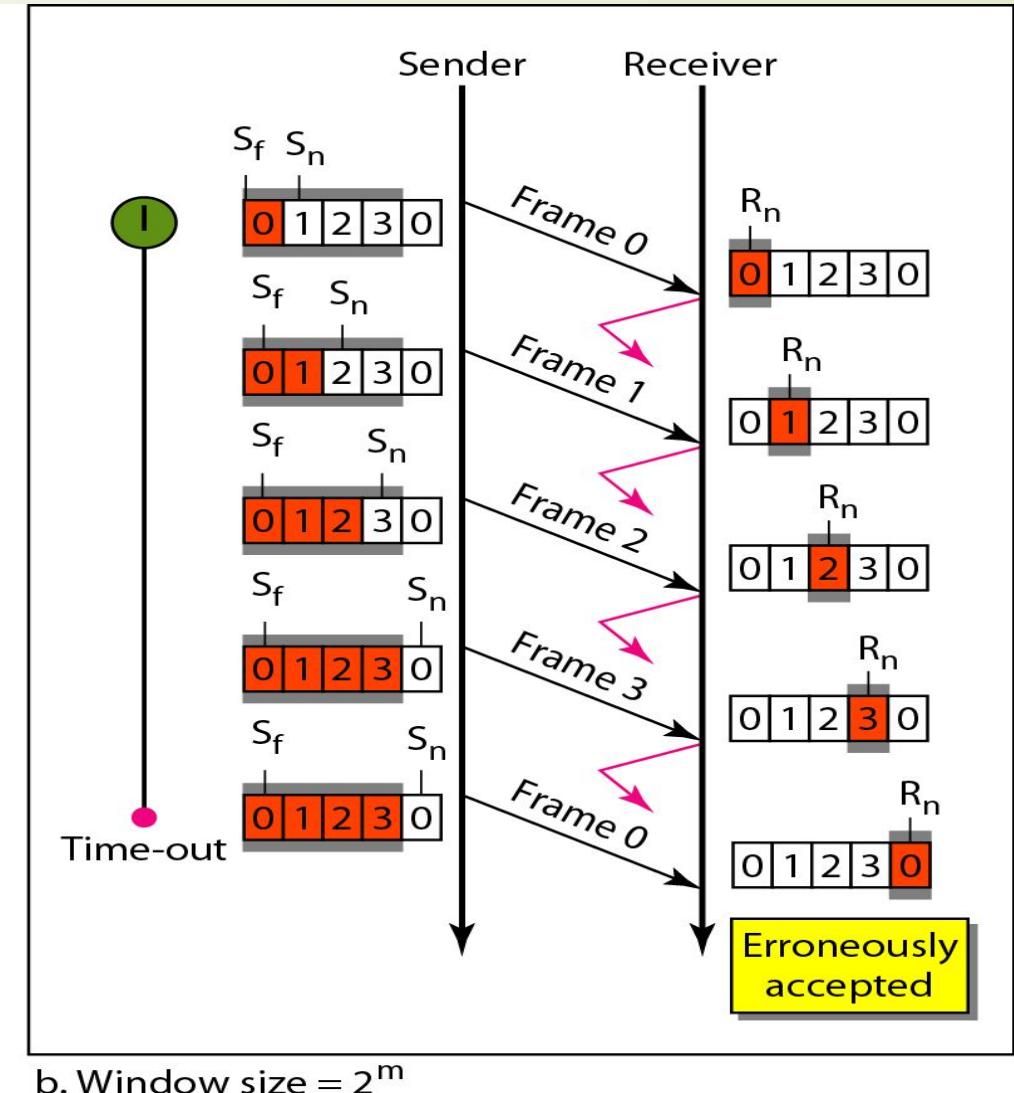
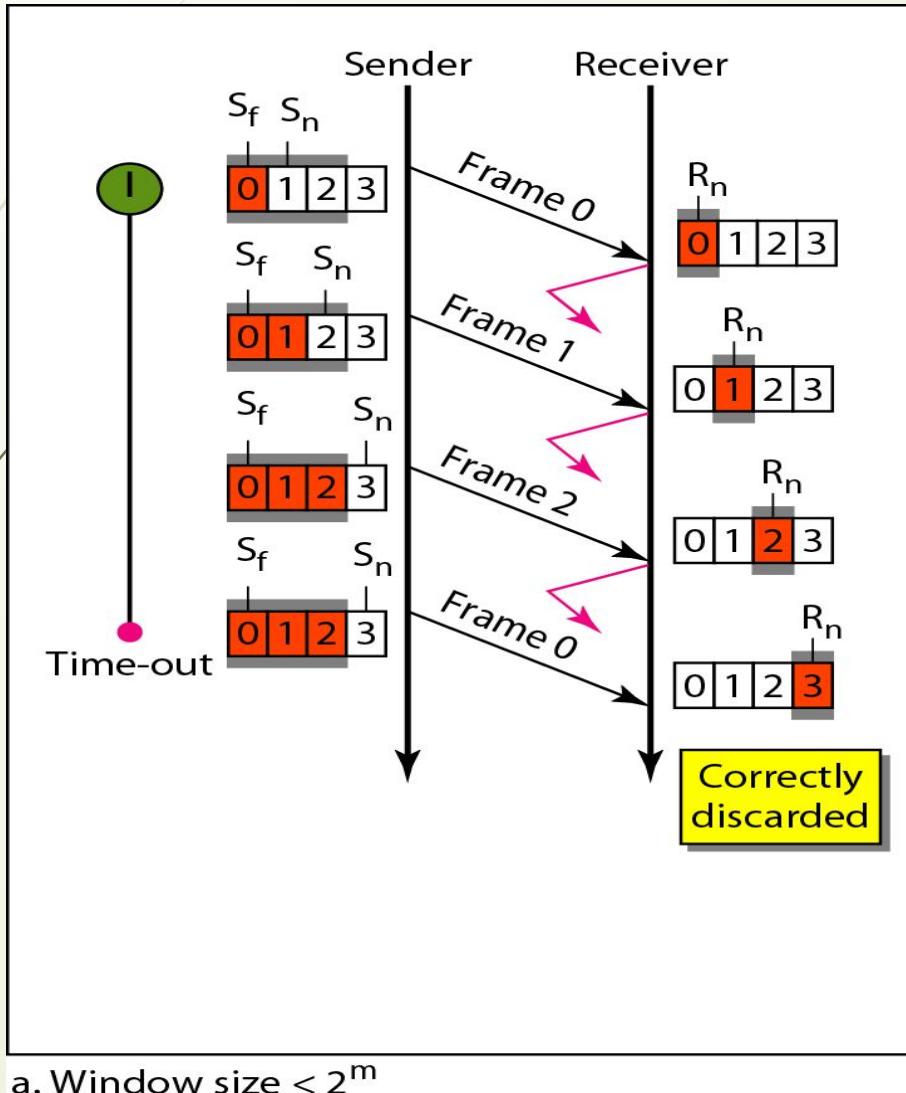
Note

- The receive window is an abstract concept defining an imaginary box of size 1 with one single variable R_n .
The window slides.
- when a correct frame has arrived; sliding occurs one slot at a time.

Design of Go-Back-N ARQ



Why Window size should be less than 2^m for Go-Back-N ARQ





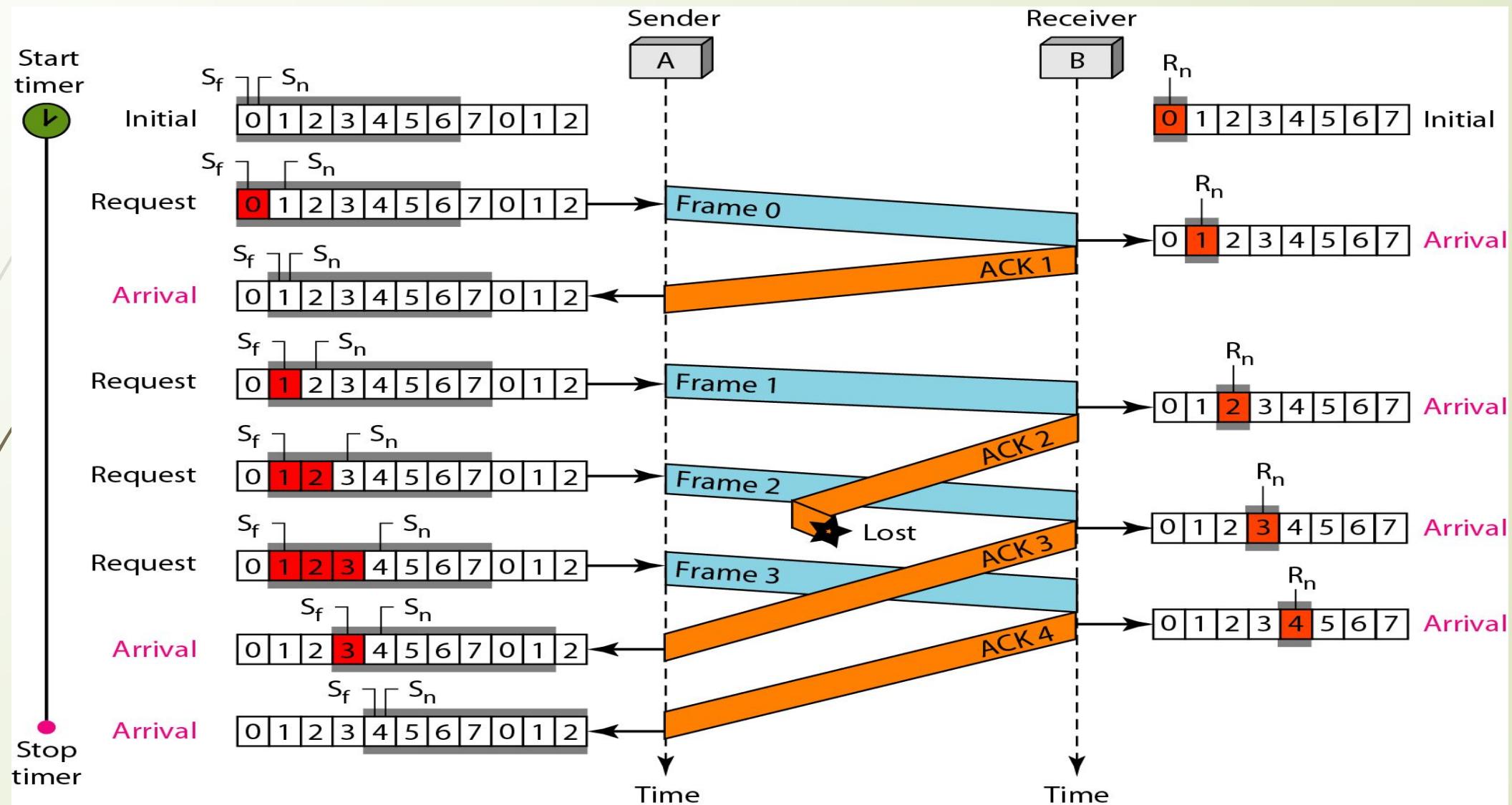
Note

- In Go-Back-N ARQ, the size of the send window must be less than 2^m ;
- The size of the receiver window is always 1.

Example

- Figure 4.4 shows an example of Go-Back-N. This is an example of a case where the forward channel is reliable, but the reverse is not. No data frames are lost, but some ACKs are delayed and one is lost. The example also shows how cumulative acknowledgments can help if acknowledgments are delayed or lost. After initialization, there are seven sender events. Request events are triggered by data from the network layer; arrival events are triggered by acknowledgments from the physical layer. There is no time-out event here because all outstanding frames are acknowledged before the timer expires. Note that although ACK 2 is lost, ACK 3 serves as both ACK 2 and ACK 3.

Figure 4.4 Flow diagram





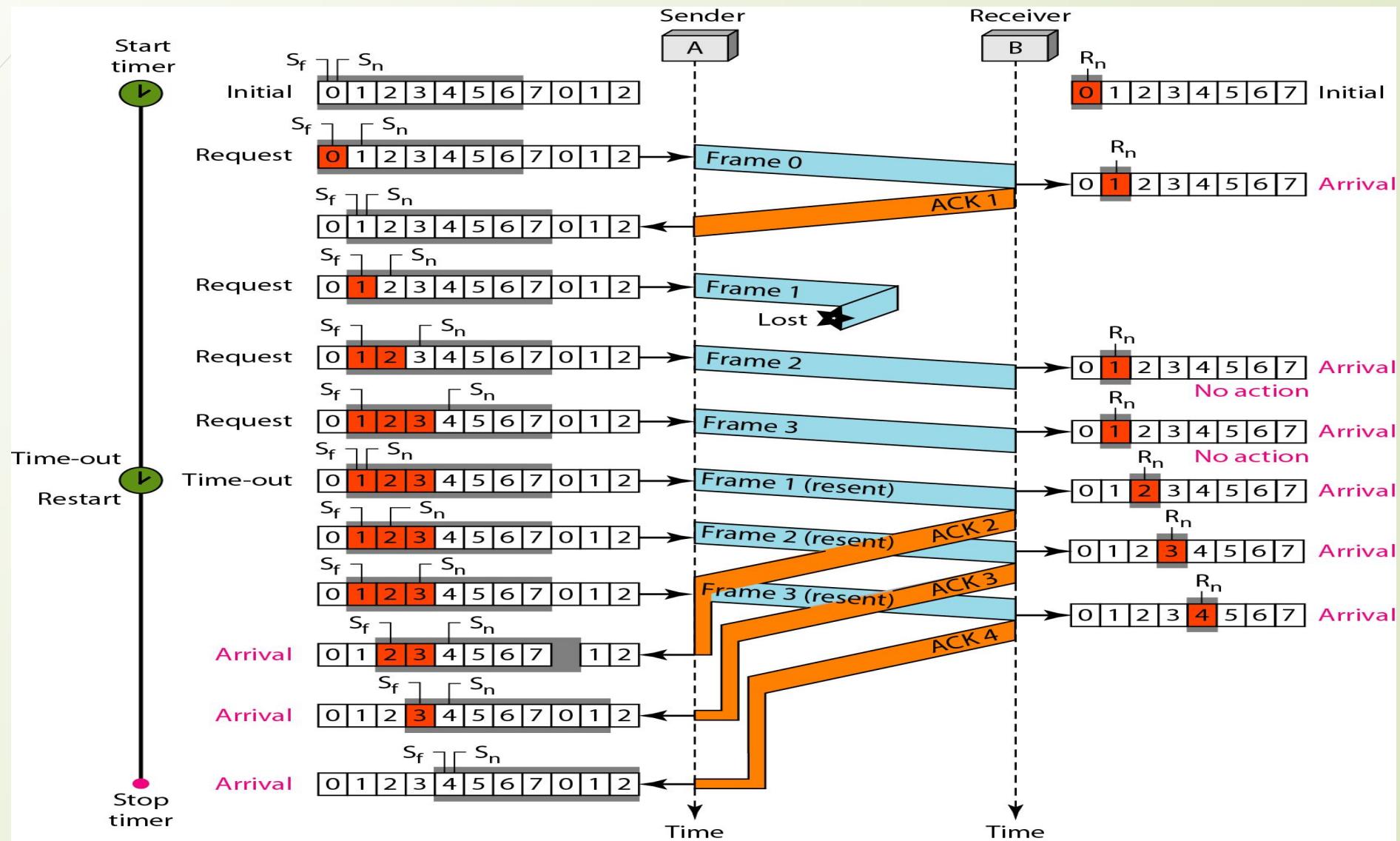
Example

- Figure 4.5 shows what happens when a frame is lost. Frames 0, 1, 2, and 3 are sent. However, frame 1 is lost. The receiver receives frames 2 and 3, but they are discarded because they are received out of order. The sender receives no acknowledgment about frames 1, 2, or 3. Its timer finally expires. The sender sends all outstanding frames (1, 2, and 3) because it does not know what is wrong. Note that the resending of frames 1, 2, and 3 is the response to one single event. When the sender is responding to this event, it cannot accept the triggering of other events. This means that when ACK 2 arrives, the sender is still busy with sending frame 3.

PREVIOUS Example (continued)

- The physical layer must wait until this event is completed and the data link layer goes back to its sleeping state. We have shown a vertical line to indicate the delay. It is the same story with ACK 3; but when ACK 3 arrives, the sender is busy responding to ACK 2. It happens again when ACK 4 arrives. Note that before the second timer expires, all outstanding frames have been sent and the timer is stopped.

Figure 4.5 Flow diagram





Note

- Stop-and-Wait ARQ is a special case of Go-Back-N ARQ in which the size of the send window is 1.

Drawbacks of Go-Back-N Protocol

- **Inefficient Use of Bandwidth:** In Go-Back-N, if a single frame is lost or corrupted, all subsequent frames must be retransmitted, even if they were received correctly. This can lead to a significant waste of bandwidth, especially in scenarios where packet loss is infrequent.

This retransmission of multiple frames increases the overhead and reduces the efficiency of the protocol, particularly in networks with high bandwidth and low error rates.

- **Increased Latency: Explanation:** Due to the need to retransmit multiple frames following a single error, Go-Back-N can introduce additional delay. The sender must wait for the acknowledgment of the retransmitted frames, which can increase the round-trip time.

This additional delay is particularly problematic in high-latency networks, where the time taken to resend multiple frames can further reduce throughput and increase the time required for data transmission.

- **Unnecessary Retransmissions:** In Go-Back-N, even if only one frame is lost or corrupted, all frames after the lost frame within the sender's window must be retransmitted, regardless of whether they were received correctly. This leads to unnecessary retransmissions, which can cause network congestion and inefficient use of network resources.

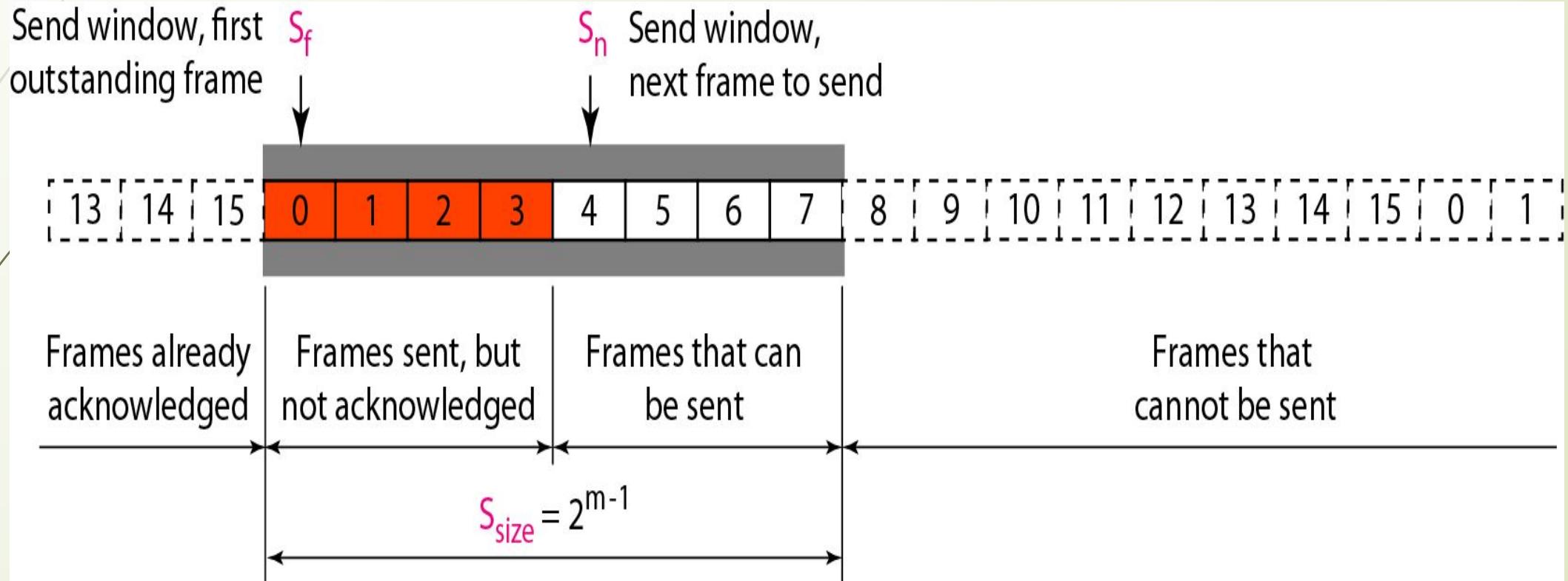
How Selective Repeat ARQ Solves These Problems

Efficient Use of Bandwidth: Selective Repeat ARQ allows the receiver to acknowledge each frame individually and buffers out-of-order frames until the missing frame is retransmitted and received. This means that only the specific lost or corrupted frames are retransmitted, not the entire set of subsequent frames. This selective retransmission leads to a more efficient use of bandwidth, as only the erroneous frames are sent again, reducing unnecessary data transmission.

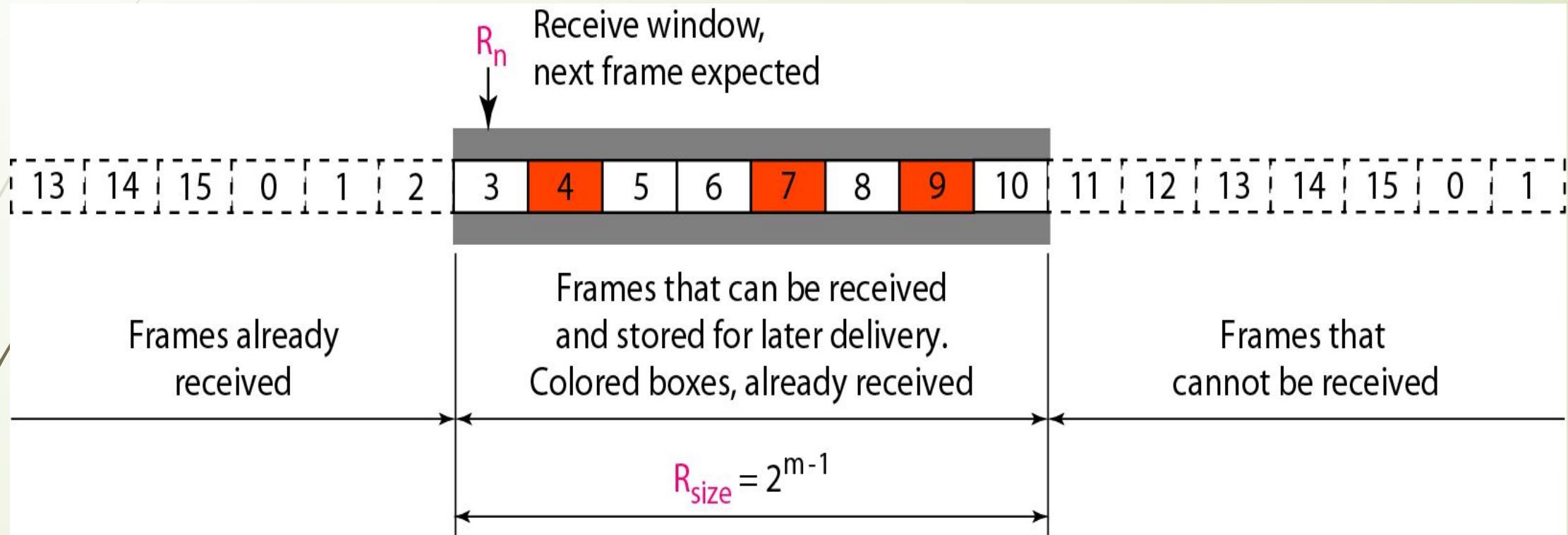
Reduced Latency: Since only the lost or corrupted frames are retransmitted, and correctly received frames are buffered, the protocol avoids the delays associated with Go-Back-N's bulk retransmissions. The receiver can deliver the correctly received frames to the higher layers without waiting for the retransmission of subsequent frames. This reduces the overall delay and improves the throughput, making Selective Repeat ARQ more suitable for networks with high latency or high bandwidth.

Minimized Retransmissions: Selective Repeat ARQ retransmits only the specific frames that were not correctly received, rather than the entire sequence of frames following an error. This minimizes the number of retransmissions, reducing the load on the network and enhancing overall efficiency.

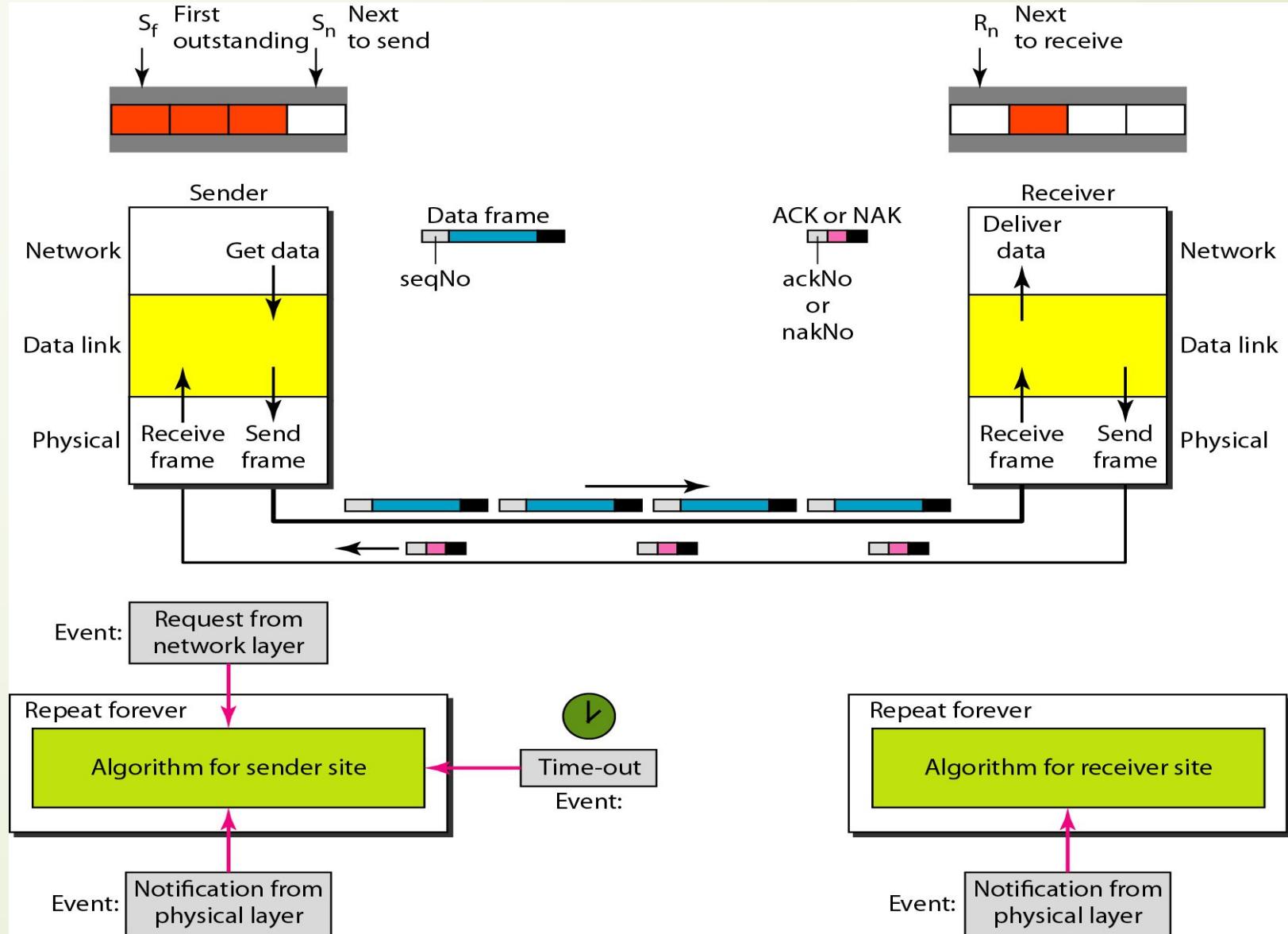
Send window for Selective Repeat ARQ



Receive window for Selective Repeat ARQ



Design of Selective Repeat ARQ

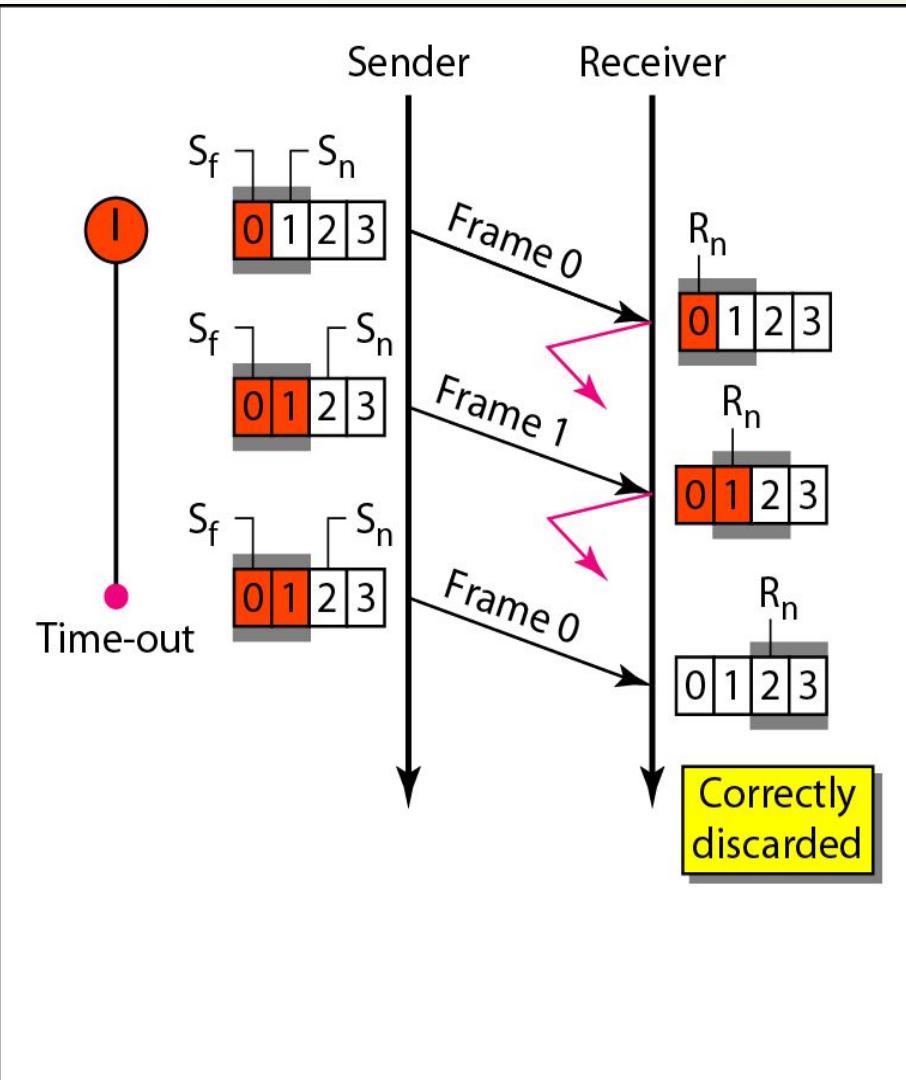




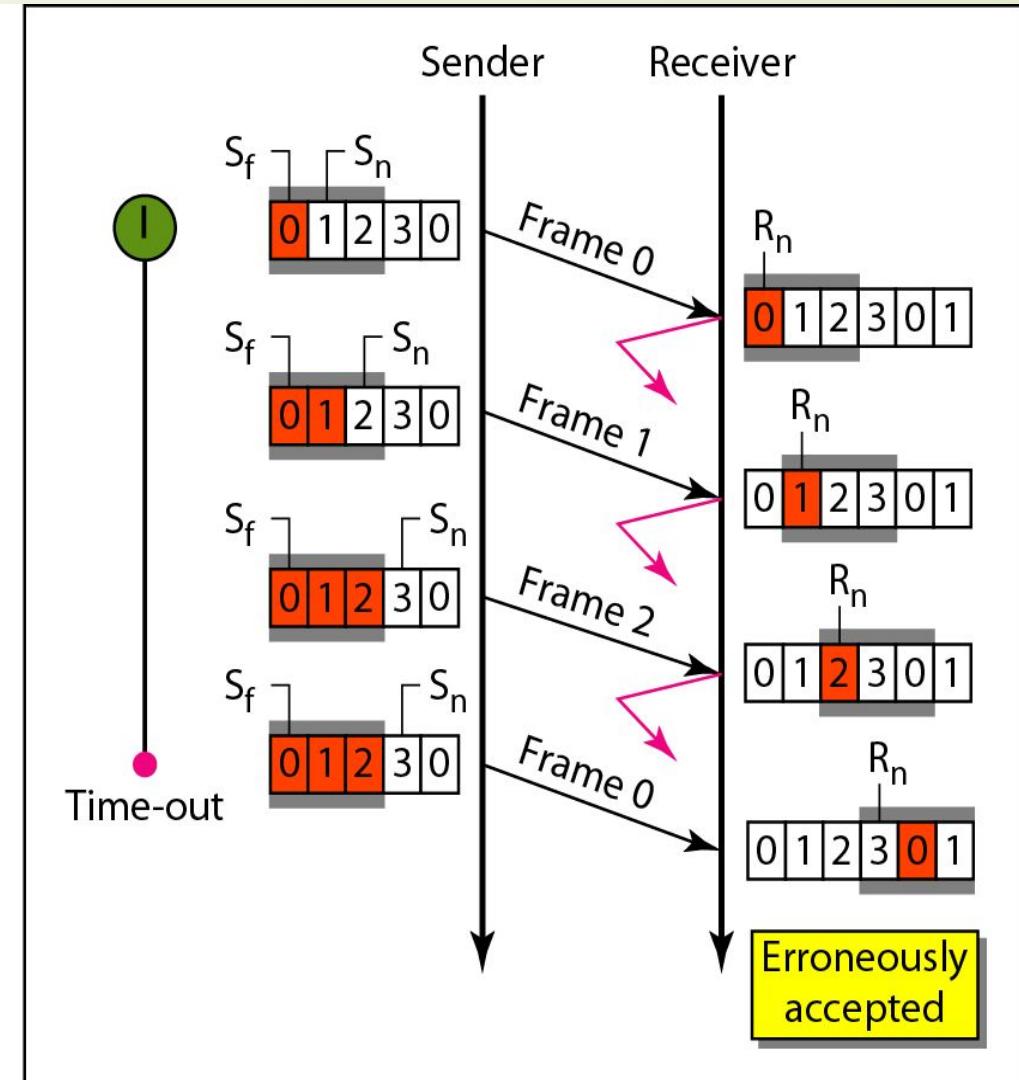
Note

- In Selective Repeat ARQ, the size of the sender and receiver window must be at most one-half of 2^m .

Selective Repeat ARQ, window size

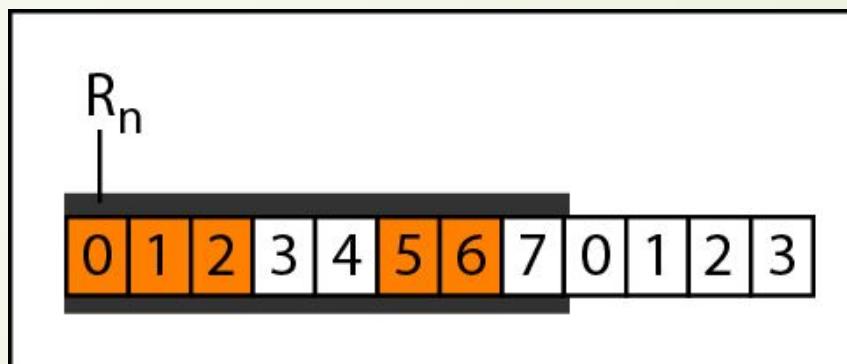


a. Window size = 2^{m-1}

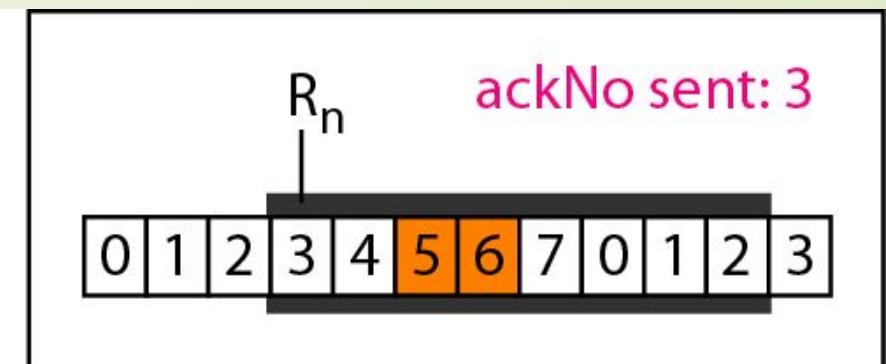


b. Window size > 2^{m-1}

Delivery of data in Selective Repeat ARQ

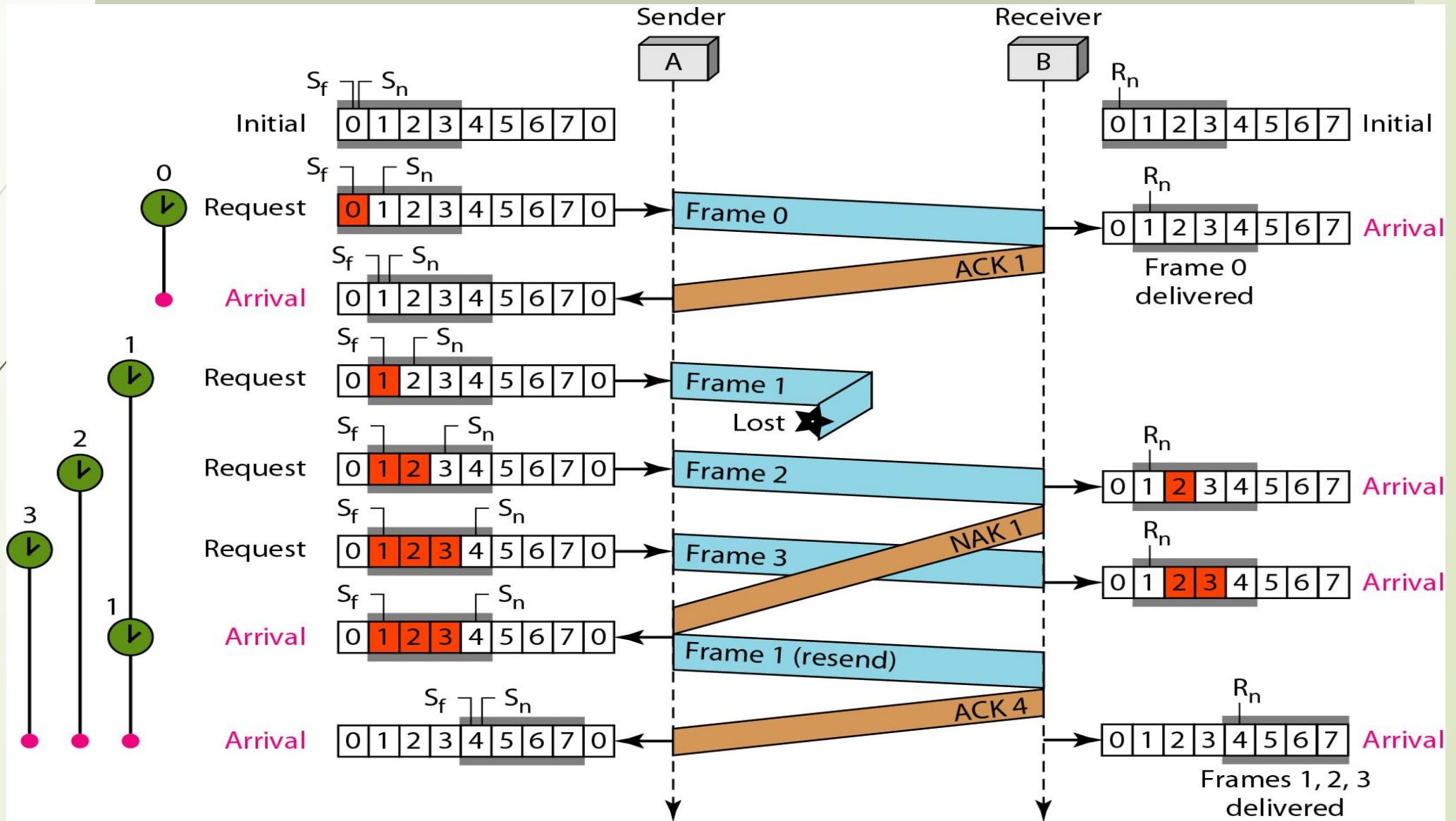


a. Before delivery



b. After delivery

Figure 4.7 *Flow diagram*



Example

- This above example is similar to Example Fig 4.3 in which frame 1 is lost. We show how Selective Repeat behaves in this case. Figure 4.7 shows the situation. One main difference is the number of timers. Here, each frame sent or resent needs a timer, which means that the timers need to be numbered (0, 1, 2, and 3). The timer for frame 0 starts at the first request, but stops when the ACK for this frame arrives. The timer for frame 1 starts at the second request, restarts when a NAK arrives, and finally stops when the last ACK arrives. The other two timers start when the corresponding frames are sent and stop at the last arrival event.



Continued

- At the receiver site we need to distinguish between the acceptance of a frame and its delivery to the network layer. At the second arrival, frame 2 arrives and is stored and marked, but it cannot be delivered because frame 1 is missing. At the next arrival, frame 3 arrives and is marked and stored, but still none of the frames can be delivered. Only at the last arrival, when finally a copy of frame 1 arrives, can frames 1, 2, and 3 be delivered to the network layer. There are two conditions for the delivery of frames to the network layer: First, a set of consecutive frames must have arrived. Second, the set starts from the beginning of the window.



CONTINUED

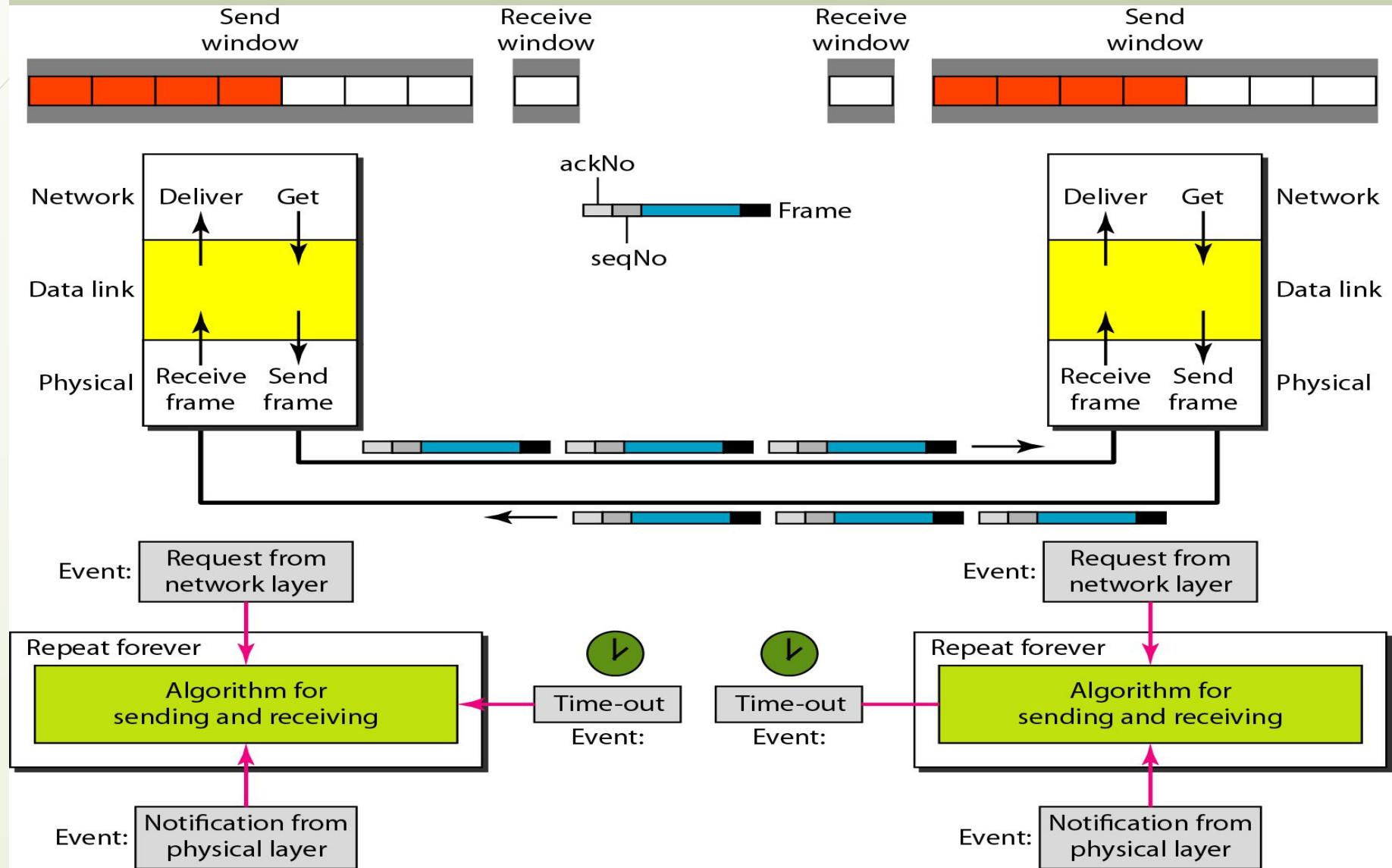
- Another important point is that a NAK is sent after the second arrival, but not after the third, although both situations look the same. The reason is that the protocol does not want to crowd the network with unnecessary NAKs and unnecessary resent frames. The second NAK would still be NAK1 to inform the sender to resend frame 1 again; this has already been done. The first NAK sent is remembered (using the nakSent variable) and is not sent again until the frame slides. A NAK is sent once for each window position and defines the first slot in the window.

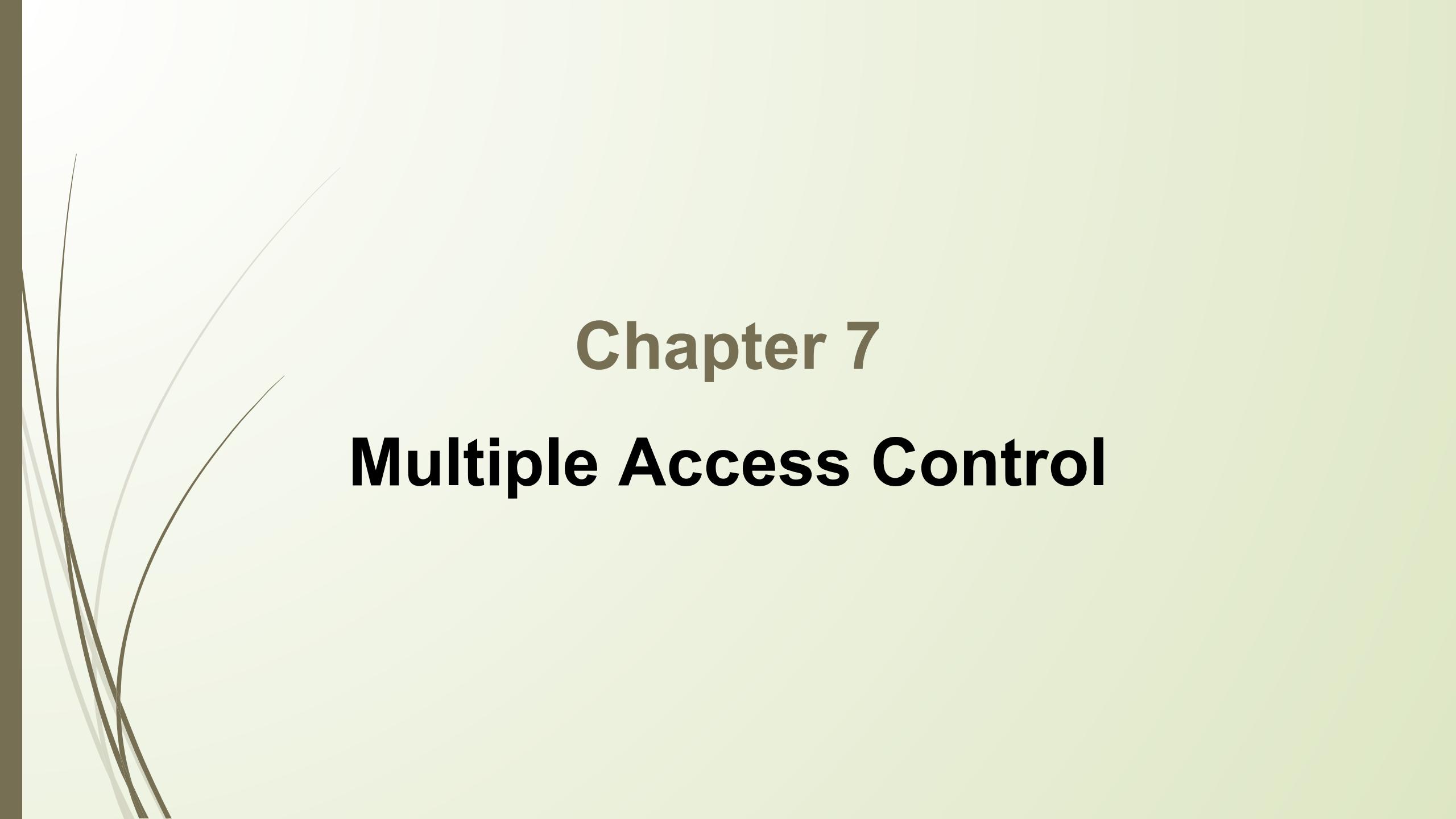


CONTINUED

- The next point is about the ACKs. Notice that only two ACKs are sent here. The first one acknowledges only the first frame; the second one acknowledges three frames. In Selective Repeat, ACKs are sent when data are delivered to the network layer. If the data belonging to n frames are delivered in one shot, only one ACK is sent for all of them.

Design of piggybacking in Go-Back-N ARQ

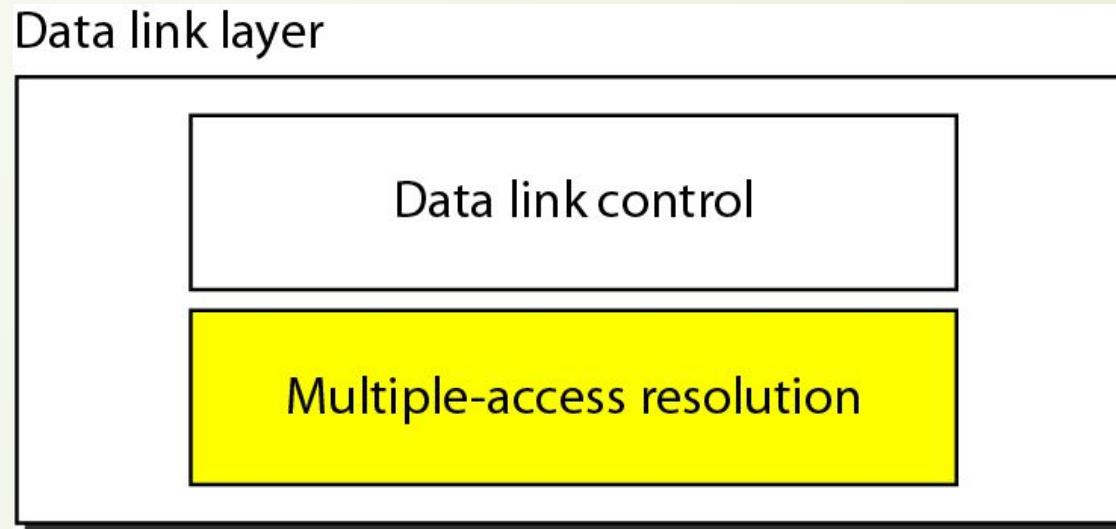




Chapter 7

Multiple Access Control

Figure 7.1 *Data link layer divided into two functionality-oriented sublayers*



Performance Parameters to evaluate a Network Performance

Packet Delivery Ratio (PDR):

Packet Delivery Ratio is the ratio of the number of packets successfully delivered to the destination to the total number of packets sent by the source. It is expressed as a percentage.

Formula:

$$\text{PDR} = \left(\frac{\text{Number of packets received by the destination}}{\text{Number of packets sent by the source}} \right) \times 100\%$$

Significance:

- **Reliability Indicator:** PDR is an indicator of the network's reliability. A high PDR means that most of the packets are successfully reaching their destination, indicating good network performance.
- **Network Health:** Low PDR could indicate issues such as high packet loss, network congestion, or poor link quality.

Performance Parameters to evaluate a Network Performance

Throughput:

Throughput refers to the rate at which data is successfully transmitted over the network. It is usually measured in bits per second (bps), but can also be expressed in packets per second (pps) or data units per time unit.

Formula:

$$\text{Throughput} = \frac{\text{Total data successfully received}}{\text{Total time taken for transmission}}$$

Significance:

Network Efficiency: Throughput is a measure of how much useful data is being transferred through the network. High throughput indicates efficient utilization of network resources.

Capacity Indicator: It reflects the actual capacity of the network to handle data traffic.

Performance Parameters to evaluate a Network Performance

Throughput:

Throughput refers to the rate at which data is successfully transmitted over the network. It is usually measured in bits per second (bps), but can also be expressed in packets per second (pps) or data units per time unit.

Formula:

$$\text{Throughput} = \frac{\text{Total data successfully received}}{\text{Total time taken for transmission}}$$

Significance:

Network Efficiency: Throughput is a measure of how much useful data is being transferred through the network. High throughput indicates efficient utilization of network resources.

Capacity Indicator: It reflects the actual capacity of the network to handle data traffic.

Performance Parameters to evaluate a Network Performance

Latency is the time it takes for a data packet to travel from the source to the destination across a network. It is typically measured in milliseconds (ms).

Components of Latency:

- 1. Propagation Delay:** The time taken for a signal to travel from the sender to the receiver.
- 2. Transmission Delay:** The time taken to push all the packet's bits onto the link.
- 3. Processing Delay:** The time taken by routers and switches to process the packet headers.
- 4. Queuing Delay:** The time a packet spends waiting in queues before being transmitted.

Significance:

Performance Indicator: Low latency is crucial for real-time applications like VoIP, online gaming, and video conferencing, where delays can significantly affect user experience.

Network Responsiveness: High latency can cause delays and slow down communication, leading to a laggy and unresponsive network.

Performance Parameters to evaluate a Network Performance

Jitter refers to the variation in the time delay (latency) of received packets. It is a measure of the variability in packet arrival times and is typically measured in milliseconds (ms).

Formula:

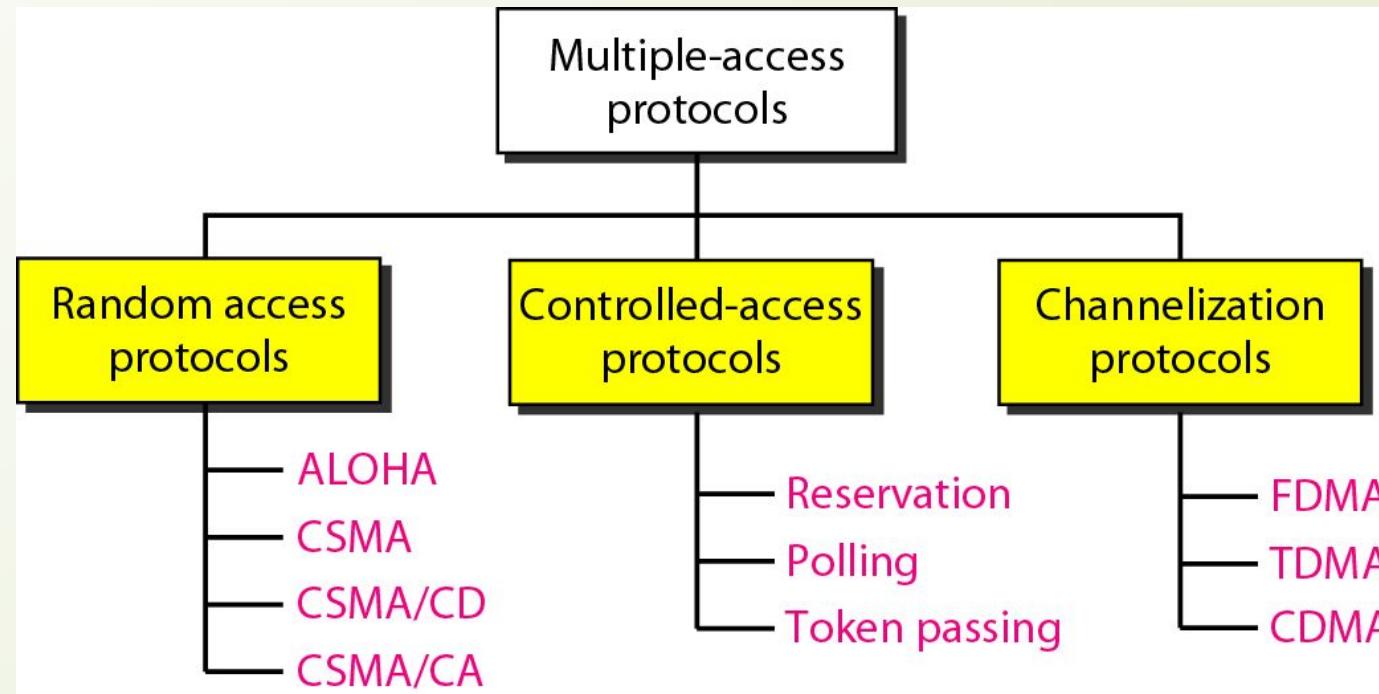
$$\text{Jitter} = |D(i + 1) - D(i)| \text{ Where } D(i)$$

Significance:

Impact on Quality: High jitter can lead to poor quality in real-time communications such as voice and video. It can cause packets to arrive out of order, leading to choppy audio or video.

Stability Indicator: Low jitter indicates a stable and predictable network, which is essential for applications that require a steady stream of data.

Figure 7.2 *Taxonomy of multiple-access protocols discussed in this chapter*



7-1 RANDOM ACCESS

In **random access** or **contention** methods, no station is superior to another station and none is assigned the control over another. No station permits, or does not permit, another station to send. At each instance, a station that has data to send uses a procedure defined by the protocol to make a decision on whether or not to send.

[Topics discussed in this section:](#)

ALOHA

Carrier Sense Multiple Access

Carrier Sense Multiple Access with Collision Detection

Carrier Sense Multiple Access with Collision Avoidance

Pure ALOHA

Transmission Process:

- Devices transmit data whenever they have data to send.
- After transmission, the device waits for an acknowledgment.
- If no acknowledgment is received (due to a collision), the device waits for a random time before retransmitting.

Collisions: Occur when two or more devices transmit at the same time, leading to garbled messages.

Figure 7.3 *Frames in a pure ALOHA network*

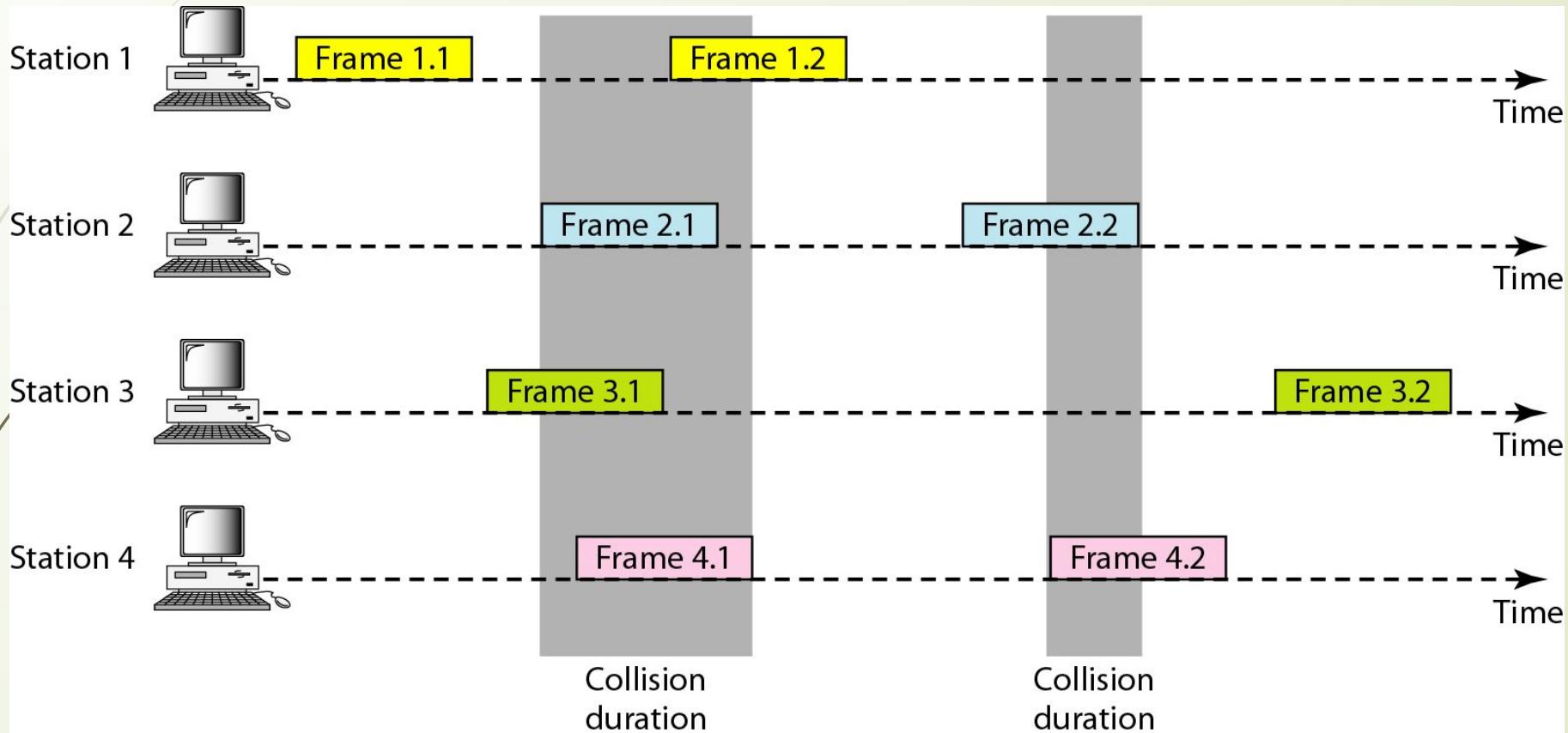
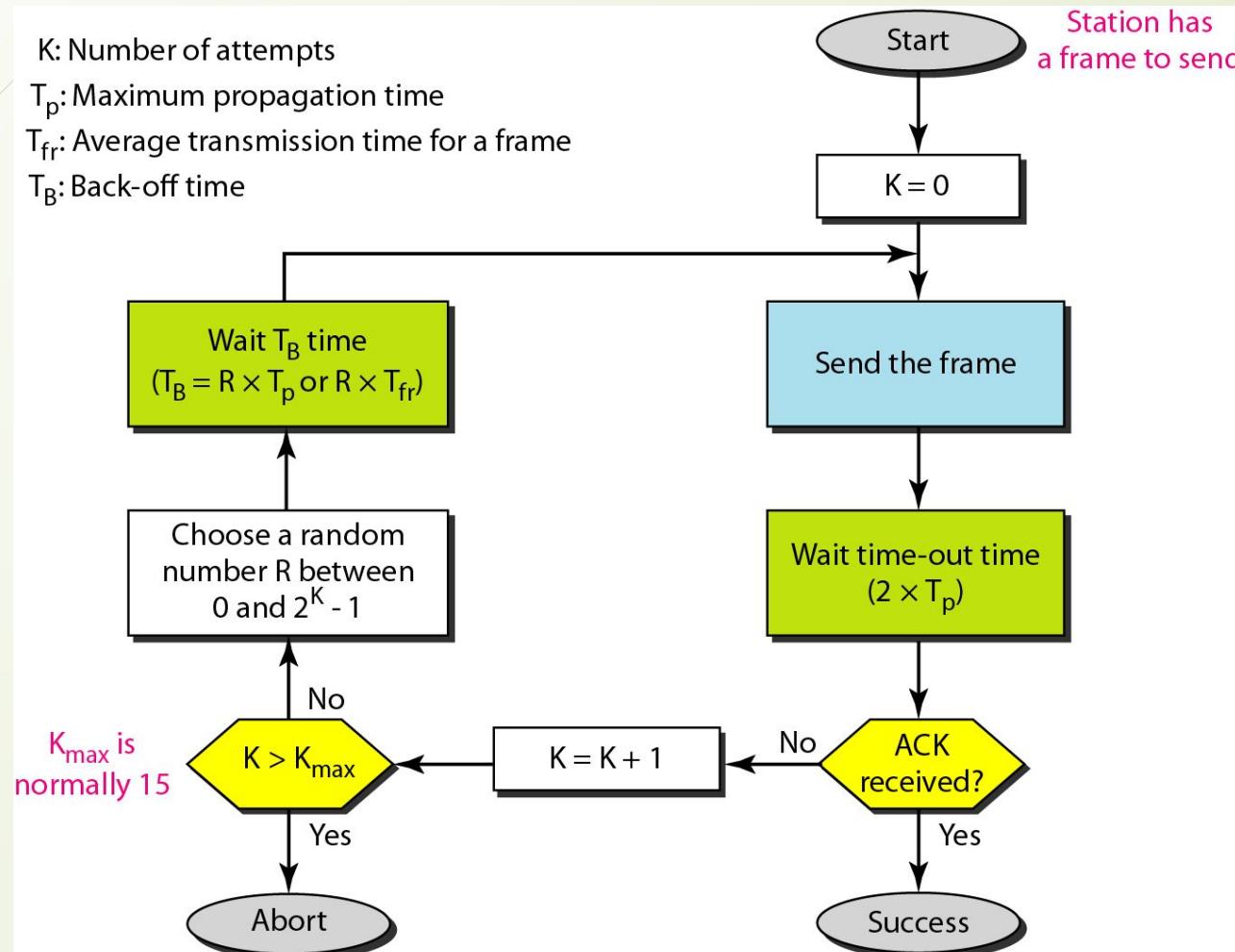
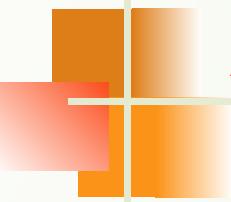


Figure 7.4 Procedure for pure ALOHA protocol





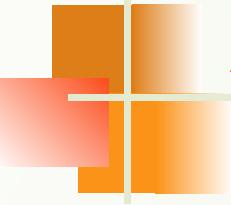
Example 7.1

The stations on a wireless ALOHA network are a maximum of 600 km apart. If we assume that signals propagate at 3×10^8 m/s, we find

$$T_p = (600 \times 10^3) / (3 \times 10^8) = 2 \text{ ms}.$$

Now we can find the value of T_B for different values of K .

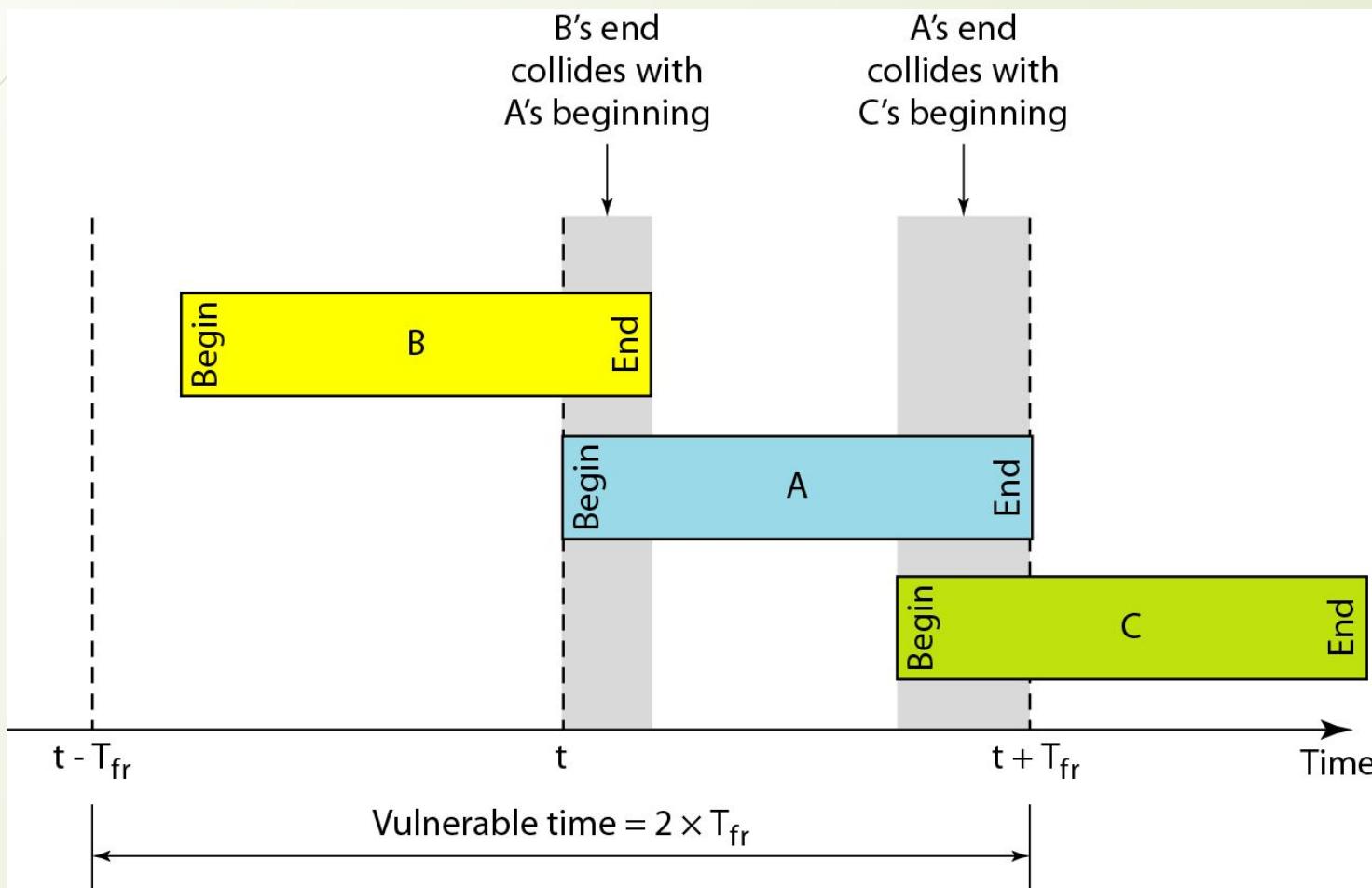
- a. For $K = 1$, the range is $\{0, 1\}$. The station needs to generate a random number with a value of 0 or 1. This means that T_B is either 0 ms (0×2) or 2 ms (1×2), based on the outcome of the random variable.

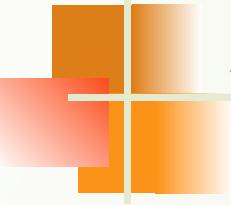


Example 7.1 (continued)

- b. For $K = 2$, the range is $\{0, 1, 2, 3\}$. This means that T_B can be 0, 2, 4, or 6 ms, based on the outcome of the random variable.*
- c. For $K = 3$, the range is $\{0, 1, 2, 3, 4, 5, 6, 7\}$. This means that T_B can be 0, 2, 4, . . . , 14 ms, based on the outcome of the random variable.*
- d. We need to mention that if $K > 10$, it is normally set to 10.*

Figure 7.5 Vulnerable time for pure ALOHA protocol



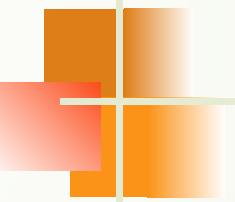


Example 7.2

A pure ALOHA network transmits 200-bit frames on a shared channel of 200 kbps. What is the requirement to make this frame collision-free?

Solution

Average frame transmission time T_{fr} is 200 bits/200 kbps or 1 ms. The vulnerable time is $2 \times 1 \text{ ms} = 2 \text{ ms}$. This means no station should send later than 1 ms before this station starts transmission and no station should start sending during the one 1-ms period that this station is sending.



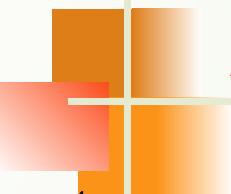
Note

The throughput for pure ALOHA is

$$S = G \times e^{-2G}$$

The maximum throughput

$$S_{\max} = 0.184 \text{ when } G = (1/2).$$



Example 7.3

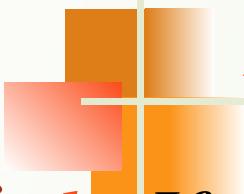
A pure ALOHA network transmits 200-bit frames on a shared channel of 200 kbps. What is the throughput if the system (all stations together) produces

- a. 1000 frames per second*
- b. 500 frames per second*
- c. 250 frames per second.*

Solution

The frame transmission time is 200/200 kbps or 1 ms.

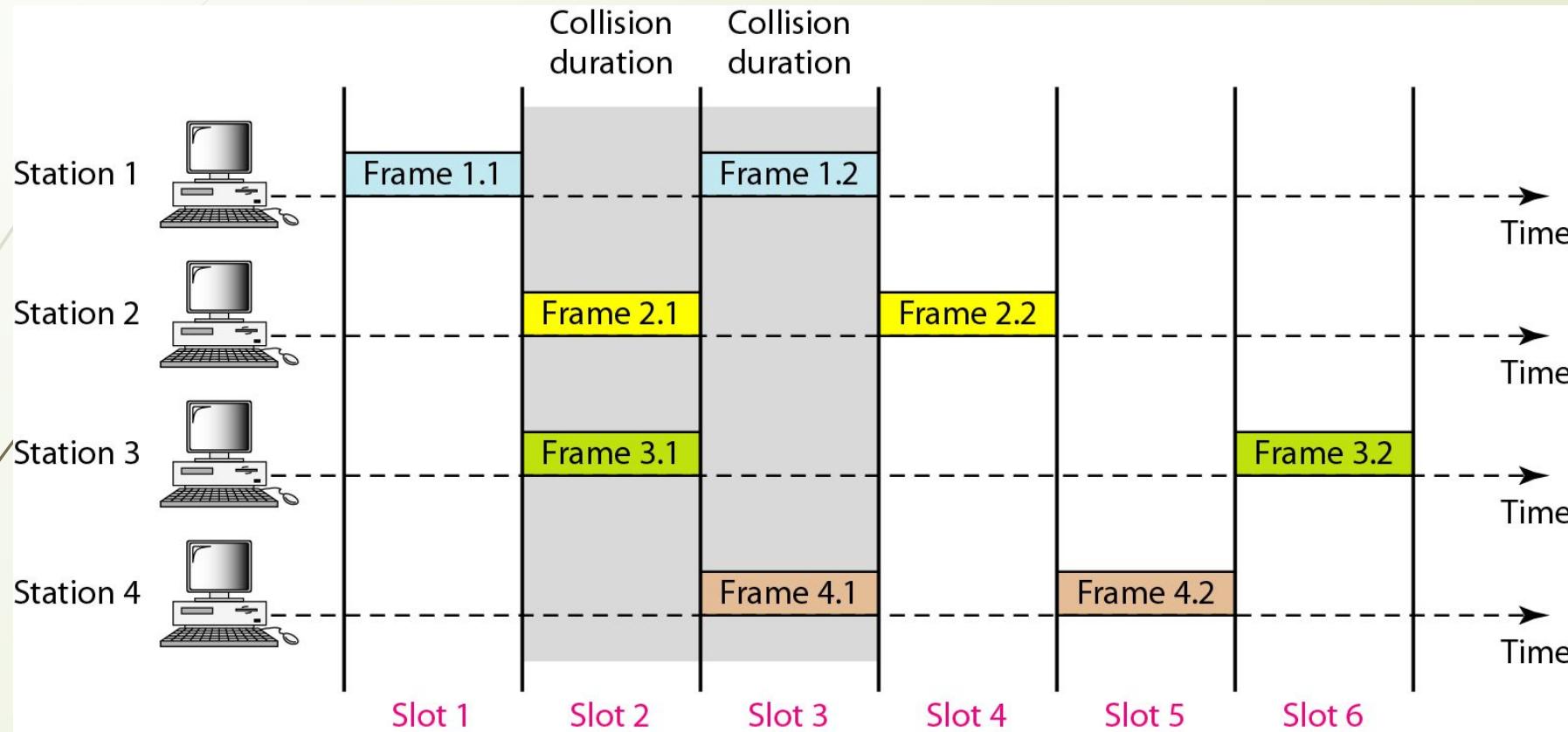
- a. If the system creates 1000 frames per second, this is 1 frame per millisecond. The load is 1. In this case $S = G \times e^{-2G}$ or $S = 0.135$ (13.5 percent). This means that the throughput is $1000 \times 0.135 = 135$ frames. Only 135 frames out of 1000 will probably survive.*

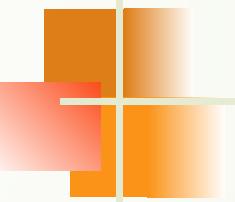


Example 7.3 (continued)

- b. If the system creates 500 frames per second, this is $(1/2)$ frame per millisecond. The load is $(1/2)$. In this case $S = G \times e^{-2G}$ or $S = 0.184$ (18.4 percent). This means that the throughput is $500 \times 0.184 = 92$ and that only 92 frames out of 500 will probably survive. Note that this is the maximum throughput case, percentagewise.
- c. If the system creates 250 frames per second, this is $(1/4)$ frame per millisecond. The load is $(1/4)$. In this case $S = G \times e^{-2G}$ or $S = 0.152$ (15.2 percent). This means that the throughput is $250 \times 0.152 = 38$. Only 38 frames out of 250 will probably survive.

Figure 7.6 *Frames in a slotted ALOHA network*





Note

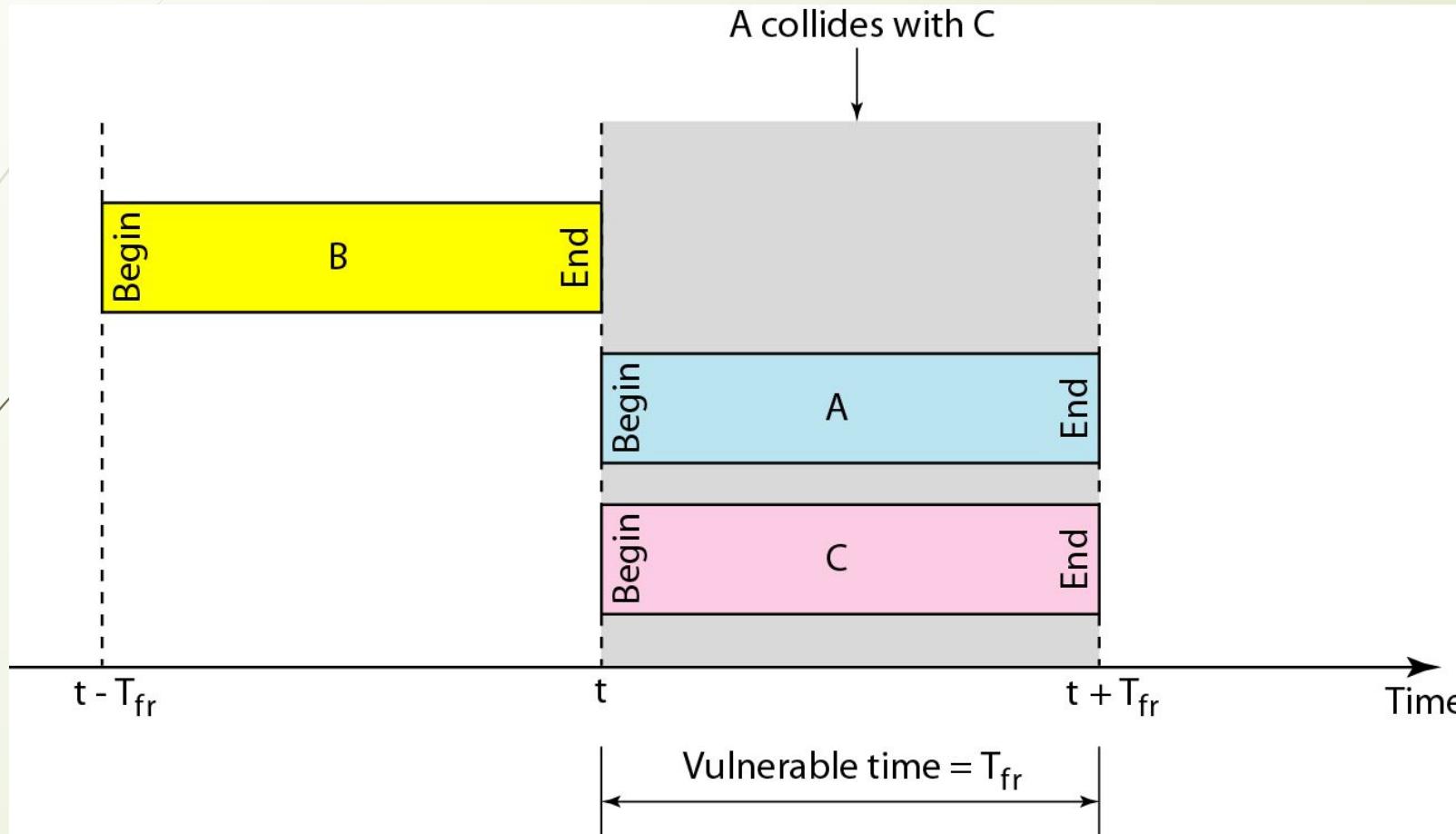
The throughput for slotted ALOHA is

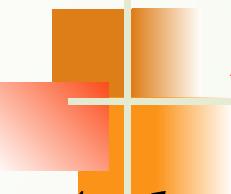
$$S = G \times e^{-G}.$$

The maximum throughput

$$S_{\max} = 0.368 \text{ when } G = 1.$$

Figure 7.7 Vulnerable time for slotted ALOHA protocol





Example 7.4

A slotted ALOHA network transmits 200-bit frames on a shared channel of 200 kbps. What is the throughput if the system (all stations together) produces

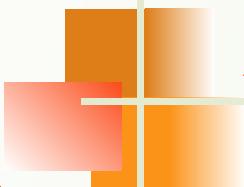
- a. 1000 frames per second b. 500 frames per second*
- c. 250 frames per second.*

Solution

The frame transmission time is 200/200 kbps or 1 ms.

- a. If the system creates 1000 frames per second, this is 1 frame per millisecond. The load is 1. In this case $S = G \times e^{-G}$ or $S = 0.368$ (36.8 percent). This means that the throughput is $1000 \times 0.0368 = 368$ frames.*

Only 386 frames out of 1000 will probably survive.



Example 7.4 (continued)

- b. If the system creates 500 frames per second, this is $(1/2)$ frame per millisecond. The load is $(1/2)$. In this case $S = G \times e^{-G}$ or $S = 0.303$ (30.3 percent). This means that the throughput is $500 \times 0.0303 = 151$. Only 151 frames out of 500 will probably survive.
- c. If the system creates 250 frames per second, this is $(1/4)$ frame per millisecond. The load is $(1/4)$. In this case $S = G \times e^{-G}$ or $S = 0.195$ (19.5 percent). This means that the throughput is $250 \times 0.195 = 49$. Only 49 frames out of 250 will probably survive.

Figure 7.8 Space/time model of the collision in CSMA

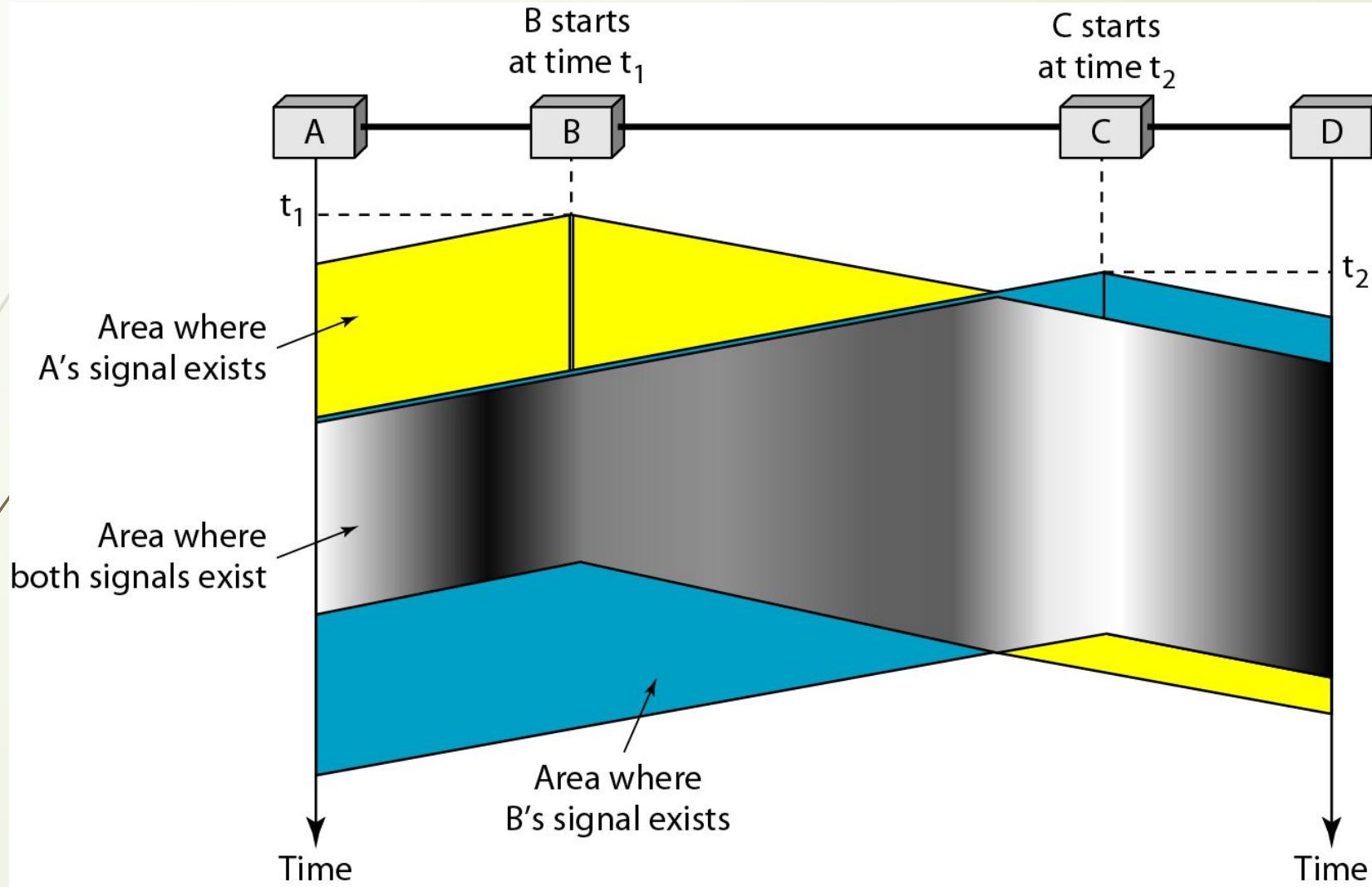


Figure 7.9 Vulnerable time in CSMA

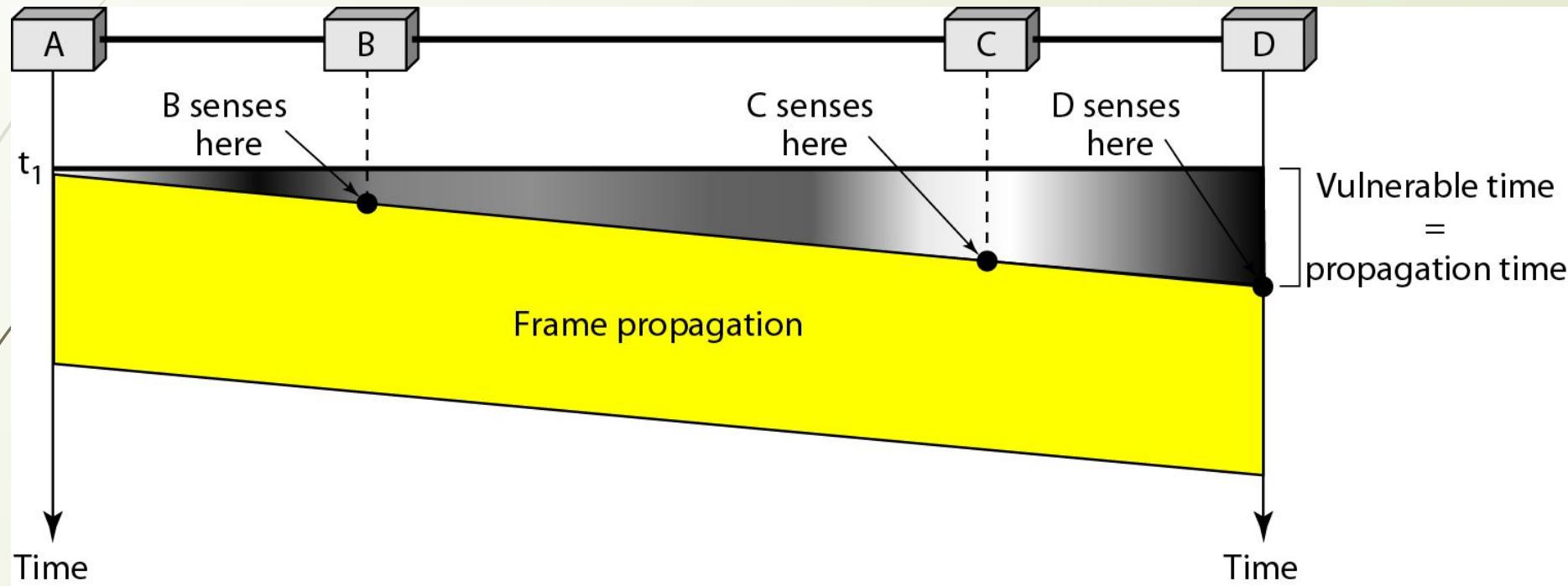


Figure 7.10 Behavior of three persistence methods

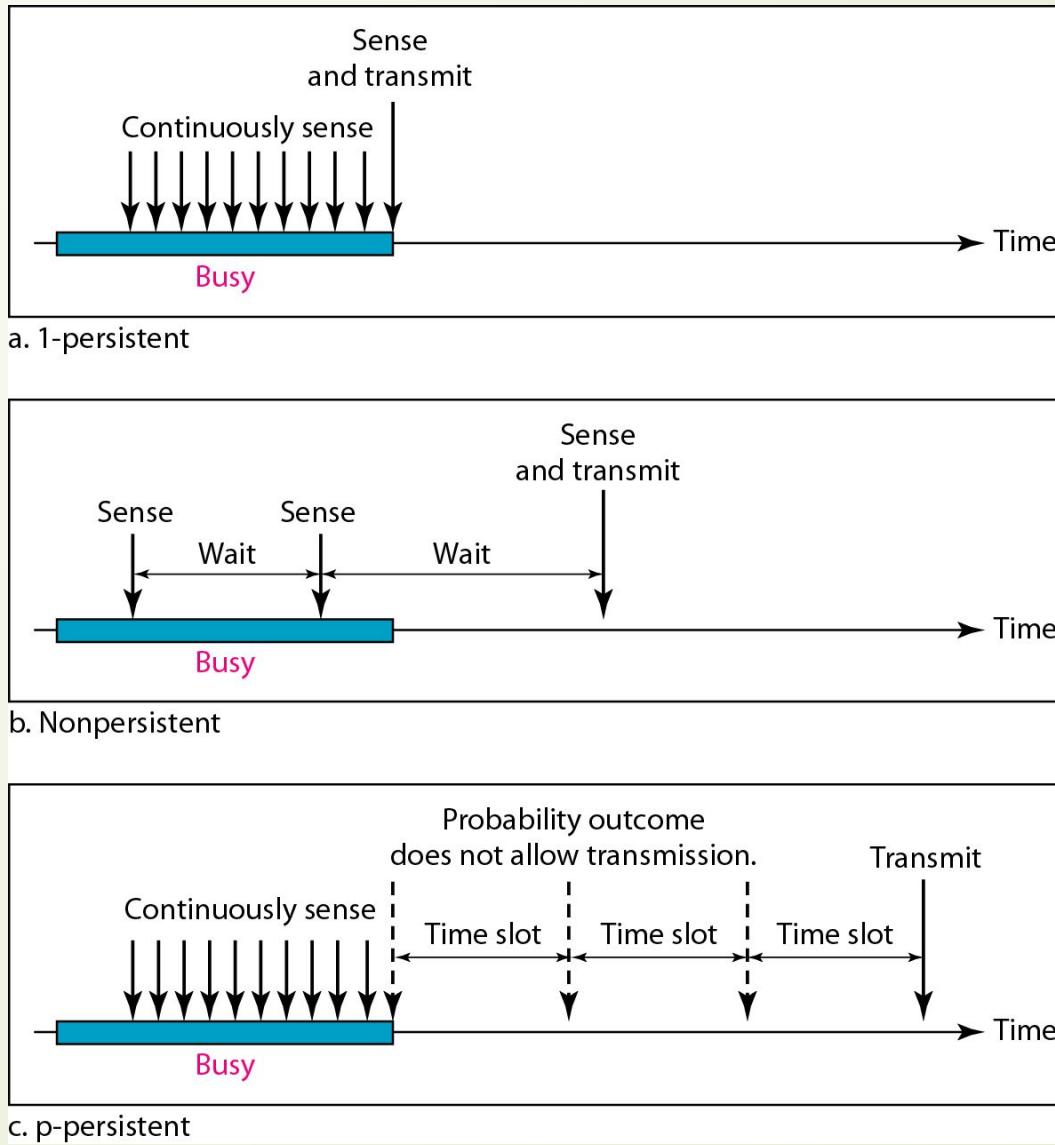


Figure 7.11 Flow diagram for three persistence methods

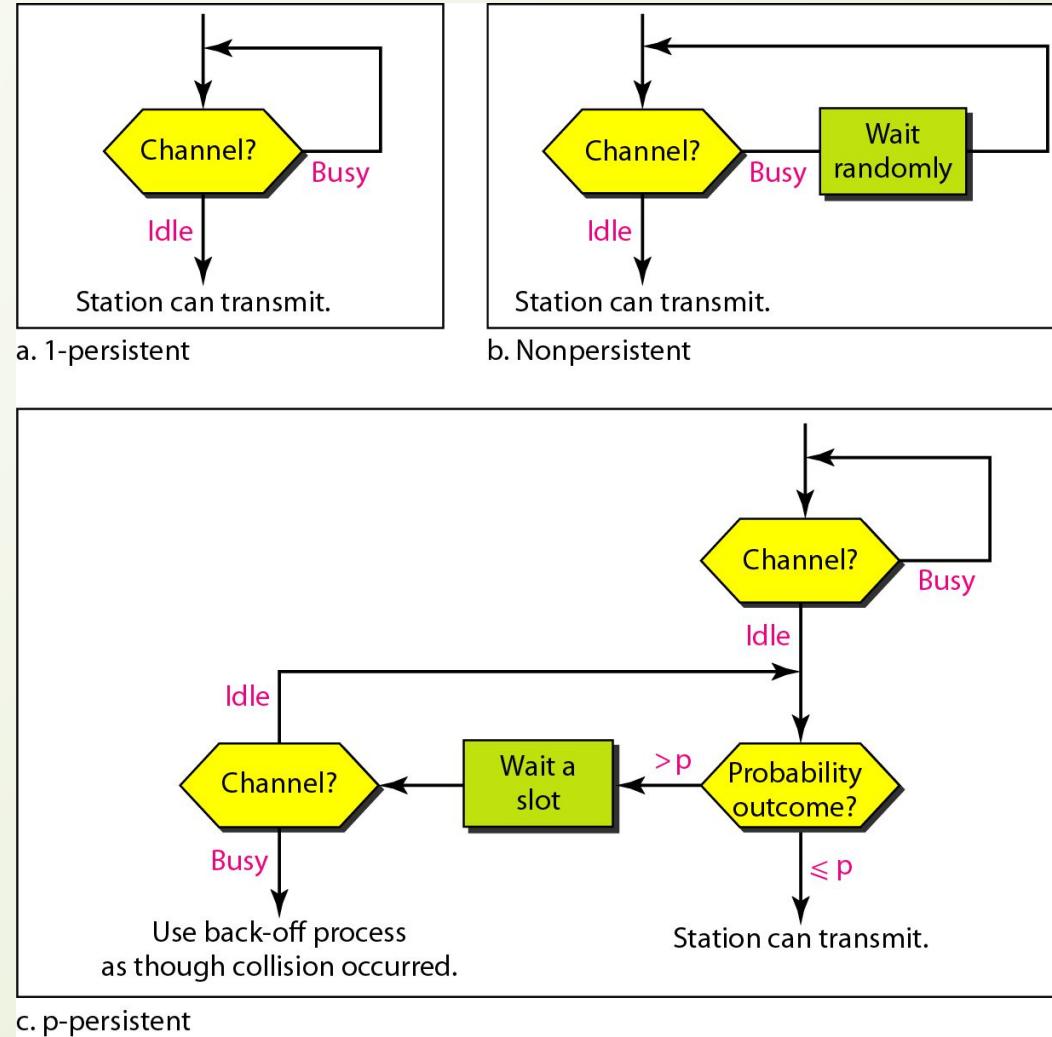


Figure 7.12 Collision of the first bit in CSMA/CD

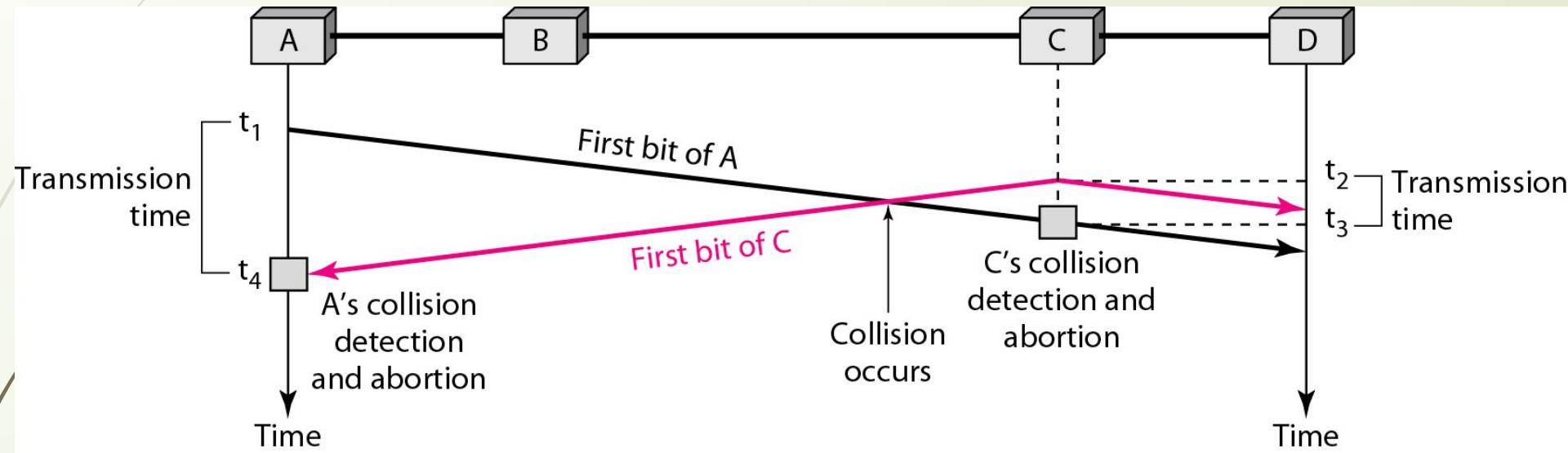
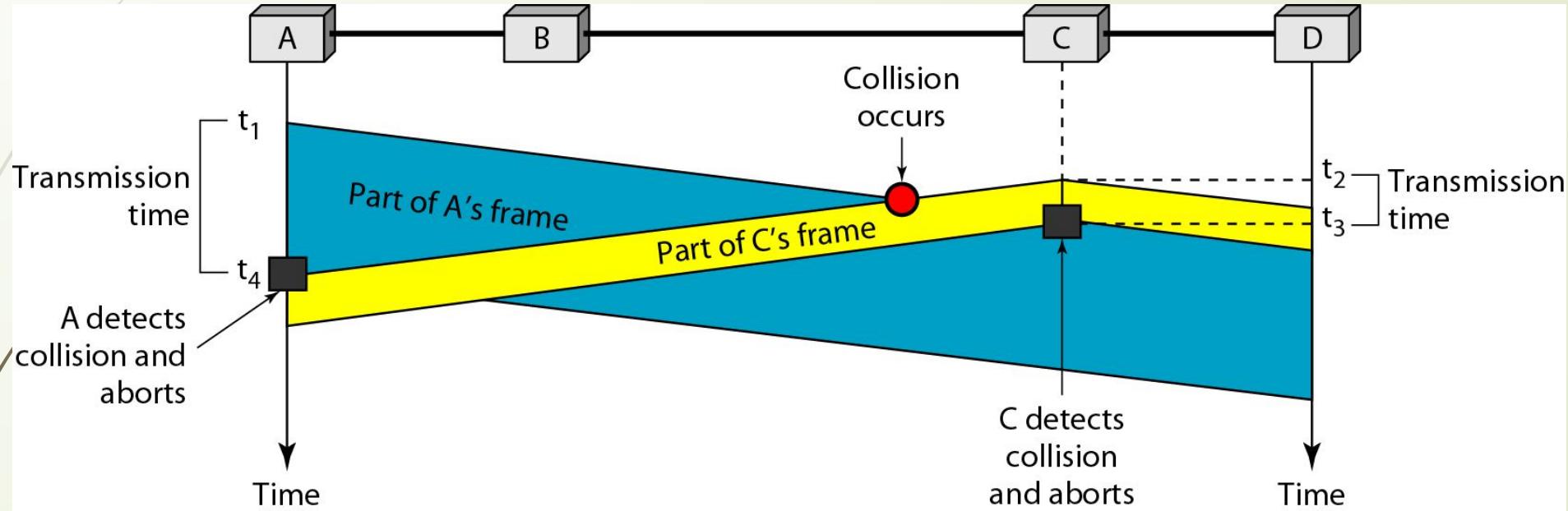
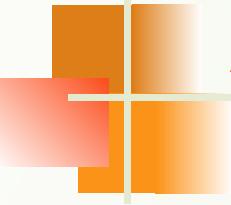


Figure 7.13 Collision and abortion in CSMA/CD





Example 7.5

A network using CSMA/CD has a bandwidth of 10 Mbps. If the maximum propagation time (including the delays in the devices and ignoring the time needed to send a jamming signal, as we see later) is 25.6 μ s, what is the minimum size of the frame?

Solution

The frame transmission time is $T_{fr} = 2 \times T_p = 51.2 \mu$ s. This means, in the worst case, a station needs to transmit for a period of 51.2 μ s to detect the collision. The minimum size of the frame is $10 \text{ Mbps} \times 51.2 \mu\text{s} = 512$ bits or 64 bytes. This is actually the minimum size of the frame for Standard Ethernet.

Figure 7.14 Flow diagram for the CSMA/CD

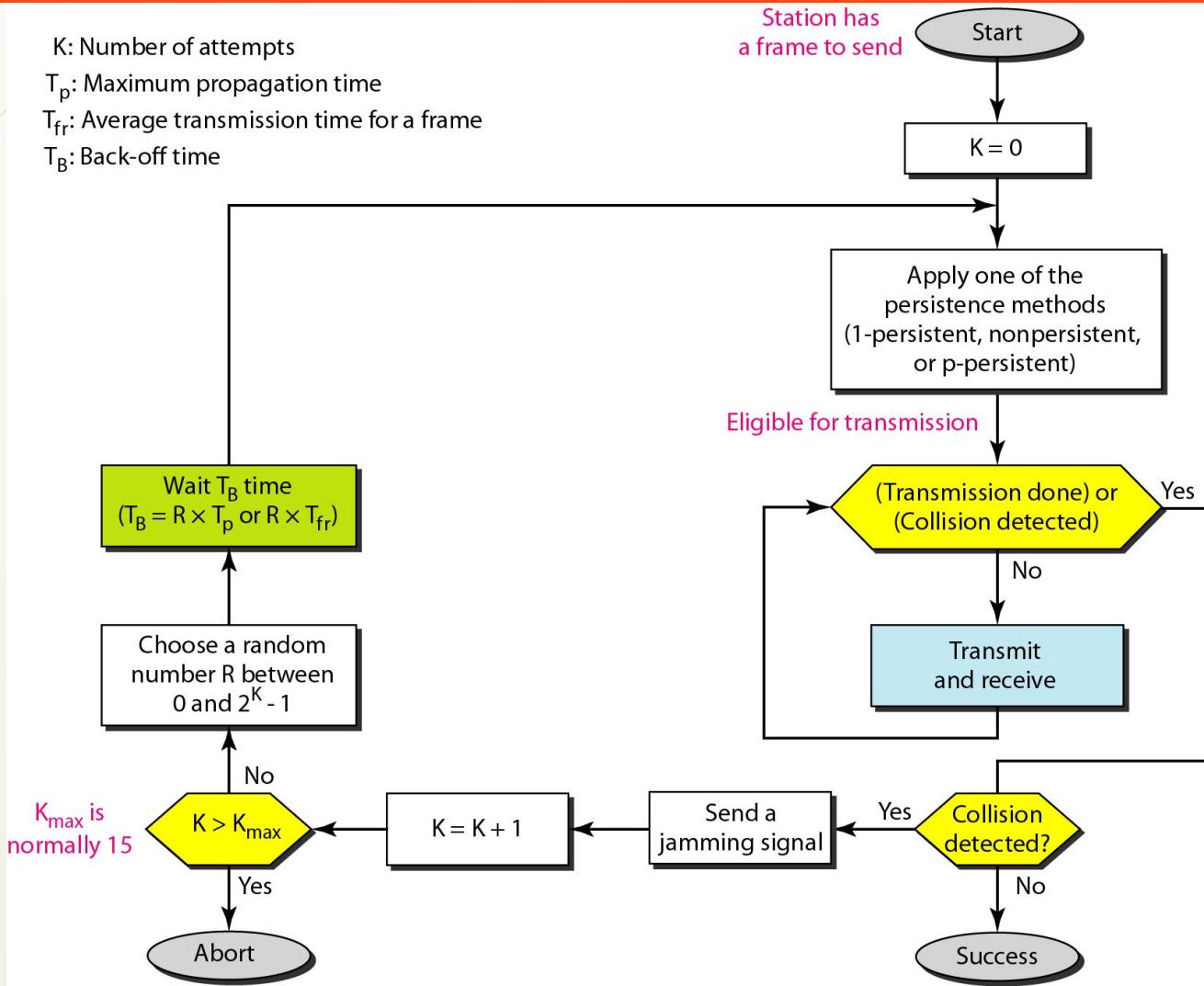


Figure 7.15 Energy level during transmission, idleness, or collision

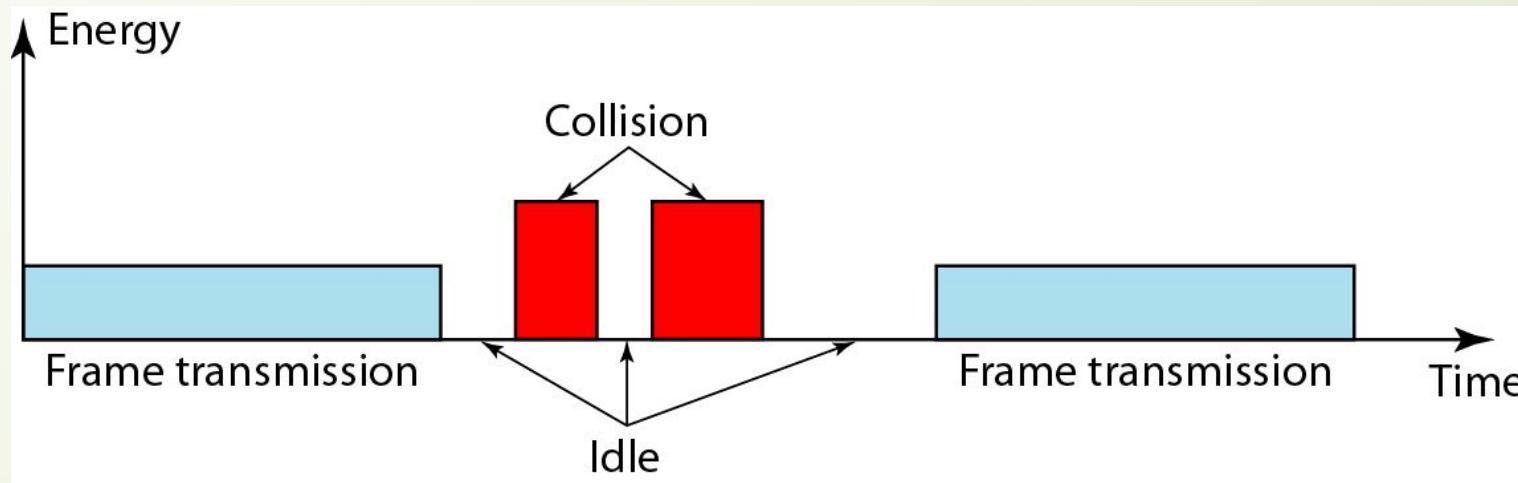
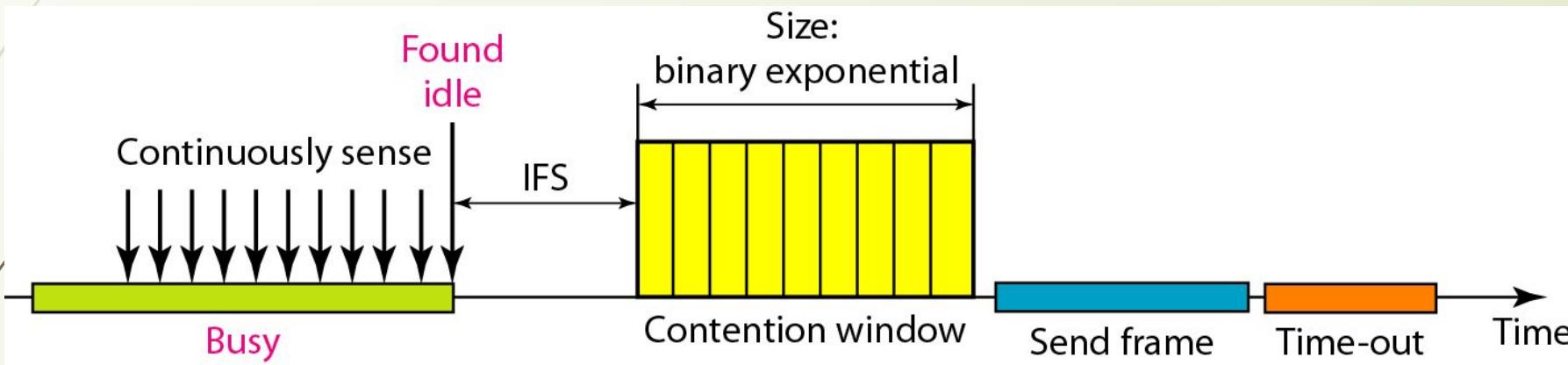
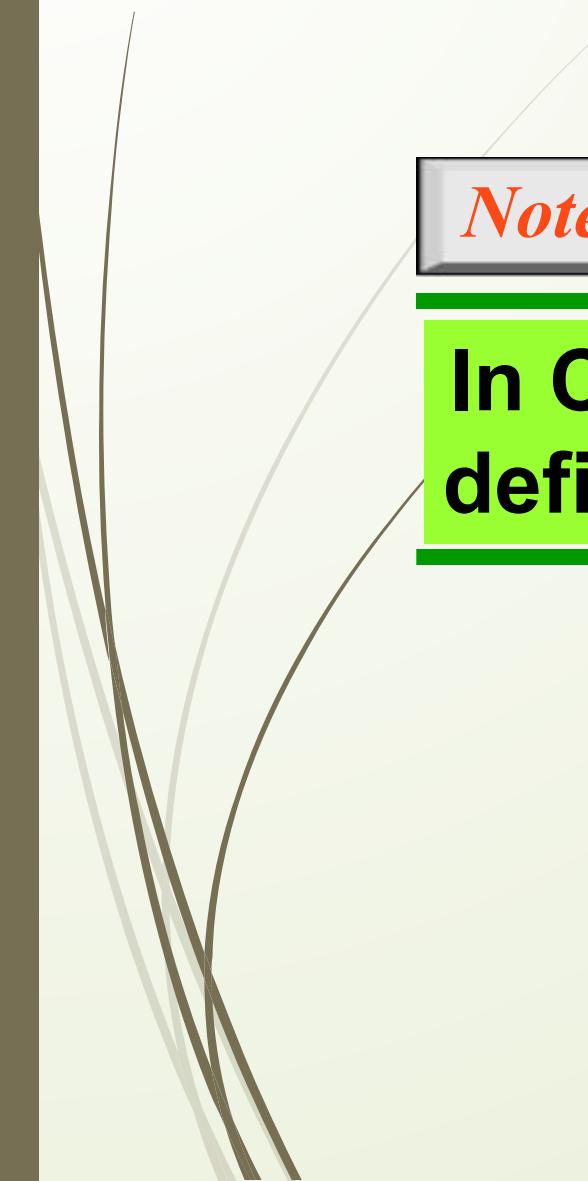


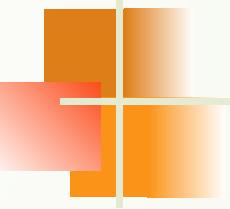
Figure 7.16 Timing in CSMA/CA





Note

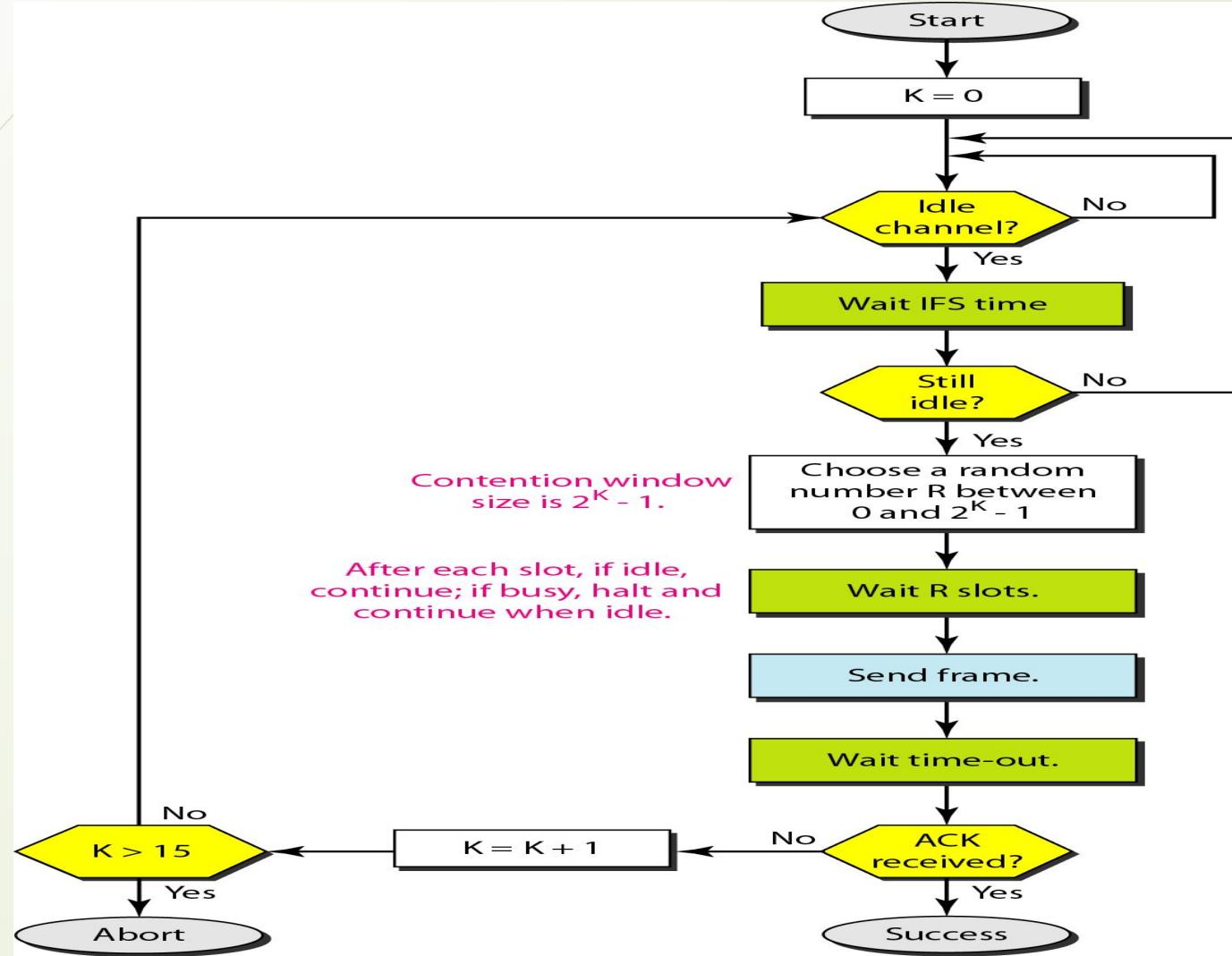
In CSMA/CA, the IFS can also be used to define the priority of a station or a frame.



Note

In CSMA/CA, if the station finds the channel busy, it does not restart the timer of the contention window; it stops the timer and restarts it when the channel becomes idle.

Figure 7.17 Flow diagram for CSMA/CA



7-2 CONTROLLED ACCESS

In **controlled access**, the stations consult one another to find which station has the right to send. A station cannot send unless it has been authorized by other stations. We discuss three popular controlled-access methods.

Topics discussed in this
section:

Reservation

Polling

Token Passing

Figure 7.18 Reservation access method

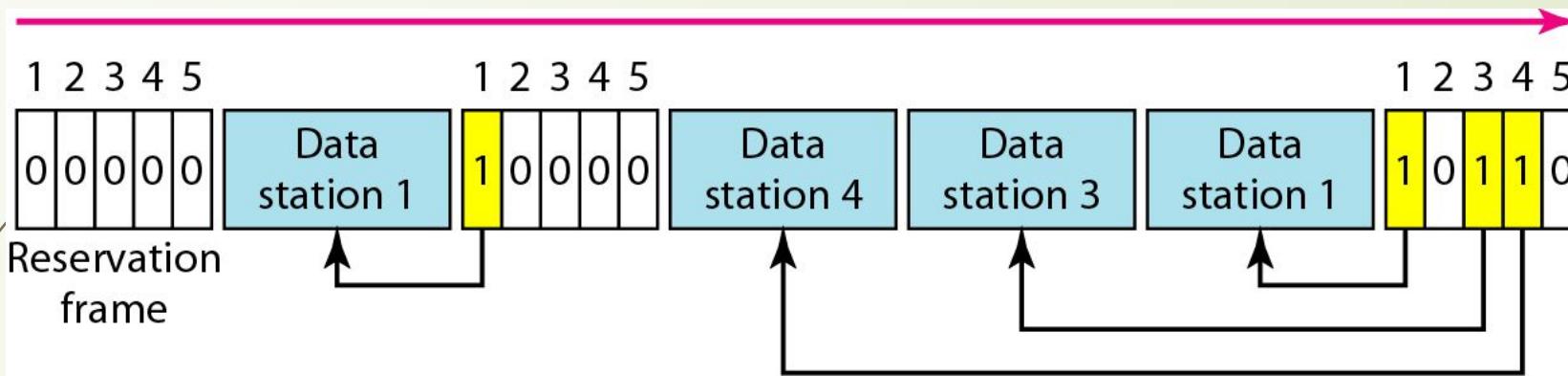


Figure 7.19 Select and poll functions in polling access method

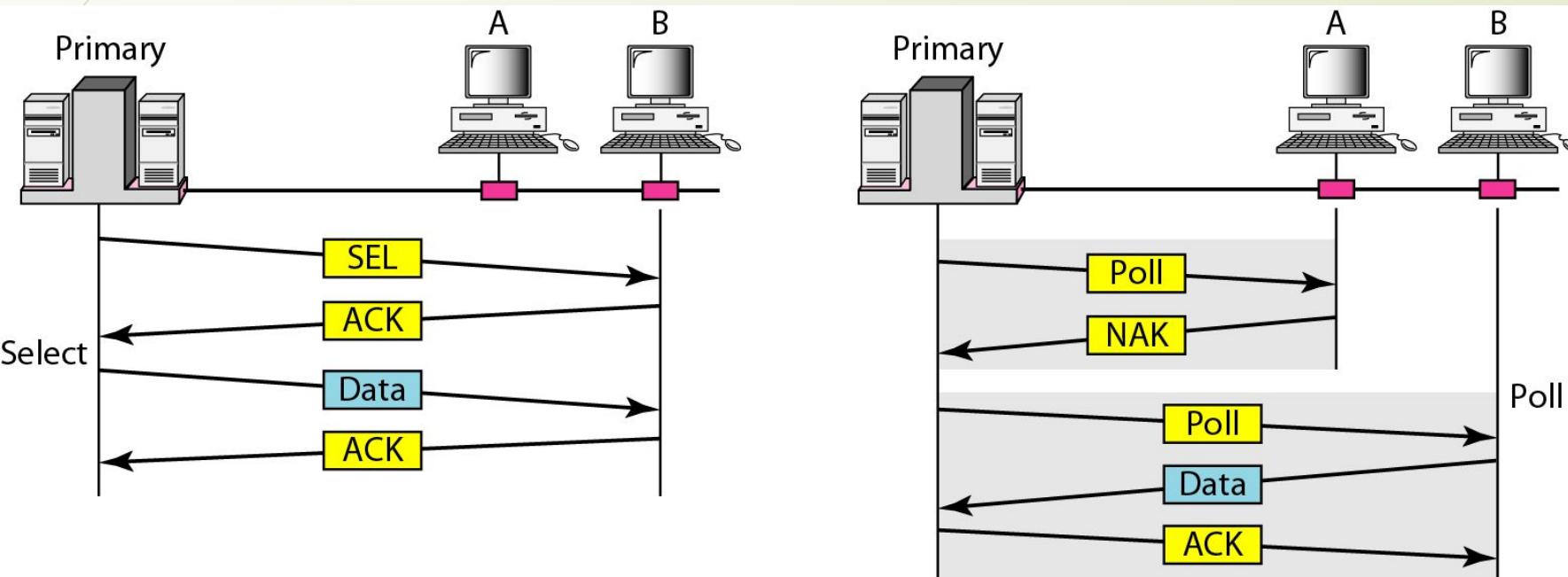
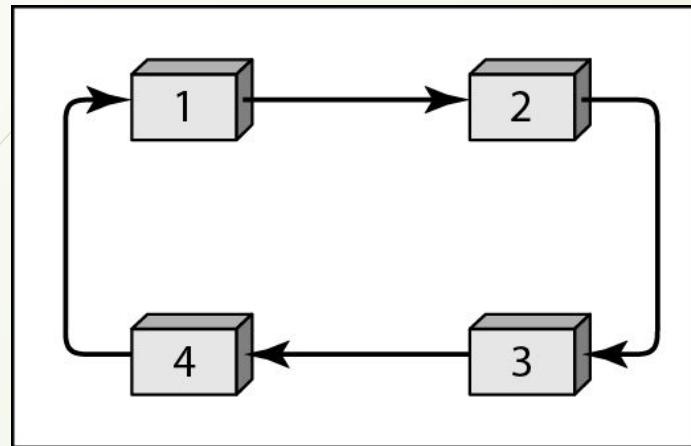
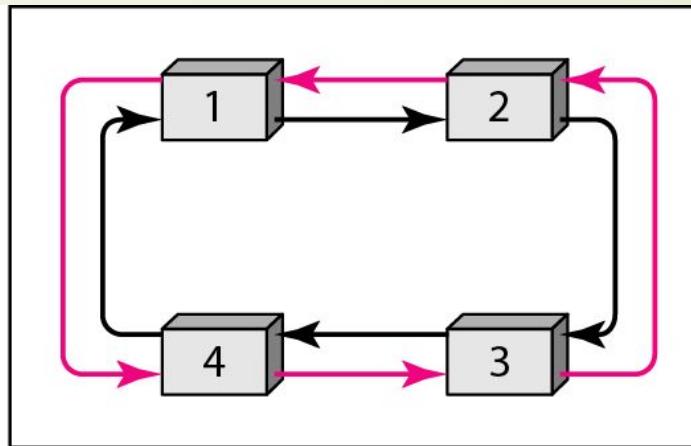


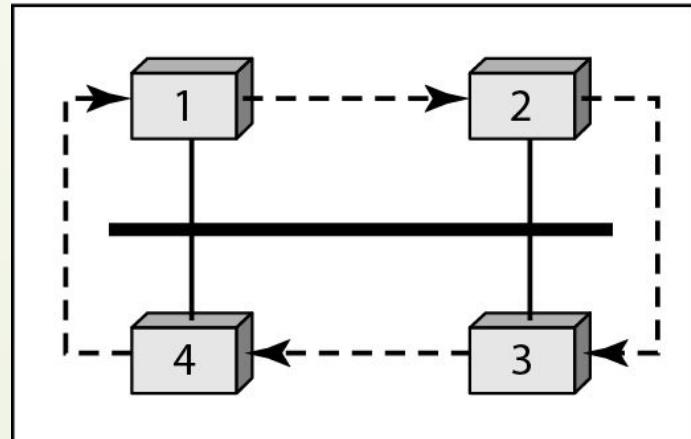
Figure 7.20 *Logical ring and physical topology in token-passing access method*



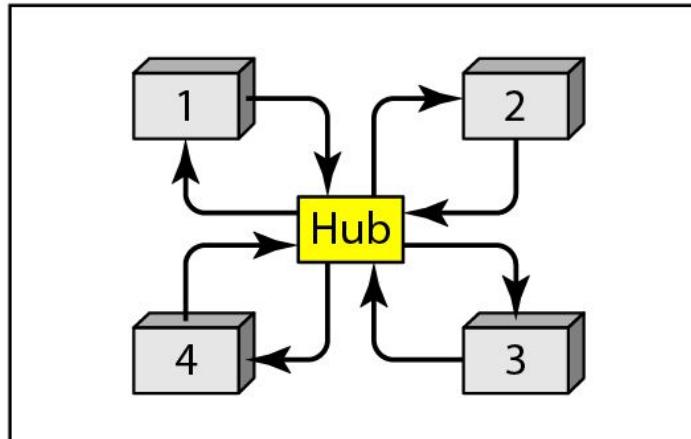
a. Physical ring



b. Dual ring



c. Bus ring



d. Star ring

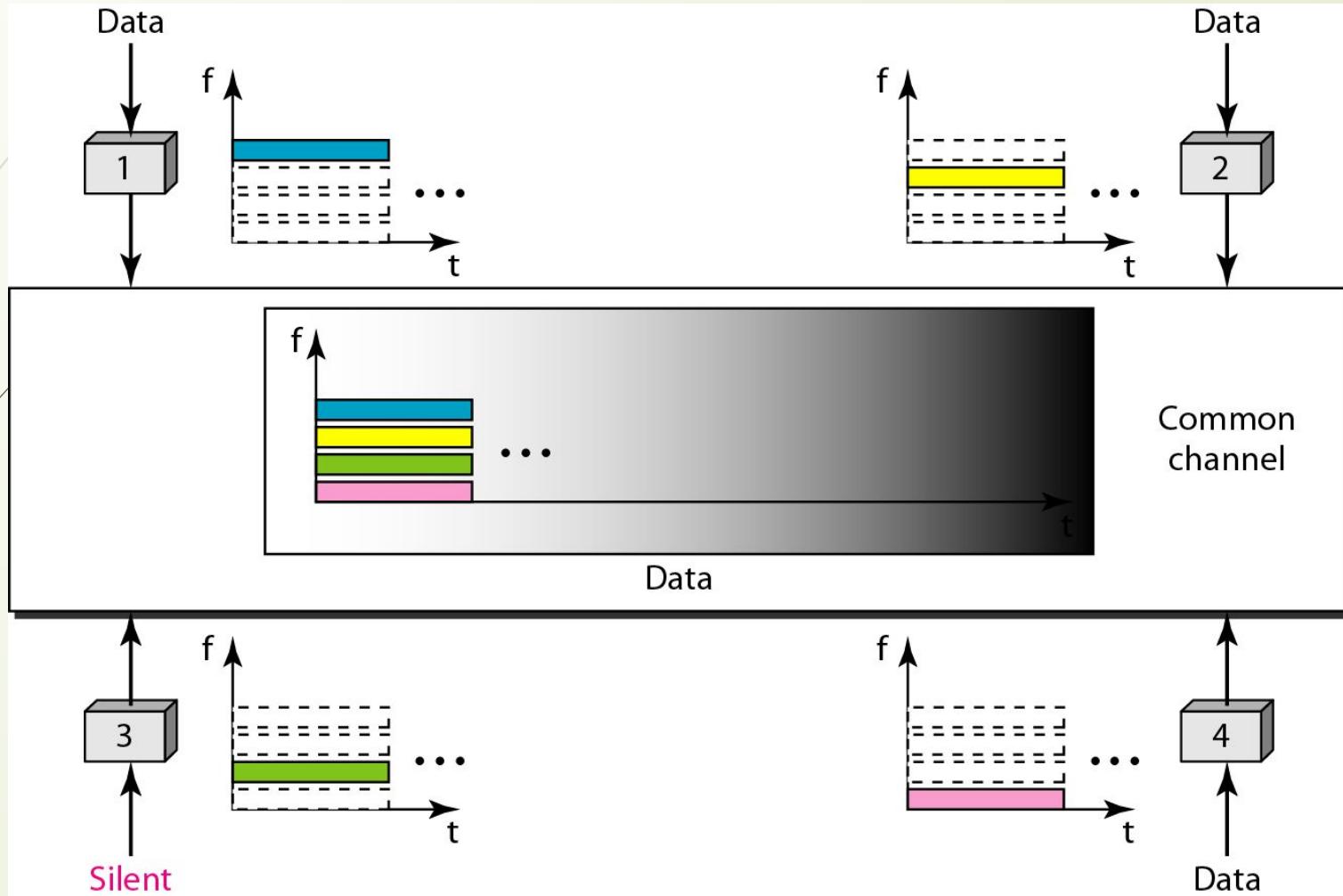
7-3 CHANNELIZATION

Channelization is a multiple-access method in which the available bandwidth of a link is shared in time, frequency, or through code, between different stations. In this section, we discuss three channelization protocols.

Topics discussed in this
section:

Frequency-Division Multiple Access (FDMA)
Time-Division Multiple Access (TDMA)
Code-Division Multiple Access (CDMA)

Figure 7.21 Frequency-division multiple access (FDMA)

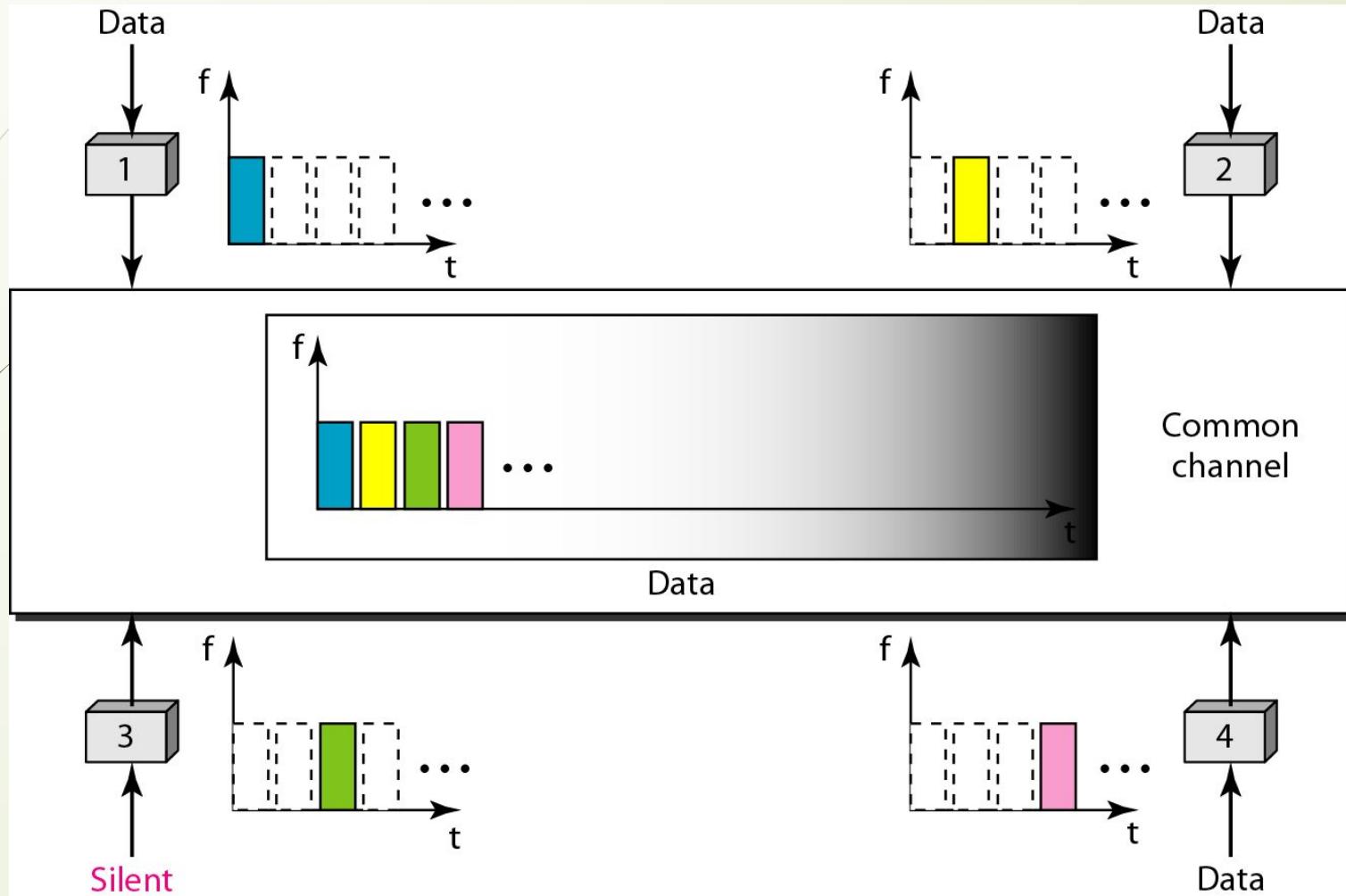


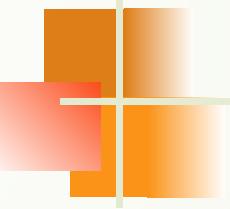


Note

In FDMA, the available bandwidth of the common channel is divided into bands that are separated by guard bands.

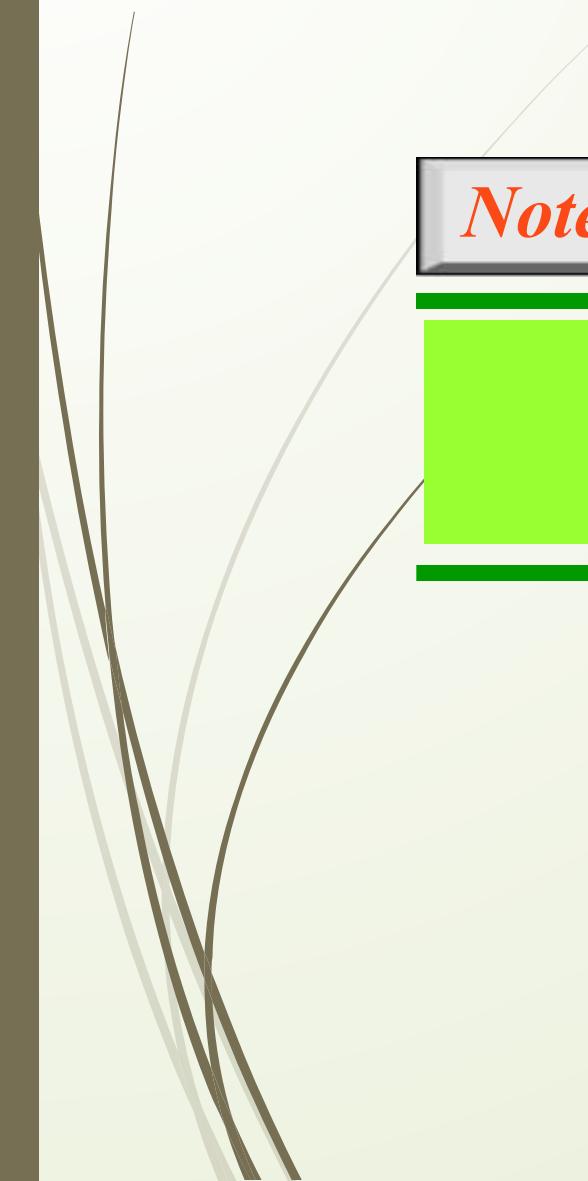
Figure 7.22 Time-division multiple access (TDMA)





Note

In TDMA, the bandwidth is just one channel that is timeshared between different stations.



Note

In CDMA, one channel carries all transmissions simultaneously.

Figure 7.23 Simple idea of communication with code

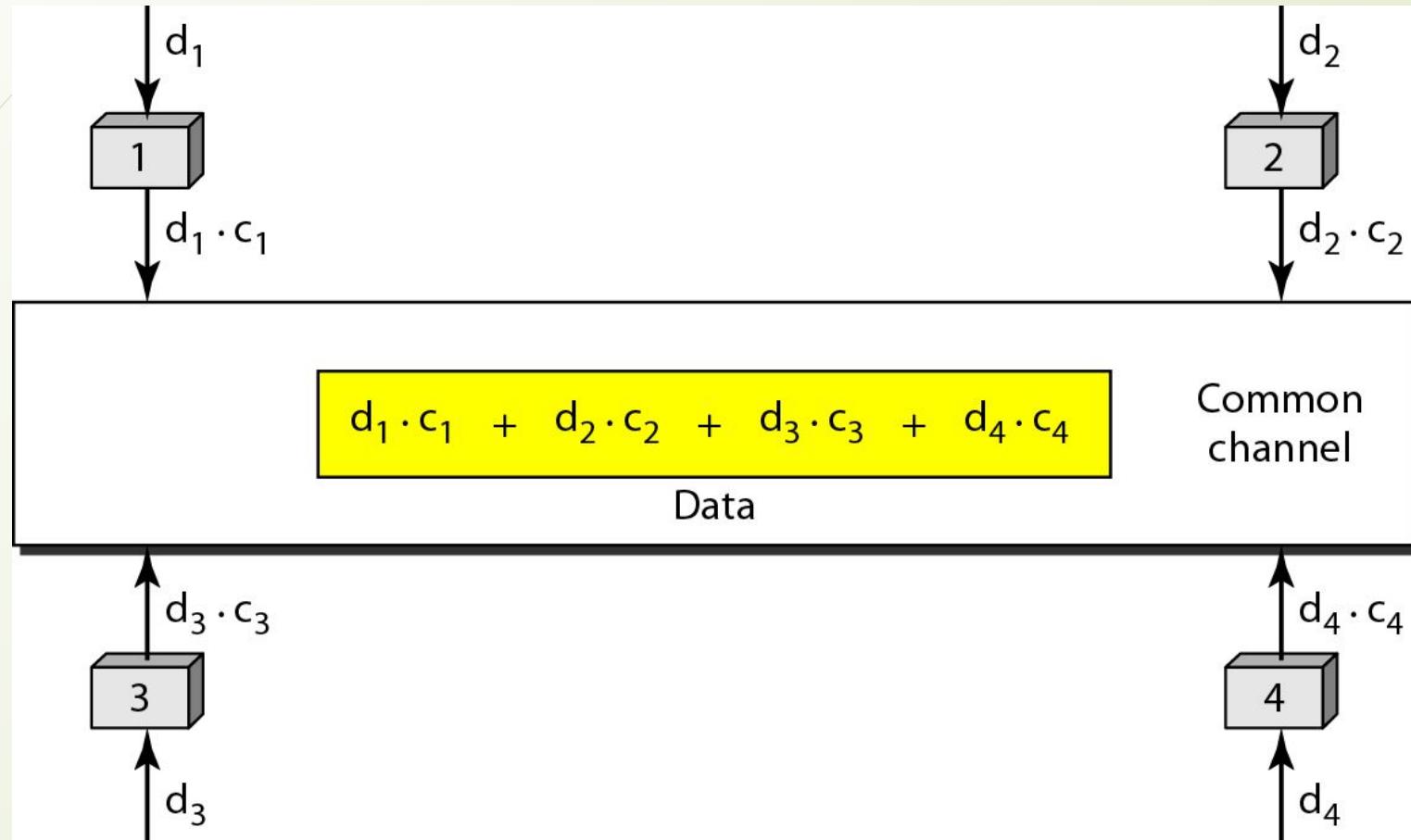


Figure 7.24 *Chip sequences*

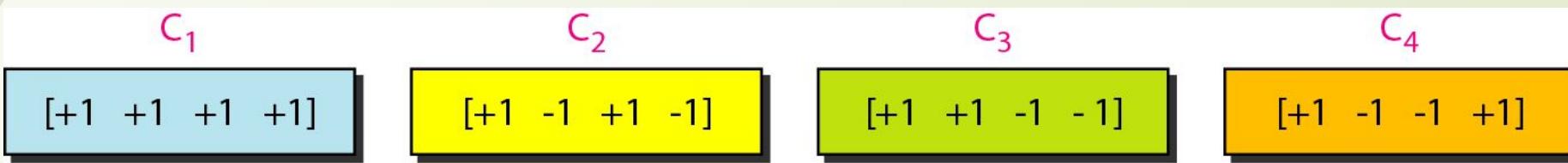


Figure 7.25 *Data representation in CDMA*

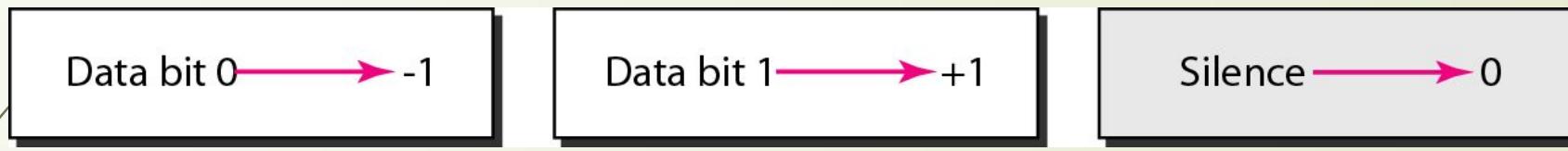


Figure 7.26 Sharing channel in CDMA

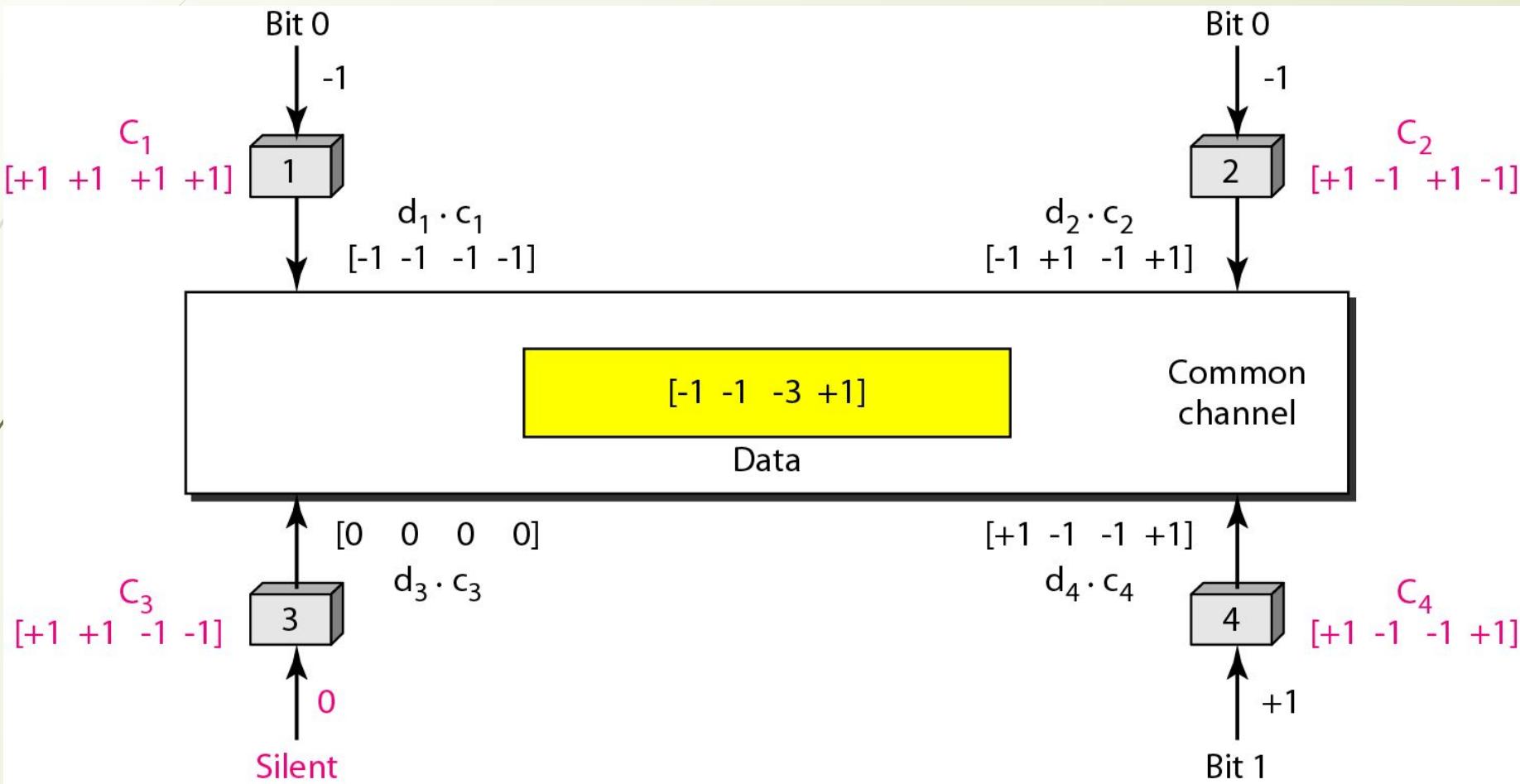


Figure 7.27 Digital signal created by four stations in CDMA

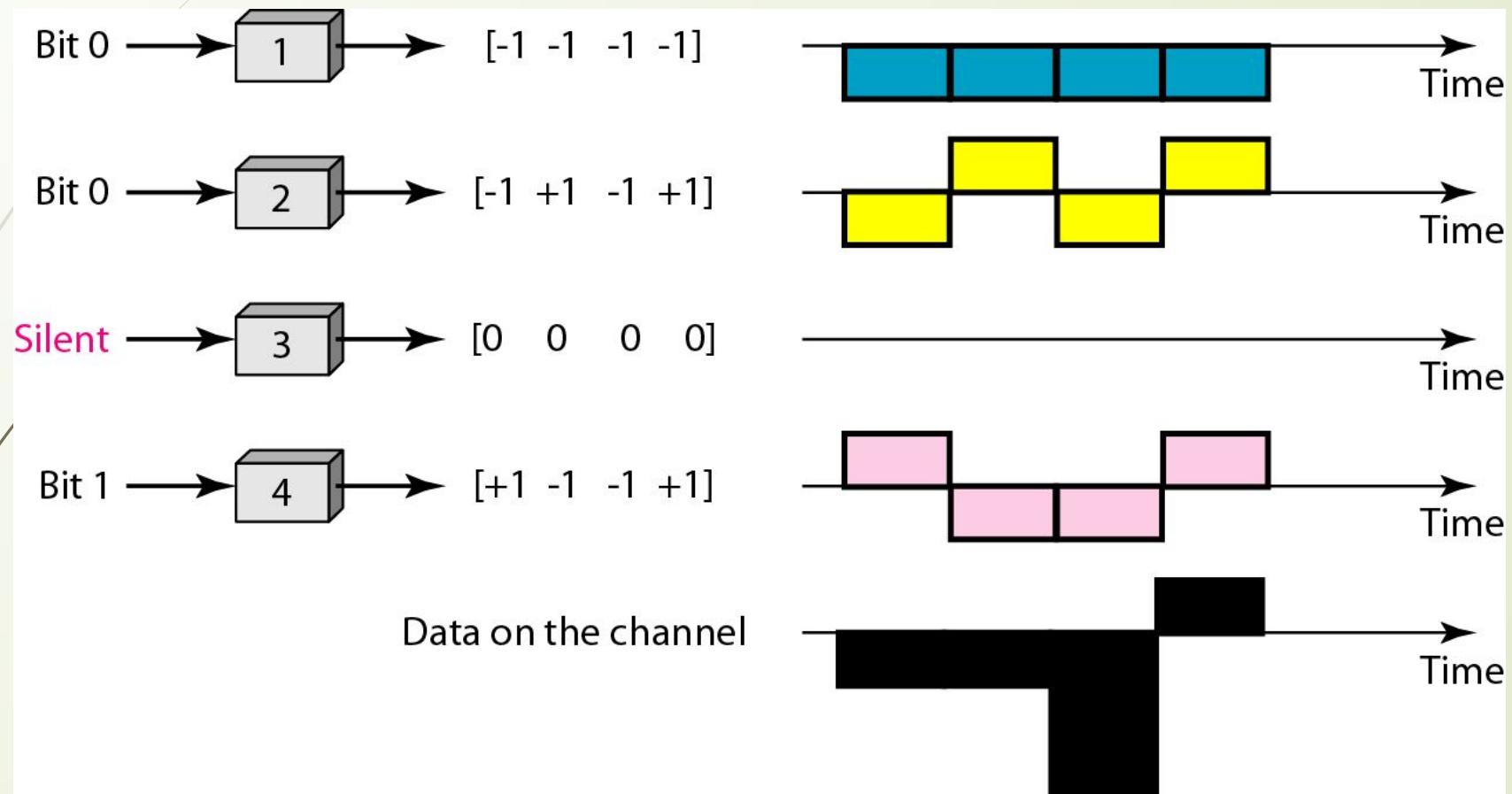


Figure 7.28 Decoding of the composite signal for one in CDMA

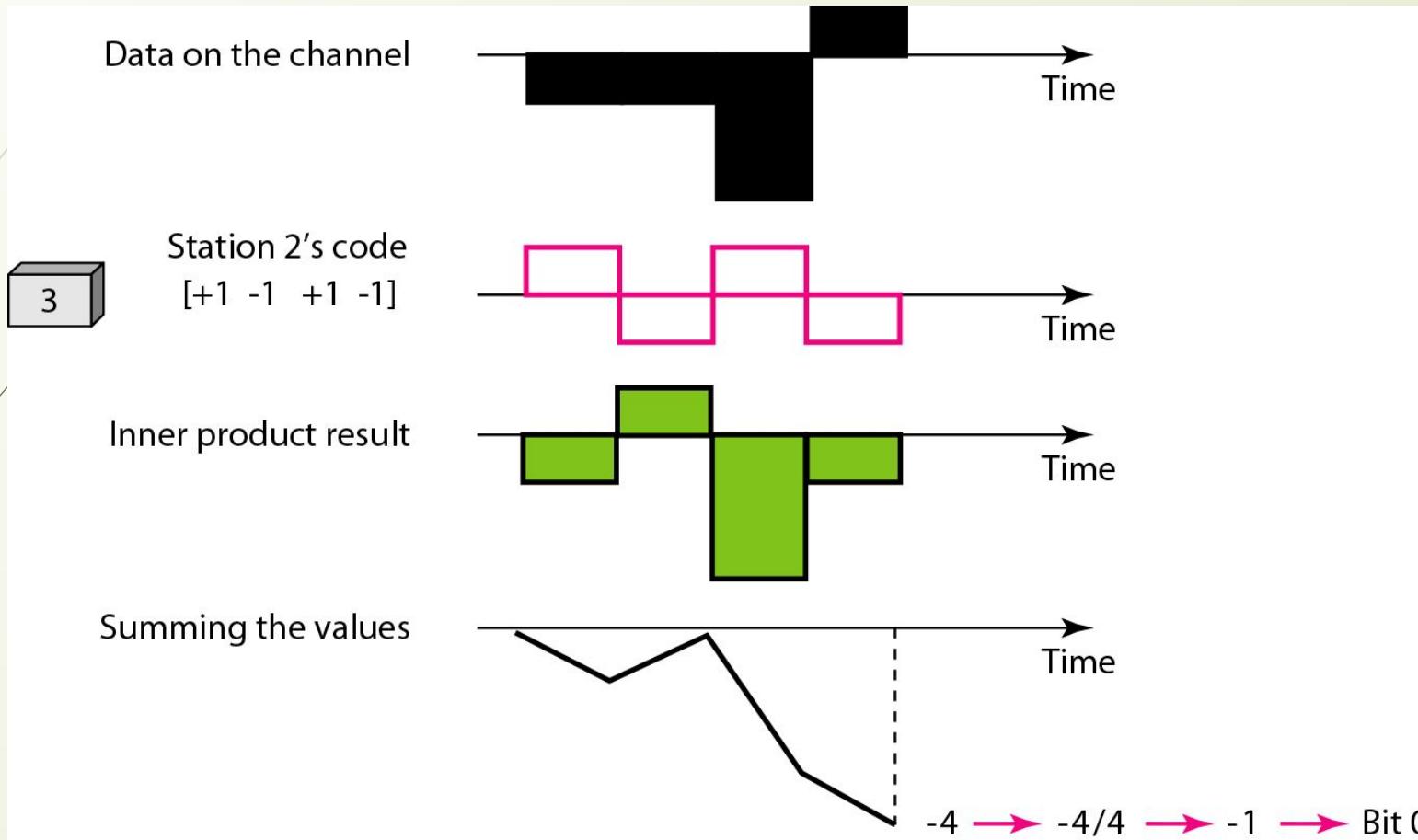


Figure 7.29 General rule and examples of creating Walsh tables

$$W_1 = \begin{bmatrix} +1 \end{bmatrix}$$

$$W_{2N} = \begin{bmatrix} W_N & W_N \\ W_N & \overline{W}_N \end{bmatrix}$$

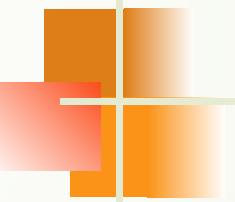
a. Two basic rules

$$W_1 = \begin{bmatrix} +1 \end{bmatrix}$$

$$W_2 = \begin{bmatrix} +1 & +1 \\ +1 & -1 \end{bmatrix}$$

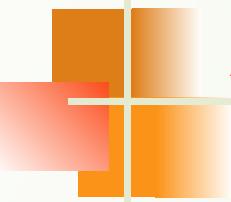
$$W_4 = \begin{bmatrix} +1 & +1 & +1 & +1 \\ +1 & -1 & +1 & -1 \\ +1 & +1 & -1 & -1 \\ +1 & -1 & -1 & +1 \end{bmatrix}$$

b. Generation of W_1 , W_2 , and W_4



Note

The number of sequences in a Walsh table needs to be $N = 2^m$.



Example 7.6

Find the chips for a network with

- a. Two stations**
- b. Four stations**

Solution

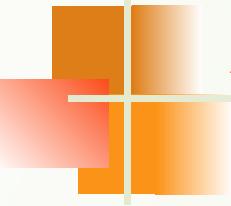
We can use the rows of W_2 and W_4 in Figure 12.29:

- a. For a two-station network, we have**

$$[+1 \ +1] \text{ and } [+1 \ -1].$$

- b. For a four-station network we have**

$$[+1 \ +1 \ +1 \ +1], \ [+1 \ -1 \ +1 \ -1], \\ [+1 \ +1 \ -1 \ -1], \text{ and } [+1 \ -1 \ -1 \ +1].$$

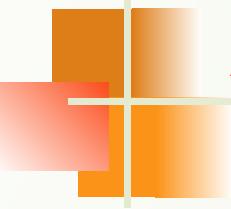


Example 7.7

What is the number of sequences if we have 90 stations in our network?

Solution

The number of sequences needs to be 2^m . We need to choose $m = 7$ and $N = 2^7$ or 128. We can then use 90 of the sequences as the chips.



Example 7.8

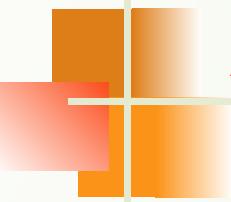
Prove that a receiving station can get the data sent by a specific sender if it multiplies the entire data on the channel by the sender's chip code and then divides it by the number of stations.

Solution

Let us prove this for the first station, using our previous four-station example. We can say that the data on the channel

$$D = (d_1 \cdot c_1 + d_2 \cdot c_2 + d_3 \cdot c_3 + d_4 \cdot c_4).$$

The receiver which wants to get the data sent by station 1 multiplies these data by c_1



Example 7.8 (continued)

$$\begin{aligned}D \cdot c_1 &= (d_1 \cdot c_1 + d_2 \cdot c_2 + d_3 \cdot c_3 + d_4 \cdot c_4) \cdot c_1 \\&= d_1 \cdot c_1 \cdot c_1 + d_2 \cdot c_2 \cdot c_1 + d_3 \cdot c_3 \cdot c_1 + d_4 \cdot c_4 \cdot c_1 \\&= d_1 \times N + d_2 \times 0 + d_3 \times 0 + d_4 \times 0 \\&= d_1 \times N\end{aligned}$$

When we divide the result by N , we get d_1 .



HDLC

- **High-Level Data Link Control (HDLC)** is a bit-oriented synchronous data link layer protocol developed by the International Organization for Standardization (ISO). It is one of the most fundamental protocols in the field of data communication and networking, forming the basis for many other network protocols.

Key Features of HDLC

1. **Bit-Oriented Protocol:** HDLC is a bit-oriented protocol, meaning it considers data as a continuous stream of bits rather than as discrete characters. This allows HDLC to transmit any type of binary data efficiently.
2. **Synchronous Communication:** HDLC is used for synchronous communication, where the sender and receiver are synchronized in time, meaning data is transmitted at a constant rate, with both ends sharing the same clock signal.
3. **Error Detection and Correction:** HDLC provides mechanisms for detecting and possibly correcting errors in the data transmission process. It uses a cyclic redundancy check (CRC) to ensure data integrity.

Frame Structure of HDLC

1. **Frame Structure:** HDLC encapsulates data into frames, which are the basic units of communication. Each frame consists of several fields:
 - **Flag Field:** This is the start and end delimiter of a frame, consisting of the bit pattern 01111110 (0x7E). The flag indicates the beginning and end of a frame.
 - **Address Field:** This identifies the address of the secondary station or destination.
 - **Control Field:** This field is used for flow and error control. It indicates the type of frame and carries sequence numbers for proper ordering of frames.
 - **Information Field:** This field carries the actual user data. The length of this field is variable, depending on the amount of data being transmitted.
 - **Frame Check Sequence (FCS):** This is a CRC field used for error detection.

Types of HDLC Frames

HDLC uses three types of frames, each serving a different purpose:

1. Information (I) Frames:

1. These frames are used for transmitting user data and can carry both data and control information.
2. The Control field of I-frames includes a sequence number that helps in ensuring proper data sequencing and error correction.

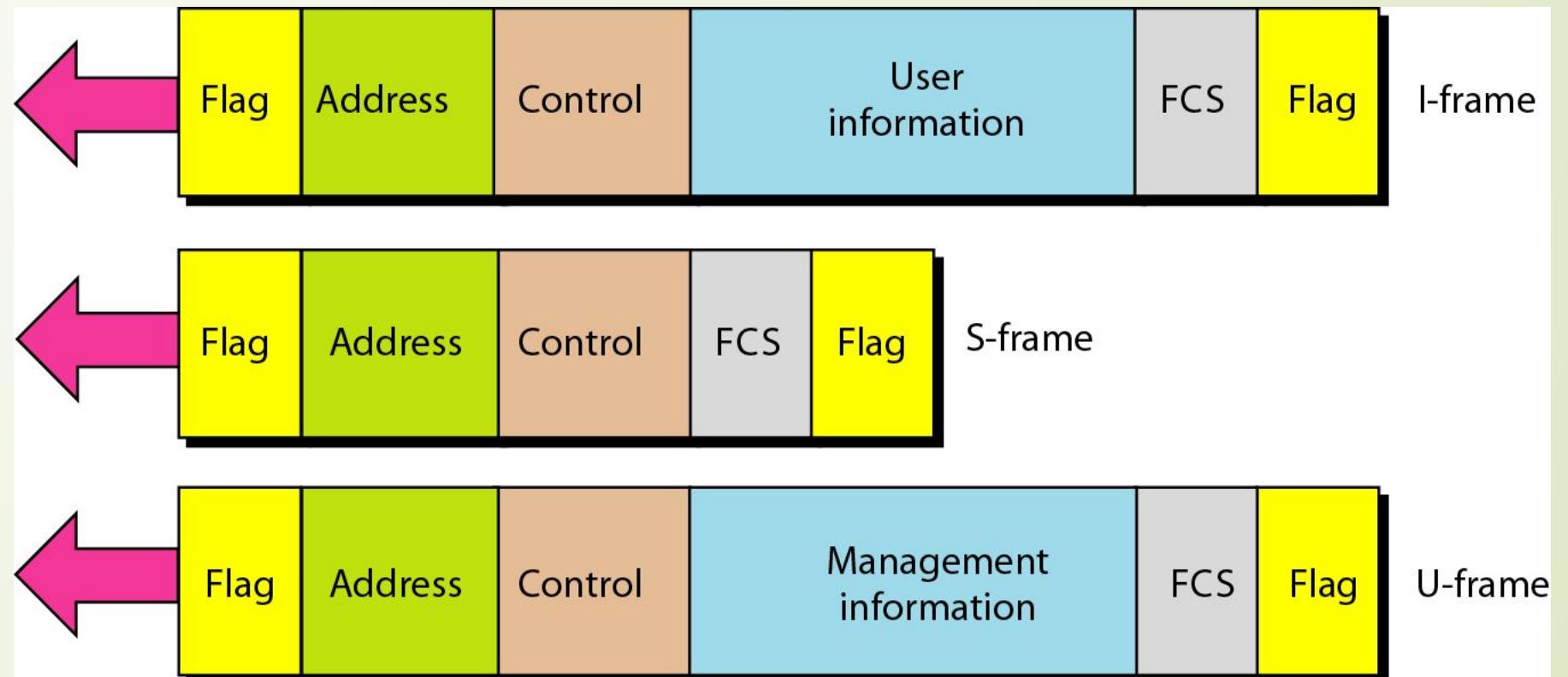
2. Supervisory (S) Frames:

1. These frames are used for controlling the flow of data and error recovery.
2. They do not carry user data but include control information, such as acknowledgment of received frames, requests for retransmission, and flow control commands.

3. Unnumbered (U) Frames:

1. U-frames are used for various network management tasks, such as establishing or terminating a connection, resetting the link, and exchanging control information without sequence numbering.

HDLC frames



HDLC Operation Modes

HDLC supports different operational modes, depending on the nature of the communication:

1. Normal Response Mode (NRM):

1. In this mode, communication is controlled by a primary station, which initiates communication with secondary stations. Secondary stations can only respond to requests from the primary station.
2. This is typically used in multi-point configurations, where a central controller communicates with multiple devices.

2. Asynchronous Balanced Mode (ABM):

1. ABM allows both stations to initiate communication independently, making it a balanced mode of operation.
2. This mode is often used in point-to-point communication, where two stations can send and receive data simultaneously.

3. Asynchronous Response Mode (ARM):

1. Similar to NRM, but the secondary stations can initiate communication without waiting for the primary station. However, the primary station can still control the line.
2. This mode is rarely used.

Normal Response Mode

- **Normal Response Mode (NRM)** is typically used in configurations where a central device (called the **primary station**) controls communication with other devices (called **secondary stations**). The communication is unidirectional from the primary station to the secondary station, and the secondary station can only send data when explicitly allowed (polled) by the primary station.
- **Characteristics:**
 - **Master-Slave Relationship:** In NRM, the **primary station** is the master, while the **secondary stations** are slaves. The primary station initiates all communication.
 - **Polling Mechanism:** The secondary station cannot send data spontaneously. It must wait for the primary station to poll or request data. When polled, the secondary station responds by sending data.
 - **Flow Control:** Since the primary station controls all communication, it can manage the flow of data and ensure orderly transmission.
 - **Error Control:** The primary station is responsible for handling any errors and retransmitting data if needed.

Normal Response Mode

□ Use Case:

- **Multipoint Networks:** NRM is commonly used in **multipoint networks** where multiple secondary devices are connected to a single primary station. For example, legacy **IBM mainframe networks** (Synchronous Data Link Control, or SDLC) often used NRM for managing communication with terminals.

□ Advantages:

- Simplified communication for centralized systems.
- The primary station has full control over the network.

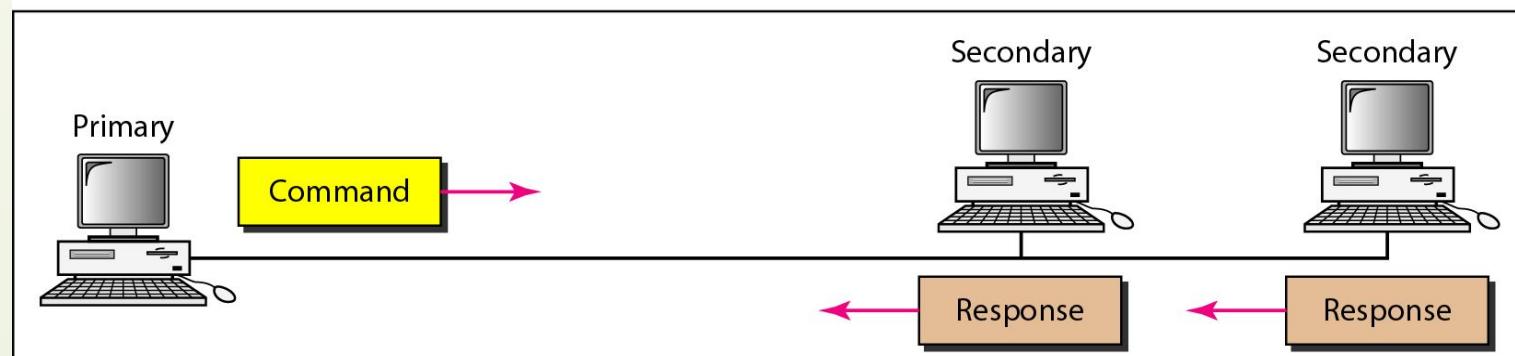
□ Disadvantages:

- Secondary stations are passive and must wait for the primary station's permission to send data, which can lead to inefficient use of network resources.

Normal response mode



a. Point-to-point



b. Multipoint

Asynchronous balanced mode

- **Asynchronous Balanced Mode (ABM)** is the most flexible and widely used mode of HDLC. In ABM, both devices on the link (known as **stations**) have equal status, meaning either can initiate communication. ABM allows for full-duplex, peer-to-peer communication without the need for a master-slave relationship.
- **Characteristics:**
 - **Balanced Relationship:** Both stations (peers) can initiate data transmission independently without the need for a polling mechanism.
 - **Full-Duplex Communication:** ABM allows both stations to send and receive data simultaneously, enabling more efficient communication.
 - **Error and Flow Control:** Both stations are responsible for error detection, acknowledgment, and retransmission of frames. This makes the link more robust and decentralized in terms of control.
 - **Simplicity in Peer Communication:** Since there is no primary/secondary distinction, ABM simplifies peer-to-peer communication, making it suitable for modern, bidirectional network links.

Asynchronous balanced mode

□ Use Case:

- **Point-to-Point Links:** ABM is commonly used in **point-to-point** communication systems where two devices need to exchange data without a central controller. It is widely used in WAN protocols like **X.25**, **Frame Relay**, and modern **PPP (Point-to-Point Protocol)**, which evolved from HDLC.

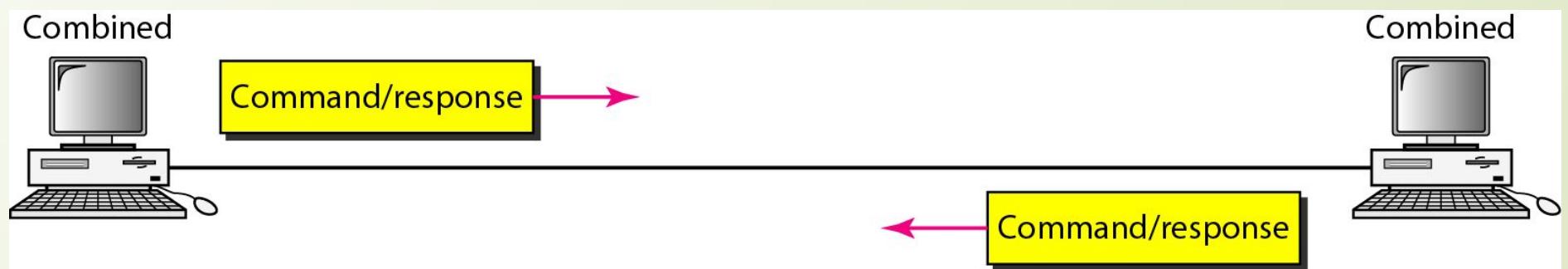
□ Advantages:

- Provides flexibility, allowing any station to initiate communication, resulting in higher efficiency.
- Supports full-duplex communication, maximizing link utilization.

□ Disadvantages:

- Requires more complex error and flow control mechanisms compared to NRM or ARM, since both stations are involved in managing the link.

Asynchronous balanced mode



Asynchronous Response mode

- **Asynchronous Response Mode (ARM)** also follows a **master-slave model**, similar to NRM, but with one key difference: the secondary stations can initiate communication with the primary station without being explicitly polled. However, the primary station still has control over the link and can manage data flow.
- **Characteristics:**
 - **Master-Slave Relationship:** Like NRM, the primary station has overall control, but the secondary stations are allowed more flexibility in initiating communication.
 - **Unsolicited Responses:** Secondary stations do not need to wait for a poll from the primary station to send data. They can start communication whenever they need to, though the primary station can still manage or restrict their responses.
 - **Control of Communication:** Although secondary stations can initiate communication, the primary station remains responsible for overall control of the link, including error recovery, flow control, and link management.

Asynchronous Response mode

□ Use Case:

- **Legacy Networks:** ARM was not widely adopted but can be used in systems where the secondary stations occasionally need to send data without waiting for permission, but the network design still requires a central controller.

□ Advantages:

- Allows secondary stations to initiate communication, increasing efficiency compared to NRM.

□ Disadvantages:

- The mode is rarely used in modern networks and can be less efficient than ABM due to the continued master-slave relationship.

Error Handling in HDLC

HDLC employs several mechanisms for error detection and correction:

- **Cyclic Redundancy Check (CRC):** Each frame includes a CRC code in the FCS field. The CRC code is calculated from the frame's content. Upon receiving the frame, the receiver recalculates the CRC and compares it with the one in the FCS field. If the two match, the frame is considered error-free.
- **Acknowledgments and Retransmissions:** HDLC uses acknowledgments (in S-frames) to confirm the successful receipt of frames. If a frame is not acknowledged, it is assumed to be lost or corrupted and is retransmitted.

Applications of HDLC

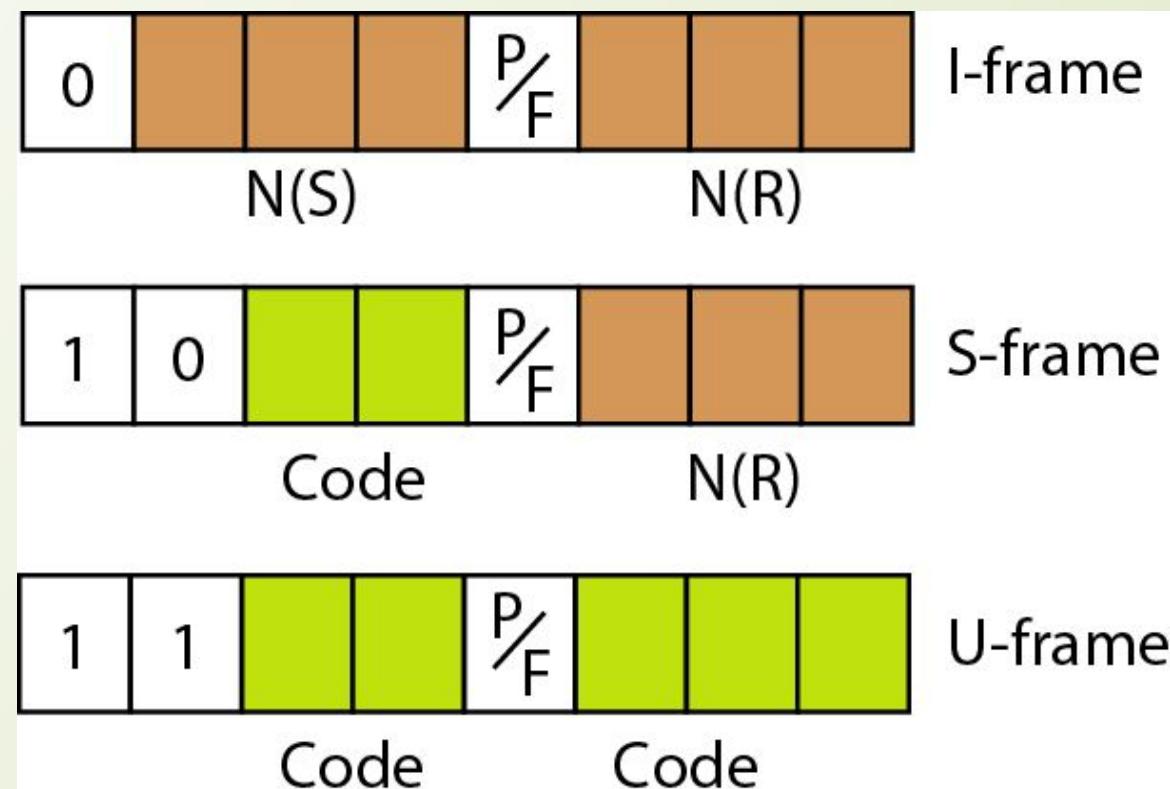
- HDLC is widely used in various communication systems, including:
 - **WAN (Wide Area Network) Protocols:** HDLC is used in point-to-point communication links in WAN protocols like X.25, Frame Relay, and ISDN.
 - **PPP (Point-to-Point Protocol):** PPP, used in many internet connections, is based on HDLC. However, PPP includes additional features like authentication and network layer protocol multiplexing.
 - **Embedded Systems:** HDLC is used in communication between devices in embedded systems due to its robustness and simplicity.



Note

- HDLC is not limited to **wired communication**; it can also be used in **wireless communication** environments. While HDLC was initially designed for point-to-point and multipoint connections over **synchronous serial links**, which are often wired (such as in leased lines, X.25 networks, or Frame Relay), its principles and mechanisms have been adapted for use in various communication systems, including **wireless networks**.

Control field format for the different frame types



U-frame control command and response

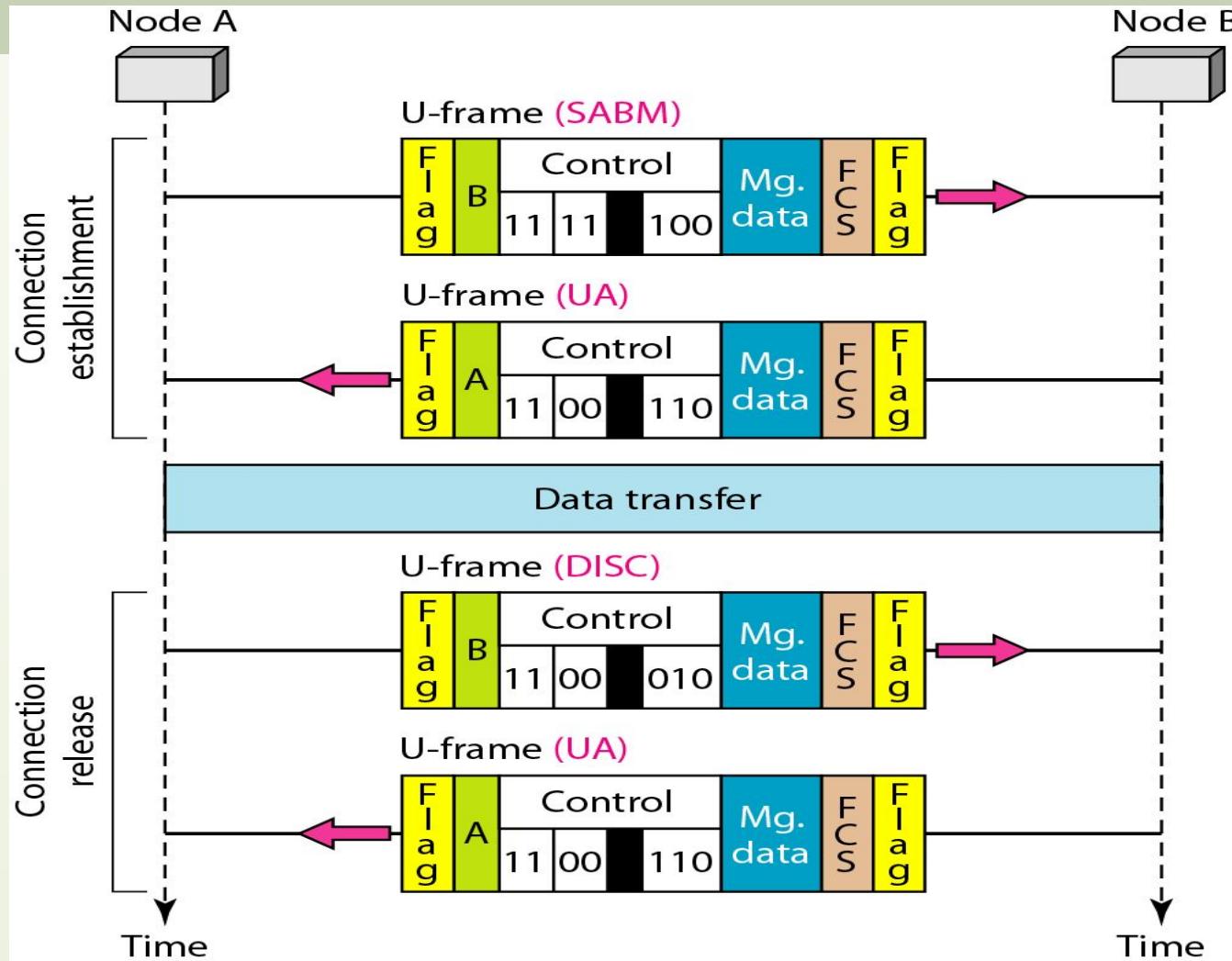
<i>Code</i>	<i>Command</i>	<i>Response</i>	<i>Meaning</i>
00 001	SNRM		Set normal response mode
11 011	SNRME		Set normal response mode, extended
11 100	SABM	DM	Set asynchronous balanced mode or disconnect mode
11 110	SABME		Set asynchronous balanced mode, extended
00 000	UI	UI	Unnumbered information
00 110		UA	Unnumbered acknowledgment
00 010	DISC	RD	Disconnect or request disconnect
10 000	SIM	RIM	Set initialization mode or request information mode
00 100	UP		Unnumbered poll
11 001	RSET		Reset
11 101	XID	XID	Exchange ID
10 001	FRMR	FRMR	Frame reject



Example

- Figure 4.8 shows how **U-frames** can be used for connection establishment and connection release. Node A asks for a connection with a set asynchronous balanced mode (SABM) frame; node B gives a positive response with an unnumbered acknowledgment (UA) frame. After these two exchanges, data can be transferred between the two nodes (not shown in the figure). After data transfer, node A sends a DISC (disconnect) frame to release the connection; it is confirmed by node B responding with a UA (unnumbered acknowledgment).

Figure 4.8 Example of connection and disconnection





EXAMPLE

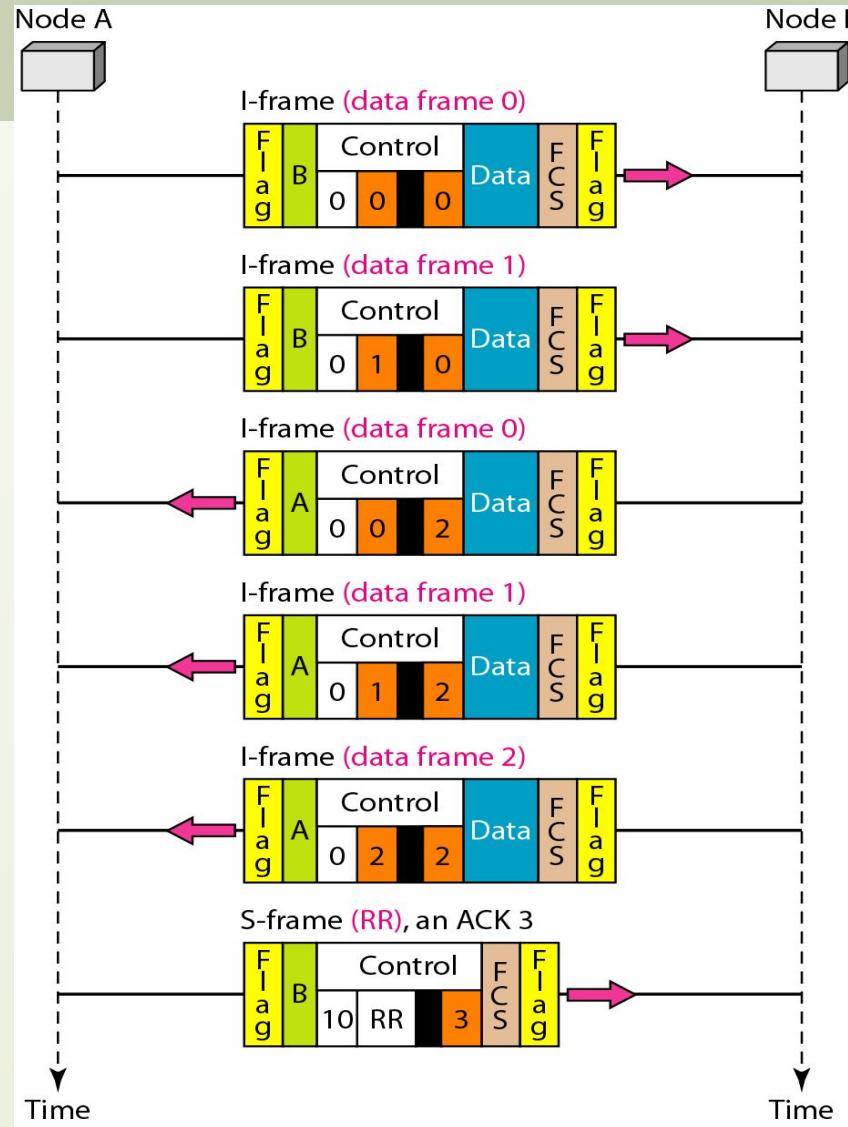
- Figure 4.9 shows an exchange using piggybacking. Node A begins the exchange of information with an I-frame numbered 0 followed by another I-frame numbered 1. Node B piggybacks its acknowledgment of both frames onto an I-frame of its own. Node B's first I-frame is also numbered 0 [N(S) field] and contains a 2 in its N(R) field, acknowledging the receipt of A's frames 1 and 0 and indicating that it expects frame 2 to arrive next. Node B transmits its second and third I-frames (numbered 1 and 2) before accepting further frames from node A.



CONTINUED

- Its N(R) information, therefore, has not changed: B frames 1 and 2 indicate that node B is still expecting A's frame 2 to arrive next. Node A has sent all its data. Therefore, it cannot piggyback an acknowledgment onto an I-frame and sends an S-frame instead. The RR code indicates that A is still ready to receive. The number 3 in the N(R) field tells B that frames 0, 1, and 2 have all been accepted and that A is now expecting frame number 3.

Figure 4.9 Example of piggybacking without error

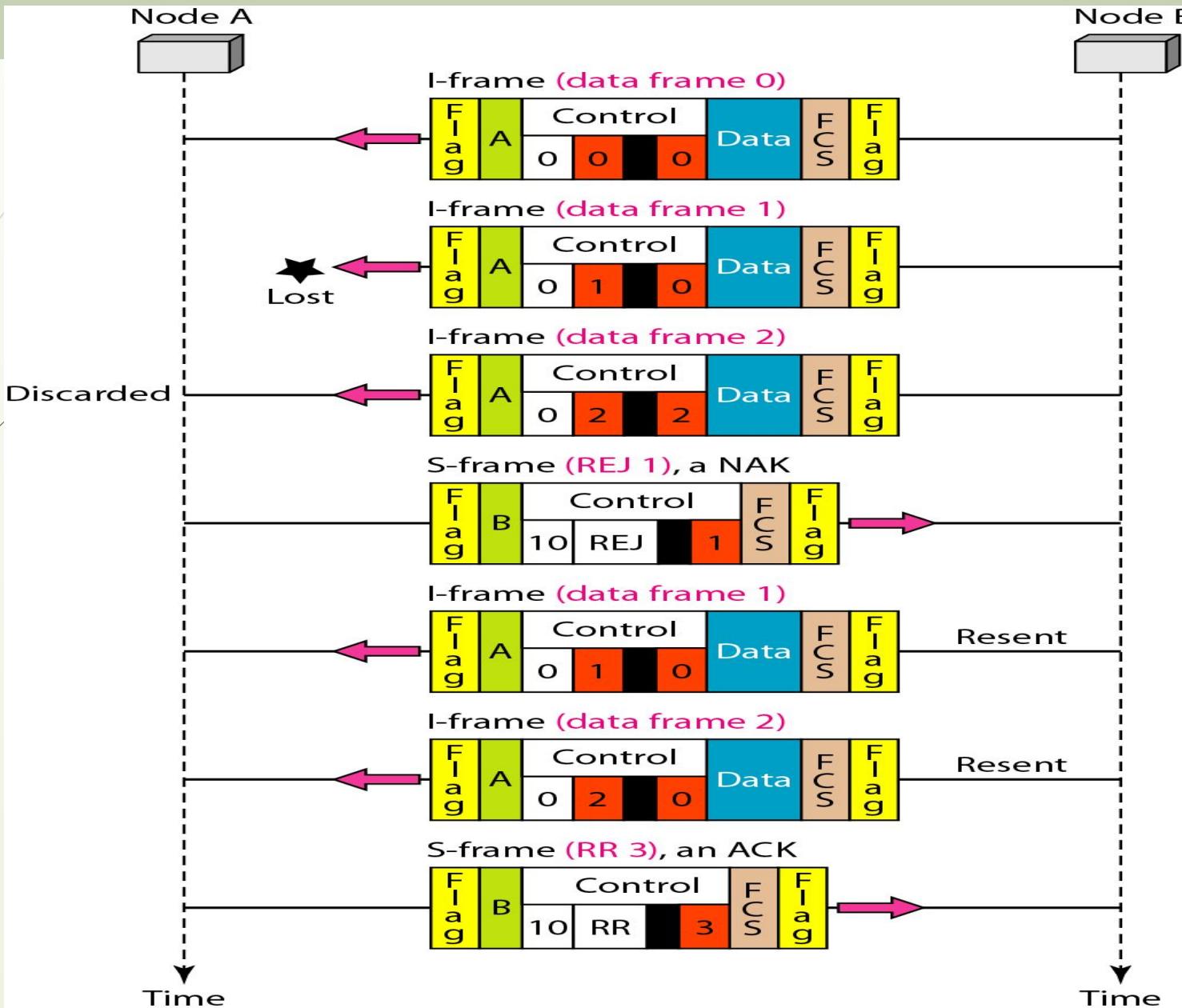


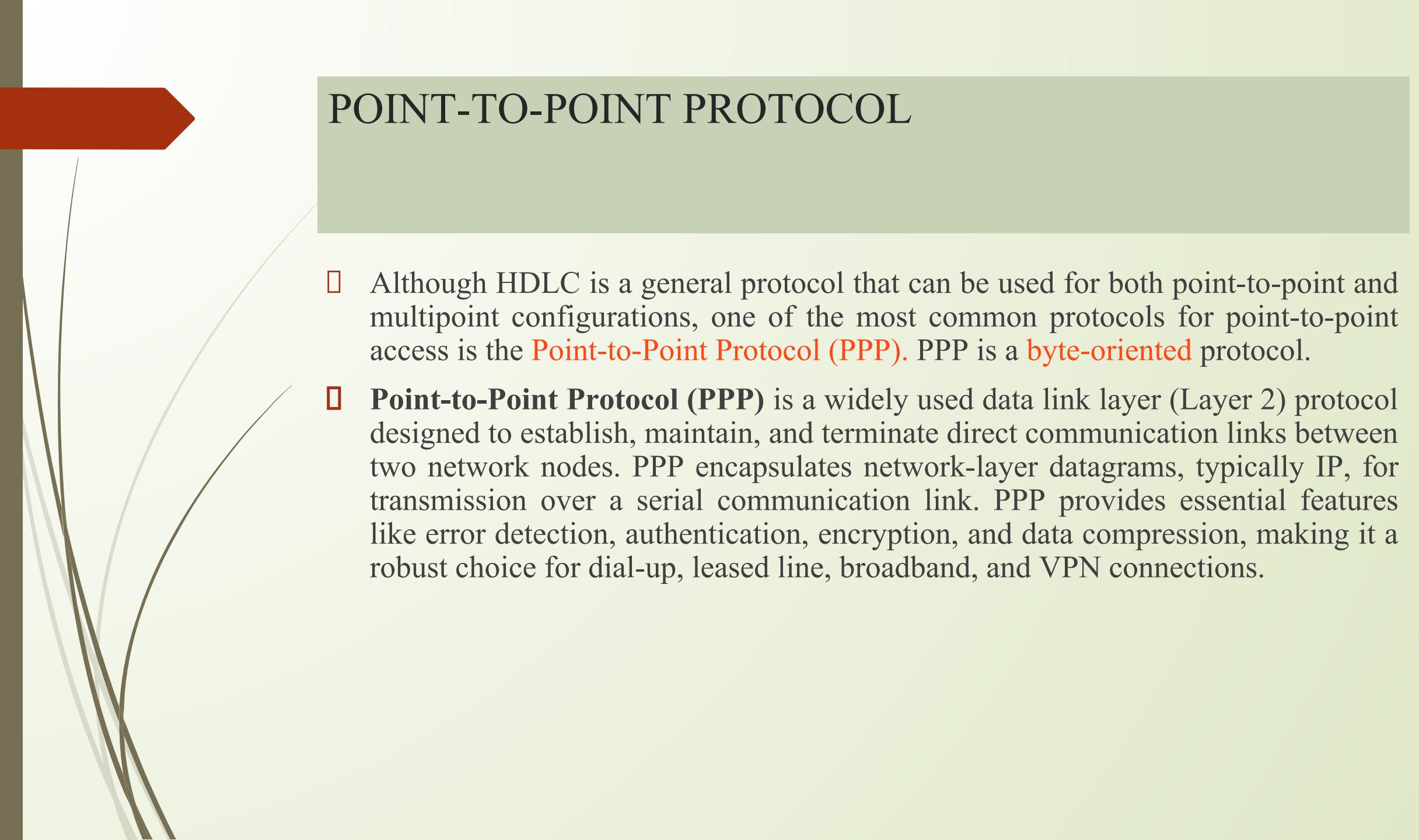


EXAMPLE

- Figure 4.10 shows an exchange in which a frame is lost. Node B sends three data frames (0, 1, and 2), but frame 1 is lost. When node A receives frame 2, it discards it and sends a REJ frame for frame 1. Note that the protocol being used is Go-Back-N with the special use of an REJ frame as a NAK frame. The NAK frame does two things here: It confirms the receipt of frame 0 and declares that frame 1 and any following frames must be resent. Node B, after receiving the REJ frame, resends frames 1 and 2. Node A acknowledges the receipt by sending an RR frame (ACK) with acknowledgment number 3.

Figure 4.10 Example of piggybacking with error





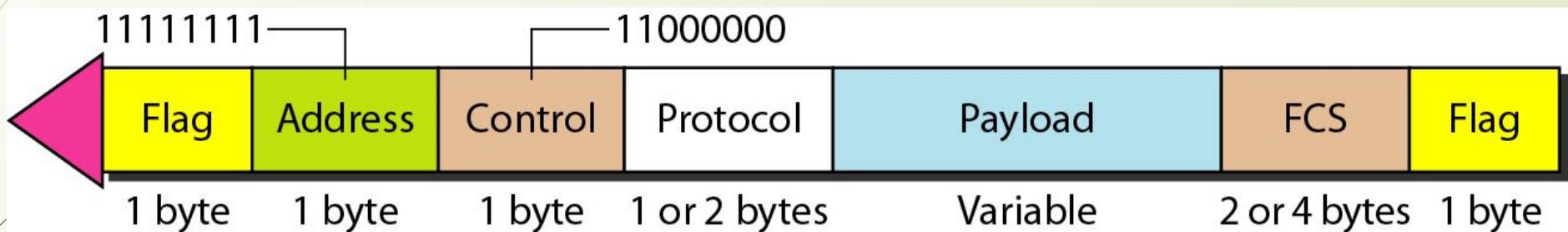
POINT-TO-POINT PROTOCOL

- Although HDLC is a general protocol that can be used for both point-to-point and multipoint configurations, one of the most common protocols for point-to-point access is the **Point-to-Point Protocol (PPP)**. PPP is a **byte-oriented** protocol.
- **Point-to-Point Protocol (PPP)** is a widely used data link layer (Layer 2) protocol designed to establish, maintain, and terminate direct communication links between two network nodes. PPP encapsulates network-layer datagrams, typically IP, for transmission over a serial communication link. PPP provides essential features like error detection, authentication, encryption, and data compression, making it a robust choice for dial-up, leased line, broadband, and VPN connections.

PPP Architecture and Frame Structure

- PPP is made up of three main components:
 1. **Encapsulation:** Defines how data is encapsulated in a PPP frame for transmission.
 2. **Link Control Protocol (LCP):** Negotiates, configures, and manages the connection.
 3. **Network Control Protocol (NCP):** Negotiates and configures the network layer protocol used (e.g., IP, IPv6).

PPP frame format



Flag (1 byte): Marks the beginning and end of a PPP frame. The flag is always set to 0x7E (01111110).

Address (1 byte): Since PPP is used in point-to-point links, this field is always set to 0xFF (broadcast address).

Control (1 byte): Typically set to 0x03, representing an unnumbered information frame (UI) without flow control.

Protocol (1 or 2 bytes): Identifies the type of payload in the information field (e.g., IP, IPv6, Link Control Protocol, etc.).

Information: Contains the actual data being transmitted. The size of this field can vary.

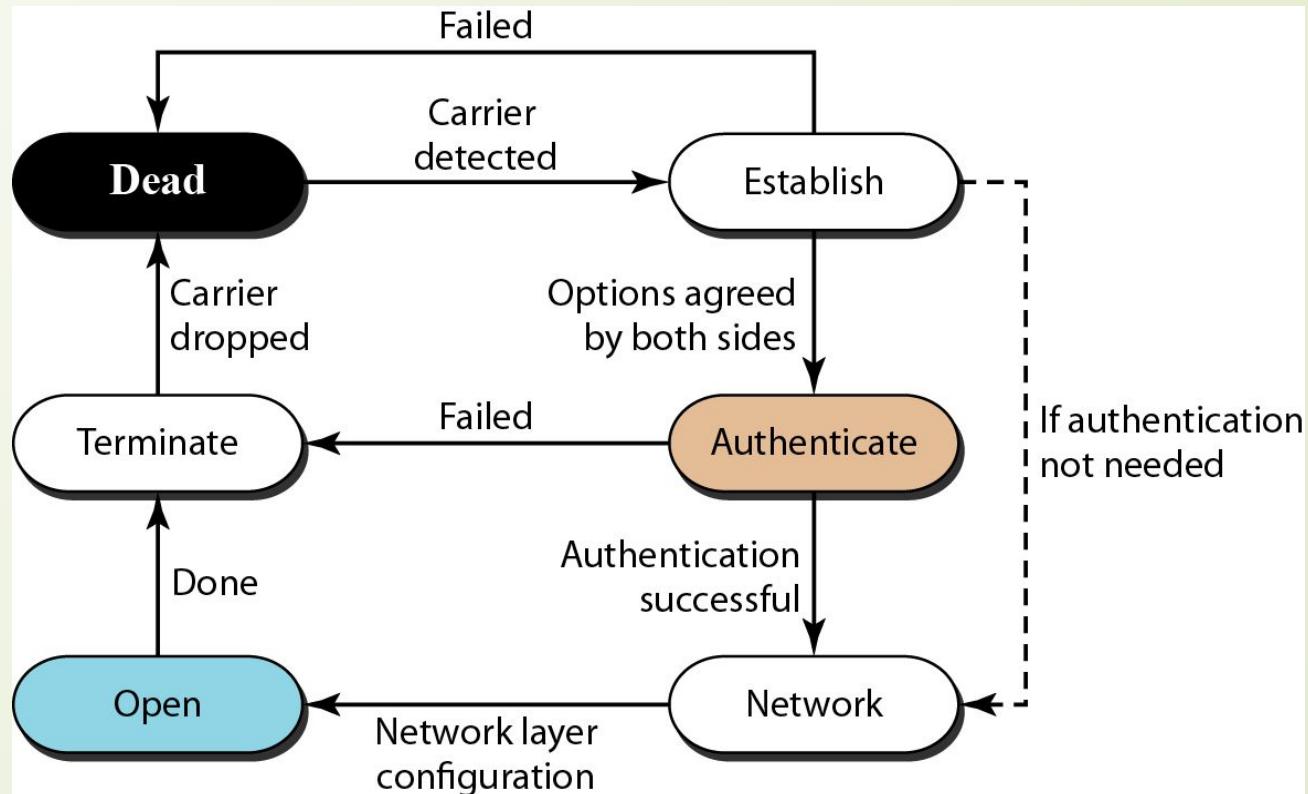
Frame Check Sequence (FCS) (2 or 4 bytes): Provides a CRC (Cyclic Redundancy Check) value for error detection.



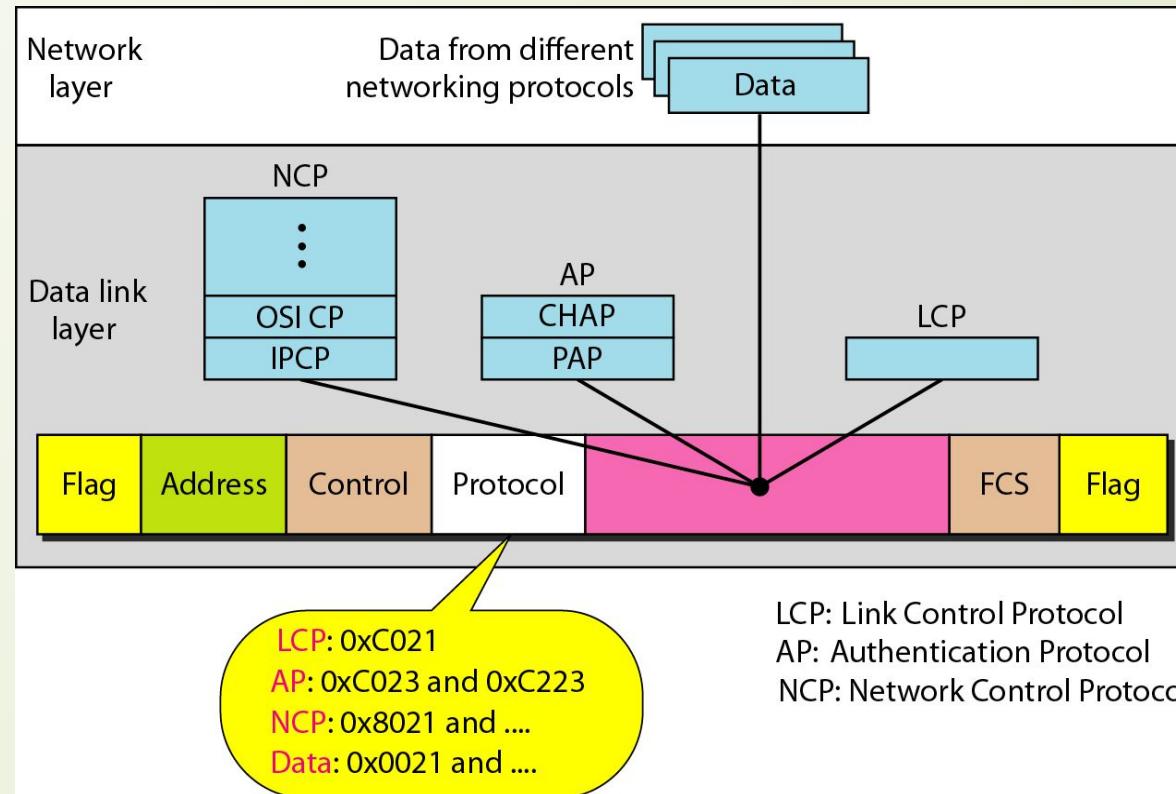
Note

□ PPP is a byte-oriented protocol using byte stuffing with the escape byte 01111101.

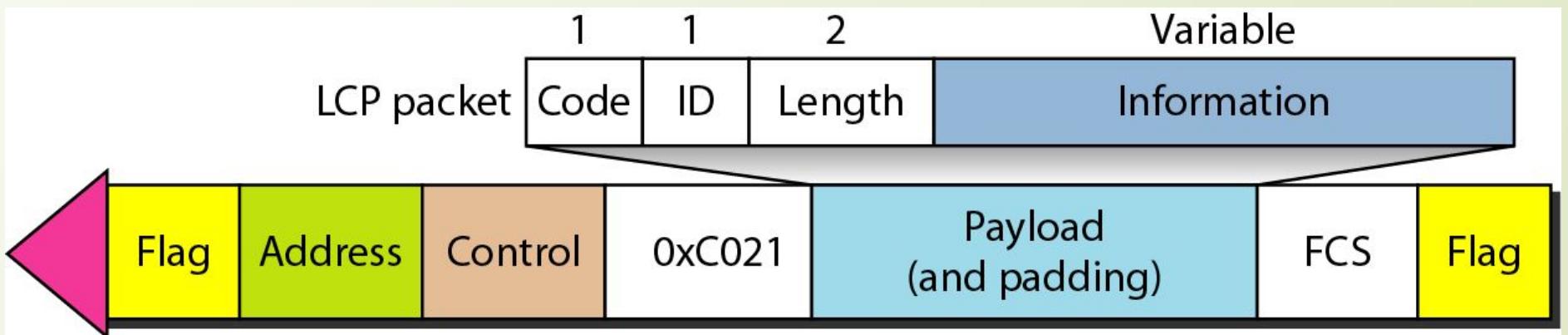
Transition phases



Multiplexing in PPP

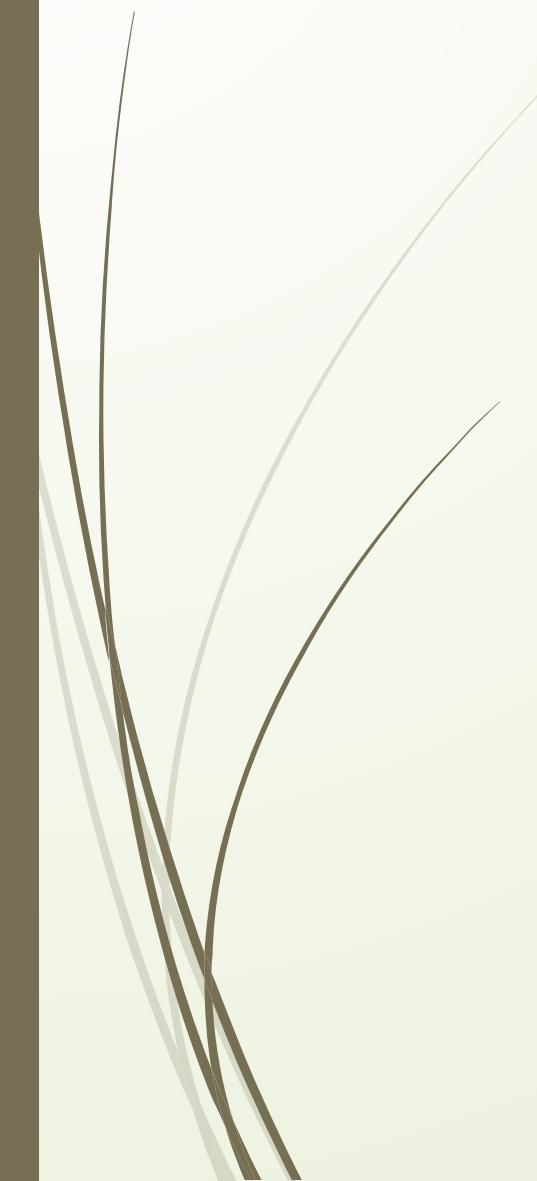


LCP packet encapsulated in a frame



LCP packets

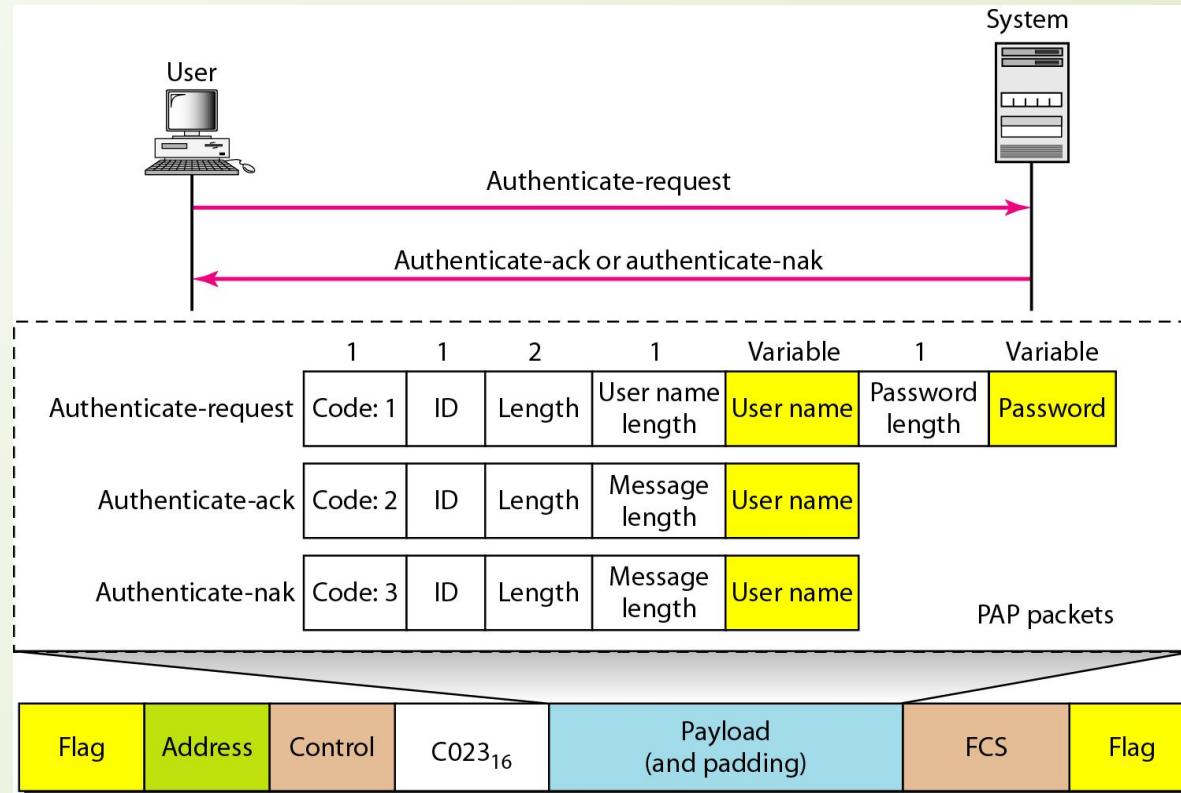
<i>Code</i>	<i>Packet Type</i>	<i>Description</i>
0x01	Configure-request	Contains the list of proposed options and their values
0x02	Configure-ack	Accepts all options proposed
0x03	Configure-nak	Announces that some options are not acceptable
0x04	Configure-reject	Announces that some options are not recognized
0x05	Terminate-request	Request to shut down the line
0x06	Terminate-ack	Accept the shutdown request
0x07	Code-reject	Announces an unknown code
0x08	Protocol-reject	Announces an unknown protocol
0x09	Echo-request	A type of hello message to check if the other end is alive
0x0A	Echo-reply	The response to the echo-request message
0x0B	Discard-request	A request to discard the packet



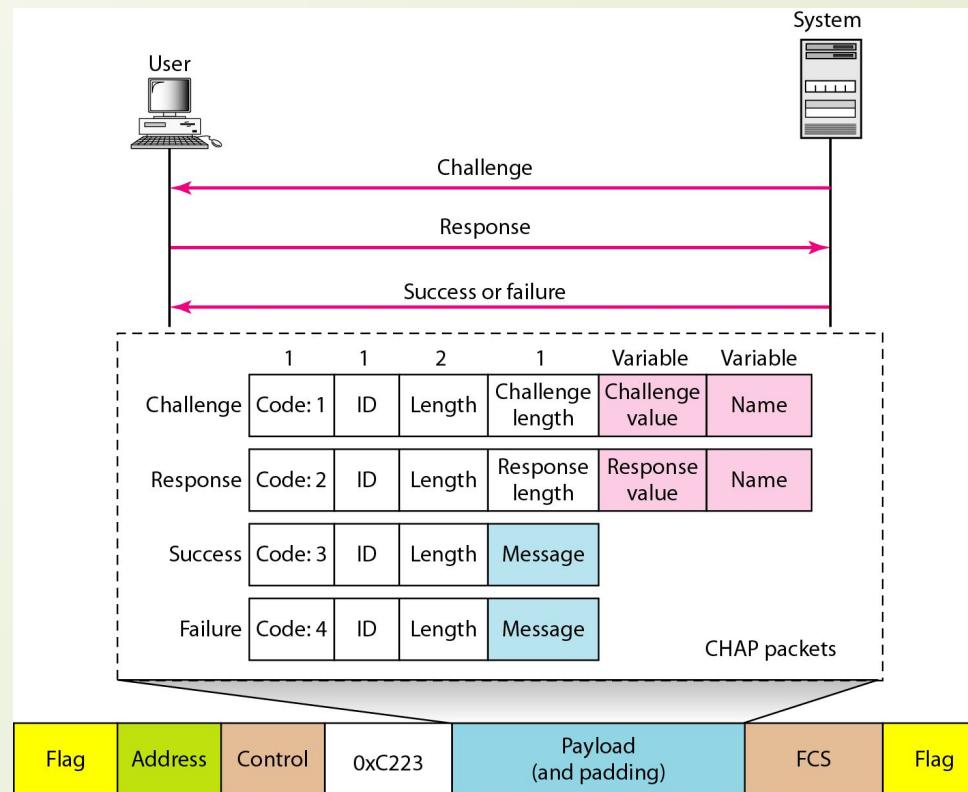
Common options

<i>Option</i>	<i>Default</i>
Maximum receive unit (payload field size)	1500
Authentication protocol	None
Protocol field compression	Off
Address and control field compression	Off

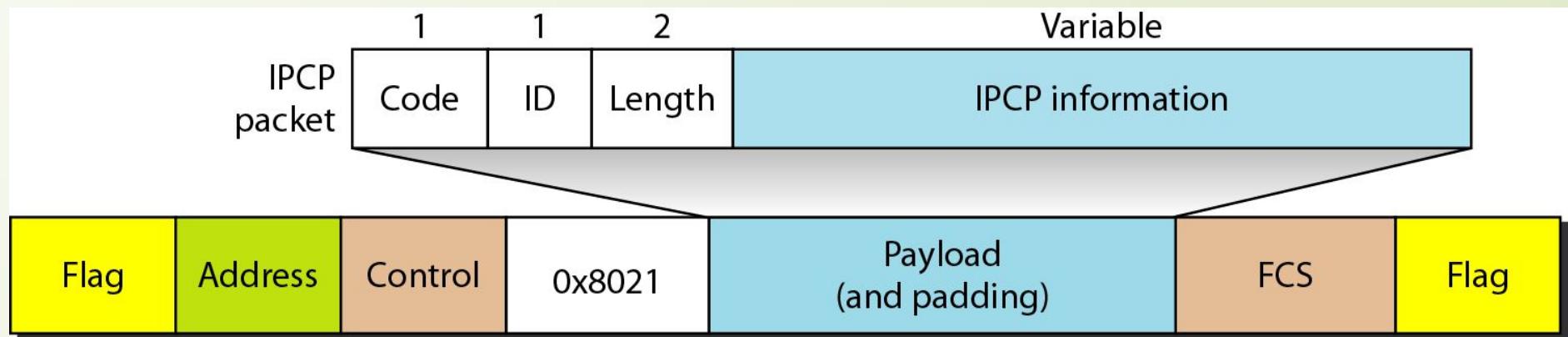
PAP packets encapsulated in a PPP frame



CHAP packets encapsulated in a PPP frame



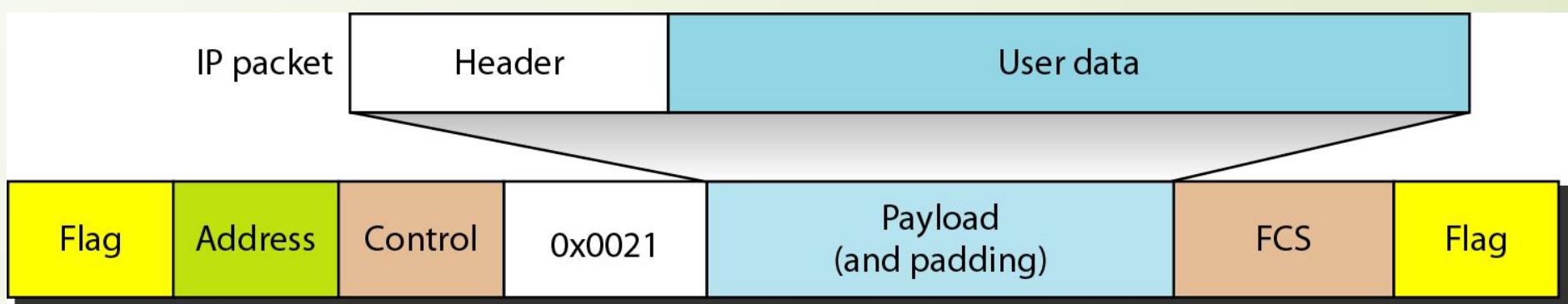
IPCP packet encapsulated in PPP frame



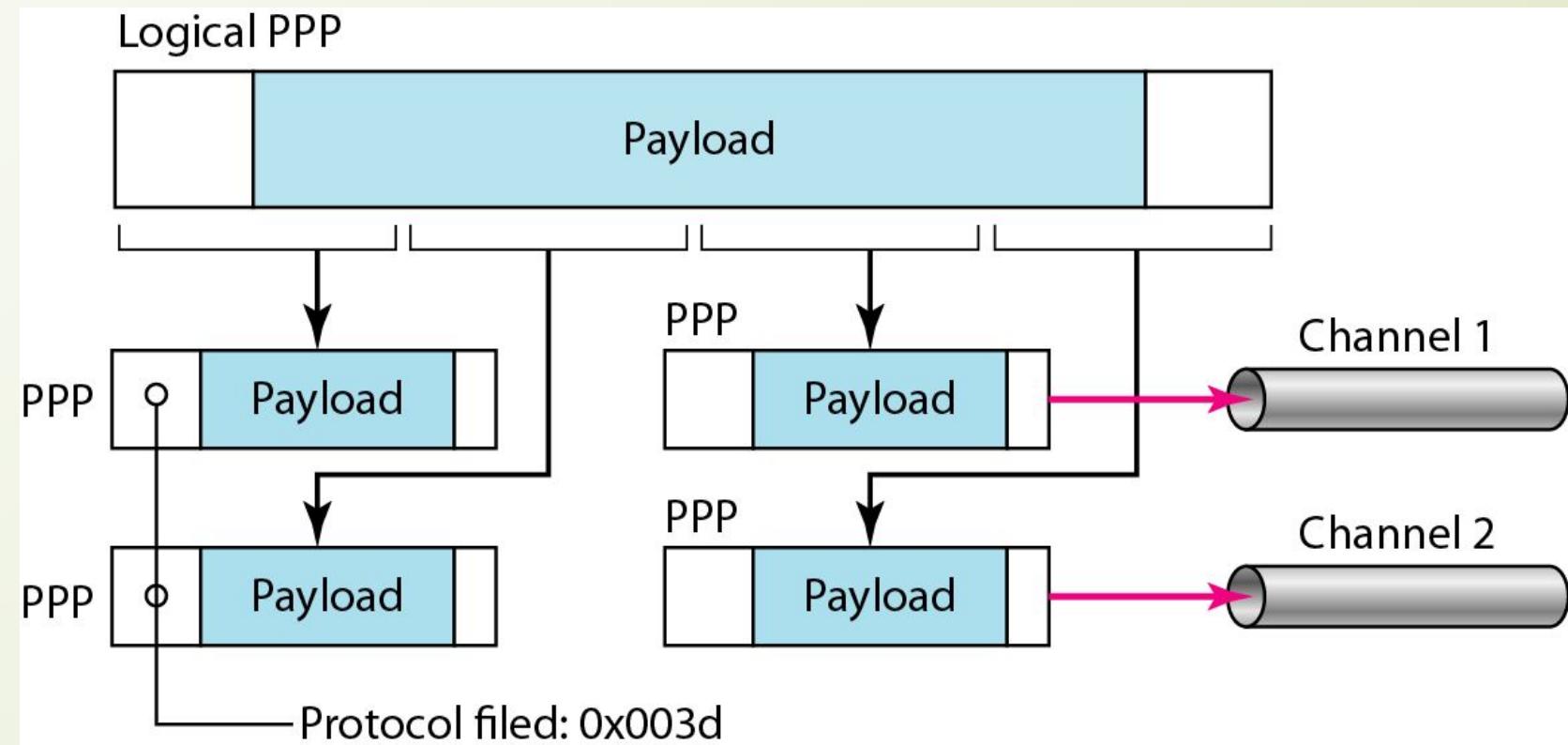
Code value for IPCP packets

<i>Code</i>	<i>IPCP Packet</i>
0x01	Configure-request
0x02	Configure-ack
0x03	Configure-nak
0x04	Configure-reject
0x05	Terminate-request
0x06	Terminate-ack
0x07	Code-reject

IP datagram encapsulated in a PPP frame



Multilink PPP





EXAMPLE

- Let us go through the phases followed by a network layer packet as it is transmitted through a PPP connection. Figure 4.11 shows the steps. For simplicity, we assume unidirectional movement of data from the user site to the system site (such as sending an e-mail through an ISP).

- The first two frames show link establishment. We have chosen two options (not shown in the figure): using PAP for authentication and suppressing the address control fields. Frames 3 and 4 are for authentication. Frames 5 and 6 establish the network layer connection using IPCP.

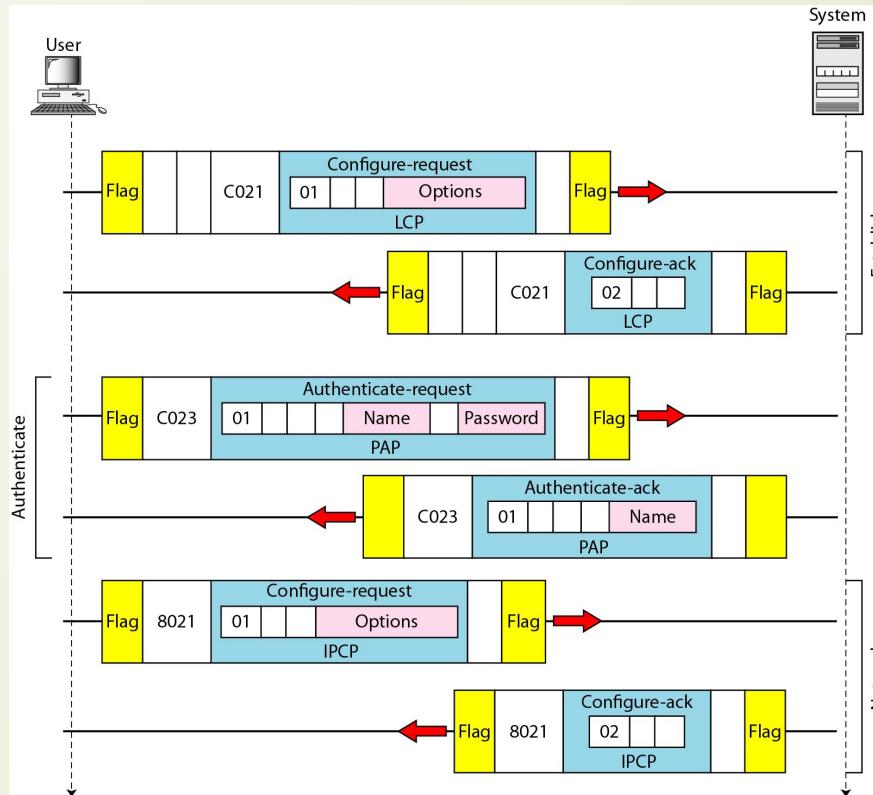


CONTINUED

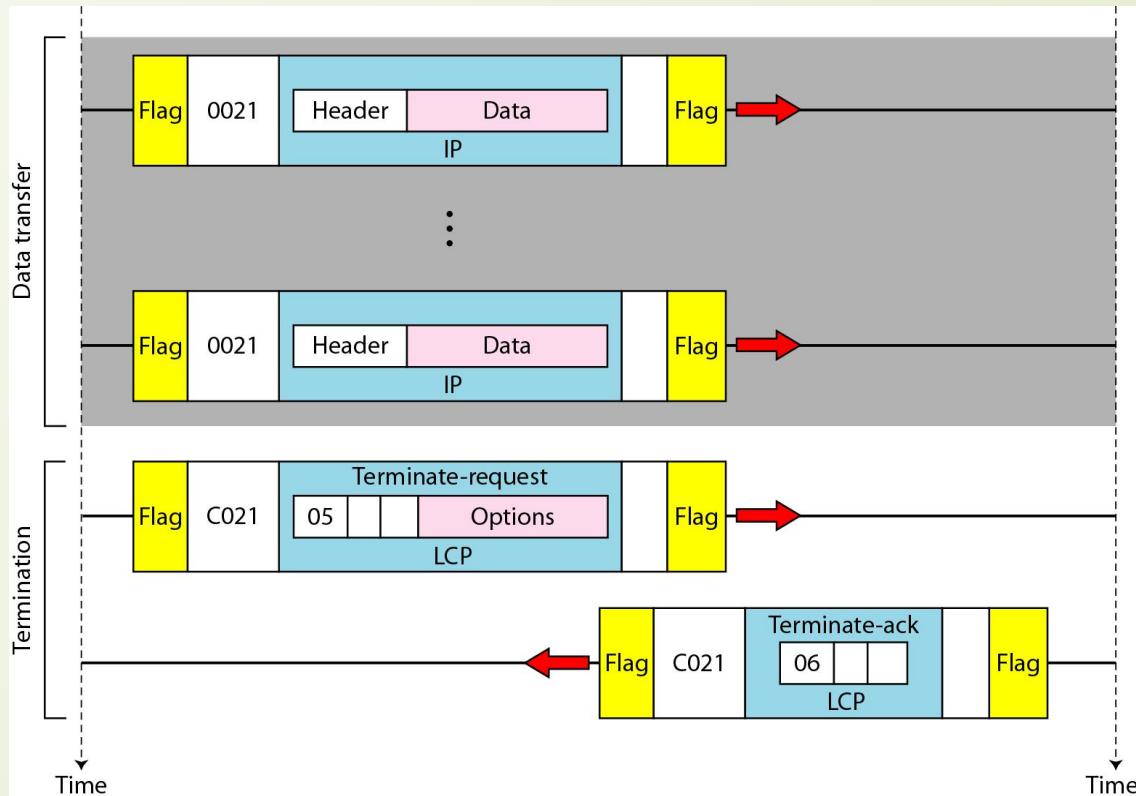
- The next several frames show that some IP packets are encapsulated in the PPP frame. The system (receiver) may have been running several network layer protocols, but it knows that the incoming data must be delivered to the IP protocol because the NCP protocol used before the data transfer was IPCP.

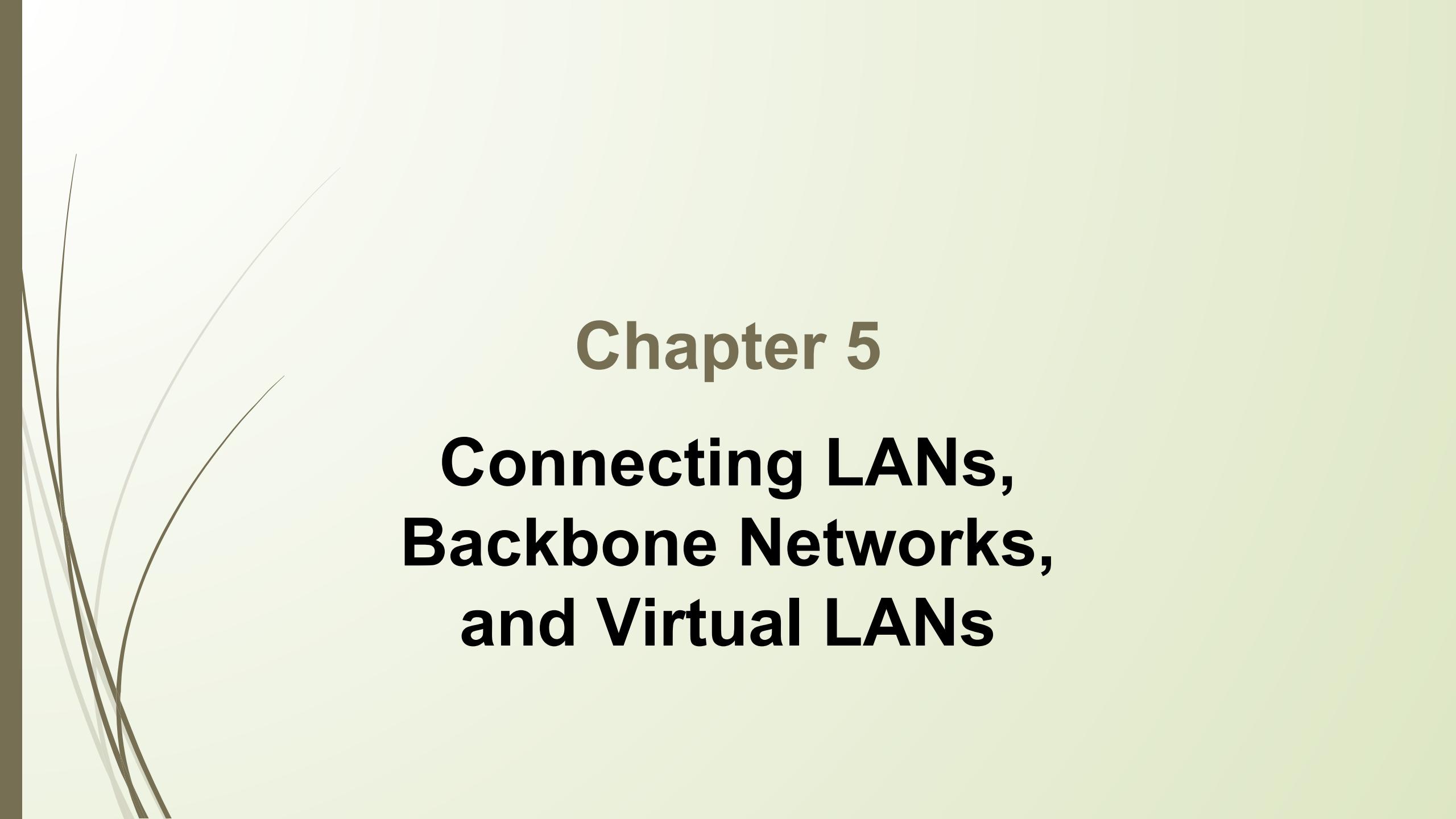
- After data transfer, the user then terminates the data link connection, which is acknowledged by the system. Of course the user or the system could have chosen to terminate the network layer IPCP and keep the data link layer running if it wanted to run another NCP protocol

Figure 4.11



CONTINUED





Chapter 5

Connecting LANs, Backbone Networks, and Virtual LANs

5-1 CONNECTING DEVICES

In this section, we divide connecting devices into five different categories based on the layer in which they operate in a network.

Topics discussed in this section:

Passive Hubs

Active Hubs

Bridges

Two-Layer Switches

Routers

Three-Layer Switches

Gateways

Figure 5.1 *Five categories of connecting devices*

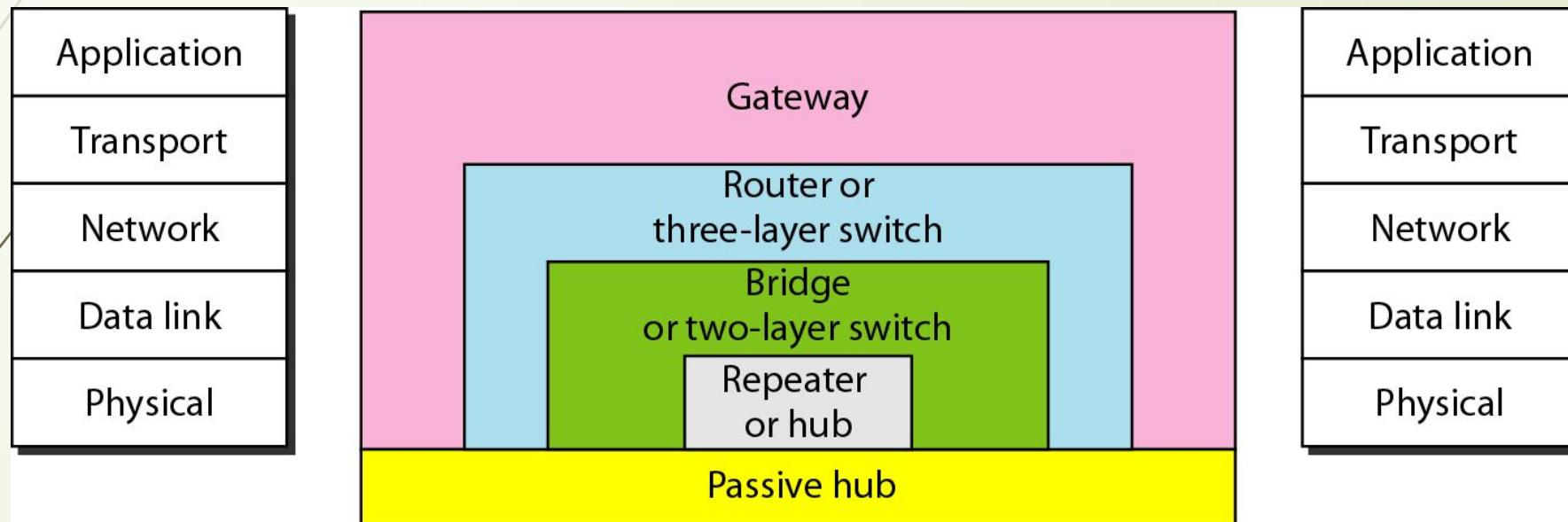
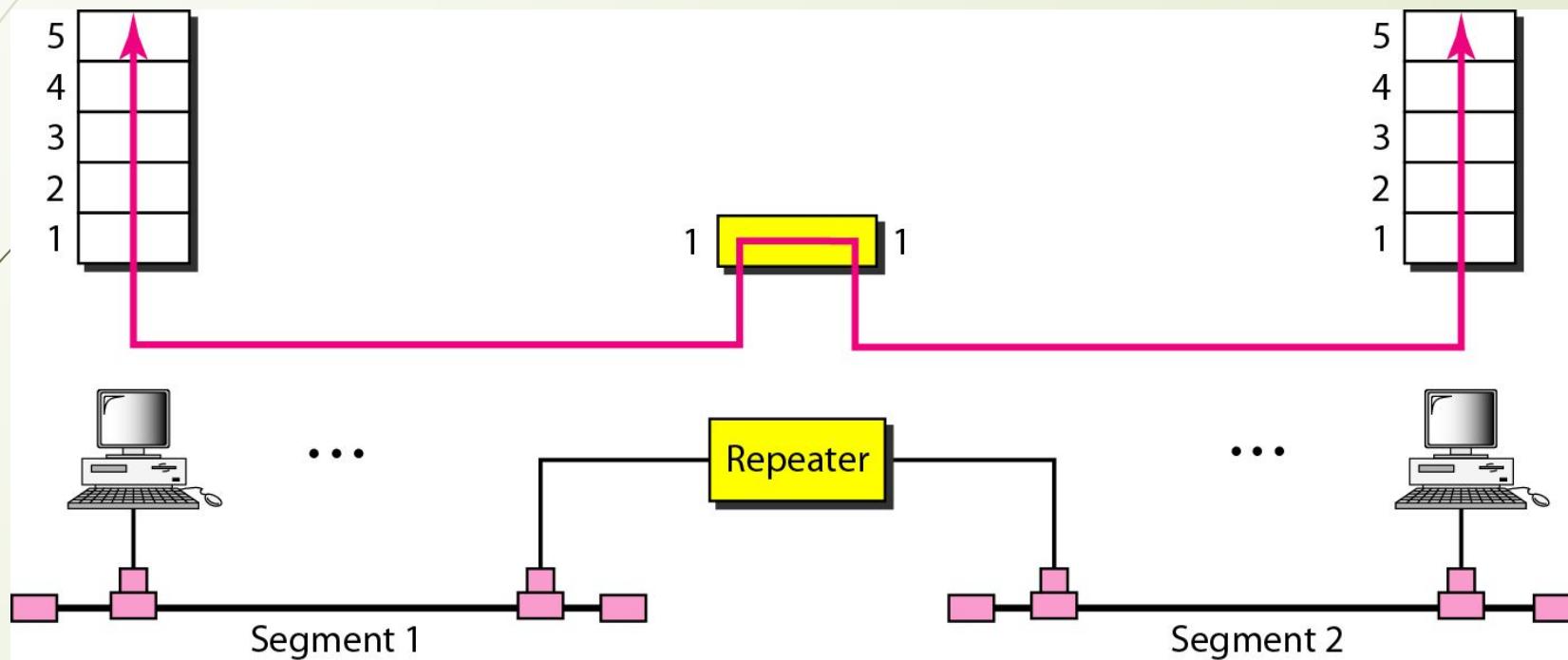


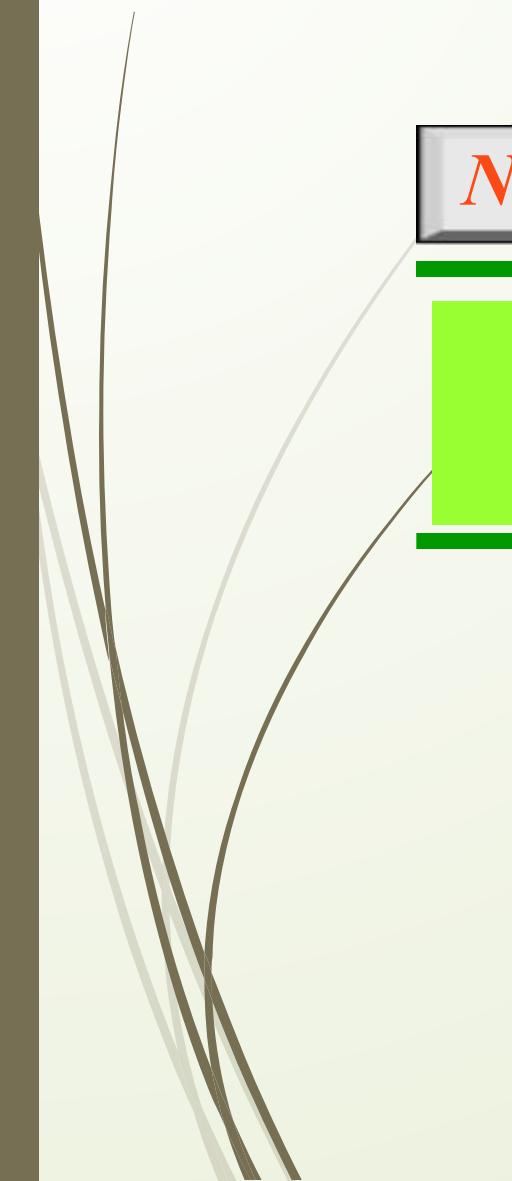
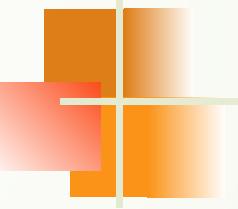
Figure 5.2 *A repeater connecting two segments of a LAN*





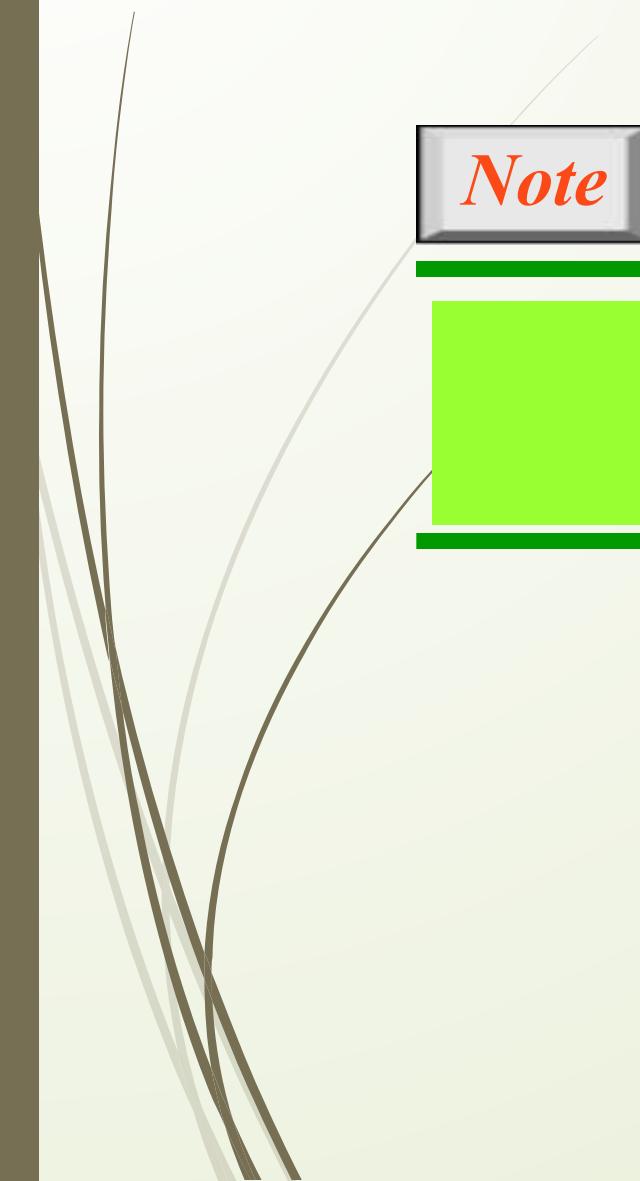
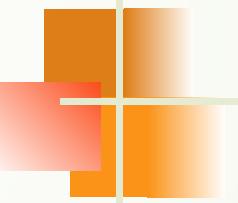
Note

A repeater connects segments of a LAN.



Note

**A repeater forwards every frame;
it has no filtering capability.**



Note

**A repeater is a regenerator,
not an amplifier.**

Figure 5.3 Function of a repeater

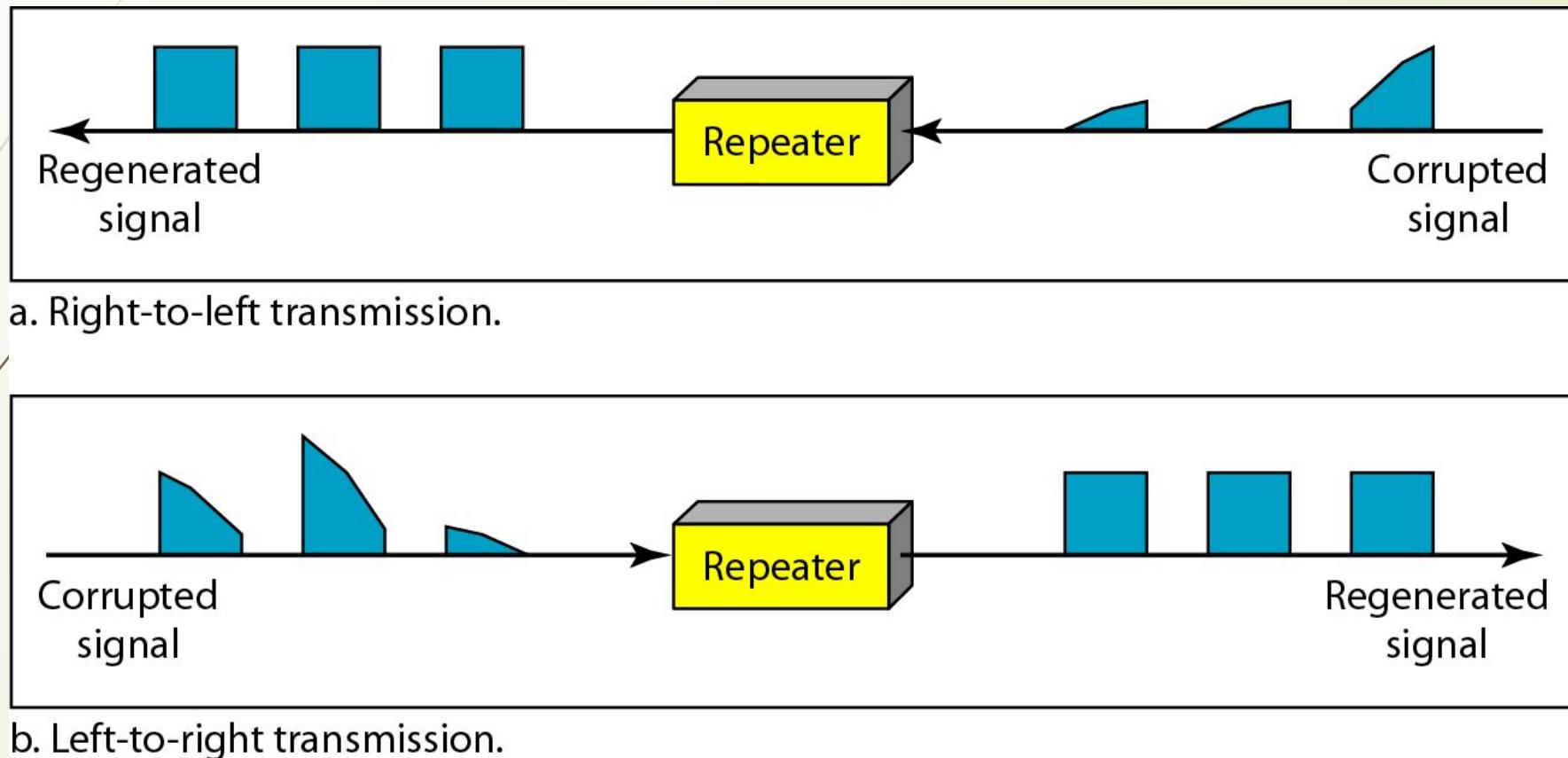
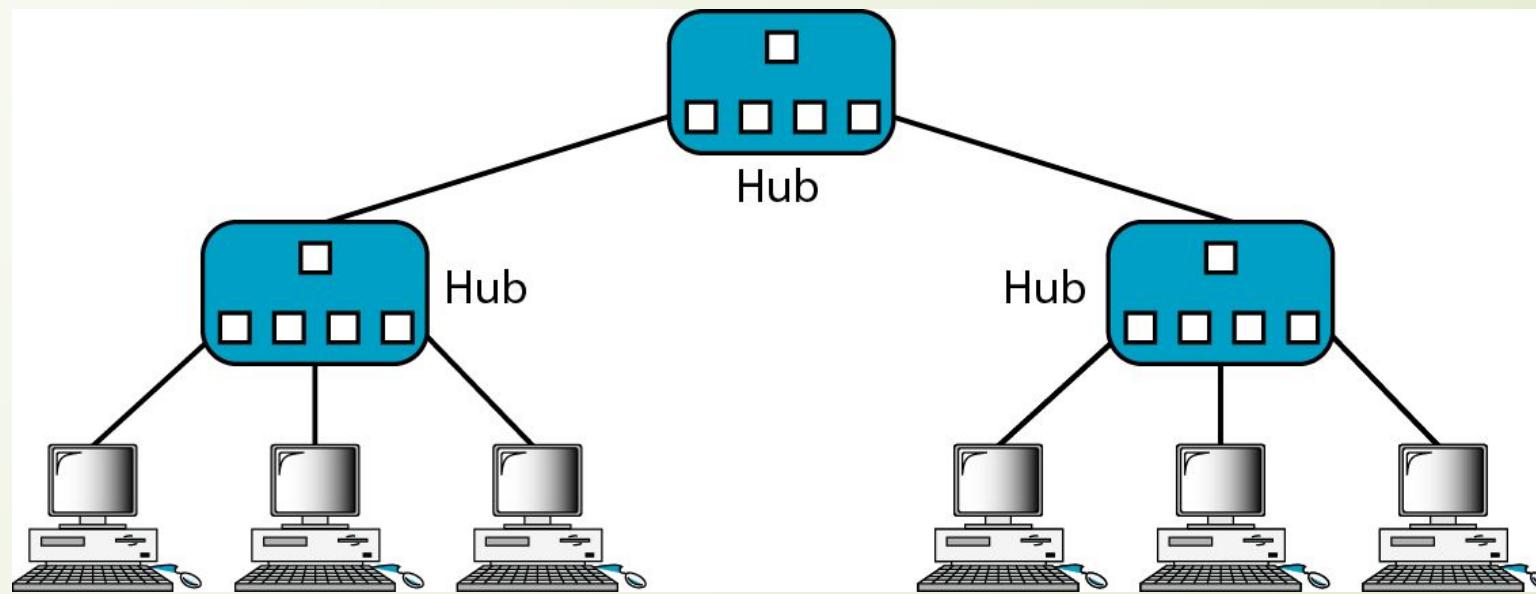
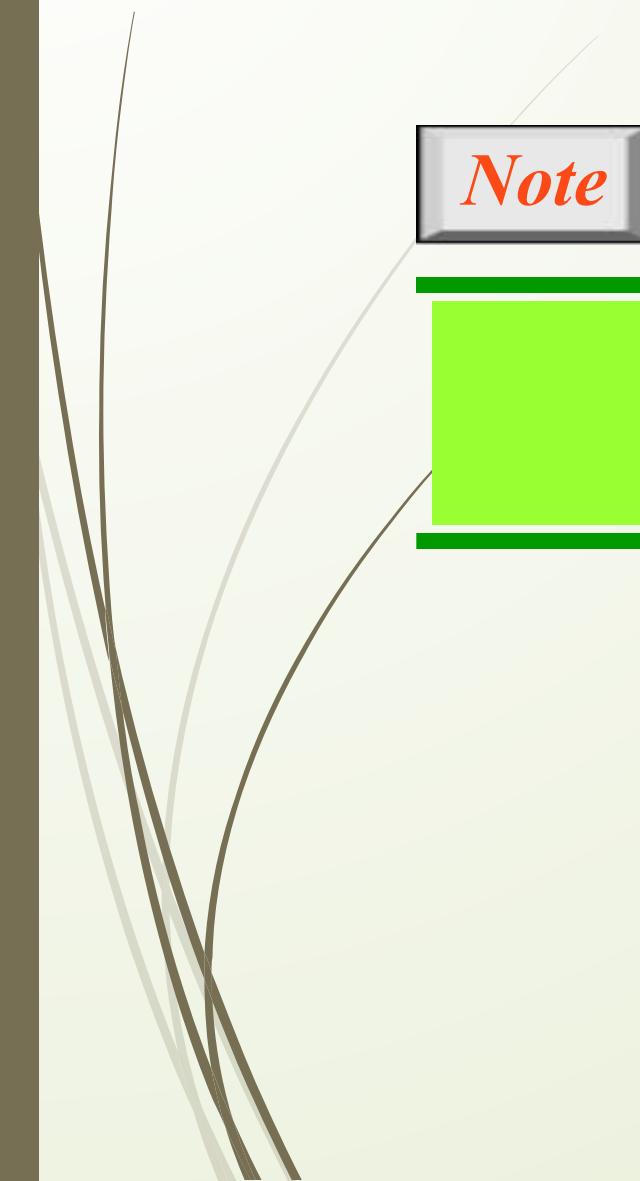
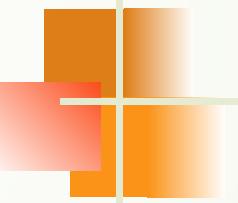


Figure 5.4 *A hierarchy of hubs*

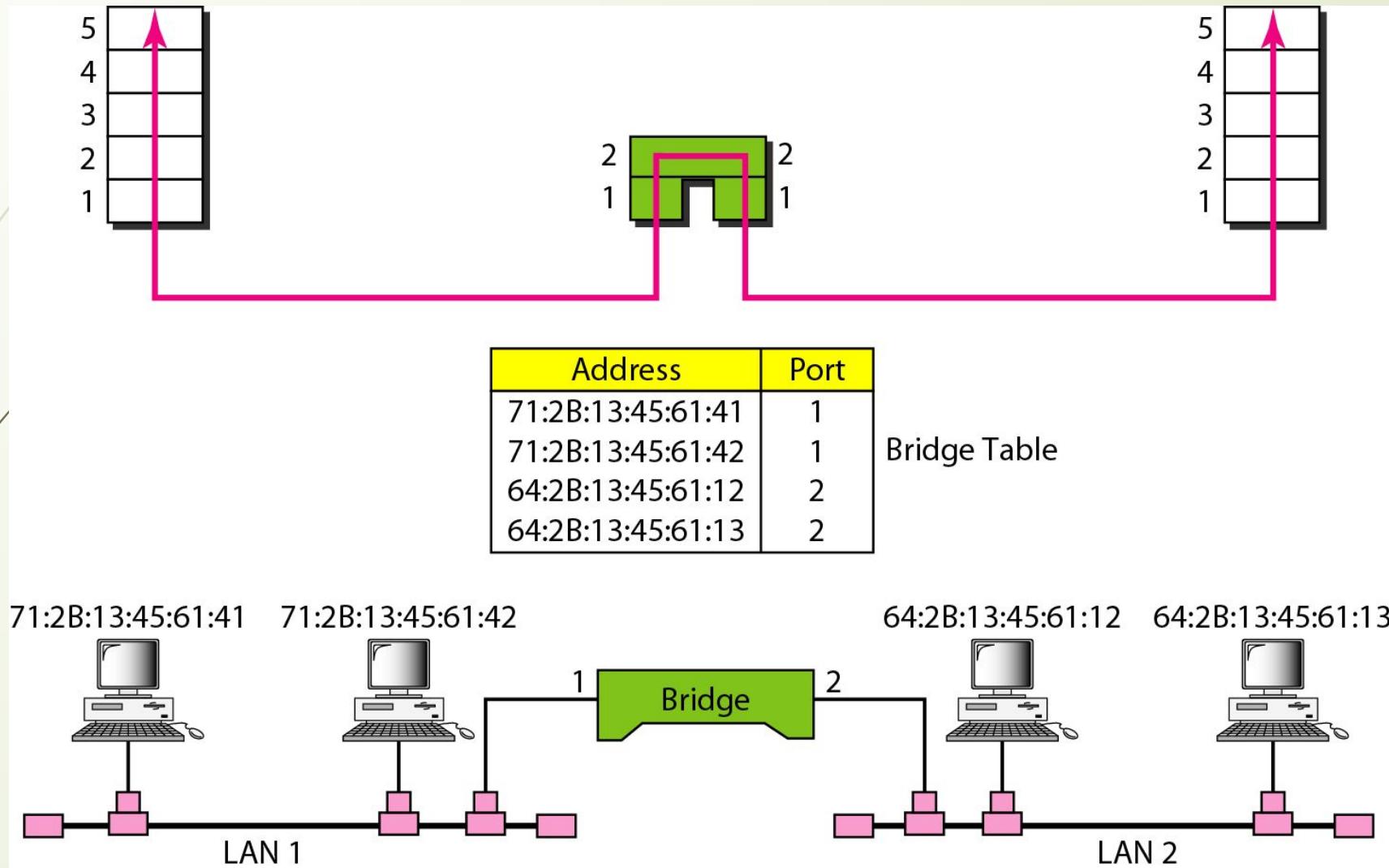




Note

A bridge has a table used in filtering decisions.

Figure 5.5 A bridge connecting two LANs

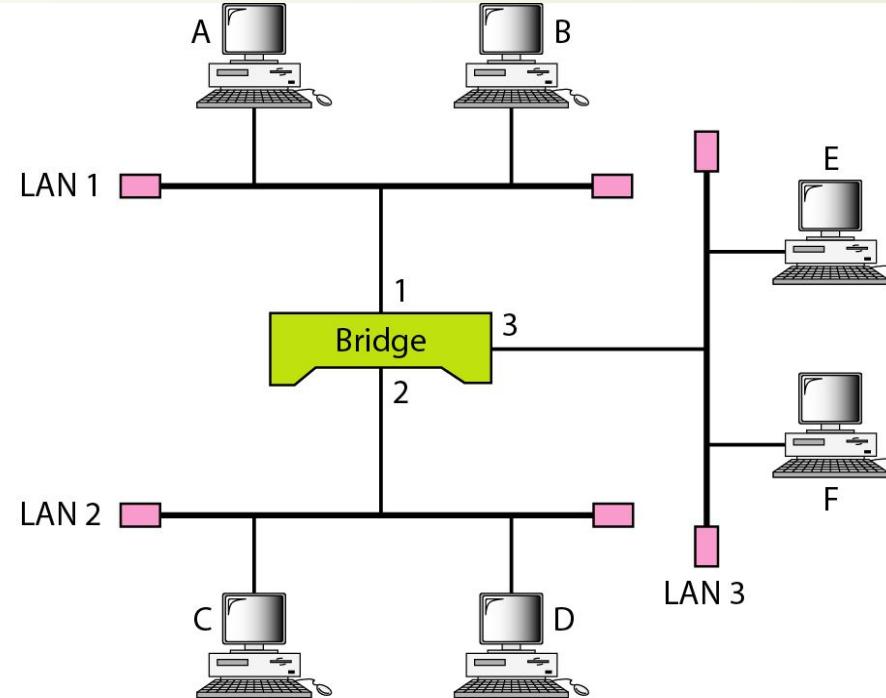




Note

**A bridge does not change the physical
(MAC) addresses in a frame.**

Figure 5.6 A learning bridge and the process of learning



Address	Port

a. Original

Address	Port
A	1

b. After A sends a frame to D

Address	Port
A	1
E	3

c. After E sends a frame to A

Address	Port
A	1
E	3
B	1

d. After B sends a frame to C

Figure 5.7 Loop problem in a learning bridge

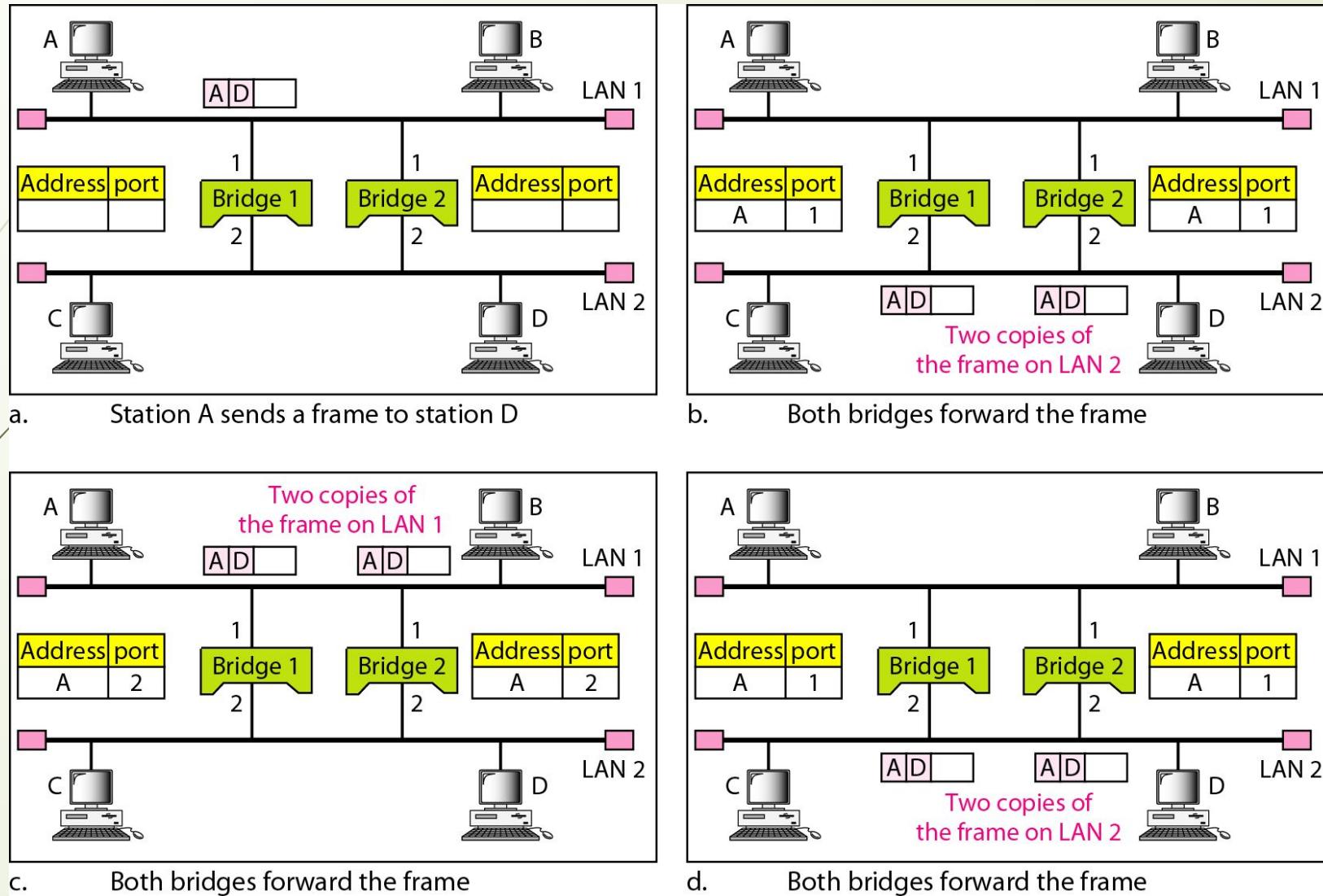
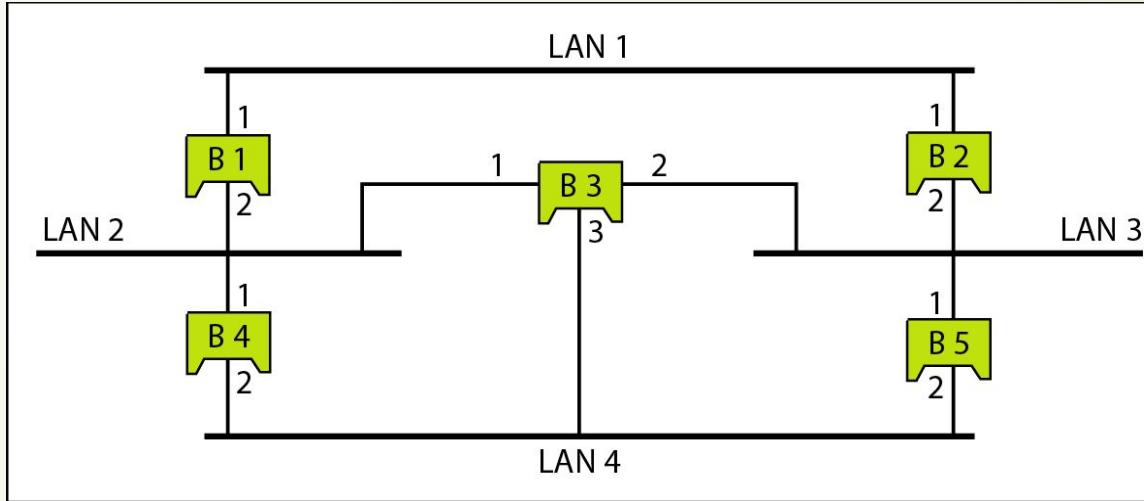
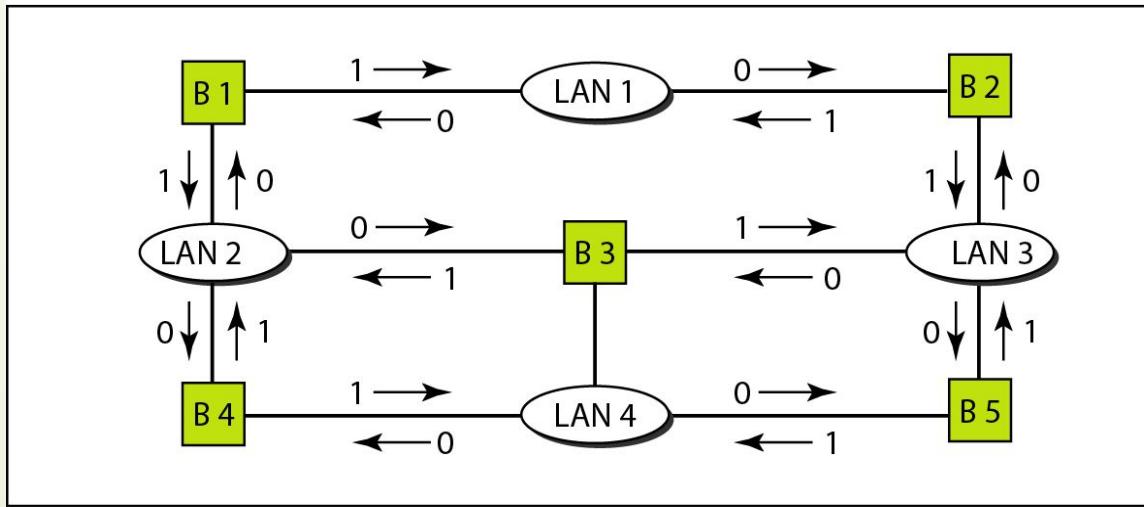


Figure 5.8 A system of connected LANs and its graph representation

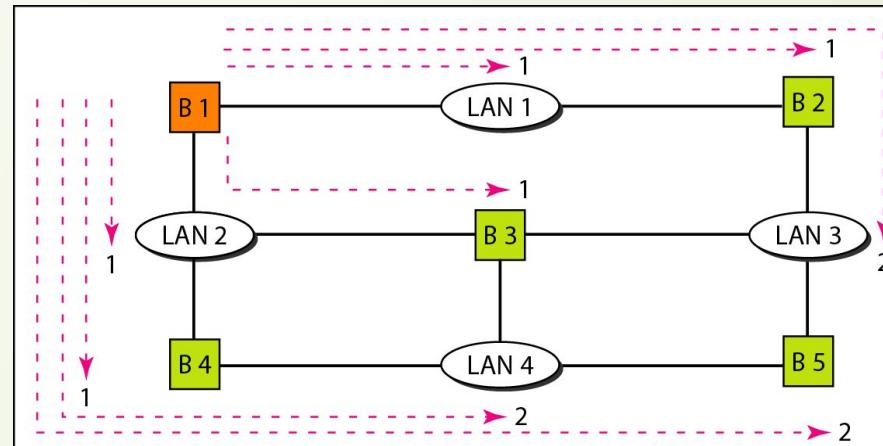


a. Actual system

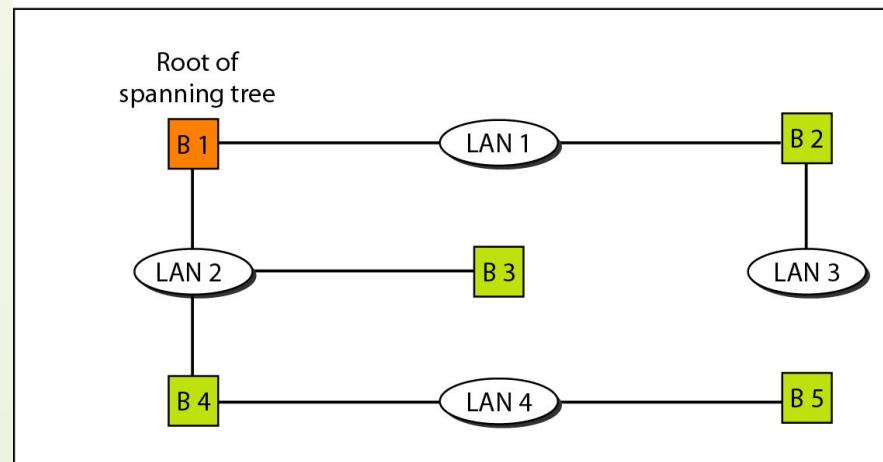


b. Graph representation with cost assigned to each arc

Figure 5.9 *Finding the shortest paths and the spanning tree in a system of bridges*

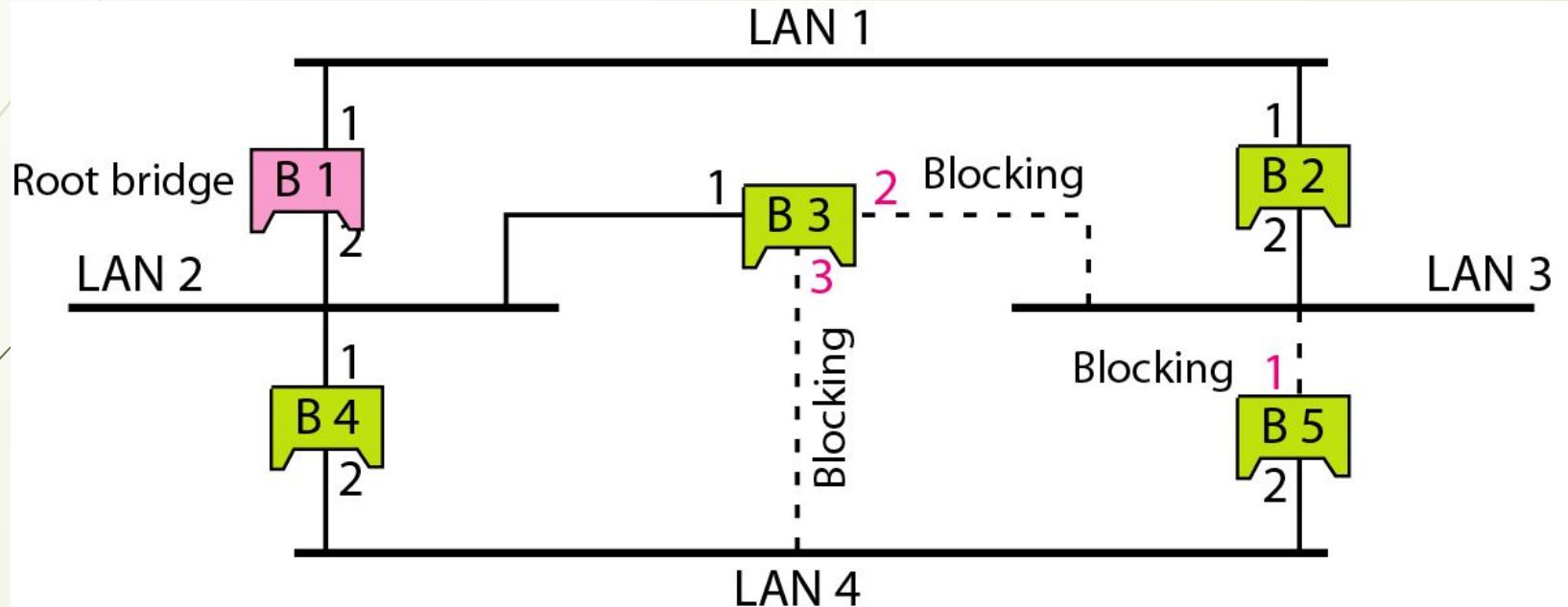


a. Shortest paths



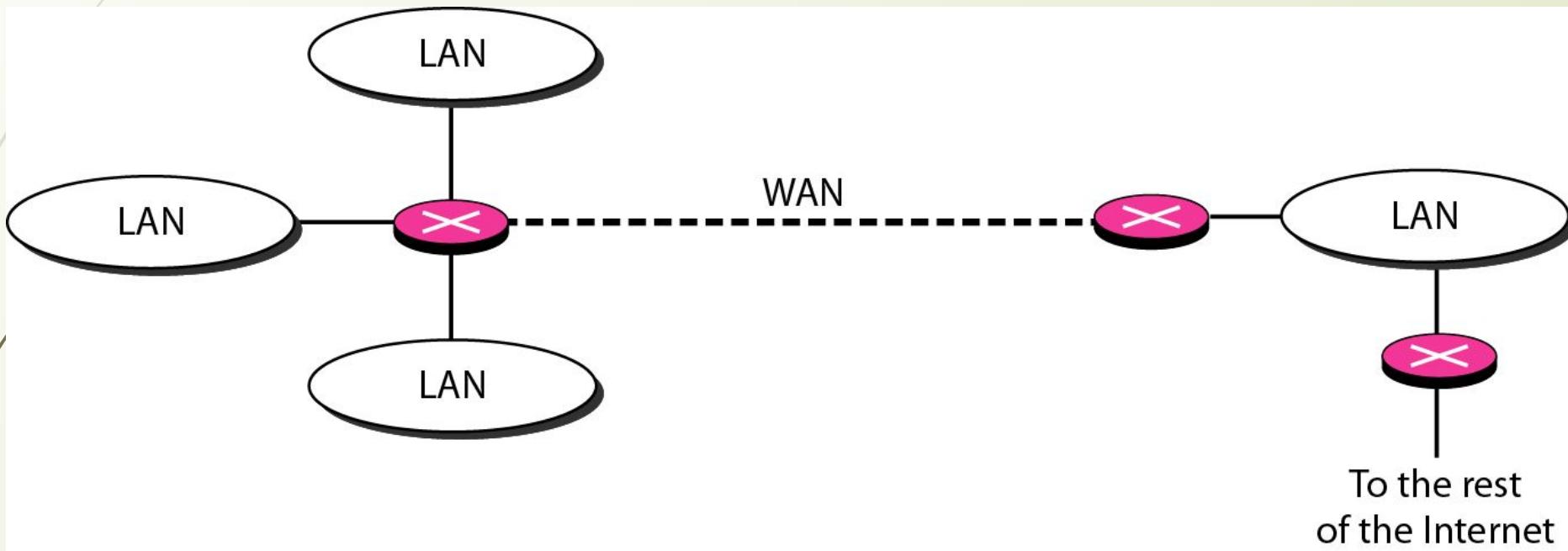
b. Spanning tree

Figure 5.10 Forwarding and blocking ports after using spanning tree algorithm



Ports 2 and 3 of bridge B3 are blocking ports (no frame is sent out of these ports). Port 1 of bridge B5 is also a blocking port (no frame is sent out of this port).

Figure 5.11 *Routers connecting independent LANs and WANs*



Sr. No	Hub	Switch	Router
1.	Hub is a physical layer device i.e. layer 1.	Switch is a data link layer device i.e. layer 2.	Router is a network layer device i.e. layer 3.
2.	A Hub works on the basis of broadcasting.	Switch works on the basis of MAC address.	A router works on the basis of IP address.
3.	A Hub is a multiport repeater in which a signal introduced at the input of any port appears at the other ports.	A Switch is a tele-communication device which receives a message from any device connected to it and then transmits the message only to the device for which the message is intended.	A route reads the header of incoming packet and forward it to the port for which it is intended there by determines the route. It can also perform filtering and encapsulation.
4.	Hub is not an intelligent device that may include amplifier or repeater.	A Switch is an intelligent device as it passes on the message to the selective device by inspecting the address.	A router is more sophisticated and intelligent device as it can read IP address and direct the packets to another network with specified IP address. Moreover routers can build address tables that helps in routing decisions.
5.	At least single network is required to connect.	At least single network is required to connect.	Router needs at least two networks to connect.
6.	Hub is cheaper as compared to switch and router.	Switch is an expensive device than hub.	Router is a relatively much more expensive device than hub and switch.

5-2 BACKBONE NETWORKS

A backbone network allows several LANs to be connected. In a backbone network, no station is directly connected to the backbone; the stations are part of a LAN, and the backbone connects the LANs.

Topics discussed in this section:

Bus Backbone

Star Backbone

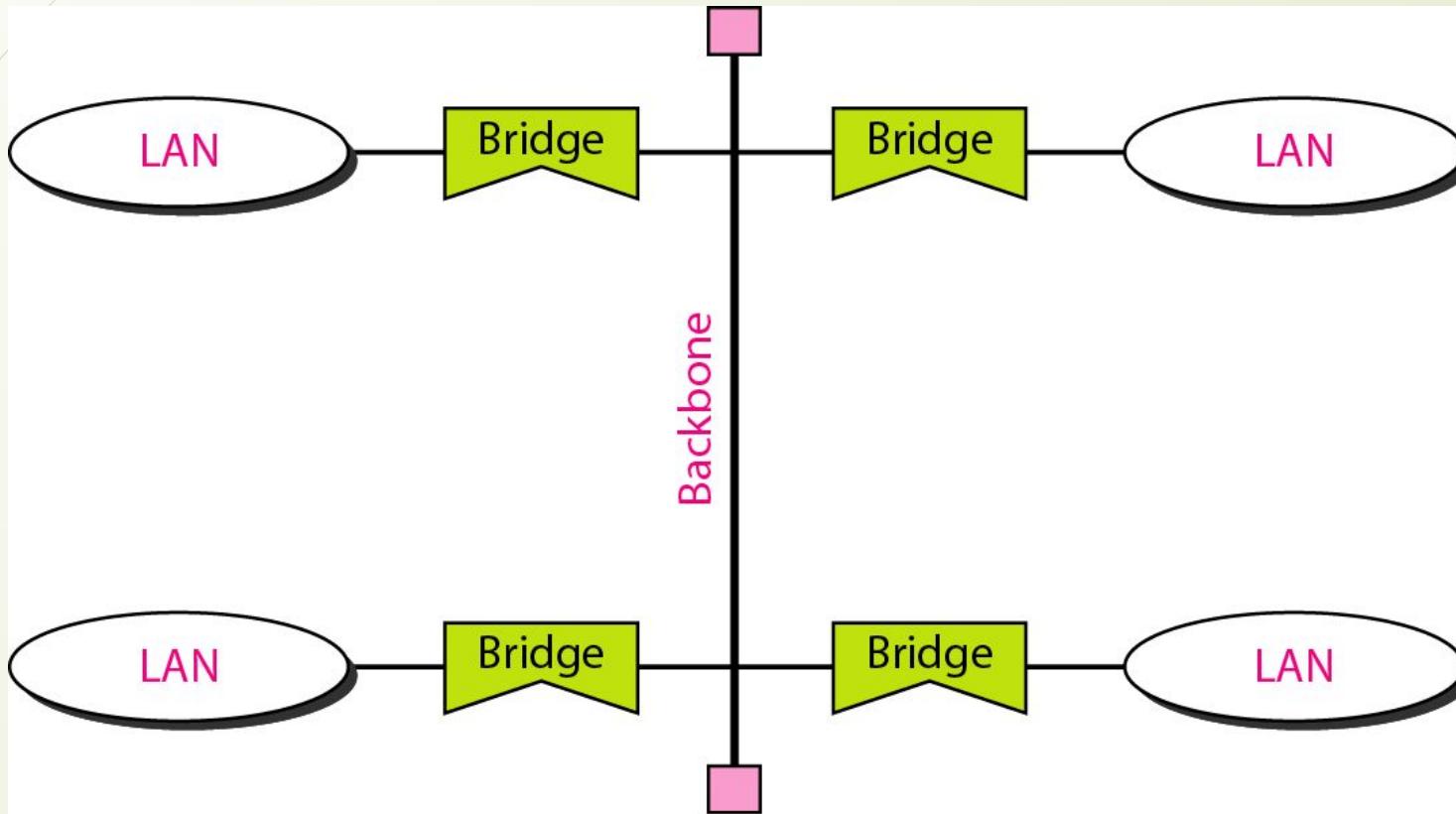
Connecting Remote LANs

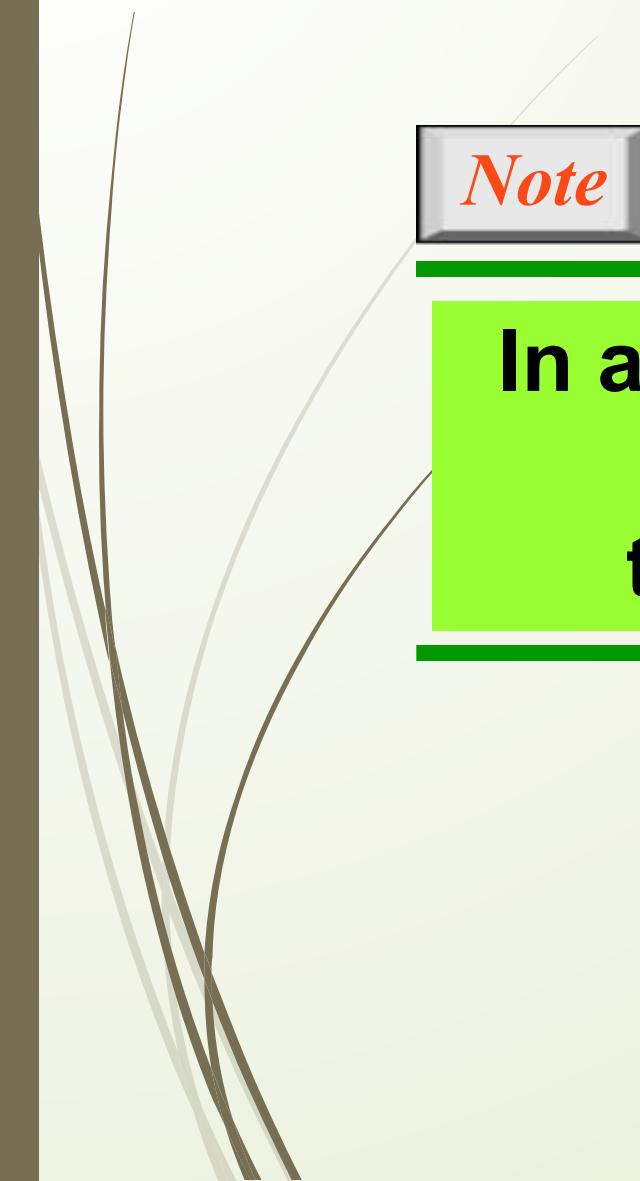
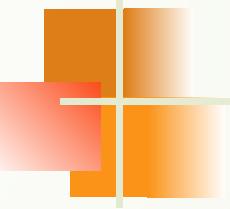


Note

**In a bus backbone, the topology
of the backbone is a bus.**

Figure 5.12 Bus backbone





Note

**In a star backbone, the topology of the backbone is a star;
the backbone is just one switch.**

Figure 5.13 *Star backbone*

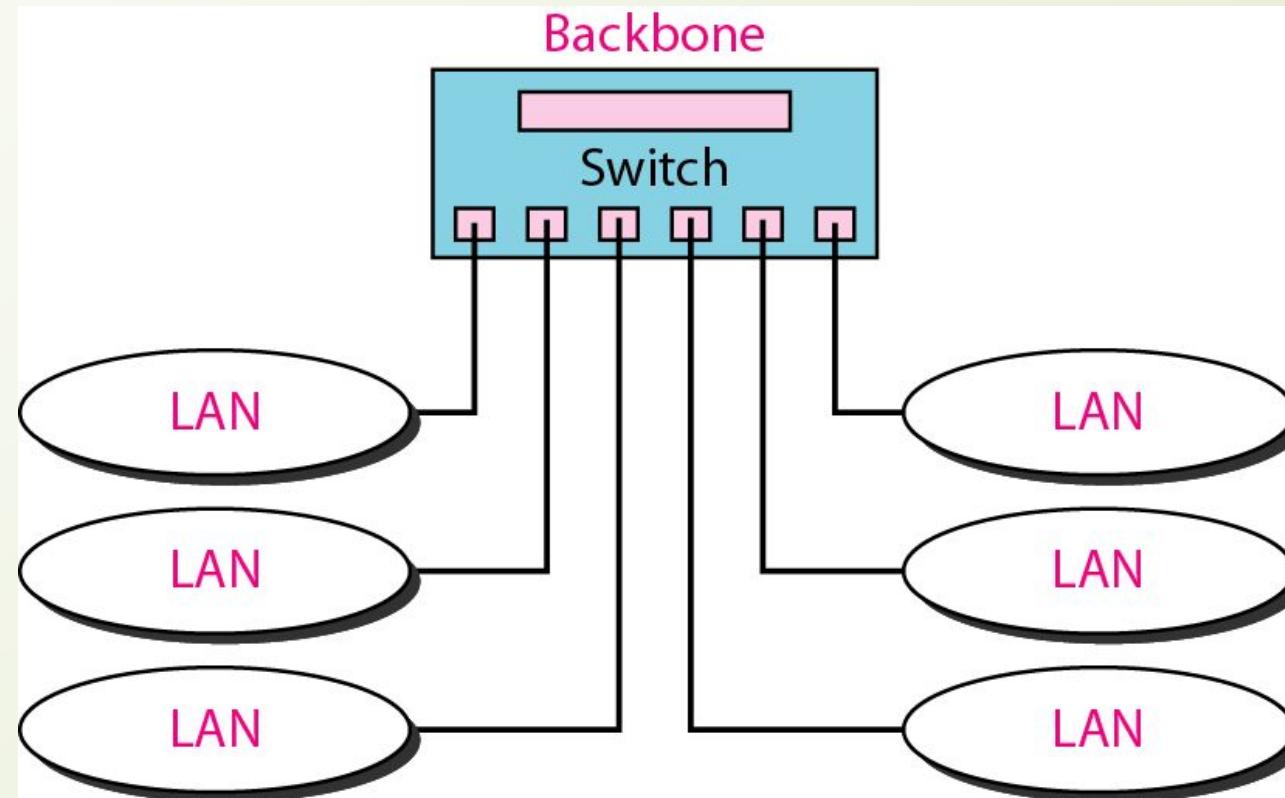
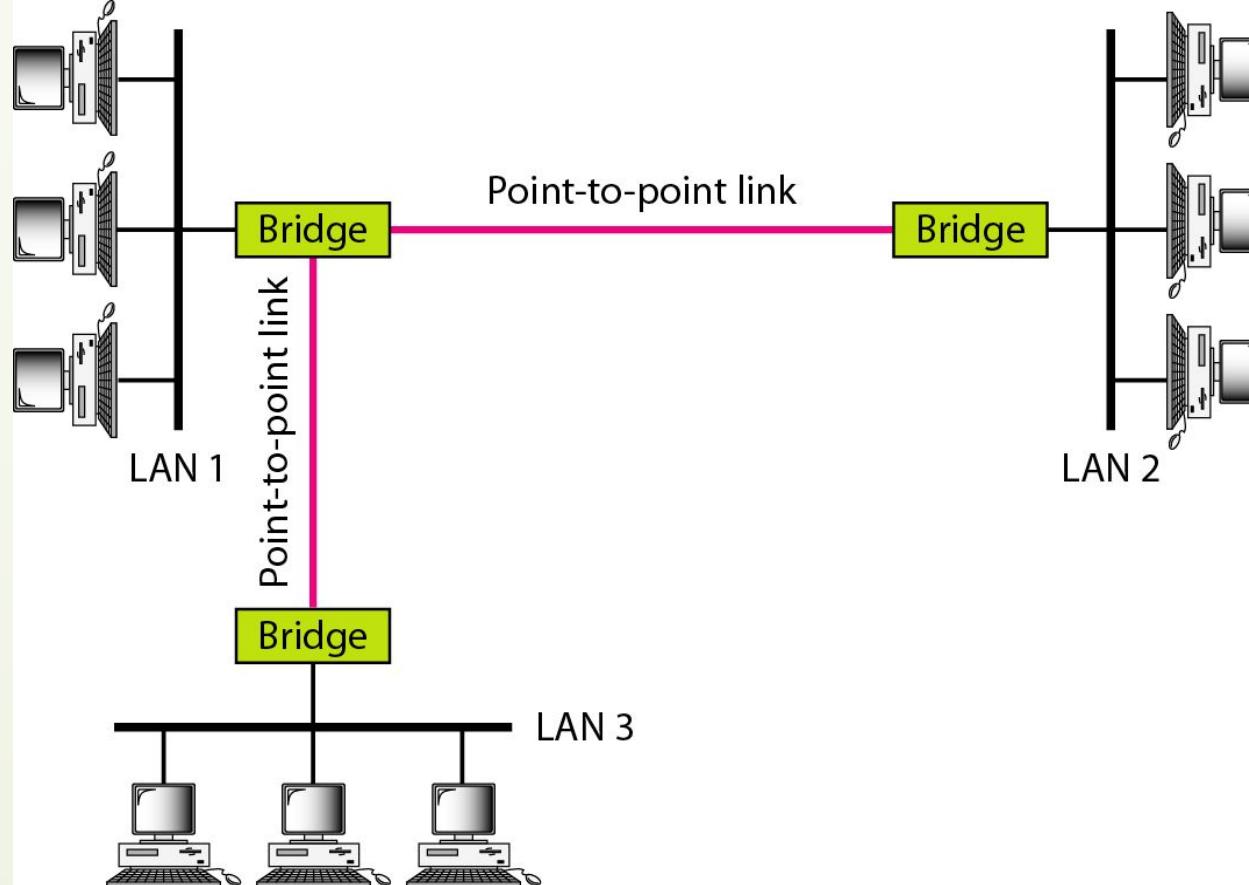


Figure 5.14 Connecting remote LANs with bridges





Note

A point-to-point link acts as a LAN in a remote backbone connected by remote bridges.

5-3 VIRTUAL LANs

We can roughly define a virtual local area network (VLAN) as a local area network configured by software, not by physical wiring.

Topics discussed in this section:

Membership

Configuration

Communication between Switches

IEEE Standard

Advantages

Figure 5.15 *A switch connecting three LANs*

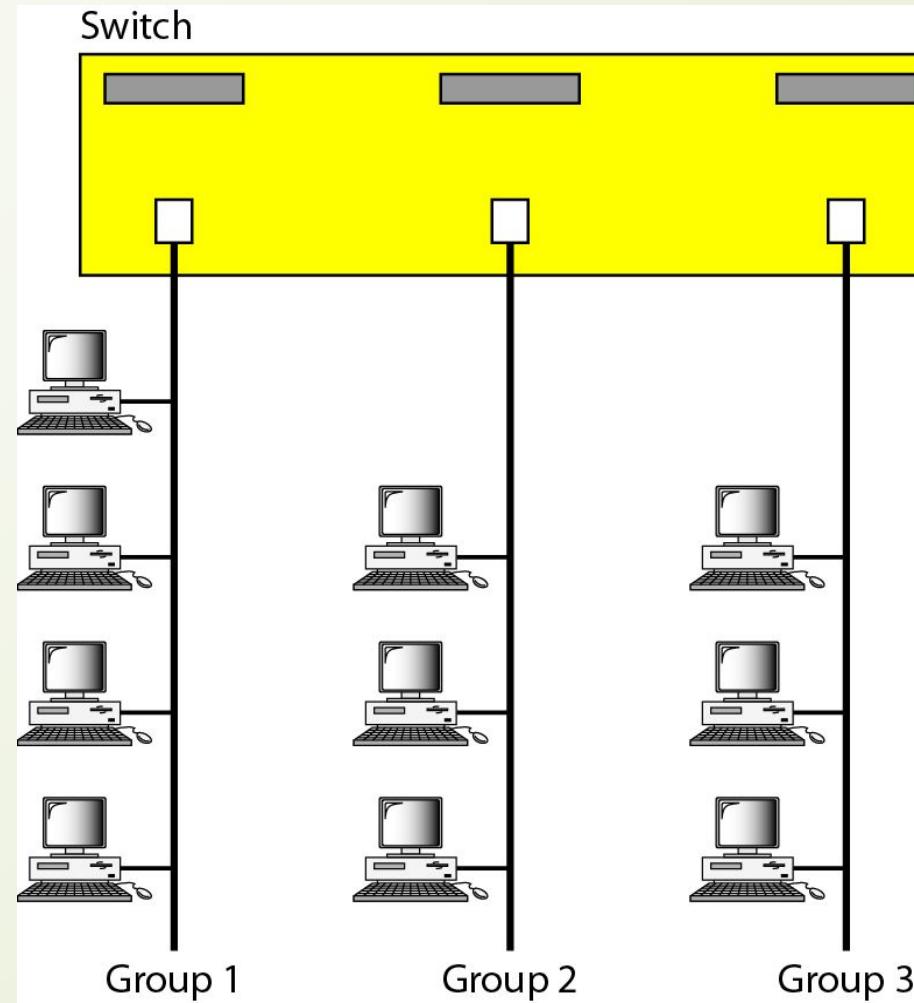


Figure 5.16 A switch using VLAN software

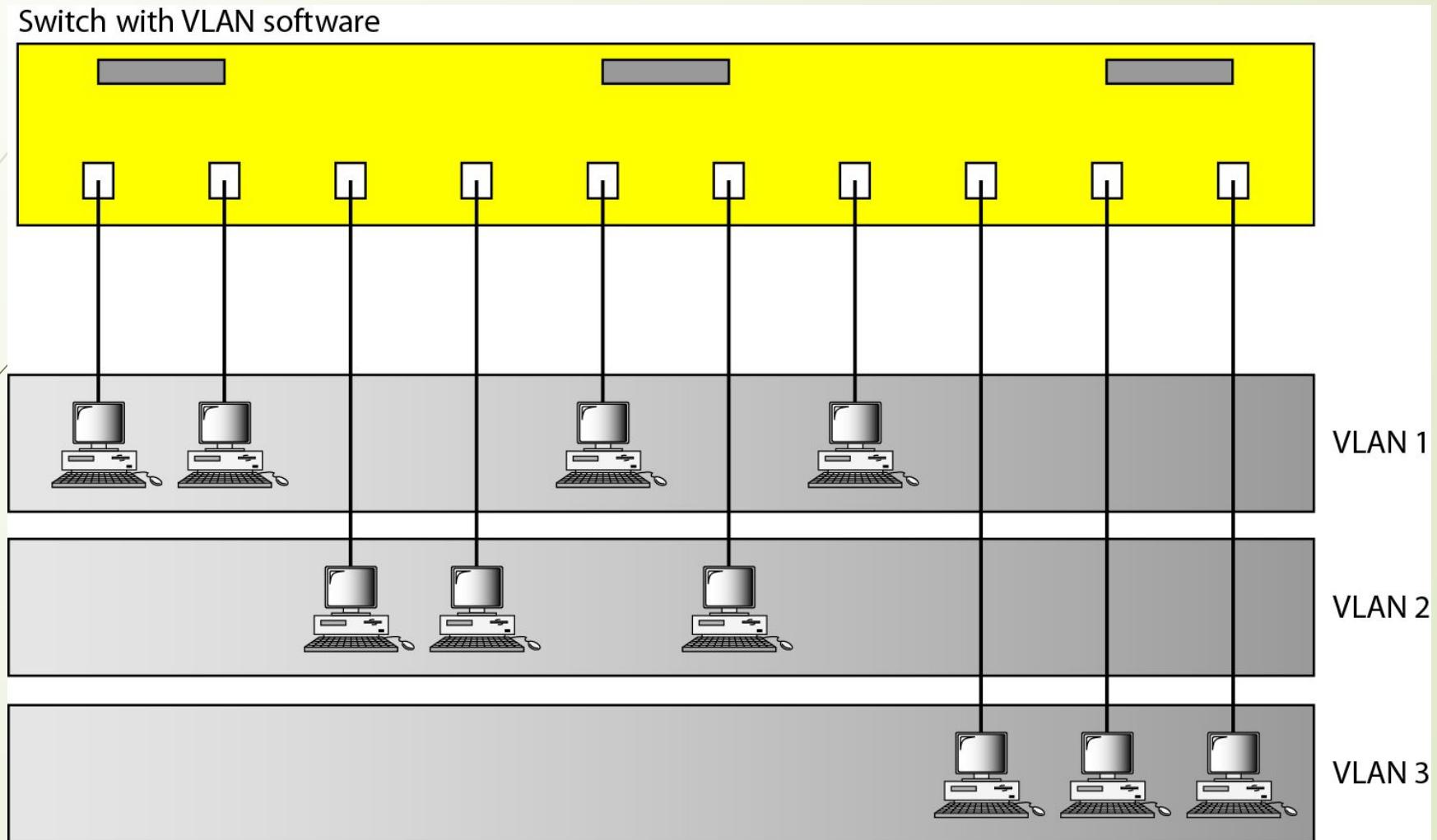
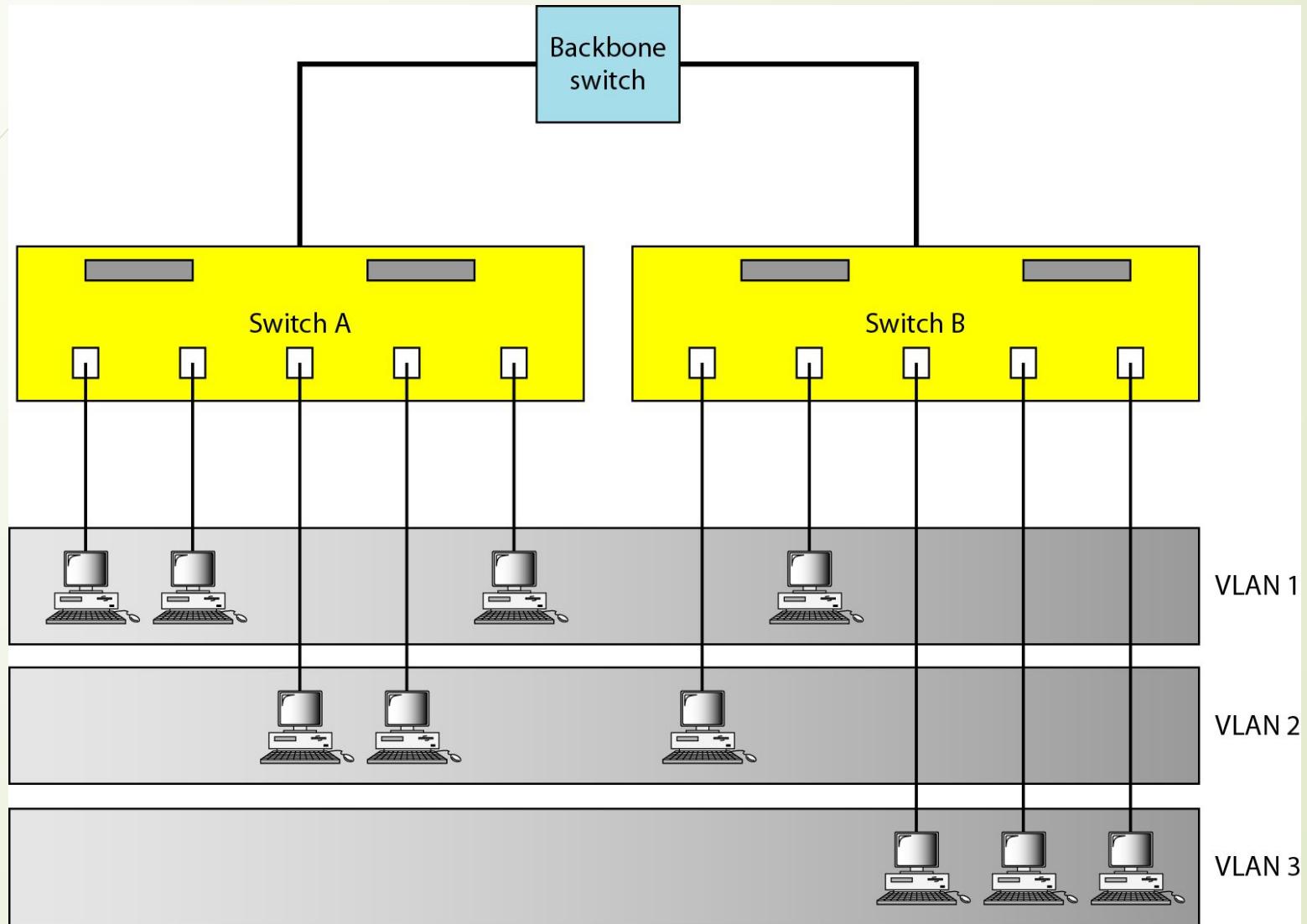
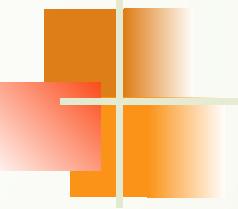


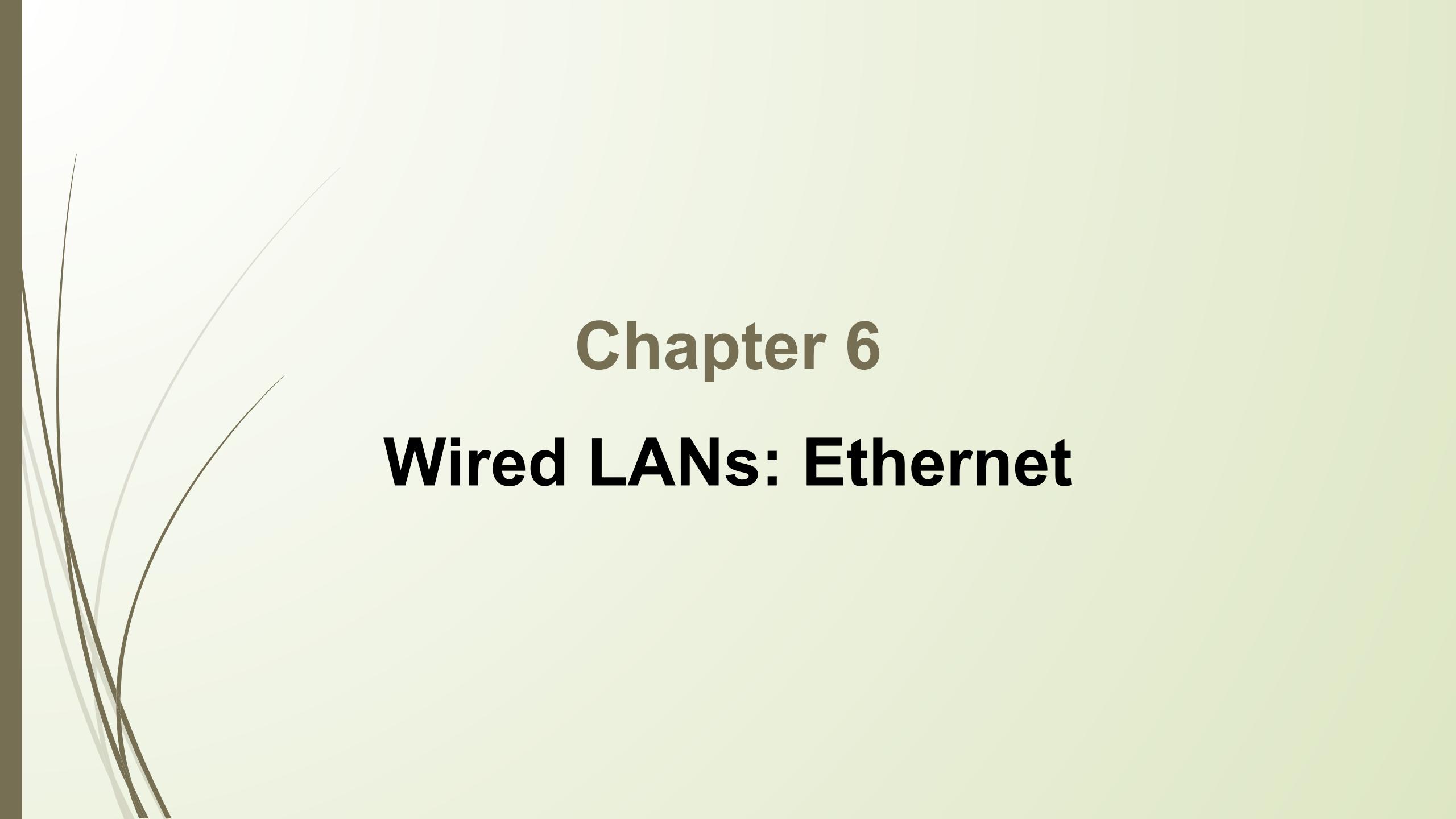
Figure 5.17 Two switches in a backbone using VLAN software





Note

VLANs create broadcast domains.



Chapter 6

Wired LANs: Ethernet

6-1 IEEE STANDARDS

In 1985, the Computer Society of the IEEE started a project, called Project 802, to set standards to enable intercommunication among equipment from a variety of manufacturers. Project 802 is a way of specifying functions of the physical layer and the data link layer of major LAN protocols.

Topics discussed in this section:

Data Link Layer
Physical Layer

Figure 6.1 IEEE standard for LANs

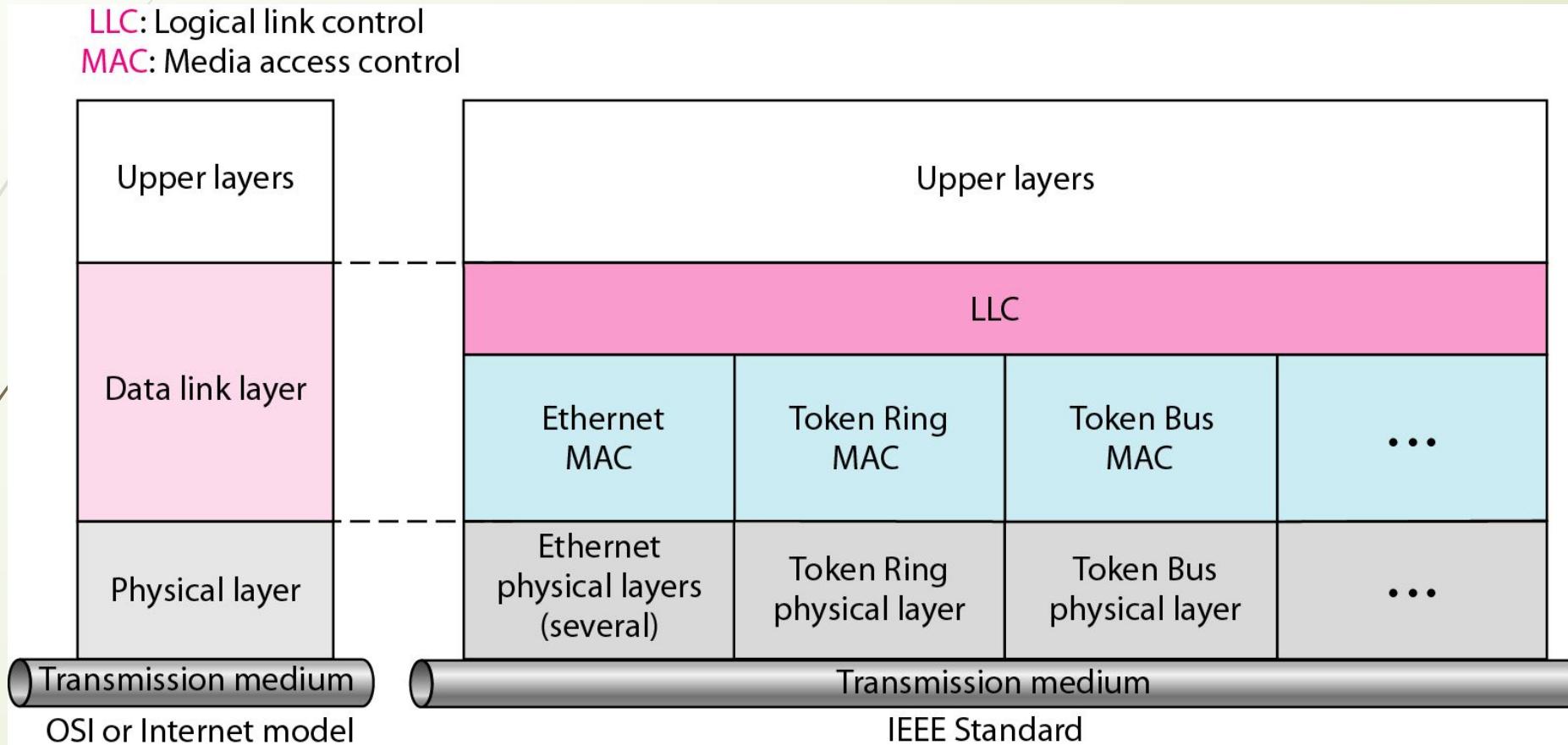
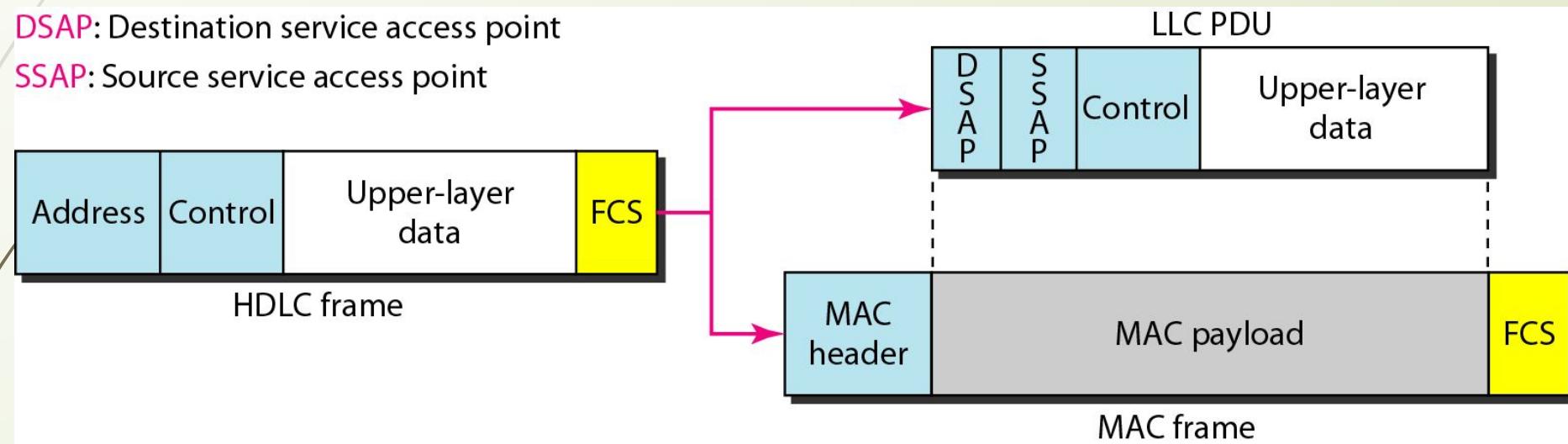


Figure 6.2 *HDLC frame compared with LLC and MAC frames*



6-2 STANDARD ETHERNET

The original Ethernet was created in 1976 at Xerox's Palo Alto Research Center (PARC). Since then, it has gone through four generations. We briefly discuss the Standard (or traditional) Ethernet in this section.

Topics discussed in this section:

MAC Sublayer

Physical Layer

Figure 6.3 *Ethernet evolution through four generations*

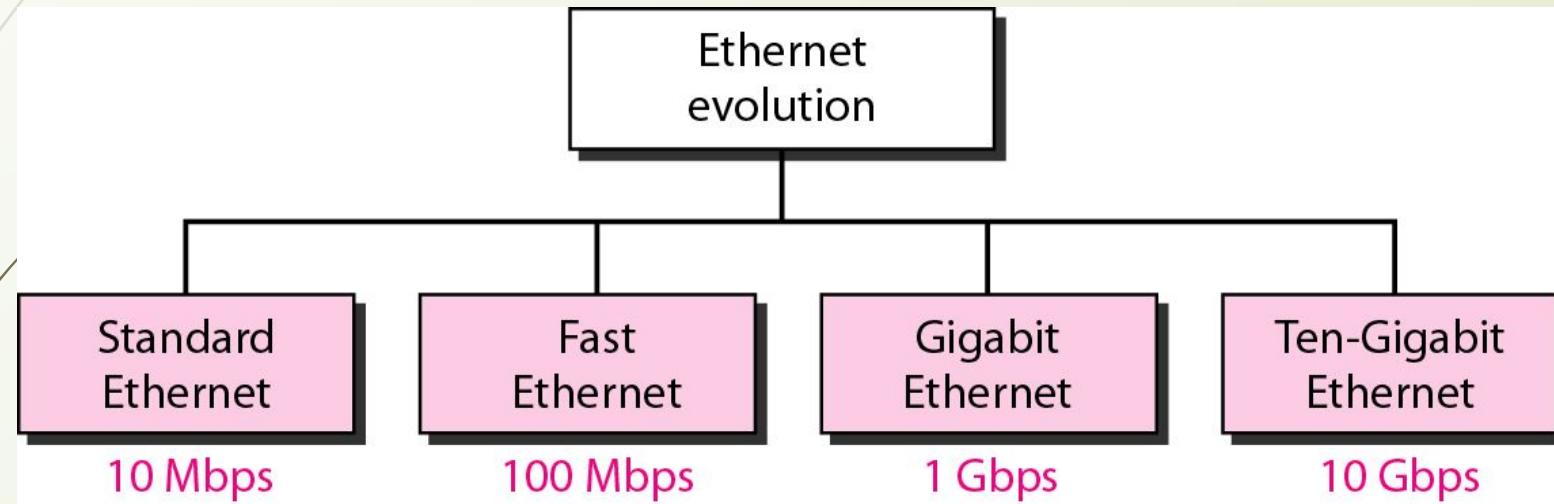


Figure 6.4 802.3 MAC frame

Preamble: 56 bits of alternating 1s and 0s.

SFD: Start frame delimiter, flag (10101011)

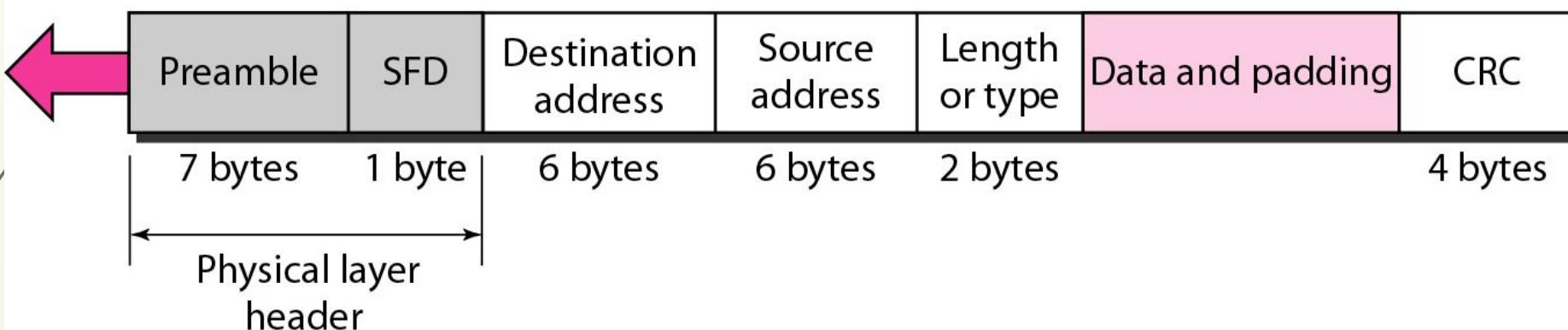
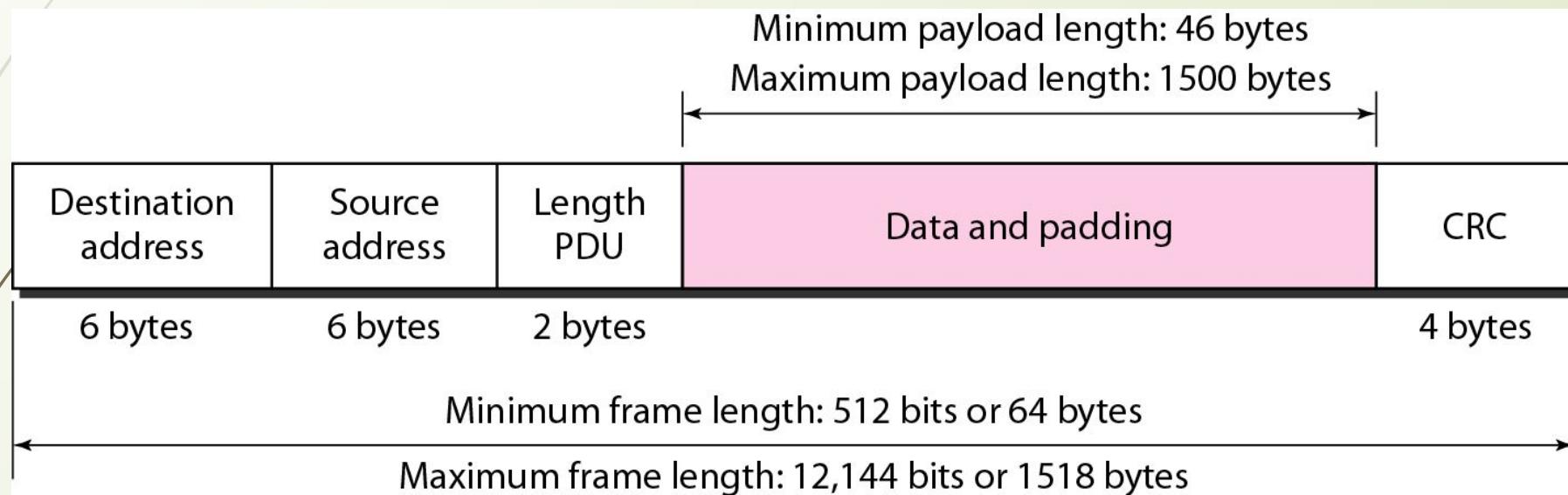
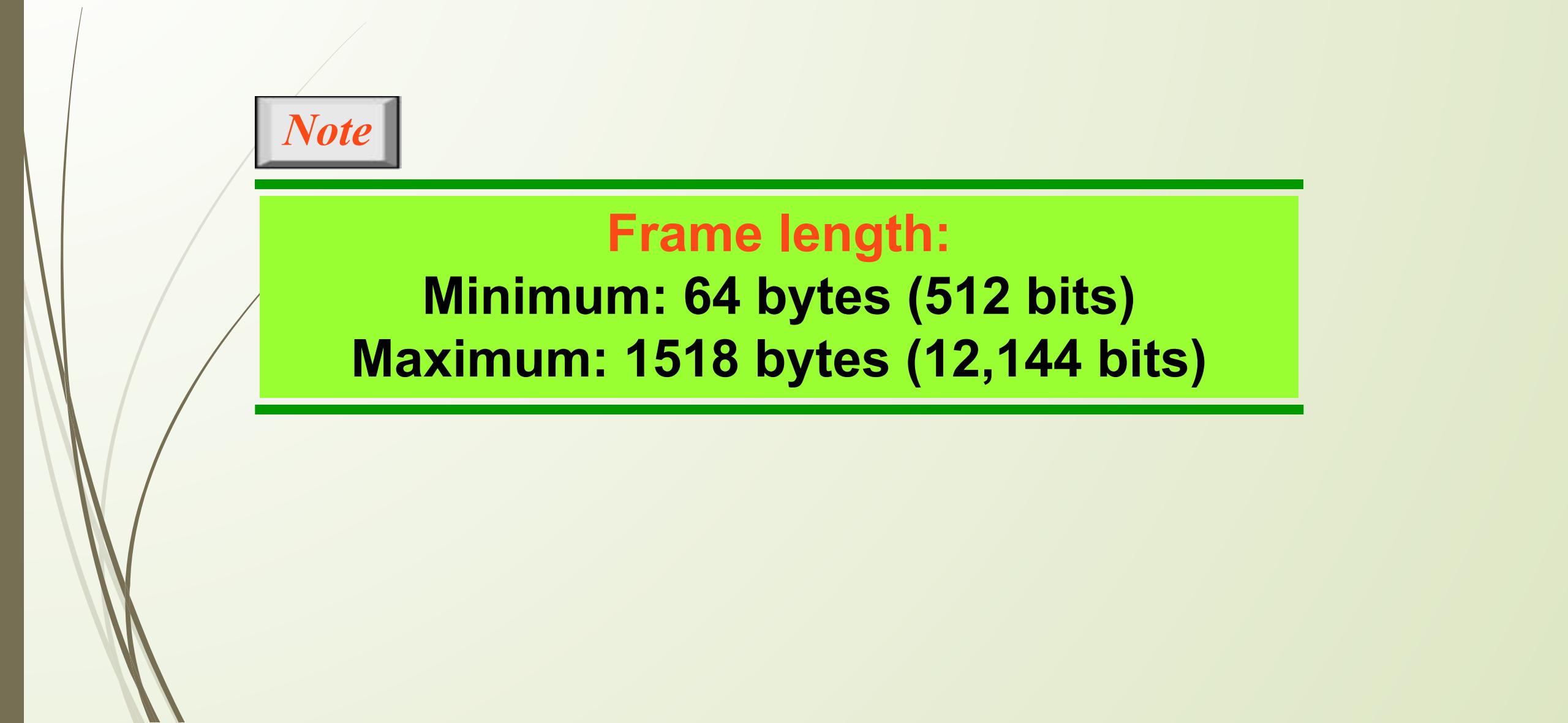


Figure 6.5 *Minimum and maximum lengths*





Note

Frame length:

Minimum: 64 bytes (512 bits)

Maximum: 1518 bytes (12,144 bits)

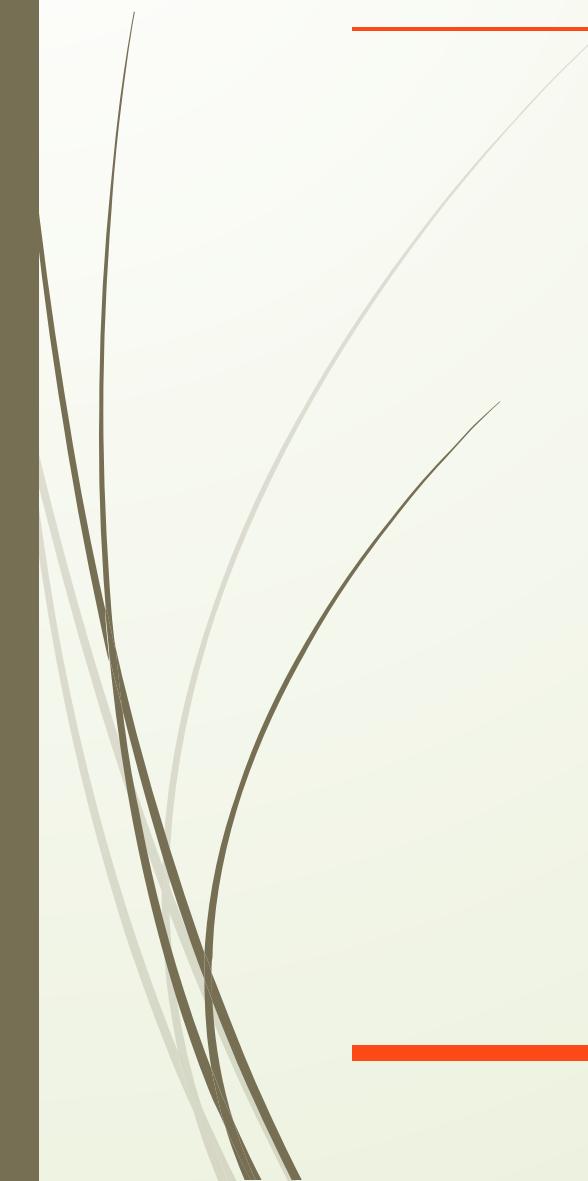
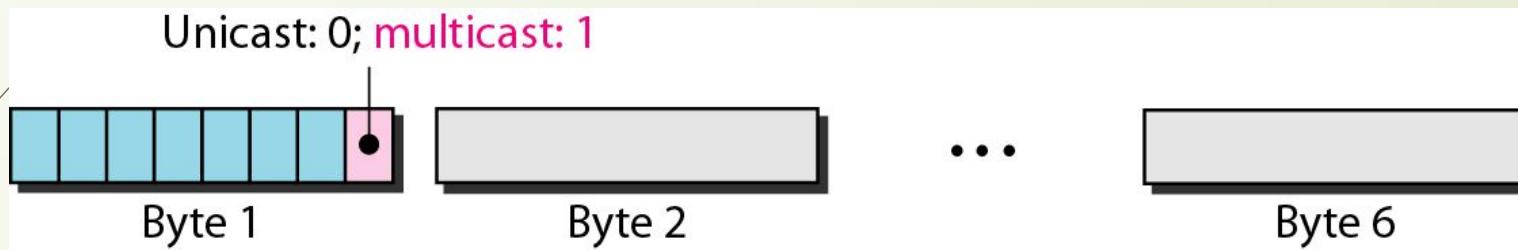


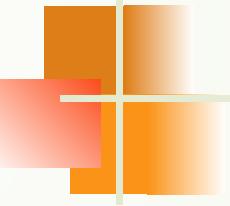
Figure 6.6 *Example of an Ethernet address in hexadecimal notation*

06 : 01 : 02 : 01 : 2C : 4B

6 bytes = 12 hex digits = 48 bits

Figure 6.7 Unicast and multicast addresses

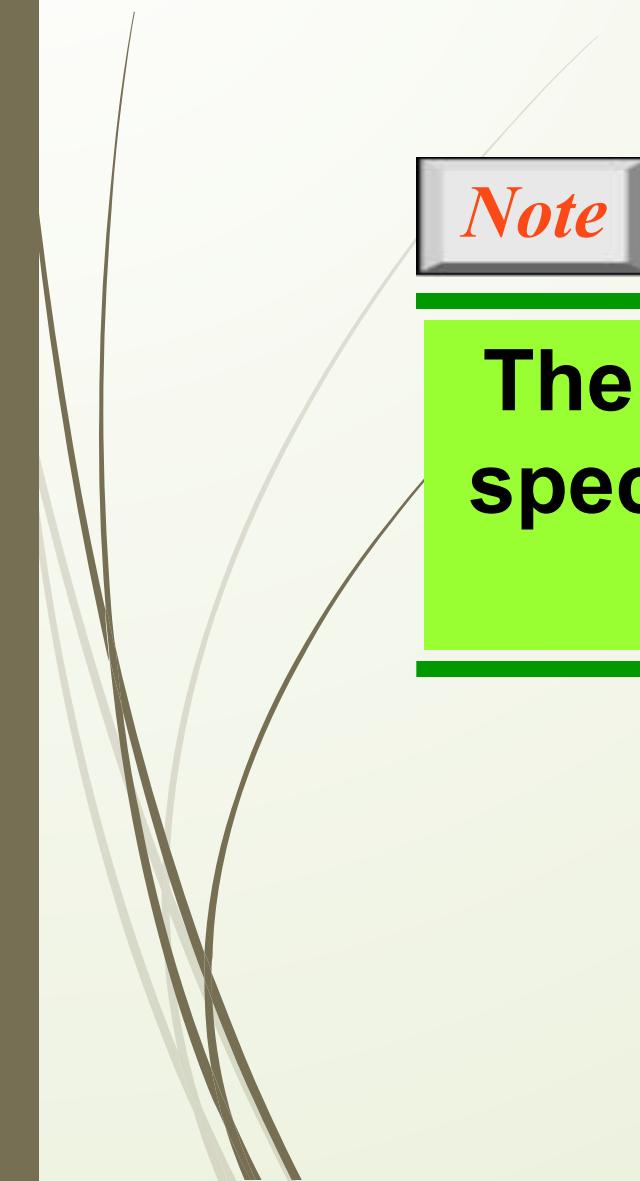




Note

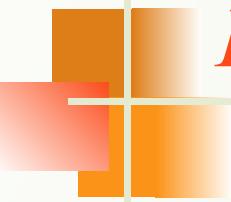
The least significant bit of the first byte defines the type of address.

**If the bit is 0, the address is unicast;
otherwise, it is multicast.**



Note

The broadcast destination address is a special case of the multicast address in which all bits are 1s.



Example 6.1

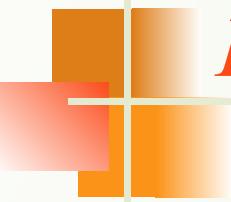
Define the type of the following destination addresses:

- a.* **4A:30:10:21:10:1A**
- b.* **47:20:1B:2E:08:EE**
- c.* **FF:FF:FF:FF:FF:FF**

Solution

To find the type of the address, we need to look at the second hexadecimal digit from the left. If it is even, the address is unicast. If it is odd, the address is multicast. If all digits are F's, the address is broadcast. Therefore, we have the following:

- a.* **This is a unicast address because A in binary is 1010.**
- b.* **This is a multicast address because 7 in binary is 0111.**
- c.* **This is a broadcast address because all digits are F's.**



Example 6.2

Show how the address 47:20:1B:2E:08:EE is sent out on line.

Solution

The address is sent left-to-right, byte by byte; for each byte, it is sent right-to-left, bit by bit, as shown below:



11100010 00000100 11011000 01110100 00010000 01110111

Figure 6.8 Categories of Standard Ethernet

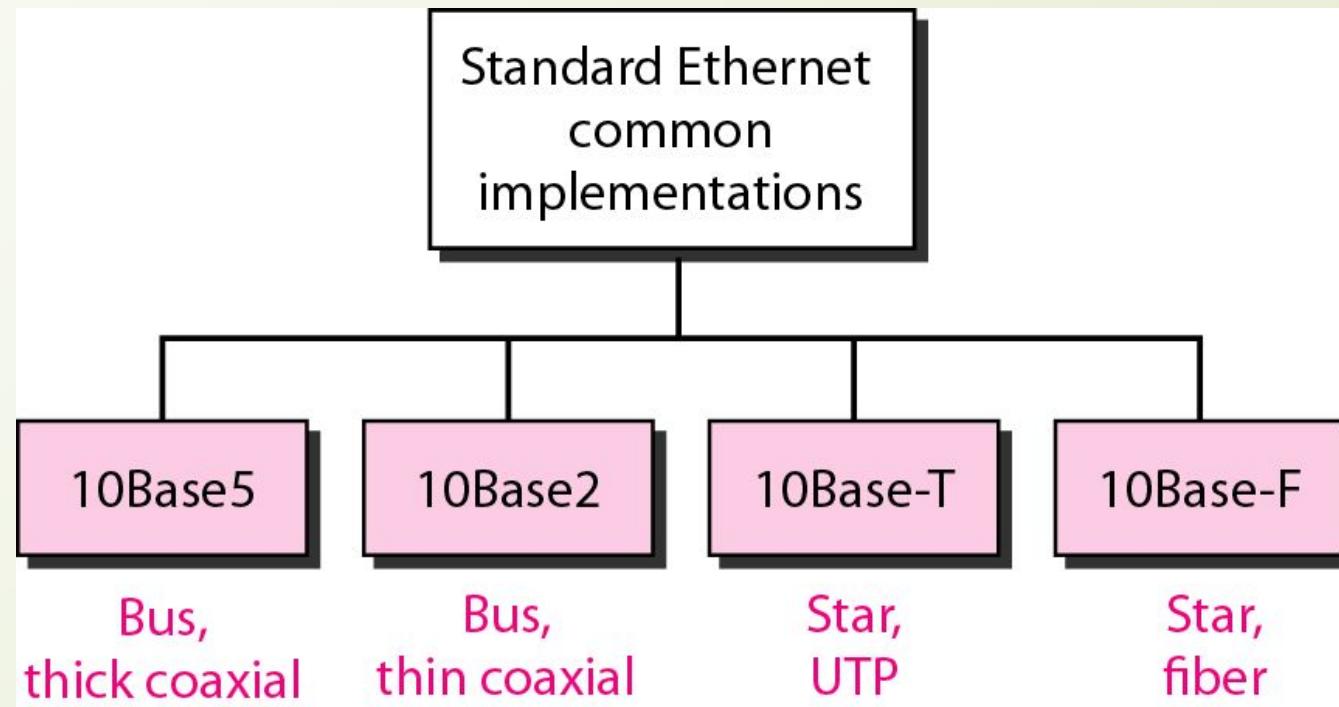


Figure 6.9 Encoding in a Standard Ethernet implementation

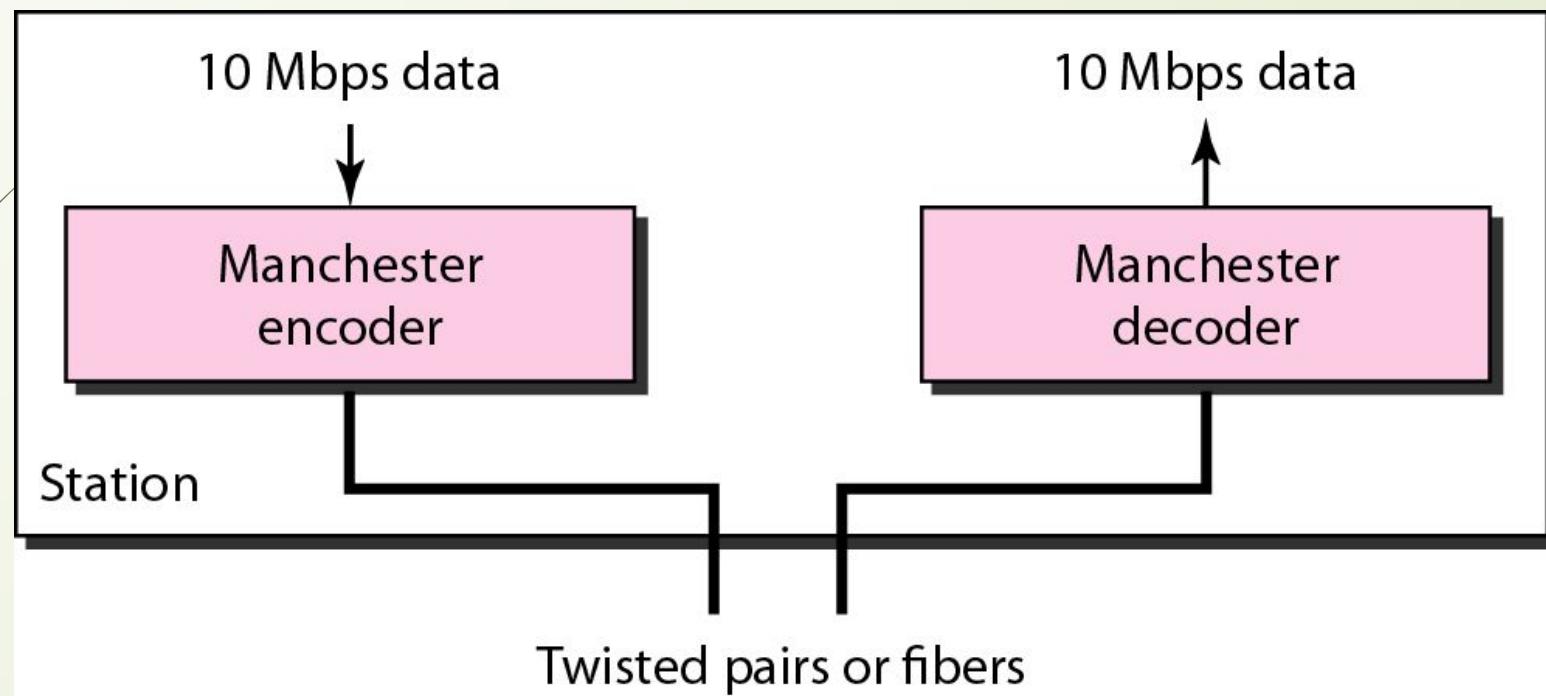


Figure 6.10 *10Base5 implementation*

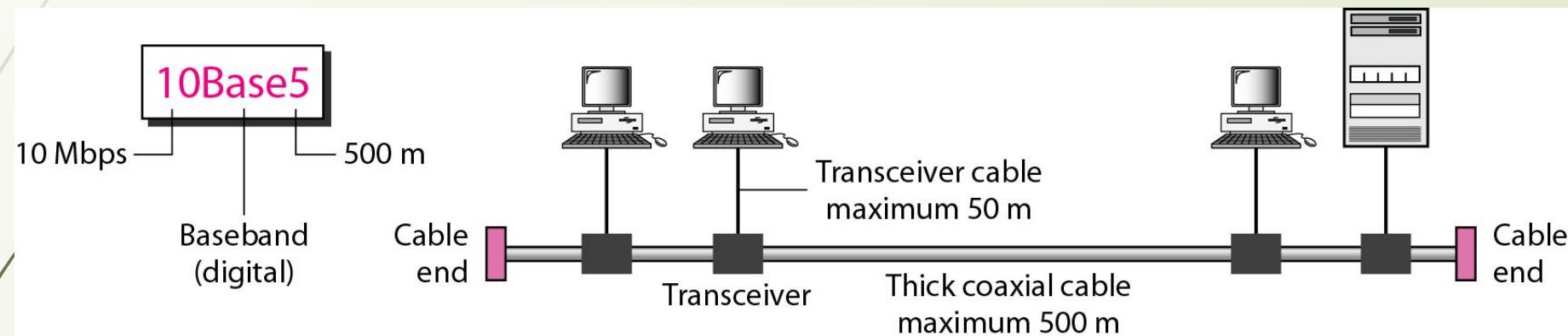


Figure 6.11 *10Base2 implementation*

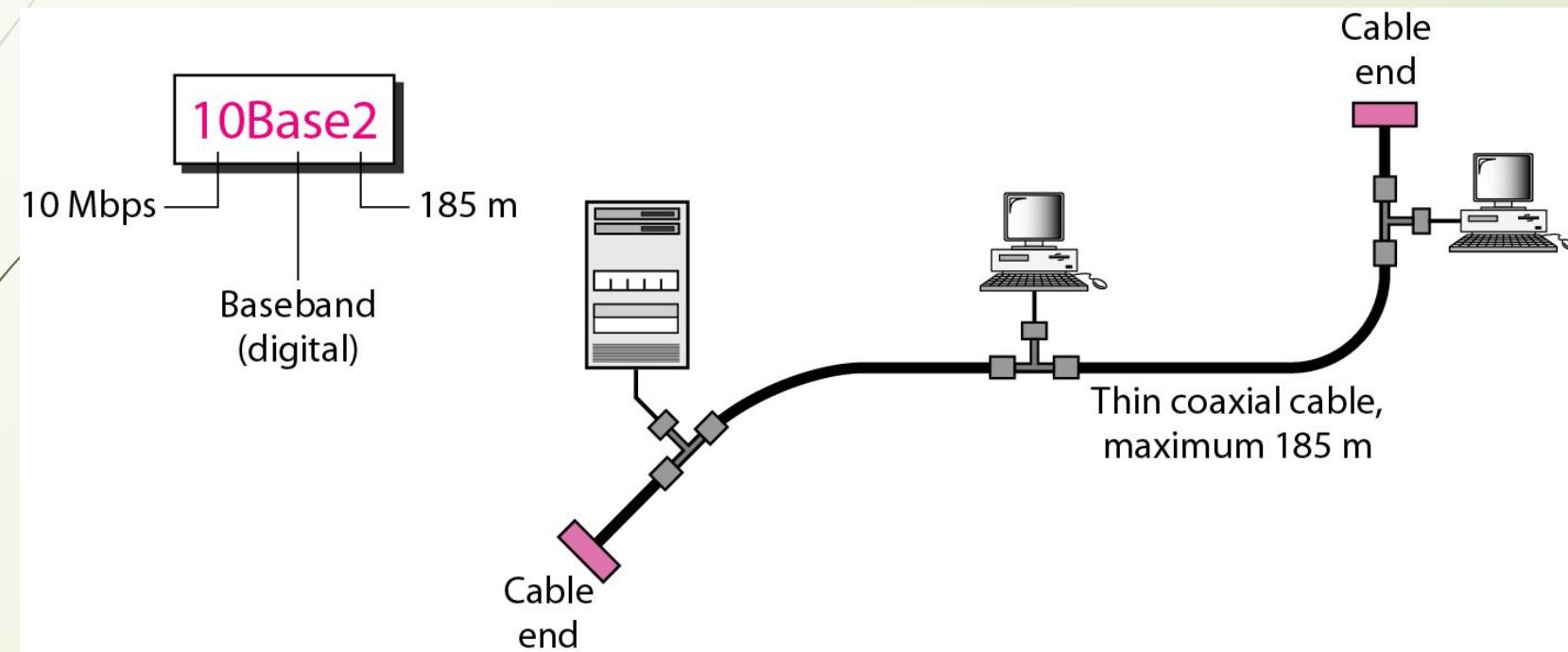


Figure 6.12 *10Base-T implementation*

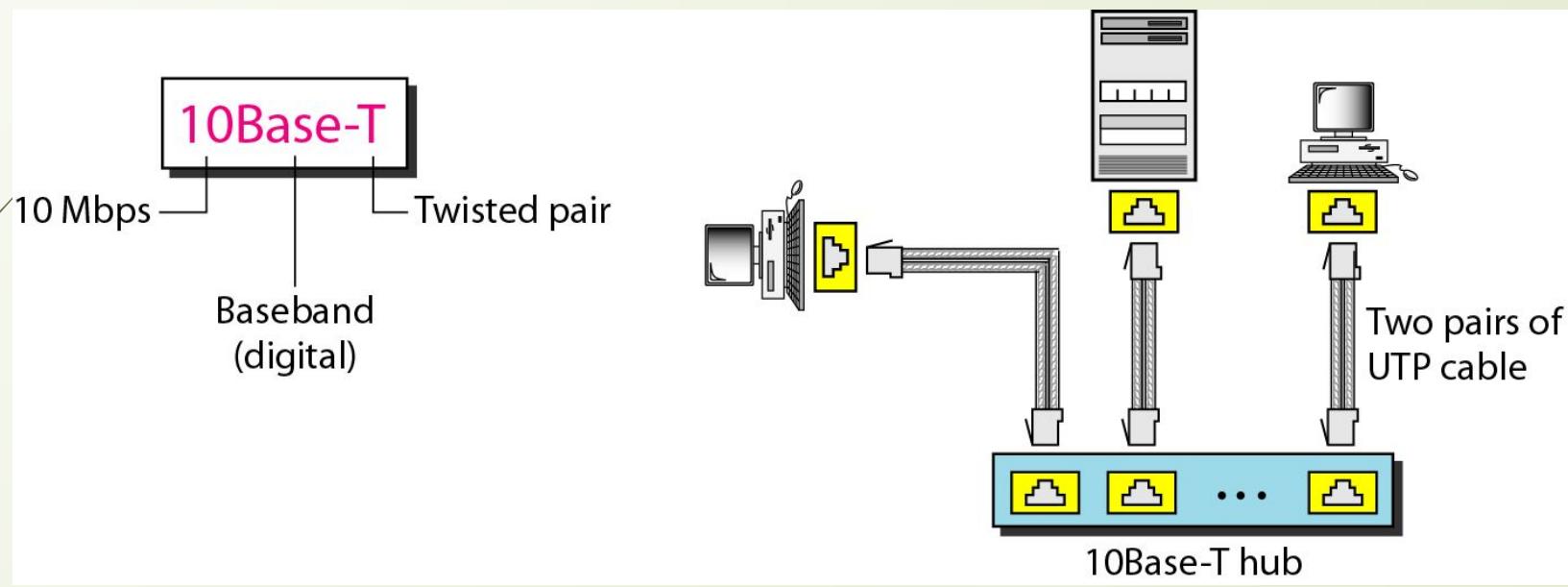


Figure 6.13 *10Base-F implementation*

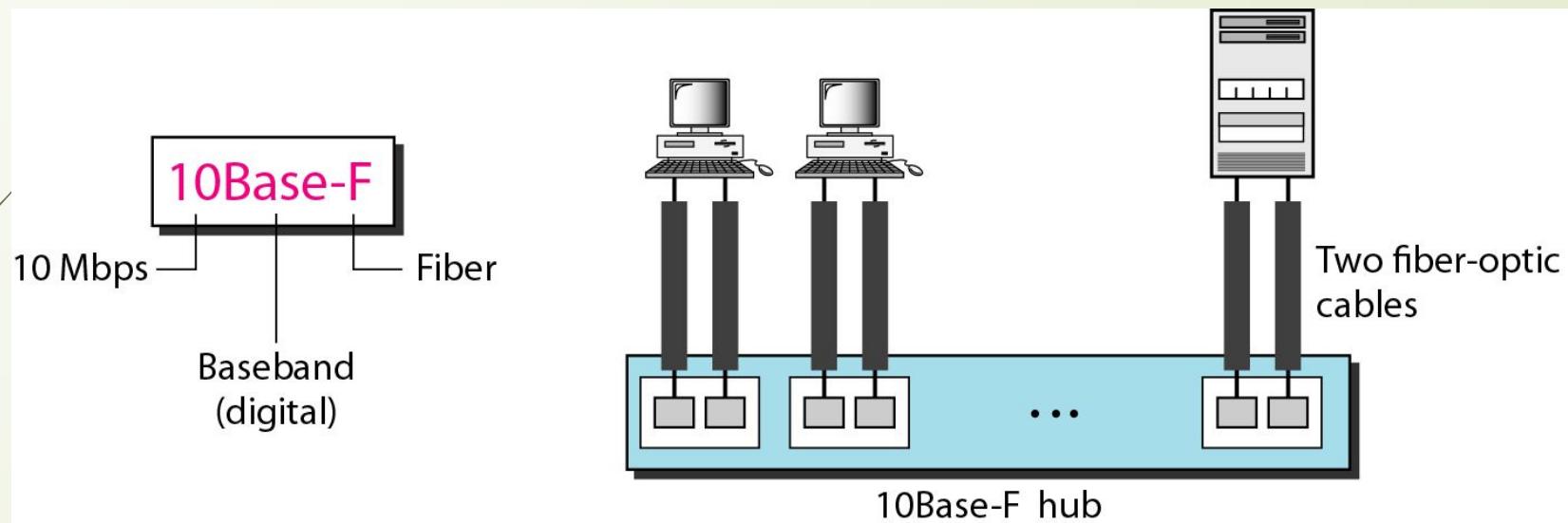


Table 6.1 *Summary of Standard Ethernet implementations*

<i>Characteristics</i>	<i>10Base5</i>	<i>10Base2</i>	<i>10Base-T</i>	<i>10Base-F</i>
Media	Thick coaxial cable	Thin coaxial cable	2 UTP	2 Fiber
Maximum length	500 m	185 m	100 m	2000 m
Line encoding	Manchester	Manchester	Manchester	Manchester

6-3 CHANGES IN THE STANDARD

The 10-Mbps Standard Ethernet has gone through several changes before moving to the higher data rates. These changes actually opened the road to the evolution of the Ethernet to become compatible with other high-data-rate LANs.

Topics discussed in this section:

Bridged Ethernet

Switched Ethernet

Full-Duplex Ethernet

Figure 6.14 *Sharing bandwidth*

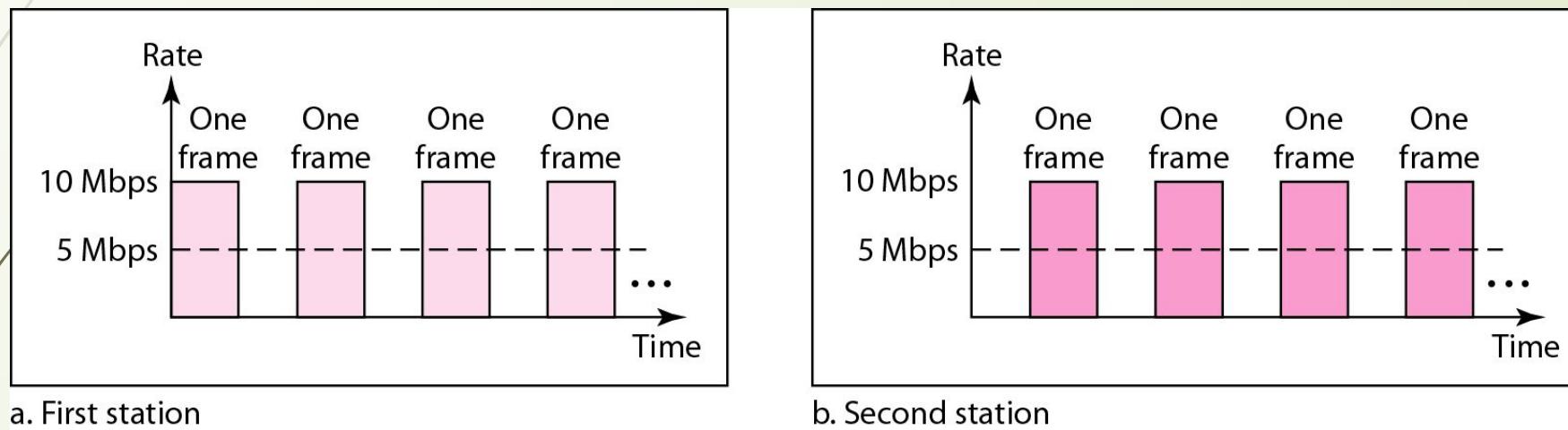
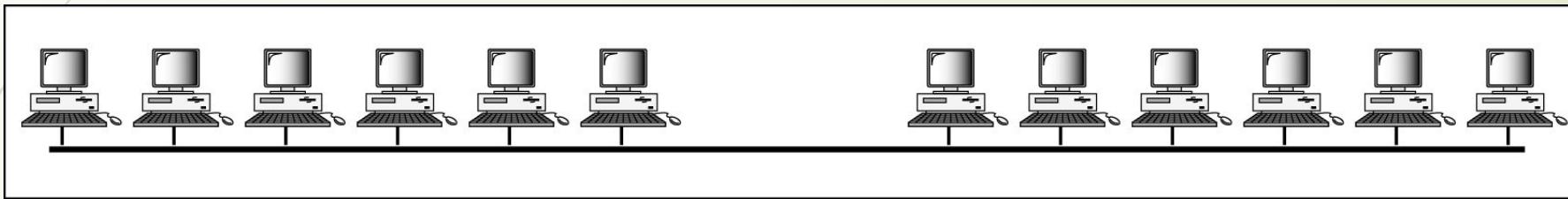
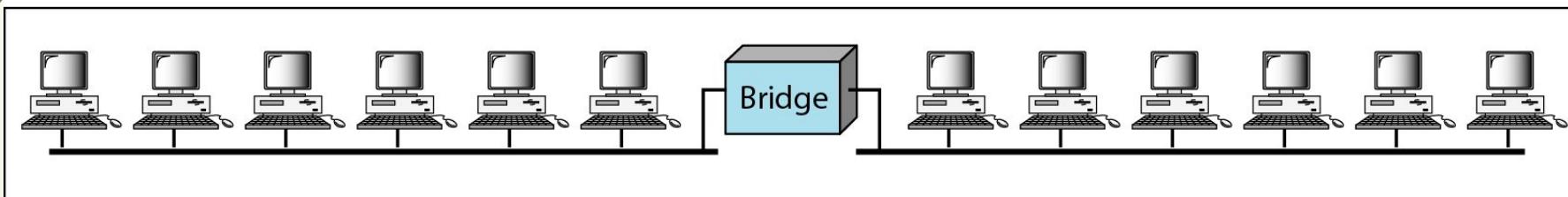


Figure 6.15 *A network with and without a bridge*



a. Without bridging



b. With bridging

Figure 6.16 Collision domains in an unbridged network and a bridged network

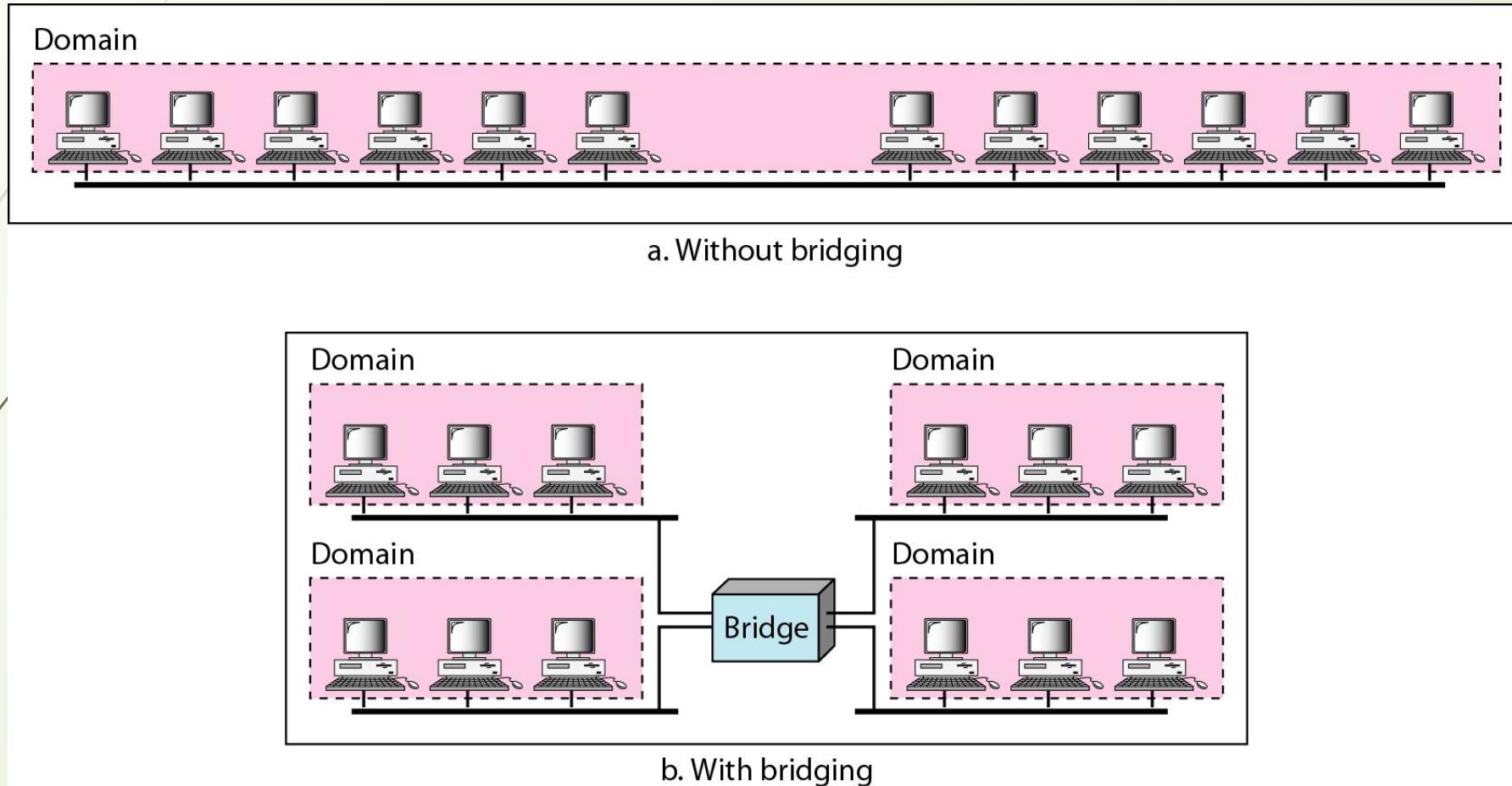


Figure 6.17 *Switched Ethernet*

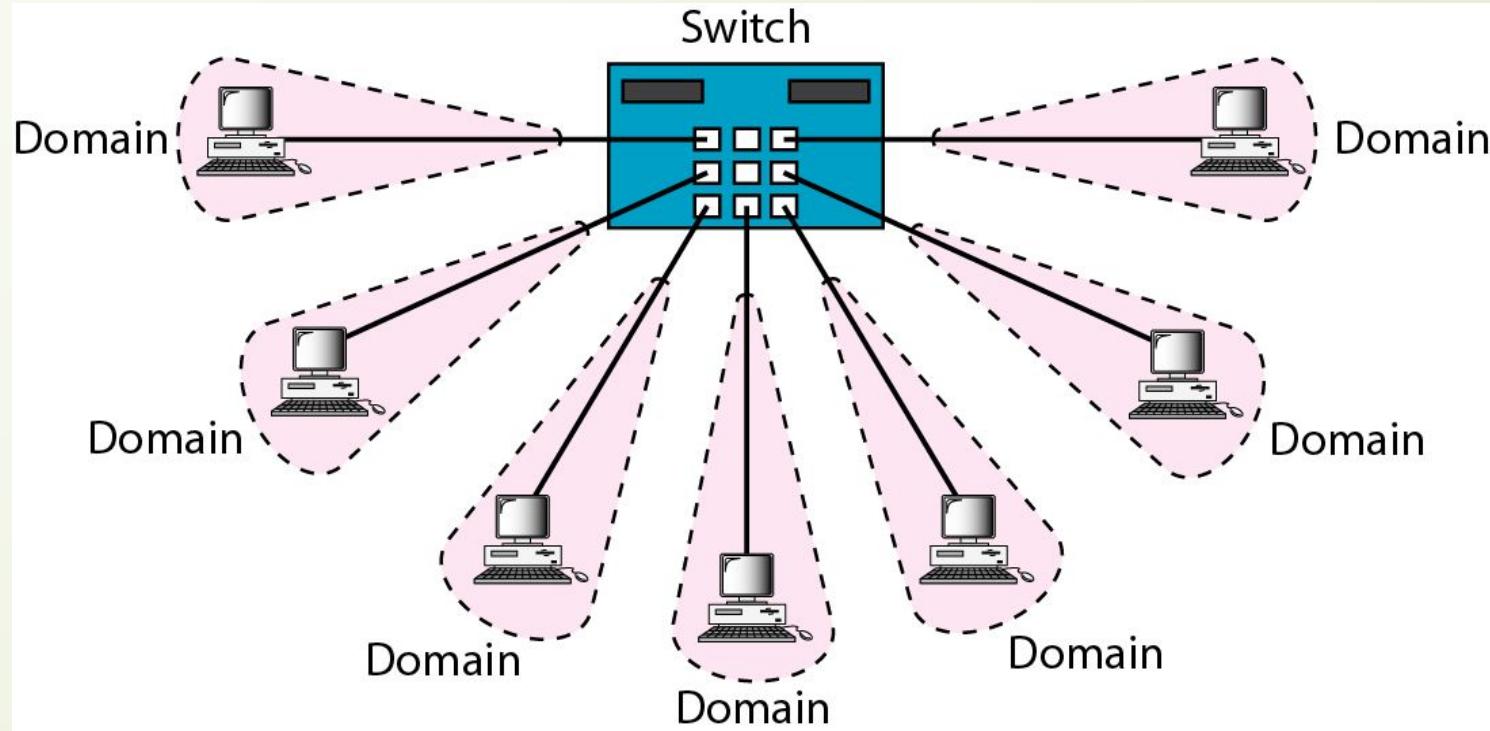
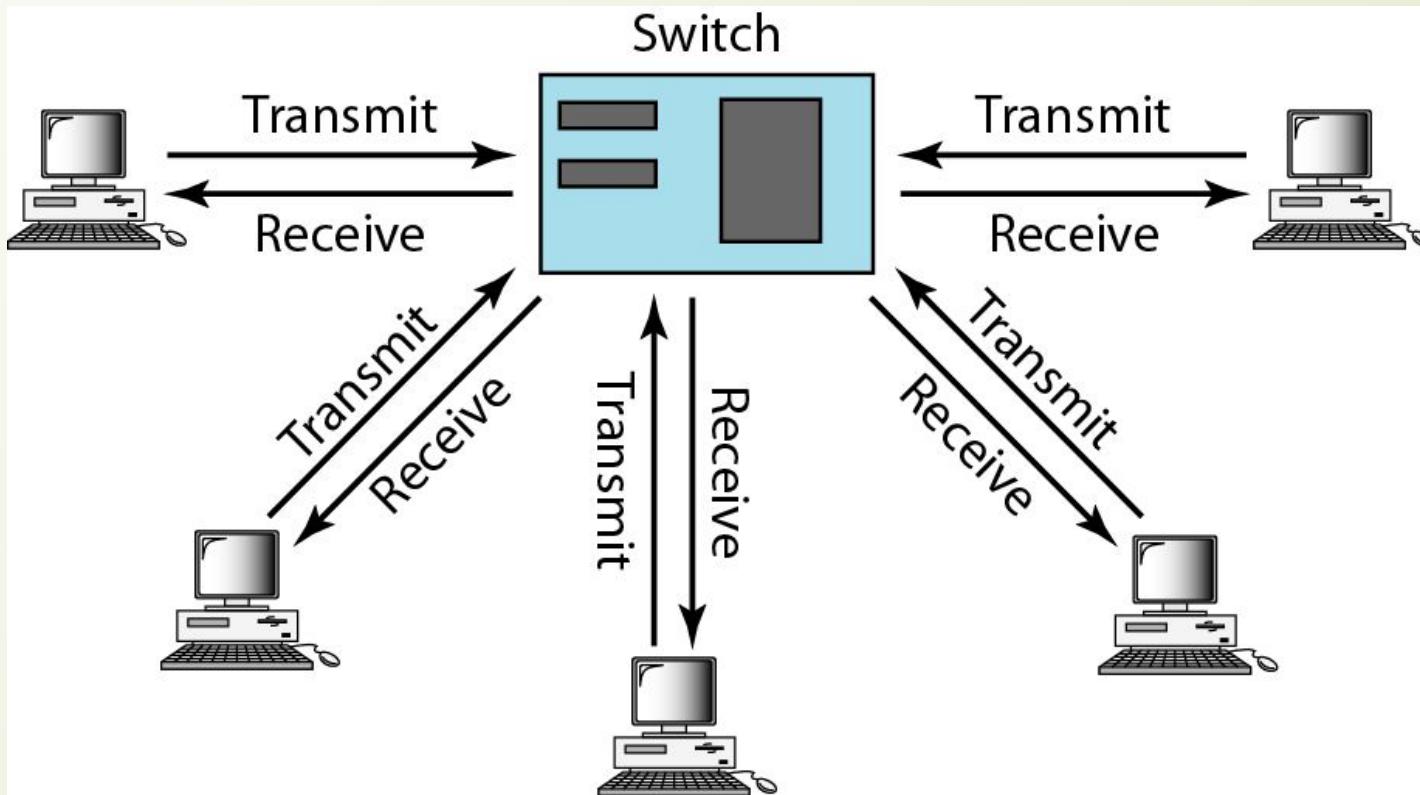


Figure 6.18 Full-duplex switched Ethernet



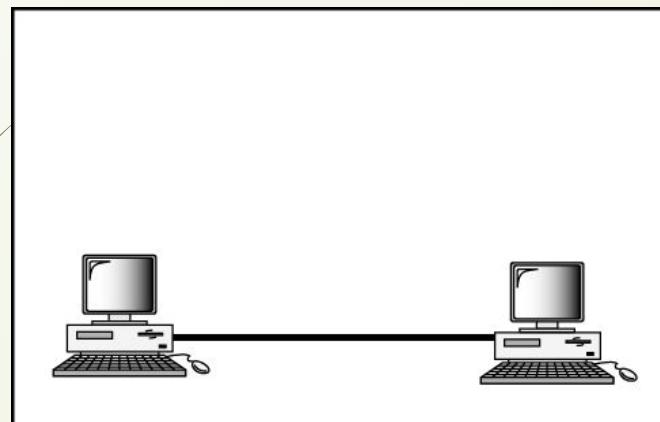
6-4 FAST ETHERNET

Fast Ethernet was designed to compete with LAN protocols such as FDDI or Fiber Channel. IEEE created Fast Ethernet under the name 802.3u. Fast Ethernet is backward-compatible with Standard Ethernet, but it can transmit data 10 times faster at a rate of 100 Mbps.

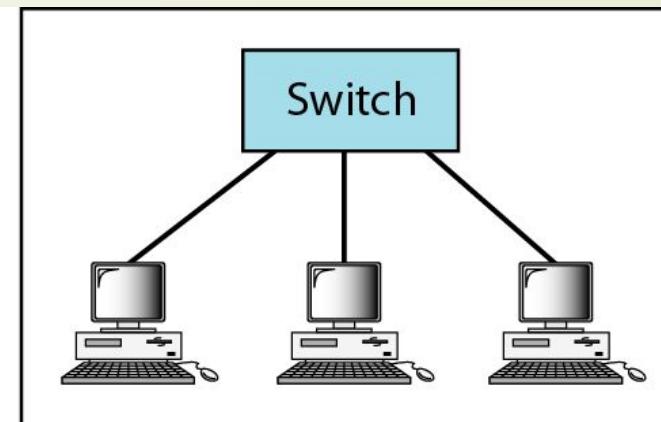
Topics discussed in this section:

MAC Sublayer
Physical Layer

Figure 6.19 *Fast Ethernet topology*



a. Point-to-point



b. Star

Figure 6.20 *Fast Ethernet implementations*

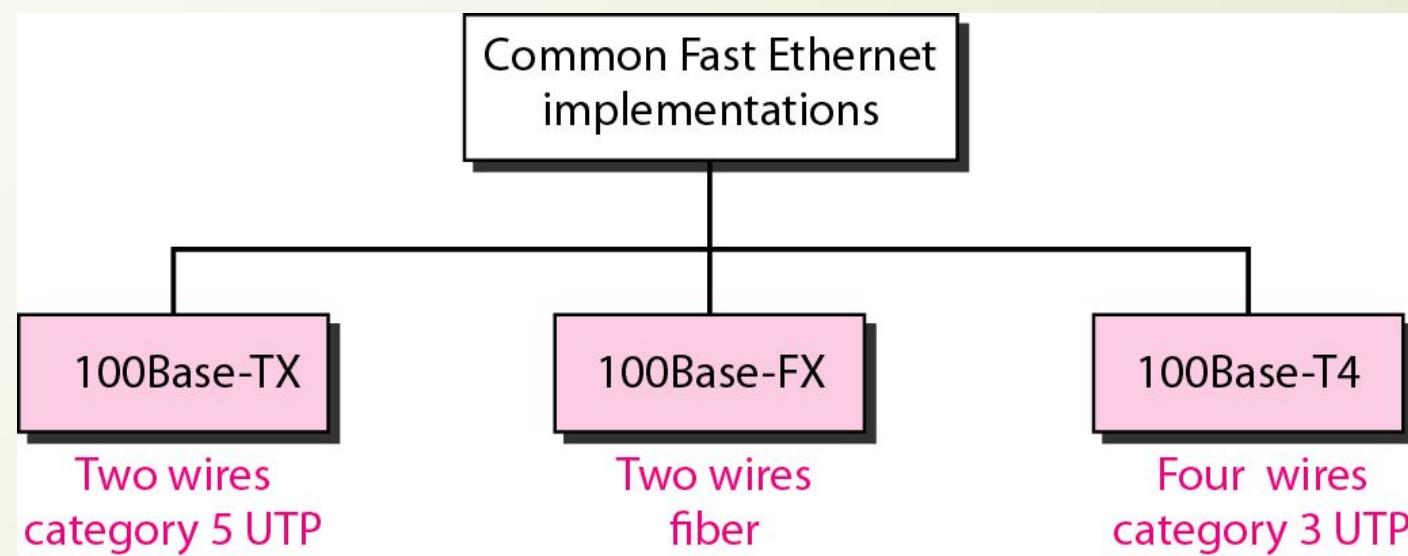


Figure 6.21 Encoding for Fast Ethernet implementation

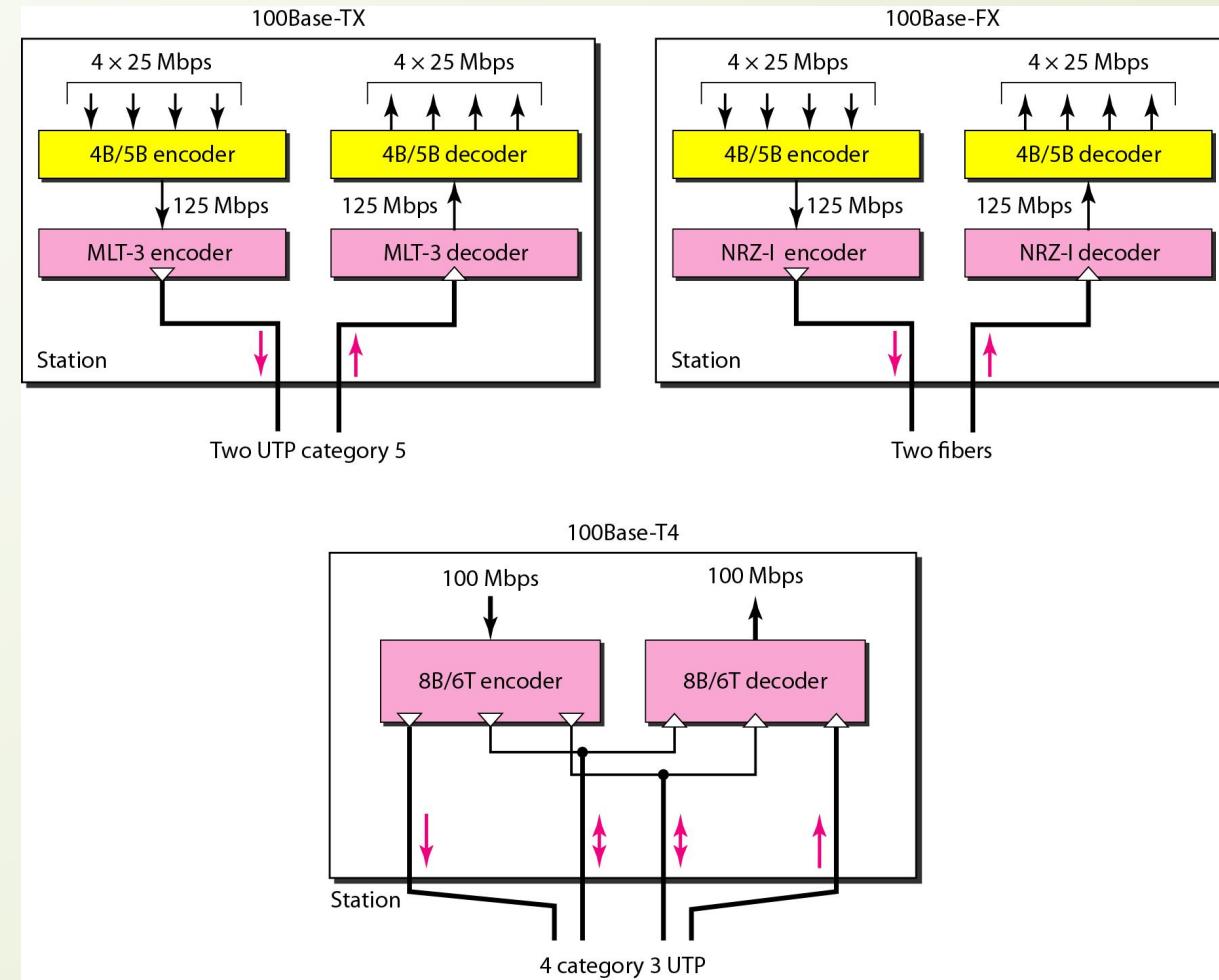


Table 6.2 *Summary of Fast Ethernet implementations*

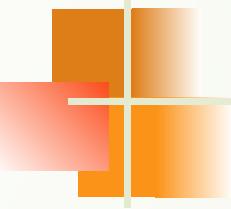
<i>Characteristics</i>	<i>100Base-TX</i>	<i>100Base-FX</i>	<i>100Base-T4</i>
Media	Cat 5 UTP or STP	Fiber	Cat 4 UTP
Number of wires	2	2	4
Maximum length	100 m	100 m	100 m
Block encoding	4B/5B	4B/5B	
Line encoding	MLT-3	NRZ-I	8B/6T

6-5 GIGABIT ETHERNET

The need for an even higher data rate resulted in the design of the Gigabit Ethernet protocol (1000 Mbps). The IEEE committee calls the standard 802.3z.

Topics discussed in this section:

MAC Sublayer
Physical Layer
Ten-Gigabit Ethernet



Note

In the full-duplex mode of Gigabit Ethernet, there is no collision; the maximum length of the cable is determined by the signal attenuation in the cable.

Figure 6.22 *Topologies of Gigabit Ethernet*

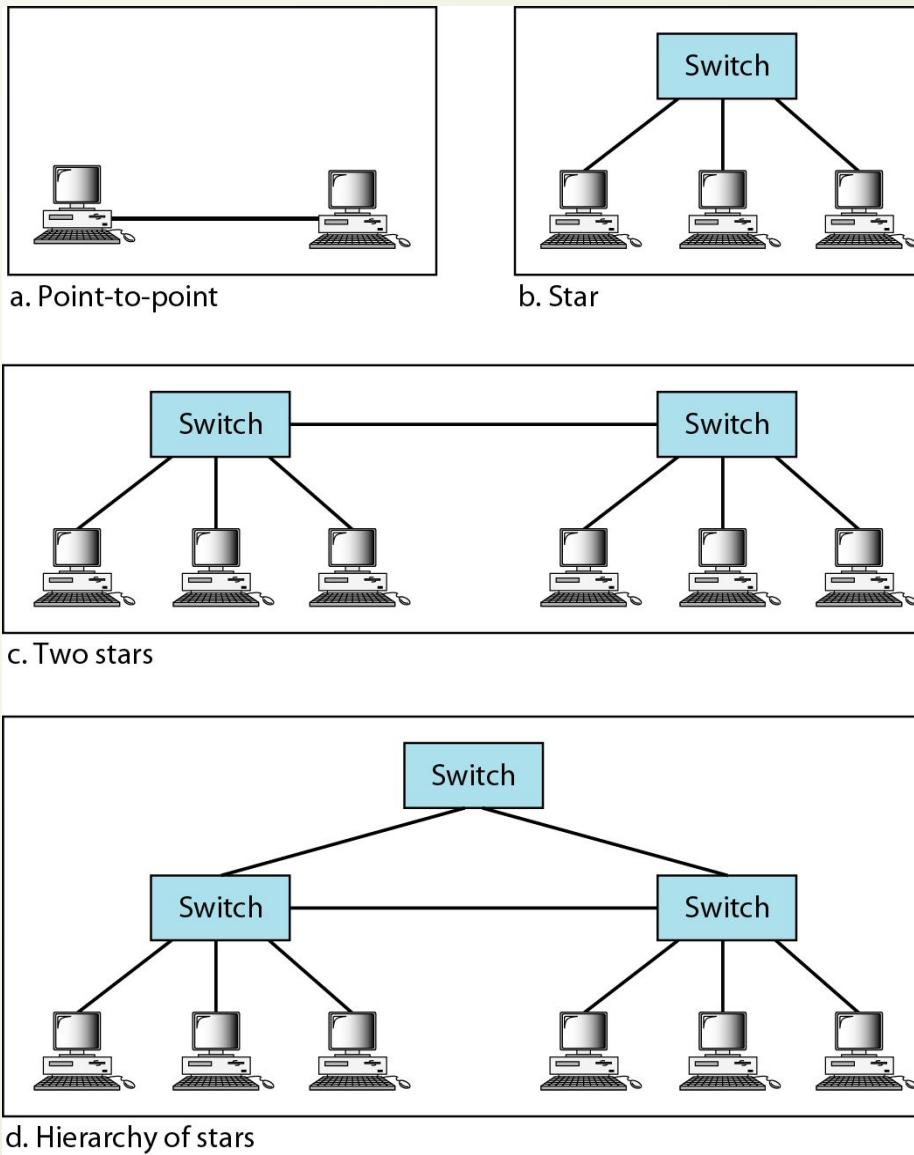


Figure 6.23 *Gigabit Ethernet implementations*

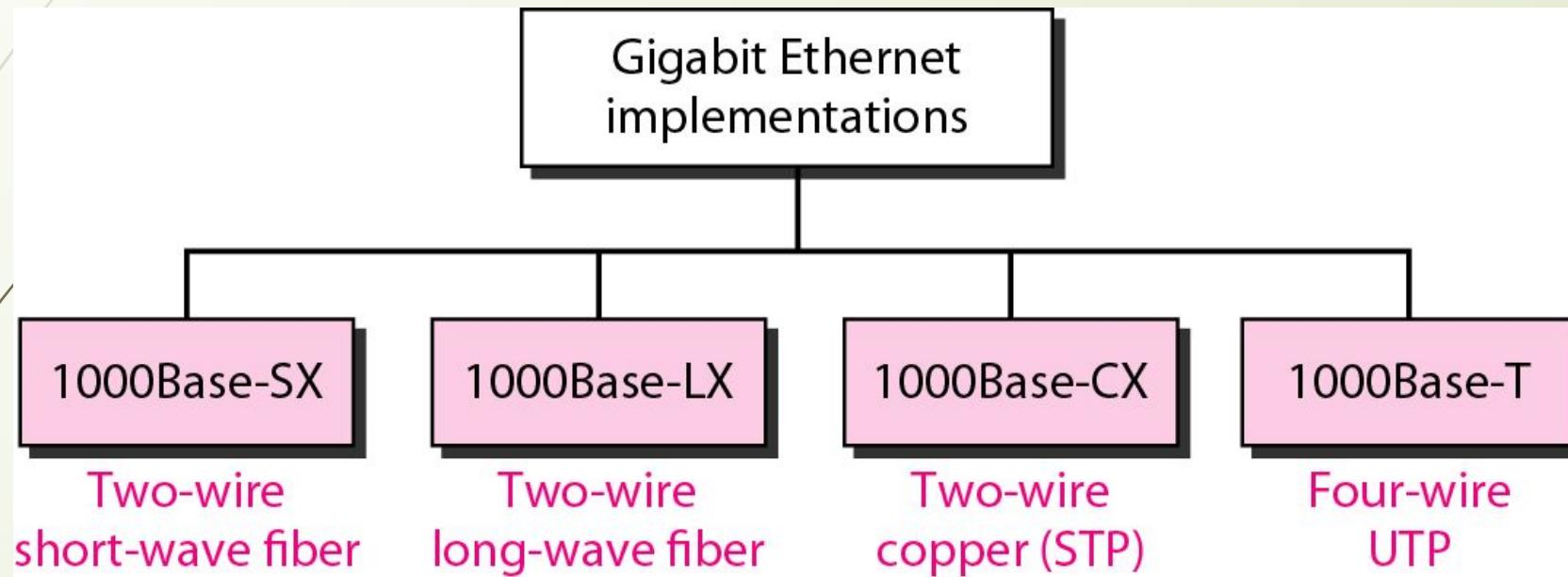


Figure 6.24 Encoding in Gigabit Ethernet implementations

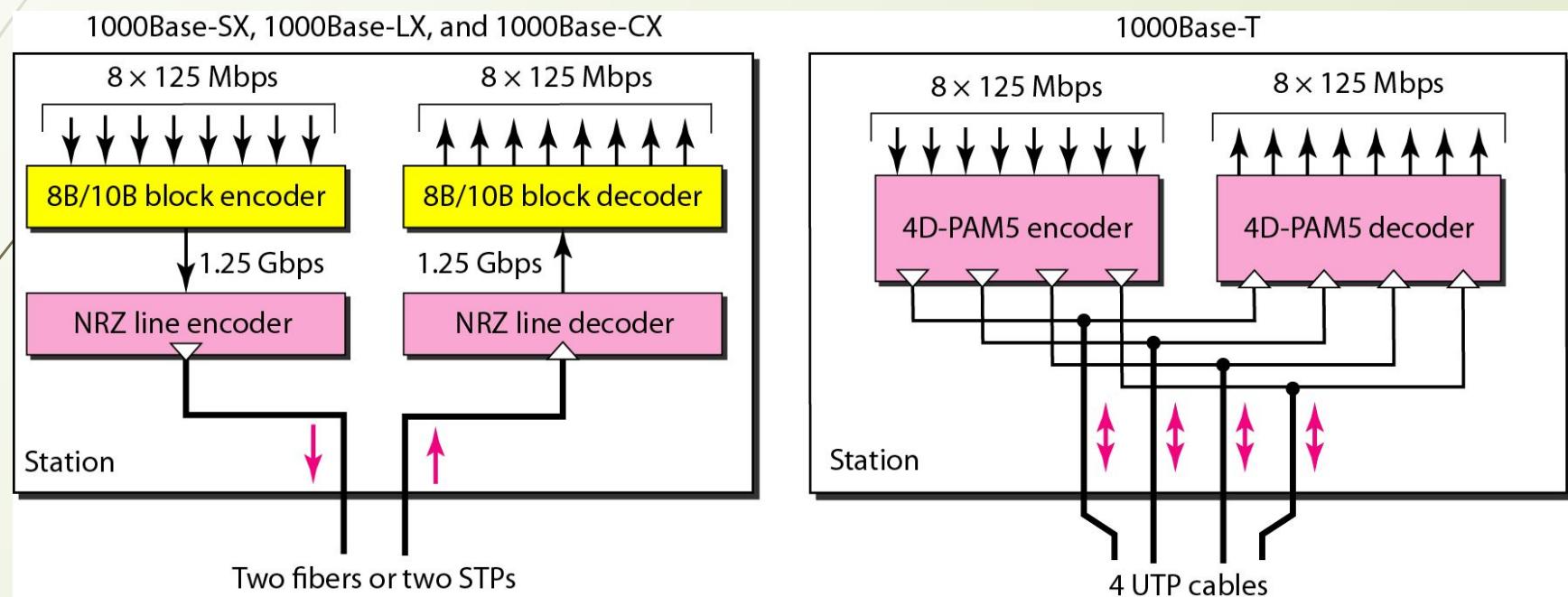


Table 6.3 *Summary of Gigabit Ethernet implementations*

<i>Characteristics</i>	<i>1000Base-SX</i>	<i>1000Base-LX</i>	<i>1000Base-CX</i>	<i>1000Base-T</i>
Media	Fiber short-wave	Fiber long-wave	STP	Cat 5 UTP
Number of wires	2	2	2	4
Maximum length	550 m	5000 m	25 m	100 m
Block encoding	8B/10B	8B/10B	8B/10B	
Line encoding	NRZ	NRZ	NRZ	4D-PAM5

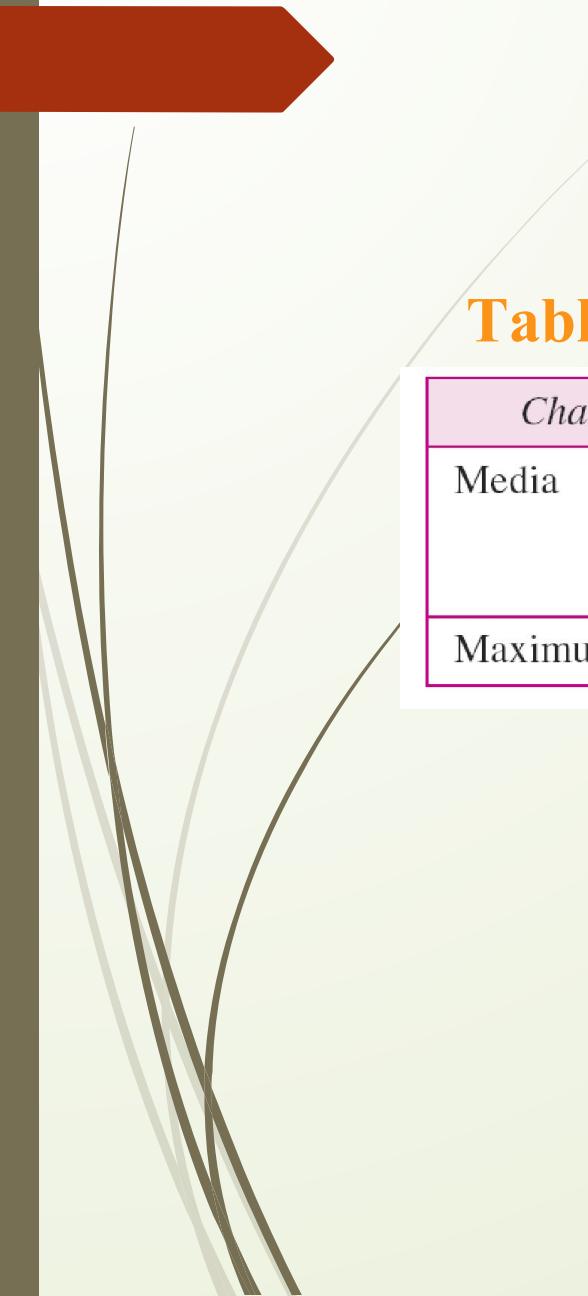
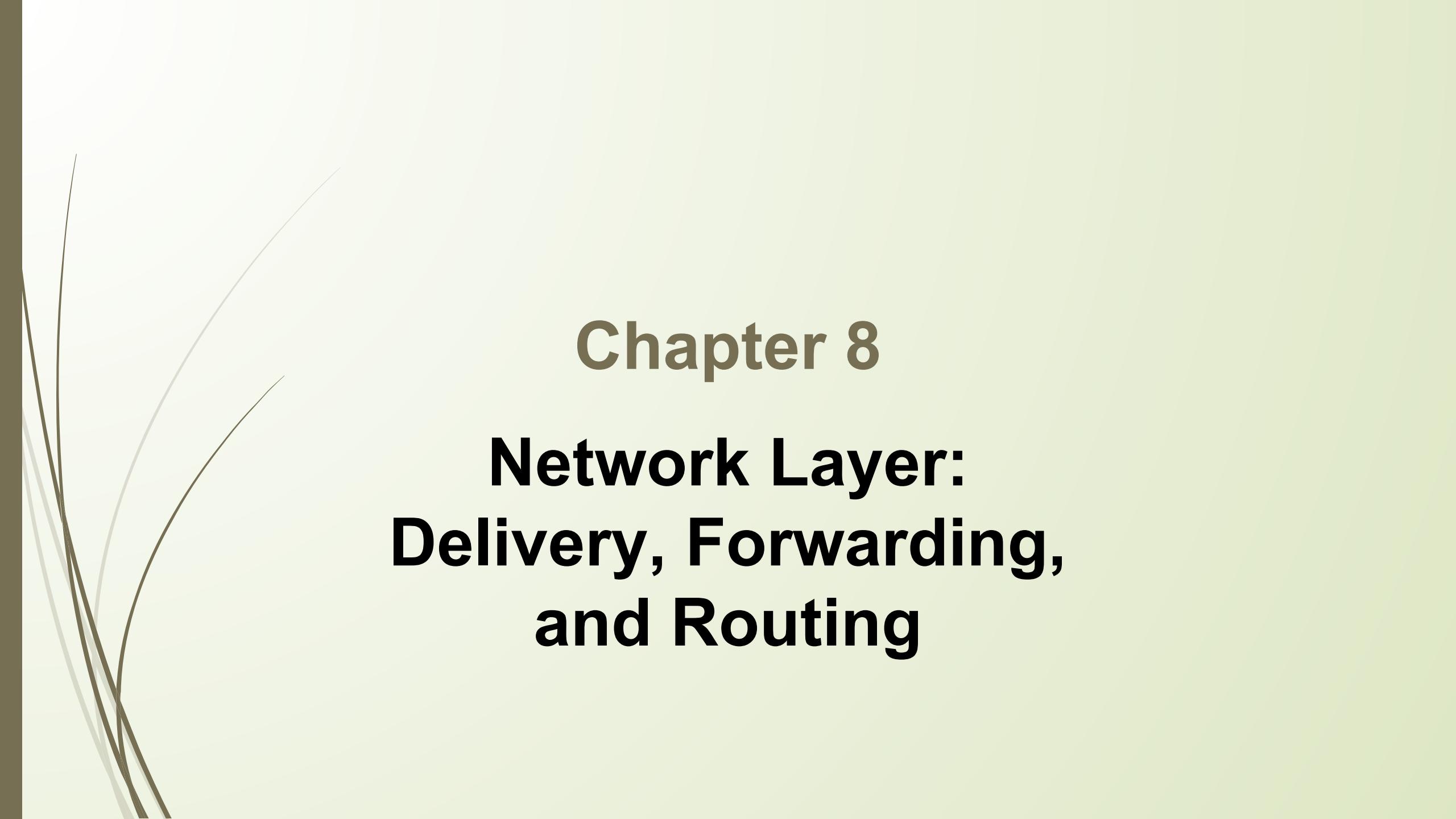


Table 6.4 *Summary of Ten-Gigabit Ethernet implementations*

Characteristics	10GBase-S	10GBase-L	10GBase-E
Media	Short-wave 850-nm multimode	Long-wave 1310-nm single mode	Extended 1550-mm single mode
Maximum length	300 m	10 km	40 km



Chapter 8

Network Layer: Delivery, Forwarding, and Routing

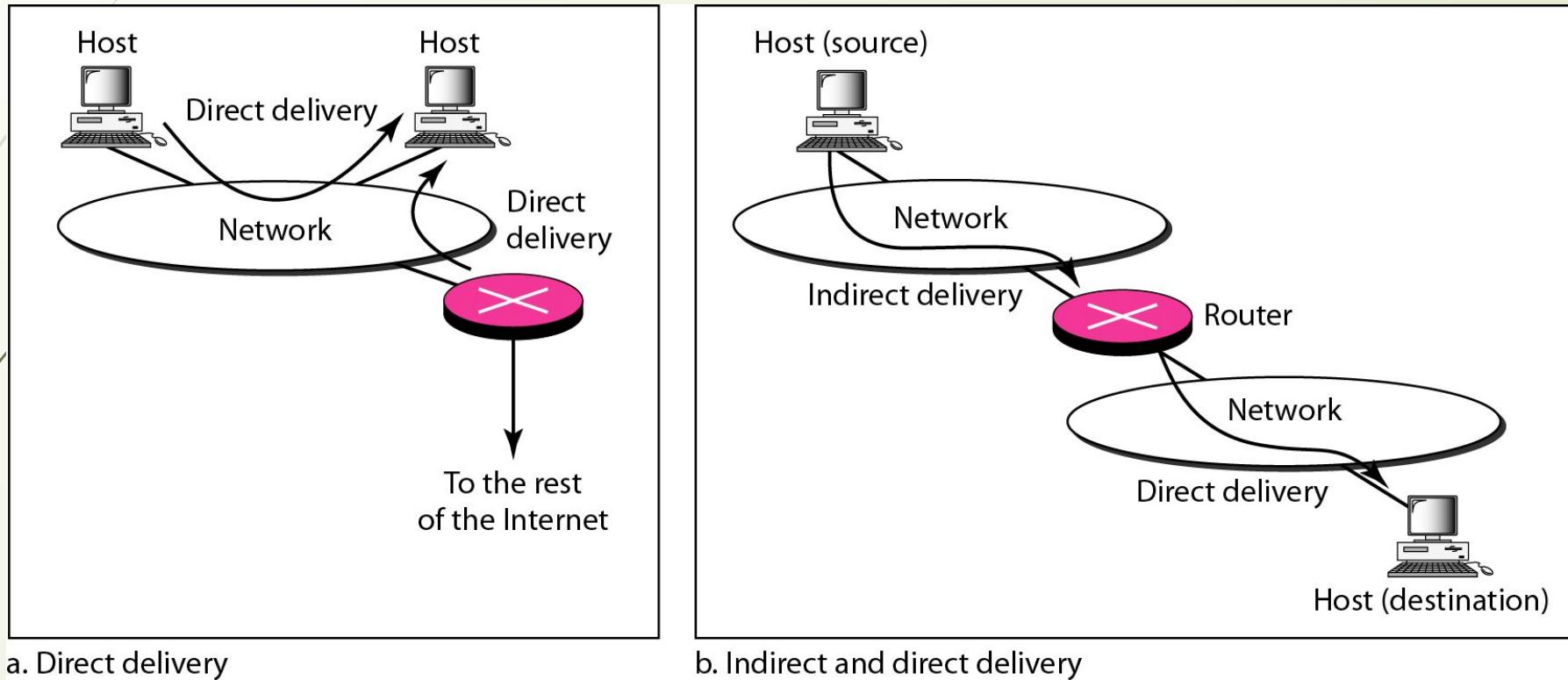
8-1 DELIVERY

The network layer supervises the handling of the packets by the underlying physical networks. We define this handling as the delivery of a packet.

Topics discussed in this section:

Direct Versus Indirect Delivery

Figure 8.1 Direct and indirect delivery



Forwarding means to place the packet in its route to its destination. Forwarding requires a host or a router to have a routing table. When a host has a packet to send or when a router has received a packet to be forwarded, it looks at this table to find the route to the final destination.

Topics discussed in this section:

Forwarding Techniques

Forwarding Process

Routing Table

Figure 8.2 Route method versus next-hop method

a. Routing tables based on route

Destination	Route
Host B	R1, R2, host B

Routing table
for host A

Destination	Route
Host B	R2, host B

Routing table
for R1

Destination	Route
Host B	Host B

Routing table
for R2

b. Routing tables based on next hop

Destination	Next hop
Host B	R1

Destination	Next hop
Host B	R2

Destination	Next hop
Host B	---

Host A



Network

R1

Network

R2

Host B



Network

Figure 8.3 Host-specific versus network-specific method

Routing table for host S based on host-specific method

Destination	Next hop
A	R1
B	R1
C	R1
D	R1

Routing table for host S based on network-specific method

Destination	Next hop
N2	R1

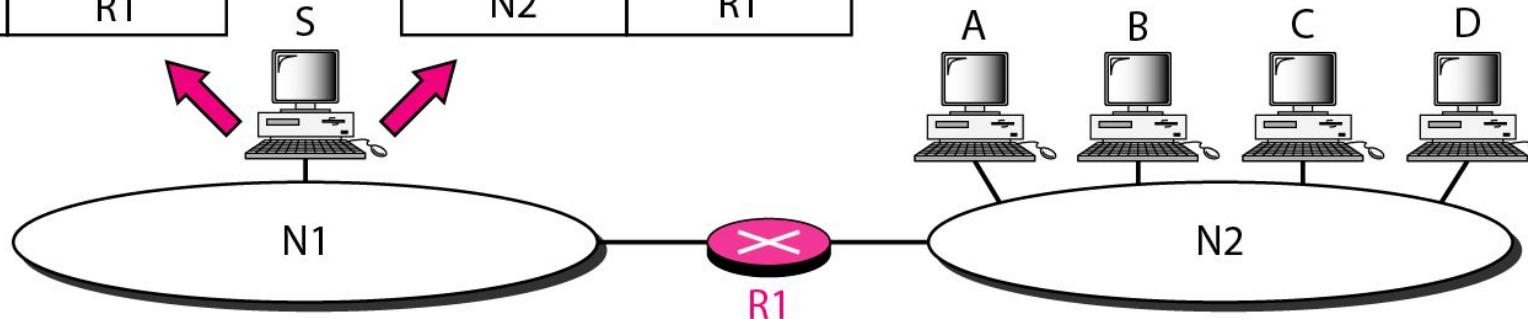


Figure 8.4 *Default method*

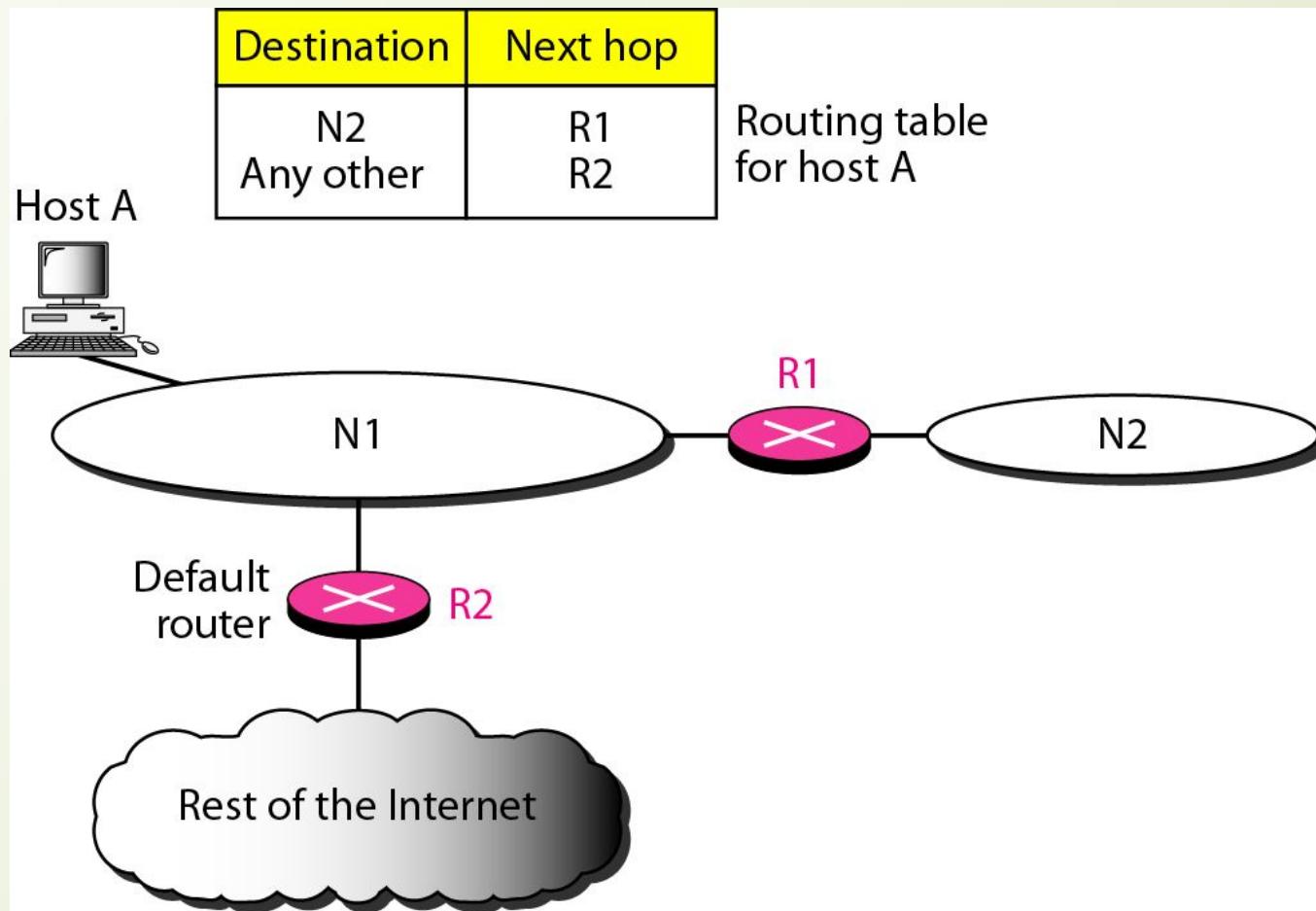
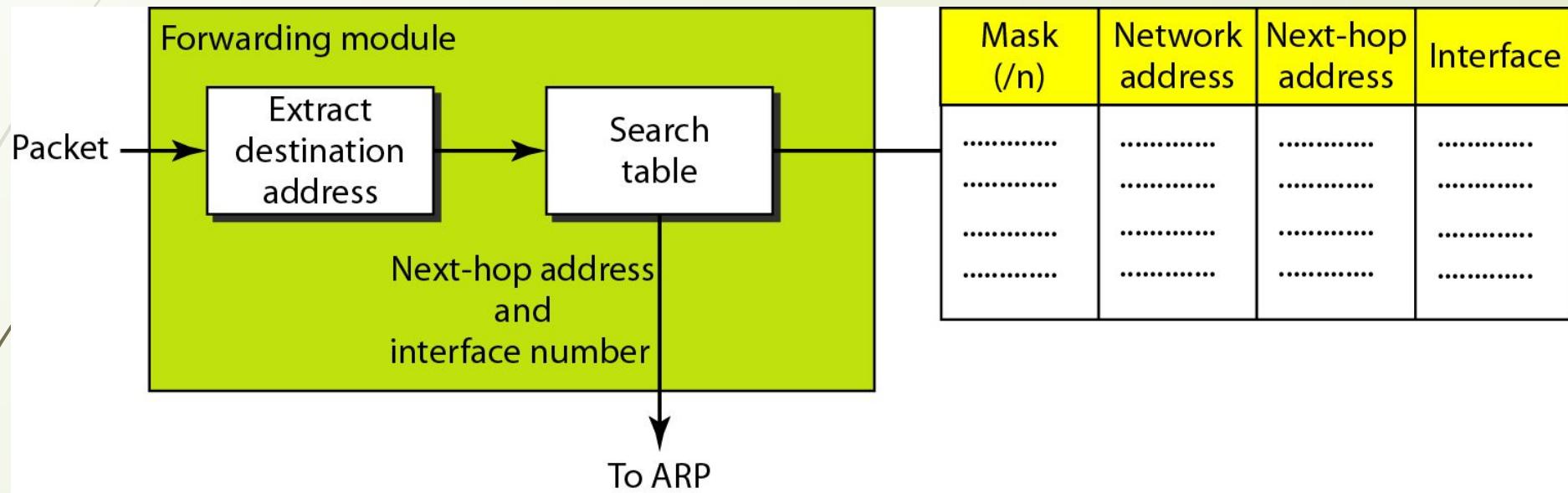
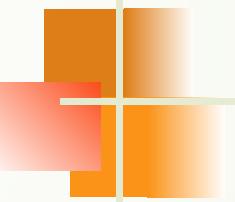


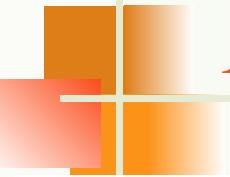
Figure 8.5 Simplified forwarding module in classless address





Note

In classless addressing, we need at least four columns in a routing table.



Example 8.1

Make a routing table for router R1, using the configuration in Figure 8.6.

Solution

Table 8.1 shows the corresponding table.

Figure 8.6 Configuration for Example 8.1

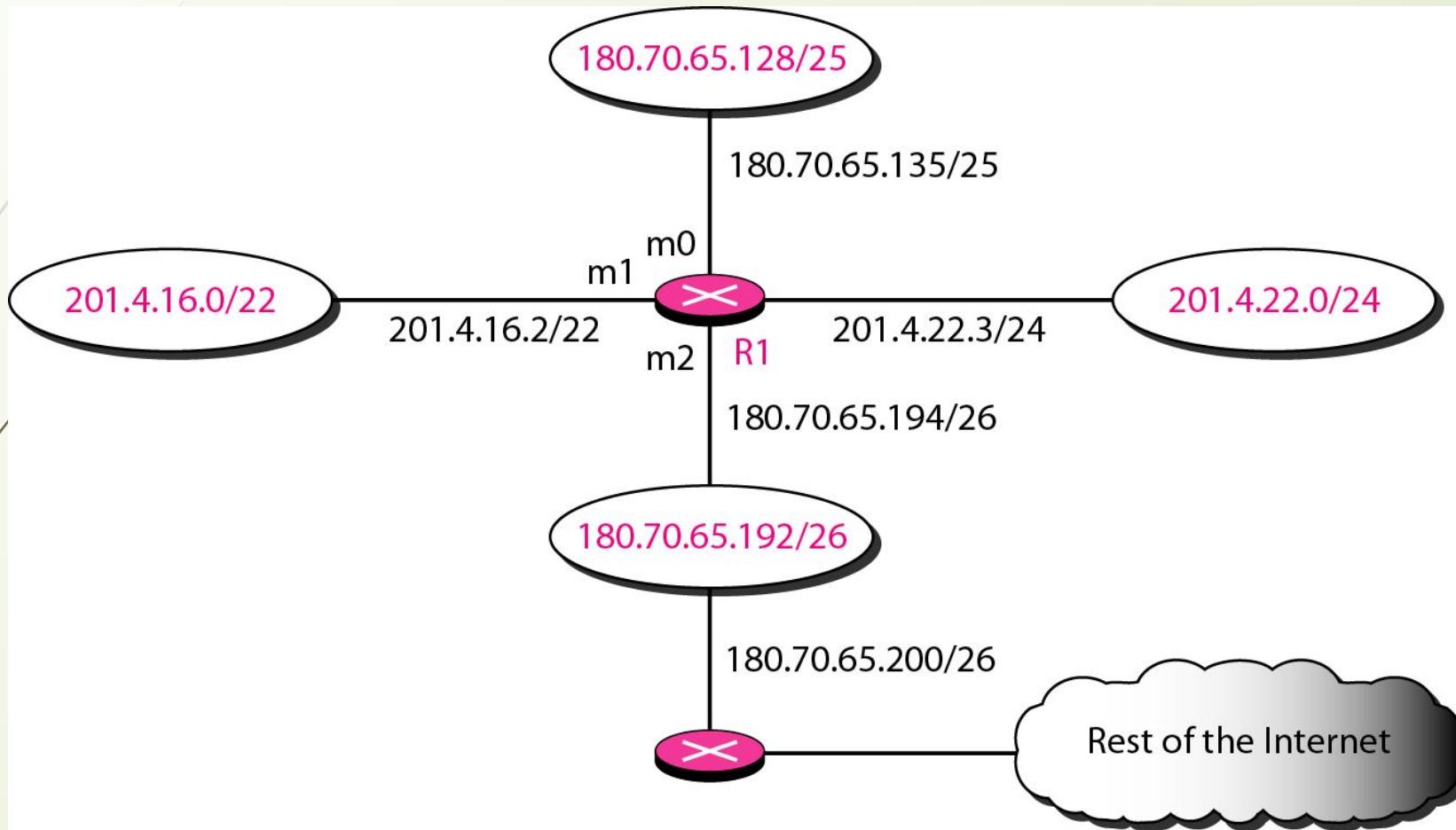
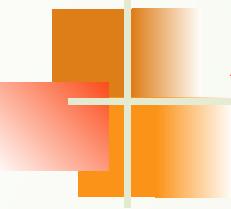


Table 8.1 *Routing table for router R1 in Figure 8.6*

<i>Mask</i>	<i>Network Address</i>	<i>Next Hop</i>	<i>Interface</i>
/26	180.70.65.192	—	m2
/25	180.70.65.128	—	m0
/24	201.4.22.0	—	m3
/22	201.4.16.0	m1
Any	Any	180.70.65.200	m2



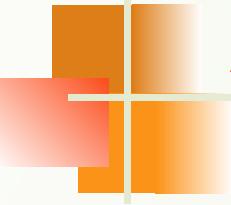
Example 8.2

Show the forwarding process if a packet arrives at R1 in Figure 8.6 with the destination address 180.70.65.140.

Solution

The router performs the following steps:

- 1. The first mask (/26) is applied to the destination address. The result is 180.70.65.128, which does not match the corresponding network address.*
- 2. The second mask (/25) is applied to the destination address. The result is 180.70.65.128, which matches the corresponding network address. The next-hop address and the interface number m0 are passed to ARP for further processing.*



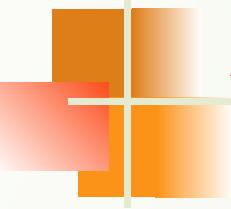
Example 8.3

Show the forwarding process if a packet arrives at R1 in Figure 8.6 with the destination address 201.4.22.35.

Solution

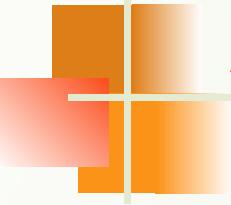
The router performs the following steps:

- 1. The first mask (/26) is applied to the destination address. The result is 201.4.22.0, which does not match the corresponding network address.*
- 2. The second mask (/25) is applied to the destination address. The result is 201.4.22.0, which does not match the corresponding network address (row 2).*



Example 8.3 (continued)

3. The third mask (/24) is applied to the destination address. The result is 201.4.22.0, which matches the corresponding network address. The destination address of the packet and the interface number m3 are passed to ARP.

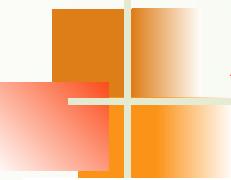


Example 8.4

Show the forwarding process if a packet arrives at R1 in Figure 8.6 with the destination address 18.24.32.78.

Solution

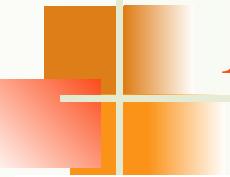
This time all masks are applied, one by one, to the destination address, but no matching network address is found. When it reaches the end of the table, the module gives the next-hop address 180.70.65.200 and interface number m2 to ARP. This is probably an outgoing package that needs to be sent, via the default router, to someplace else in the Internet.



Example 8.5

As an example of hierarchical routing, let us consider Figure 8.9. A regional ISP is granted 16,384 addresses starting from 120.14.64.0. The regional ISP has decided to divide this block into four subblocks, each with 4096 addresses. Three of these subblocks are assigned to three local ISPs; the second subblock is reserved for future use. Note that the mask for each block is /20 because the original block with mask /18 is divided into 4 blocks.

The first local ISP has divided its assigned subblock into 8 smaller blocks and assigned each to a small ISP. Each small ISP provides services to 128 households, each using four addresses.



Example 8.5 (continued)

The second local ISP has divided its block into 4 blocks and has assigned the addresses to four large organizations.

The third local ISP has divided its block into 16 blocks and assigned each block to a small organization. Each small organization has 256 addresses, and the mask is /24.

There is a sense of hierarchy in this configuration. All routers in the Internet send a packet with destination address 120.14.64.0 to 120.14.127.255 to the regional ISP.

Figure 8.9 Hierarchical routing with ISPs

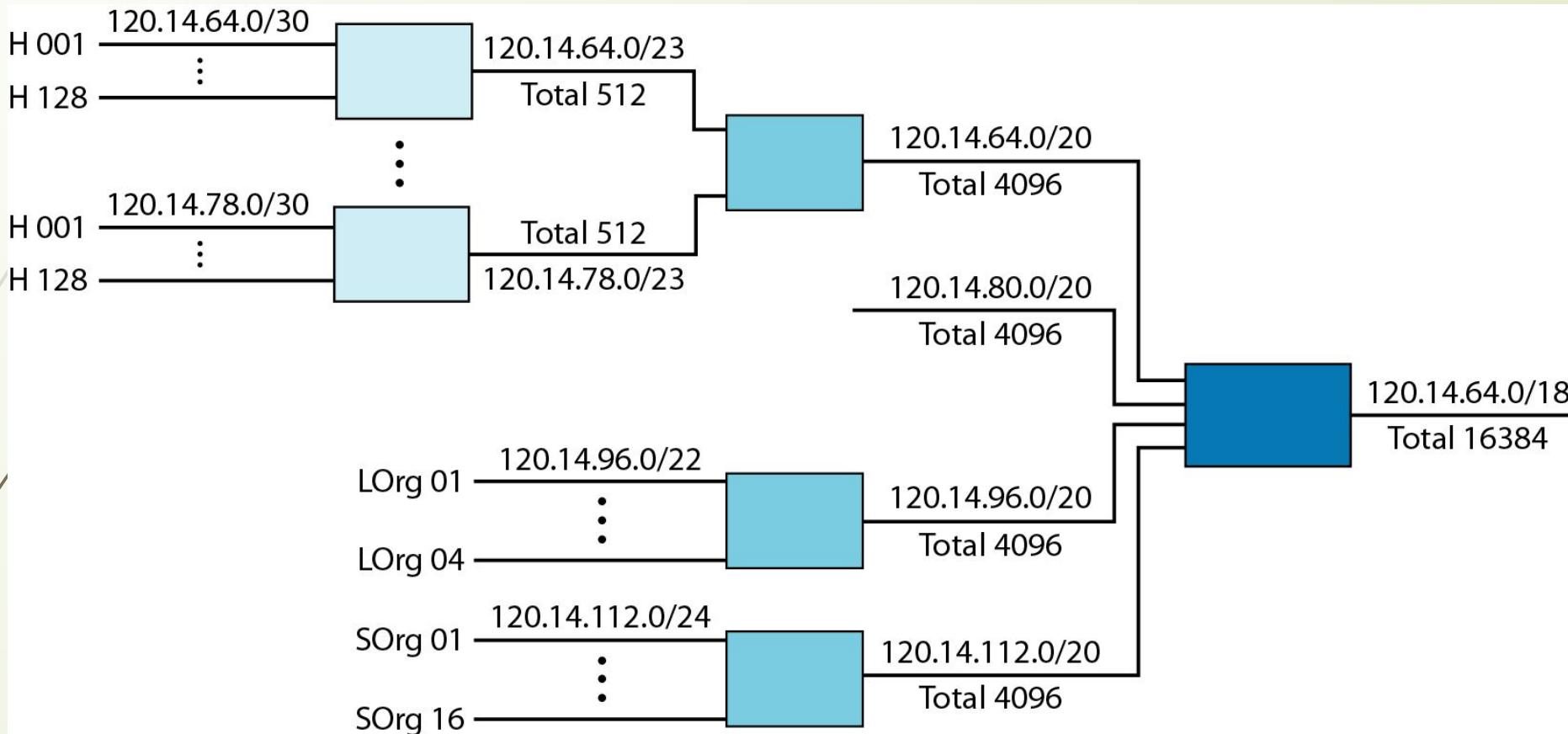


Figure 8.10 *Common fields in a routing table*





A routing table can be either static or dynamic. A static table is one with manual entries. A dynamic table is one that is updated automatically when there is a change somewhere in the Internet. A routing protocol is a combination of rules and procedures that lets routers in the Internet inform each other of changes.

Topics discussed in this section:

Optimization

Intra- and Interdomain Routing

Distance Vector Routing and RIP

Link State Routing and OSPF

Path Vector Routing and BGP

Figure 8.12 Autonomous systems

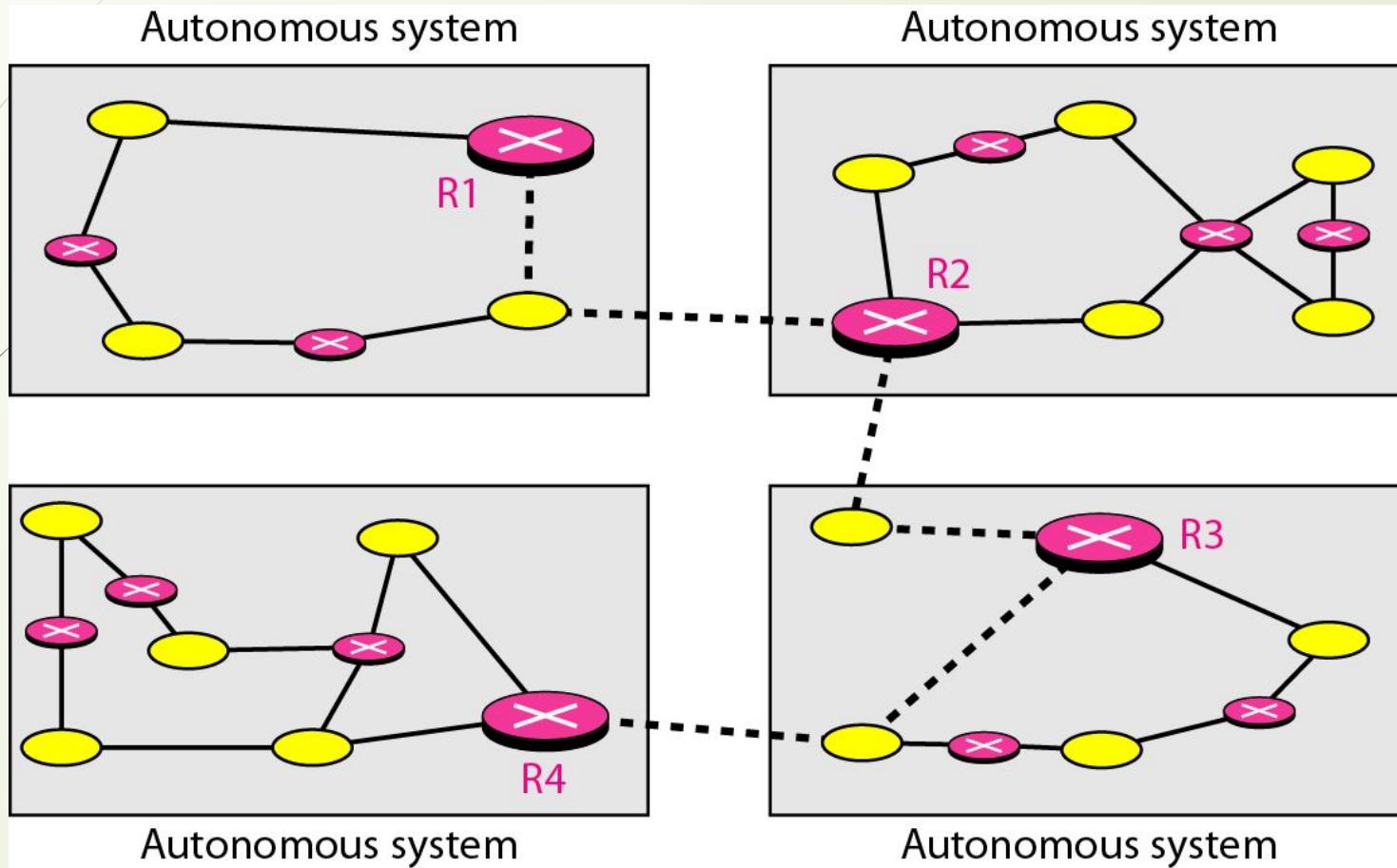


Figure 8.13 *Popular routing protocols*

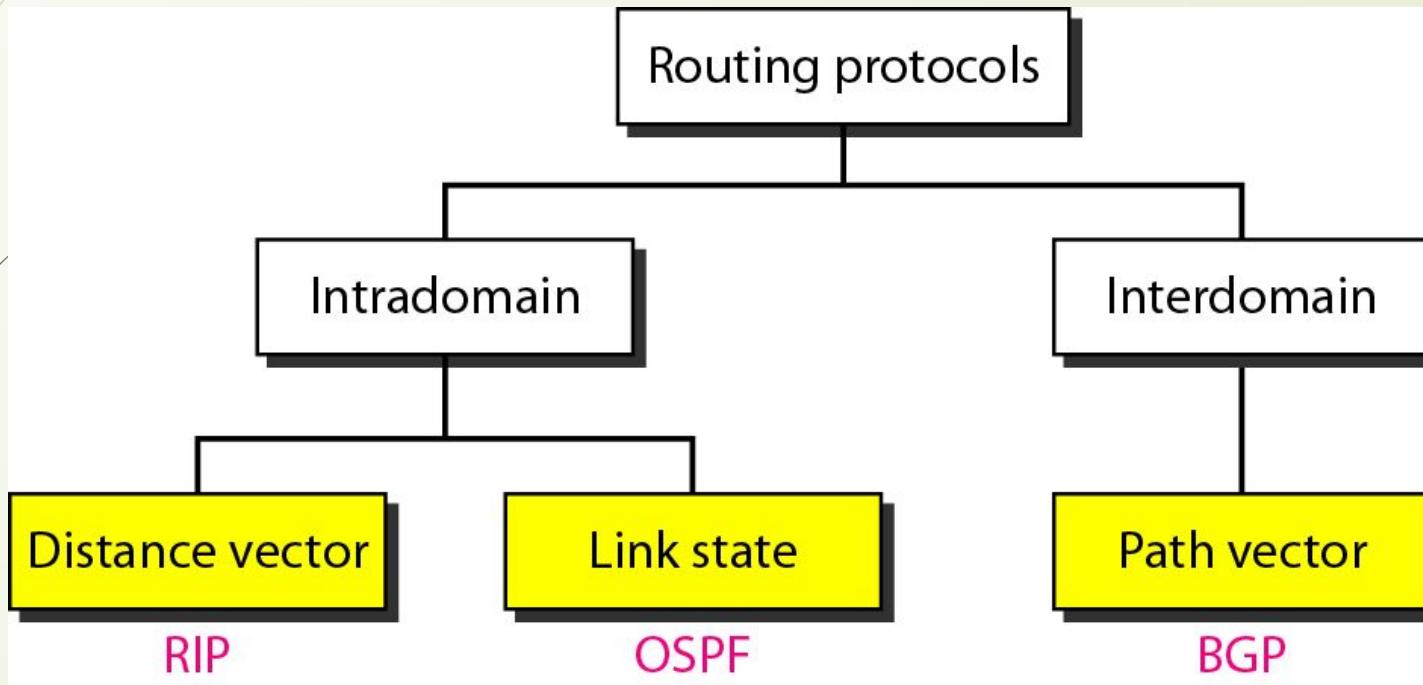


Figure 8.14 Distance vector routing tables

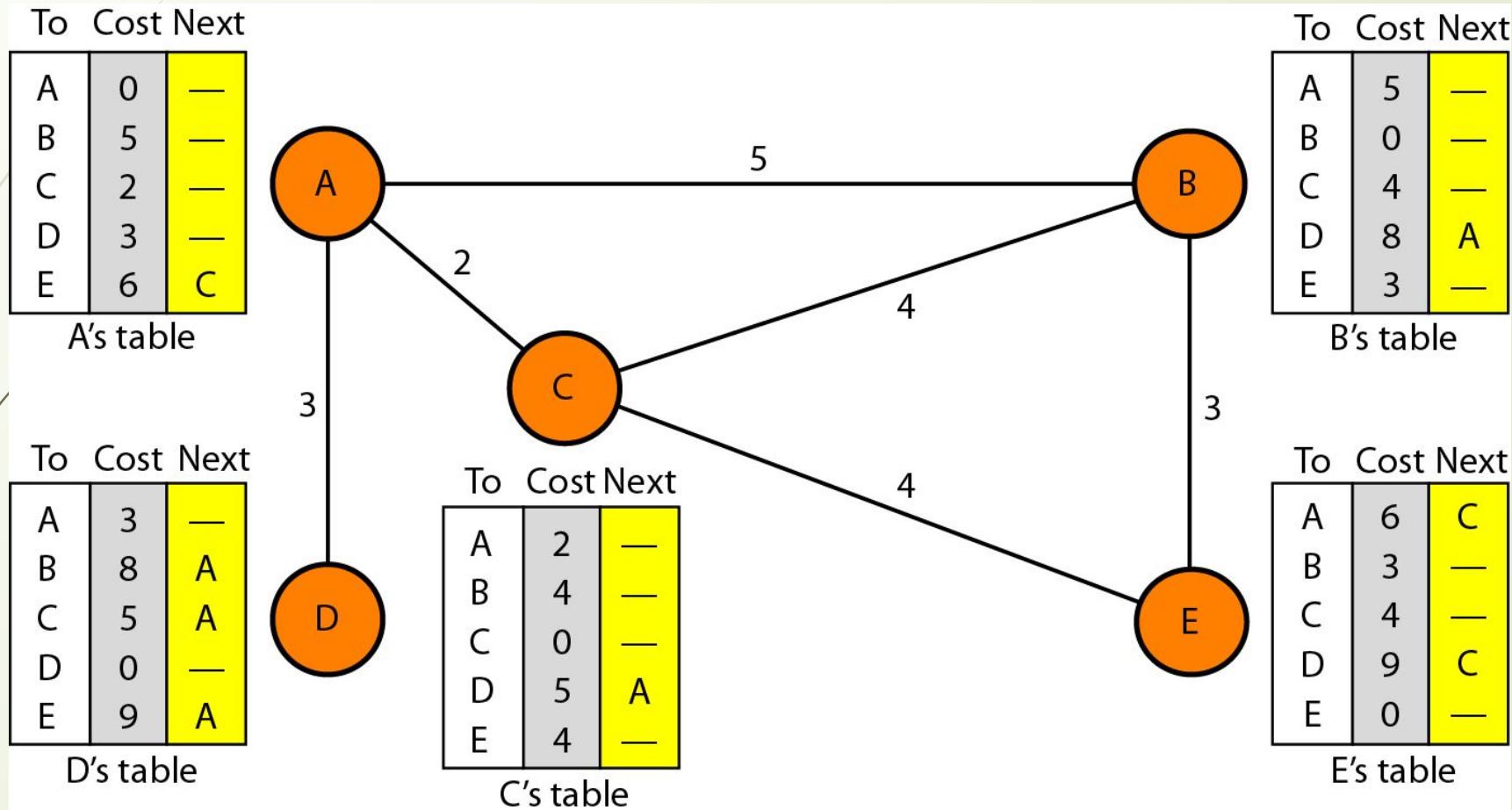
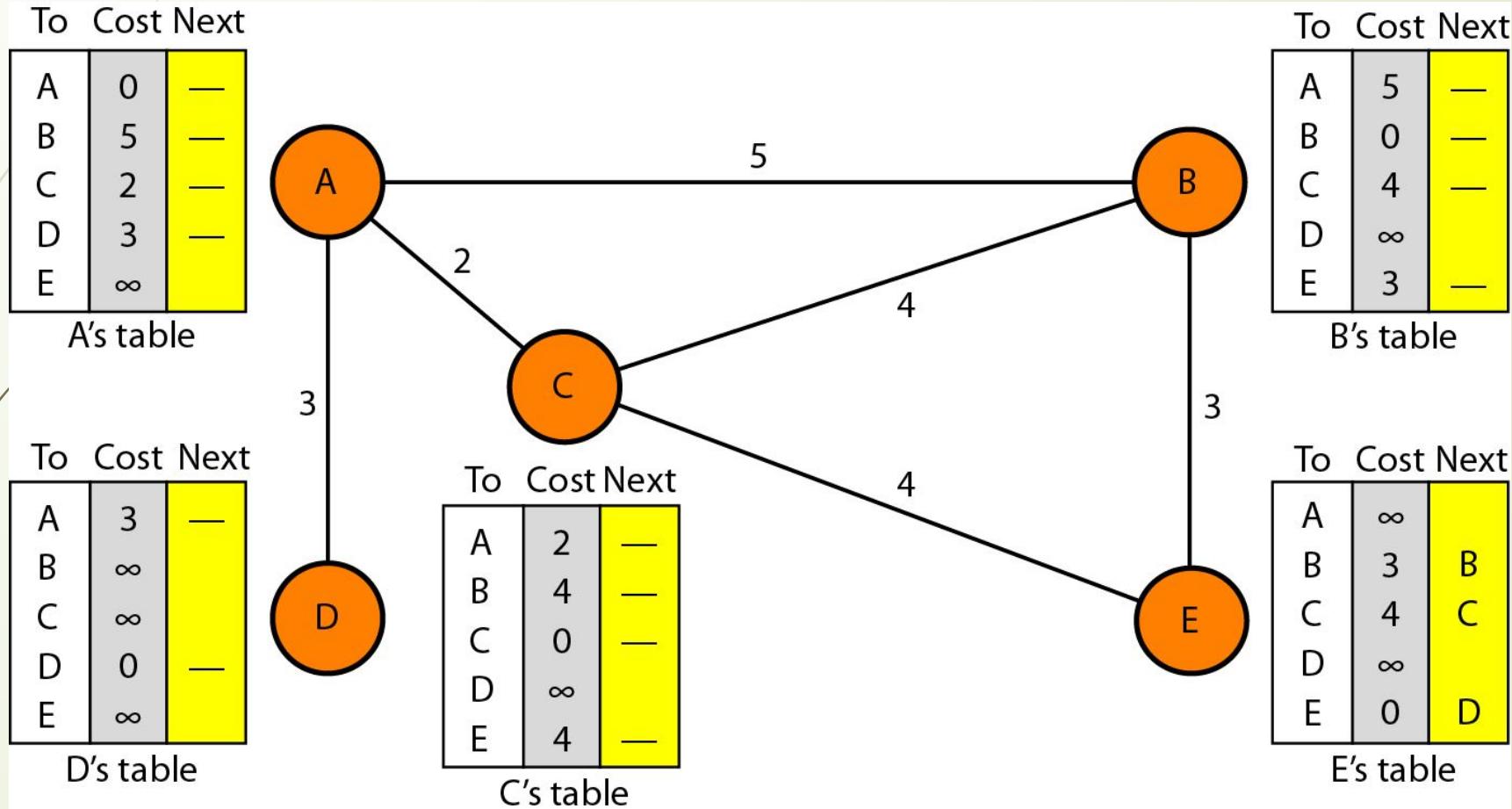


Figure 8.15 Initialization of tables in distance vector routing





Note

In distance vector routing, each node shares its routing table with its immediate neighbors periodically and when there is a change.

Figure 8.16 Updating in distance vector routing

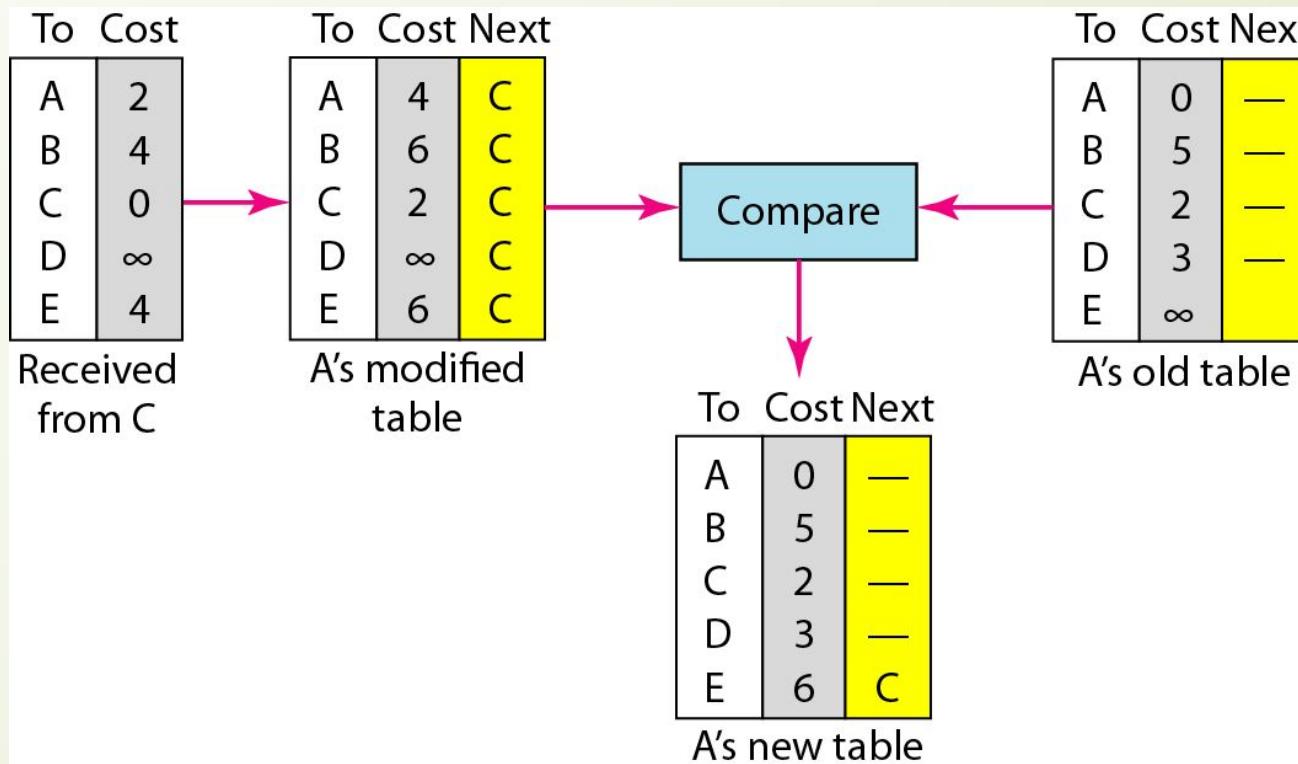


Figure 8.17 Two-node instability

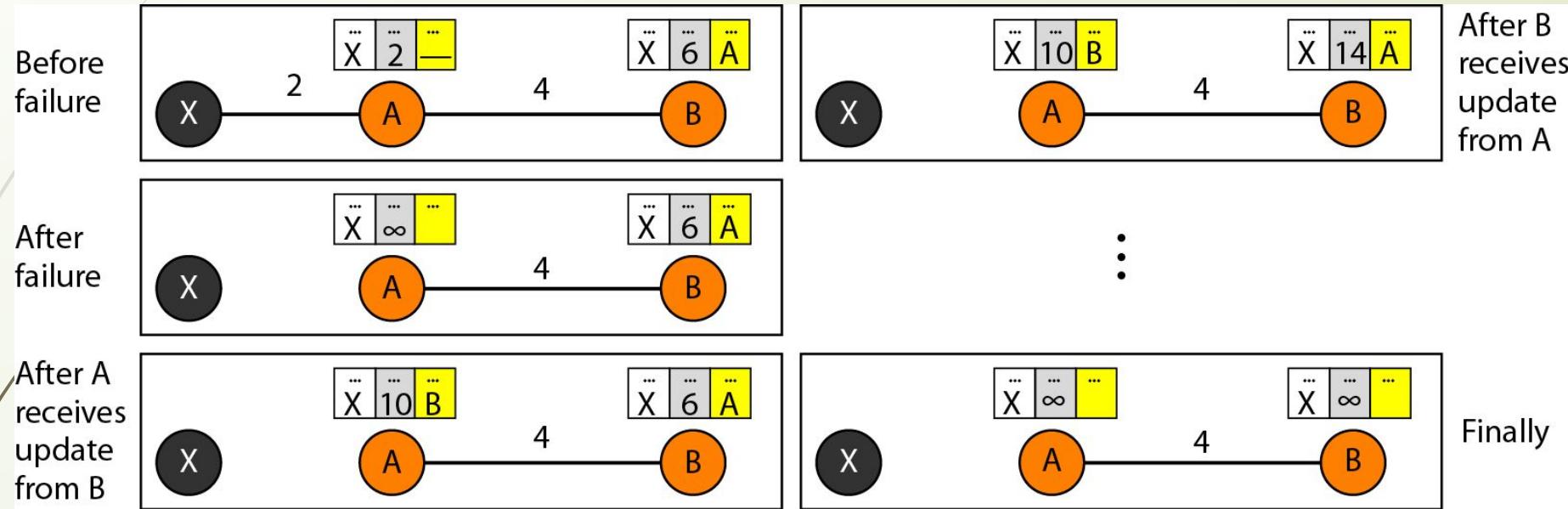


Figure 8.18 Three-node instability

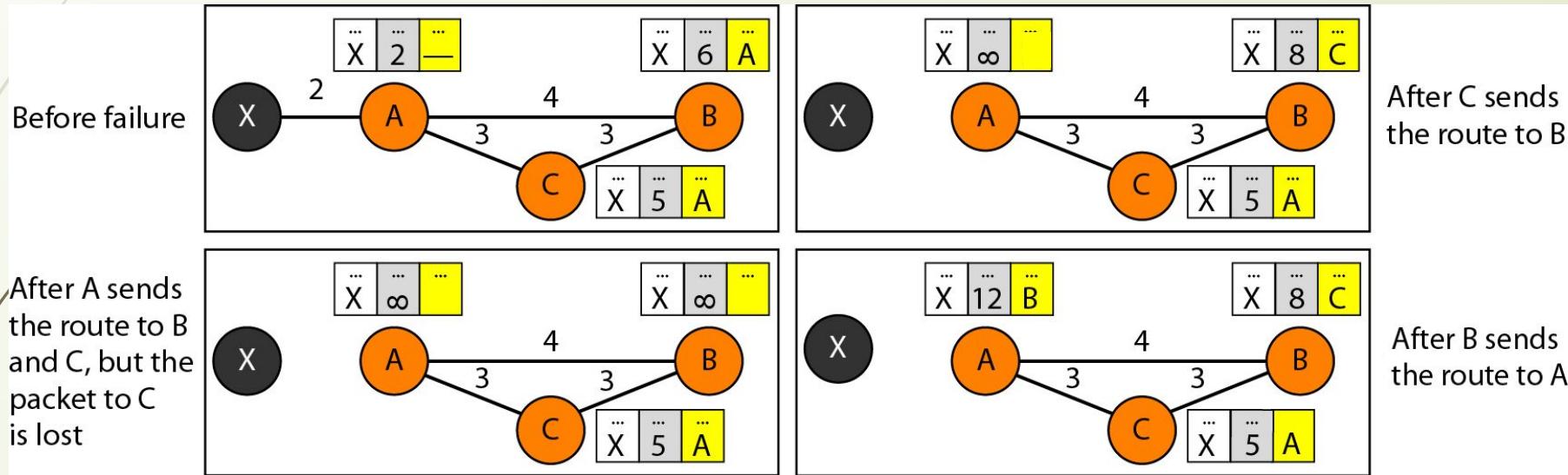


Figure 8.19 Example of a domain using RIP

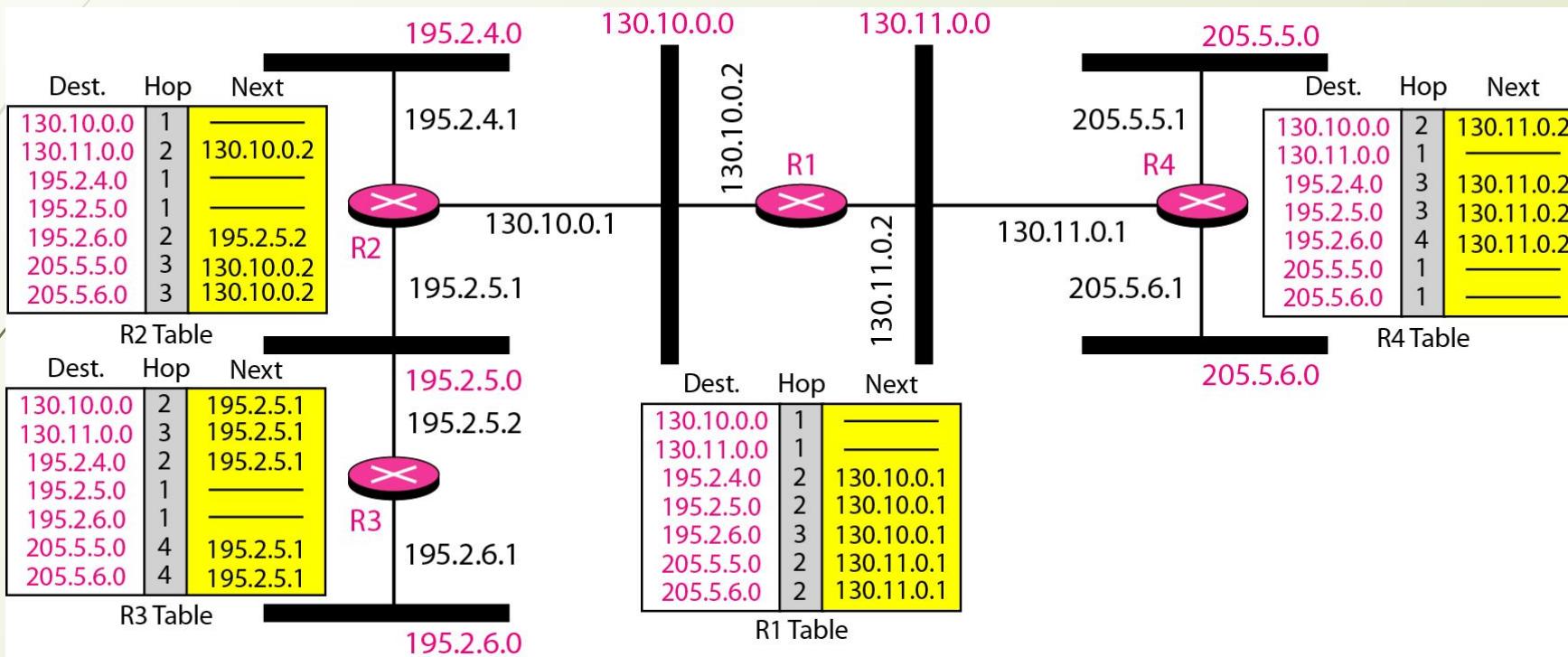


Figure 8.20 Concept of link state routing

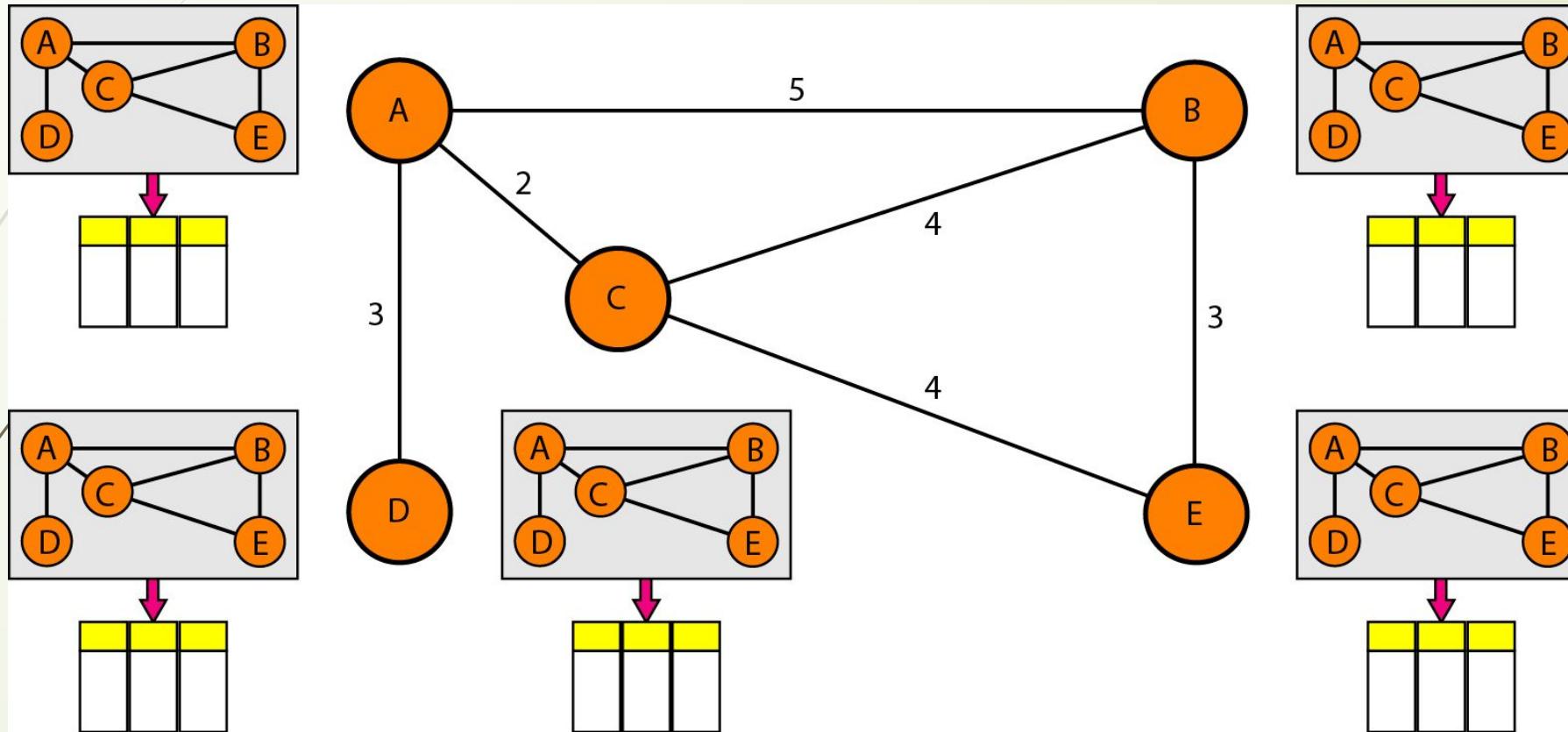


Figure 8.21 *Link state knowledge*

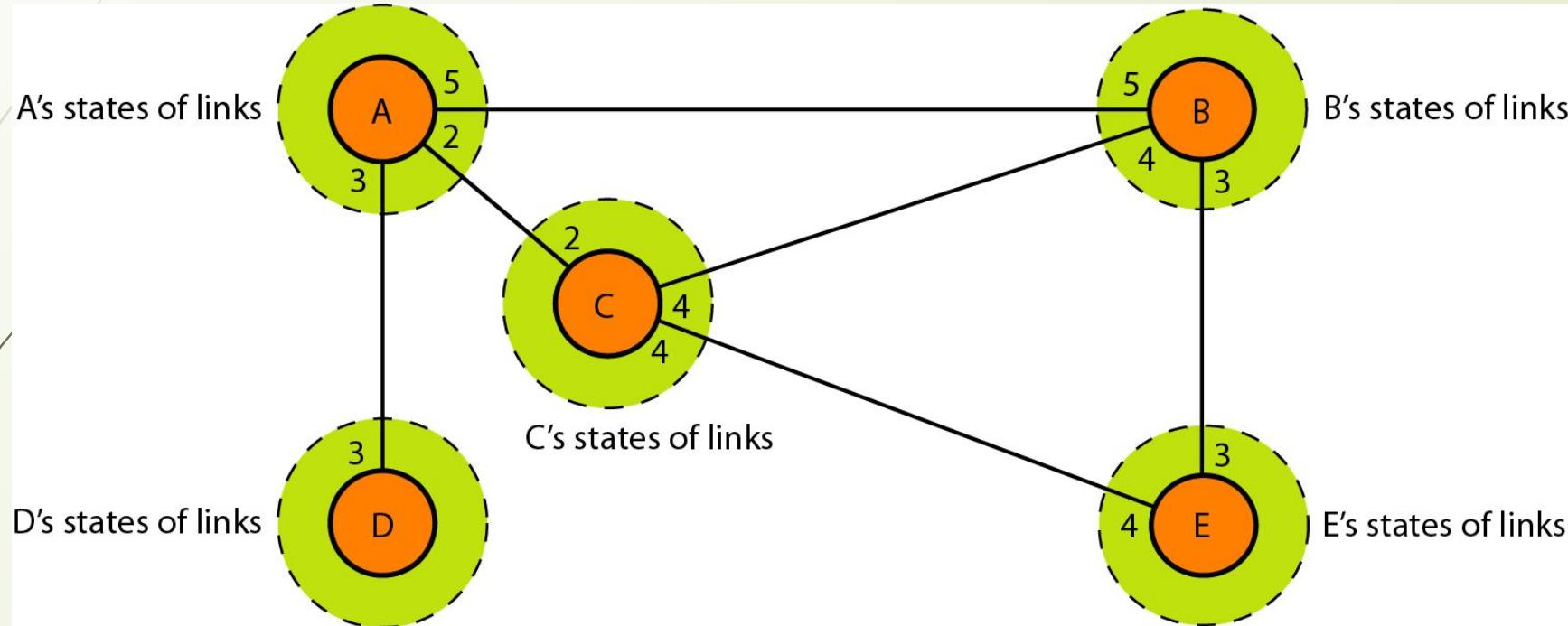


Figure 8.22 Dijkstra algorithm

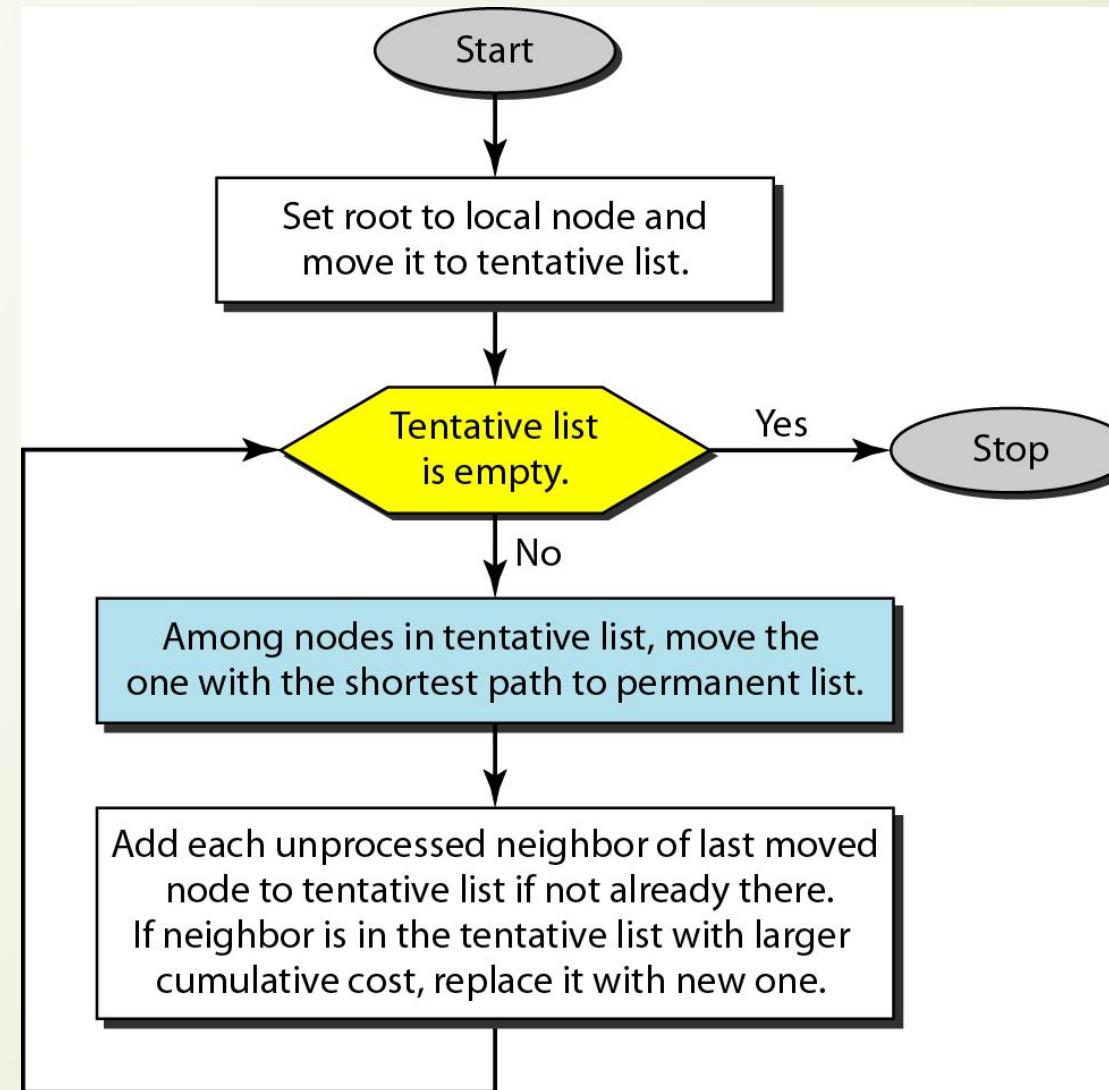


Figure 8.23 Example of formation of shortest path tree

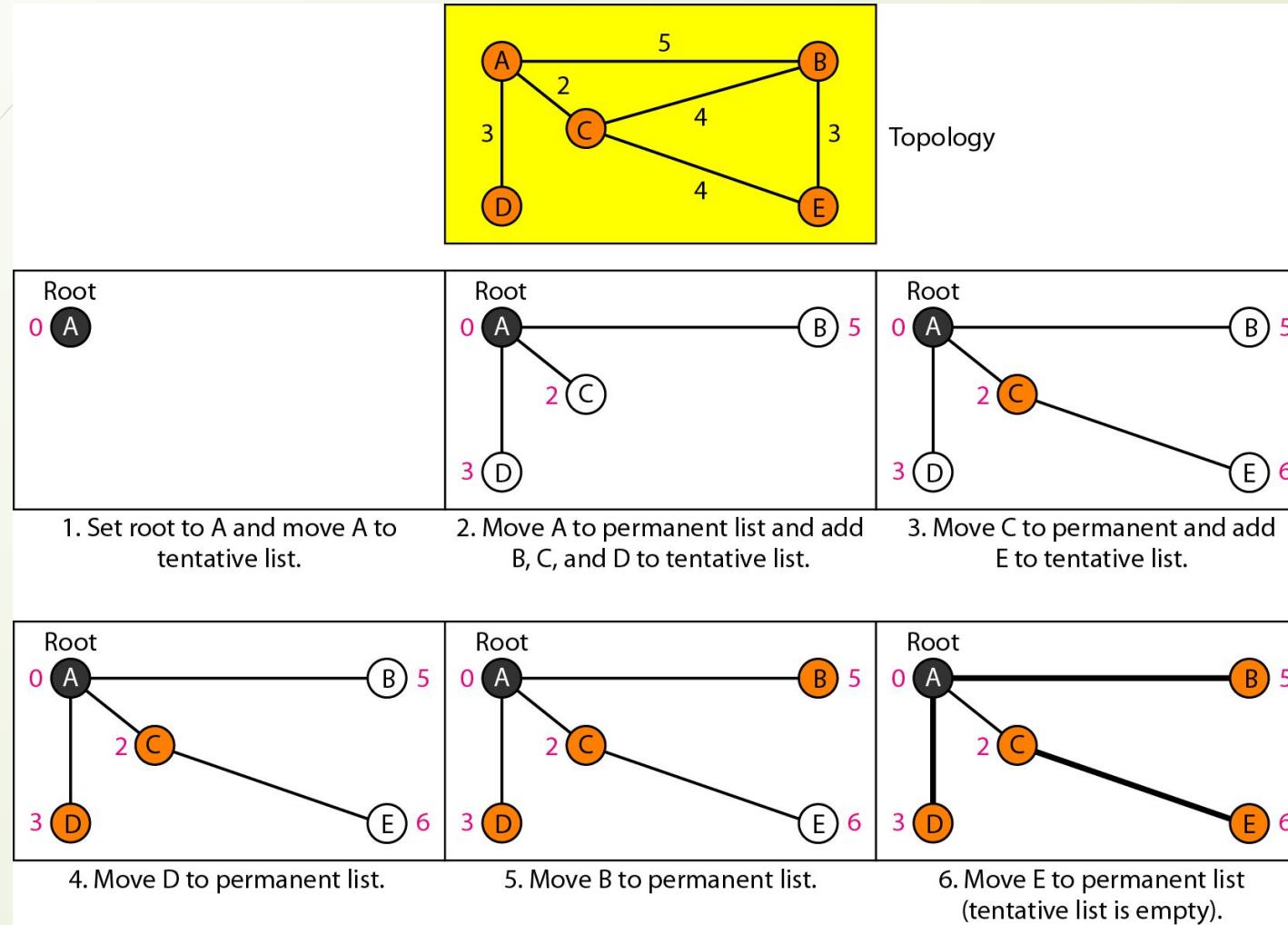


Table 8.2 *Routing table for node A*

<i>Node</i>	<i>Cost</i>	<i>Next Router</i>
A	0	—
B	5	—
C	2	—
D	3	—
E	6	C

Figure 8.24 Areas in an autonomous system

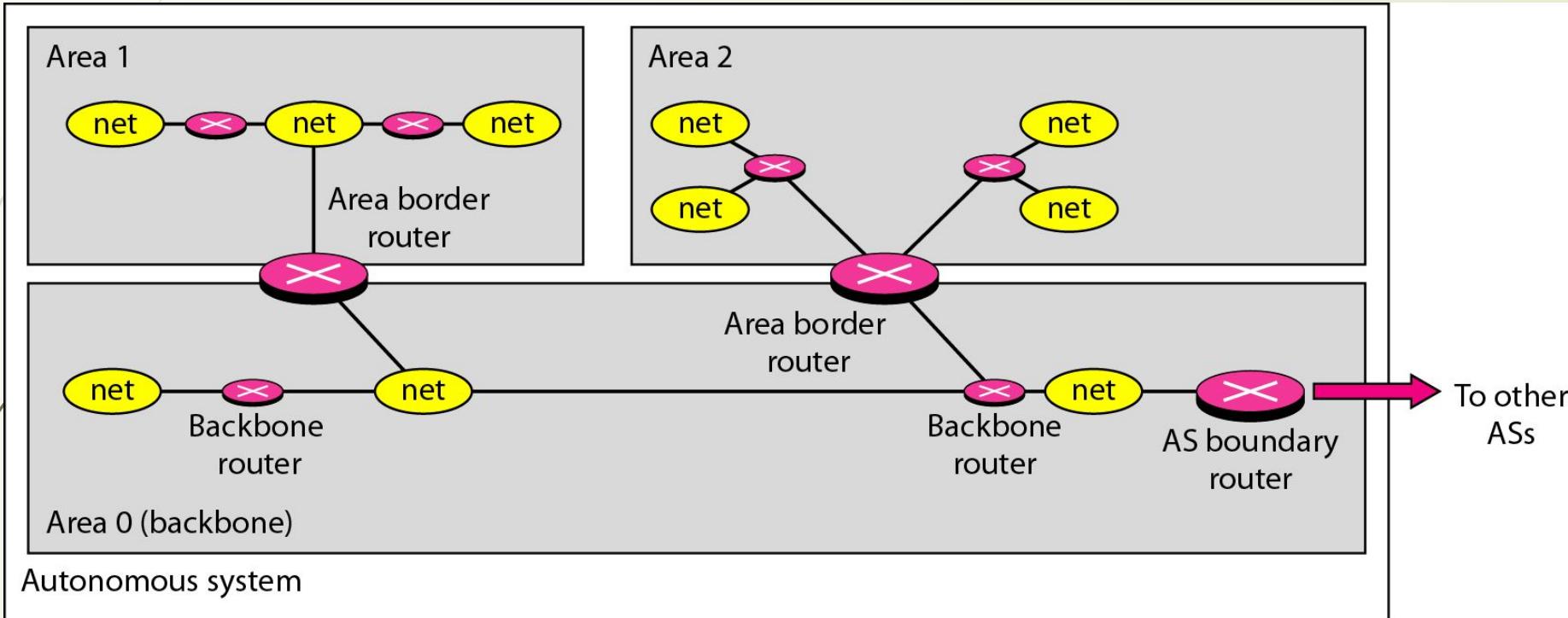


Figure 8.25 *Types of links*

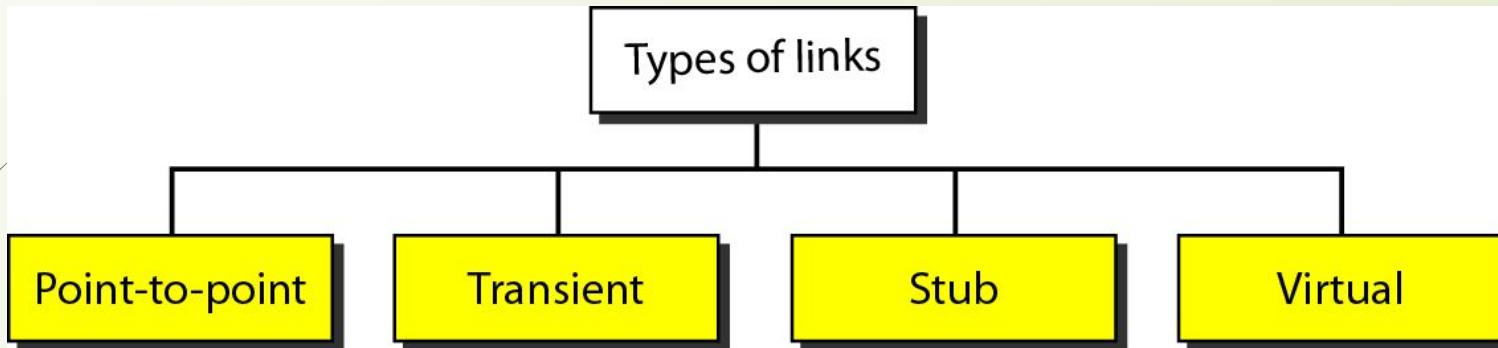


Figure 8.26 *Point-to-point link*



Figure 8.27 *Transient link*

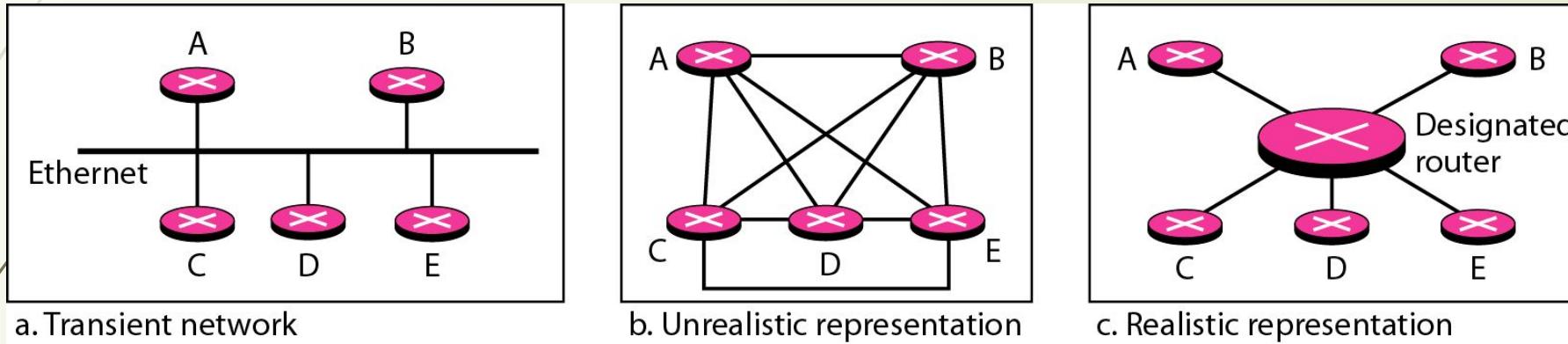


Figure 8.28 *Stub link*

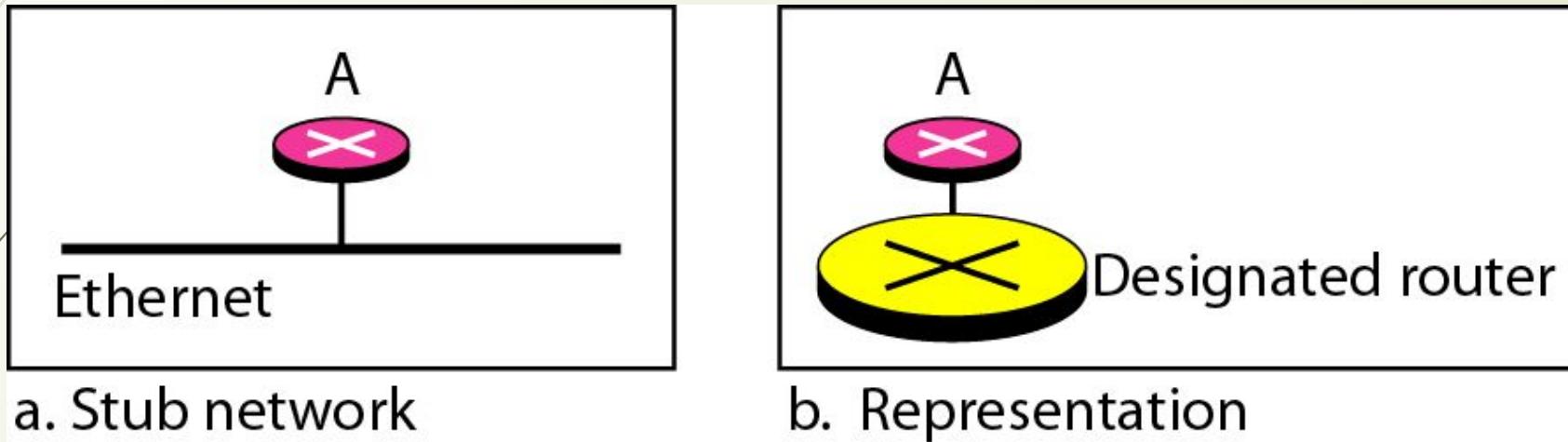


Figure 8.29 Example of an AS and its graphical representation in OSPF

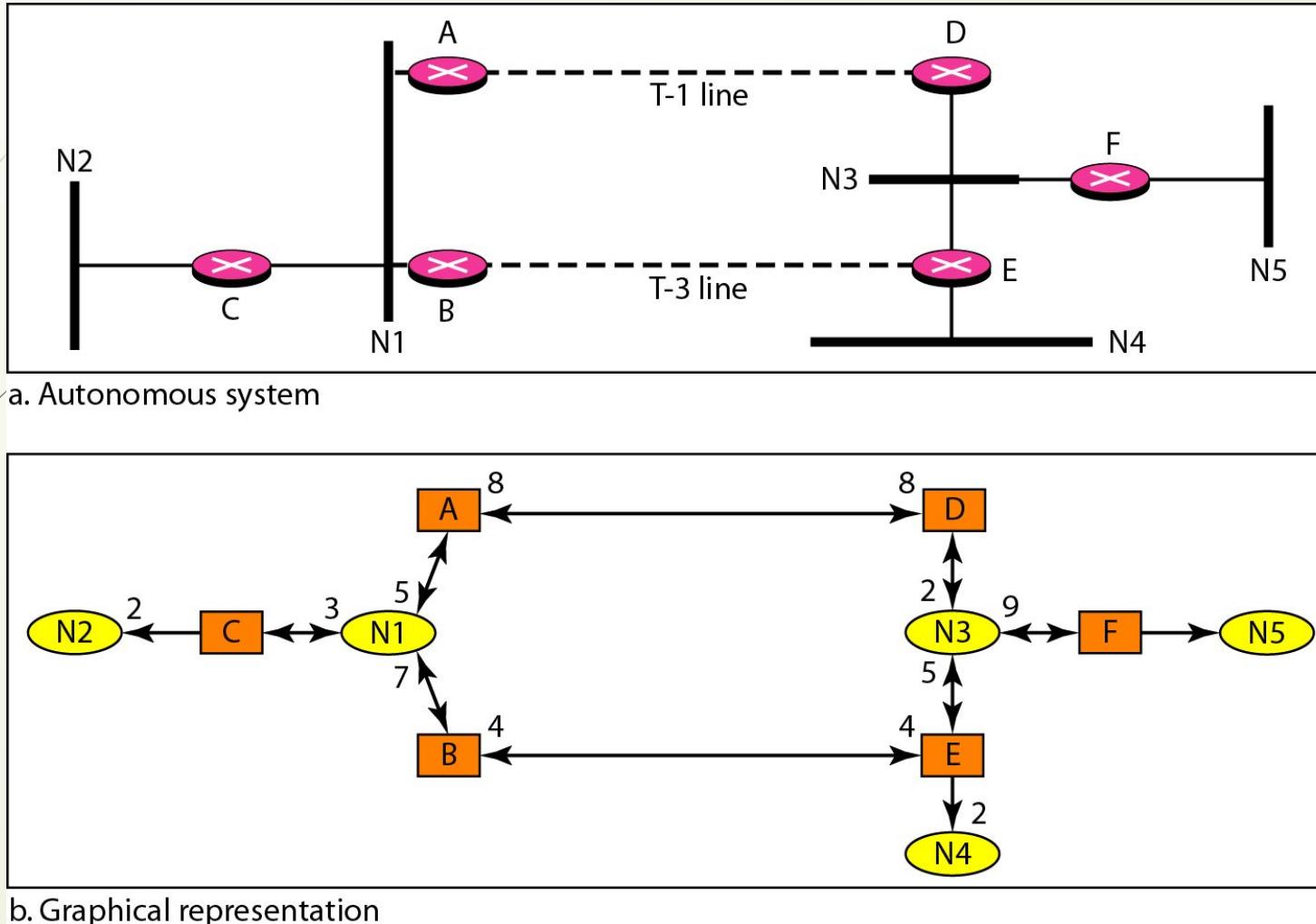


Figure 8.30 Initial routing tables in path vector routing

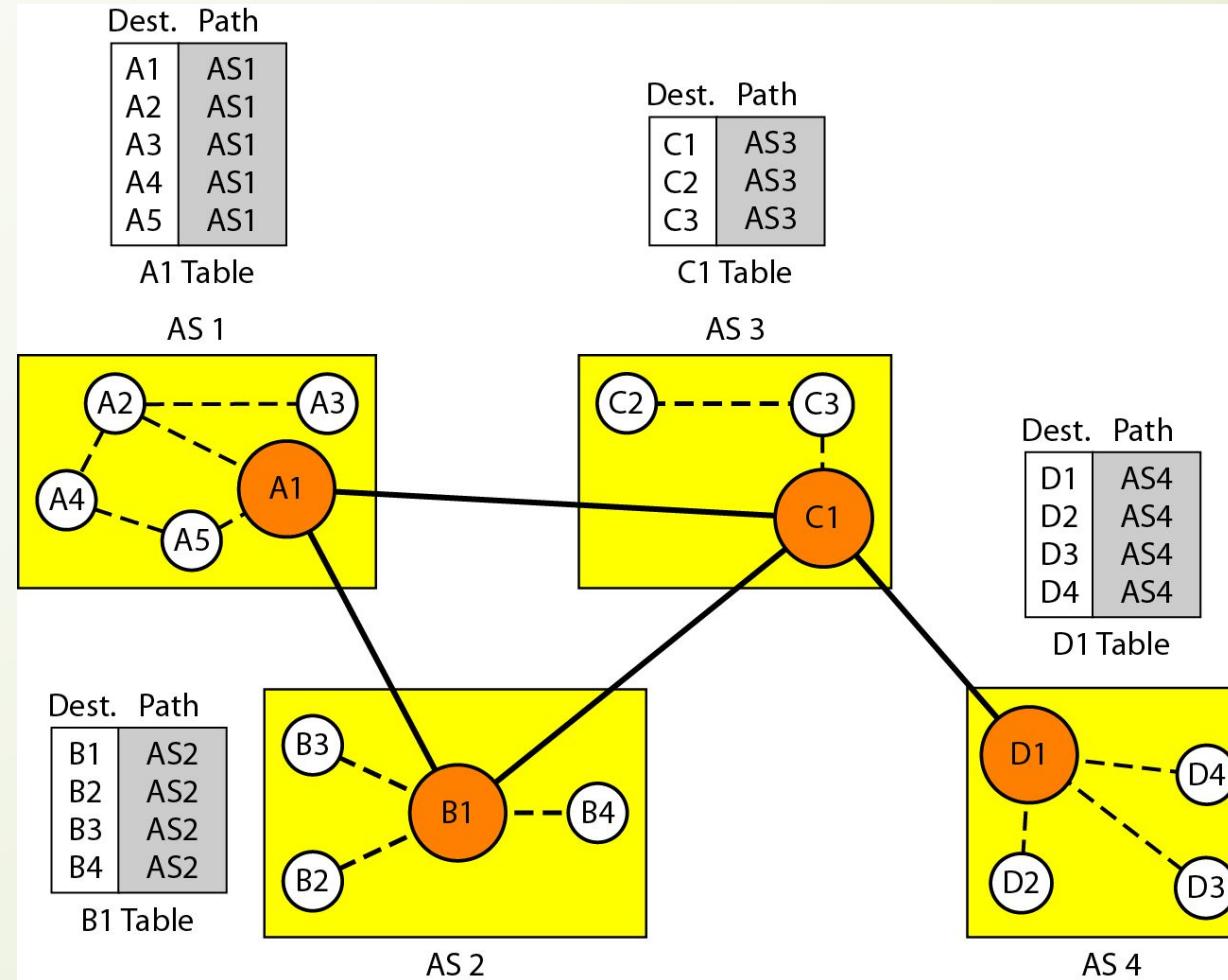


Figure 8.31 *Stabilized tables for three autonomous systems*

Dest.	Path
A1	AS1
...	
A5	AS1
B1	AS1-AS2
...	
B4	AS1-AS2
C1	AS1-AS3
...	
C3	AS1-AS3
D1	AS1-AS2-AS4
...	
D4	AS1-AS2-AS4

A1 Table

Dest.	Path
A1	AS2-AS1
...	
A5	AS2-AS1
B1	AS2
...	
B4	AS2
C1	AS2-AS3
...	
C3	AS2-AS3
D1	AS2-AS3-AS4
...	
D4	AS2-AS3-AS4

B1 Table

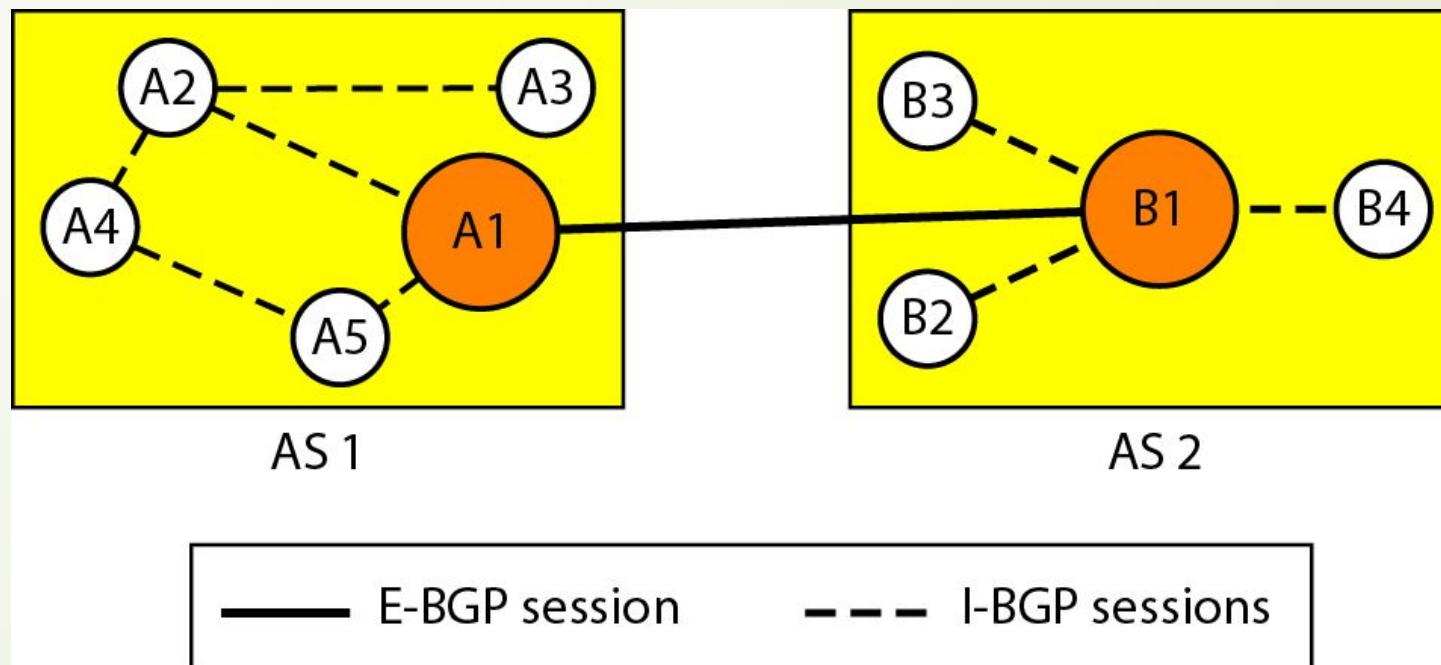
Dest.	Path
A1	AS3-AS1
...	
A5	AS3-AS1
B1	AS3-AS2
...	
B4	AS3-AS2
C1	AS3
...	
C3	AS3
D1	AS3-AS4
...	
D4	AS3-AS4

C1 Table

Dest.	Path
A1	AS4-AS3-AS1
...	
A5	AS4-AS3-AS1
B1	AS4-AS3-AS2
...	
B4	AS4-AS3-AS2
C1	AS4-AS3
...	
C3	AS4-AS3
D1	AS4
...	
D4	AS4

D1 Table

Figure 8.32 Internal and external BGP sessions



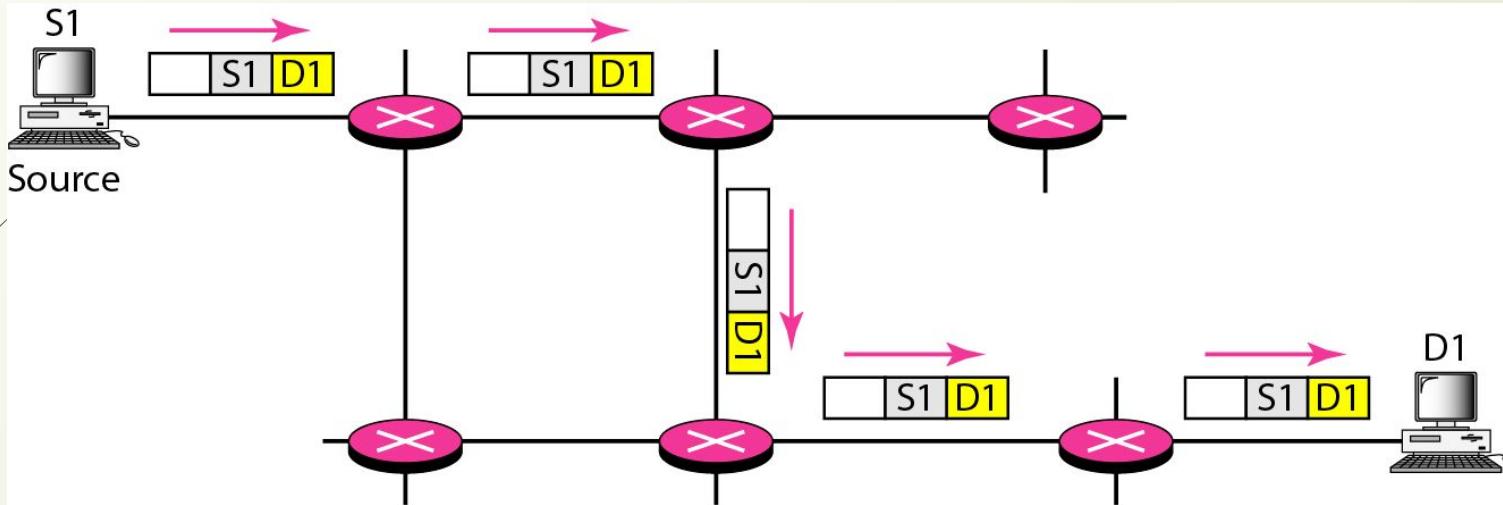
8-4 MULTICAST ROUTING PROTOCOLS

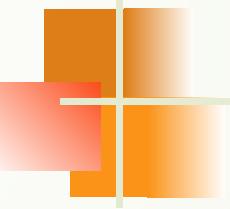
In this section, we discuss multicasting and multicast routing protocols.

Topics discussed in this section:

**Unicast, Multicast, and Broadcast Applications
Multicast Routing
Routing Protocols**

Figure 8.33 Unicasting

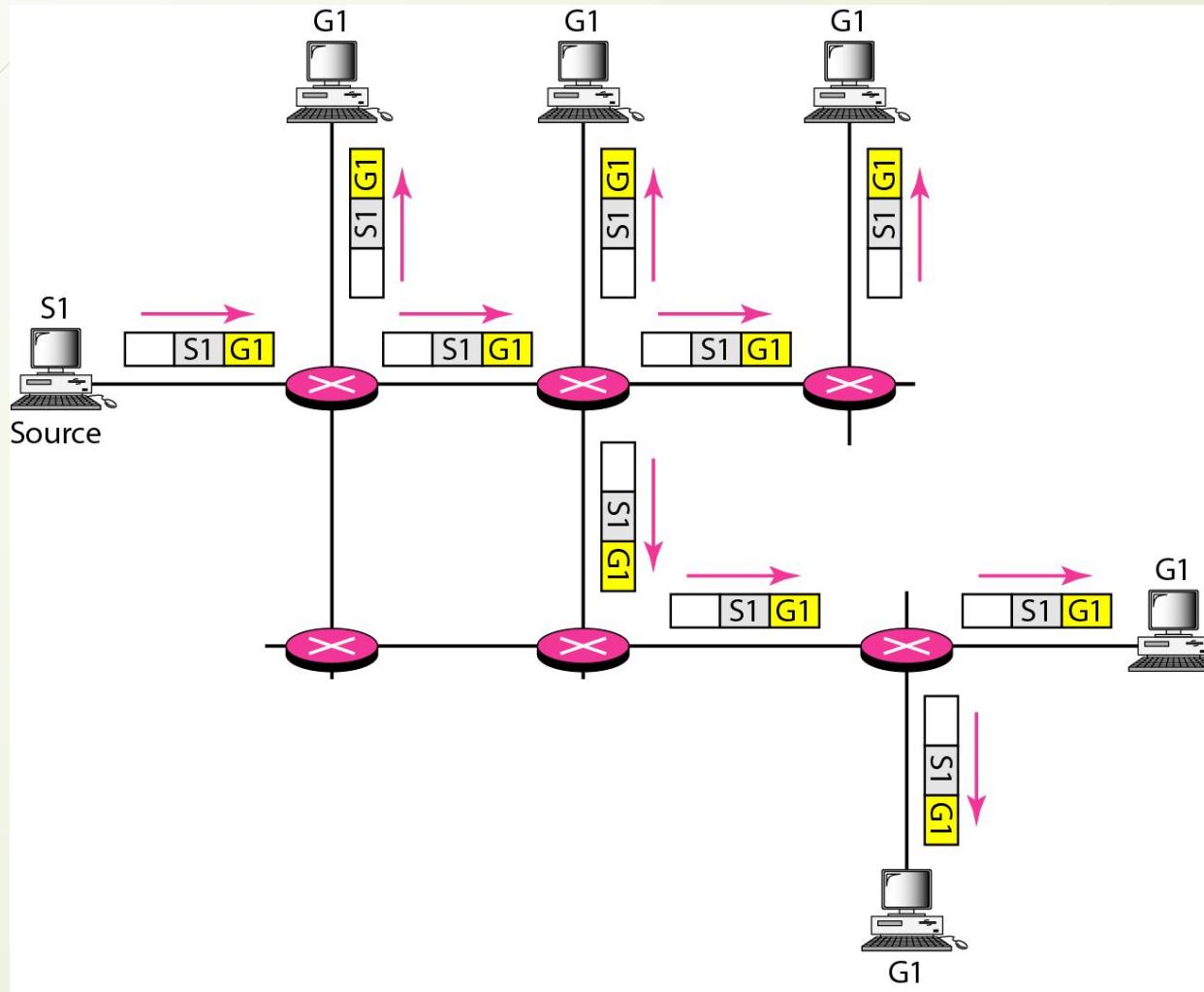




Note

In unicasting, the router forwards the received packet through only one of its interfaces.

Figure 8.34 Multicasting

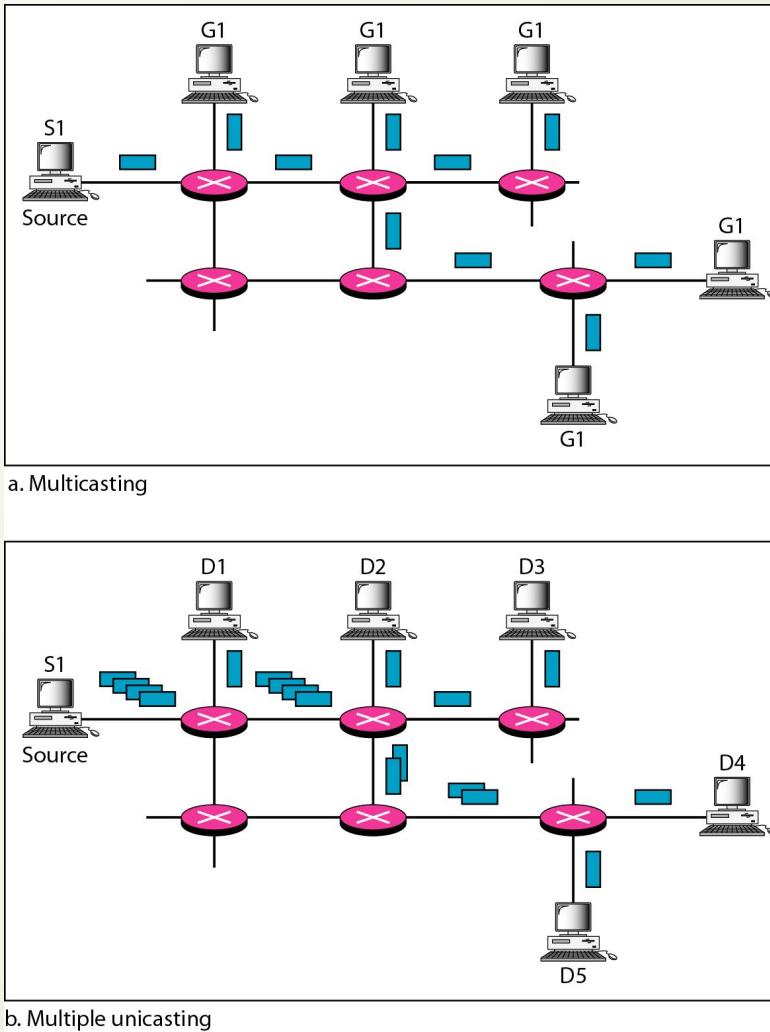




Note

In multicasting, the router may forward the received packet through several of its interfaces.

Figure 8.35 Multicasting versus multiple unicasting





Note

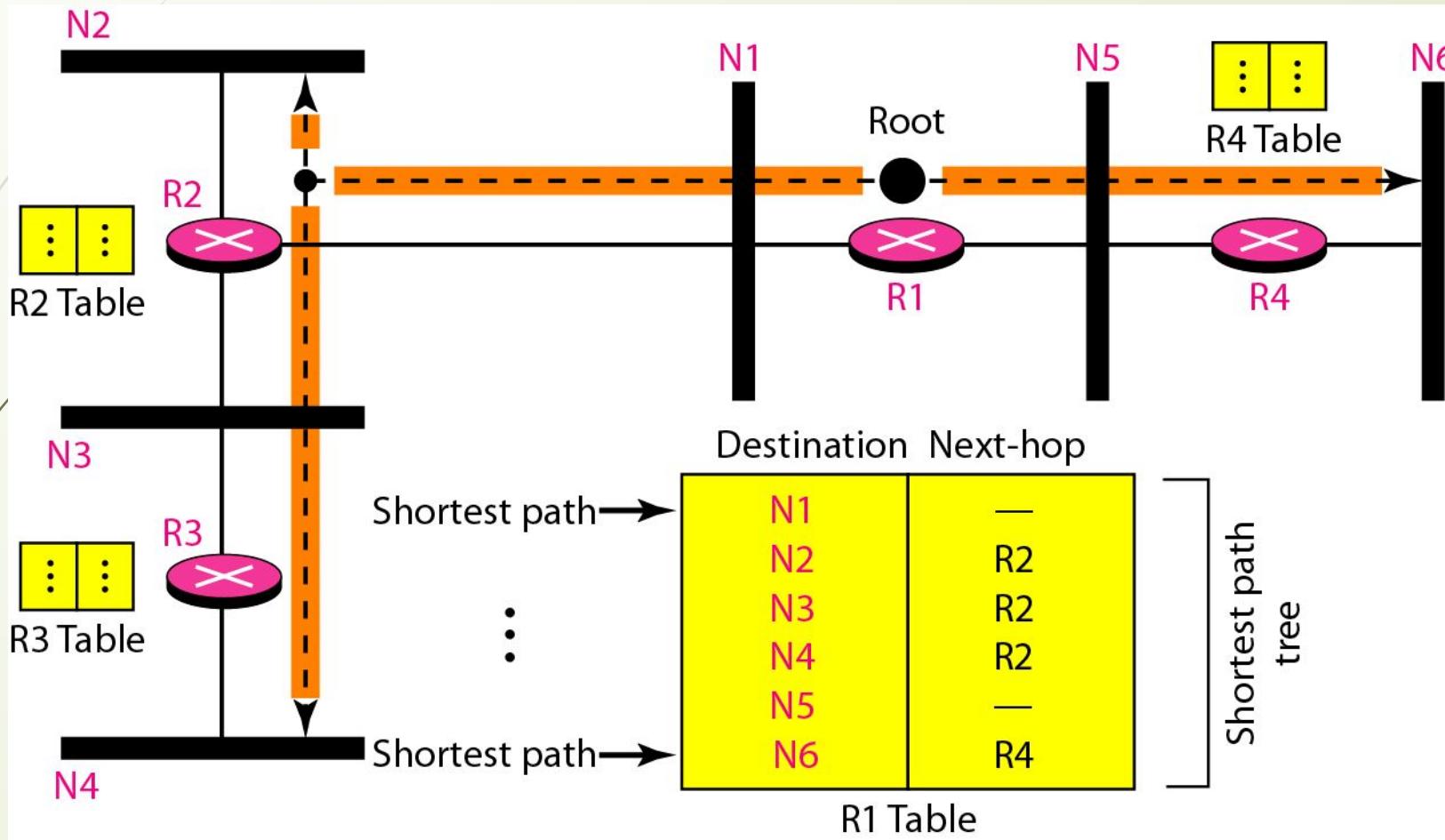
Emulation of multicasting through multiple unicasting is not efficient and may create long delays, particularly with a large group.

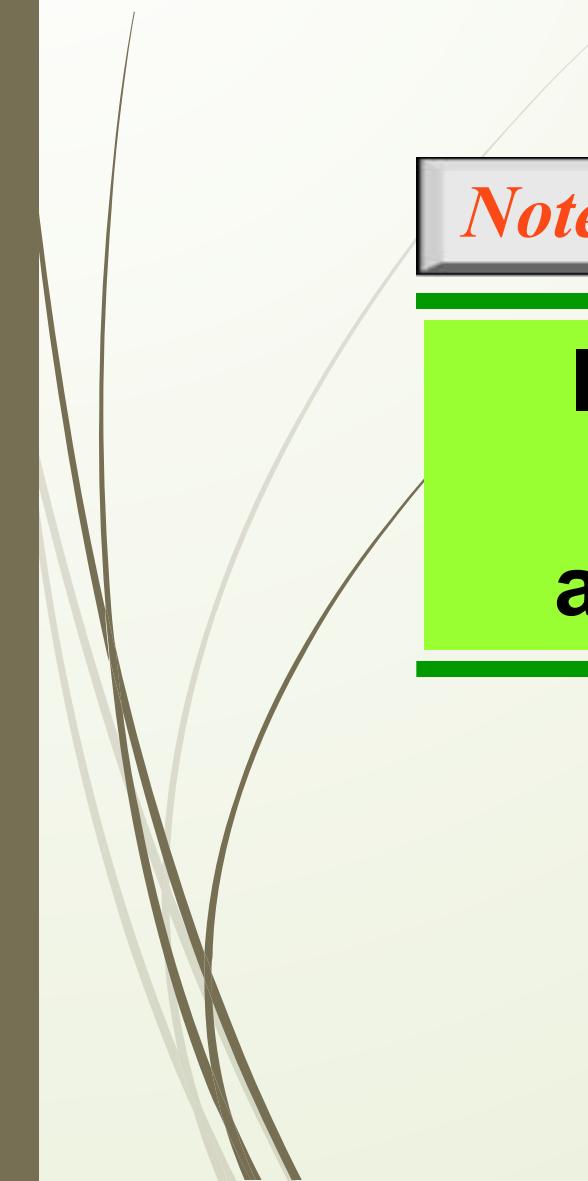


Note

In unicast routing, each router in the domain has a table that defines a shortest path tree to possible destinations.

Figure 8.36 Shortest path tree in unicast routing

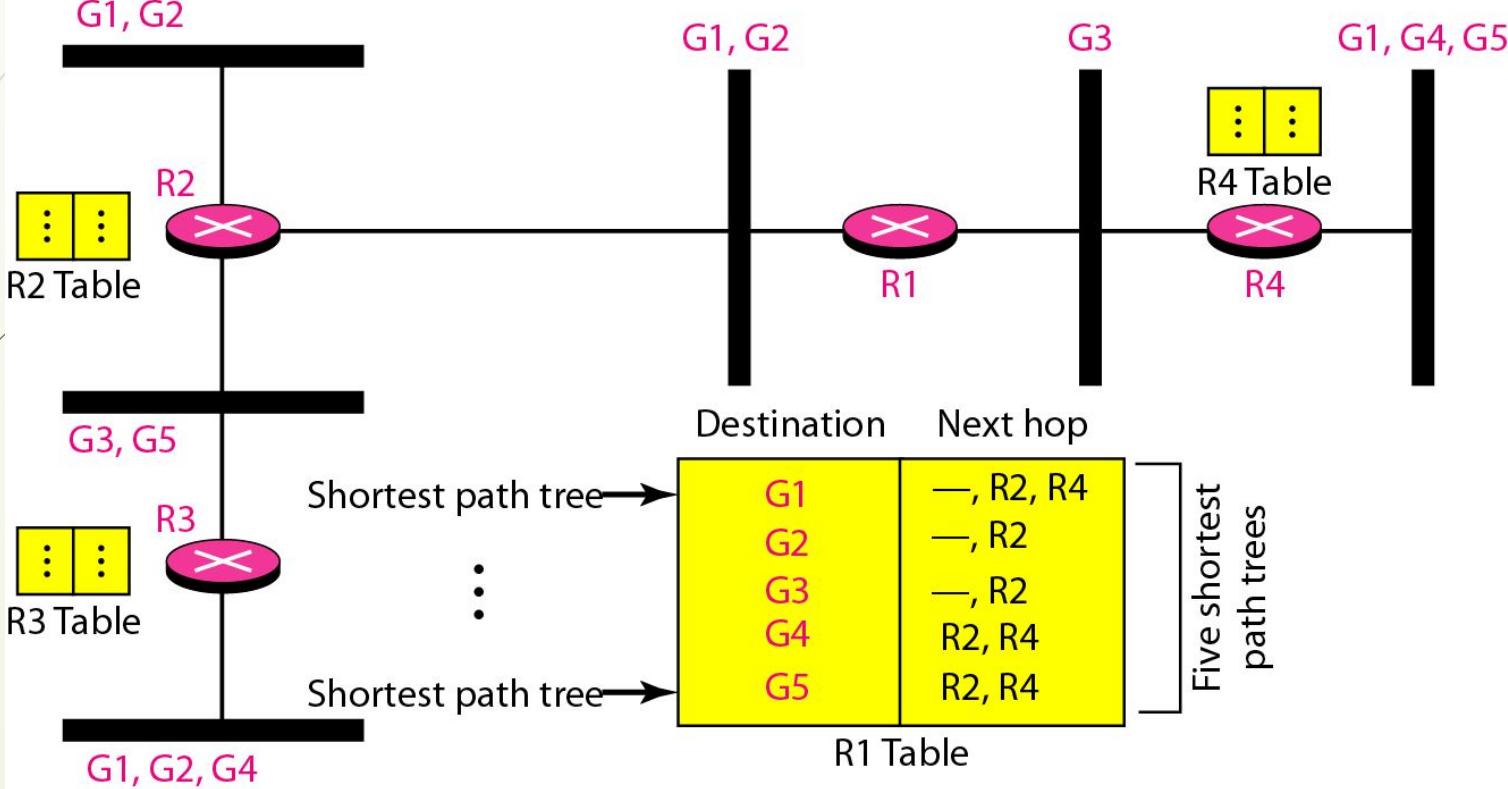


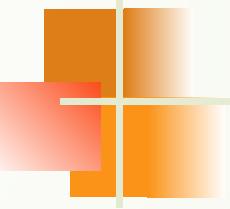


Note

In multicast routing, each involved router needs to construct a shortest path tree for each group.

Figure 8.37 Source-based tree approach

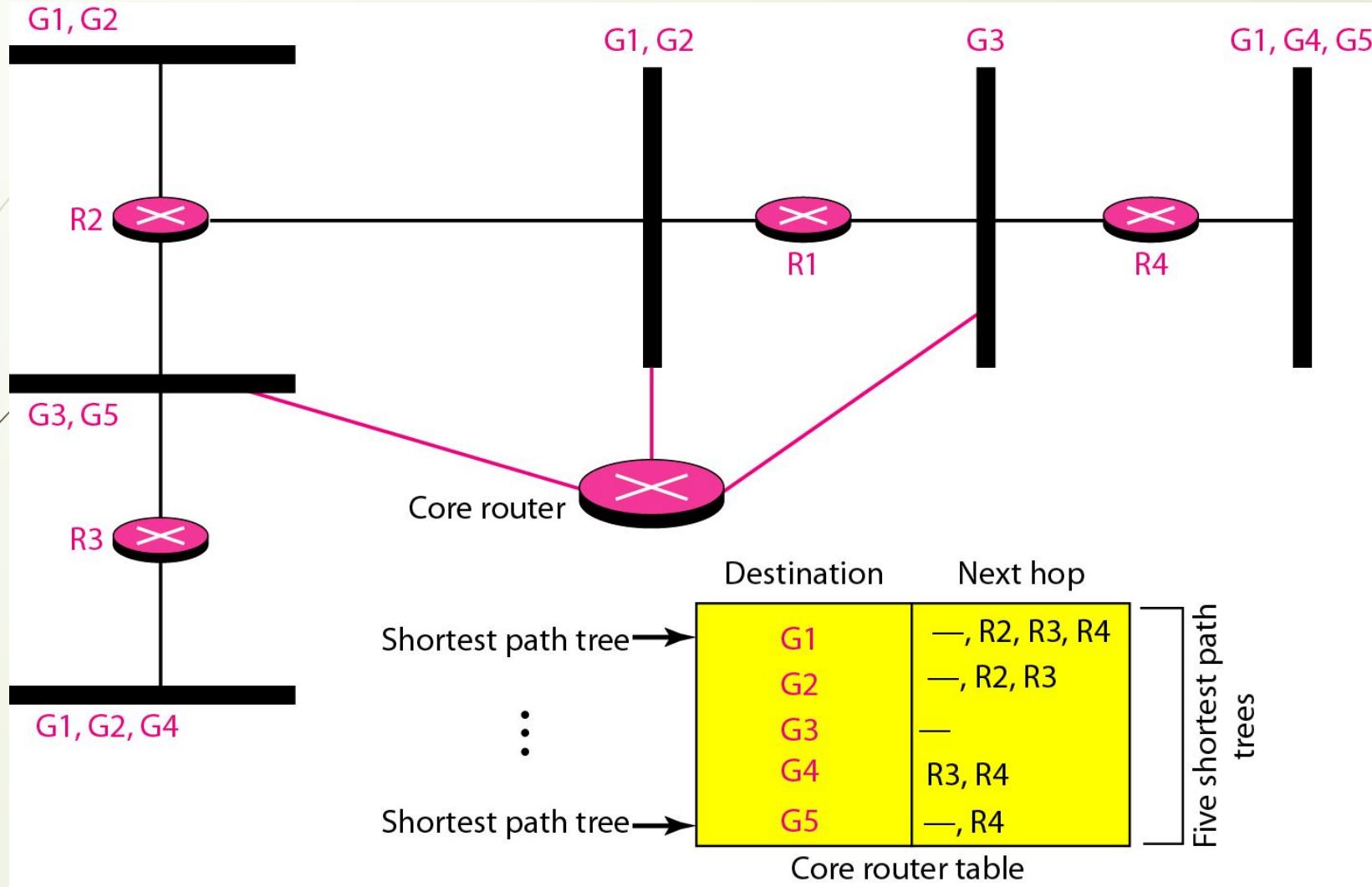


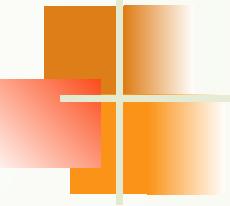


Note

In the source-based tree approach, each router needs to have one shortest path tree for each group.

Figure 8.38 Group-shared tree approach

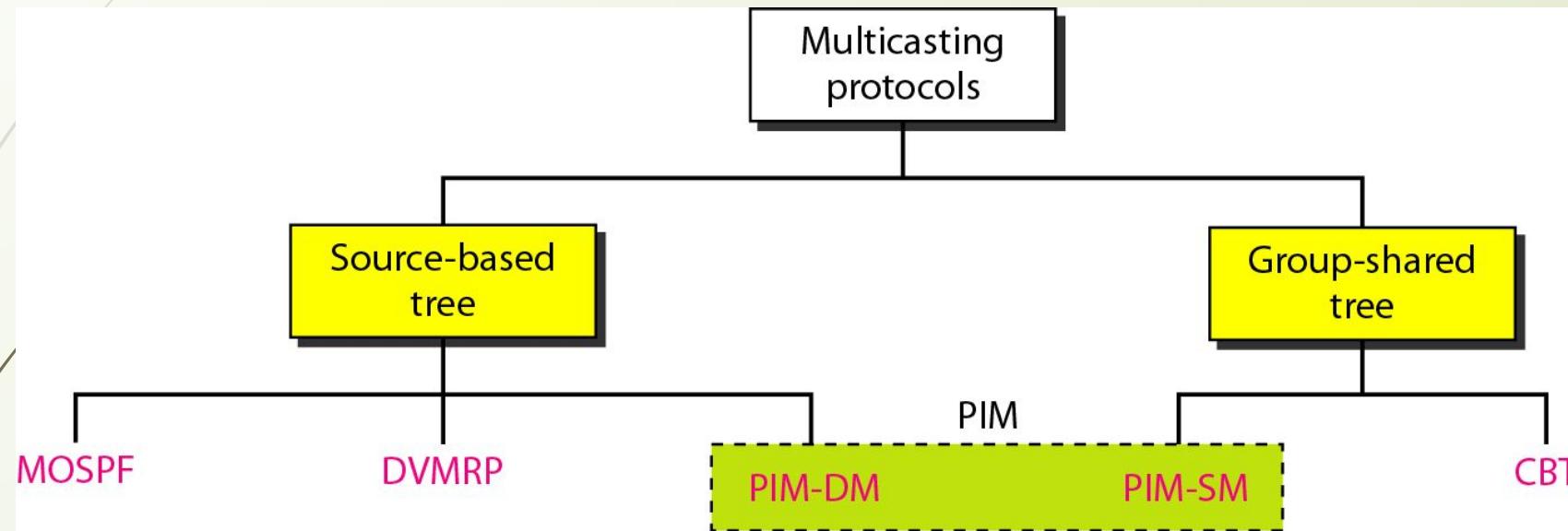


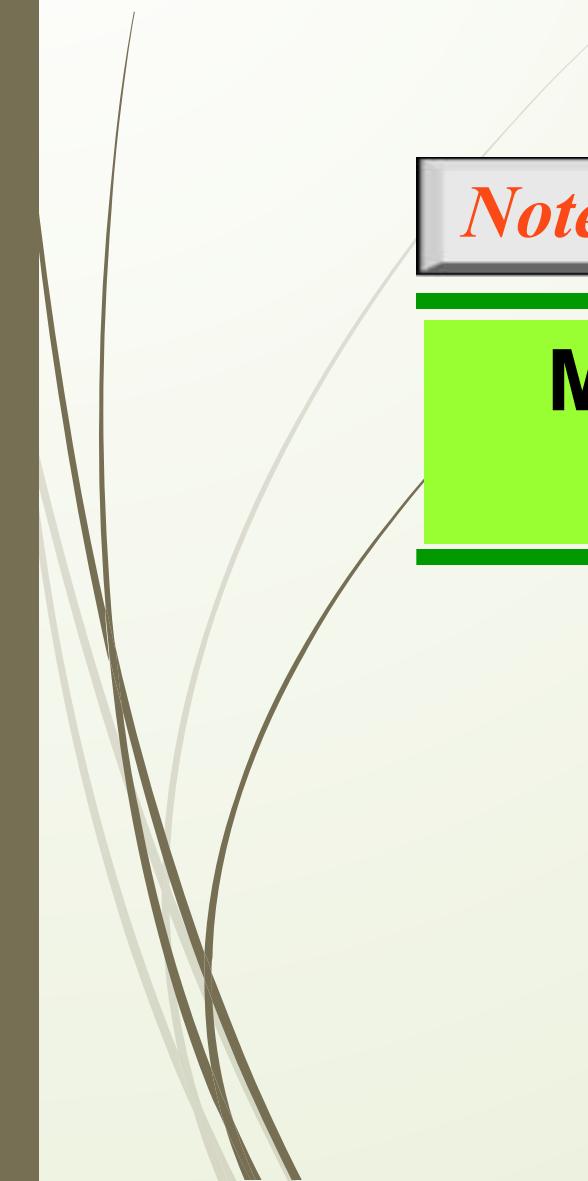


Note

In the group-shared tree approach, only the core router, which has a shortest path tree for each group, is involved in multicasting.

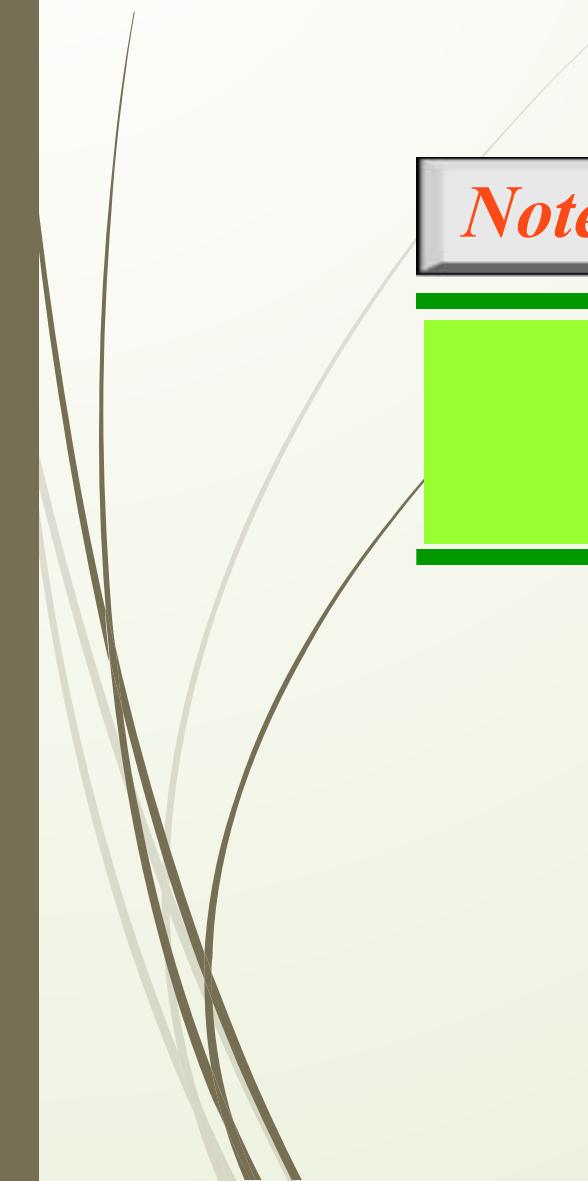
Figure 8.39 *Taxonomy of common multicast protocols*





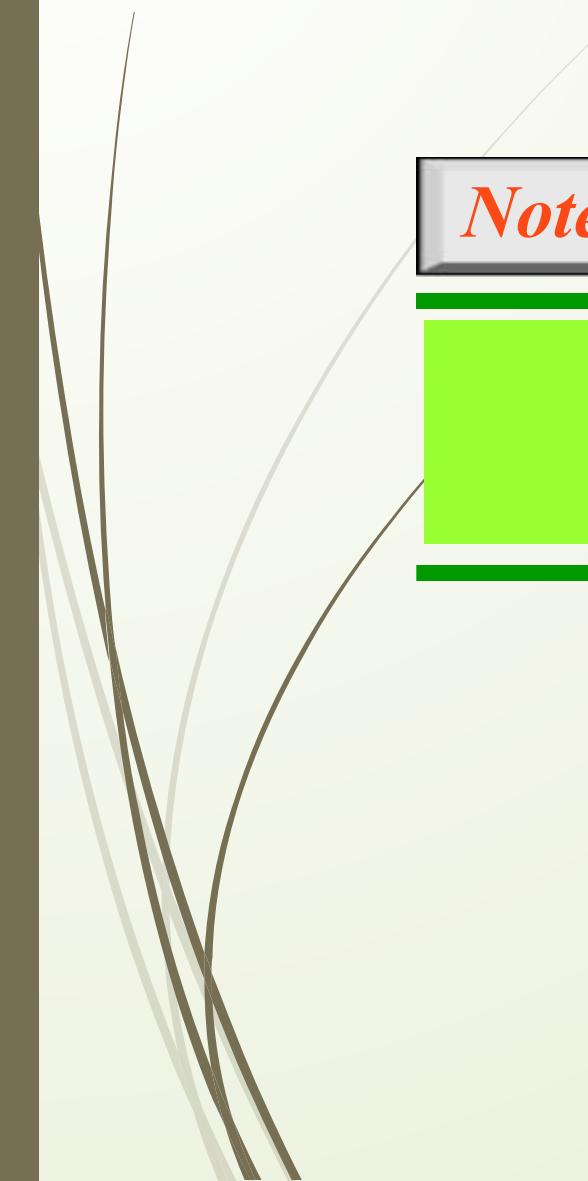
Note

Multicast link state routing uses the source-based tree approach.



Note

Flooding broadcasts packets, but creates loops in the systems.



Note

RPF eliminates the loop in the flooding process.

Figure 8.40 Reverse path forwarding (RPF)

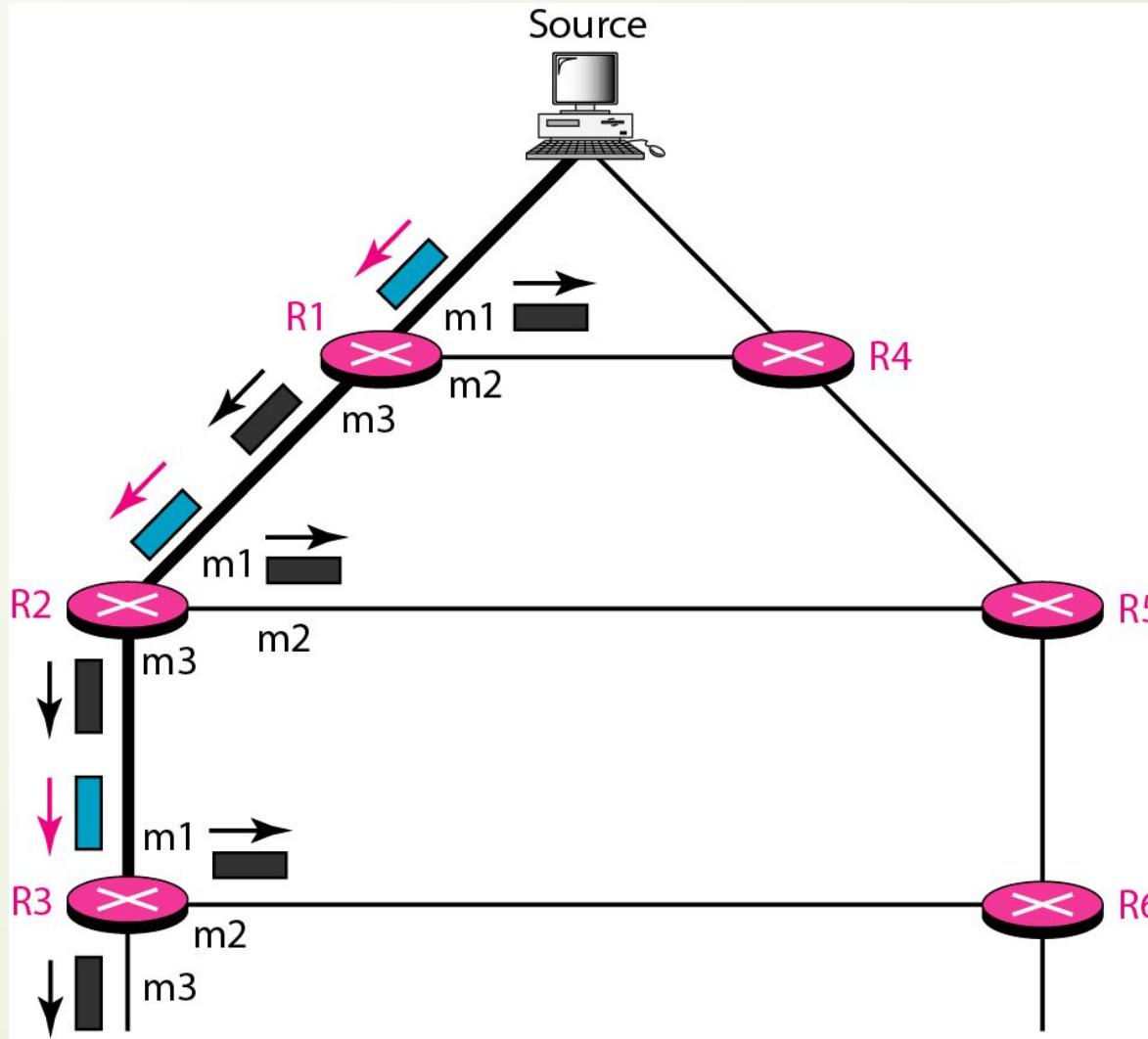


Figure 8.41 Problem with RPF

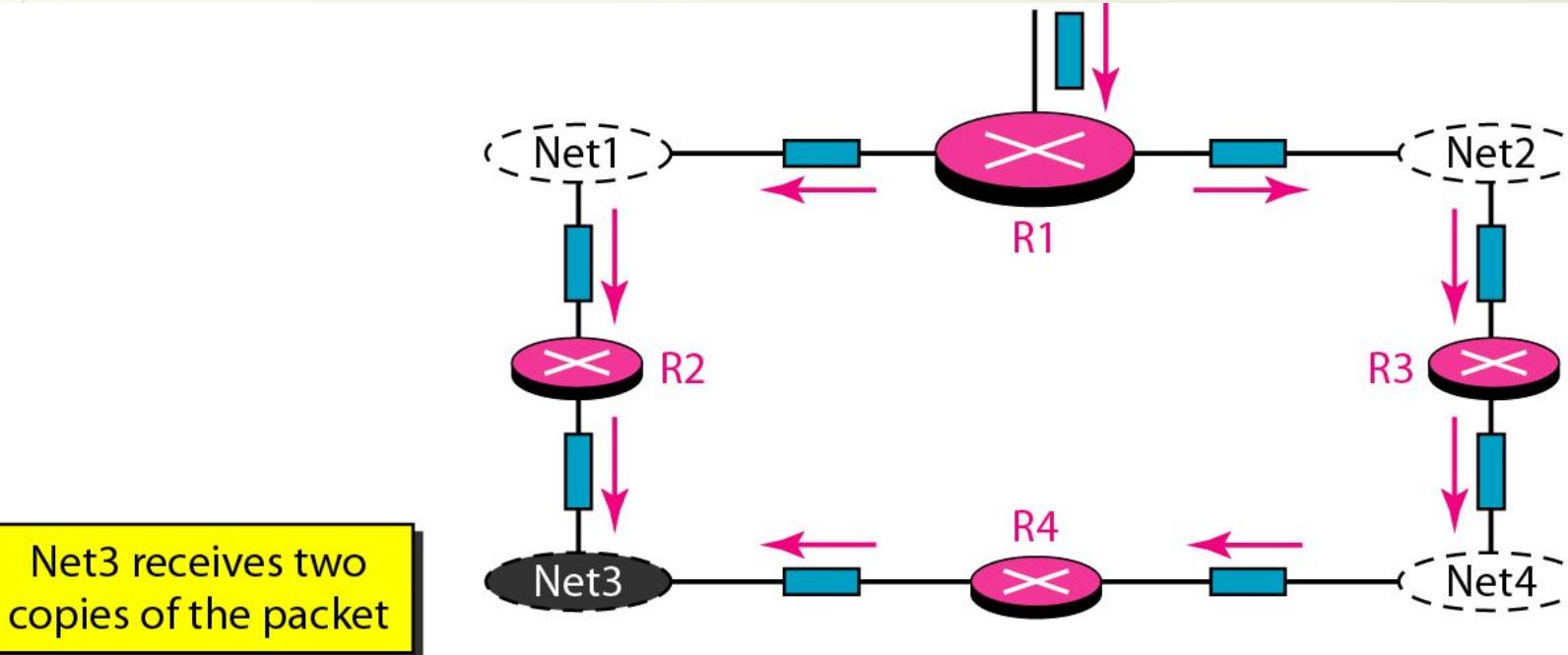
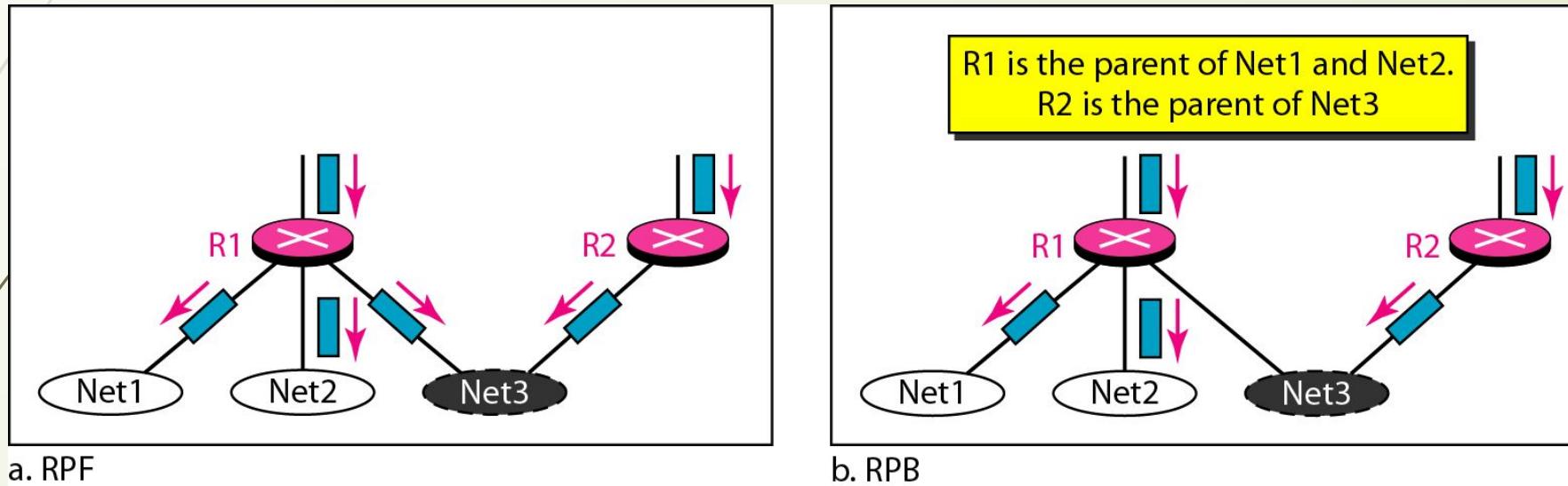
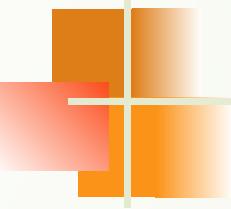


Figure 8.42 RPF Versus RPB

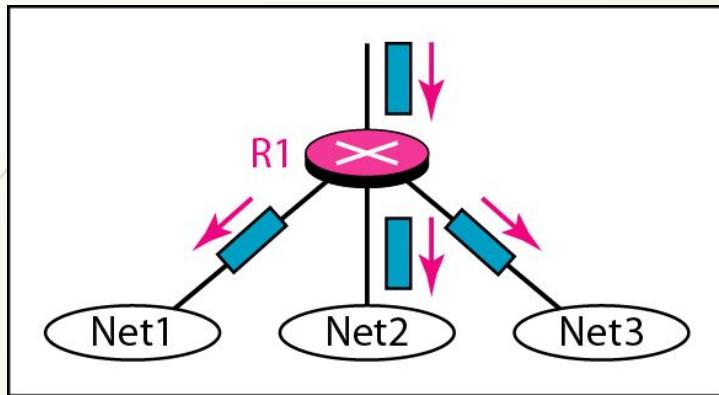




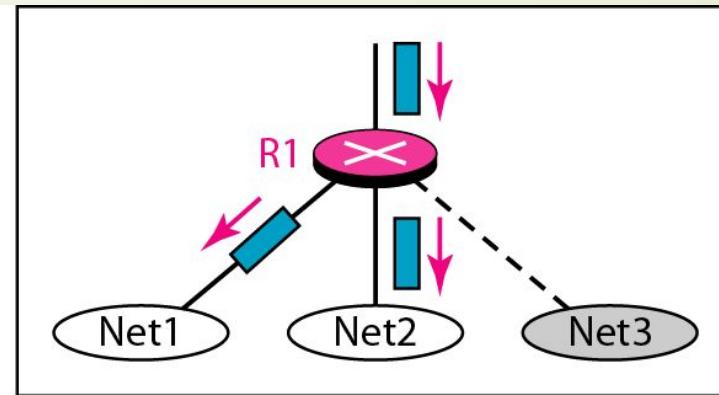
Note

RPB creates a shortest path broadcast tree from the source to each destination. It guarantees that each destination receives one and only one copy of the packet.

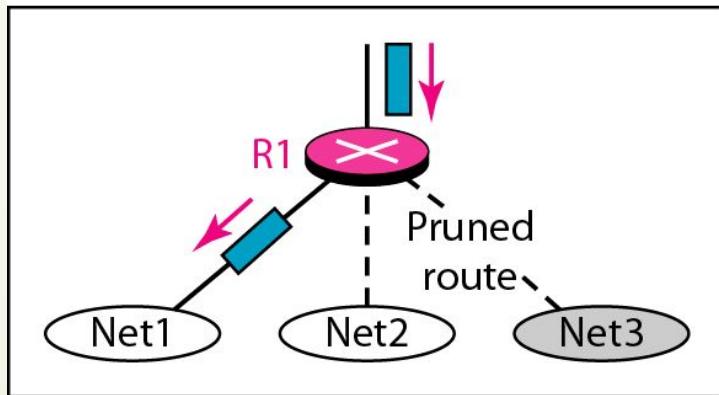
Figure 8.43 RPF, RPB, and RPM



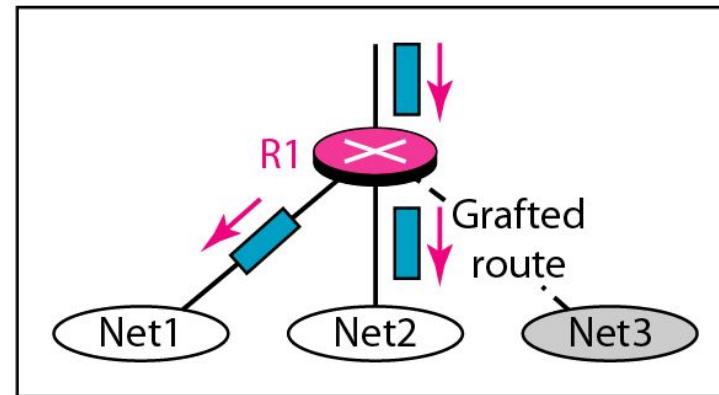
a. RPF



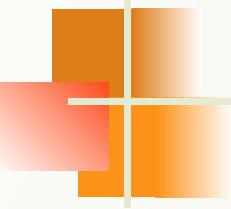
b. RPB



c. RPM (after pruning)



d. RPM (after grafting)



Note

**RPM adds pruning and grafting to RPB
to create a multicast shortest
path tree that supports dynamic
membership changes.**

Figure 8.44 Group-shared tree with rendezvous router

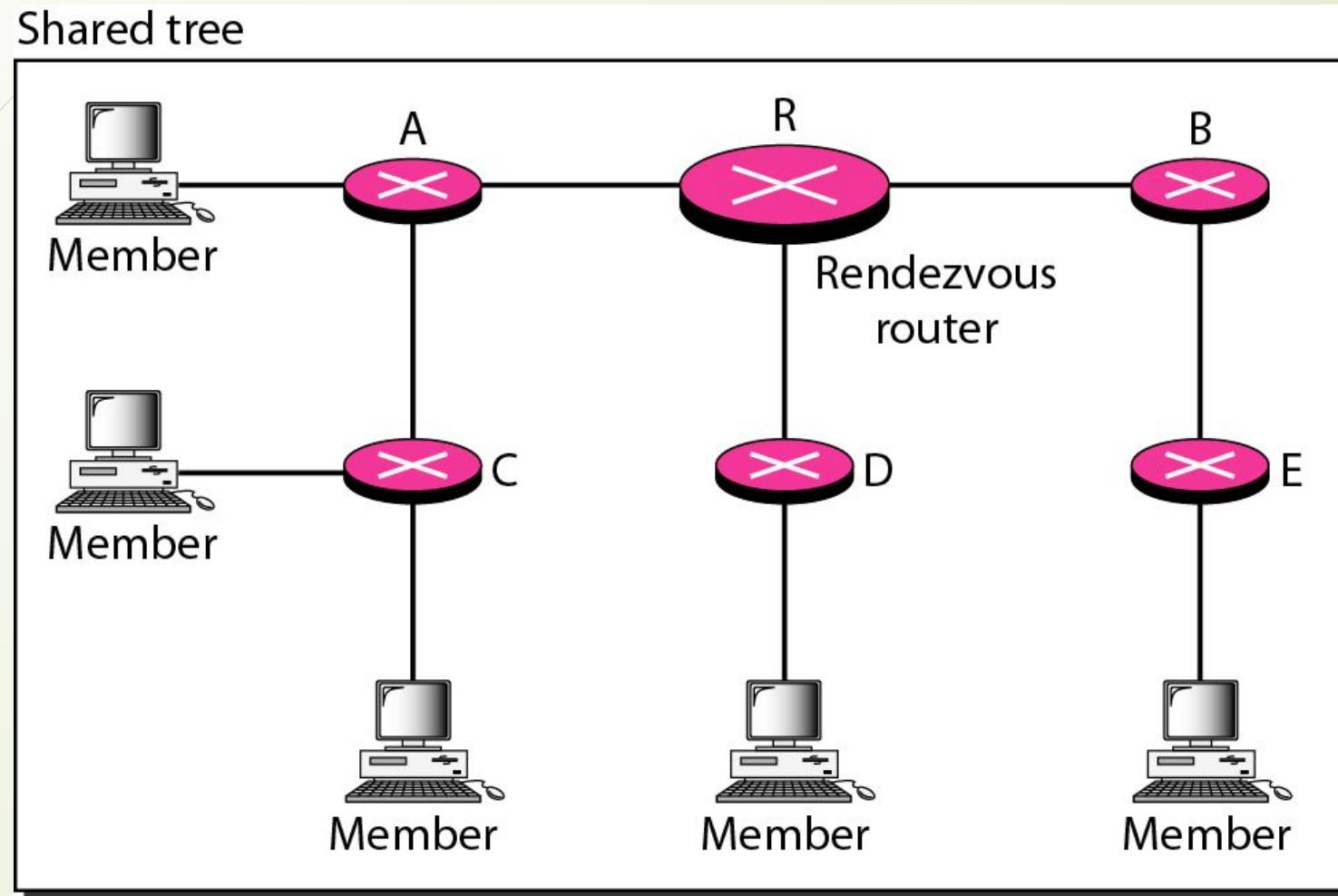
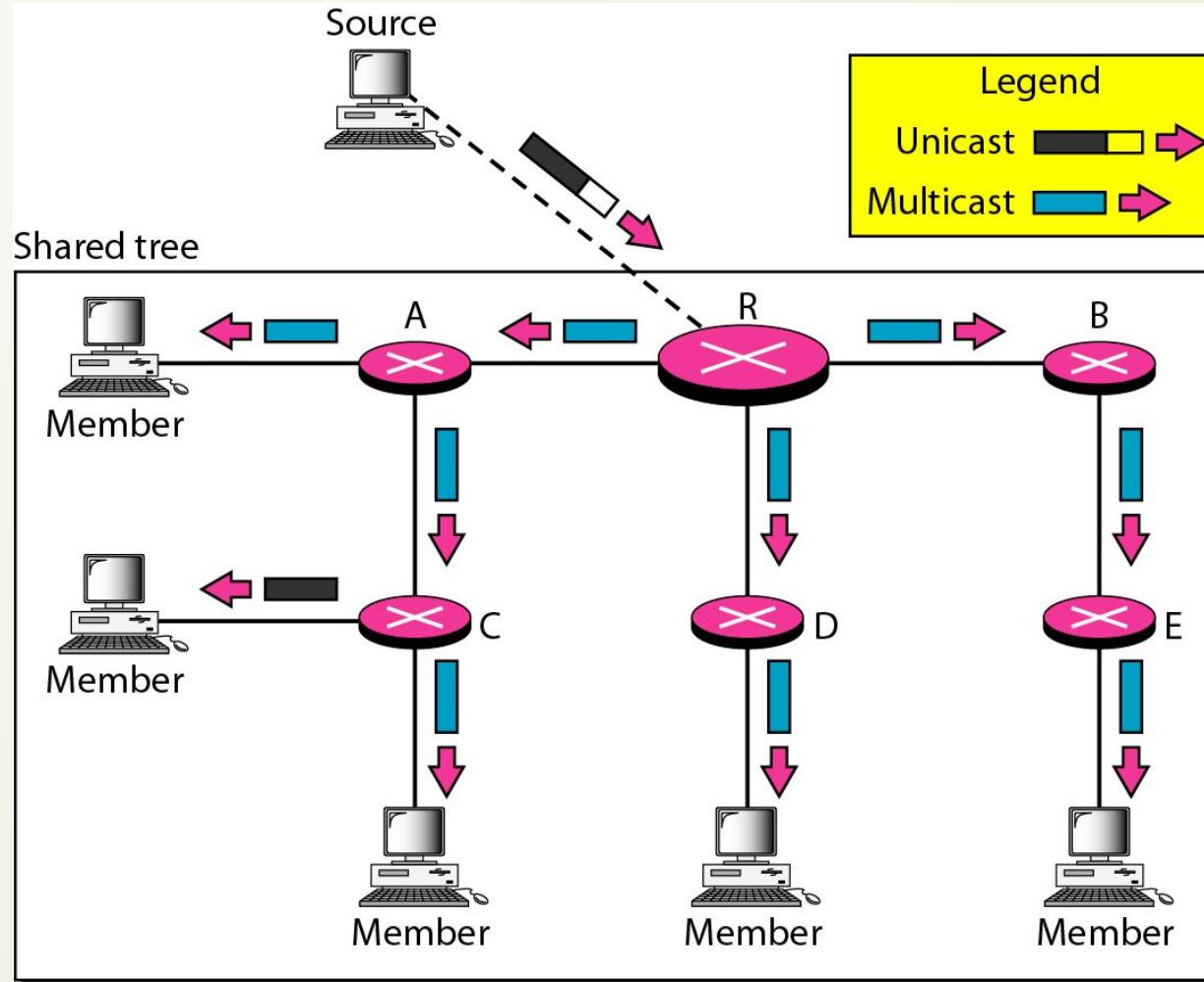
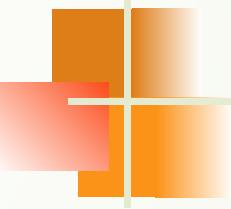


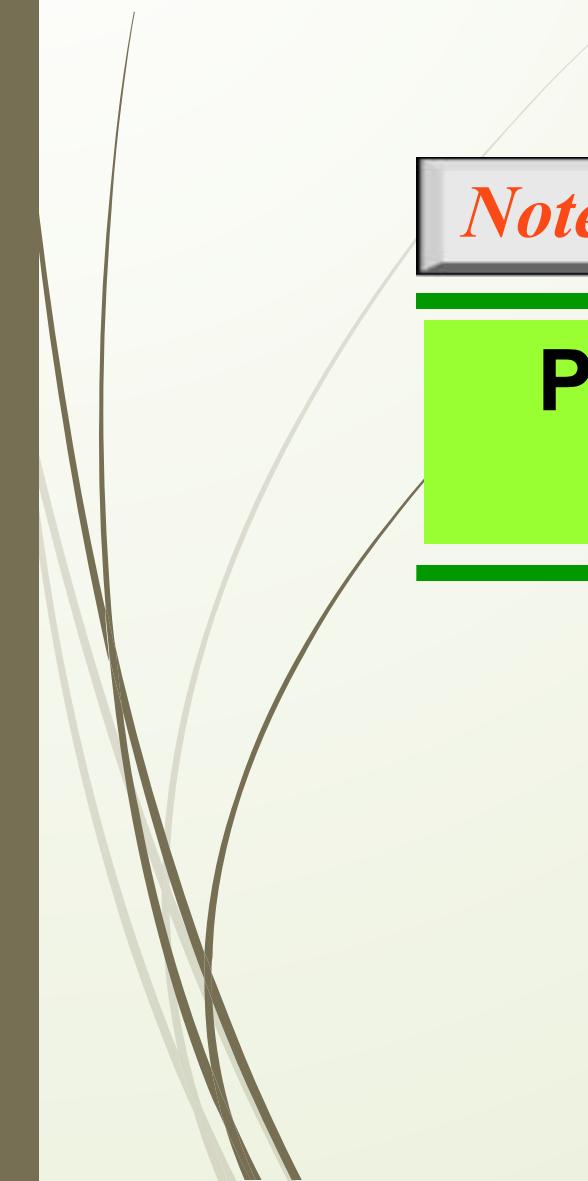
Figure 8.45 *Sending a multicast packet to the rendezvous router*





Note

In CBT, the source sends the multicast packet (encapsulated in a unicast packet) to the core router. The core router decapsulates the packet and forwards it to all interested interfaces.



Note

PIM-DM is used in a dense multicast environment, such as a LAN.



Note

PIM-DM uses RPF and pruning and grafting strategies to handle multicasting.

However, it is independent of the underlying unicast protocol.



Note

PIM-SM is used in a sparse multicast environment such as a WAN.



Note

PIM-SM is similar to CBT but uses a simpler procedure.

Figure 8.46 *Logical tunneling*

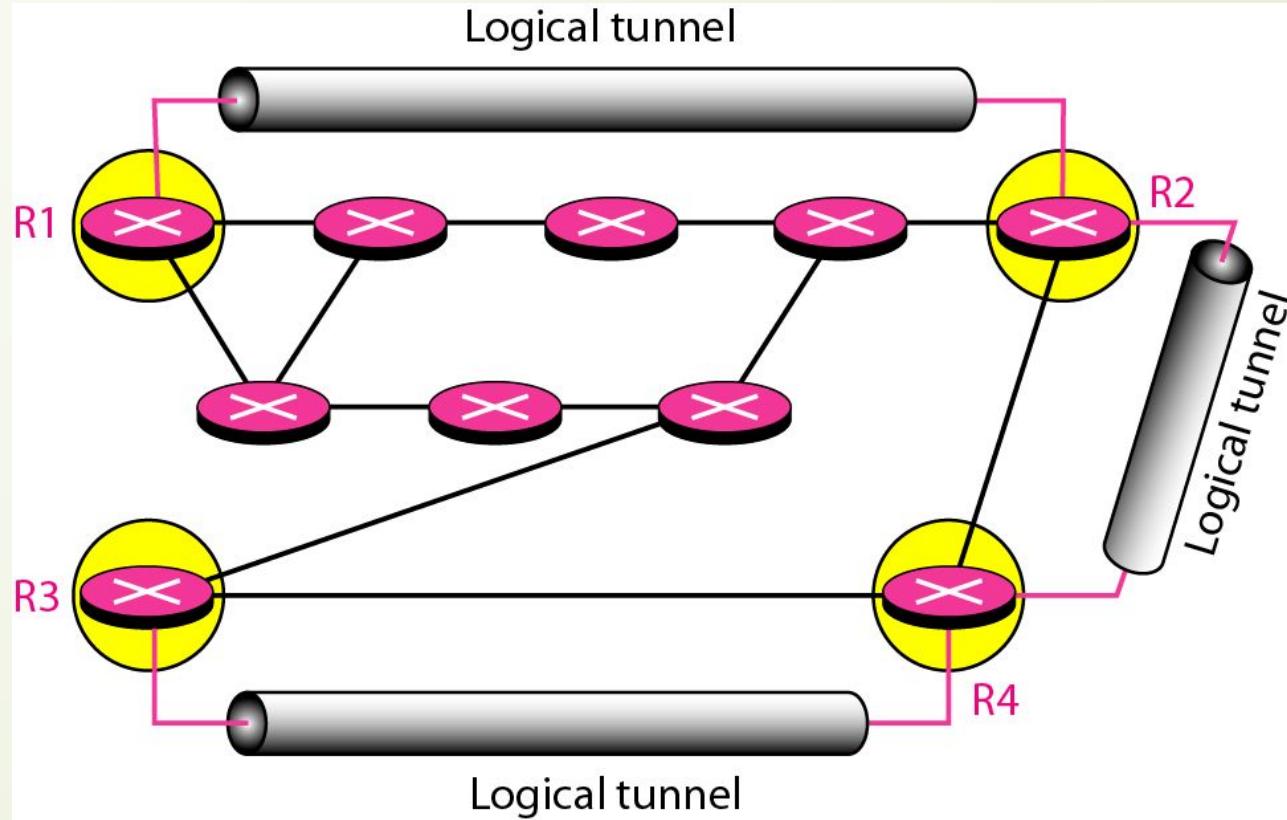
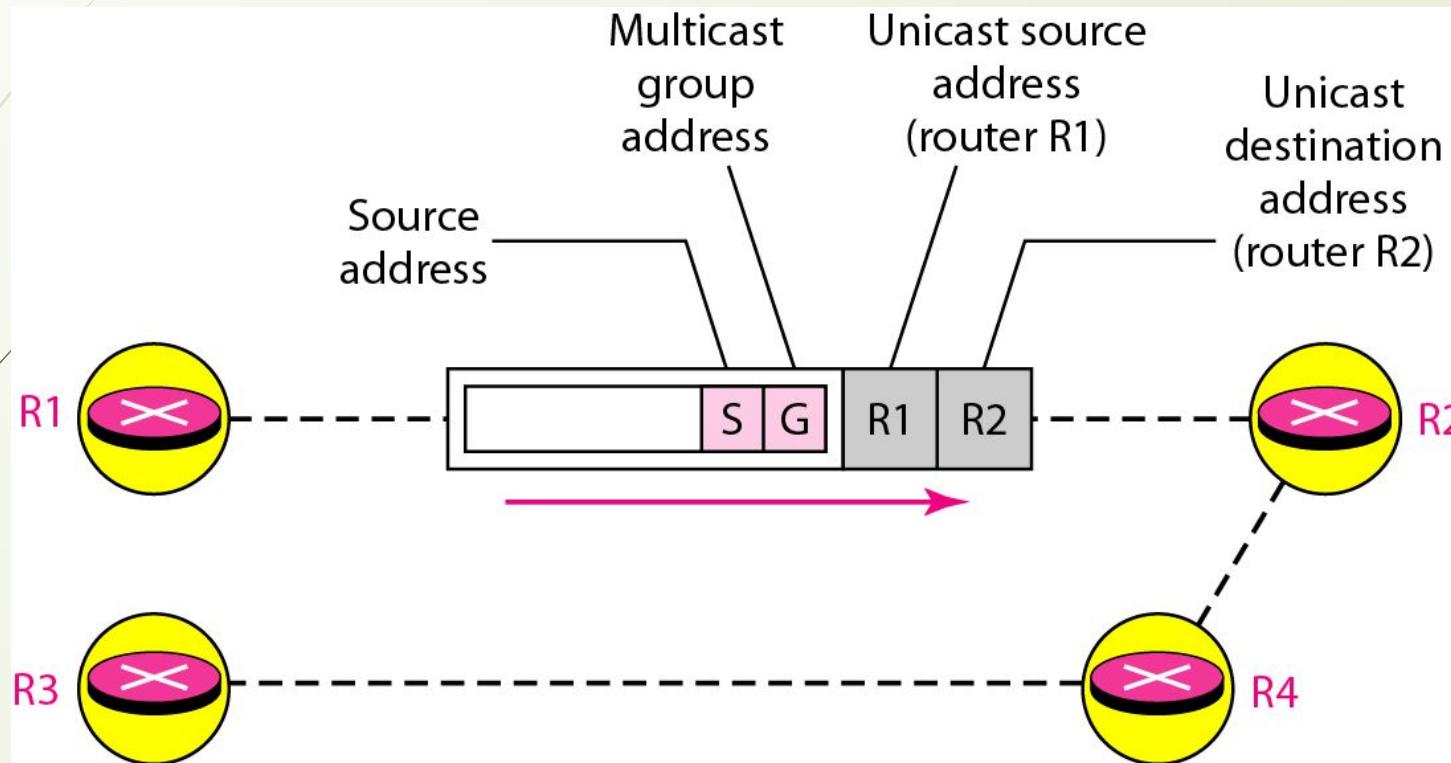


Figure 8.47 MBONE





Chapter 9

Network Layer: Logical Addressing

9-1 IPv4 ADDRESSES

An IPv4 address is a 32-bit address that uniquely and universally defines the connection of a device (for example, a computer or a router) to the Internet.

Topics discussed in this section:

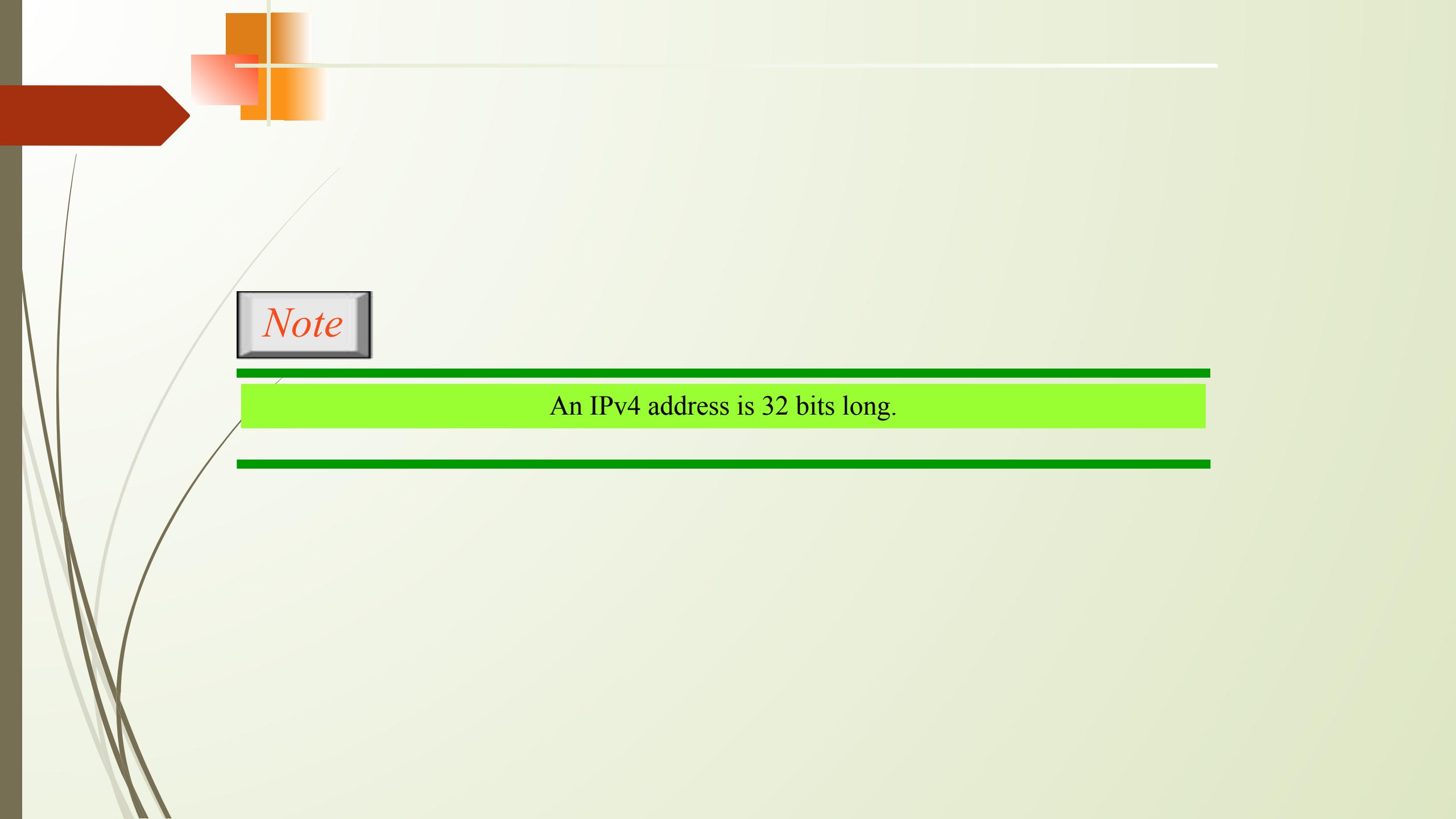
Address Space

Notations

Classful Addressing

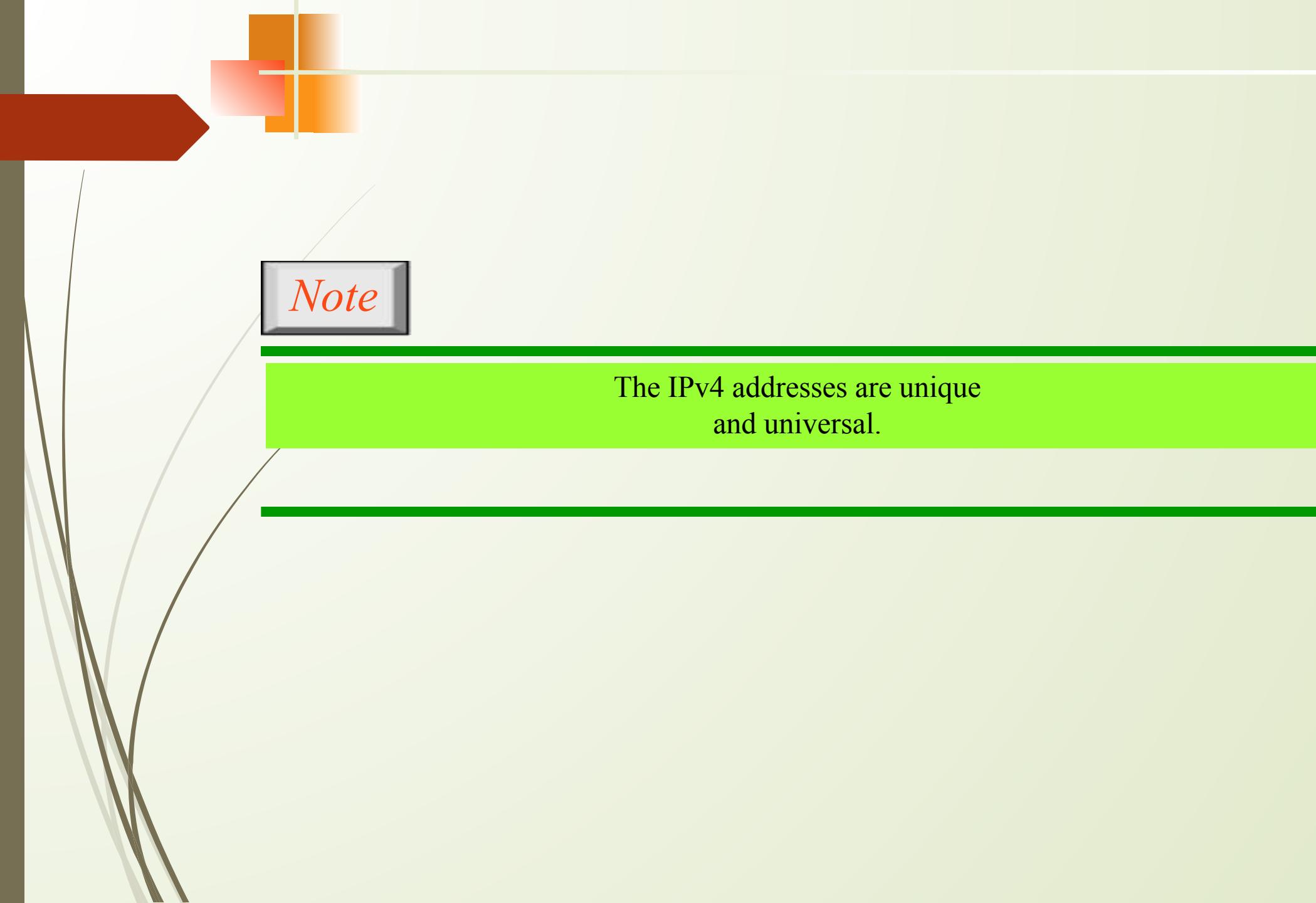
Classless Addressing

Network Address Translation (NAT)



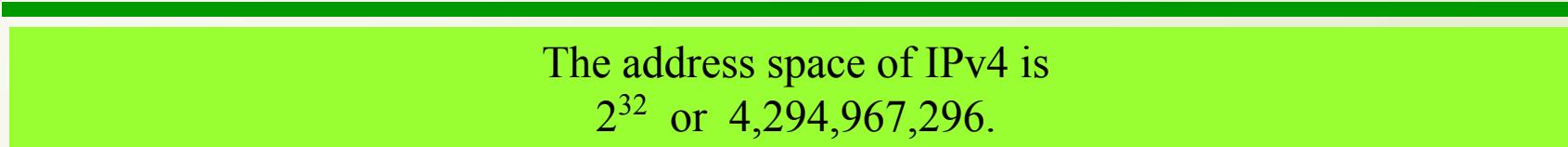
Note

An IPv4 address is 32 bits long.



Note

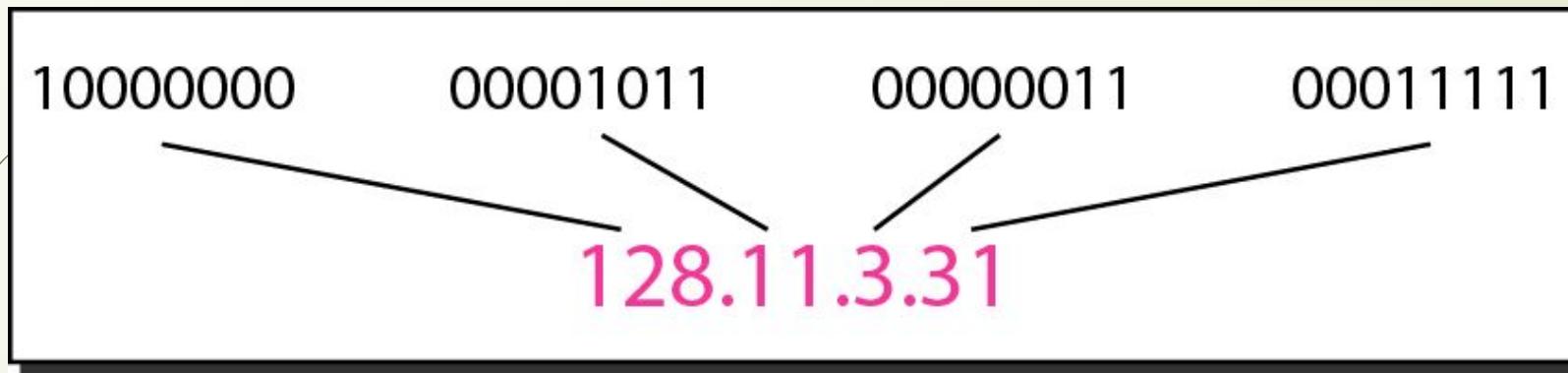
The IPv4 addresses are unique
and universal.

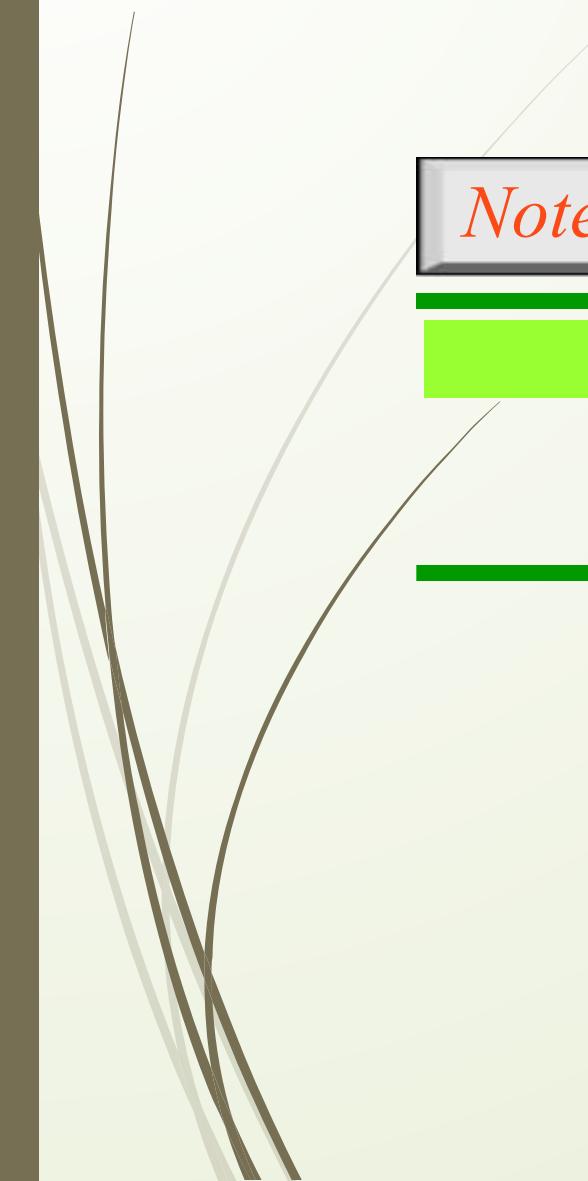
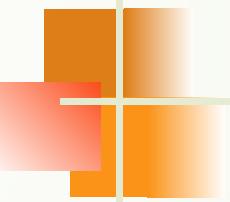


Note

The address space of IPv4 is
 2^{32} or 4,294,967,296.

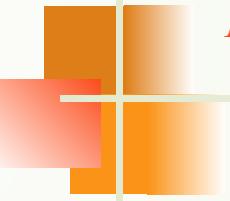
Figure 9.1 Dotted-decimal notation and binary notation for an IPv4 address





Note

Numbering systems are reviewed in Appendix B.



Example 9.1

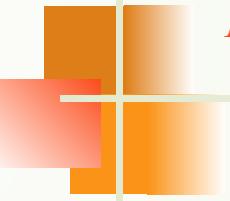
Change the following IPv4 addresses from binary notation to dotted-decimal notation.

- a. 10000001 00001011 00001011 11101111
- b. 11000001 10000011 00011011 11111111

Solution

We replace each group of 8 bits with its equivalent decimal number (see Appendix B) and add dots for separation.

- a. 129.11.11.239
- b. 193.131.27.255



Example 9.2

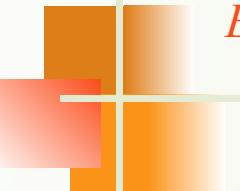
Change the following IPv4 addresses from dotted-decimal notation to binary notation.

- a. 111.56.45.78
- b. 221.34.7.82

Solution

We replace each decimal number with its binary equivalent (see Appendix B).

- a. 01101111 00111000 00101101 01001110
- b. 11011101 00100010 00000111 01010010



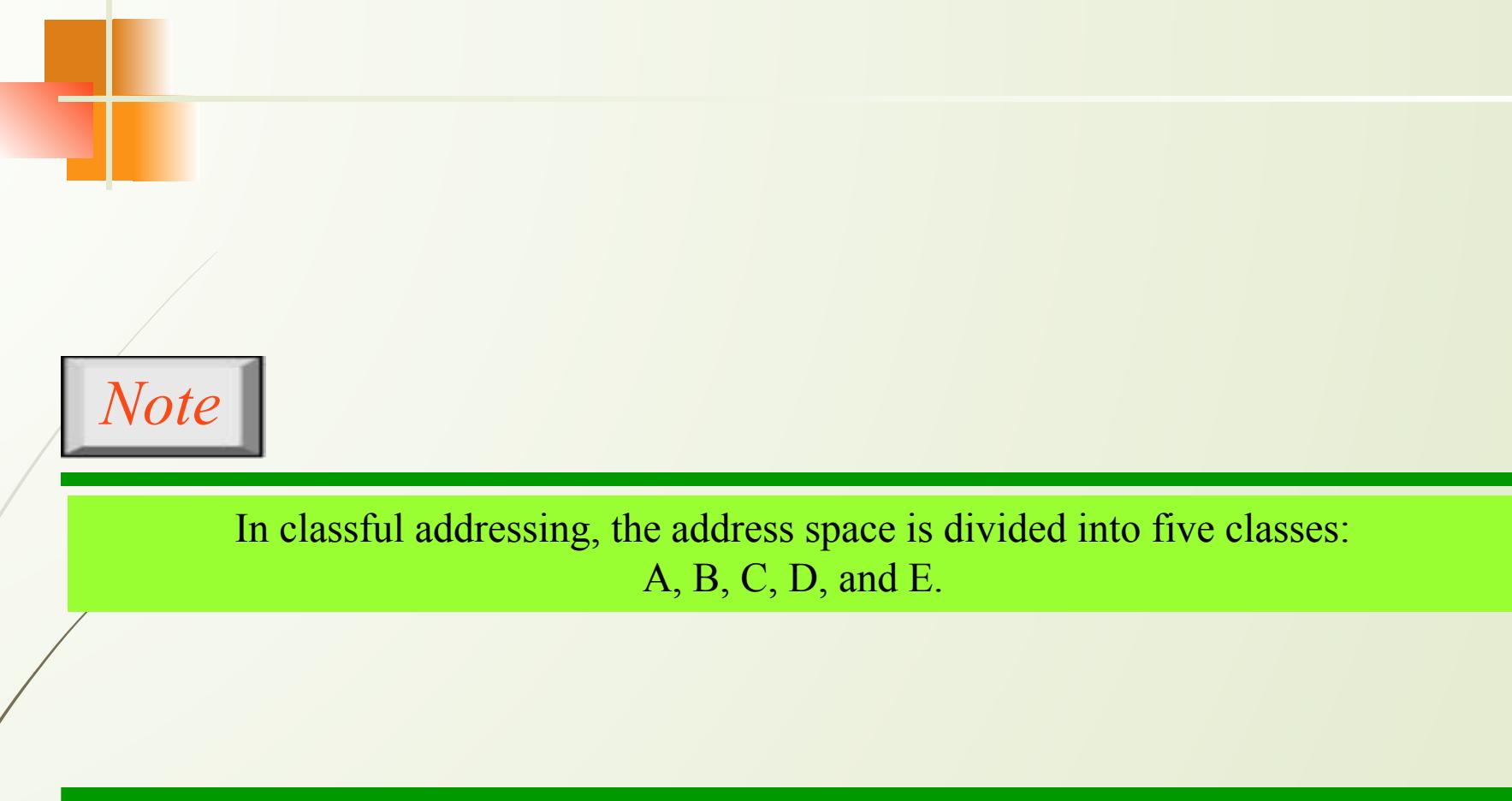
Example 9.3

Find the error, if any, in the following IPv4 addresses.

- a. 111.56.045.78
- b. 221.34.7.8.20
- c. 75.45.301.14
- d. 11100010.23.14.67

Solution

- a. *There must be no leading zero (045).*
- b. *There can be no more than four numbers.*
- c. *Each number needs to be less than or equal to 255.*
- d. *A mixture of binary notation and dotted-decimal notation is not allowed.*



Note

In classful addressing, the address space is divided into five classes:
A, B, C, D, and E.

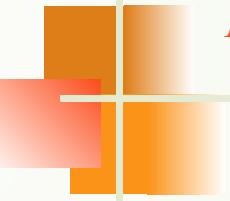
Figure 9.2 *Finding the classes in binary and dotted-decimal notation*

	First byte	Second byte	Third byte	Fourth byte
Class A	0			
Class B	10			
Class C	110			
Class D	1110			
Class E	1111			

a. Binary notation

	First byte	Second byte	Third byte	Fourth byte
Class A	0–127			
Class B	128–191			
Class C	192–223			
Class D	224–239			
Class E	240–255			

b. Dotted-decimal notation



Example 9.4

Find the class of each address.

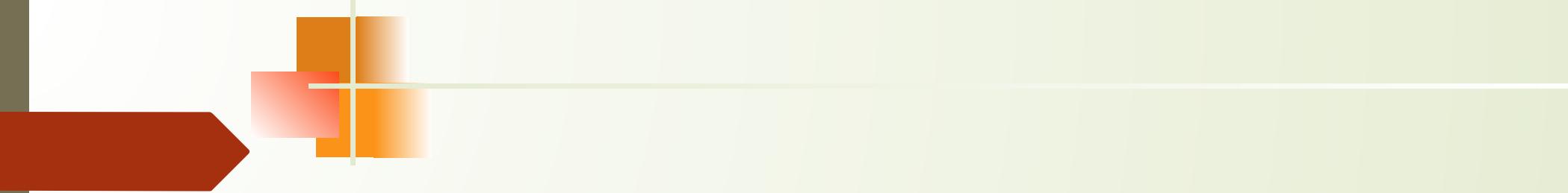
- a. 00000001 00001011 00001011 11101111
- b. 11000001 10000011 00011011 11111111
- c. 14.23.120.8
- d. 252.5.15.111

Solution

- a. *The first bit is 0. This is a class A address.*
- b. *The first 2 bits are 1; the third bit is 0. This is a class C address.*
- c. *The first byte is 14; the class is A.*
- d. *The first byte is 252; the class is E.*

Table 9.1 *Number of blocks and block size in classful IPv4 addressing*

<i>Class</i>	<i>Number of Blocks</i>	<i>Block Size</i>	<i>Application</i>
A	128	16,777,216	Unicast
B	16,384	65,536	Unicast
C	2,097,152	256	Unicast
D	1	268,435,456	Multicast
E	1	268,435,456	Reserved



Note

In classful addressing, a large part of the available addresses were wasted.

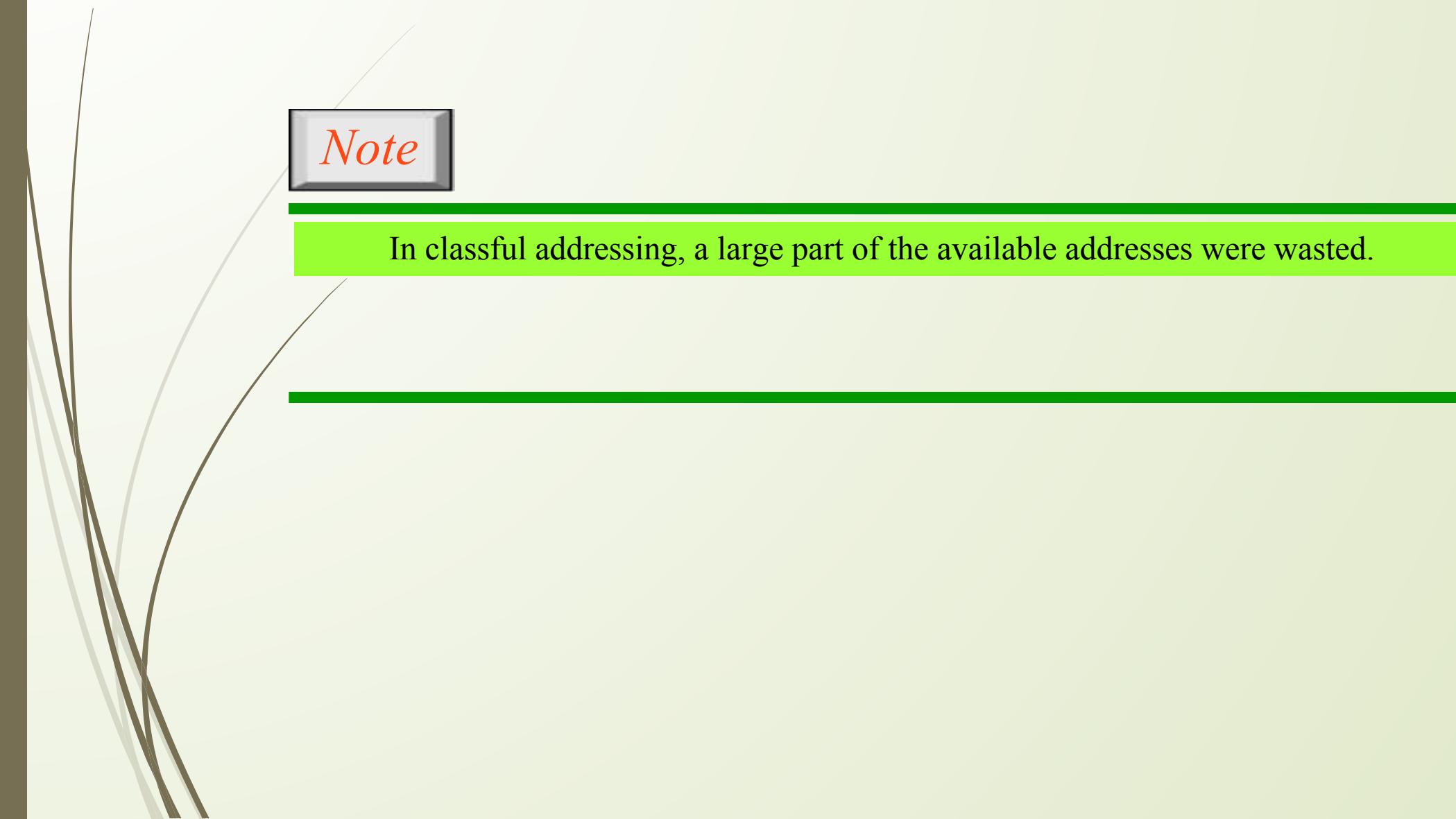


Table 9.2 *Default masks for classful addressing*

Class	Binary	Dotted-Decimal	CIDR
A	11111111 00000000 00000000 00000000	255.0.0.0	/8
B	11111111 11111111 00000000 00000000	255.255.0.0	/16
C	11111111 11111111 11111111 00000000	255.255.255.0	/24



Note

Classful addressing, which is almost obsolete, is replaced with classless addressing.

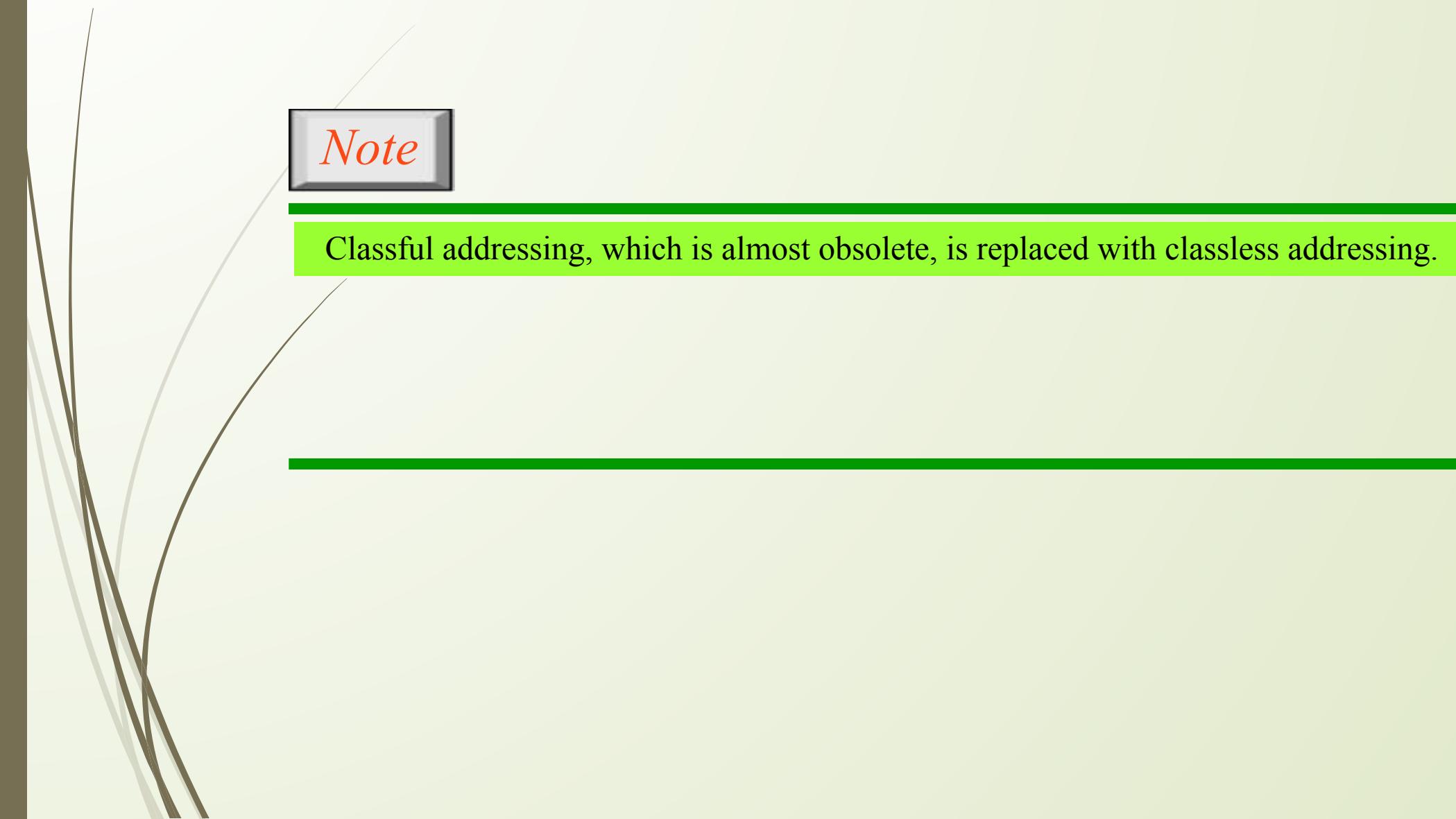
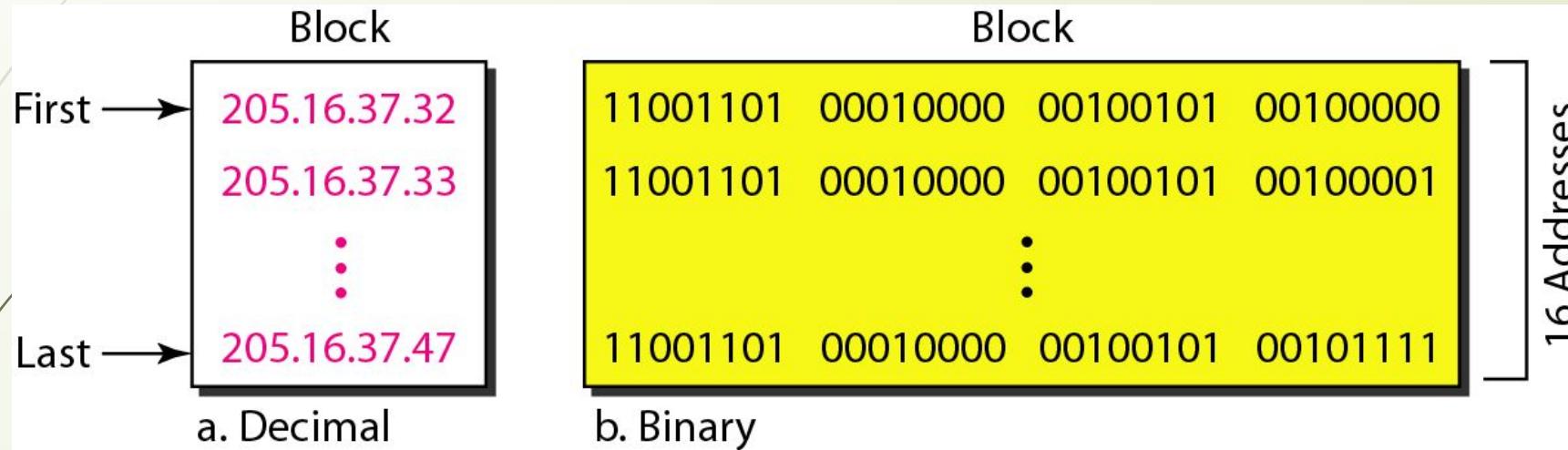
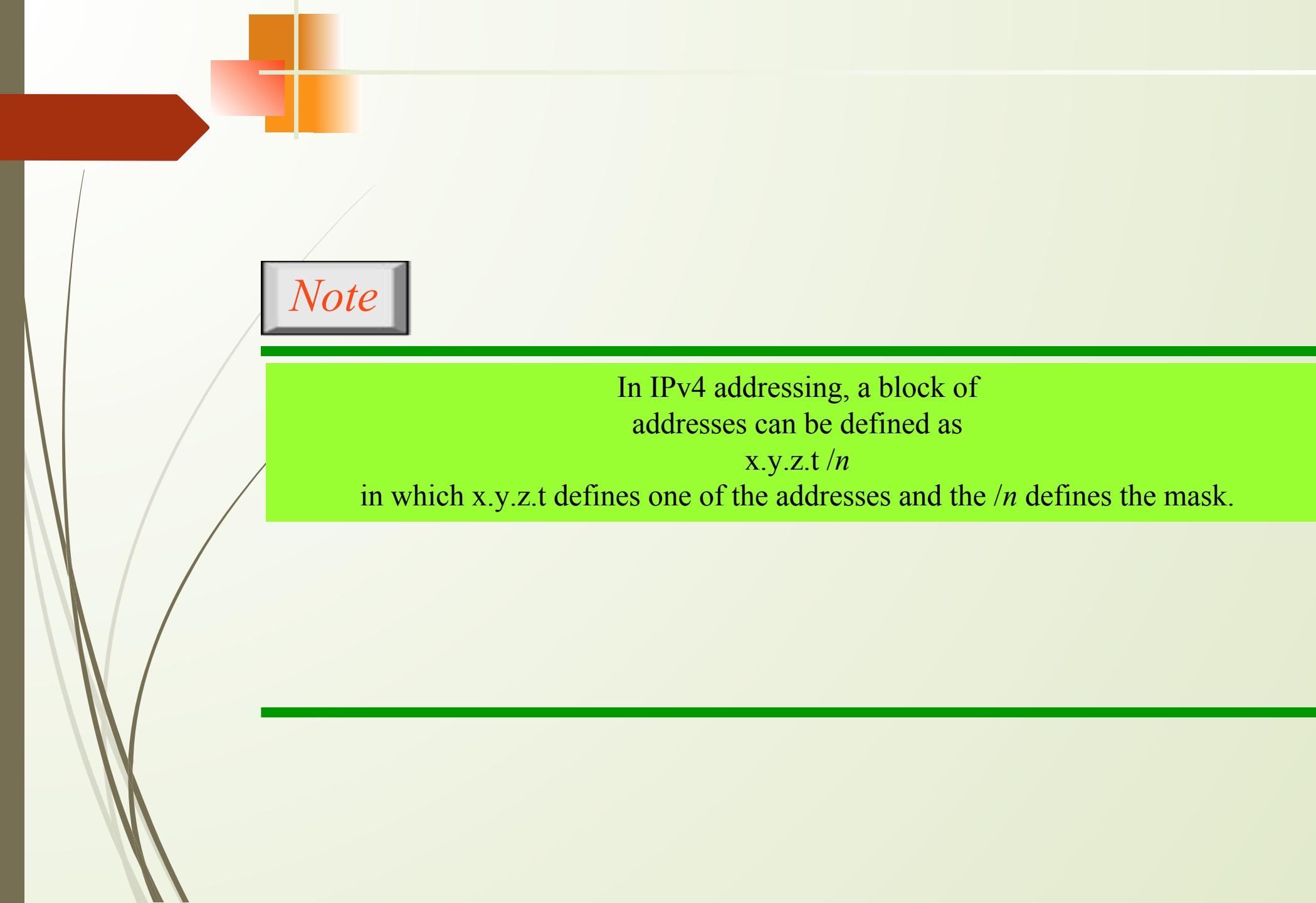


Figure 9.3 shows a block of addresses, in both binary and dotted-decimal notation, granted to a small business that needs 16 addresses.

We can see that the restrictions are applied to this block. The addresses are contiguous. The number of addresses is a power of 2 ($16 = 2^4$), and the first address is divisible by 16. The first address, when converted to a decimal number, is 3,440,387,360, which when divided by 16 results in 215,024,210.

Figure 9.3 A block of 16 addresses granted to a small organization



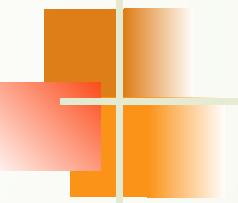


Note

In IPv4 addressing, a block of addresses can be defined as

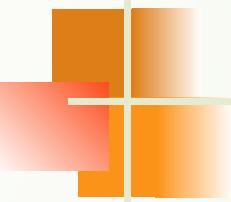
$x.y.z.t/n$

in which $x.y.z.t$ defines one of the addresses and the $/n$ defines the mask.



Note

The first address in the block can be found by setting the rightmost
 $32 - n$ bits to 0s.



Example 9.6

A block of addresses is granted to a small organization. We know that one of the addresses is 205.16.37.39/28. What is the first address in the block?

Solution

The binary representation of the given address is

11001101 00010000 00100101 00100111

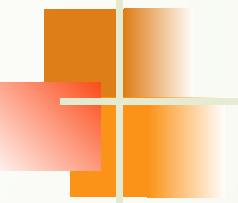
If we set 32–28 rightmost bits to 0, we get

11001101 00010000 00100101 0010000

or

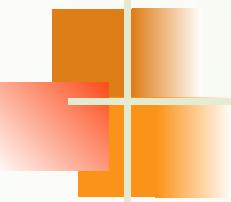
205.16.37.32.

This is actually the block shown in Figure 9.3.



Note

The last address in the block can be found by setting the rightmost
 $32 - n$ bits to 1s.



Example 9.7

Find the last address for the block in Example 9.6.

Solution

The binary representation of the given address is

11001101 00010000 00100101 00100111

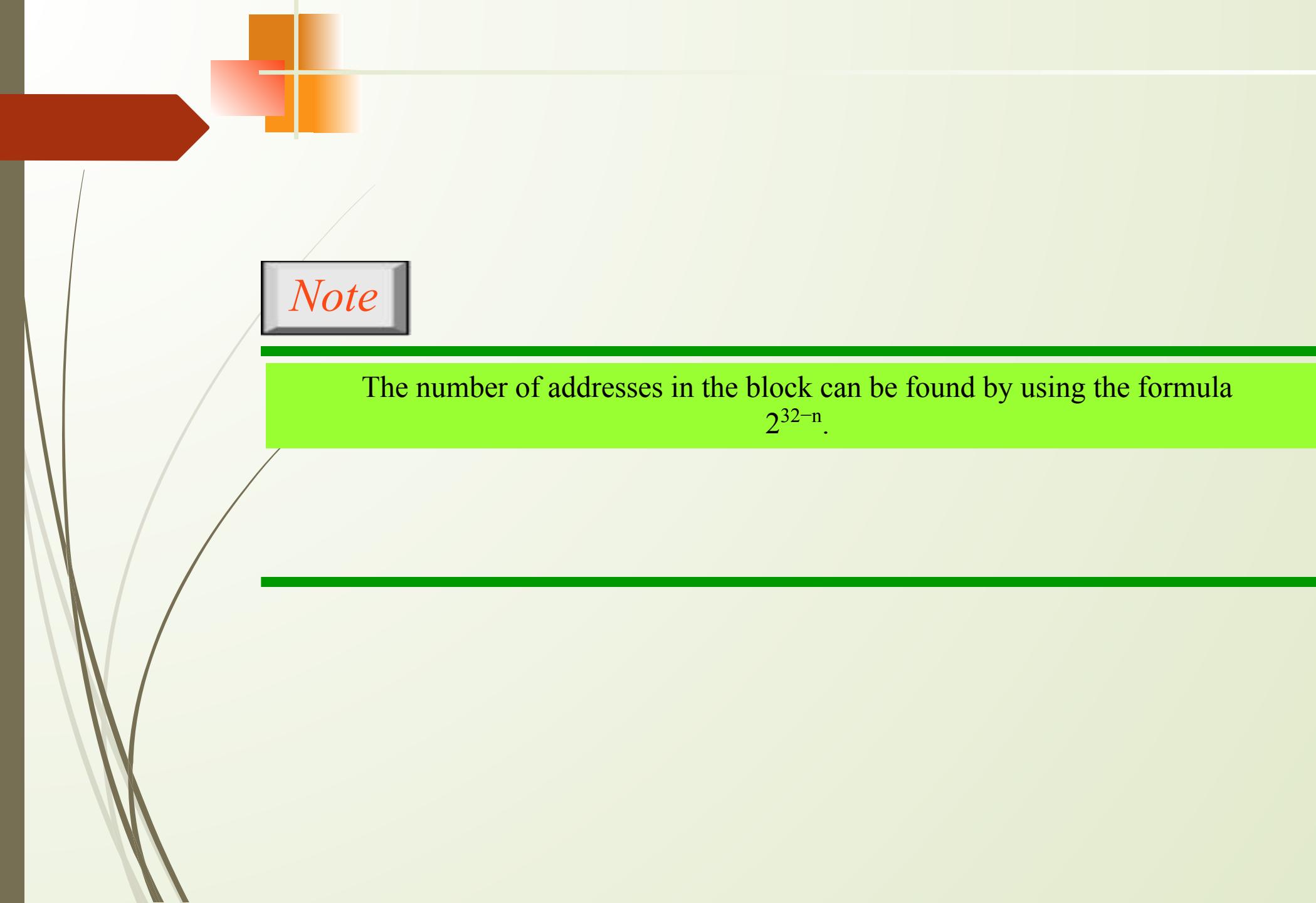
If we set 32 – 28 rightmost bits to 1, we get

11001101 00010000 00100101 00101111

or

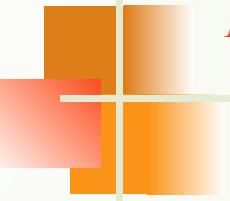
205.16.37.47

This is actually the block shown in Figure 9.3.



Note

The number of addresses in the block can be found by using the formula
 2^{32-n} .

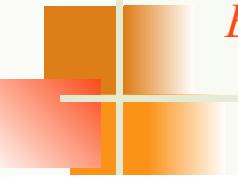


Example 9.8

Find the number of addresses in Example 9.6.

Solution

The value of n is 28, which means that number of addresses is 2^{32-28} or 16.



Example 9.9

Another way to find the first address, the last address, and the number of addresses is to represent the mask as a 32-bit binary (or 8-digit hexadecimal) number. This is particularly useful when we are writing a program to find these pieces of information. In Example 9.5 the /28 can be represented as

11111111 11111111 11111111 11110000

(twenty-eight 1s and four 0s).

Find

- The first address
- The last address
- The number of addresses.

Solution

a. The first address can be found by ANDing the given addresses with the mask. ANDing here is done bit by bit. The result of ANDing 2 bits is 1 if both bits are 1s; the result is 0 otherwise.

Address:	11001101 00010000 00100101 00100111
Mask:	11111111 11111111 11111111 11110000
First address:	11001101 00010000 00100101 00100000

b. The last address can be found by ORing the given addresses with the complement of the mask. ORing here is done bit by bit. The result of ORing 2 bits is 0 if both bits are 0s; the result is 1 otherwise. The complement of a number is found by changing each 1 to 0 and each 0 to 1.

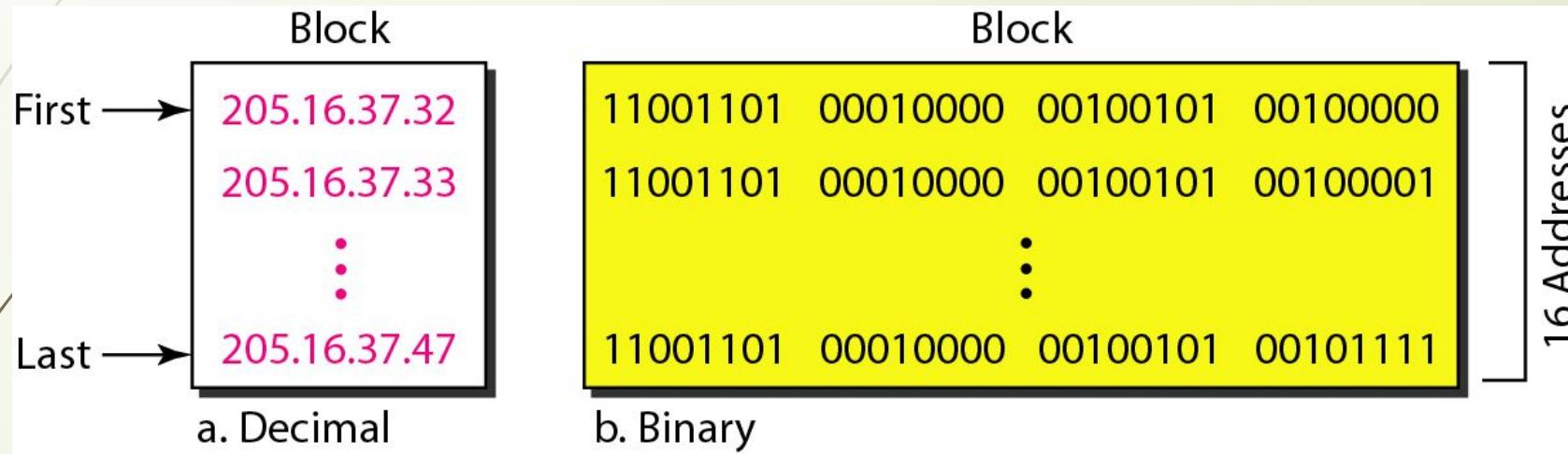
Address:	11001101	00010000	00100101	00100111
Mask complement:	00000000	00000000	00000000	00001111
Last address:	11001101	00010000	00100101	00101111

- c. The number of addresses can be found by complementing the mask, interpreting it as a decimal number, and adding 1 to it.

Mask complement: **00000000 00000000 00000000 00001111**

Number of addresses: $15 + 1 = 16$

Figure 9.4 A network configuration for the block 205.16.37.32/28





Note

The first address in a block is normally not assigned to any device; it is used as the network address that represents the organization to the rest of the world.

Figure 9.5 Two levels of hierarchy in an IPv4 address

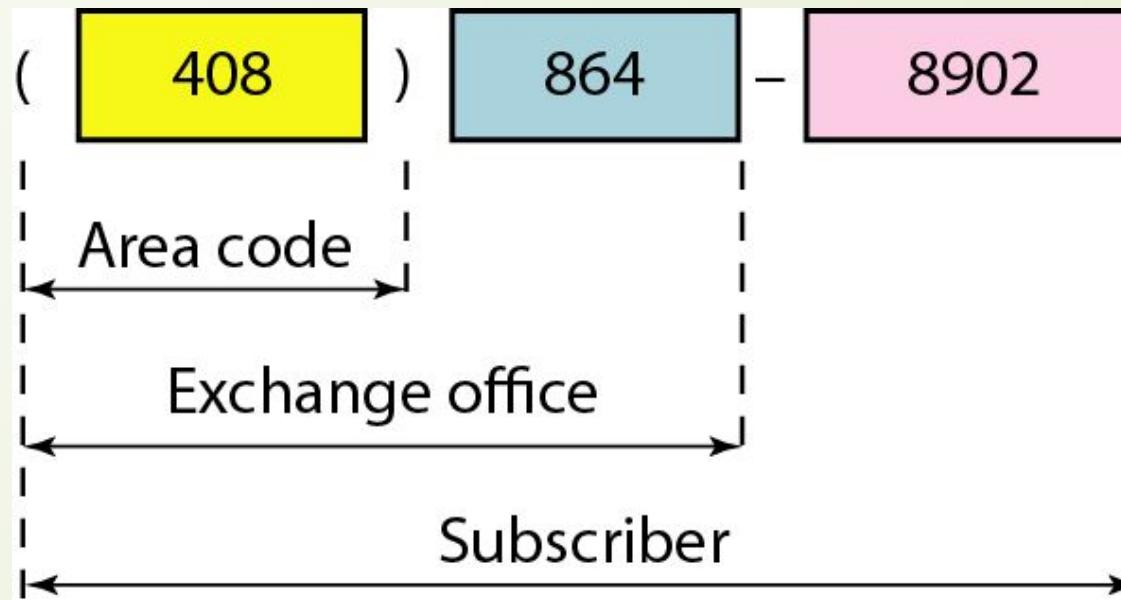
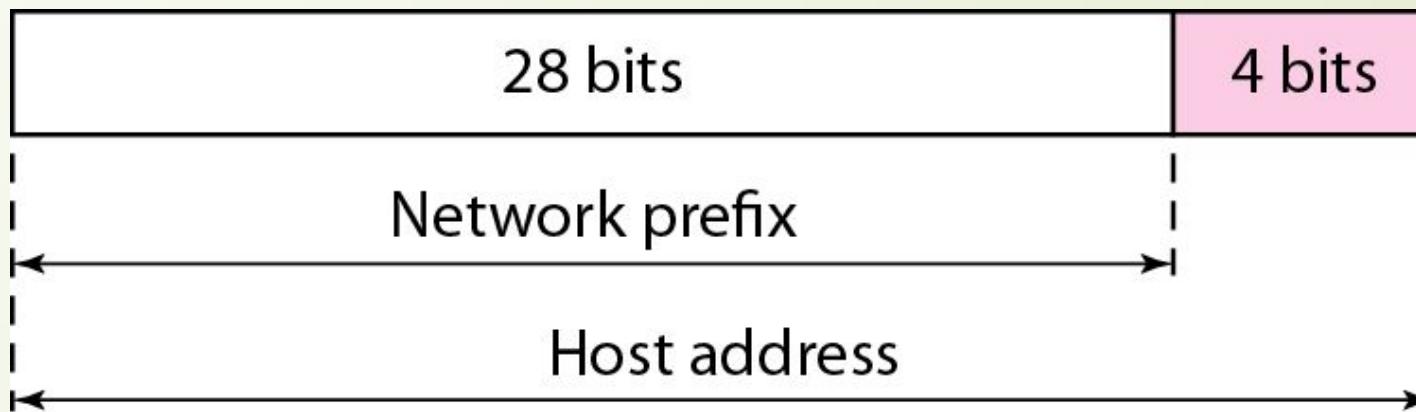
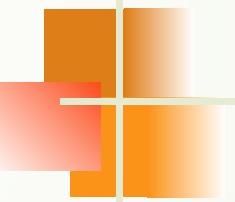


Figure 9.6 A frame in a character-oriented protocol





Note

Each address in the block can be considered as a two-level hierarchical structure:
the leftmost n bits (prefix) define the network;
the rightmost $32 - n$ bits define the host.

Figure 9.7 Configuration and addresses in a subnetted network

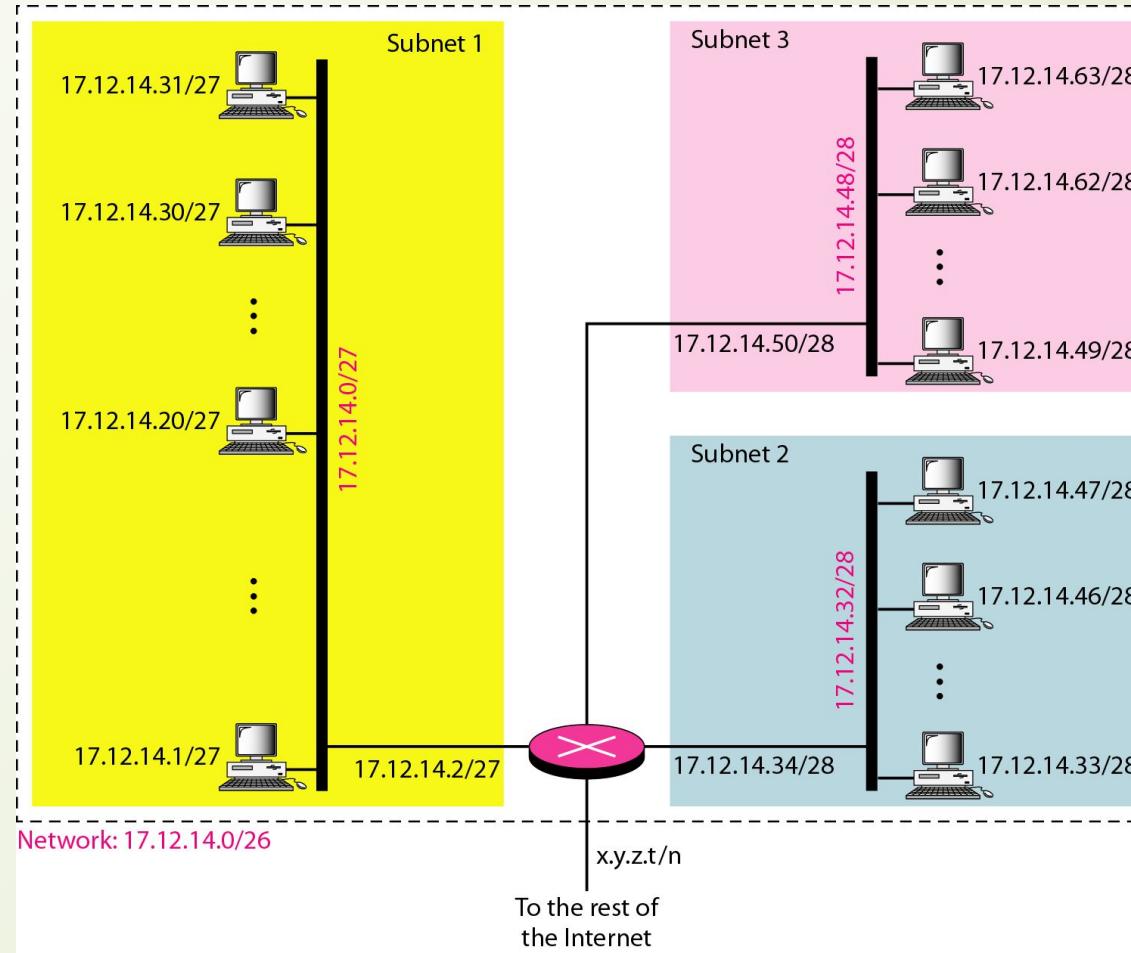
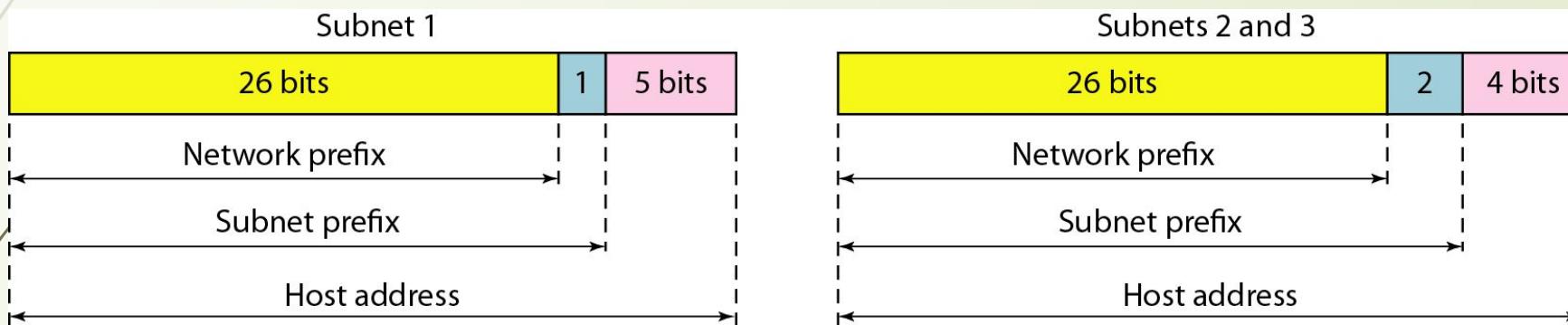
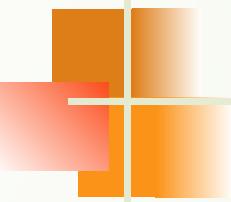


Figure 9.8 Three-level hierarchy in an IPv4 address





Example 9.10

An ISP is granted a block of addresses starting with 190.100.0.0/16 (65,536 addresses). The ISP needs to distribute these addresses to three groups of customers as follows:

- a. The first group has 64 customers; each needs 256 addresses.*
- b. The second group has 128 customers; each needs 128 addresses.*
- c. The third group has 128 customers; each needs 64 addresses.*

Design the subblocks and find out how many addresses are still available after these allocations.

Solution

Figure 9.9 shows the situation.

Group 1

For this group, each customer needs 256 addresses. This means that 8 ($\log_2 256$) bits are needed to define each host. The prefix length is then $32 - 8 = 24$. The addresses are

1st Customer: 190.100.0.0/24 190.100.0.255/24

2nd Customer: 190.100.1.0/24 190.100.1.255/24

...

64th Customer: 190.100.63.0/24 190.100.63.255/24

Total = $64 \times 256 = 16,384$

Group 2

For this group, each customer needs 128 addresses. This means that $7 (\log_2 128)$ bits are needed to define each host. The prefix length is then $32 - 7 = 25$. The addresses are

1st Customer: 190.100.64.0/25 190.100.64.127/25

2nd Customer: 190.100.64.128/25 190.100.64.255/25

...

128th Customer: 190.100.127.128/25 190.100.127.255/25

Total = $128 \times 128 = 16,384$

Group 3

For this group, each customer needs 64 addresses. This means that $6 (\log_2 64)$ bits are needed to each host. The prefix length is then $32 - 6 = 26$. The addresses are

1st Customer: 190.100.128.0/26 190.100.128.63/26

2nd Customer: 190.100.128.64/26 190.100.128.127/26

...

128th Customer: 190.100.159.192/26 190.100.159.255/26

Total = $128 \times 64 = 8192$

Number of granted addresses to the ISP: 65,536

Number of allocated addresses by the ISP: 40,960

Number of available addresses: 24,576

Figure 9.9 An example of address allocation and distribution by an ISP

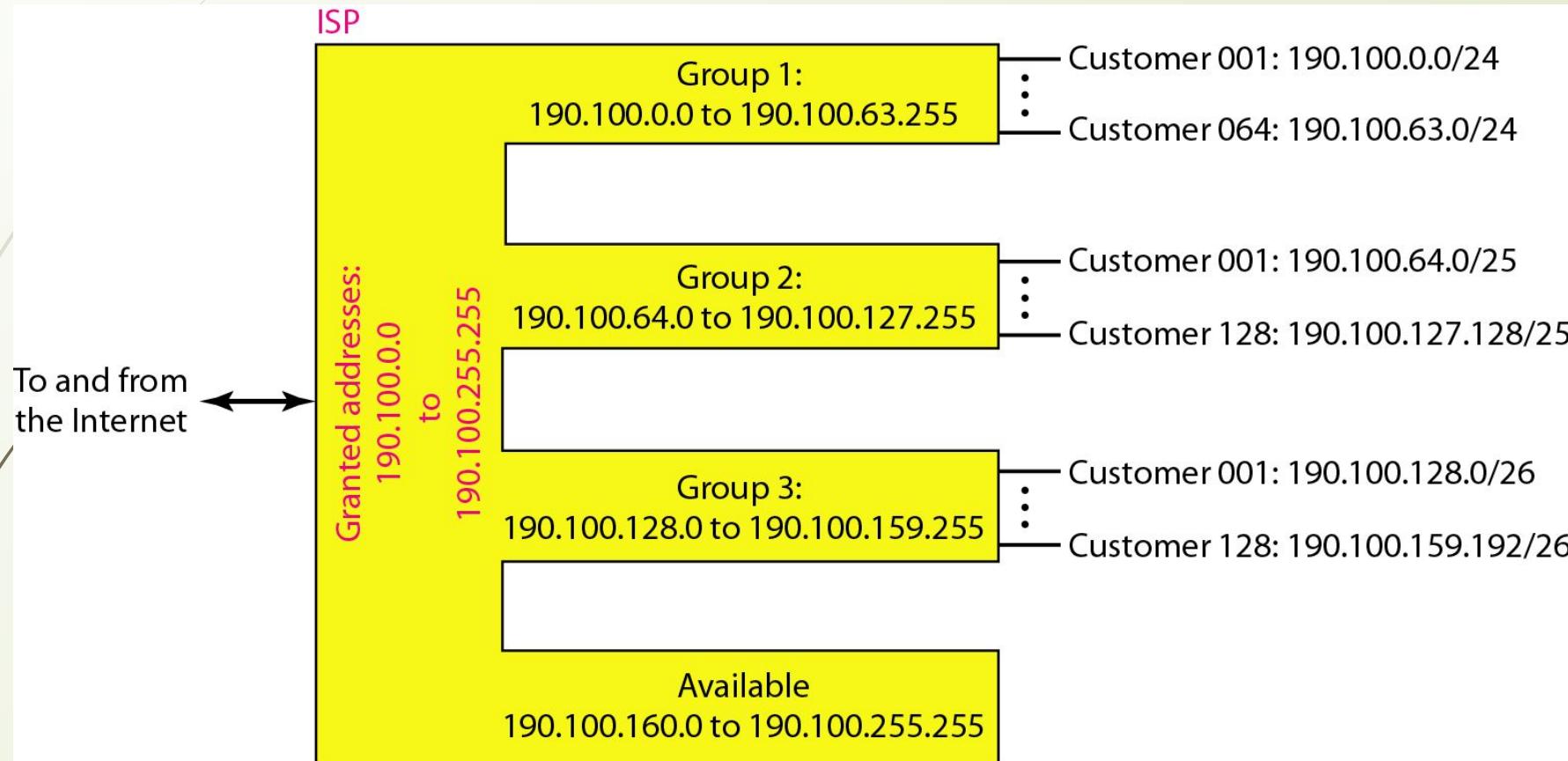




Table 9.3 *Addresses for private networks*

<i>Range</i>	<i>Total</i>
10.0.0.0 to 10.255.255.255	2^{24}
172.16.0.0 to 172.31.255.255	2^{20}
192.168.0.0 to 192.168.255.255	2^{16}

Figure 9.10 A NAT implementation

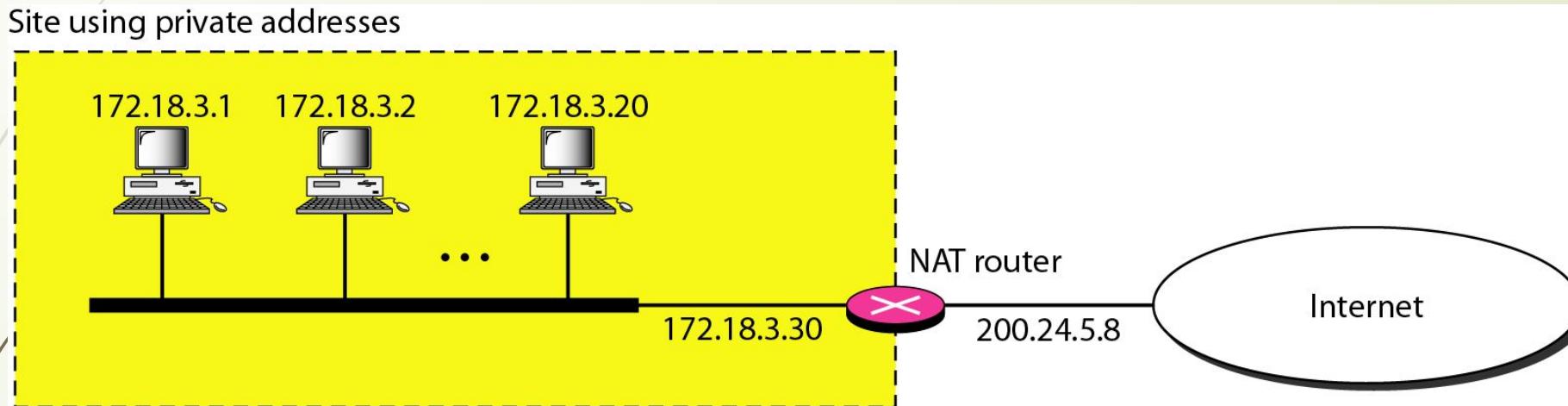


Figure 9.11 Addresses in a NAT

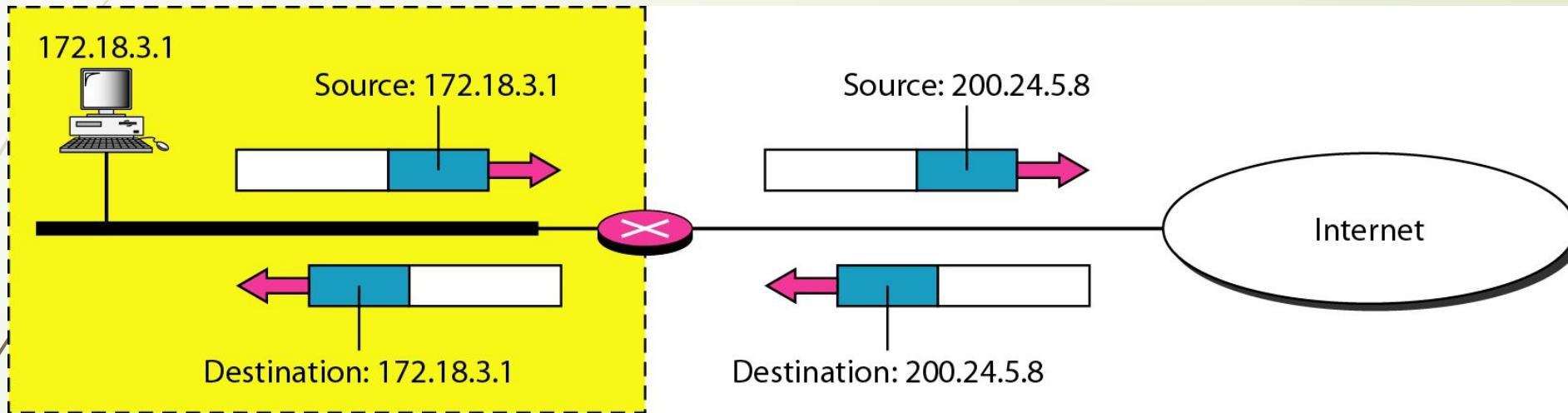
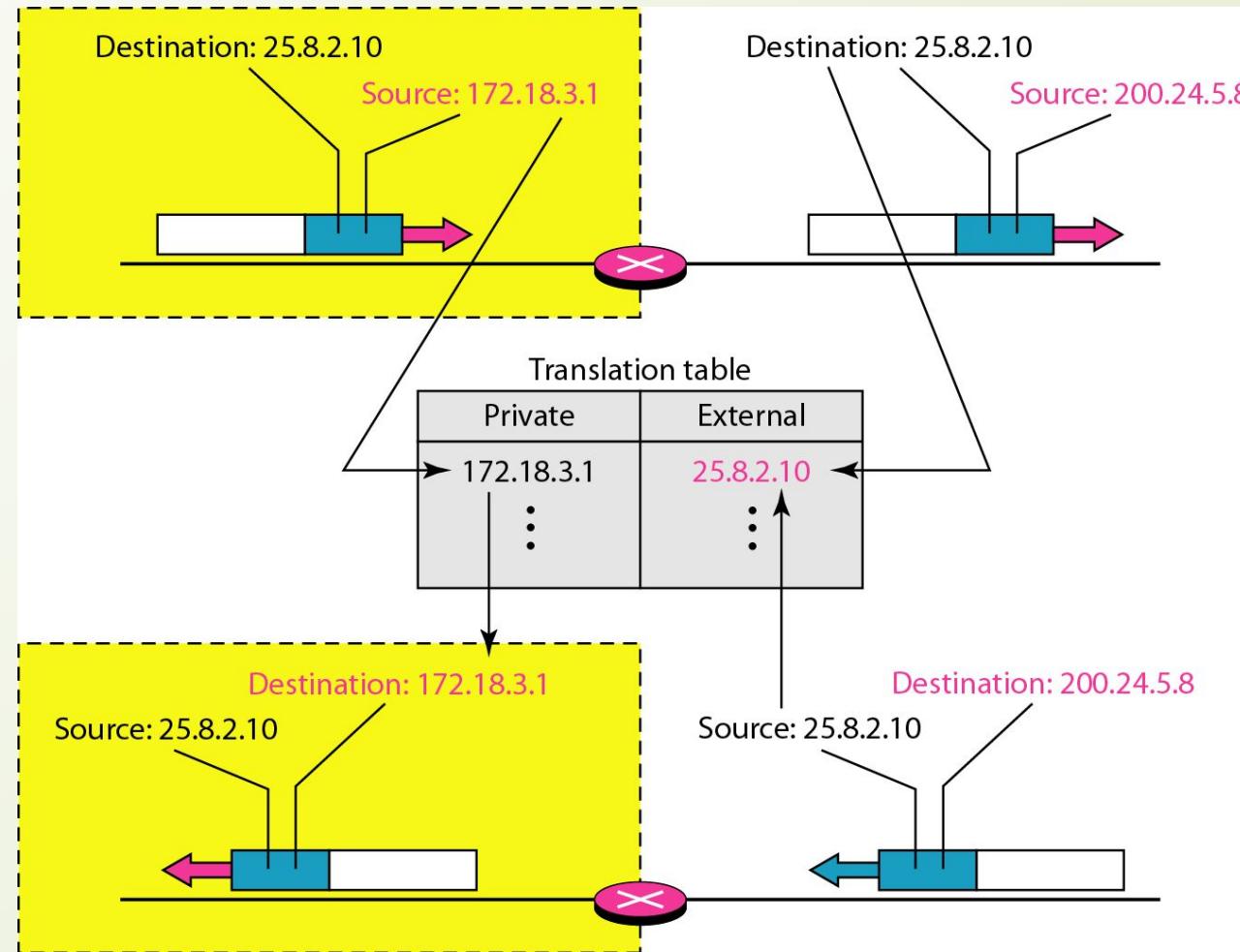


Figure 9.12 NAT address translation



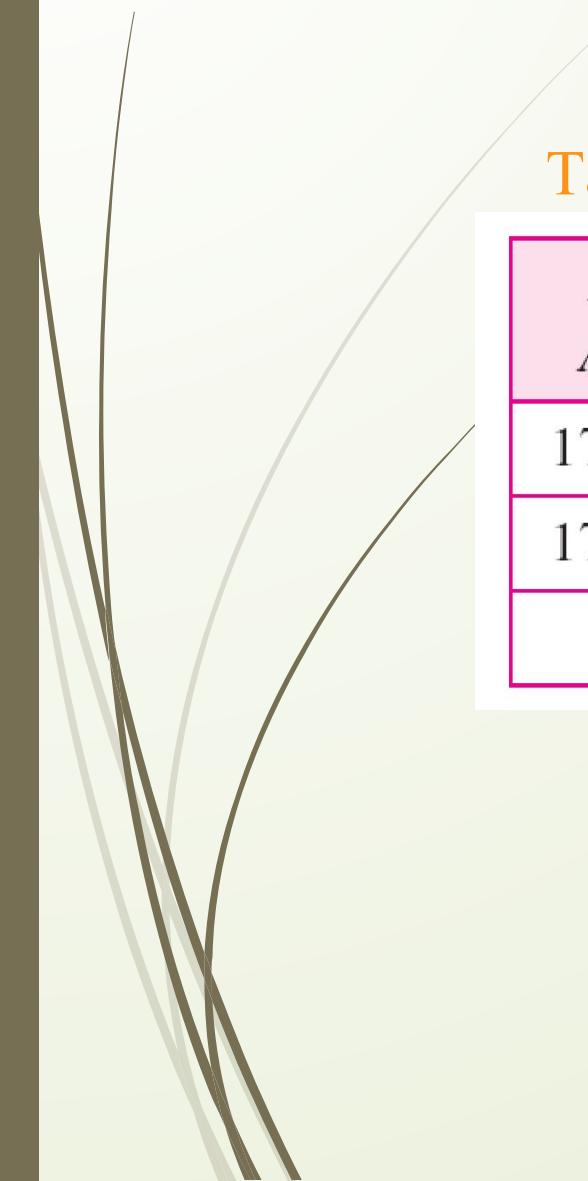
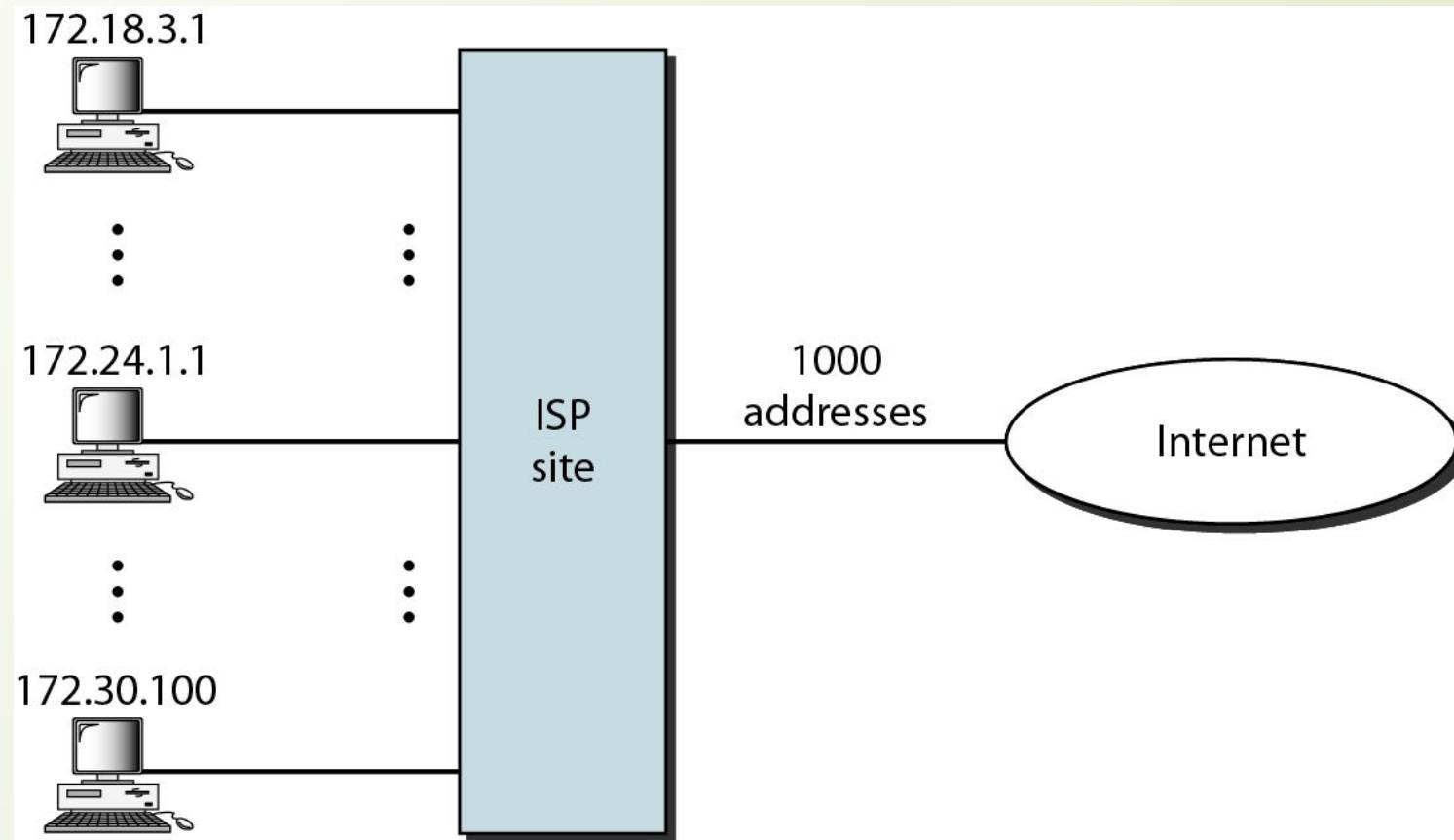


Table 9.4 *Five-column translation table*

<i>Private Address</i>	<i>Private Port</i>	<i>External Address</i>	<i>External Port</i>	<i>Transport Protocol</i>
172.18.3.1	1400	25.8.3.2	80	TCP
172.18.3.2	1401	25.8.3.2	80	TCP
...

Figure 9.13 An ISP and NAT



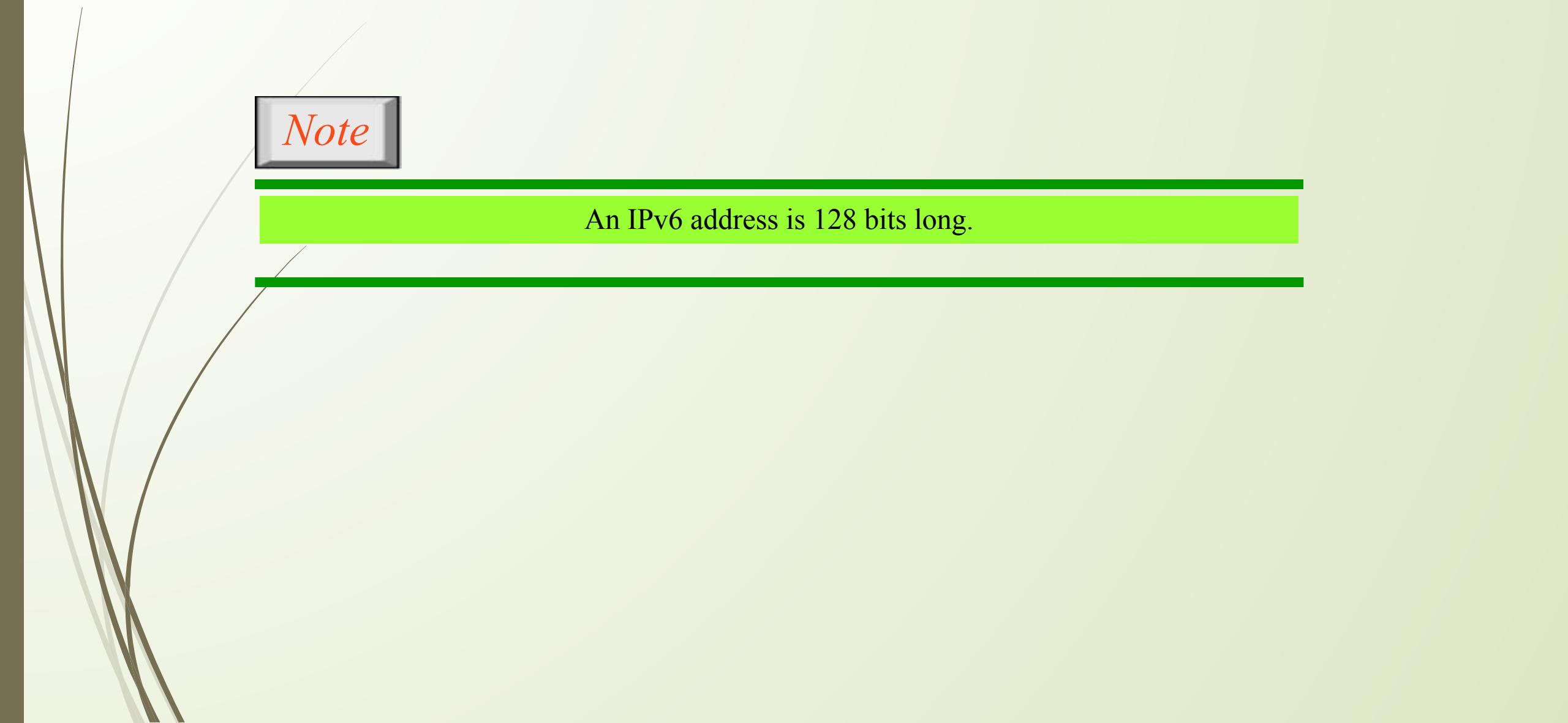
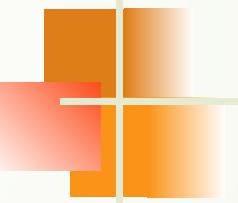
9-2 IPv6 ADDRESSES

Despite all short-term solutions, address depletion is still a long-term problem for the Internet. This and other problems in the IP protocol itself have been the motivation for IPv6.

Topics discussed in this section:

Structure

Address Space



Note

An IPv6 address is 128 bits long.

Figure 9.14 IPv6 address in binary and hexadecimal colon notation

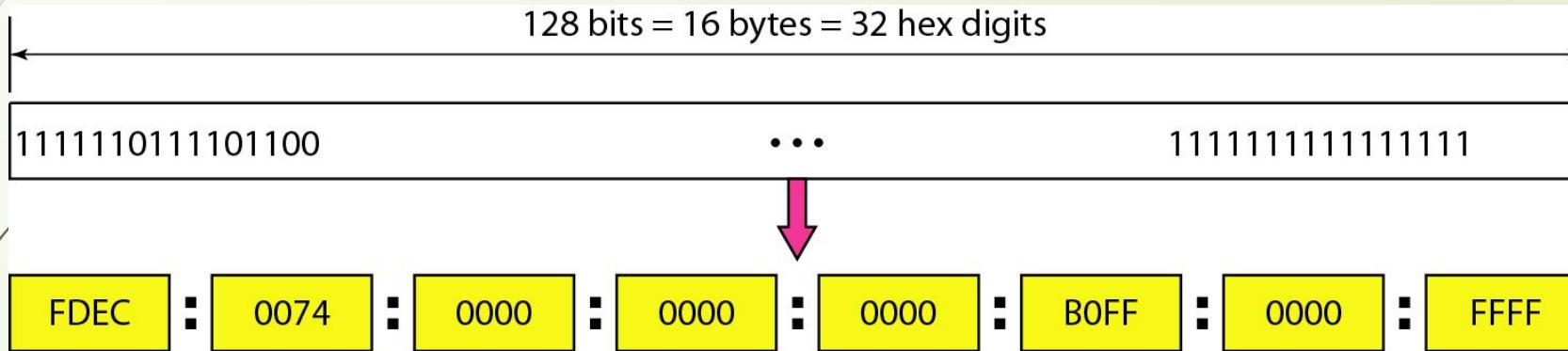
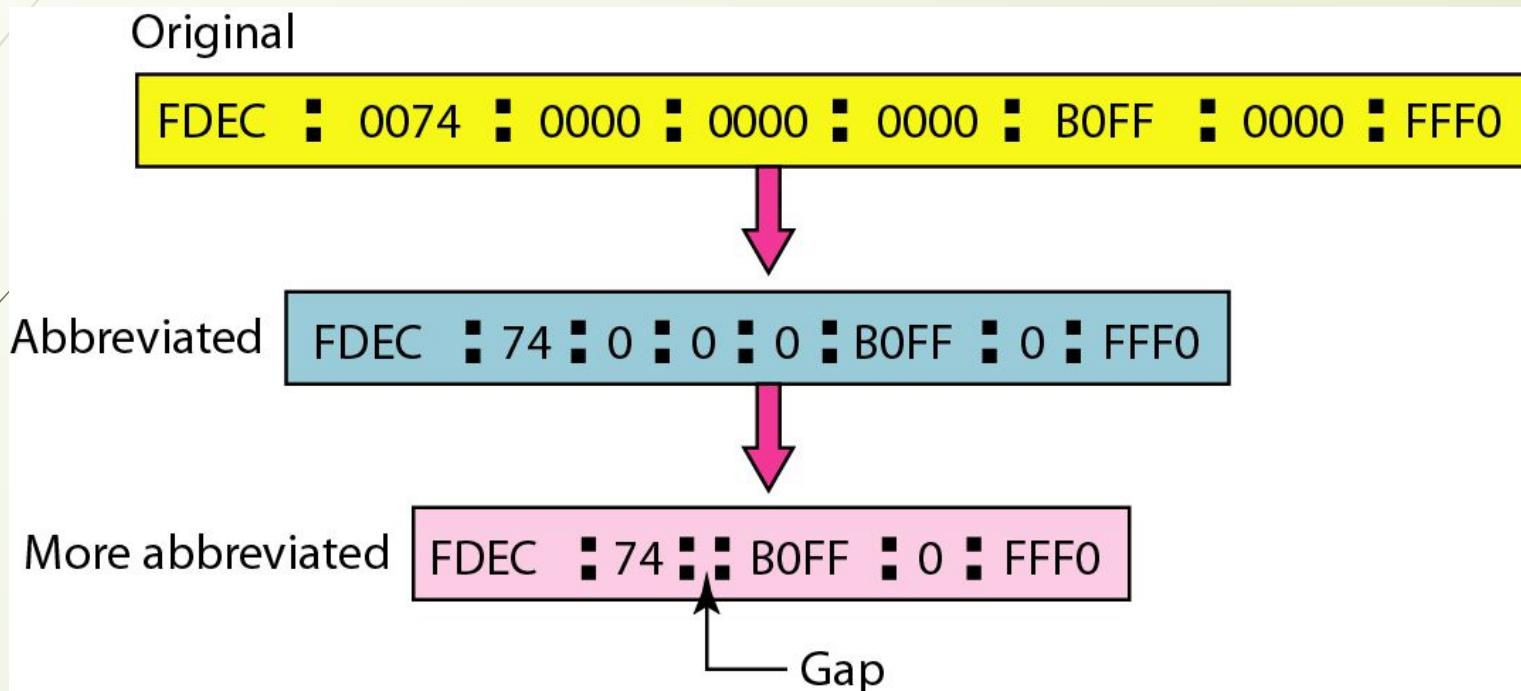
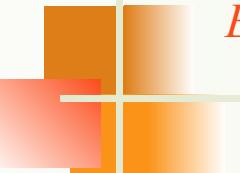


Figure 9.15 Abbreviated IPv6 addresses





Example 9.11

Expand the address $0:15::1:12:1213$ to its original.

Solution

We first need to align the left side of the double colon to the left of the original pattern and the right side of the double colon to the right of the original pattern to find how many 0s we need to replace the double colon.

XXXX:XXXX:XXXX:XXXX:XXXX:XXXX:XXXX:XXXX

0: 15: : 1: 12:1213

This means that the original address is.

0000:0015:0000:0000:0000:0001:0012:1213

Table 9.5 *Type prefixes for IPv6*

Type Prefix	Type	Fraction
0000 0000	Reserved	1/256
0000 0001	Unassigned	1/256
0000 001	ISO network addresses	1/128
0000 010	IPX (Novell) network addresses	1/128
0000 011	Unassigned	1/128
0000 1	Unassigned	1/32
0001	Reserved	1/16
001	Reserved	1/8
010	Provider-based unicast addresses	1/8

Table 9.5 *Type prefixes for IPv6 addresses (continued)*

Type Prefix	Type	Fraction
011	Unassigned	1/8
100	Geographic-based unicast addresses	1/8
101	Unassigned	1/8
110	Unassigned	1/8
1110	Unassigned	1/16
1111 0	Unassigned	1/32
1111 10	Unassigned	1/64
1111 110	Unassigned	1/128
1111 1110 0	Unassigned	1/512
1111 1110 10	Link local addresses	1/1024
1111 1110 11	Site local addresses	1/1024
1111 1111	Multicast addresses	1/256

Figure 9.16 Prefixes for provider-based unicast address

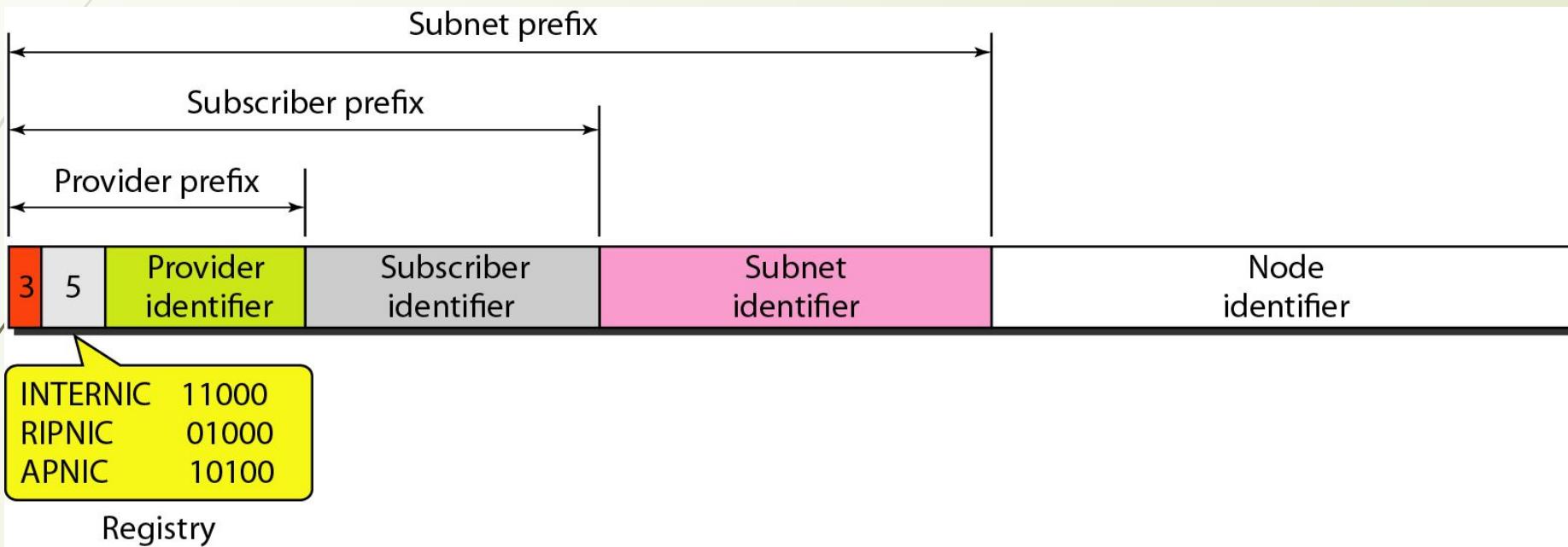


Figure 9.17 Multicast address in IPv6

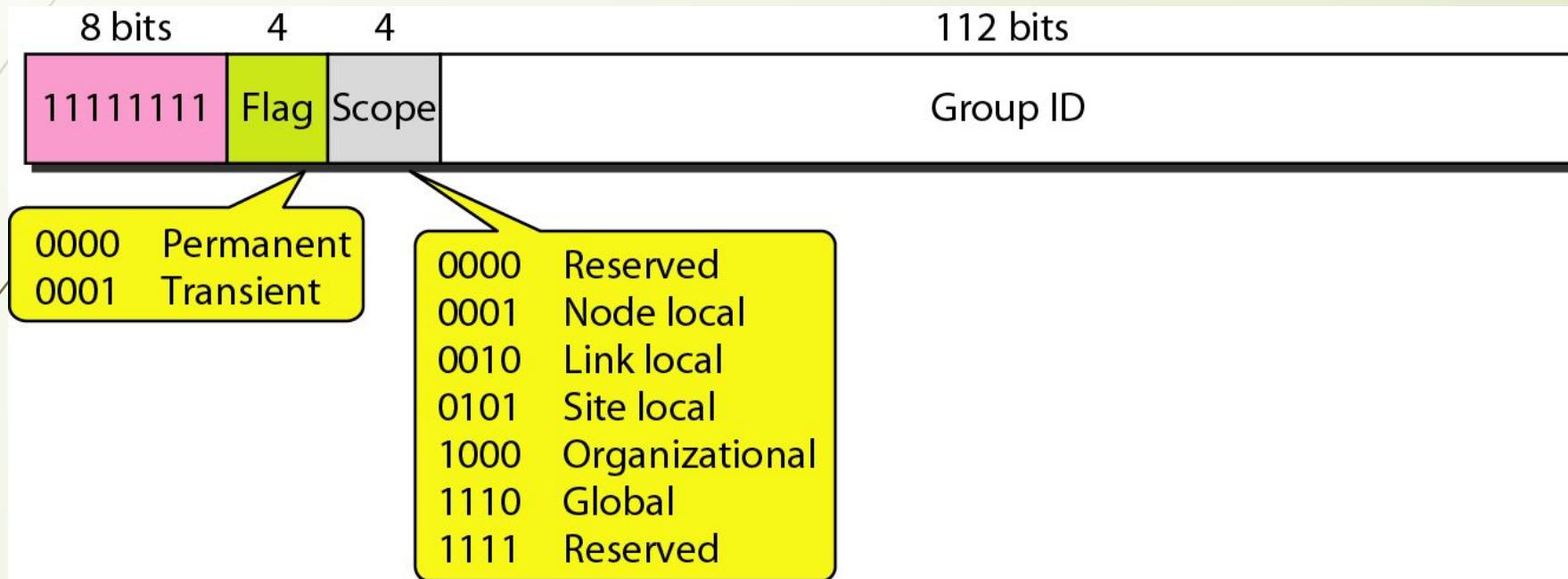


Figure 9.18 *Reserved addresses in IPv6*

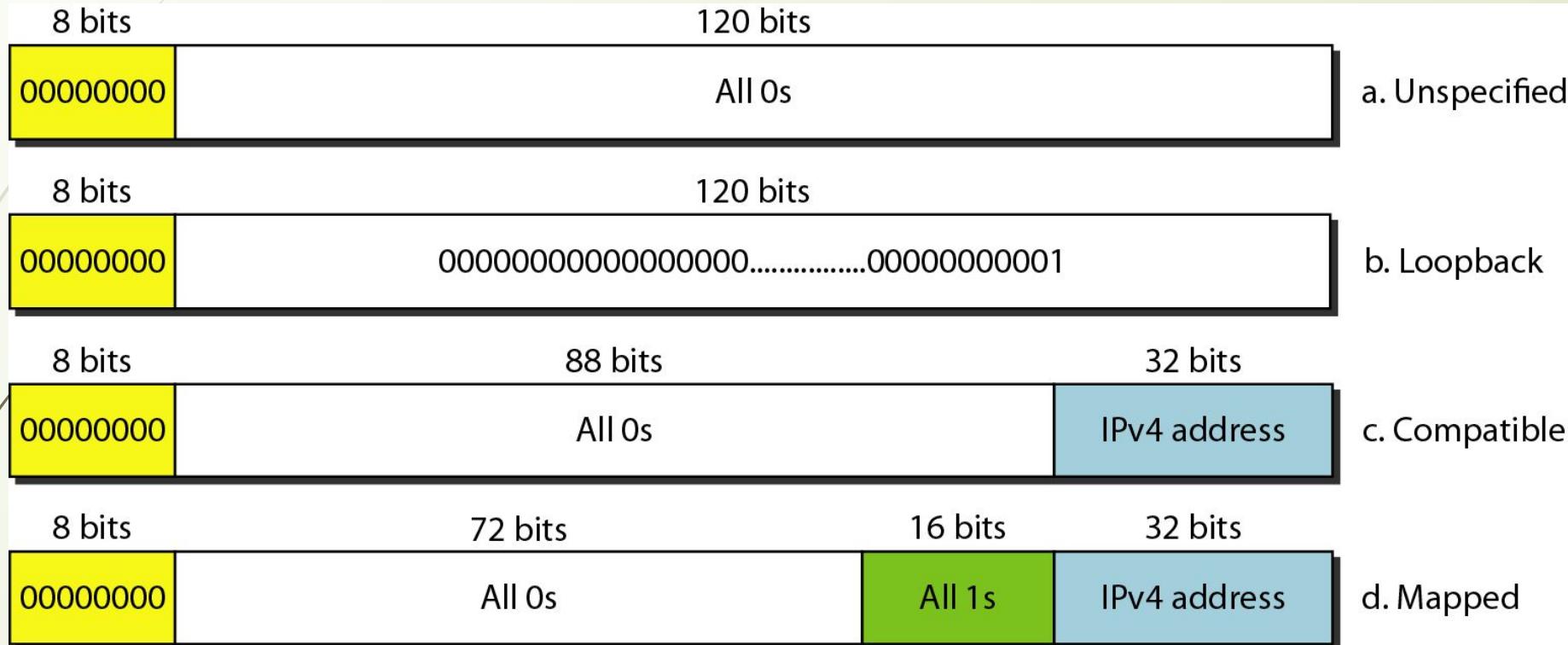
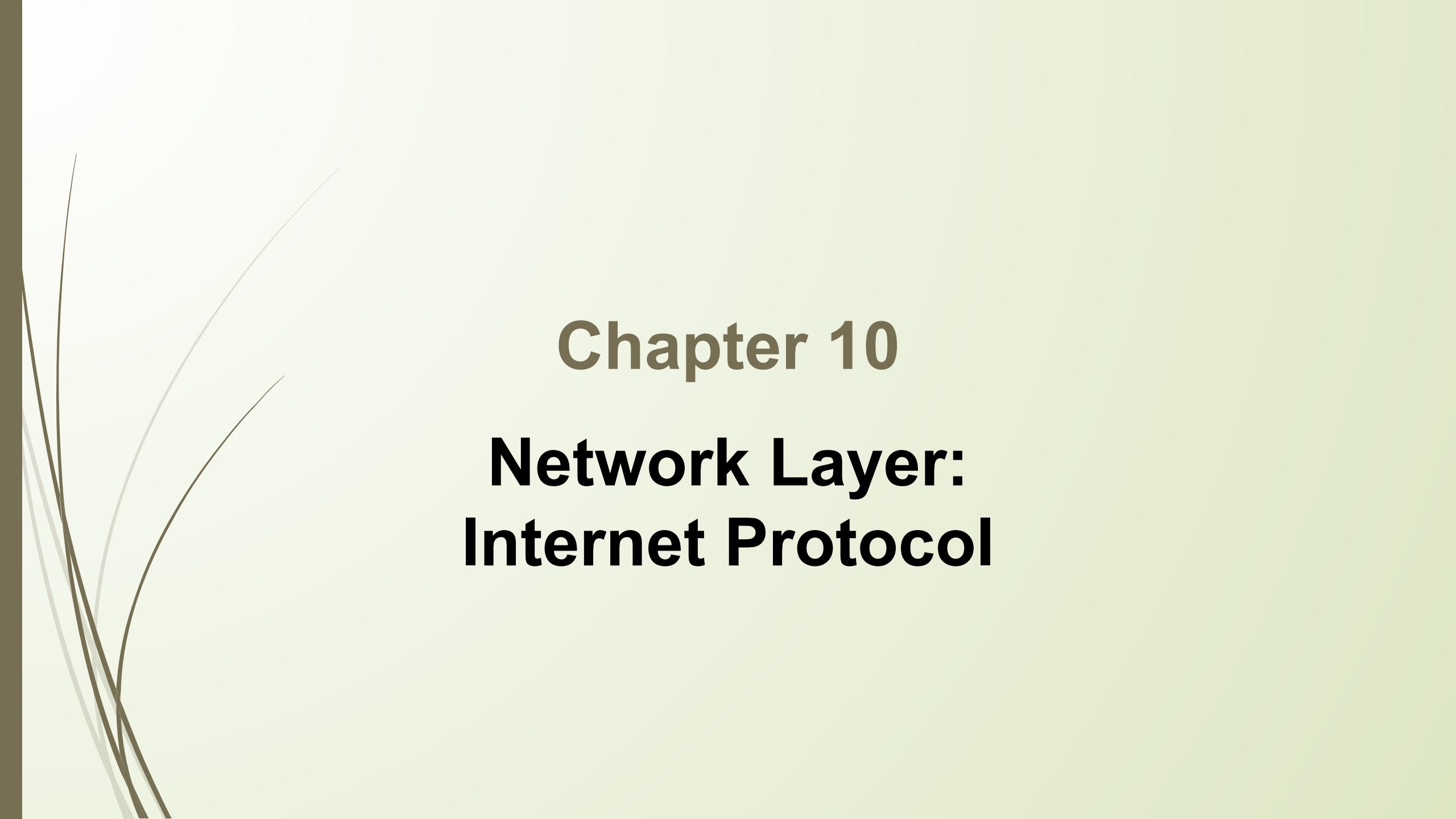


Figure 9.19 Local addresses in IPv6





Chapter 10

Network Layer: Internet Protocol

10-1 INTERNETWORKING

In this section, we discuss internetworking, connecting networks together to make an internetwork or an internet.

Topics discussed in this section:

Need for Network Layer

Internet as a Datagram Network

Internet as a Connectionless Network

Figure 10.1 *Links between two hosts*

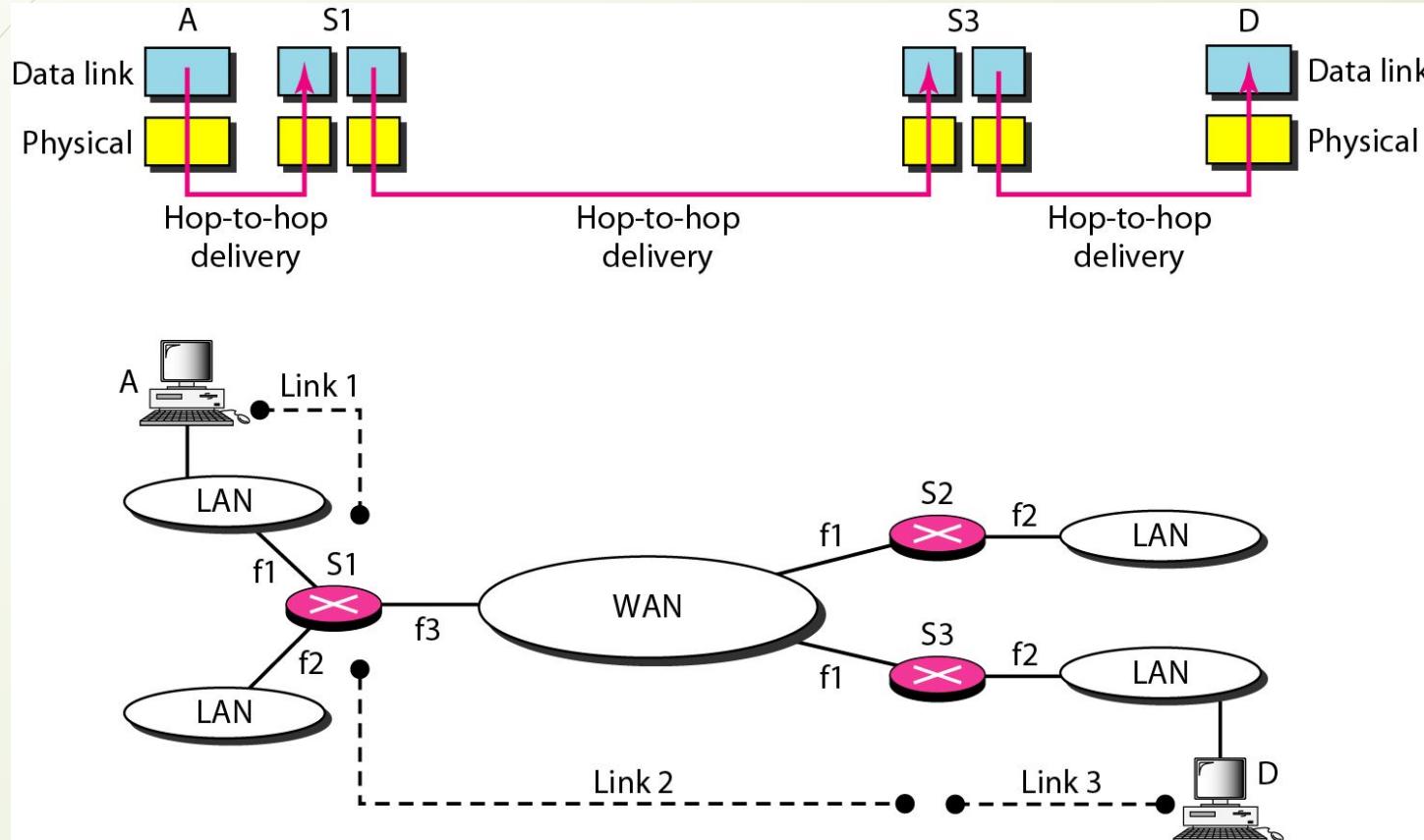


Figure 10.2 Network layer in an internetwork

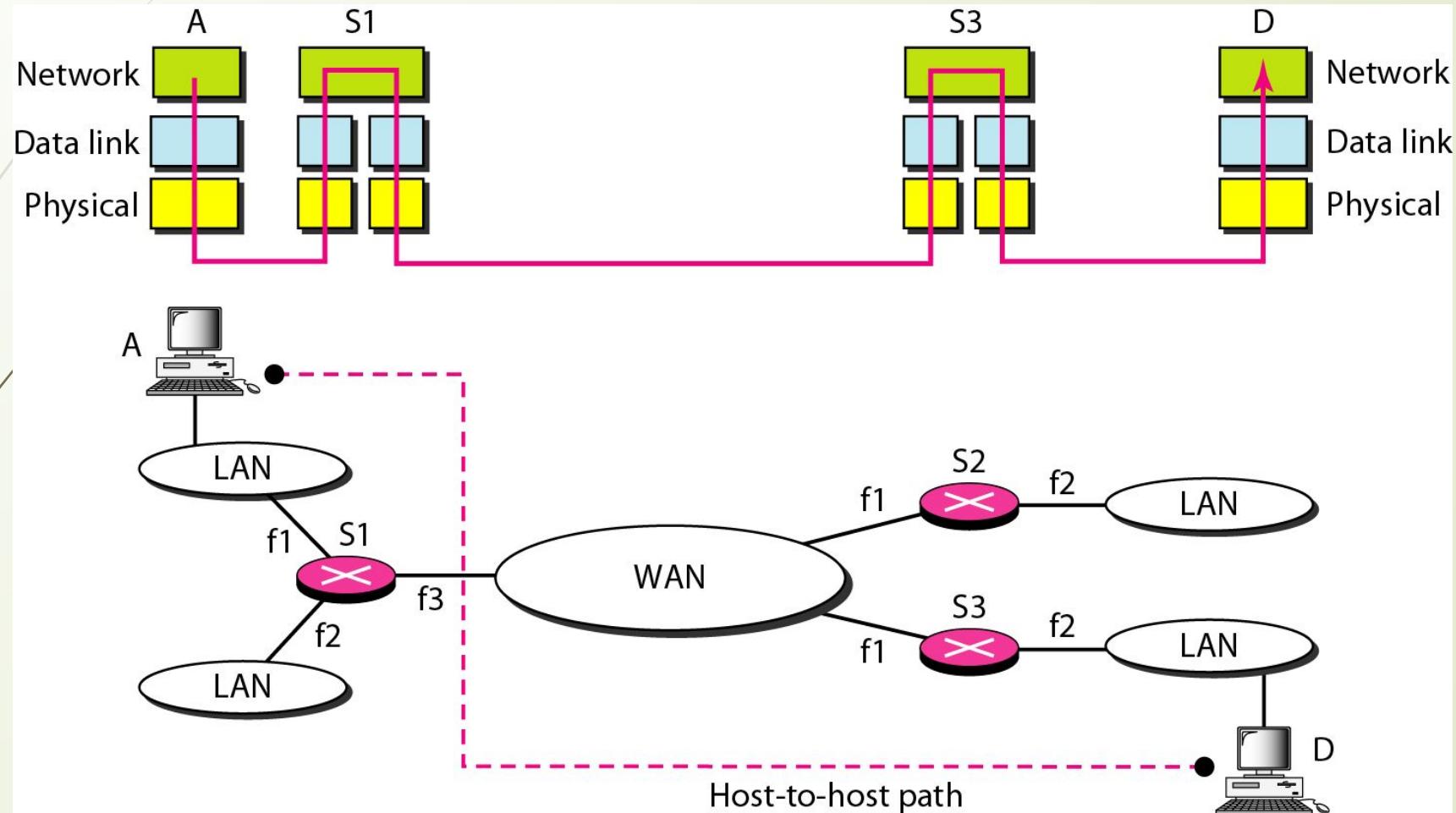


Figure 10.3 Network layer at the source, router, and destination

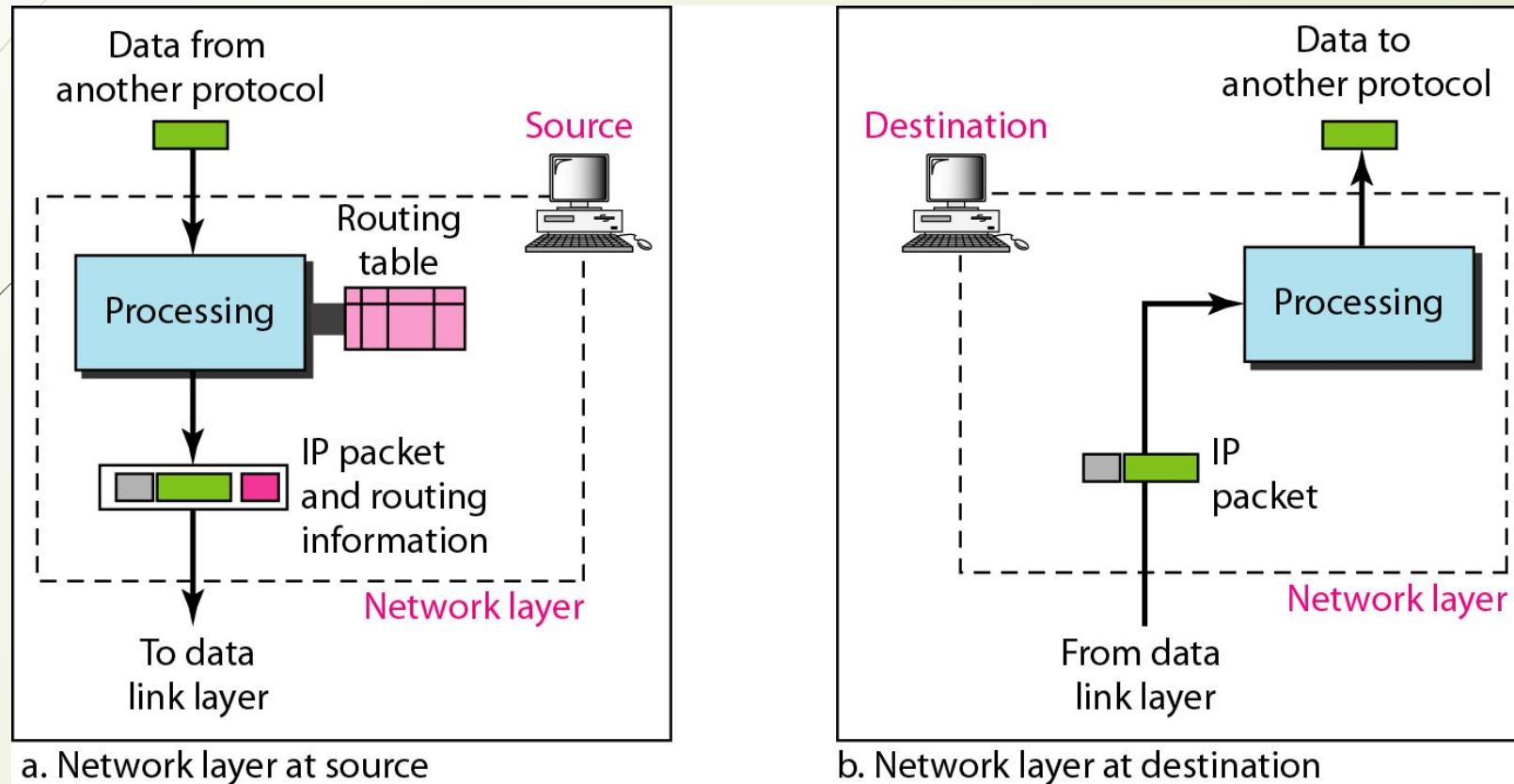
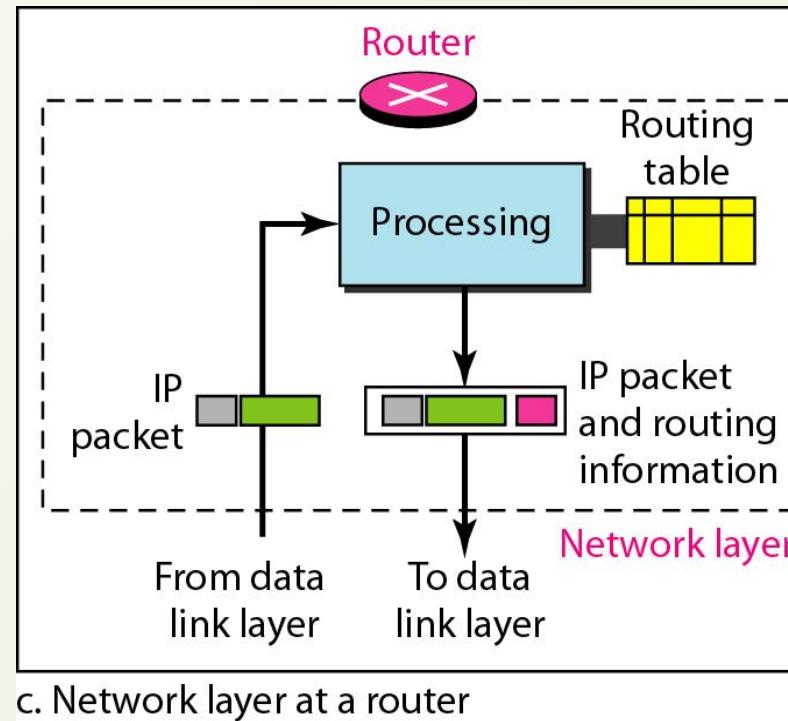


Figure 10.3 Network layer at the source, router, and destination (continued)





Note

Switching at the network layer in the Internet uses the datagram approach to packet switching.



Note

Communication at the network layer in the Internet is connectionless.

*The Internet Protocol version 4 (**IPv4**) is the delivery mechanism used by the TCP/IP protocols.*

Topics discussed in this section:

Datagram

Fragmentation

Checksum

Options

Figure 10.4 Position of IPv4 in TCP/IP protocol suite

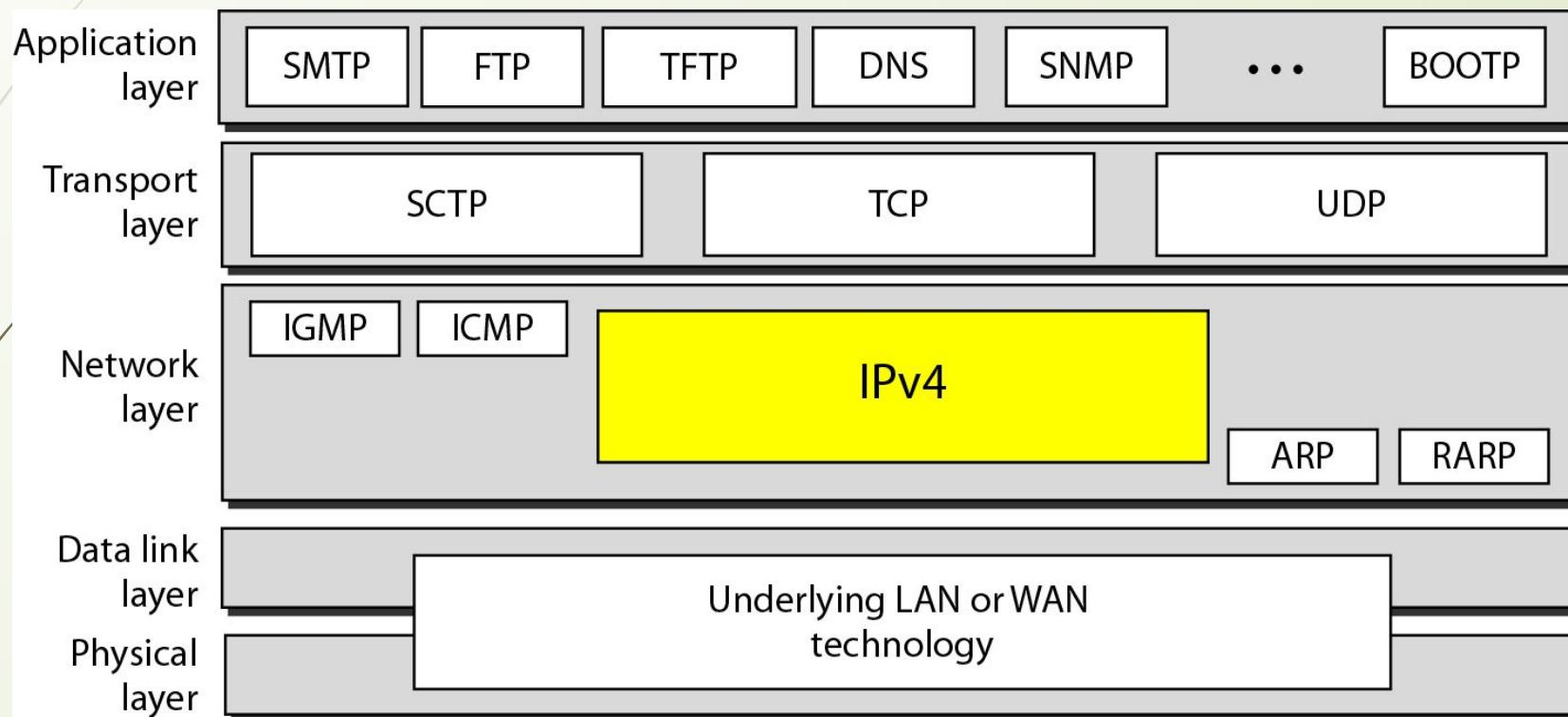


Figure 10.5 IPv4 datagram format

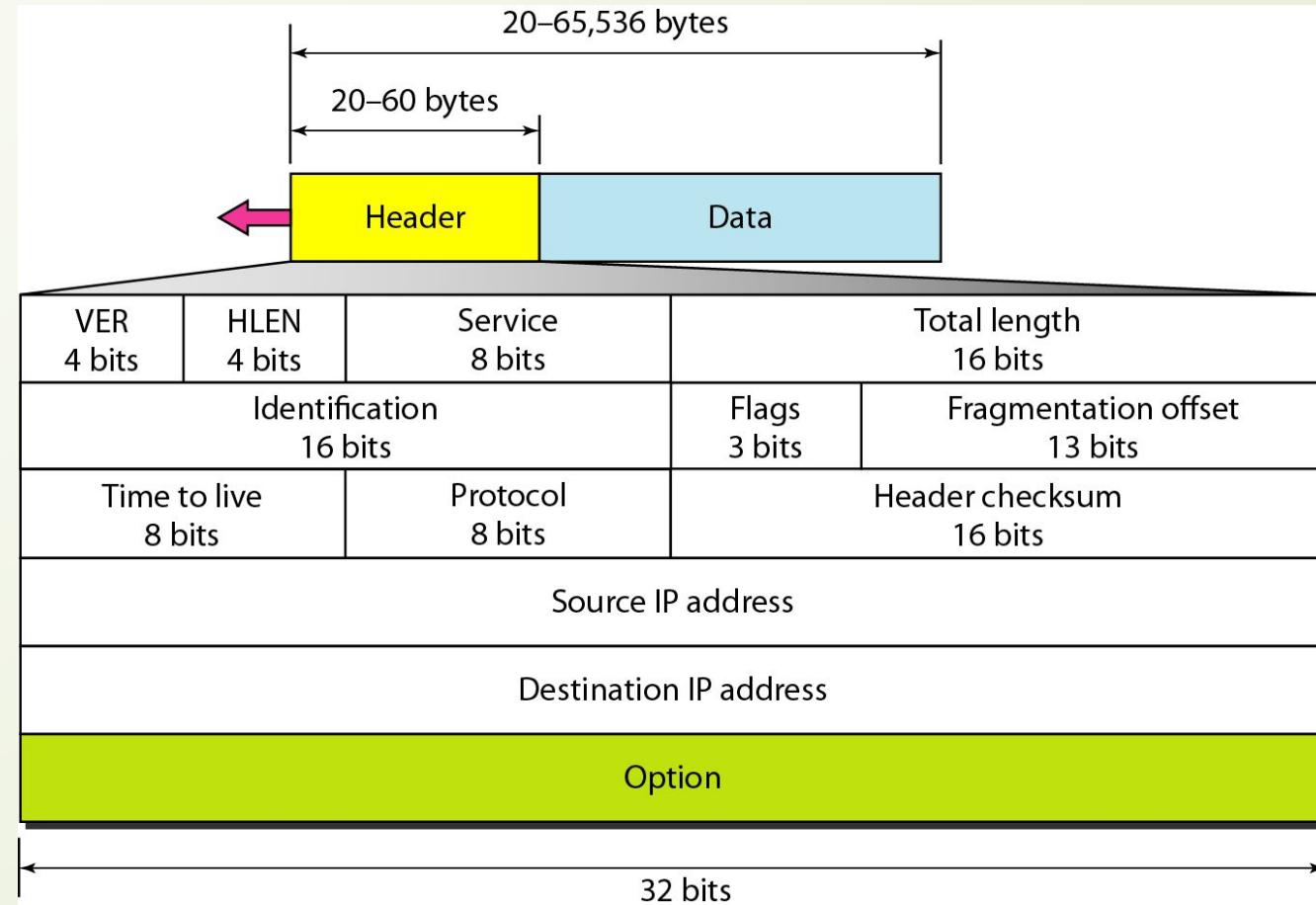
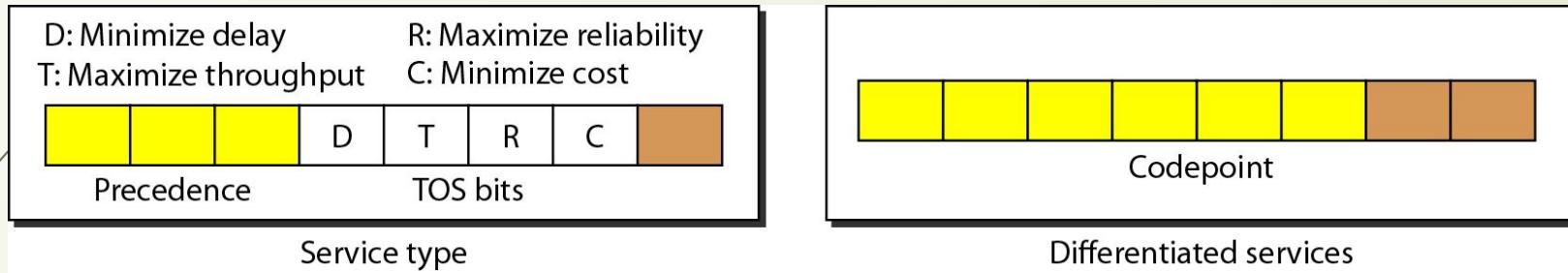
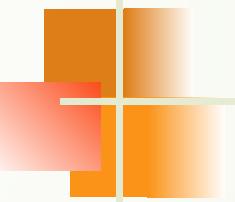


Figure 10.6 *Service type or differentiated services*





Note

The precedence subfield was part of version 4, but never used.

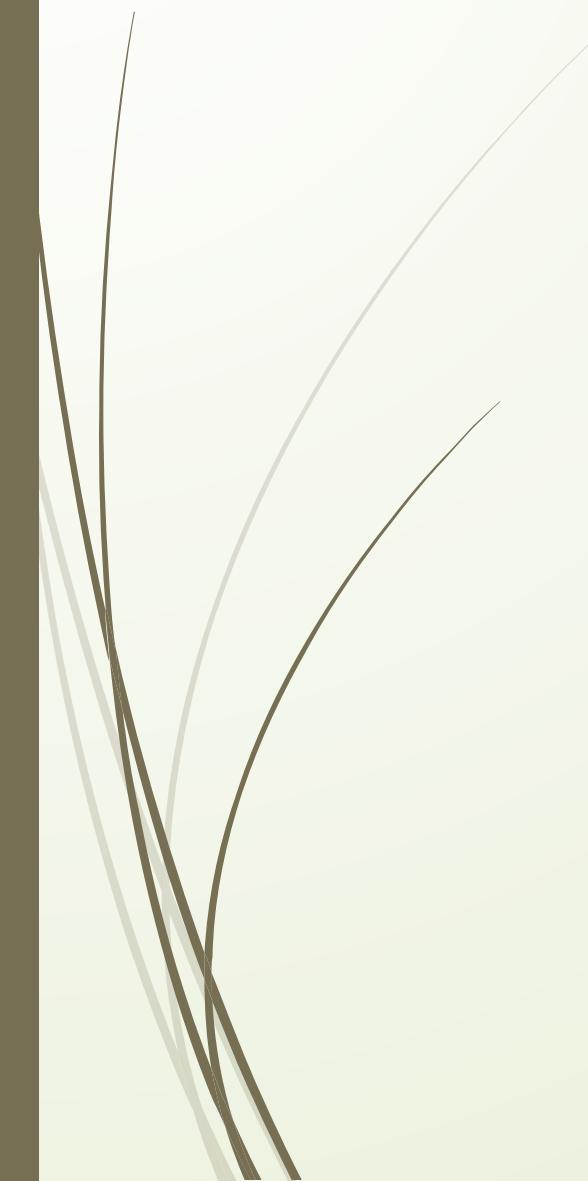


Table 10.1 *Types of service*

<i>TOS Bits</i>	<i>Description</i>
0000	Normal (default)
0001	Minimize cost
0010	Maximize reliability
0100	Maximize throughput
1000	Minimize delay

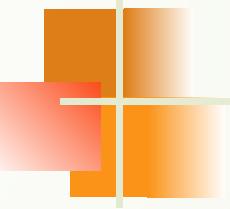
Table 10.2 *Default types of service*

<i>Protocol</i>	<i>TOS Bits</i>	<i>Description</i>
ICMP	0000	Normal
BOOTP	0000	Normal
NNTP	0001	Minimize cost
IGP	0010	Maximize reliability
SNMP	0010	Maximize reliability
TELNET	1000	Minimize delay
FTP (data)	0100	Maximize throughput
FTP (control)	1000	Minimize delay
TFTP	1000	Minimize delay
SMTP (command)	1000	Minimize delay
SMTP (data)	0100	Maximize throughput
DNS (UDP query)	1000	Minimize delay
DNS (TCP query)	0000	Normal
DNS (zone)	0100	Maximize throughput



Table 10.3 *Values for*

<i>Value</i>	<i>Protocol</i>
1	ICMP
2	IGMP
6	TCP
17	UDP
89	OSPF



Note

The total length field defines the total length of the datagram including the header.

Figure 10.7 *Encapsulation of a small datagram in an Ethernet frame*

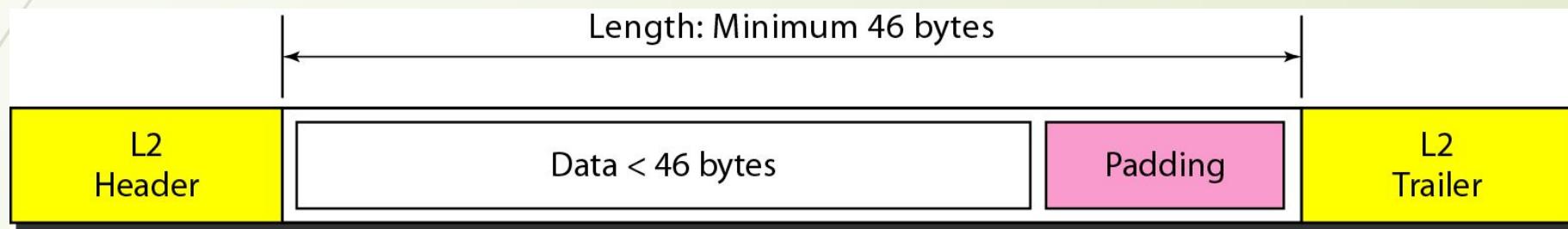
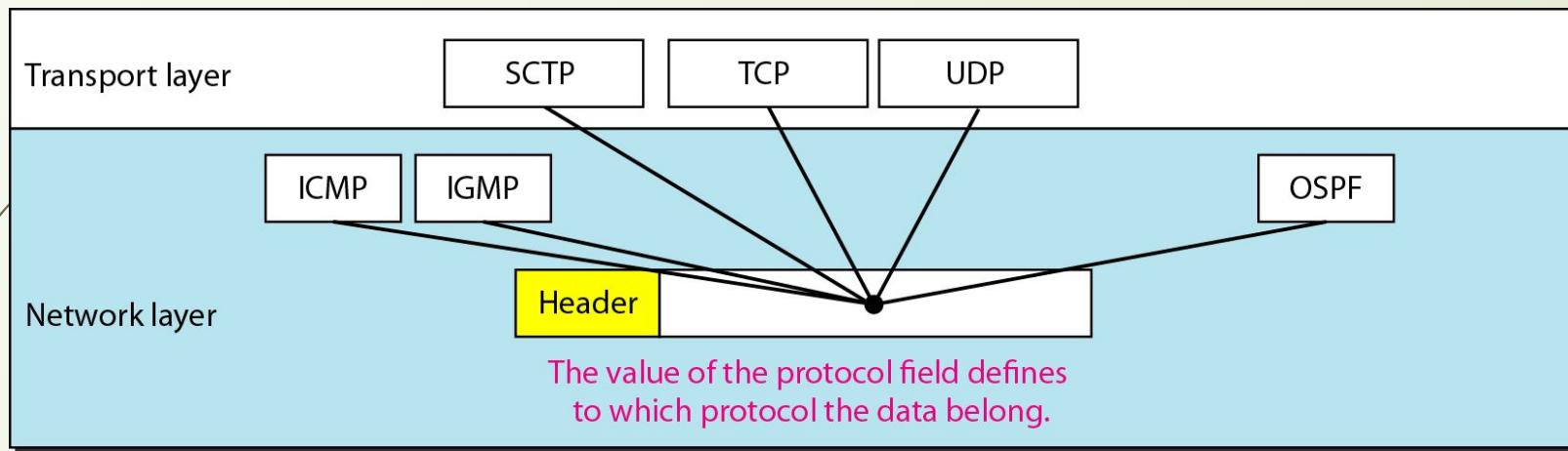


Figure 10.8 *Protocol field and encapsulated data*



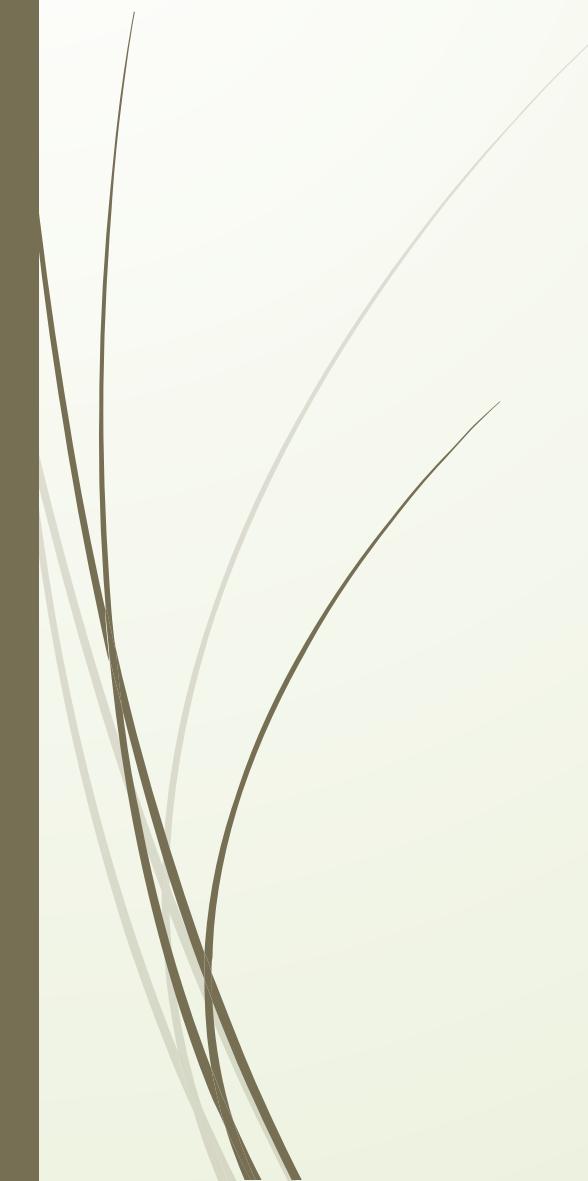
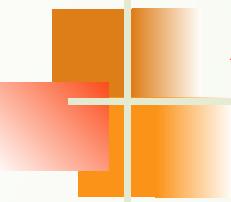


Table 10.4 *Protocol values*

<i>Value</i>	<i>Protocol</i>
1	ICMP
2	IGMP
6	TCP
17	UDP
89	OSPF



Example 10.1

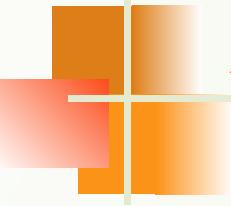
An IPv4 packet has arrived with the first 8 bits as shown:

01000010

The receiver discards the packet. Why?

Solution

There is an error in this packet. The 4 leftmost bits (0100) show the version, which is correct. The next 4 bits (0010) show an invalid header length ($2 \times 4 = 8$). The minimum number of bytes in the header must be 20. The packet has been corrupted in transmission.

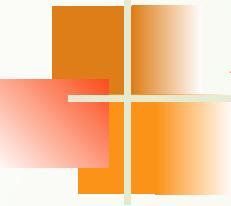


Example 10.2

In an IPv4 packet, the value of HLEN is 1000 in binary. How many bytes of options are being carried by this packet?

Solution

The HLEN value is 8, which means the total number of bytes in the header is 8×4 , or 32 bytes. The first 20 bytes are the base header, the next 12 bytes are the options.

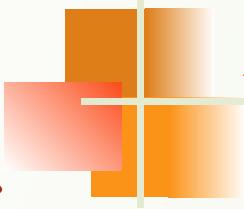


Example 10.3

In an IPv4 packet, the value of HLEN is 5, and the value of the total length field is 0x0028. How many bytes of data are being carried by this packet?

Solution

The HLEN value is 5, which means the total number of bytes in the header is 5×4 , or 20 bytes (no options). The total length is 40 bytes, which means the packet is carrying 20 bytes of data ($40 - 20$).



Example 10.4

An IPv4 packet has arrived with the first few hexadecimal digits as shown.

0x45000028000100000102...

How many hops can this packet travel before being dropped? The data belong to what upper-layer protocol?

Solution

To find the time-to-live field, we skip 8 bytes. The time-to-live field is the ninth byte, which is 01. This means the packet can travel only one hop. The protocol field is the next byte (02), which means that the upper-layer protocol is IGMP.

Figure 10.9 Maximum transfer unit (MTU)

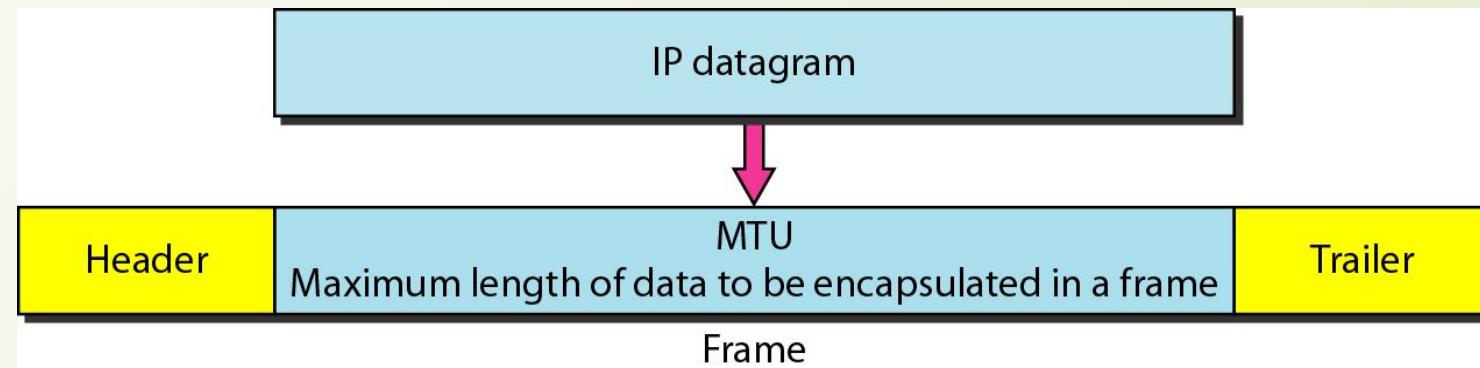


Table 10.5 *MTUs for some networks*

<i>Protocol</i>	<i>MTU</i>
Hyperchannel	65,535
Token Ring (16 Mbps)	17,914
Token Ring (4 Mbps)	4,464
FDDI	4,352
Ethernet	1,500
X.25	576
PPP	296

Figure 10.10 *Flags used in fragmentation*



Figure 10.11 *Fragmentation example*

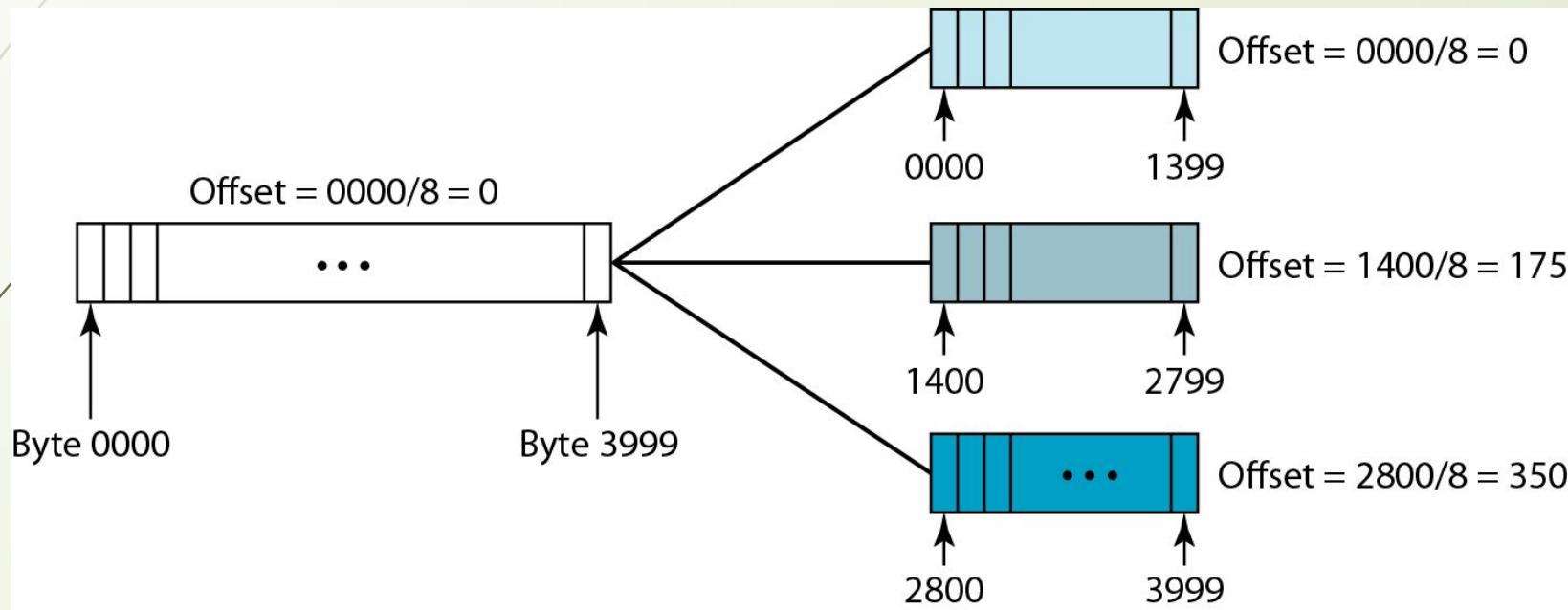
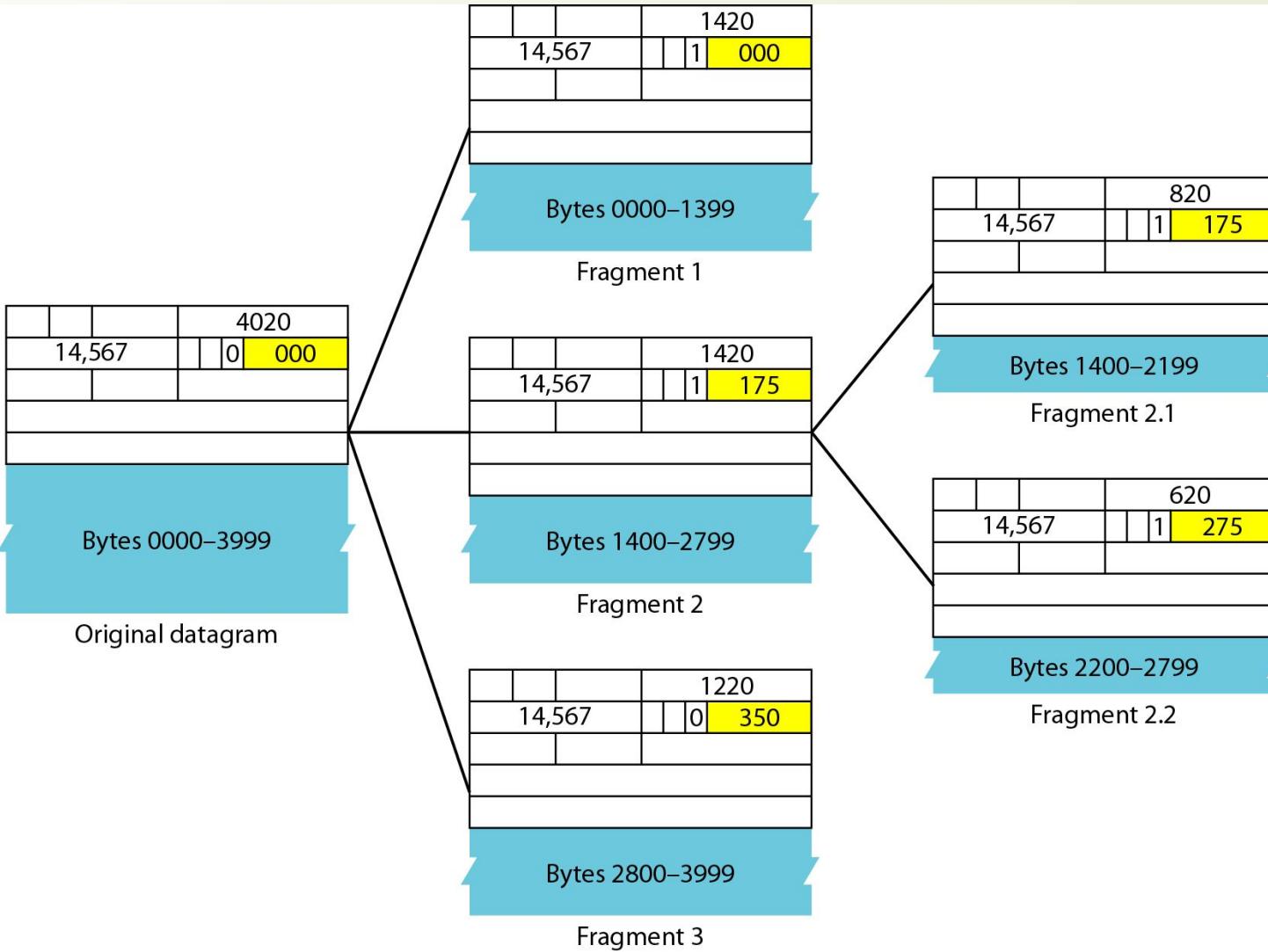
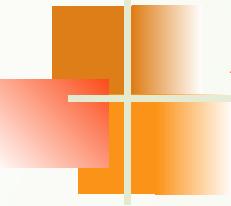


Figure 10.12 *Detailed fragmentation example*



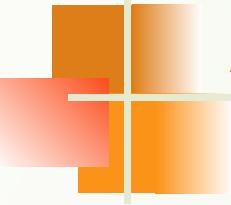


Example 10.5

A packet has arrived with an M bit value of 0. Is this the first fragment, the last fragment, or a middle fragment? Do we know if the packet was fragmented?

Solution

If the M bit is 0, it means that there are no more fragments; the fragment is the last one. However, we cannot say if the original packet was fragmented or not. A non-fragmented packet is considered the last fragment.

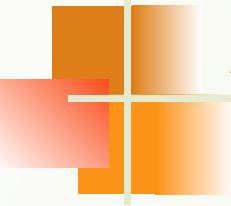


Example 10.6

A packet has arrived with an M bit value of 1. Is this the first fragment, the last fragment, or a middle fragment? Do we know if the packet was fragmented?

Solution

If the M bit is 1, it means that there is at least one more fragment. This fragment can be the first one or a middle one, but not the last one. We don't know if it is the first one or a middle one; we need more information (the value of the fragmentation offset).

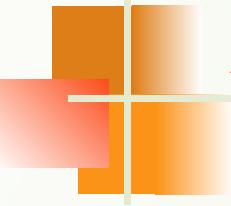


Example 10.7

A packet has arrived with an M bit value of 1 and a fragmentation offset value of 0. Is this the first fragment, the last fragment, or a middle fragment?

Solution

Because the M bit is 1, it is either the first fragment or a middle one. Because the offset value is 0, it is the first fragment.

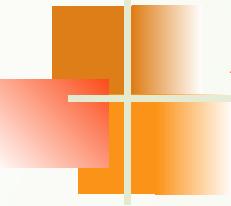


Example 10.8

A packet has arrived in which the offset value is 100. What is the number of the first byte? Do we know the number of the last byte?

Solution

To find the number of the first byte, we multiply the offset value by 8. This means that the first byte number is 800. We cannot determine the number of the last byte unless we know the length.

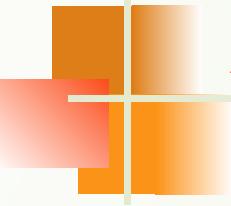


Example 10.9

A packet has arrived in which the offset value is 100, the value of HLEN is 5, and the value of the total length field is 100. What are the numbers of the first byte and the last byte?

Solution

The first byte number is $100 \times 8 = 800$. The total length is 100 bytes, and the header length is 20 bytes (5×4), which means that there are 80 bytes in this datagram. If the first byte number is 800, the last byte number must be 879.



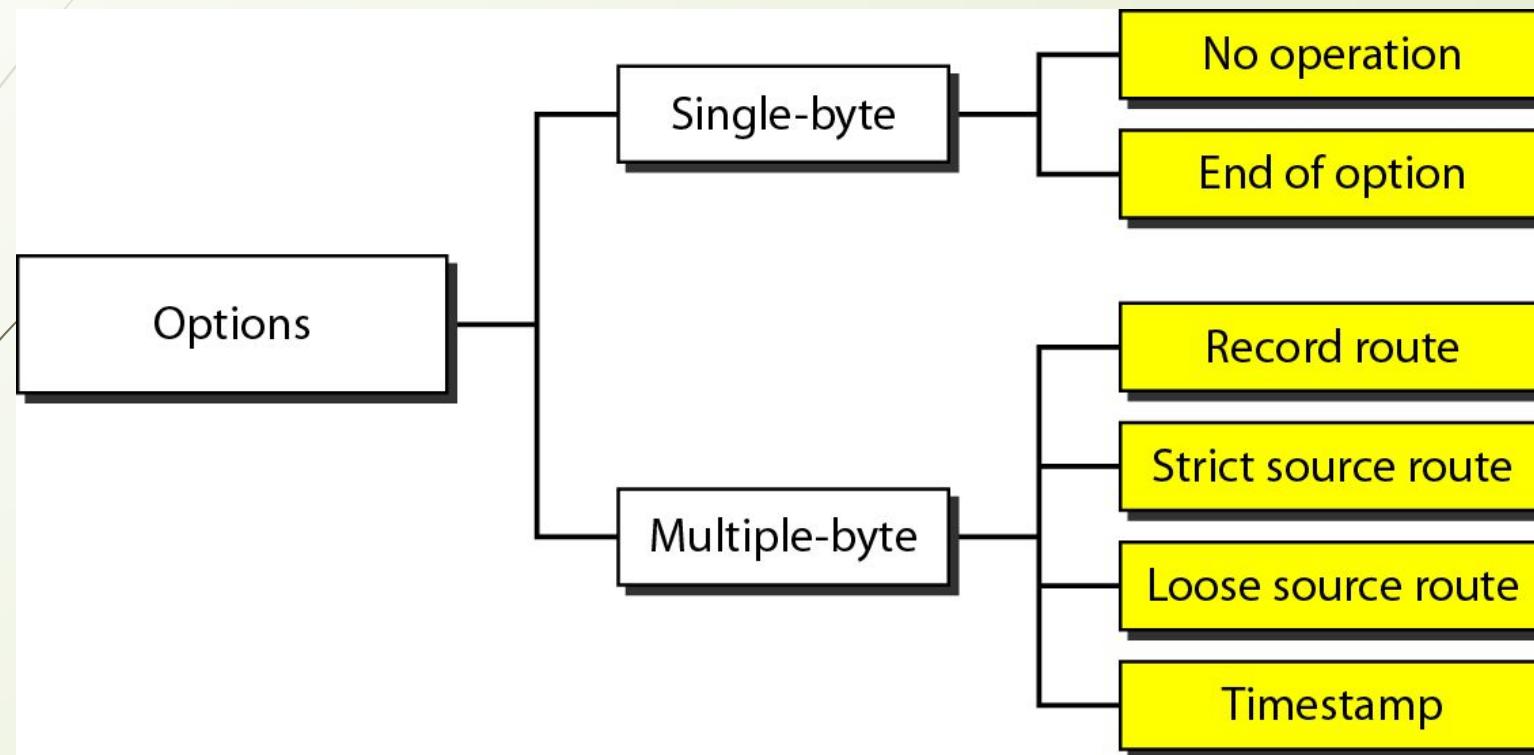
Example 10.10

Figure 10.13 shows an example of a checksum calculation for an IPv4 header without options. The header is divided into 16-bit sections. All the sections are added and the sum is complemented. The result is inserted in the checksum field.

Figure 10.13 Example of checksum calculation in IPv4

4	5	0	28			
1		0	0			
4	17	0		↑		
10.12.14.5						
12.6.7.9						
4, 5, and 0	→	4	5	0		
28	→	0	0	1 C		
1	→	0	0	0 1		
0 and 0	→	0	0	0 0		
4 and 17	→	0	4	1 1		
0	→	0	0	0 0		
10.12	→	0	A	0 C		
14.5	→	0	E	0 5		
12.6	→	0	C	0 6		
7.9	→	0	7	0 9		
Sum	→	7	4	4 E		
Checksum	→	8	B	B 1		

Figure 10.14 *Taxonomy of options in IPv4*



The network layer protocol in the TCP/IP protocol suite is currently IPv4. Although IPv4 is well designed, data communication has evolved since the inception of IPv4 in the 1970s. IPv4 has some deficiencies that make it unsuitable for the fast-growing Internet.

Topics discussed in this section:

Advantages

Packet Format

Extension Headers

Figure 10.15 IPv6 datagram header and payload

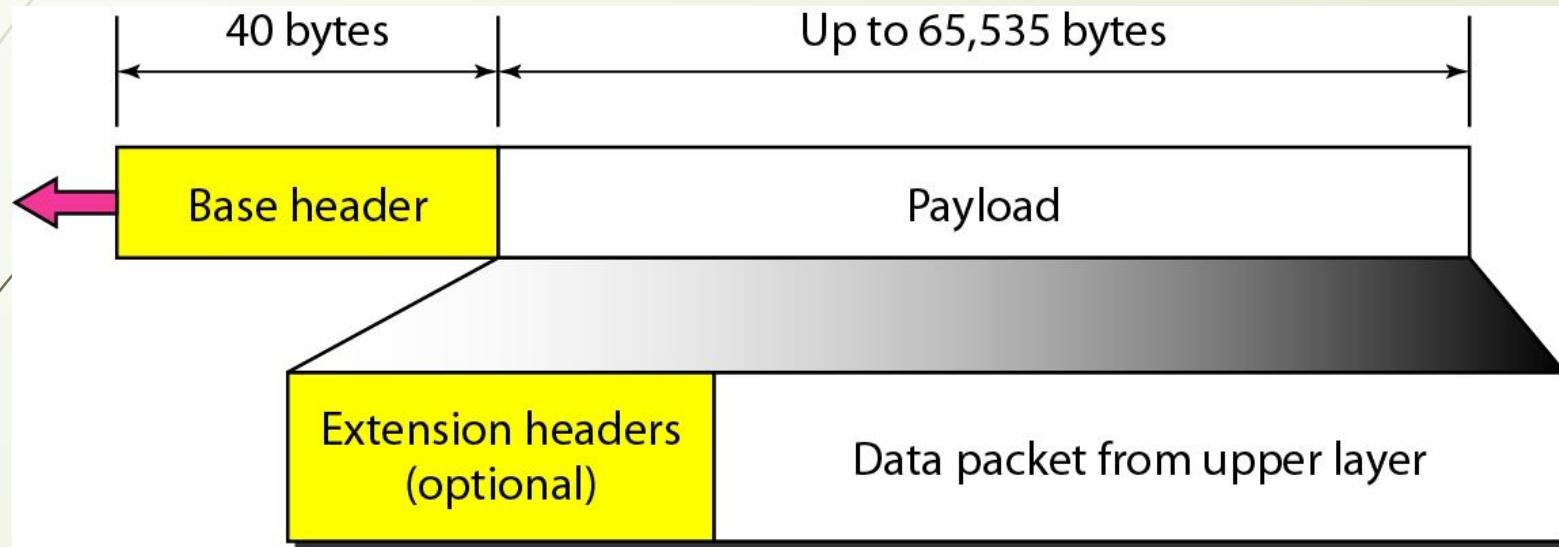


Figure 10.16 Format of an IPv6 datagram

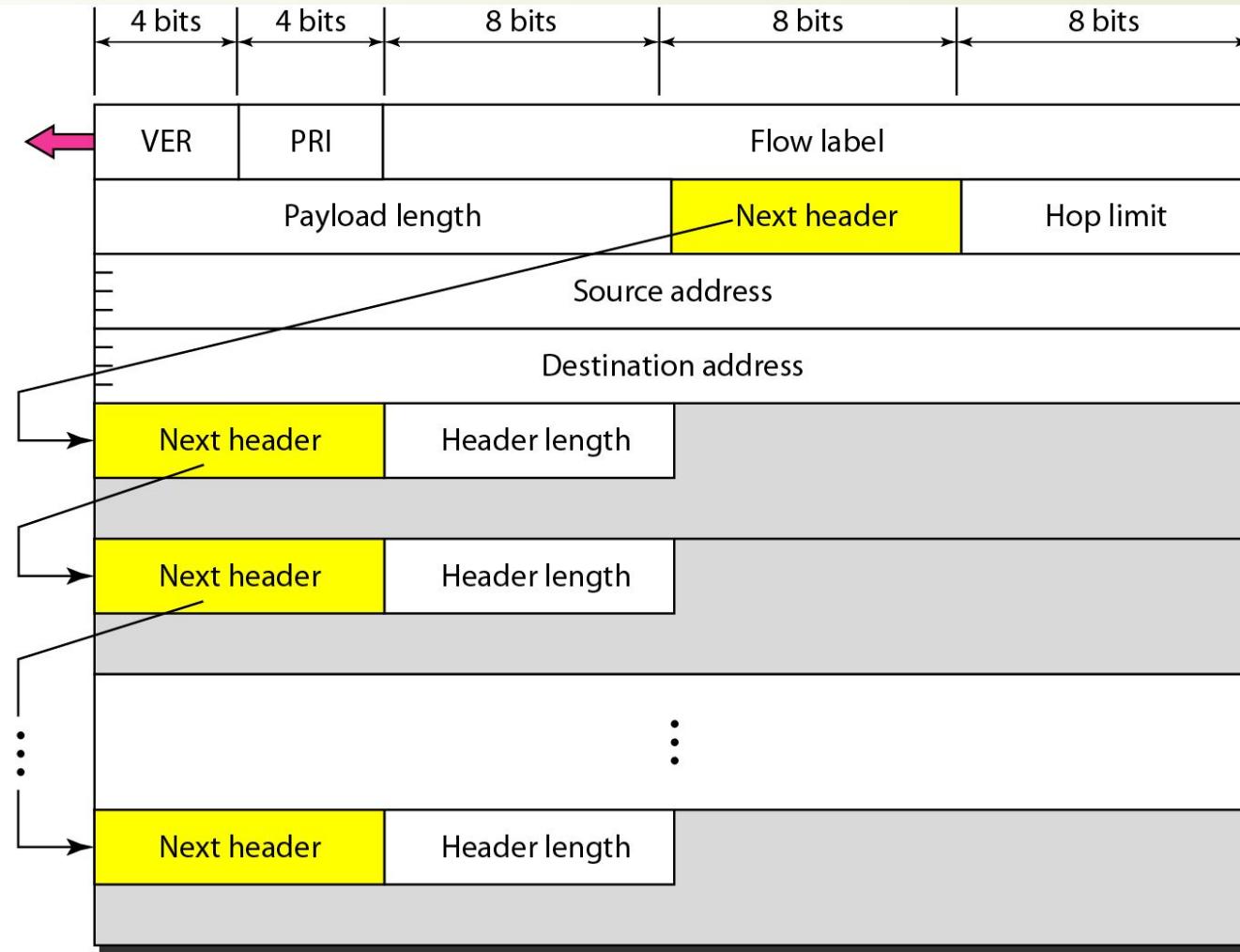


Table 10.6 *Next header codes for IPv6*

<i>Code</i>	<i>Next Header</i>
0	Hop-by-hop option
2	ICMP
6	TCP
17	UDP
43	Source routing
44	Fragmentation
50	Encrypted security payload
51	Authentication
59	Null (no next header)
60	Destination option

Table 10.7 *Priorities for congestion-controlled traffic*

<i>Priority</i>	<i>Meaning</i>
0	No specific traffic
1	Background data
2	Unattended data traffic
3	Reserved
4	Attended bulk data traffic
5	Reserved
6	Interactive traffic
7	Control traffic

Table 10.8 *Priorities for noncongestion-controlled traffic*

<i>Priority</i>	<i>Meaning</i>
8	Data with greatest redundancy
...	...
15	Data with least redundancy

Table 10.9 *Comparison between IPv4 and IPv6 packet headers*

Comparison
1. The header length field is eliminated in IPv6 because the length of the header is fixed in this version.
2. The service type field is eliminated in IPv6. The priority and flow label fields together take over the function of the service type field.
3. The total length field is eliminated in IPv6 and replaced by the payload length field.
4. The identification, flag, and offset fields are eliminated from the base header in IPv6. They are included in the fragmentation extension header.
5. The TTL field is called hop limit in IPv6.
6. The protocol field is replaced by the next header field.
7. The header checksum is eliminated because the checksum is provided by upper-layer protocols; it is therefore not needed at this level.
8. The option fields in IPv4 are implemented as extension headers in IPv6.

Figure 10.17 *Extension header types*

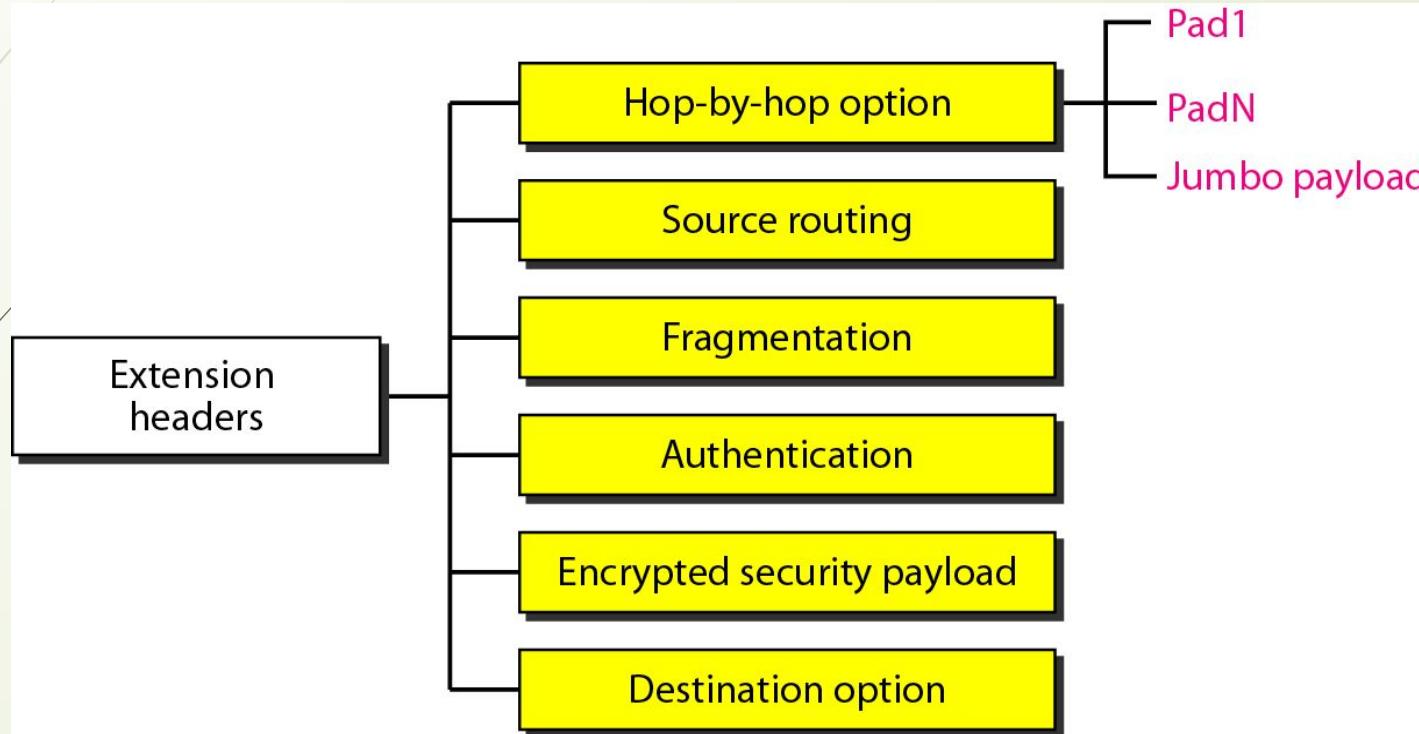


Table 10.10 *Comparison between IPv4 options and IPv6 extension headers*

<i>Comparison</i>
1. The no-operation and end-of-option options in IPv4 are replaced by Pad1 and PadN options in IPv6.
2. The record route option is not implemented in IPv6 because it was not used.
3. The timestamp option is not implemented because it was not used.
4. The source route option is called the source route extension header in IPv6.
5. The fragmentation fields in the base header section of IPv4 have moved to the fragmentation extension header in IPv6.
6. The authentication extension header is new in IPv6.
7. The encrypted security payload extension header is new in IPv6.

10-4 TRANSITION FROM IPv4 TO IPv6

Because of the huge number of systems on the Internet, the transition from IPv4 to IPv6 cannot happen suddenly. It takes a considerable amount of time before every system in the Internet can move from IPv4 to IPv6. The transition must be smooth to prevent any problems between IPv4 and IPv6 systems.

Topics discussed in this section:

Dual Stack

Tunneling

Header Translation

Figure 10.18 *Three transition strategies*

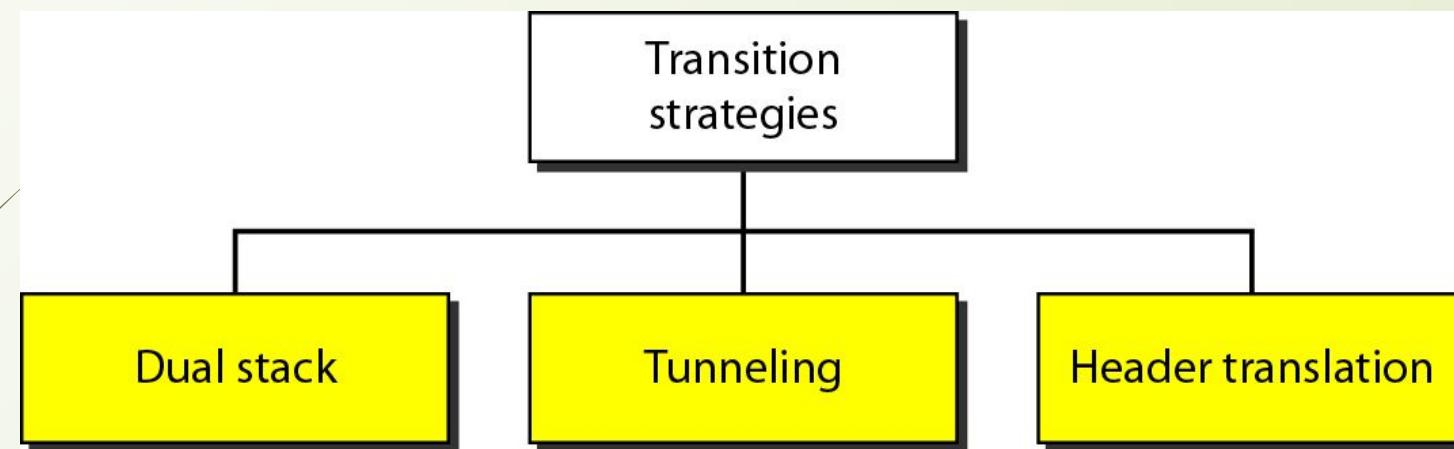


Figure 10.19 Dual stack

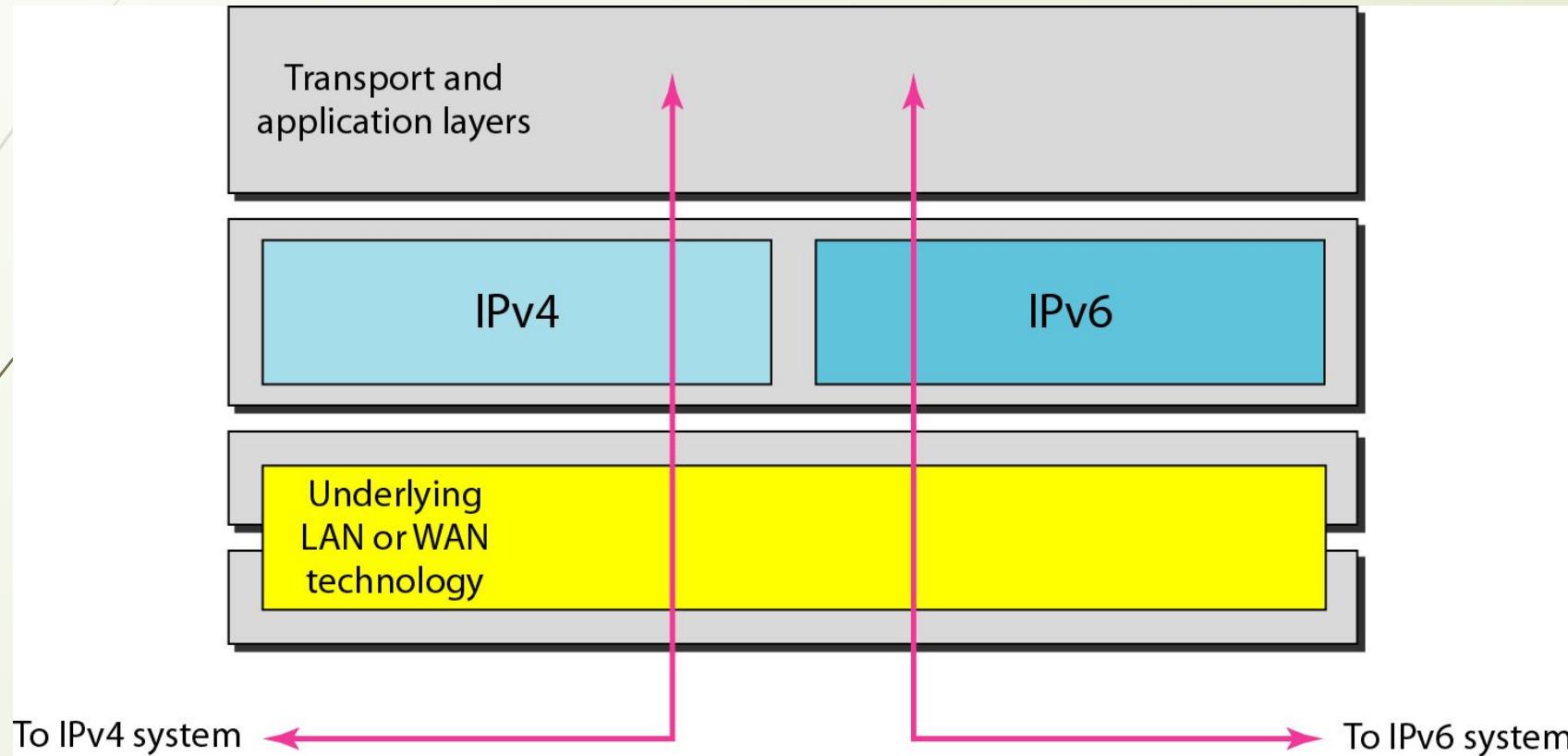


Figure 10.20 Tunneling strategy

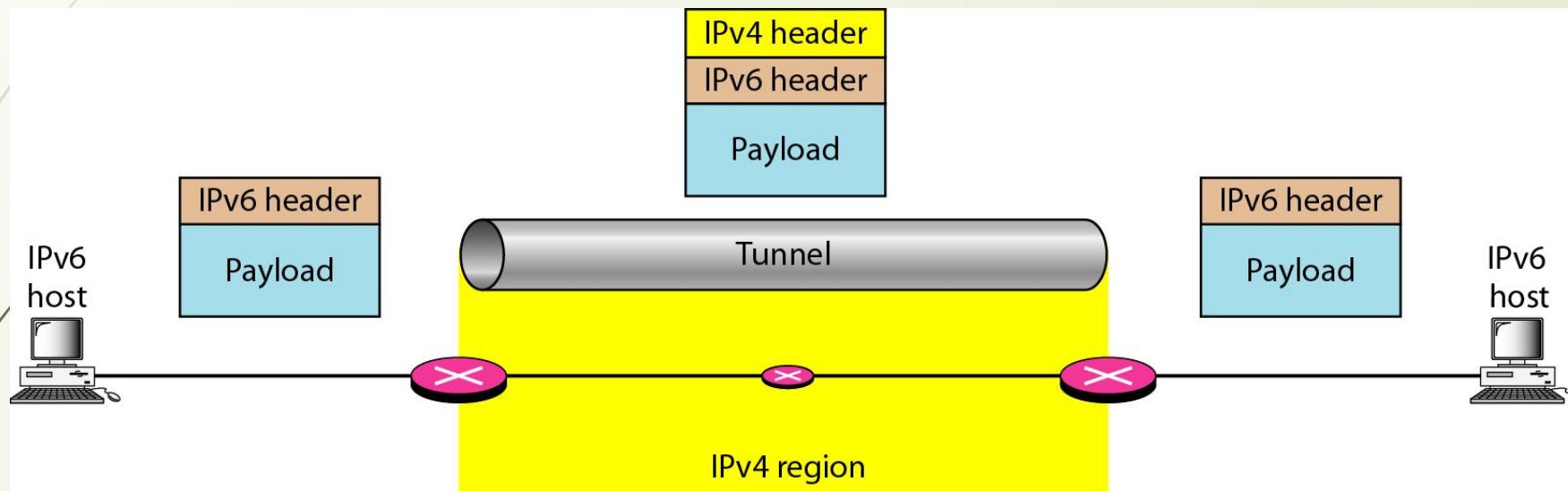


Figure 10.21 Header translation strategy

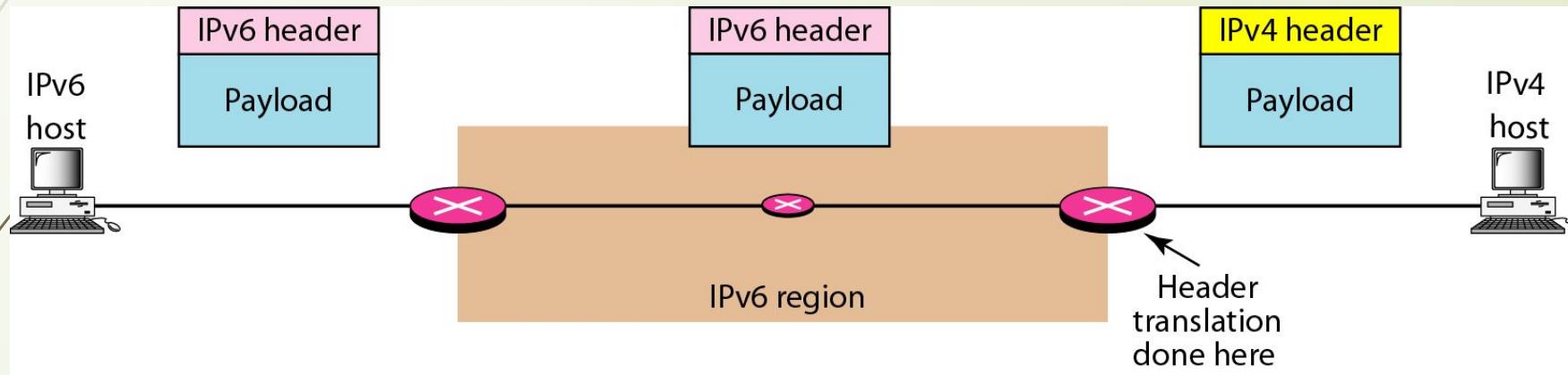
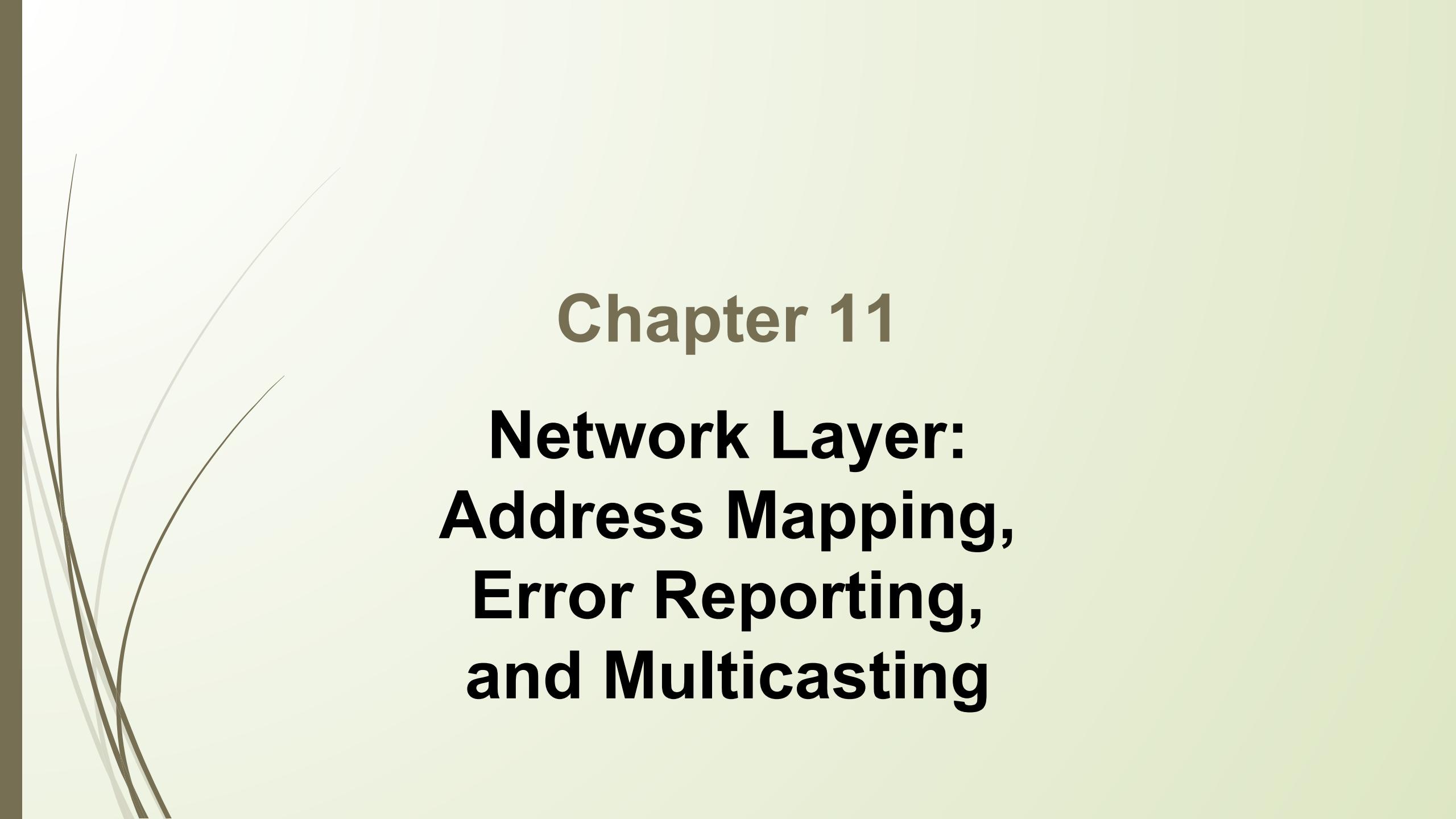


Table 10.11 *Header*

<i>Header Translation Procedure</i>
1. The IPv6 mapped address is changed to an IPv4 address by extracting the rightmost 32 bits.
2. The value of the IPv6 priority field is discarded.
3. The type of service field in IPv4 is set to zero.
4. The checksum for IPv4 is calculated and inserted in the corresponding field.
5. The IPv6 flow label is ignored.
6. Compatible extension headers are converted to options and inserted in the IPv4 header. Some may have to be dropped.
7. The length of IPv4 header is calculated and inserted into the corresponding field.
8. The total length of the IPv4 packet is calculated and inserted in the corresponding field.



Chapter 11

Network Layer: Address Mapping, Error Reporting, and Multicasting

11-1 ADDRESS MAPPING

*The delivery of a packet to a host or a router requires two levels of addressing: **logical** and **physical**. We need to be able to map a logical address to its corresponding physical address and vice versa. This can be done by using either static or dynamic mapping.*

Topics discussed in this section:

Mapping Logical to Physical Address

Mapping Physical to Logical Address

Figure 11.1 ARP operation

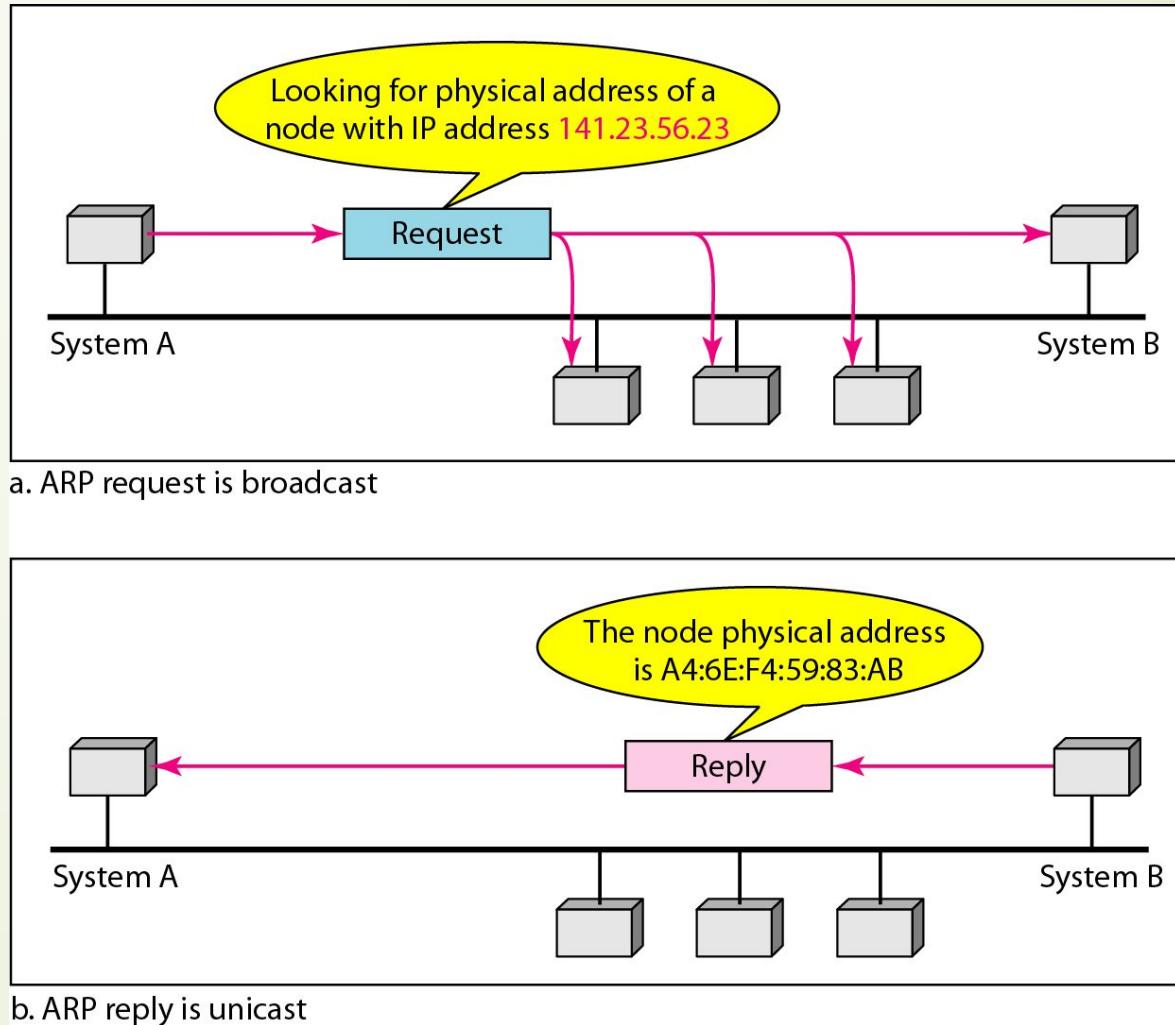


Figure 11.2 ARP packet

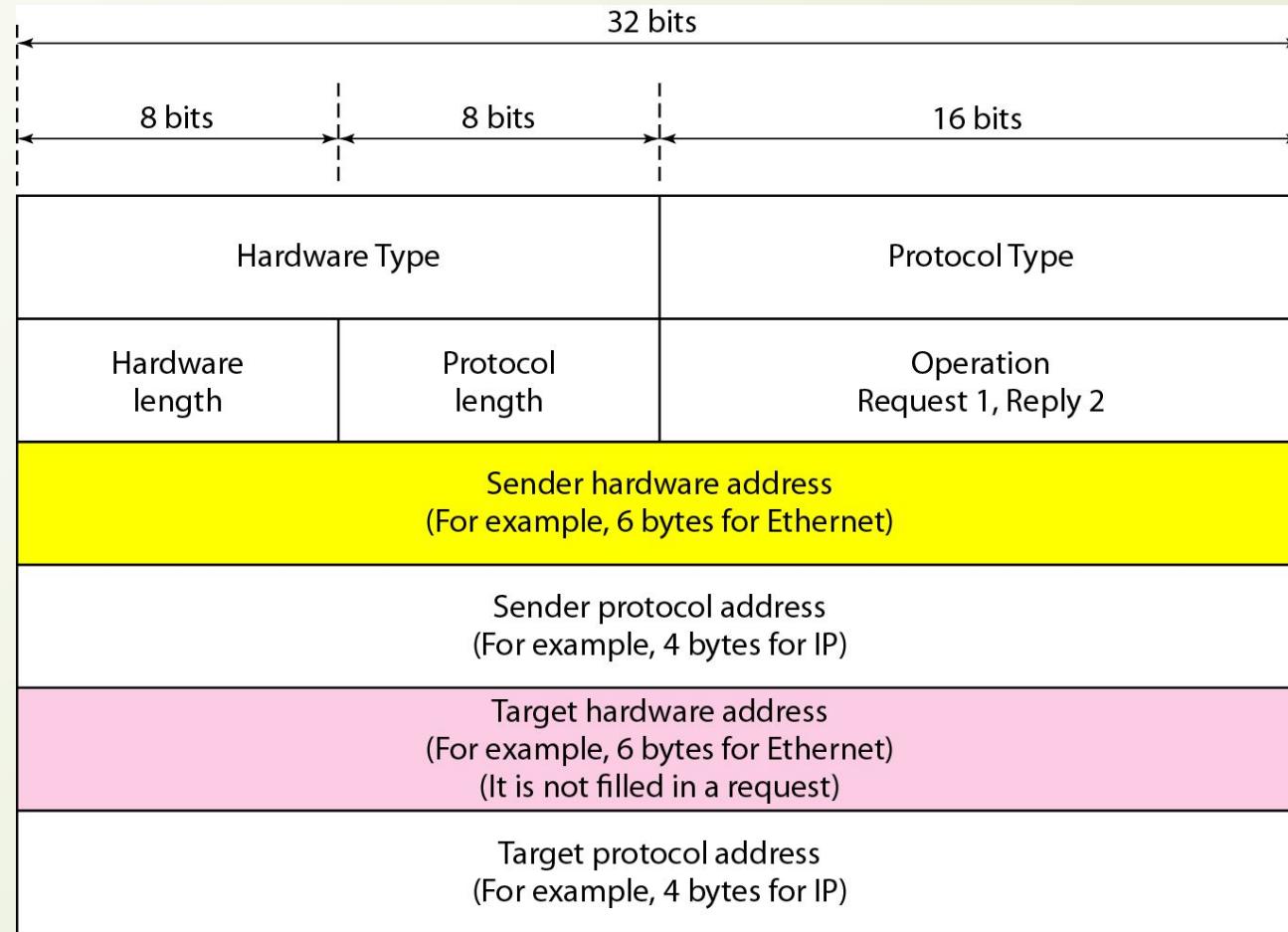


Figure 11.3 *Encapsulation of ARP packet*

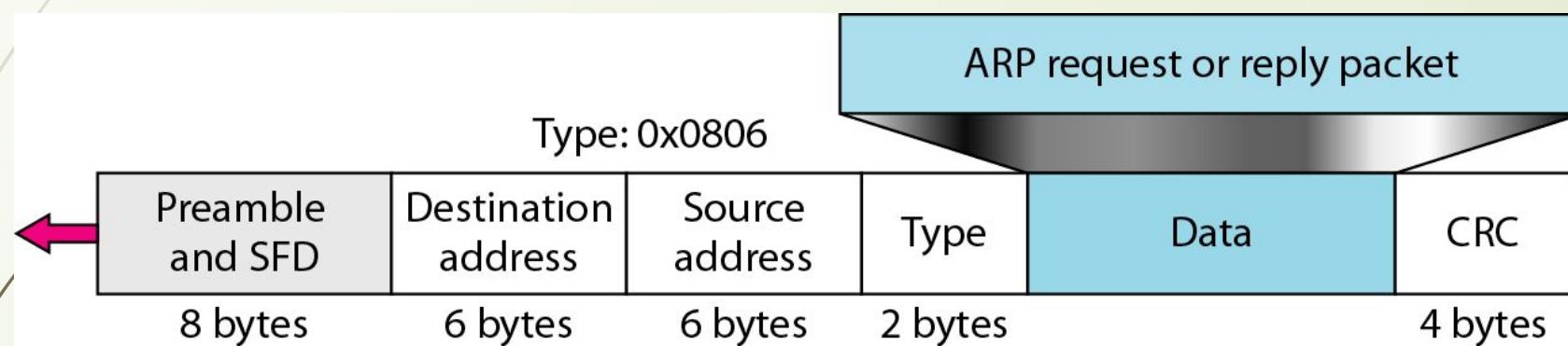
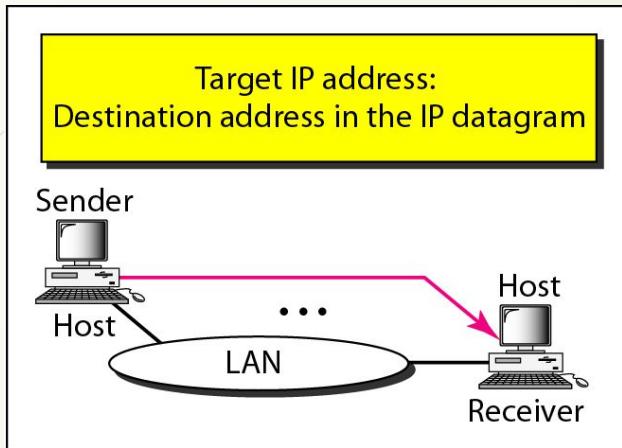
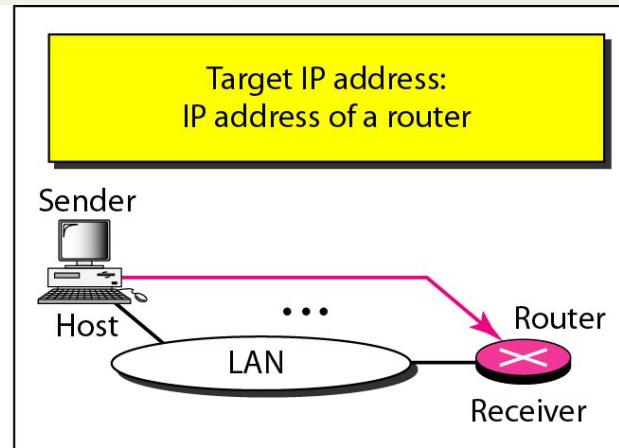


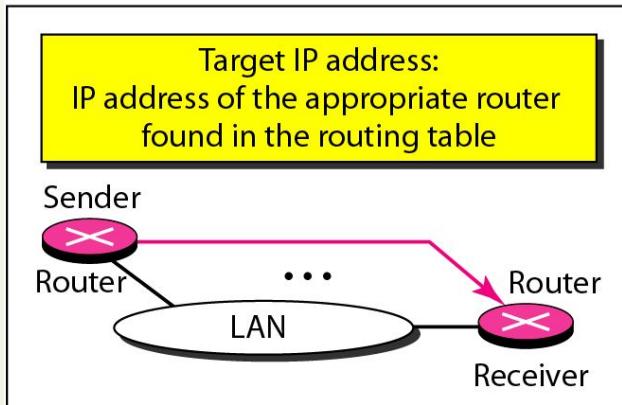
Figure 11.4 Four cases using ARP



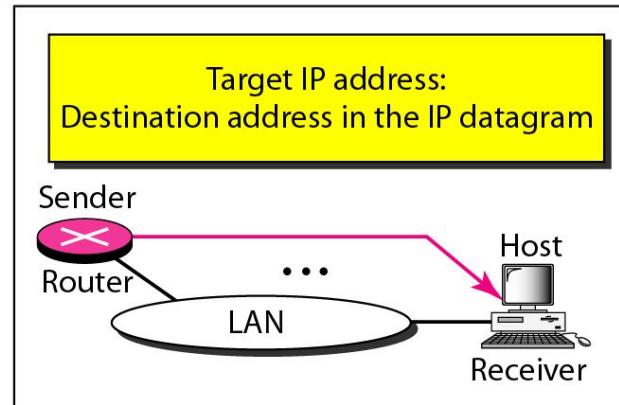
Case 1. A host has a packet to send to another host on the same network.



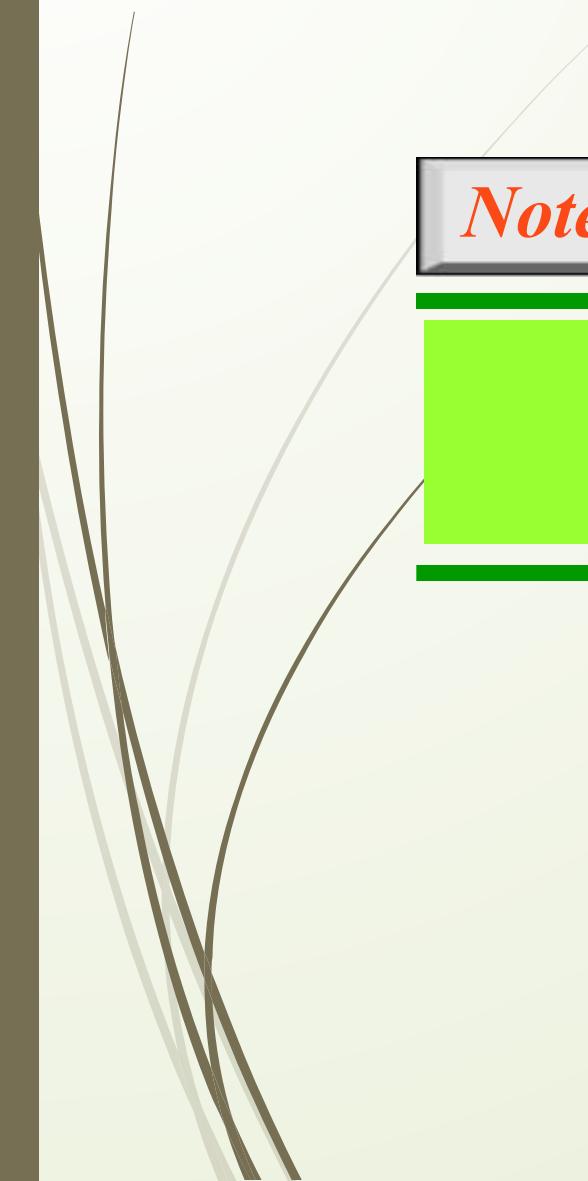
Case 2. A host wants to send a packet to another host on another network.
It must first be delivered to a router.



Case 3. A router receives a packet to be sent to a host on another network. It must first be delivered to the appropriate router.

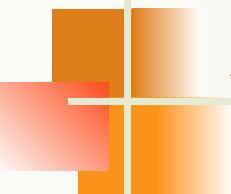


Case 4. A router receives a packet to be sent to a host on the same network.



Note

**An ARP request is broadcast;
an ARP reply is unicast.**



Example 11.1

A host with IP address 130.23.43.20 and physical address B2:34:55:10:22:10 has a packet to send to another host with IP address 130.23.43.25 and physical address A4:6E:F4:59:83:AB. The two hosts are on the same Ethernet network. Show the ARP request and reply packets encapsulated in Ethernet frames.

Solution

Figure 11.5 shows the ARP request and reply packets. Note that the ARP data field in this case is 28 bytes, and that the individual addresses do not fit in the 4-byte boundary. That is why we do not show the regular 4-byte boundaries for these addresses.

Figure 11.5 Example 11.1, an ARP request and reply

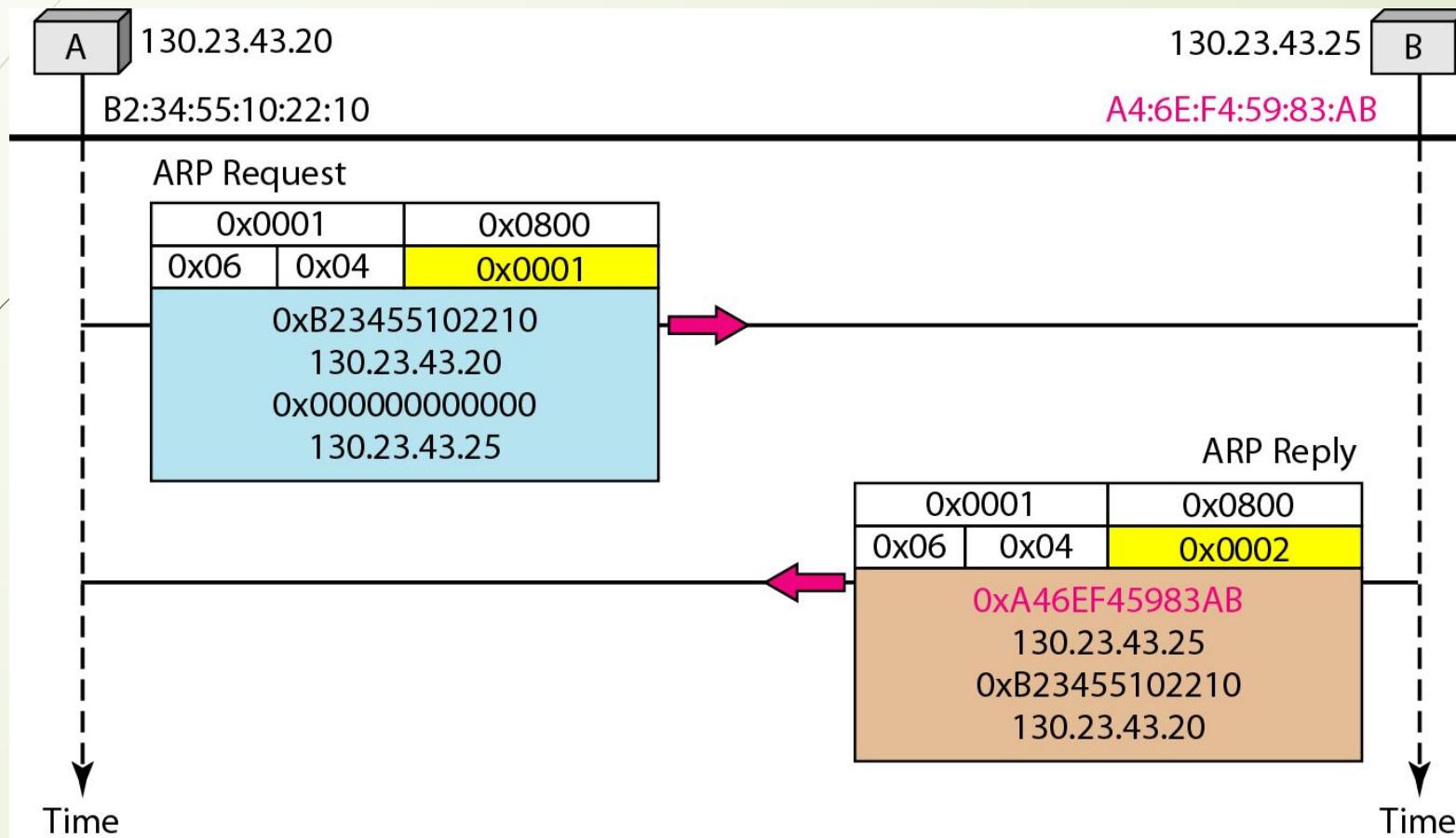


Figure 11.6 *Proxy ARP*

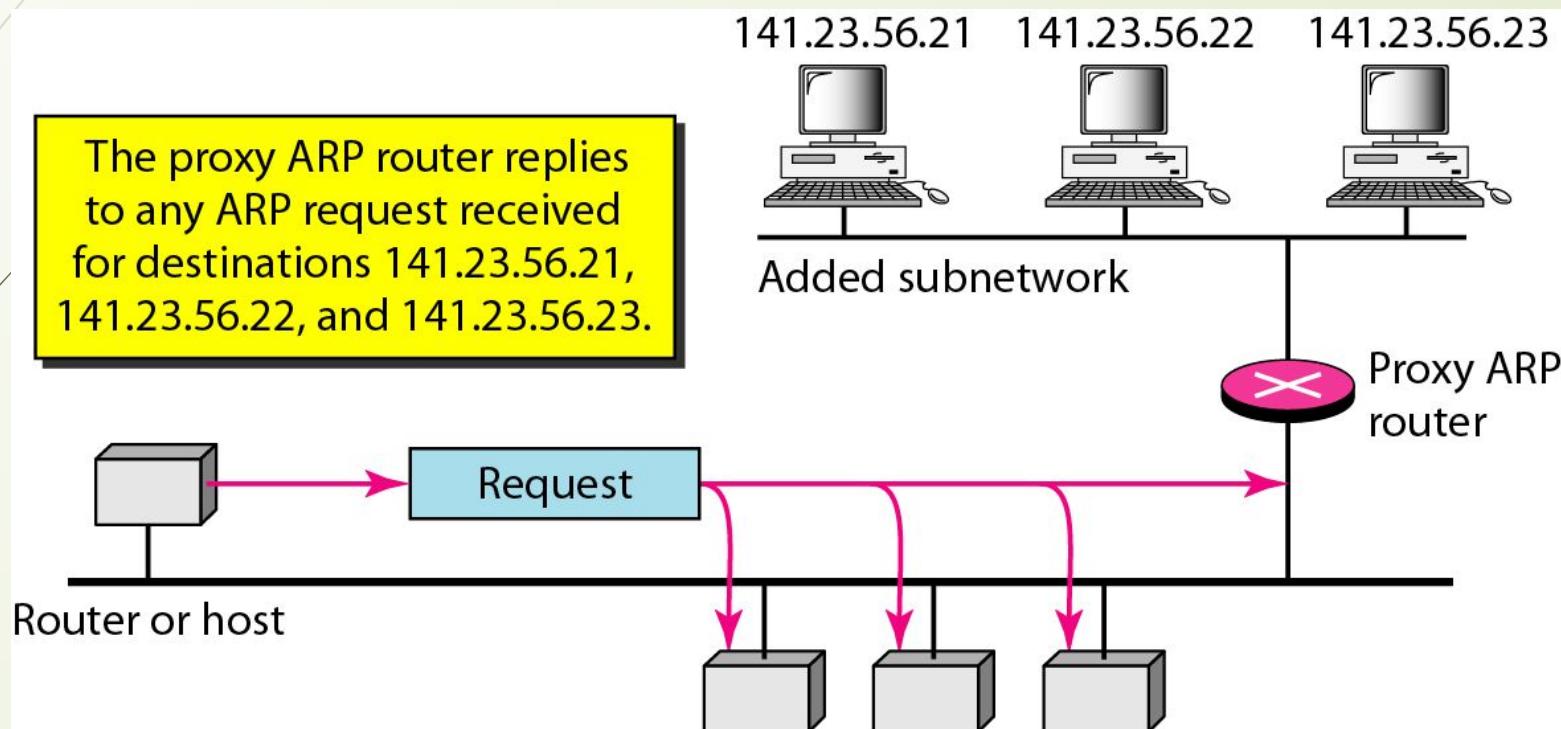
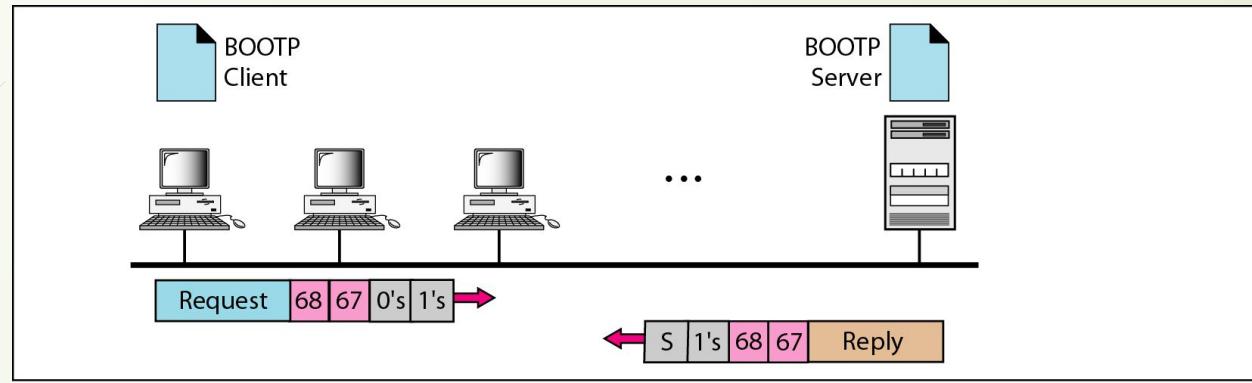
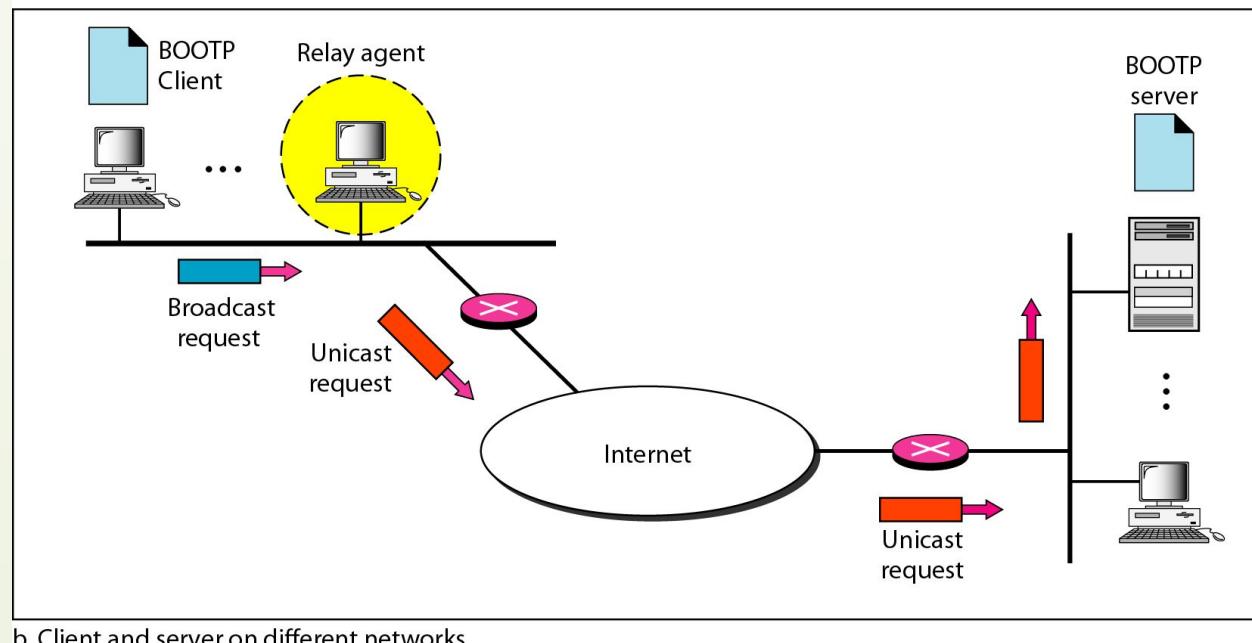


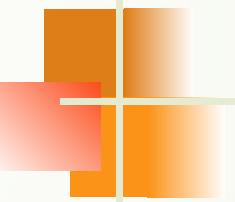
Figure 11.7 BOOTP client and server on the same and different networks



a. Client and server on the same network



b. Client and server on different networks



Note

DHCP provides static and dynamic address allocation that can be manual or automatic.

The IP protocol has no error-reporting or error-correcting mechanism. The IP protocol also lacks a mechanism for host and management queries. The Internet Control Message Protocol (ICMP) has been designed to compensate for the above two deficiencies. It is a companion to the IP protocol.

Topics discussed in this section:

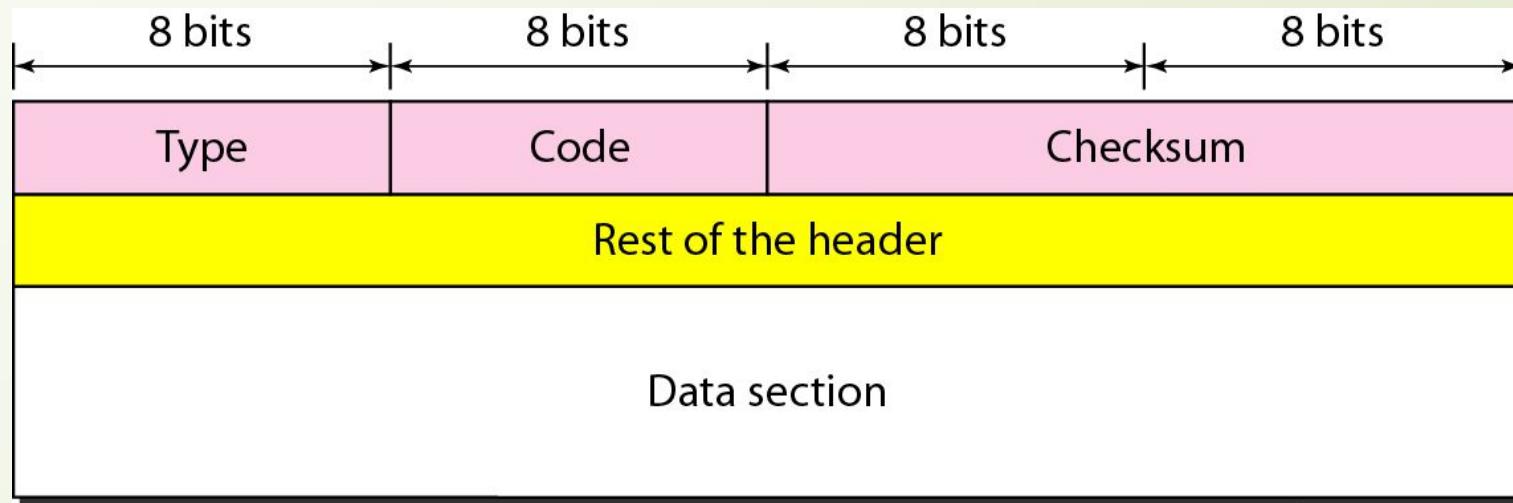
Types of Messages

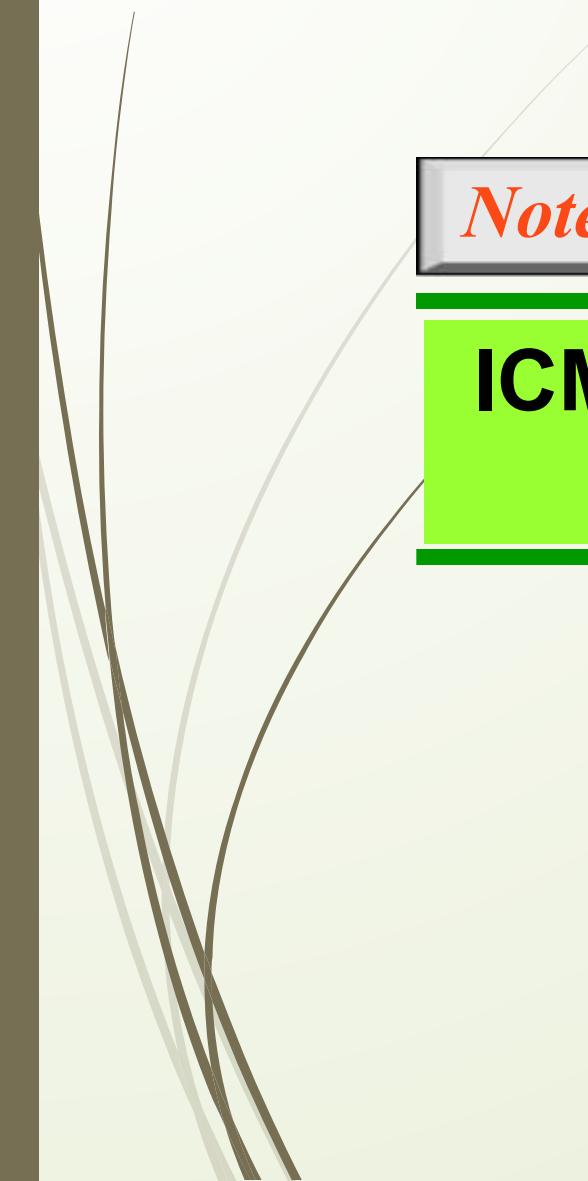
Message Format

Error Reporting and Query

Debugging Tools

Figure 11.8 *General format of ICMP messages*

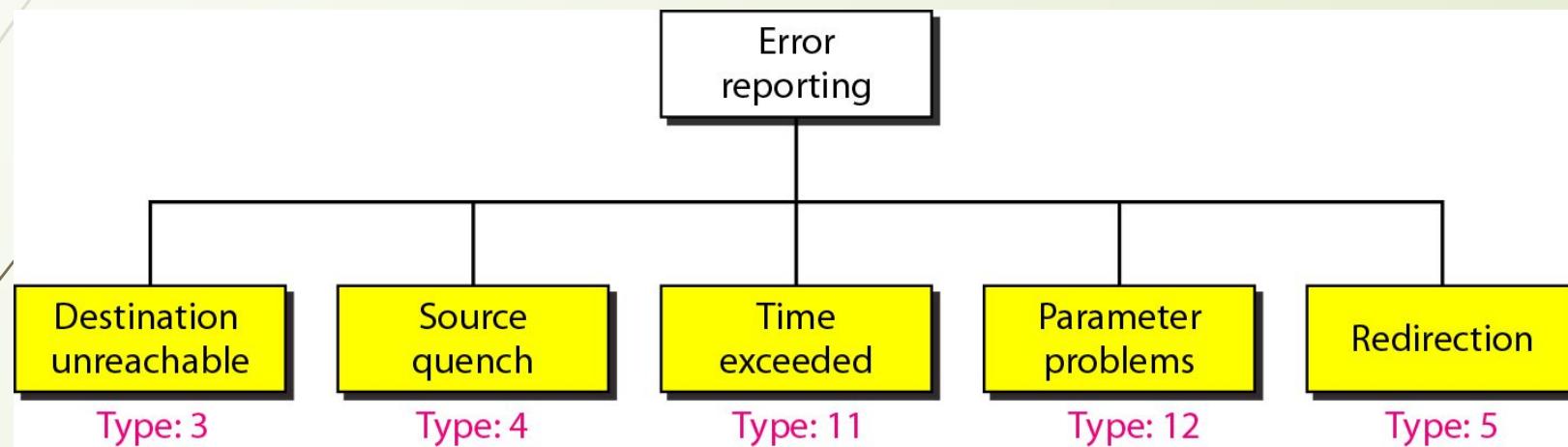


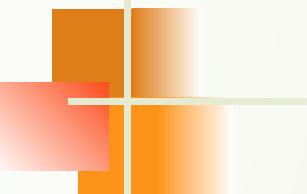


Note

ICMP always reports error messages to the original source.

Figure 11.9 Error-reporting messages





Note

Important points about ICMP error messages:

- No ICMP error message will be generated in response to a datagram carrying an ICMP error message.
- No ICMP error message will be generated for a fragmented datagram that is not the first fragment.
- No ICMP error message will be generated for a datagram having a multicast address.
- No ICMP error message will be generated for a datagram having a special address such as 127.0.0.0 or 0.0.0.0.

Figure 11.10 *Contents of data field for the error messages*

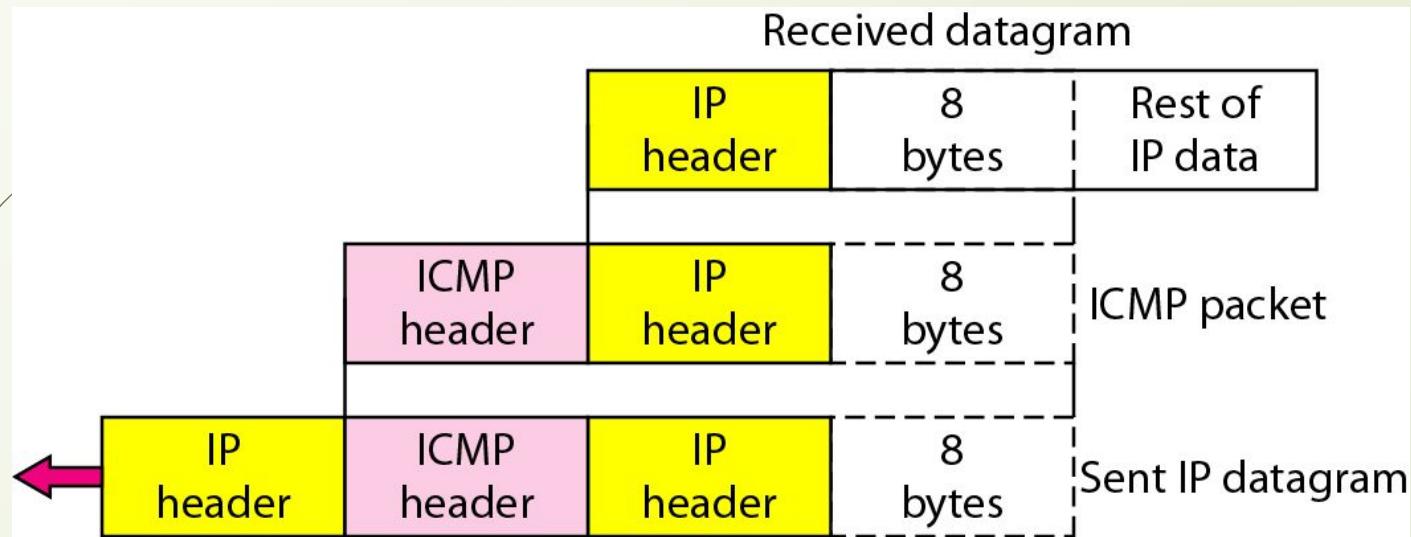


Figure 11.11 *Redirection concept*

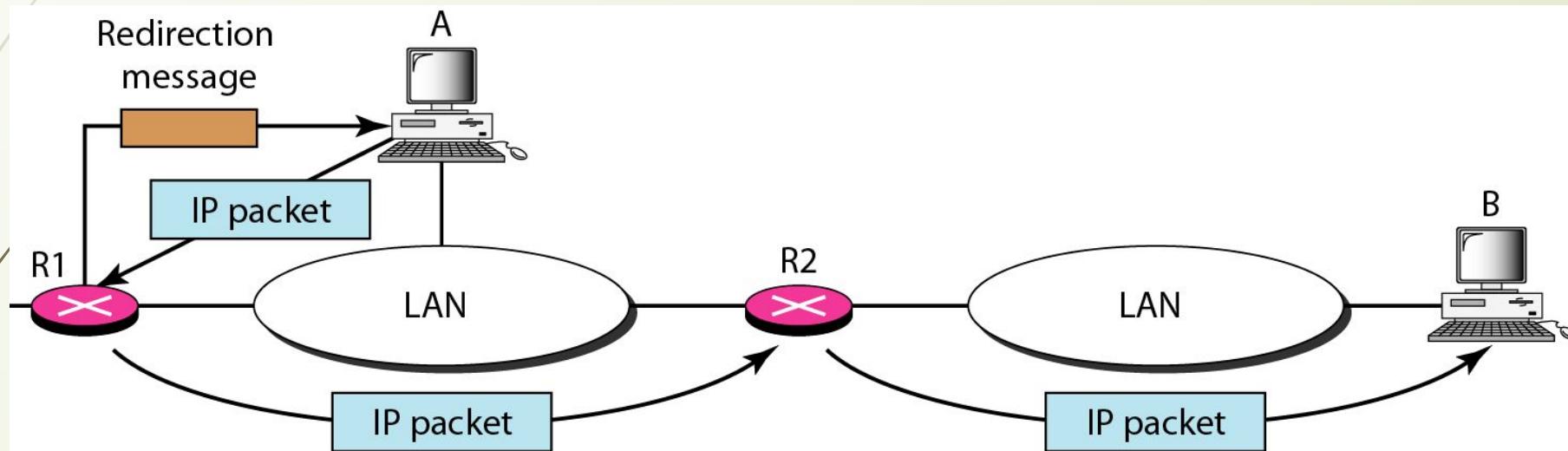


Figure 11.12 *Query messages*

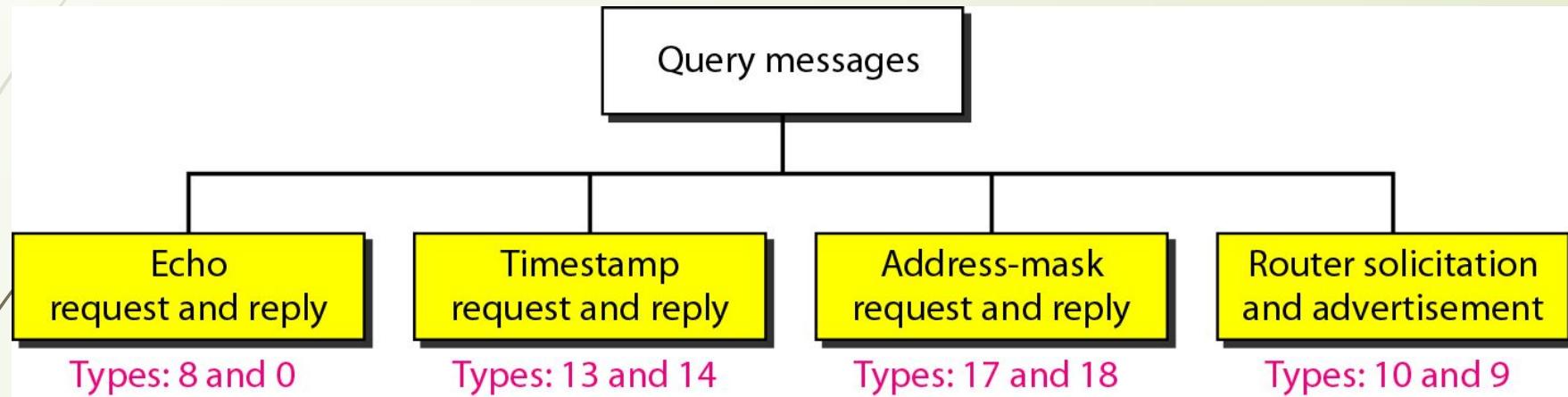
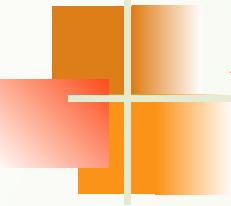


Figure 11.13 *Encapsulation of ICMP query messages*



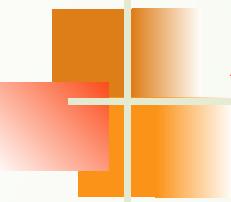


Example 11.2

Figure 11.14 shows an example of checksum calculation for a simple echo-request message. We randomly chose the identifier to be 1 and the sequence number to be 9. The message is divided into 16-bit (2-byte) words. The words are added and the sum is complemented. Now the sender can put this value in the checksum field.

Figure 11.14 Example of checksum calculation

8	0	0	
1		9	
TEST			
8 & 0	→	00001000	00000000
0	→	00000000	00000000
1	→	00000000	00000001
9	→	00000000	00001001
T & E	→	01010100	01000101
S & T	→	01010011	01010100
		Sum	→ 10101111 10100011
		Checksum	→ 01010000 01011100



Example 11.3

We use the ping program to test the server fhda.edu. The result is shown on the next slide. The ping program sends messages with sequence numbers starting from 0. For each probe it gives us the RTT time. The TTL (time to live) field in the IP datagram that encapsulates an ICMP message has been set to 62. At the beginning, ping defines the number of data bytes as 56 and the total number of bytes as 84. It is obvious that if we add 8 bytes of ICMP header and 20 bytes of IP header to 56, the result is 84. However, note that in each probe ping defines the number of bytes as 64. This is the total number of bytes in the ICMP packet ($56 + 8$).

Example 11.3 (continued)

\$ ping fhda.edu

PING fhda.edu (153.18.8.1) 56 (84) bytes of data.

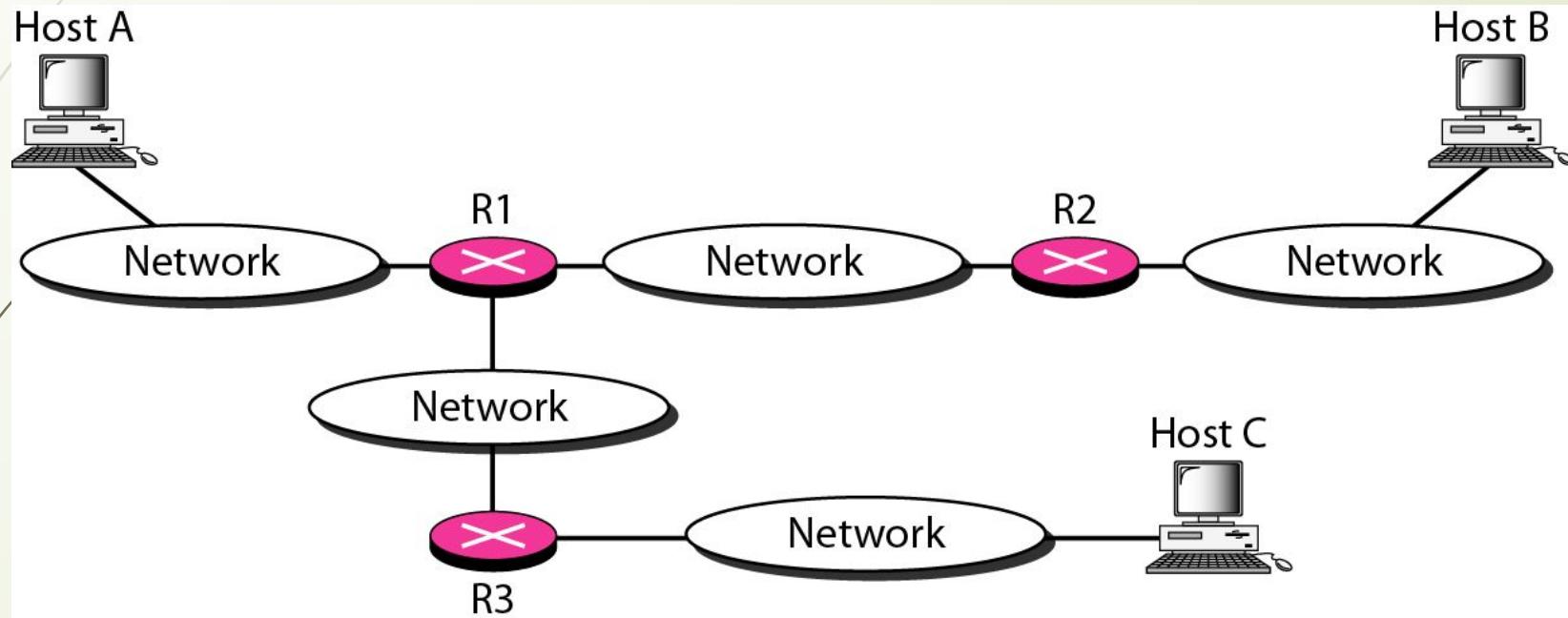
64 bytes from tiptoe.fhda.edu (153.18.8.1): icmp_seq=0	ttl=62	time=1.91 ms
64 bytes from tiptoe.fhda.edu (153.18.8.1): icmp_seq=1	ttl=62	time=2.04 ms
64 bytes from tiptoe.fhda.edu (153.18.8.1): icmp_seq=2	ttl=62	time=1.90 ms
64 bytes from tiptoe.fhda.edu (153.18.8.1): icmp_seq=3	ttl=62	time=1.97 ms
64 bytes from tiptoe.fhda.edu (153.18.8.1): icmp_seq=4	ttl=62	time=1.93 ms
64 bytes from tiptoe.fhda.edu (153.18.8.1): icmp_seq=5	ttl=62	time=2.00 ms
64 bytes from tiptoe.fhda.edu (153.18.8.1): icmp_seq=6	ttl=62	time=1.94 ms
64 bytes from tiptoe.fhda.edu (153.18.8.1): icmp_seq=7	ttl=62	time=1.94 ms
64 bytes from tiptoe.fhda.edu (153.18.8.1): icmp_seq=8	ttl=62	time=1.97 ms
64 bytes from tiptoe.fhda.edu (153.18.8.1): icmp_seq=9	ttl=62	time=1.89 ms
64 bytes from tiptoe.fhda.edu (153.18.8.1): icmp_seq=10	ttl=62	time=1.98 ms

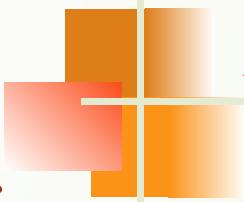
--- fhda.edu ping statistics ---

11 packets transmitted, 11 received, 0% packet loss, time 10103ms

rtt min/avg/max = 1.899/1.955/2.041 ms

Figure 11.15 *The traceroute program operation*



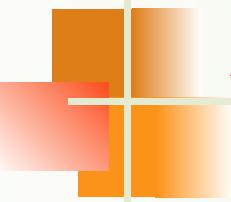


Example 11.4

We use the traceroute program to find the route from the computer `voyager.deanza.edu` to the server `fhda.edu`. The following shows the result:

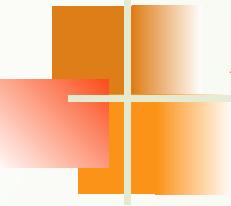
```
$ traceroute fhda.edu
traceroute to fhda.edu (153.18.8.1), 30 hops max, 38 byte packets
 1 Dcore.fhda.edu      (153.18.31.254)   0.995 ms   0.899 ms   0.878 ms
 2 Dbackup.fhda.edu    (153.18.251.4)     1.039 ms   1.064 ms   1.083 ms
 3 tiptoe.fhda.edu     (153.18.8.1)       1.797 ms   1.642 ms   1.757 ms
```

The unnumbered line after the command shows that the destination is 153.18.8.1. The packet contains 38 bytes: 20 bytes of IP header, 8 bytes of UDP header, and 10 bytes of application data. The application data are used by traceroute to keep track of the packets.



Example 11.4 (continued)

The first line shows the first router visited. The router is named Dcore.fhda.edu with IP address 153.18.31.254. The first round-trip time was 0.995 ms, the second was 0.899 ms, and the third was 0.878 ms. The second line shows the second router visited. The router is named Dbackup.fhda.edu with IP address 153.18.251.4. The three round-trip times are also shown. The third line shows the destination host. We know that this is the destination host because there are no more lines. The destination host is the server fhda.edu, but it is named tiptoe.fhda.edu with the IP address 153.18.8.1. The three round-trip times are also shown.



Example 11.5

In this example, we trace a longer route, the route to xerox.com (see next slide). Here there are 17 hops between source and destination. Note that some round-trip times look unusual. It could be that a router was too busy to process the packet immediately.

Example 11.5 (continued)

```
$ traceroute xerox.com
```

traceroute to xerox.com (13.1.64.93), 30 hops max, 38 byte packets

1	Dcore.fhda.edu	(153.18.31.254)	0.622 ms	0.891 ms	0.875 ms
2	Ddmz.fhda.edu	(153.18.251.40)	2.132 ms	2.266 ms	2.094 ms
3	Cinic.fhda.edu	(153.18.253.126)	2.110 ms	2.145 ms	1.763 ms
4	cenic.net	(137.164.32.140)	3.069 ms	2.875 ms	2.930 ms
5	cenic.net	(137.164.22.31)	4.205 ms	4.870 ms	4.197 ms
...
14	snfc21.pbi.net	(151.164.191.49)	7.656 ms	7.129 ms	6.866 ms
15	sbcglobal.net	(151.164.243.58)	7.844 ms	7.545 ms	7.353 ms
16	pacbell.net	(209.232.138.114)	9.857 ms	9.535 ms	9.603 ms
17	209.233.48.223	(209.233.48.223)	10.634 ms	10.771 ms	10.592 ms
18	alpha.Xerox.COM	(13.1.64.93)	11.172 ms	11.048 ms	10.922 ms

The IP protocol can be involved in two types of communication: unicasting and multicasting. The Internet Group Management Protocol (IGMP) is one of the necessary, but not sufficient, protocols that is involved in multicasting. IGMP is a companion to the IP protocol.

Topics discussed in this section:

Group Management

IGMP Messages and IGMP Operation

Encapsulation

Netstat Utility

Figure 11.16 *IGMP message types*

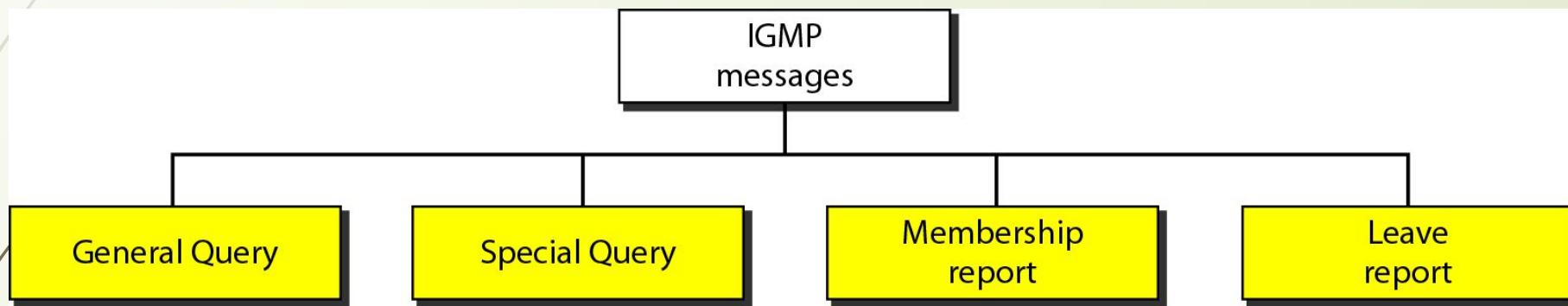


Figure 11.17 IGMP message format

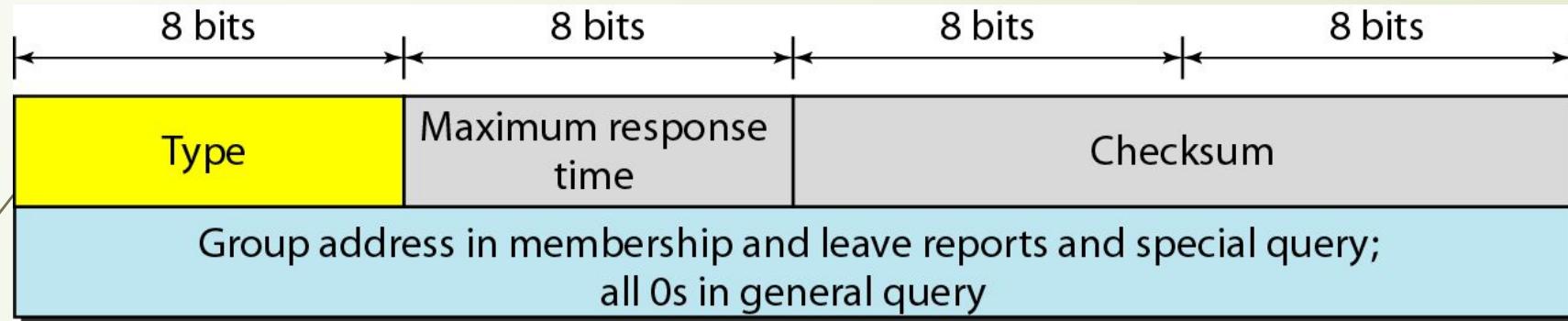
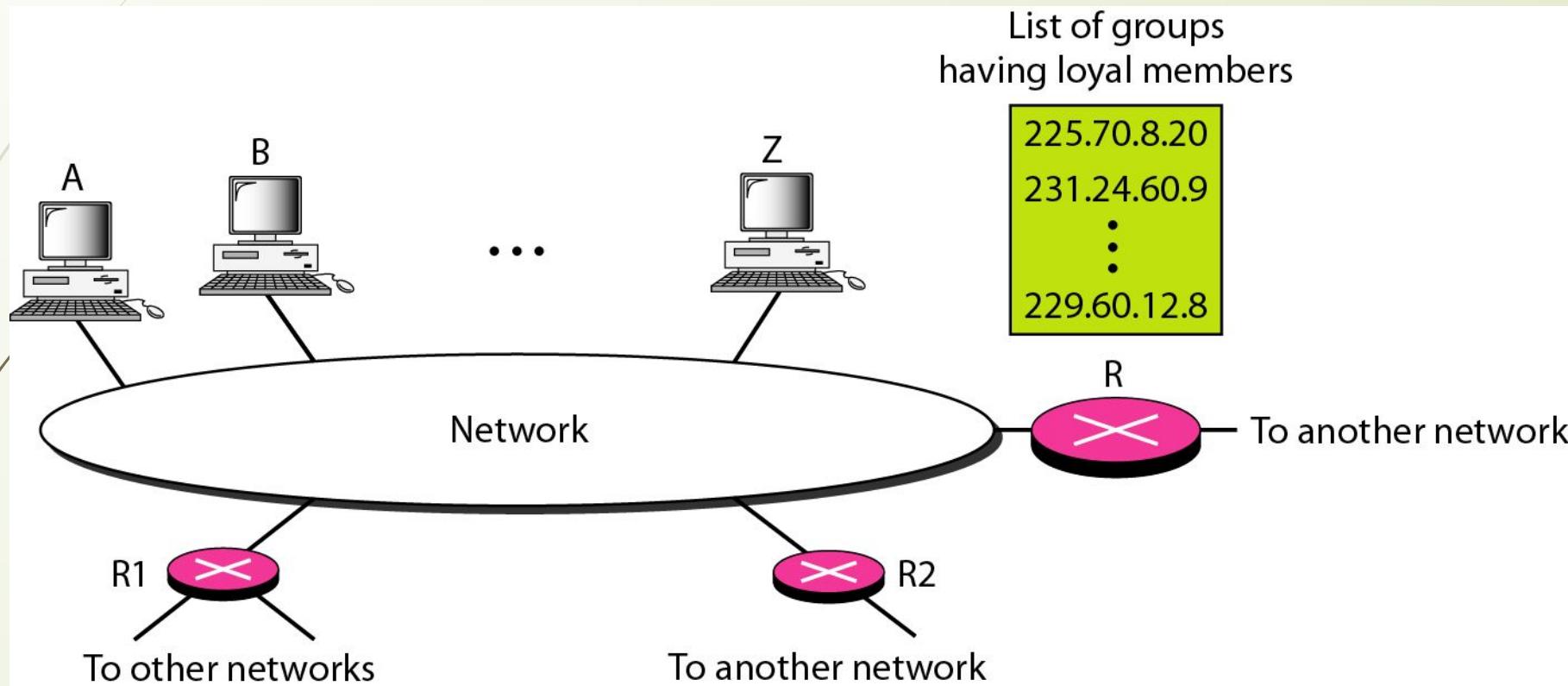
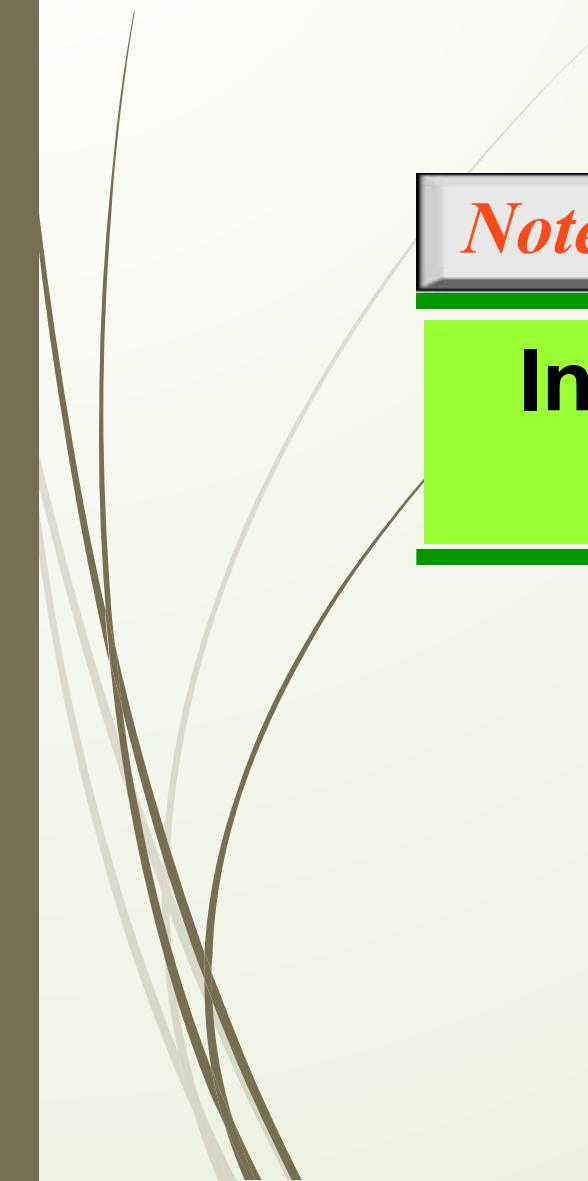


Table 11.1 *IGMP type*

<i>Type</i>	<i>Value</i>
General or special query	0x11 or 00010001
Membership report	0x16 or 00010110
Leave report	0x17 or 00010111

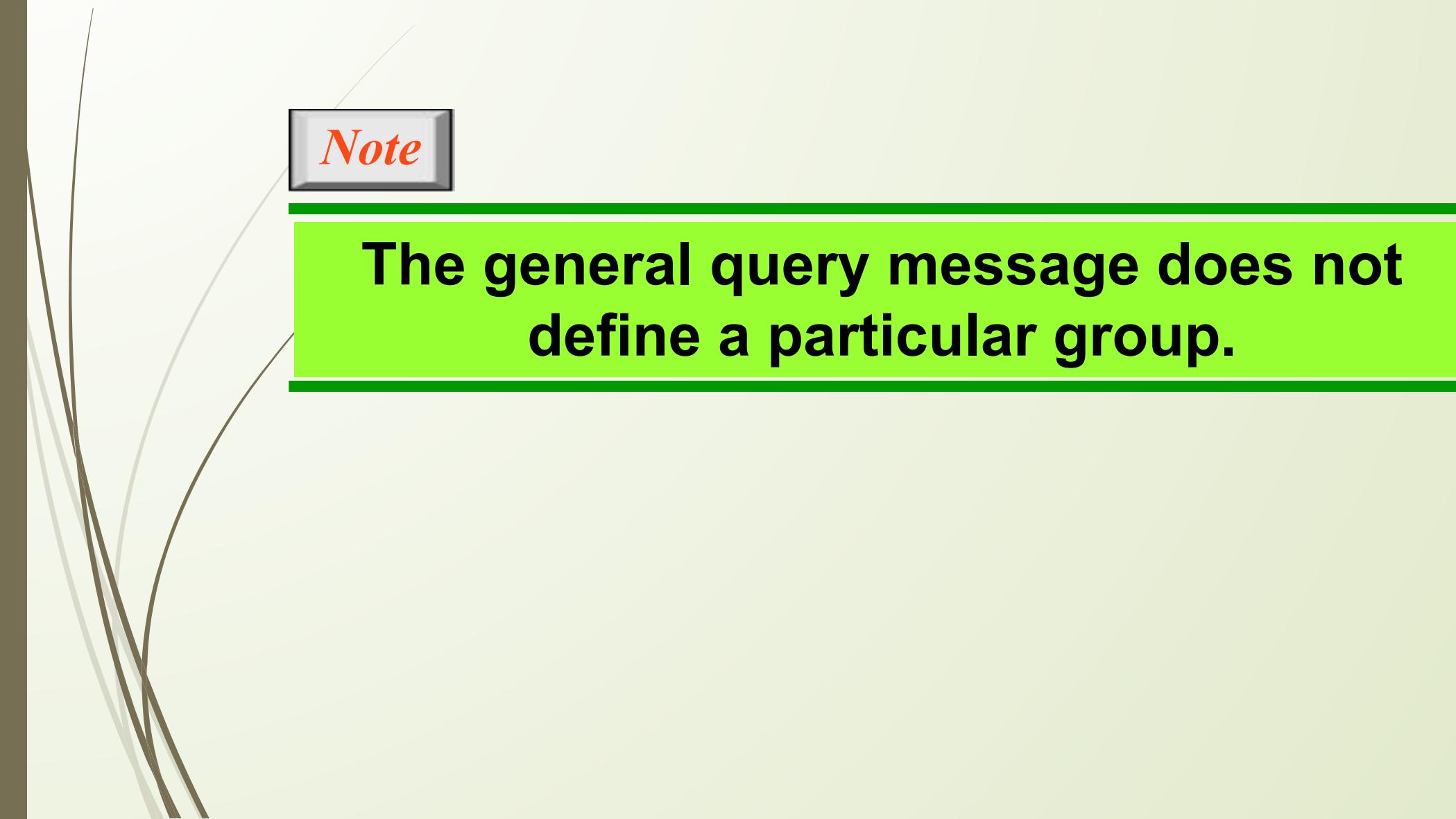
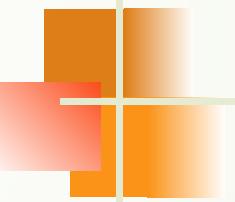
Figure 11.18 *IGMP operation*





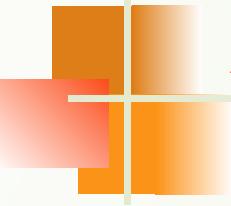
Note

In IGMP, a membership report is sent twice, one after the other.



Note

The general query message does not define a particular group.



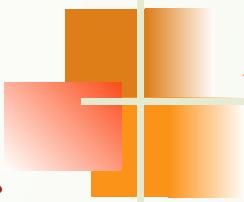
Example 11.6

Imagine there are three hosts in a network, as shown in Figure 11.19. A query message was received at time 0; the random delay time (in tenths of seconds) for each group is shown next to the group address. Show the sequence of report messages.

Solution

The events occur in this sequence:

- a. **Time 12:** The timer for 228.42.0.0 in host A expires, and a membership report is sent, which is received by the router and every host including host B which cancels its timer for 228.42.0.0.



Example 11.6 (continued)

- b. Time 30: The timer for 225.14.0.0 in host A expires, and a membership report is sent which is received by the router and every host including host C which cancels its timer for 225.14.0.0.*
- c. Time 50: The timer for 238.71.0.0 in host B expires, and a membership report is sent, which is received by the router and every host.*
- d. Time 70: The timer for 230.43.0.0 in host C expires, and a membership report is sent, which is received by the router and every host including host A which cancels its timer for 230.43.0.0.*

Figure 11.19 Example 11.6

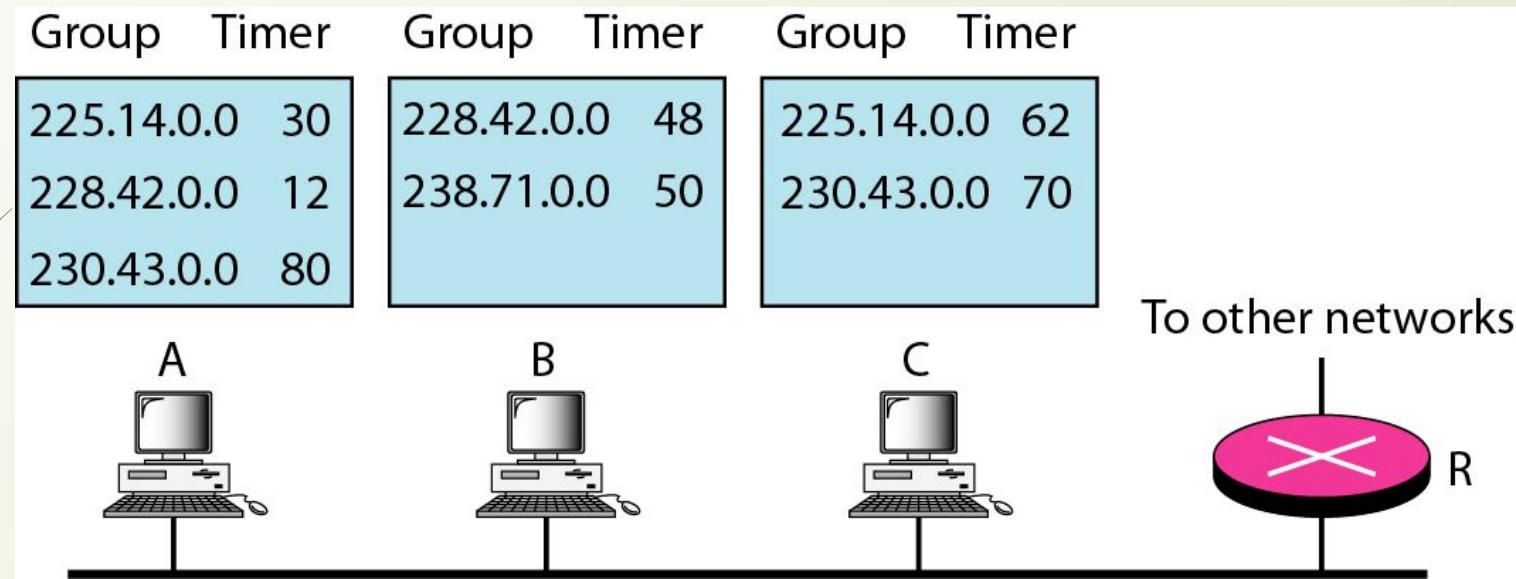
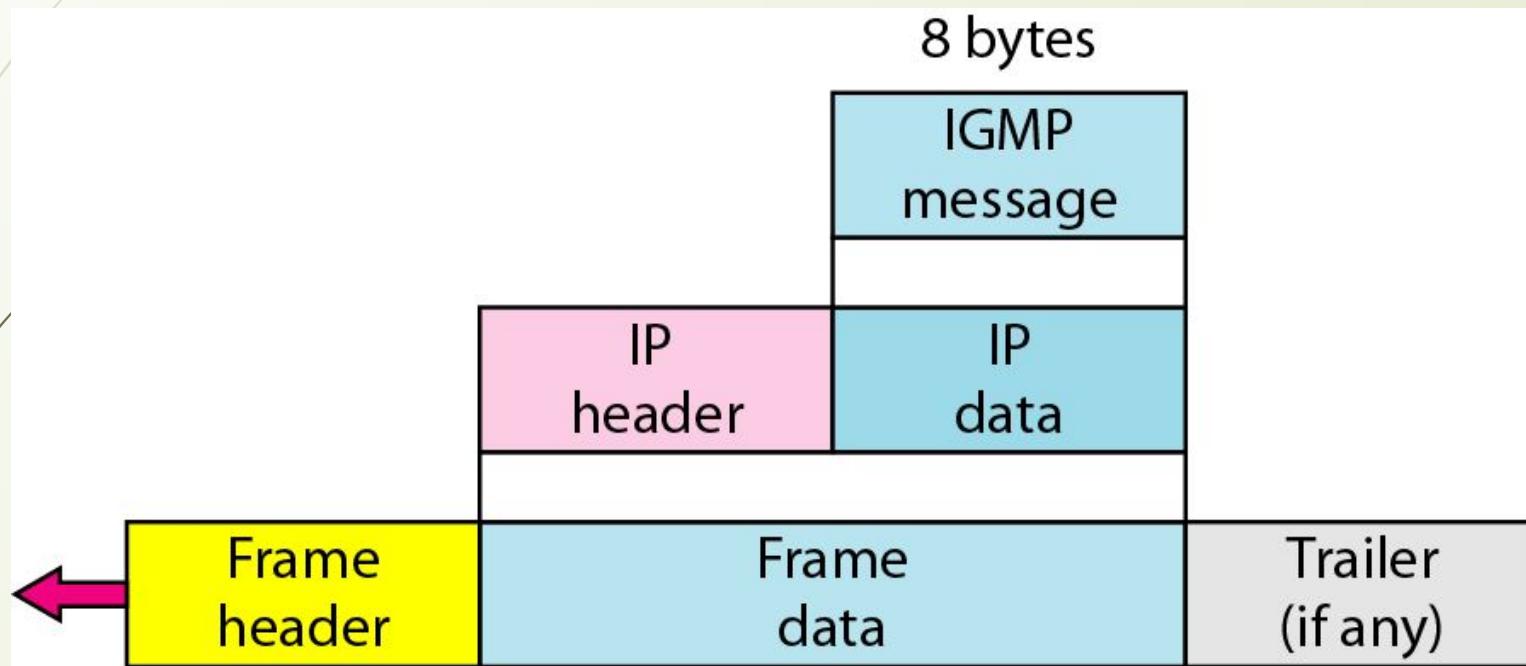
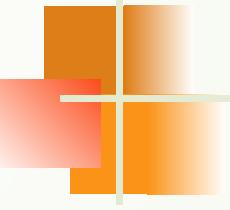


Figure 11.20 *Encapsulation of IGMP packet*





Note

The IP packet that carries an IGMP packet has a value of 1 in its TTL field.

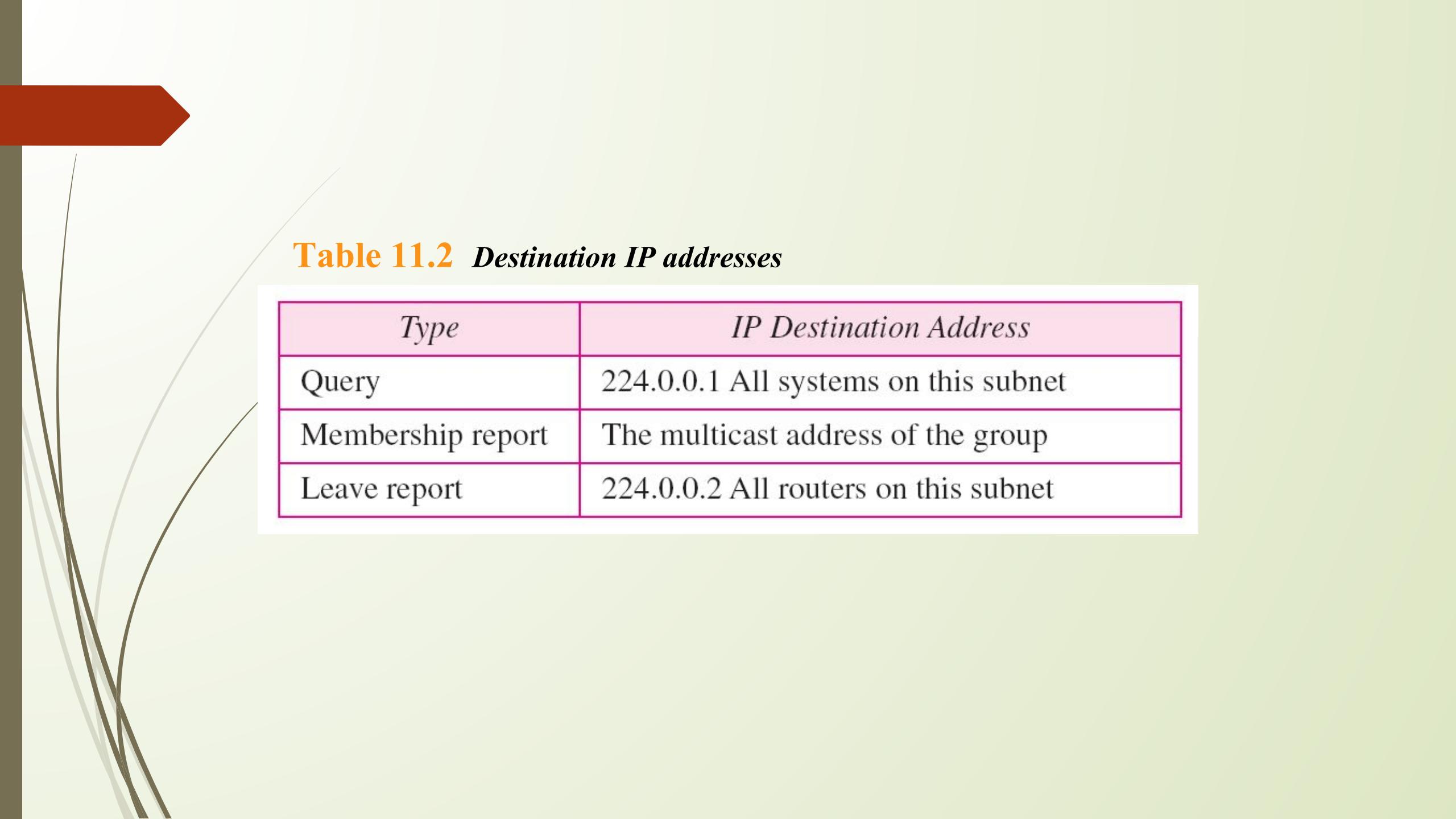
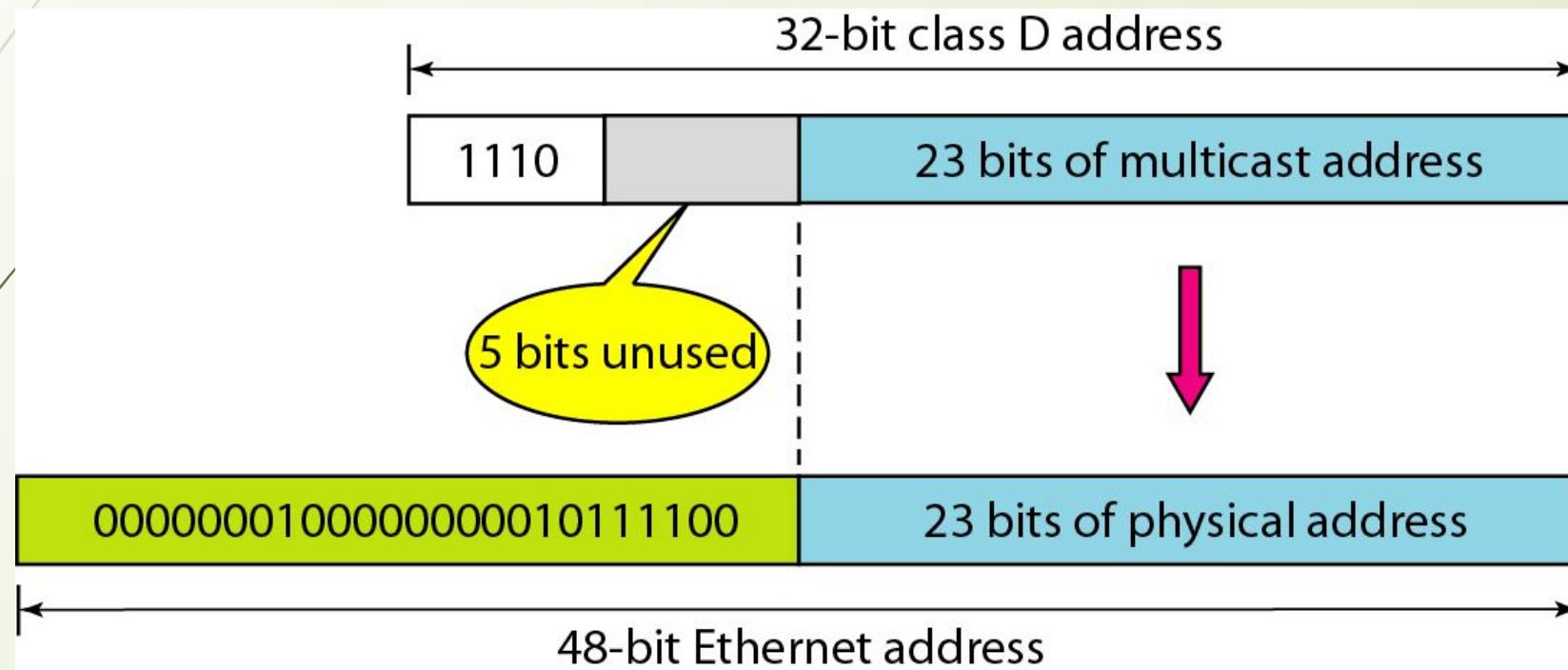


Table 11.2 *Destination IP addresses*

Type	IP Destination Address
Query	224.0.0.1 All systems on this subnet
Membership report	The multicast address of the group
Leave report	224.0.0.2 All routers on this subnet

Figure 11.21 *Mapping class D to Ethernet physical address*

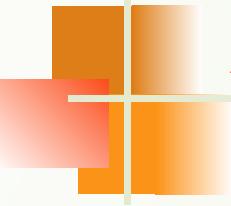




Note

**An Ethernet multicast physical address
is in the range**

01:00:5E:00:00:00 to 01:00:5E:7F:FF:FF.



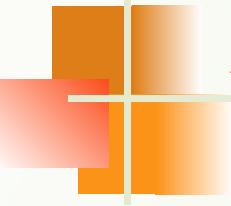
Example 11.7

Change the multicast IP address 230.43.14.7 to an Ethernet multicast physical address.

Solution

We can do this in two steps:

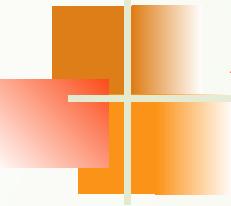
- a. *We write the rightmost 23 bits of the IP address in hexadecimal. This can be done by changing the rightmost 3 bytes to hexadecimal and then subtracting 8 from the leftmost digit if it is greater than or equal to 8. In our example, the result is 2B:0E:07.*



Example 11.7 (continued)

- b. We add the result of part a to the starting Ethernet multicast address, which is 01:00:5E:00:00:00. The result is*

01:00:5E:2B:0E:07



Example 11.8

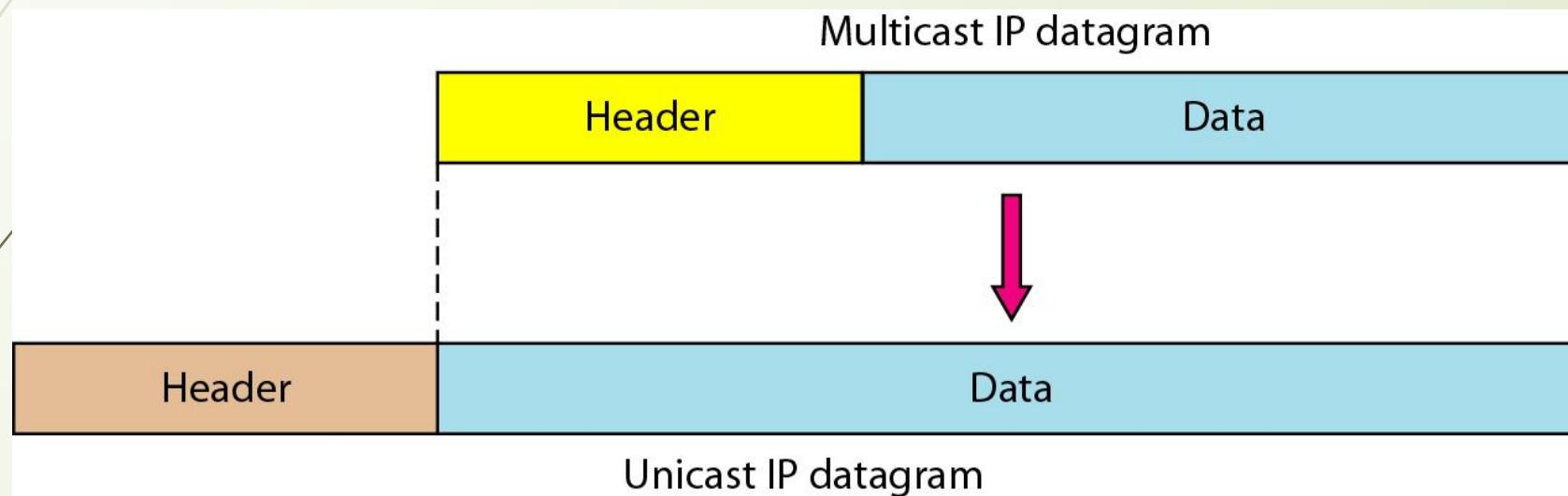
Change the multicast IP address 238.212.24.9 to an Ethernet multicast address.

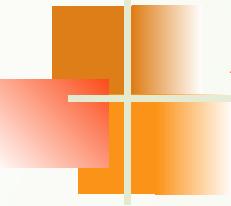
Solution

- a. The rightmost 3 bytes in hexadecimal is D4:18:09. We need to subtract 8 from the leftmost digit, resulting in 54:18:09.*
- b. We add the result of part a to the Ethernet multicast starting address. The result is*

01:00:5E:54:18:09

Figure 11.22 *Tunneling*

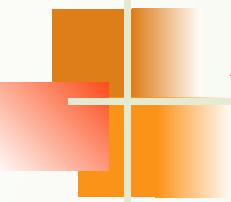




Example 11.9

We use netstat (see next slide) with three options: -n, -r, and -a. The -n option gives the numeric versions of IP addresses, the -r option gives the routing table, and the -a option gives all addresses (unicast and multicast). Note that we show only the fields relative to our discussion. “Gateway” defines the router, “Iface” defines the interface.

Note that the multicast address is shown in color. Any packet with a multicast address from 224.0.0.0 to 239.255.255.255 is masked and delivered to the Ethernet interface.



Example 11.9 (continued)

```
$ netstat -nra
```

Kernel IP routing table

Destination	Gateway	Mask	Flags	Iface
153.18.16.0	0.0.0.0	255.255.240.0	U	eth0
169.254.0.0	0.0.0.0	255.255.0.0	U	eth0
127.0.0.0	0.0.0.0	255.0.0.0	U	lo
224.0.0.0	0.0.0.0	224.0.0.0	U	eth0
0.0.0.0	153.18.31.254	0.0.0.0	UG	eth0

11-4 ICMPv6

We discussed IPv6 in Chapter 10. Another protocol that has been modified in version 6 of the TCP/IP protocol suite is ICMP (ICMPv6). This new version follows the same strategy and purposes of version 4.

Topics discussed in this section:

Error Reporting

Query

Figure 11.23 *Comparison of network layers in version 4 and version 6*

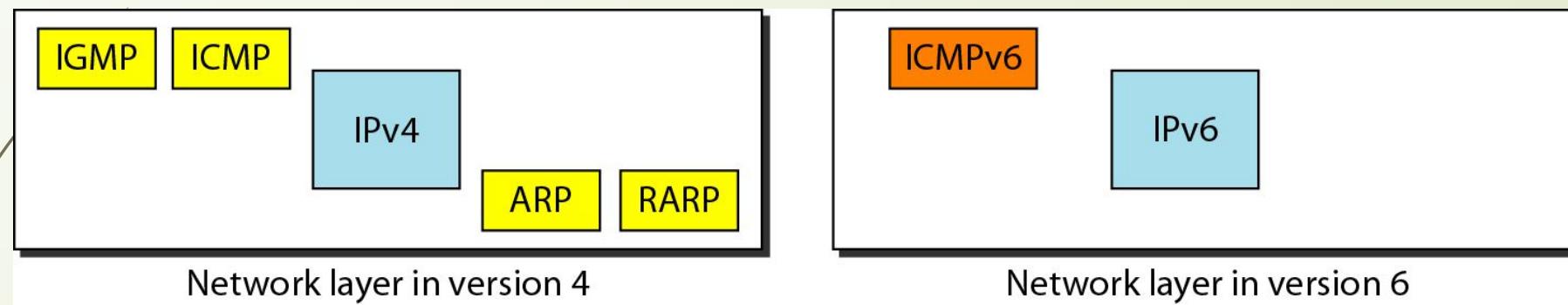


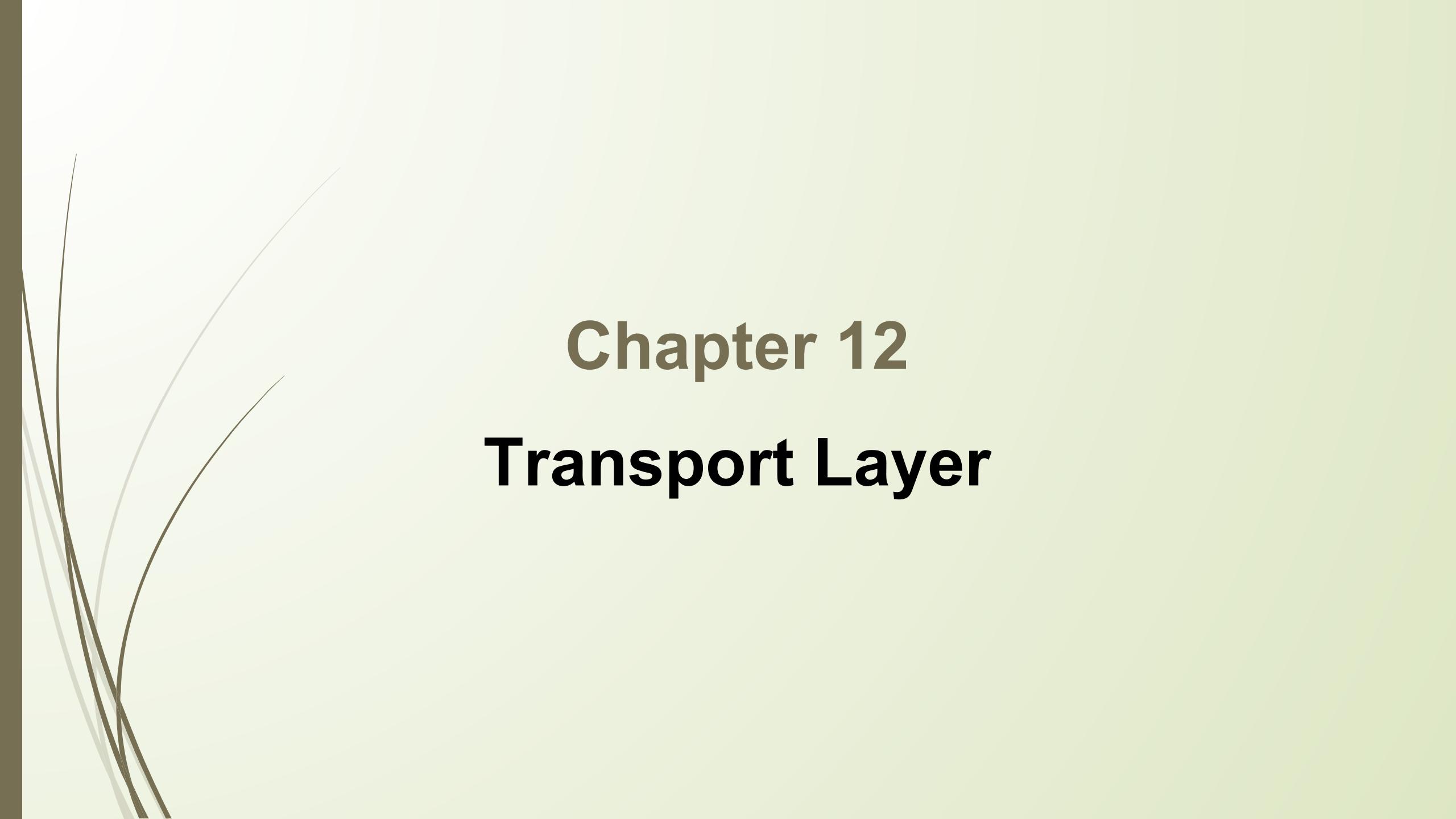


Table 11.3 Comparison of error-reporting messages in ICMPv4 and ICMPv6

Type of Message	Version 4	Version 6
Destination unreachable	Yes	Yes
Source quench	Yes	No
Packet too big	No	Yes
Time exceeded	Yes	Yes
Parameter problem	Yes	Yes
Redirection	Yes	Yes

Table 11.4 *Comparison of query messages in ICMPv4 and ICMPv6*

Type of Message	Version 4	Version 6
Echo request and reply	Yes	Yes
Timestamp request and reply	Yes	No
Address-mask request and reply	Yes	No
Router solicitation and advertisement	Yes	Yes
Neighbor solicitation and advertisement	ARP	Yes
Group membership	IGMP	Yes



Chapter 12

Transport Layer

12-1 PROCESS-TO-PROCESS DELIVERY

The transport layer is responsible for process-to-process delivery—the delivery of a packet, part of a message, from one process to another. Two processes communicate in a client/server relationship, as we will see later.

Topics discussed in this section:

Client/Server Paradigm

Multiplexing and Demultiplexing

Connectionless Versus Connection-Oriented Service

Reliable Versus Unreliable

Three Protocols



Note

The transport layer is responsible for process-to-process delivery.

Figure 12.1 *Types of data deliveries*

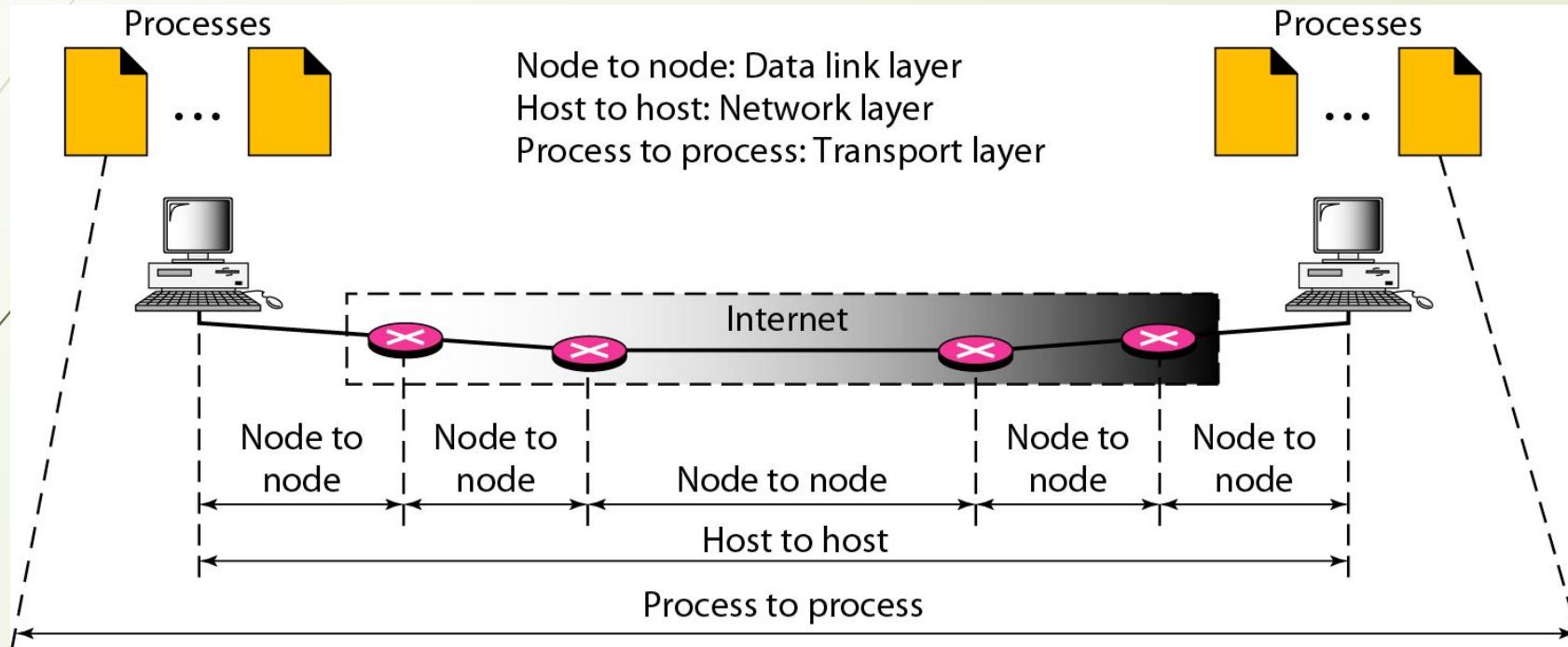


Figure 12.2 Port numbers

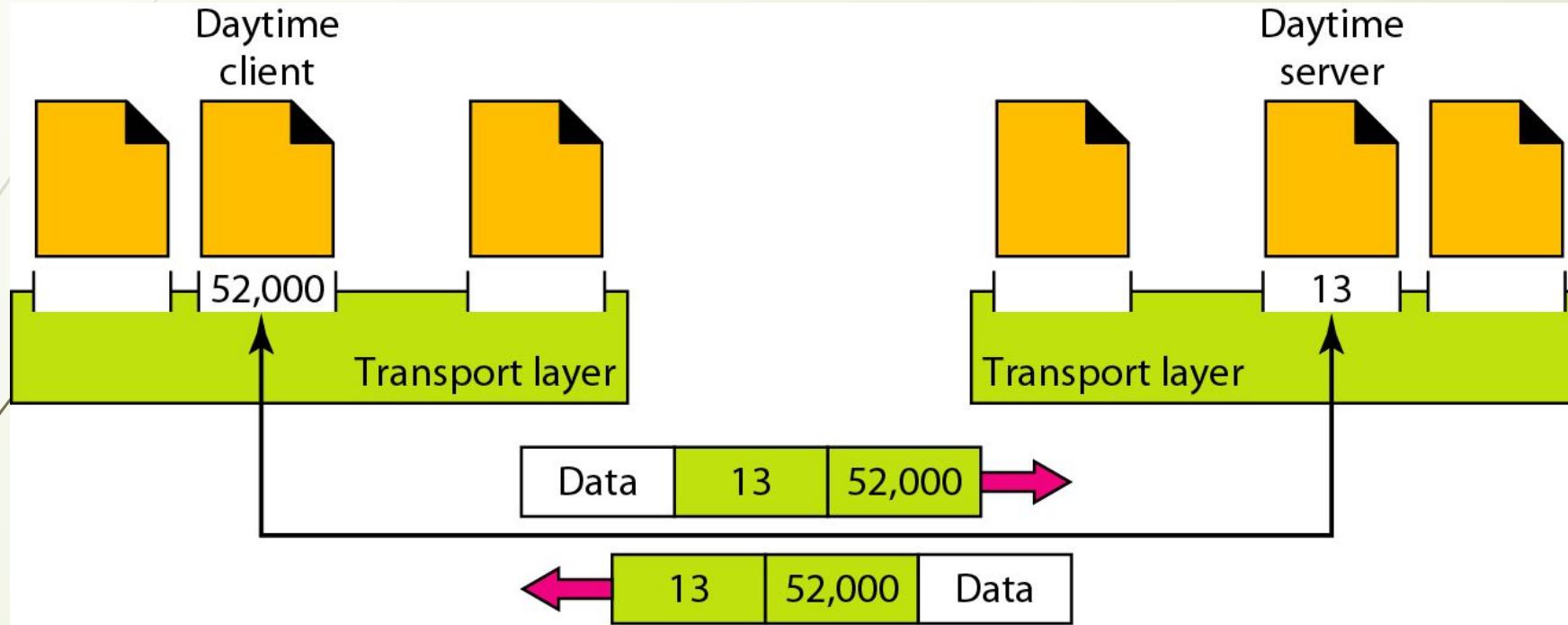


Figure 12.3 *IP addresses versus port numbers*

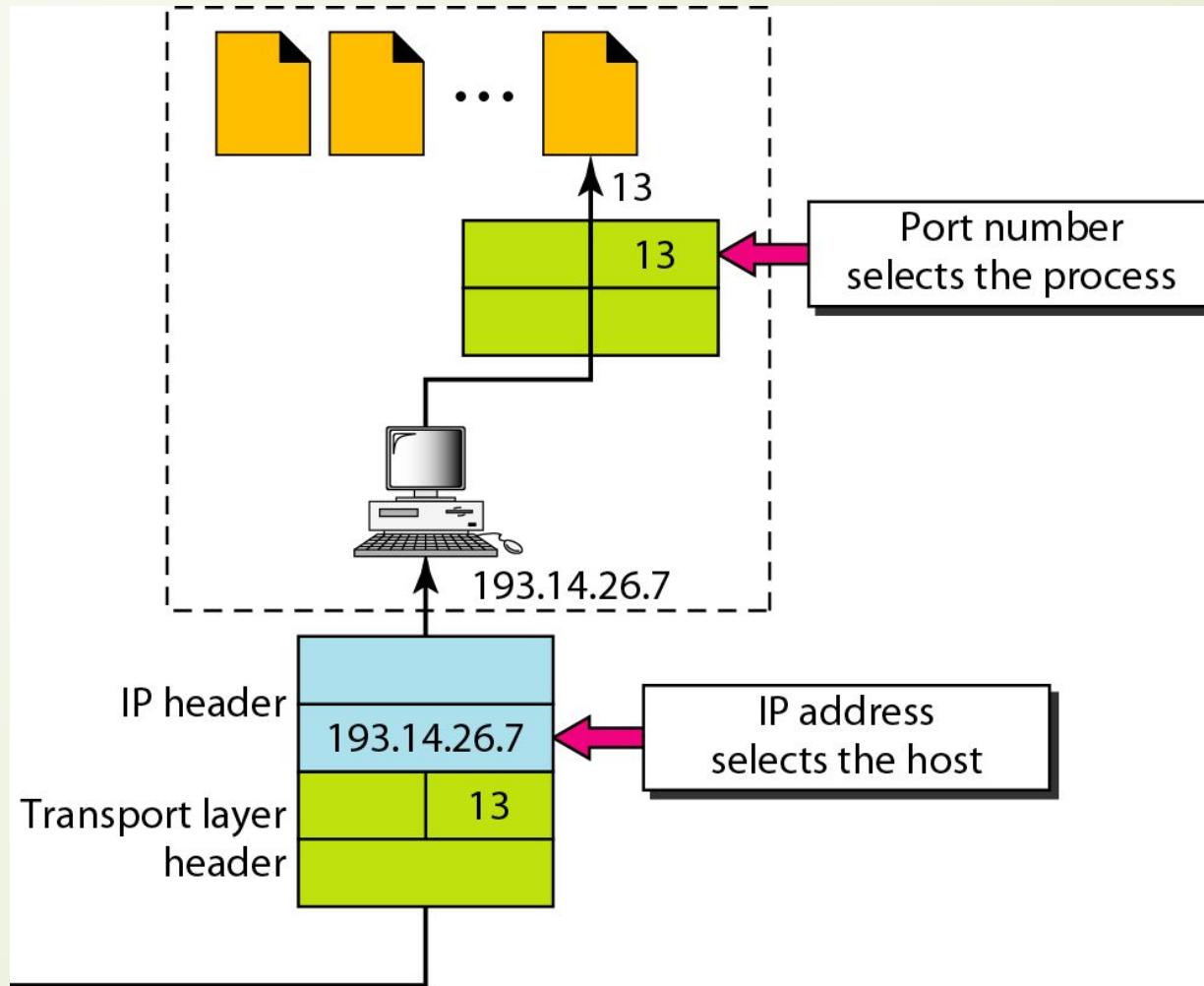


Figure 12.4 IANA ranges

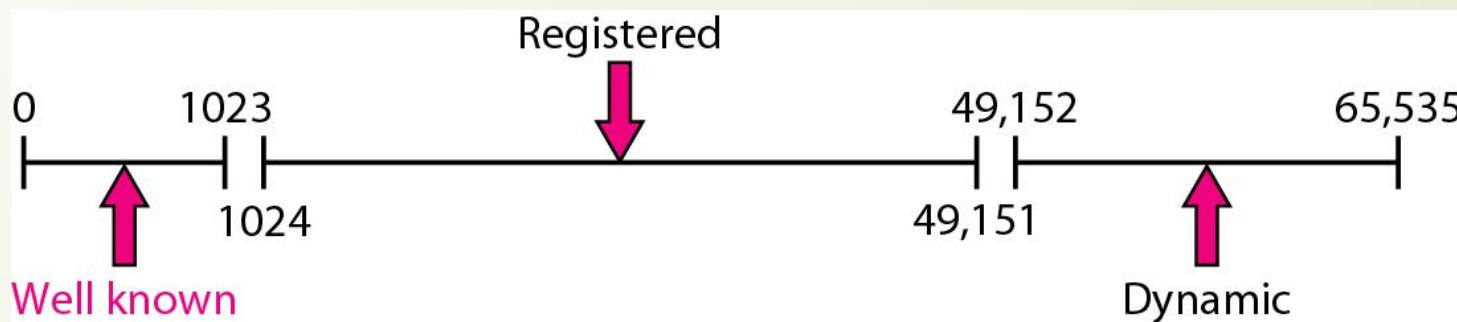


Figure 12.5 *Socket address*



Figure 12.6 Multiplexing and demultiplexing

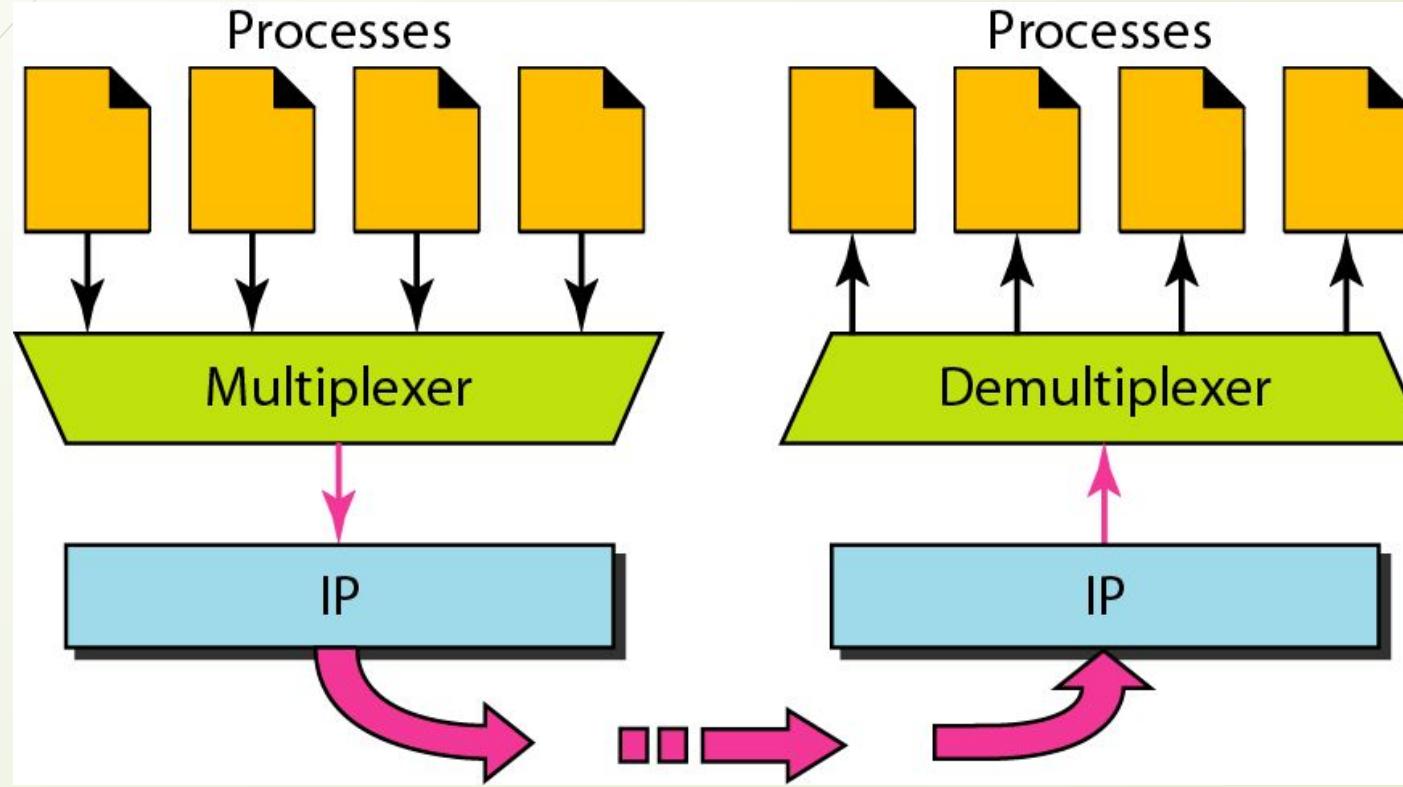


Figure 12.7 Error control

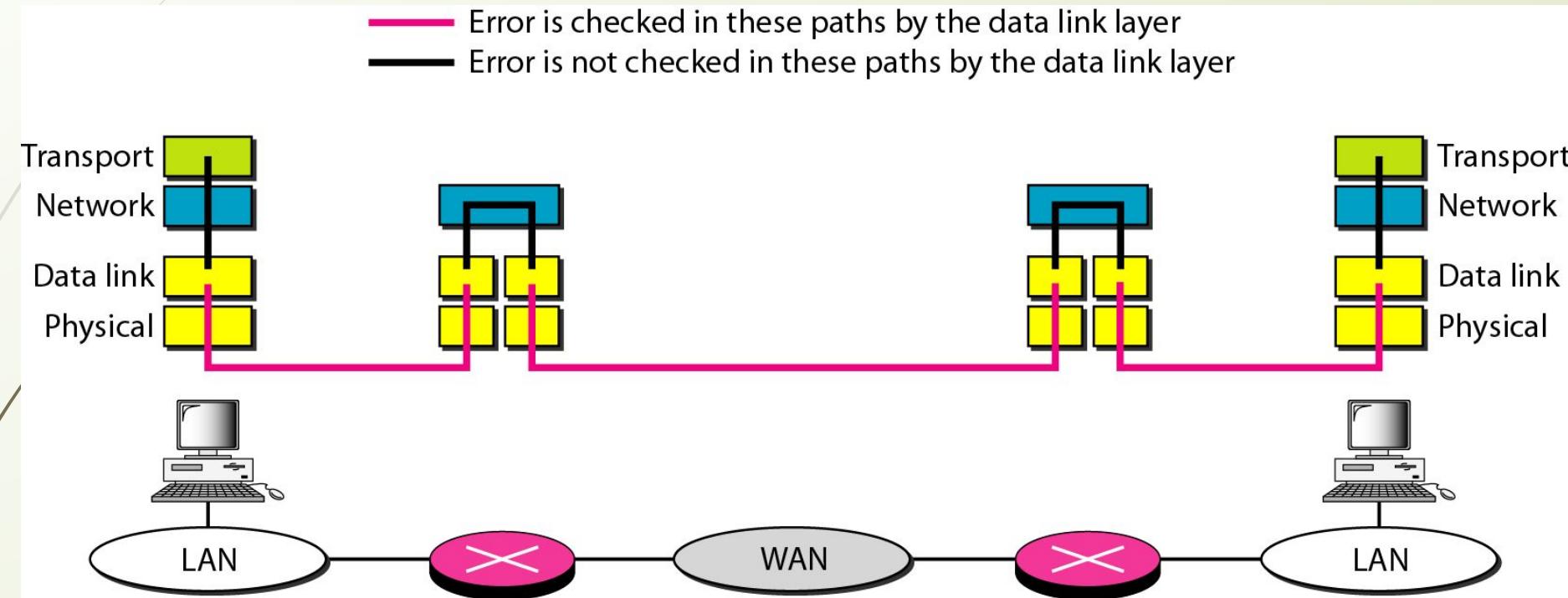
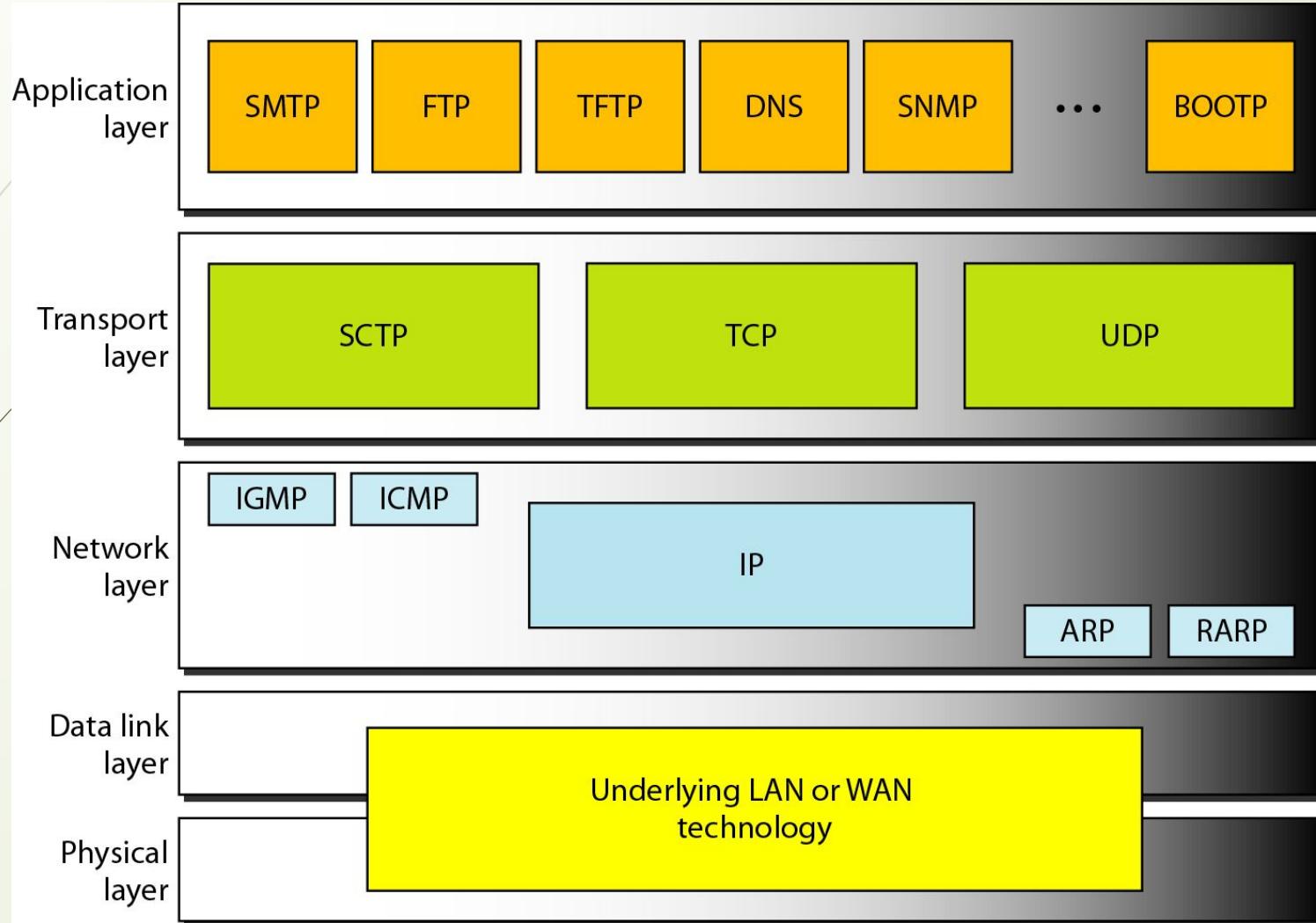


Figure 12.8 Position of UDP, TCP, and SCTP in TCP/IP suite



12-2 USER DATAGRAM PROTOCOL (UDP)

The User Datagram Protocol (UDP) is called a connectionless, unreliable transport protocol. It does not add anything to the services of IP except to provide process-to-process communication instead of host-to-host communication.

Topics discussed in this section:

Well-Known Ports for UDP

User Datagram

Checksum

UDP Operation

Use of UDP

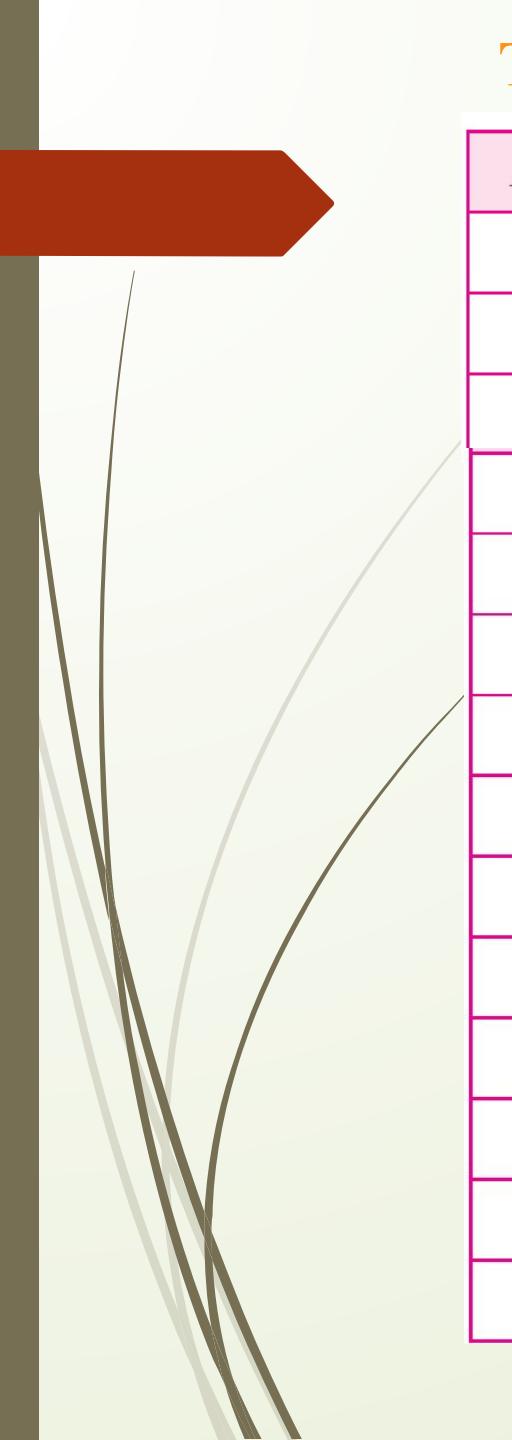
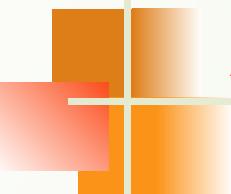


Table 12.1 *Well-known ports used with UDP*

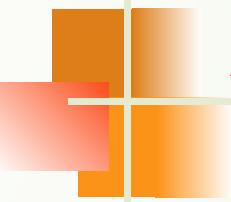
<i>Port</i>	<i>Protocol</i>	<i>Description</i>
7	Echo	Echoes a received datagram back to the sender
9	Discard	Discards any datagram that is received
11	Users	Active users
13	Daytime	Returns the date and the time
17	Quote	Returns a quote of the day
19	Chargen	Returns a string of characters
53	Nameserver	Domain Name Service
67	BOOTPs	Server port to download bootstrap information
68	BOOTPc	Client port to download bootstrap information
69	TFTP	Trivial File Transfer Protocol
111	RPC	Remote Procedure Call
123	NTP	Network Time Protocol
161	SNMP	Simple Network Management Protocol
162	SNMP	Simple Network Management Protocol (trap)



Example 12.1

In UNIX, the well-known ports are stored in a file called /etc/services. Each line in this file gives the name of the server and the well-known port number. We can use the grep utility to extract the line corresponding to the desired application. The following shows the port for FTP. Note that FTP can use port 21 with either UDP or TCP.

```
$ grep ftp /etc/services
ftp          21/tcp
ftp          21/udp
```



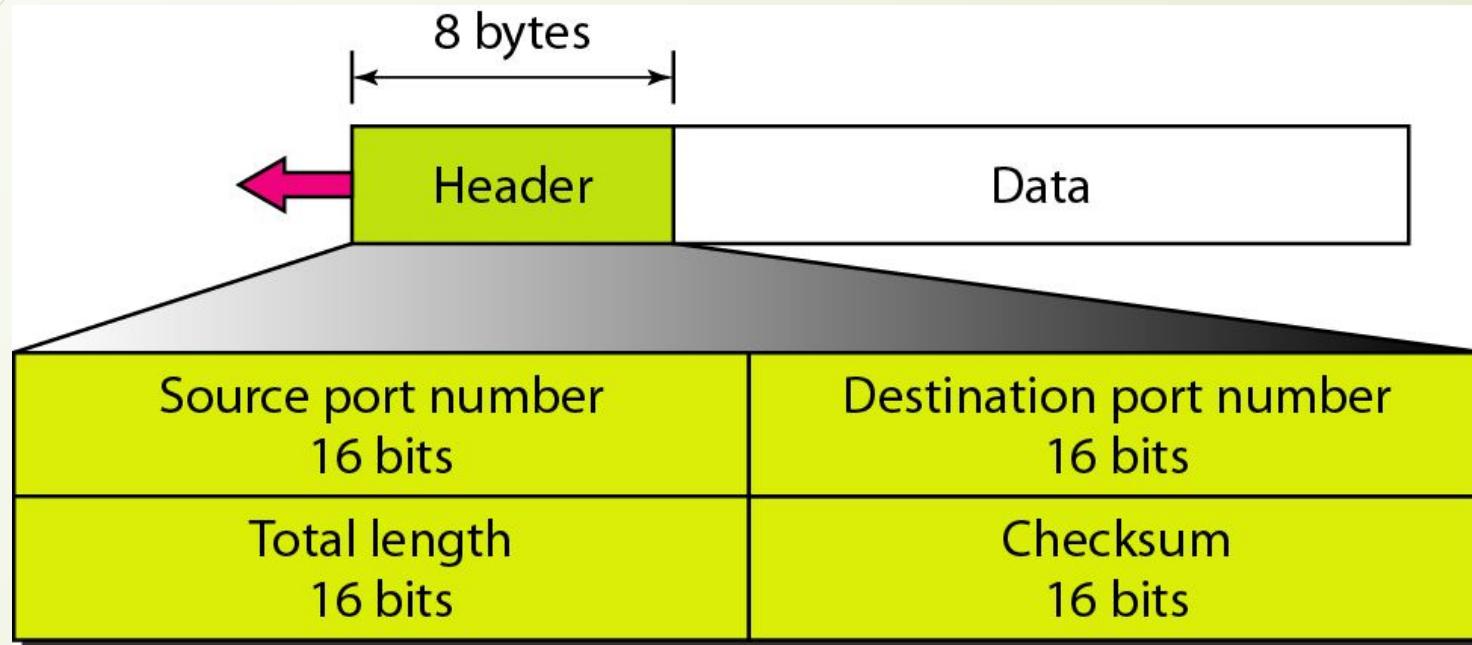
Example 12.1 (continued)

SNMP uses two port numbers (161 and 162), each for a different purpose, as we will see in Chapter 28.

```
$ grep snmp /etc/services
```

snmp	161/tcp	#Simple Net Mgmt Proto
snmp	161/udp	#Simple Net Mgmt Proto
snmptrap	162/udp	#Traps for SNMP

Figure 12.9 *User datagram format*

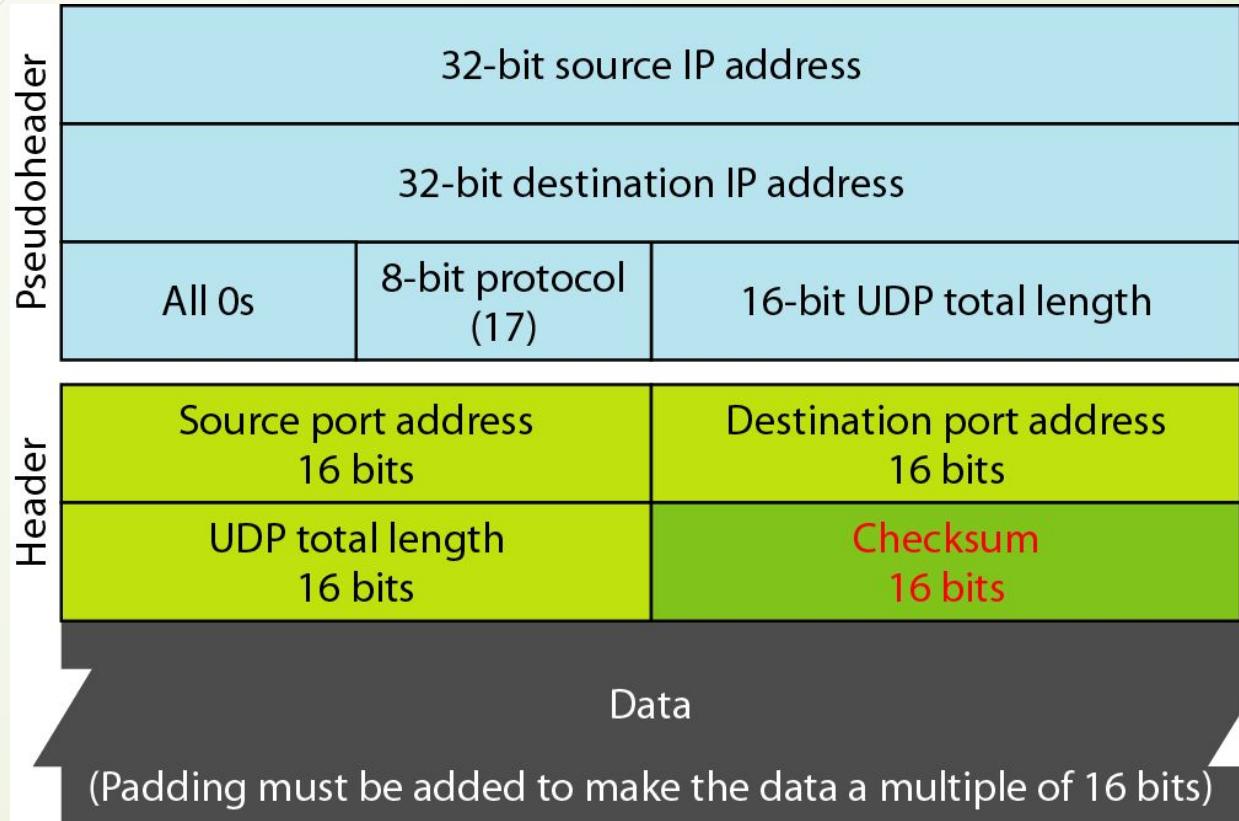


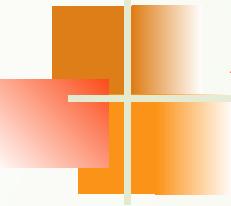


Note

UDP length
= IP length – IP header's length

Figure 12.10 Pseudoheader for checksum calculation

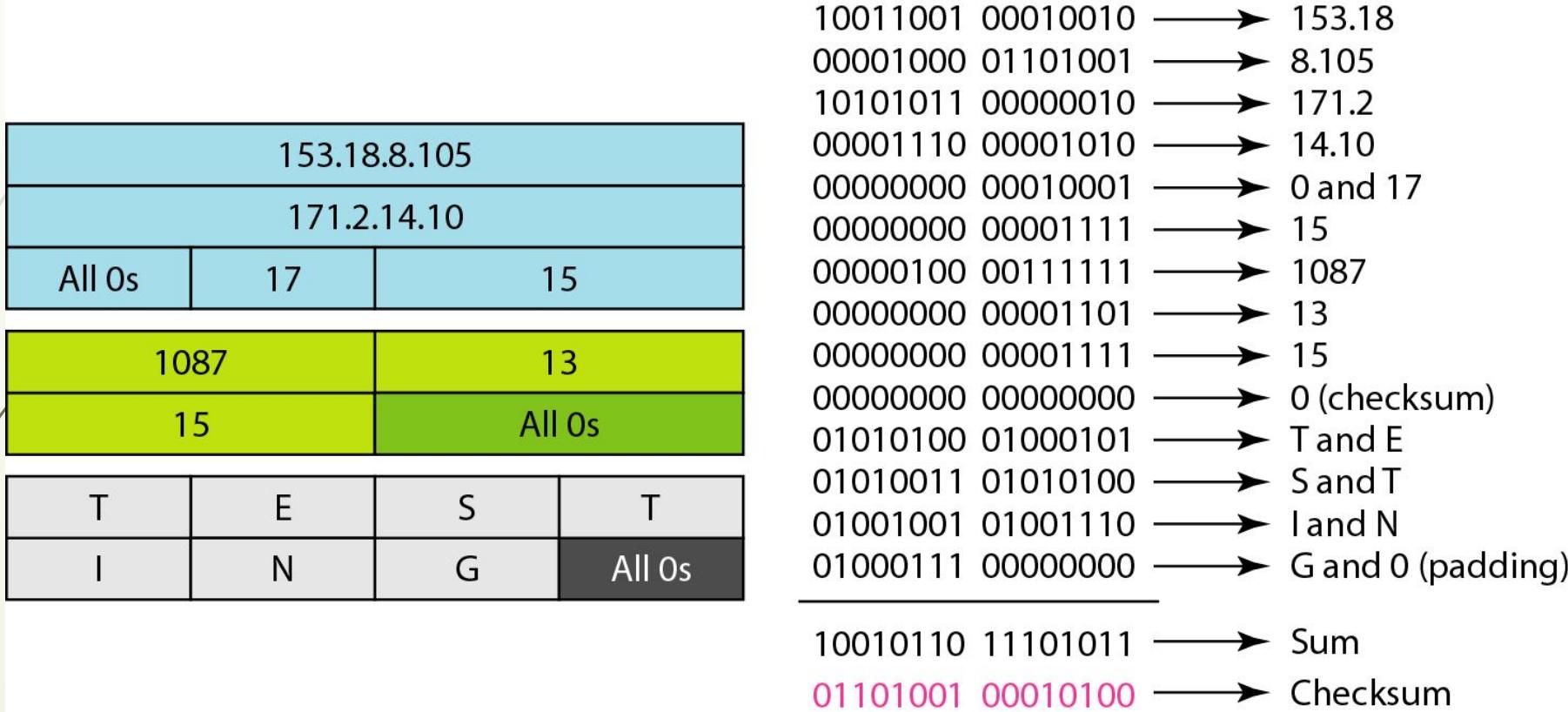




Example 12.2

Figure 12.11 shows the checksum calculation for a very small user datagram with only 7 bytes of data. Because the number of bytes of data is odd, padding is added for checksum calculation. The pseudoheader as well as the padding will be dropped when the user datagram is delivered to IP.

Figure 12.11 *Checksum calculation of a simple UDP user datagram*



TCP is a connection-oriented protocol; it creates a virtual connection between two TCPS to send data. In addition, TCP uses flow and error control mechanisms at the transport level.

Topics discussed in this section:

TCP Services

TCP Features

Segment

A TCP Connection

Flow Control

Error Control

Figure 12.13 *Stream delivery*

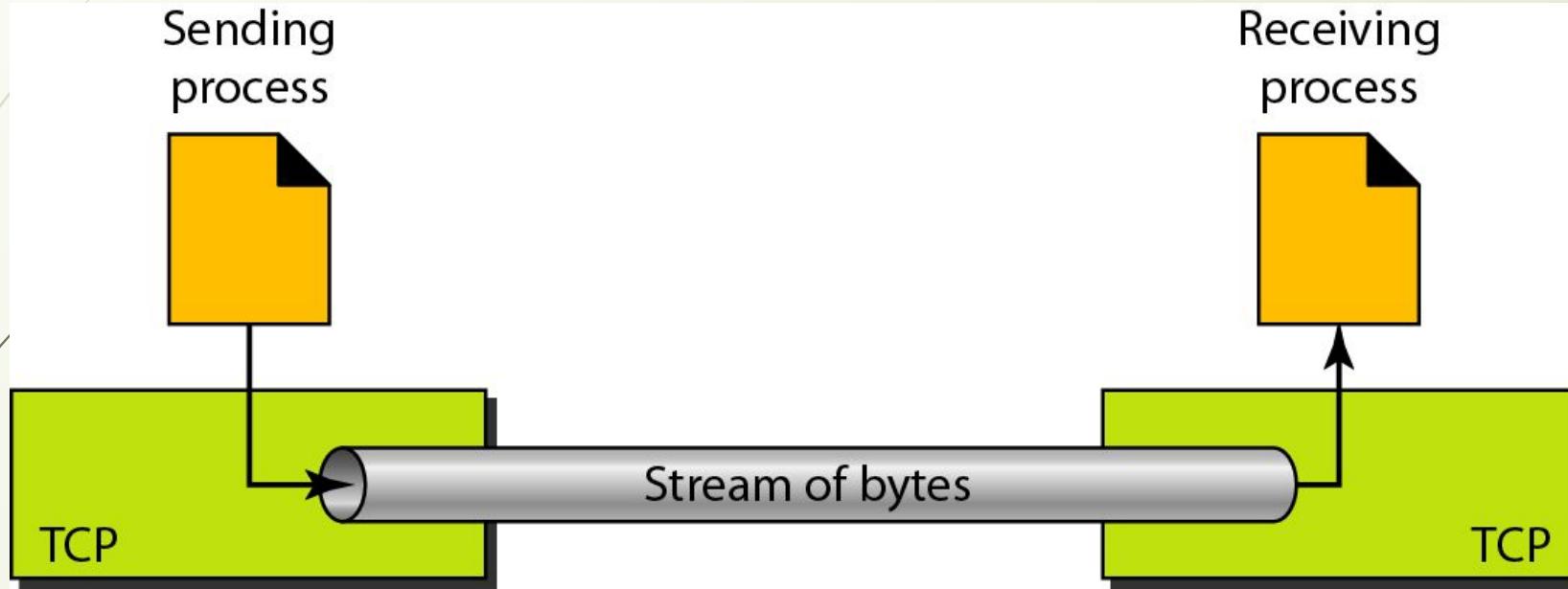


Figure 12.14 *Sending and receiving buffers*

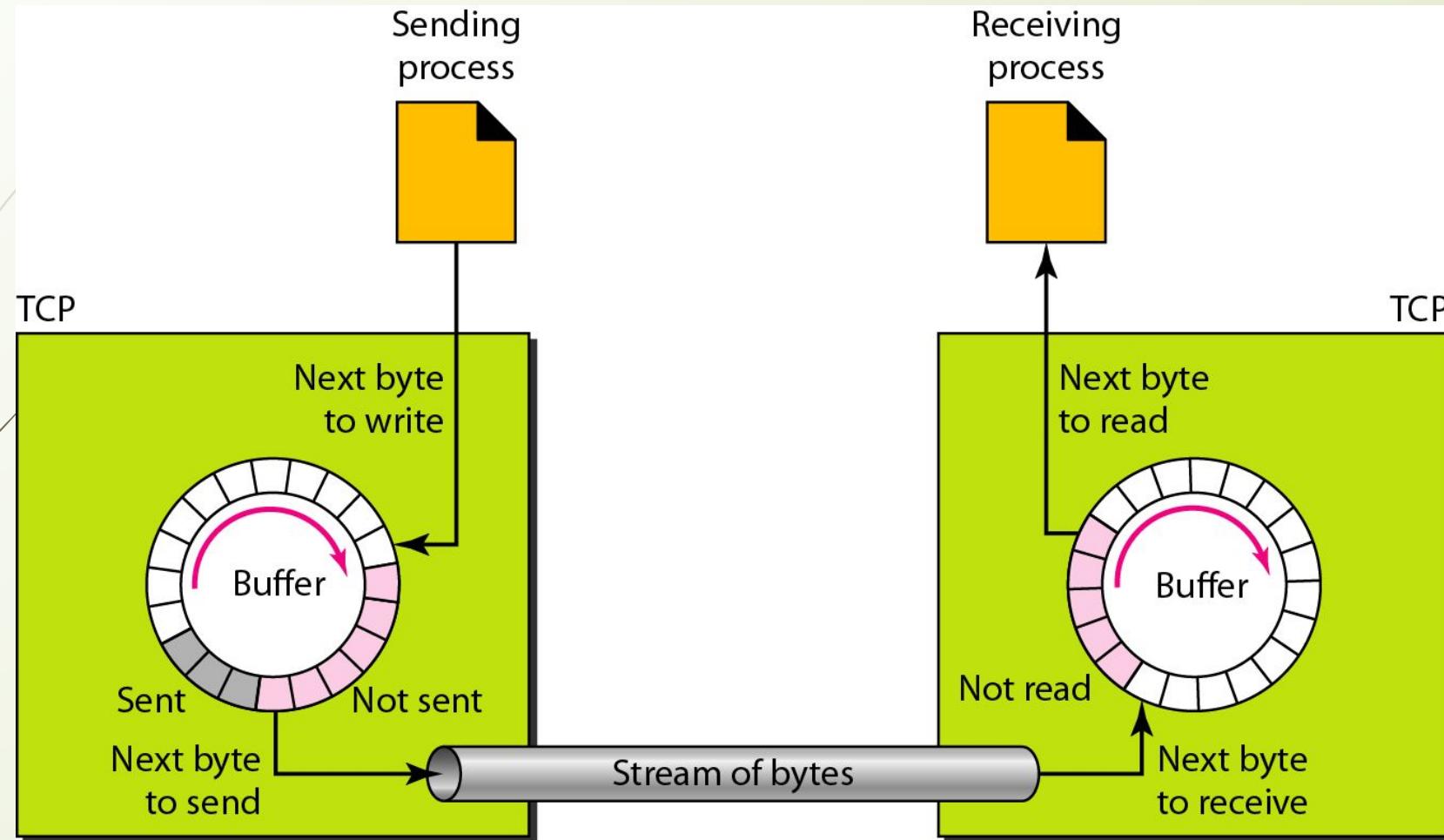
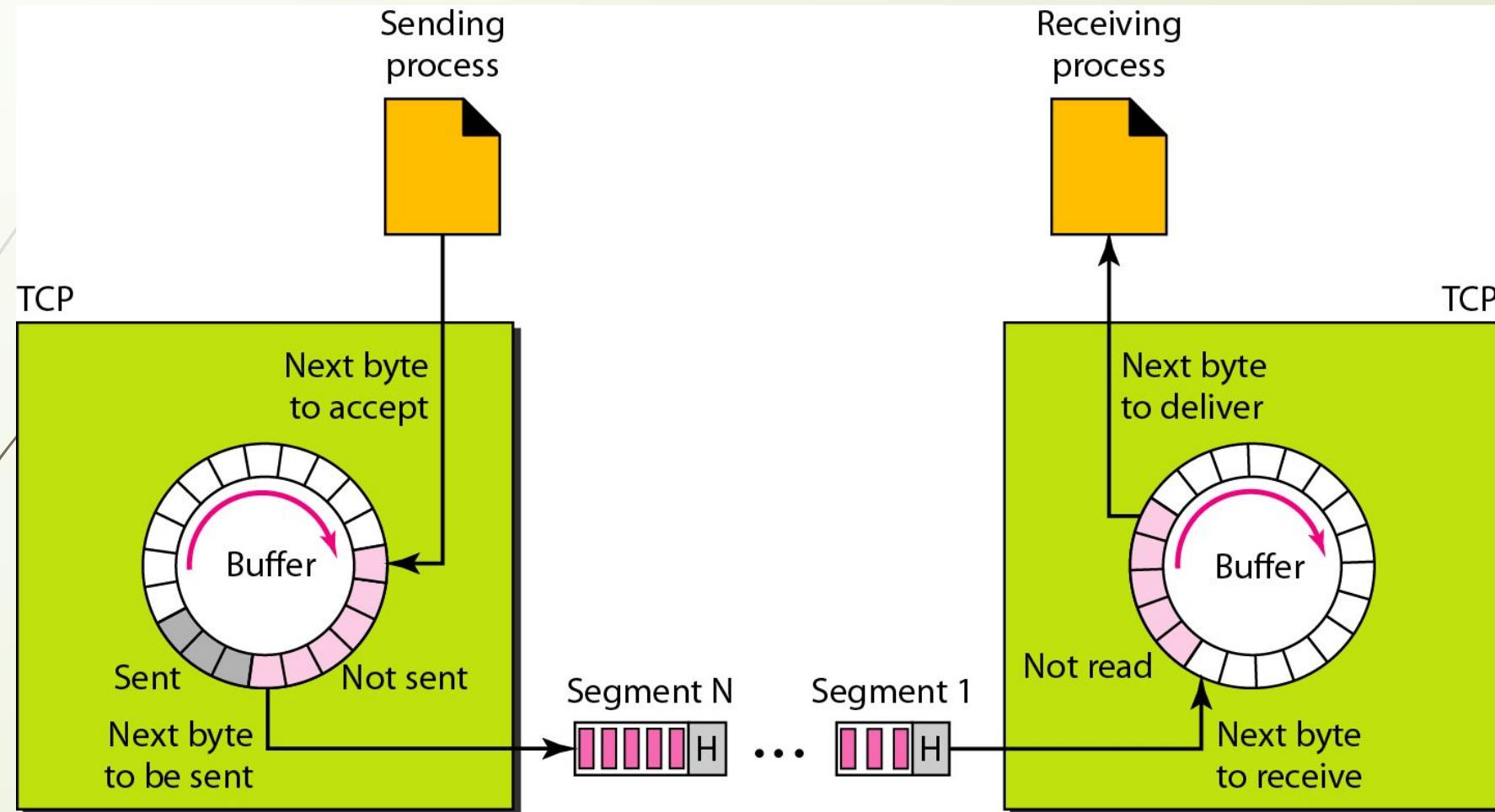
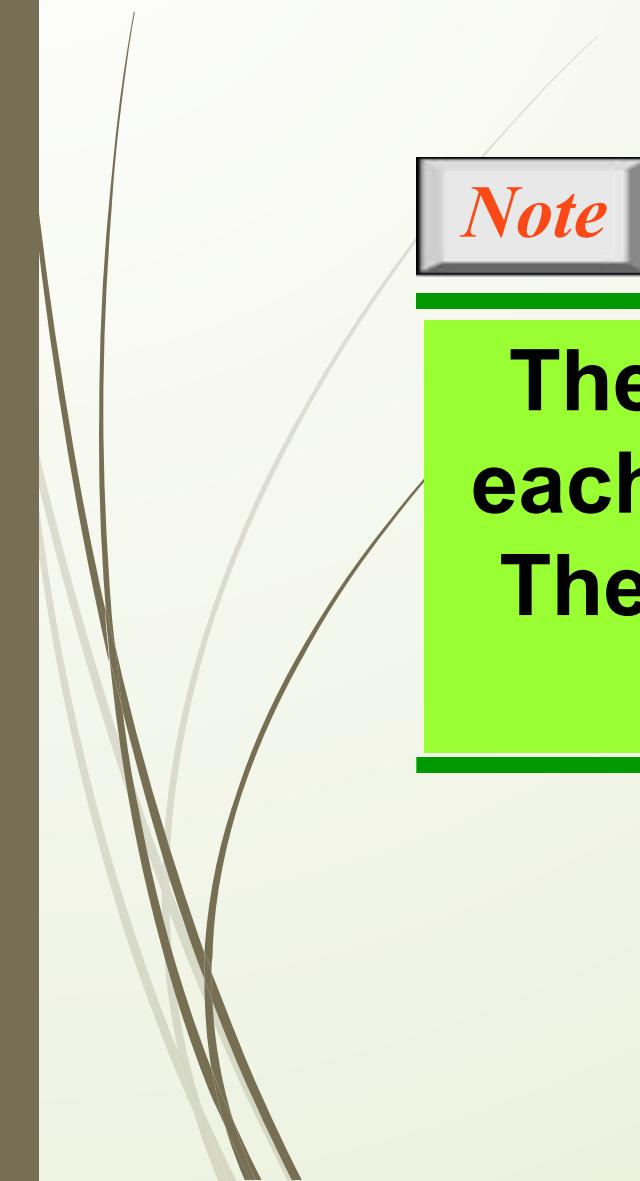


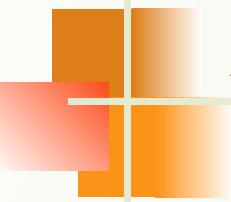
Figure 12.15 *TCP segments*





Note

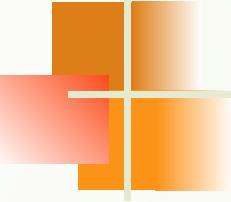
**The bytes of data being transferred in each connection are numbered by TCP.
The numbering starts with a randomly generated number.**



Example 12.3

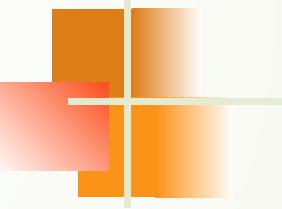
The following shows the sequence number for each segment:

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 of 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.**

Figure 12.16 TCP segment format

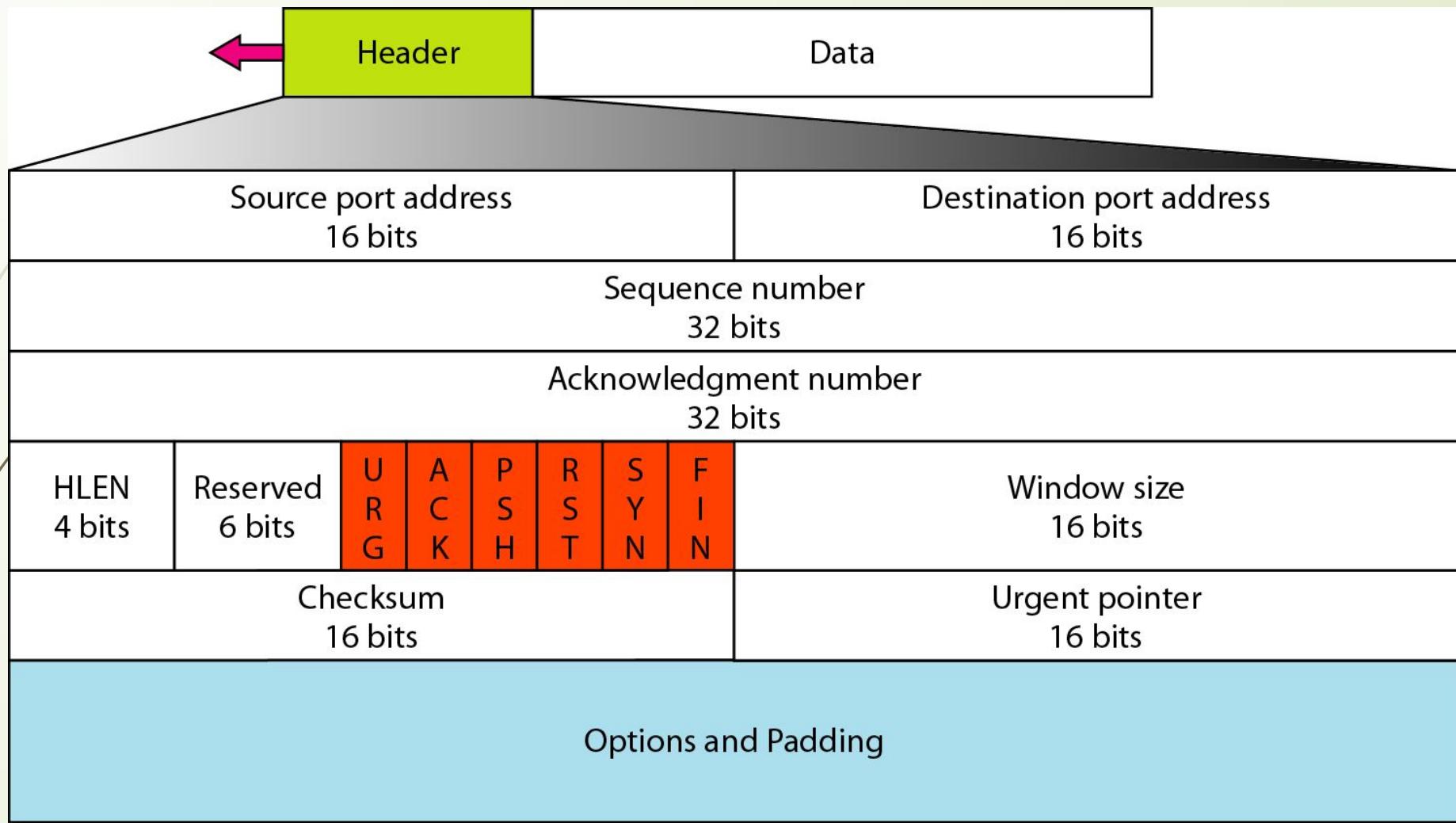


Figure 12.17 *Control field*

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

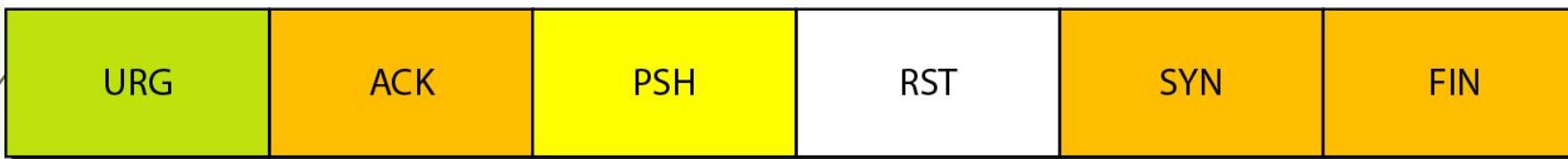
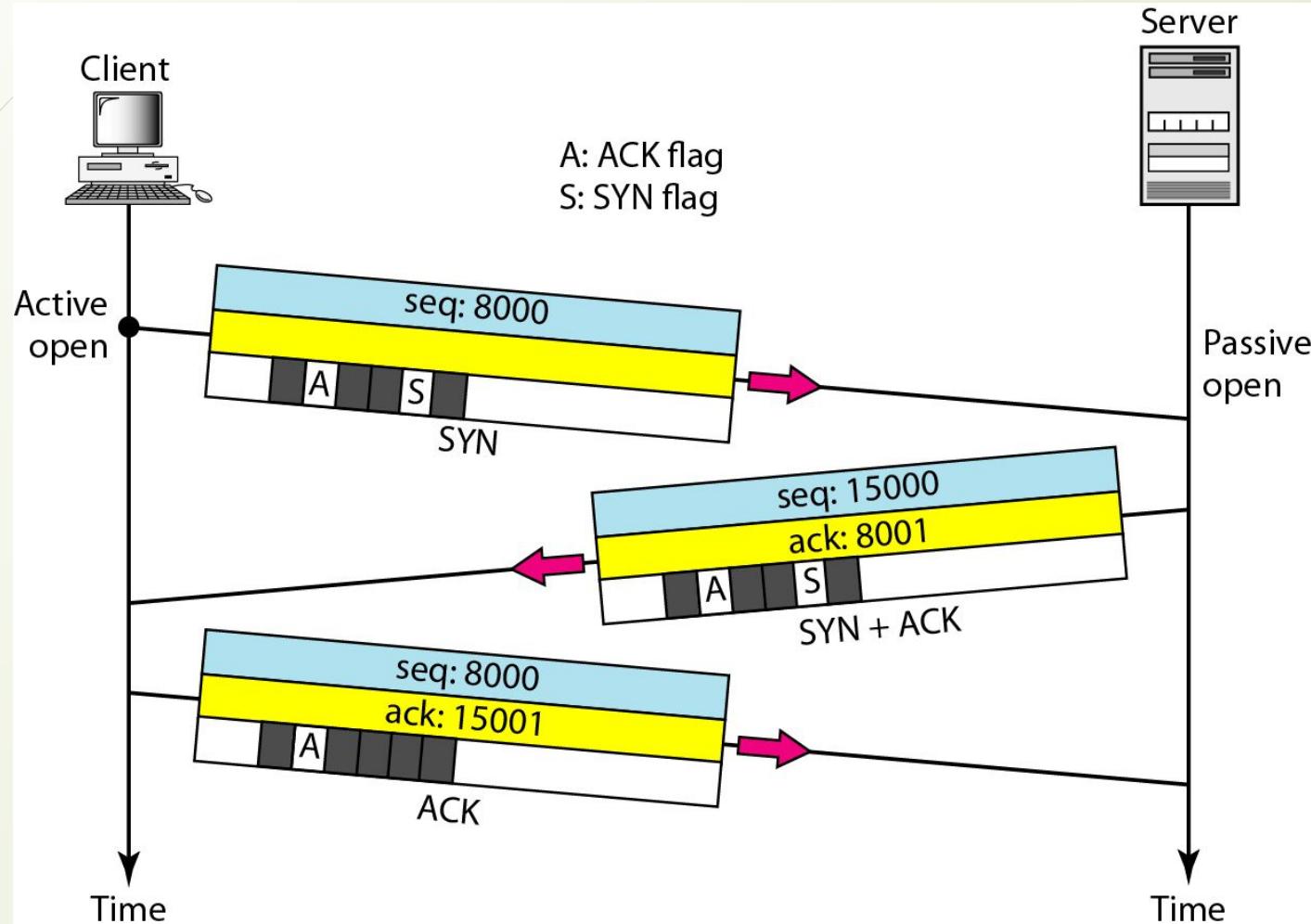


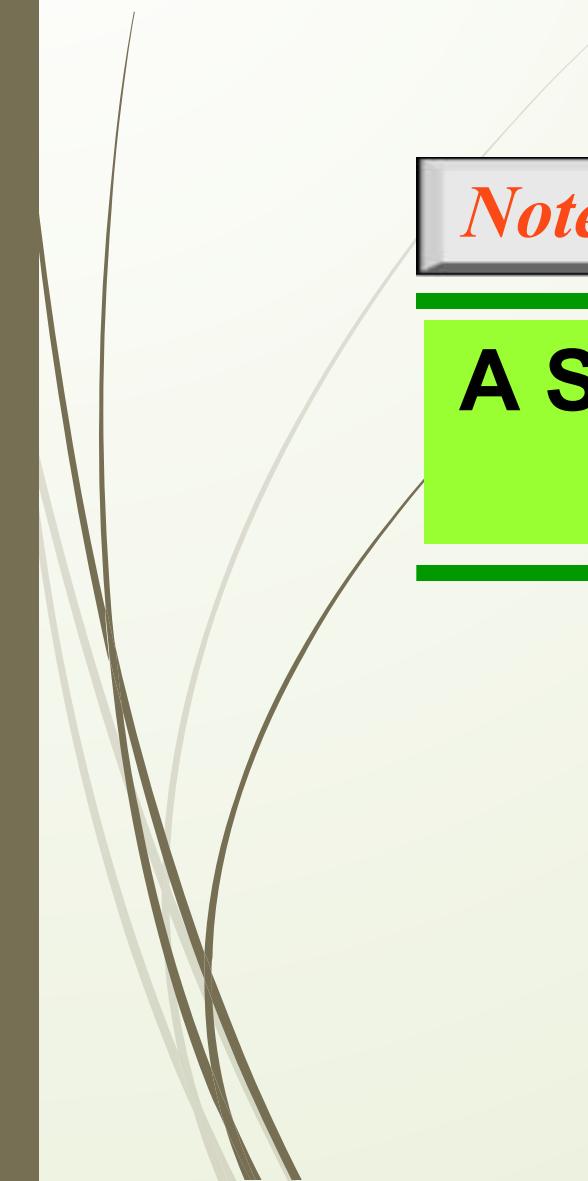


Table 12.3 *Description of flags in the control field*

<i>Flag</i>	<i>Description</i>
URG	The value of the urgent pointer field is valid.
ACK	The value of the acknowledgment field is valid.
PSH	Push the data.
RST	Reset the connection.
SYN	Synchronize sequence numbers during connection.
FIN	Terminate the connection.

Figure 12.18 Connection establishment using three-way handshaking





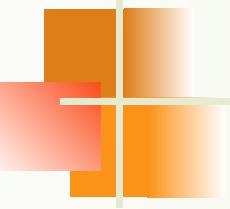
Note

A SYN segment cannot carry data, but it consumes one sequence number.



Note

A SYN + ACK segment cannot carry data, but does consume one sequence number.



Note

**An ACK segment, if carrying no data,
consumes no sequence number.**

Figure 12.19 Data transfer

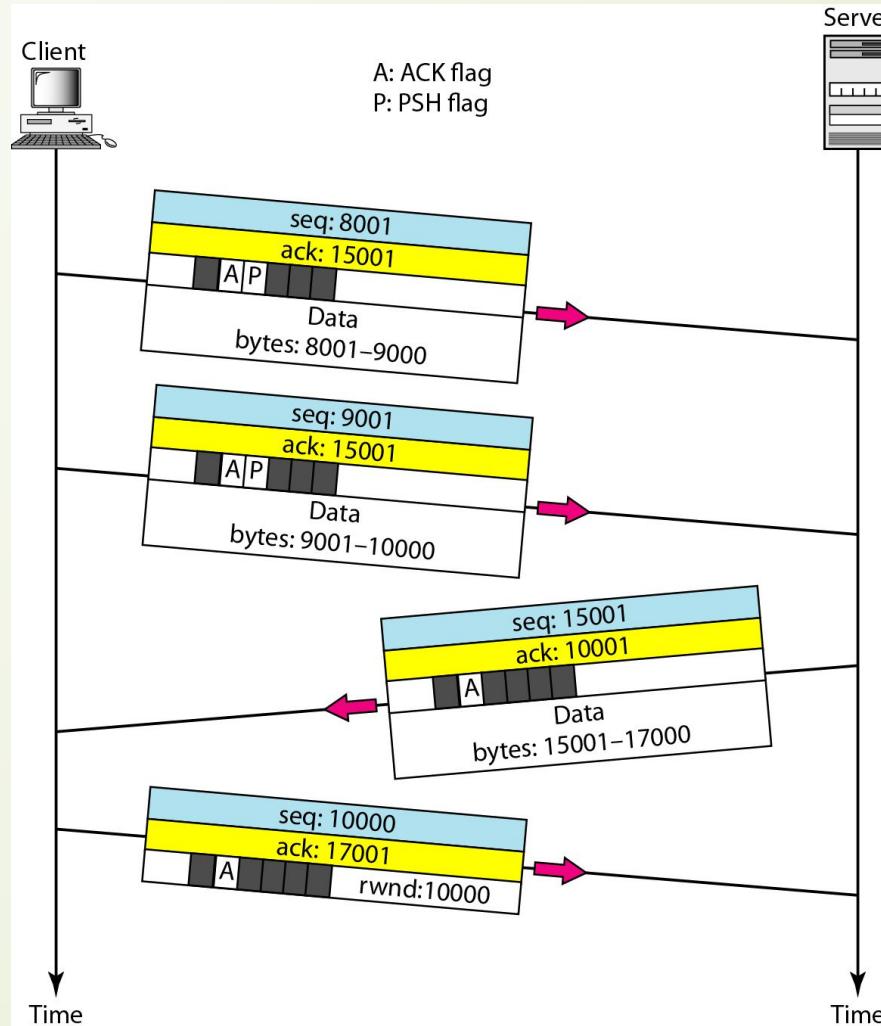
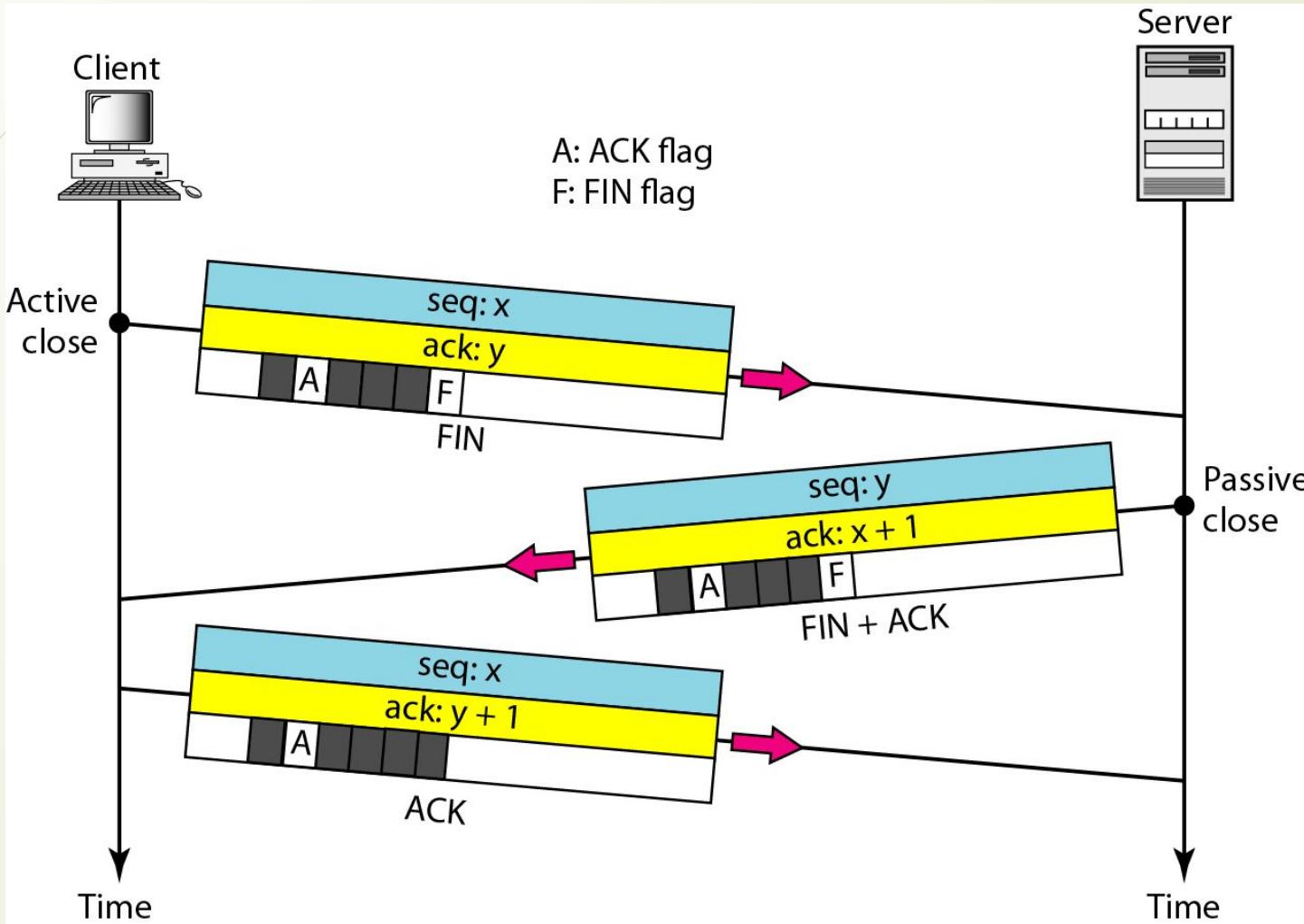
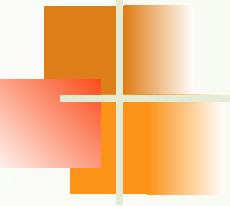


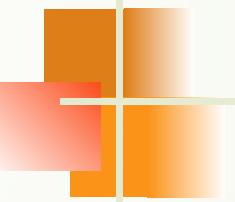
Figure 12.20 Connection termination using three-way handshaking





Note

The FIN segment consumes one sequence number if it does not carry data.



Note

**The FIN + ACK segment consumes
one sequence number if it
does not carry data.**

Figure 12.21 Half-close

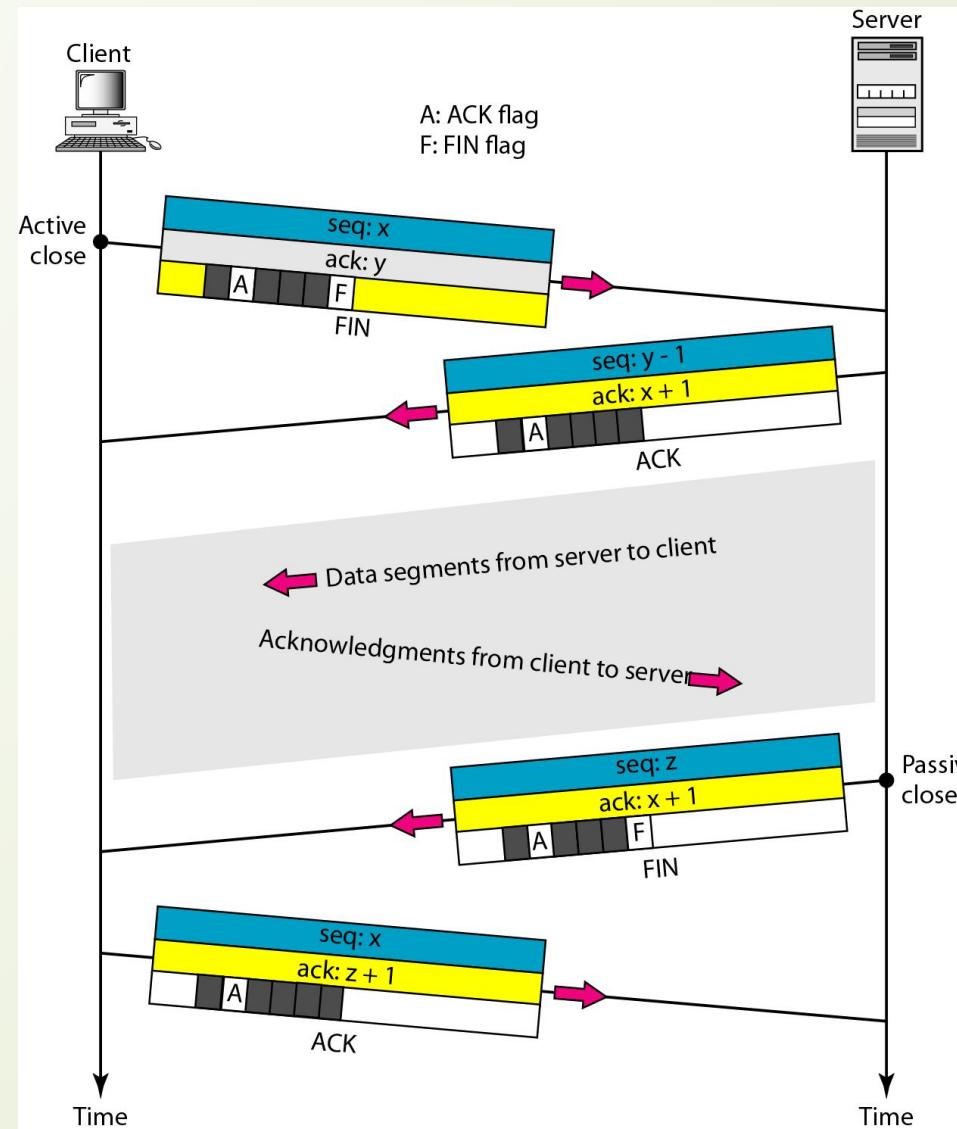
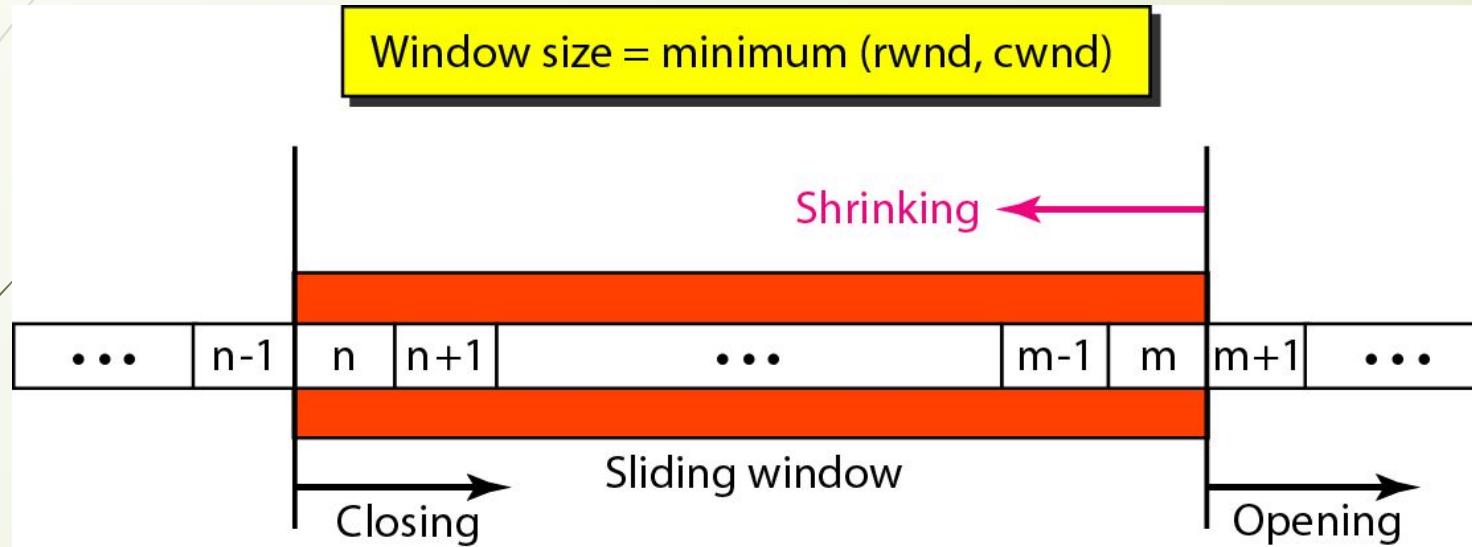
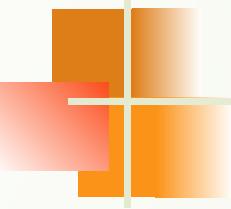


Figure 12.22 Sliding window

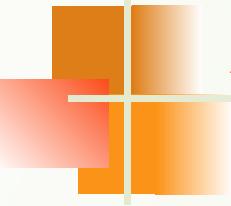




Note

A sliding window is used to make transmission more efficient as well as to control the flow of data so that the destination does not become overwhelmed with data.

TCP sliding windows are byte-oriented.

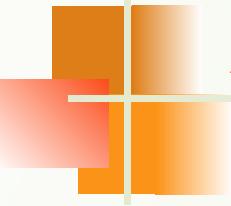


Example 12.4

What is the value of the receiver window (rwnd) for host A if the receiver, host B, has a buffer size of 5000 bytes and 1000 bytes of received and unprocessed data?

Solution

The value of rwnd = 5000 – 1000 = 4000. Host B can receive only 4000 bytes of data before overflowing its buffer. Host B advertises this value in its next segment to A.

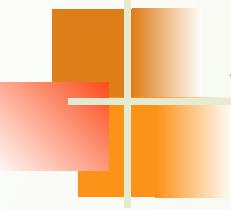


Example 12.5

What is the size of the window for host A if the value of rwnd is 3000 bytes and the value of cwnd is 3500 bytes?

Solution

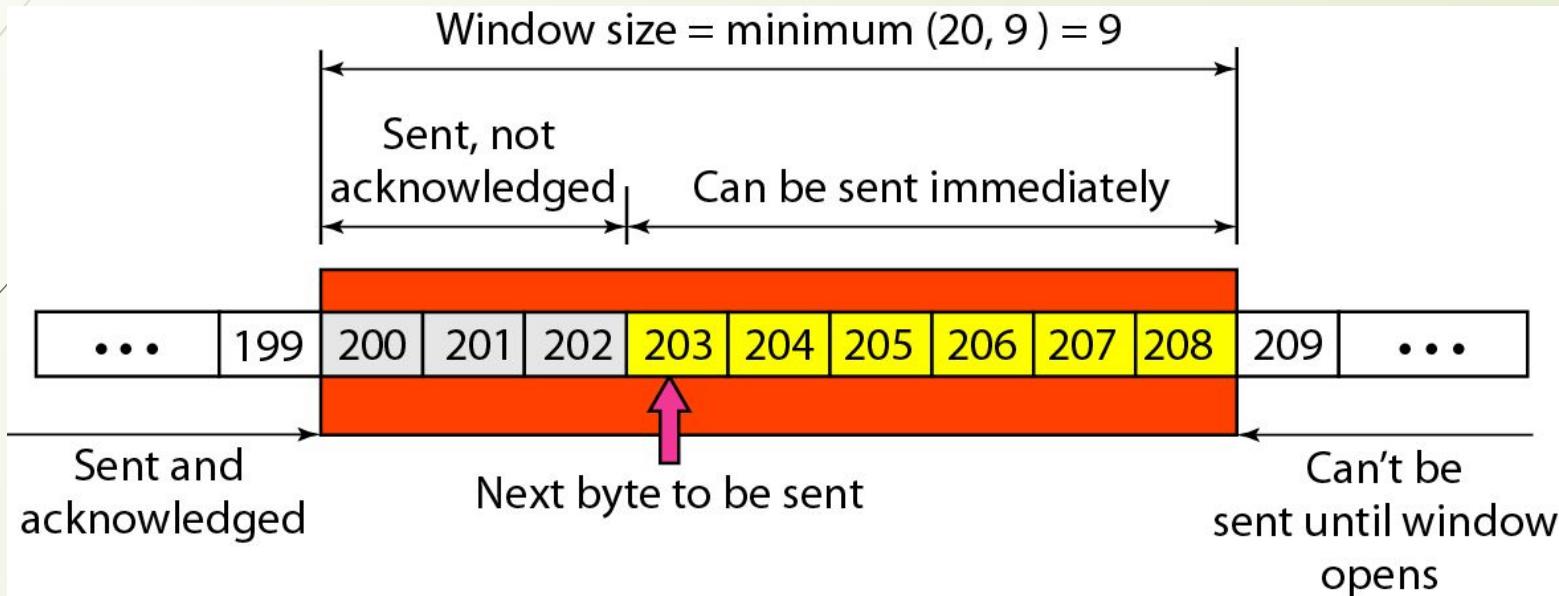
The size of the window is the smaller of rwnd and cwnd, which is 3000 bytes.



Example 12.6

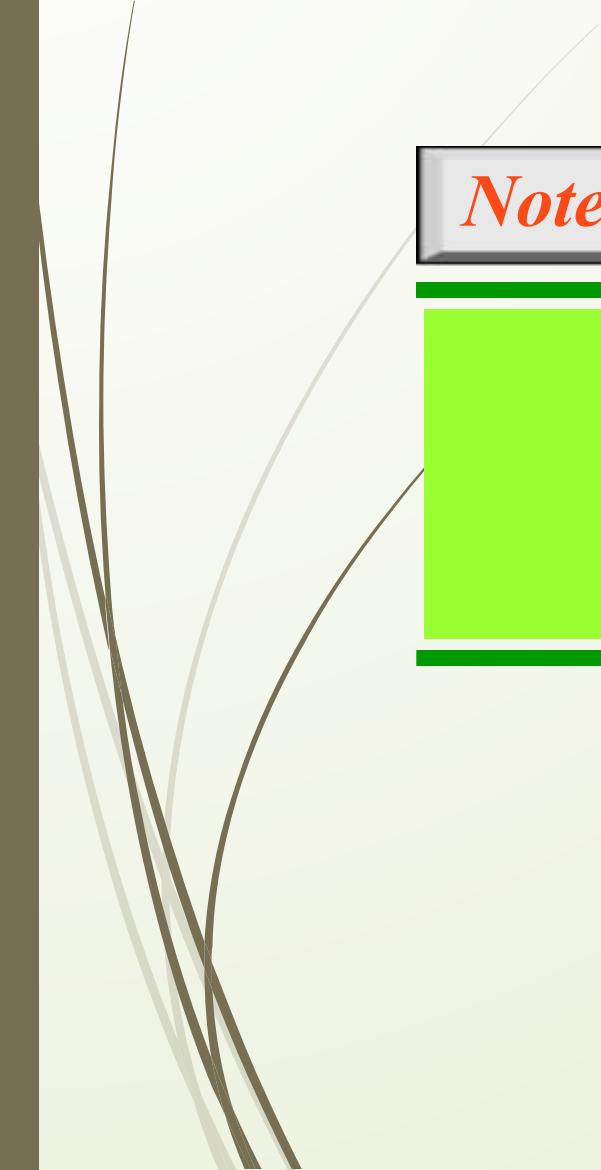
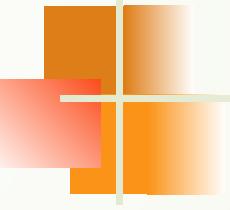
Figure 12.23 shows an unrealistic example of a sliding window. The sender has sent bytes up to 202. We assume that cwnd is 20 (in reality this value is thousands of bytes). The receiver has sent an acknowledgment number of 200 with an rwnd of 9 bytes (in reality this value is thousands of bytes). The size of the sender window is the minimum of rwnd and cwnd, or 9 bytes. Bytes 200 to 202 are sent, but not acknowledged. Bytes 203 to 208 can be sent without worrying about acknowledgment. Bytes 209 and above cannot be sent.

Figure 12.23 Example 12.6



Some points about TCP sliding windows:

- ❑ The size of the window is the lesser of rwnd and cwnd.**
- ❑ The source does not have to send a full window's worth of data.**
- ❑ The window can be opened or closed by the receiver, but should not be shrunk.**
- ❑ The destination can send an acknowledgment at any time as long as it does not result in a shrinking window.**
- ❑ The receiver can temporarily shut down the window; the sender, however, can always send a segment of 1 byte after the window is shut down.**



Note

ACK segments do not consume sequence numbers and are not acknowledged.



Note

In modern implementations, a retransmission occurs if the retransmission timer expires or three duplicate ACK segments have arrived.



Note

No retransmission timer is set for an ACK segment.



Note

Data may arrive out of order and be temporarily stored by the receiving TCP, but TCP guarantees that no out-of-order segment is delivered to the process.

Figure 12.24 Normal operation

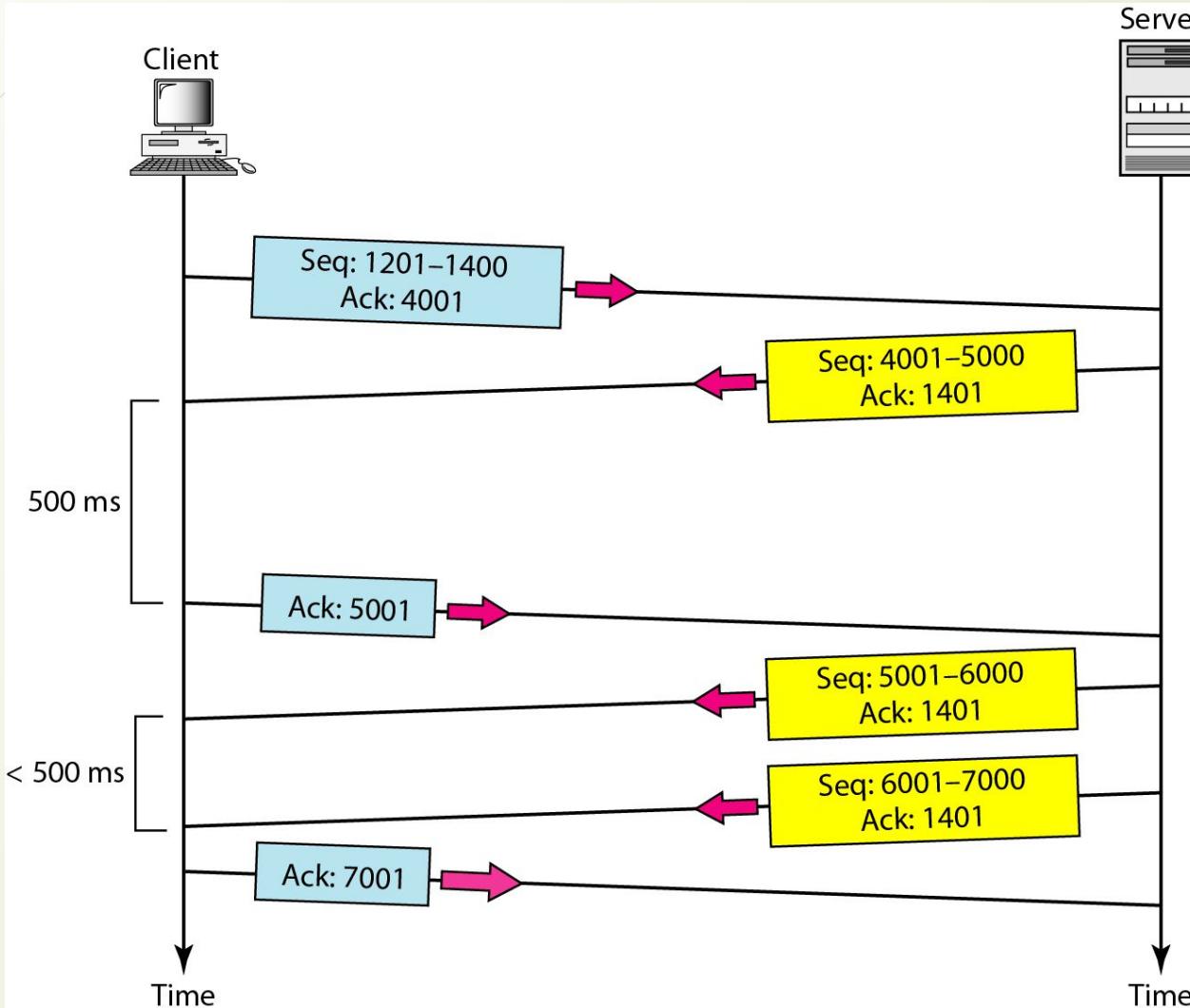
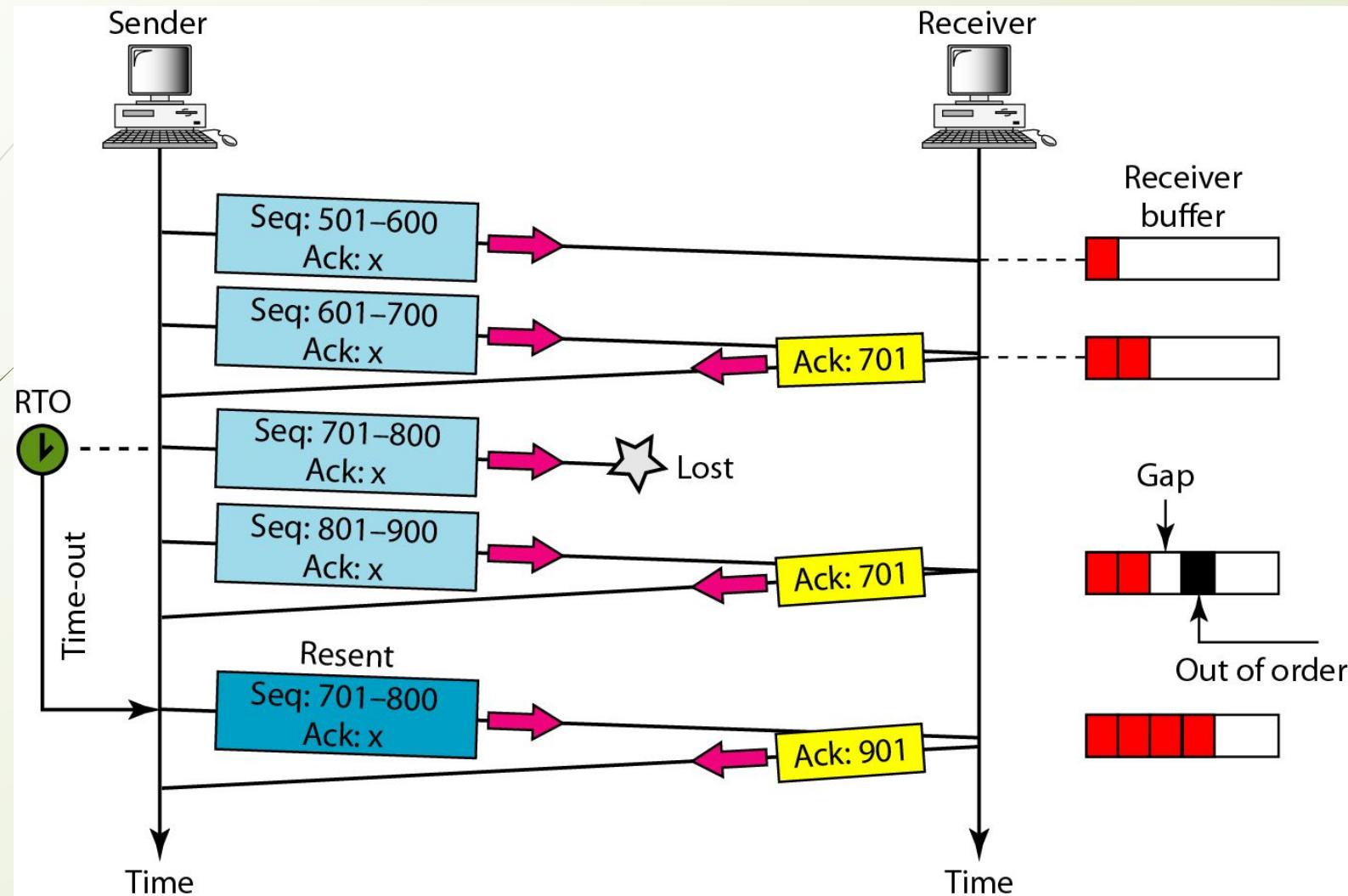
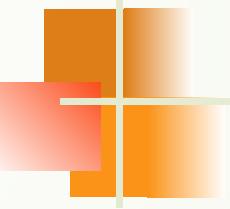


Figure 12.25 Lost segment





Note

The receiver TCP delivers only ordered data to the process.

Stream Control Transmission Protocol (SCTP) is a new reliable, message-oriented transport layer protocol. SCTP, however, is mostly designed for Internet applications that have recently been introduced. These new applications need a more sophisticated service than TCP can provide.

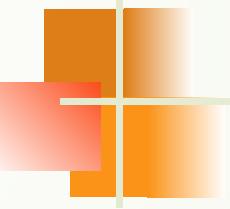
Topics discussed in this section:

SCTP Services and Features

Packet Format

An SCTP Association

Flow Control and Error Control



Note

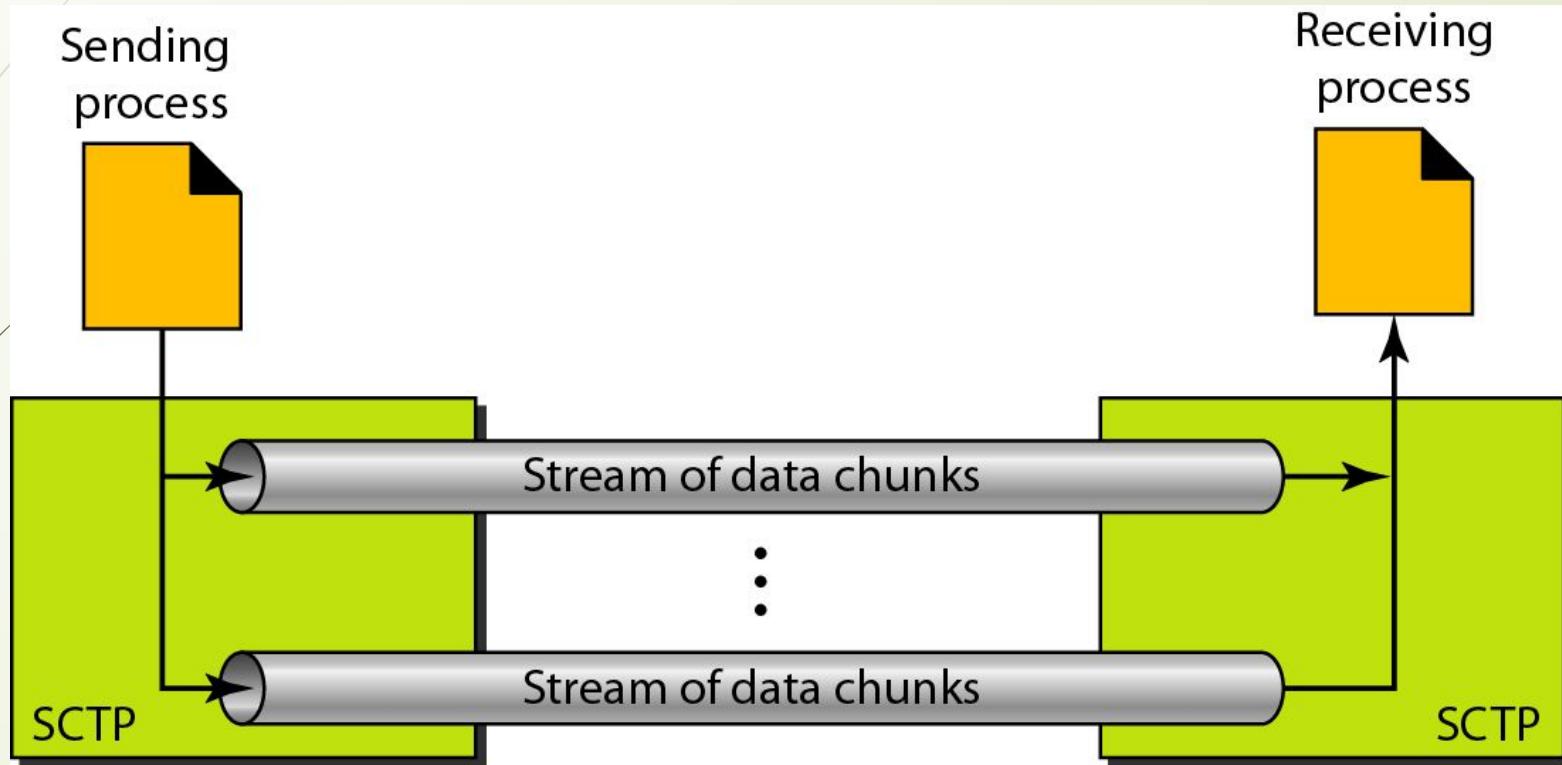
SCTP is a message-oriented, reliable protocol that combines the best features of UDP and TCP.

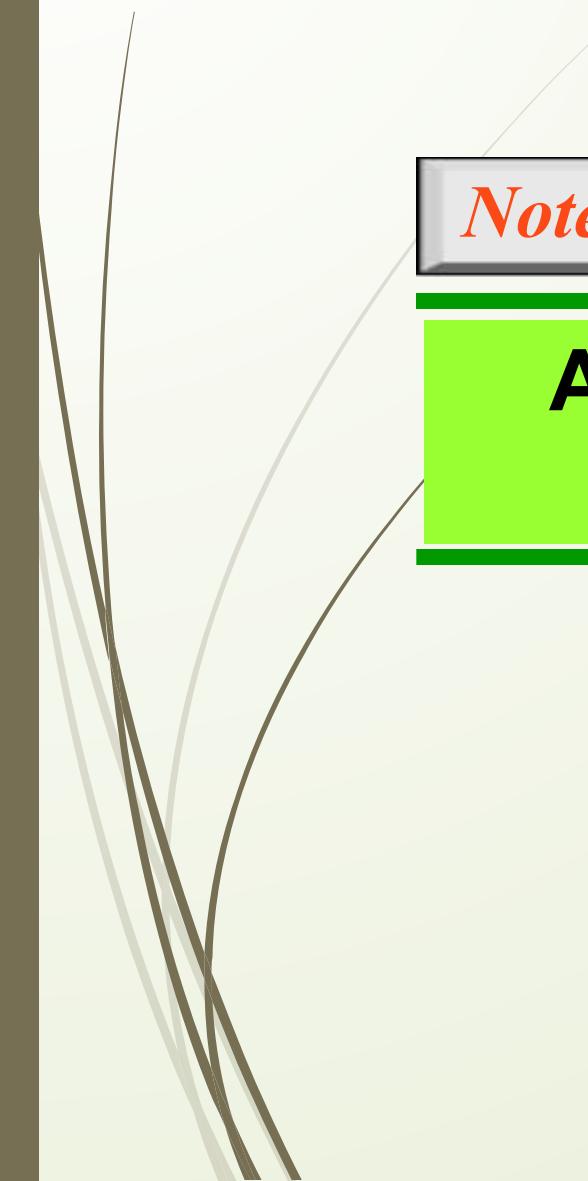


Table 12.4 Some SCTP applications

<i>Protocol</i>	<i>Port Number</i>	<i>Description</i>
IUA	9990	ISDN over IP
M2UA	2904	SS7 telephony signaling
M3UA	2905	SS7 telephony signaling
H.248	2945	Media gateway control
H.323	1718, 1719, 1720, 11720	IP telephony
SIP	5060	IP telephony

Figure 12.27 *Multiple-stream concept*

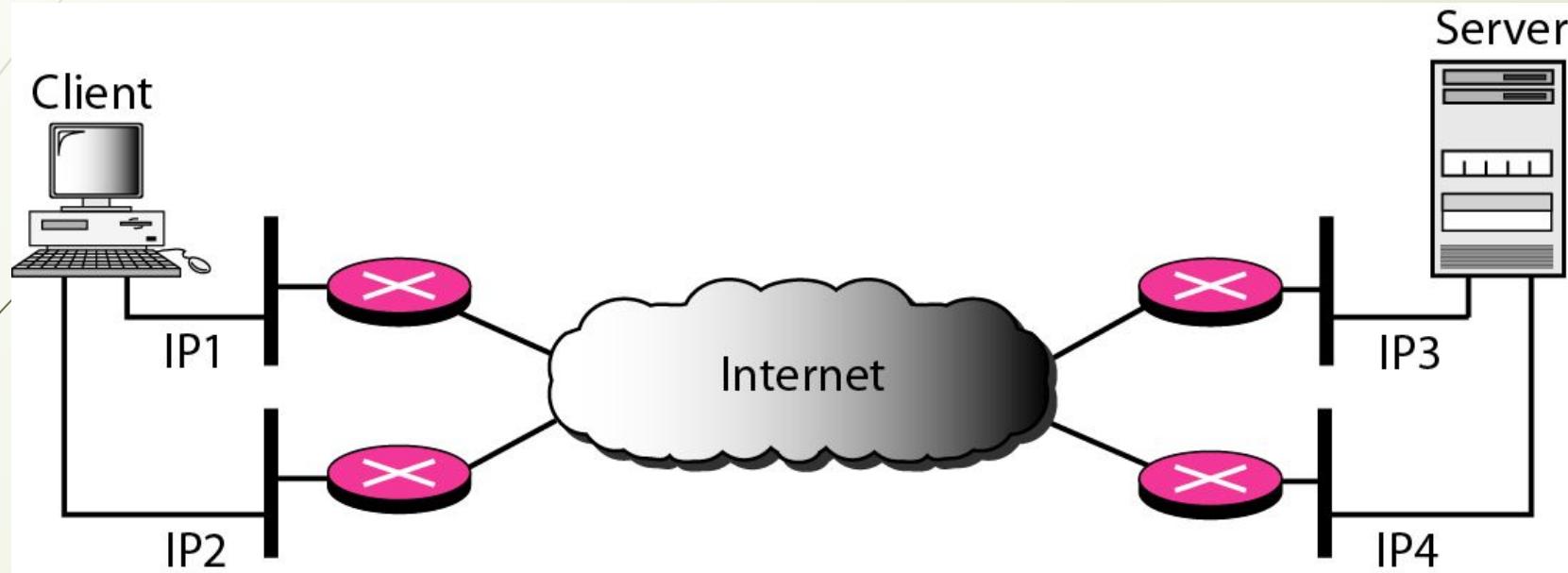


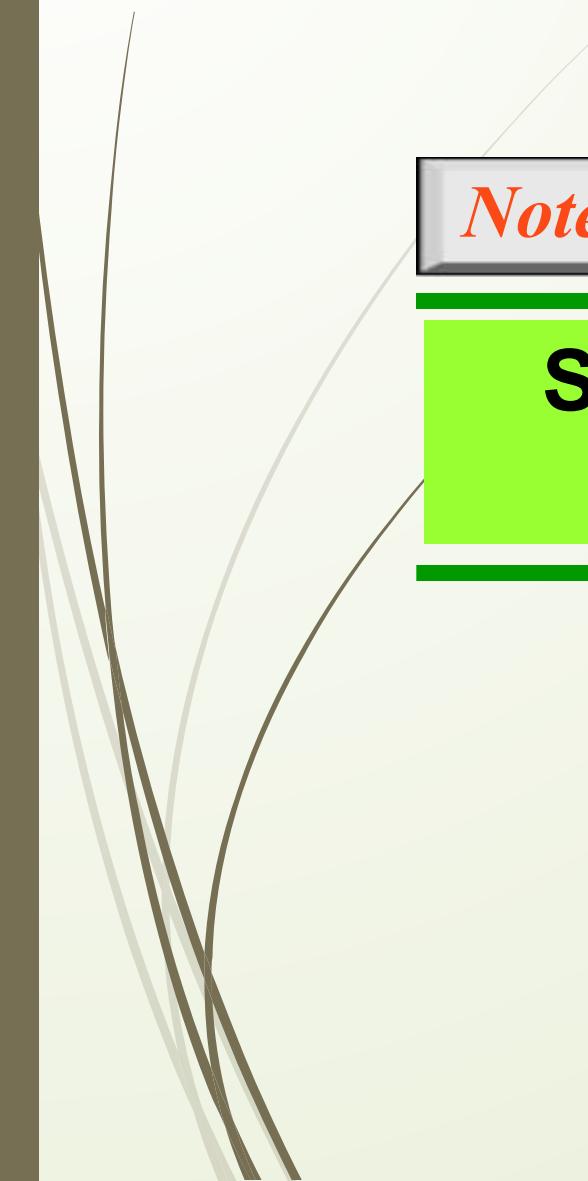


Note

An association in SCTP can involve multiple streams.

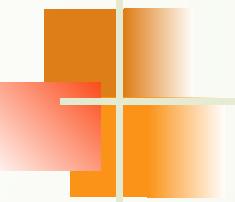
Figure 12.28 *Multihoming concept*





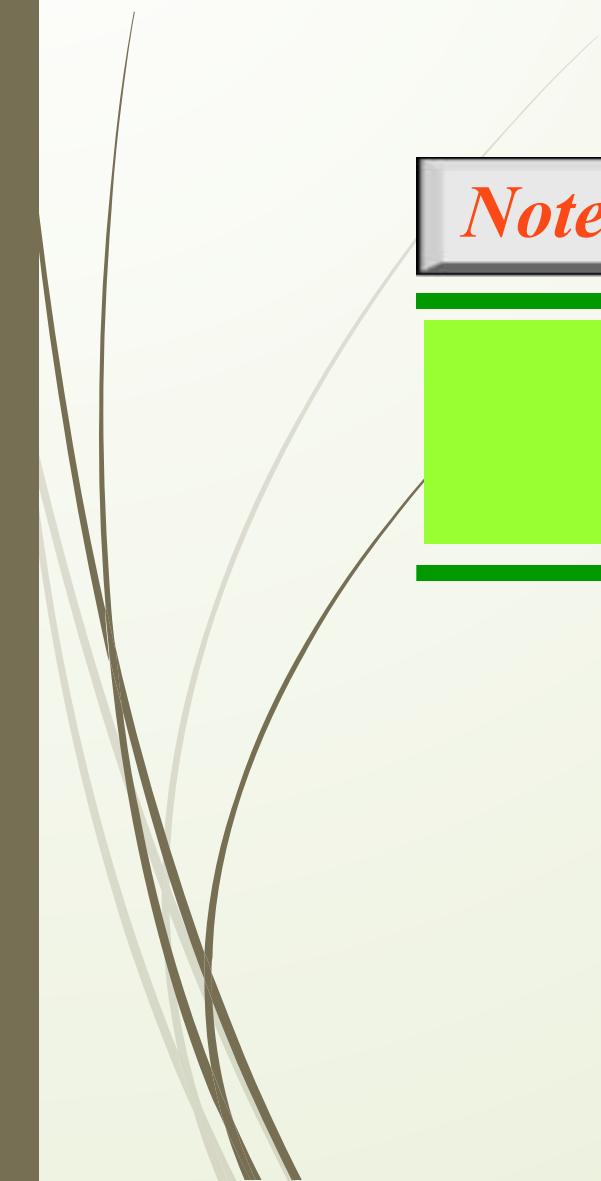
Note

SCTP association allows multiple IP addresses for each end.



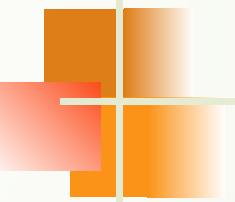
Note

In SCTP, a data chunk is numbered using a TSN.



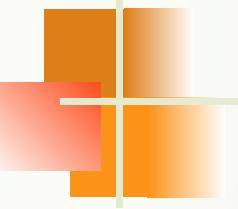
Note

To distinguish between different streams, SCTP uses an SI.



Note

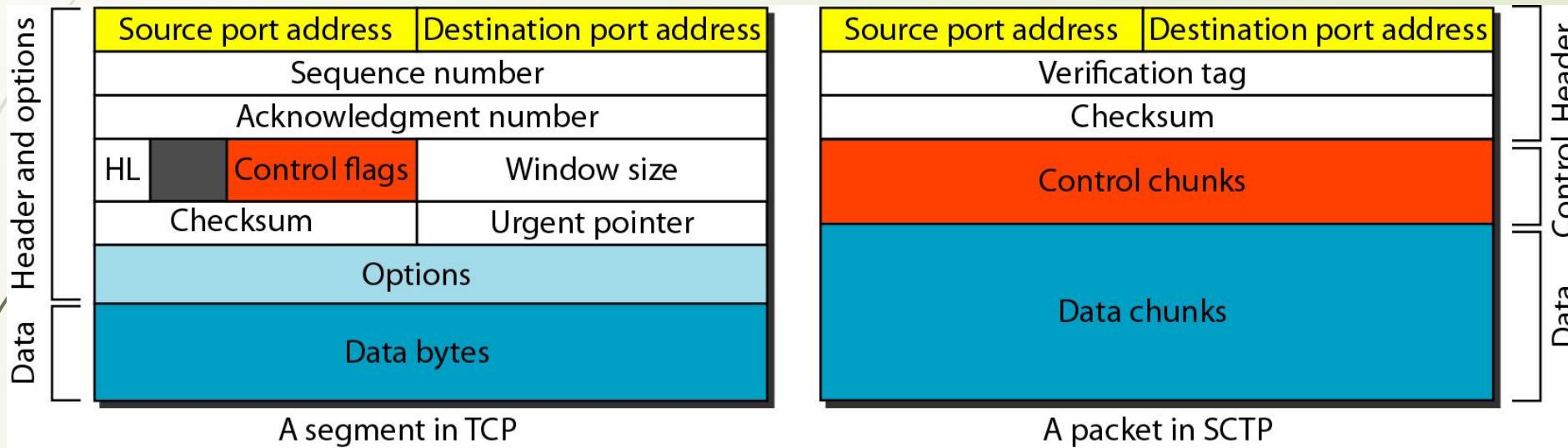
**To distinguish between different data chunks belonging to the same stream,
SCTP uses SSNs.**

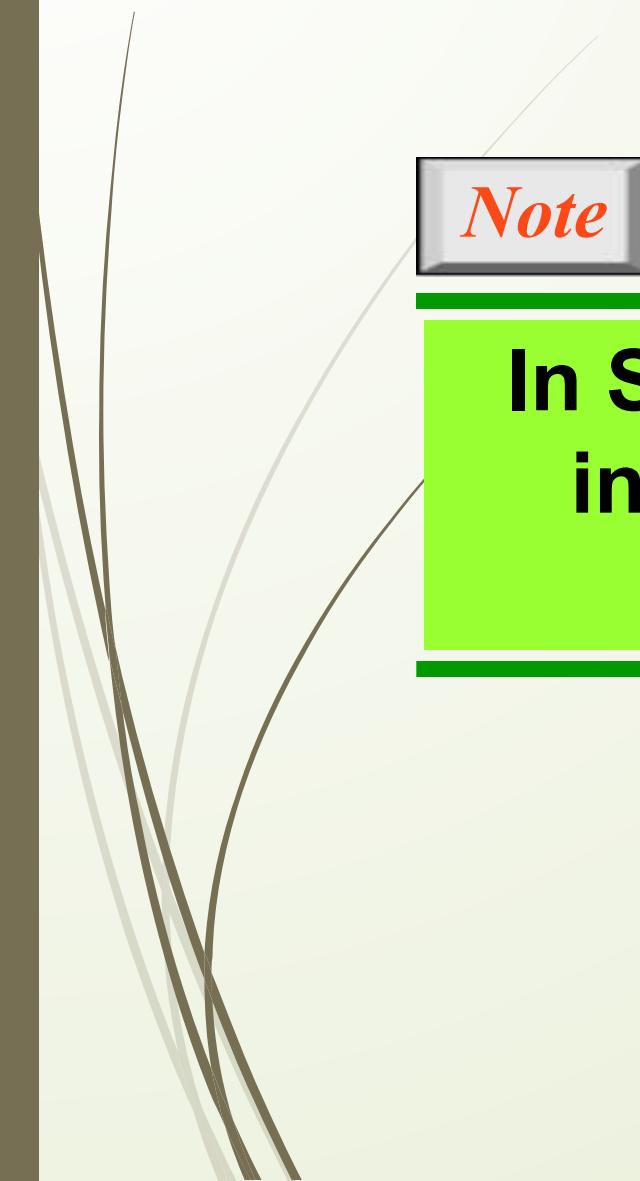


Note

TCP has segments; SCTP has packets.

Figure 12.29 Comparison between a TCP segment and an SCTP packet

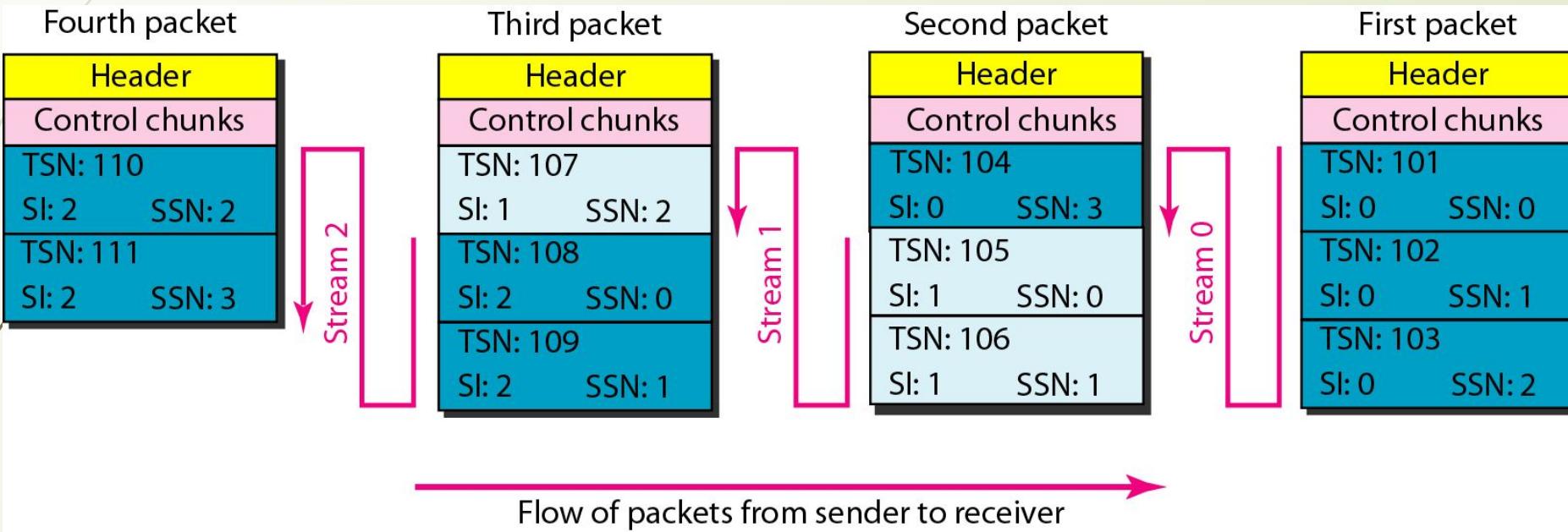




Note

In SCTP, control information and data information are carried in separate chunks.

Figure 12.30 *Packet, data chunks, and streams*

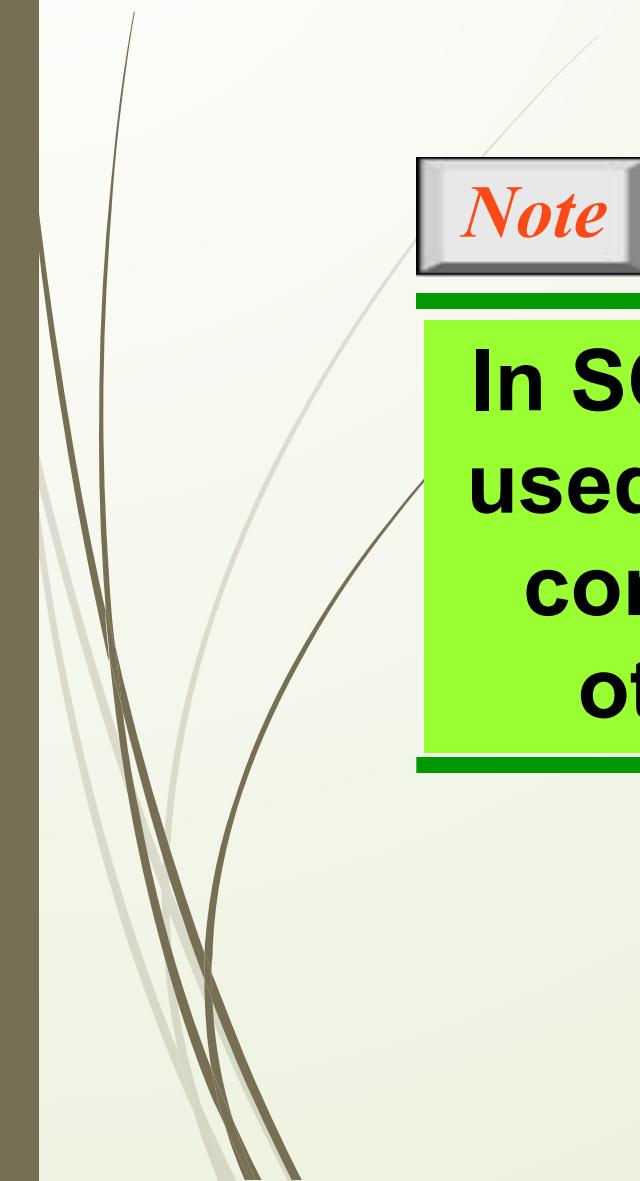




Note

Data chunks are identified by three items: TSN, SI, and SSN.

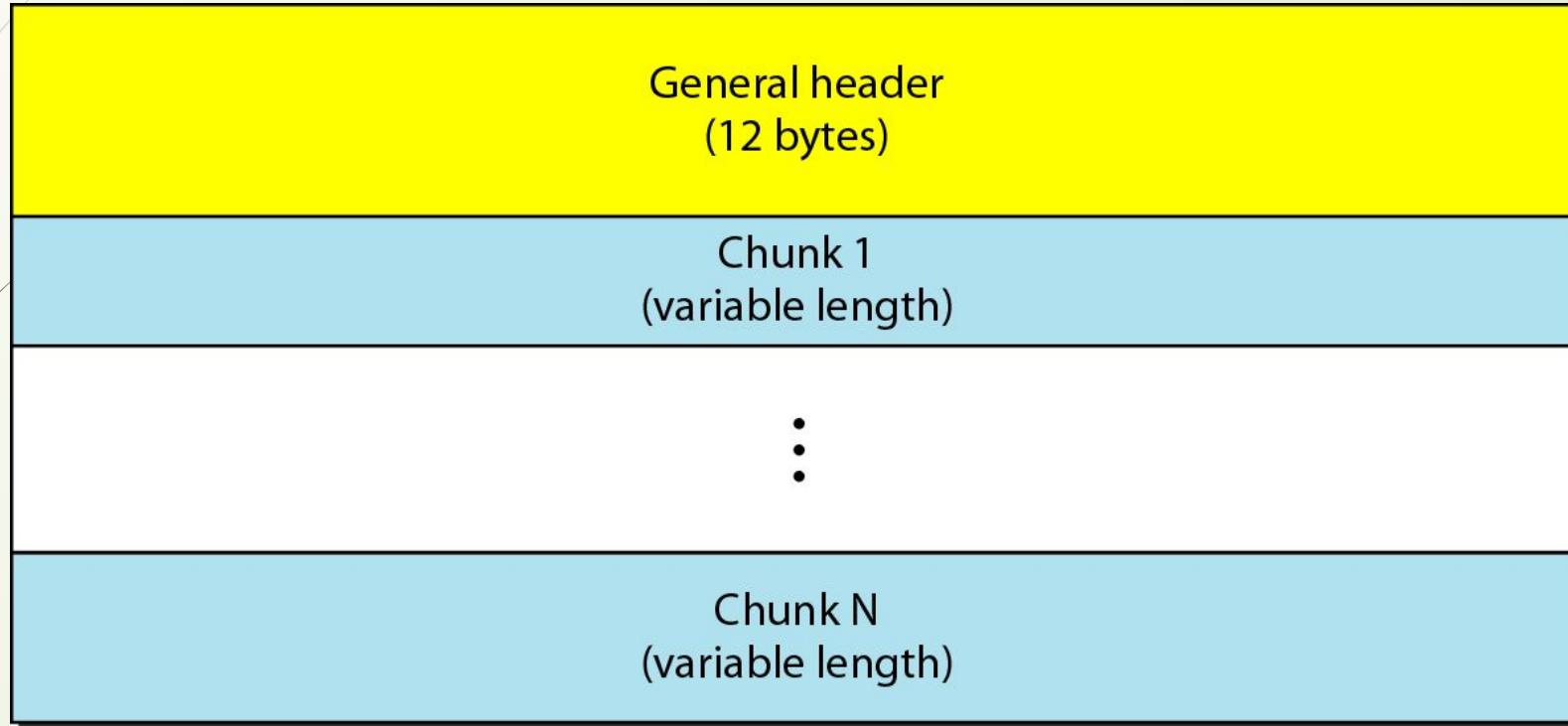
TSN is a cumulative number identifying the association; SI defines the stream; SSN defines the chunk in a stream.

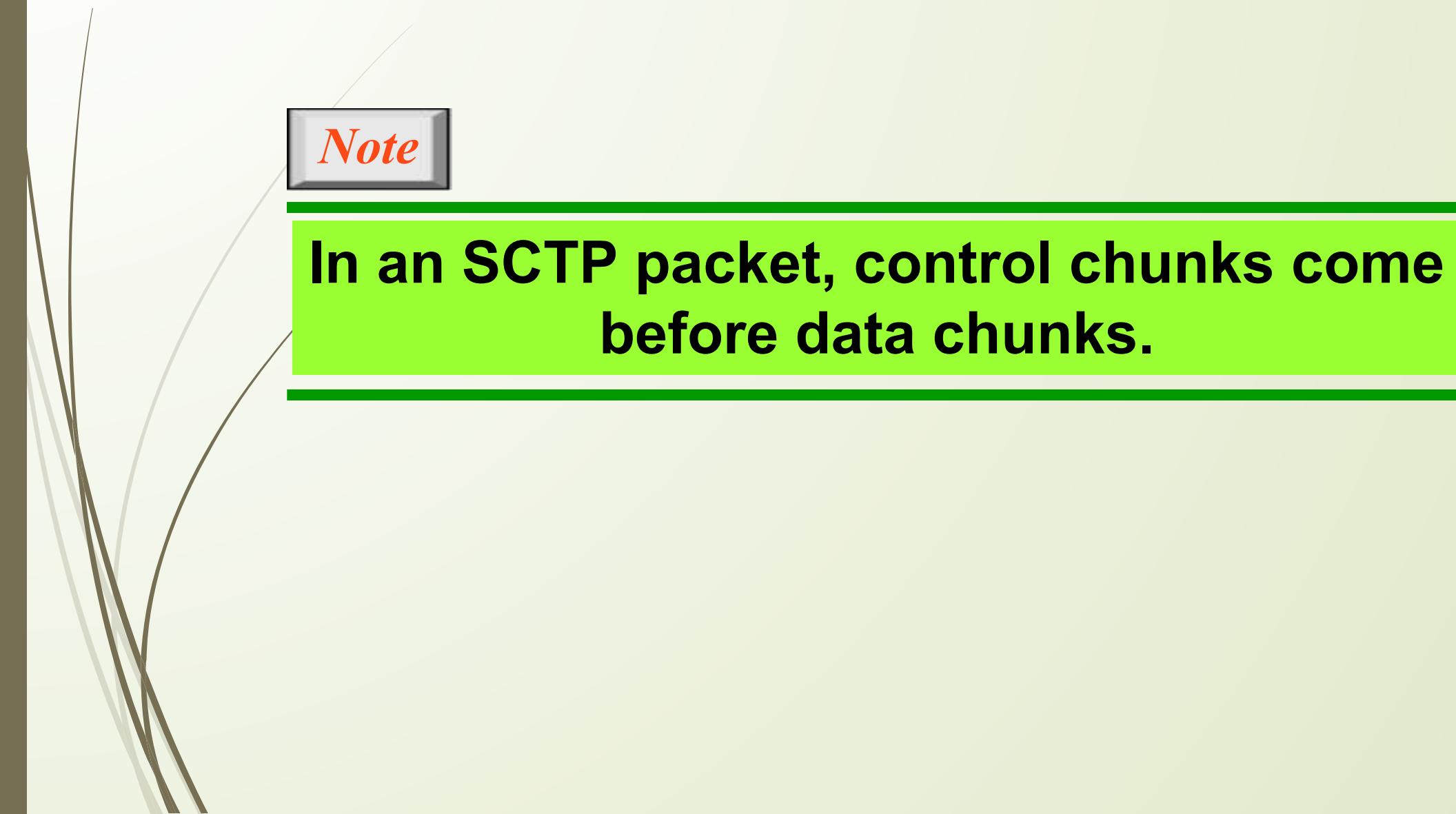


Note

In SCTP, acknowledgment numbers are used to acknowledge only data chunks; control chunks are acknowledged by other control chunks if necessary.

Figure 12.31 *SCTP packet format*





Note

In an SCTP packet, control chunks come before data chunks.

Figure 12.32 General header

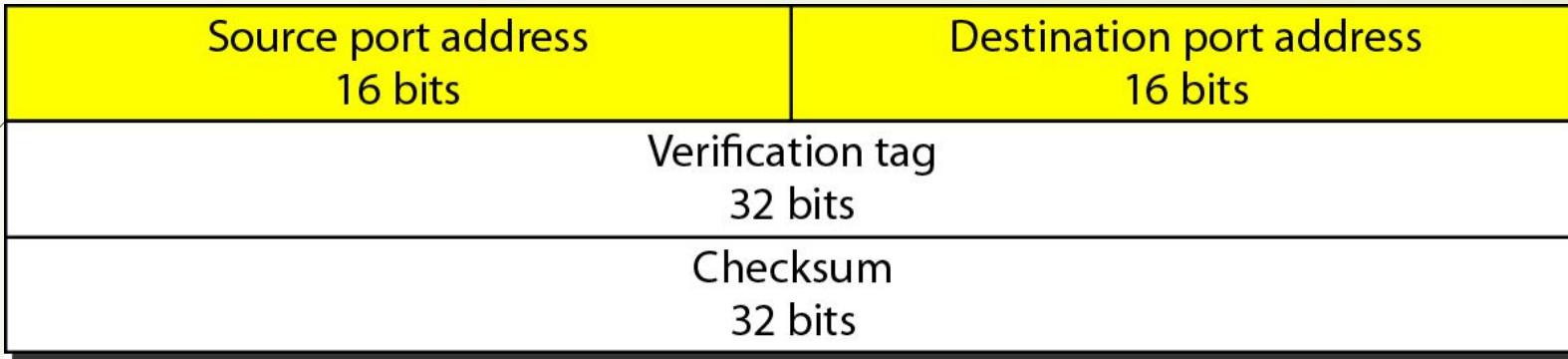
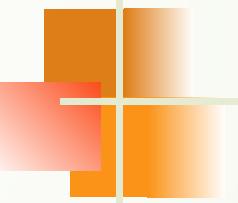


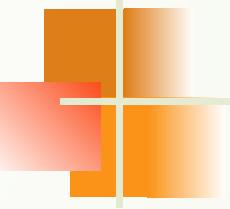
Table 12.5 *Chunks*

Type	Chunk	Description
0	DATA	User data
1	INIT	Sets up an association
2	INIT ACK	Acknowledges INIT chunk
3	SACK	Selective acknowledgment
4	HEARTBEAT	Probes the peer for liveliness
5	HEARTBEAT ACK	Acknowledges HEARTBEAT chunk
6	ABORT	Aborts an association
7	SHUTDOWN	Terminates an association
8	SHUTDOWN ACK	Acknowledges SHUTDOWN chunk
9	ERROR	Reports errors without shutting down
10	COOKIE ECHO	Third packet in association establishment
11	COOKIE ACK	Acknowledges COOKIE ECHO chunk
14	SHUTDOWN COMPLETE	Third packet in association termination
192	FORWARD TSN	For adjusting cumulative TSN



Note

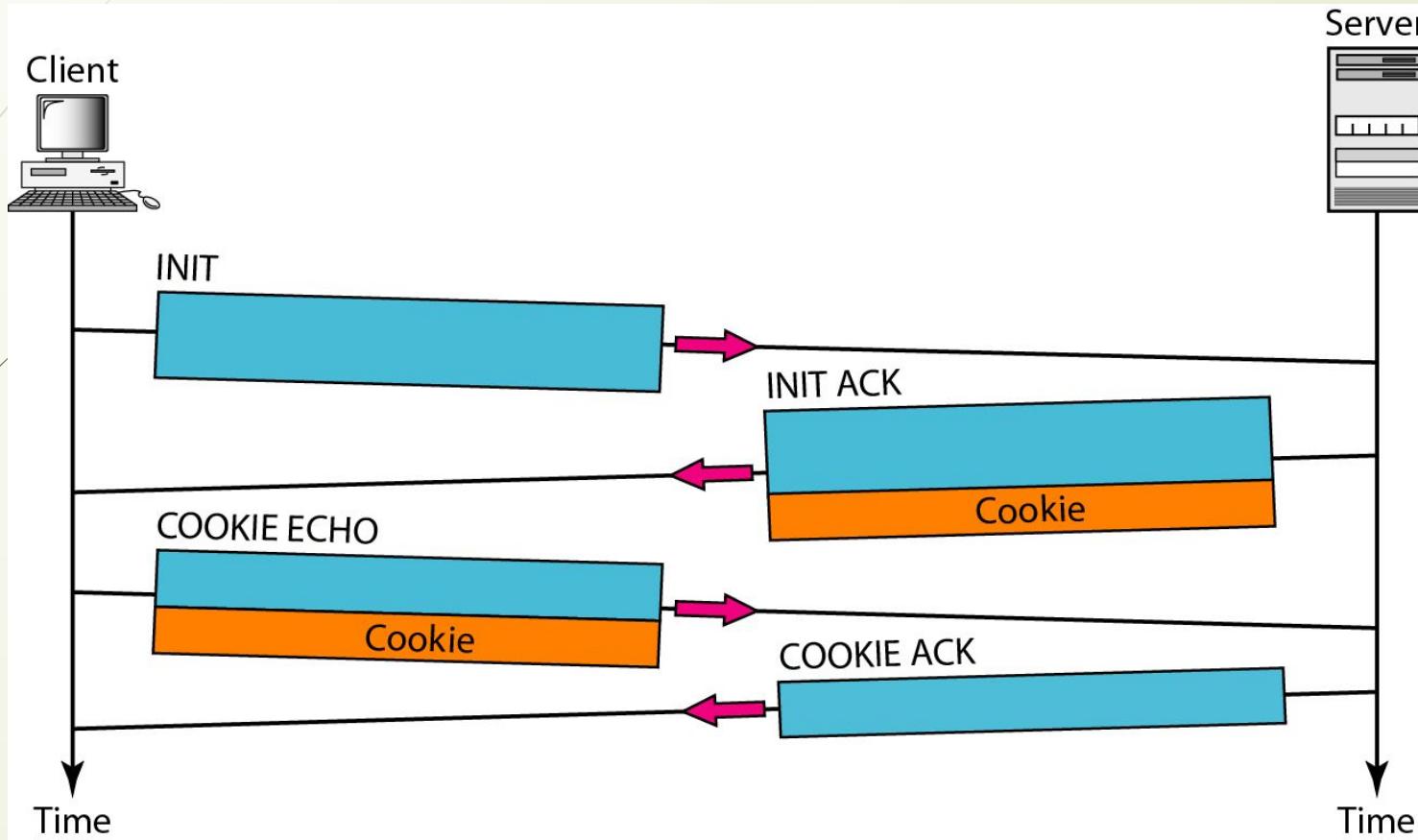
A connection in SCTP is called an association.

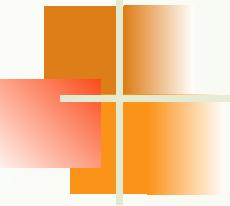


Note

**No other chunk is allowed in a packet carrying an INIT or INIT ACK chunk.
A COOKIE ECHO or a COOKIE ACK chunk can carry data chunks.**

Figure 12.33 Four-way handshaking



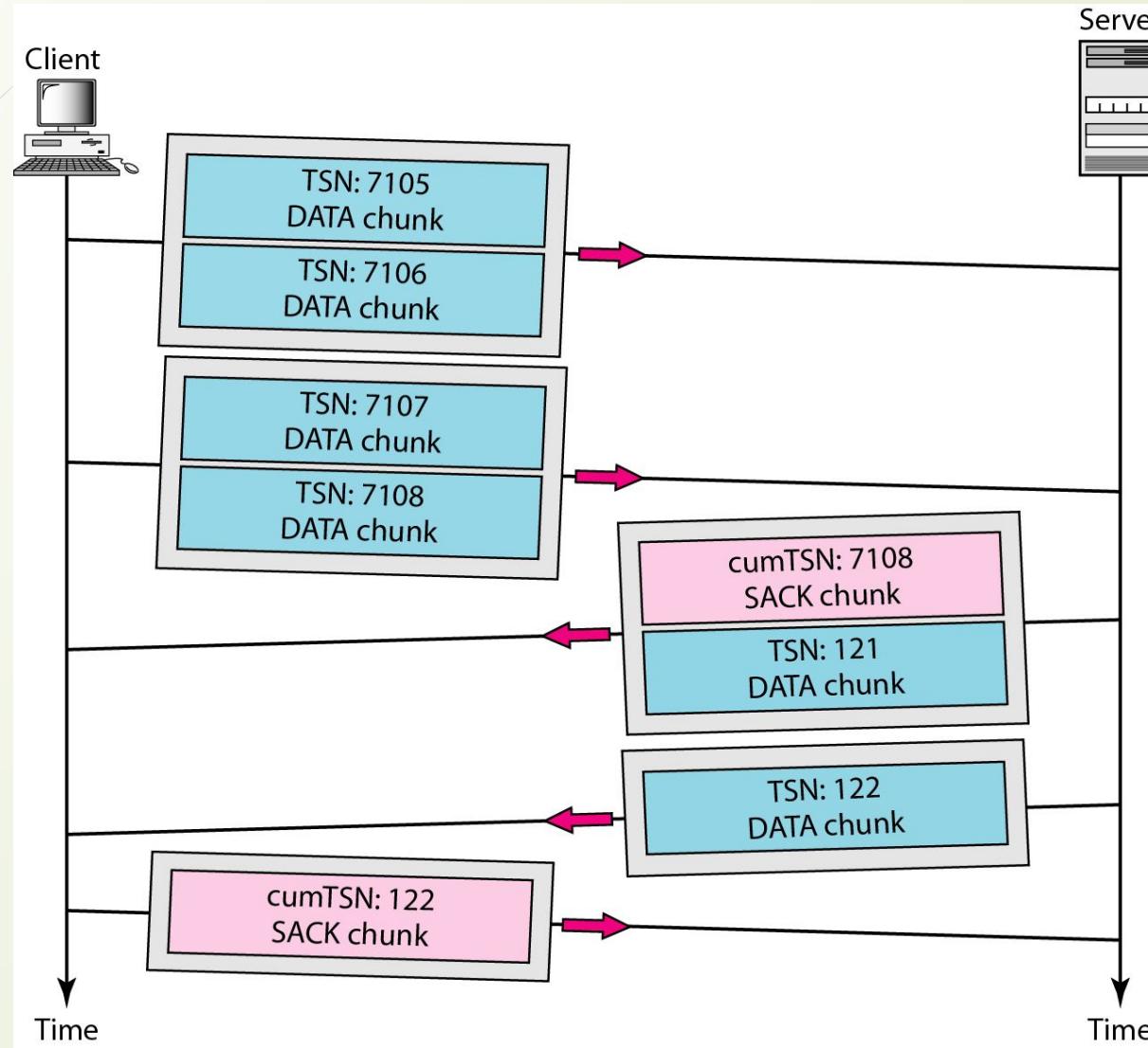


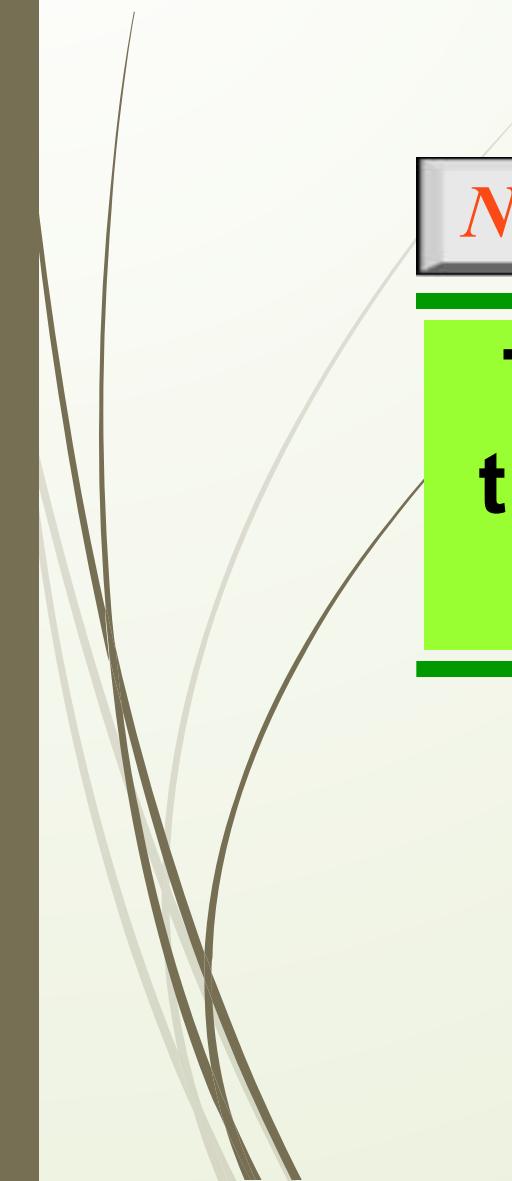
Note

**In SCTP, only DATA chunks
consume TSNs;
DATA chunks are the only chunks
that are acknowledged.**



Figure 12.34 Simple data transfer





Note

The acknowledgment in SCTP defines the cumulative TSN, the TSN of the last data chunk received in order.

Figure 12.35 Association termination

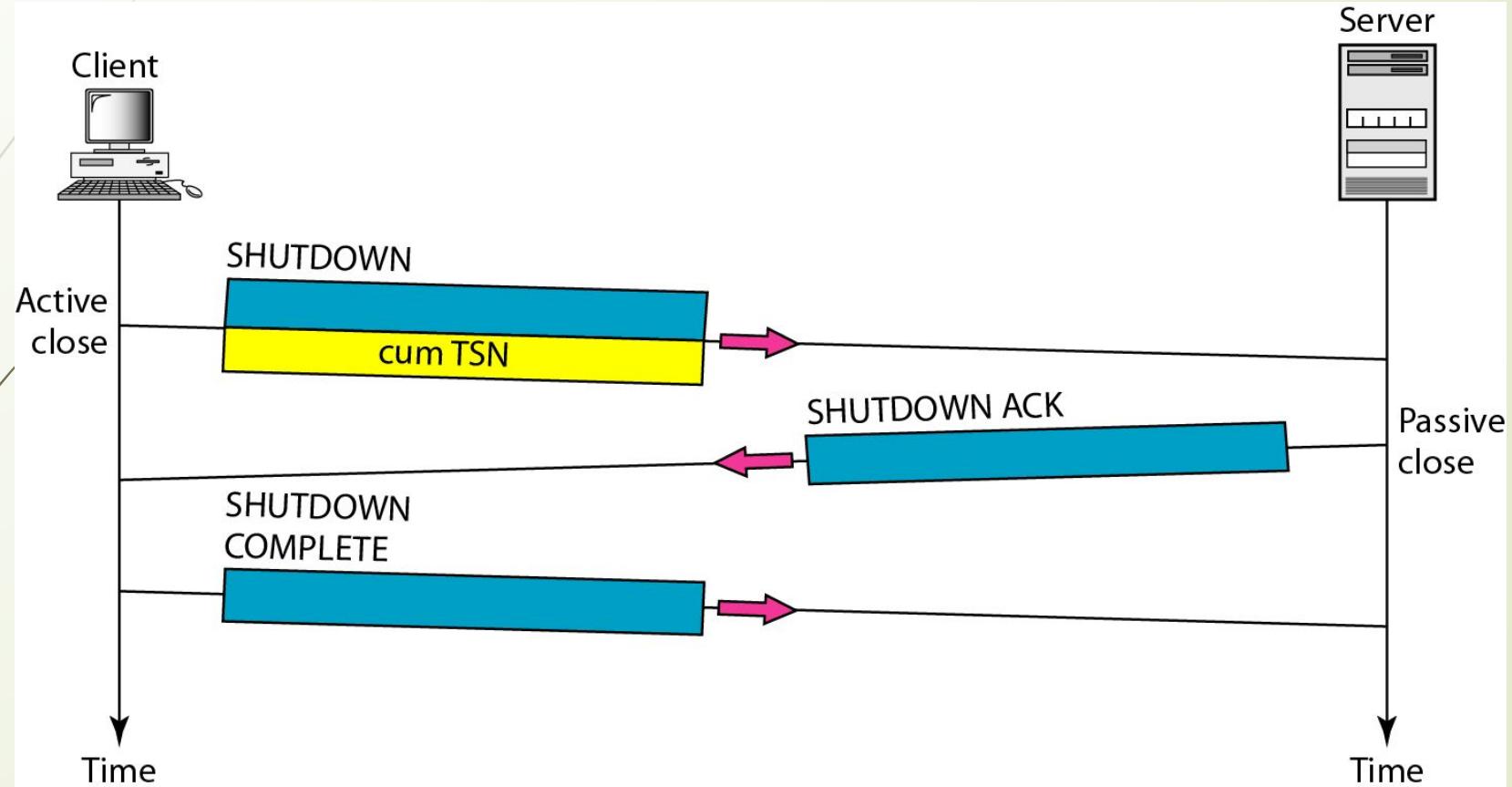


Figure 12.36 Flow control, receiver site

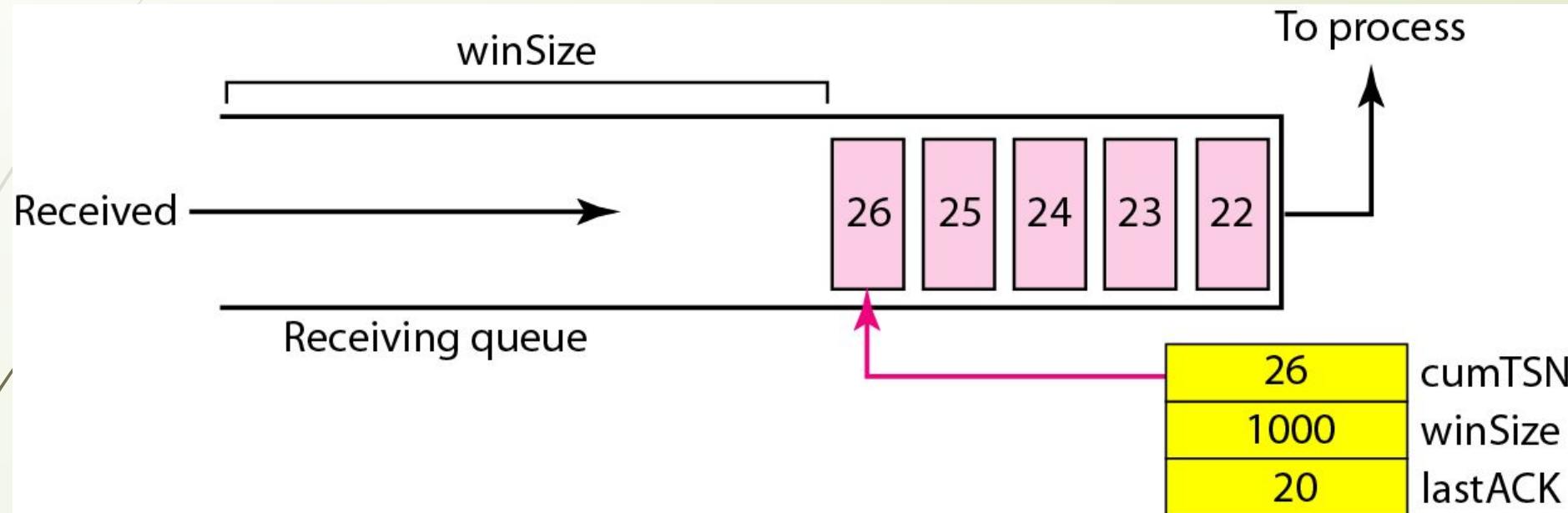


Figure 12.37 Flow control, sender site

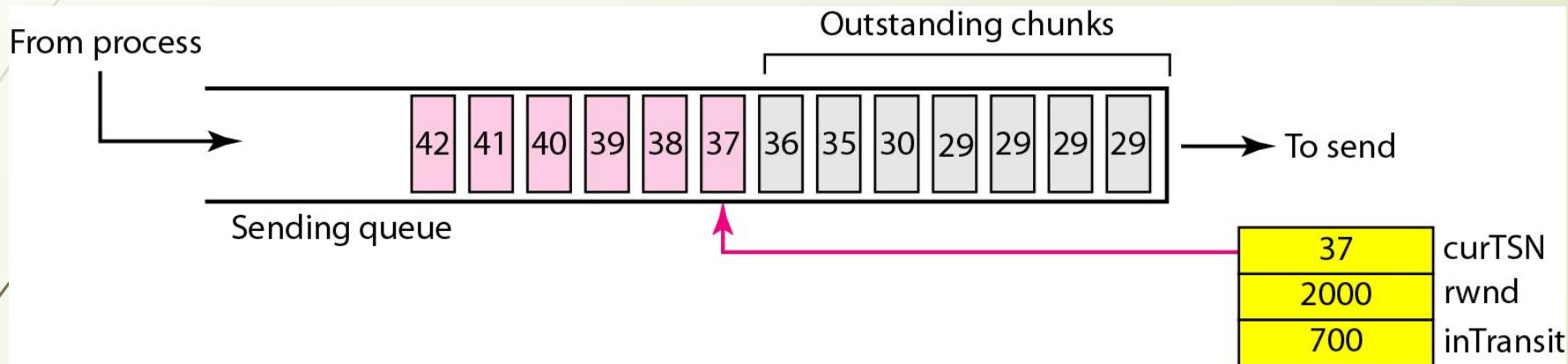


Figure 12.38 Flow control scenario

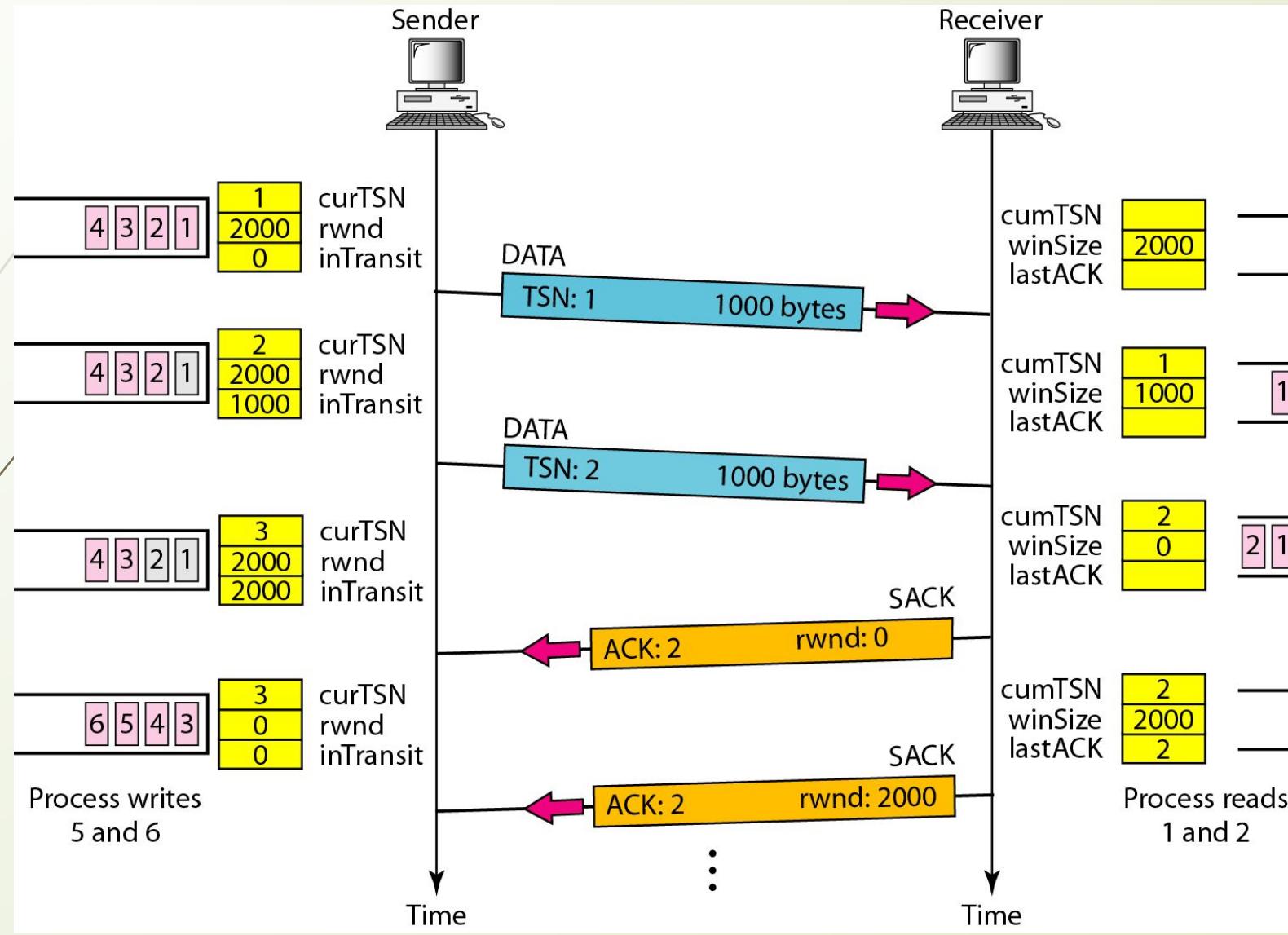


Figure 12.39 Error control, receiver site

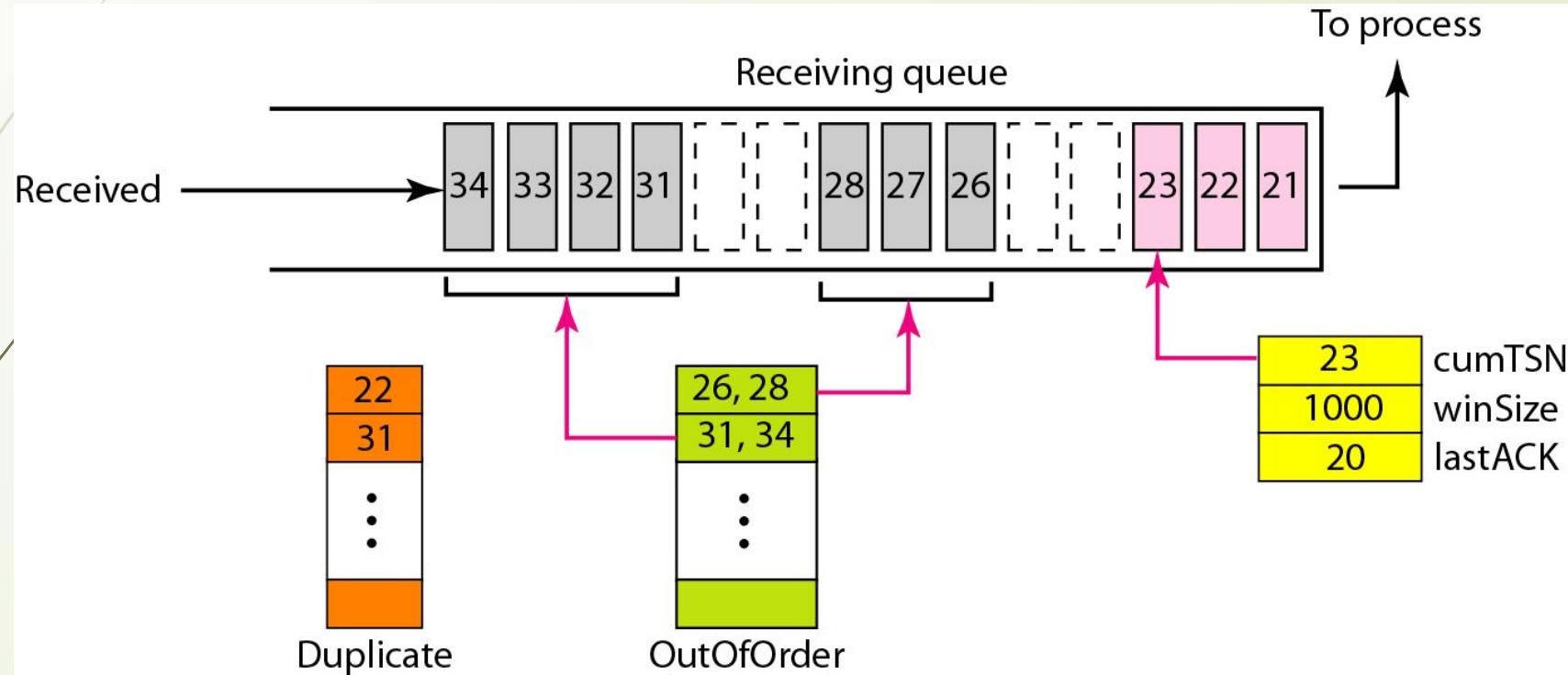
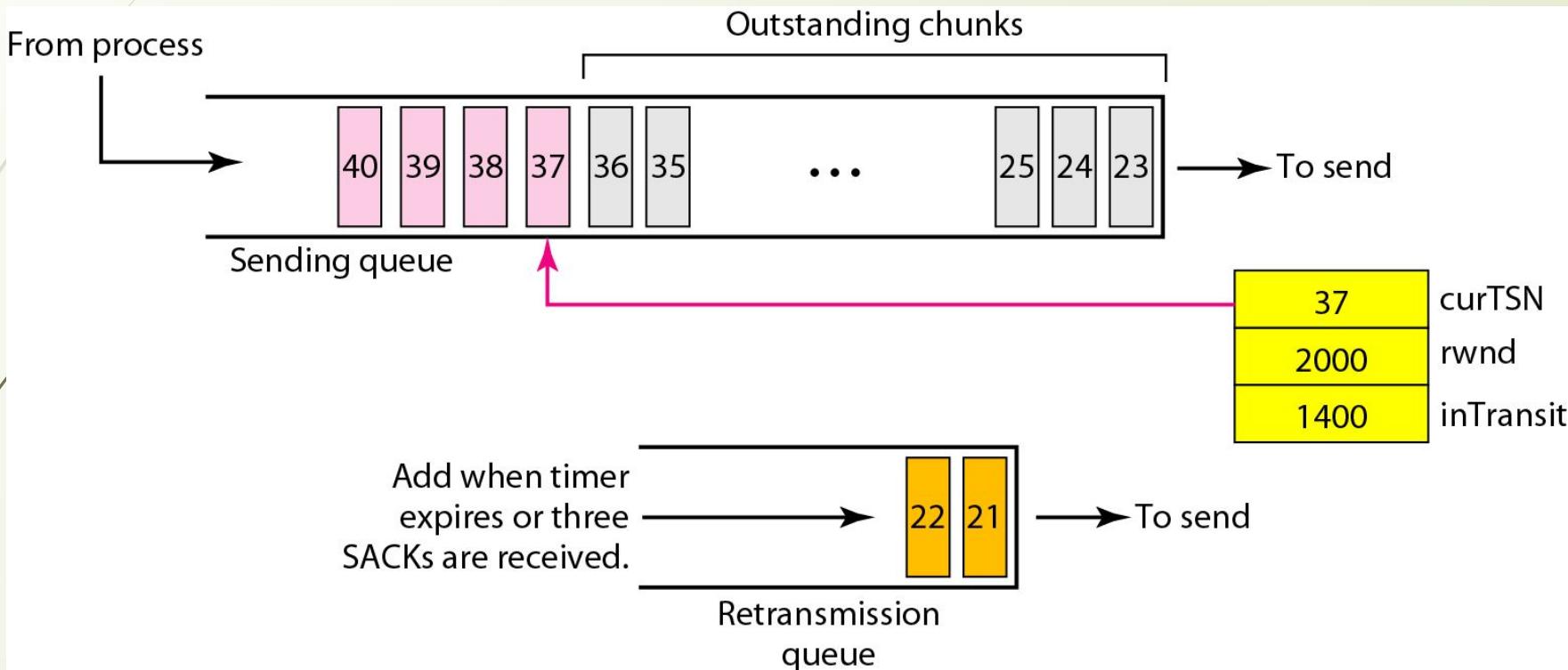
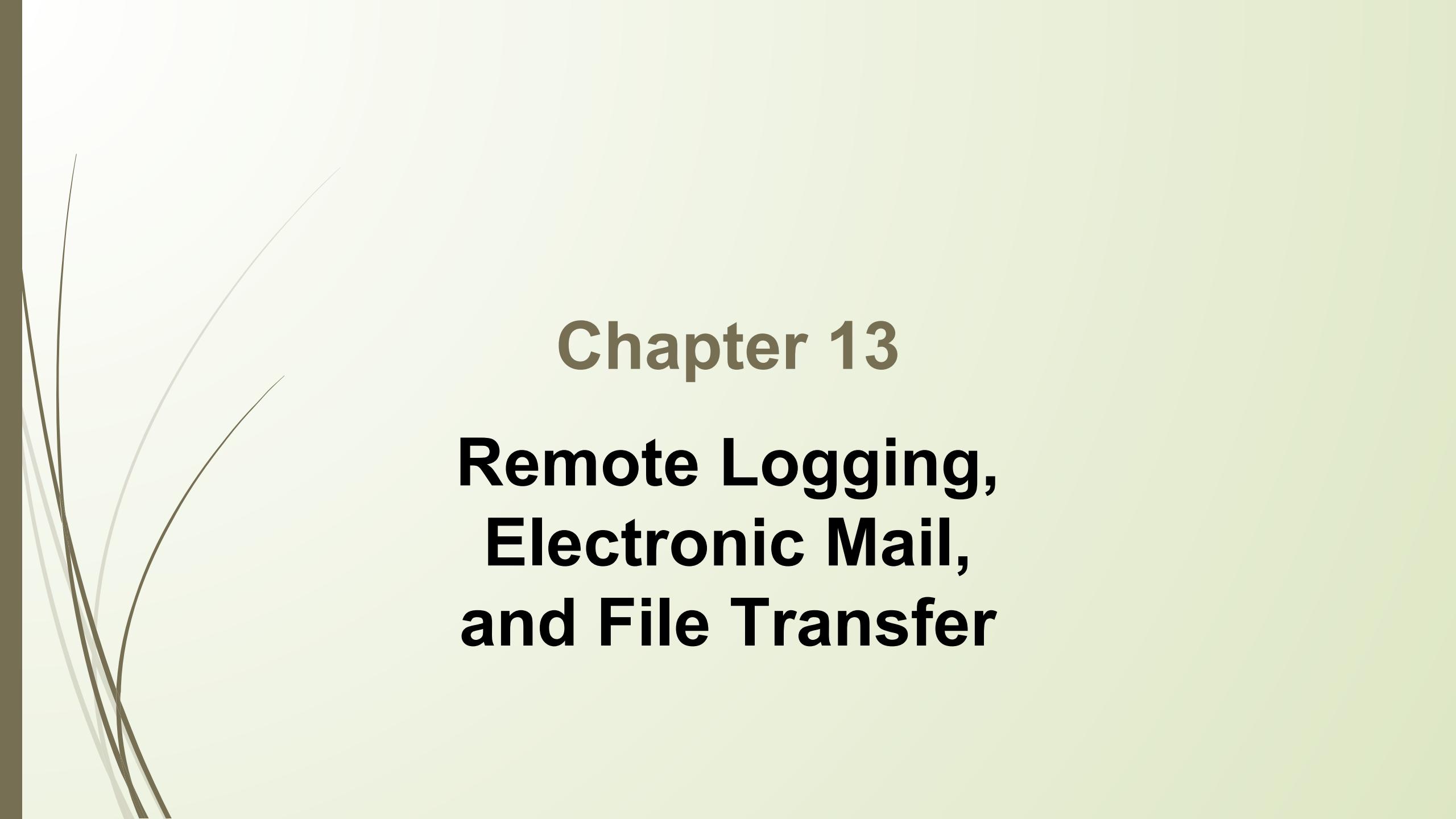


Figure 12.40 Error control, sender site





Chapter 13

Remote Logging, Electronic Mail, and File Transfer

13-1 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 13.1 Local and remote log-in

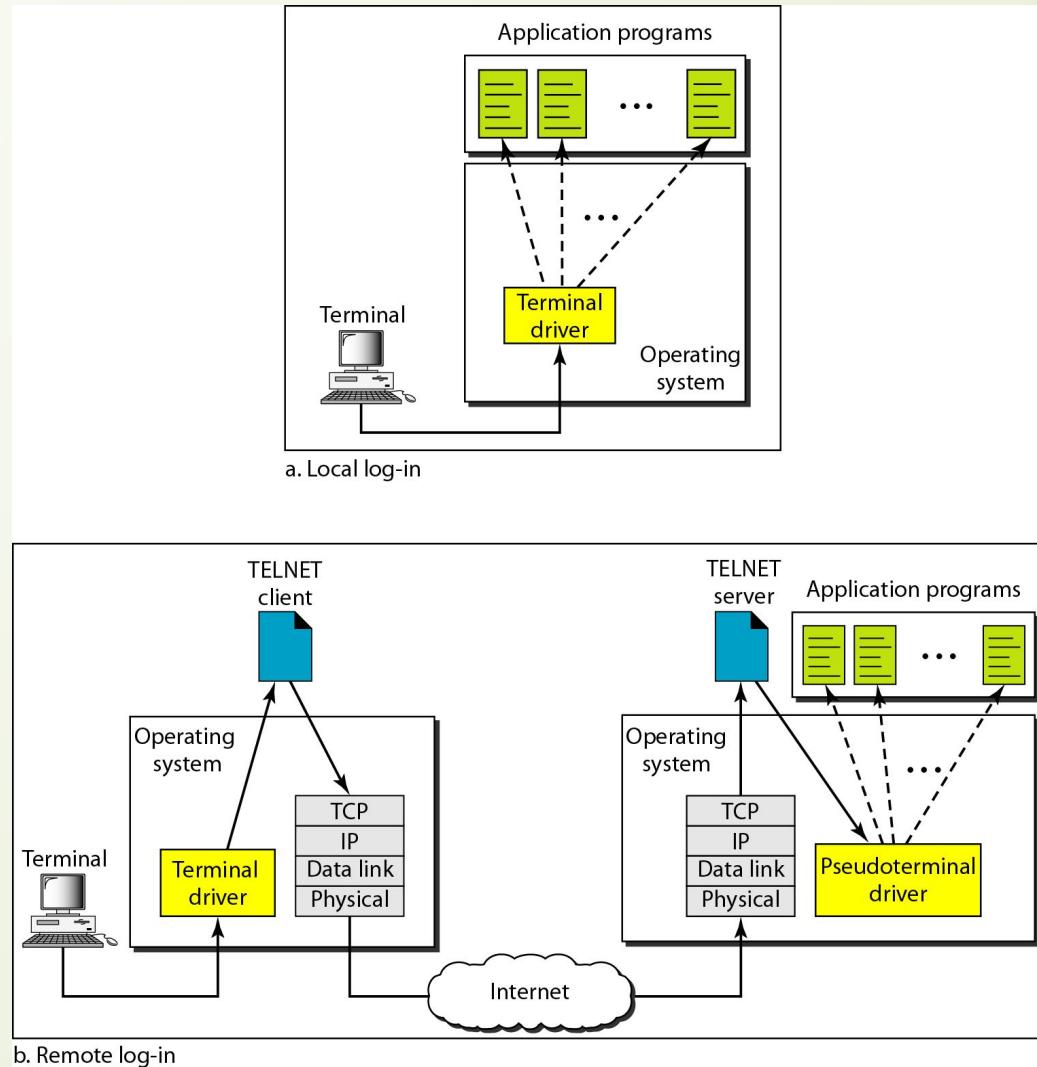
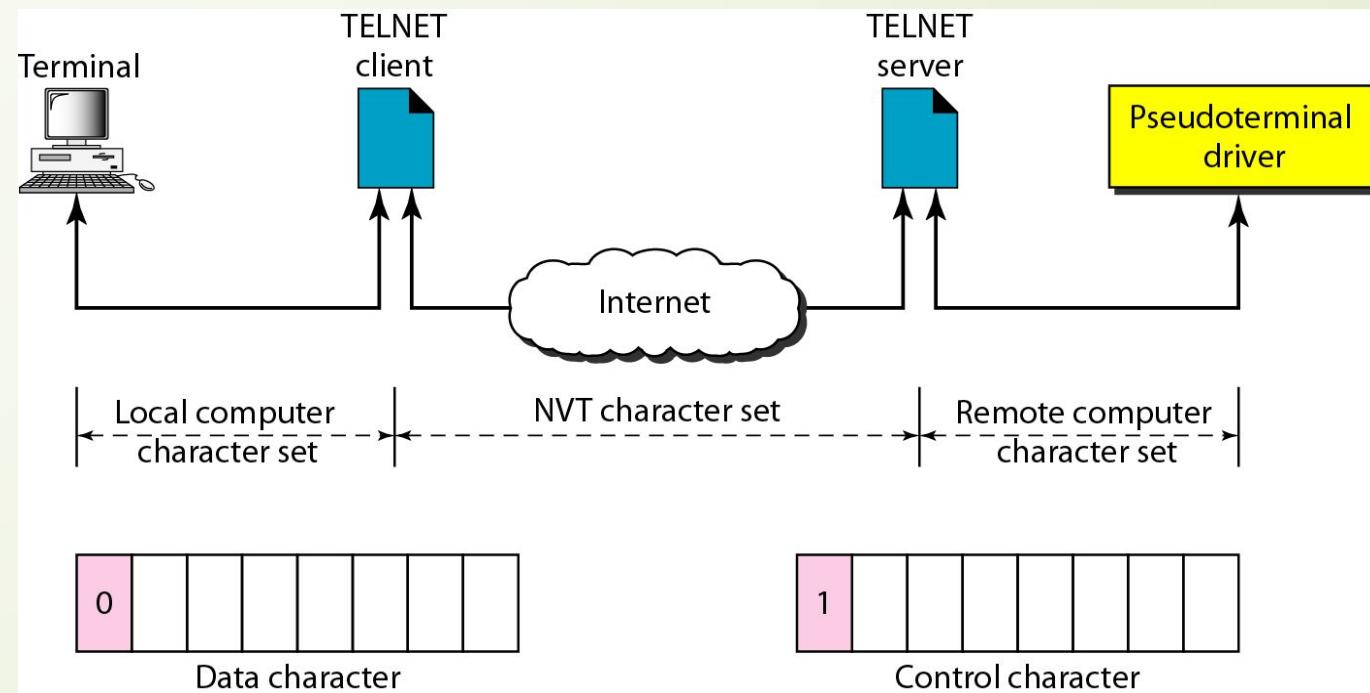


Figure 13.2 *Concept of Network Virtual Terminal (NVT)*



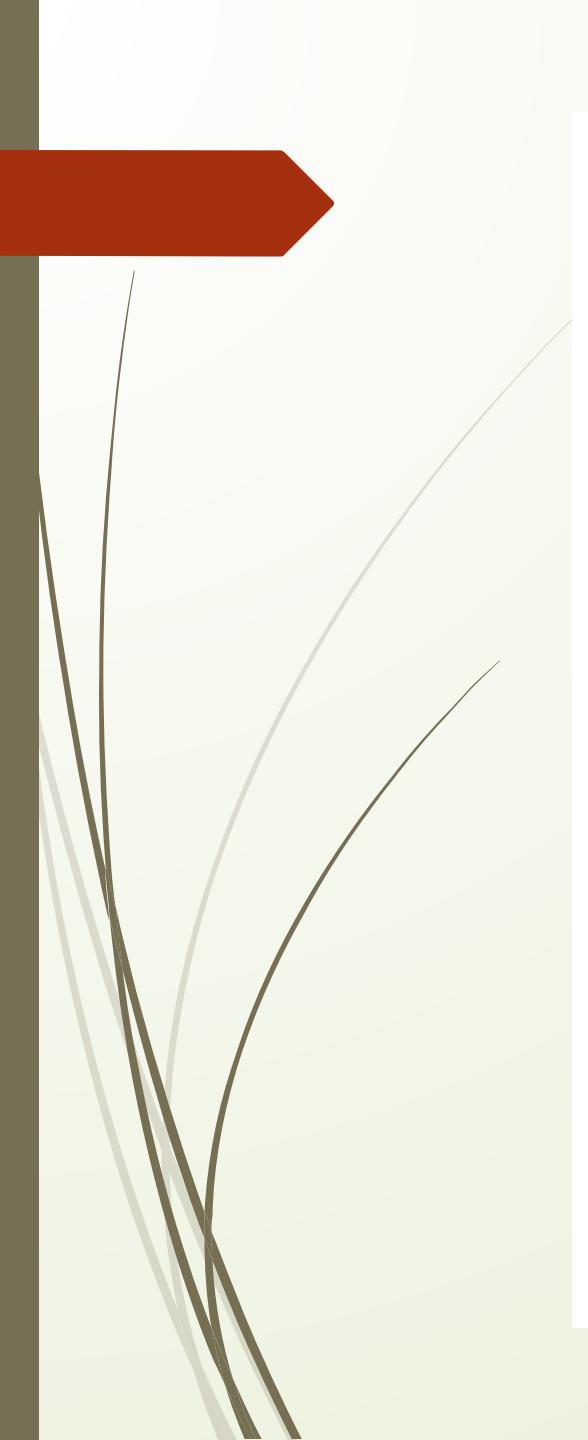


Table 13.1 Some NVT control characters

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 13.3 An example of embedding





Table 13.2 *Options*

<i>Code</i>	<i>Option</i>	<i>Meaning</i>
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 13.3 NVT character set for option negotiation

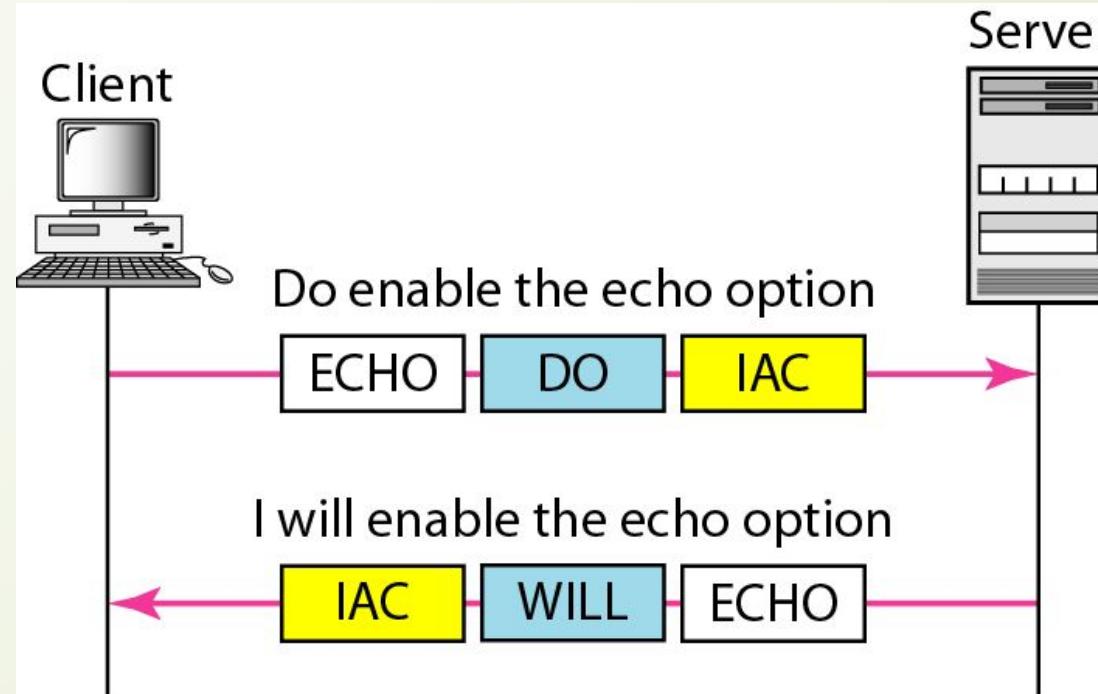
Character	Decimal	Binary	Meaning
WILL	251	11111011	1. Offering to enable 2. Accepting a request to enable
WONT	252	11111100	1. Rejecting a request to enable 2. Offering to disable 3. Accepting a request to disable
DO	253	11111101	1. Approving an offer to enable 2. Requesting to enable
DONT	254	11111110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



Example 13.1

Figure 13.4 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 13.4 Example 13.1: Echo option



13-2 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

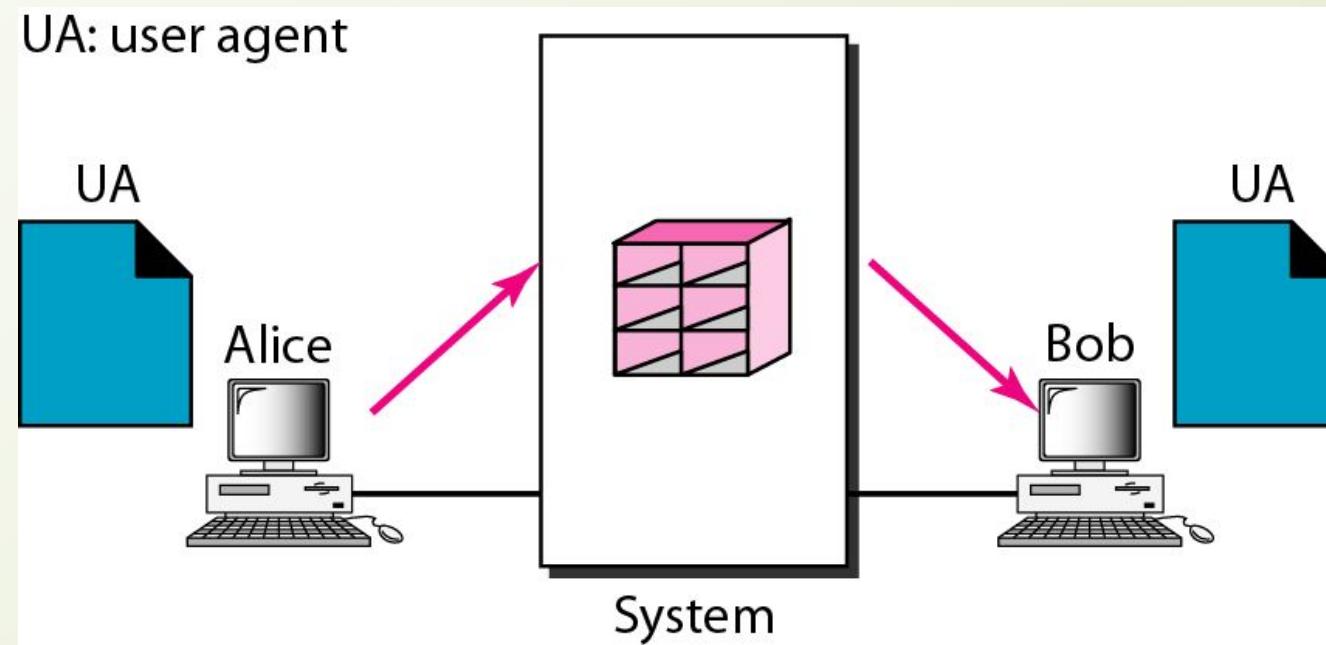
User Agent

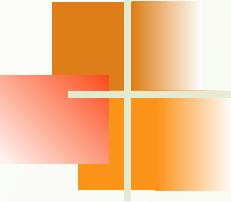
Message Transfer Agent: SMTP

Message Access Agent: POP and IMAP

Web-Based Mail

Figure 13.6 First scenario in electronic mail

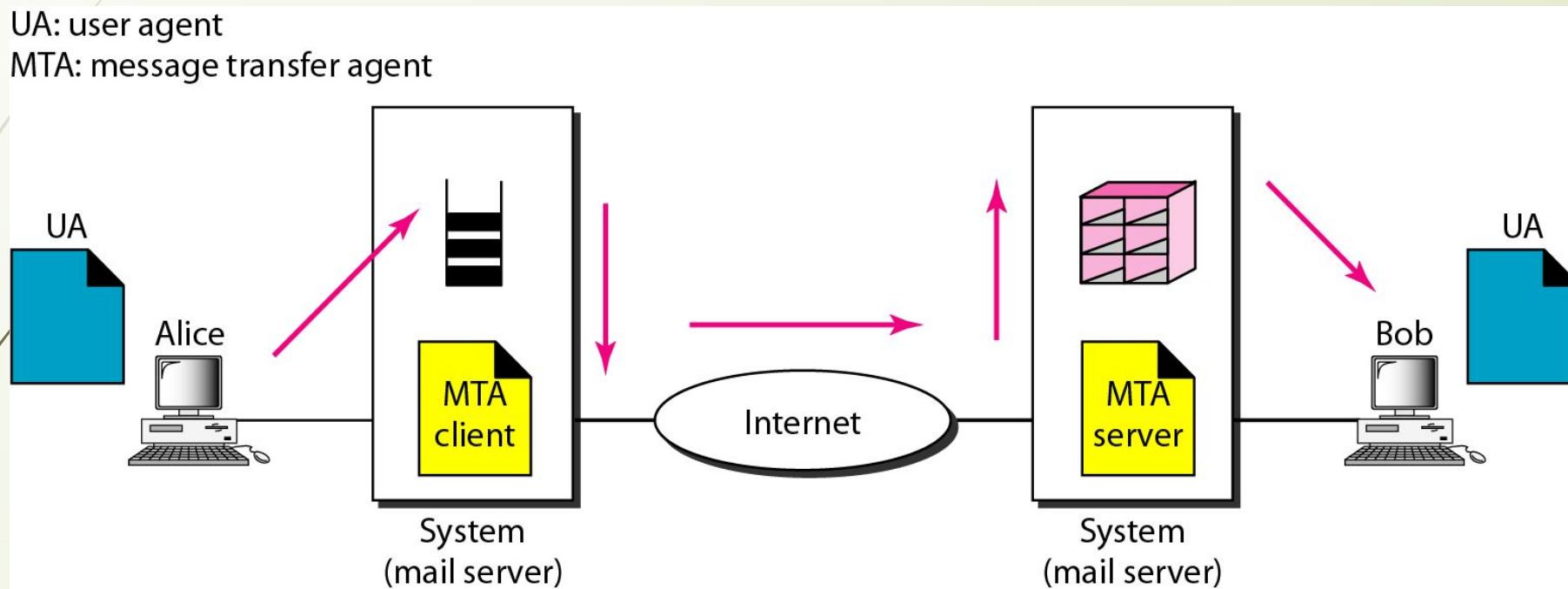


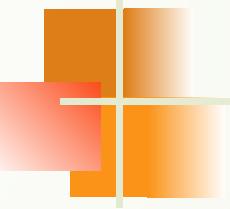


Note

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

Figure 13.7 Second scenario in electronic mail

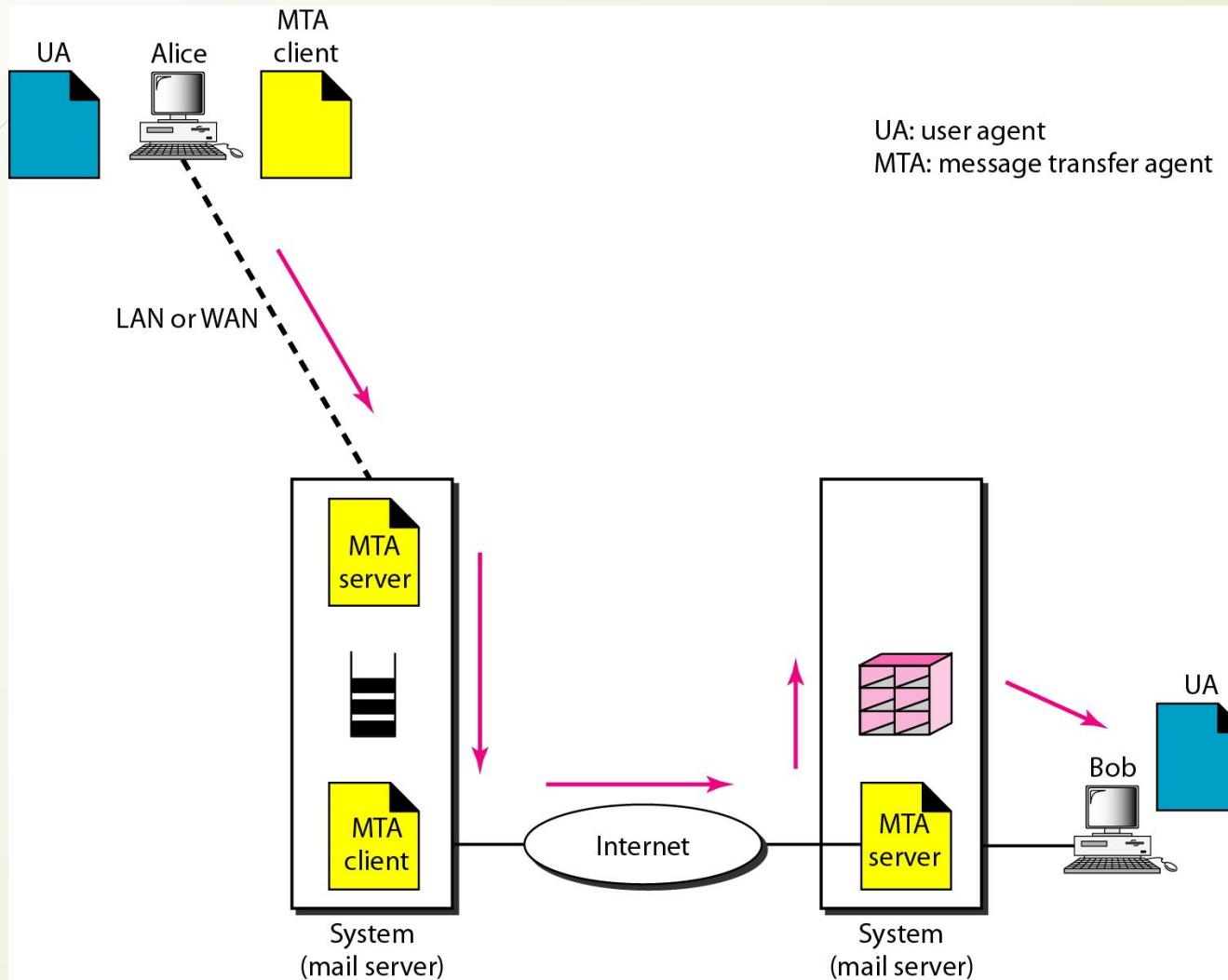


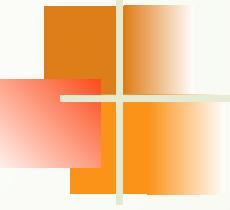


Note

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 13.8 *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 13.9 Fourth scenario in electronic mail

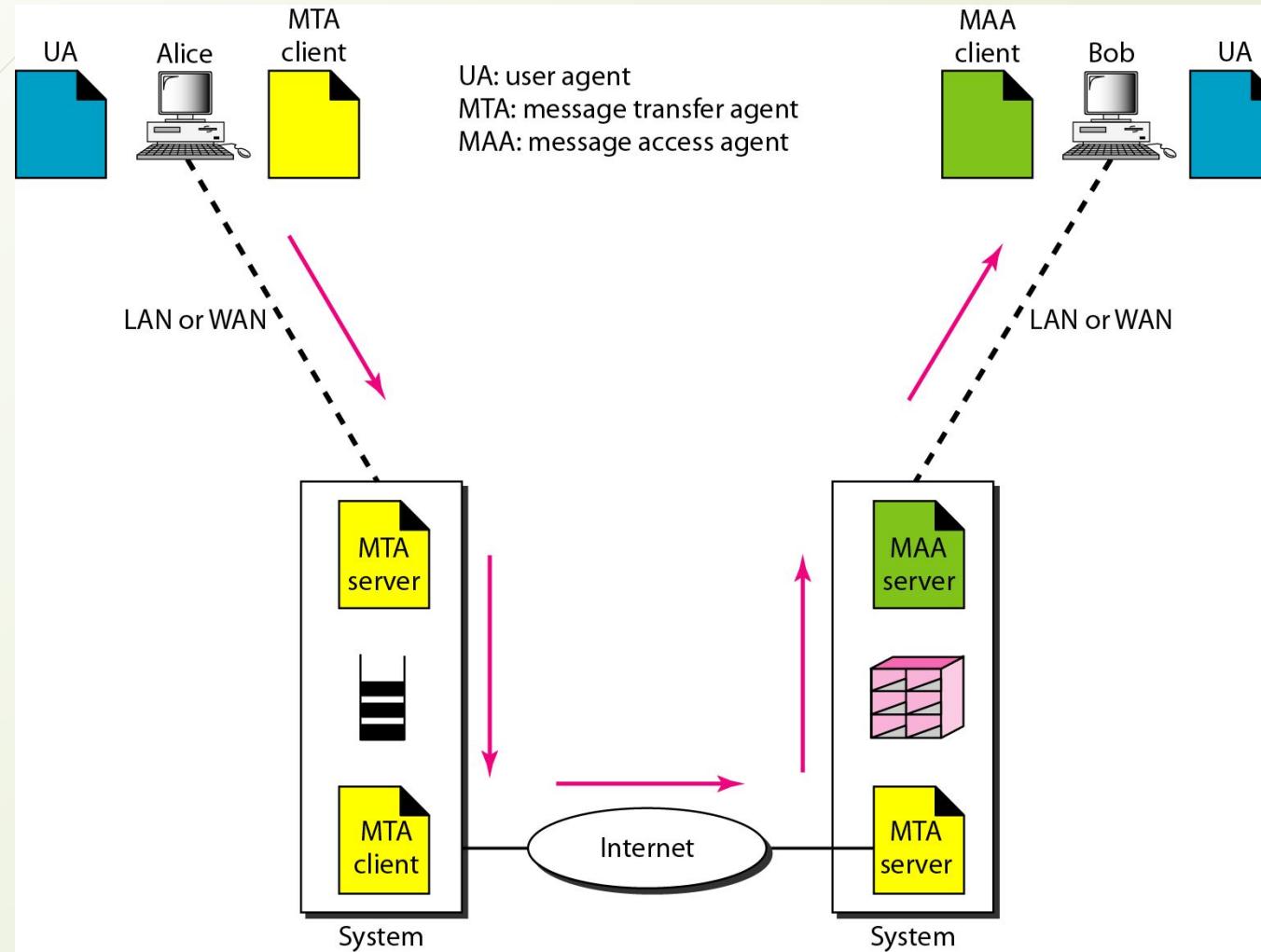
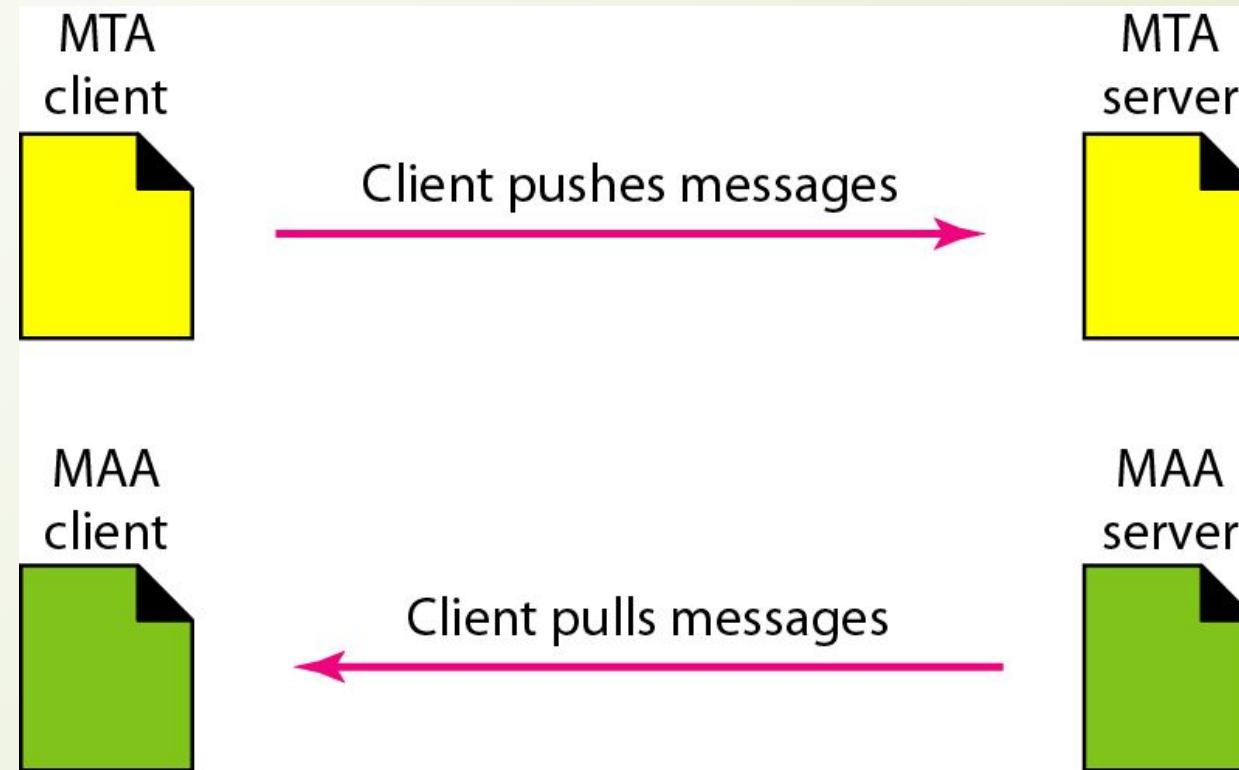
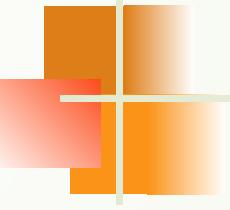


Figure 13.10 *Push versus pull in electronic email*



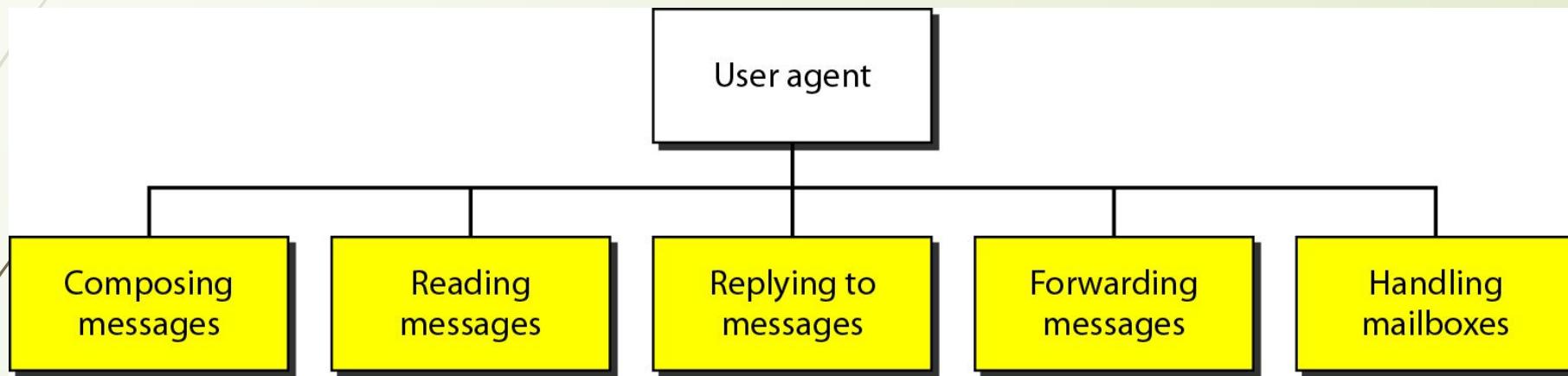


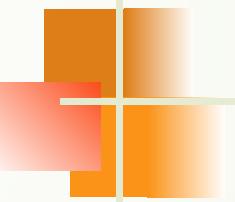
Note

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.

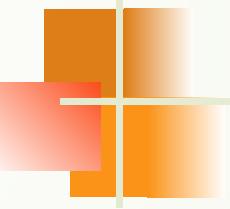
Figure 13.11 *Services of user agent*





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 13.12 Format of an e-mail

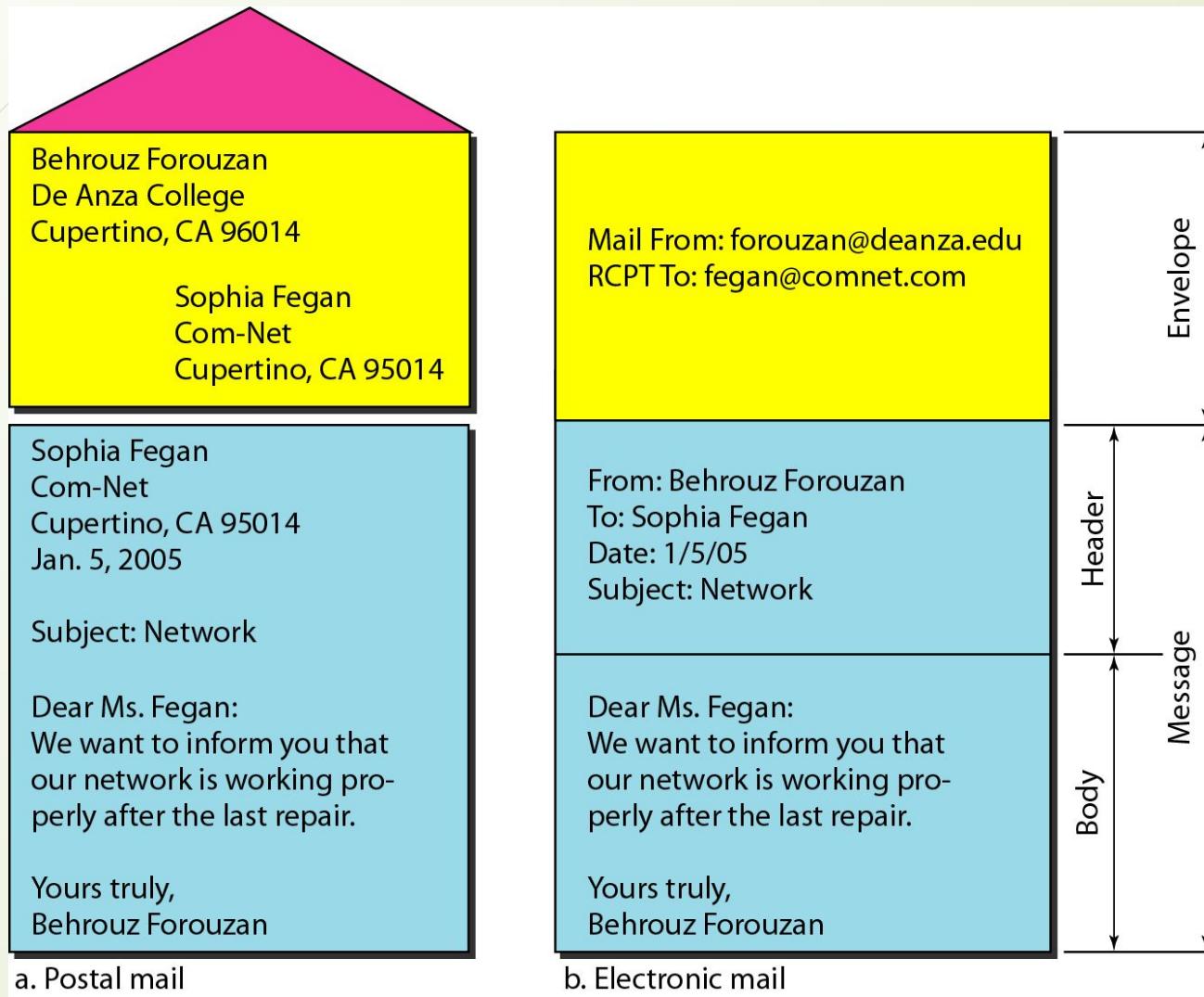


Figure 13.13 *E-mail address*

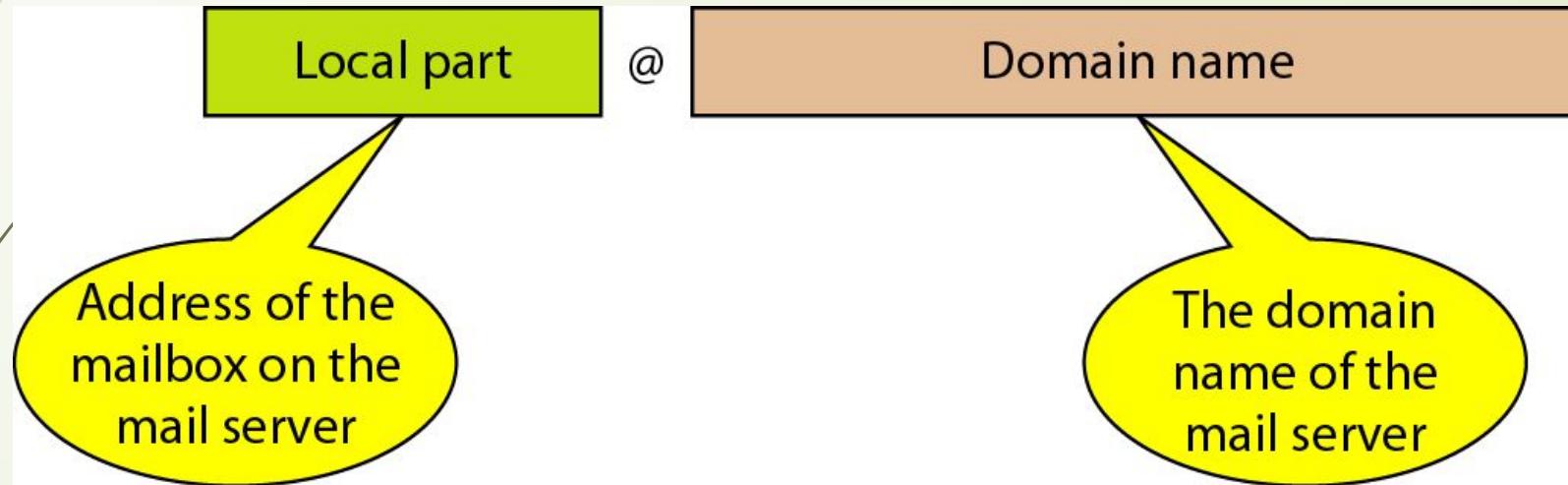


Figure 13.14 Multipurpose Internet Mail Extensions (MIME)

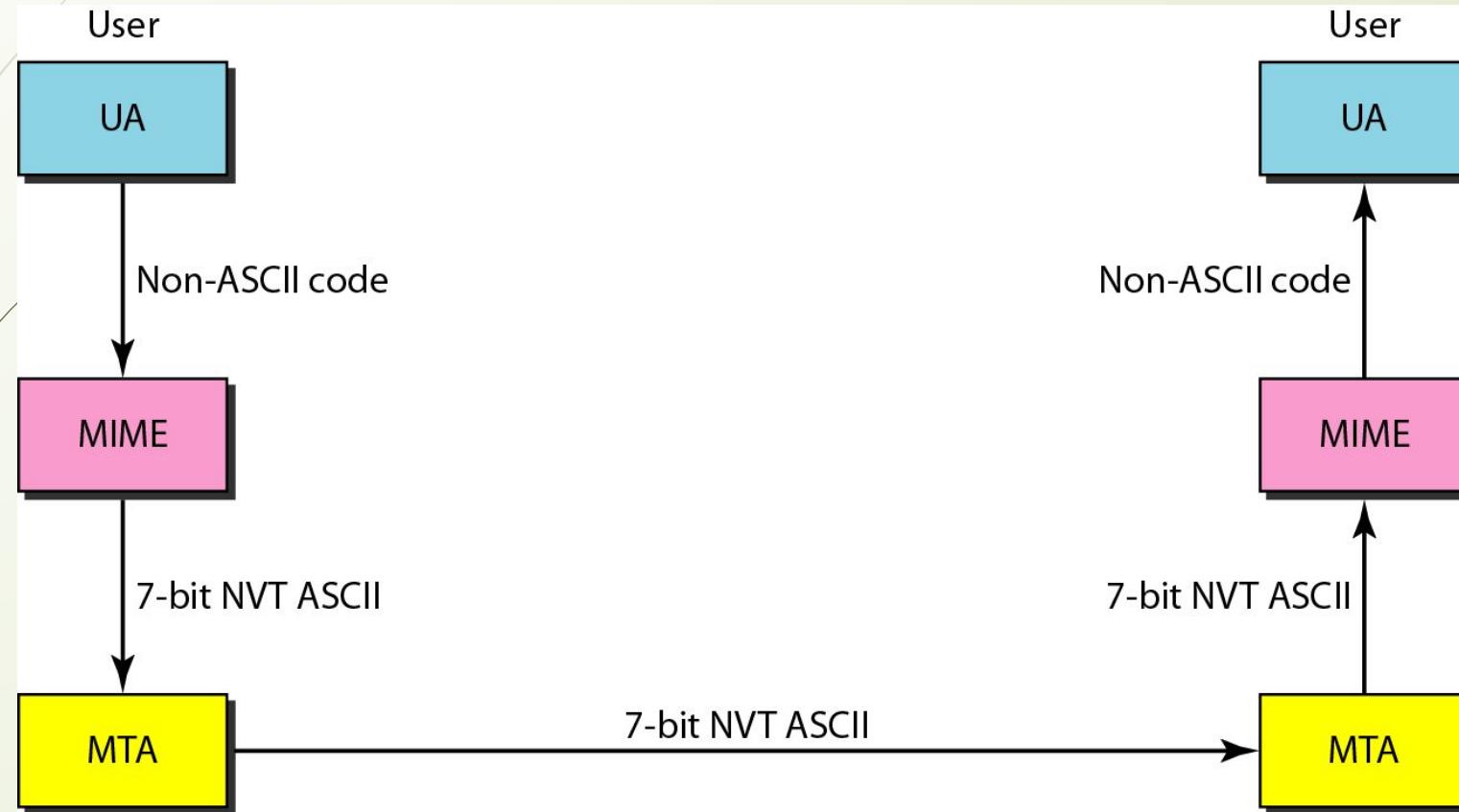


Figure 13.15 *MIME header*

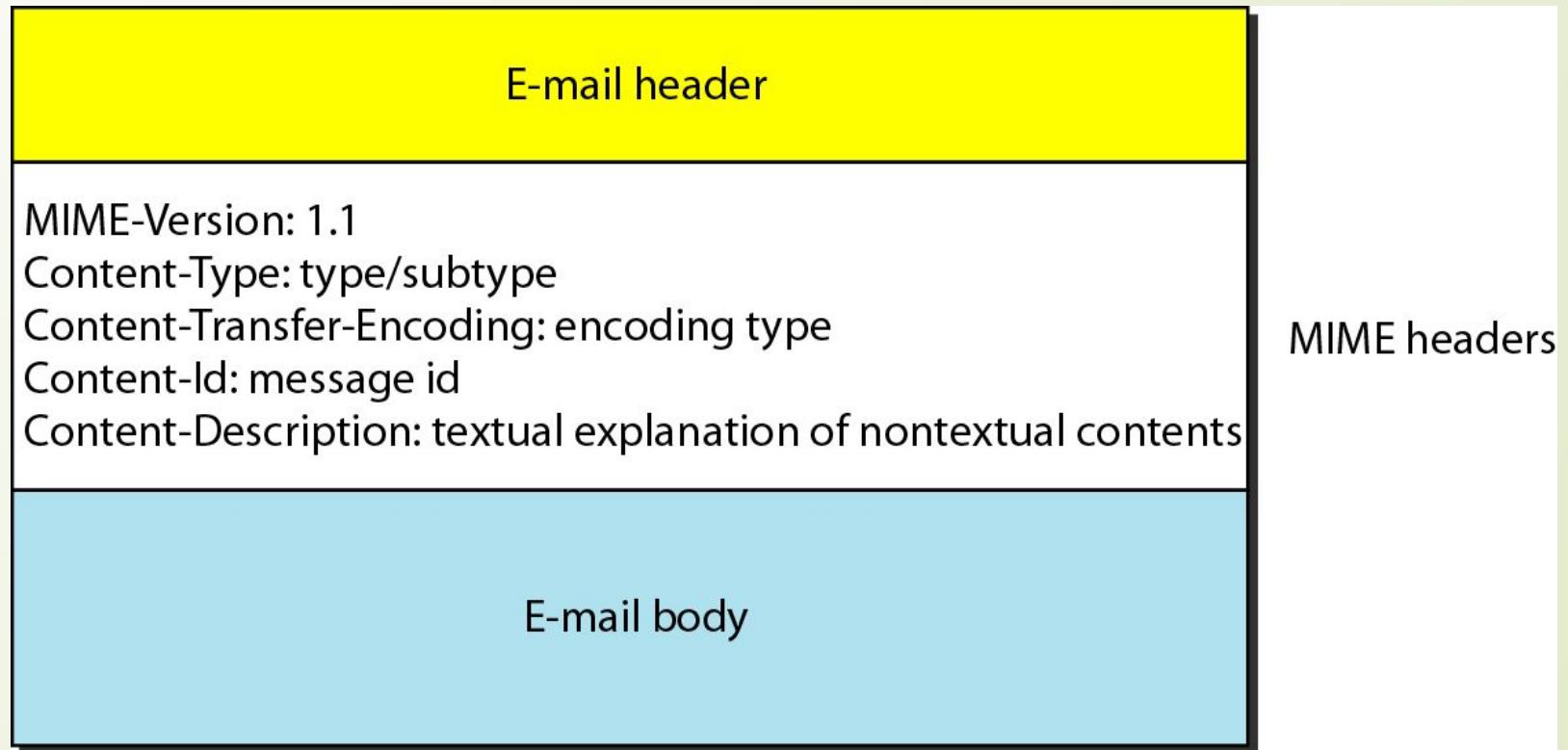


Table 13.5 *Data types and subtypes in MIME*

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)

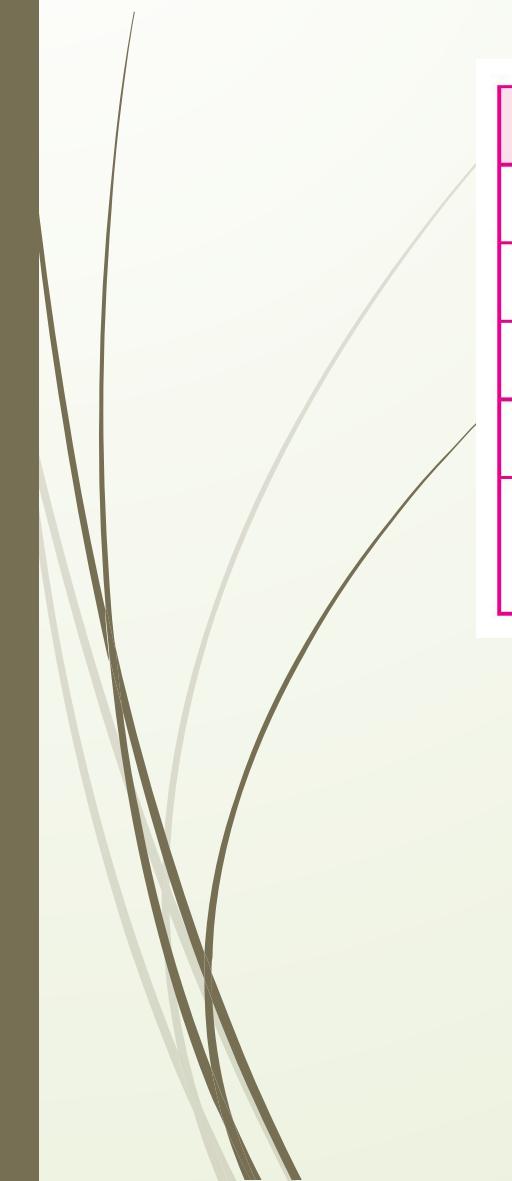


Table 13.6 *Content-transfer-encoding*

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

Figure 13.16 *SMTP range*

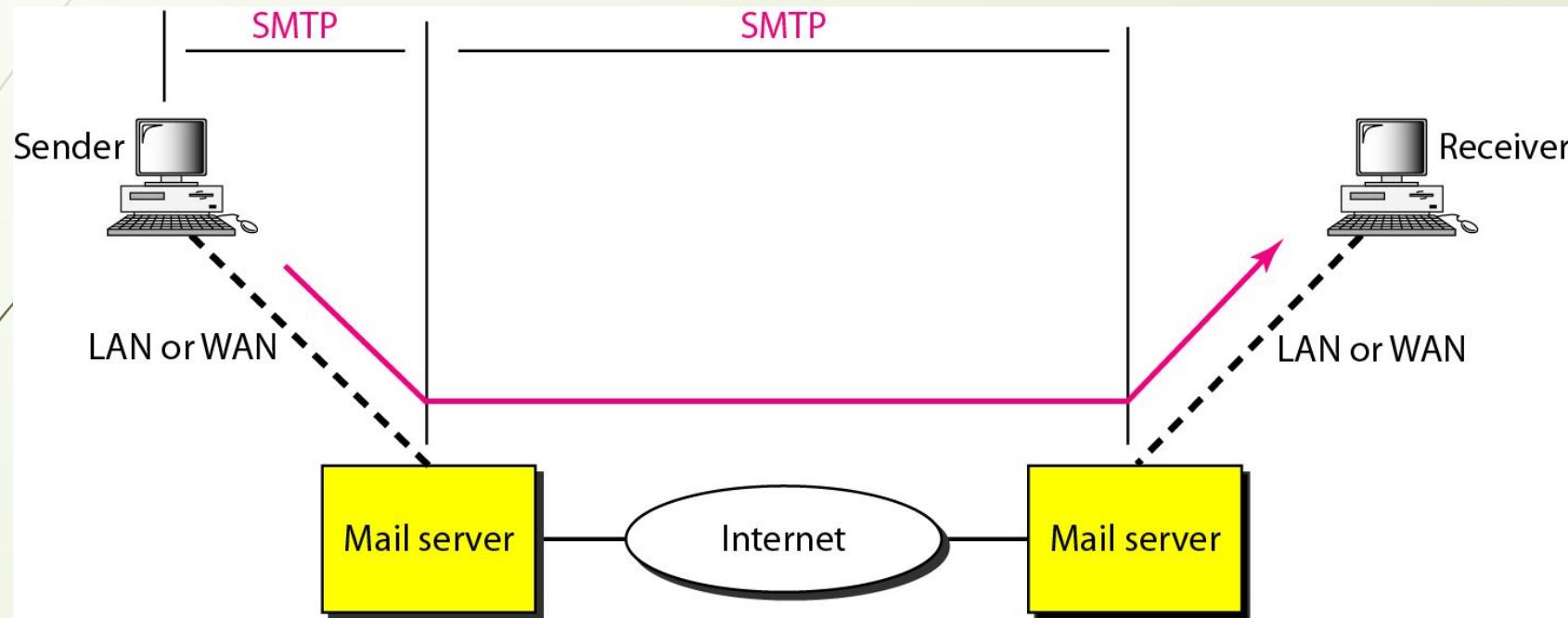


Figure 13.17 *Commands and responses*

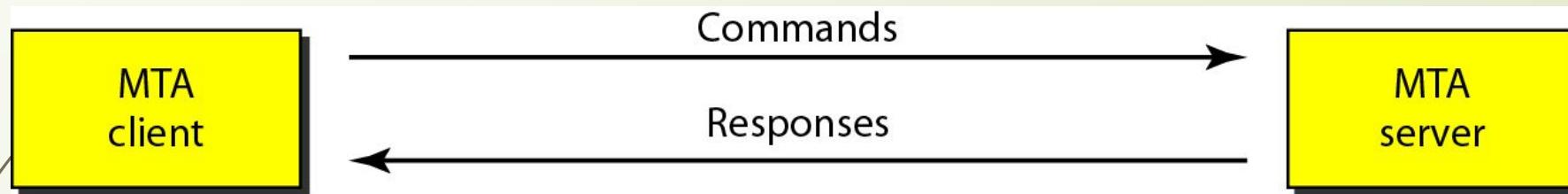




Figure 13.18 *Command format*

Keyword: argument(s)



Table 13.7 *Commands*

<i>Keyword</i>	<i>Argument(s)</i>
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 13.8 Responses

<i>Code</i>	<i>Description</i>
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 13.8 Responses (continued)

<i>Code</i>	<i>Description</i>
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 13.19 POST OFFICE PROTOCOL *POP3* and Internet Mail Access Protocol version 4 (*IMAP4*)

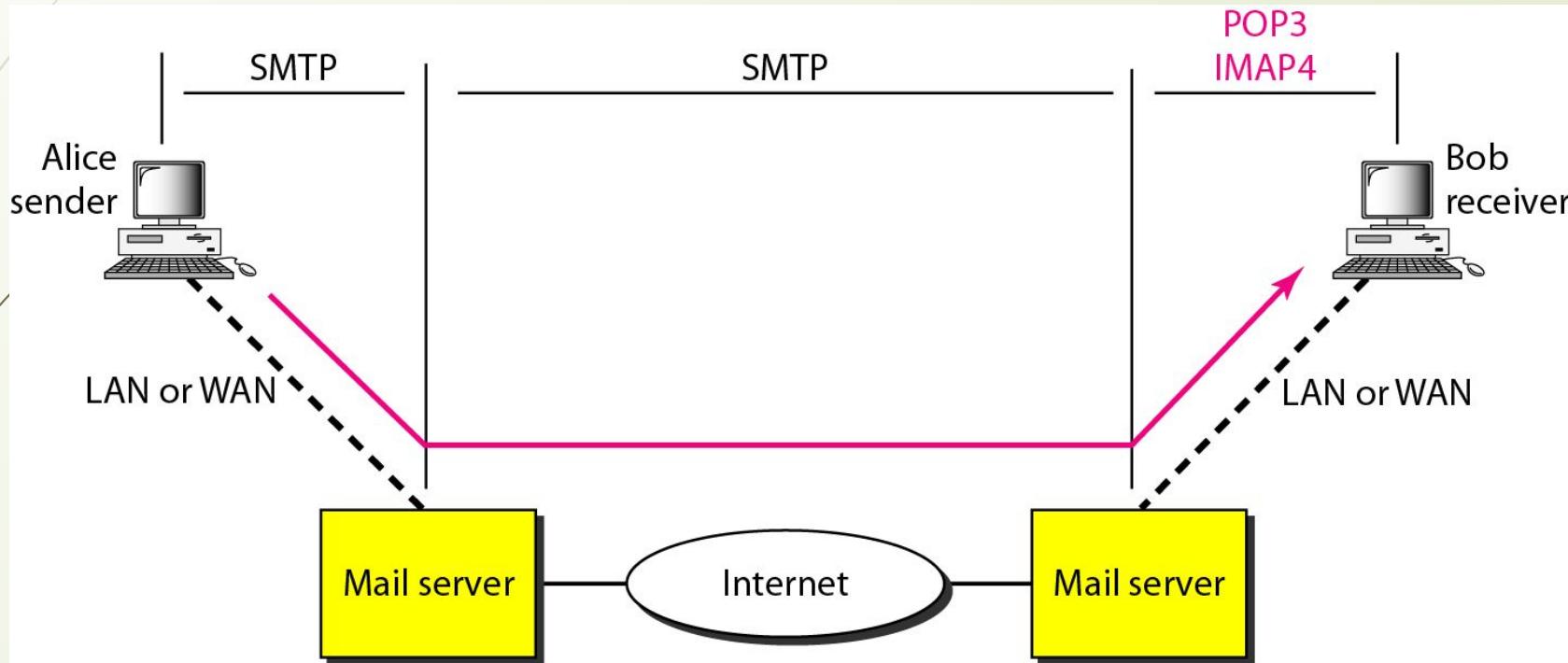
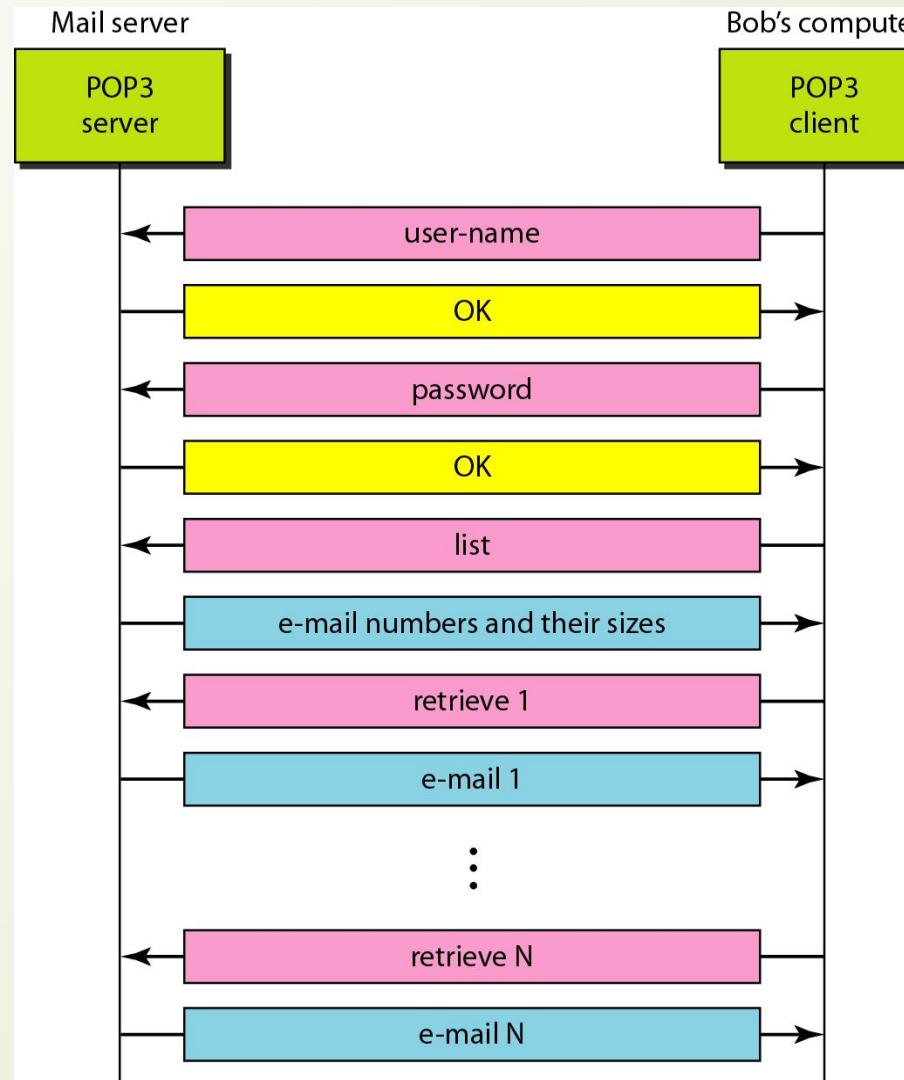


Figure 13.20 *The exchange of commands and responses in POP3*



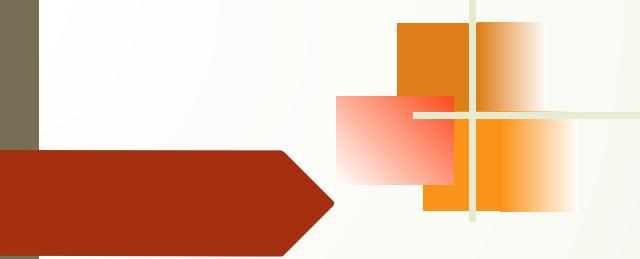
13-3 FILE TRANSFER

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 13.21 FTP

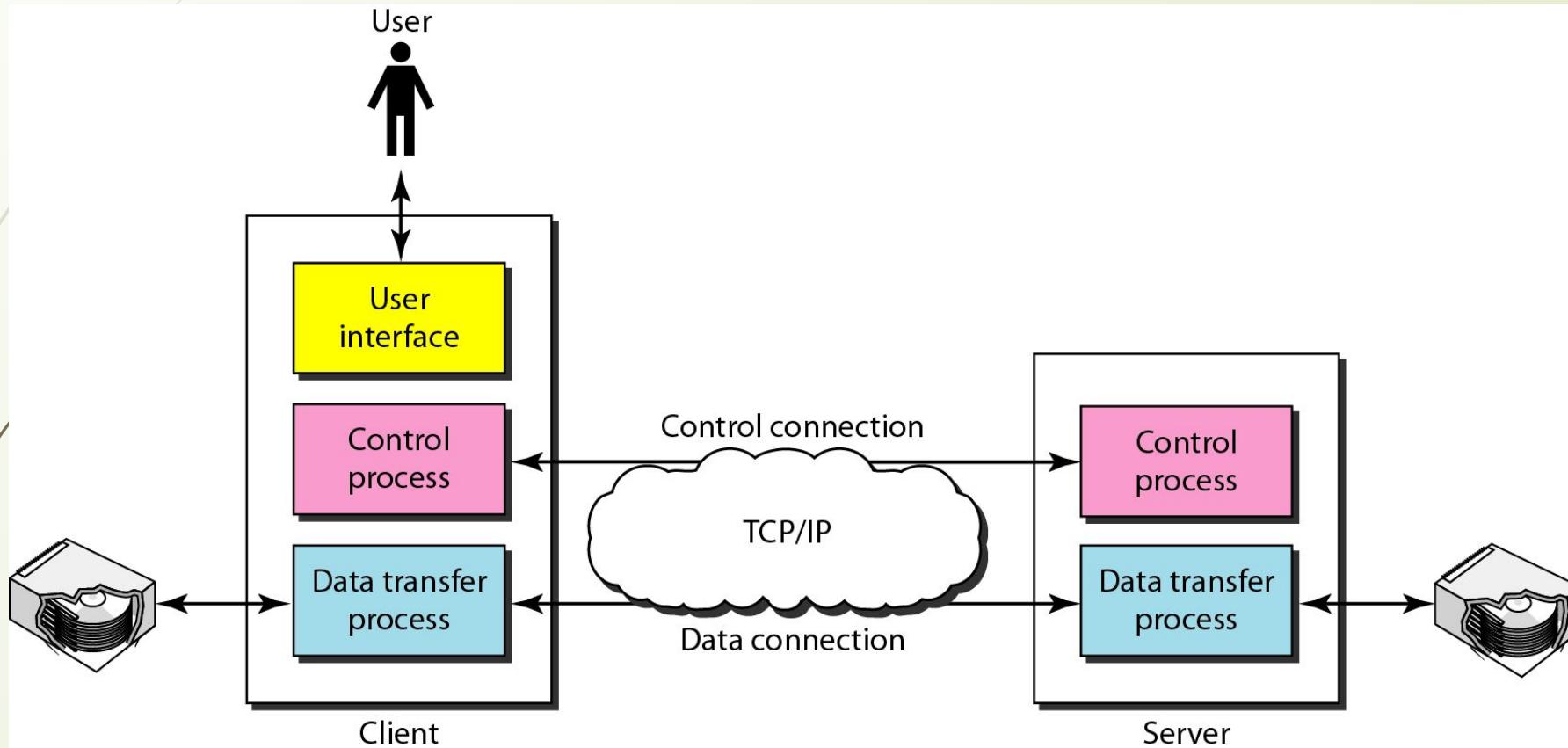


Figure 13.22 Using the control connection

FTP uses the same approach as SMTP to communicate across the control connection.

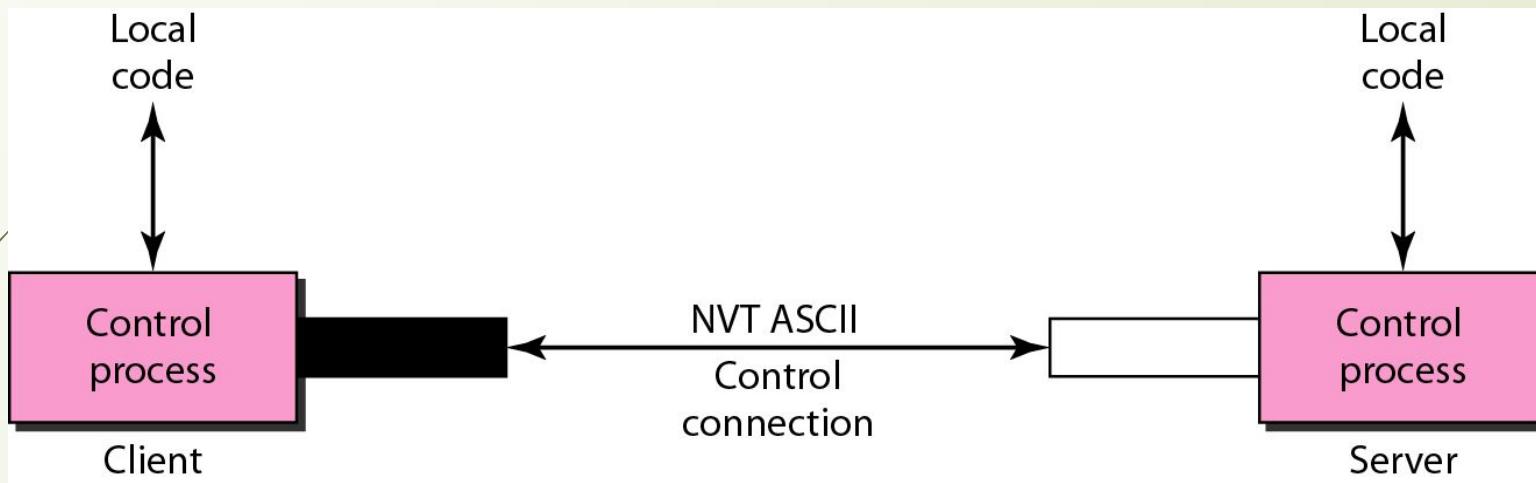
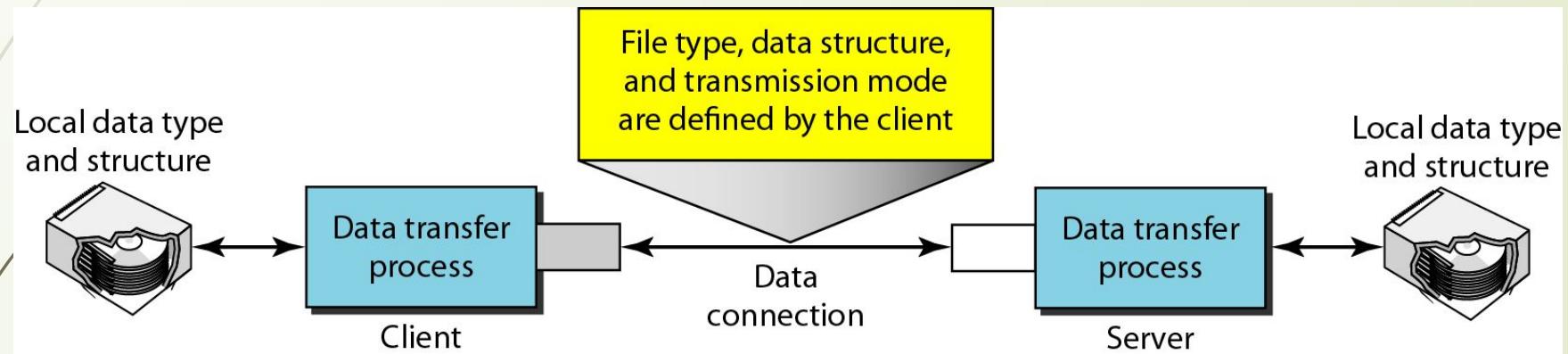
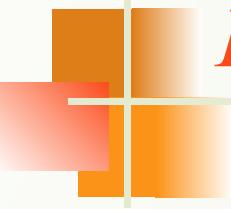


Figure 13.23 *Using the data connection*

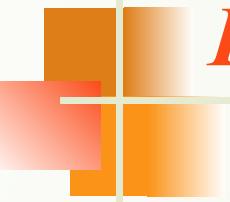




Example 13.4

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

- 1. After the control connection is created, the FTP server sends the 220 response.*
- 2. The client sends its name.*
- 3. The server responds with 331.*



Example 13.4 (continued)

- 4.** *The client sends the password (not shown).*
- 5.** *The server responds with 230 (user log-in is OK).*
- 6.** *The client sends the list command (ls reports) to find the list of files on the directory named report.*
- 7.** *Now the server responds with 150 and opens the data connection.*
- 8.** *The server then sends the list of the files or directories on the data connection.*
- 9.** *The client sends a QUIT command.*
- 10.** *The server responds with 221.*

Example 13.4 (continued)

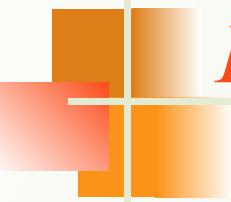
```
$ ftp voyager.deanza.fhda.edu  
Connected to voyager.deanza.fhda.edu.  
220 (vsFTPd 1.2.1)  
530 Please login with USER and PASS.  
Name (voyager.deanza.fhda.edu:forouzan): forouzan  
331 Please specify the password.  
Password:  
230 Login successful.  
Remote system type is UNIX.  
Using binary mode to transfer files.  
ftp> ls reports  
227 Entering Passive Mode (153,18,17,11,238,169)  
150 Here comes the directory listing.
```

drwxr-xr-x	2	3027	411	4096 Sep 24 2002	business
drwxr-xr-x	2	3027	411	4096 Sep 24 2002	personal
drwxr-xr-x	2	3027	411	4096 Sep 24 2002	school

```
226 Directory send OK.
```

```
ftp> quit
```

```
221 Goodbye.
```

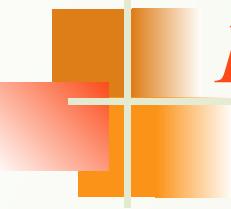


Example 13.5

We show an example of anonymous FTP. We assume that some public data are available at internic.net.

```
$ ftp internic.net
Connected to internic.net
220 Server ready
Name: anonymous
331 Guest login OK, send "guest" as password
Password: guest
```

continued on next slide



Example 13.5 (continued)

```
ftp > pwd
```

```
257 '/' is current directory
```

```
ftp > ls
```

```
200 OK
```

```
150 Opening ASCII mode
```

```
bin
```

```
...
```

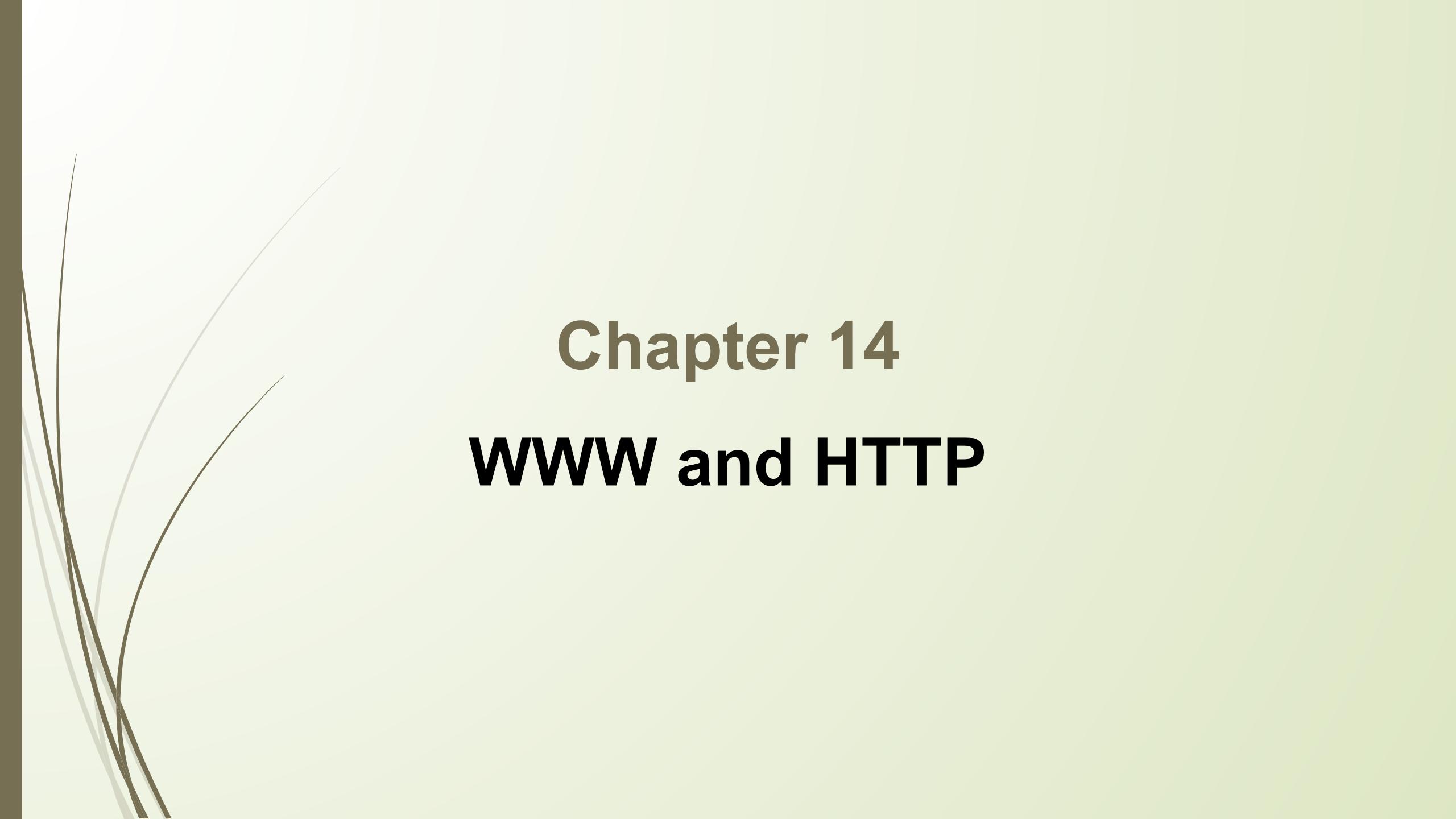
```
...
```

```
...
```

```
ftp > close
```

```
221 Goodbye
```

```
ftp > quit
```



Chapter 14

www and HTTP

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 14.1 *Architecture of WWW*

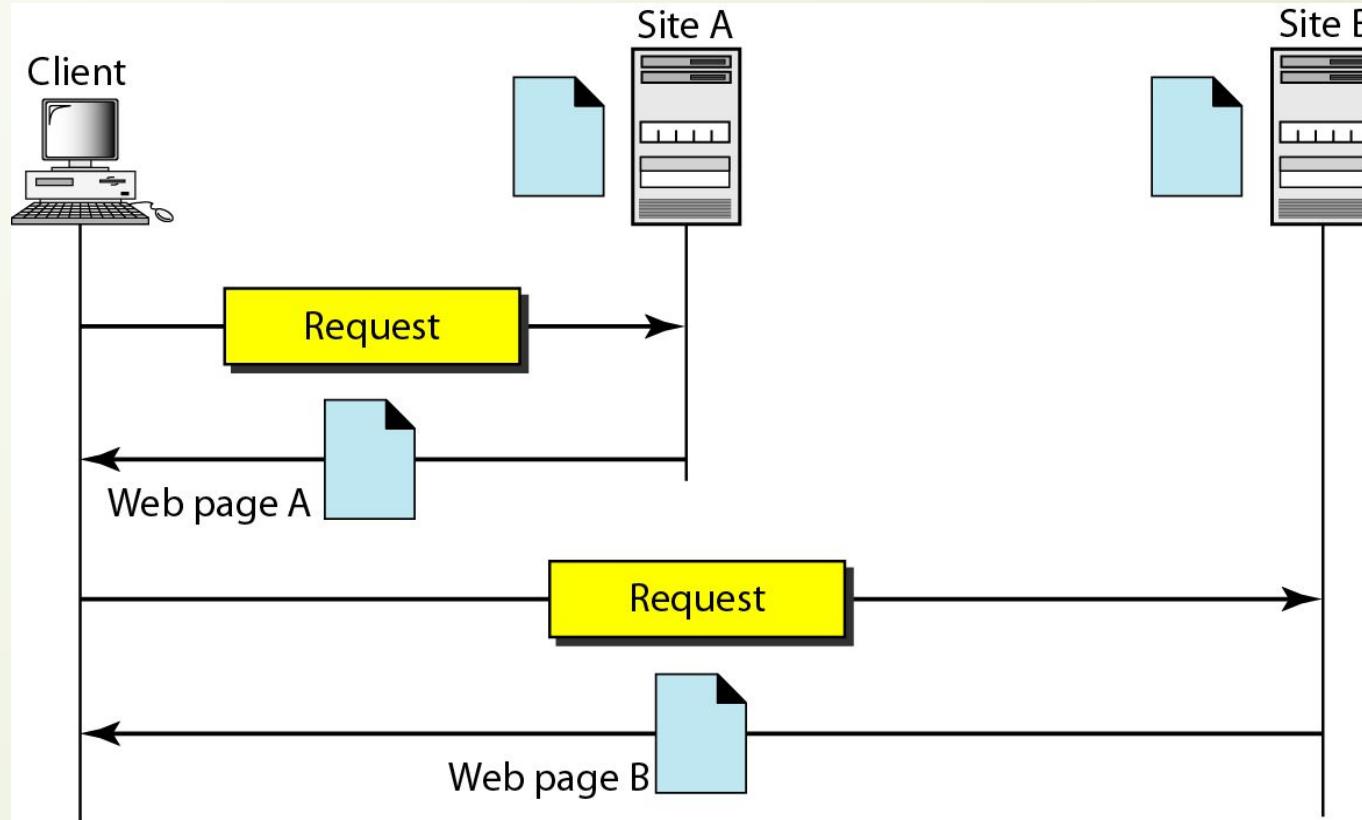


Figure 14.2 *Browser*

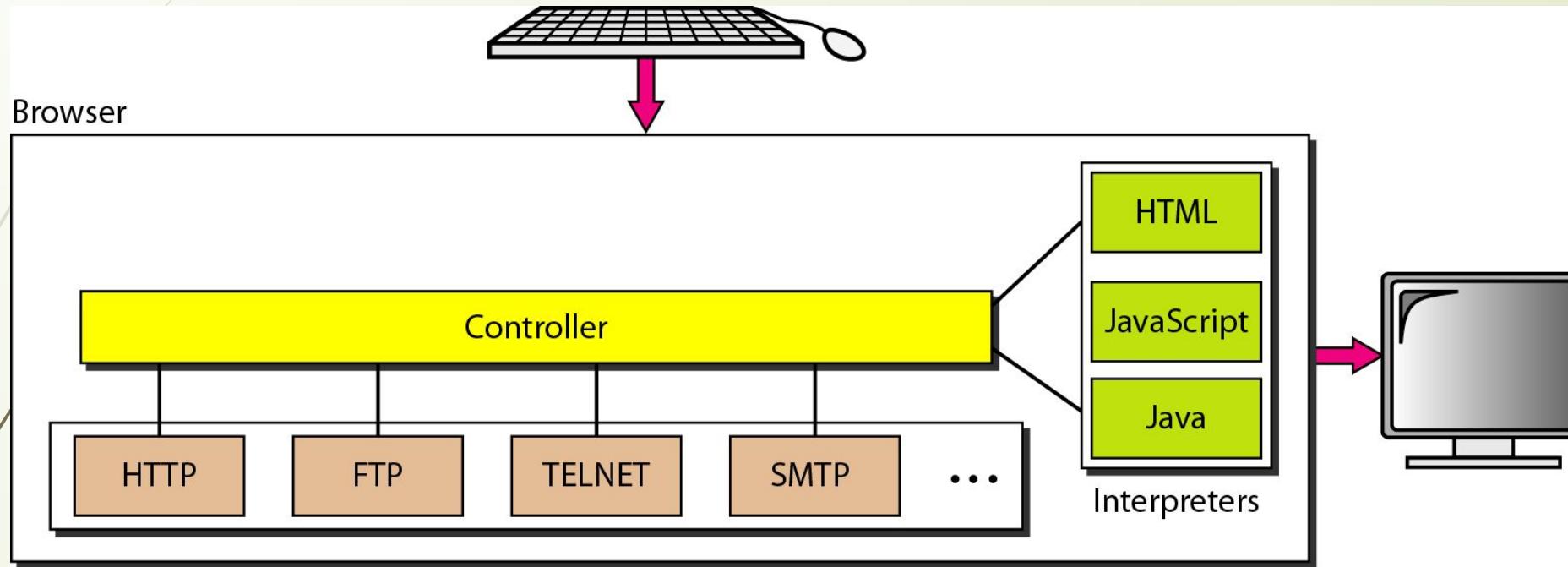
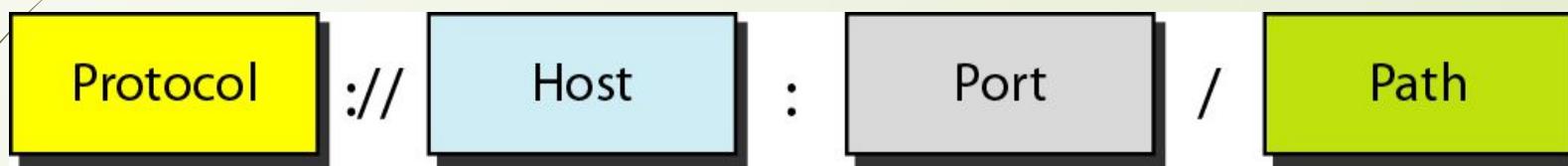




Figure 14.3 URL



Cookies

The World Wide Web was originally designed as a stateless entity. A client sends a request; a server responds. Their relationship is over. The original design of WWW, retrieving publicly available documents, exactly fits this purpose. Today the Web has other functions; some are listed here.

- 1 . Some websites need to allow access to registered clients only.
2. Websites are being used as electronic stores that allow users to browse through the store, select wanted items, put them in an electronic cart, and pay at the end with a credit card.
3. Some websites are used as portals: the user selects the Web pages he wants to see.
4. Some websites are just advertising.

Creation and Storage of Cookies

The creation and storage of cookies depend on the implementation; however, the principle is the same.

1. When a server receives a request from a client, it stores information about the client in a file or a string. The information may include the domain name of the client, the contents of the cookie (information the server has gathered about the client such as name, registration number, and so on), a timestamp, and other information depending on the implementation.
2. The server includes the cookie in the response that it sends to the client.
3. When the client receives the response, the browser stores the cookie in the cookie directory, which is sorted by the domain server name.

Using Cookies

When a client sends a request to a server, the browser looks in the cookie directory to see if it can find a cookie sent by that server. If found, the cookie is included in the request. When the server receives the request, it knows that this is an old client, not a new one. Note that the contents of the cookie are never read by the browser or disclosed to the user.

14-2 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 14.4 *Static document*

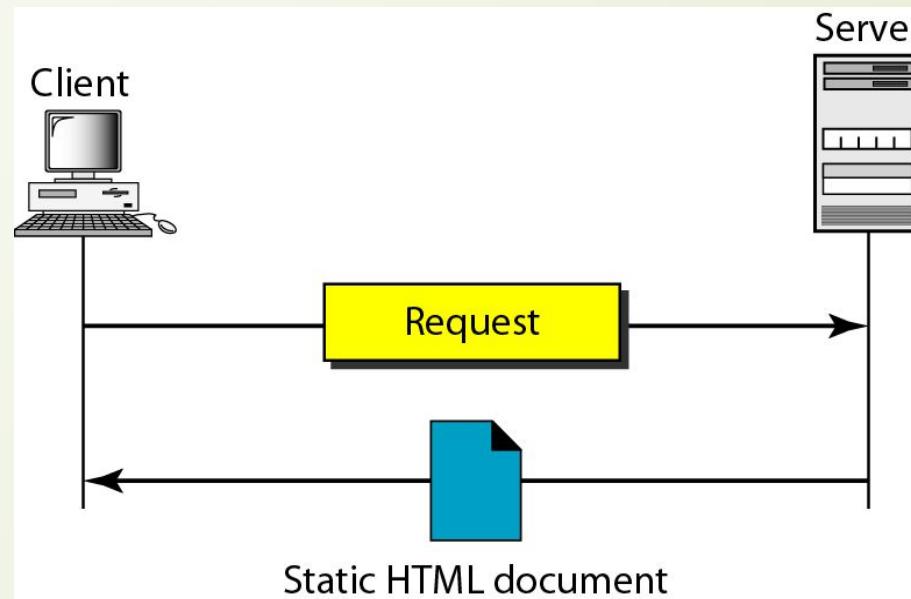




Figure 14.5 *Boldface tags*

Bold tag

End bold

**** This is the text to be boldfaced.****

Figure 14.6 *Effect of boldface tags*

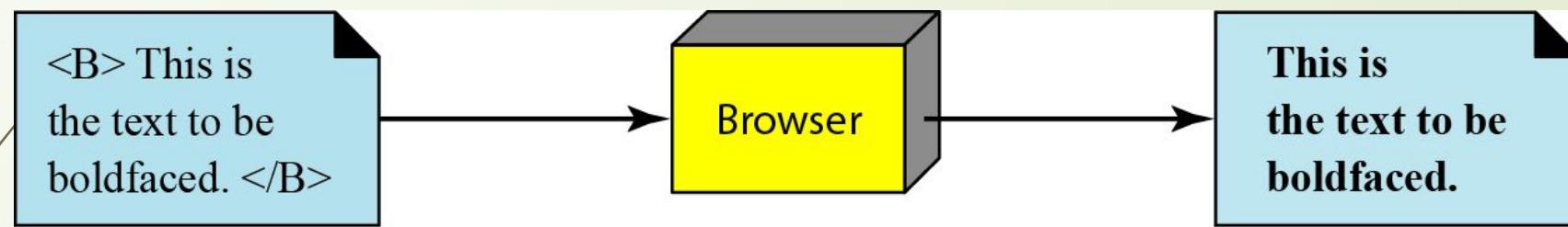


Figure 14.7 *Beginning and ending tags*

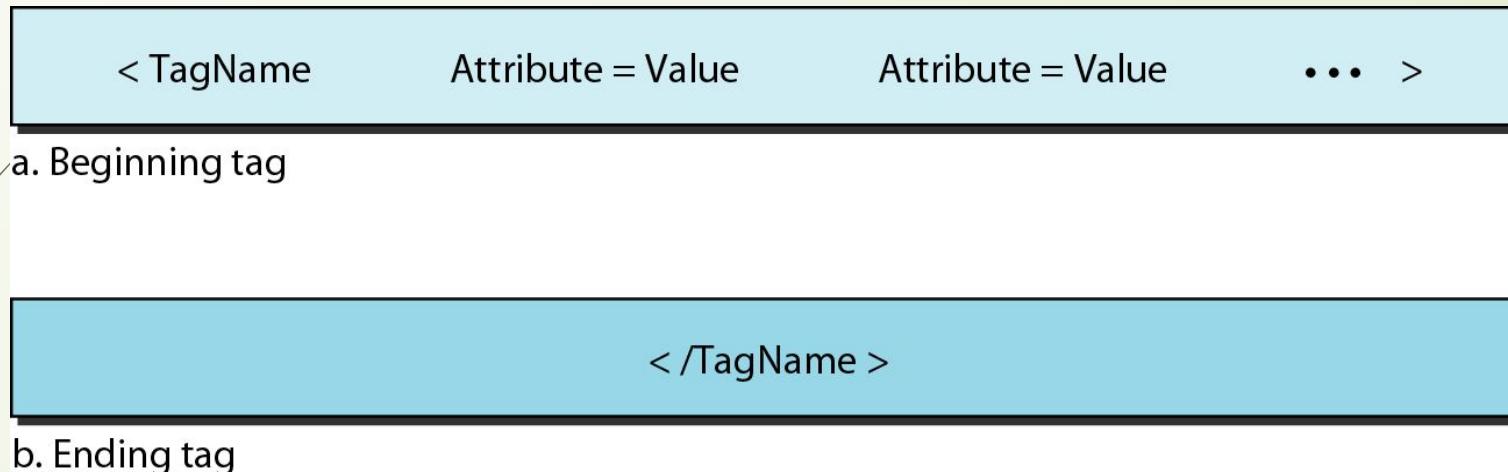
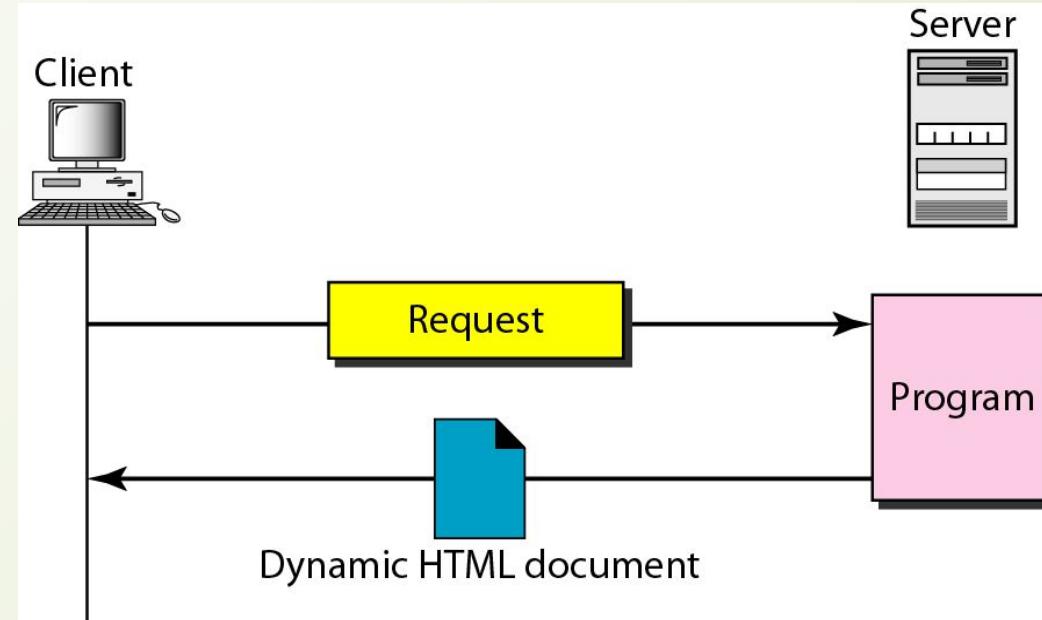


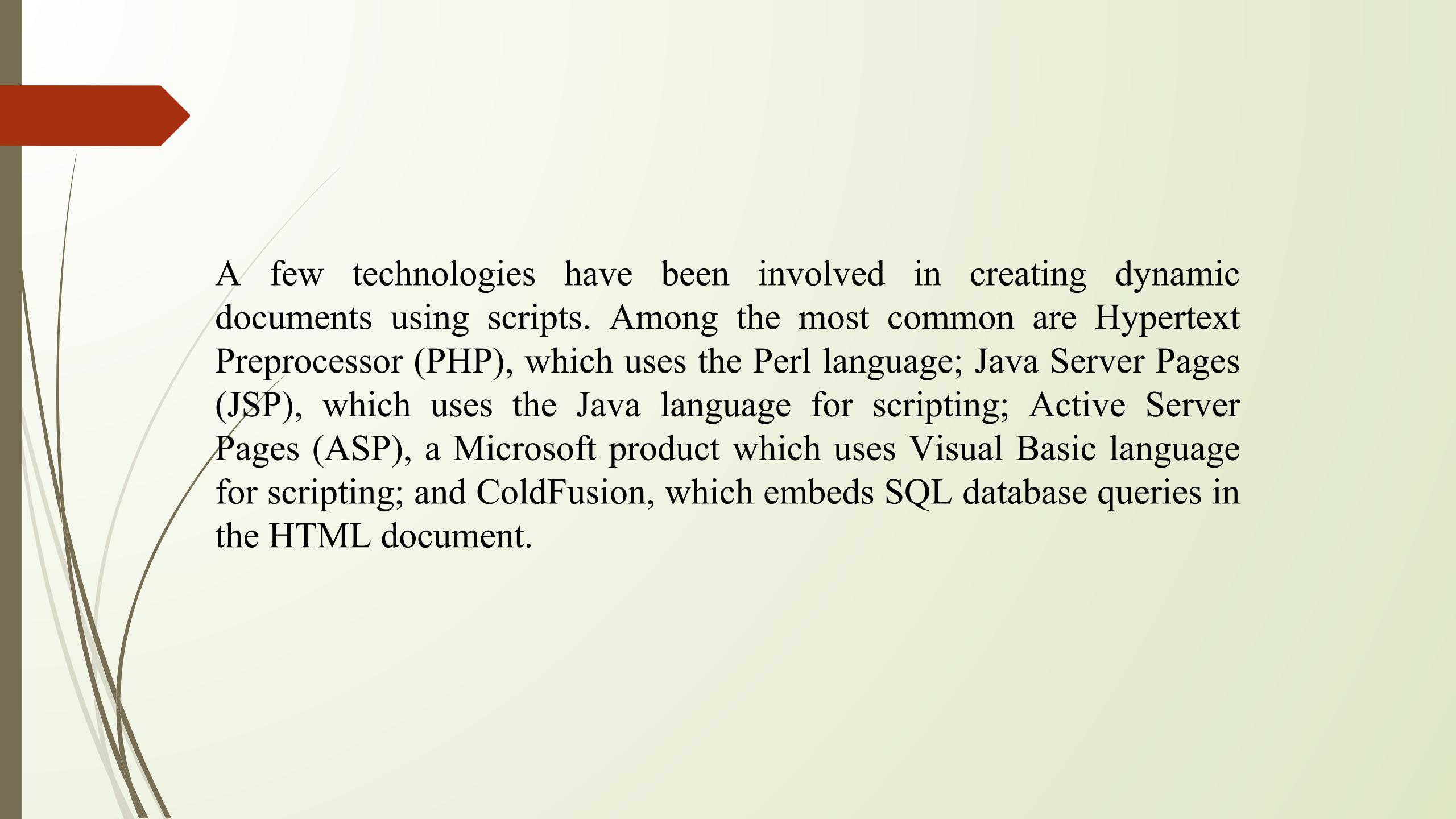
Figure 14.8 *Dynamic document using CGI*



A very simple example of a dynamic document is the retrieval of the time and date from a server.

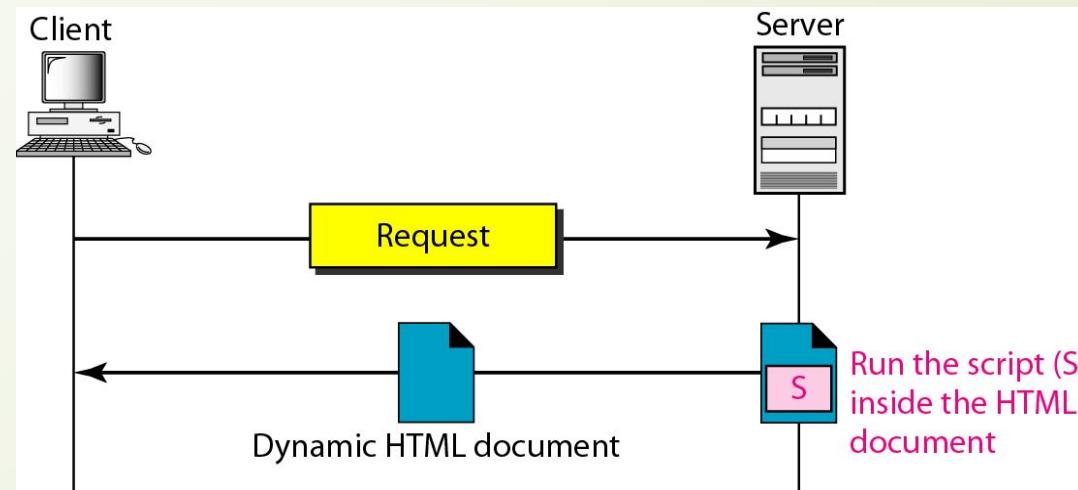


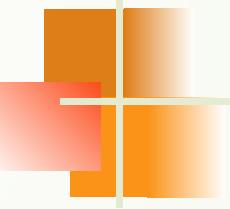
The problem with CGI technology is the inefficiency that results if part of the dynamic document that is to be created is fixed and not changing from request to request. For example, assume that we need to retrieve a list of spare parts, their availability, and prices for a specific car brand. Although the availability and prices vary from time to time, the name, description, and the picture of the parts are fixed. If we use CGI, the program must create an entire document each time a request is made. The solution is to create a file containing the fixed part of the document using IHTML and embed a script, a source code, that can be run by the server to provide the varying availability and price section.



A few technologies have been involved in creating dynamic documents using scripts. Among the most common are Hypertext Preprocessor (PHP), which uses the Perl language; Java Server Pages (JSP), which uses the Java language for scripting; Active Server Pages (ASP), a Microsoft product which uses Visual Basic language for scripting; and ColdFusion, which embeds SQL database queries in the HTML document.

Figure 14.9 Dynamic document using server-site script





Note

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

Figure 14.10 *Active document using Java applet*

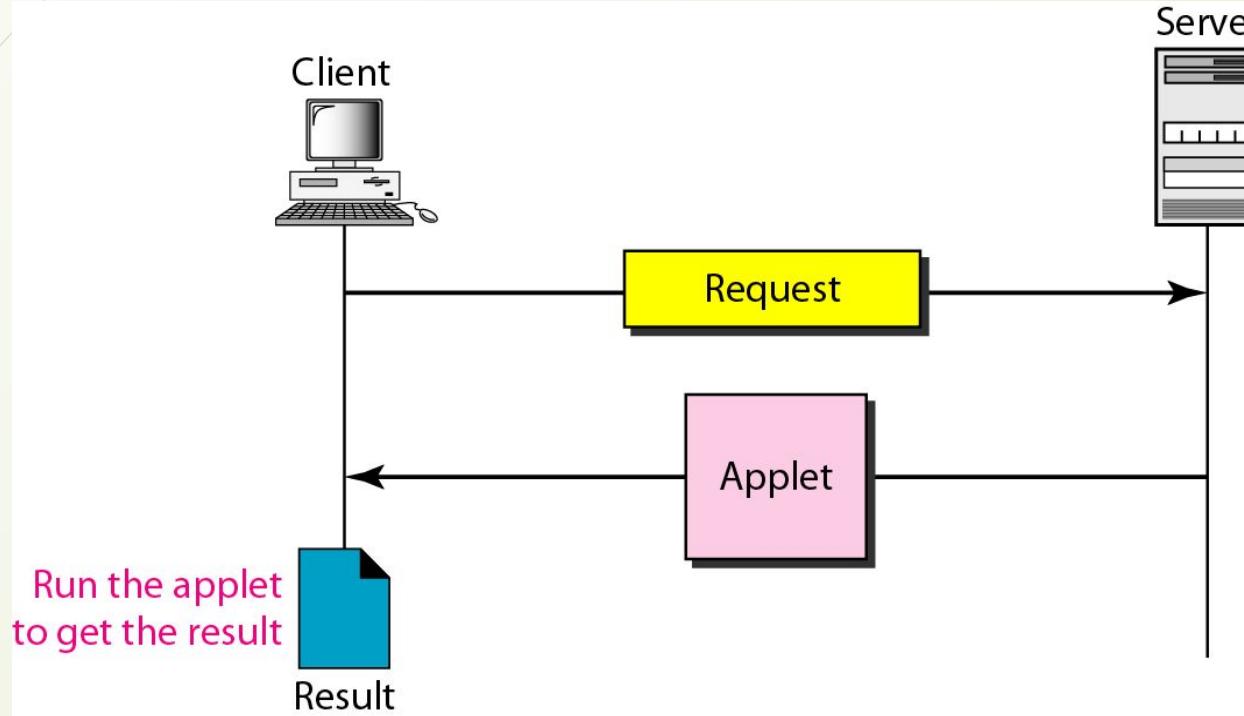
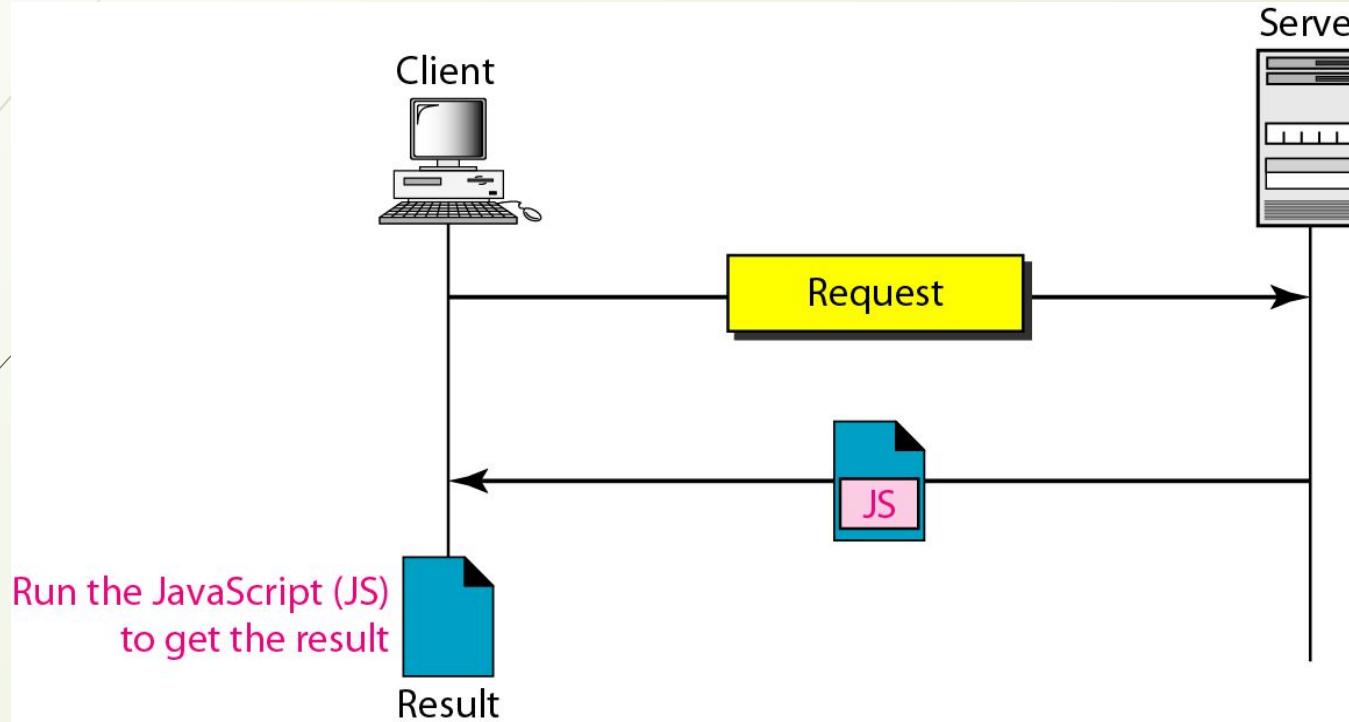
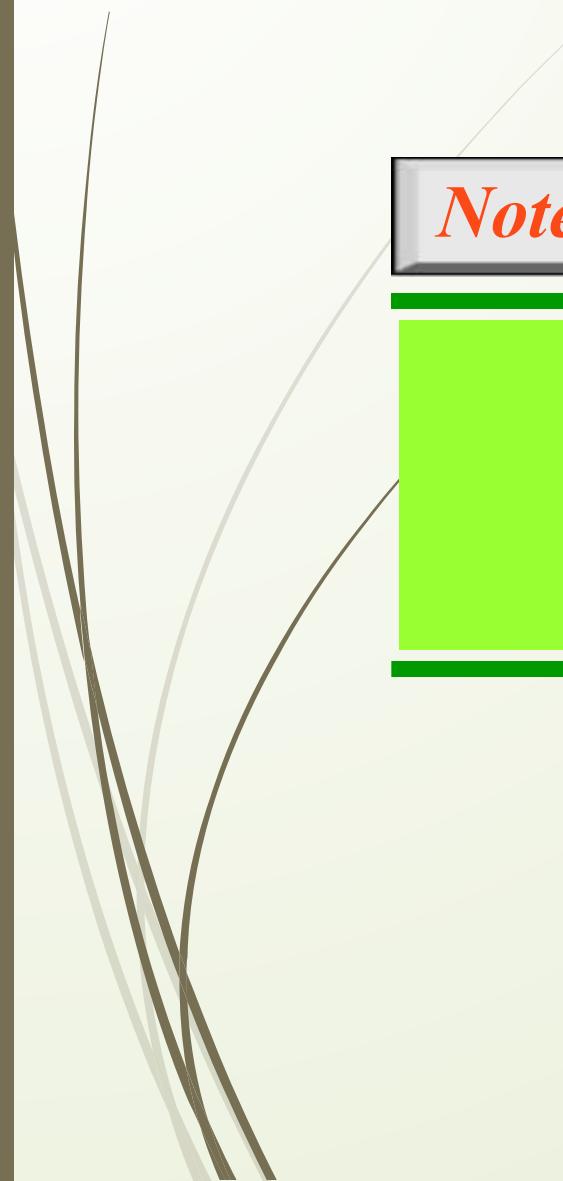
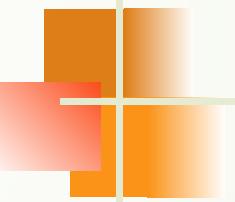


Figure 14.11 *Active document using client-site script*



The idea of scripts in dynamic documents can also be used for active documents. If the active part of the document is small, it can be written in a scripting language; then it can be interpreted and run by the client at the same time.



Note

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

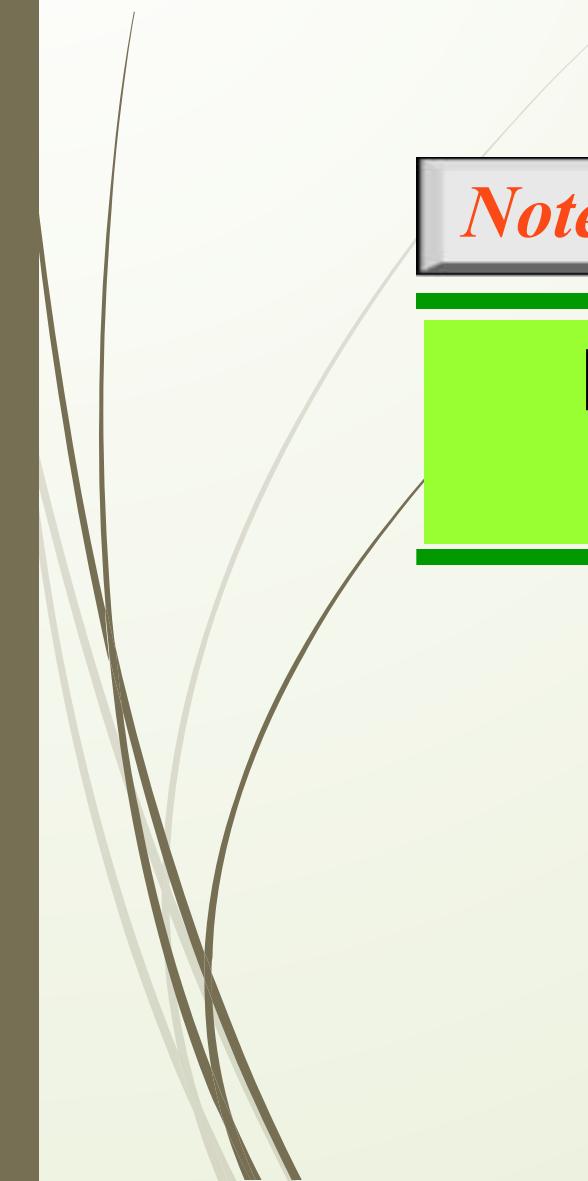
14-3 HTTP

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:

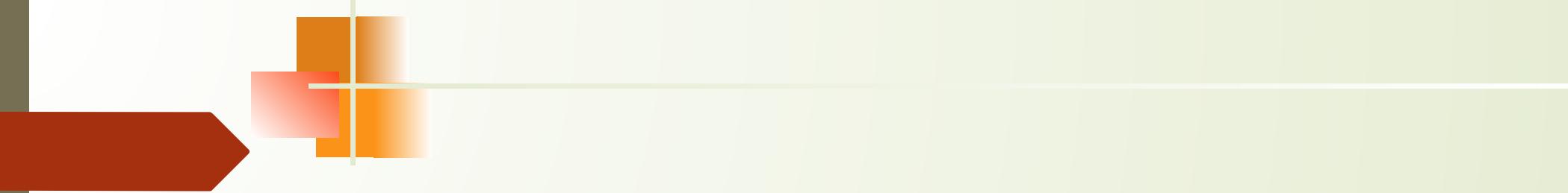
HTTP Transaction

Persistent Versus Nonpersistent Connection



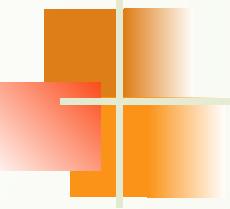
Note

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



Note

It is similar to FTP because It transfers files and uses the services of TCP. However, it is much simpler than FTP because it uses only one TCP connection. There is no separate control connection.



Note

HTTP is like SMTP because the data transferred between the client and the server look like SMTP messages. In addition, the format of the messages is controlled by MIME-like headers.

Figure 14.12 *HTTP transaction*

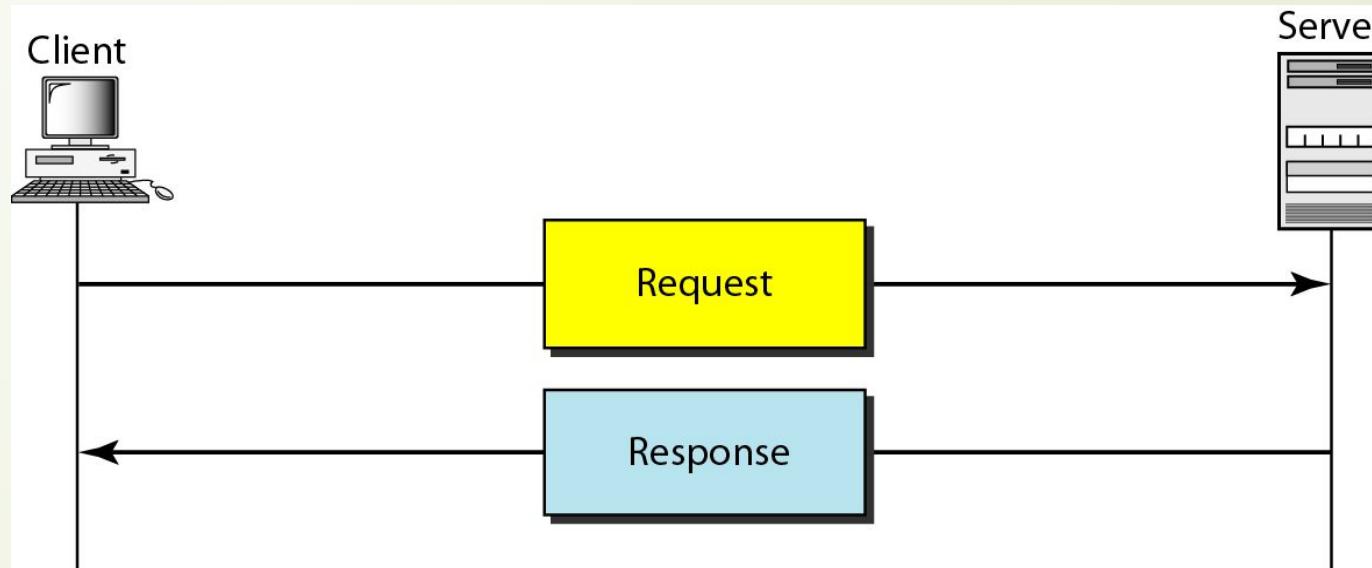




Figure 14.13 Request and response messages

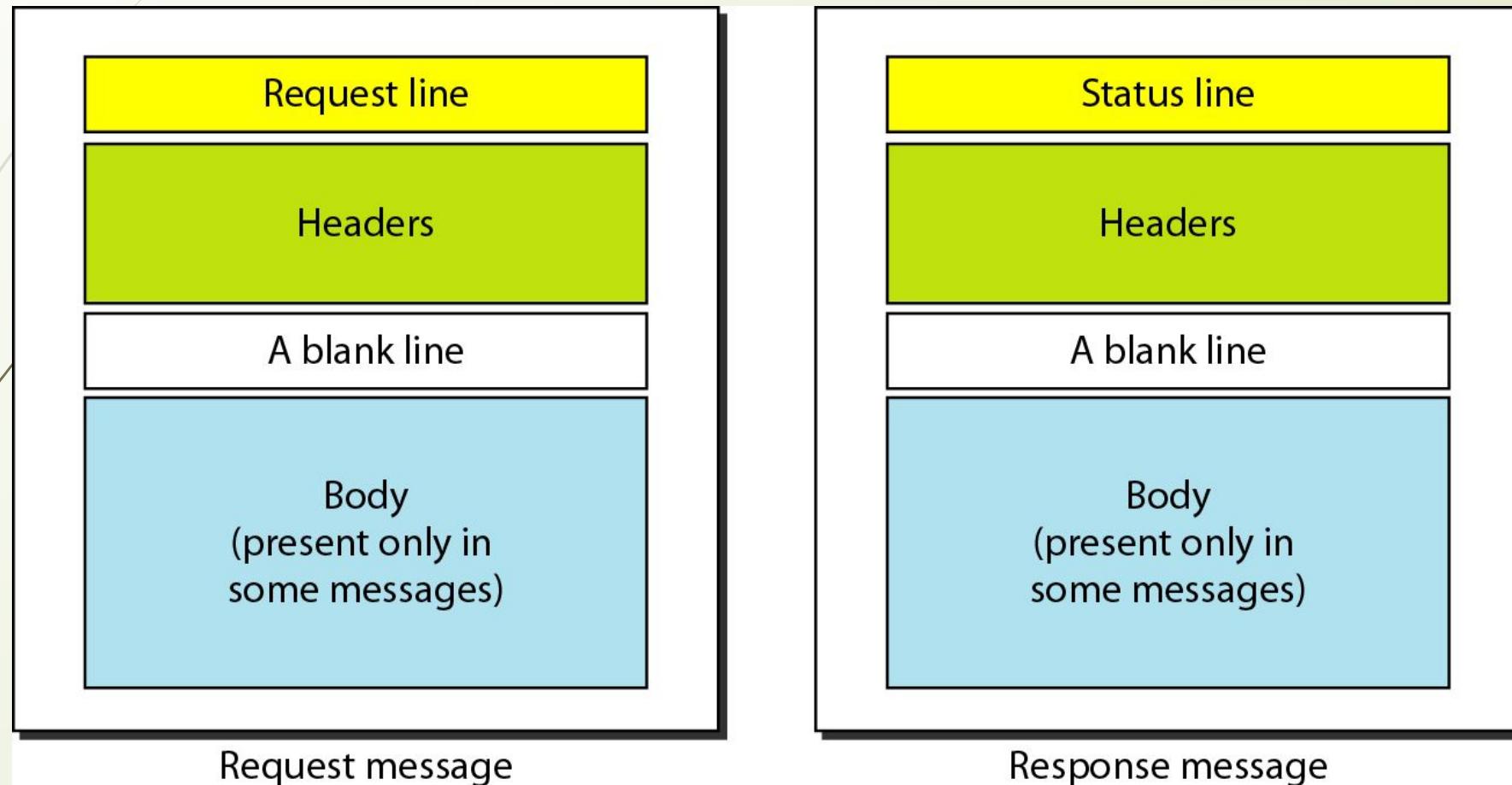


Figure 14.14 Request and status lines

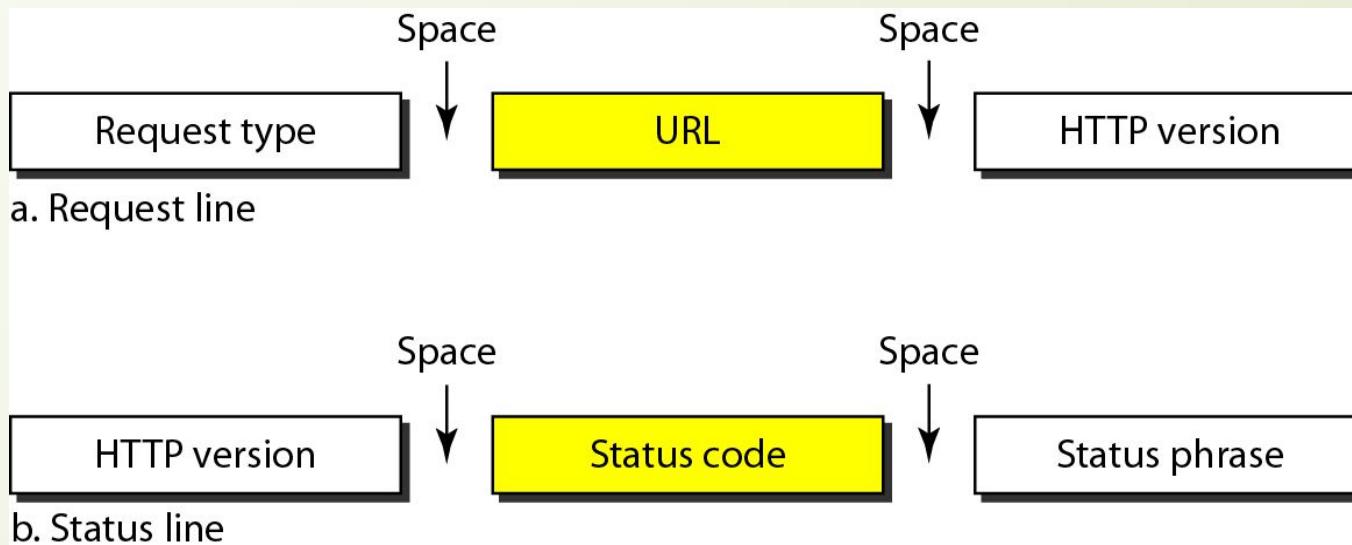




Table 14.1 *Methods*

<i>Method</i>	<i>Action</i>
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 14.2 *Status codes*

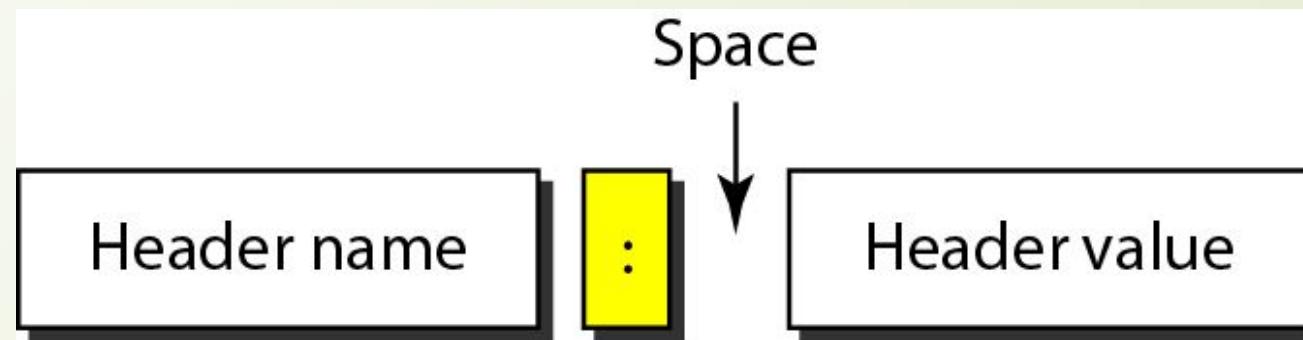
<i>Code</i>	<i>Phrase</i>	<i>Description</i>
Informational		
100	Continue	The initial part of the request has been received, and the client may continue with its request.
101	Switching	The server is complying with a client request to switch protocols defined in the upgrade header.
Success		
200	OK	The 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 14.2 Status codes (*continued*)

<i>Code</i>	<i>Phrase</i>	<i>Description</i>
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.

Figure 14.15 *Header format*



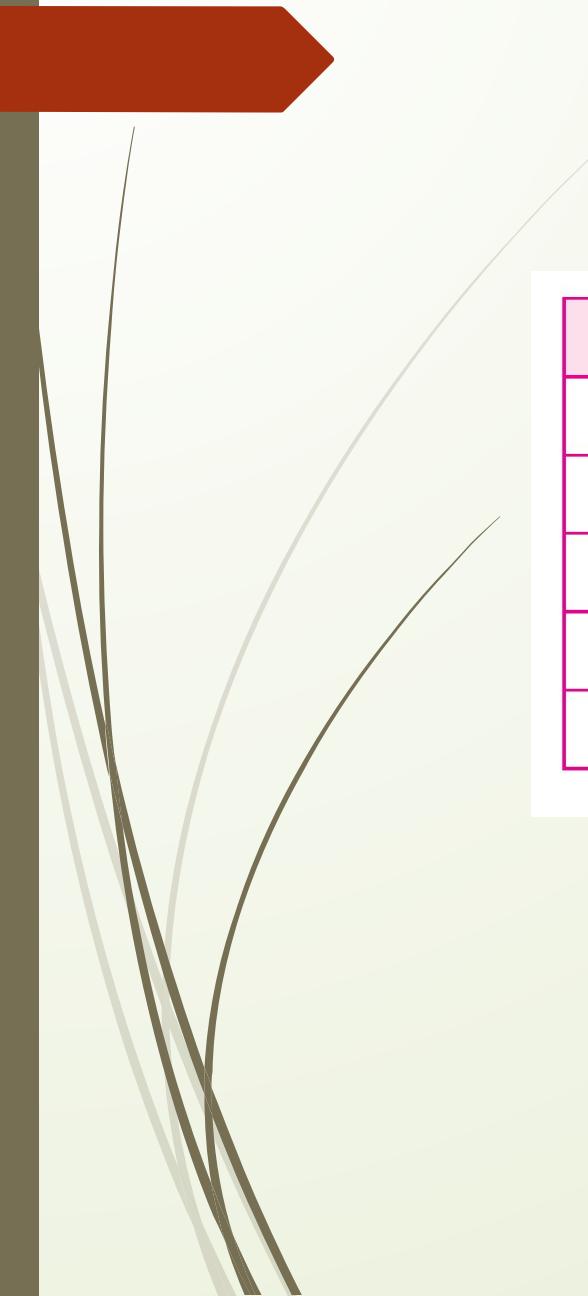


Table 14.3 *General headers*

<i>Header</i>	<i>Description</i>
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 14.4 Request headers

<i>Header</i>	<i>Description</i>
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



Table 14.5 Response headers

<i>Header</i>	<i>Description</i>
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

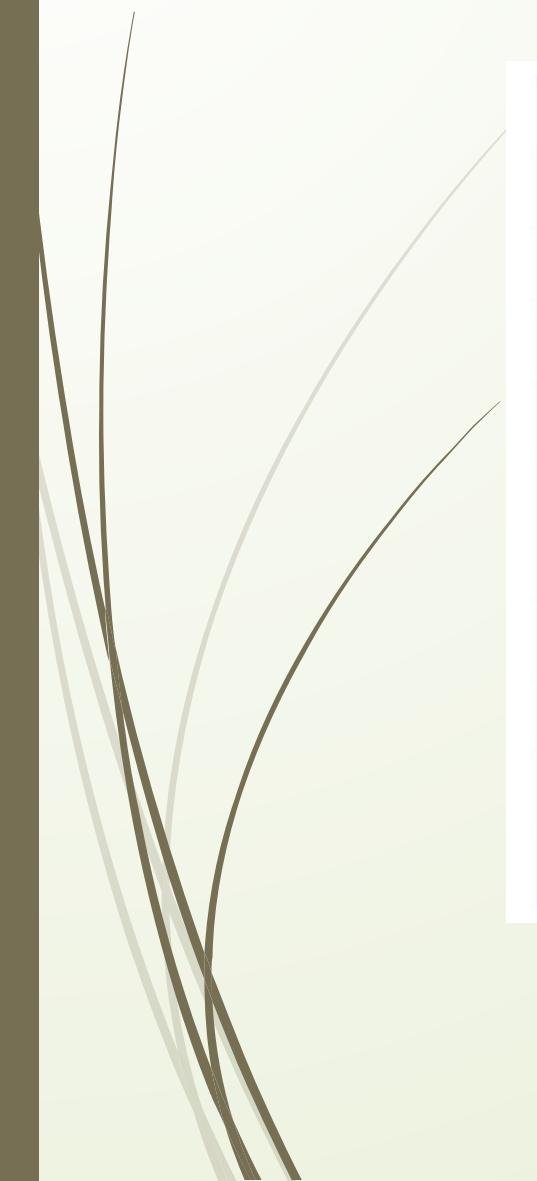
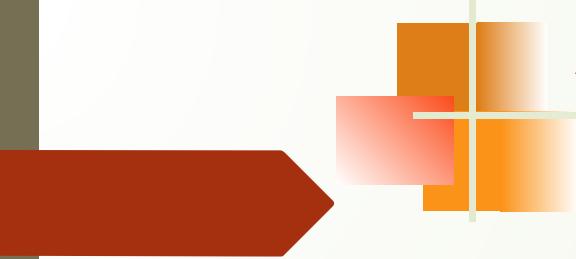


Table 14.6 *Entity headers*

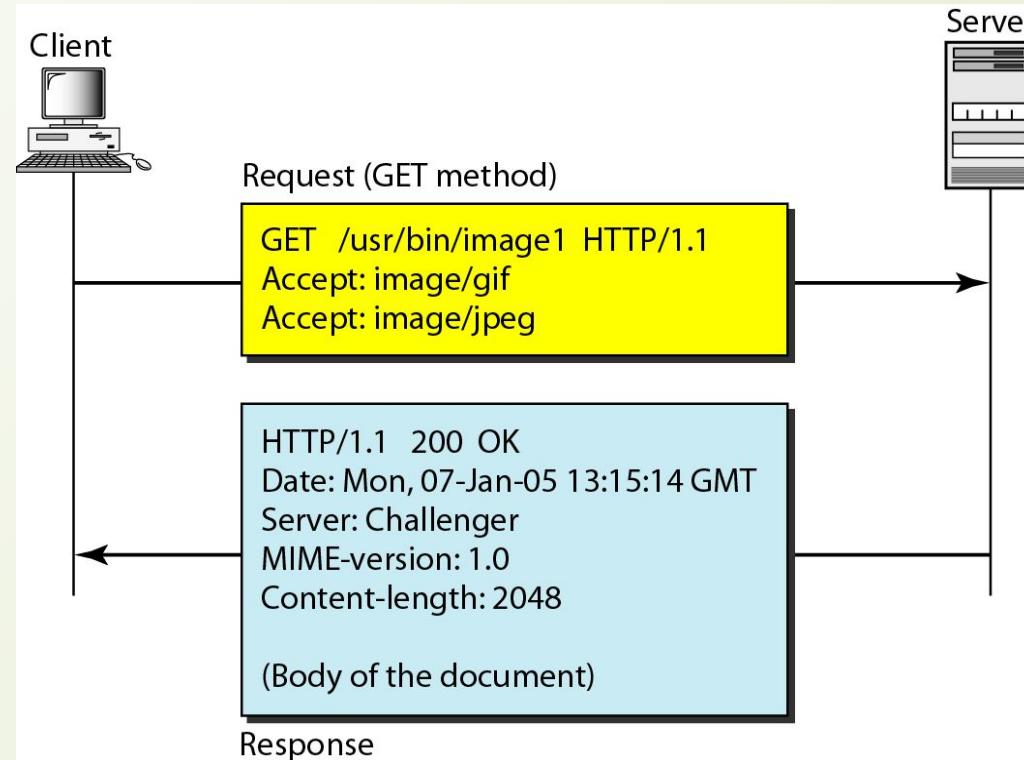
<i>Header</i>	<i>Description</i>
Allow	Lists valid methods that can be used with a URL
Content-encoding	Specifies the encoding scheme
Content-language	Specifies the language
Content-length	Shows the length of the document
Content-range	Specifies the range of the document
Content-type	Specifies the medium 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

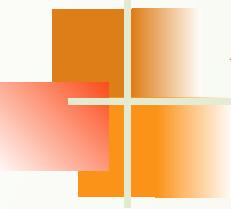


Example 14.1

This example retrieves a document. We use the GET method to retrieve an image with the path /usr/bin/image1. The request line shows the method (GET), the URL, and the HTTP version (1.1). The header has two lines that show that the client can accept images in the GIF or JPEG format. The request does not have a body. The response message contains the status line and four lines of header. The header lines define the date, server, MIME version, and length of the document. The body of the document follows the header (see Figure 27.16).

Figure 14.16 Example 14.1

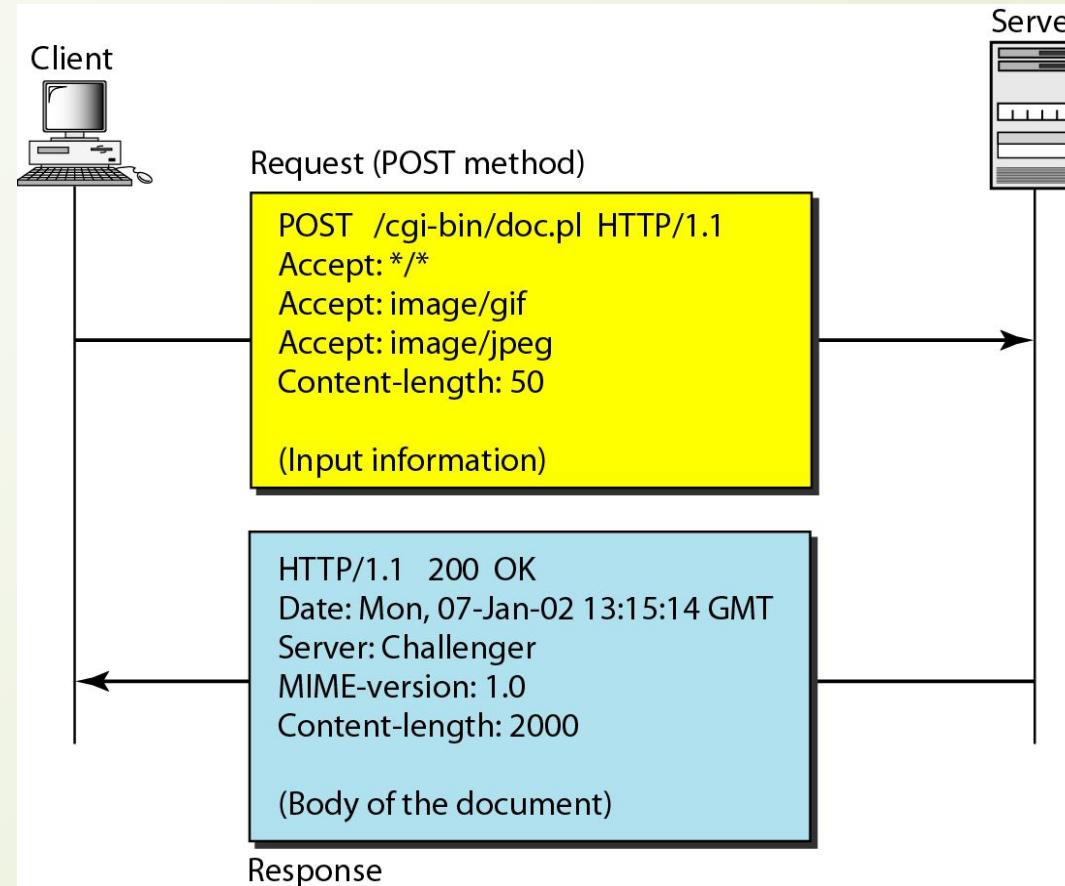


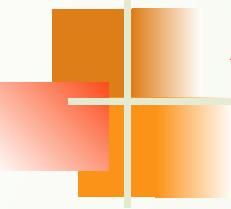


Example 14.2

In this example, the client wants to send data to the server. We use the POST method. The request line shows the method (POST), URL, and HTTP version (1.1). There are four lines of headers. The request body contains the input information. The response message contains the status line and four lines of headers. The created document, which is a CGI document, is included as the body (see Figure 14.17).

Figure 14.17 Example 14.2





Example 14.3

HTTP uses ASCII characters. A client can directly connect to a server using TELNET, which logs into port 80 (see next slide). The next three lines show that the connection is successful. We then type three lines. The first shows the request line (GET method), the second is the header (defining the host), the third is a blank, terminating the request. The server response is seven lines starting with the status line. The blank line at the end terminates the server response. The file of 14,230 lines is received after the blank line (not shown here). The last line is the output by the client.

Example 14.3 (continued)

```
$ telnet www.mhhe.com 80
```

```
Trying 198.45.24.104 . . .
```

```
Connected to www.mhhe.com (198.45.24.104).
```

```
Escape character is '^]'.  
GET /engcs/compsci/forouzan HTTP/1.1
```

```
From: forouzanbehrouz@fhda.edu
```

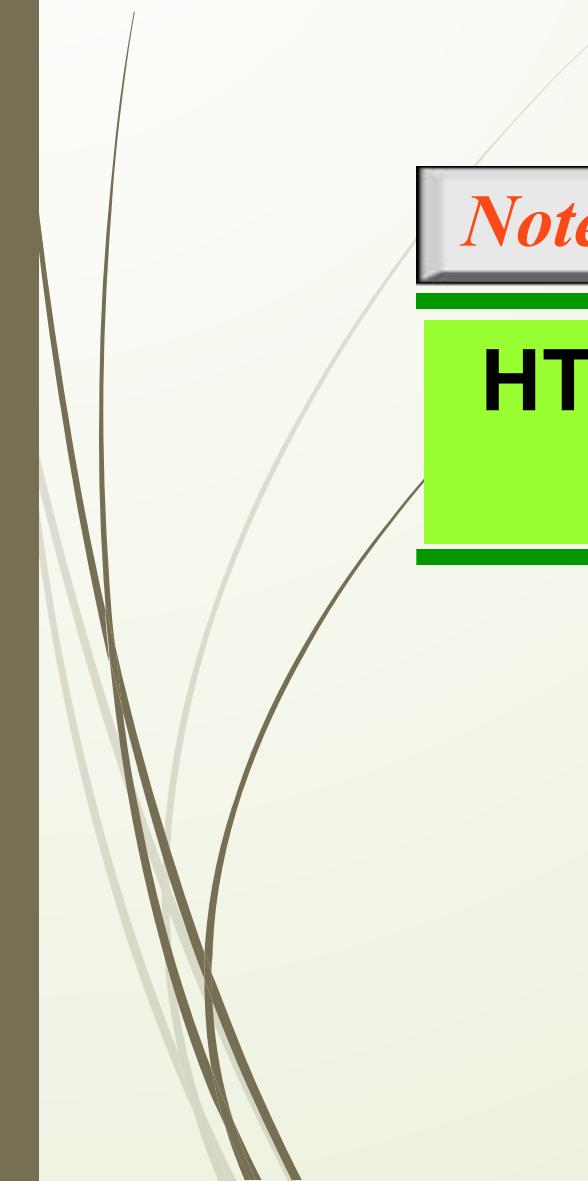
HTTP/1.1 200 OK

Date: Thu, 28 Oct 2004 16:27:46 GMT

Server: Apache/1.3.9 (Unix) ApacheJServ/1.1.2 PHP/4.1.2 PHP/3.0.18

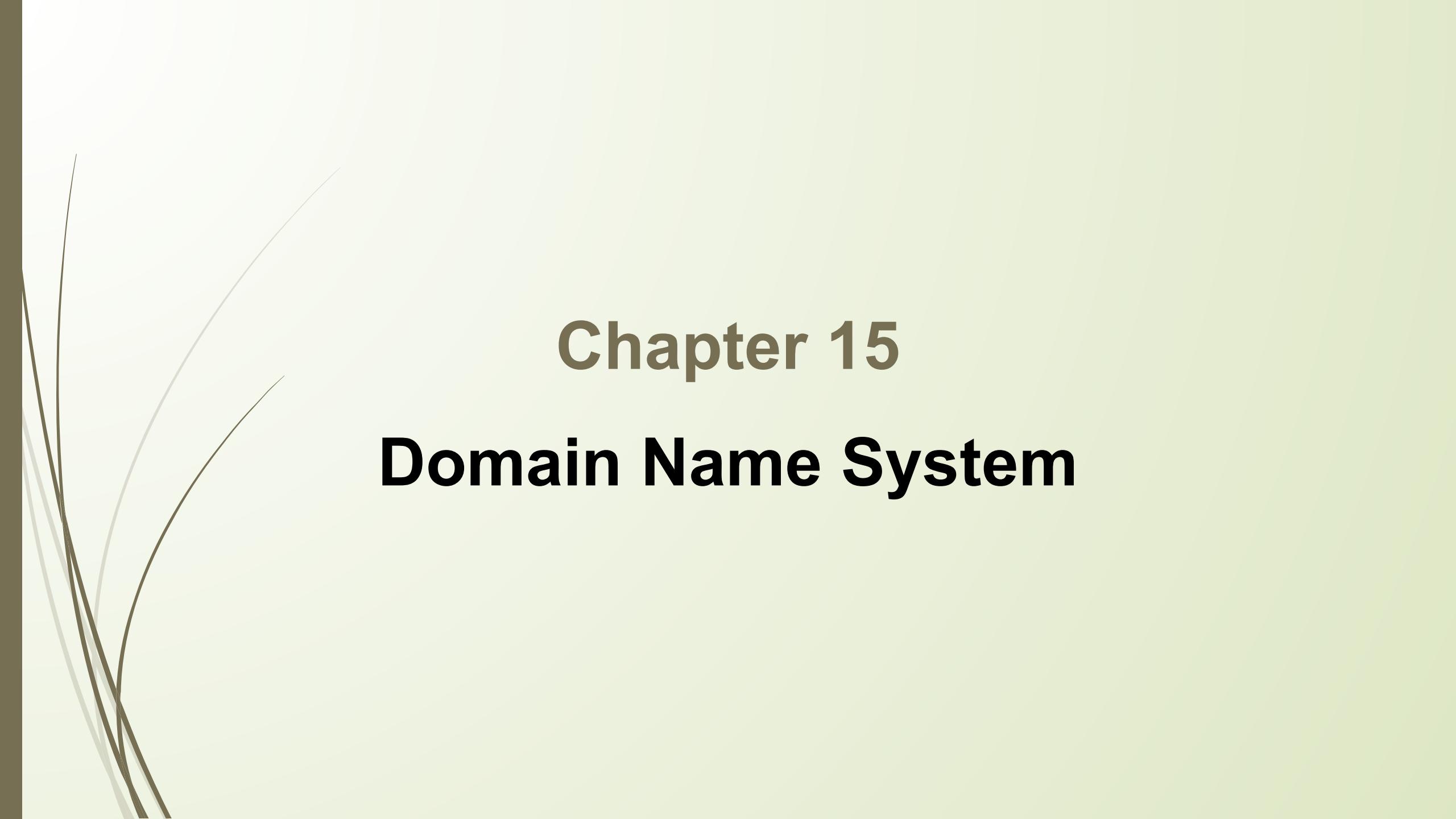
MIME-version:1.0

Content-Type: text/html



Note

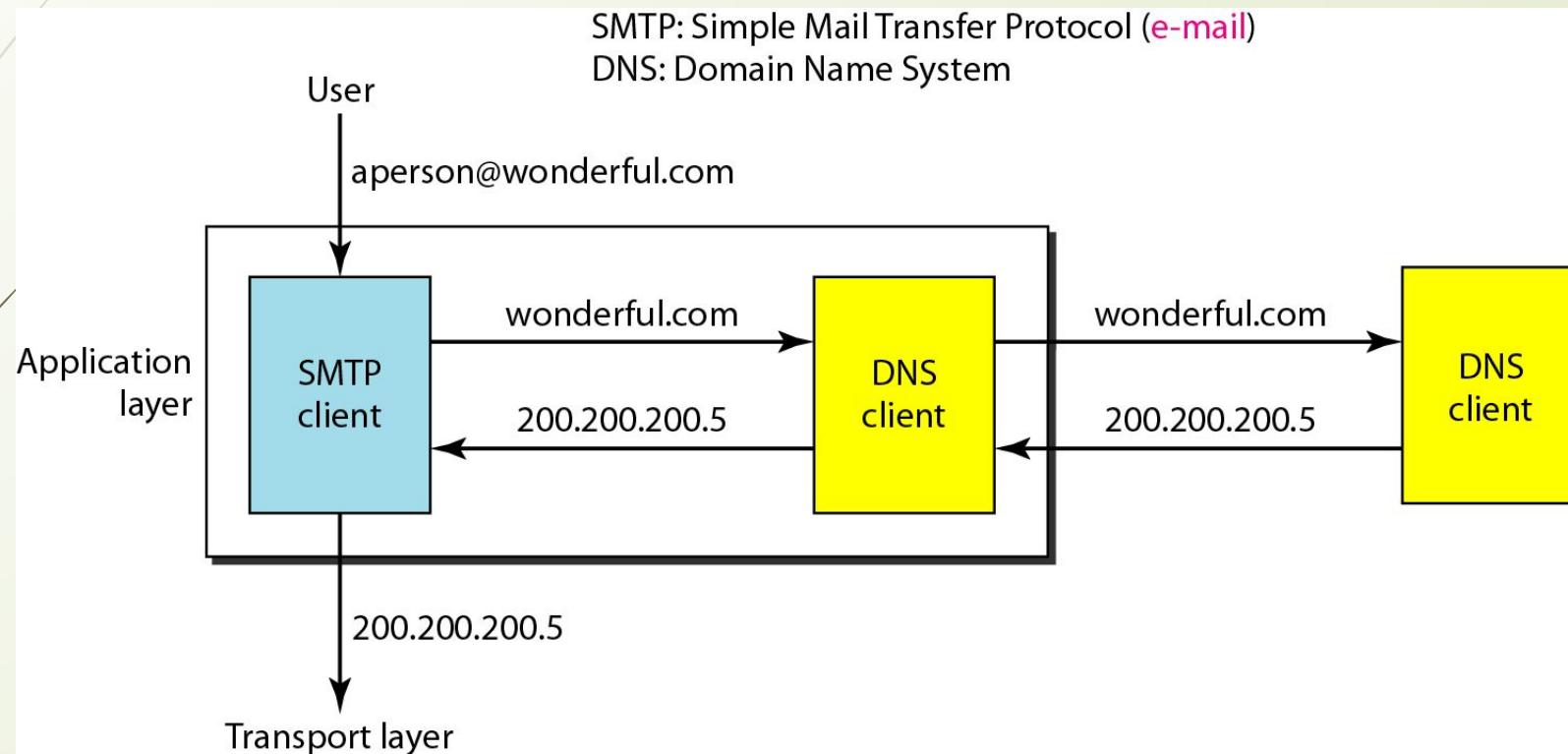
HTTP version 1.1 specifies a persistent connection by default.



Chapter 15

Domain Name System

Figure 15.1 Example of using the DNS service



15-1 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.

Topics discussed in this section:

Flat Name Space

Hierarchical Name Space

15-2 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 15.2 *Domain name space*

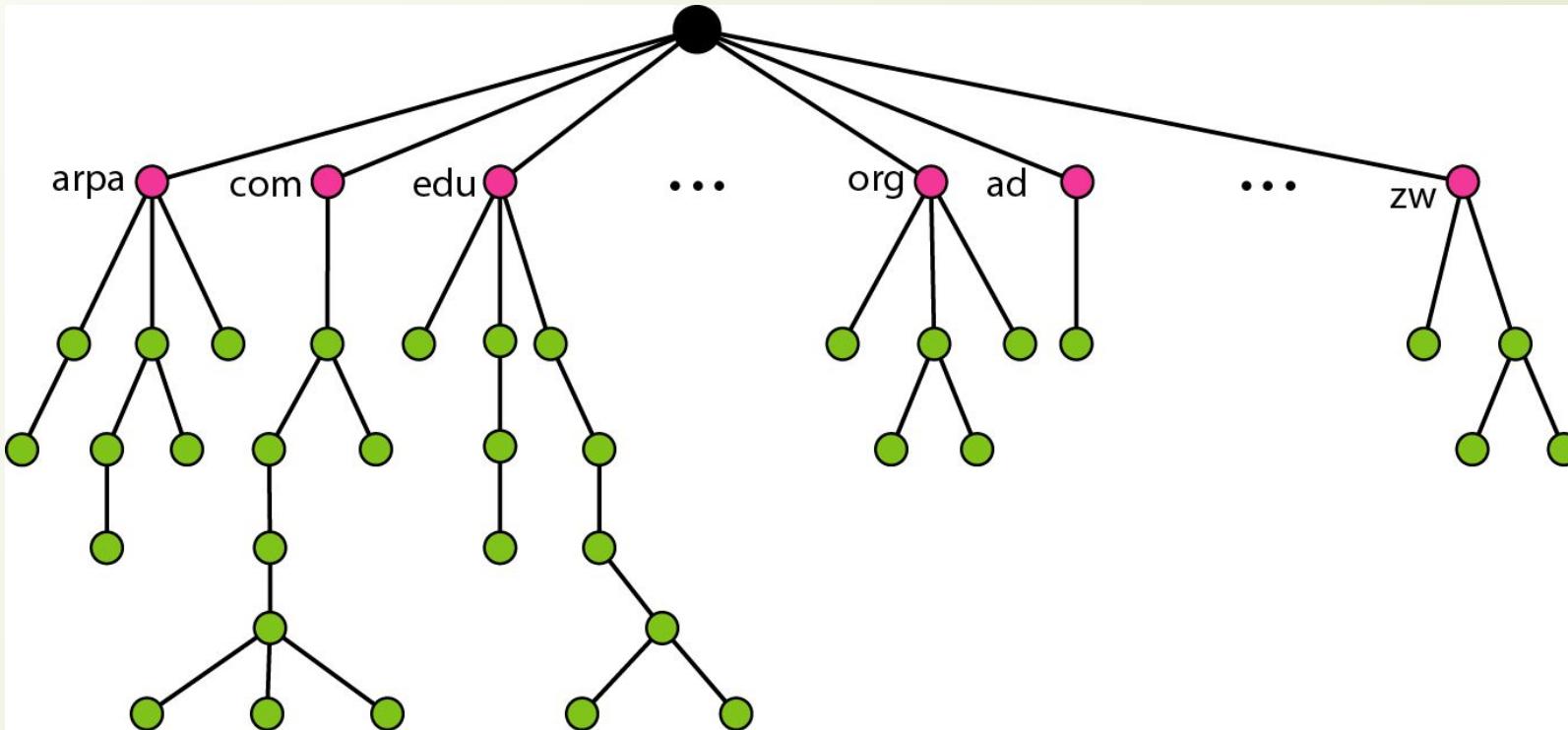


Figure 15.3 Domain names and labels

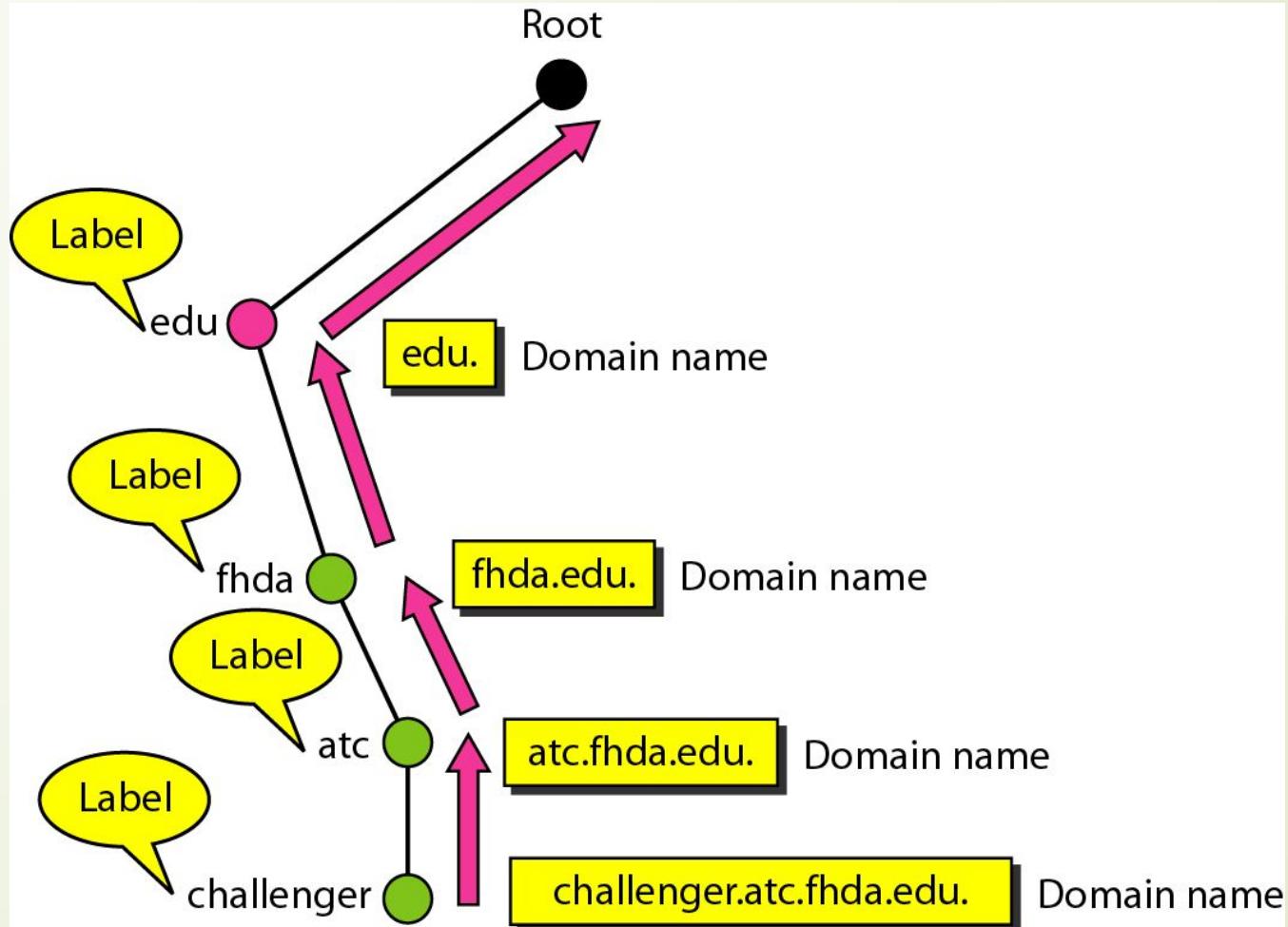


Figure 15.4 FQDN and PQDN

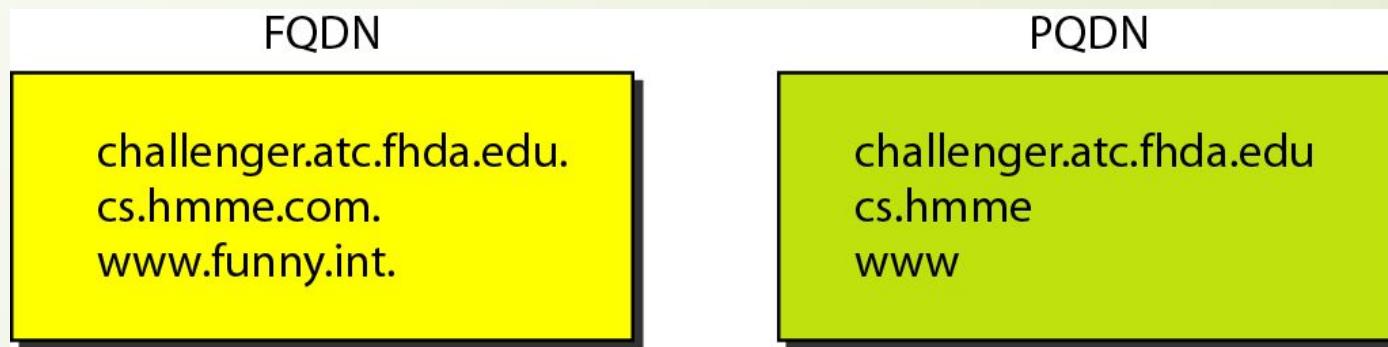
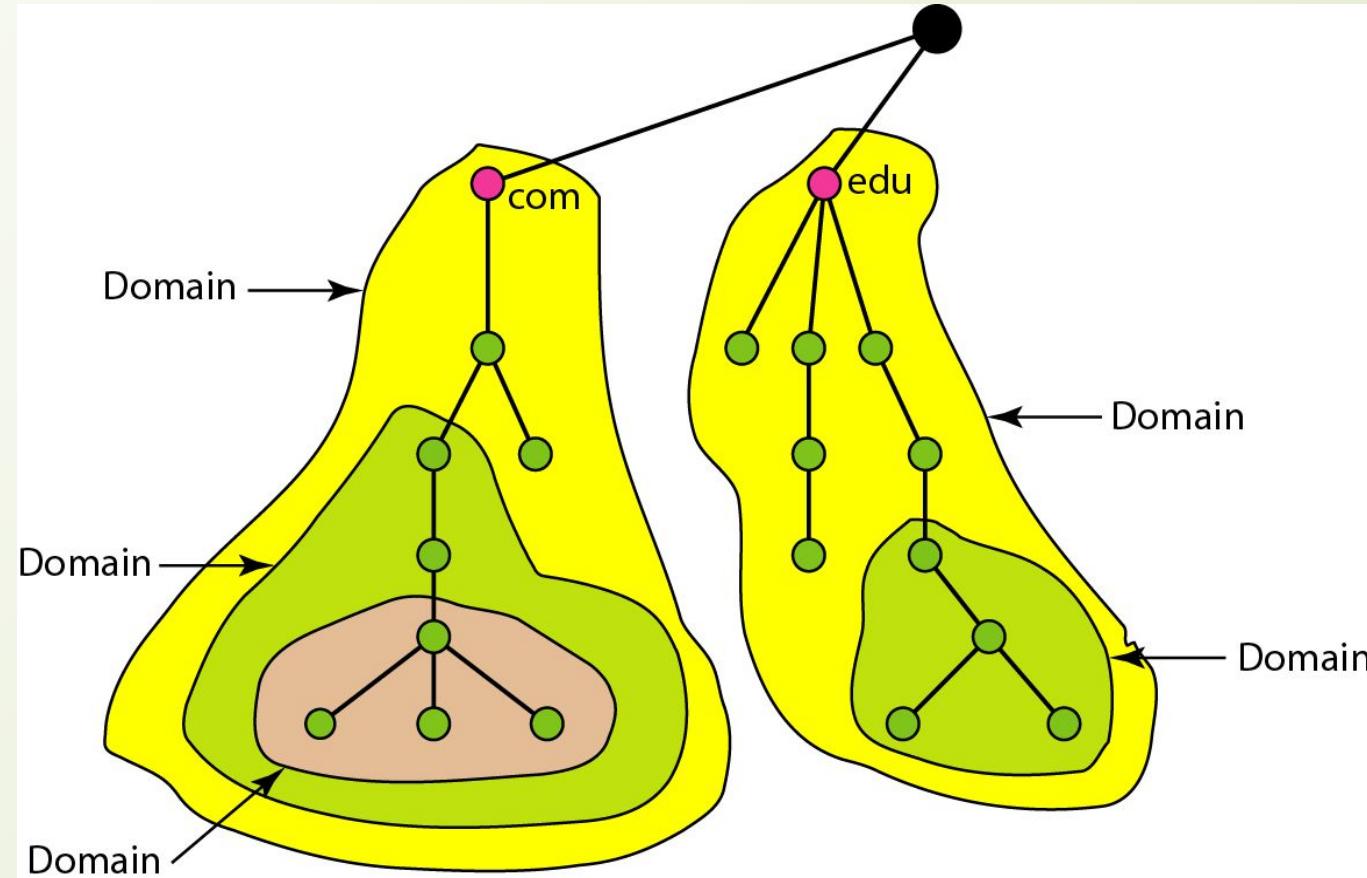


Figure 15.5 Domains



15-3 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 15.6 *Hierarchy of name servers*

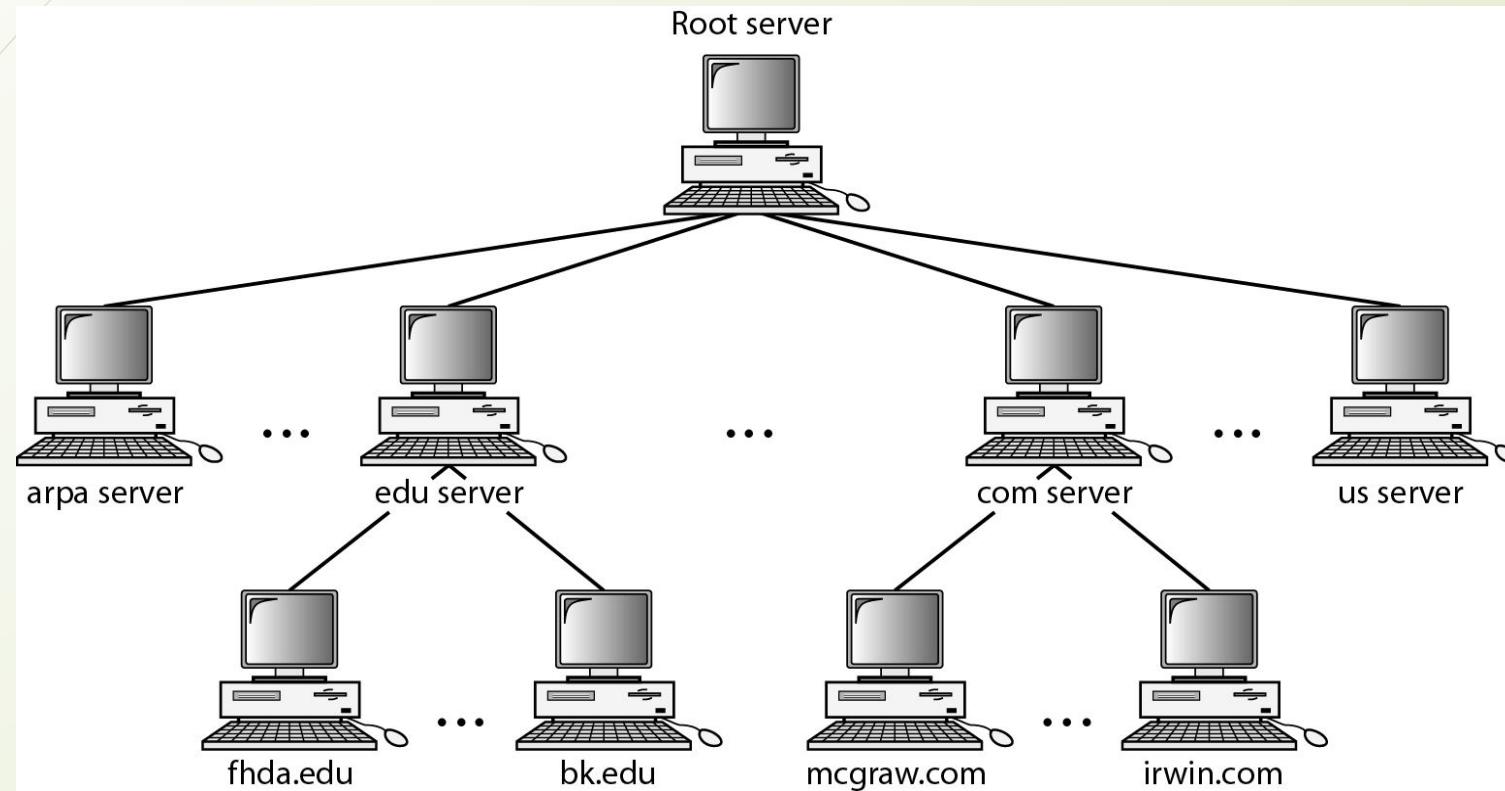
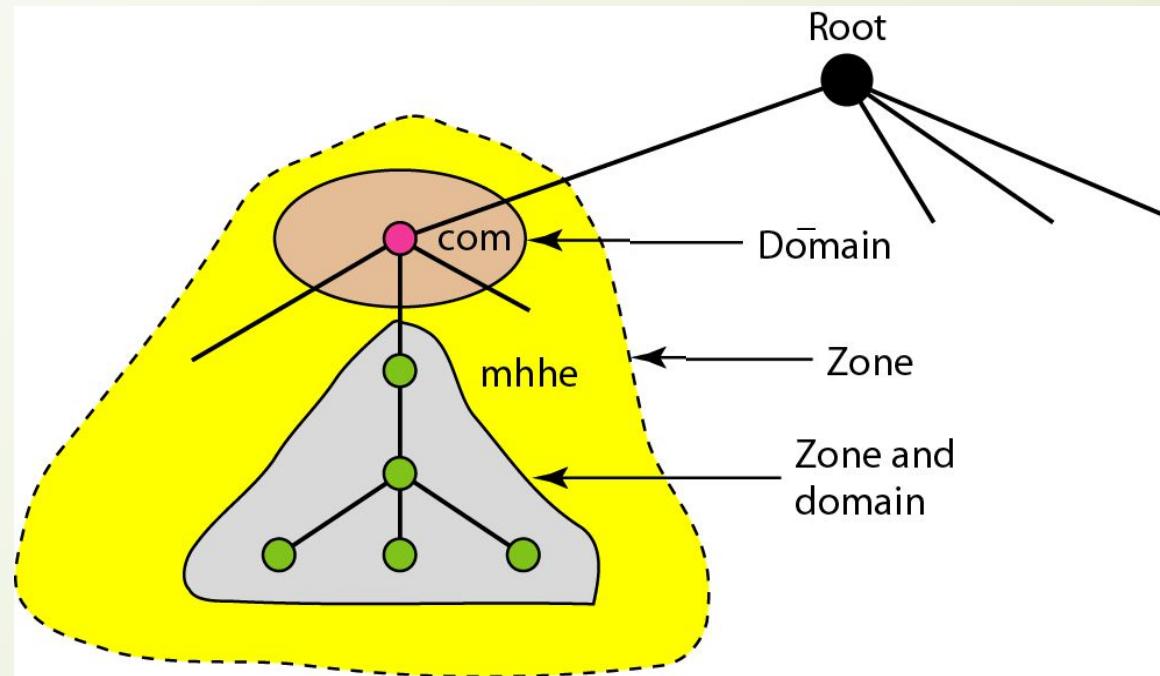
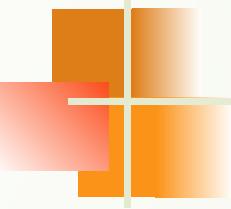


Figure 15.7 Zones and domains





Note

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.

15-4 DNS IN THE INTERNET

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

Topics discussed in this section:

Generic Domains

Country Domains

Inverse Domain

Figure 15.8 DNS IN THE INTERNET

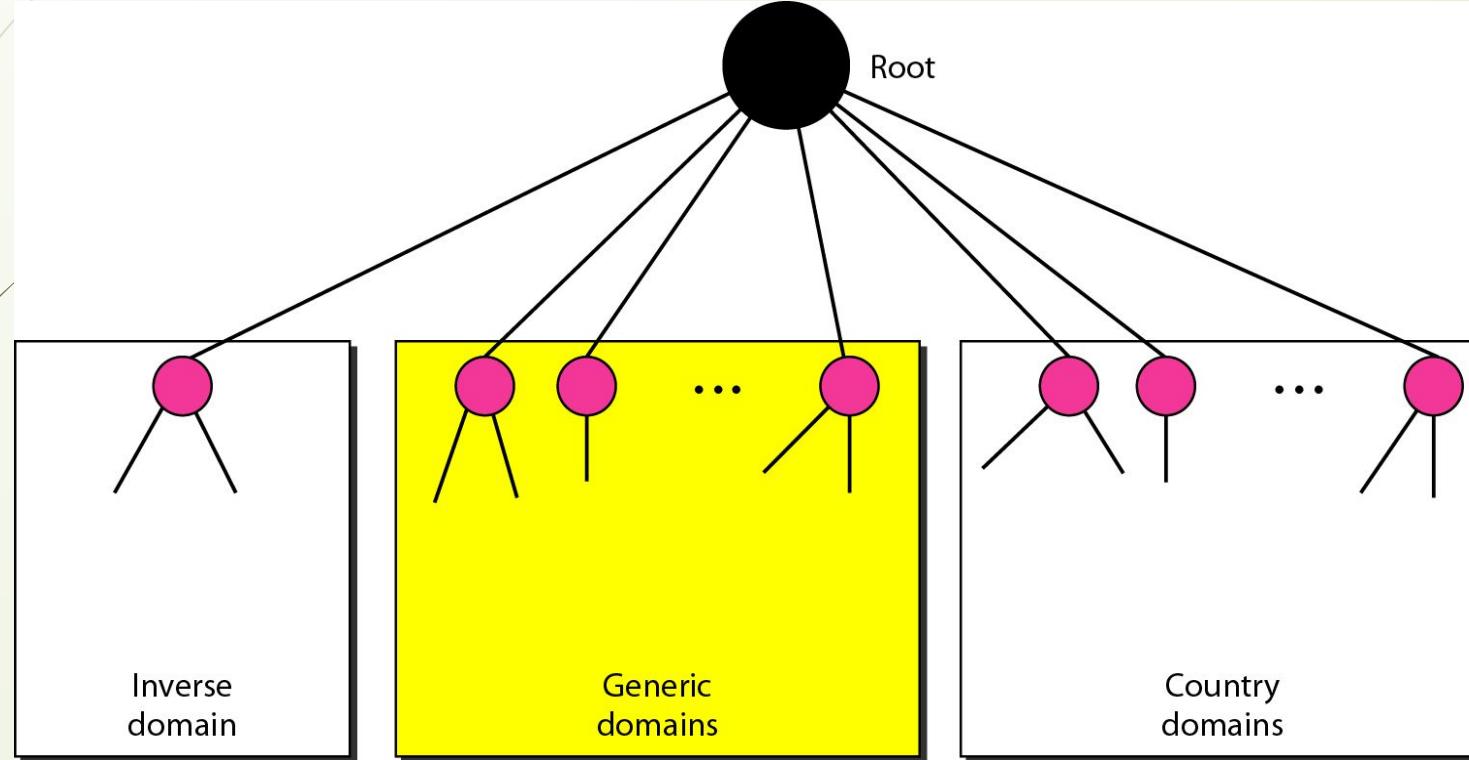


Figure 15.9 Generic domains

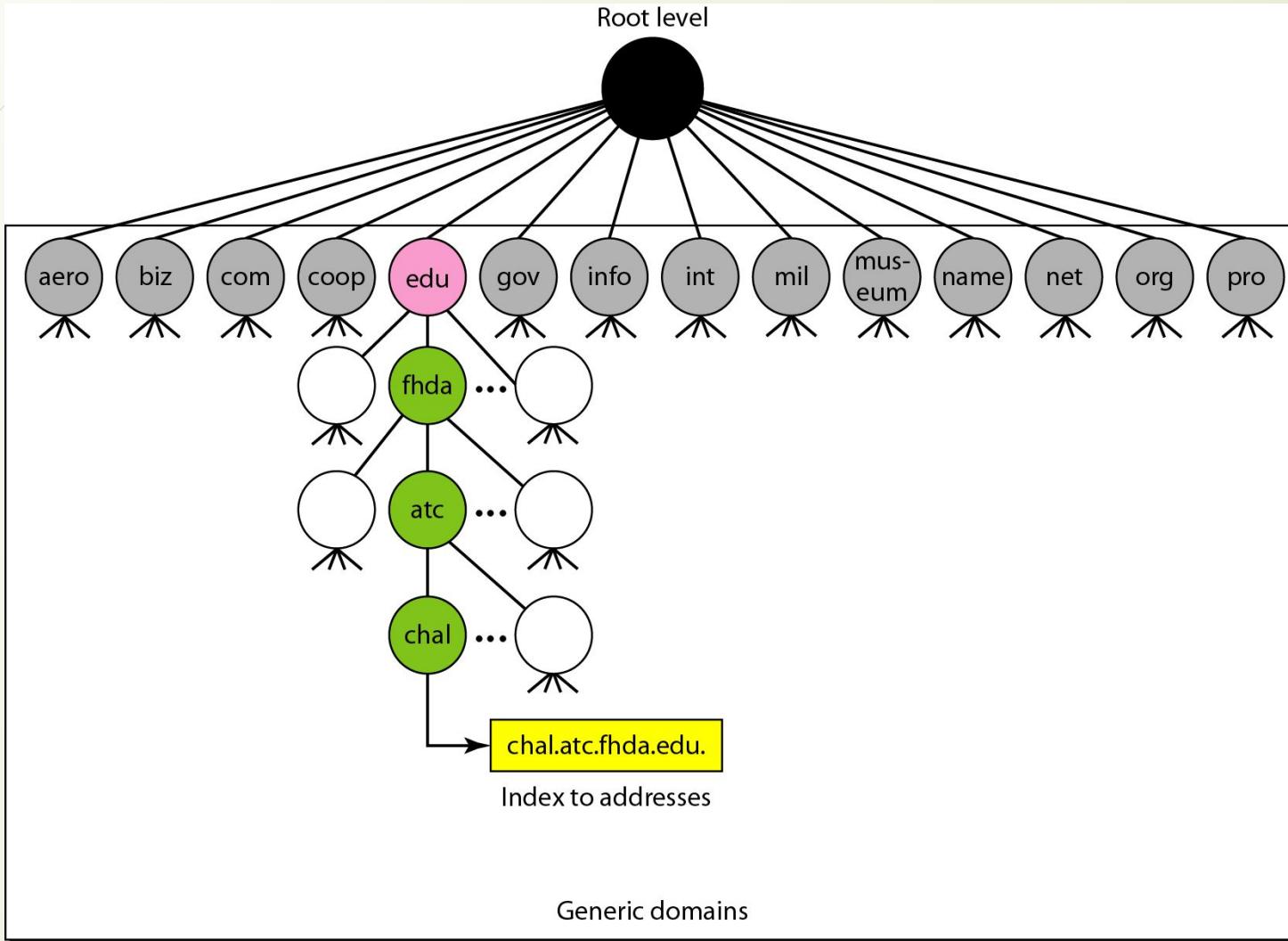




Table 15.1 *Generic domain labels*

<i>Label</i>	<i>Description</i>
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 15.10 *Country domains*

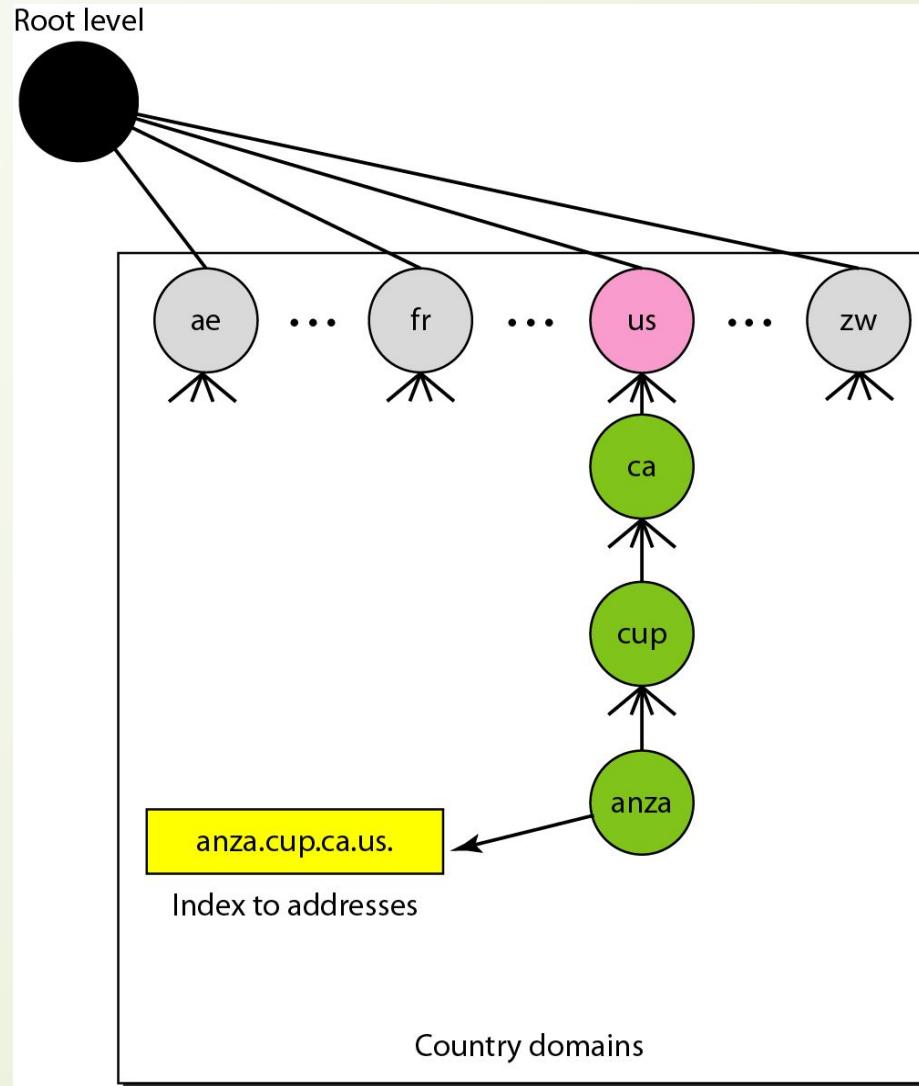
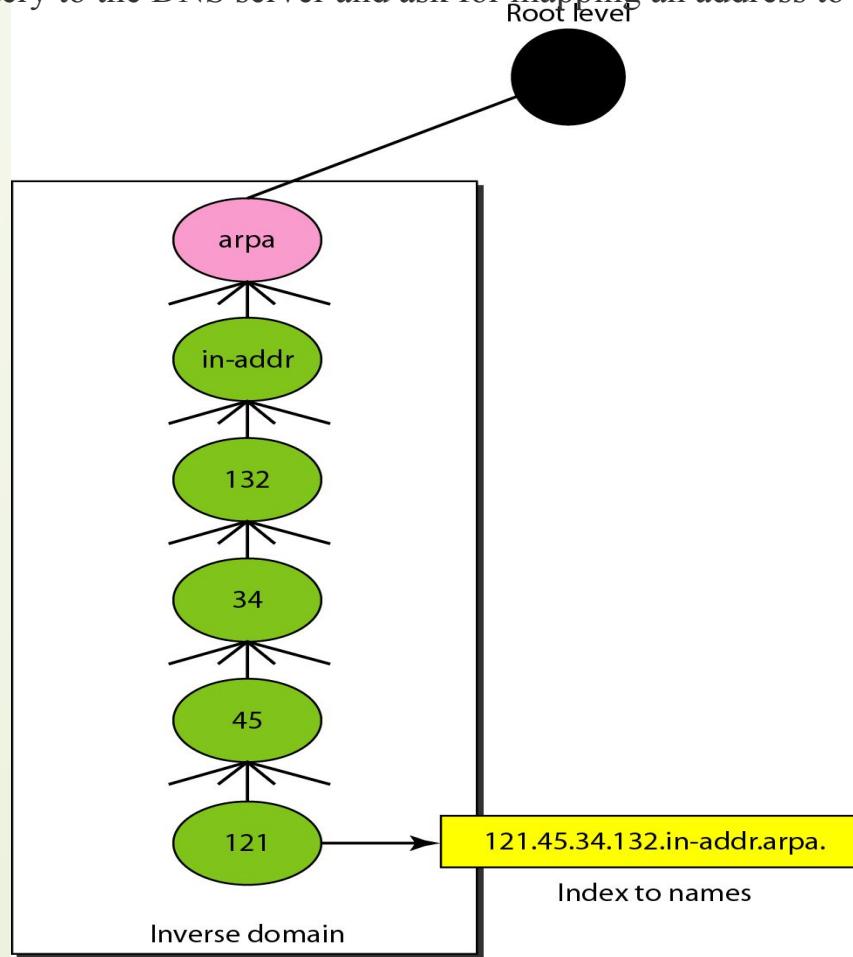


Figure 15.11 *Inverse domain*

The inverse domain is used for mapping an address to a name. When the server has received a request from the client, and the server contains the files of only authorized clients. To determine whether the client is on the authorized list or not, it sends a query to the DNS server and ask for mapping an address to the name



15-5 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

Caching

Figure 15.12 *Recursive resolution*

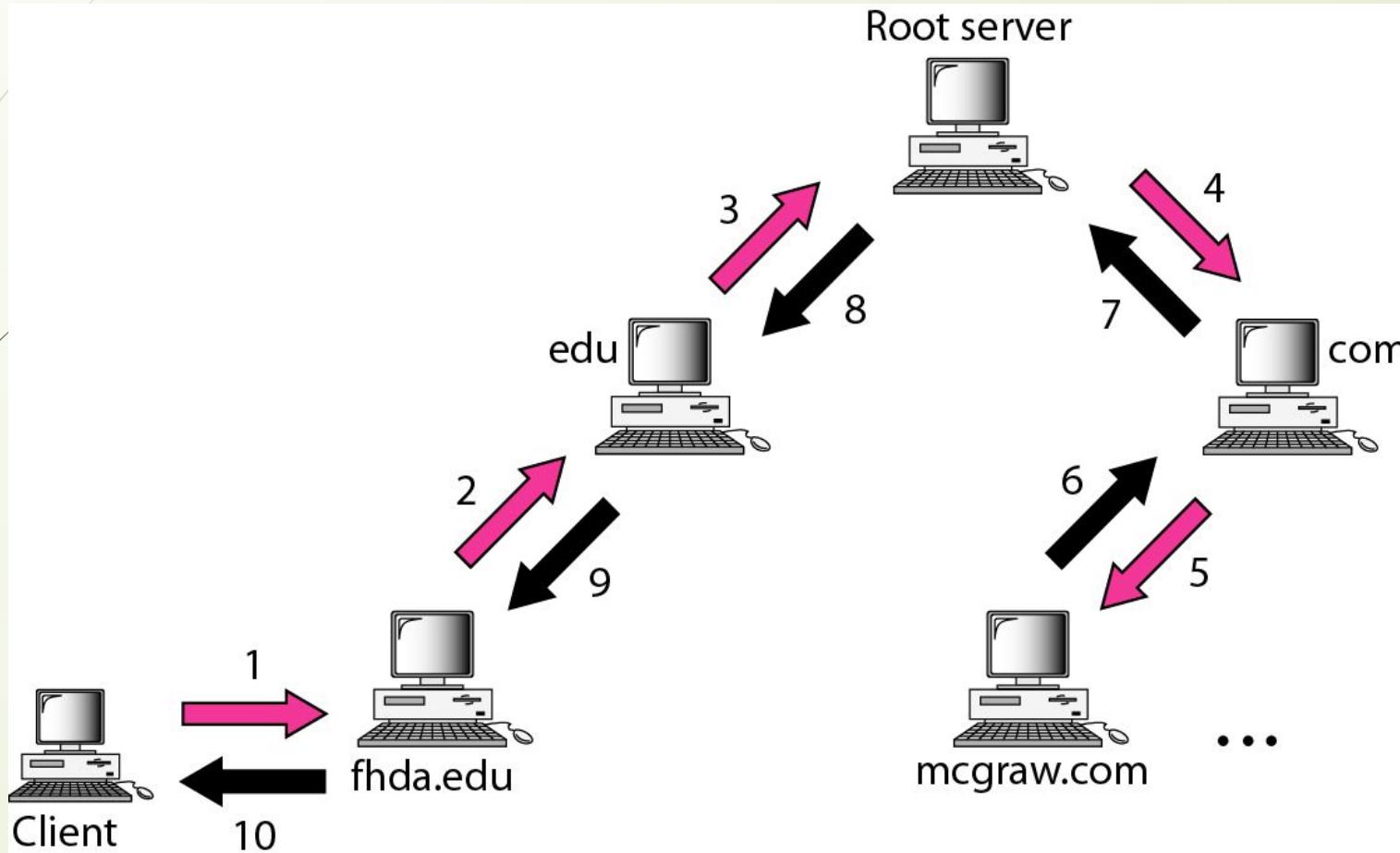
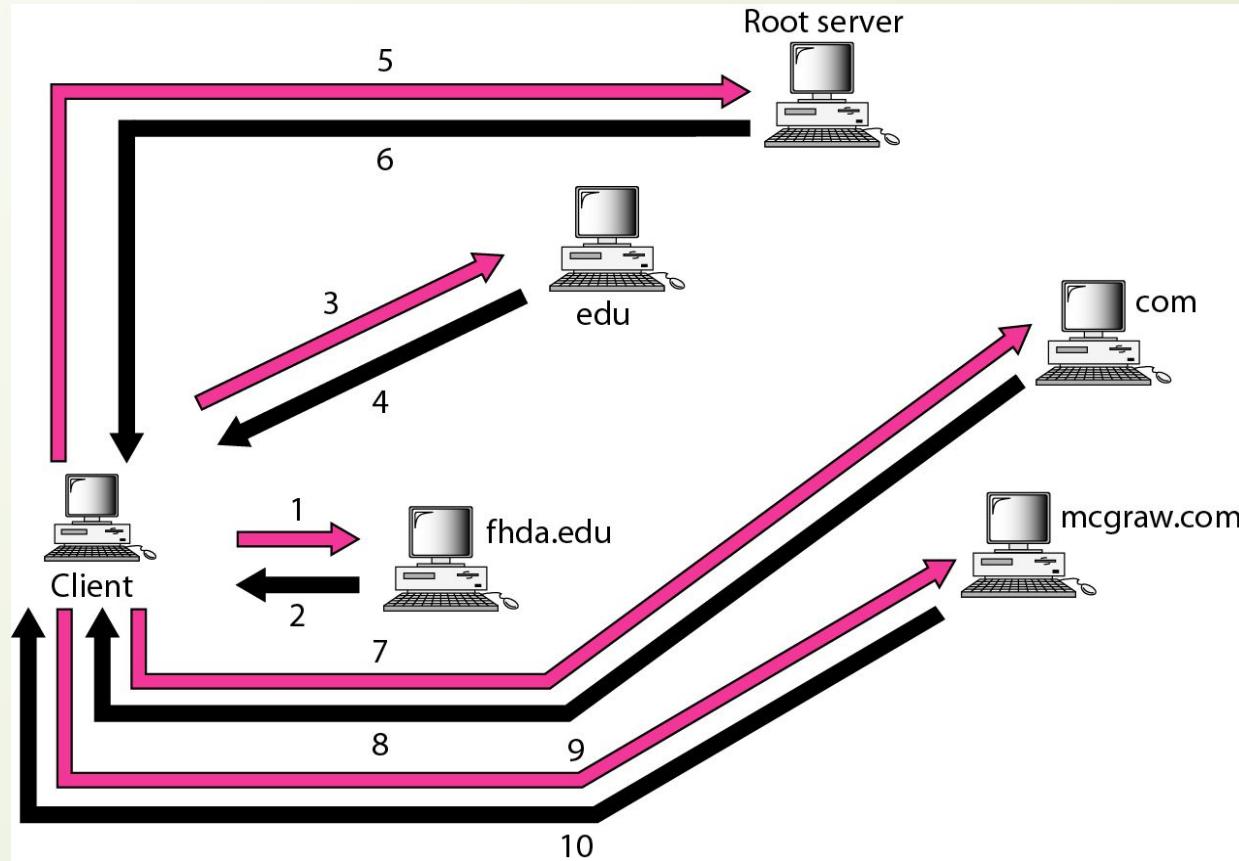


Figure 15.13 Iterative resolution



15-6 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 15.14 *Query and response messages*

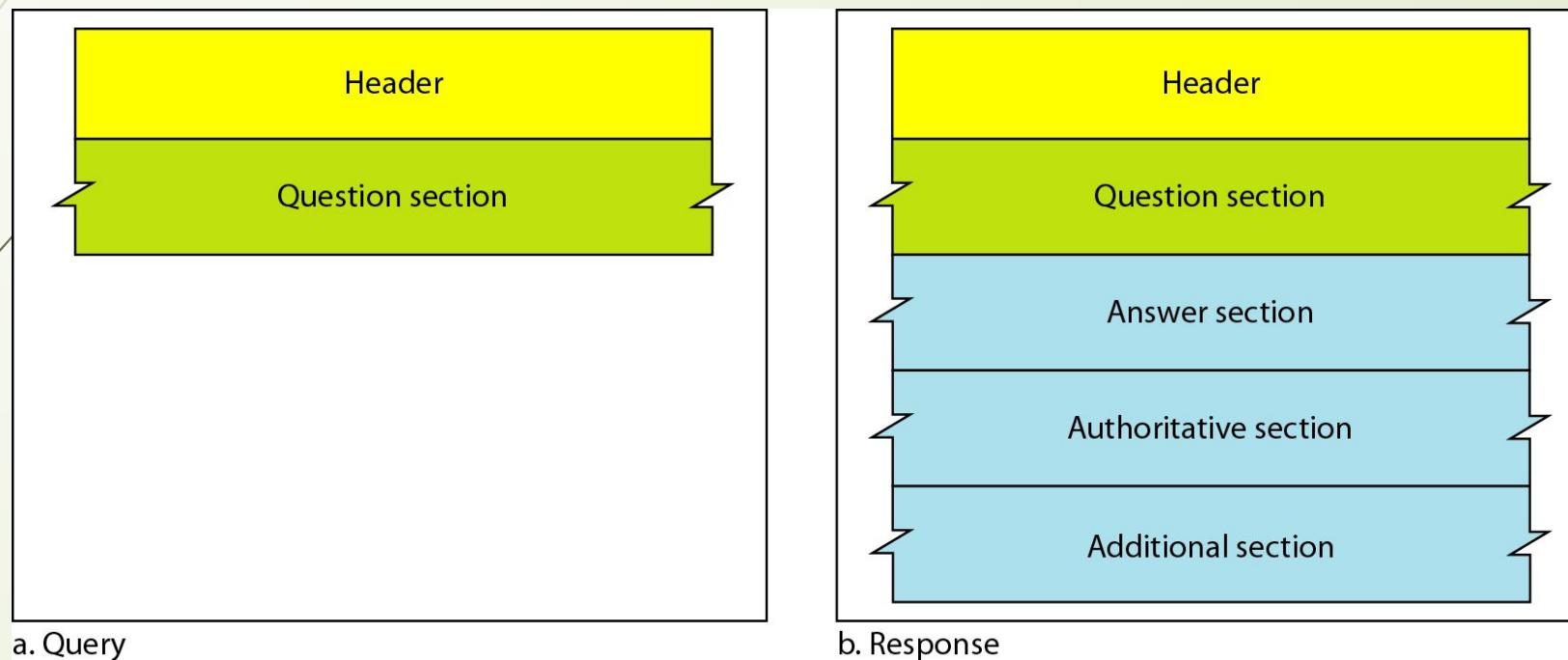




Figure 15.15 *Header format*

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

15-7 TYPES OF RECORDS

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

15-8 REGISTRARS

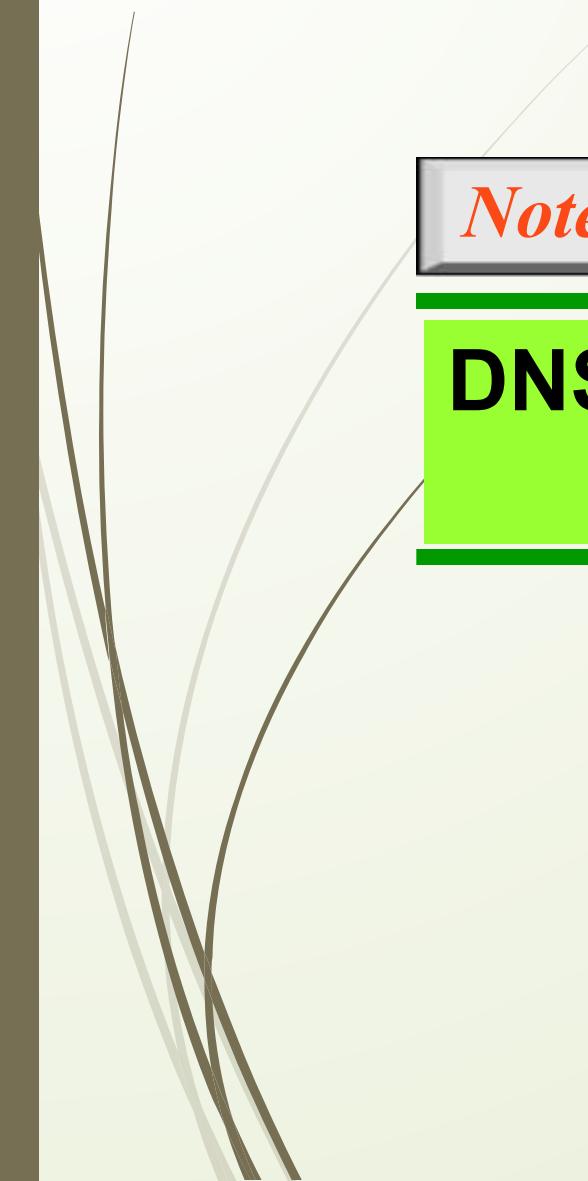
How are new domains added to DNS? This is done through a registrar, a commercial entity accredited by Internet Corporation for Assigned Names and Numbers (ICANN). A registrar first verifies that the requested domain name is unique and then enters it into the DNS database. A fee is charged.

15-9 DYNAMIC DOMAIN NAME SYSTEM (DDNS)

The DNS master file must be updated dynamically. The Dynamic Domain Name System (DDNS) therefore was devised to respond to this need. In DDNS, when a binding between a name and an address is determined, the information is sent, usually by DHCP to a primary DNS server. The primary server updates the zone. The secondary servers are notified either actively or passively.

15-10 ENCAPSULATION

DNS can use either UDP or TCP. In both cases the well-known port used by the server is port 53. UDP is used when the size of the response message is less than 512 bytes because most UDP packages have a 512-byte packet size limit. If the size of the response message is more than 512 bytes, a TCP connection is used.



Note

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