

CSE1004 NETWORK AND COMMUNICATION

**Networking Principles and layered
architecture**

Networking Principles and layered architecture

- Data Communications and Networking:
 - A Communications Model
- Data Communications:
 - Evolution of network
 - Requirements
 - Applications
 - Network Topology (Line configuration, Data Flow),
- Protocols and Standards, Network Models (OSI, TCP/IP)

DEFINTION & APPLICATIONS

DEFINTION:

- A computer network is defined as a group of two or more computer systems linked together. It is done to enable the computers to communicate and share available resources.
- Many types of computer networks, including the following: LAN, MAN, WAN, CAN, PAN, HAN....
- Network benefits: Sharing and Connectivity

APPLICATIONS:

- **Sharing of resources such as printers.**
- **Sharing of expensive software's and database.**
- **Communication from one computer to another computer**
- **Exchange of data and information among users via network.**
- **Sharing of information over geographically wide areas.**

Network Benefits: SHARING RESOURCES

- Types of resources are:
 1. **Hardware:** A network allows users to share many hardware devices such as printers, modems, fax machines, CD ROM, players, etc.
 2. **Software:** sharing software resources reduces the cost of software installation, saves space on hard disk.

OTHER BENEFITS OF COMPUTER NETWORK

- Increased speed
- Reduced cost
- Improved security
- Centralized software managements
- Electronic mail
- Flexible access

DISADVATAGES OF NETWORKS

- o High cost of installation
- o Requires time for administration
- o Failure of server
- o Cable faults

DATA COMMUNICATIONS

DC- is the exchange of data between two devices by means of any transmission medium.

Characteristics:

- 1.Delivery,**
- 2. Accuracy and**
- 3. Timeliness.**

1. The data must be delivered to the correct destination.
2. The data must be delivered accurately. i.e. without alteration.
3. The system must deliver data in a timely manner.
e.g. Real time application.

Network

- **Network:** Is a group or system of interconnected people or things.
- Example: "the company has a network of 20 branches".
- Computer Networks: A network is defined as a group of two or more computer systems linked together. There are many types of computer networks: LAN, MAN, WAN, CAN, PAN, HAN.....

Network Characteristics

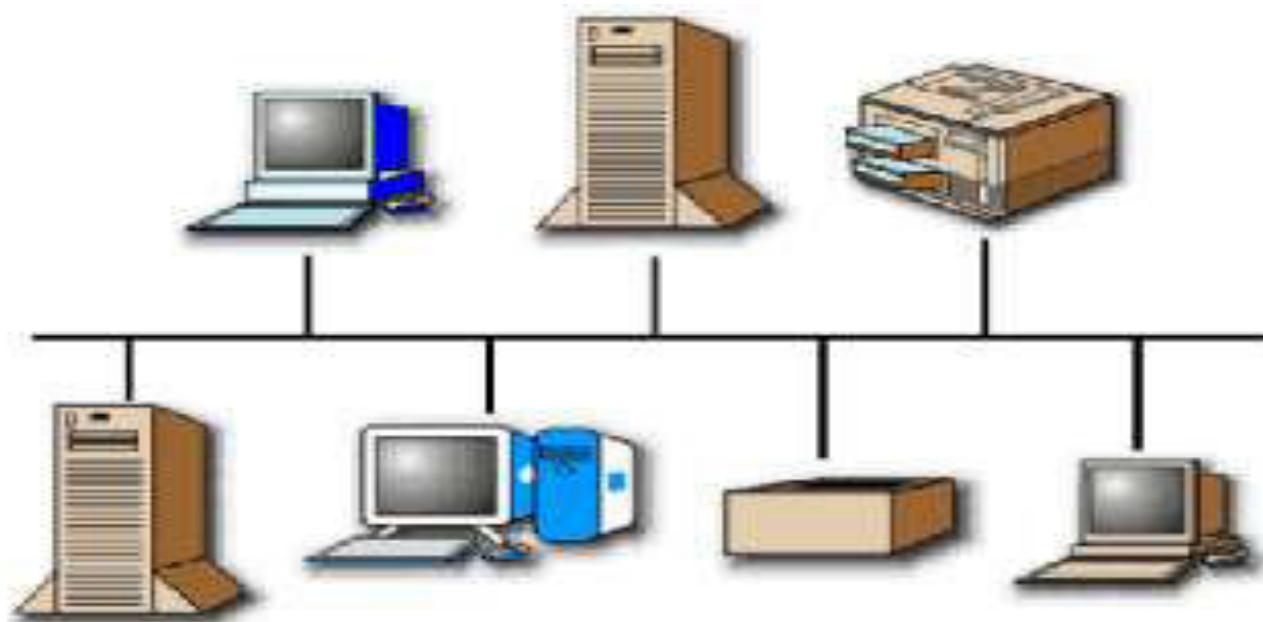
- **Topology** : The geometric arrangement of a computer system.
- **Protocol** : The protocol defines a common set of rules and signals that computers on the network use to communicate. One of the most popular protocols for LANs is called Ethernet.
- **Architecture** : Networks can be broadly classified as using either a peer to peer or client/server architecture.

Network Goals:

- Resource sharing.
- High reliability.
- Saving Money and Time consuming

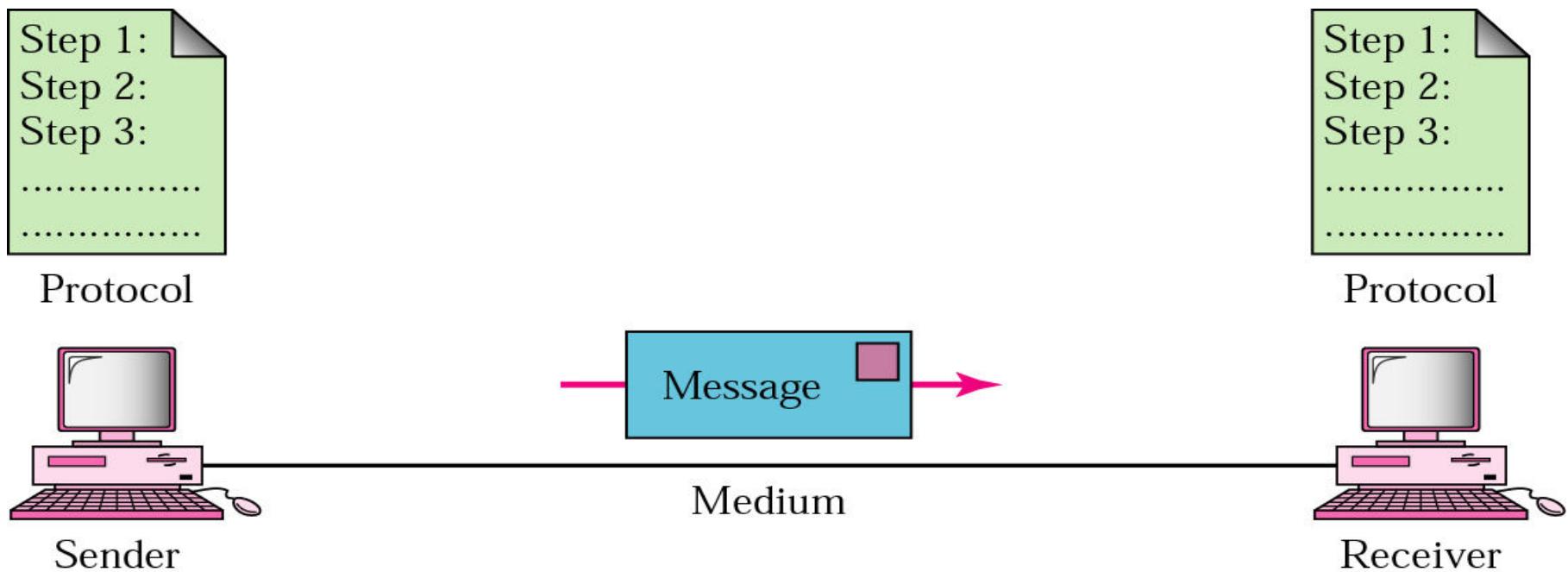
What is a Network?

- A network is a set of devices (node) connected by media links.
- A computer network may be defined as an interconnected collection of autonomous computers.
- A network is a collection of computers, printers, routers, switches, and other devices that are able to communicate with each other over some transmission media.



Components of data Communication:

1.Message, 2.Sender, 3.Receiver, 4.Medium and 5.Protocol.



Components of data Communication

Message : It is the data to be communicated. It consists of text, numbers, pictures, sound, or video or any combination of these.

Sender : It is the device that sends the data message. It can be a computer, workstation, telephone handset, video camera.

Receiver : It is the device that receiver the message. It can be a computer, workstation, telephone, and television.

Medium : Transmission medium is the physical path by which a message travels from sender to receiver. Example it consists of twisted pair wire, co axial cable, fiber optical, laser or radio waves.

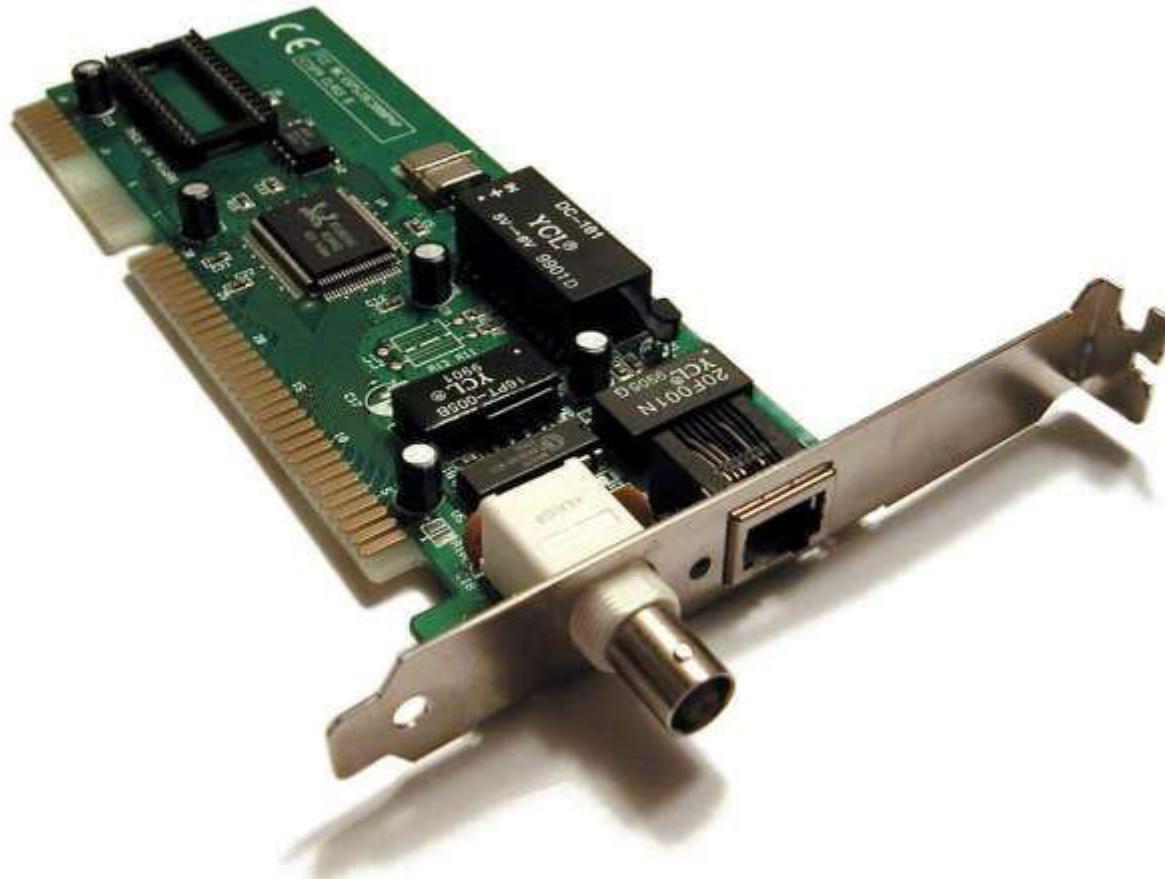
Protocol: It is a set of rules that govern data communication.

Without a protocol two devices are connected but not

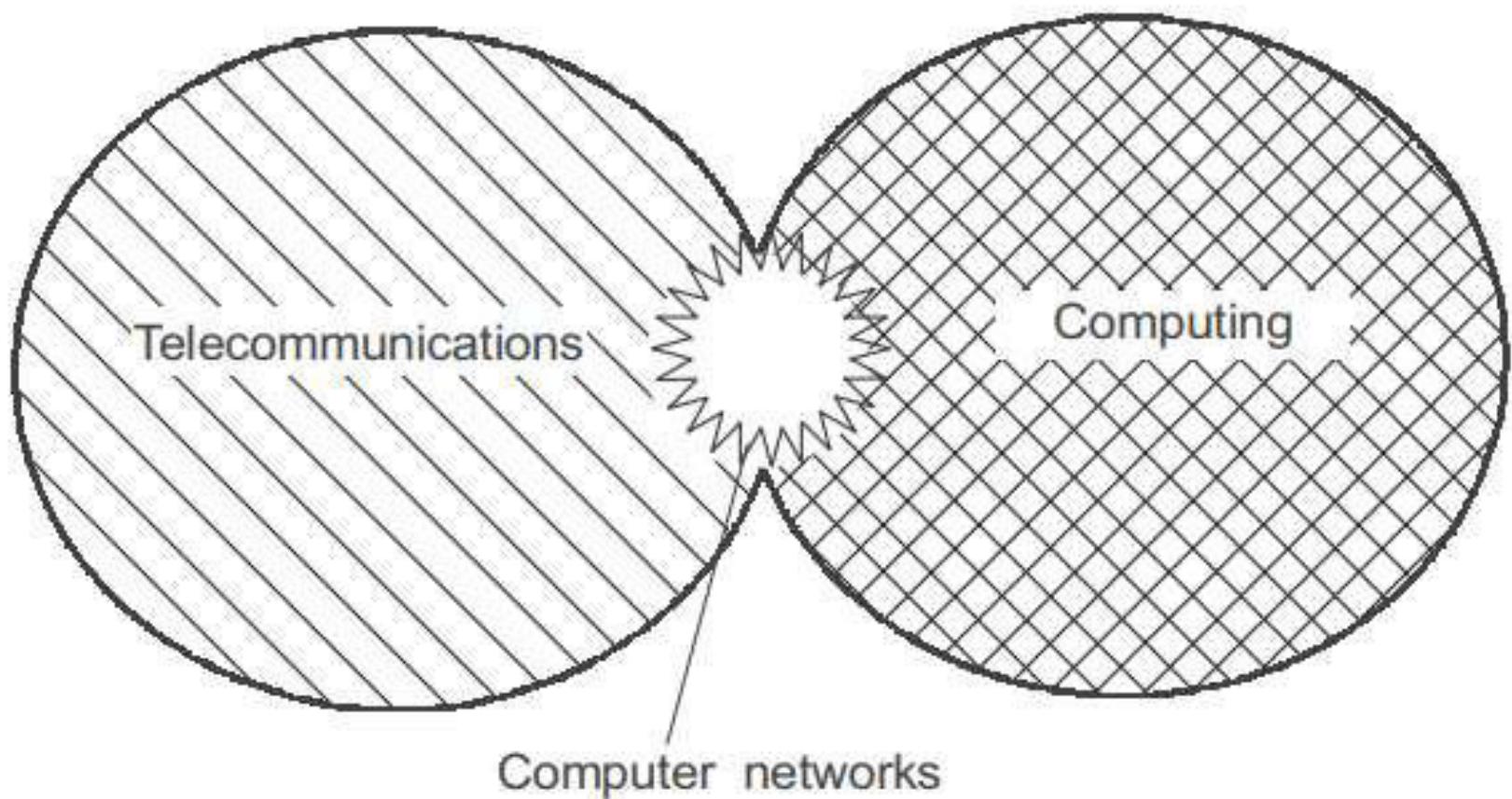
COMPONENTS OF COMPUTER NETWORK

- Two or more computers
- Cables as links between the computers
- A network interfacing card(NIC) on each computer
- Switches
- Software called operating system(OS)

Network interfacing card(NIC)



1. Evolution of network



Evol....Roots of Computer networks

- Computer networks: transmitting information over along distances. This implementation is done by various methods of data encoding and multiplexing in telecommunications systems.
- Batch processing systems: 1950
- Multiterminal systems: Prototype of the computer network(1960).

Batch processing systems

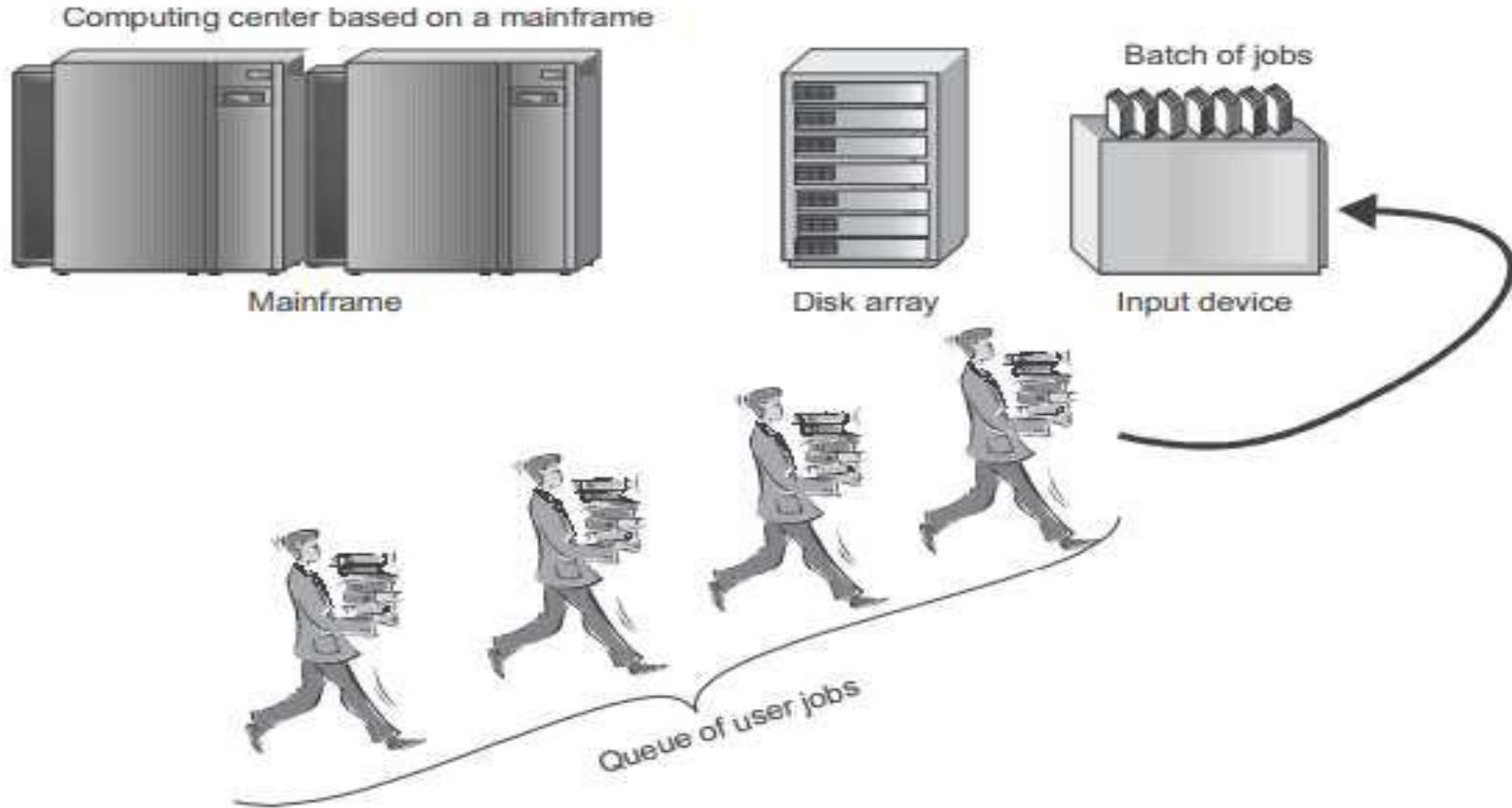


Figure 1.2 Centralized system based on a mainframe

Multiterminal systems

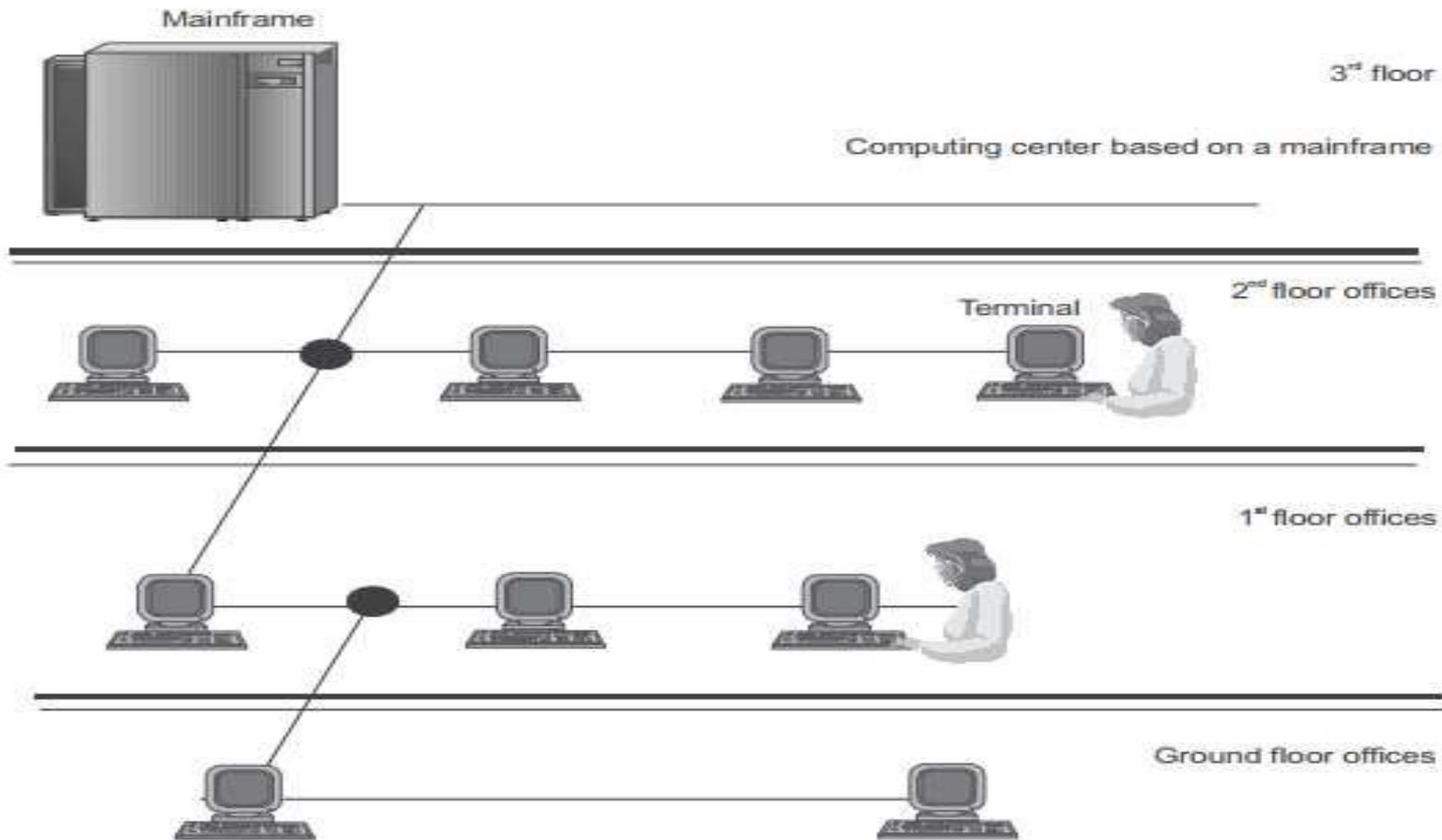


Figure 1.3 Multiterminal system as a prototype of a computer network

Evol...First Computer Networks

- First WAN:
 - Multilayer architecture of communications protocols
 - Packet switching technology
 - Packet routing heterogeneous networks

2. Data Communications: Requirements

The screenshot shows a web browser window with the title "6 Data Communication Requirements". The browser has multiple tabs open, including "Sent Mail", "WhatsApp", "Prime Video: MMOV", "Mail", and "6 Data Communication Req.". The main content area displays the title "6 Data Communication Requirements" and a sub-section "6.1 Interaction". Below this is a sequence diagram titled "Figure 6-1. Interaction Diagram". The diagram shows two participants: "Web Client System" and "Web Enabled DICOM Server". A sequence of messages is shown: 1. Discovery request (GET HTTP Request) from the Web Client System to the Web Enabled DICOM Server. 2. Discovery result (HTTP Response to the GET Request) from the Web Enabled DICOM Server back to the Web Client System.

6 Data Communication Requirements

6.1 Interaction

The interaction shall be as shown in [Figure 6-1](#).

Multiple communications modes are possible

- URI based using HTTP Get: WADO-URI request
- Web Services (WS) using HTTP Post: WADO-WS, either:
 - a. DICOM Requester (Retrieve Imaging Document Set)
 - b. Rendered Requester (Retrieve Rendered Imaging Document Set)
 - c. Metadata Requester (Retrieve Imaging Document Set Metadata)

3. Applications of Networks:

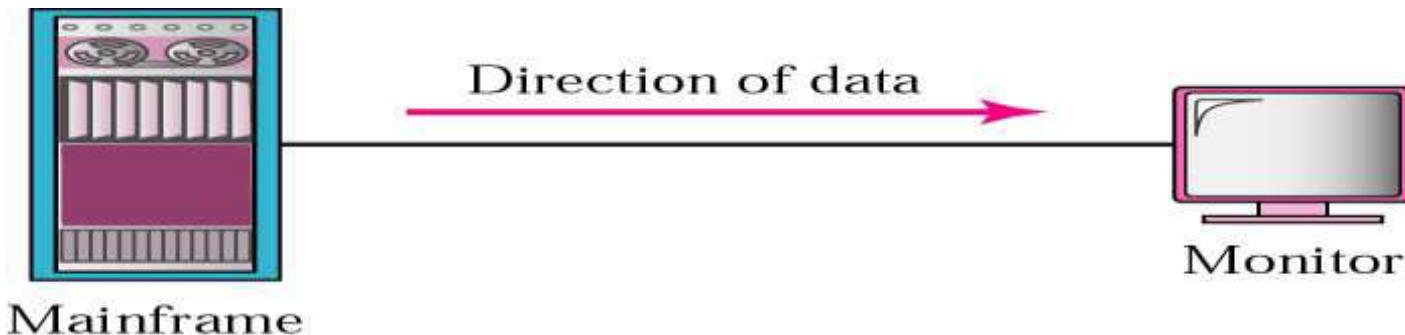
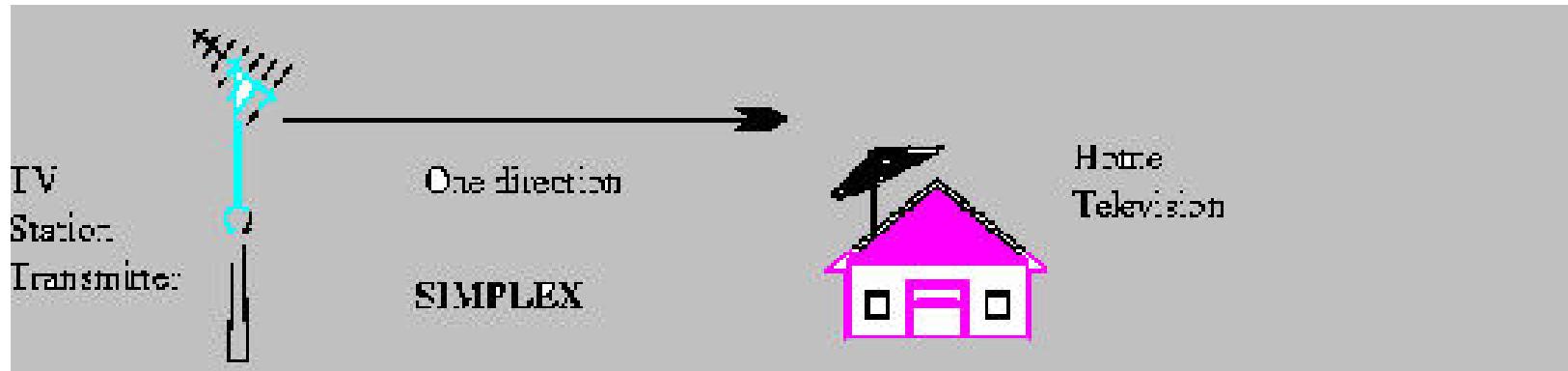
- Electronic messaging. (E-Mail)
- Electronic data Interchange. (E-Com.).
- Teleconferencing.
- Cellular Telephone.
- Cable TV.
- On-line Marketing , Sales, ticket reservations (boats, hotels, theaters)
- Financial Services. (E- Cash).
- Manufacturing.
- Information Services.

DIRECTION OF DATA FLOW

DIRECTION OF DATA FLOW:

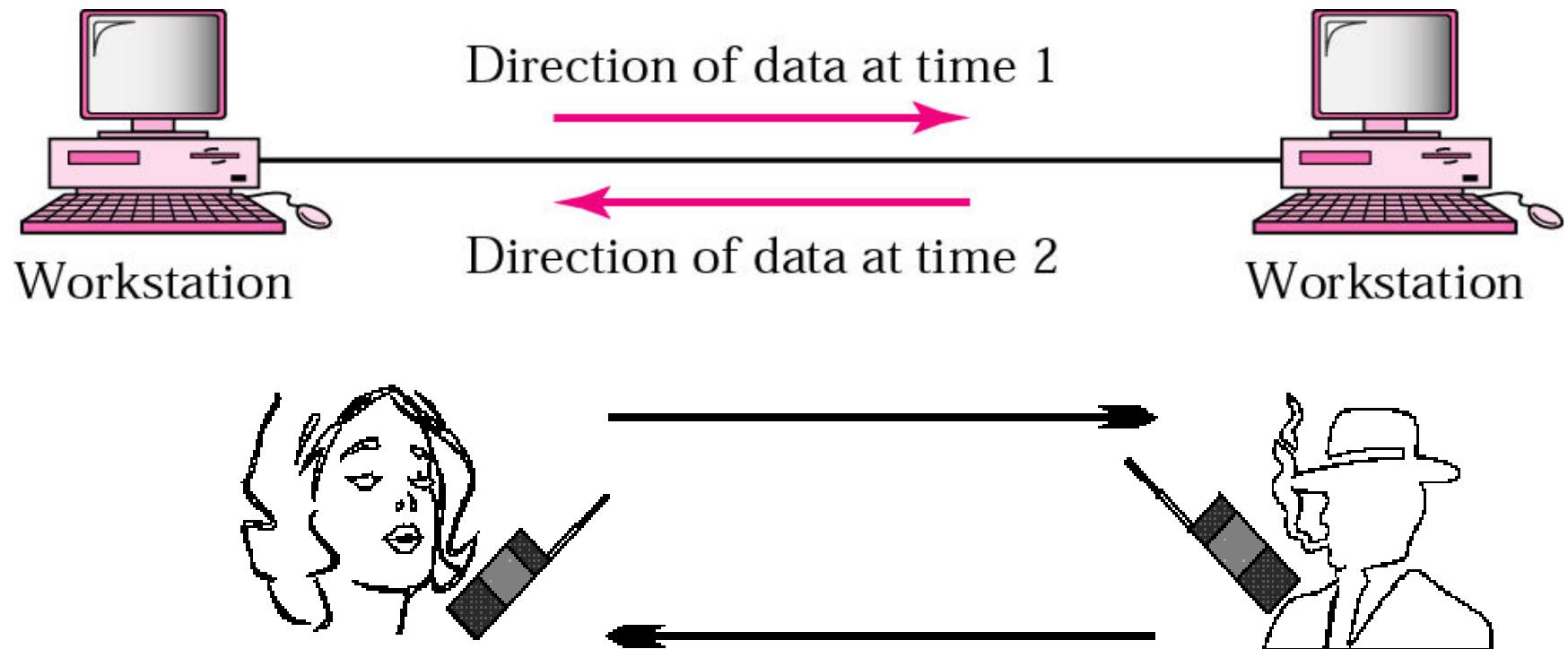
Simplex:

Data flows in only one direction on the data communication line (medium). E.g. Radio and Television broadcasts. They go from the TV station to your home television.



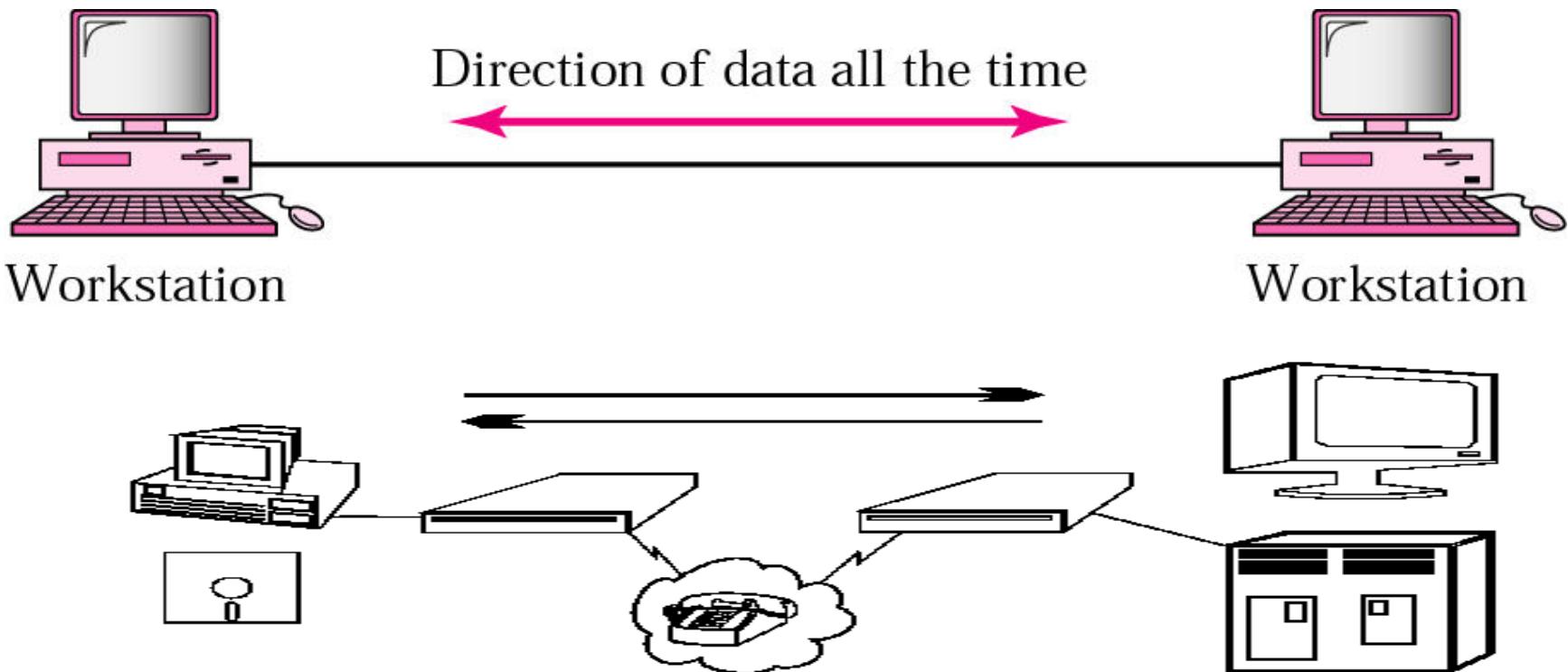
Half-Duplex:

Data flows in both directions but only one direction at a time on the data communication line. Ex. Conversation on walkie-talkies is a half-duplex data flow. Each person takes turns talking. If both talk at once - nothing occurs!



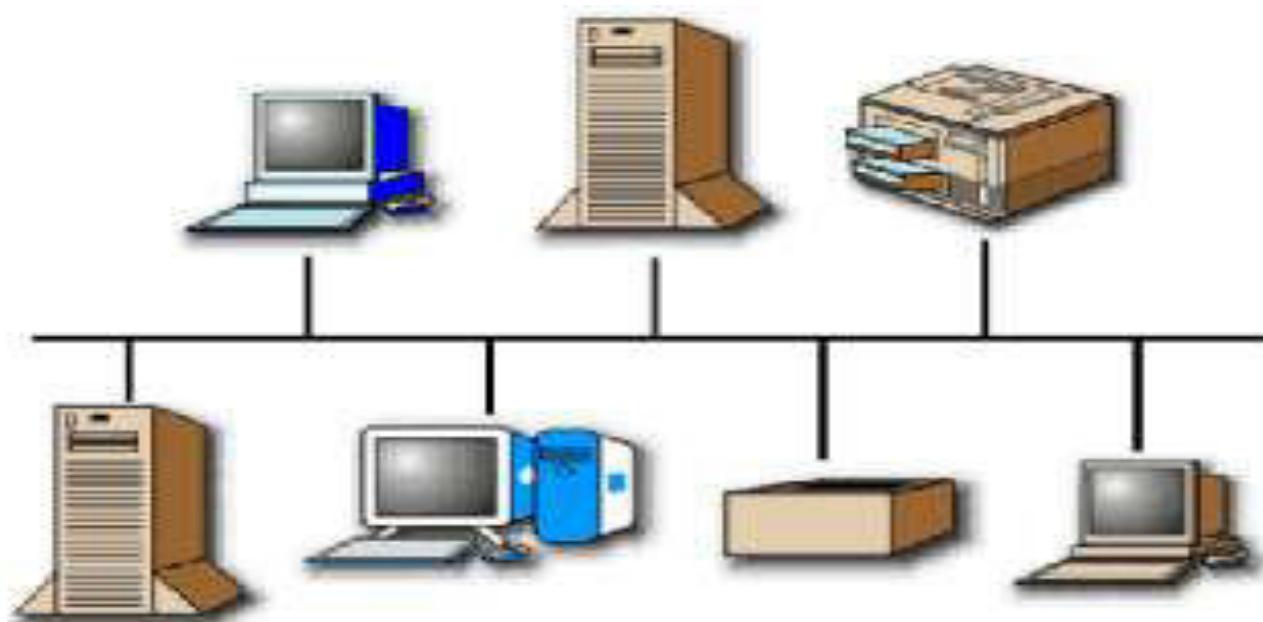
Full-Duplex:

Data flows in both directions simultaneously at the same time.
Ex. Modems are configured to flow data in both directions.



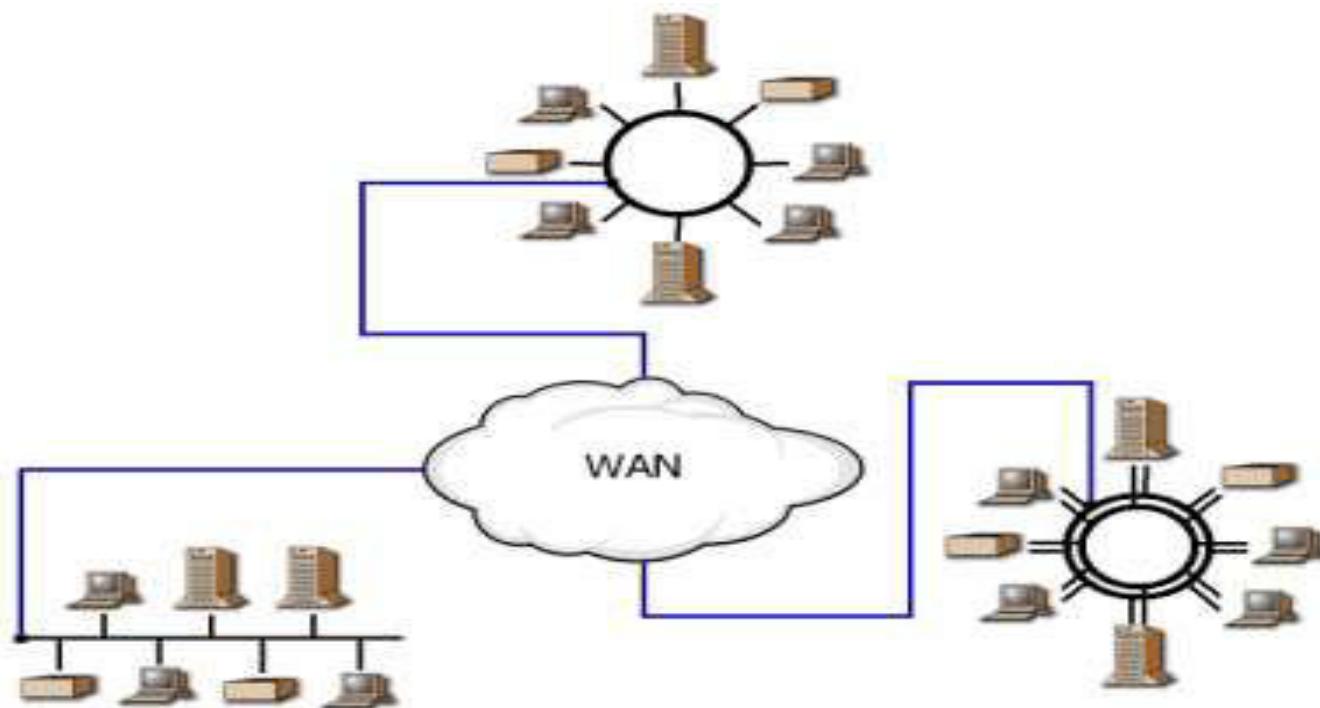
What is a Network?

- A network is a set of devices (node) connected by media links.
- A computer network may be defined as an interconnected collection of autonomous computers.
- A network is a collection of computers, printers, routers, switches, and other devices that are able to communicate with each other over some transmission media.

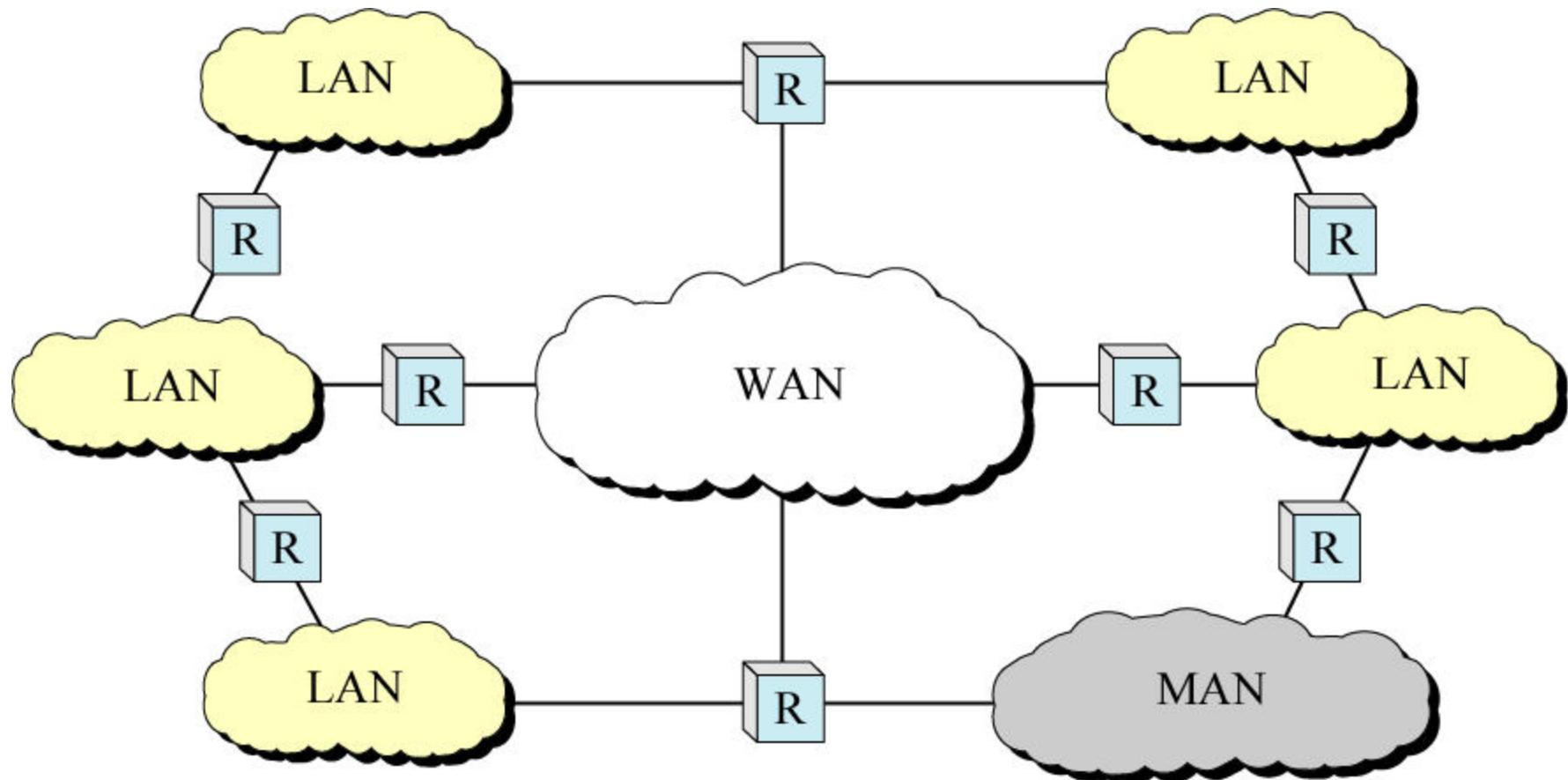


What is an Internetwork? (i.e. Networks of Networks)

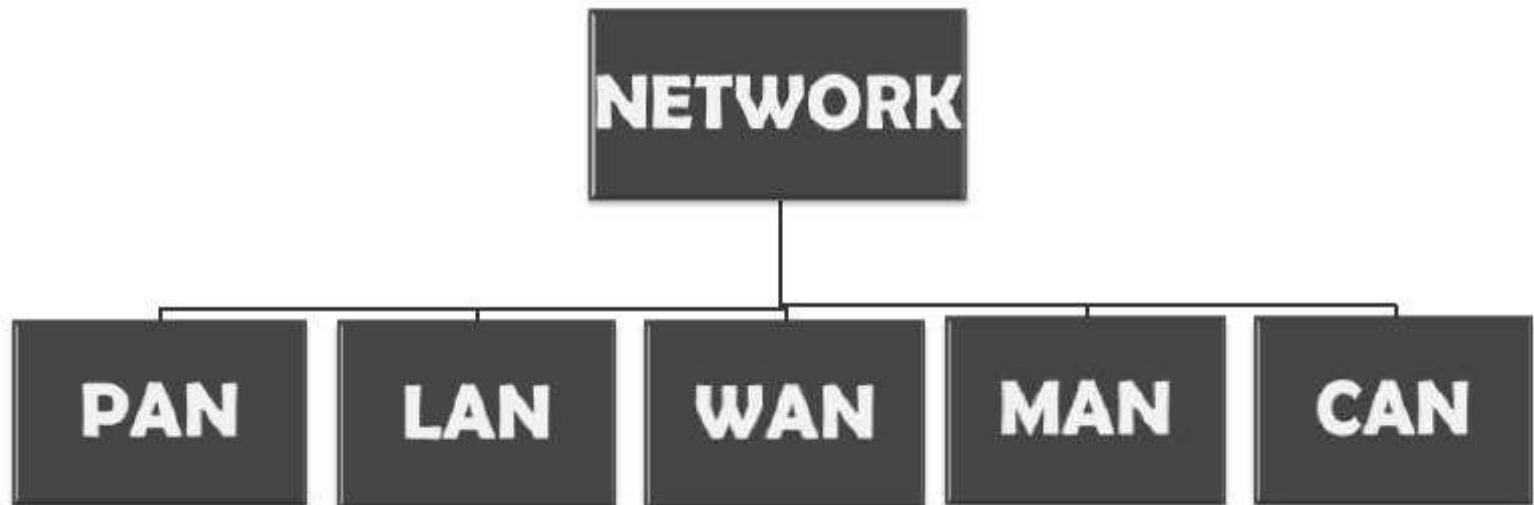
- An **Internetwork** is a collection of independent remote networks, LANs and WANs, and their connecting devices. They function together as one large network sharing connectivity resources.



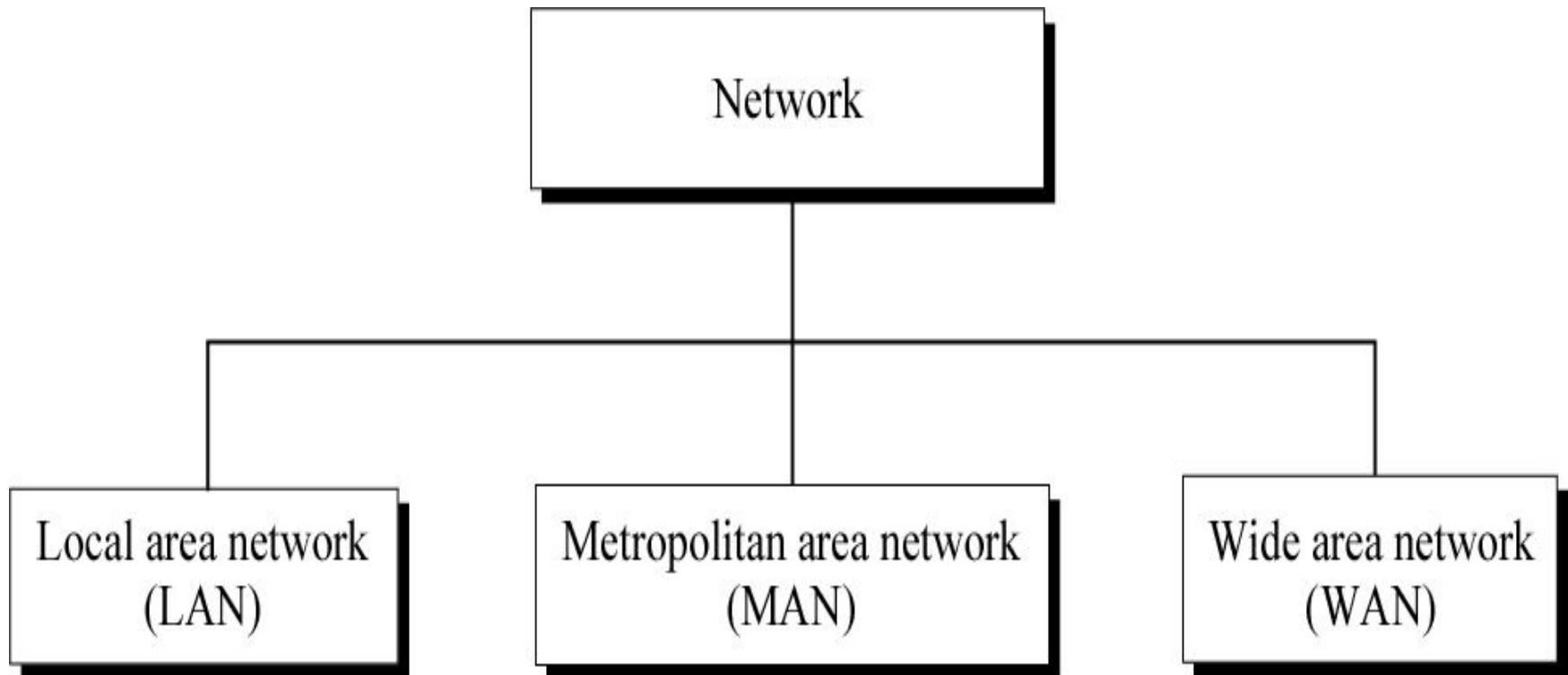
Internet (Internet)



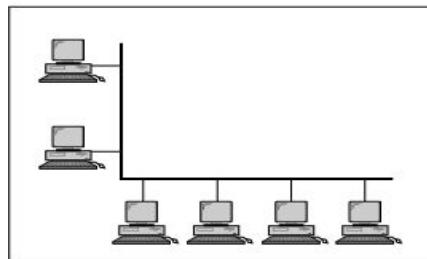
CLASSIFICATION OF AREA BY THEIR GEOGRAPHY



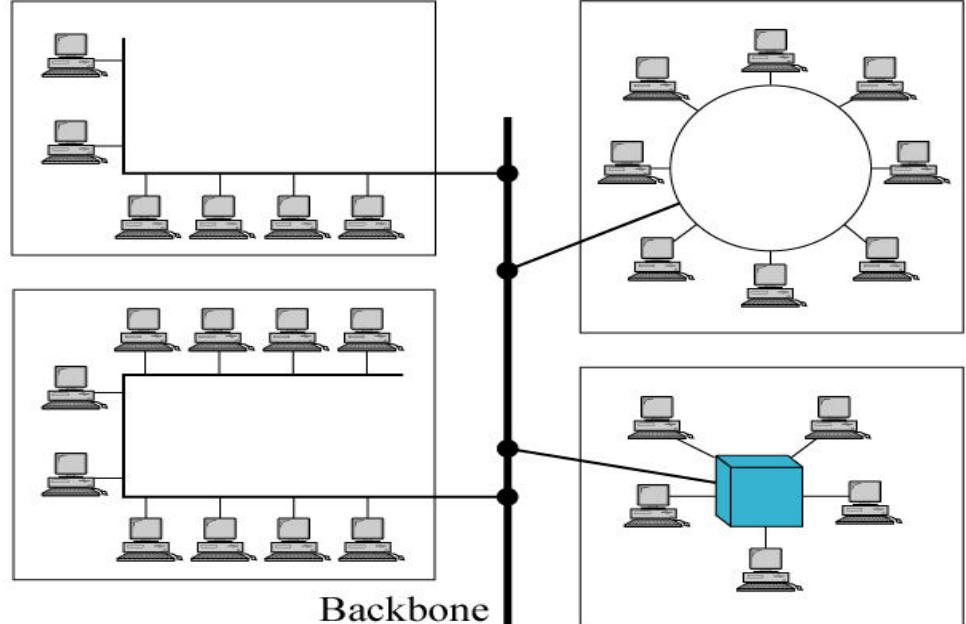
Categories of Networks



LAN – Local Area Network



a. Single-building LAN



b. Multiple-building LAN

The network can be categorized based on **its size, its ownership, the distance it covers, and its physical architecture.**

Interprocessor Distance:

1. LAN :

10m – Room, 100m - Building and 1km or 2 km – upto Campus.

LAN (Local Area Network)

- It covers a small geographical area with in a building or up to a few kilometers outside
- They are widely used to connect PC with in a office.
- LAN has distinguished from other networks by three characters.
 - size
 - their transmission technology
 - their Topology
- LAN run at speeds of 10 Mbps to 100 Mbps. or (100/1000Mbps)
- Different Topologies will be used for LAN Connectivity.
 - Bus / RING
- IEEE 802.3 known as Ethernet is an typical example for LAN

Advantages of LAN :

LAN provides a cost-effective multi-user computer environment.

A LAN is suited to any type of application.

Any number of users can be accommodated.

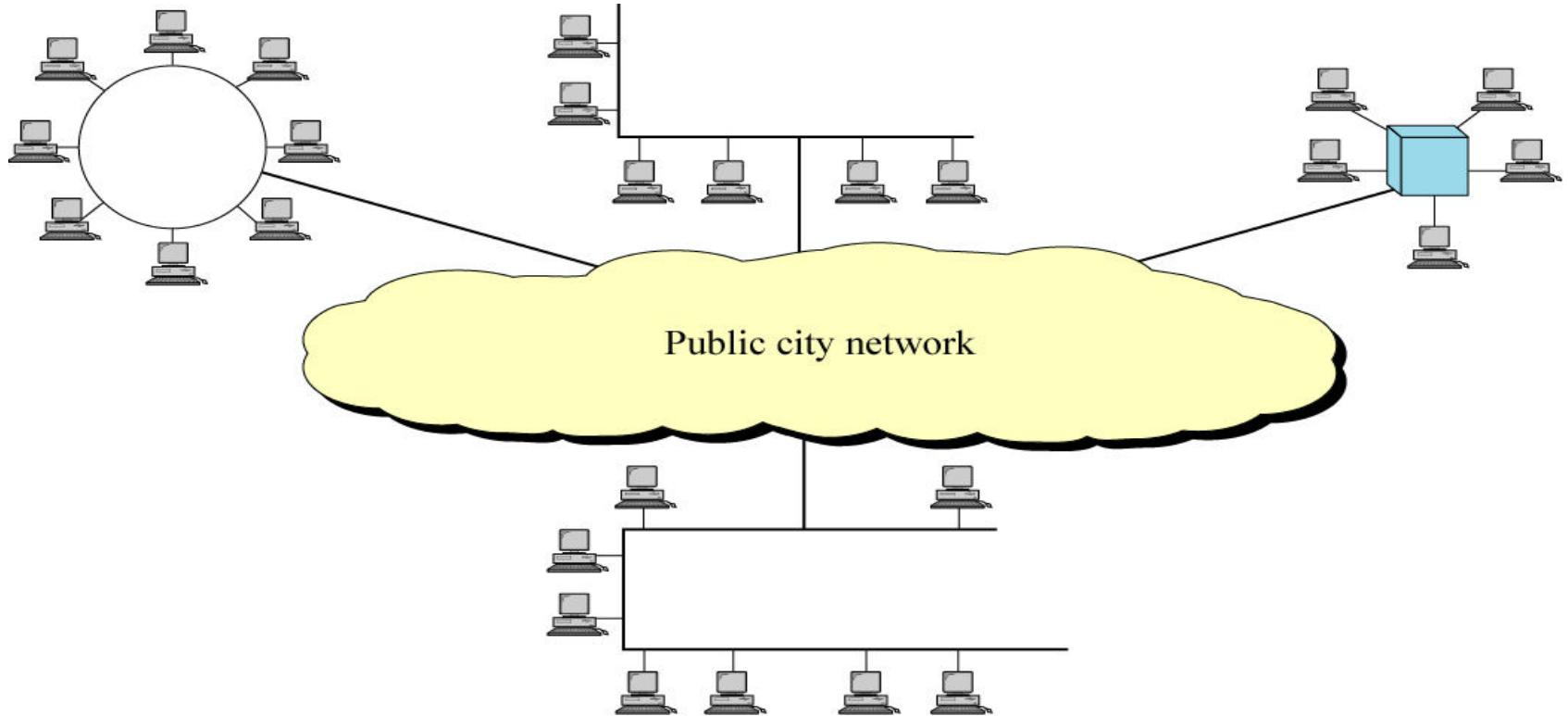
It is flexible and growth-oriented.

Data transfer rates in the 4 to 10 Mbps range.

Today speeds are normally 100 or 1000 Mbps.

It provide data integrity.

MAN (Metropolitan Area Network)



2. MAN.

10km or 20 km – upto City level.

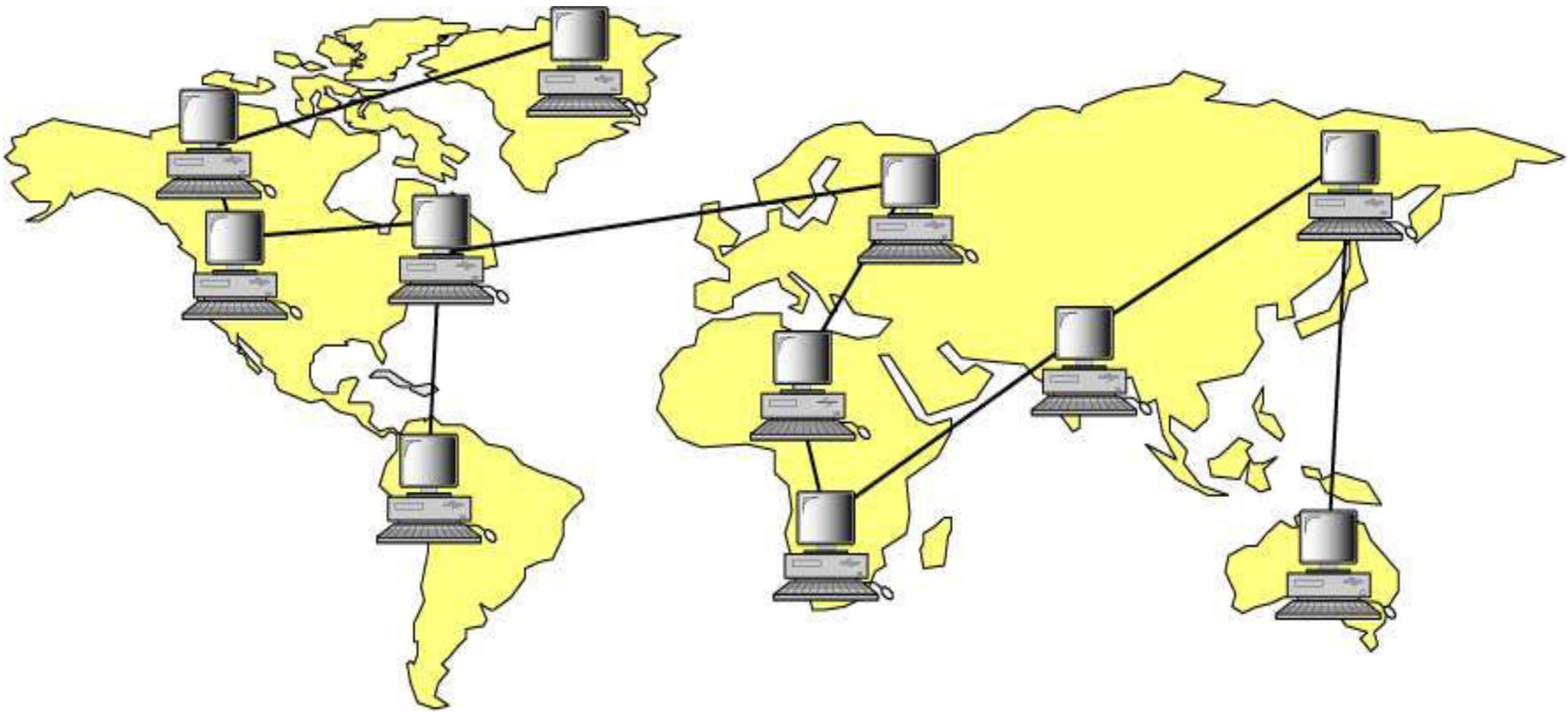
MAN (Metropolitan Area Network)

- MAN is a bigger network covers a group of nearby offices in a city .up to 10 – 20 kilometers range.
- MAN supports both voice and data. The typical example is Local Cable Network..
- LAN has distinguished from other networks by two characters.
 - standard that is adopted by them.
 - DQDB (Distributed Queue Dual Bus) – 802.6
- MAN run at speeds of 150 Mbps.
- Typical Topology will be used for MAN Connectivity.
 - BUS
- IEEE 802.6 known as Ethernet is an typical example for LAN.
- It may be a single network such as a cable TV network or it may be a means of connecting a number of LANs into a large network so that resources may be shared LAN-to-LAN as well as device-to-device.

MAN (Metropolitan Area Network) – Cont...

- MAN provides the transfer rates from 34 to 150 Mbps.
- A MAN is designed with two unidirectional buses.
- Each Bus is independent of the other in the transfer of traffic.
- The topology can be designed as an open bus or closed bus configuration.
- It can support both data and voice.
- The high speed links between LANs within a MAN are made possible by fiber-optic connection.

WAN - (Wide Area Network)

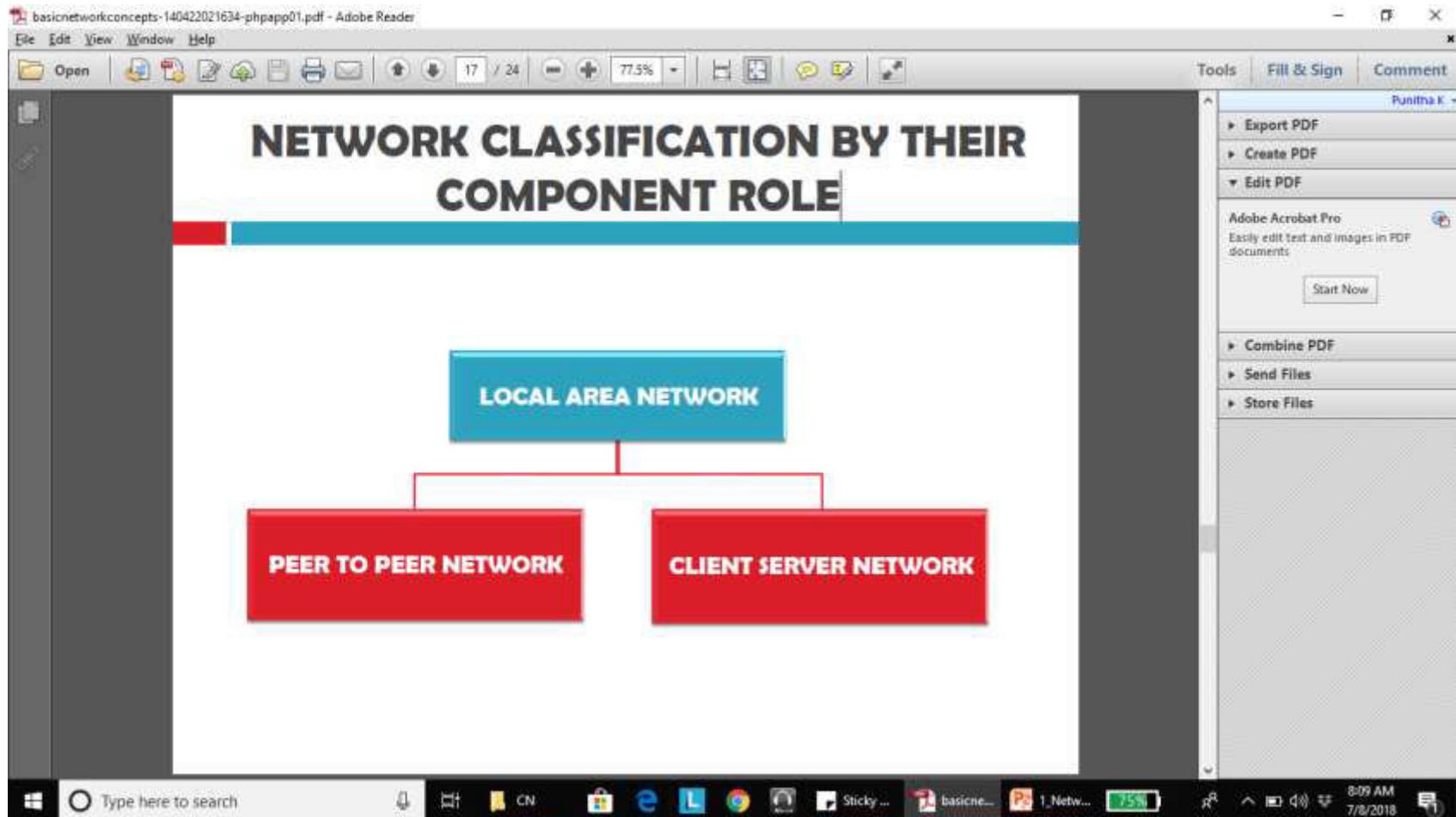


3. WAN : - 100km – upto Country level , 1000km – upto continent and 10,000km – upto Planet level.(The Internet).

WAN (Wide Area Network)

- WAN covers a large geographical area , country or continent.
- Hosts / Subnet
- The job of the Subnet is to carry the messages from host to host. subnet is an area in which the actual communication takes place.
- Subnet Consists of Two Distinct Components.
 - Transmission Lines
 - Switching Elements (Specialized Systems)
- Packet Switched Nodes / Router
- Inside the Subnet routers have a connectivity among themselves.
- Store and Forward Concept
- All the Topologies are applicable
- Works at 100 Mbps to 1000 Mbps.

NETWORK CLASSIFICATION BY THEIR COMPONENT ROLE



PEER TO PEER NETWORK

- In peer to peer network each computer is responsible for making its own resources available to other computers on the network.
- Each computer is responsible for setting up and maintaining its own security for these resources.
- Each computer is responsible for accessing the required network resources from peer to peer relationships.
- This network is useful for a small network containing less than 10 computers on a single LAN. Each computer can function as both client and server and do not have a central control system.
- There are no servers in peer network. Peer networks are amplified into home group.

ADVANTAGES & DISADVANTAGES OF PEER TO PEER NETWORK

The screenshot shows a Microsoft Windows desktop with an Adobe Acrobat Reader window open. The window title is "basicnetworkconcepts-140422021634-phpapp01.pdf - Adobe Reader". The main content of the PDF is as follows:

ADVANTAGES & DISADVANTAGES OF PEER TO PEER NETWORK

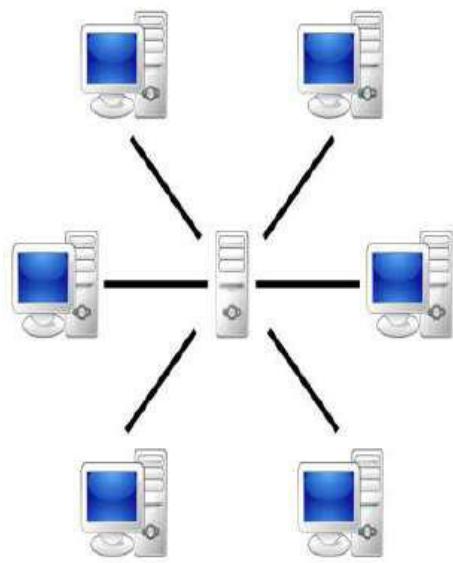
Advantages:

- Use less expensive computer hardware
- Easy to administer
- No NOS required
- More built in redundancy
- Easy setup & low cost

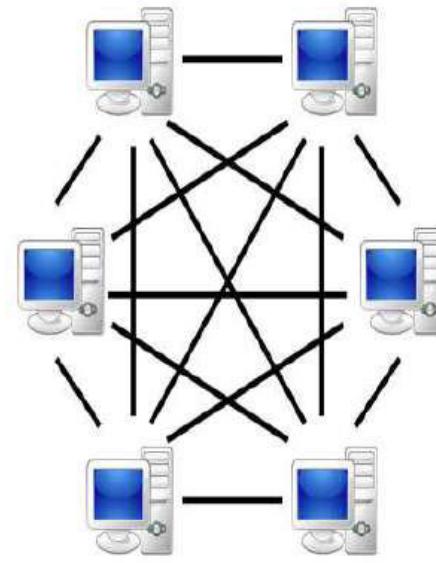
Disadvantages:

- Not very secure
- No central point of storage or file archiving
- Additional load on computer because of resource sharing
- Hard to maintain version control

The Adobe Acrobat interface includes a toolbar at the top, a menu bar with File, Edit, View, Window, Help, and a sidebar on the right labeled "Punitha K." with options like Export PDF, Create PDF, and Edit PDF.



Server-based



P2P-network

CLIENT/SERVER NETWORK

- In client-server network relationships, certain computers act as server and other act as clients. A server is simply a computer, that available the network resources and provides service to other computers when they request it. A client is the computer running a program that requests the service from a server.
- Local area network(LAN) is based on client server network relationship.
- A client-server network is one in which all available network resources such as files, directories, applications and shared devices, are centrally managed and hosted and then are accessed by client.
- Client serve network are defined by the presence of servers on a network that provide security and administration of the network.

ADVANTAGES AND DISADVANTAGES OF CLIENT- SERVER NETWORK

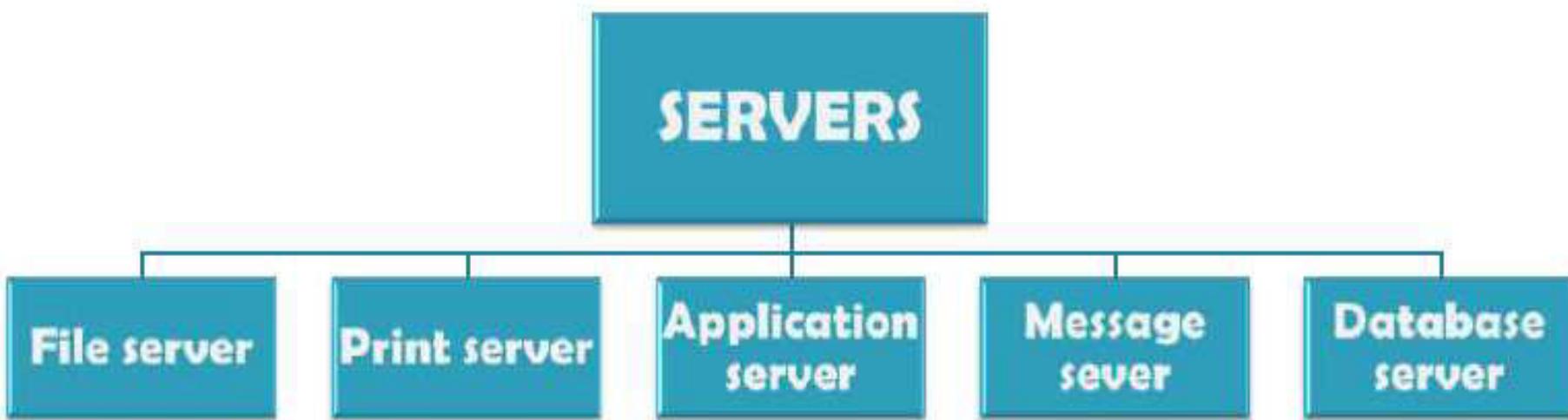
Advantages:

- **Very secure**
- **Better performance**
- **Centralized backup**
- **very reliable**

Disadvantages:

- **requires professional administration**
- **More hardware-intensive**
- **More software intensive**
- **Expensive dedicated software**

TYPES OF SERVERS



TYPES OF SERVERS

- **File server:** provides services for storing, retrieving and moving data. User can read/write/exchange/manage files with help of file servers
- **Printer server:** used for controlling and managing printing on the network. It also offers the fax service to the network users.
- **Application server:** helps to share expensive software and additional computing power by the computers in a network.
- **Message server:** used to co-ordinate the

Types of network devices

- Hub
- Switch
- Router
- Bridge
- Gateway
- Modem
- Repeater
- Access Point



REPEATER

- A repeater is an electronic device that amplifies the signal it receives.
- It receives a signal and retransmits it at a higher level or higher power so that the signal can cover longer distances, more than 100 meters for standard LAN cables.
- Repeaters work on the Physical layer.

REPEATER



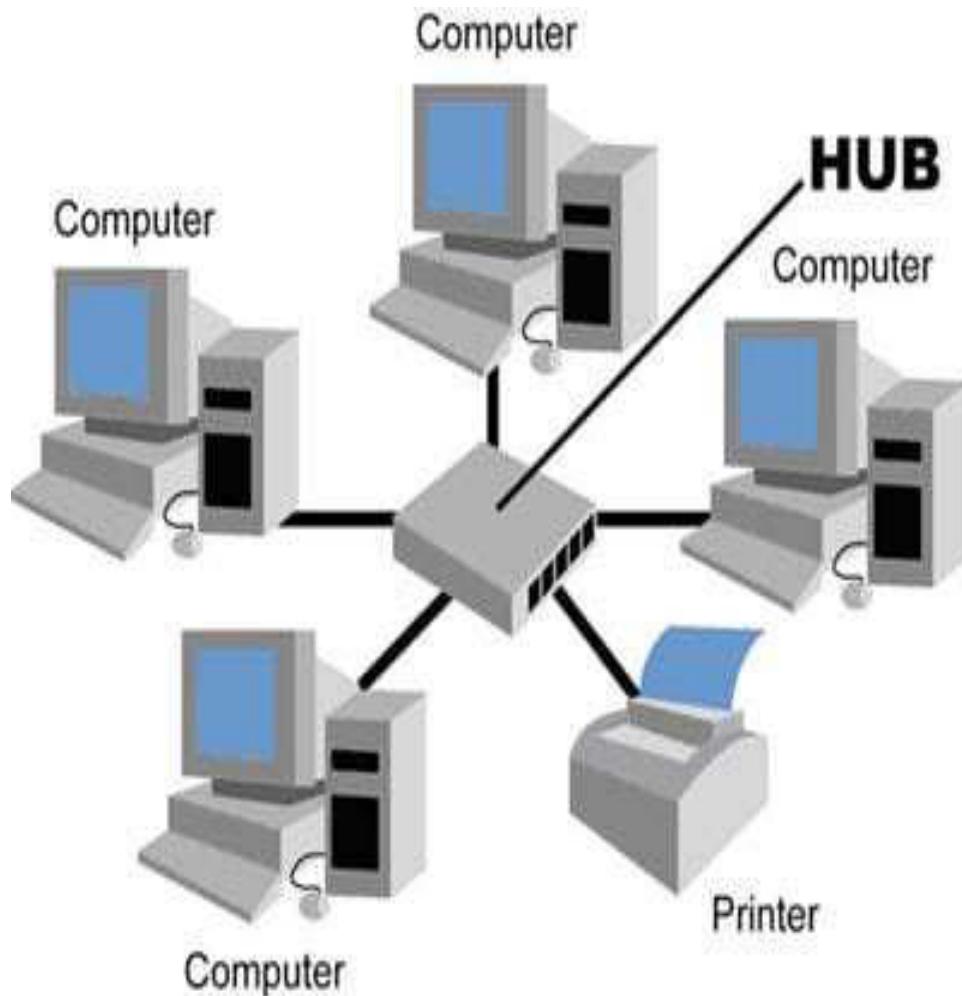
HUB

- Hubs connect multiple computer networking devices together.
- A hub also acts as a repeater in that it amplifies signals that deteriorate after traveling long distances over connecting cables.
- A hub is the simplest in the family of network connecting devices because it connects LAN components with identical protocols.

HUB



HUB



Switch

- Switches generally have a **more intelligent role than hubs**.
- A switch is a multiport device that improves network efficiency.
- The switch maintains limited routing information about nodes in the internal network, and it allows connections to systems like hubs or routers.
- Strands of LANs are usually connected using switches.
- Generally, switches **can read the hardware addresses of incoming packets** to transmit them to the appropriate destination.

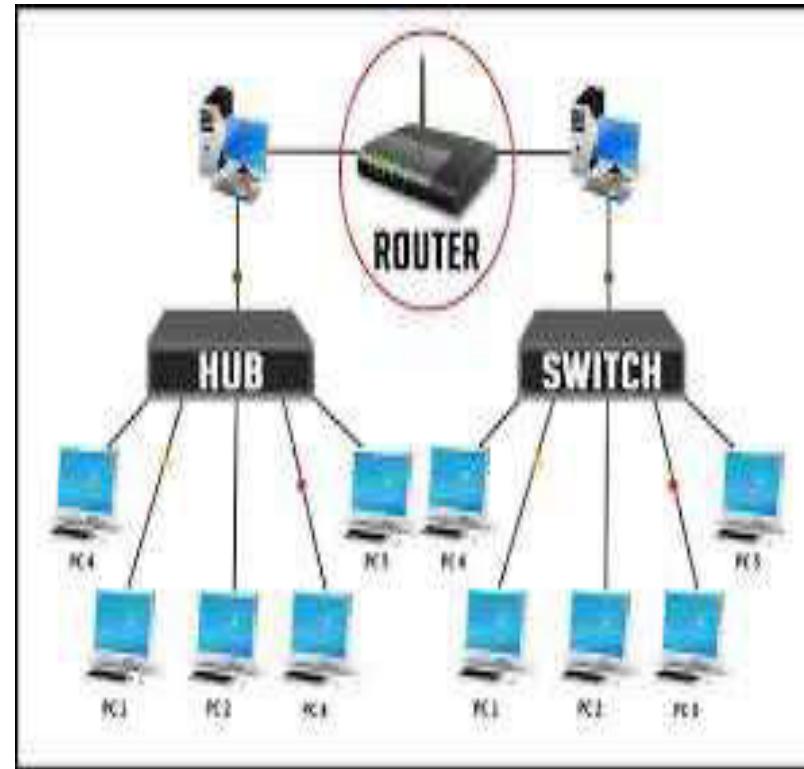
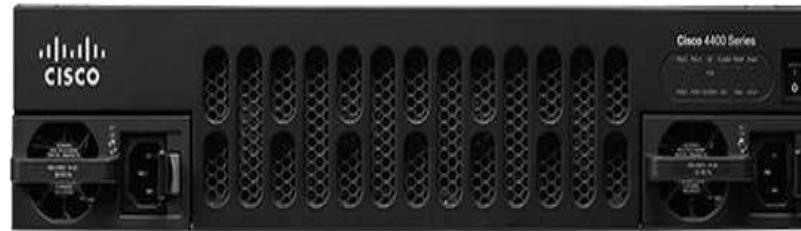
SWITCH



ROUTER

- Routers help transmit packets to their destinations by charting a path through the sea of interconnected networking devices using different network topologies.
- **Routers are intelligent devices**, and they store information about the networks they're connected to.
- Routers are general-purpose devices that interconnect two or more heterogeneous networks.

ROUTER

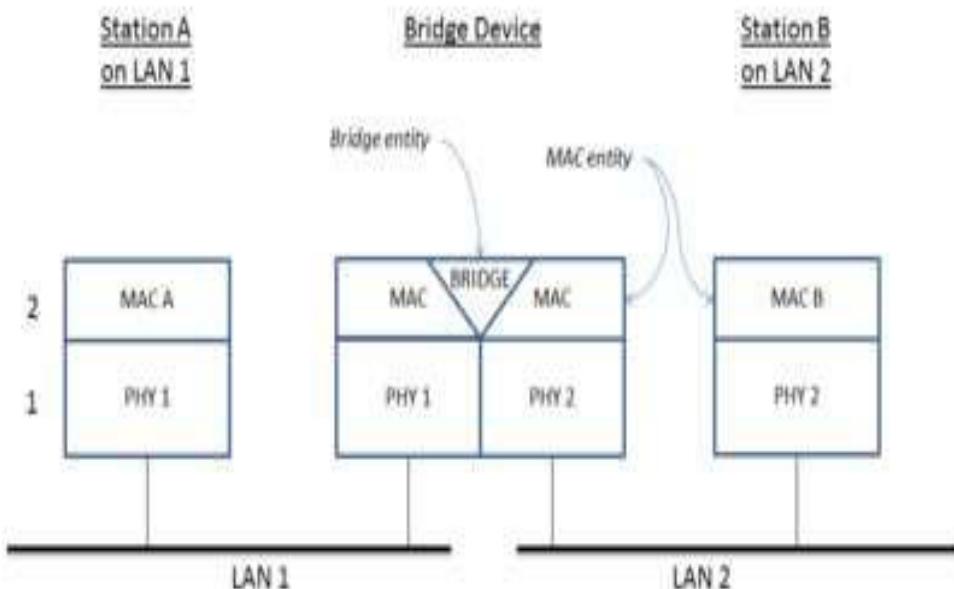


BRIDGE

- Bridges are used to connect two or more hosts or network segments together.
- The basic role of bridges in network architecture is storing and forwarding frames between the different segments that the bridge connects.

BRIDGE

A bridge connecting two LAN segments.



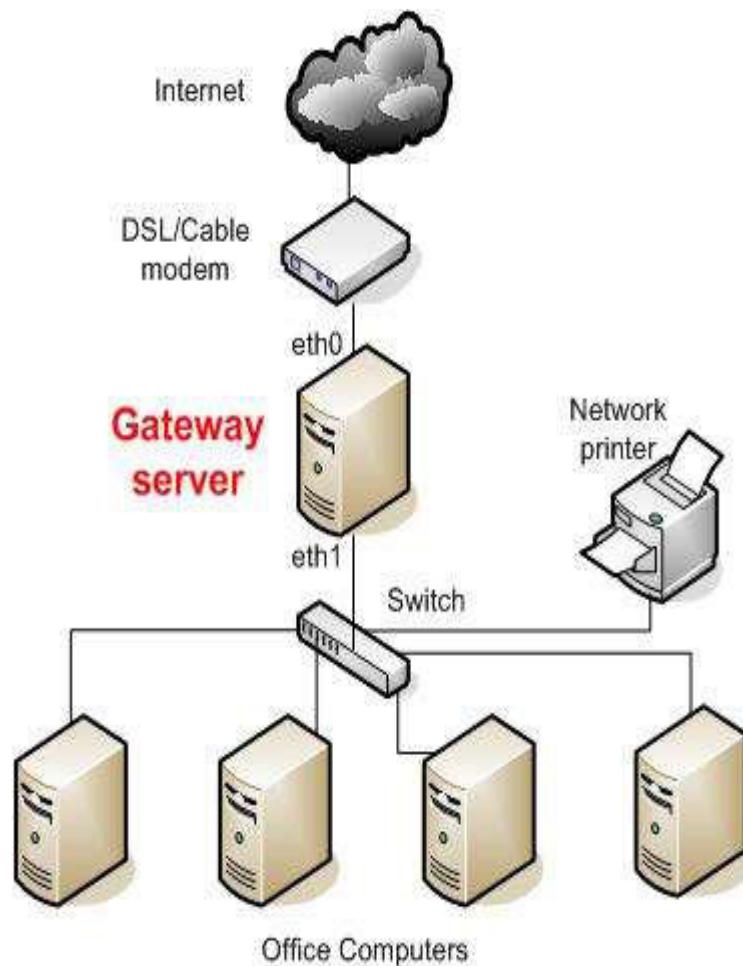
GATEWAY

- Gateways normally work at the Transport and Session layers of the OSI model.
- Gateways provide translation between networking technologies such as Open System Interconnection (OSI) and Transmission Control Protocol/Internet Protocol (TCP/IP).
- Gateways connect two or more autonomous networks, each with its own routing algorithms, protocols, topology, domain name service, and network administration procedures and policies.

GATEWAY...

- Gateways perform all of the functions of routers and more.
- In fact, a router with added translation functionality is a gateway.
- The function that does the translation between different network technologies is called a protocol converter.

GATEWAY



MODEM – Modulator+Demodulator

- Modems (modulators-demodulators) are used to transmit digital signals over analog telephone lines.
- Thus, digital signals are converted by the modem into analog signals of different frequencies and transmitted to a modem at the receiving location.
- The receiving modem performs the reverse transformation and provides a digital output to a device connected to a modem, usually a computer.
- Modems work on both the Physical and Data Link layers.

MODEM



ACCESS POINT

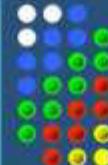
- While an access point (AP) can technically involve either a wired or wireless connection, it commonly means a wireless device.
- An AP works at the second OSI layer, the Data Link layer, and it can operate either as a bridge connecting a standard wired network to wireless devices or as a router passing data transmissions from one access point to another.
- Access points typically are separate network devices with a built-in antenna, transmitter and adapter.

ACCESS POINT



4. Topology

- Network topology is the arrangement of the elements of a communication network. Network topology can be used to define or describe the arrangement of various types of telecommunication networks, including command and control radio networks, industrial field busses and computer networks.



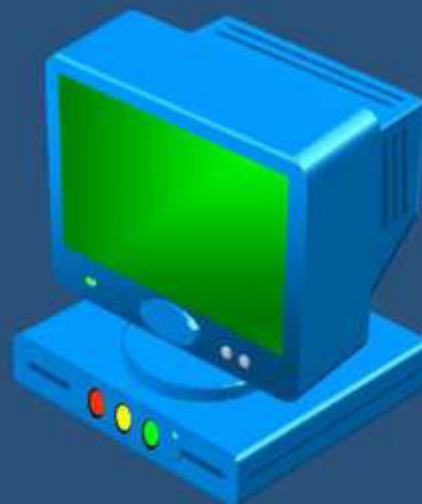
Network TOPOLOGIES

What are network topologies?

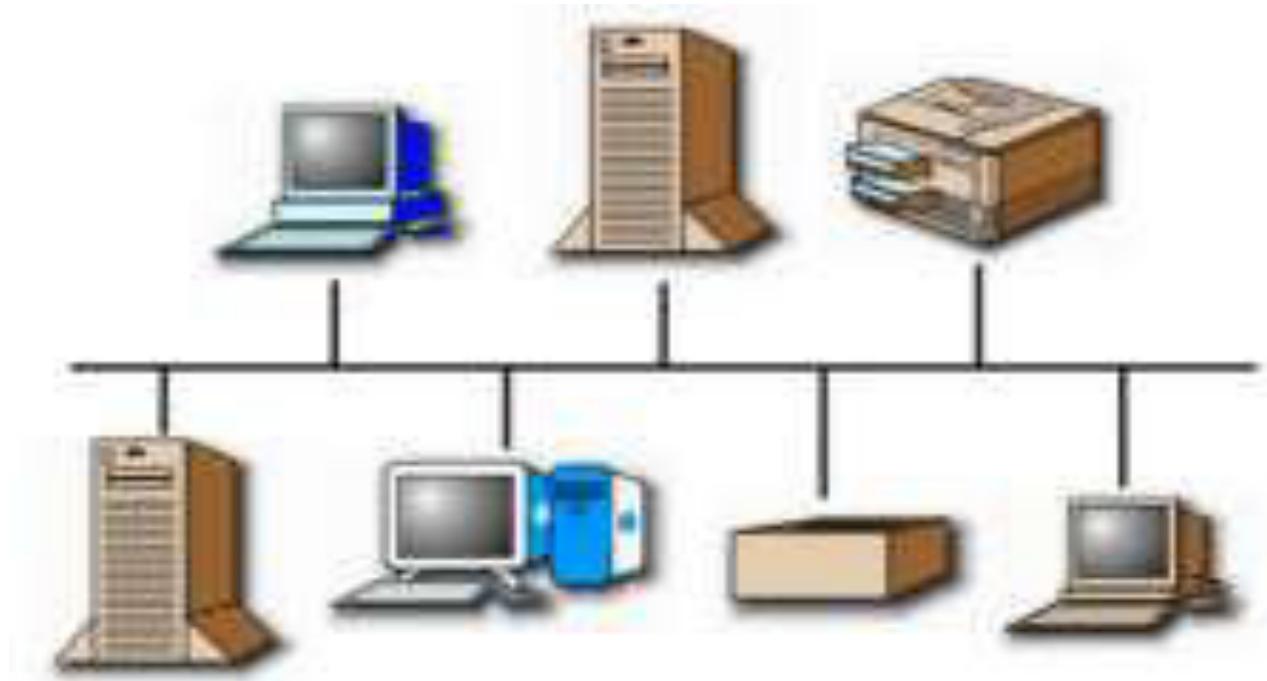
A topology is the layout of how a network communicates with different devices.

There are a couple of different categories of topologies.

Wired and wireless.

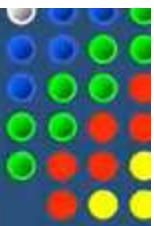


Bus Topology Network



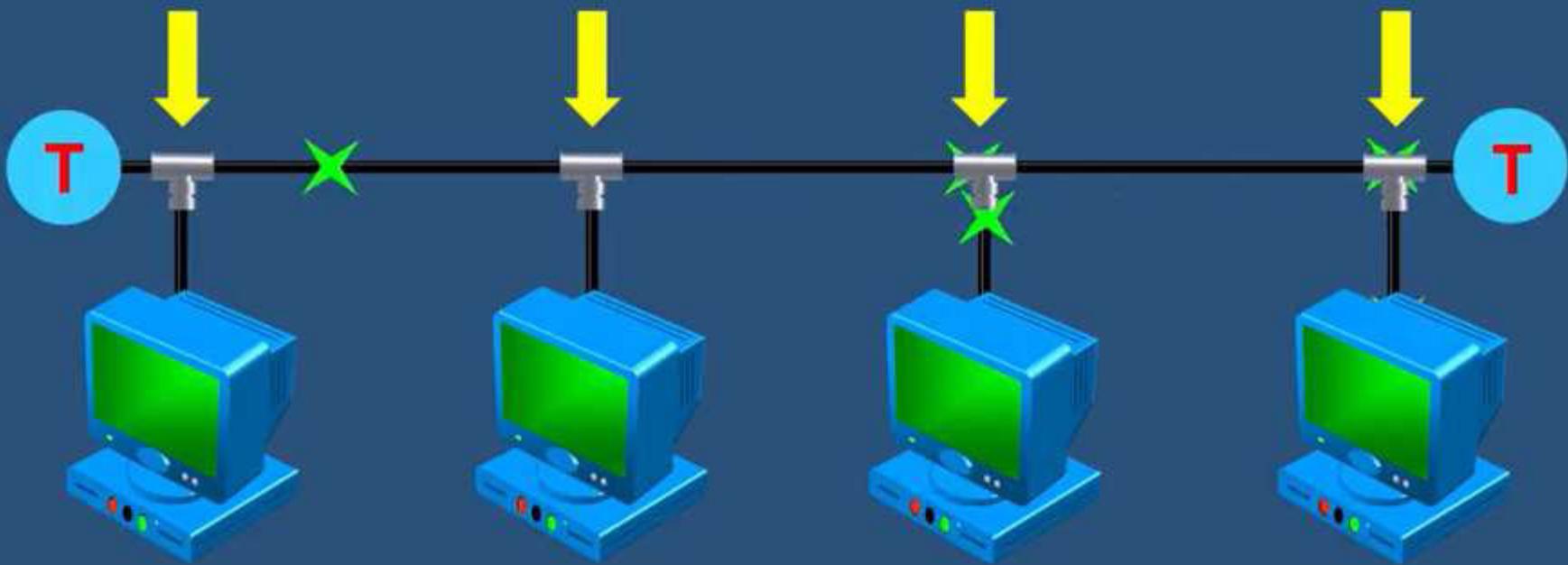
- A single cable connects each workstation in a linear, daisy-chained fashion.
- * Signals are broadcasted to all stations, but stations only act on the frames addressed to them

Bus TOPOLOGY



COAXIAL CABLE

BNC CONNECTOR

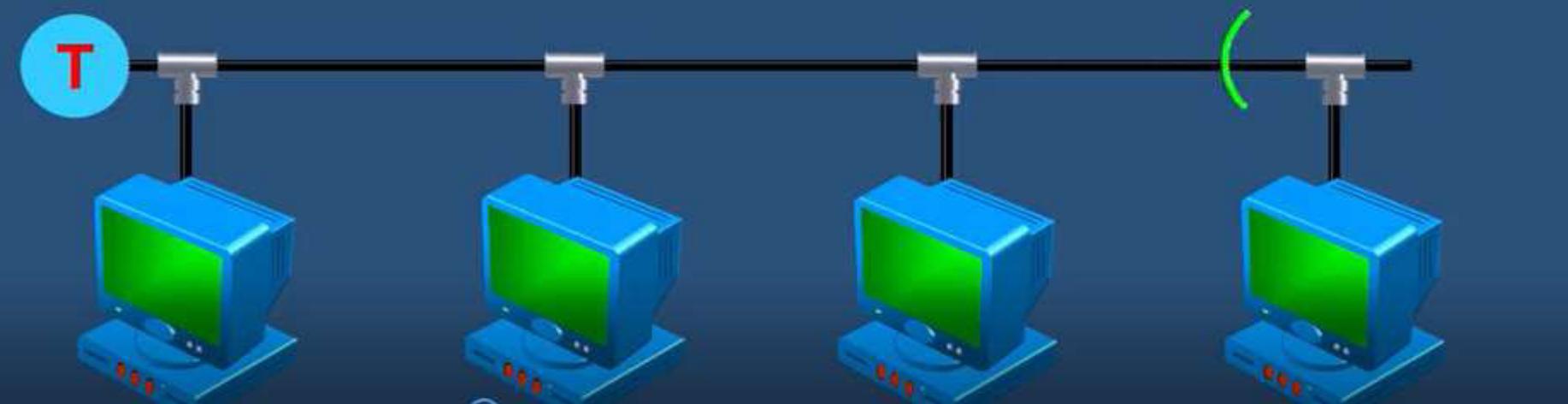


Bus TOPOLOGY

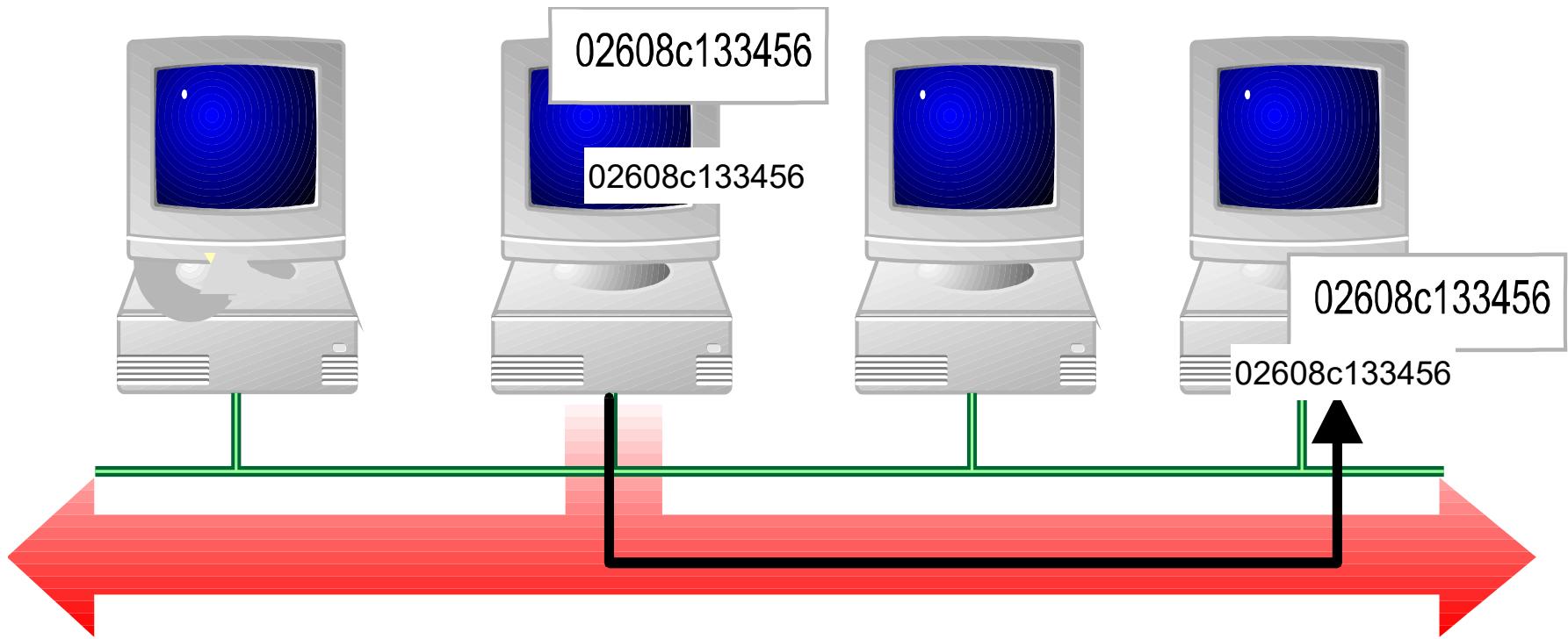


COAXIAL CABLE

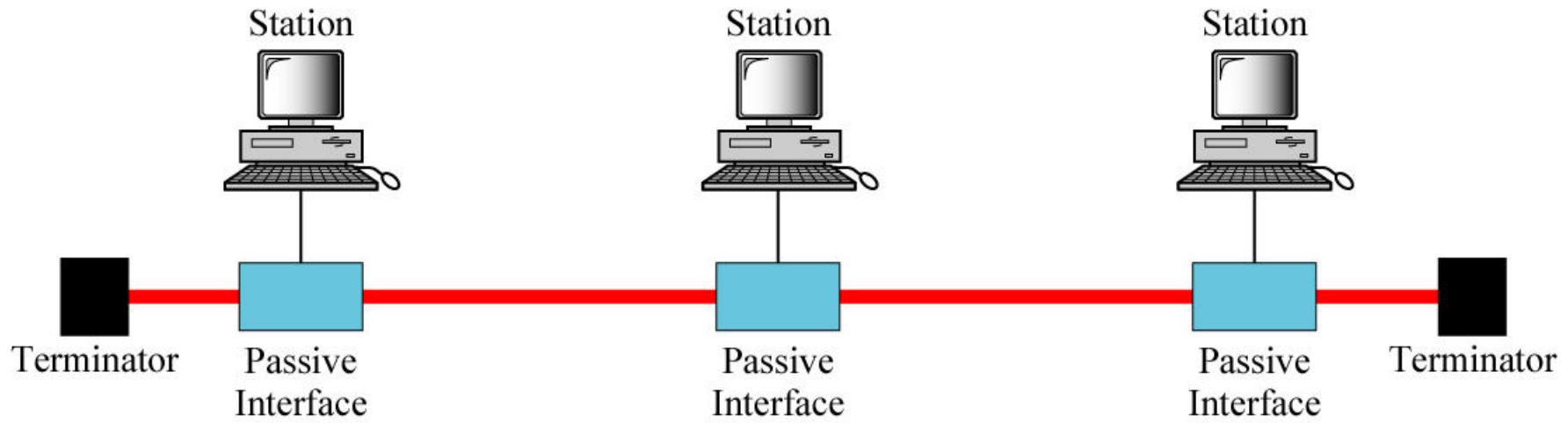
BNC CONNECTOR



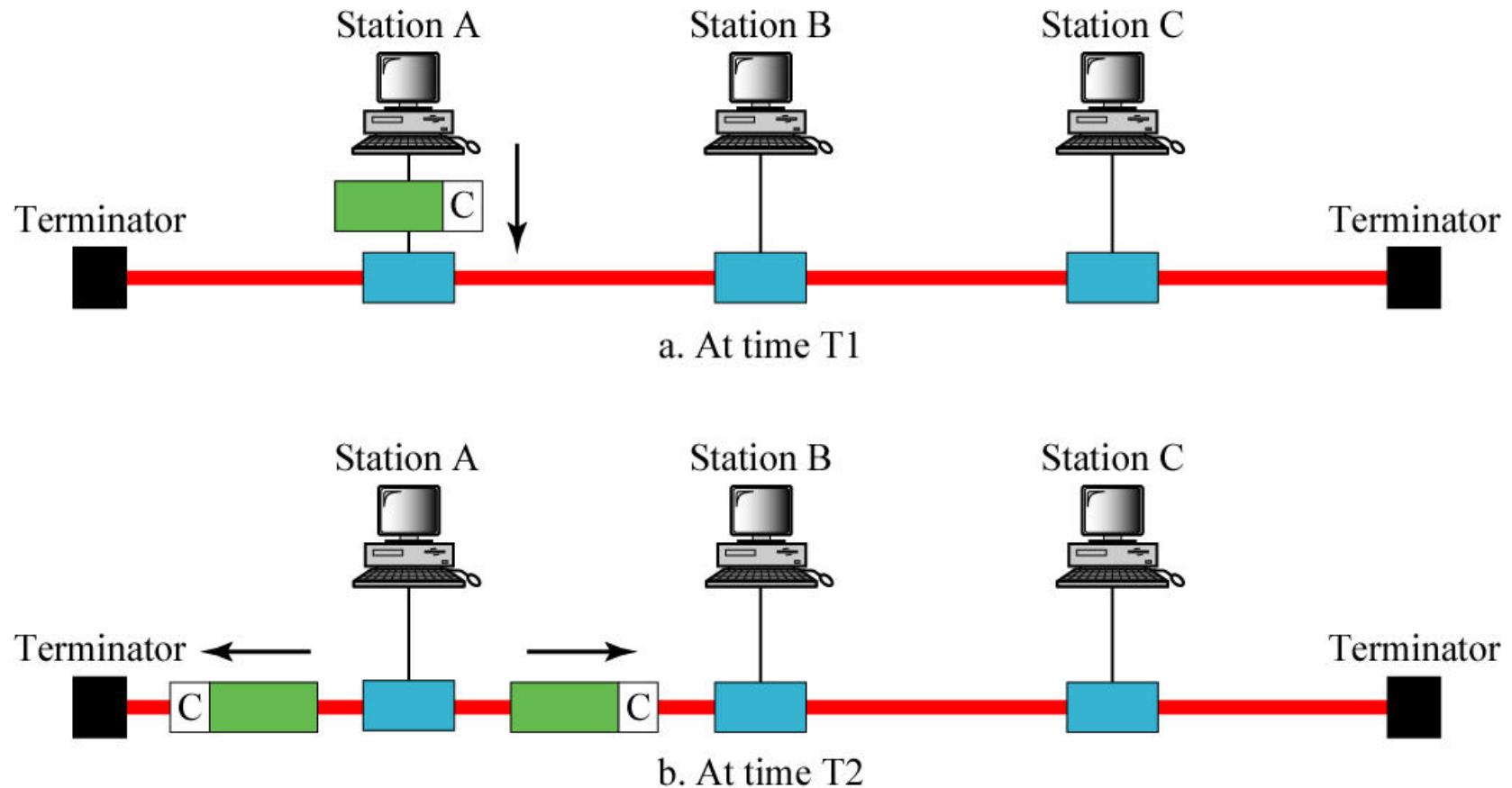
Data is sent to all computers, but only the destination computer accepts



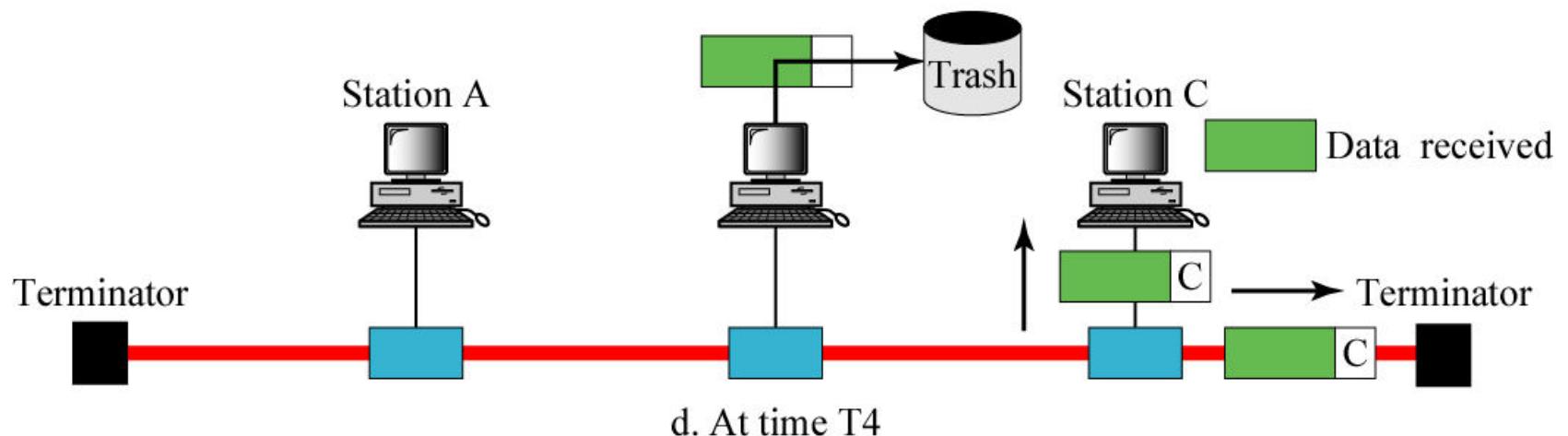
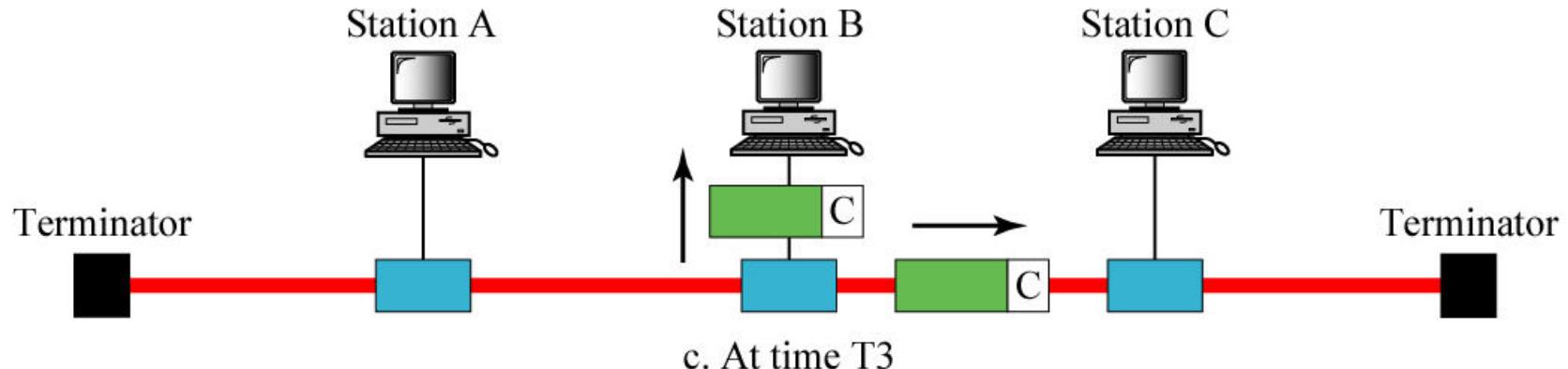
Bus Topology



Bus Topology Operation



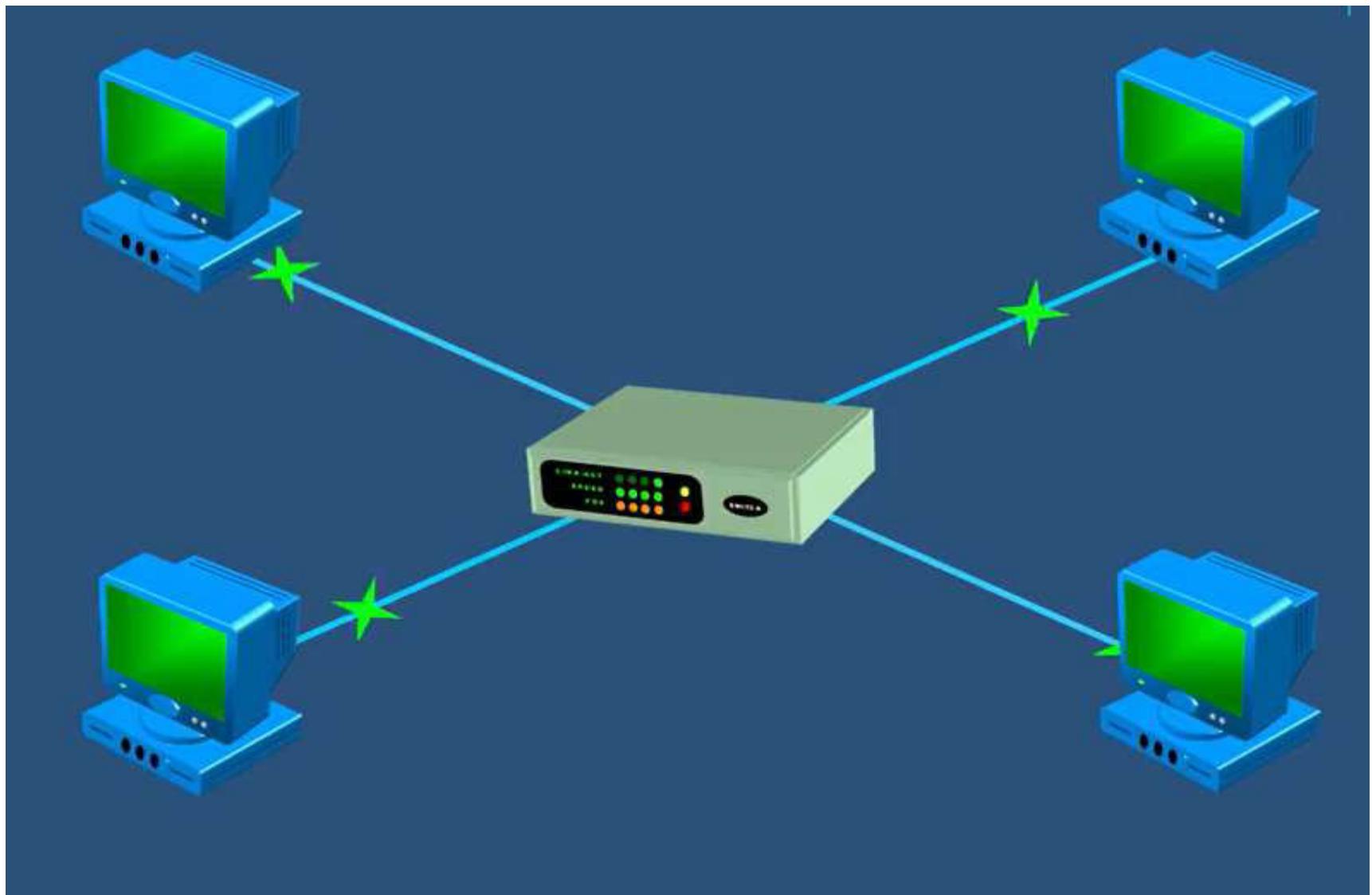
Bus Topology Operation



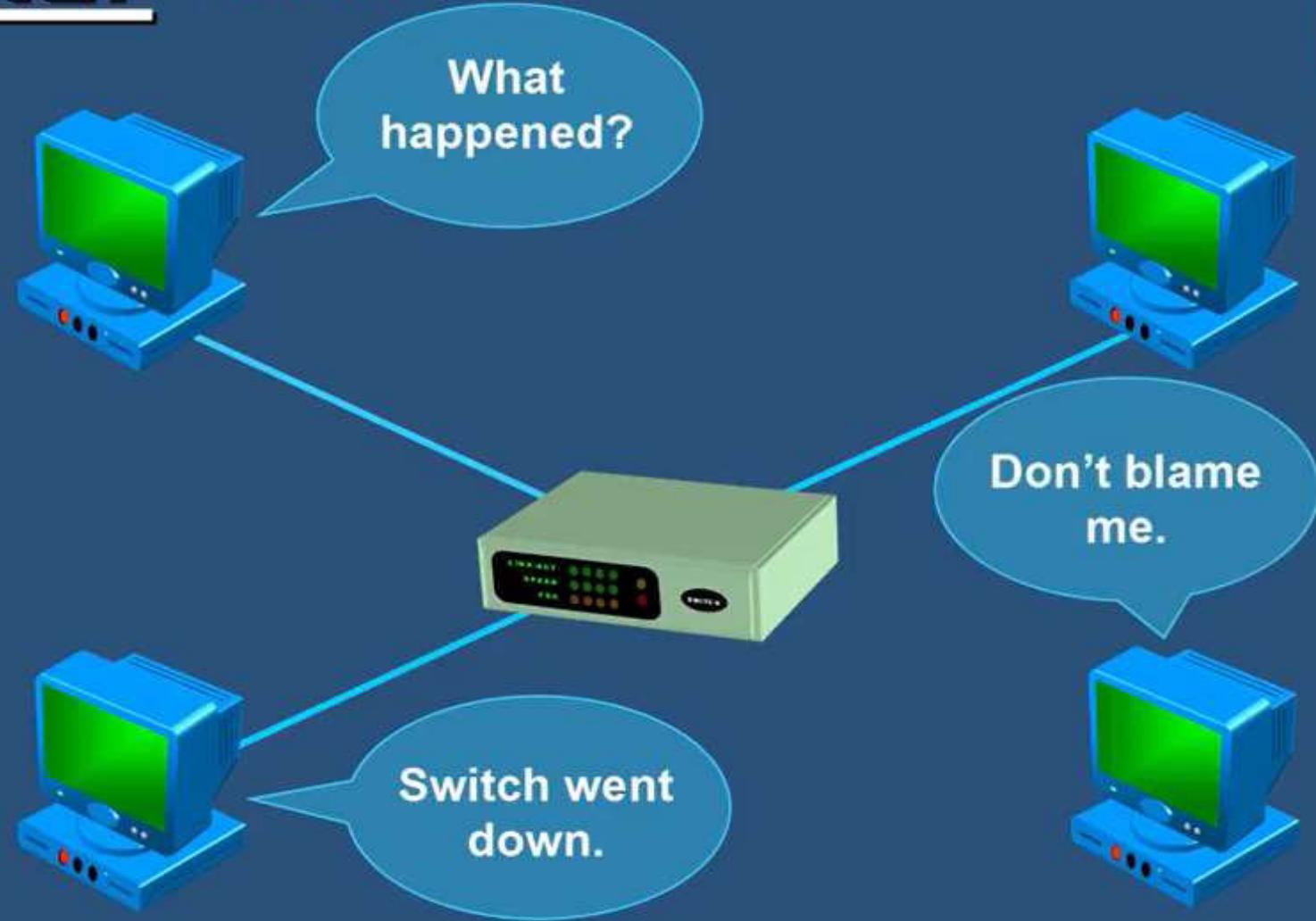
Advantages and Disadvantages of Bus Topology

- Reliable in very small network.
 - Easy to setup.
 - Easy to extend the network with the help of repeaters.
 - Easy to connect the segments with a barrel connector.
-
- Heavy network traffic will slow down the network.
 - Each barrel connector weakens the signal.
 - It is difficult to trouble shoot.
 - Termination is required on both the end systems.
 - Any break in the cable brings the entire network down

Simple star network



Star TOPOLOGY



Simple star network

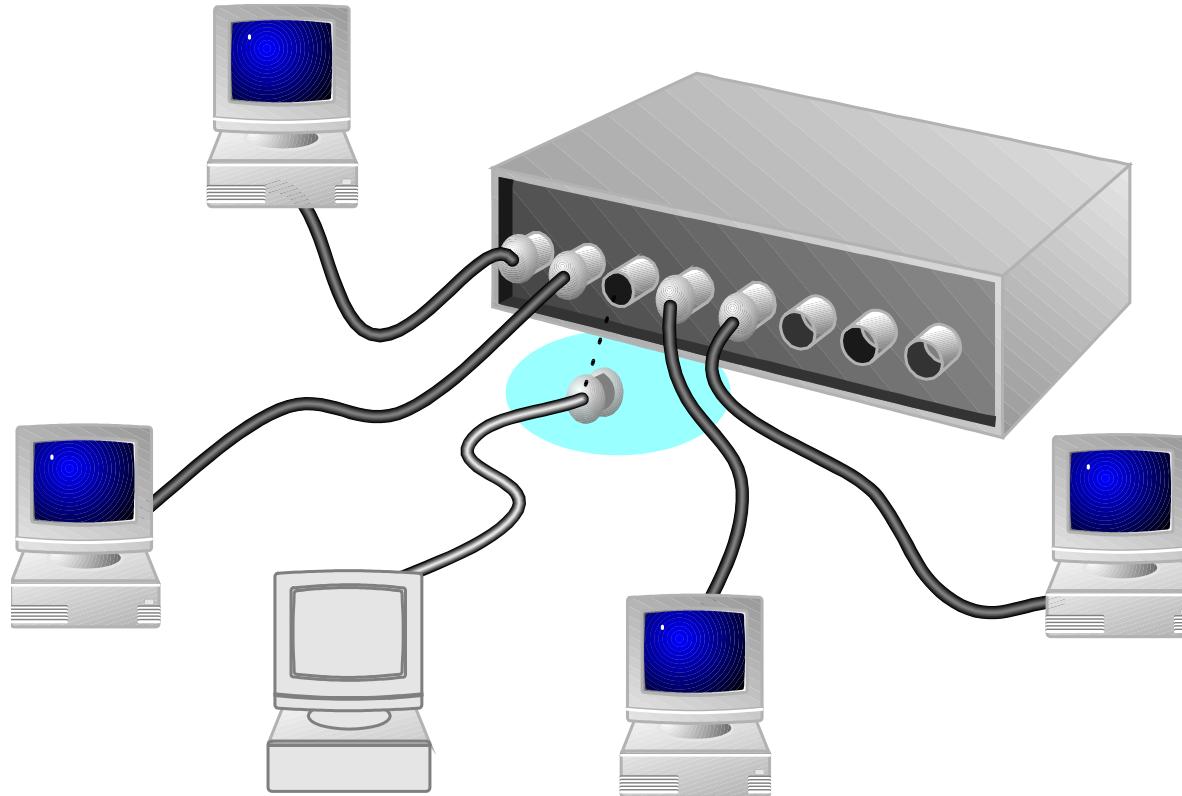


In a star topology, each station is connected to a central hub or concentrator that functions as a multi-port repeater. Each station broadcasts to all of the devices connected to the hub.

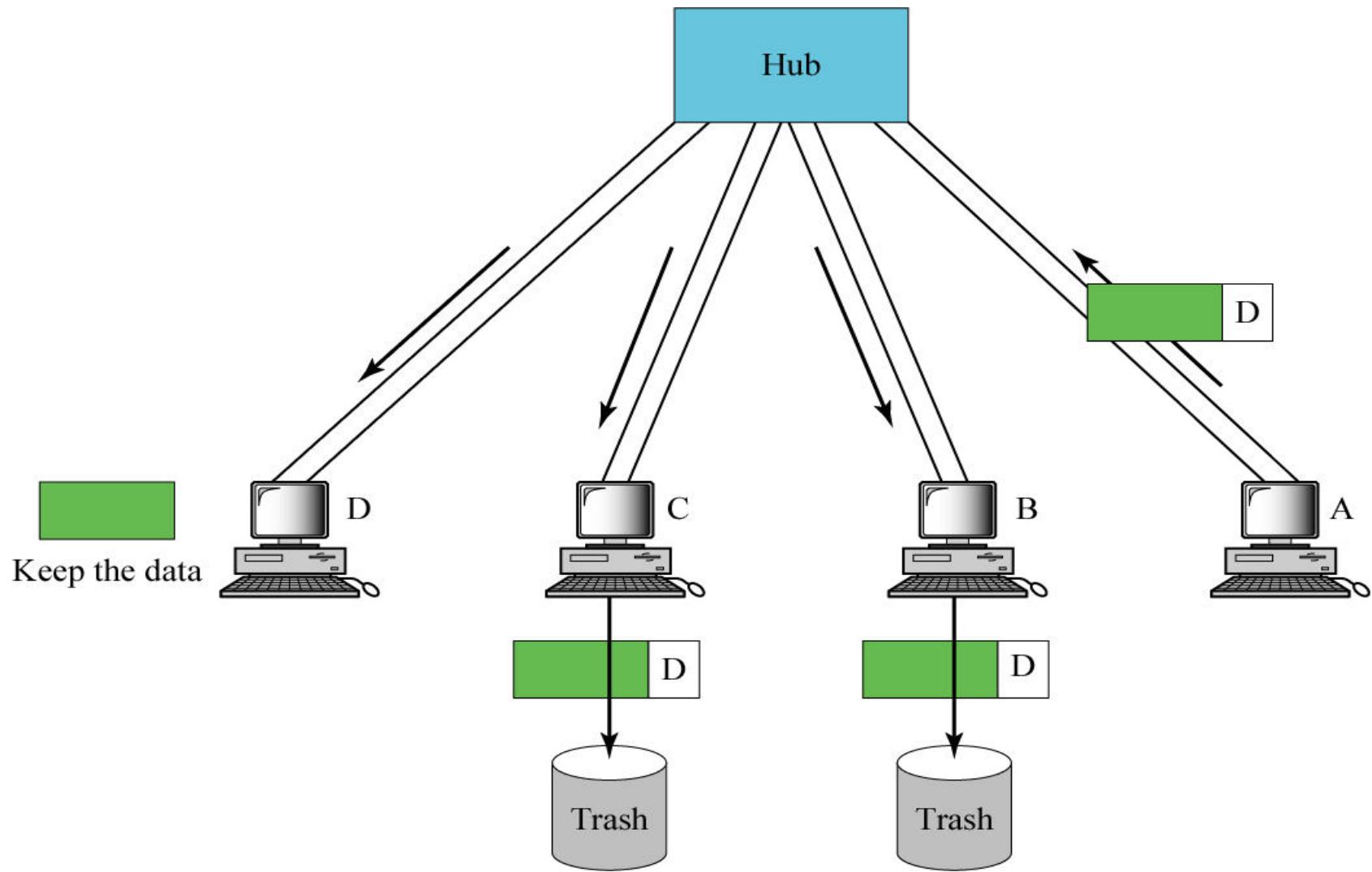
Advantages and Disadvantages of Star Topology

- It easy to modify and add more computers in the network.
 - Easy to trouble shoot.
 - Single computer failure do not effect the network.
 - Other cable types can be used in the same network.
-
- If the central hub fail the network is down.
 - Most of the star network require a central device to rebroadcast or switch the network traffic.
 - Network cabling is more .

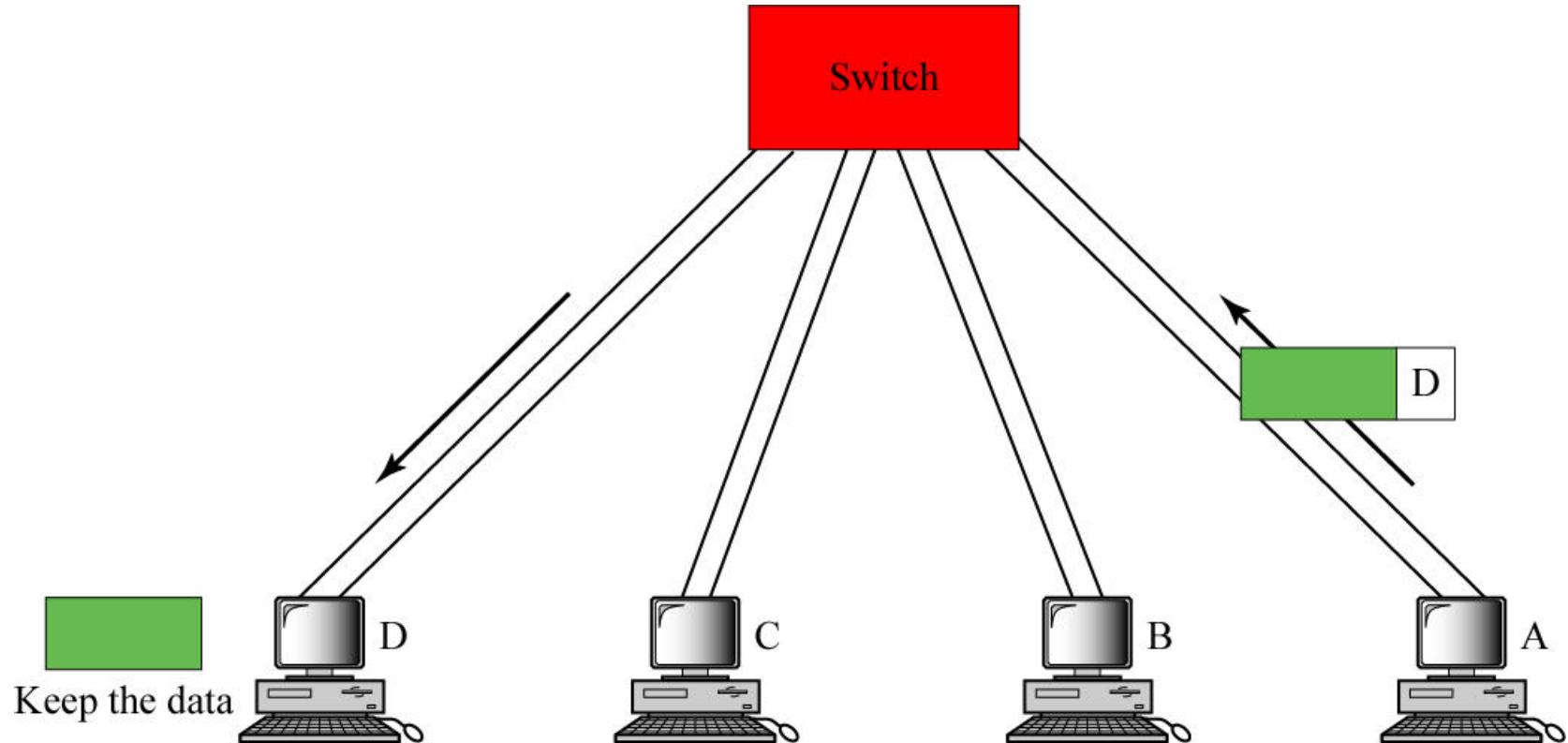
A break or unplugged cable takes down the only unplugged computer



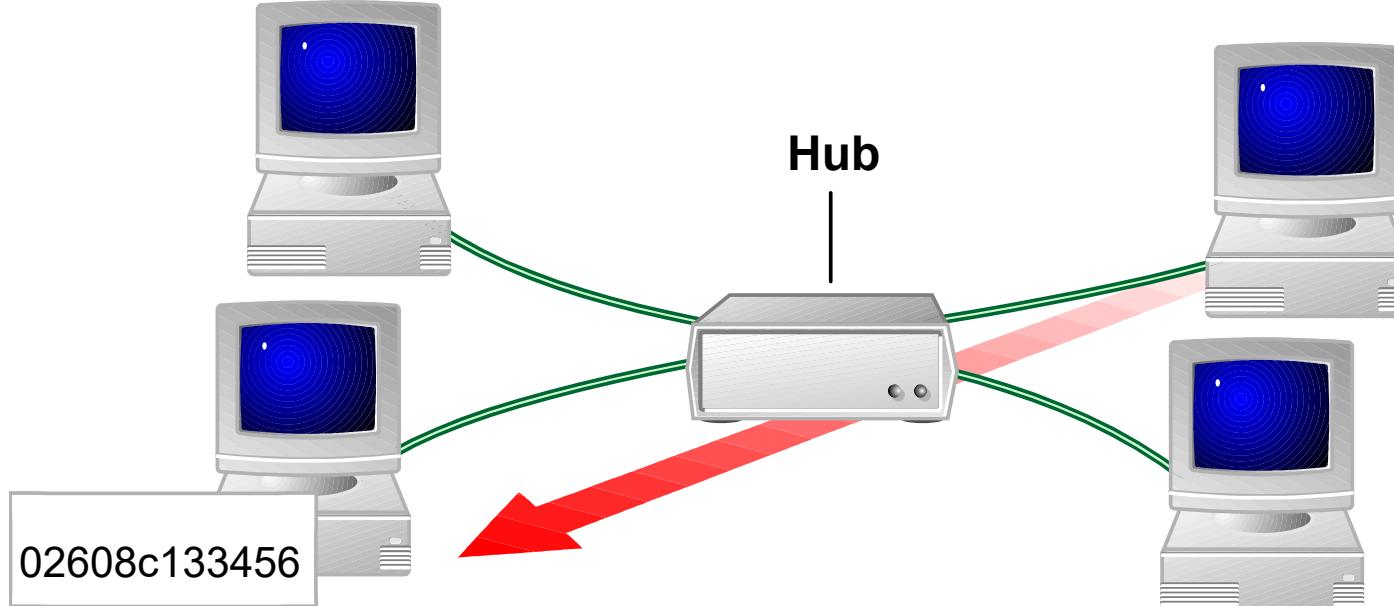
Using a Hub in a Star Topology



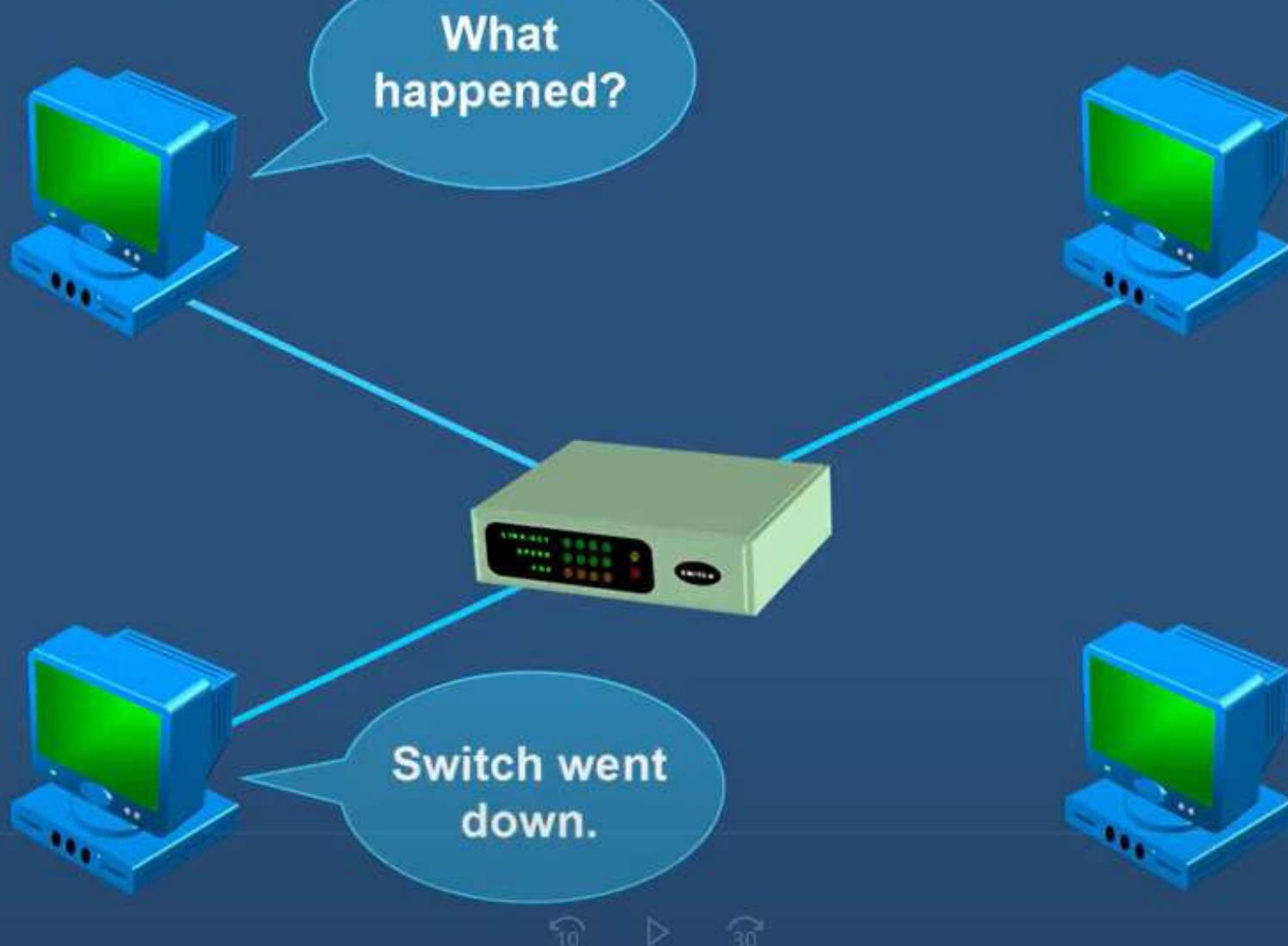
Using a Switch in a Star Topology



A hub is the central point in a star topology



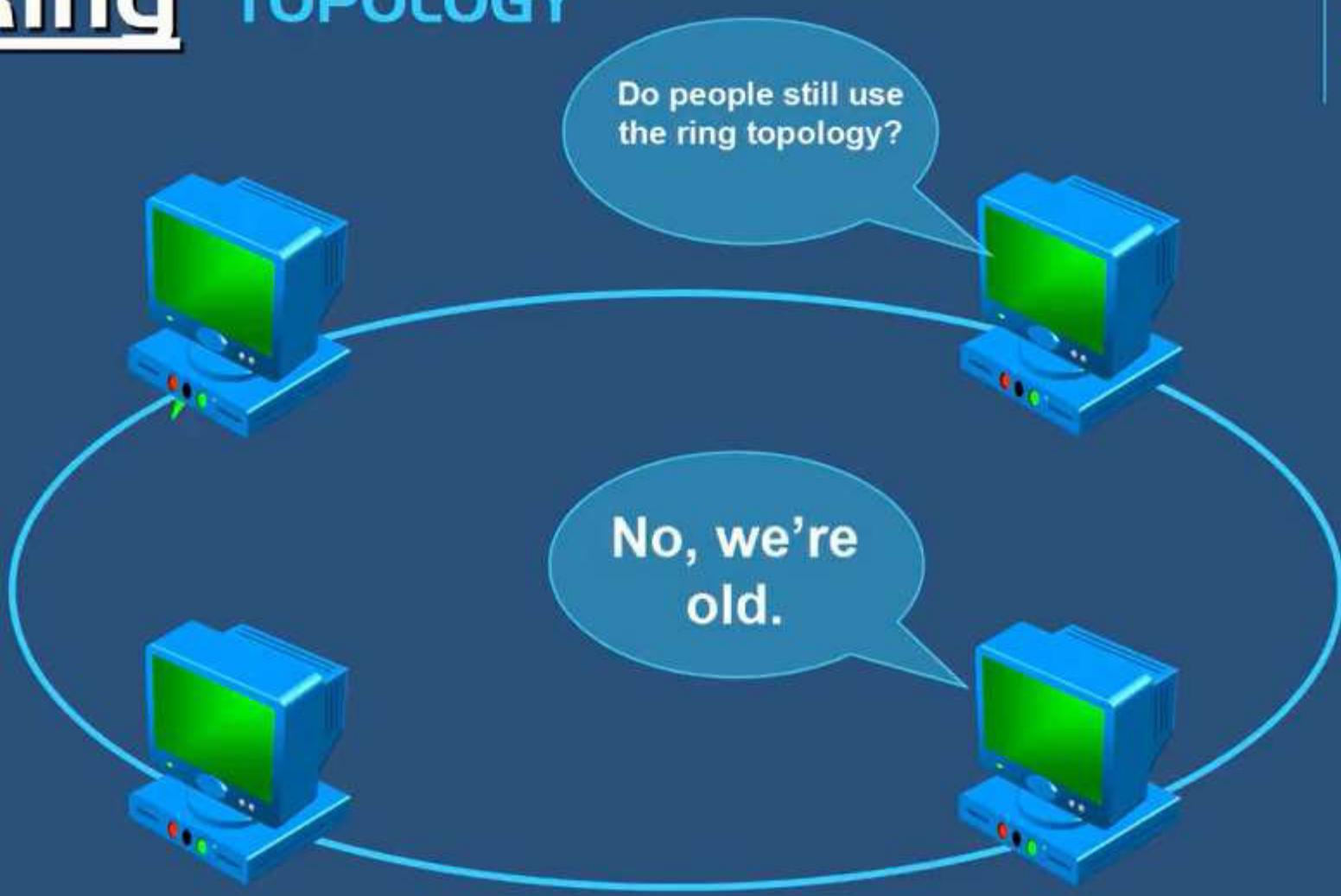
Star TOPOLOGY



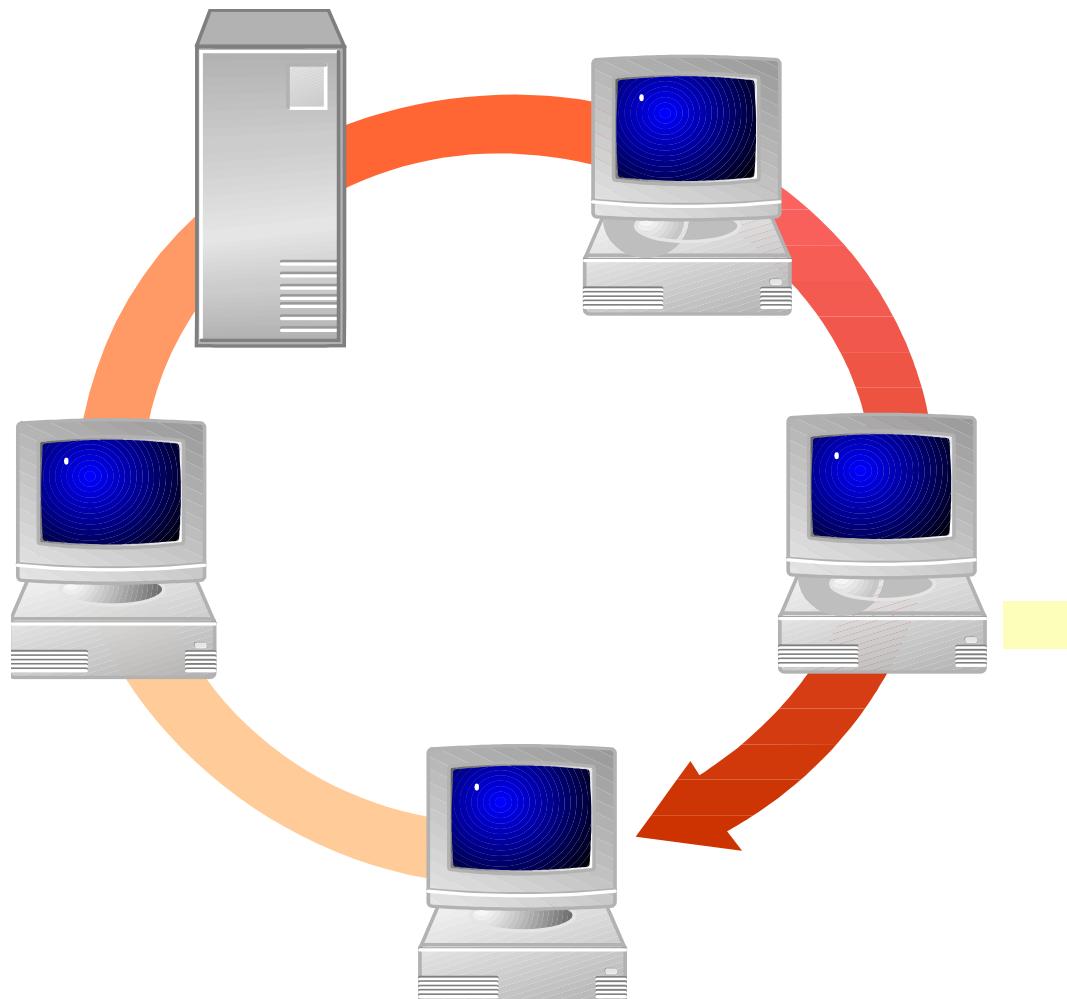
Ring Topology

- A ring topology is a LAN architecture that consists of a series of devices connected to one another by unidirectional transmission links to form a single closed loops.
- In a ring network, every device has exactly two neighbors for communication purposes.
- All messages travel through a ring in the same direction (effectively either "clockwise" or "counterclockwise").
- A failure in any cable or device breaks the loop and can take down the entire network.

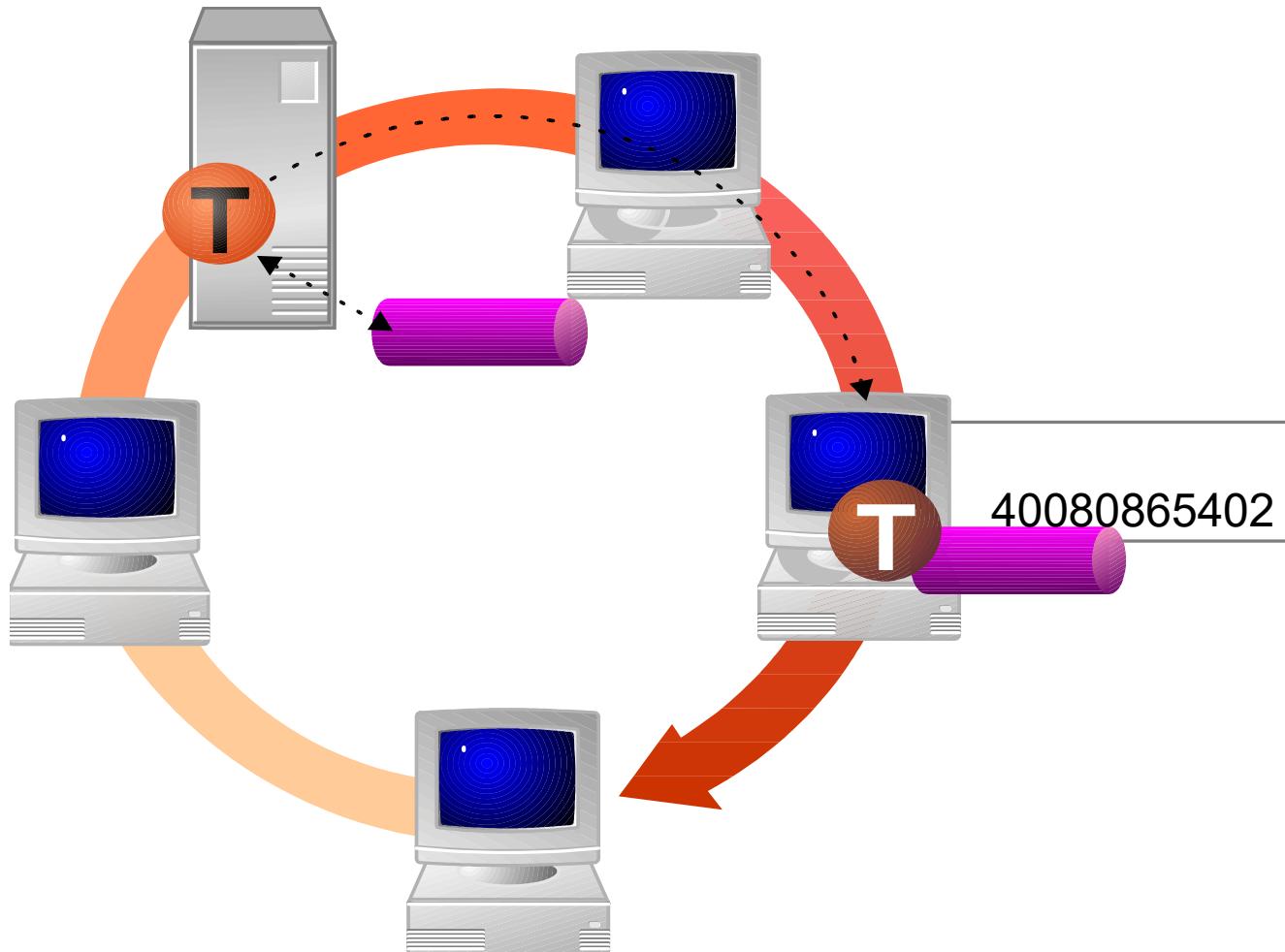
Ring TOPOLOGY



Simple Ring Network Showing Logical Ring



A computer grabs the token and passes it around the ring



Advantages and Disadvantages of Ring Topology

- No computers can monopolize the network because every computer is given equal access.
- The network traffic is in a single direction.
- If one computer fails the entire network is down.
- It is difficult to troubleshoot and also adding or removing the computers disrupts the network.
- Network reconfiguration is difficult.
- Difficult to diagnose faults.
- Topology affects the access protocol.

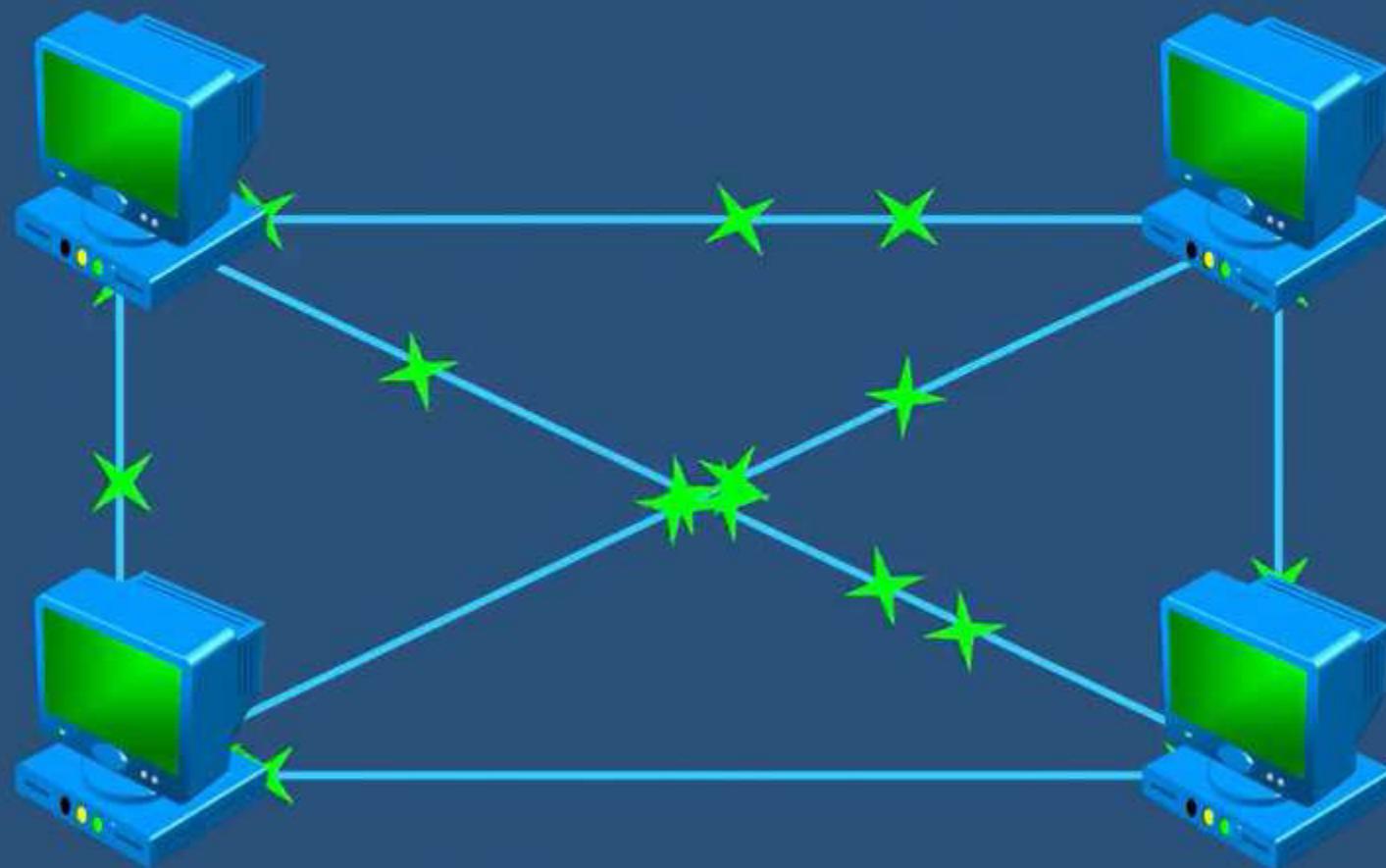
Mesh Topology.

Mesh topologies involve the concept of *routes*. Unlike each of the previous topologies, messages sent on a mesh network can take any of several possible paths from source to destination.

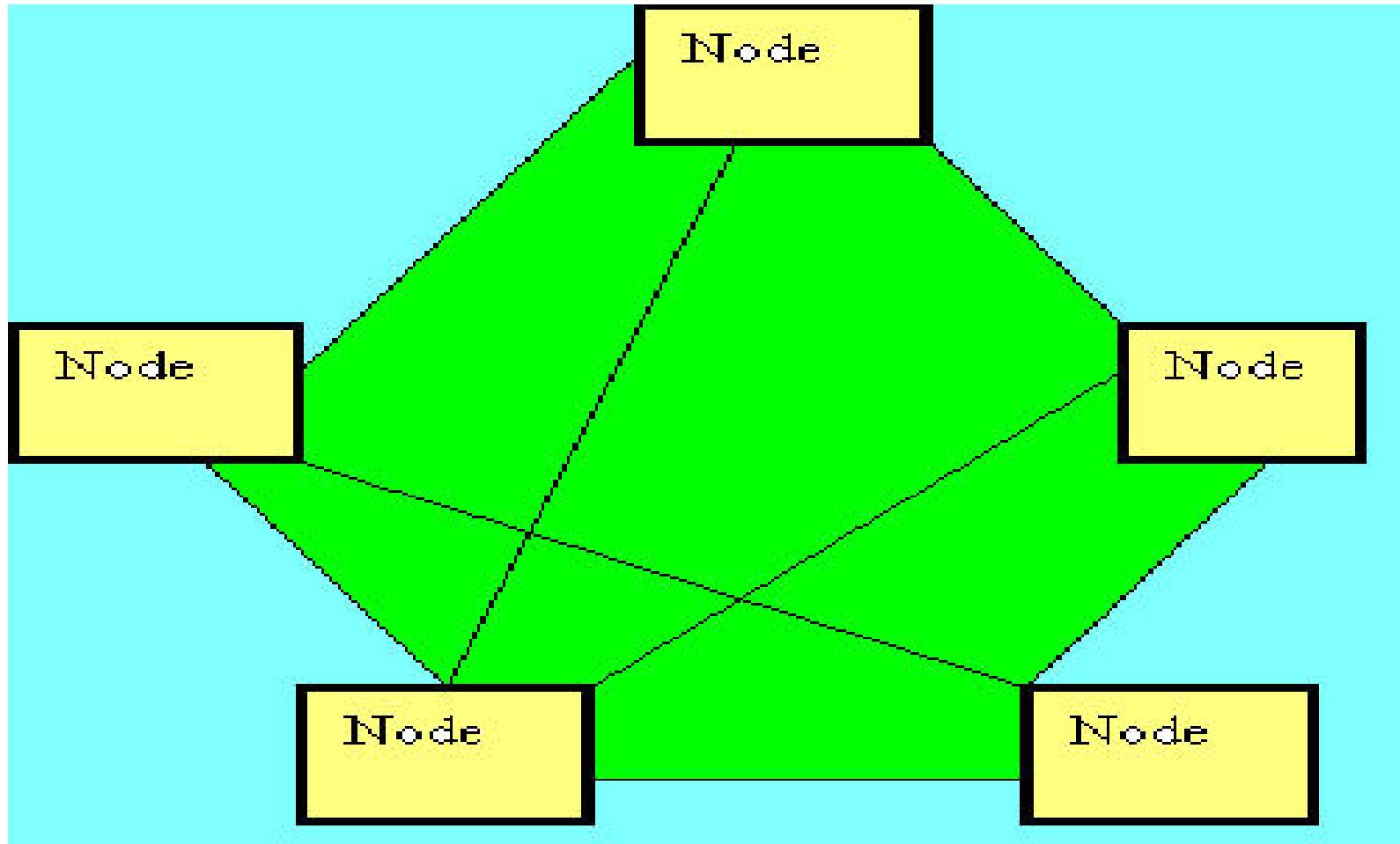
(Recall that in a ring, although two cable paths exist, messages can only travel in one direction.)

Some WANs, like the Internet, employ mesh routing..
Fig. Mesh Topology.

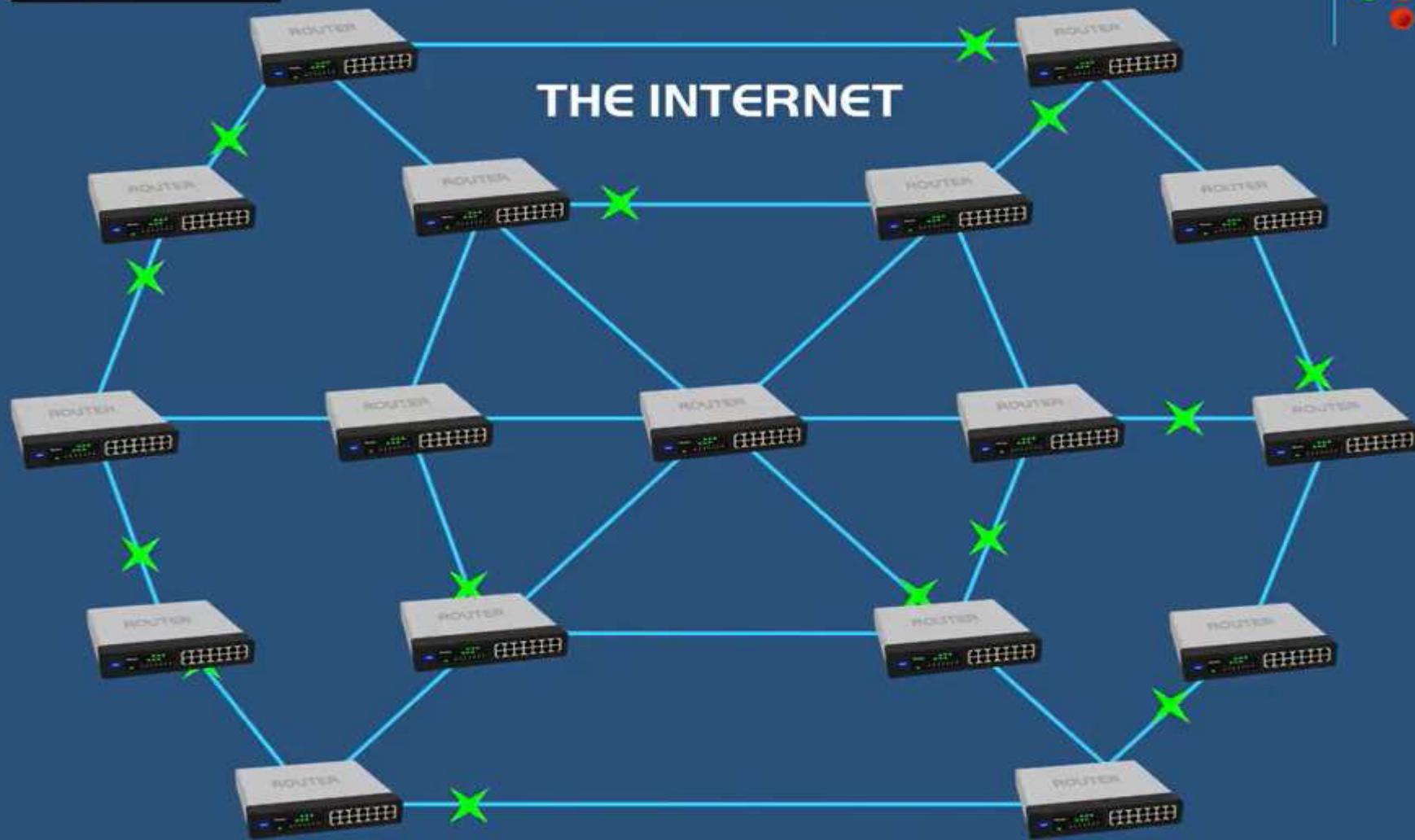
MESH TOPOLOGY



Mesh Topology.



Mesh TOPOLOGY



Tree Topology :

Tree topologies integrate multiple star topologies together onto a bus. In its simplest form, only hub devices connect directly to the tree bus, and each hub functions as the "root" of a tree of devices. This bus/star hybrid approach supports future expandability of the network much better than a bus (limited in the number of devices due to the broadcast traffic it generates) or a star (limited by the number of hub ports) alone.

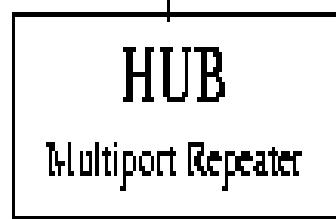
Example of tree topology can be seen in the cable TV. The main cable from the main office is divided into main branches with each branch divided into smaller branches and so on. The hubs are used when a cable is divided.

Level 1

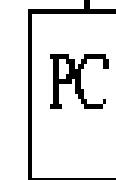
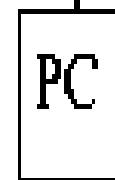
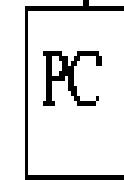
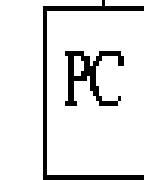
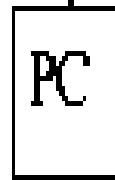
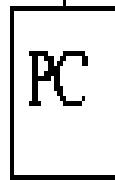


Maximum of 4 repeaters
between any two PCs

Level 2



Level 3



Wireless Topology

Infrastructure TOPOLOGY



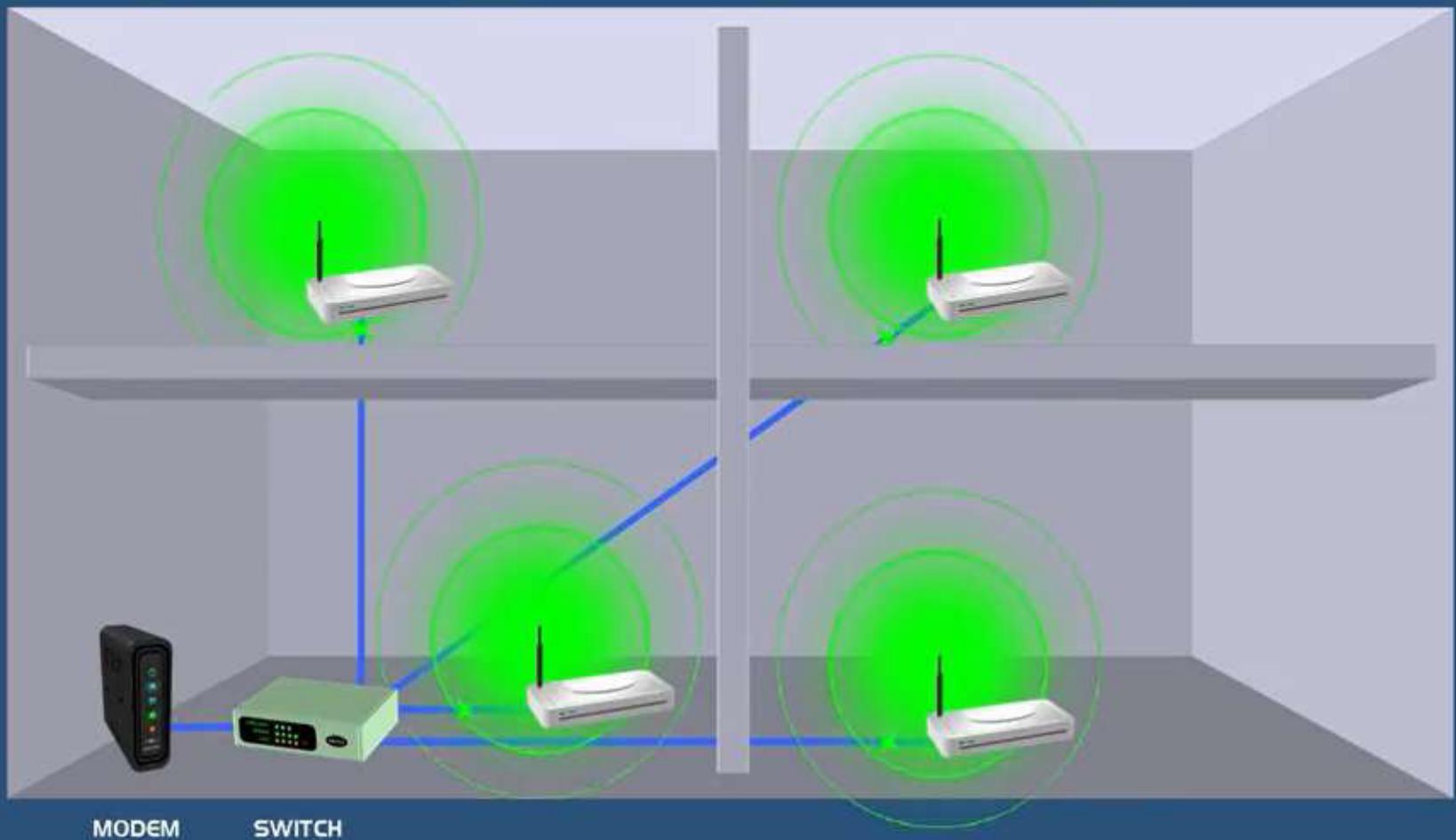
Ad hoc TOPOLOGY



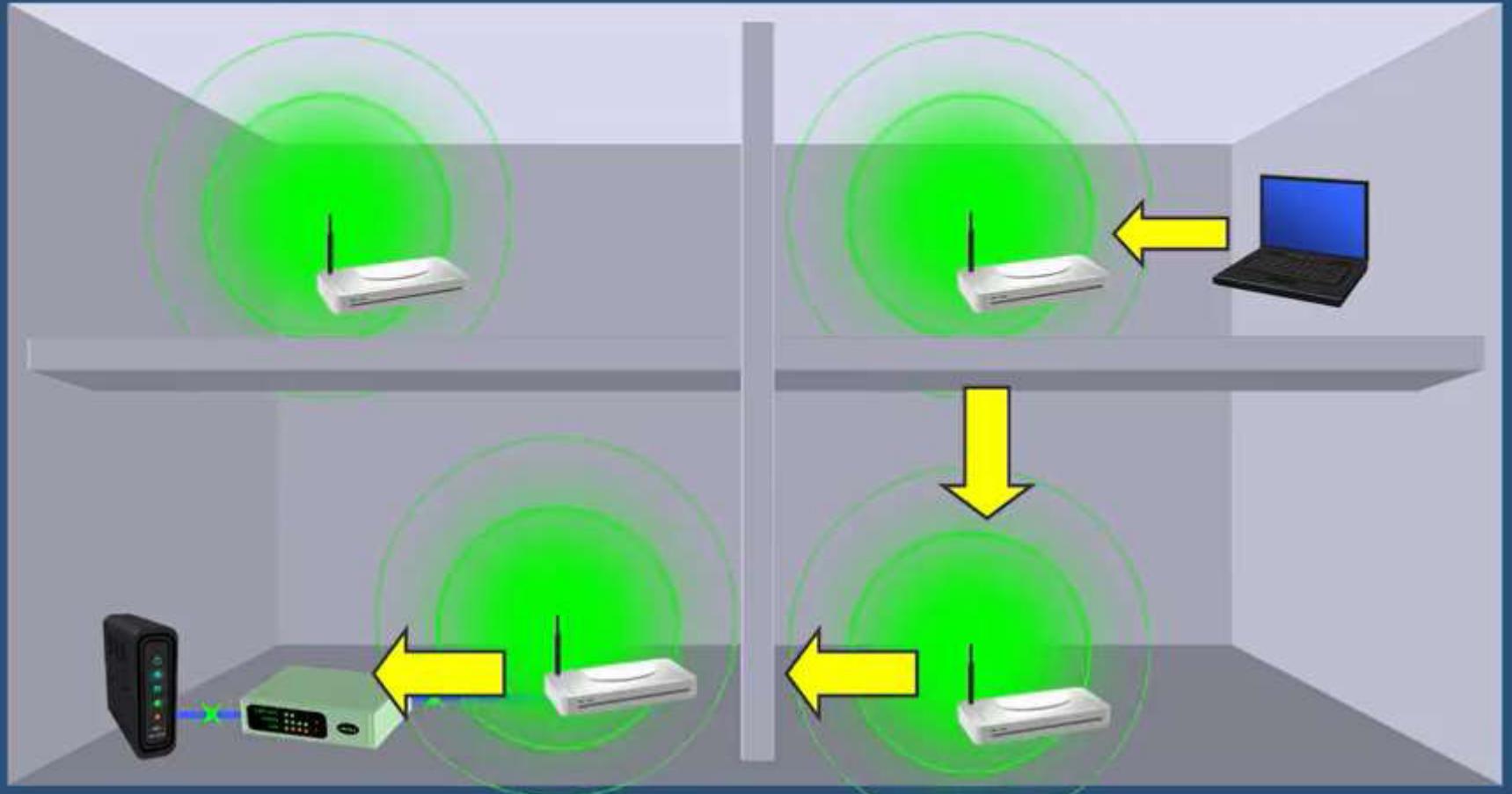
~~CABLES~~ ~~SERVERS~~
~~ROUTERS~~ ~~WAPs~~



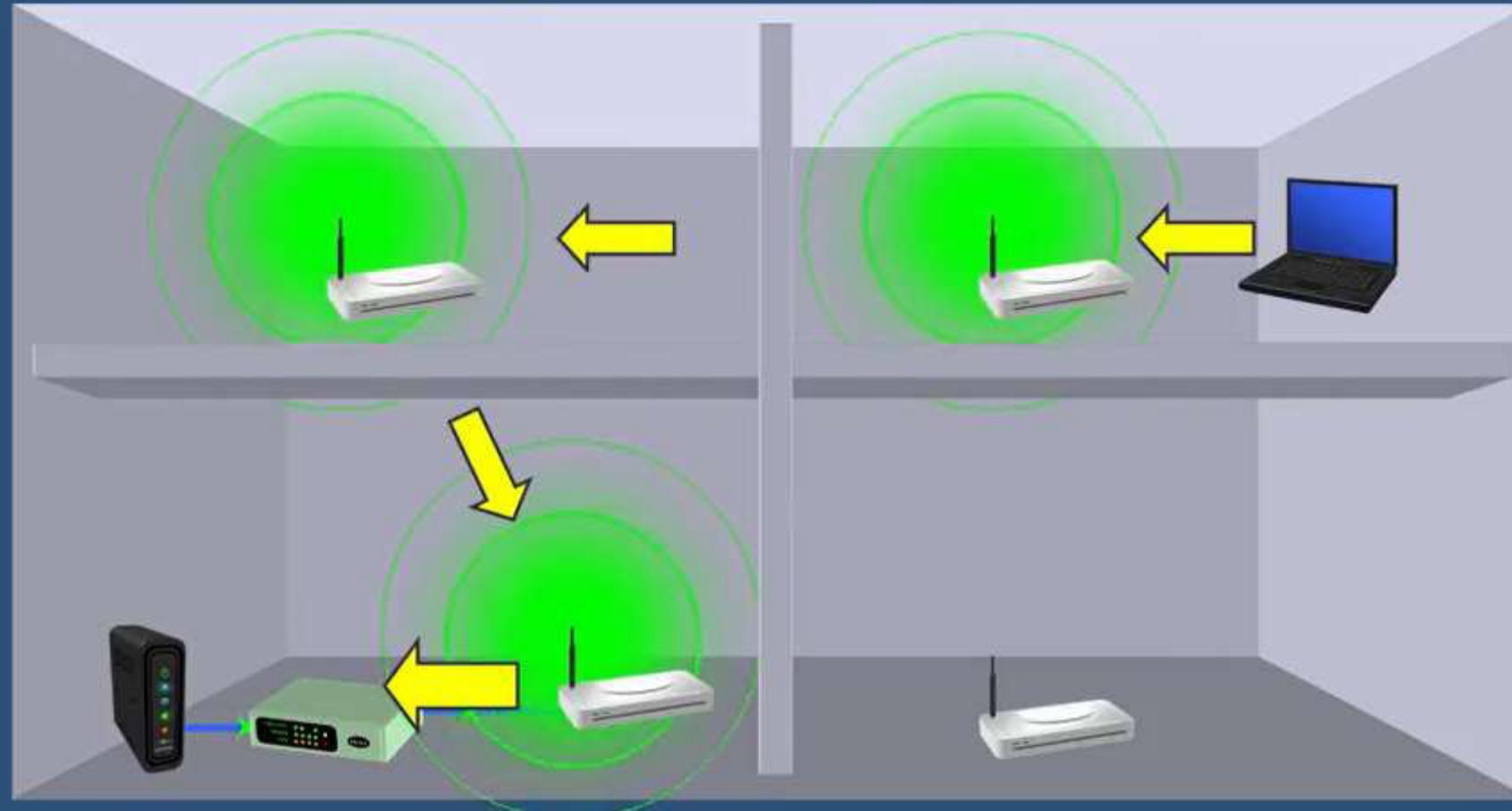
Wireless Mesh TOPOLOGY



Wireless Mesh TOPOLOGY



Wireless Mesh TOPOLOGY



Protocol :

It is defined as a set of rules and regulations used for communication.

The key elements are given below.

1. Syntax: Refers to the structure of data, meaning the order in which they are presented.

2. Semantics: The meaning of each section of bits.

3. Timing: Refers to two characteristics: - When data should be sent and how fast they can be sent.

PROTOCOLS AND STANDARDS:

This standard defines structured cabling, a telecommunication cabling system that can support virtually any voice, imaging or data applications that an end user chooses.

- Electronic Industries Association (EIA),
- Telecommunications Industry Association (TIA) and other leading telecommunication companies worked cooperatively to create ANSI/TIA/EIA-568-A standard for commercial buildings.

Standards:

Something established for use as a rule or basis of comparison in measuring or judging capacity, quantity, content, extent, value, quality, etc.

Definitions

- Rules and conventions for the exchange of information
 - Open Systems
- Who makes the rules and conventions?
 - Many local, regional, and international organizations
 - ISO, ITU, IEEE, ANSI, ECMA
- **Open Systems Interconnection Standards (OSI)**
 - Packet Switched Public Data Network (PSPDN)
 - Circuit Switched Public Data Network (CSPDN)
 - Public Switched Telephone Network (PSTN)
 - Integrated Services Digital Network (ISDN)
 - Local Area Network (LAN)
- V-series
 - Connecting equipment to a Public Switched Telephone Network (PSTN)
- X-series
 - Connecting equipment to a Public Switched Data Network (PSDN)
- I-series
 - Connecting equipment to an Integrated Services Digital Network (ISDN)

Standards Organizations

- ITU - **International Telecommunication Union** which develops worldwide standards for telecommunication technologies.
- CCITT - **Consultative Committee for International Telegraph and Telephone**. Responsible for development of Communication standards.
- IEEE - **Institute of Electrical and Electronic Engineers**.
- ISO - **International Standardization Organization**. Responsible for a wide range of standards including networking standards.

Popular Protocols

- **TCP/IP** - Transmission Control Protocol/Internet Protocol. Name of suite of protocols to support the implementation of worldwide internet works.
- **X.25** - ITU's standard that defines how connections between terminal equipment and computers are maintained.
- **SMDS** - Switched Multi-megabit Data Service. High speed packet switched WAN networking technology offered by phone companies.

Popular Protocols (Cont.)

- ISDN - Integrated Services Digital Network. Communication protocol offered by phone companies which allows phone networks to carry voice, video, and data.
- CDPD - Cellular Digital Packet Data. Standard for 2-way wireless data communication over high frequency cellular phone channels.
- DQDB - Distributed Queue Dual Bus. Data link layer protocol designed for metropolitan area networks.
- CDMA - Code Division Multiple Access.

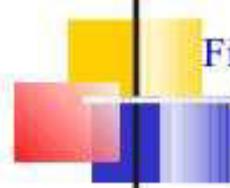
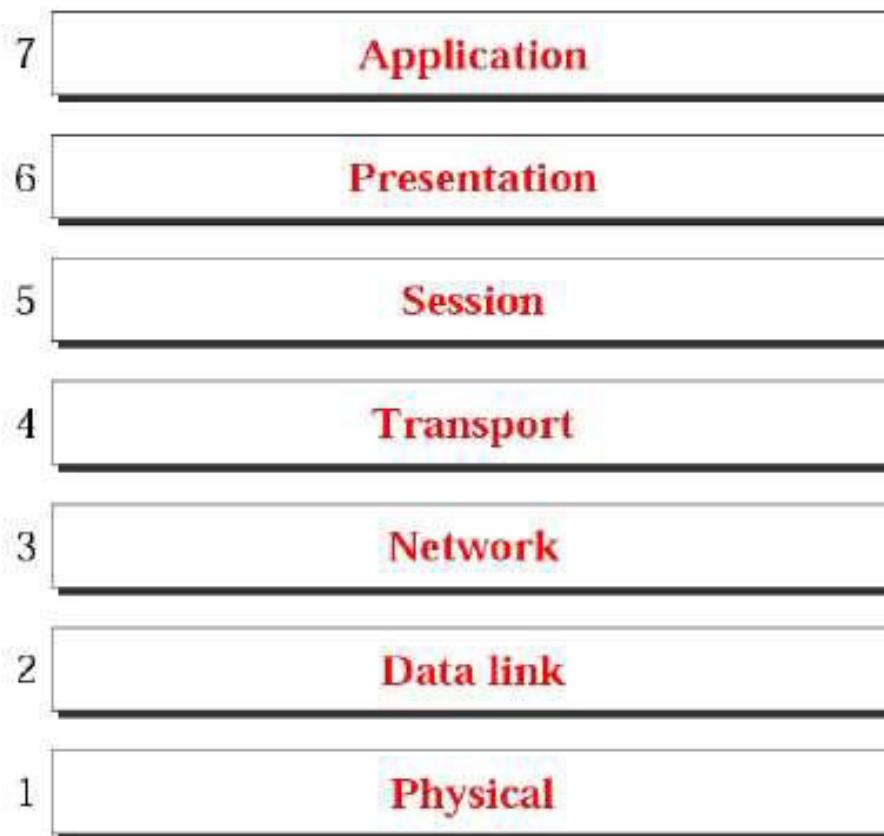
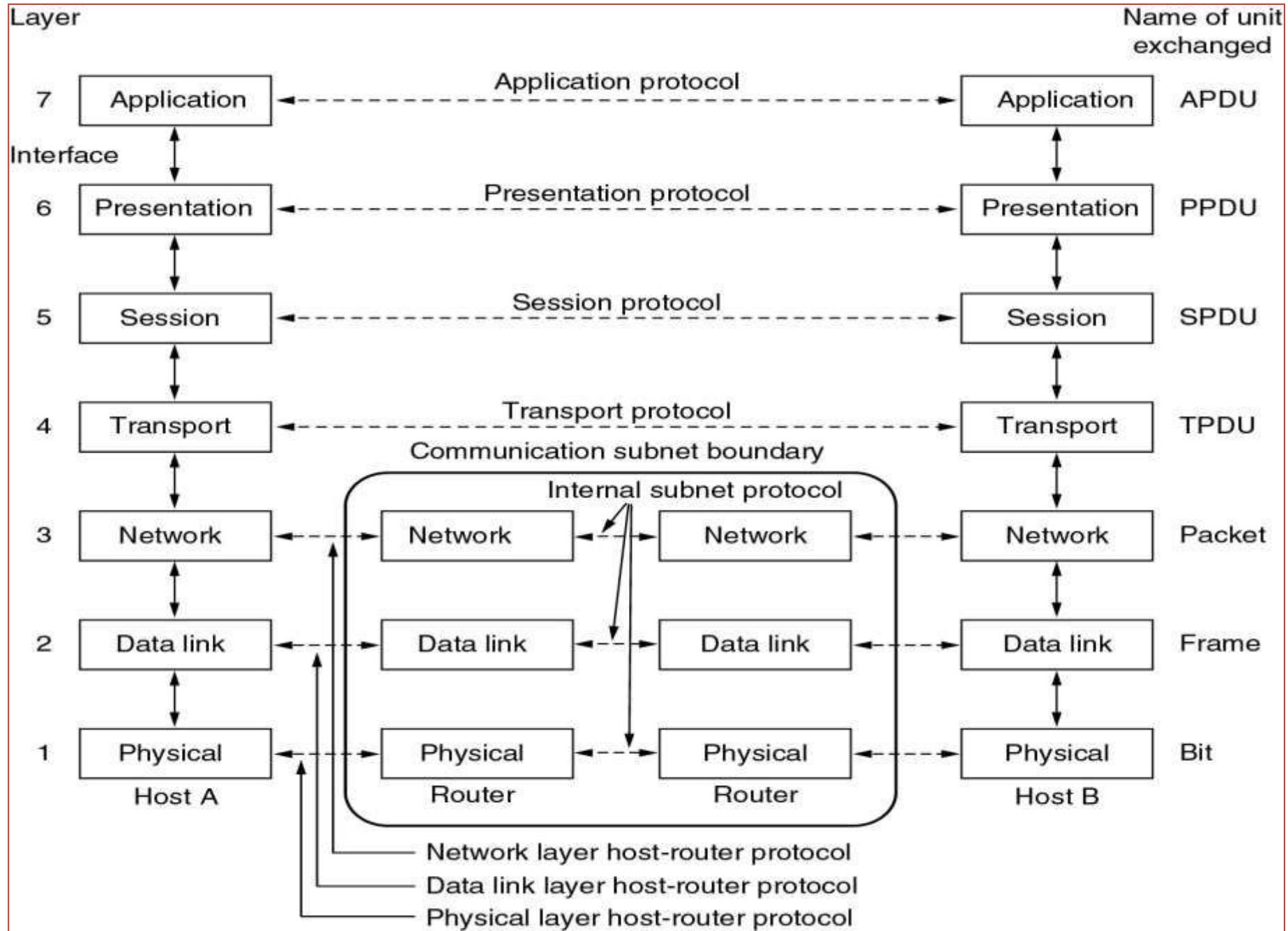


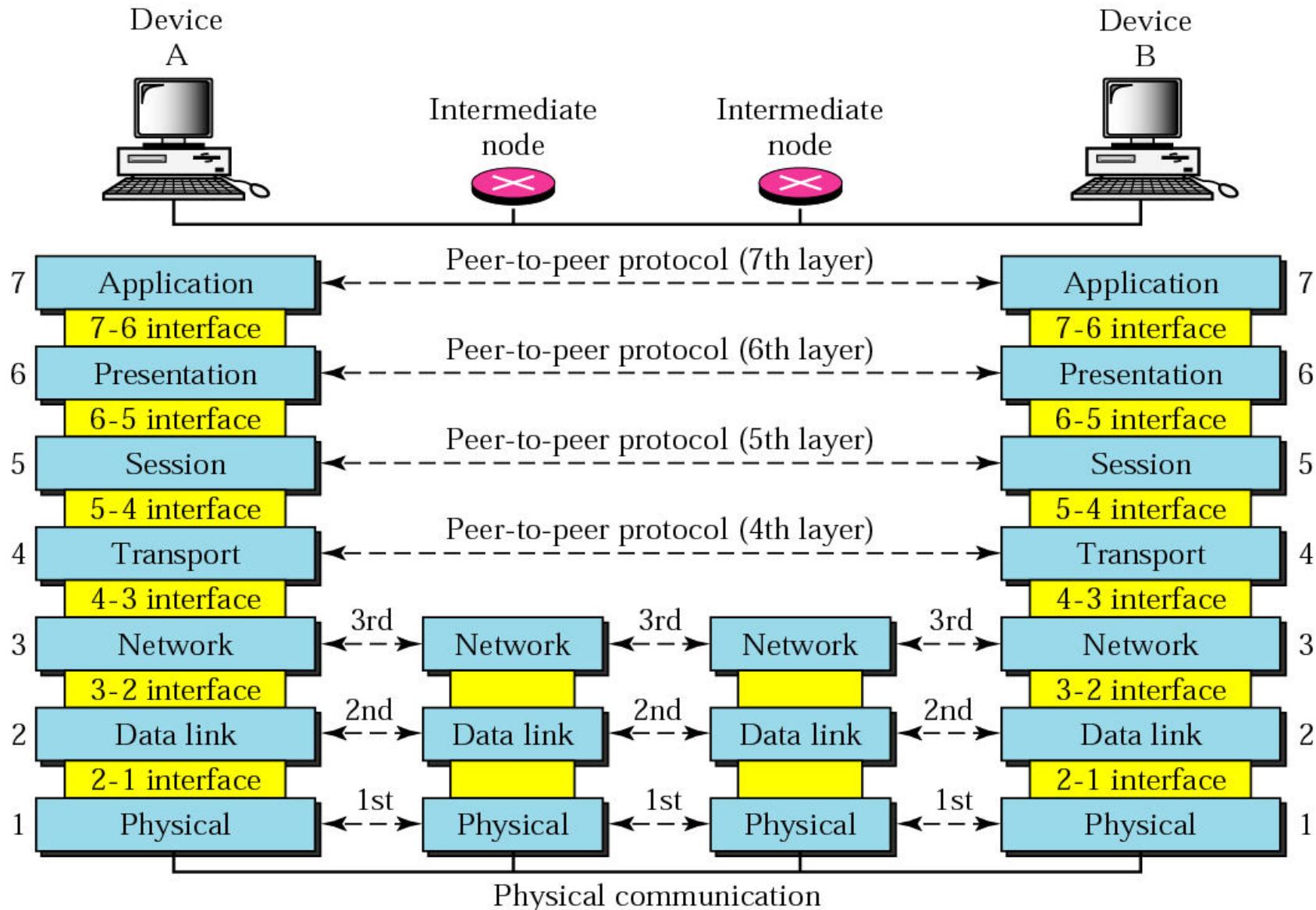
Figure 2.1 *The OSI model*



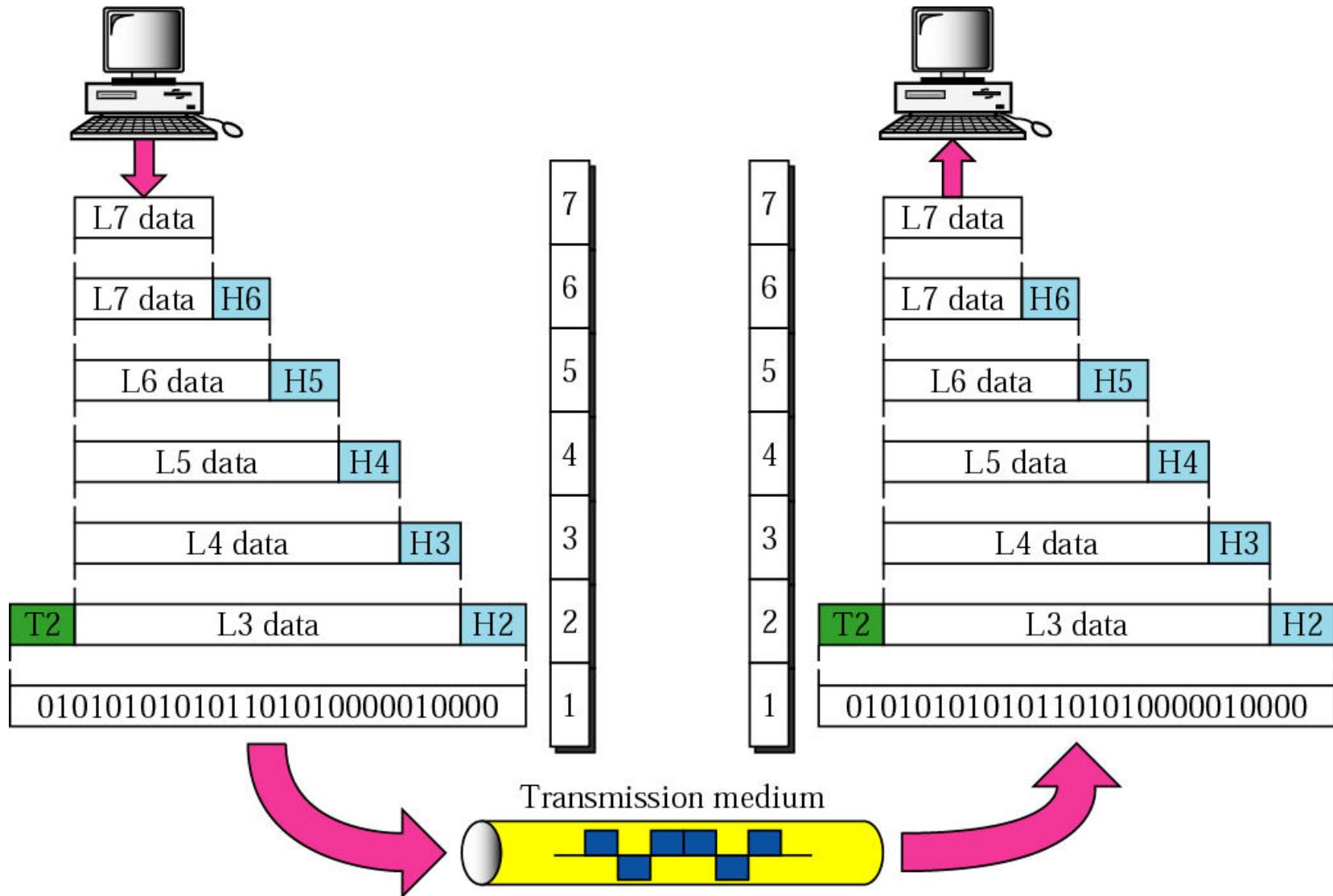
OSI Reference Model



OSI layers



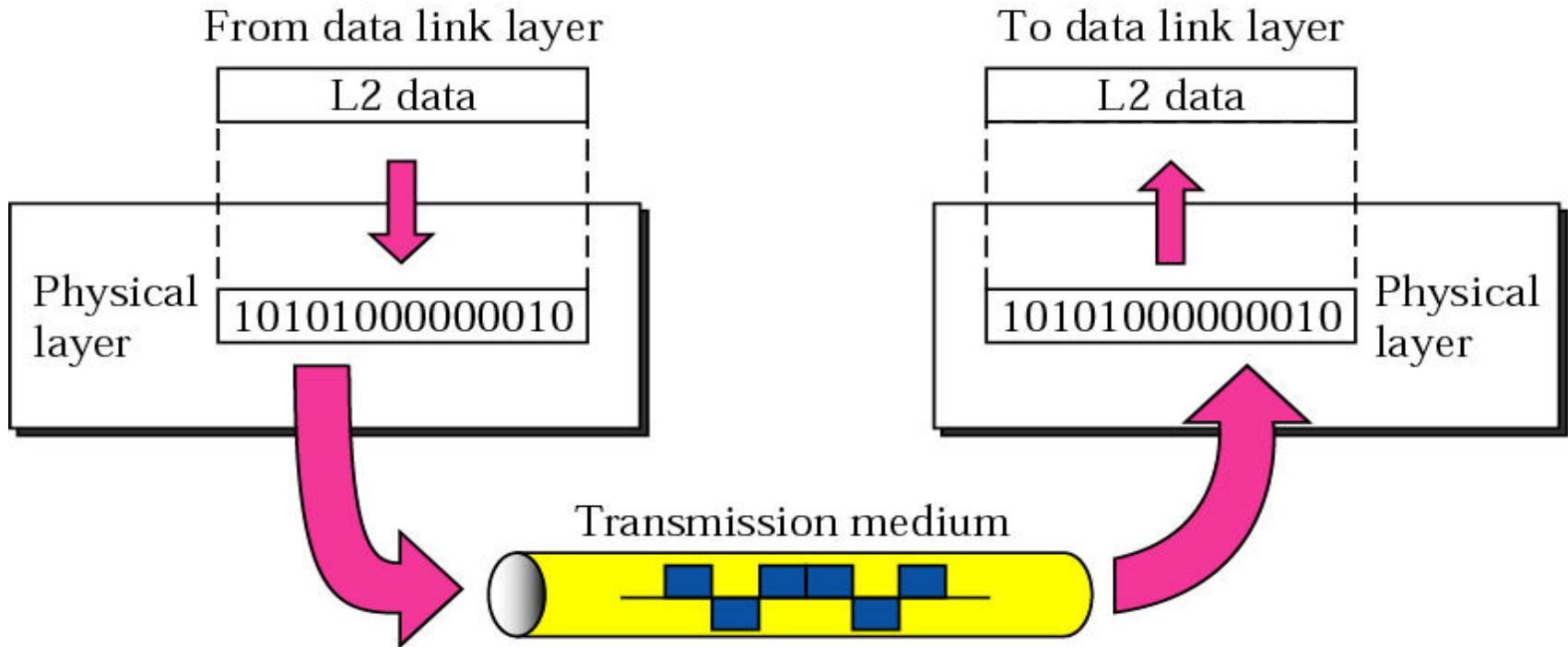
An exchange using the OSI model



Physical Layer

- The Physical layer is concerned with sending raw **bits** between adjacent nodes across the medium..
- The bits sent as 0's and 1's will be received as 0's and 1's only.
- The Physical layer has to take care of the following factors.
 - **Signal Encoding** : How are the bits 0 and 1 to be represented.
 - **Medium** : What is the medium used, and are its properties.
 - **Signal type** : Are analog signals used or digital.
 - **Bandwidth** : Which of base band or broadband communication used.
 - Whether the transmission is serial or parallel
 - What is the topology used

Physical Layer



Physical Layer

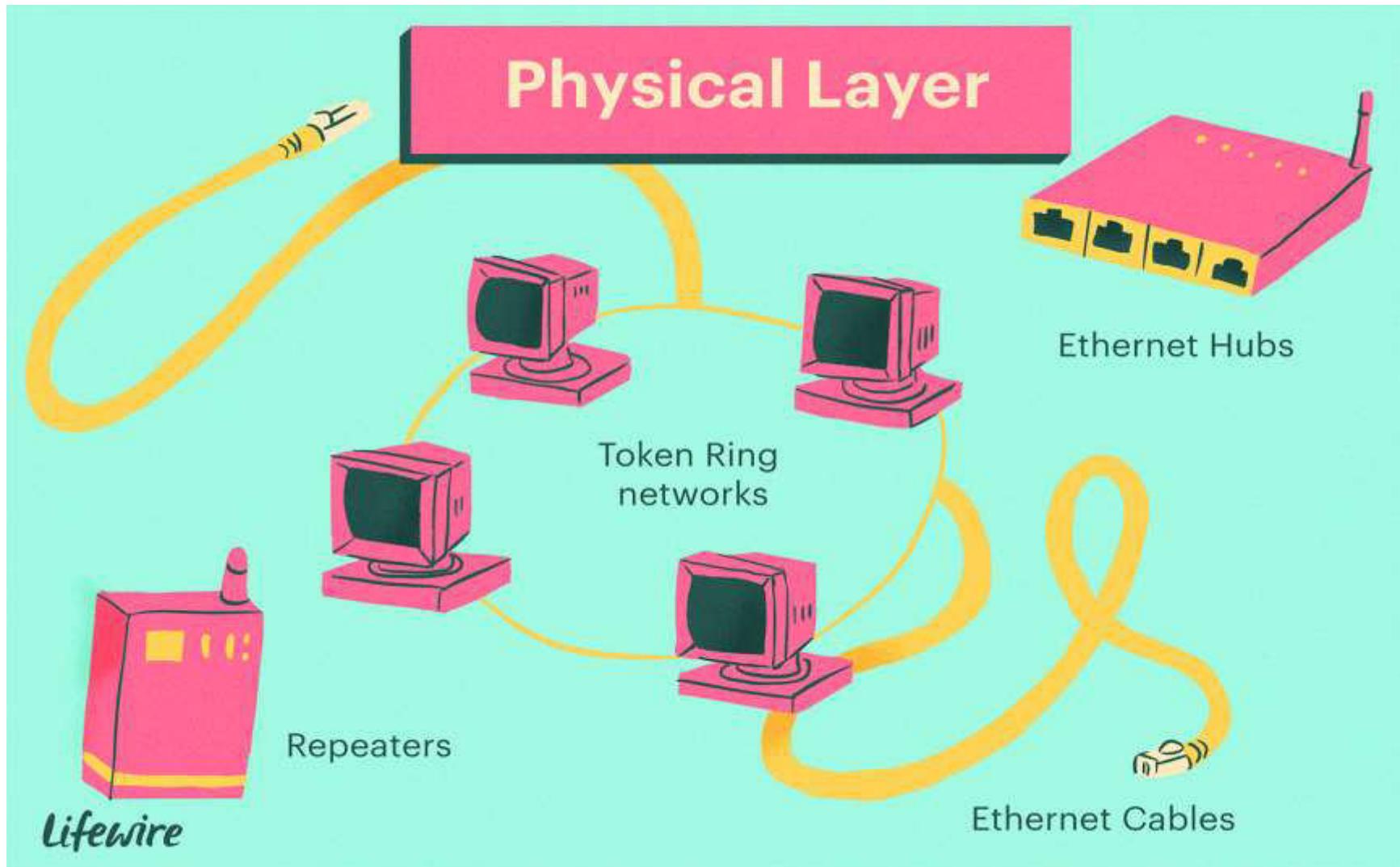
The Physical Layer



Physical Layer

- Deals with mechanical, electrical and procedural interfacing
- Provides collision detection
- Specifies cables, connectors, and other components
- Transmits raw information over communication channel
- Establishes, maintains, and disconnects physical links
- Includes software device drivers for communication interfaces

Physical Layer

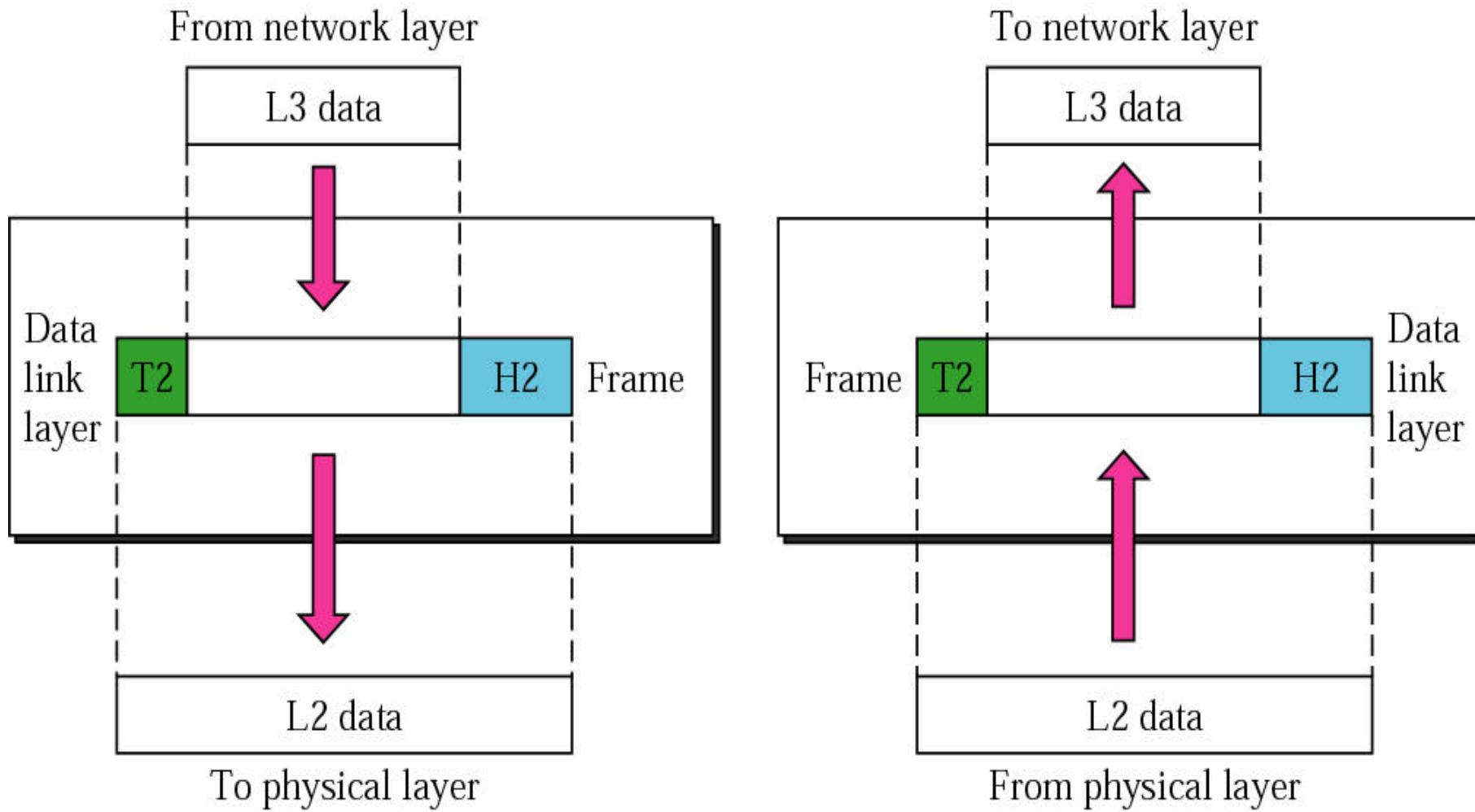


Lifewire

DATA LINK LAYER

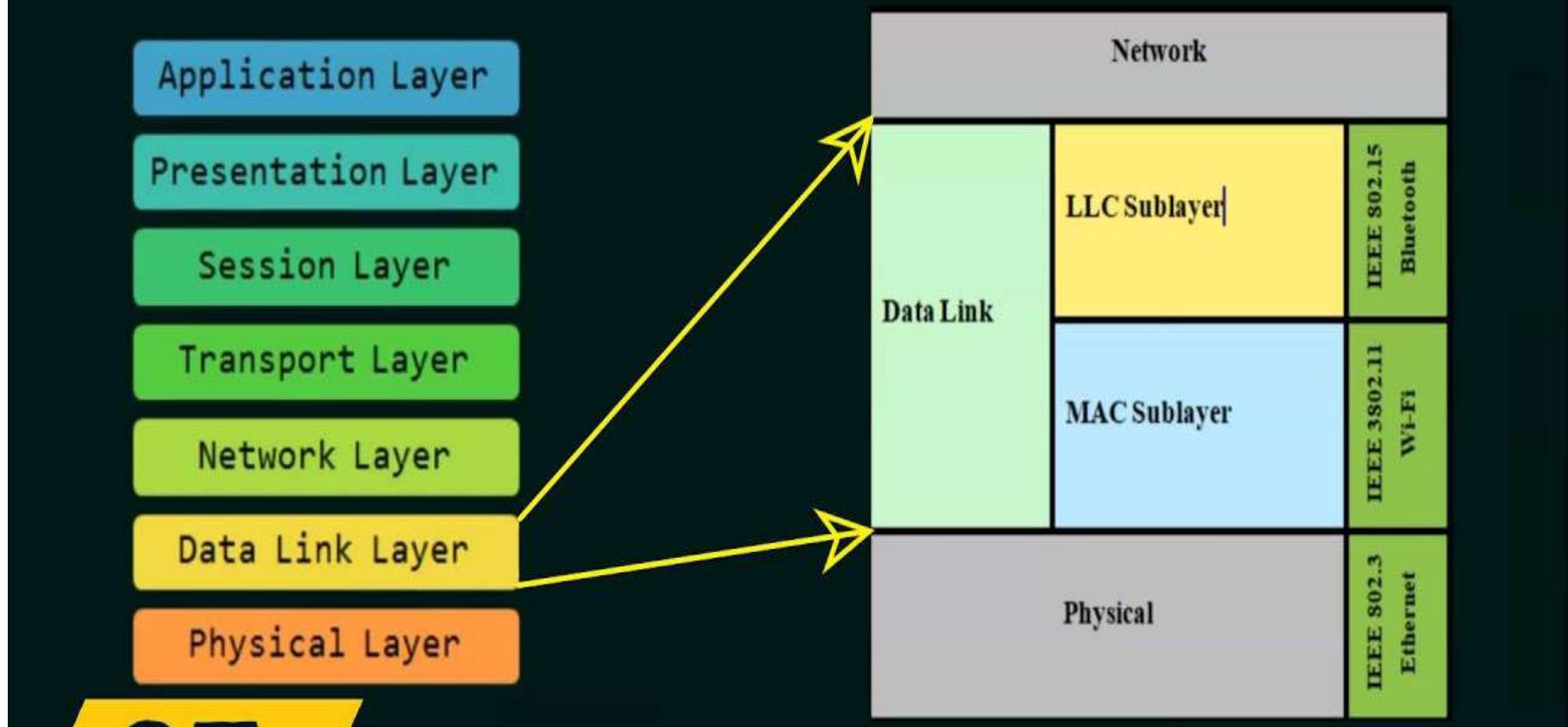
- It is responsible for transmitting a group of bits between the adjacent nodes called **Frames**.
- It has to construct the frame after receives the bits from the physical layer.
- The DLL has to check the CRC to ensure the correctness of the frame. If incorrect, it asks for retransmission.
- If the receiver is slow, then the transmitter has to make some agreement with the receiver to ensure correct delivery.
- Discarding of duplicate frames will be done at the receiving end.
- Retransmission will be done at the sending end only when necessary.
 - If the Timer has Elapsed
 - No Acknowledgement within specified time.
 - if the ACK. Is damaged.
- DLL is divided into two sub layers
 - **Logical Link Control**
 - **Medium Access Control**
- Headers and trailers are added , containing the physical addresses of the adjacent nodes.
- DLL has to handle the error detection and correction and deliver the undamaged frame **only at the network level.**

Data Link Layer



Data Link Layer

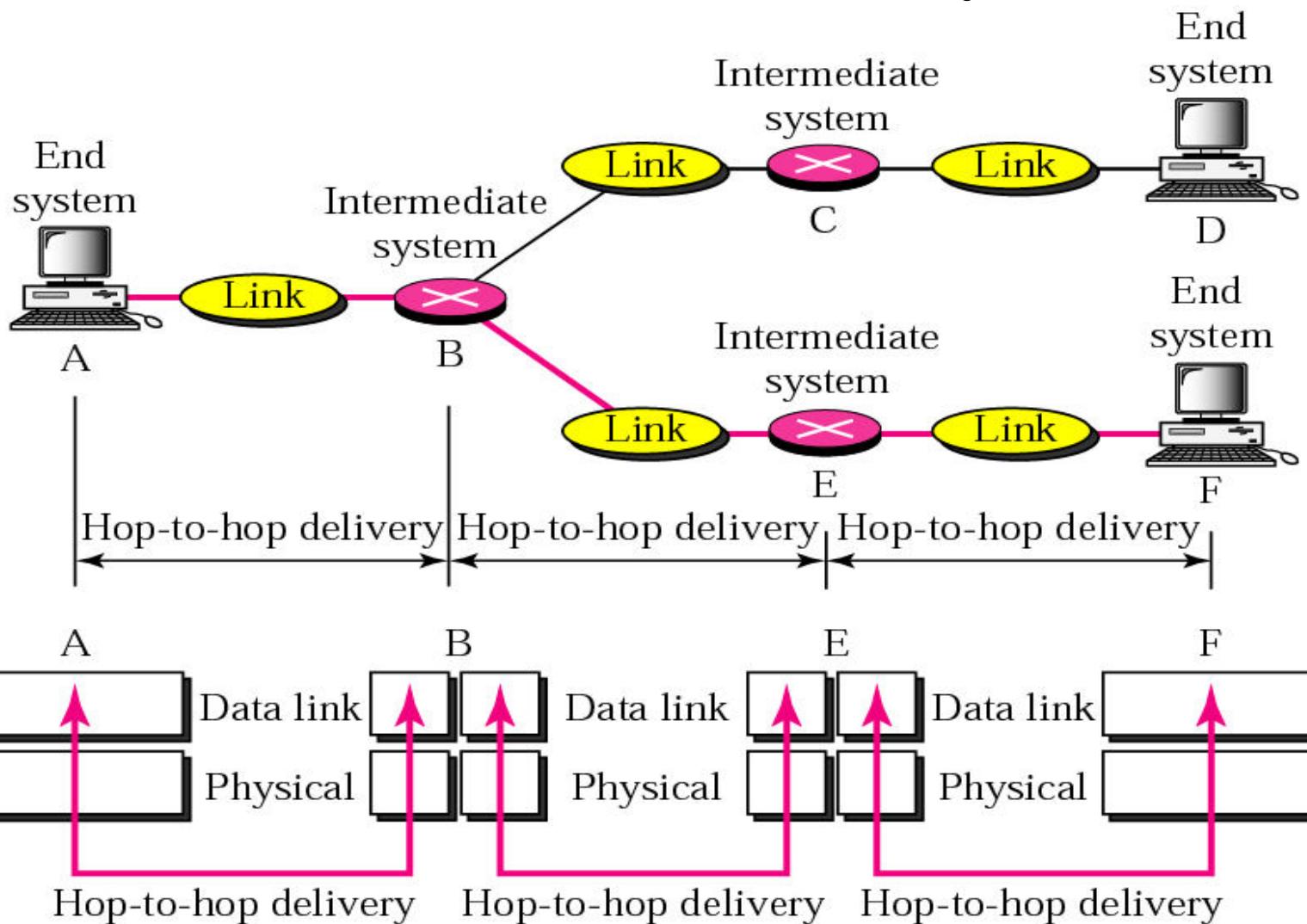
Data Link Layer (Sub-layers)



Data Link Layer

- Provides reliable transfer of data.
- Breaks data (packets) into frames.
- Adds bits for error detection/correction.
- Manages access to and use of the channel.
- Solve problems caused by lost, damaged, and duplicate frames.
- Sends acknowledgments.
- Adds flags to indicate beginning and end of message.
- Connectionless or connection oriented services.
- IEEE MAC and LLC support.

Node-to-node delivery



NETWORK LAYER

Network Layer

Formats the data into packets to be delivered up to the Transport layer

Or updates the destination address and pushes the frame back down to the lower layers.

IP Addresses

Lifewire

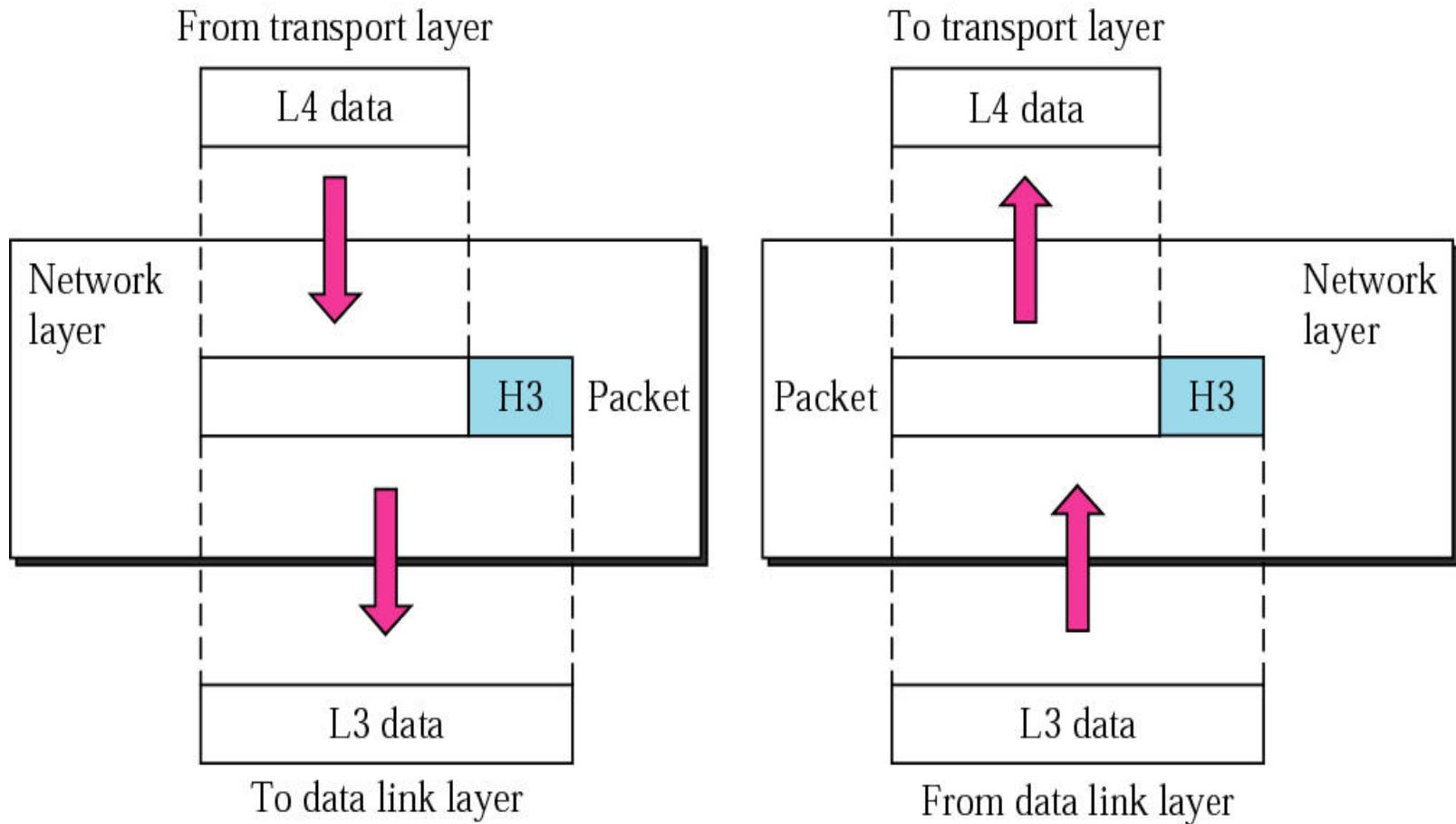
NETWORK LAYER

- It is responsible for routing a packet within the subnet.
i.e) from the source node to the destination node across multiple nodes in the same network or across multiple networks.
- It is also responsible for tackling the **congestion problem** at a node, when there are too many packets stored at a node to be forwarded to the next node.
- It has to take care of **controlling the flow of information**.
- Transmission of messages will be done by **Packets**.
- It offers two types of services .
 - **Virtual Circuit (Telephone)**
 - Complete route should be established before transmission begins as first phase
 - Data transfer happens after that as second phase
 - Call termination as third phase
 - Order of delivering of messages will be maintained

NETWORK LAYER – (Cont...)

- **Data Gram (Postal)**
 - It can follow any route it wants. Selection of route is decided on availability.
 - Order of delivering of messages cant maintained.
- When there is only one small network based on broadcast , this layer is will do minimum functionality.
- It takes care of interpreting the logical addresses to physical addresses.

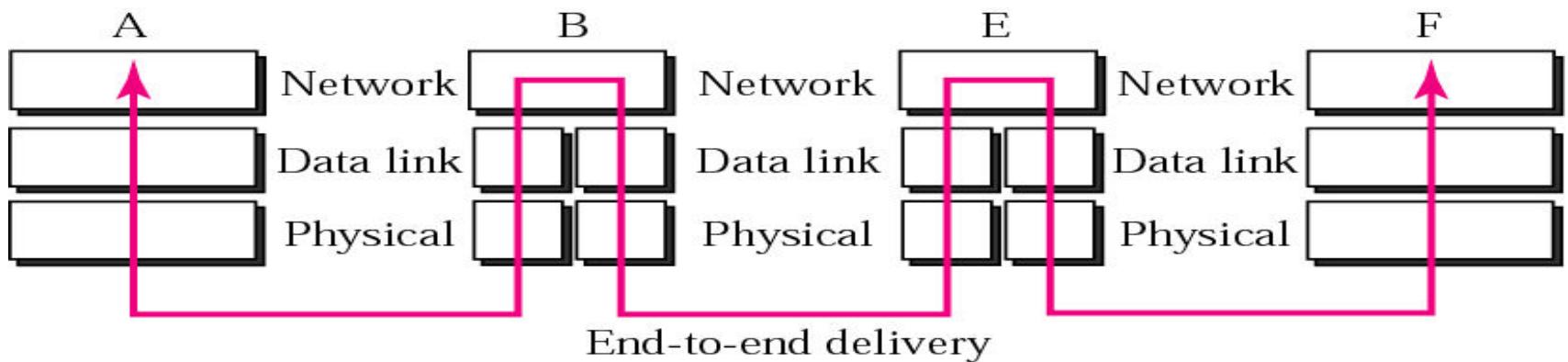
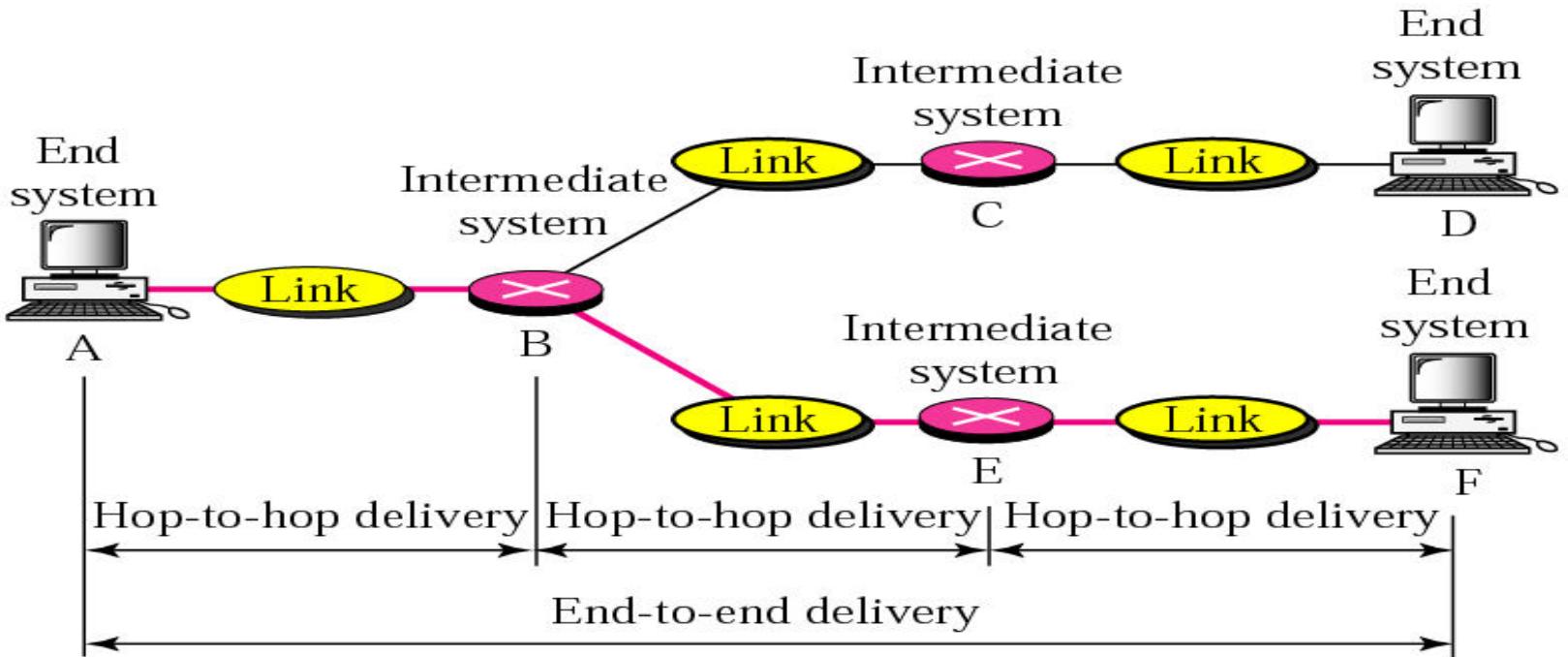
Network Layer



Network Layer

- Establishes, maintains and terminates connections
- Determines how packets are routed
- Divides transport messages into packets and reassembles them
- Performs congestion control, flow control
- Provides virtual circuit or datagram services
- Recognizes message priorities
- Sends messages in proper order
- Handles internetworking

End-to-end delivery



Transport Layer

Delivers data across network connections like TCP



Different transport protocols may support a range of optional capabilities including:



Error recovery



Flow Control

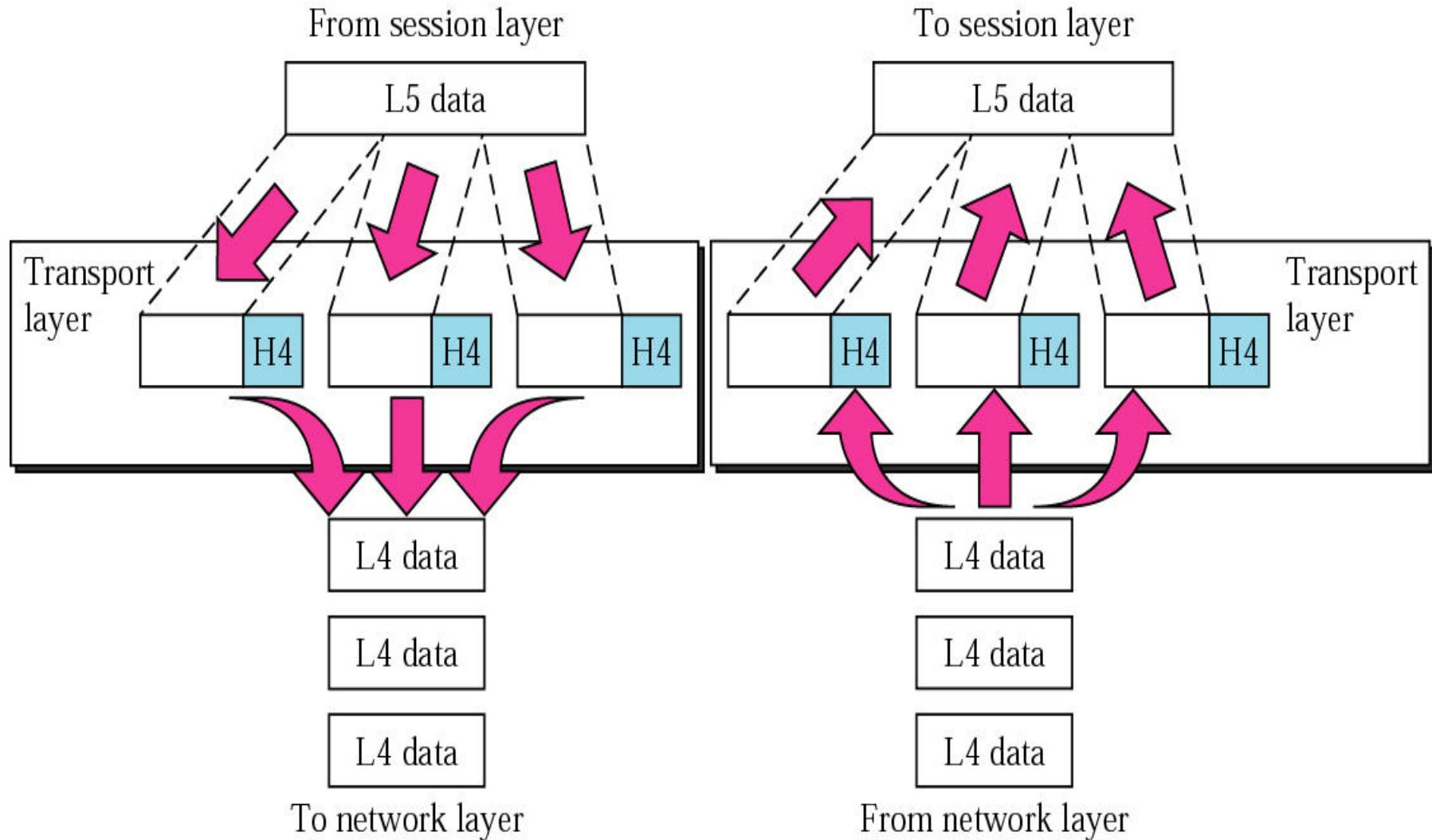


Support for re-transmission

TRANSPORT LAYER

- The Transport layer is the first end-to-end layer.
- The Transport layer breaks the messages in to number of packets, numbers them by adding sequence numbers at the source , and uses the same at the destination to reassemble the original message.
- A header at the transport layer contains information that helps to send the messages to the corresponding layer at the destination node, although the messages broken in to packets may travel through a number of intermediate nodes.
- It ensures end – to –end error free delivery to the hosts.
- The transport layer ensures that the complete message arrives at the receiver , and in the proper order.
- The Transport layer enables communication between two applications running on different computers.
- The Transport layer receives data from the session layer on the source computer which needs to be sent to the other computer.
- The Transport layer might create a logical connection between the source and the destination for the duration of the complete message transfer.
- It provides multiple transport for data flow.

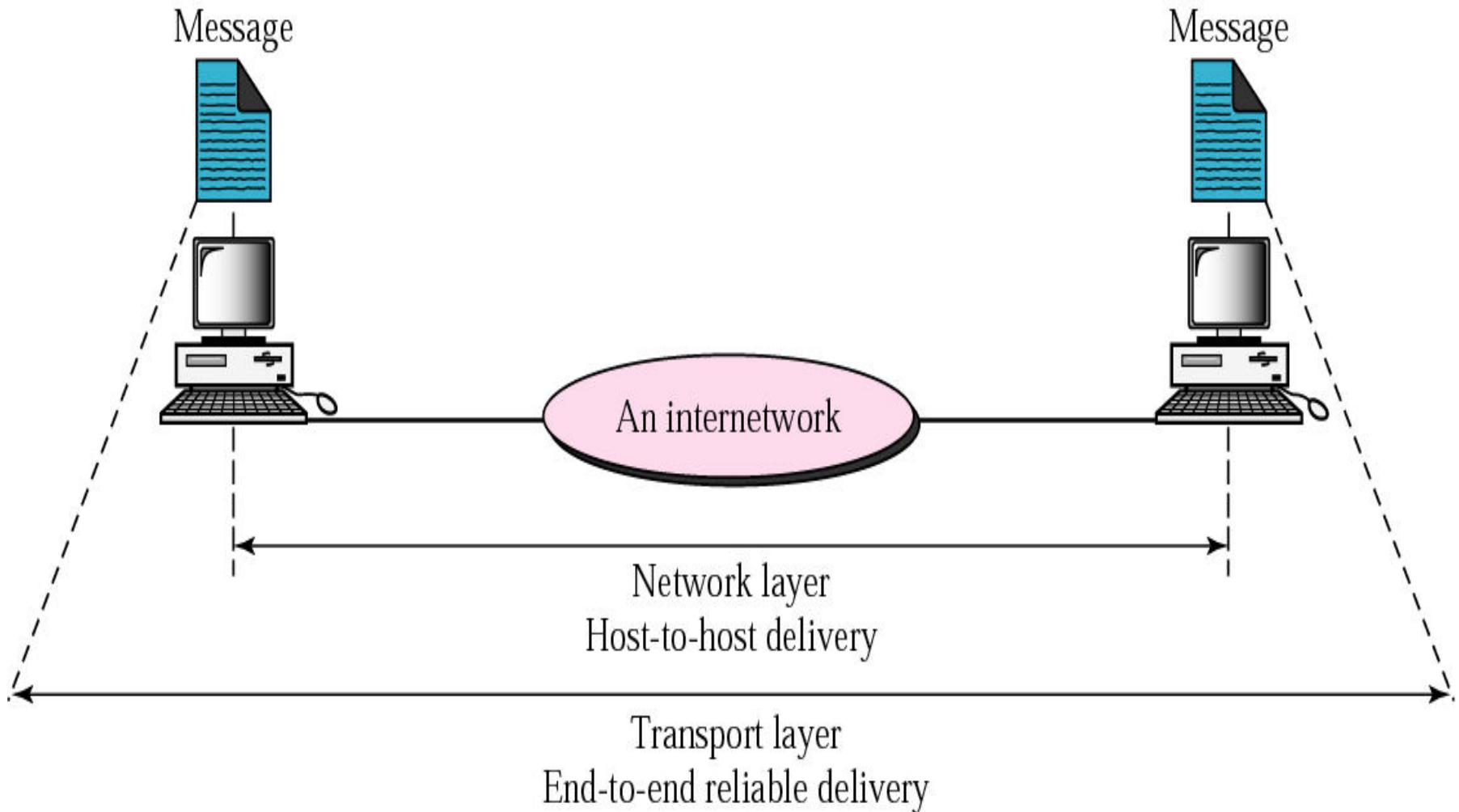
Transport Layer



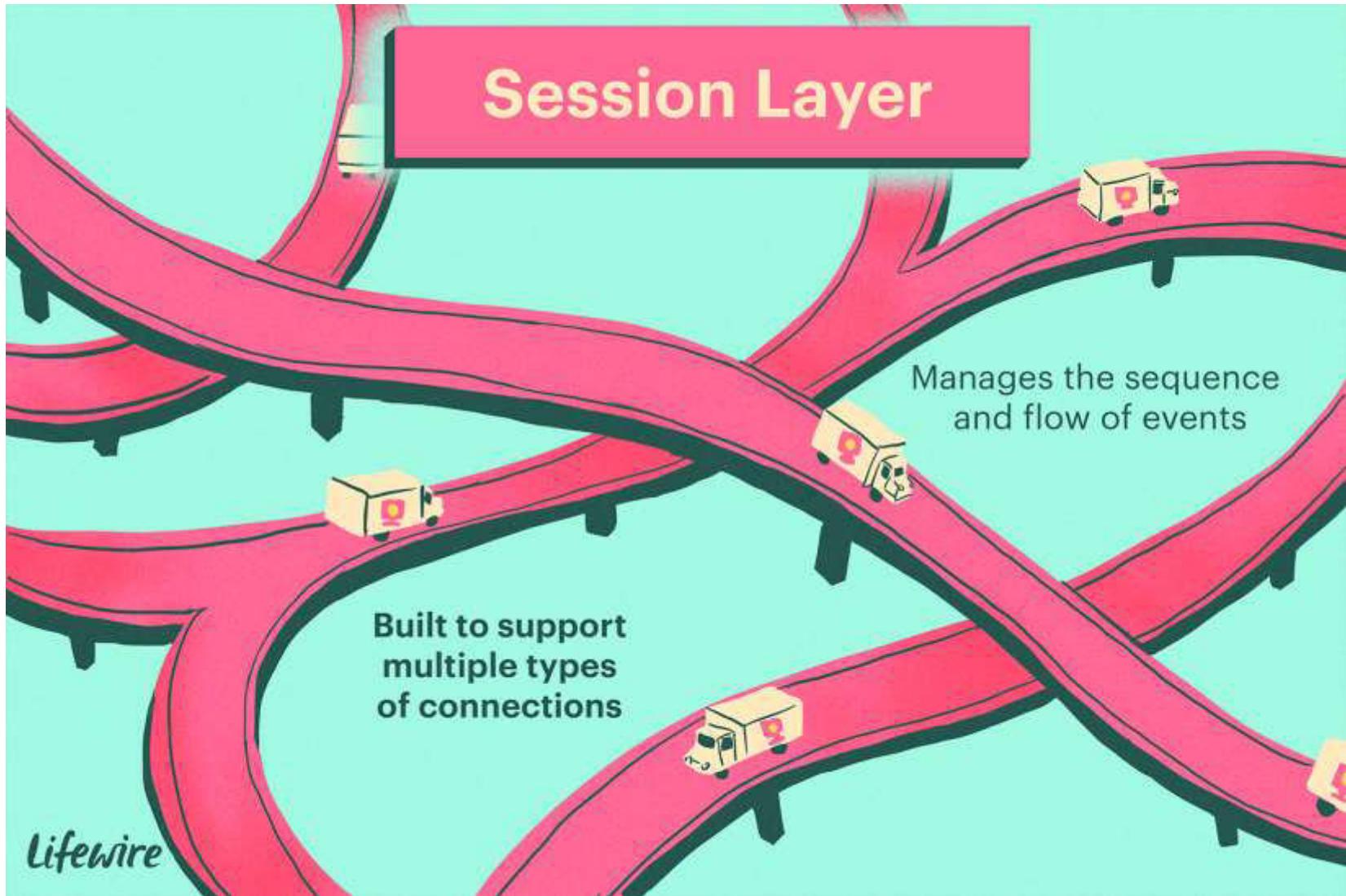
Transport Layer

- Establishes reliable end-to-end transport session (error detection and recovery), once path has been established.
- Fragmentation of message into packets (if not handled by layer 3).
- Multiplexing of several sessions from same source and all going to same destination.
- Creates distinct (Different) network connections.
- Monitors quality of service.
- Disassembles and assembles session messages.
- Flow control (if not done by layer 3).

Reliable end-to-end delivery of a message



SESSION LAYER

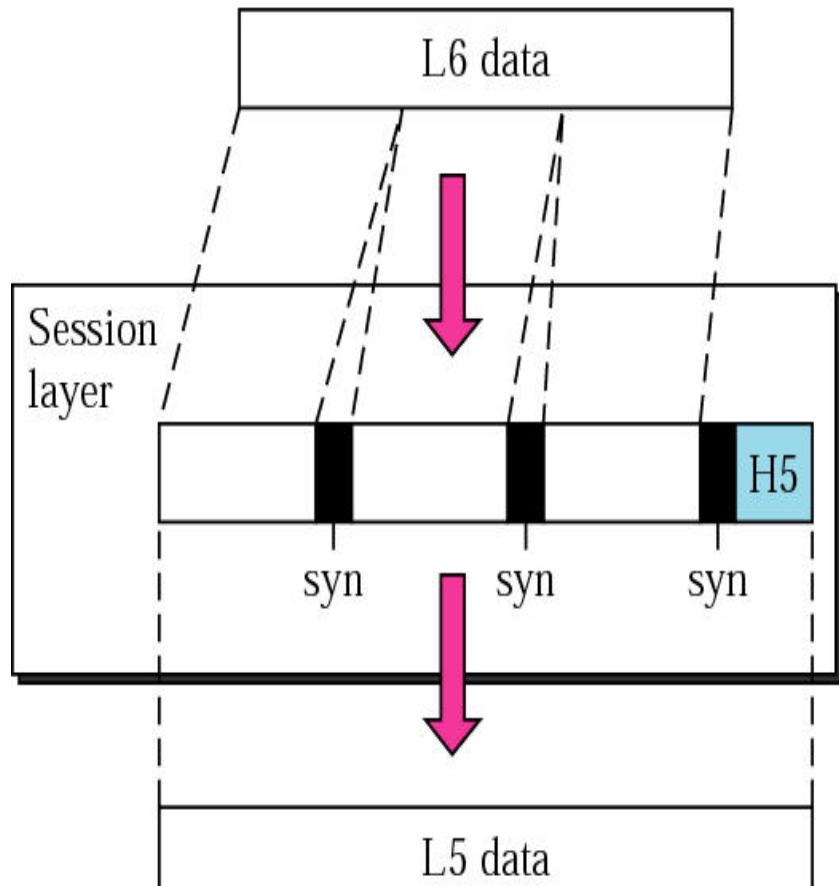


SESSION LAYER

- **The Session layer is to establish , maintain and synchronize the interaction between two communicating nodes.**
- A connection between two ends is called a session.
- It is responsible for Remote Login Process.
- It makes sure that a session once established is closed only after the successful completion.
- **It divides a session into sub sessions for avoiding the retransmission of entire messages by adding the checkpoint feature.**
- **It decides the order in which data needs to be passed to the transport layer.**
- **It also decides which user application sends data , and at what point of time, and whether the communication is simplex, half-duplex, or full duplex.**

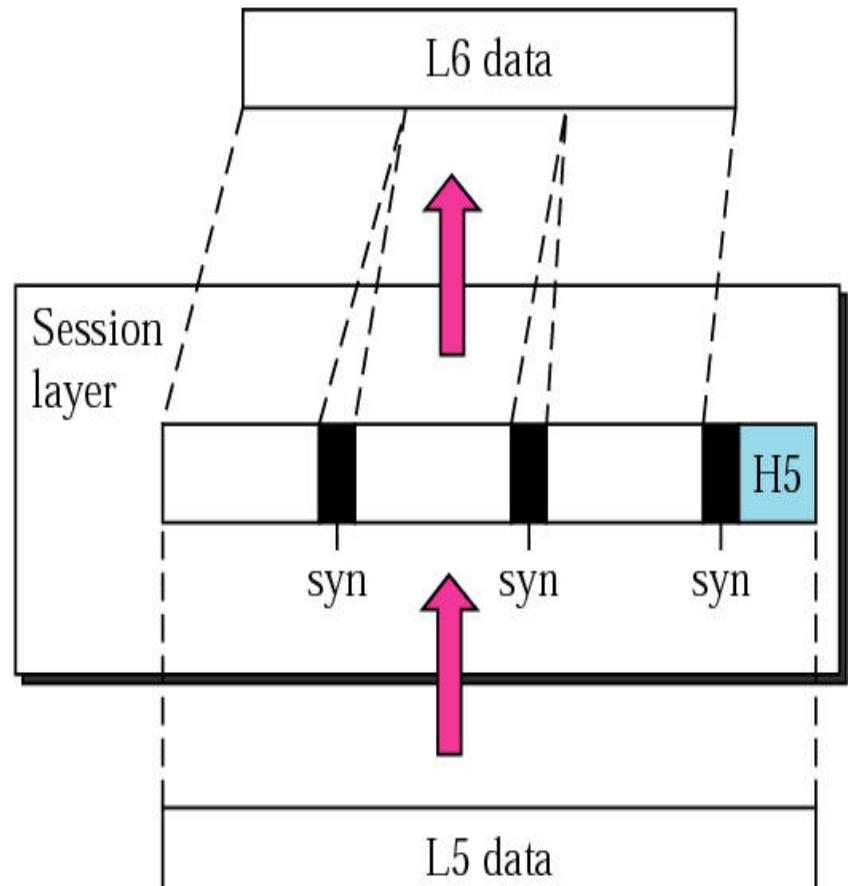
Session Layer

From presentation layer



To transport layer

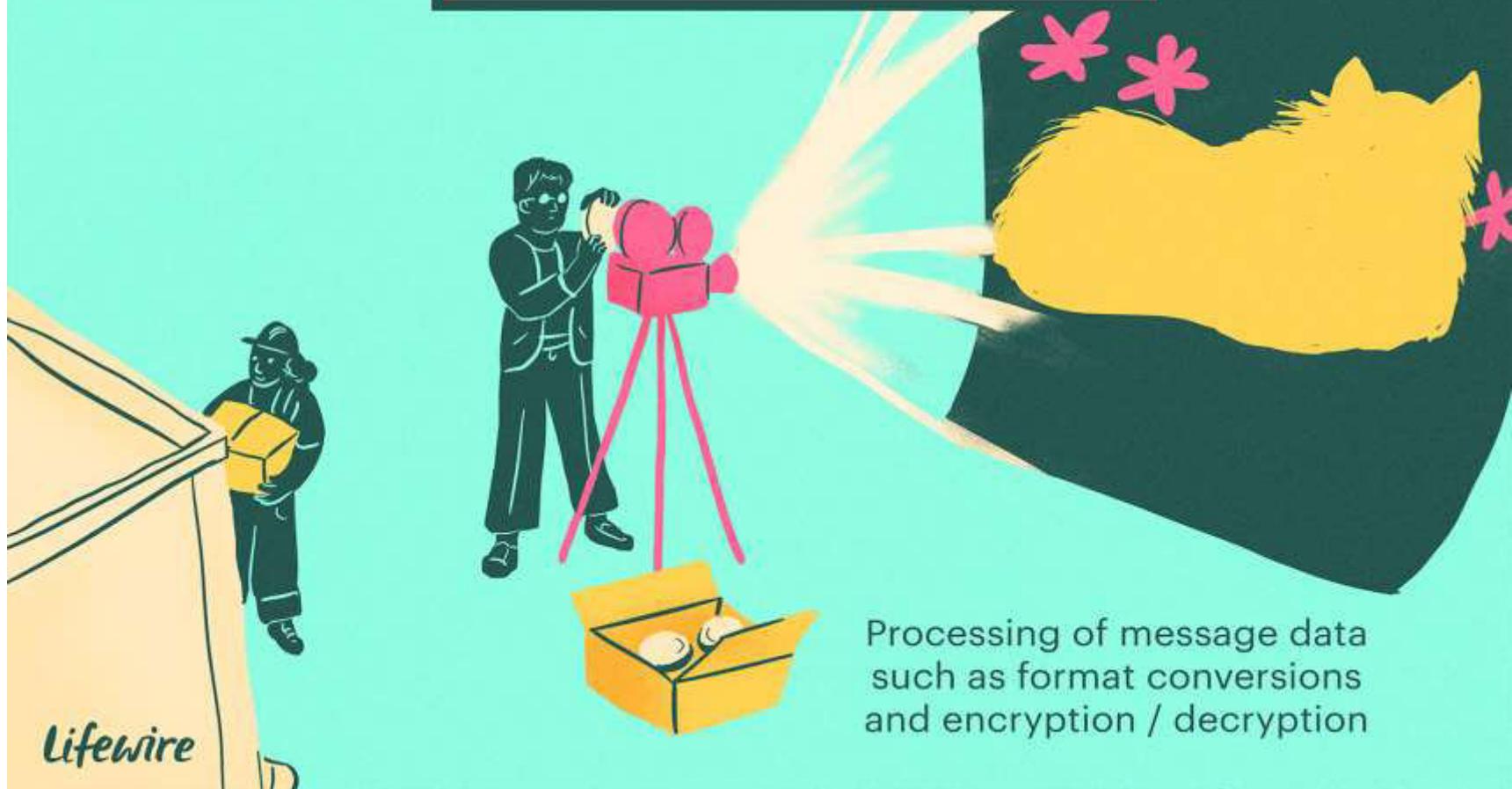
To presentation layer



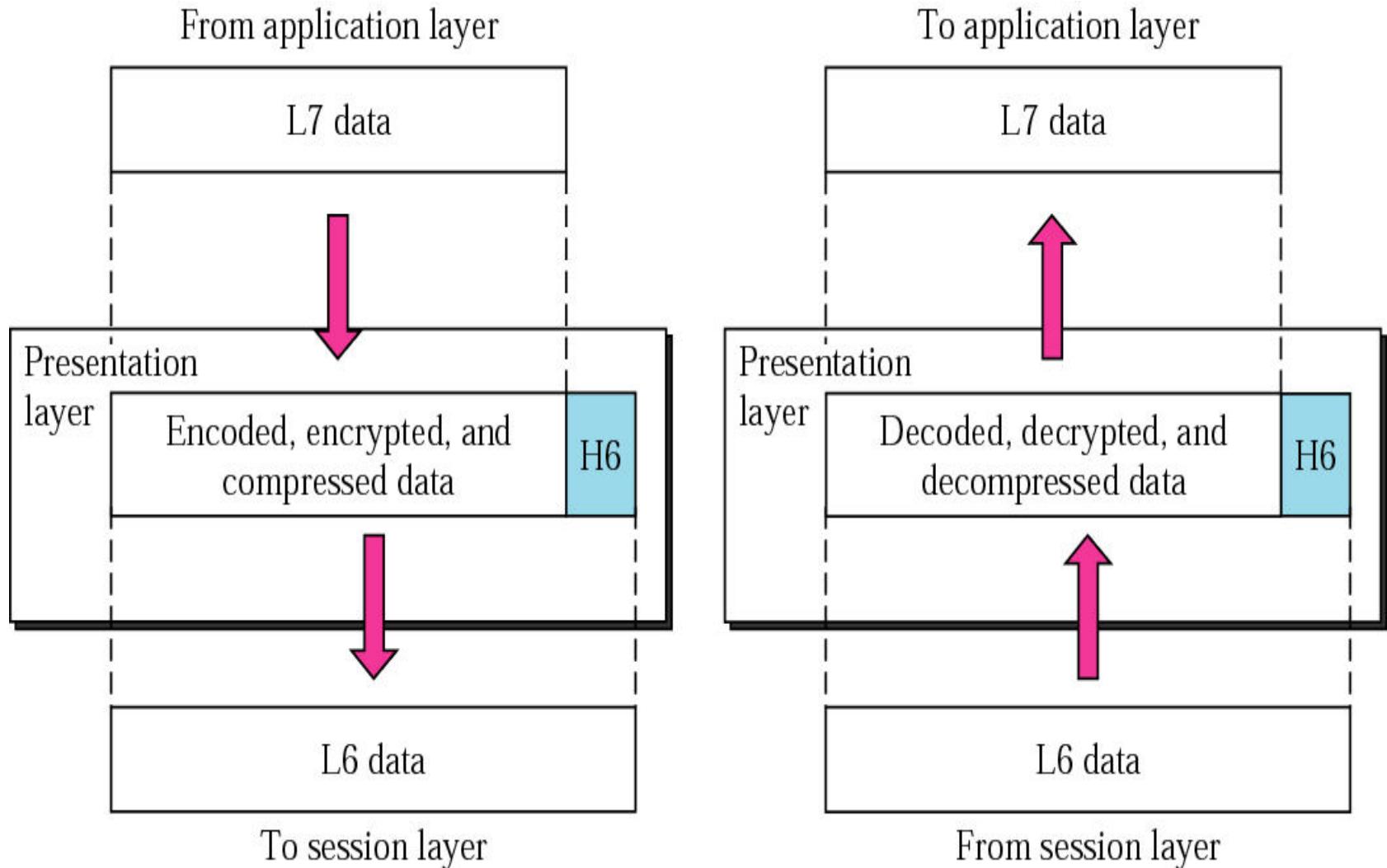
From transport layer

Presentation Layer

Presentation Layer



Presentation Layer



Presentation Layer

- The presentation layer is responsible for presenting data in the format the user can understand.
- The presentation layer can also provide security measures.
- It may encrypt data before sending it to the lower layers for transfer.
- The Presentation layer at the other end would decrypt the data after receiving it.
 - Data encryption, security, compression and code conversion
 - Make sure data is encoded in standard form (ASCII)
 - Handles pass-through of services from session to application layer

APPLICATION LAYER



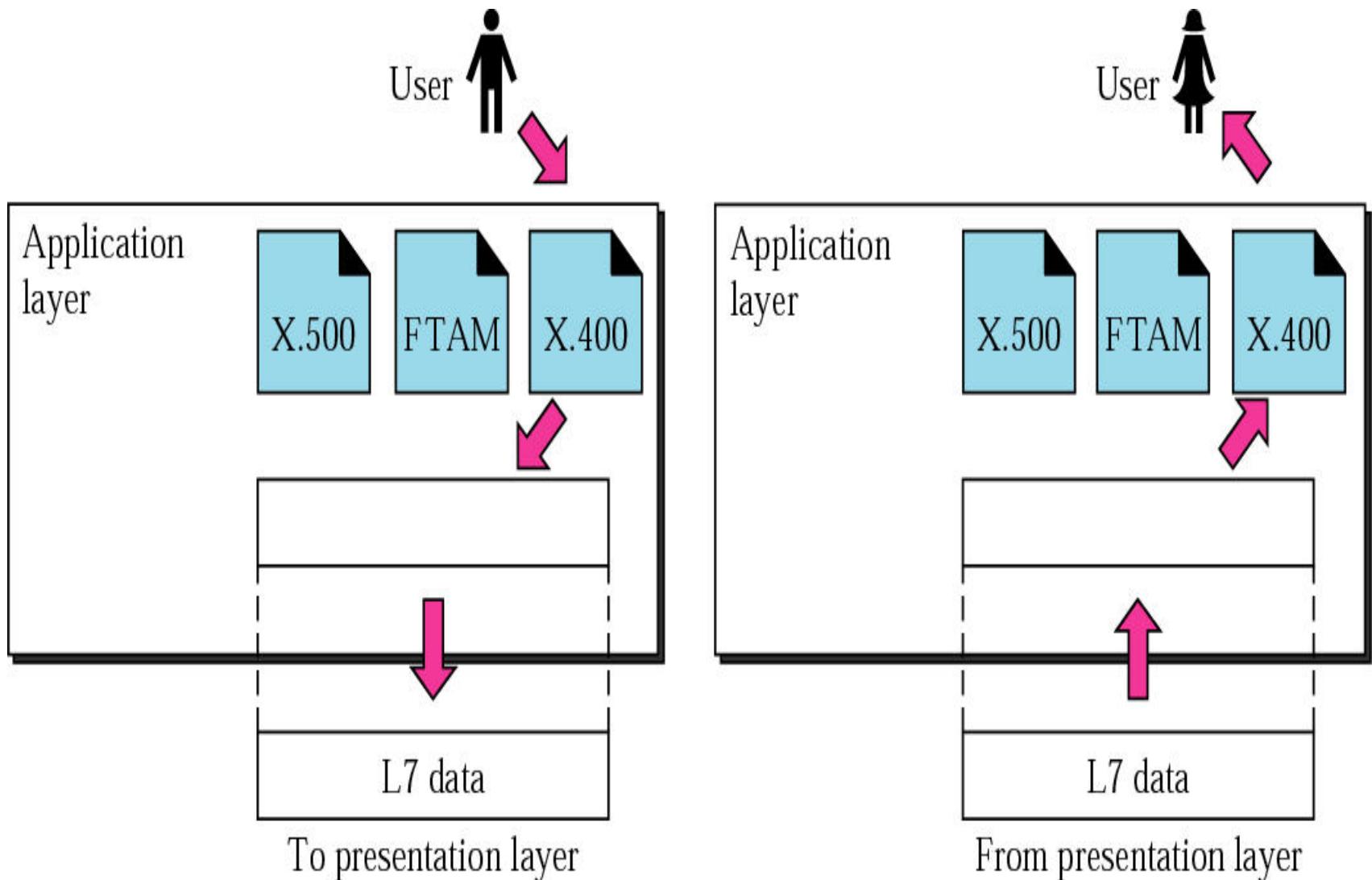
APPLICATION LAYER

- Login, password check
- Agreement on semantics for information exchange
- File transfer, access and management
- Message handling, email
- Job transfer and manipulation
- Directory service
- System management
- Industry protocols
- Database access and management
- Virtual terminals

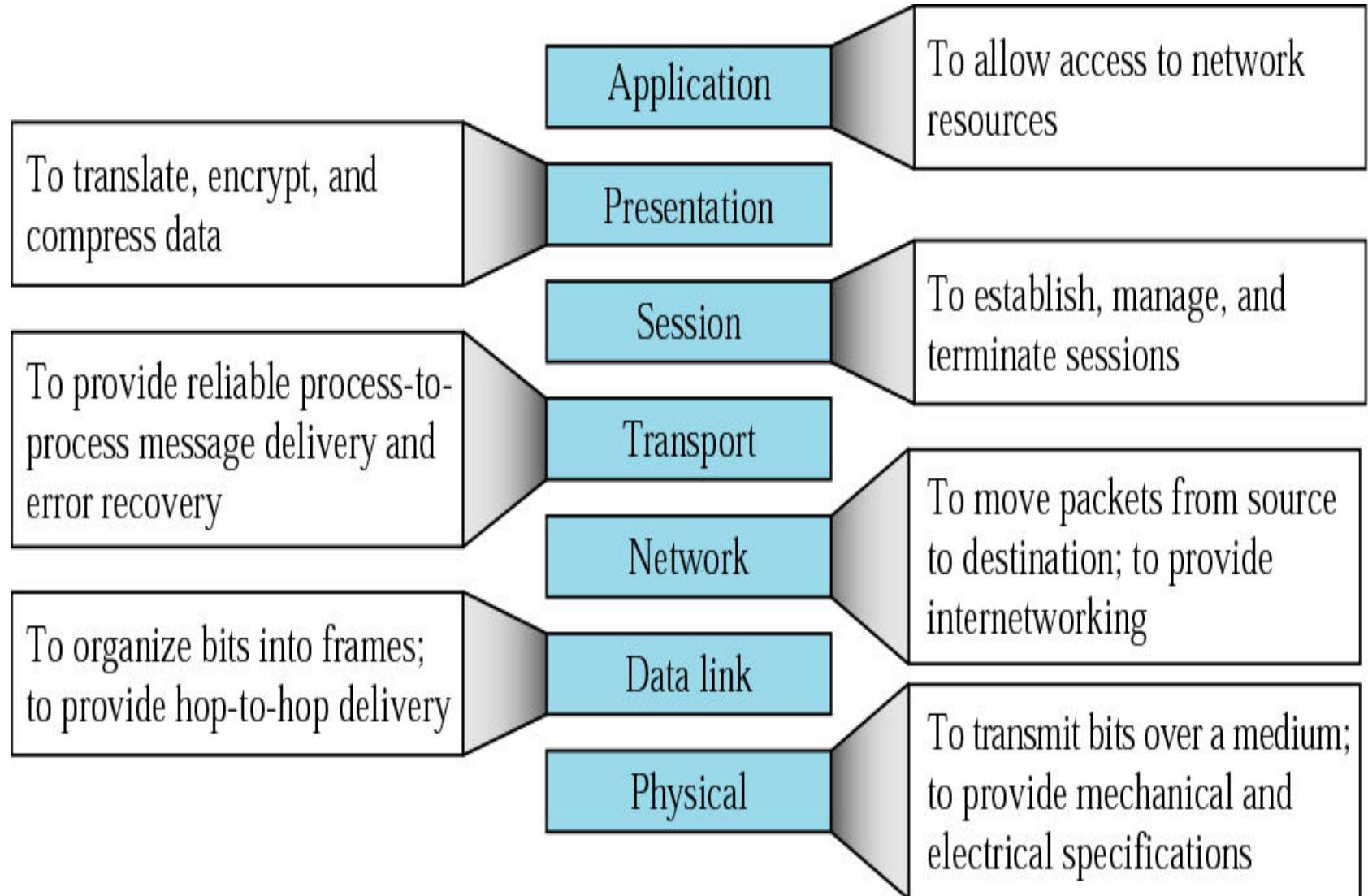
APPLICATION LAYER – (Cont...)

- It is the topmost layer enables a user to access the network.
- This layer provides user interface for network applications such as remote login (TELNET) www, Remote File Transfer (FTP), Electronic Mail (E-Mail) and allows to access the remote data base.
- It allows a user to access, download or upload files from / to a remote host.
- It allows the user to use the mail services.
- Accessing the WebPages is also a utility of this layer.
- All the emulating software's are accessible in this layer.
- The user and the application programs interact with a physical network at this layer.
- All the user application tools available in this layer only.

Application Layer



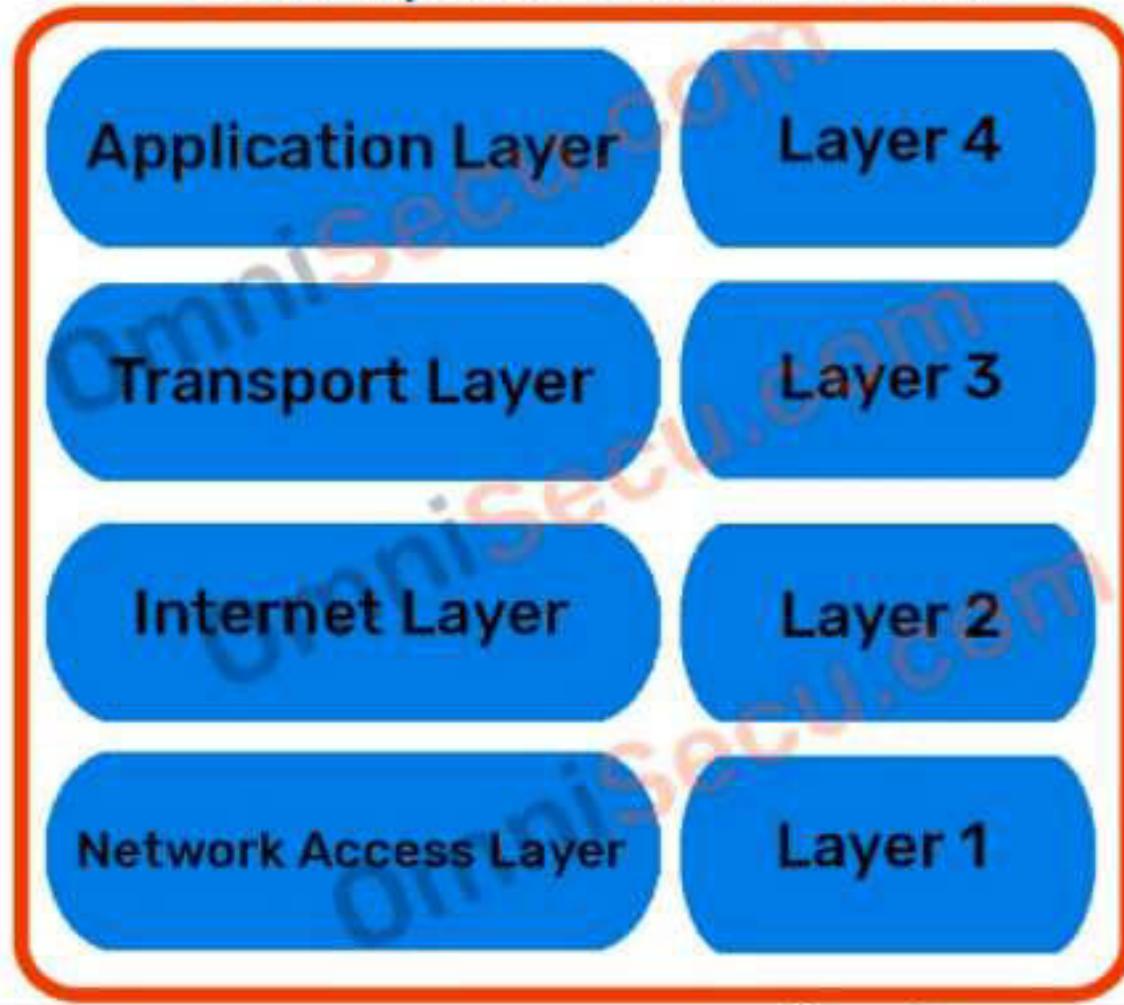
Summary of layers

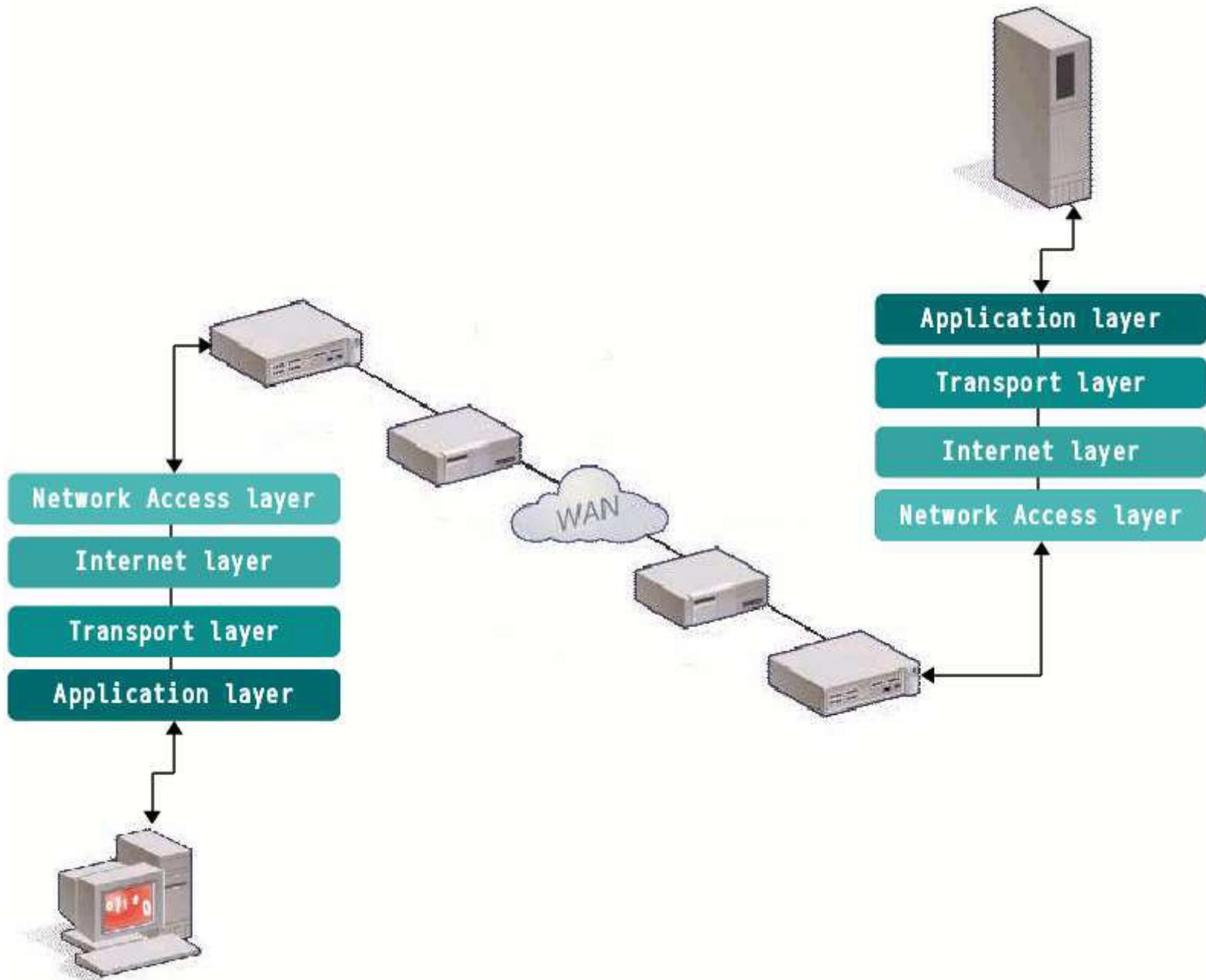


TCP/IP Model

- The **OSI Model** was designed to describe the functions of the communication system by dividing the communication procedure into smaller and simpler components.
- TCP/IP model, it was designed and developed by Department of Defense (DoD) in 1960s and is based on standard protocols, Transmission Control Protocol/Internet Protocol.
- The **TCP/IP model** is a concise version of the OSI model. It contains four layers:
 - Application Layer
 - Transport Layer
 - Internet Layer
 - Network Access Layer

Four Layered TCP/IP Model





Application layer

- This layer contains all application protocols that use the Transport layer.
- Application protocols include FTP, HTTP, DNS, NFS, SMTP, Telnet
- To send data, the application calls up a Transport layer protocol, such as TCP.

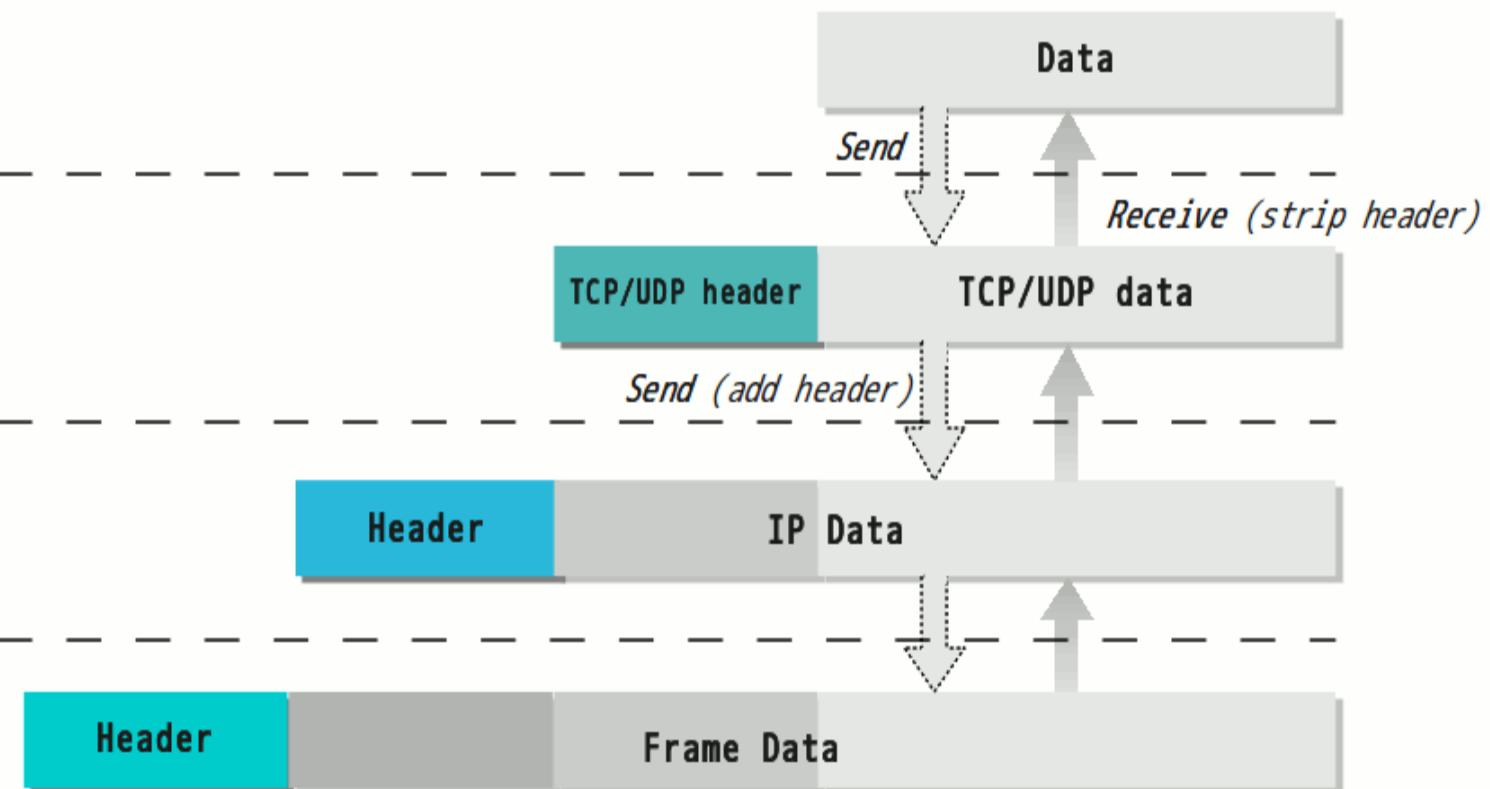
Transport Layer

Application layer

Transport layer

Internet layer

Network Access layer



Transport layer

- TCP and UDP are the most important protocols in this layer, delivering data between application and internet layers.
- TCP provides reliable data delivery service with error detection and error correction. UDP provides a connectionless delivery service.
- When called by an application, TCP wraps the data into a TCP packet (also called TCP segment). contains a TCP header followed by the application data
- TCP then hands the packet to IP.
- TCP keeps track of what data belongs to what process.

Internet layer

- This is above the Network Access layer, and it provides the packet delivery service on which TCP/IP networks are built.
- It provides a routing mechanism allowing for packets to be transmitted across one or more different networks.
- The Internet Protocol (IP) runs in this layer and provides a way to transport datagrams across the network.
- It is a connectionless protocol and does not provide error control, relying on protocols in the other layers to provide error detection and recovery.

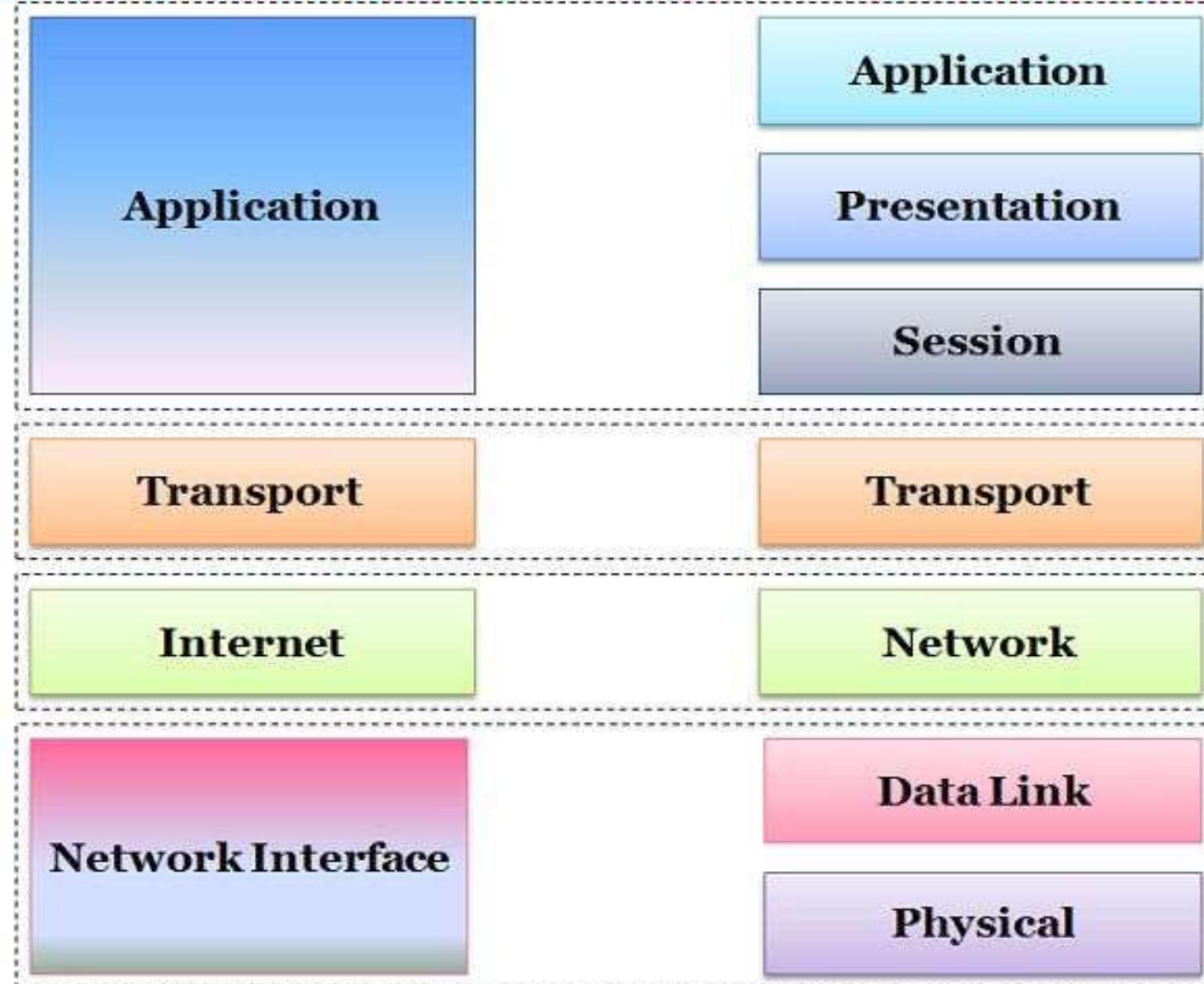
Network access layer

- Protocols in this layer are designed to move packets (IP datagrams) between the internet layer interface of two different hosts on the same physical link
- Network Interface:
 - Each networking device has a corresponding interface in the kernel
 - Ethernet interfaces: eth0, eth1
 - PPP interfaces: ppp0, ppp1 (Point-to-Point Protocol)
 - FDDI interfaces: fddi0, fddi1 (Fiber Distributed Data Interface)

TCP/IP MODEL

VS

OSI MODEL



2.3 TCP/IP Protocol Suite

The **TCP/IP protocol suite** is made of five layers: physical, data link, network, transport, and application. The first four layers provide physical standards, network interface, internetworking, and transport functions that correspond to the first four layers of the OSI model. The three topmost layers in the OSI model, however, are represented in TCP/IP by a single layer called the application layer.

The topics discussed in this section include:

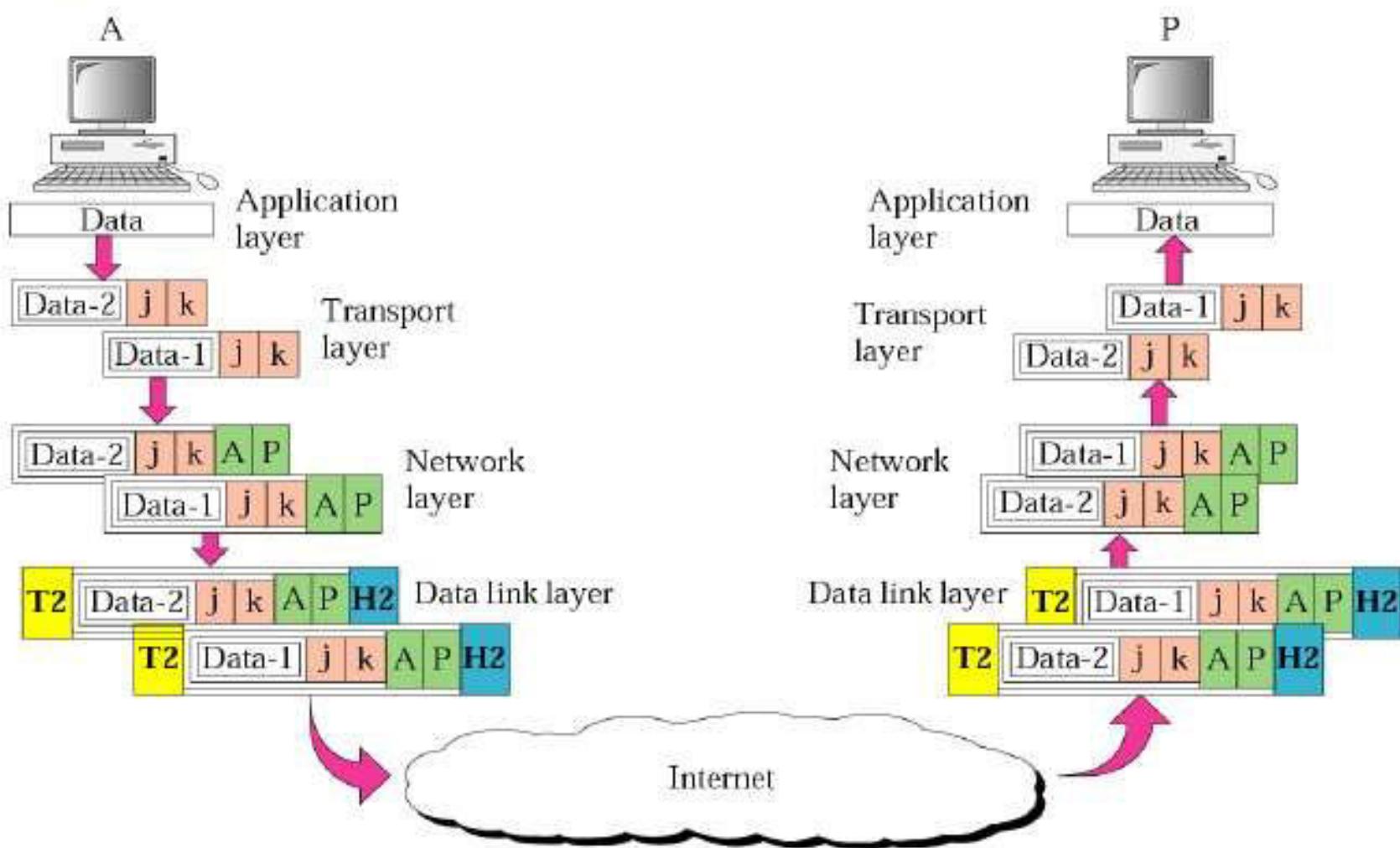
Physical and Data Link Layers

Network Layer

Transport Layer

Application Layer

Figure 2.20 Port addresses



Circuit and Packet switching



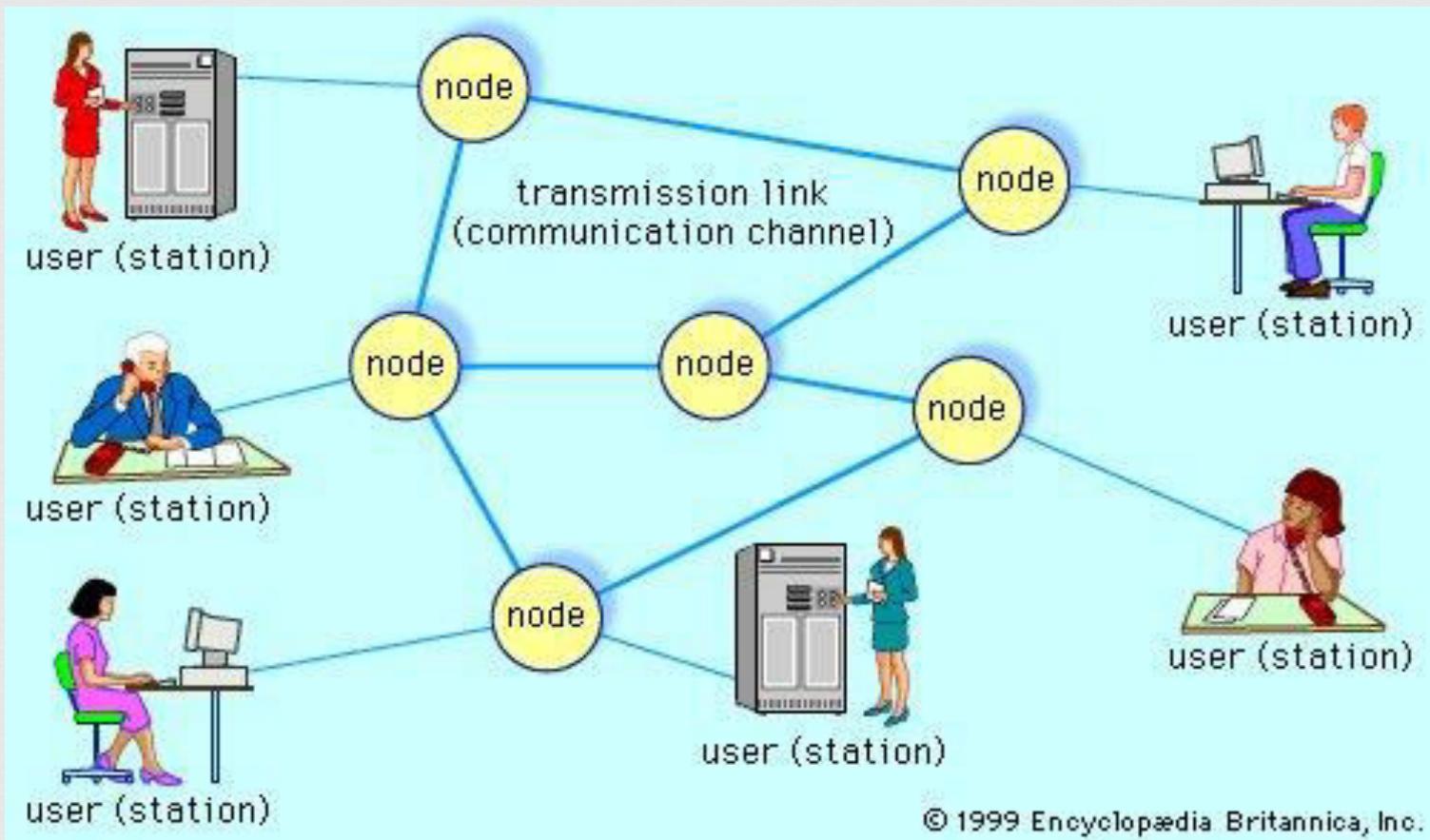
MODULE 2

Overview



- ❖ Switched Communications Networks:
 - ❖ Circuit Switching
 - ❖ Packet Switching
 - ❖ Comparison of Circuit Switching and Packet Switching
 - ❖ Implementing Network Software, Networking
- ❖ Parameters(Transmission Impairment, Data Rate and Performance)

❖ **Telecommunications network**, electronic system of links and switches, and the controls that govern their operation, that allows for data transfer and exchange among multiple users



- ❖ When several users of telecommunications media wish to communicate with one another, they must be organized into some form of network.
- ❖ In theory, each user can be given a direct point-to-point link to all the other users in what is known as a fully connected topology (telephone), but in practice this technique is impractical and expensive—especially for a large and dispersed network.
- ❖ Method is inefficient, since most of the links will be idle at any given time.
- ❖ Modern telecommunications networks avoid these issues by establishing a linked network of switches, or nodes, such that each user is connected to one of the nodes. Each link in such a network is called a communications channel. Wire, fibre-optic cable, and radio waves may be used for different communications channels.

Types of networks



- ❖ Switched communications network
- ❖ Broadcast network
- ❖ Network access
- ❖ Scheduled access
- ❖ Random access
- ❖ Carrier sense multiple access

Switched communications network

- ❖ A switched communications  network transfers data from source to destination through a series of network nodes.
- ❖ Switching can be done in one of two ways. In a circuit-switched network, and a packet-switched network

Switched Communications Networks

Switching Techniques

In large networks there might be multiple paths linking sender and receiver. Information may be switched as it travels through various communication channels. There are three typical switching techniques available for digital traffic.

- Circuit Switching
- Message Switching
- Packet Switching

Circuit Switching

- **Circuit switching** is a technique that directly connects the sender and the receiver in an unbroken path.
- Telephone switching equipment, for example, establishes a path that connects the caller's telephone to the receiver's telephone by making a physical connection.
- With this type of switching technique, once a connection is established, a dedicated path exists between both ends until the connection is terminated.
- Routing decisions must be made when the circuit is first established, but there are no decisions made after that time.

Circuit switching

Advantages:

- The communication channel (once established) is dedicated.

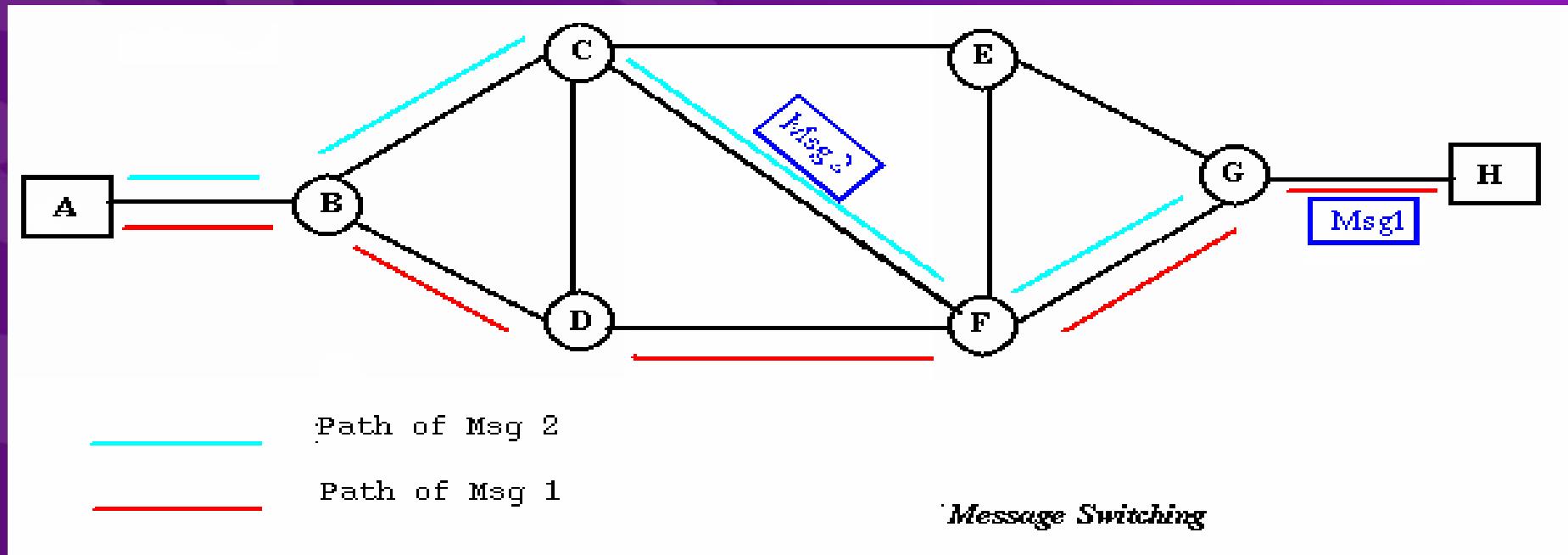
Disadvantages:

- Possible **long wait** to establish a connection, (10 sec, more on long-distance or international calls.) during which no data can be transmitted.
- **More expensive** than any other switching techniques, because a dedicated path is required for each connection. **Inefficient use** of the communication channel, because the channel is not used when the connected systems are not using it.

Message Switching

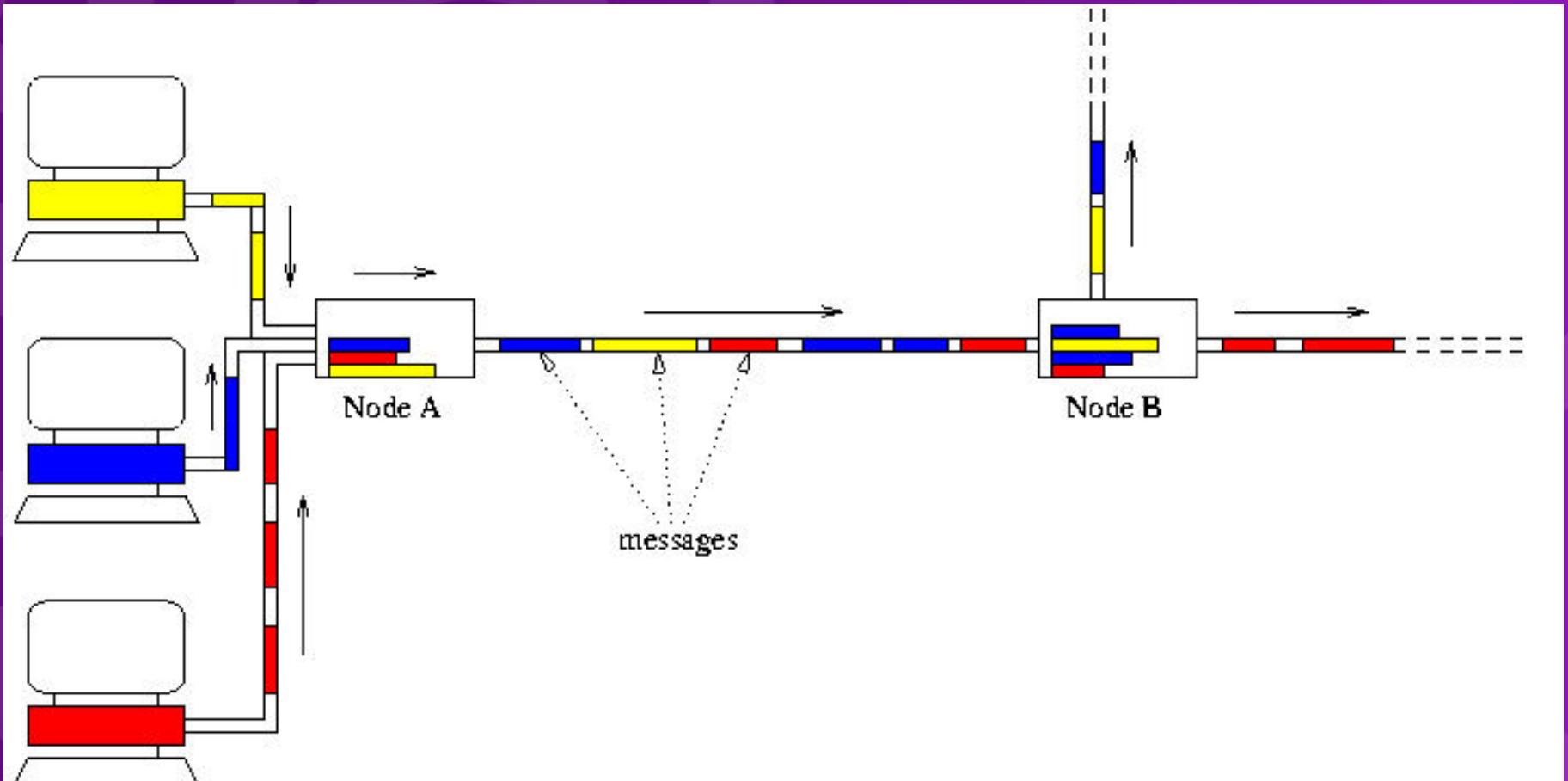
- There is **no need to establish a dedicated path between two stations.**
- When a station sends a message, the destination address is appended to the message.
- The message is then transmitted through the network, in its entirety, from node to node.
- Each node receives the entire message, stores it in its entirety on disk, and then transmits the message to the next node.
- This type of network is called a **store-and-forward network.**

Message Switching



- A message-switching node is typically a general-purpose computer.
- The device **needs sufficient secondary-storage capacity to store the incoming messages**, which could be long.
- A **time delay** is introduced using this type of scheme due to **store- and-forward time**, plus the **time required to find the next node in the transmission path**.

Message Switching



Message Switching

Advantages:

- Channel efficiency can be greater compared to circuit-switched systems, because more devices are **sharing the channel**.
- **Traffic congestion can be reduced**, because messages may be temporarily stored in route.
- **Message priorities** can be established due to store-and-forward technique.
- **Message broadcasting** can be achieved with the use of broadcast address appended in the message.

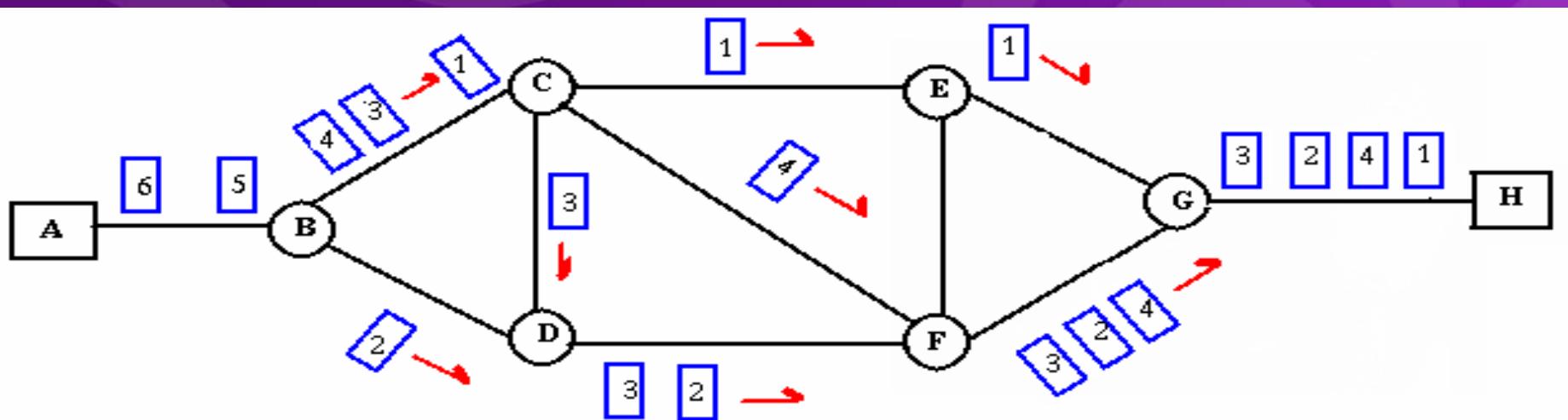
Message Switching

Disadvantages:

- Message switching is not **compatible** with interactive applications.
- Store-and-forward devices are **expensive**, because they must have large disks to hold potentially long messages.

Packet Switching

- *Packet switching* can be seen as a solution that tries to combine the advantages of message and circuit switching and to minimize the disadvantages of both.
- The methods of packet switching: Datagram and virtual circuit.



Packet Switching

- In both packet switching methods, a message is broken into small parts, called **packets**.
- Each packet is tagged with appropriate source and destination **addresses**.
- Since packets have a strictly defined maximum length, they can be stored in **main memory** instead of disk, therefore access delay and cost are minimized.
- Also the transmission speeds, between nodes, are optimized.
- With current technology, packets are generally accepted onto the network on a first-come, first-served FIFO **basis**. If the network becomes overloaded, packets are delayed or discarded ('`dropped").

Packet size

- The size of the packet can vary from 180 bits, the size for the Data kit® virtual circuit switch designed by Bell Labs for communications and business applications.
- To 1,024 or 2,048 bits for the 1PSS® switch, also designed by Bell Labs for public data networking; to 53 bytes for ATM switching, such as Lucent Technologies' packet switches.

Packet switching

- The analog signal from your **phone** is converted into a digital data stream. That series of digital bits is then divided into relatively tiny clusters of bits, called **packets**. Each packet has at its beginning the digital address -- a long number -- to which it is being sent. The system blasts out all those tiny packets, as fast as it can, and they travel across the nation's digital backbone systems to their destination: the telephone, or rather the telephone system, of the person you're calling.
- They do not necessarily travel together; they do **not travel** sequentially. They don't even all travel via the **same route**. But eventually they arrive at the right point -- that digital address added to the front of each string of digital data -- and at their destination are reassembled into the correct order, then converted to analog form, so your friend can understand what you're saying.

Packet Switching: Datagram

- Datagram packet switching is similar to message switching in that each packet is a **self-contained** unit with complete addressing information attached.
- This fact allows packets to take a variety of possible paths through the network.
- So the packets, each with the same destination address, do not follow the same route, and they may arrive out of sequence at the exit point node (or the destination).
- Reordering is done at the destination point based on the sequence number of the packets.
- It is possible for a packet to be destroyed if one of the nodes on its way is crashed momentarily. Thus all its queued packets may be lost.

Packet Switching: Virtual Circuit

- In the virtual circuit approach, a preplanned route is established before any data packets are sent.
- A logical connection is established when
 - a sender send a "**call request packet**" to the receiver and the receiver send back an acknowledge packet "**call accepted packet**" to the sender if the receiver agrees on conversational parameters.
- The conversational parameters can be maximum packet sizes, path to be taken, and other variables necessary to establish and maintain the conversation.
- Virtual circuits imply acknowledgements, flow control, and error control, so virtual circuits are reliable.
- That is, they have the capability to inform upper-protocol layers if a transmission problem occurs.

Packet Switching:Virtual Circuit

- In virtual circuit, the route between stations does not mean that this is a dedicated path, as in circuit switching.
- A packet is still buffered at each node and queued for output over a line.
- The difference between virtual circuit and datagram approaches:
 - With virtual circuit, the node does not need to make a routing decision for each packet.
 - It is made only once for all packets using that virtual circuit.

Packet Switching: Virtual Circuit

VC's offer guarantees that

- the packets sent arrive in the order sent
- with no duplicates or omissions
- with no errors (with high probability)
regardless of how they are implemented internally.

Advantages of packet switching

Advantages:

- Packet switching is **cost effective**, because switching devices do not need massive amount of secondary storage.
- Packet switching offers **improved delay characteristics**, because there are no long messages in the queue(max packet size is fixed).
- Packet can be **rerouted** if there is any problem, such as, busy or disabled links.
- The advantage of packet switching is that many network users can **share** the same channel at the **same time**. Packet switching can maximize link **efficiency** by making optimal use of link bandwidth.

Disadvantages of packet switching

Disadvantages:

- **Protocols** for packet switching are typically more **complex**.
- It can add some **initial costs** in implementation.
- If packet is lost, sender needs to **retransmit** the data.
- Another disadvantage is that packet-switched systems still can't deliver the same quality as dedicated circuits in applications requiring very **little delay** - like voice conversations or moving images.

Implementing Network Software



- ❧ 1 Introduction
- ❧ 2 Sockets
- ❧ 3 Reading and Writing
- ❧ 4 Writing the Server side
- ❧ 5 The Knock Knock Server
- ❧ 6 The Knock Knock Protocol
- ❧ 7 The Knock Knock Client
- ❧ 8 Running the Programs
- ❧ 9 Supporting Multiple Clients

Implementing Network Software



- ❖ In Client/Server applications, the server provides services and the client consume these services. The communication between client and server should be reliable what means that no data can be dropped, it just must arrive in the same order that was sent.
- ❖ TCP provides a reliable one for this purpose. It is called point to point communication and it creates a channel between client and server where the connection starts and ends in a *socket*.

Implementing Network Software - Sockets



A socket is one end point of two way communication link between programs running on the network. In java, Sockets are classes that represent the connection between client (class `Socket`) and server (class `ServerSocket`). A socket is bound to a port number so that the TCP layer can identify the application that data is destined to be sent to.

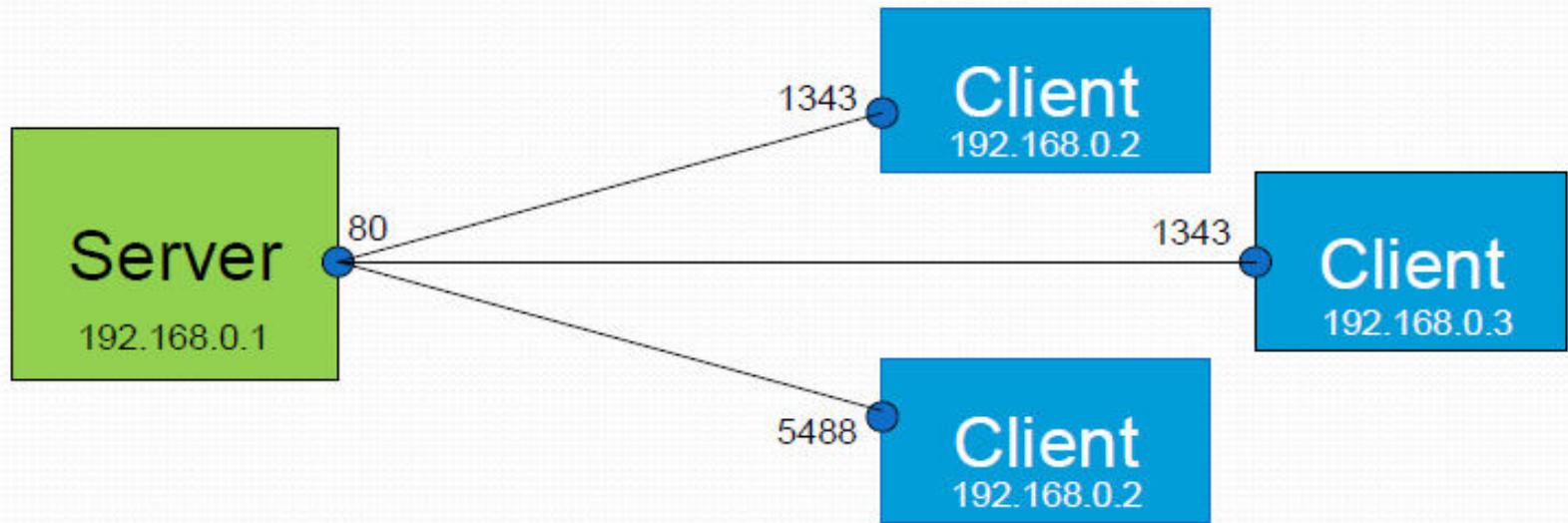
An endpoint is a combination of an IP address and a port number. Every TCP connection can be uniquely identified by its endpoints. That way you can have multiple connections between your host and the server.

Sockets

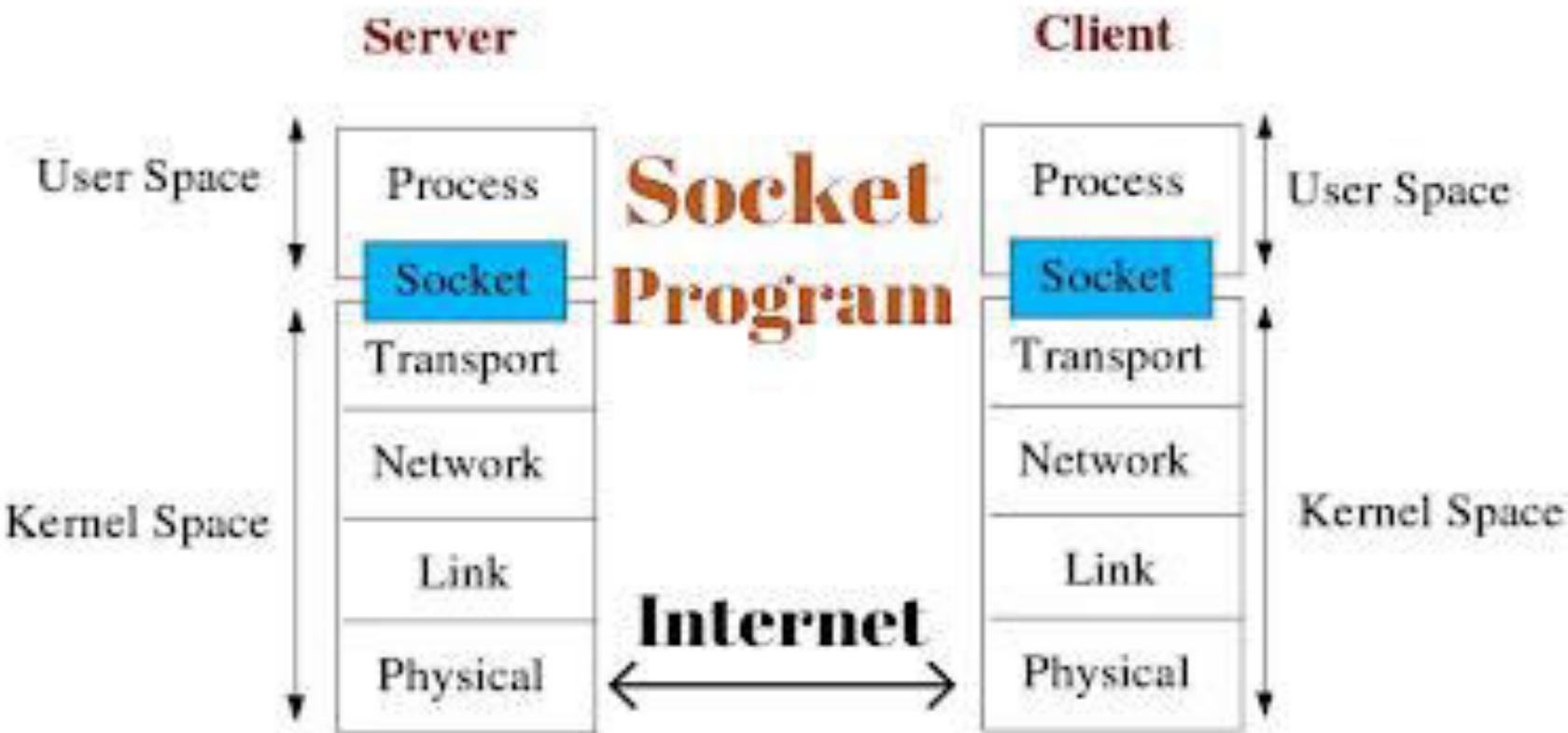


- ❖ End-point of inter process communication
- ❖ Provides an interface to send data to/from the network through a port
- ❖ Exists on either side of connection
- ❖ What exactly creates a Socket?
 - <IP address, Port #> tuple
- ❖ What makes a connection?
 - {Source<IP address, Port #>, Destination <IP address, Port #>}
 - i.e. source socket - destination socket pair uniquely identifies a connection

Example



Socket Programming in Networking



Implementing Network Software

☞ Sockets



TYPES



- ❖ Two essential types of sockets:
 - ❖ Stream Sockets
 - Connection oriented
 - Rely on TCP to provide reliable two way connected communication
 - ❖ Datagram Sockets
 - Rely on UDP
 - Connection is unreliable

Use of Sockets



- ❖ Connection-based sockets communicate client-server: the server waits for a connection from the client
- ❖ Connectionless sockets are peer-to peer: each process is symmetric.

Sockets : Client-Server application



Algorithm for TCP client

- ❖ - Find the IP address and port number of server
- ❖ - Create a TCP socket
- ❖ - Connect the socket to server (Server must be up and listening for new requests)
- ❖ - Send/receive data with server using the socket
- ❖ - Close the connection

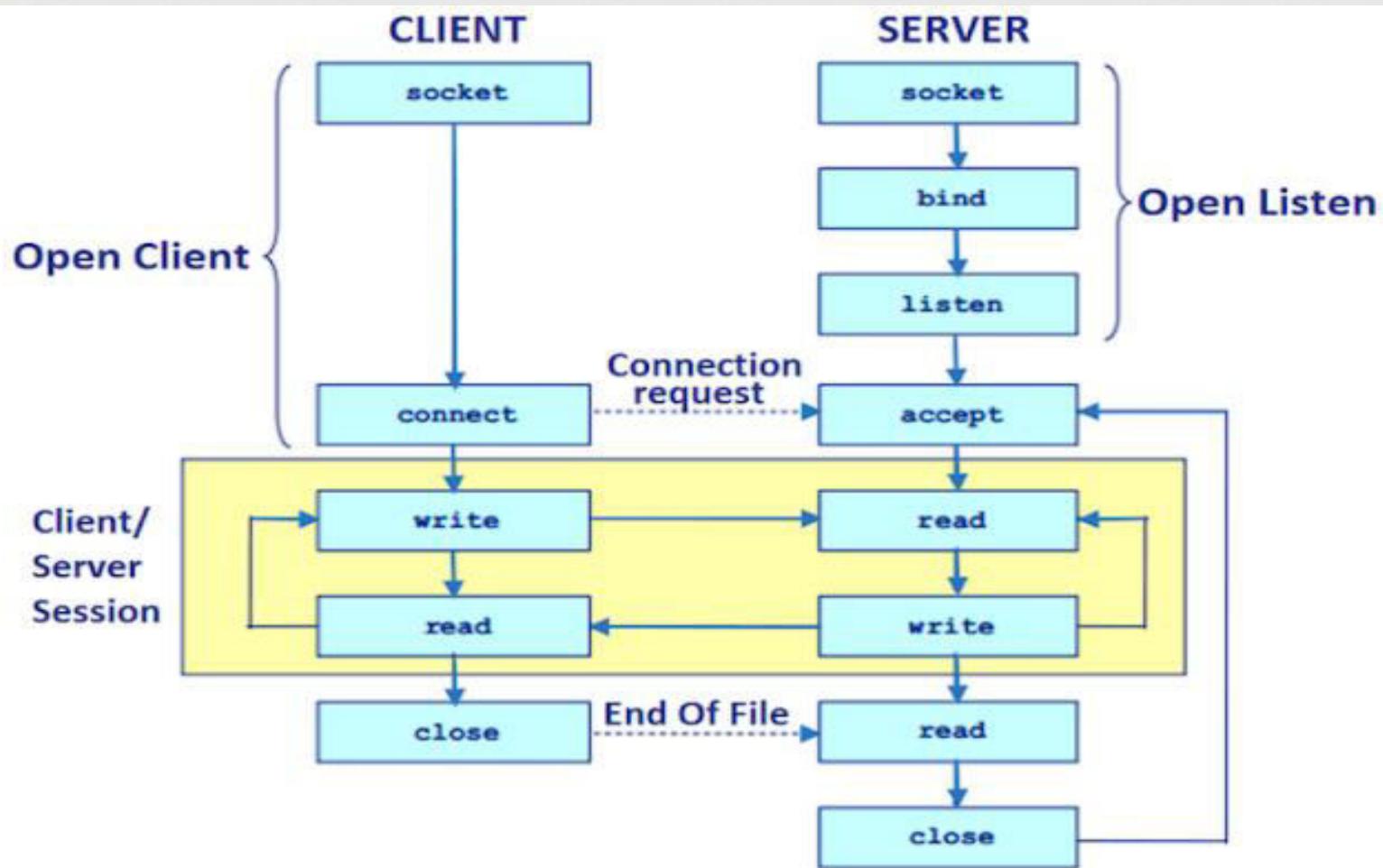
Sockets : Client-Server application



❖ Algorithm for TCP SERVER

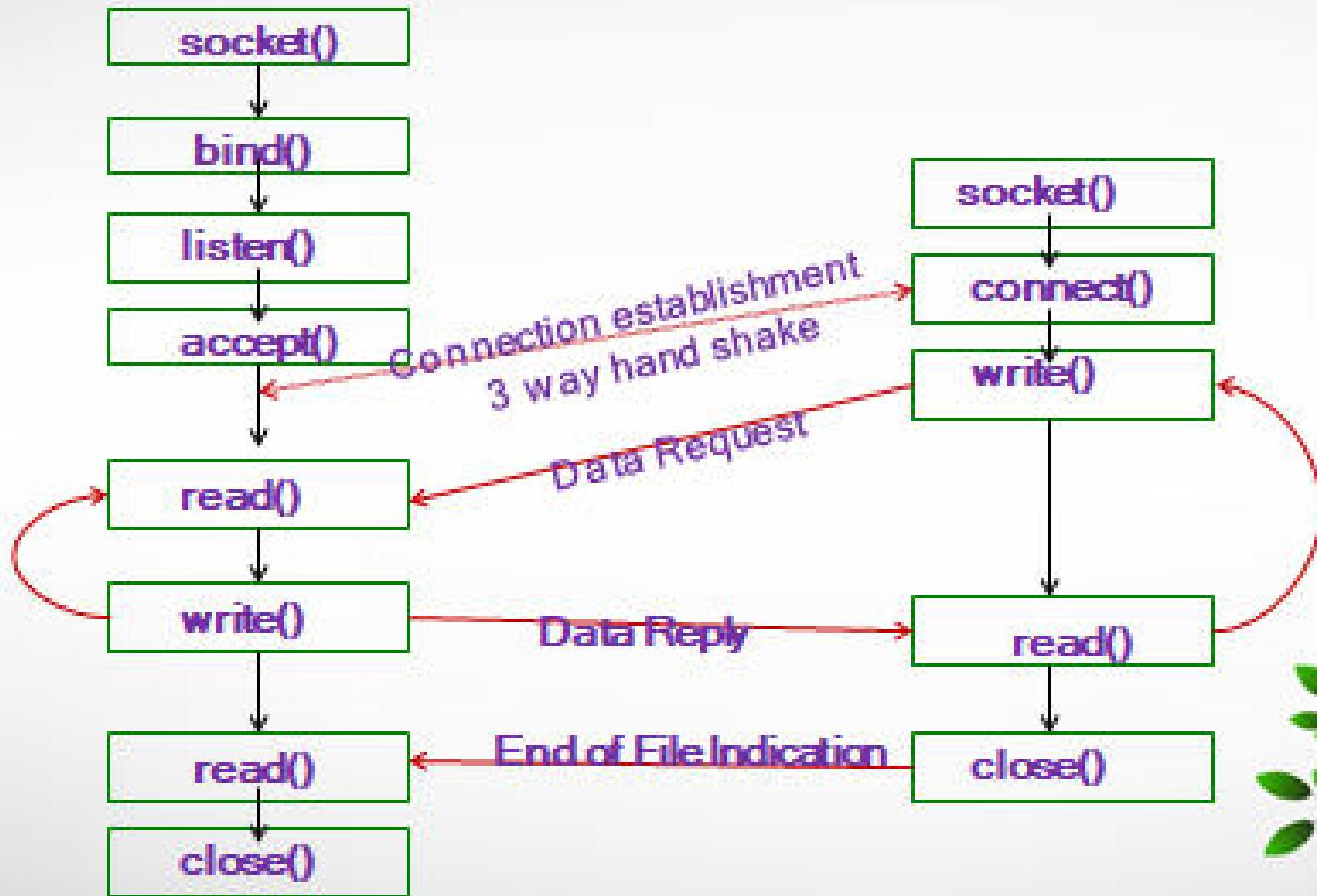
- ❖ - Find the port number of server
- ❖ - Create a TCP *server socket*
- ❖ - Bind the *server socket* to server Port number (this is the port to which clients will connect)
- ❖ - Accept a new connection from client
- ❖ - Send/receive data with client using the *client socket*
- ❖ - Close the connection with client

Sockets : Client-Server application



Client Server Architecture

- Basic Socket functions for Elementary TCP Client Server



Feature	TCP	UDP
Connection status	Requires an established connection to transmit data <i>(connection should be closed once transmission is complete)</i>	Connectionless protocol with no requirements for opening, maintaining, or terminating a connection
Data sequencing	Able to sequence	Unable to sequence
Guaranteed delivery	Can guarantee delivery of data to the destination router	Cannot guarantee delivery of data to the destination
Retransmission of data	Retransmission of lost packets is possible	No retransmission of lost packets

Feature	TCP	UDP
Error checking	Extensive error checking and acknowledgment of data	Basic error checking mechanism using checksums
Method of transfer	Data is read as a byte stream; messages are transmitted to segment boundaries	UDP packets with defined boundaries; sent individually and checked for integrity on arrival
Speed	Slower than UDP	Faster than TCP
Broadcasting	Does not support Broadcasting	Does support Broadcasting
Optimal use	Used by HTTPS, HTTP, SMTP, POP, FTP, etc	Video conferencing streaming, DNS, VoIP, etc

Noisy data



- ❖ Noisy data: data with a large amount of additional meaningless information in it called noise.
- ❖ Noisy data means corrupted data. It also includes any data that cannot be understood and interpreted correctly by machines, such as unstructured text.
- ❖ Noisy data can adversely affect the results of any data analysis and skew conclusions if not handled properly. Statistical analysis is sometimes used to weed the noise out of noisy data

Sources of noise

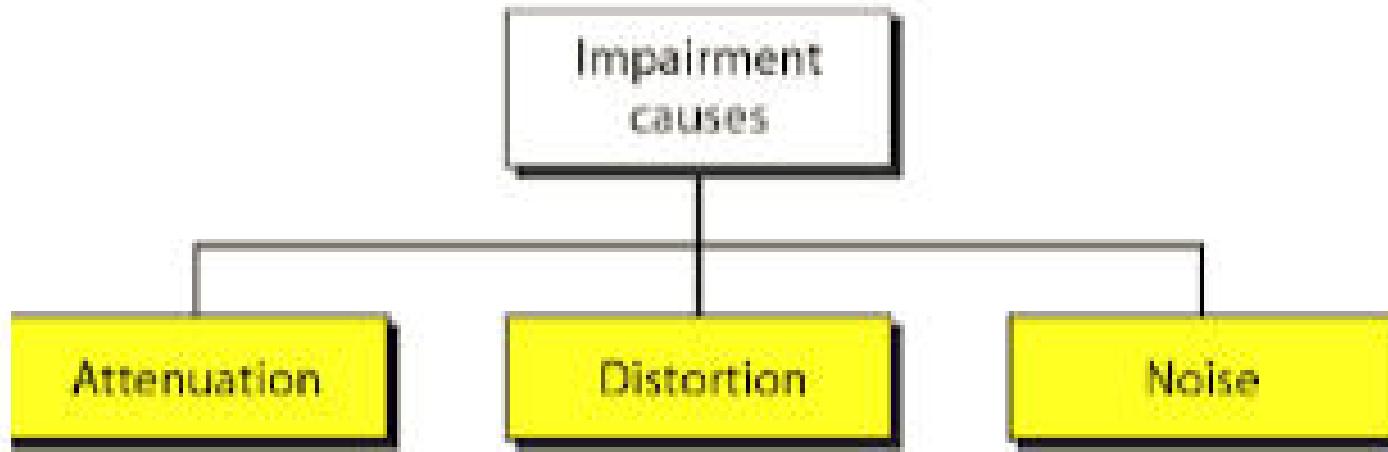


- ❖ Differences in real-world measured data from the true values come about from by multiple factors affecting the measurement.
- ❖ **Random noise** (*white noise*) in a signal is measured as the Signal-to-Noise Ratio. Random noise is an unavoidable problem. It affects the data collection and data preparation processes, where errors commonly occur.



- ❖ Signals travel through transmission media, which are not perfect. The imperfection causes signal impairment.
- ❖ This means that the signal at the beginning of the medium is not the same as the signal at the end of the medium. What is sent is not what is received.
- ❖ Three causes of impairment are attenuation, distortion, and noise

Impairments

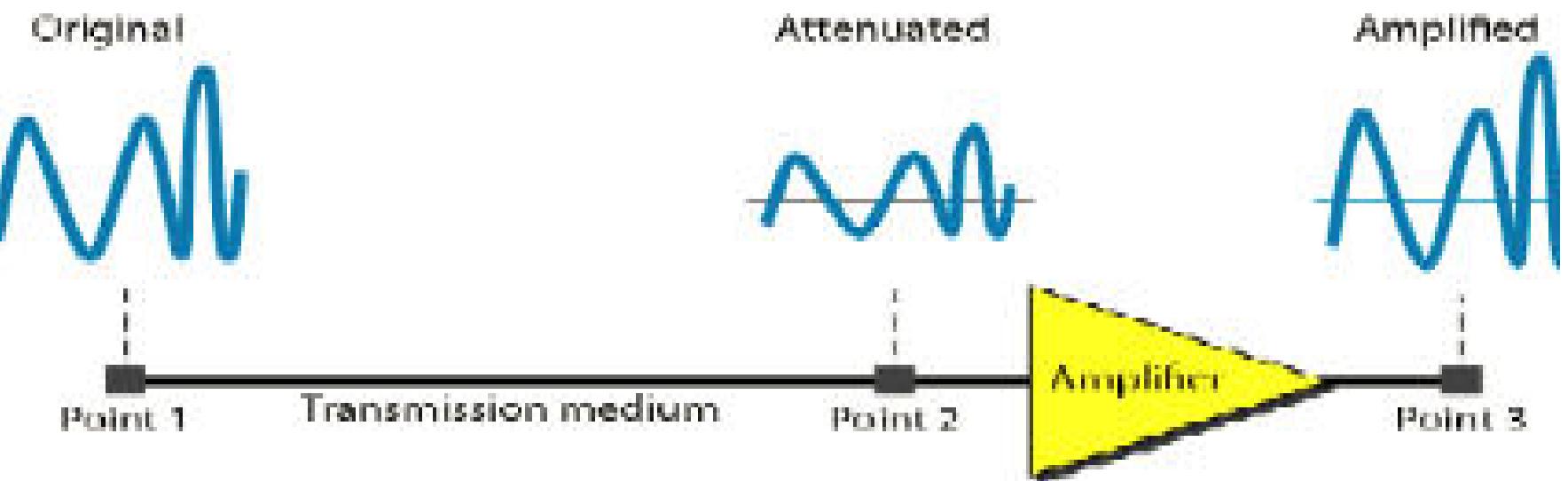


Attenuation



- ❖ Means loss of energy -> weaker signal
- ❖ When a signal travels through a medium it loses energy overcoming the resistance of the medium.
- ❖ That is why a wire carrying electric signals gets warm, if not hot, after a while. Some of the electrical energy in the signal is converted to heat.
- ❖ Amplifiers are used to compensate for this loss of energy by amplifying the signal.

Measurement of Attenuation



Decibel

- ❖ To show that a signal has lost or gained strength, engineers use the unit of the decibel.
- ❖ The decibel (dB) measures the relative strengths of two signals or one signal at two different points. Note that the decibel is negative if a signal is attenuated and positive if a signal is amplified.

$$dB = 10 \log_{10} \frac{P_2}{P_1}$$

- ❖ Variables P_1 and P_2 are the powers of a signal at points 1 and 2, respectively. In this case, because power is proportional to the square of the voltage, the formula is $dB = 20 \log 10 (V_2 / V_1)$.

Example 1

- Suppose a signal travels through a transmission medium and its power is reduced to one-half.

This means that $P_2 = \frac{1}{2} P_1$. In this case, the attenuation (loss of power) can be calculated as

$$10 \log_{10} \frac{P_2}{P_1} = 10 \log_{10} \frac{0.5 P_1}{P_1} = 10 \log_{10} 0.5 = 10(-0.3) = -3 \text{ dB}$$

Example 2



- ❖ A signal travels through an amplifier, and its power is increased 10 times. This means that $P_2 = 10P_1$. In this case, the amplification (gain of power) can be calculated as

$$10 \log_{10} \frac{P_2}{P_1} = 10 \log_{10} \frac{10P_1}{P_1} = 10 \log_{10} 10 = 10(1) = 10 \text{ dB}$$

Example 3



Sometimes the decibel is used to measure signal power in milliwatts. In this case, it is referred to as dBm and is calculated as $\text{dBm} = 10 \log_{10} P_m$, where P_m is the power in milliwatts. Calculate the power of a signal with $\text{dBm} = -30$.

Solution

We can calculate the power in the signal as

$$\begin{aligned}\text{dBm} &= 10 \log_{10} P_m = -30 \\ \log_{10} P_m &= -3 \quad P_m = 10^{-3} \text{ mW}\end{aligned}$$

Distortion

- ❖ Distortion means that the signal changes its form or shape. Distortion can occur in a composite signal made of different frequencies. Each signal component has its own propagation speed (see the next section) through a medium and, therefore, its own delay in arriving at the final destination.
- ❖ Differences in delay may create a difference in phase if the delay is not exactly the same as the period duration. In other words, signal components at the receiver have phases different from what they had at the sender.

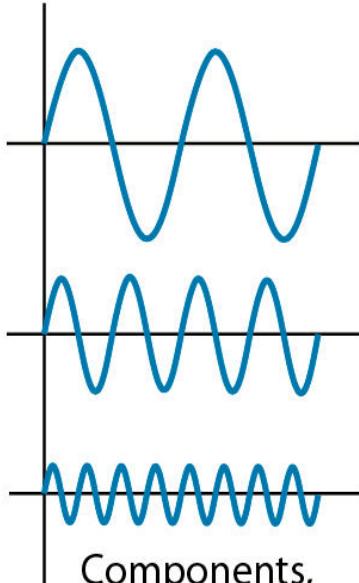
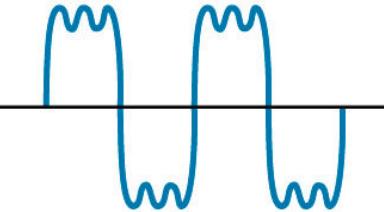
Distortion

- ❖ Distortion is known as the alteration of the original signal. This may happen due to the **properties of the medium**. There are many types of distortion such as amplitude distortion, harmonic distortion, and phase distortion.
- ❖ For electromagnetic waves polarization distortions also occurs. When the *distortion occurs, shape of the waveform is changed.*
- ❖ For example, amplitude distortion happens if all the parts of the signals are not equally amplified. This happens in wireless transmissions because the medium get changed by the time. The receivers should be able to identify these distortions.

Distortion



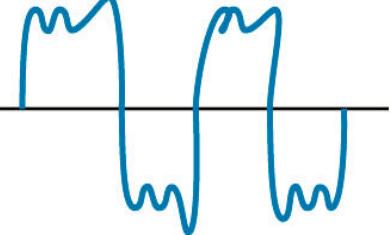
Composite signal
sent



Components,
in phase

At the sender

Composite signal
received



Components,
out of phase

At the receiver

Distortion



Types:

- ❖ Amplitude distortion
- ❖ Delay / Phase Distortion
- ❖ Frequency related distortion

Difference between attenuation and distortion

- ❖ Although scaled down in amplitude, shape of waveform does not change in attenuation unlike in distortion.
- ❖ Removal of the effects of attenuation is easier than removing the effects of distortion.
- ❖ If the attenuation happens in different amounts for the different parts of the signal, it is a distortion.

Noise



- ❖ The random and unpredicted electrical signal (coming from both internal or external portion of the system) which interfere the reception of actual required signal is called- noise.
- ❖ Noise can be characterized by statistical parameter such as averaged/ squared noise, current/ voltage etc.
- ❖ Noise is unwanted signals that are inserted somewhere between transmission and reception. Noise is the major limiting factor in communications system performance. Noise may be divided into four categories:
- ❖ Thermal noise, Intermodulation noise, Cross talk, Impulse noise ...

Thermal noise

Thermal noise in watts present in a bandwidth of B Hertz can be expressed as: $N = kTB$ (Unit:Watts)

Where T = temperature in kelvins ,

k = Boltzmann's constant = $1.38 * 10^{-23}$ J/K

B is bandwidth.

In decibel-watts, $N = 10 \log(kTB)$

$$\text{i.e } N = -228.6 + 10 \log T + 10 \log B$$

EXAMPLE : Given a receiver with an effective noise temperature of 294 K and a 10-MHz bandwidth, the thermal noise level at the receiver's output is

$$\begin{aligned}N &= -228.6 + 10 \log T + 10 \log B \\&= -228.6 + 10 \log (294) + 10 \log (10^7) \\&= -133.9 \text{ dBW}\end{aligned}$$

Thermal noise

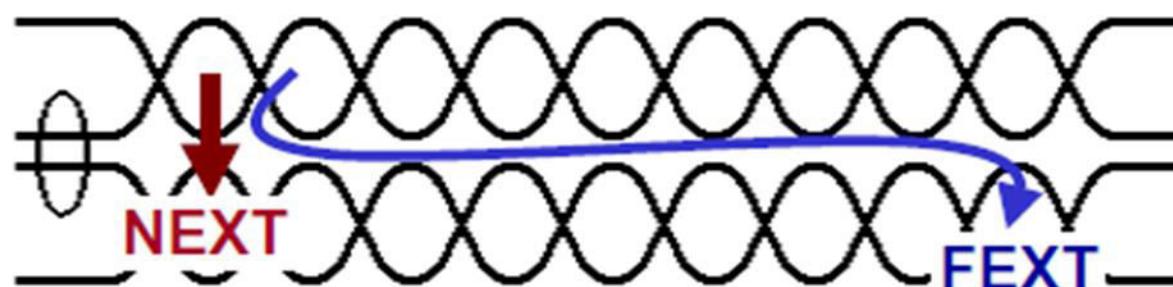


- ❖ Thermal noise is due to thermal agitation of electrons. It is present in all electronic devices and transmission media and is a function of temperature.
- ❖ Thermal noise is uniformly distributed across the bandwidths typically used in communications systems and hence is often referred to as white noise.
- ❖ Thermal noise is particularly significant for satellite communication. The noise is assumed to be independent of frequency. Thus the thermal noise in watts present in a bandwidth of B Hertz can be expressed as

❖ **Intermodulation noise:** When signals at different frequencies share the same transmission medium, the result may be intermodulation noise. The effect of intermodulation noise is to produce signals at a frequency that is the sum or difference of the two original frequencies or multiples of those frequencies. For example, the mixing of signals at frequencies f_1 and f_2 might produce energy at the frequency f_1+f_2 .

❖ **Crosstalk:** Crosstalk has been experienced by anyone who, while using the telephone, has been able to hear another conversation; it is an unwanted coupling between signal paths. It can occur by electrical coupling between nearby twisted pairs or, rarely coax cable lines carrying multiple signals. Crosstalk can also occur when microwave antennas pick up unwanted signals.

Crosstalk



- **NEXT (near-end crosstalk)**
 - interference in a wire at the transmitting end of a signal sent on a different wire
- **FEXT (far-end crosstalk)**
 - interference in a wire at the receiving end of a signal sent on a different wire

❖ **Impulse noise:** Impulse noise is non continuous, consisting of irregular pulses or noise spikes of short duration and of relatively high amplitude. It is generated from a variety of causes, including external electromagnetic disturbances, such as lightning, and faults and flaws in the communications system.

Signal-to-noise ratio (SNR)

SNR is the ratio of two powers, it is often described in decibel units, SNR_{dB}, defined as

$$\text{SNR}_{\text{dB}} = 10 \log_{10} \text{SNR}$$

$$\text{Signal TO Noise Ratio} = 10 \log \frac{\text{Signal power}}{\text{Noise Power}} = 10 \log \frac{P_s}{P_n}$$

A receiver has an input signal power of $1.2\mu\text{W}$. The noise power is $0.80\mu\text{W}$. What is the signal to noise ratio?

- Signal to Noise Ratio = $10 \log (1.2/0.8)$
= $10 \log 1.5$
= $10 (0.176)$
= 1.76 dB

Bit error rate(BER)



- The BER (Bit Error Rate) is the probability of a single bit being corrupted in a define time interval
- BER of 10^{-5} means on average 1 bit in 10^{-5} will be corrupted
 - A BER of 10^{-5} over voice-graded line is typical.
 - BERs of less than 10^{-6} over digital communication is common.

Bit error rate(BER)



❖ Several factors that effect BER:

- Bandwidth
- S/N
- Transmission medium
- Transmission distance
- Environment
- Performance of transmitter and receiver

Effect of noise practice



- E_b/N_0 = signal energy to noise energy ratio

$$\begin{aligned}\frac{E_b}{N_0} &= \frac{S^*W}{R^*N} = \frac{S^*W}{N^*R} \\ &= S/N + 10 \log W - 10 \log R \quad (\text{dB})\end{aligned}$$

S= signal power in watts

R= data rate

W= bandwidth

N= noise power in received signal

Channel capacity

- ❖ The maximum rate at which data can be transmitted over a given communication path, or channel, under given conditions, is referred to as the channel capacity.
- ❖ There are four concepts that relate to one another
 - ❖ **Data rate:** The rate, in bits per second (bps), at which data can be communicated.
 - ❖ **Bandwidth:** The bandwidth of the transmitted signal as constrained by the transmitter and the nature of the transmission medium, expressed in cycles per second, or Hertz
 - ❖ **Noise:** The average level of noise over the communications path
 - ❖ **Error rate:** The rate at which errors occur, where an error is the reception of a 1 when a 0 was transmitted or the reception of a 0 when a 1 was transmitted.

Signal-to-noise ratio (SNR)

The signal-to-noise ratio is defined as

$$\text{SNR} = \frac{\text{Average signal power}}{\text{Average noise power}}$$

We need to consider the average signal power and the average noise power because these may change with time. Figure 3.30 shows the idea of SNR.

SNR is actually the ratio of what is wanted (signal) to what is not wanted (noise). A high SNR means the signal is less corrupted by noise; a low SNR means the signal is more corrupted by noise.

Because SNR is the ratio of two powers, it is often described in decibel units, SNR_{dB}, defined as

$$\text{SNR}_{\text{db}} = 10 \log_{10} \text{SNR}$$

DATA RATE LIMITS



- ❖ A very important consideration in data communications is how fast we can send data, in bits per second. over a channel.
- ❖ Data rate depends on three factors:
 1. The bandwidth available
 2. The level of the signals we use
 3. The quality of the channel (the level of noise)

Two theoretical formulas were developed to calculate the data rate: one by Nyquist for a noiseless channel. Another by Shannon for a noisy channel.

Nyquist Bandwidth



- ❖ In noiseless environment, the limitation on data rate is simply the bandwidth of the signal. A formulation of this limitation, due to Nyquist, states that “ if the rate of signal transmission is $2B$, then a signal with frequencies no greater than B is sufficient to carry the signal rate”.
- ❖ Given a bandwidth of B , the highest signal rate that can be carried is $2B$. $C = 2 \times B \times \log_2 M$

Where C is BitRate or capacity , B is Bandwidth , M is number of levels.

Nyquist Bandwidth



- ❖ Consider a noiseless channel with a bandwidth of 3000 Hz transmitting a signal with two signal levels. What is the maximum bit rate?

$$\text{BitRate} = 2 \times 3000 \times \log_2 2 = 6000 \text{ bps}$$

Shannon's Law

The maximum data rate of a noisy channel whose bandwidth W Hz, and whose signal-to-noise ratio is S/N , is given by

$$C = W \log_2 \left(1 + \frac{S}{N} \right)$$

W = bandwidth in Hz

S = average signal power in watts

N = random noise power in watts

Let $W = 3300 - 300$ Hz = 3000 Hz

Assume a typical decibel ratio of 30 dB, thus $S/N = 1000$

$$\begin{aligned} C &= 3000 \times \log_2 (1000) \\ &\sim 30 \text{ Kbps} \end{aligned}$$

Claud Shannon carried Nyquist's work further and extended it to the case of a channel subject to random noise. Shannon's theorem give the theoretical upper bound to the capacity of a link as a function of the signal-to-noise ratio, measured in dB.

As an example, consider a voice channel has a bandwidth 3000 Hz and transmit data with normally has S/N = 30 dB or 1000.

$$\begin{aligned}C &= 3000 \log_2(1+1000) \\&= 29,897 \text{ bps}\end{aligned}$$

This is the limit of today's 28.8-Kbps modems. Higher data rates are achieved if the quality (SNR) of the phone network improves or by using compression.

Bandwidth efficiency

$$B = C/W$$

Bandwidth of the channel
Channel capacity

- Typical values range from 0.25 to 3.0 bps Hz⁻¹



Noise figure

I/P → Comm Device → O/P

$$F = \frac{\text{SNR}_{\text{in}}}{\text{SNR}_{\text{out}}}$$

where SNR_{in} and SNR_{out} are the input and output signal-to-noise ratios

Noise-equivalent temperature

It is a reference measurement between a minimum noise level due to thermal noise and total (ext.+ int.) noise.

$$T_{eq} = T_o (F-1)$$

DATA RATE LIMITS

- ❖ It is very important in data communications that how fast we can send data, in bits per second over a channel.
- ❖ Data rate depends on three factors:
 1. The bandwidth available
 2. The level of the signals we use
 3. The quality of the channel (the level of noise)
- ❖ Two theoretical formulas were developed to calculate the data rate: one by **Nyquist** for a noiseless channel. another by **Shannon** for a noisy channel.

Noiseless Channel: Nyquist Bit Rate

$$\text{BitRate} = 2 \times \text{bandwidth} \times \log_2 L$$

- Where L is the number of signal levels used to represent data, and BitRate is the bit rate in bits per second.
- Example: Consider a noiseless channel with a bandwidth of 3000 Hz transmitting a signal with two signal levels. Calculate the maximum bit rate.



❖ BitRate = $2 \times 3000 \times \log_2 2 = 6000$ bps

Example 2



- ❖ We need to send 265 kbps over a noiseless channel with a bandwidth of 20 kHz. How many signallevels do we need?



- ❖ $265,000 = 2 \times 20,000 \times \log_2 L$
- ❖ $\log_2 L = 6.625$ $L = 2^{6.625} = 98.7$ levels
- ❖ Since this result is not a power of 2, we need to either increase the number of levels or reduce the bit rate. If we have 128 levels, the bit rate is 280 kbps. If we have 64 levels, the bit rate is 240 kbps.

Noisy Channel: Shannon Capacity

- ❖ Capacity =bandwidth X $\log_2 (1 + \text{SNR})$
- ❖ A telephone line normally has a bandwidth of 3000 Hz (300 to 3300 Hz) assigned for data communications. The signal-to-noise ratio is usually 3162. For this channel the capacity is calculated as



$$\text{C} = B \log_2 (1 + \text{SNR}) = 3000 \log_2 (1 + 3162) = 3000 \log_2 3163$$

$$= 3000 \times 11.62 = 34,860 \text{ bps}$$

PERFORMANCE



- ❖ Bandwidth
- ❖ Throughput
- ❖ Latency (Delay)
- ❖ Jitter

Bandwidth

- ❖ Two different measuring values: bandwidth in hertz and bandwidth in bits per second.
- ❖ *Bandwidth in Hertz*: Bandwidth in hertz is the range of frequencies contained in a composite signal or the range of frequencies a channel can pass. For example, we can say the bandwidth of a subscriber telephone line is 4 kHz.
- ❖ *Bandwidth in Bits per Seconds*: refer to the number of bits per second that a channel, a link, or even a network can transmit. For example, one can say the bandwidth of a Fast Ethernet network (or the links in this network) is a maximum of 100 Mbps.

Throughput

- ❖ The throughput is a measure of how fast we can actually send data through a network. A link may have a bandwidth of B bps, but we can only send T bps through this link with T always less than B .
- ❖ In other words, the bandwidth is a potential measurement of a link; the throughput is an actual measurement of how fast we can send data.

❖ A network with bandwidth of 10 Mbps can pass only an average of 12,000 frames per minute with each frame carrying an average of 10,000 bits. What is the throughput of this network?



We can calculate the throughput as

$$\text{Throughput} = \frac{12,000 \times 10,000}{60} = 2 \text{ Mbps}$$

Latency (Delay)

- ❖ The latency or delay defines how long it takes for an entire message to completely arrive at the destination from the time the first bit is sent out from the source. We can say that latency is made of four components: propagation time, transmission time, queuing time and processing delay.
- ❖ Latency = propagation time + transmission time + queuing time + processing delay



- ❖ 1. *Propagation Time*: measures the time required for a bit to travel from the source to the destination. The propagation time is calculated by dividing the distance by the propagation speed.

$$\text{Propagation time} = \frac{\text{Distance}}{\text{Propagation speed}}$$

What is the propagation time if the distance between the two points is 12,000 km? Assume the propagation speed to be 2.4×10^8 mls in cable.



$$\text{Propagation time} = \frac{12000 \times 1000}{2.4 \times 10^8} = 50 \text{ ms}$$

Jitter



- ❖ Jitter is a problem if different packets of data encounter different delays and the application using the data at the receiver site is time-sensitive (audio and video data, for example).
- ❖ If the delay for the first packet is 20 ms, for the second is 45 ms, and for the third is 40 ms, then the real-time application that uses the packets endures jitter.

Transmission time

The time required for transmission of
message depends on the size of the
message and the bandwidth of the
channel.

$$\text{Transmission time} = \frac{\text{Message size}}{\text{Bandwidth}}$$

Data Link Layer

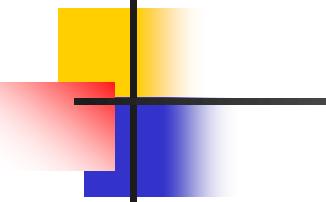
Module 3

Overview

- Error Detection- CRC, Checksum
- Correction – Hamming Code
- Flow control mechanism
 - Sliding Window Protocol, GoBack - N
 - Selective Repeat
- Multiple access Aloha - Slotted Aloha
 - CSMA, CSMA/CD
- Multiple Access Networks (IEEE 802.3)
- Token Ring(IEEE 802.5)
- Wireless Networks (IEEE 802.11, 802.15)

Basic concepts

- ★ Networks must be able to transfer data from one device to another with complete accuracy.
- ★ Data can be corrupted during transmission.
- ★ For reliable communication, errors must be detected and corrected.
- ★ **Error detection and correction** are implemented either at the **data link layer** or the **transport layer** of the OSI model.



Note

**Data can be corrupted
during transmission.**

**Some applications require that
errors be detected and corrected.**

10-1 INTRODUCTION

Let us first discuss some issues related, directly or indirectly, to error detection and correction.

Topics discussed in this section:

Types of Errors

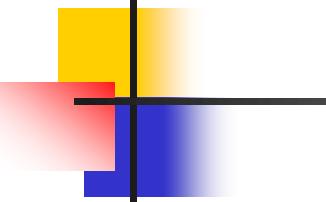
Redundancy

Detection Versus Correction

Forward Error Correction Versus Retransmission

Coding

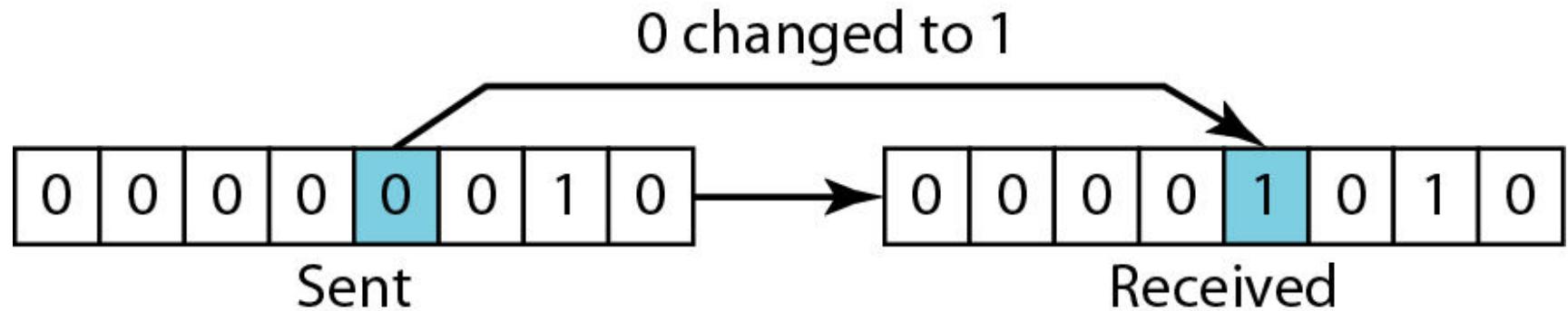
Modular Arithmetic

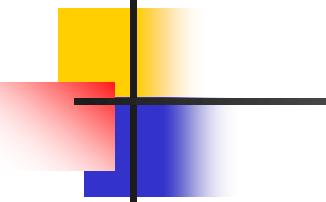


Note

In a single-bit error, only 1 bit in the data unit has changed.

Single-bit error

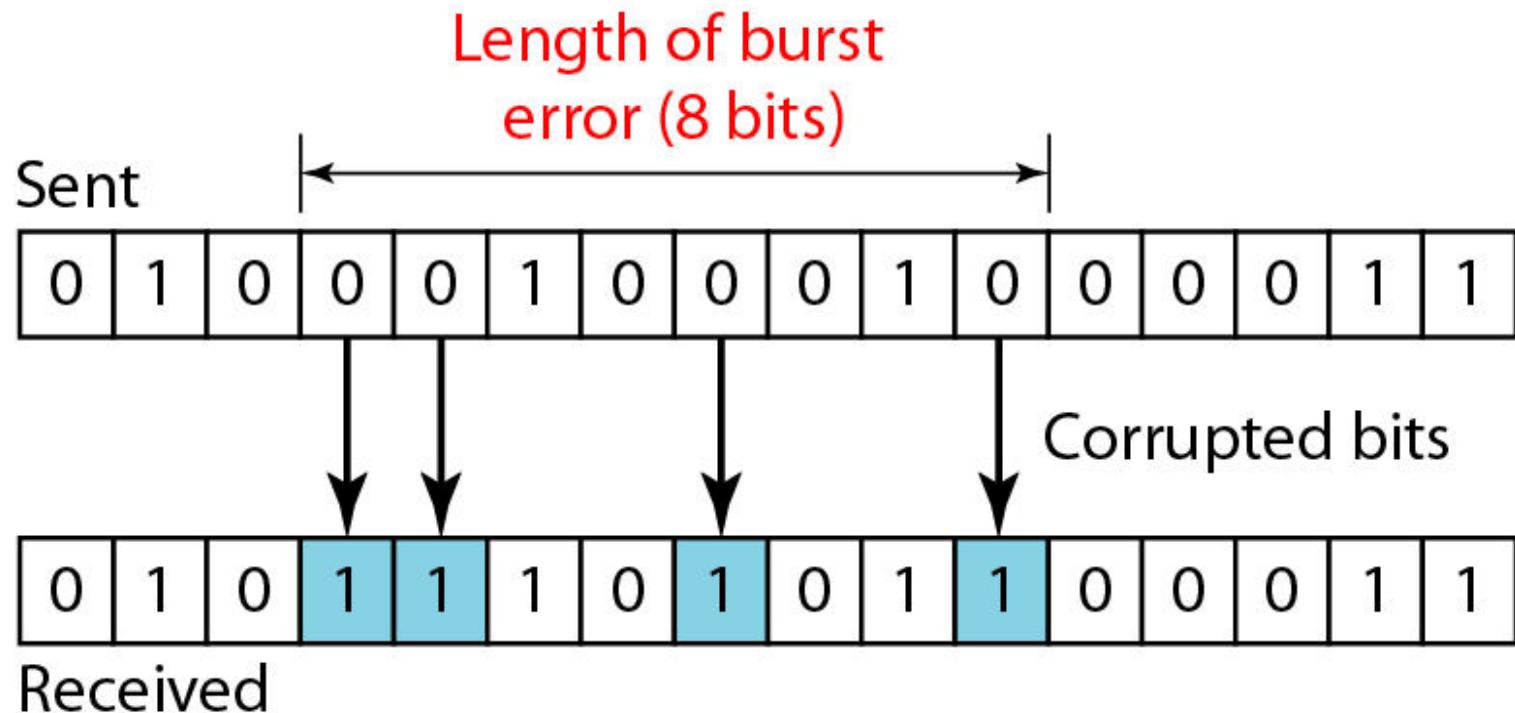


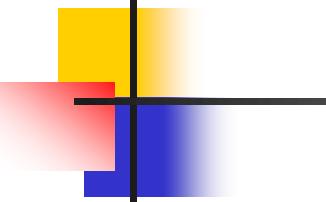


Note

A burst error means that 2 or more bits in the data unit have changed.

Burst error of length 8

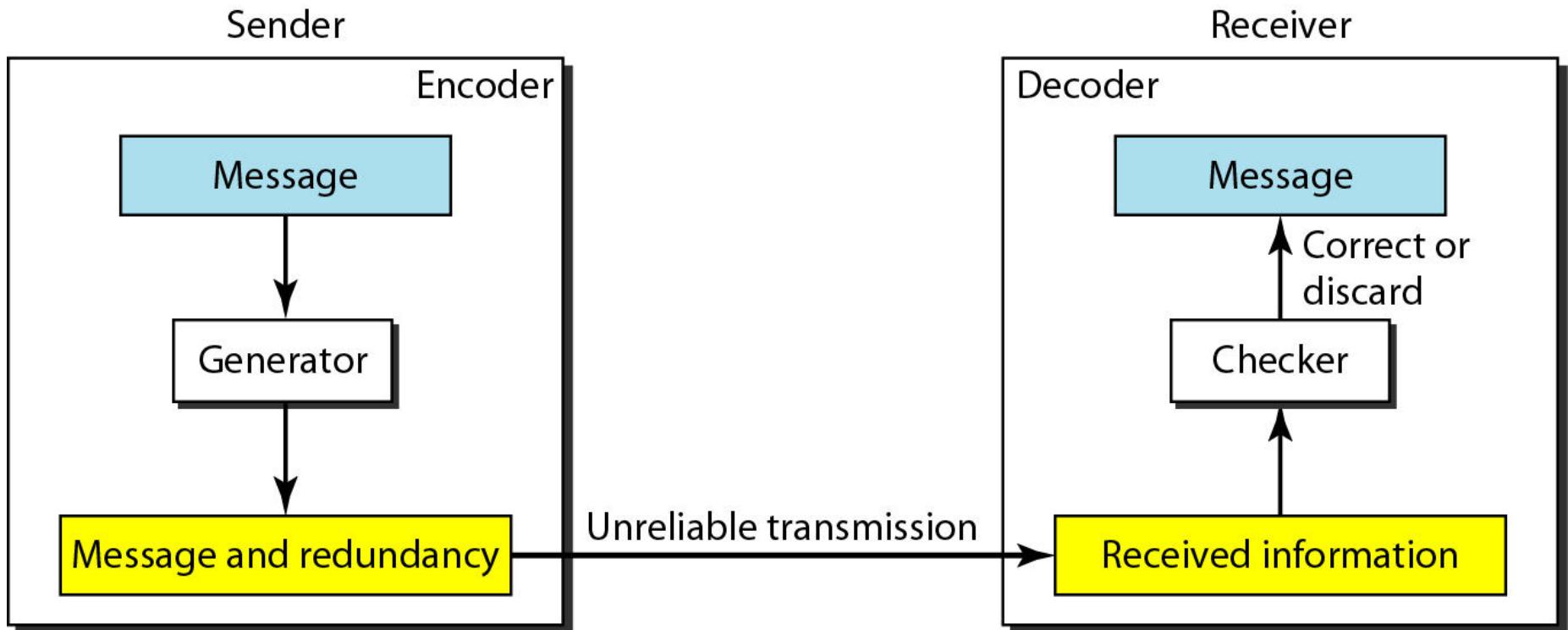




Note

To detect or correct errors, we need to send extra (redundant) bits with data.

The structure of encoder and decoder



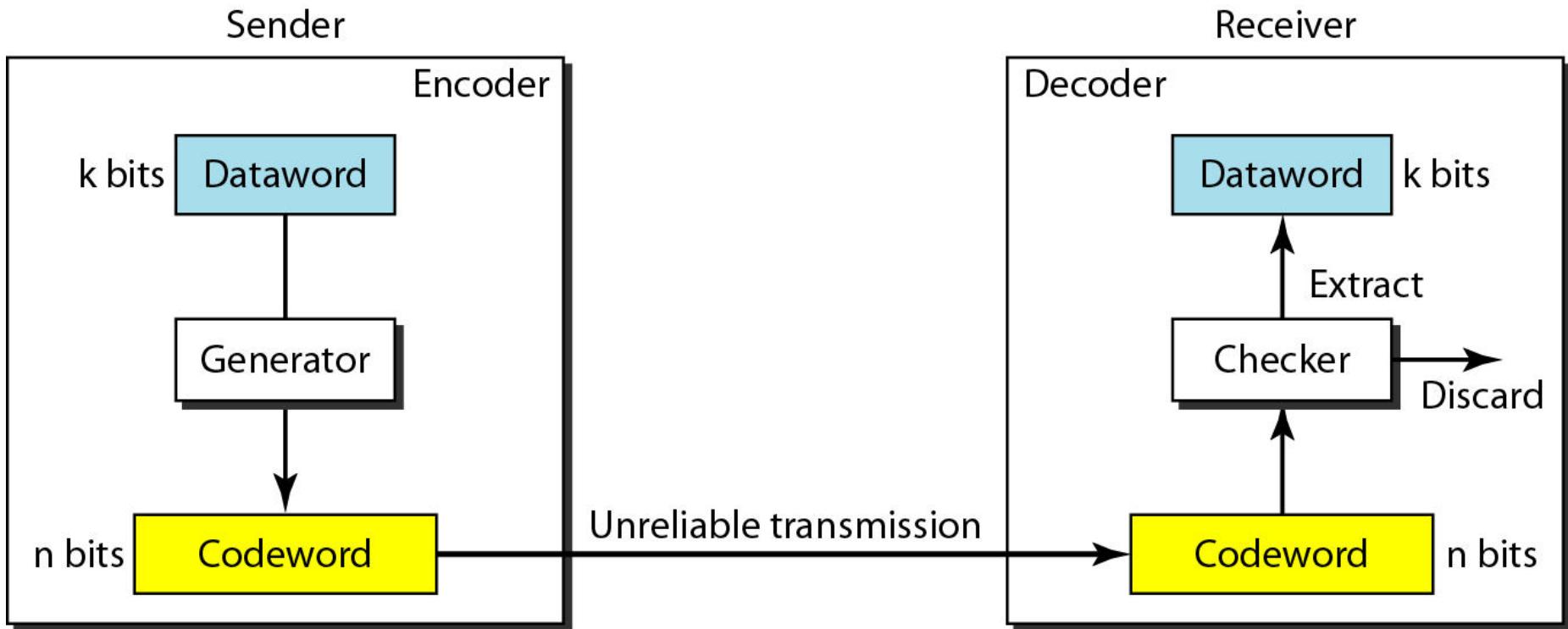
BLOCK CODING

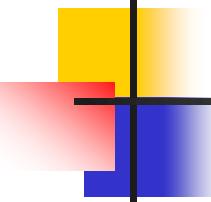
*In block coding, we divide our message into blocks, each of k bits, called **datawords**. We add r redundant bits to each block to make the length $n = k + r$. The resulting n -bit blocks are called **codewords**.*

Error Detection

- Enough redundancy is added to detect an error.
- The receiver knows an error occurred but does not know which bit(s) is(are) in error.
- Has less overhead than error correction.

Process of error detection in block coding



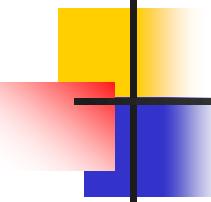


Example

Let us assume that $k = 2$ and $n = 3$.

Assume the sender encodes the dataword 01 as 011 and sends it to the receiver. Consider the following cases:

- 1. The receiver receives 011. It is a valid codeword. The receiver extracts the dataword 01 from it.*



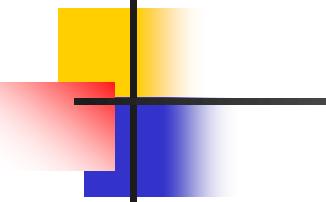
Example (continued)

- 2.** *The codeword is corrupted during transmission, and 111 is received. This is not a valid codeword and is discarded.*

- 3.** *The codeword is corrupted during transmission, and 000 is received. This is a valid codeword. The receiver incorrectly extracts the dataword 00. Two corrupted bits have made the error undetectable.*

A code for error detection

<i>Datawords</i>	<i>Codewords</i>
00	000
01	011
10	101
11	110



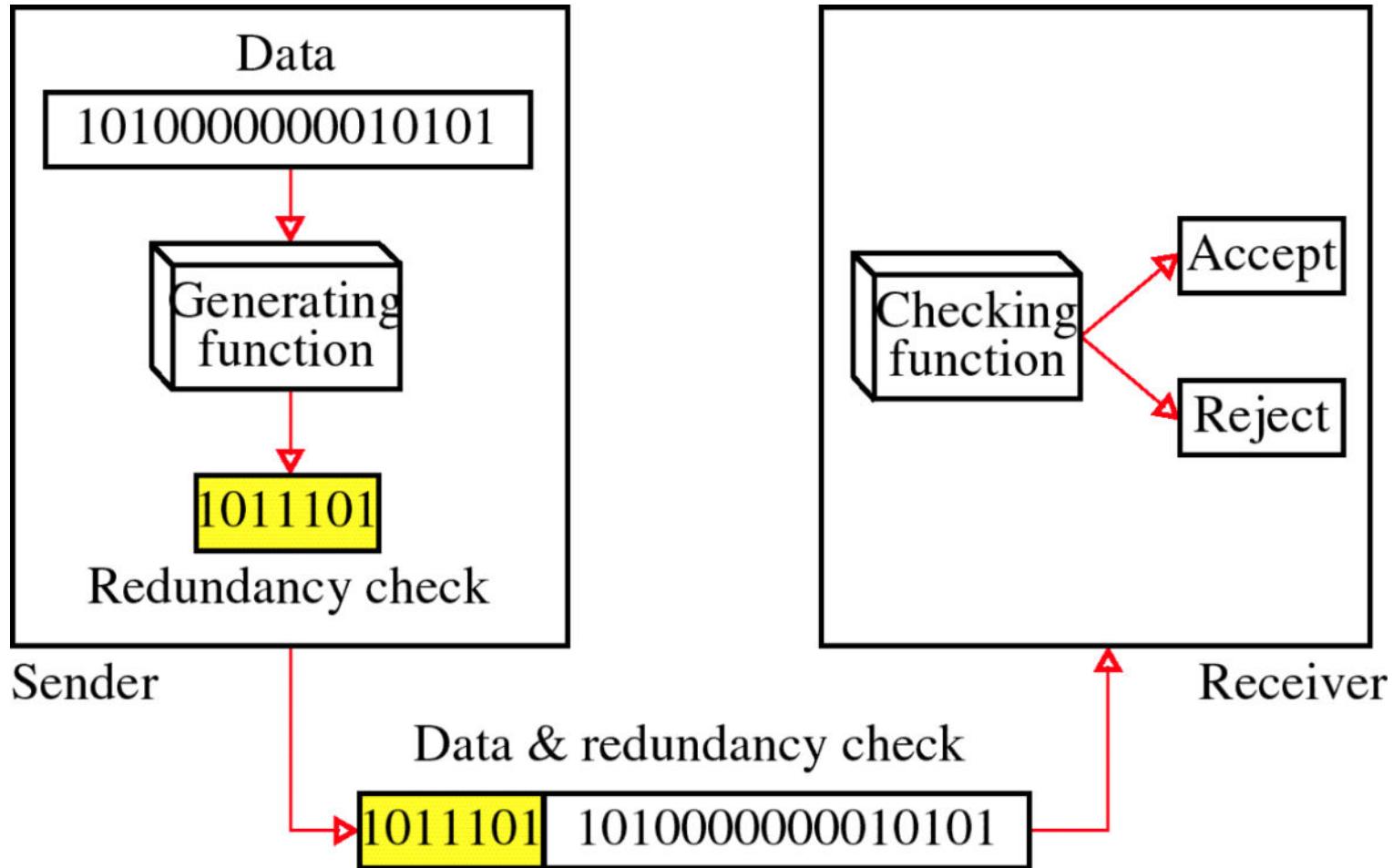
Note

An error-detecting code can detect only the types of errors for which it is designed; other types of errors may remain undetected.

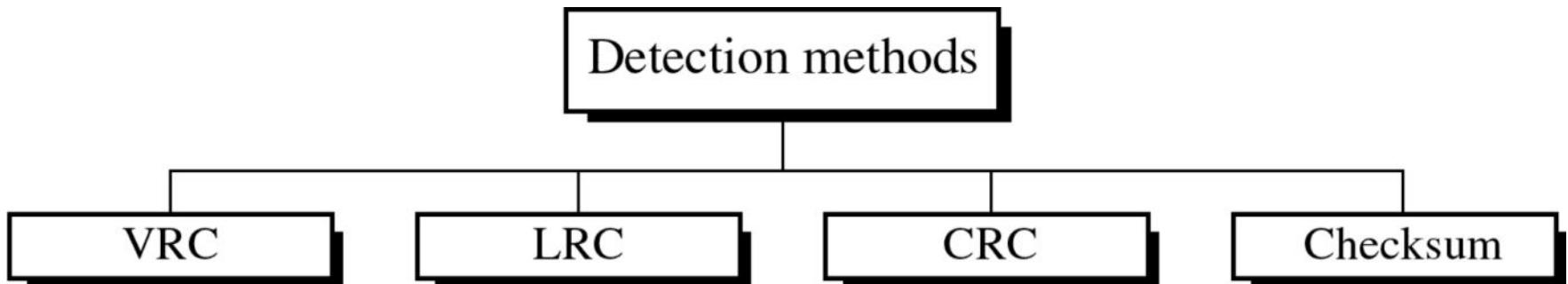
Error detection

- Error detection means to decide whether the received data is correct or not without having a copy of the original message.
- Error detection **uses the concept of redundancy**, **which means** adding extra bits for detecting errors at the destination.

Redundancy



Four types of redundancy checks are used in data communications

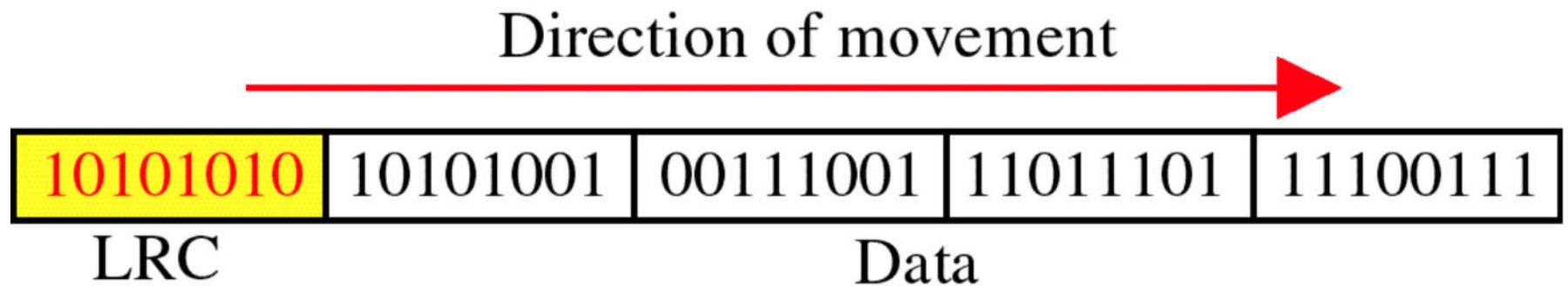


Performance

- It can detect single bit error
- It can detect burst errors only if the total number of errors is odd.

Longitudinal Redundancy Check

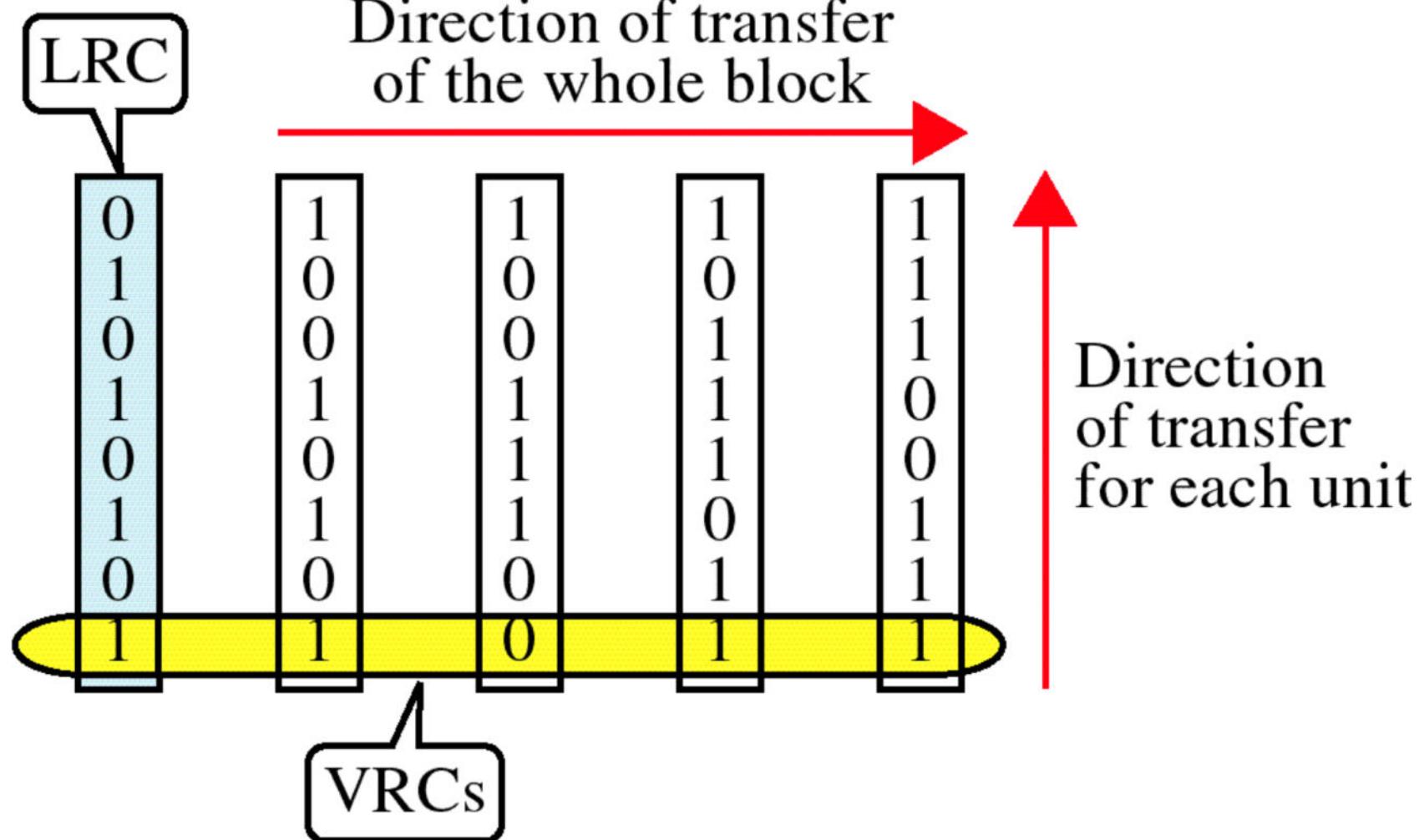
LRC



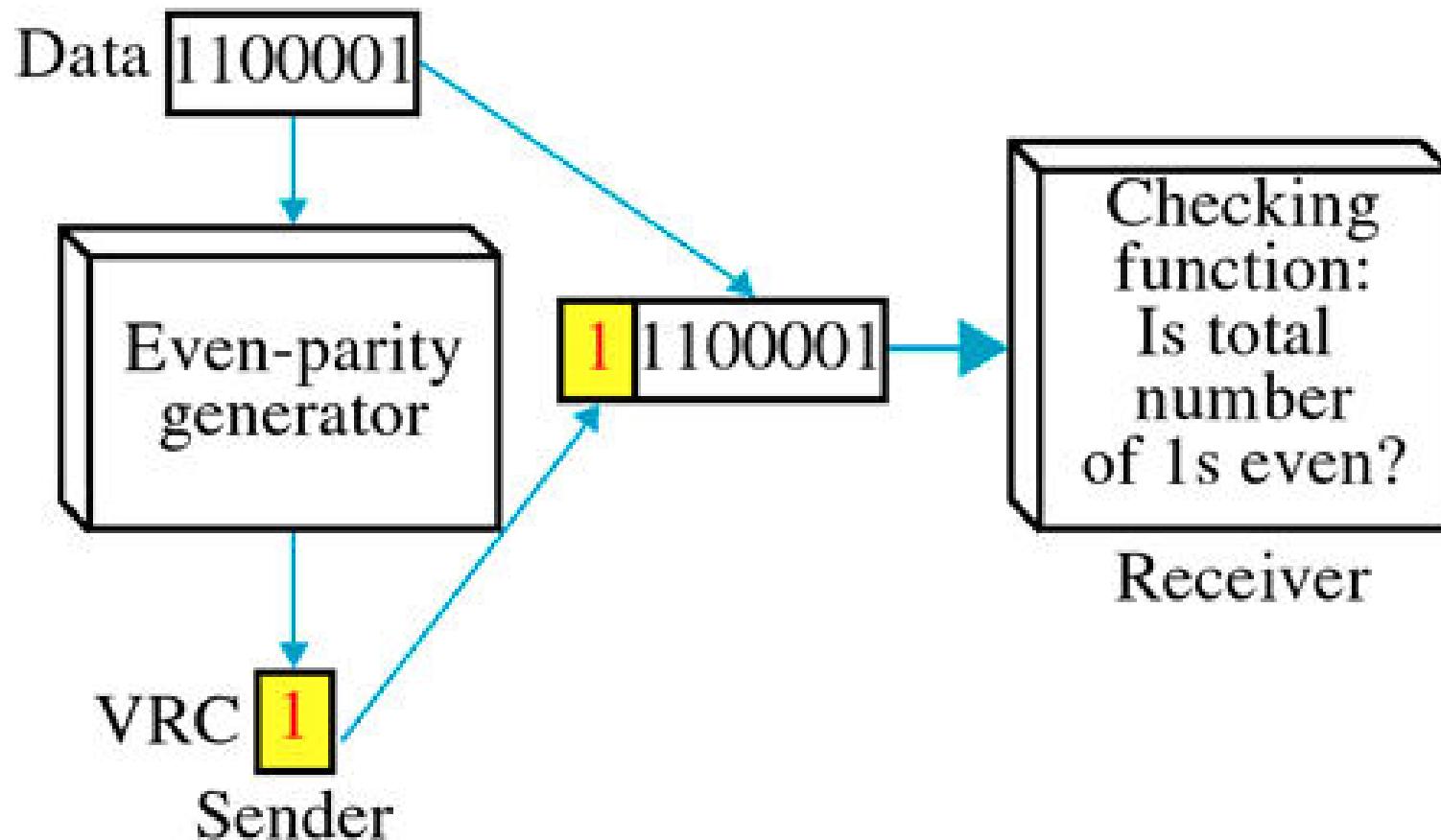
Performance

- ➔ LCR increases the likelihood of detecting burst errors.
- ➔ If two bits in one data units are damaged and two bits in exactly the same positions in another data unit are also damaged, the LRC checker will not detect an error.

VRC and LRC



Vertical Redundancy Check VRC



Original Data

10011001	11100010	00100100	10000100
----------	----------	----------	----------

Row parities

10011001	0
11100010	0
00100100	0
10000100	0
11011011	0

Column
parities



100110010	111000100	001001000	100001000	110110110
-----------	-----------	-----------	-----------	-----------

Data to be sent

2D parity check

- 2D parity check can detect and correct all 1 bit errors.

0	1	1	0	1	0	0	1
1	0	1	1	0	1	0	0
0	0	0	0	1	1	0	1
1	1	1	0	1	0	1	1
1	0	0	1	0	1	1	0
1	0	0	0	1	1	0	1

Error bit

2D parity check

- 2D parity check can detect and correct all 1 bit errors.

The diagram shows a 7x9 grid of binary digits (0s and 1s). Several bits are circled in red, indicating errors. A red arrow labeled 'e' points to the circled bit at position (5, 5). A red line labeled 'Error bit' points to the circled bit at position (3, 3). The circled bits are located at positions (5, 5), (3, 3), (6, 2), (6, 3), (6, 4), (6, 5), (6, 6), (6, 7), and (6, 8).

0	1	1	0	1	0	0	1
1	0	1	1	0	1	0	0
0	0	0	0	1	1	0	1
1	1	1	0	1	0	1	1
1	0	0	1	0	1	1	0
1	0	0	0	1	1	0	1

2D parity check

- 2D parity check can detect and correct all 1 bit errors.

0	1	1	0	1	0	0	1
1	0	1	1	0	1	0	0
0	0	0	0	1	1	0	1
1	1	1	0	1	0	1	1
1	0	0	1	0	1	1	0
1	0	0	0	1	1	0	1

Diagram illustrating a 2D parity check matrix. A red line labeled "Error bit" points to the third column of the third row, where the value is circled in red and has a red "X" drawn through it. A red wavy line also points to the same cell. A red "X" is also present at the bottom center of the diagram.

2D parity check

- 2D parity check can detect and correct all 1 bit errors.

Error bit

0	1	1	0	1	0	0	1
1	0	1	1	0	1	0	0
0	0	0	0	1	1	0	1
1	1	0	0	1	0	1	1
1	0	0	1	0	1	1	0
1	0	0	0	1	1	0	1

A red arrow points from the text "Error bit" to the value "0" in the third row, third column. A red circle highlights this "0". A red "X" is placed over the value "1" in the third row, eighth column. A red bracket highlights the entire third row.

2D parity check

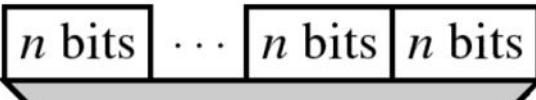
- 2D parity check can detect and correct all 1 bit errors.

Error bit

0	1	1	0	0	0	0	1
1	0	1	1	0	1	0	0
0	0	0	0	1	1	0	1
1	1	1	0	1	0	1	1
1	0	0	1	0	1	1	0
1	0	0	0	1	1	0	1

Checksum

Section K Section 1



Section 1 $n \text{ bits}$

Section 2 $n \text{ bits}$

.....

.....

Section K $n \text{ bits}$

Sum $n \text{ bits}$

Complement

.....

.....

Checksum

Sender

Section k Section 1



Checksum

Section 1 $n \text{ bits}$

Section 2 $n \text{ bits}$

.....

.....

Section K $n \text{ bits}$

Checksum $n \text{ bits}$

Sum Red Box

All 1s, accept
Otherwise, reject

Receiver

At the sender

- ⇒ The unit is divided into k sections, each of n bits.
- ⇒ All sections are added together using one's complement to get the sum.
- ⇒ The sum is complemented and becomes the checksum.
- ⇒ The checksum is sent with the data

At the receiver

- ⇒ The unit is divided into k sections, each of n bits.
- ⇒ All sections are added together using one's complement to get the sum.
- ⇒ The sum is complemented.
- ⇒ If the result is zero, the data are accepted: otherwise, they are rejected.

Performance

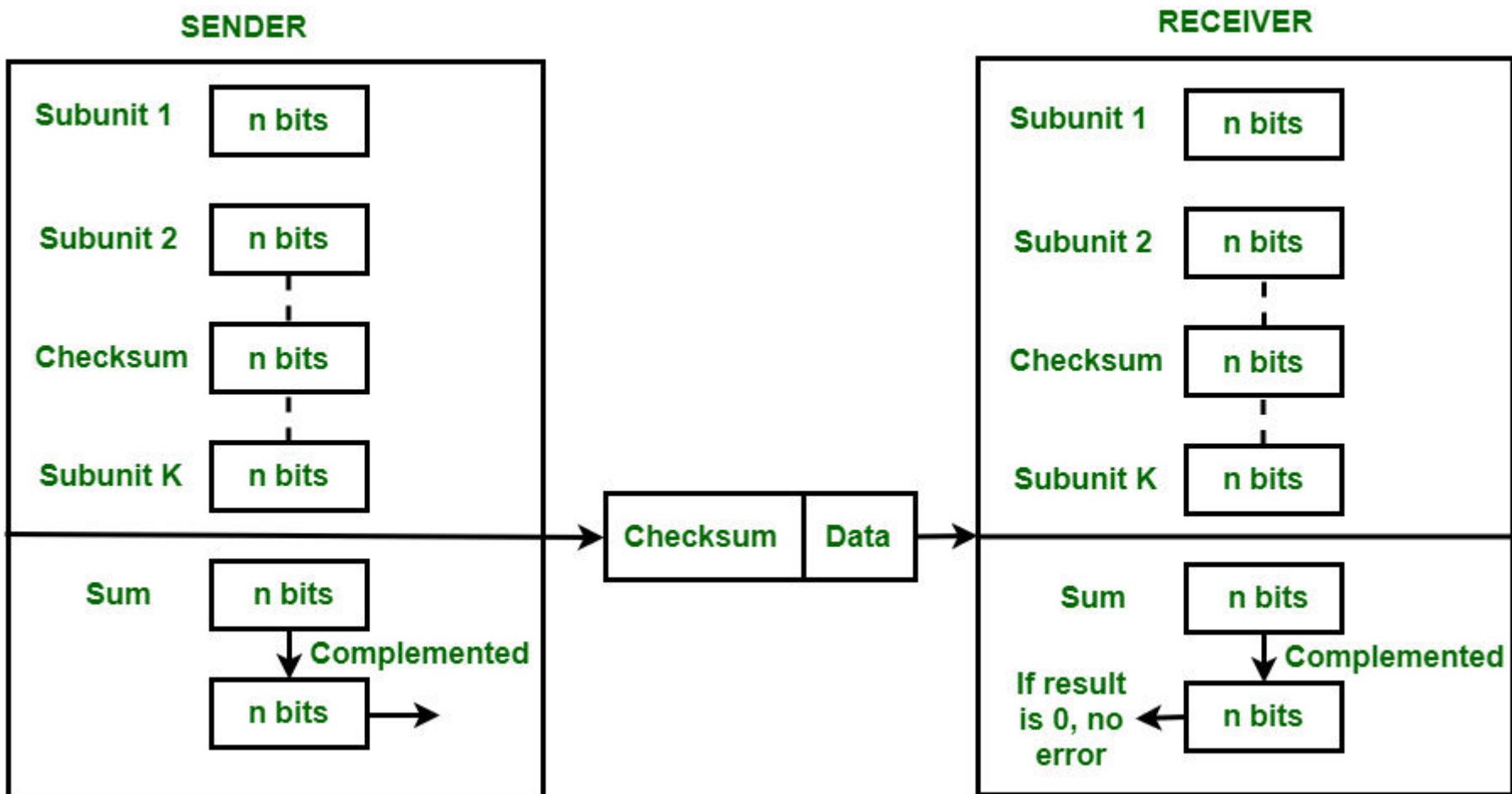
- The checksum detects all errors involving an odd number of bits.
- It detects most errors involving an even number of bits.
- If one or more bits of a segment are damaged and the corresponding bit or bits of opposite value in a second segment are also damaged, the sums of those columns will not change and the receiver will not detect a problem.

CHECKSUM

- If the set of numbers is (7, 11, 12, 0, 6), we send (7, 11, 12, 0, 6, 36), where 36 is the sum of the original numbers.
- We can make the job of the receiver easier if we send the negative (complement) of the sum, called the *checksum*. In this case, we send (7, 11, 12, 0, 6, -36). The receiver can add all the numbers received (including the checksum). If the result is 0, it assumes no error; otherwise, there is an error.

One's Complement

- How can we represent the number 21 in one's complement arithmetic using only four bits?
- **Solution**
- The number 21 in binary is 10101 (it needs five bits). We can wrap the leftmost bit and add it to the four rightmost bits. We have $(0101 + 1) = 0110$ or 6.



Check Sum Example

Original Data

10011001	11100010	00100100	10000100
----------	----------	----------	----------

1 2 3 4

$k=4, m=8$

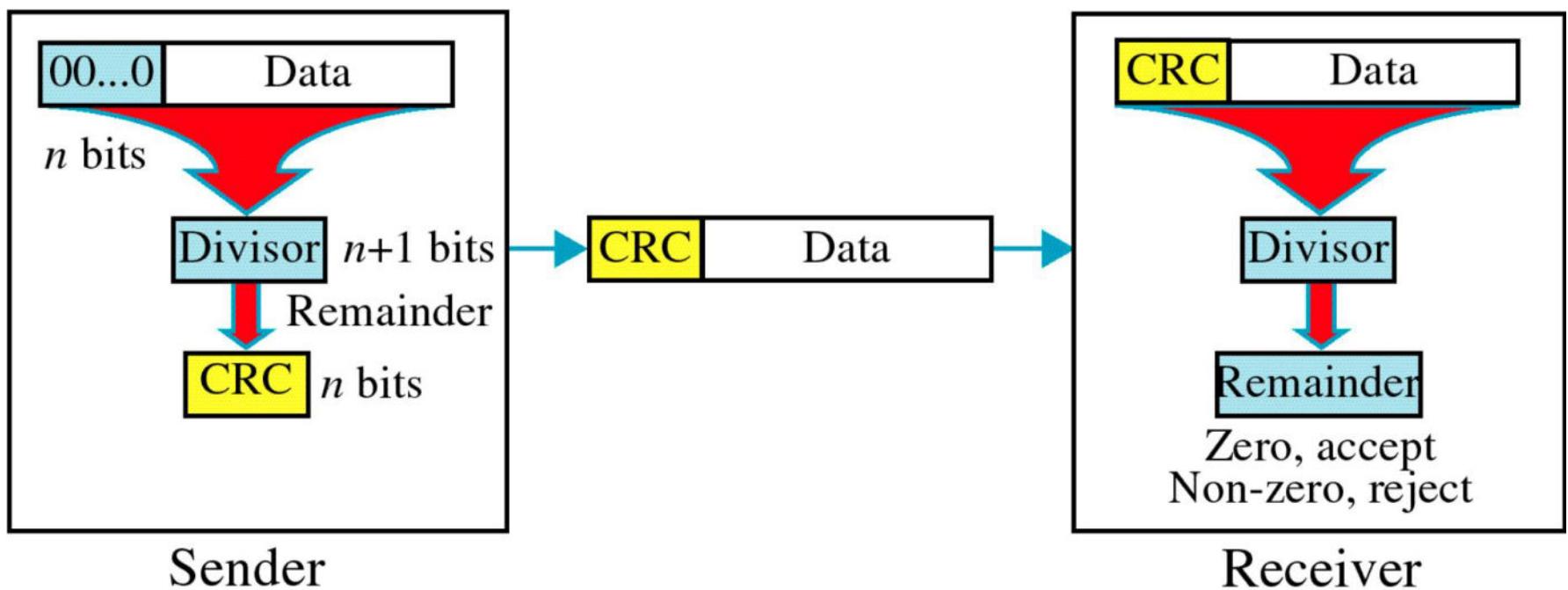
Receiver

Sender

$$\begin{array}{r} 1 \quad 10011001 \\ 2 \quad 11100010 \\ \hline 101111011 \\ \quad \quad \quad 1 \\ 01111100 \\ \hline 00100100 \\ 10100000 \\ \hline 10000100 \\ \hline 100100100 \\ \quad \quad \quad 1 \\ \hline \text{Sum: } 00100101 \\ \hline \text{CheckSum: } 11011010 \end{array}$$

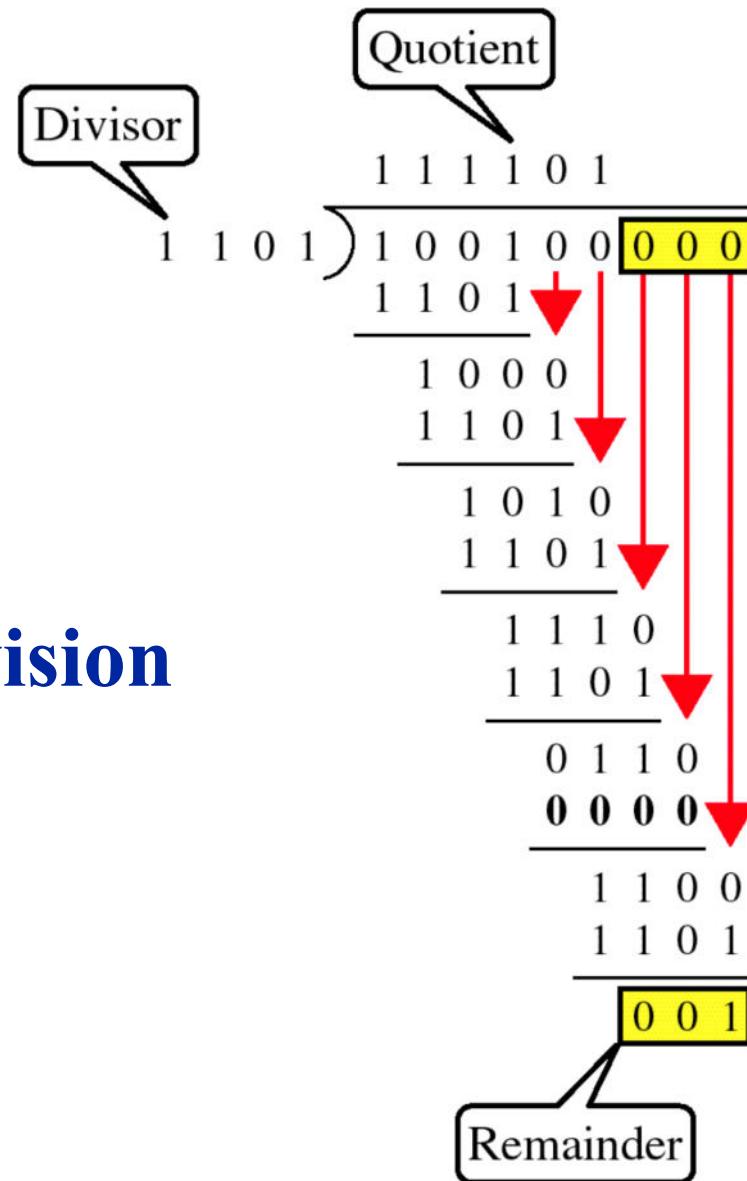
$$\begin{array}{r} 1 \quad 10011001 \\ 2 \quad 11100010 \\ \hline 01111011 \\ \quad \quad \quad 1 \\ 01111100 \\ \hline 00100100 \\ 10100000 \\ \hline 10000100 \\ \hline 00100100 \\ \quad \quad \quad 1 \\ \hline 00100101 \\ 11011010 \\ \hline \text{Sum: } 11111111 \\ \text{Complement: } 00000000 \\ \text{Conclusion: Accept Data} \end{array}$$

Cyclic Redundancy Check CRC



Cyclic Redundancy Check

- Given a k -bit frame or message, the transmitter generates an n -bit sequence, known as a *frame check sequence (FCS)*, so that the resulting frame, consisting of $(k+n)$ bits, is exactly divisible by some predetermined number.
- The receiver then divides the incoming frame by the same number and, if there is no remainder, assumes that there was no error.



Binary Division

original message

1010000

@ means X-OR

Generator polynomial

$$x^3 + 1$$

$$\rightarrow 1 \cdot x^3 + 0 \cdot x^2 + 0 \cdot x^1 + 1 \cdot x^0$$

CRC generator

1001 4-bit

If CRC generator is of n bit then append $(n-1)$ zeros in the end of original message

Sender



$$\begin{array}{r} 1001 \mid 101000000 \\ @1001 \\ \hline 001100000 \end{array}$$

$$\begin{array}{r} @1001 \\ \hline 01010000 \\ @1001 \\ \hline 00110000 \end{array}$$

$$\begin{array}{r} @1001 \\ \hline 01010 \\ @1001 \\ \hline 00110 \end{array}$$

$$\begin{array}{r} @1001 \\ \hline 0011 \\ + 011 \\ \hline 101000011 \end{array}$$

Message to be transmitted

$$\begin{array}{r} 1001 \mid 1010000011 \\ @1001 \\ \hline 0011000011 \end{array}$$

$$\begin{array}{r} @1001 \\ \hline 01010011 \\ @1001 \\ \hline 0011011 \end{array}$$

$$\begin{array}{r} @1001 \\ \hline 01001 \\ @1001 \\ \hline 0000 \end{array}$$

Receiver

Zero means data is accepted

Polynomial

$$x^7 + x^5 + x^2 + x + 1$$

Polynomial and Divisor

Polynomial

$$x^7 + x^5 + x^2 + x + 1$$

$$x^6 \quad x^4 \quad x^3$$

1 0 1 0 0 1 1 1

Divisor

Standard Polynomials

Name	Polynomial	Application
CRC-8	$x^8 + x^2 + x + 1$	ATM header
CRC-10	$x^{10} + x^9 + x^5 + x^4 + x^2 + 1$	ATM AAL
CRC-16	$x^{16} + x^{12} + x^5 + 1$	HDLC
CRC-32	$x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$	LANs

CRC-12

$$x^{12} + x^{11} + x^3 + x + 1$$

CRC-16

$$x^{16} + x^{15} + x^2 + 1$$

CRC-ITU

$$x^{16} + x^{12} + x^5 + 1$$

CRC-32

$$x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$$

Error Correction

It can be handled in two ways:

- 1) receiver can have the sender retransmit the entire data unit.
- 2) The receiver can use an error-correcting code, which automatically corrects certain errors.

Single-bit error correction

To correct an error, the receiver reverses the value of the altered bit. To do so, it must know which bit is in error.

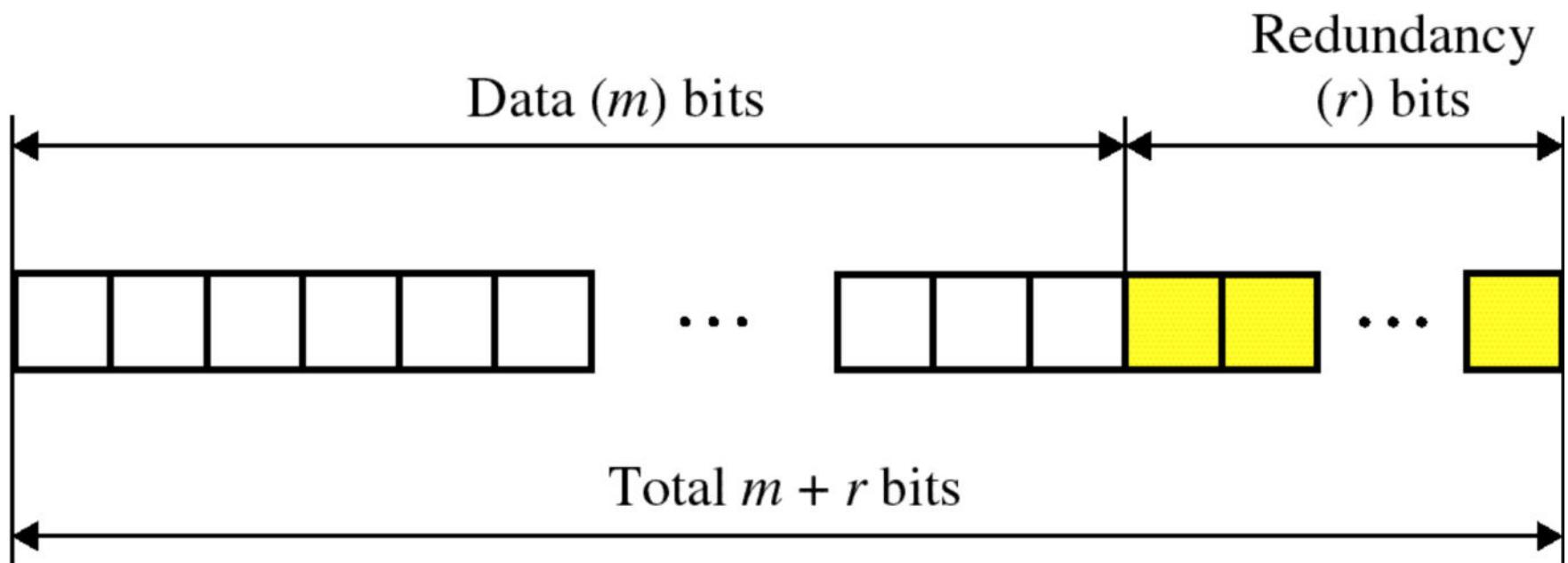
Number of redundancy bits needed

- Let data bits = m
 - Redundancy bits = r
- ∴ Total message sent = $m+r$

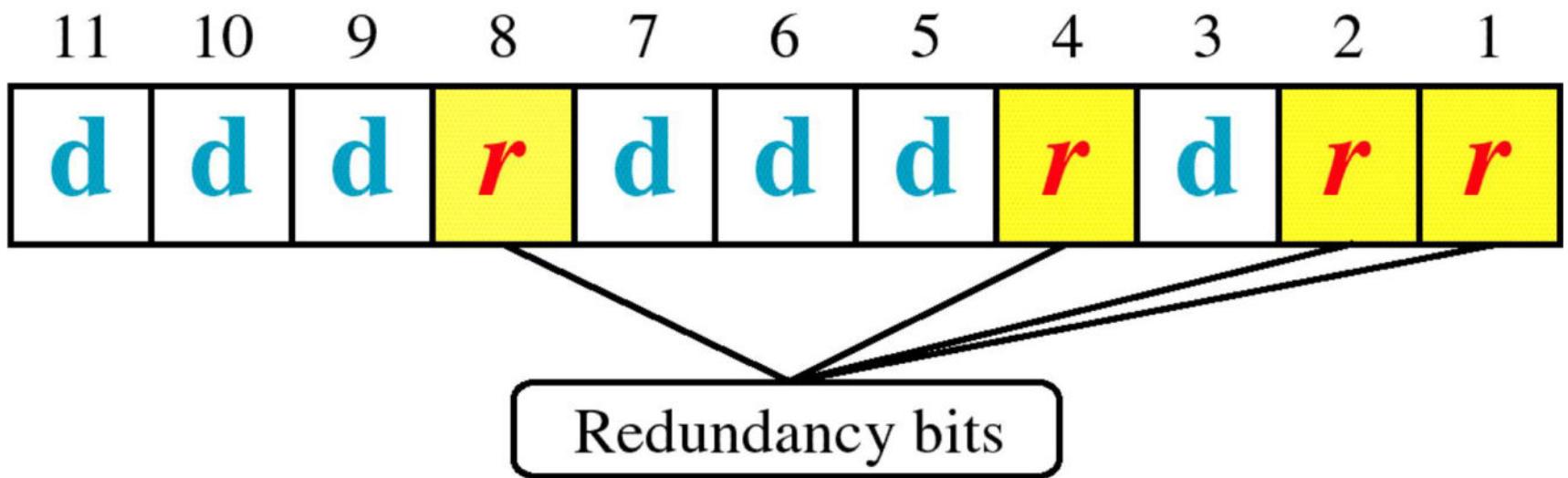
The value of r must satisfy the following relation:

$$2^r \geq m+r+1$$

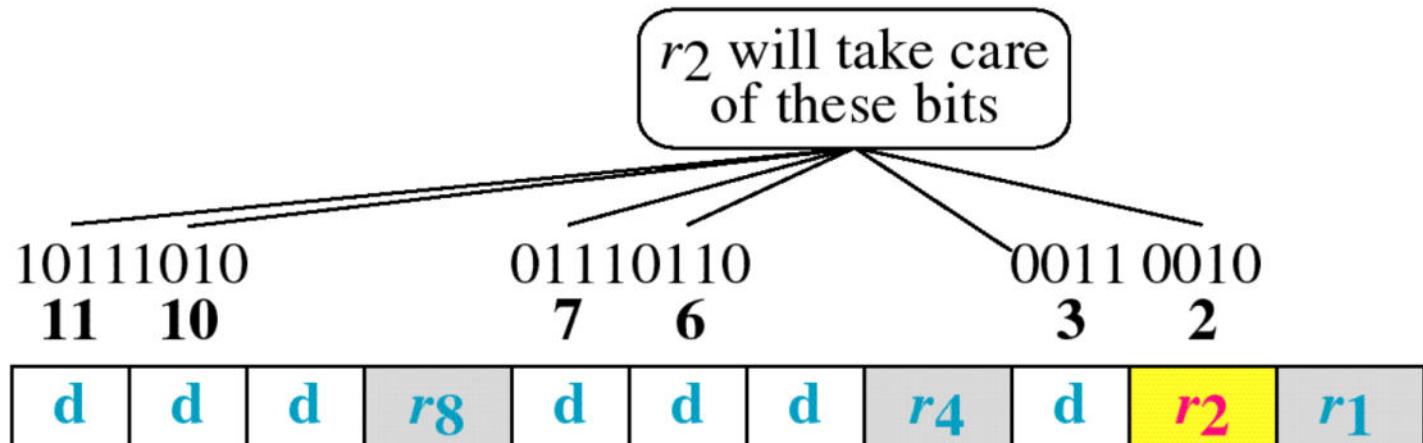
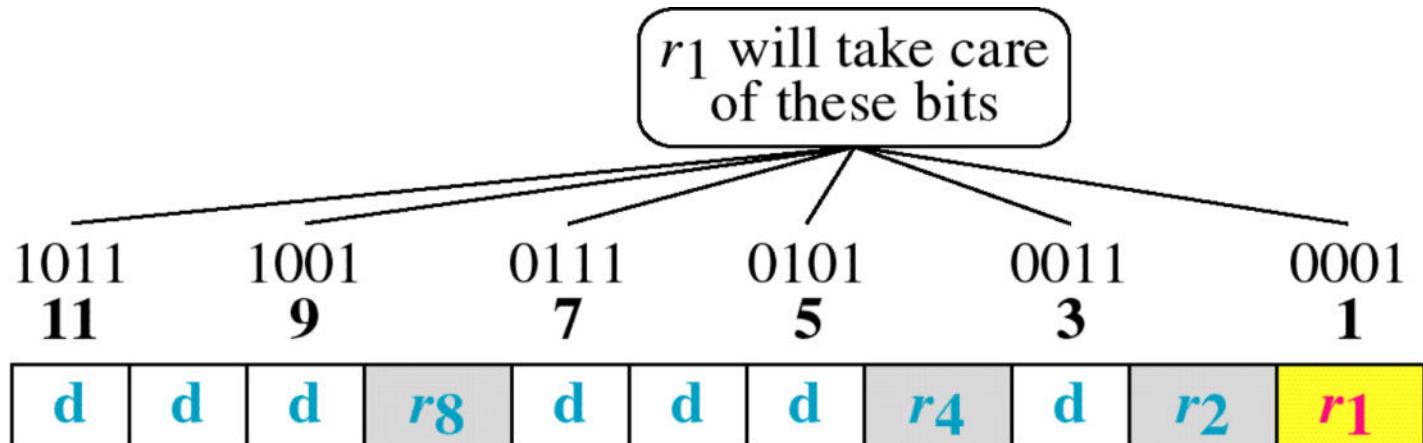
Error Correction



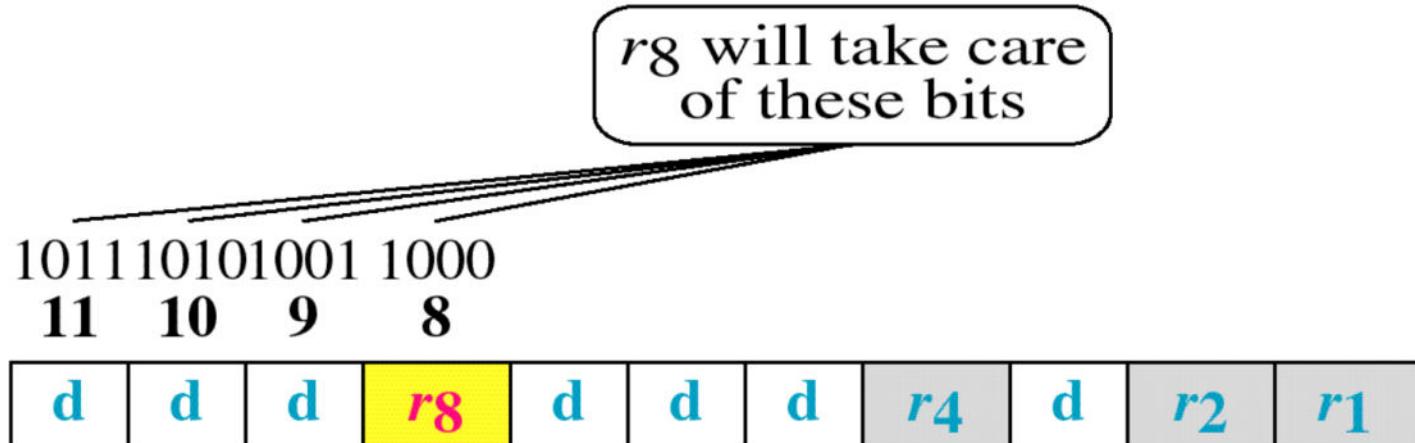
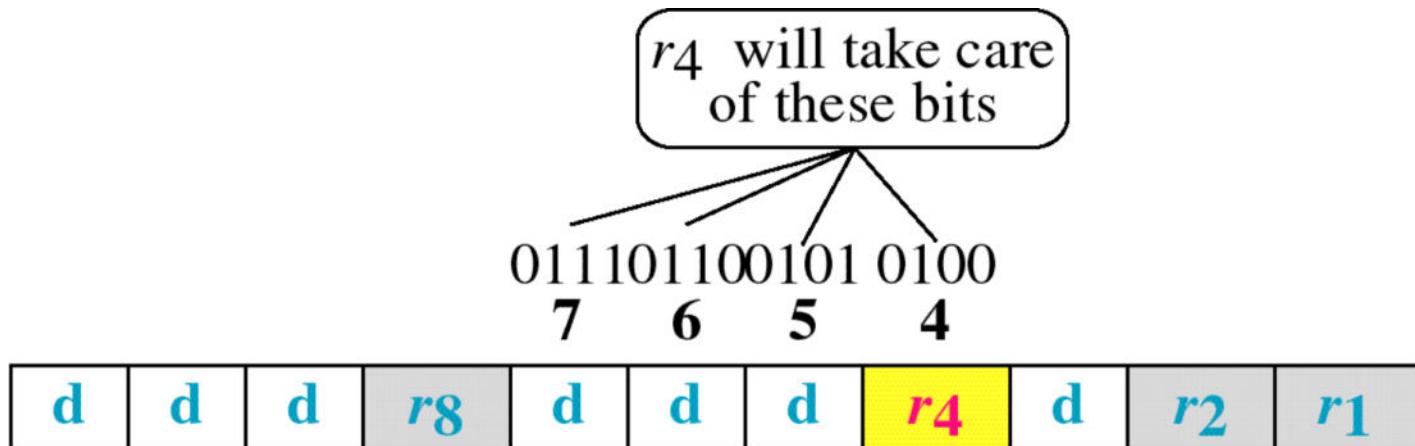
Hamming Code



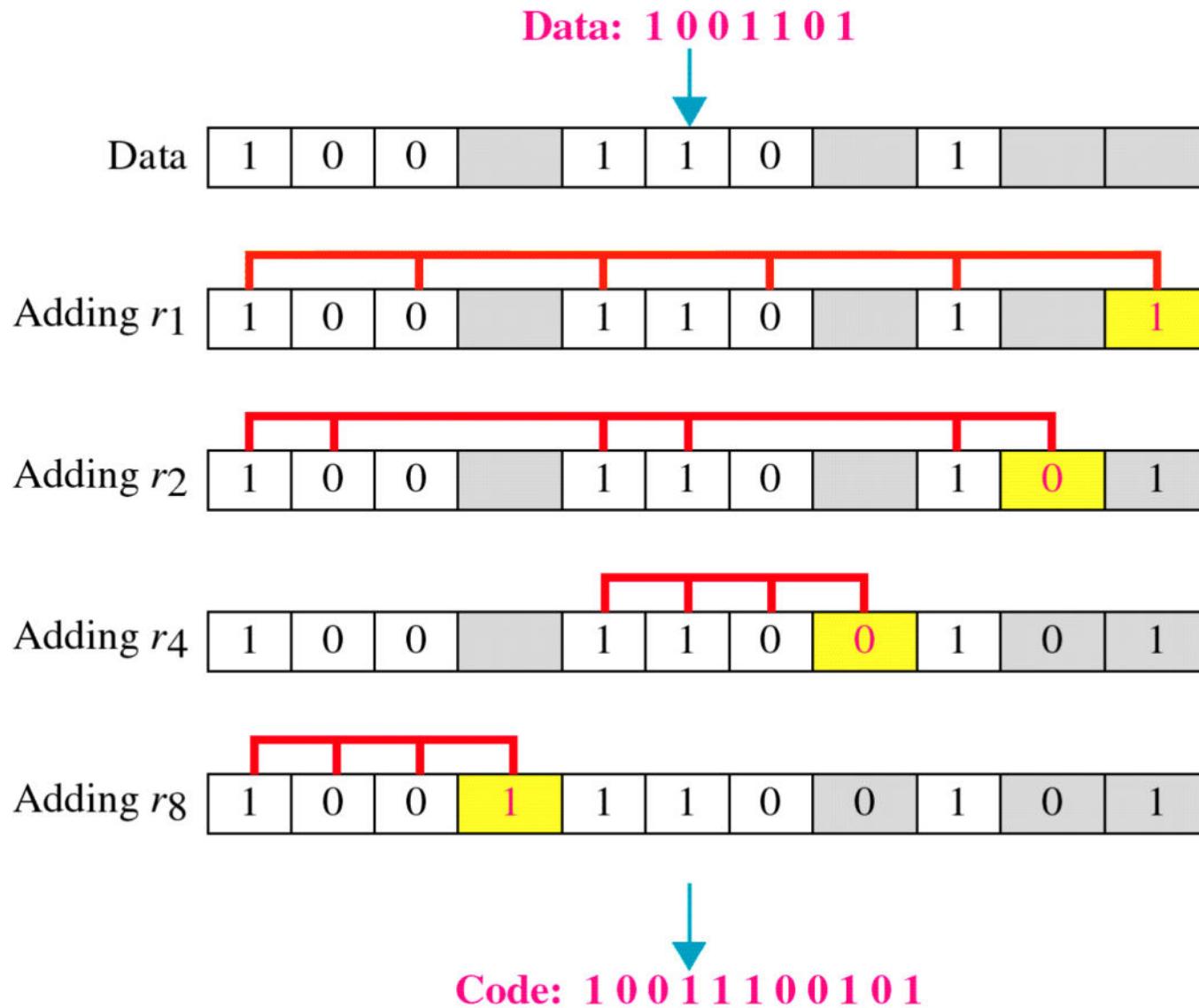
Hamming Code



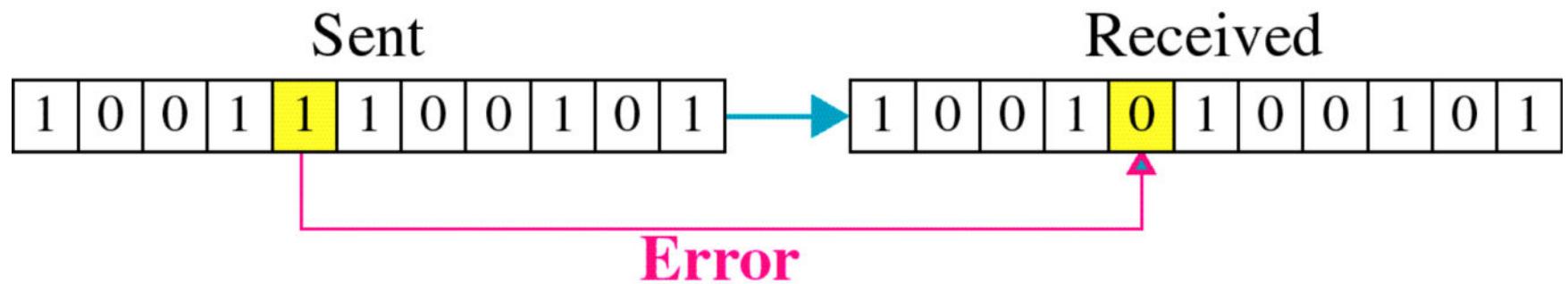
Hamming Code



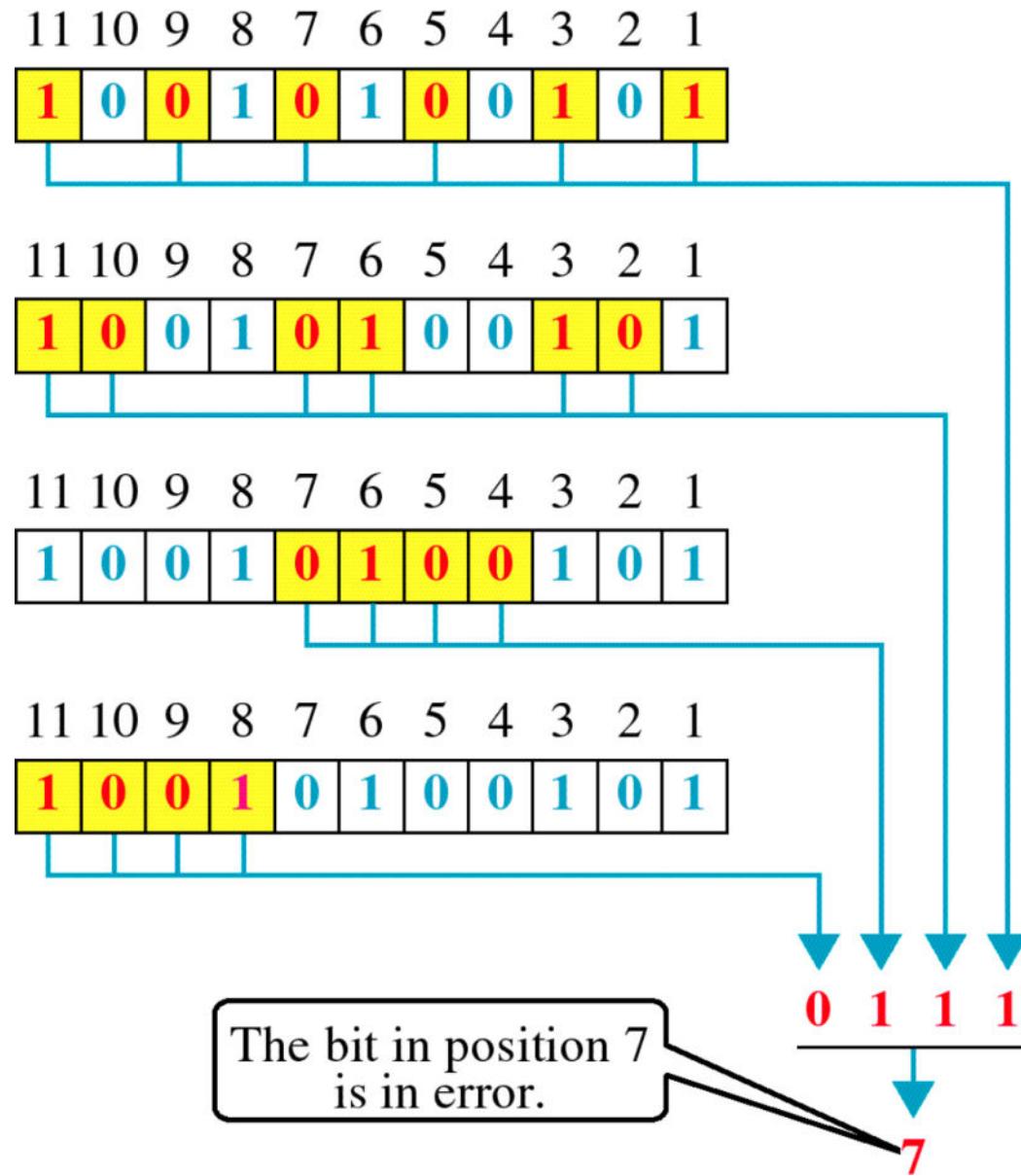
Example of Hamming Code

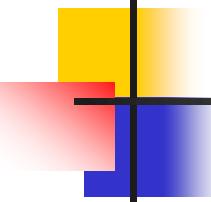


Single-bit error



Error Detection





Example 10.3

Let us add more redundant bits to Example 10.2 to see if the receiver can correct an error without knowing what was actually sent. We add 3 redundant bits to the 2-bit dataword to make 5-bit codewords. Table 10.2 shows the datawords and codewords. Assume the dataword is 01. The sender creates the codeword 01011. The codeword is corrupted during transmission, and 01001 is received. First, the receiver finds that the received codeword is not in the table. This means an error has occurred. The receiver, assuming that there is only 1 bit corrupted, uses the following strategy to guess the correct dataword.

Example 10.3 (continued)

- 1. Comparing the received codeword with the first codeword in the table (01001 versus 00000), the receiver decides that the first codeword is not the one that was sent because there are two different bits.*
- 2. By the same reasoning, the original codeword cannot be the third or fourth one in the table.*
- 3. The original codeword must be the second one in the table because this is the only one that differs from the received codeword by 1 bit. The receiver replaces 01001 with 01011 and consults the table to find the dataword 01.*

Table 10.2 *A code for error correction (Example 10.3)*

<i>Dataword</i>	<i>Codeword</i>
00	00000
01	01011
10	10101
11	11110

Error control is both error detection and error correction. It allows the receiver to **tell** the sender of any frames lost or damaged in transmission and coordinates the retransmission of those frames by the sender.

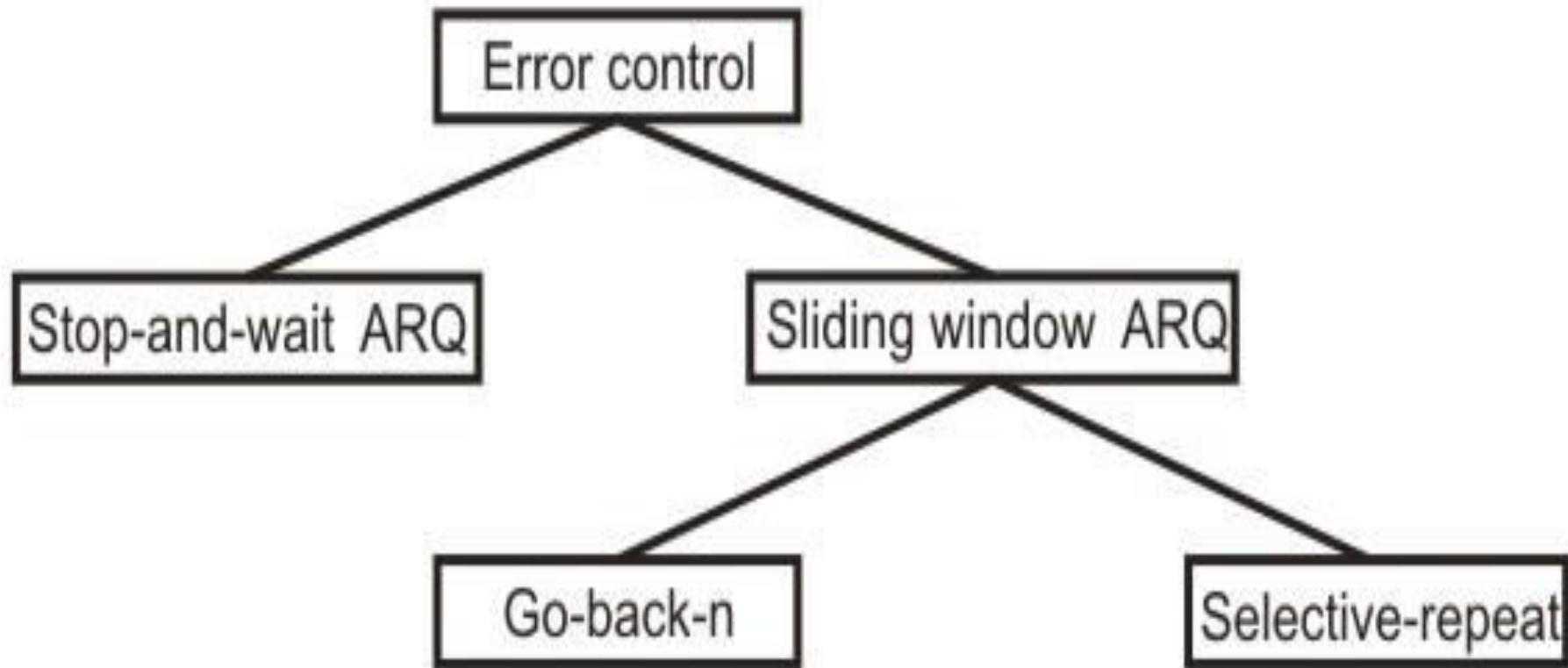
Error control in the data link layer is based on automatic repeat request (ARQ), which is the retransmission of data.

FLOW AND ERROR CONTROL

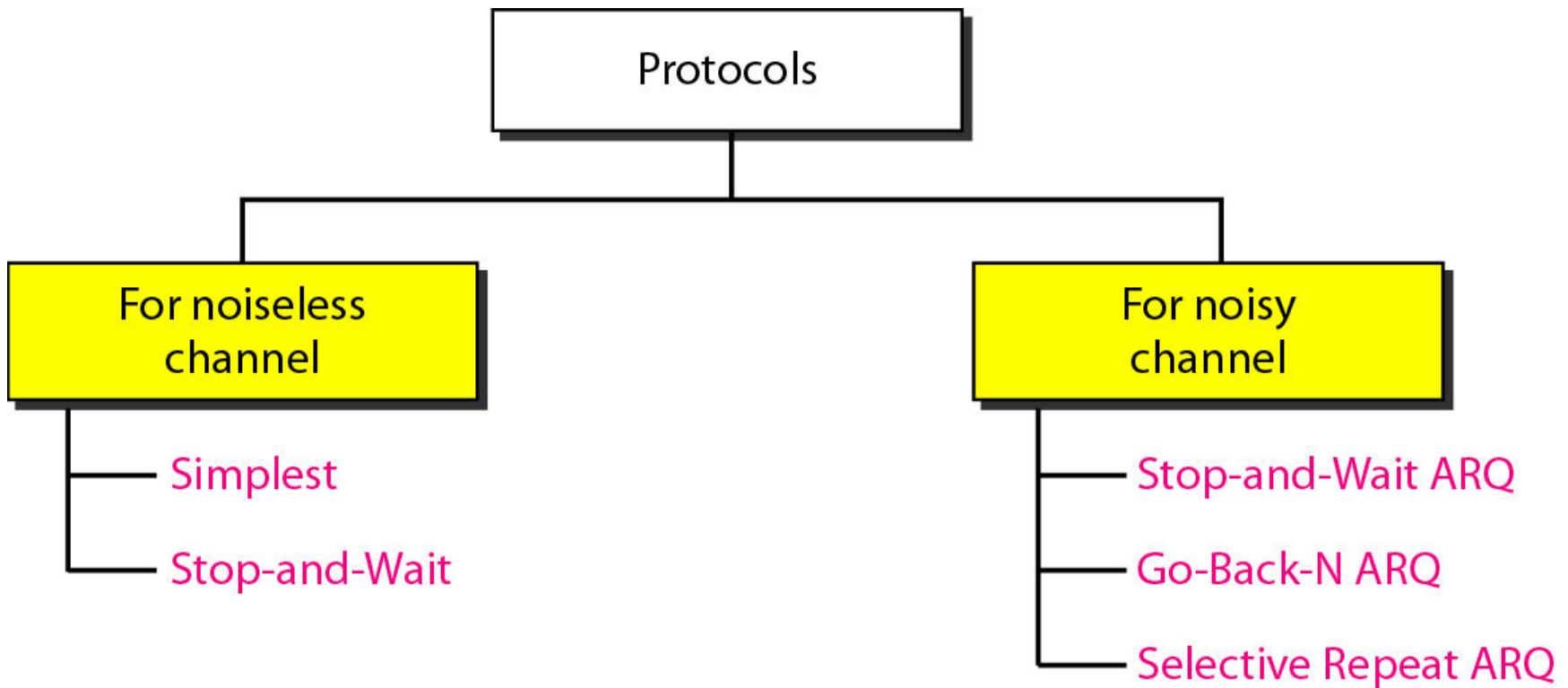
- Flow Control:
- Set of procedures that tells the sender how much data it can transmit before it must wait for an acknowledgment from the receiver
- Error Control:
- Data link layer is based on automatic repeat request, which is the retransmission of data

PROTOCOLS

- 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.

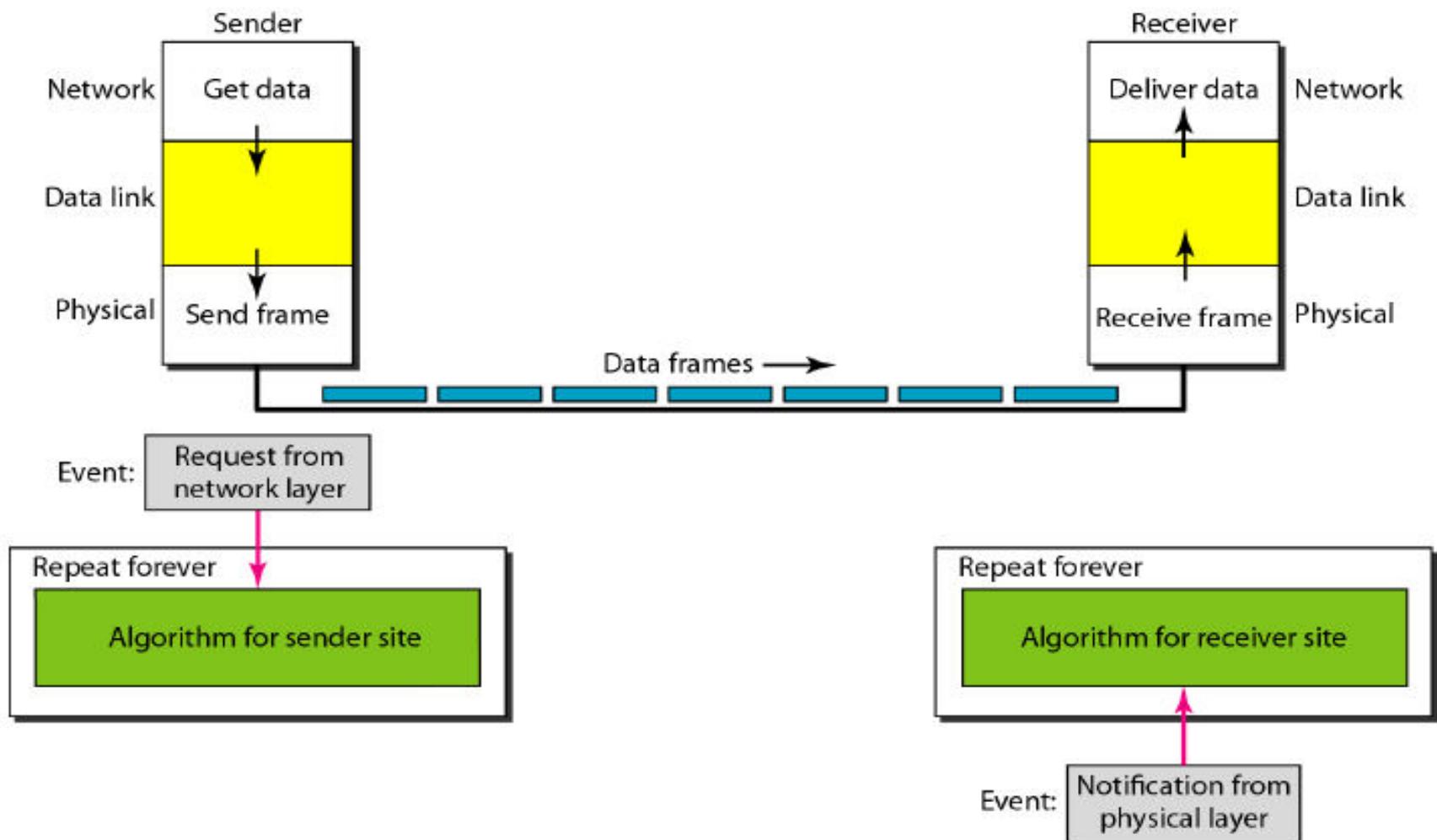


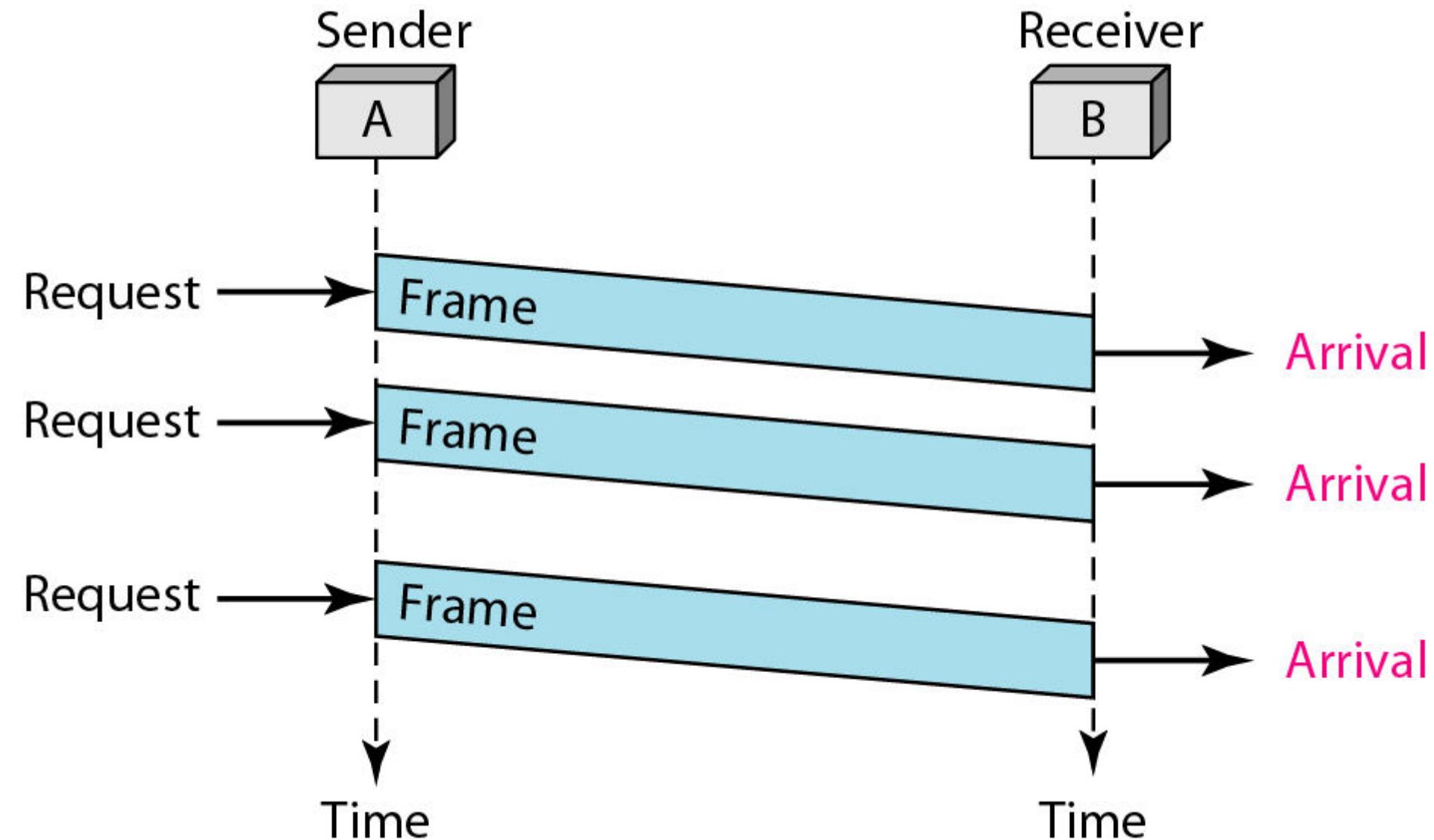
Classification of protocols



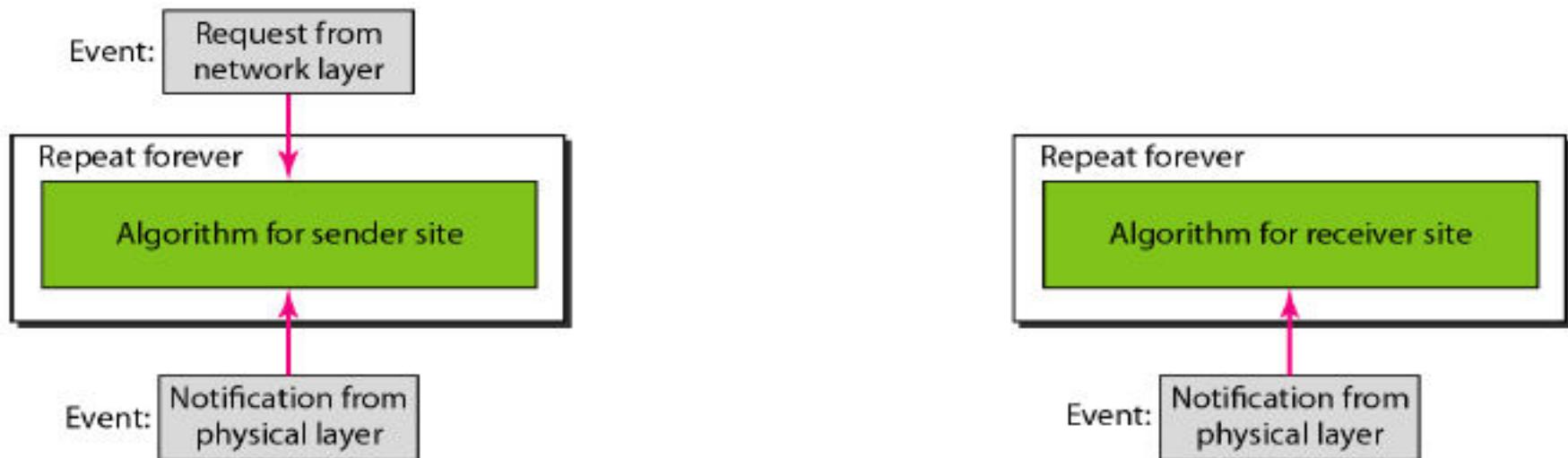
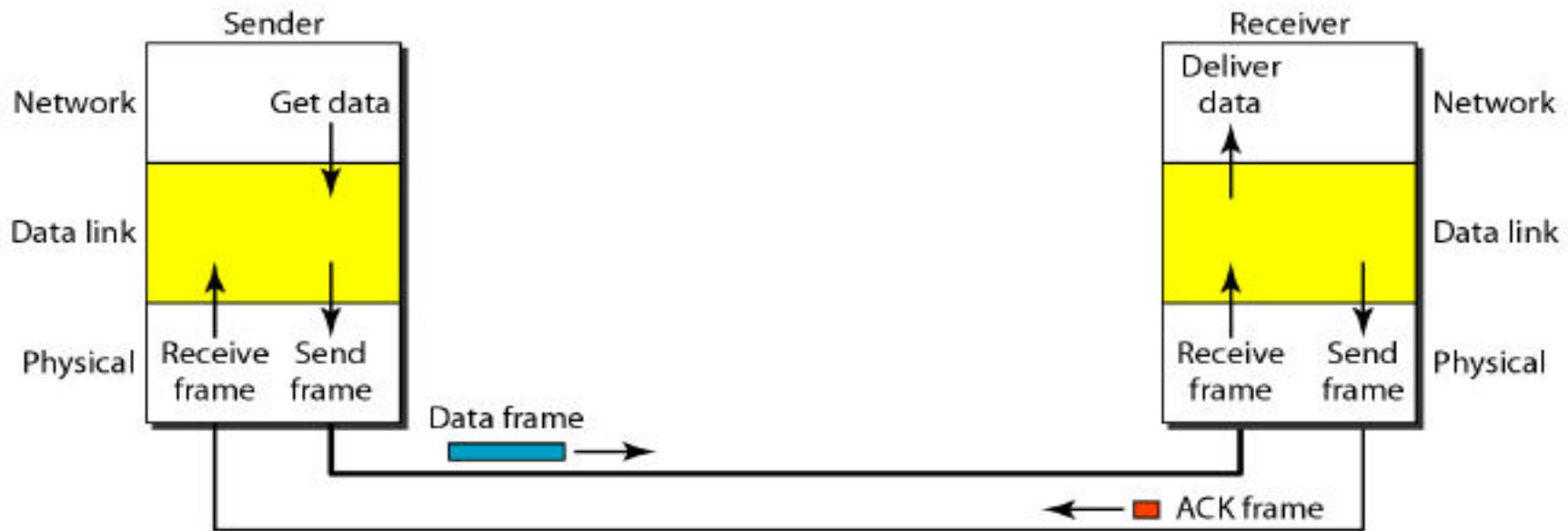
NOISELESS CHANNELS

- **Simplest Protocol:** data frames are traveling in only one direction-from the sender to receiver





Stop-and-Wait Protocol



NOISY CHANNELS

- **Stop-and-Wait Automatic Repeat Request(ARQ)**
- **Go-Back-N Automatic Repeat Request**
- **Selective Repeat Automatic Repeat Request**



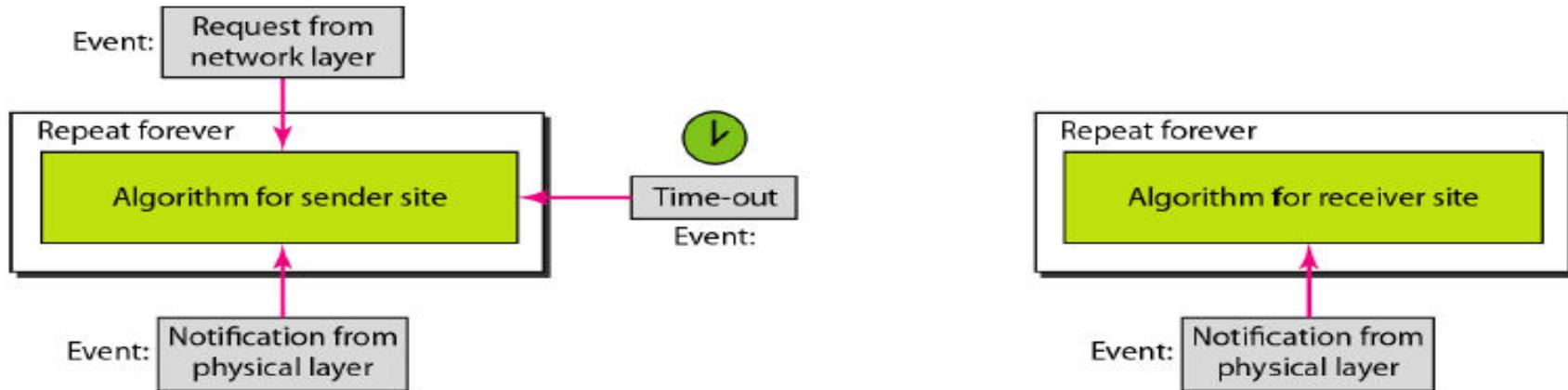
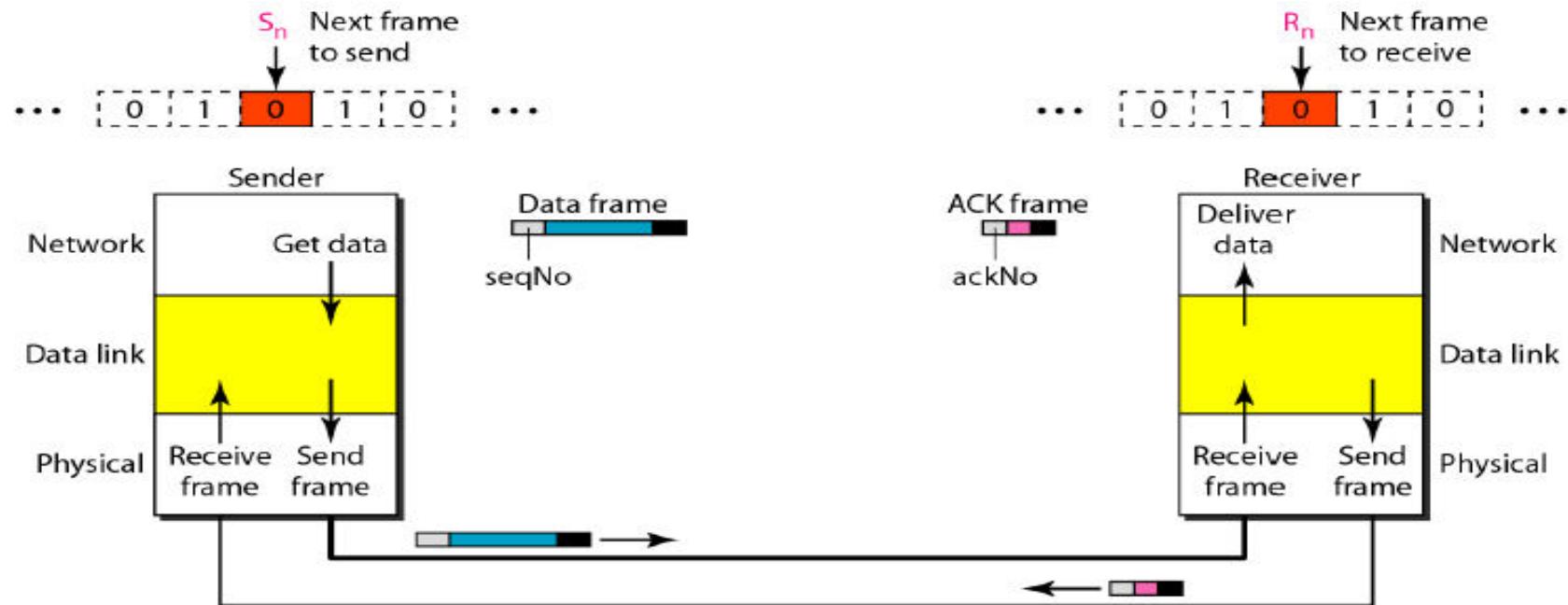
Error correction in Stop-and-Wait ARQ is done by keeping a copy of the sent frame and retransmitting of the frame when the timer expires.

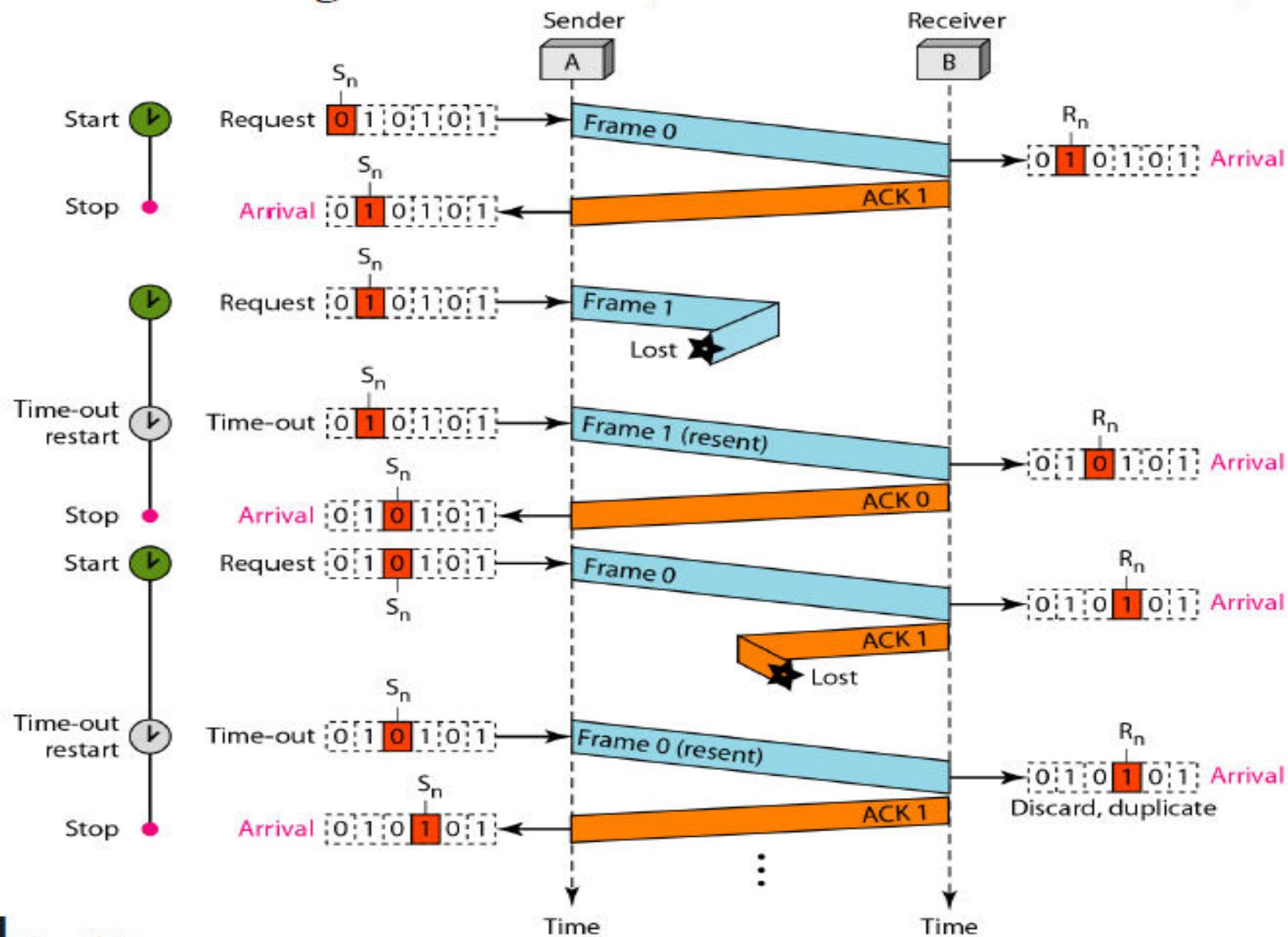
In Stop-and-Wait ARQ, we use sequence numbers to number the frames.

The sequence numbers are based on modulo-2 arithmetic.

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

Design of the Stop-and-Wait ARQ Protocol





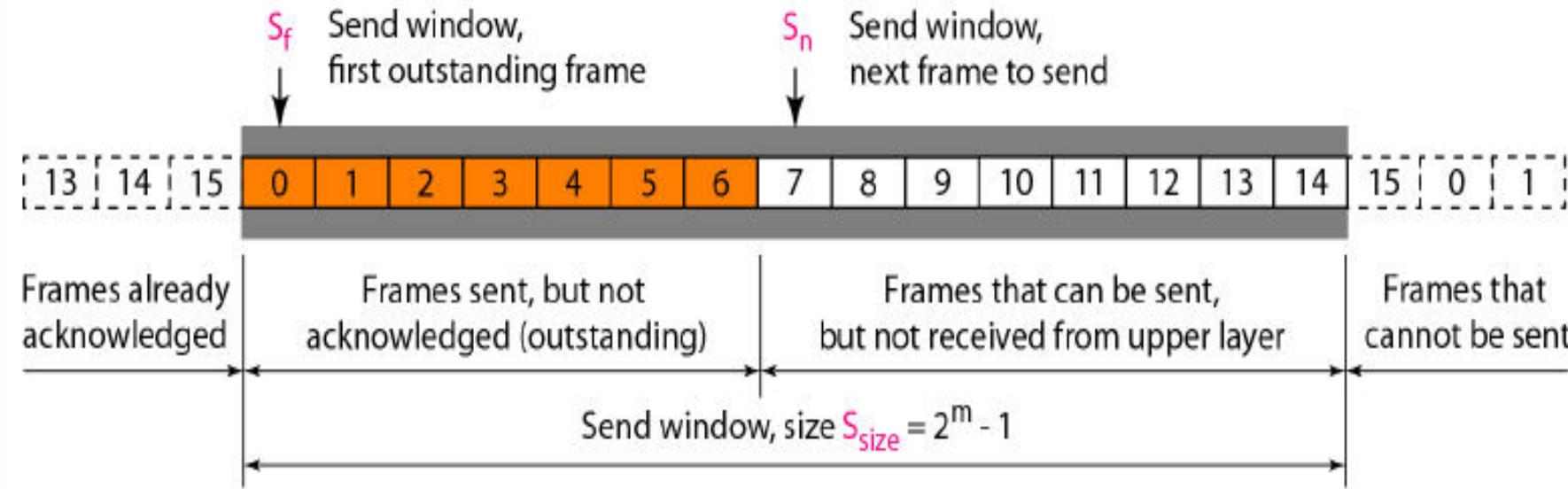
Pipelining

- Pipelining: In networking and in other areas, a task is often begun before the previous task has ended.
- There is no pipelining in Stop-and-Wait ARQ because we need to wait for a frame to reach the destination and be acknowledged before the next frame can be sent.
- Pipelining improves the efficiency of the transmission if the number of bits in transition is large with respect to the bandwidth-delay

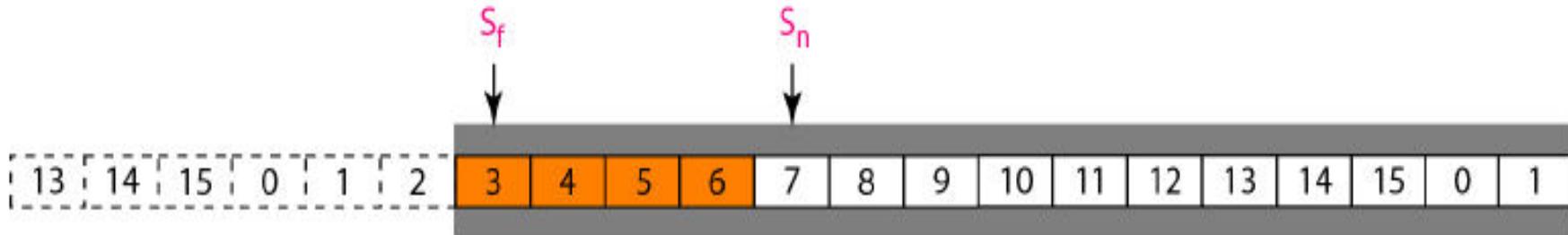
Go-Back-N Automatic Repeat Request

- To improve the efficiency of transmission (filling the pipe), multiple frames must be in transition while waiting for acknowledgment.
- Let more than one frame be outstanding to keep the channel busy while the sender is waiting for acknowledgment.
- In this protocol, can send several frames before receiving acknowledgments, we keep a copy of these frames until the acknowledgments arrive. Thus need sequence number for frames.

Sender: Go-Back-N ARQ

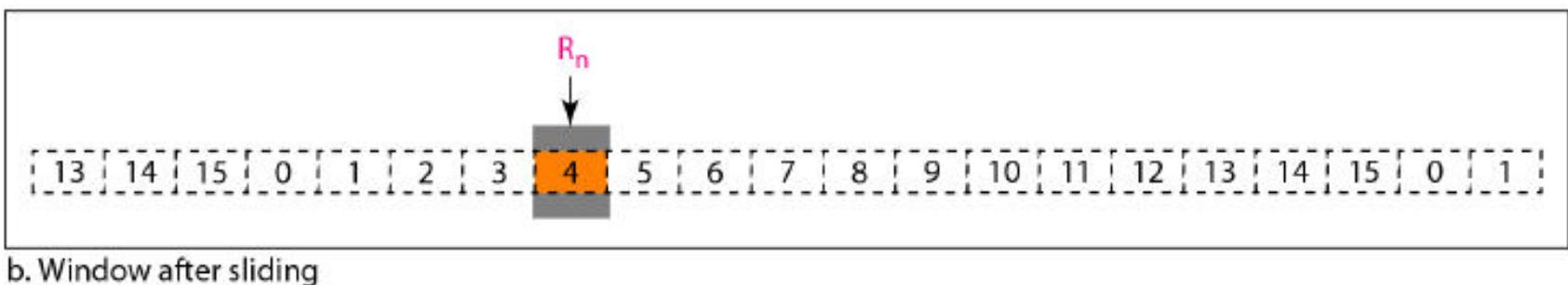
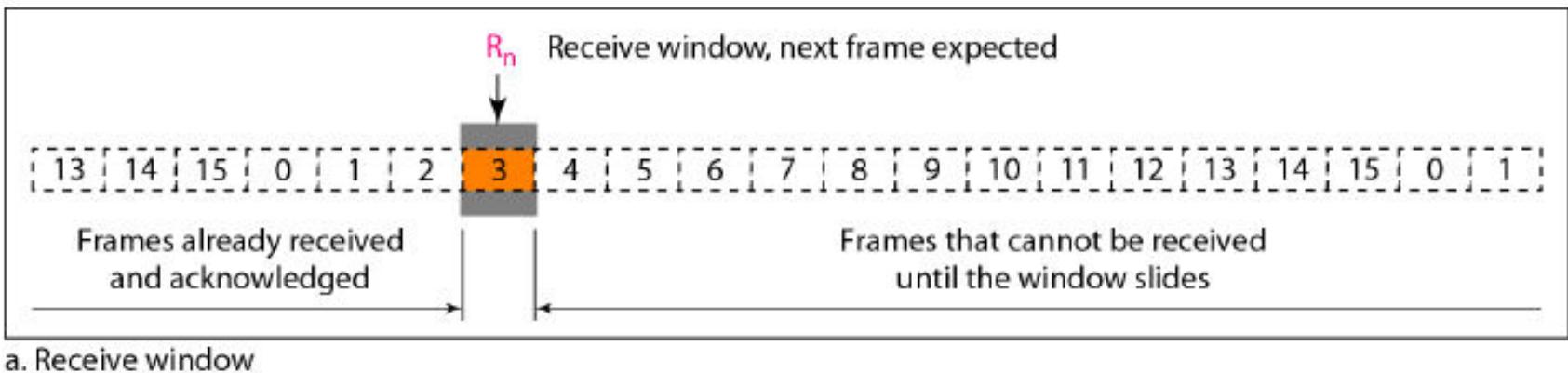


a. Send window before sliding



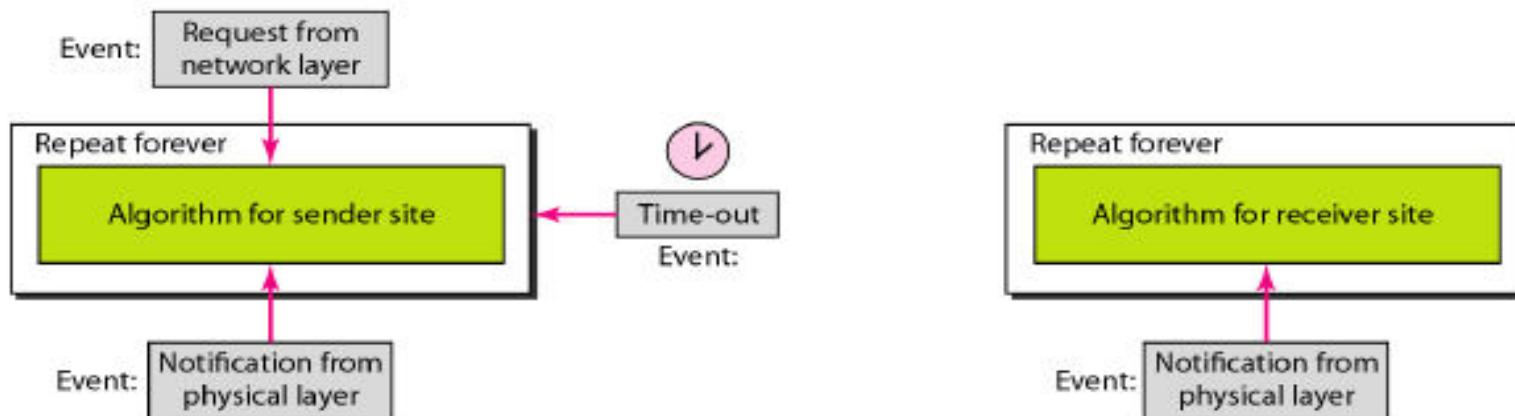
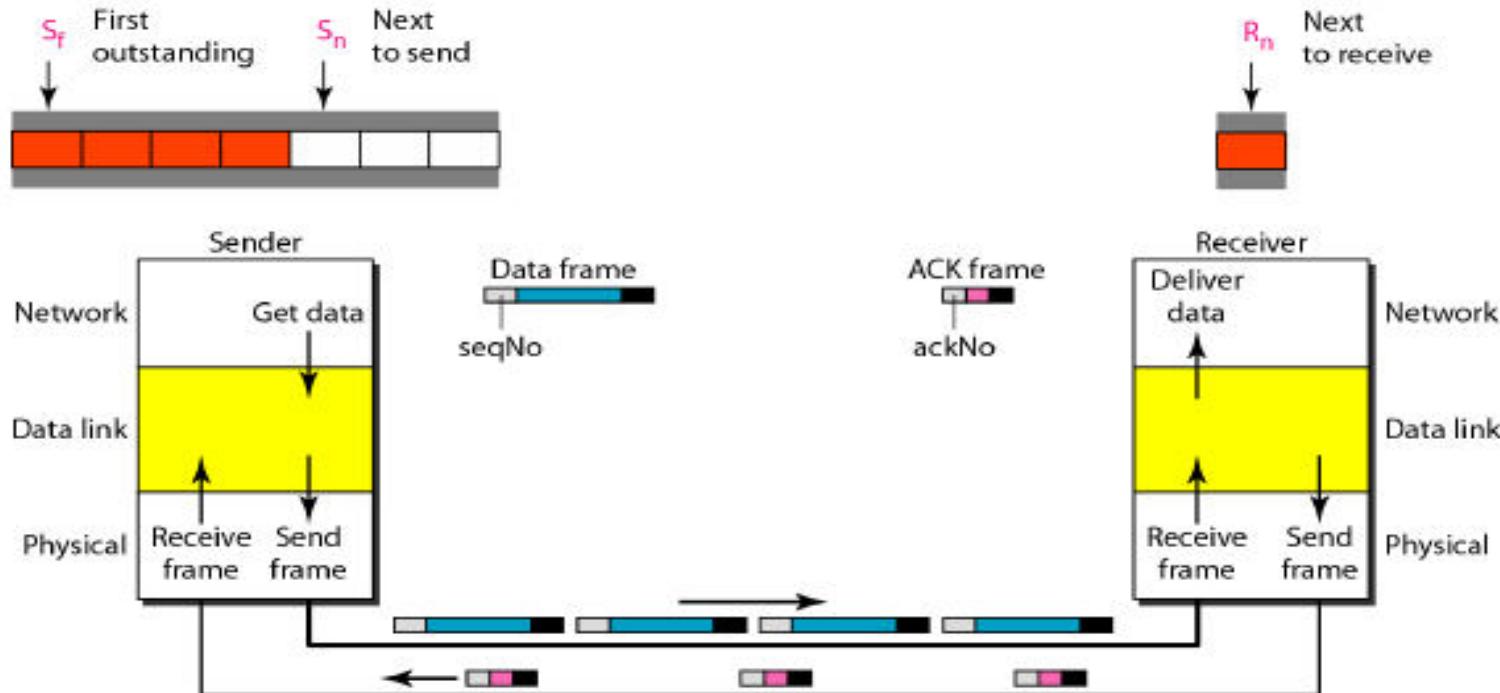
b. Send window after sliding

Receive window for Go-Back-N ARQ



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.

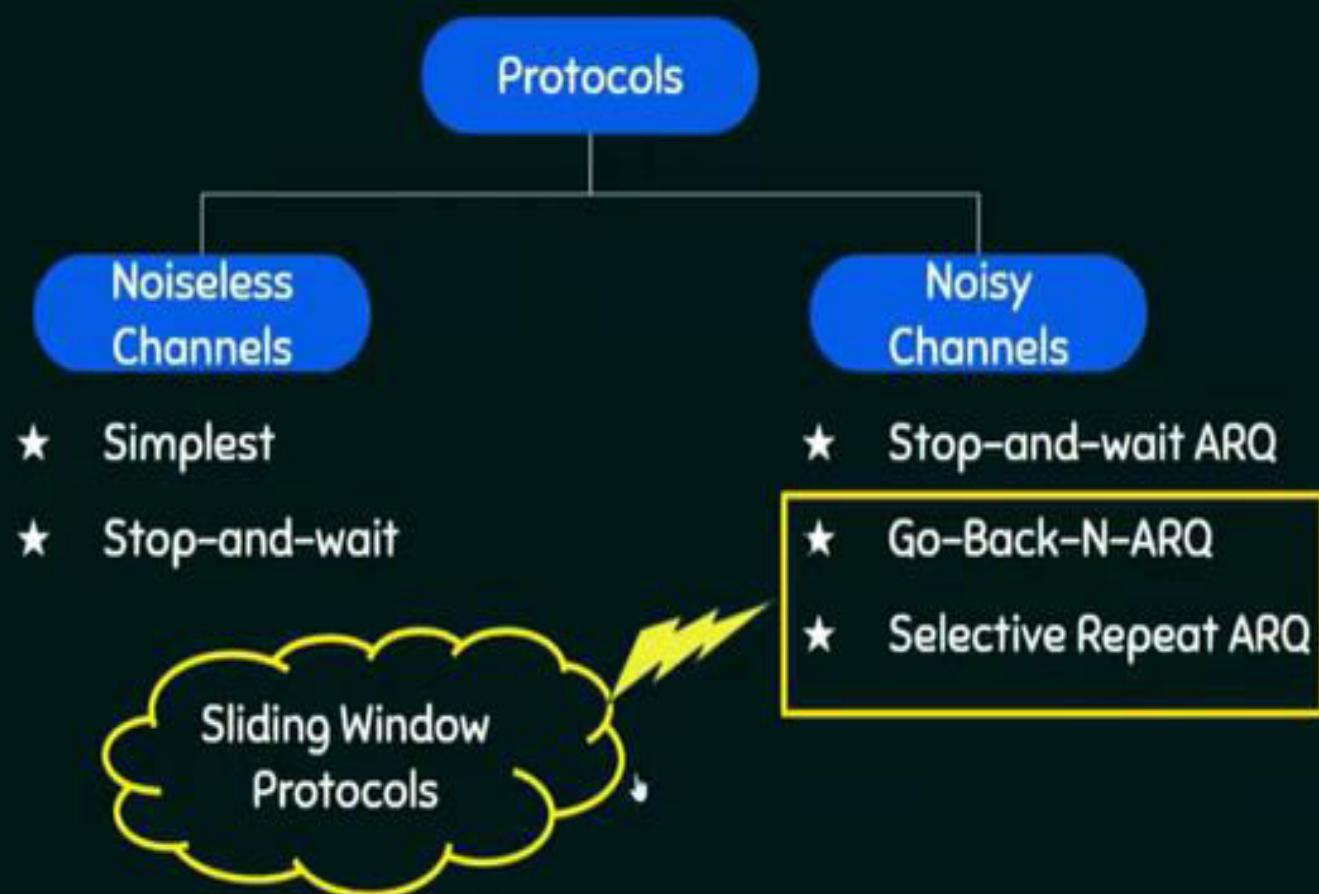
Design of Go-Back-N ARQ



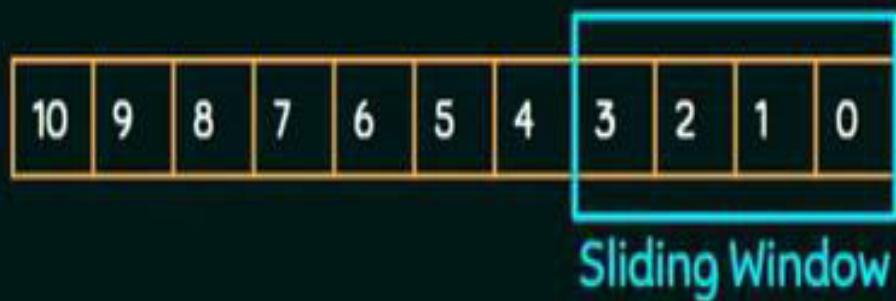
SELECTIVE REPEAT ARQ

- ★ In Selective Repeat ARQ, only the erroneous or lost frames are retransmitted, while correct frames are received and buffered.
- ★ The receiver while keeping track of sequence numbers, buffers the frames in memory and sends NACK for only frame which is missing or damaged.
- ★ The sender will send/retransmit packet for which NACK is received.

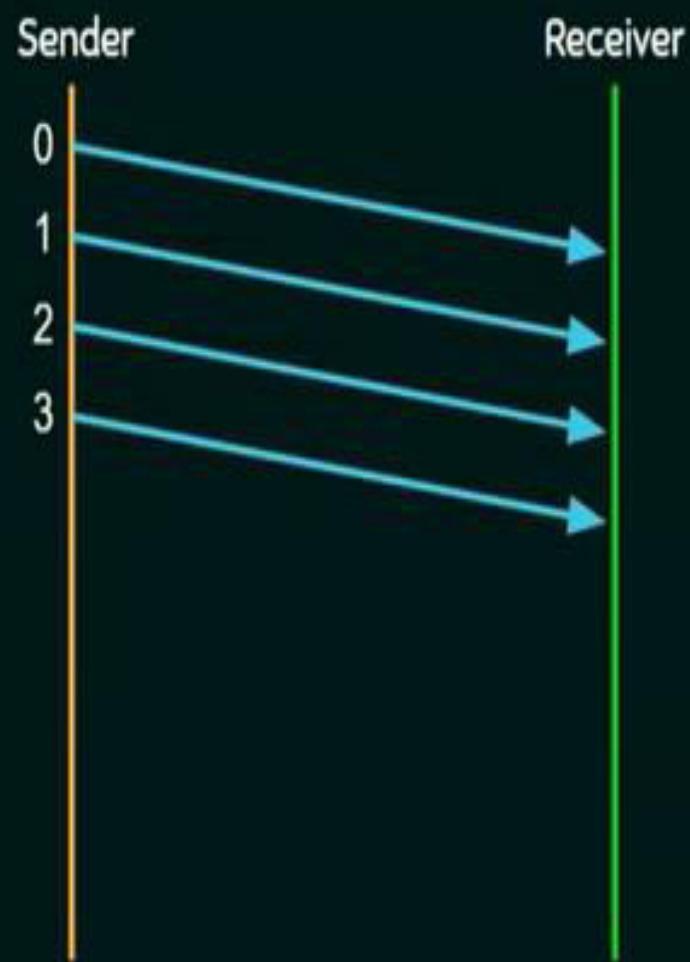
SLIDING WINDOW PROTOCOLS



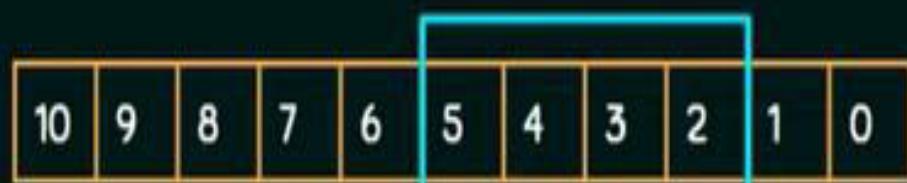
WORKING OF SELECTIVE REPEAT



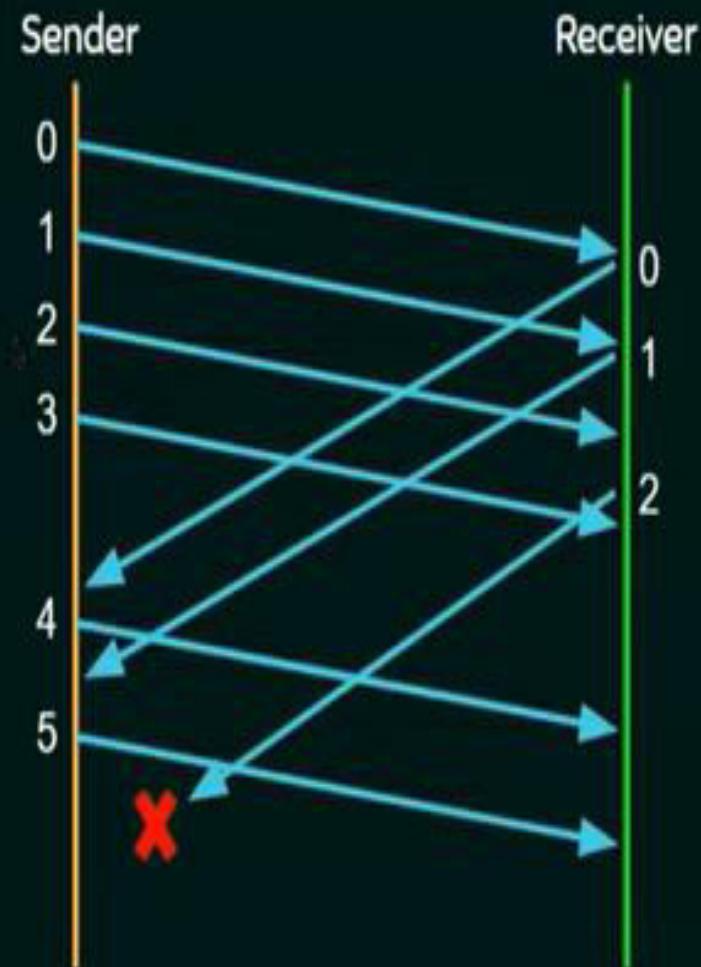
Window Size: 4



WORKING OF SELECTIVE REPEAT



Window Size: 4

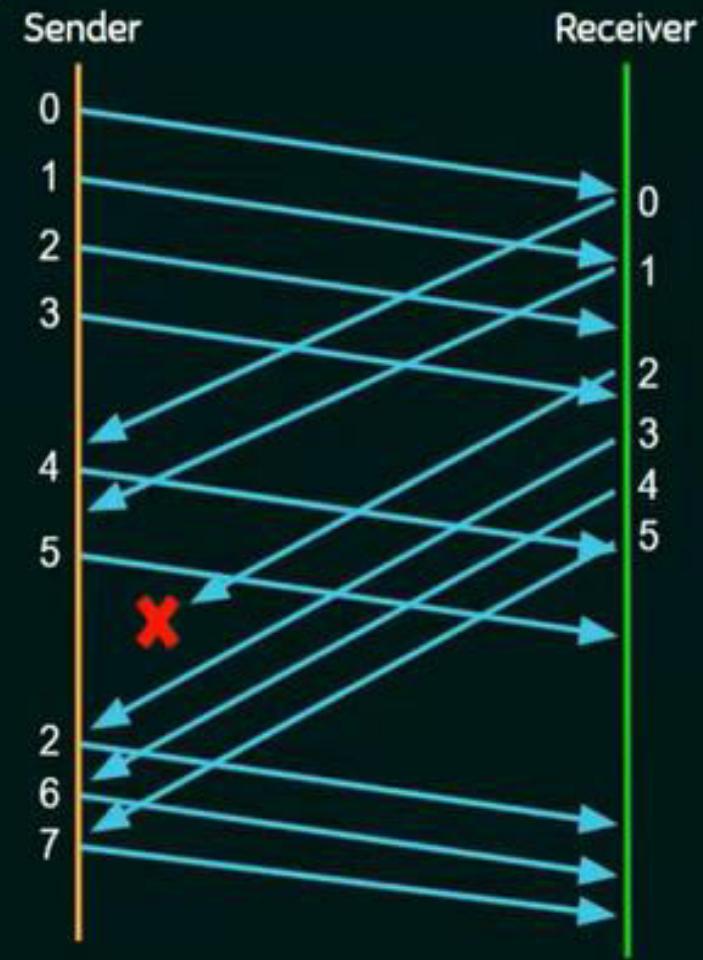


Selective Repeat Automatic Repeat Request

WORKING OF SELECTIVE REPEAT



Window Size: 4

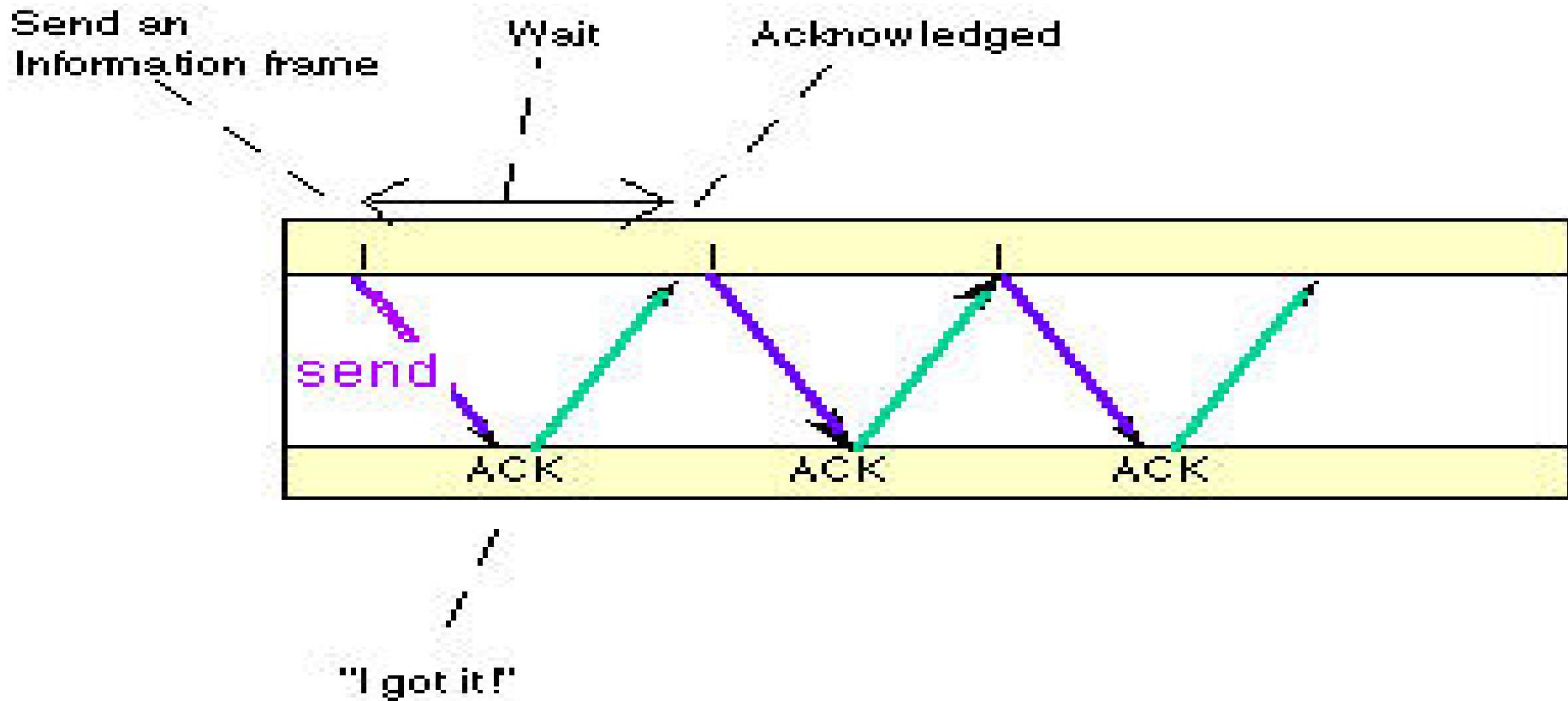


Flow Control

- Flow Control is a set of procedures that tells the sender how much data it can transmit before it must wait for an acknowledgment from the receiver
- Flow control refers to the set of procedures used to restrict the amount of data the transmitter can send before waiting for acknowledgment.
- There are two methods developed for flow control namely Stop-and-wait and Sliding-window.

Flow Control: Stop-and-Wait

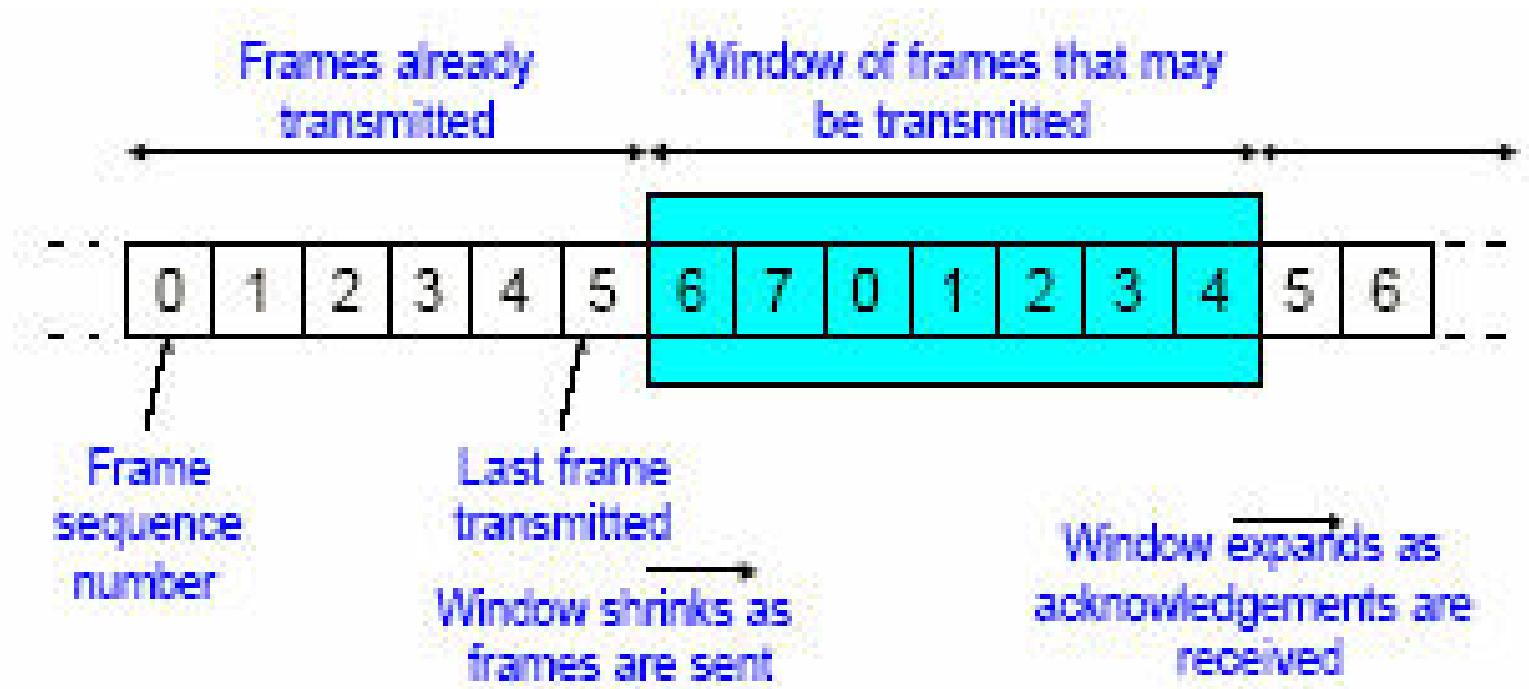
- Ping-pong
- $\text{RTTs}_1 = t_2 - t_1$



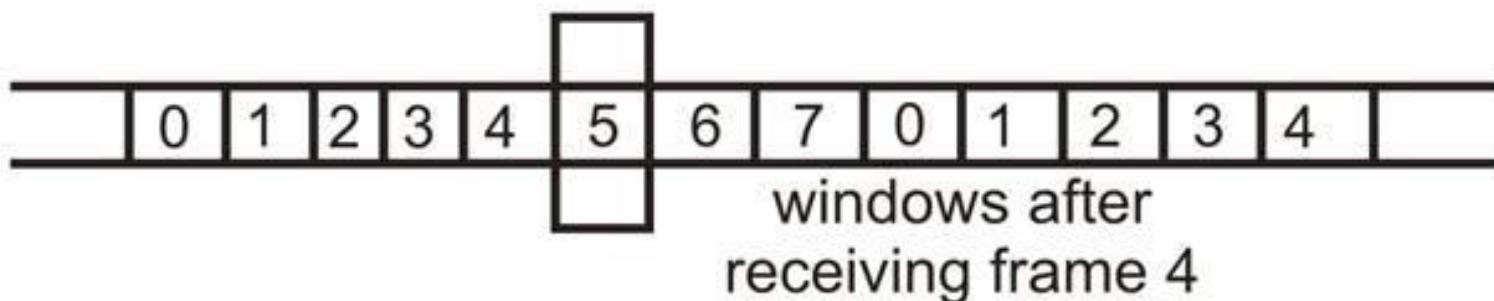
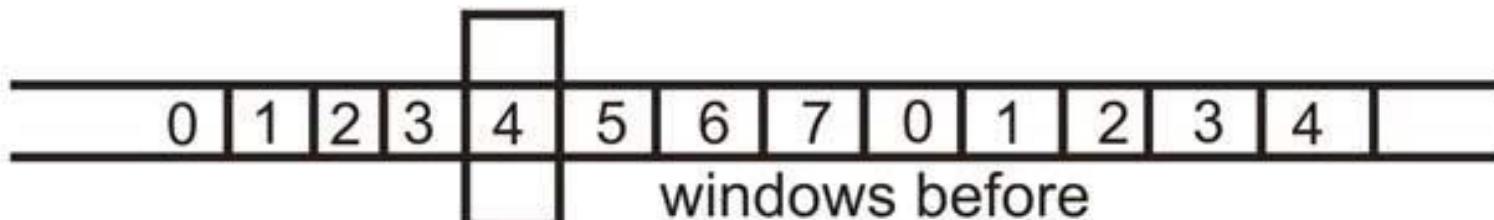
Link Utilization in Stop-and-Wait

- *Transmission time=1*
- *Propagation delay =a*
- The link utilization $U = 1/(1+2a)$,
 $a = \text{Propagation time} / \text{transmission time}$

Sliding Window: Sender



Receiver sliding Window



Sliding Window Flow Control

- Allows transmission of multiple frames
- Assigns each frame a k-bit sequence number
- Range of sequence number is $[0 \dots 2k-1]$, i.e., frames are counted modulo $2k$.
- The link utilization in case of Sliding Window Protocol
- $U = 1$, for $N > 2a + 1$
- $N/(1+2a)$, for $N < 2a + 1$
- Where N = the window size, and
 a = Propagation time / transmission time

- **Packet delivery time = Transmission time + Propagation delay.**

Multiple Access

- Data link control, a mechanism which provides a link with reliable communication
- Data link layer as two sublayers.
 - The upper sublayer is responsible for data link control, flow and error control is called the logical link control (LLC) layer.
 - The lower sublayer is responsible for resolving access to the shared media. (ie) for multiple access resolution is called the media access control (MAC) layer.

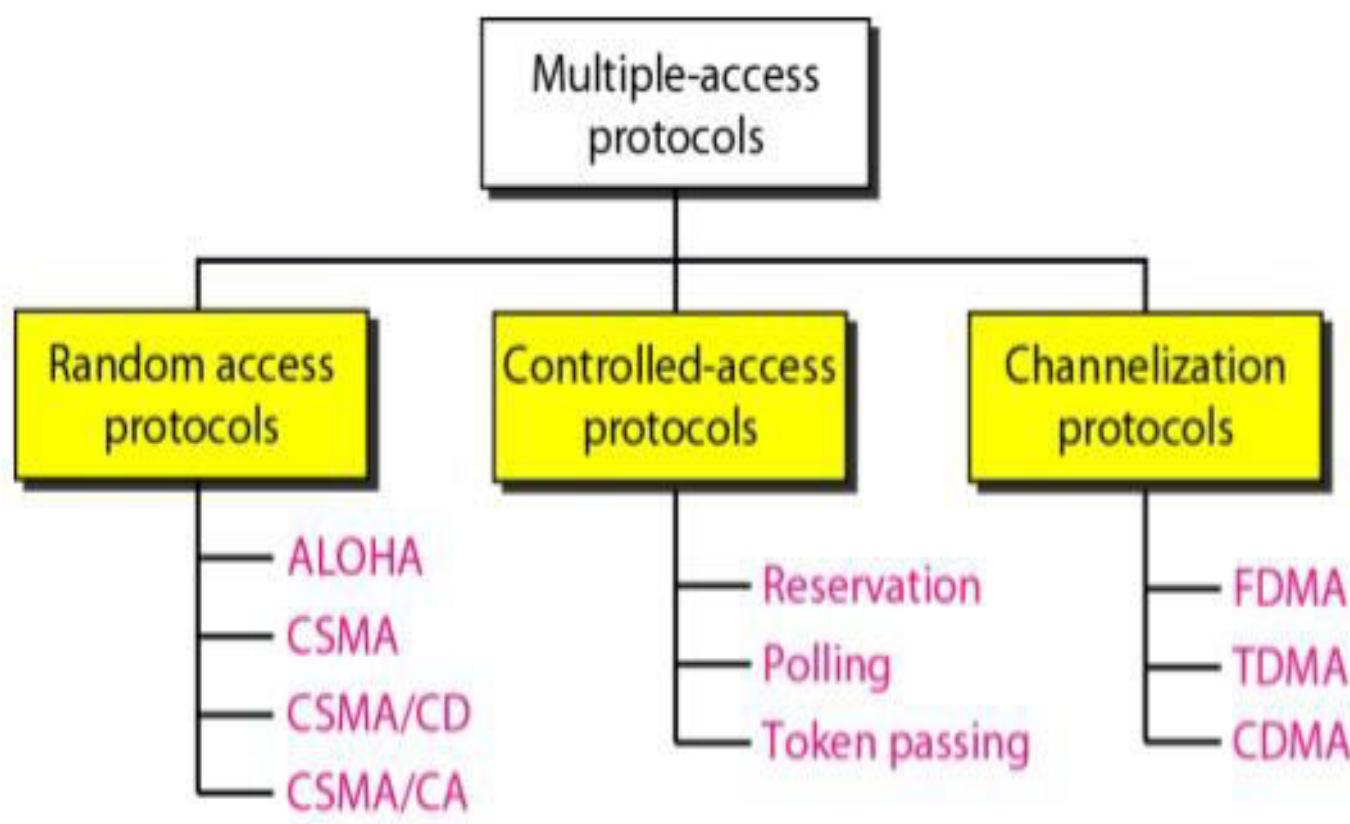
Data link layer

Data link control

Multiple-access resolution

- When nodes or stations are connected and use a common link, called a multipoint or broadcast link, we need a multiple-access protocol to coordinate access to the link.
- The problem of controlling the access to the medium. Eg. the rules of speaking.
- Many formal protocols have been devised to handle access to a shared link which has been categorized into three groups.

Types of Multiple -access protocol



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. This decision depends on the state of the medium (idle or busy).
- Each station can transmit when it desires on the condition that it follows the predefined procedure, including the testing of the state of the medium.

- In a random access method, each station has the right to the medium without being controlled by any other station. However, if more than one station tries to send, there is an access conflict-collision-and the frames will be either destroyed or modified.
- ALOHA protocol, which is used for multiple access (MA). The method has an additional procedure that forces the station to sense the medium before transmitting. This was called carrier sense multiple access (CSMA).
- This method later evolved into two parallel methods: carrier sense multiple access with collision detection (CSMA/CD), tells the station what to do when a collision is detected and carrier sense multiple access with collision avoidance (CSMA/CA). CSMA/CD, tries to avoid the collision.

- In a Random access method, each station has the right to the medium without being controlled by any other station.
- If more than one station tries to send, there is an access conflict – **COLLISION** – and the frames will be either destroyed or modified.
- To avoid access conflict, each station follows a procedure:
 - When can the station access the medium ?
 - What can the station do if the medium is busy ?
 - How can the station determine the success or failure of the transmission ?
 - What can the station do if there is an access conflict ?

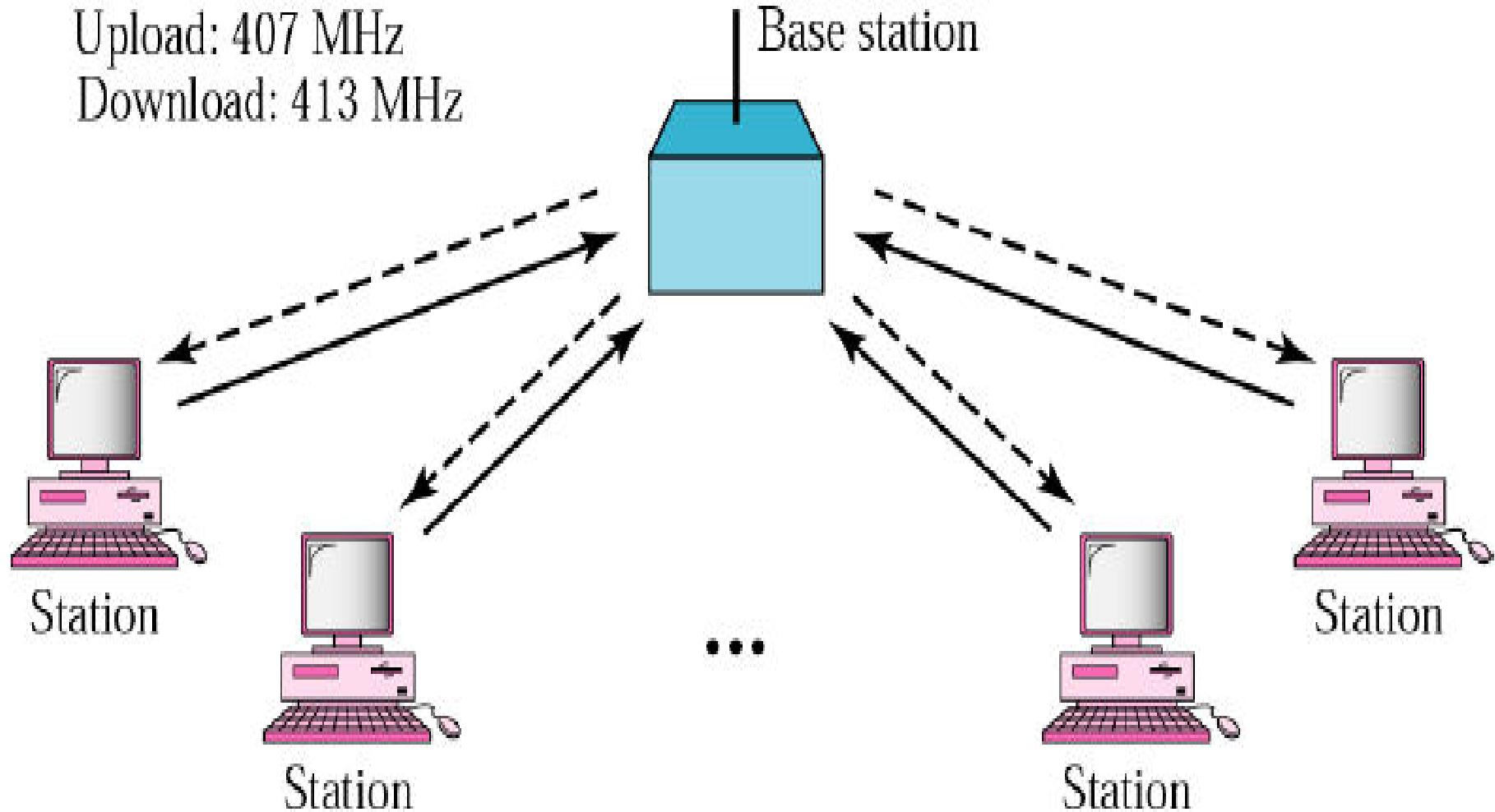
ALOHA

- ALOHA, the earliest random access method, was designed for a radio (wireless) LAN, but it can be used on any shared medium.
- It is obvious that there are potential collisions in this arrangement. The medium is shared between the stations. When a station sends data, another station may attempt to do so at the same time. The data from the two stations collide and become garbled.

ALOHA network – Multiple Access

Upload: 407 MHz

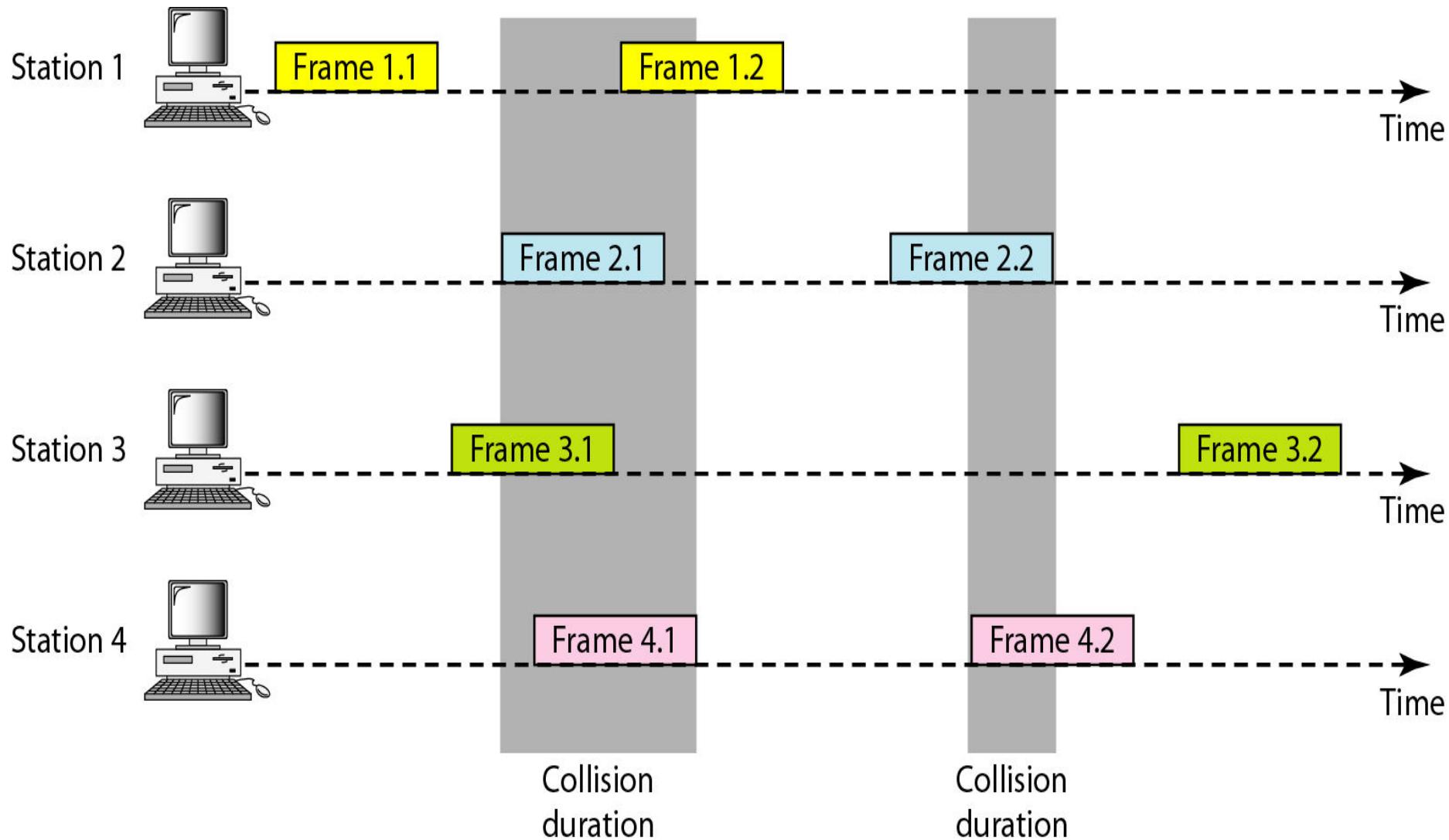
Download: 413 MHz



Pure ALOHA

- The original ALOHA protocol is called pure ALOHA. This is a simple, but elegant protocol. The idea is that each station sends a frame whenever it has a frame to send.
- However, since there is only one channel to share, there is the possibility of collision between frames from different stations.

Frames in a pure ALOHA network



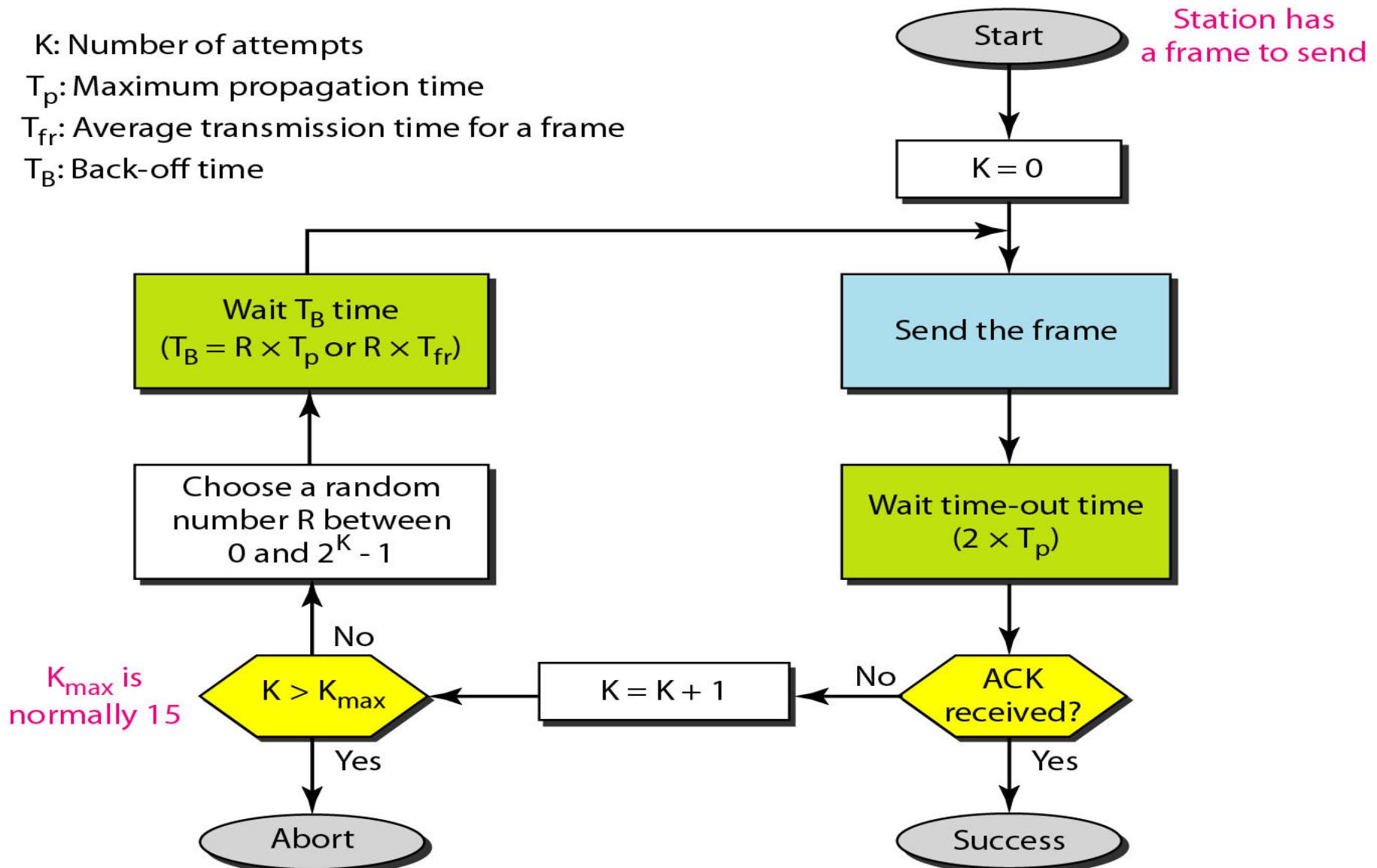
Procedure for pure ALOHA protocol

K: Number of attempts

T_p : Maximum propagation time

T_{fr} : Average transmission time for a frame

T_B : Back-off time



The throughput for pure ALOHA is

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

The maximum throughput

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

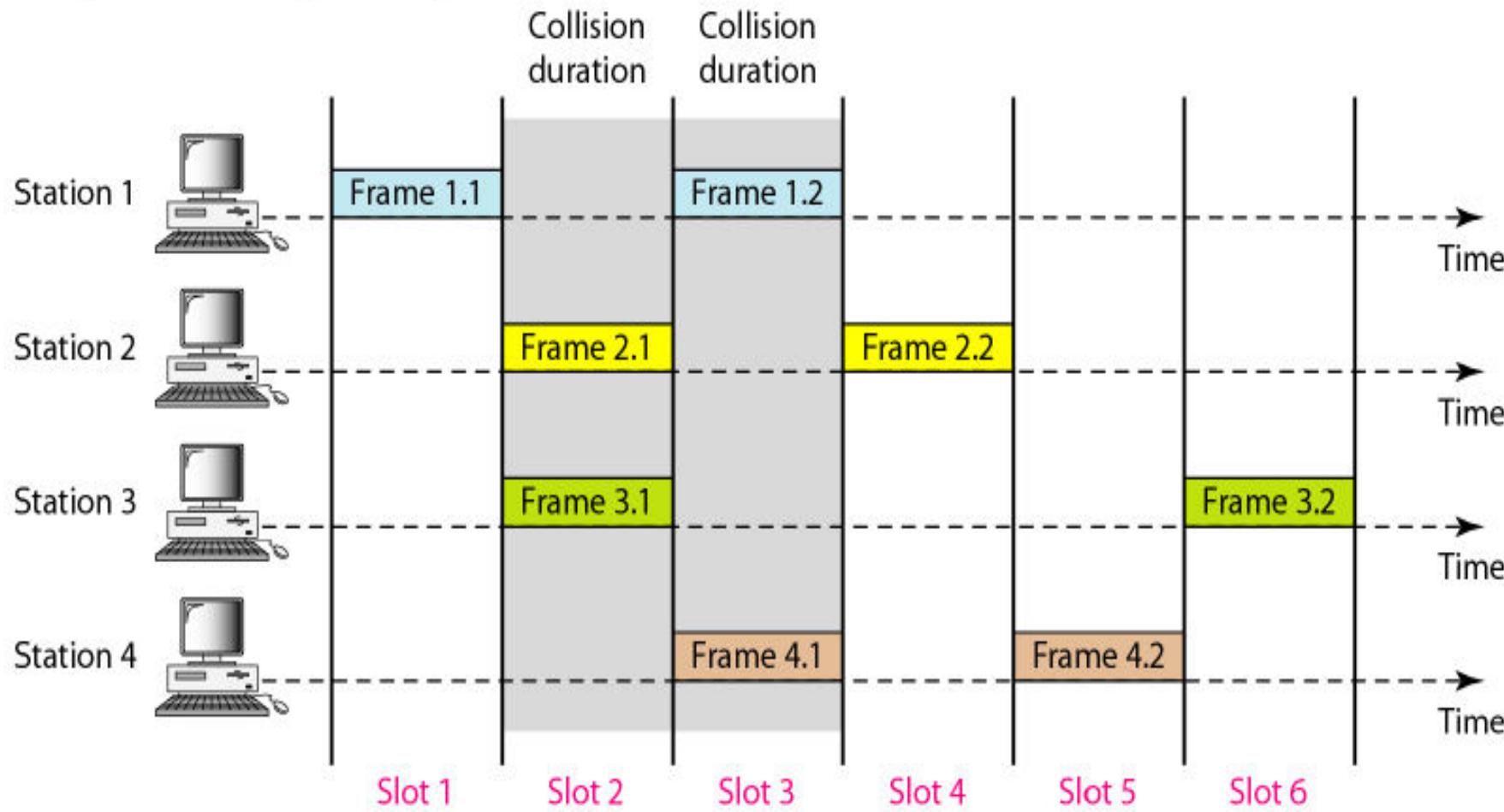
* If one-half a frame is generated during one frame transmission time, then 18.4 percent of these frames reach their destination successfully.

- ❖ S is the average number of successful transmissions, called throughput.
- ❖ G is the average number of frames generated by the system during one frame transmission time.

Slotted ALOHA

□ Slotted ALOHA

- ❖ We divide the time into slots of T_{fr} s and force the station to send only at the beginning of the time slot.



The throughput for slotted ALOHA is

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

The maximum throughput

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

- If a frame is generated during one frame transmission time, then 36.8 percent of these frames reach their destination successfully.

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.

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 $200/1200$ 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.
- 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.

ALOHA Class of Multiple Access Protocols

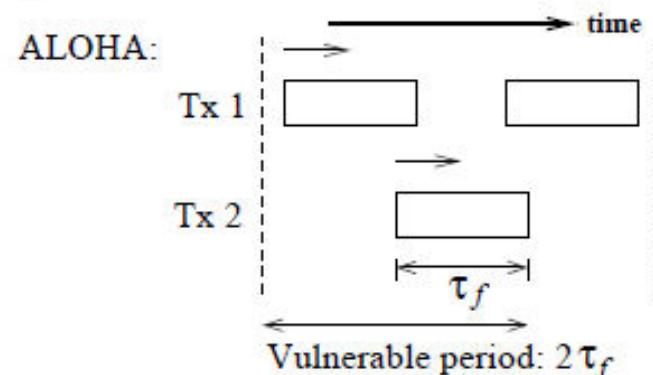
- **ALOHA**, also called pure ALOHA: Whenever a user has a frame to send, it simply transmits the frame. If collision occurs, it waits for a random period of time and re-sends it again
 - Sender can always find out if its frame was destroyed by listening to channel. For a LAN, feedback is immediate, while for a satellite there is a long delay of 270 ms before sender knows
 - If listening while transmitting is not possible, ACKs are needed, e.g. in packet radio, collision from simultaneous transmissions of multiple transmitters is detected by base station, who sends out ACK or NAK accordingly (via reverse channel)
- **Performance:** throughput S (frames/s) which defines average number of frames successfully transmitted per unit time, and **average delay** D (s) experienced by a frame
- Assuming average frame length τ_f (s) and fixed channel rate, frame transmission can be modelled by Poisson distribution with mean arrival rate λ (frames/s)

Normalised channel traffic or average number of old and new frames submitted per frame time is

$$G = \lambda\tau_f \text{ (unit in Erlang)}$$

The throughput is then given by

$$S = G \times \text{Prob}(\text{no collision})$$



ALOHA Class (continue)

- Slotted ALOHA: time is divided into slots of equal length greater or equal to average frame duration τ_f , and frame transmission can only start at beginning of a time slot
- Probability that a frame does not suffer from a collision is given by

$$P_0 = \begin{cases} e^{-2G}, & \text{ALOHA} \\ e^{-G}, & \text{slotted ALOHA} \end{cases}$$

The throughput/frame time is then

$$S = \begin{cases} G \cdot e^{-2G}, & \text{ALOHA} \\ G \cdot e^{-G}, & \text{slotted ALOHA} \end{cases}$$

- Maximum throughput of ALOHA:

$$\frac{dS}{dG} = e^{-2G} - 2Ge^{-2G} = 0 \Rightarrow G_{\max} = \frac{1}{2} \Rightarrow S_{\max} = \frac{1}{2}e^{-1} = 0.1839$$

Maximum throughput of slotted ALOHA:

$$\frac{dS}{dG} = e^{-G} - Ge^{-G} = 0 \Rightarrow G_{\max} = 1 \Rightarrow S_{\max} = e^{-1} = 0.3679$$

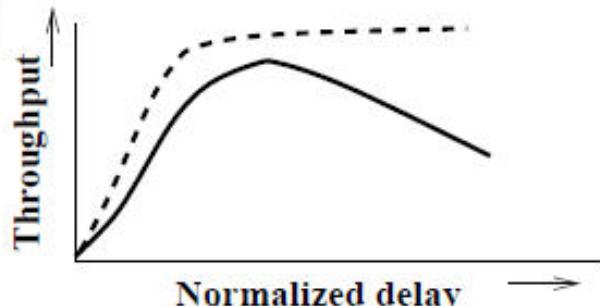
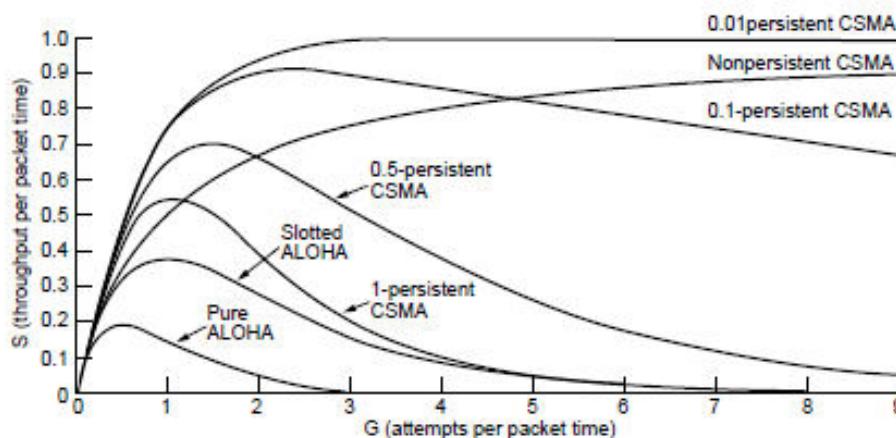
- ALOHA class is simple to implement but efficiency is low. By listening before transmitting, lots of collisions can be avoided → carrier sense multiple access (CSMA)

Carrier Sense Multiple Access

- A user wishing to transmit first listens to the medium to see if another transmission is in progress (carrier sense)
 - If the channel is in use, it must wait. If the medium is idle, it may transmit
 - **1-persistent**: a user keeps listening to see if channel is free and, as soon as the channel is idle, it transmits
 - **Nonpersistent**: when the channel is busy, it waits for a random period of time before trying to listen again. This is less greedy
 - **p-persistent**: for slotted systems. When the channel is free during current slot, it may transmit with probability p or may defer until next slot with probability $1 - p$
- **Detection** or sensing **delay** is determined by receiver hardware: a small detection time means that a user can detect a free channel rapidly
- **Propagation delay** is critical to performance: a small propagation delay means that as soon as a user launches a packet, others know quickly and will defer to transmit, thus reducing collisions
- CSMA is effective for LANs, where propagation delay is usually very small compared with frame transmission, i.e. small link parameter α
- Performance of a random access scheme is specified by S versus G and D versus G

CSMA (continue)

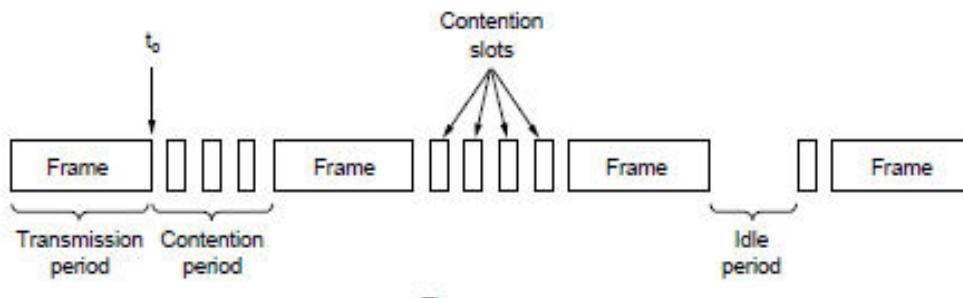
- Throughput versus load:
 - For CSMA with small p , the method performs very well in terms of throughput at high load (almost 100%). However, for smaller p , users must wait longer (larger delay) to attempt transmission
 - In the extreme case: only single user wishes to transmit, expected number of deferring is $1/p$. If $p = 0.01$, at low load, a user will wait an average of 99 time slots before transmitting on an idle line
 - For low load, slotted ALOHA is preferred due to its low delay
- Trade-off **throughput** versus **delay**: multiple access protocol with the characteristics of dashed curve is preferred
- Better performance can be achieved if user continues to listen to medium while transmitting and stops transmission immediately if collision is detected → CSMA with collision detection



CSMA with Collision Detection

- A user wishes to transmit:
 1. Listens to see if the channel is free. If the channel is idle, it transmits. If the channel is busy, it keeps listening until the channel is free, then transmits immediately (1-persistent)
 2. During the transmission, it keeps listening to detect collision. If a collision is detected, it stops transmitting immediately, and waits a random period of time before goes back to step 1.
- States of CSMA/CD:
transmission period, contention period and idle period

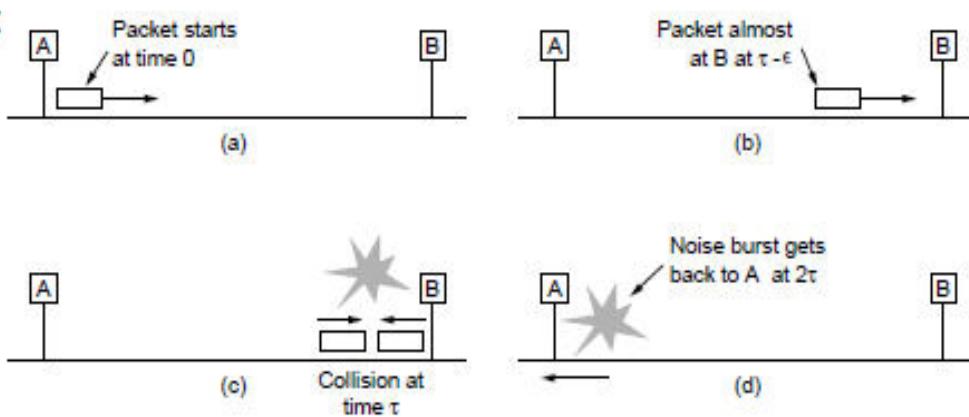
Let τ be end-to-end (two farthest users) propagation time



- Worst case time to detect collision is 2τ :

Frames should be long enough to allow collision detection prior to the end of transmission, otherwise CSMA/CD degrades to CSMA

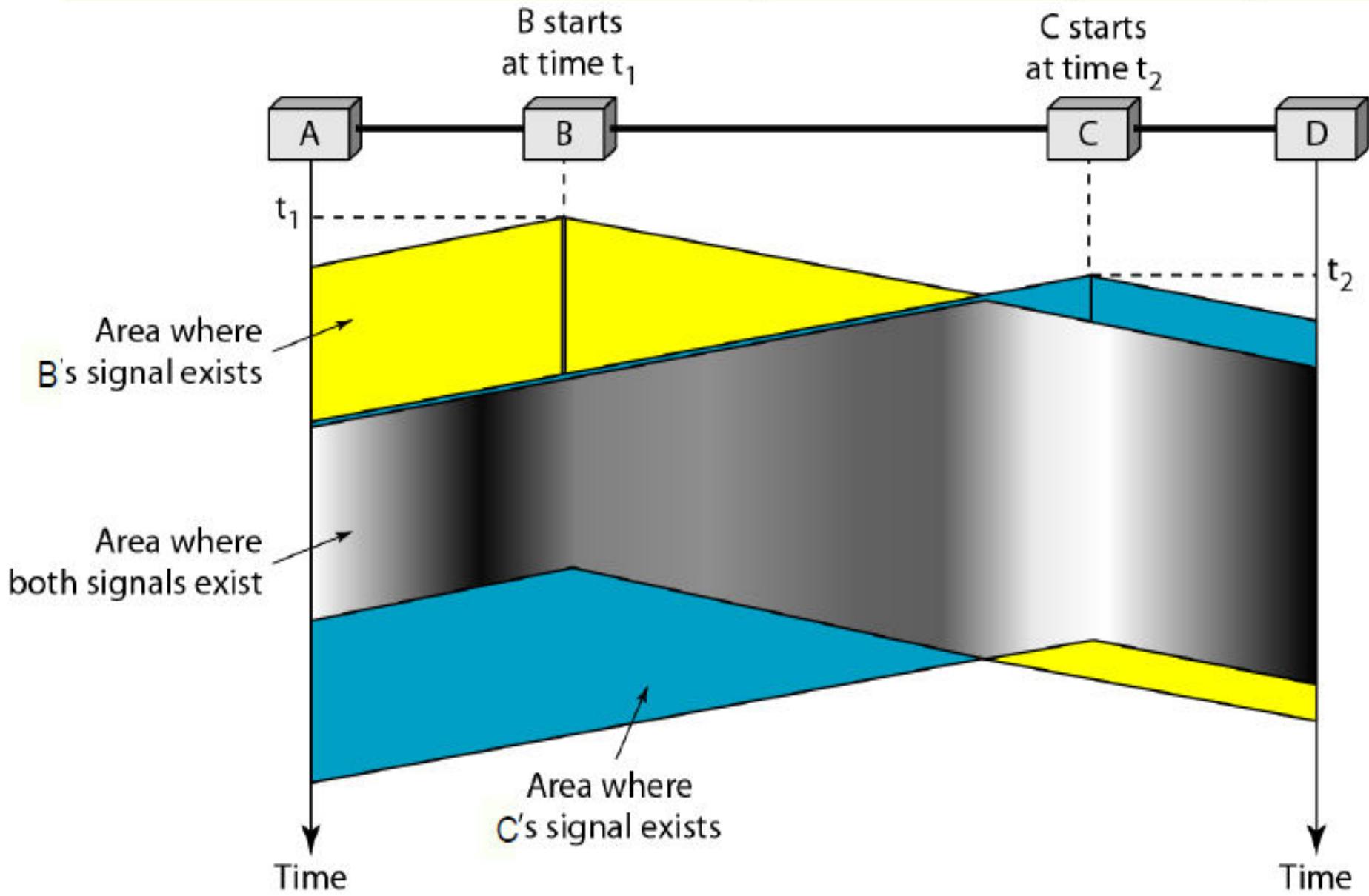
Binary exponential backoff is used: when repeatedly facing collisions, mean value of random delay is doubled



Carrier Sense Multiple Access (CSMA)

- ❑ To minimize the chance of collision and, therefore, increase the performance, the CSMA method was developed.
- ❑ CSMA is based on the principle “sense before transmit” or “listen before talk.”
- ❑ CSMA can reduce the possibility of collision, but it cannot eliminate it.
 - ❖ The possibility of collision still exists because of propagation delay; a station may sense the medium and find it idle, only because the first bit sent by another station has not yet been received.

Carrier Sense Multiple Access (CSMA)



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 $25.6 \mu\text{s}$, what is the minimum size of the frame?

Solution

The frame transmission time is $T_{fr} = 2 \times T_p = 51.2 \mu\text{s}$. This means, in the worst case, a station needs to transmit for a period of $51.2 \mu\text{s}$ to detect the collision. The minimum size of the frame is $10 \text{ Mbps} \times 51.2 \mu\text{s} = 512 \text{ bits}$ or 64 bytes. This is actually the minimum size of the frame for Standard Ethernet, as we will see in Chapter 13.

Procedure

Now let us look at the flow diagram for *CSMA/CD* in Figure 12.14. It is similar to the one for the ALOHA protocol, but there are differences.

The first difference is the addition of the persistence process. We need to sense the channel before we start sending the frame by using one of the persistence processes we discussed previously (nonpersistent, I-persistent, or p-persistent). The corresponding box can be replaced by one of the persistence processes shown in Figure 12.11.

The second difference is the frame transmission. In ALOHA, we first transmit the entire frame and then wait for an acknowledgment. In *CSMA/CD*, transmission and collision detection is a continuous process. We do not send the entire frame and then look for a collision. The station transmits and receives continuously and simultaneously (using two different ports). We use a loop to show that transmission is a continuous process. We constantly monitor in order to detect one of two conditions: either transmission is finished or a collision is detected. Either event stops transmission. When we come out of the loop, if a collision has not been detected, it means that transmission is complete; the entire frame is transmitted. Otherwise, a collision has occurred.

The third difference is the sending of a short jamming signal that enforces the collision in case other stations have not yet sensed the collision.

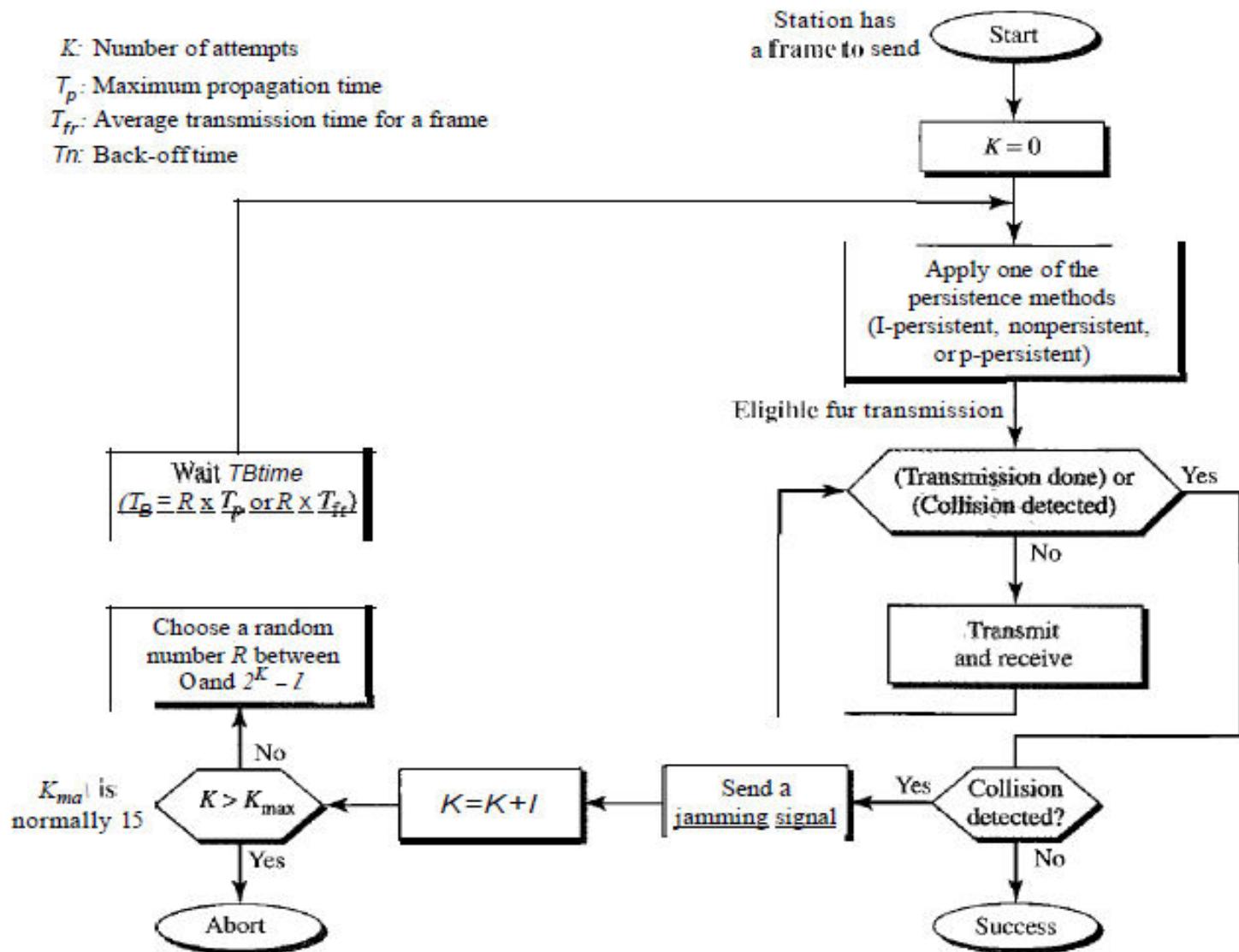
Flow diagram for the CSMA/CD

K : Number of attempts

T_p : Maximum propagation time

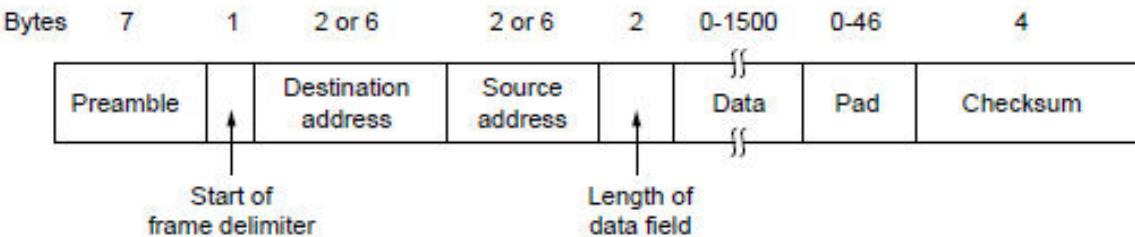
T_{fr} : Average transmission time for a frame

Tn : Back-off time



MAC for IEEE 802.3 Ethernet

- IEEE 802.3 Ethernet uses 1-persistent CSMA with CD and the frame format is:
 - Preamble:** seven 10101010 for receiver and sender clock synchronisation
 - Start of frame delimiter:** 10101011, 1-byte frame flag
 - Address:** 48 bits, the 1st bit is 0/1 for ordinary/group address and the 2nd bit is 0/1 for global/local address, and address consisting of all 1 bits is for broadcast
 - Frame \geq worst-case time to detect collision. For 10-Mbps Ethernet specification with maximum cable length 2.5 km and 4 repeaters, minimum frame time is $51.2 \mu\text{s} \rightarrow$ minimum 64 bytes
 - Data length** and **pad**: If length of data is less than 46 bytes, pad field is filled out to achieve minimum frame size of 64 bytes. This in turn requires to indicate actual data length
- MAC frame does not have control field and hence no sequence number \rightarrow it alone can only offer unacknowledged connectionless datagram services
- For connection-oriented services or for error and flow control \rightarrow LLC protocol "frames" are inserted in data fields of MAC frames
- LLC is very similar to HDLC, with address, control and data fields but no frame flag and checksum: completed layer-2 frame is MAC frame, which already has frame flag and frame checksum



CSMA with CD Performance

- Let R be data rate (bps), d be end-to-end link distance (m), V be propagation velocity (m/s), and L average frame length (bits). The **link parameter** is defined as:

$$a = \frac{\text{propagation time}}{\text{frame time}} = \frac{R d}{L V}$$

- Maximum possible **utilisation** of the channel is expressed as the **ratio of throughput to capacity**
- View time in "slots", with slot length 2τ and $\tau = \frac{d}{V}$ being end-to-end propagation time
- Recall CSMA/CD model: transmission, contention and idle periods. Under heavy load assumption
→ no idle time
- Let T_t be average transmission interval and T_c be average contention interval. The maximum utilisation or efficiency is given by

$$U = \frac{T_t}{T_t + T_c}$$

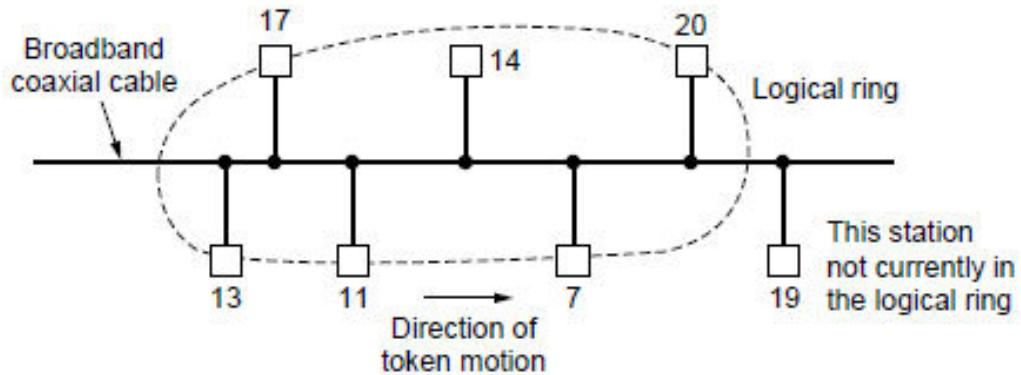
- Since $T_t = \frac{1}{2a} \times 2\tau$ and it can be shown $T_c = e \times 2\tau$,

$$U = \frac{1}{1 + 5.44a}$$

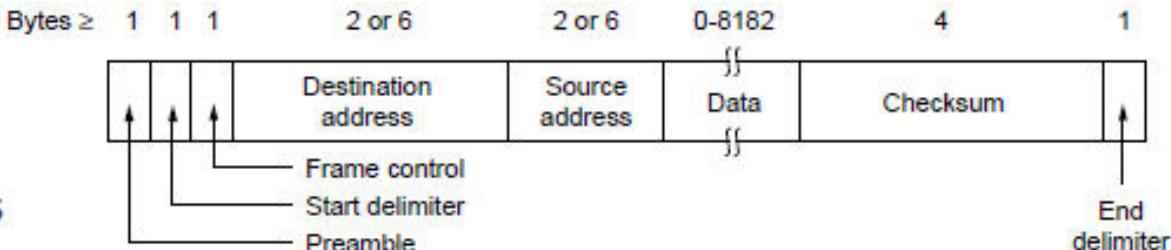
- Example.** Guided media $V = 2 \times 10^8$ (m/s), 10 Mbps LANs (Ethernet) with $\tau = 25.6 \mu s$:
Frame length 64 bytes $\Rightarrow U = 0.27$, and frame length 1024 bytes $\Rightarrow U = 0.85$

IEEE 802.4 Token Bus

- Contention protocol in IEEE 802.3 Ethernet is stochastic, i.e. worst case waiting may be unbounded. Some applications prefer known or fixed worst case waiting → round-robin
- Token bus:** physically all users are connected to a bus (as in Ethernet) but they are logically organised in a ring
- Special control frame, token, is handed from user to user in turn. User currently holding token may transmit



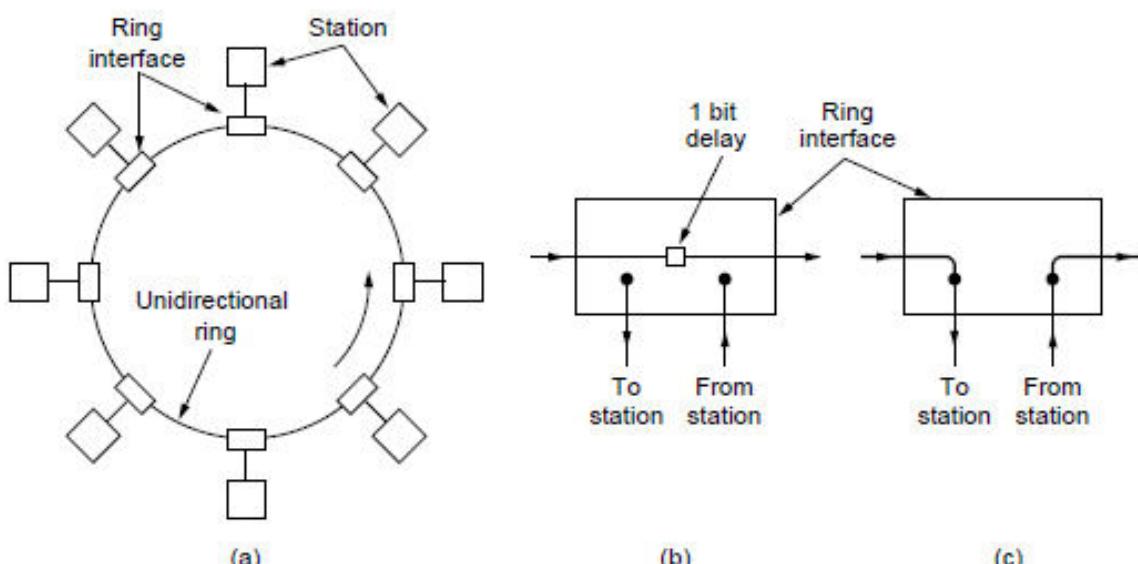
- Token bus MAC:**



- Frame control:** indicates whether a frame is data or control
- For data, frame control contains the frame's priority and a frame status indicator for destination to acknowledge correct or incorrect receipt of the frame. Otherwise destination would not be allowed to do anything since it does not have token
 - For control, it indicates type of control frame, such as claim token during initialisation, allow new stations to enter, recover from token loss, resolve contending stations for position, actual token, allow stations to leave

IEEE 802.5 Token Ring

- Token bus has very high complexity, much to do with converting a physical bus into a logical ring
- **Token ring:** Why make thing difficult? → Just physically connect stations into a ring
- **Bit physical length:** Let the data rate be R Mbps and propagation speed 200 m/ μs → A bit lasts $1/R \mu s$ or has a length $200/R$ m
e.g. for $R = 4$ Mbps, bit physical length is 50 m

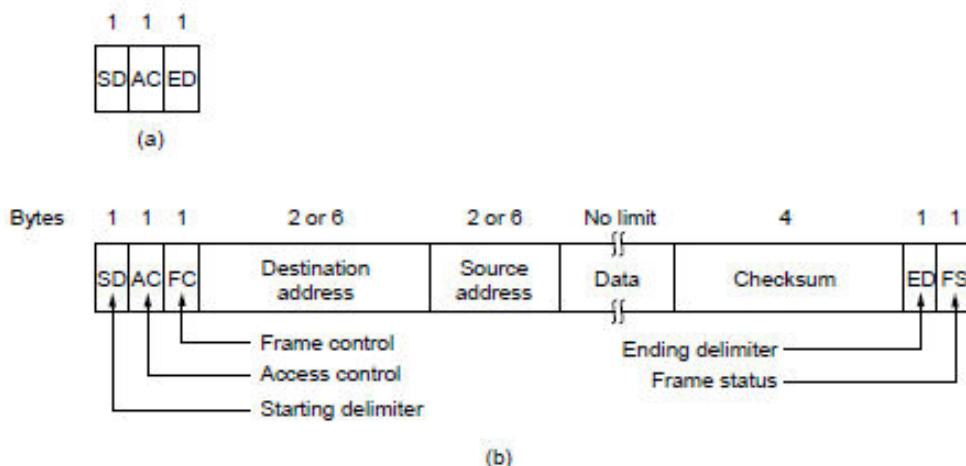


- **Ring physical length** in bits must be large enough to contain token
e.g. 1 km ring with $R = 4$ Mbps, ring physical length is 20 bits, not enough for a 24-bit token
If ring physical length is too small → insert artificial delay to increase it
- Characteristics: for 802.3 Ethernet bus, a minimum frame length is required; for 802.5 token ring, a minimum ring physical length is required

802.5 Token Ring (continue)

- MAC for token ring: with token/frame format as follows, 3-byte token circulates around the ring

- User wishing to transmit must wait for token to arrive and **captures** it (turns token into 3 bytes of a normal frame)
- It can then transmit and, when transmitted frame bits circulate back to sender, it removes them
- After user has finished Tx, it regenerates and **releases** token



- **Access control:** contains token bit (0 for token, 1 for frame), monitor, priority and reservation bits
- **Frame control:** indicate data or control frame, latter for ring maintenance/fault management
- **Token holding time:** a user is allowed to hold token for THT, default value 10 ms
- **Frame Status:** contains *A* (address recognised) and *C* (frame copied) bits, both are reset to logic 0 at sender
 - When receiver recognises destination address as its own, it sets *A* to logic 1, and if receiver is able to copy frame, it sets *C* to logic 1 → This provides automatic ACK for each frame, otherwise receiver cannot do anything as it does not have token

Token Bus and Ring MAC Performance

- Assume n active users and recall the definition of link parameter a
- Maximum utilisation or efficiency is

$$U = \begin{cases} \frac{1}{1+a/n}, & a < 1 \\ \frac{1}{a(1+1/n)}, & a > 1 \end{cases}$$

- Under a heavy load assumption, $n \rightarrow \infty$,

$$U = \begin{cases} 1, & a < 1 \\ \frac{1}{a}, & a > 1 \end{cases}$$

- Comparison of three LANs

- 802.3, 802.4 and 802.5 use roughly similar technology and get roughly similar performance
- Under most circumstances, all three perform well
- 802.3 Ethernet is most popular
- Three standards have three different frame formats → Bridging them can have serious difficulties

Issue	802.3	802.4	802.5
Performance	OK	OK	OK
Simplicity	yes	no	yes
deterministic	no	yes	yes
Priorities	no	yes	yes
Heavy-load Perf	bad	good	good
Reliability	OK	very good	good
User base	large	small	large

Summary

- For broadcast networks, data link layer is divided into medium access control and logical link control sublayers: LLC deals with point-to-point connection issues, and MAC deals with how to access shared medium
- Three medium access strategies are: contention (random access); round-robin and reservation (scheduled access)
- Contention methods: ALOHA, slotted ALOHA, CSMA, CSMA with CD
- Ethernet: CSMA with CD, frame time must be no less than $2 \times$ end-to-end propagation delay → minimum frame length
- Token bus and ring standards (round-robin based): bit has a physical length and therefore minimum ring length is needed
- Comparison of three LANs, how their MACs operate, frame formats, special features

Network Layer: Logical Addressing

Introduction

- Two Types –Physical and Logical
- Physical Addressing
 - Generated by MAC sub-layer, so termed as MAC Address
 - Denotes hardware address of Ethernet card
 - Cannot be changed
 - 48 bit long

Example: A MAC address of **2c549188c9e3** is typically displayed as 2C:54:91:88:C9:E3 or 2c-54-91-88-c9-e3.

Introduction

- Logical Addressing
 - It is logical
 - Changeable
 - Two types – IPv4 and IPv6
 - IPv4 is 32 bit long and IPv6 is 128 bit long

Introduction

- Logical Addressing
 - It is logical
 - Changeable
 - Two types – IPv4 and IPv6
 - IPv4 is 32 bit long and IPv6 is 128 bit long

Internet Addressing:

- To identify **each devices** connected in the Internet is called

Internet addressing or **IP address**.



- **Current version of protocol** is a 32-bit binary address that uniquely and universally defines the connection of a host or a router to the Internet.

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

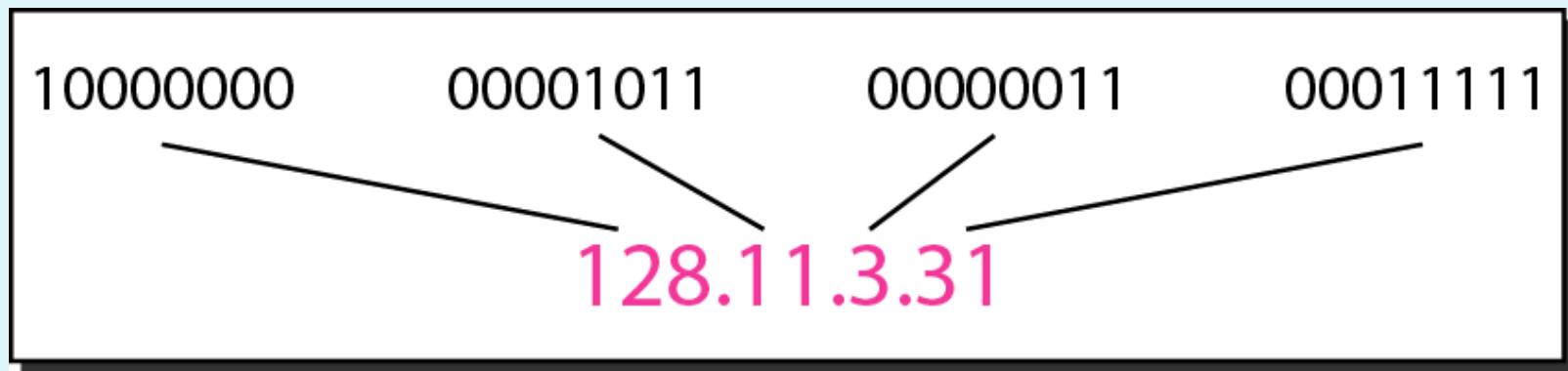
Classful Addressing

Classless Addressing

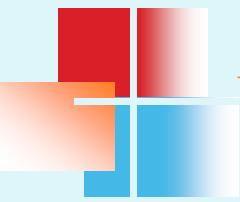
Network Address Translation (NAT)

- ▶ An IPv4 address is 32 bits long.
- ▶ The IPv4 addresses are unique and universal
- ▶ The address space of IPv4 is 2^{32} or
4,294,967,296

Figure 1 Dotted-decimal notation and binary notation for an IPv4 address



128	64	32	16	8	4	2	1
-----	----	----	----	---	---	---	---



Example 1

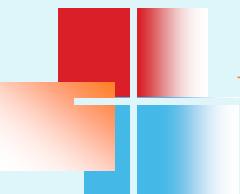
Change the following IPv4 addresses from binary notation to dotted-decimal notation.

- 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 2

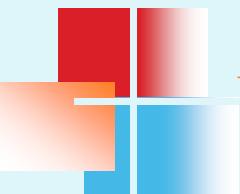
Change the following IPv4 addresses from dotted-decimal notation to binary notation.

- a. 111.56.45.78
- b. 221.34.7.82

Solution

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

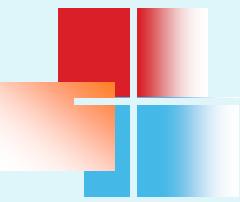
- a. 01101111 00111000 00101101 01001110
- b. 11011101 00100010 00000111 01010010



Example 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



Classful addressing

- Addressing can be divided into classful and classless.
- In **classful addressing**, the address space is divided into five classes:
A, B, C, D, and E.

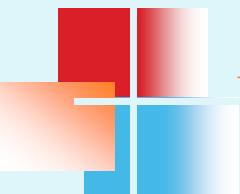
Figure.2 Finding the classes in binary and dotted-decimal notation

	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

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.*

Communication categories:

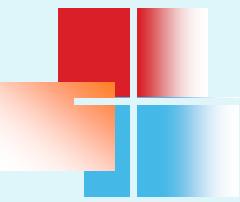
Unicast: one source to one destination(eg. A,B,C classes)

Multicast: one source to a group of destination

Reserved addresses: special purposes.

Table .1 *Number of blocks and block size in classful IPv4 addressing*

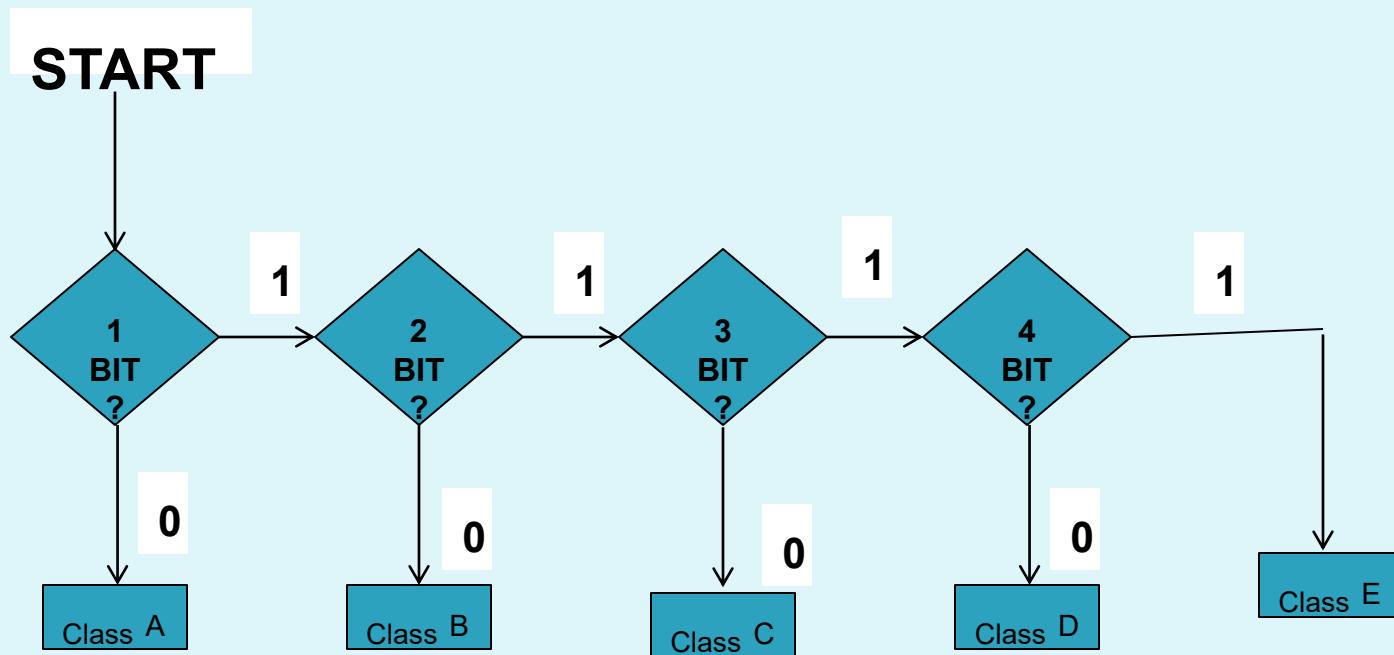
<i>Class</i>	<i>Number of Blocks</i>	<i>Block Size</i>	<i>Application</i>
A	128	16,777,216	Unicast
B	16,384	65,536	Unicast
C	2,097,152	256	Unicast
D	1	268,435,456	Multicast
E	1	268,435,456	Reserved



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

Finding the address class

Binary Notation



Classes and Blocks

- ▶ Class A is divided into 128 blocks having a different netid.
- ▶ First block covers addresses from 0.0.0.0 to 0.255.255.255(netid 0)
- ▶ Second block covers addresses from 1.0.0.0 to 1.255.255.255(netid 1)
- ▶ Last block covers addresses from 127.0.0.0 to 127.255.255.255(netid 127)
- ▶ Number of addresses in each block, 16777216. so many addresses are wasted in this class.

- ▶ Class B is divided into 64 blocks(4194304) having a different netid.
- ▶ First block covers addresses from 128.0.0.0 to 128.0.255.255(netid 128.0)
- ▶ Last block covers addresses from 191.255.0.0 to 191.255.255.255(netid 191.255)
- ▶ Number of addresses in each block, 65536. So many addresses are wasted in this class.

Classful Addressing

- Netid and Hostid
 - IP address in class A,B or C is divided netid and hostid
 - These are of varying lengths depending on class of address
 - This concept is not applicable for class D and E

Netid and hostid

Class	Byte 1	Byte 2	Byte 3	Byte 4
Class A	Netid		Hostid	
Class B		Netid		Hostid
Class C		Netid		Hostid
Class D		Multicast Address		
Class E		Reserved for Future use		

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.

Classful Addressing

- Netid and Hostid
 - In class A, first one byte define netid and three bytes define the hostid
 - In class B, first two bytes define netid and two bytes define hostid
 - In class C, first three bytes define netid and one byte defines the hostid

Classful Addressing

- Mask
 - Length of netid and hostid is predefined, still we can also use a mask (sometimes called as default mask)

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

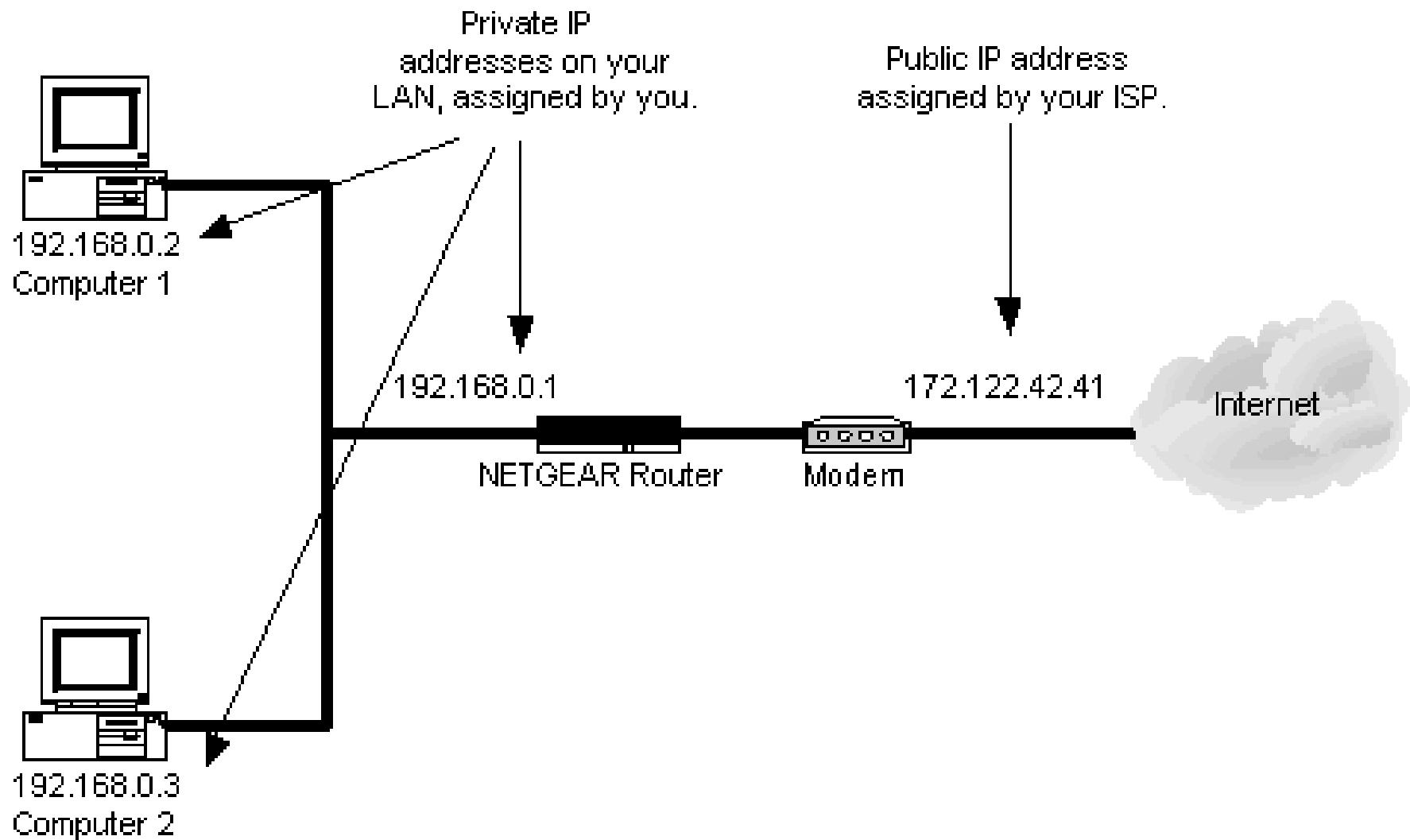
Default masks for classful addressing

**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.

CIDR Notation

- Mask is written in the form /n, where n can be 8,16,24
- This notation is also called slash notation or Classless Interdomain Routing (CIDR) notation
- Used in classless addressing

Network address



Network Address

- Several properties
 - All hostid bytes are 0
 - It defines the network to the rest of the Internet
 - First address of the block
 - From network address, class of address can be determined

Example:

- ❖ Given the address 23.56.7.91, find network address

Soln: The class is A. Only the first byte defines the netid. We can find the network address by replacing the hostid bytes (56.7.91) with 0s. So, network address is 23.0.0.0

Network address

- ▶ Given the address 23.54.22.213, find the network address.
 - The class is A, which has only one byte of netid, therefore 23.0.0.0 is the network address.
- ▶ Given the address 193.54.22.213, find the network address.
 - The class is C, which has three bytes of netid, therefore the network address is 193.54.22.0.
 - If network address is given, find the class by using the first byte.

Subnetting

- Subnetting
 - If an organization was granted a large block in class A or B, it could divide the addresses into several contiguous groups
 - These groups are assigned to smaller networks (subnets)
 - Subnet increases the number of 1s in the mask

Subnetting

- ▶ Example 1: university (netid), group of host according to the department(subnet)
- ▶ Example 2: Telephone number with STD code.
- ▶ Subnet has three levels: site, subnet, host
- ▶ 23.54.222.12 in this 23.0.0.0 is this netid and 54 is subnetid.

Subnet mask

- ▶ The subnet mask is used by the TCP/IP protocol to determine whether a host is on the local subnet or on a remote network.
- ▶ In TCP/IP, the parts of the IP address that are used as the network and host addresses are not fixed.
- ▶ Need more information supplied in another 32-bit number called a **subnet mask** (eg. 255.255.255.0).

Subnet mask

- ▶ Lining up the IP address and the subnet mask together, the network, and host portions of the address can be separated:
- ▶ 1100000010101000011101110000100 – IP address (192.168.123.132)
11111111111111111111111100000000 – Subnet mask (255.255.255.0)
- ▶ The first 24 bits (the number of ones in the subnet mask) are identified as the network address. The last 8 bits (the number of remaining zeros in the subnet mask) are identified as the host address.

Subnet mask

- ▶ It gives you the following addresses:
- ▶ 110000010101000111101100000000 – Network address (192.168.123.0)
- 00000000000000000000000010000100 – Host address (000.000.000.132)

Subnet mask

- ▶ It gives you the following addresses:
- ▶ 110000001010100011101100000000 -
Network address (192.168.123.0)

0000000000000000000000000000000010000100 - Host
address (000.000.000.132)

- ▶ When a packet arrives on the 192.168.123.0 subnet (from the local subnet or a remote network), and it has a destination address of 192.168.123.132, your computer will receive it from the network and process it.

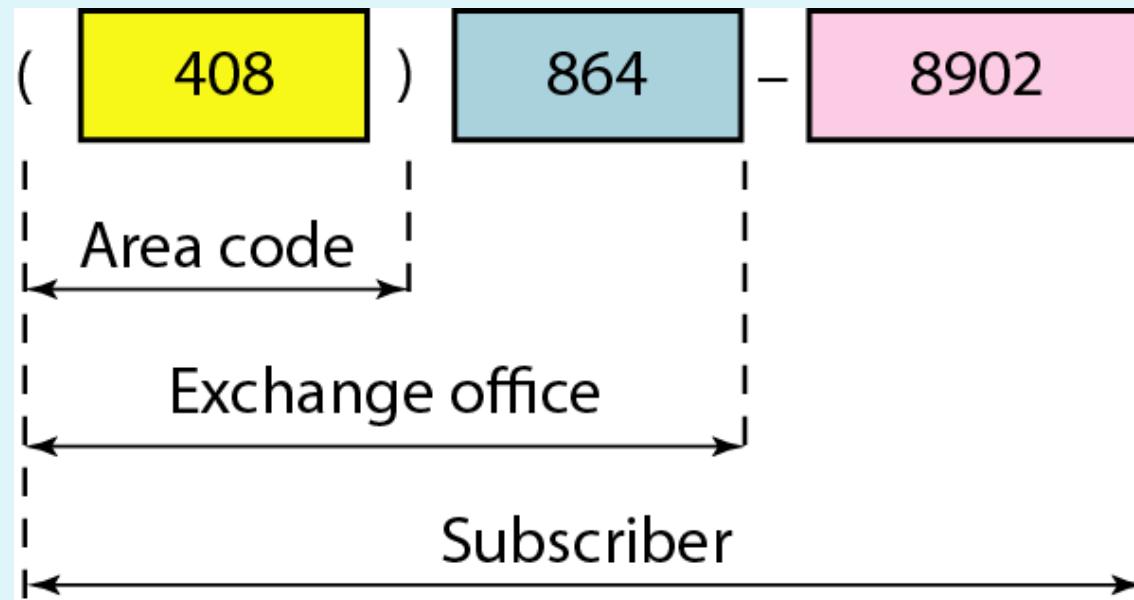
Subnetting

- ▶ Class A, B, or C TCP/IP network can be further divided, or subnetted, by a system administrator.
- ▶ 50 hosts on three networks that are connected by a TCP/IP router.
- ▶ Each of these are in a networks has 50 hosts. You are allocated in network 192.168.123.0.

Subnetting

- ▶ You are allocated the class C network 192.168.123.0.
- ▶ It means that you can use the addresses 192.168.123.1 to 192.168.123.254 for your 150 hosts.
- ▶ 192.168.123.0 and 192.168.123.255 are invalid.

Figure 3 Two levels of hierarchy in an IPv4 address



Mask

- ▶ It is 32-bit number key that helps the router that is inside the organization is called **subnet mask** as well as outside the organization called **default mask**.
- ▶ **Default mask** is a 32-bit number which is ANDed with an IP address gives network address.

11111111	11111111	00000000	00000000
----------	----------	----------	----------

Default mask
255.255.0.0

11111111	11111111	11100000	00000000
----------	----------	----------	----------

Subnet mask
255.255.224.0

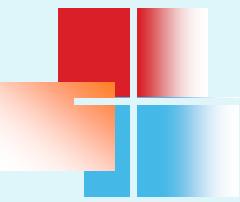
Default masks for classful addressing

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

Classless Inter-Domain Routing- CIDR

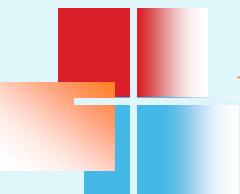
Supernetting

- ▶ Class C block has a maximum of 256 addresses which is not sufficient for some organization. **Supernetting** is the solution for it, which can combines several class C blocks to create a large range of addresses



Note

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

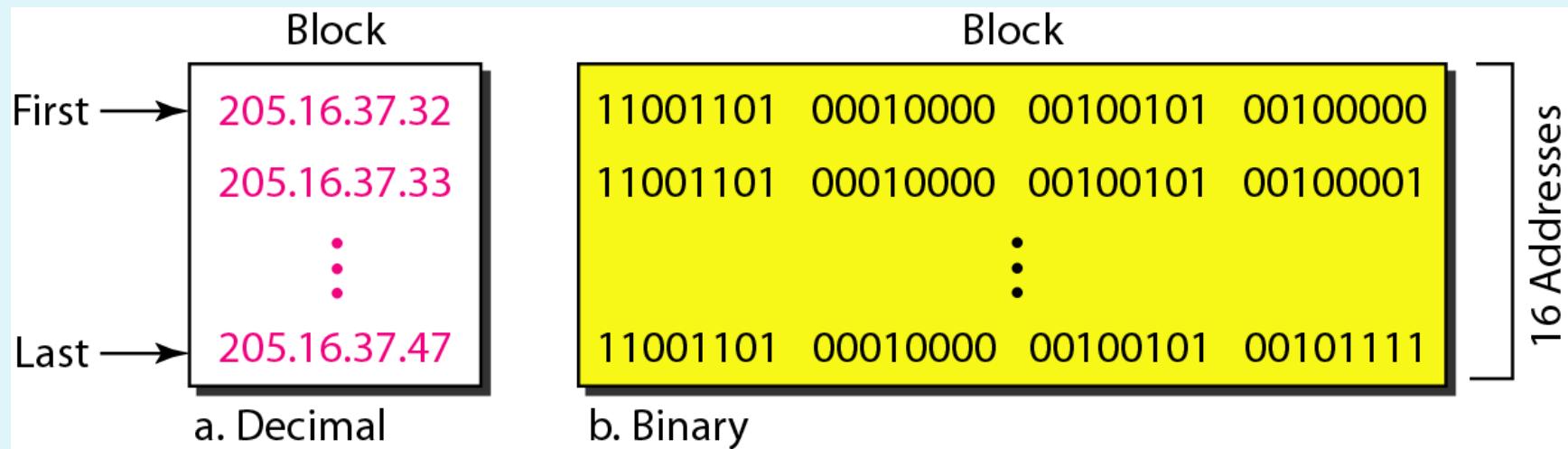


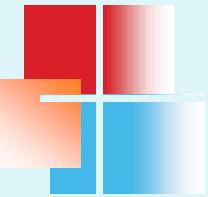
Example 5

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

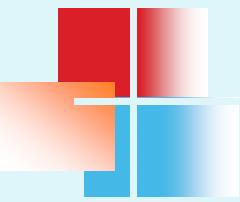
We can see that the restrictions are applied to this block. The addresses are contiguous. The number of addresses is a power of 2 ($16 = 2^4$), and the first address is divisible by 16. The first address, when converted to a decimal number, is 3,440,387,360, which when divided by 16 results in 215,024,210.

Figure .4 A block of 16 addresses granted to a small organization



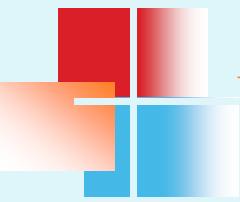


In IPv4 addressing, a block of addresses can be defined as x.y.z.t /n in which x.y.z.t defines one of the addresses and the /n defines the mask.



Note

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



Example 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

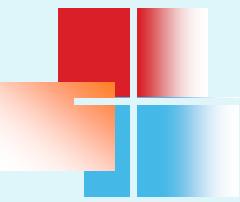
If we set 32–28 rightmost bits to 0, we get

11001101 00010000 00100101 00100000

or

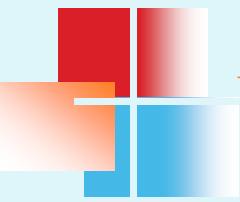
205.16.37.32.

This is actually the block shown Figure .3



Note

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



Example 7

Find the last address for the block in Example 19.6.

Solution

The binary representation of the given address is

11001101 00010000 00100101 00100111

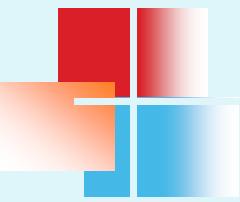
If we set 32 – 28 rightmost bits to 1, we get

11001101 00010000 00100101 00101111

or

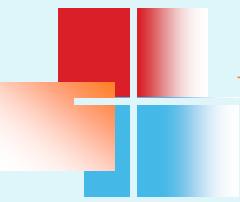
205.16.37.47

This is actually the block shown in Figure 19.3.



Note

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

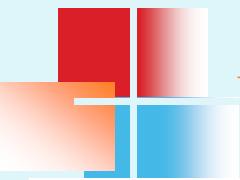


Example 8

Find the number of addresses in Example 19.6.

Solution

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



Example 9

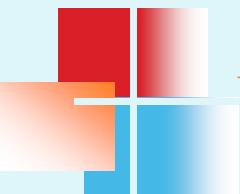
Another way to find the first address, the last address, and the number of addresses is to represent the mask as a 32-bit binary (or 8-digit hexadecimal) number. This is particularly useful when we are writing a program to find these pieces of information. In Example .5 the /28 can be represented as

11111111 11111111 11111111 11110000

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

Find

- a. *The first address*
- b. *The last address*
- c. *The number of addresses.*

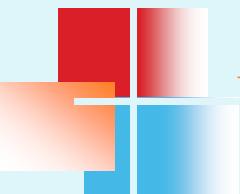


Example (continued)

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.

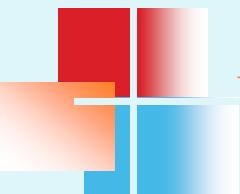
Address:	11001101	00010000	00100101	00100111
Mask:	11111111	11111111	11111111	11110000
First address:	11001101	00010000	00100101	00100000



Example 9 (continued)

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



Example 9 (continued)

- 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 5 A network configuration for the block 205.16.37.32/28

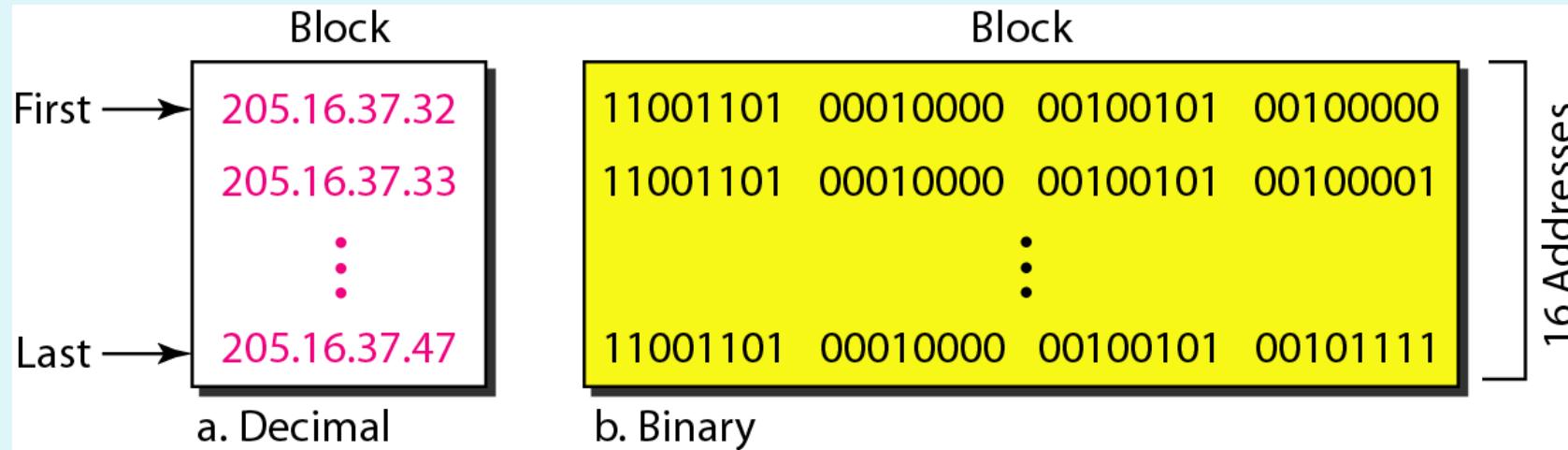
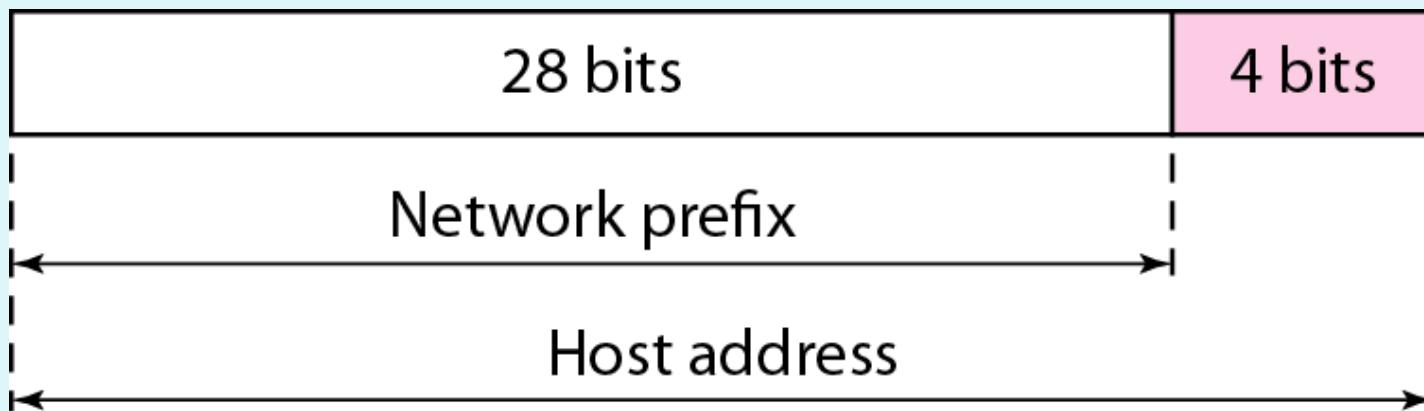
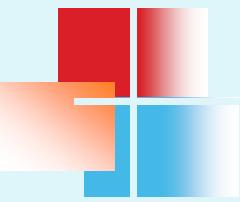


Figure 6 A frame in a character-oriented protocol





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 7 Configuration and addresses in a subnetted network

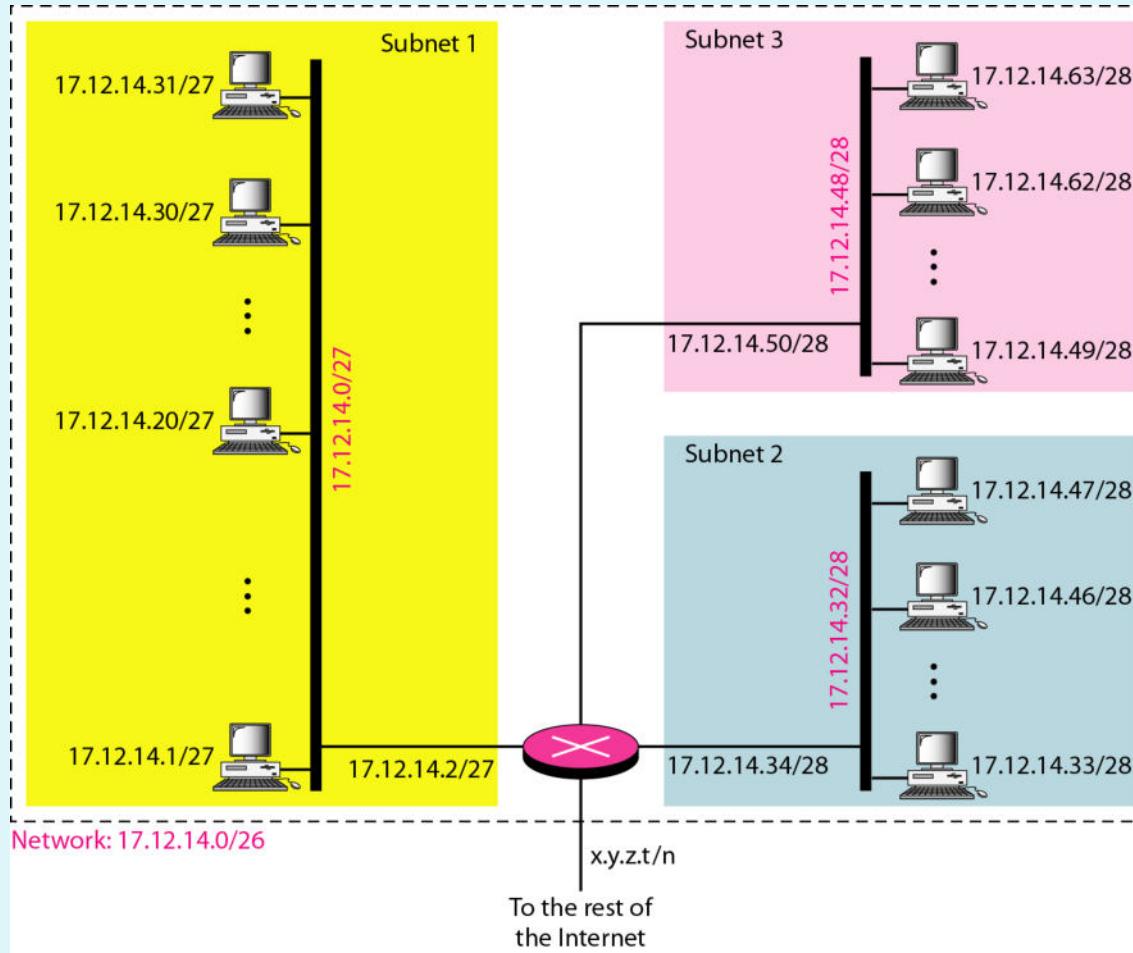
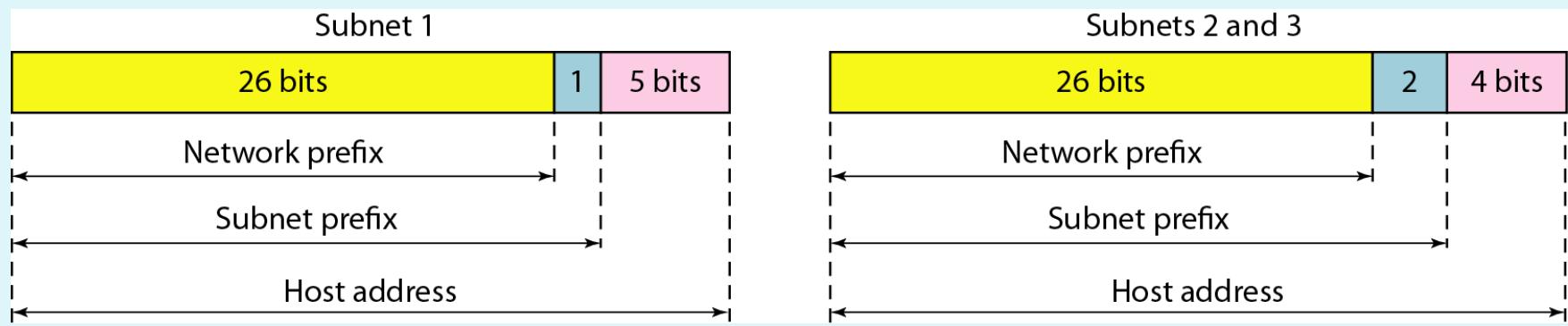
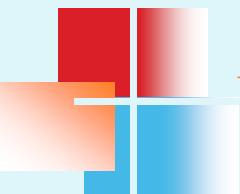


Figure 8 Three-level hierarchy in an IPv4 address



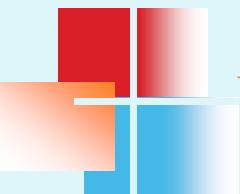


Example 10

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.



Example 10 (continued)

Solution

Figure 19.9 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

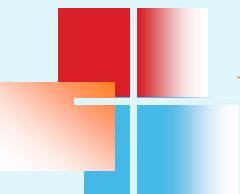
1st Customer: 190.100.0.0/24 190.100.0.255/24

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 = $64 \times 256 = 16,384$



Example 10 (continued)

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

<i>1st Customer:</i>	190.100.64.0/25	190.100.64.127/25
<i>2nd Customer:</i>	190.100.64.128/25	190.100.64.255/25
...		
<i>128th Customer:</i>	190.100.127.128/25	190.100.127.255/25
<i>Total = $128 \times 128 = 16,384$</i>		

Example 10 (continued)

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 An example of address allocation and distribution by an ISP

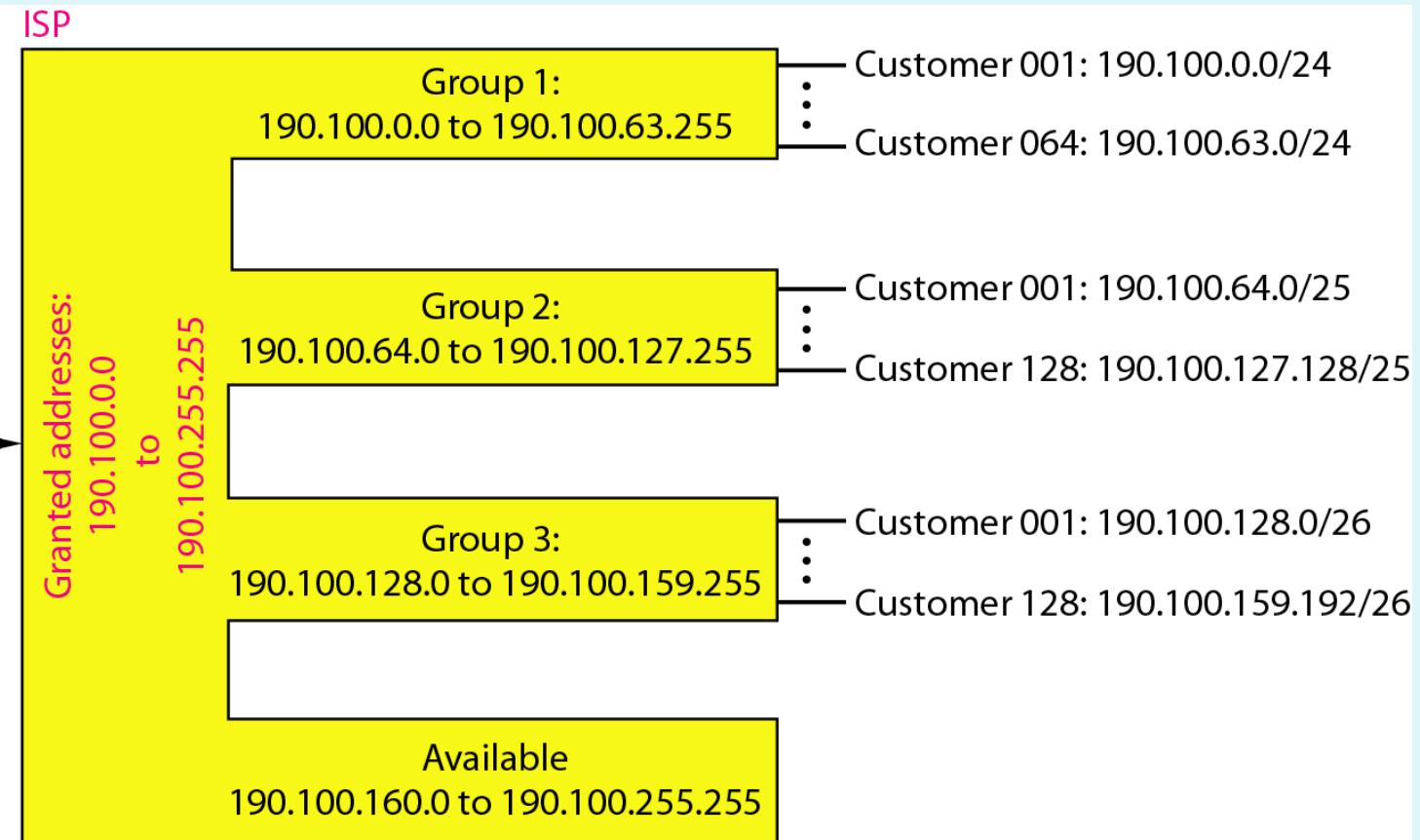


Table 19.3 *Addresses for private networks*

<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}

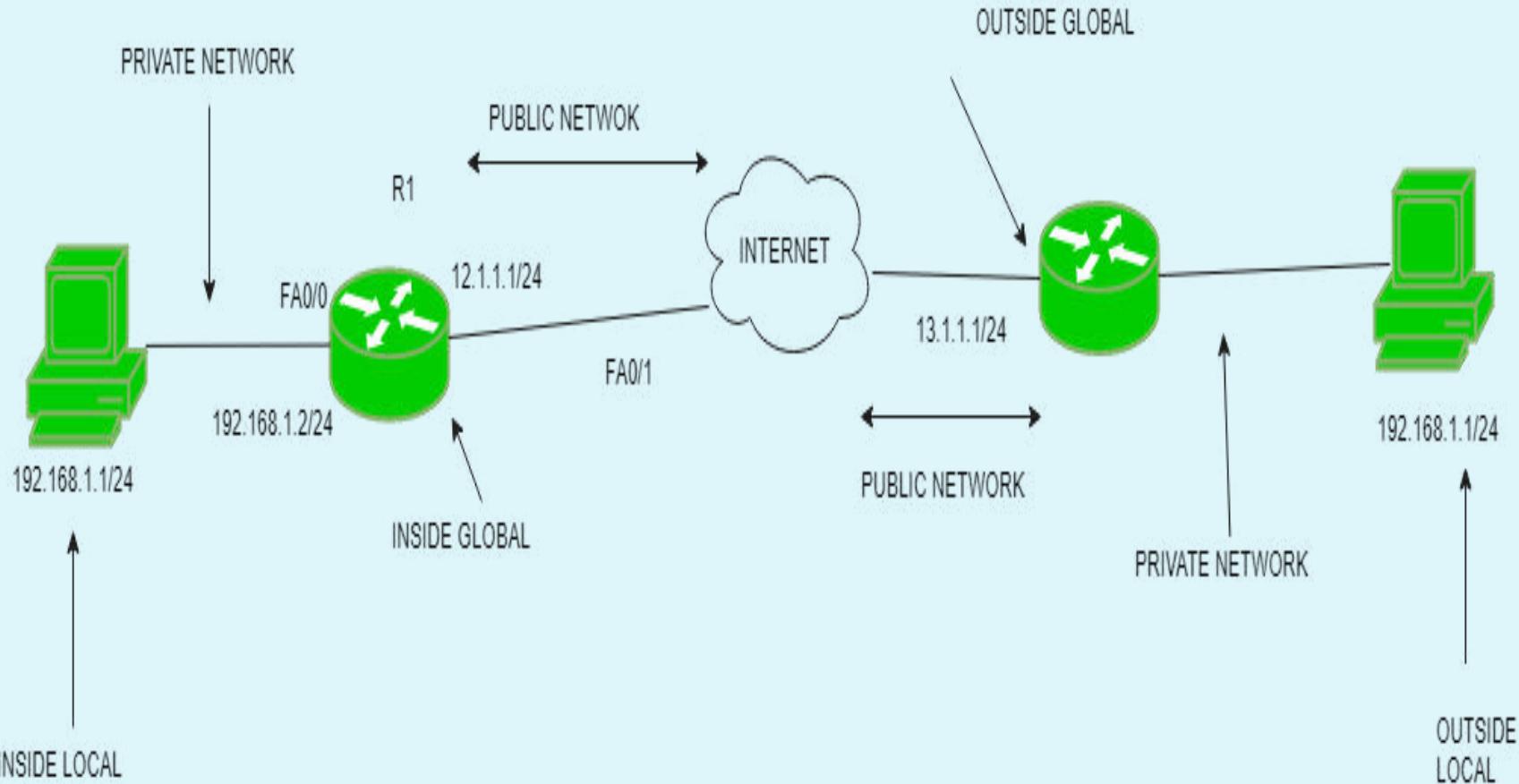
Network Address Translation (NAT)

- ▶ Process in which one or more local IP address is translated into one or more Global IP address and vice versa in order to provide Internet access to the local hosts.
- ▶ Also, it does the translation of port numbers i.e. masks the port number of the host with another port number, in the packet that will be routed to the destination.
- ▶ It then makes the corresponding entries of IP address and port number in the NAT table.
- ▶ NAT generally operates on a router or firewall.

Network Address Translation (NAT) working

- ▶ Border router is configured for NAT
- ▶ Router which has one interface in the local (inside) network and one interface in the global (outside) network

NAT inside and outside addresses



NAT inside and outside addresses

- ▶ Inside local address
- ▶ Inside global address
- ▶ Outside local address
- ▶ Outside global address

Network Address Translation (NAT) Types

- ▶ Static NAT
- ▶ Dynamic NAT
- ▶ Port Address Translation (PAT)

- ▶ **Advantages of NAT -**
- ▶ NAT conserves legally registered IP addresses.
- ▶ It provides privacy as the device's IP address, sending and receiving the traffic, will be hidden.
- ▶ Eliminates address renumbering when a network evolves.
- ▶ **Disadvantage of NAT:**
- ▶ Translation results in switching path delays.
- ▶ Certain applications will not function while NAT is enabled.
- ▶ Complicates tunneling protocols such as IPsec.
- ▶ Also, the router being a network layer device, should not tamper with port numbers(transport layer) but it has to do so because of NAT

Figure 10 A NAT implementation

Site using private addresses

172.18.3.1 172.18.3.2 172.18.3.20



...

172.18.3.30

NAT router

200.24.5.8

Internet

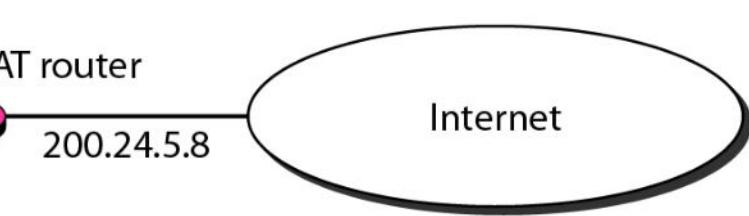


Figure 11 Addresses in a NAT

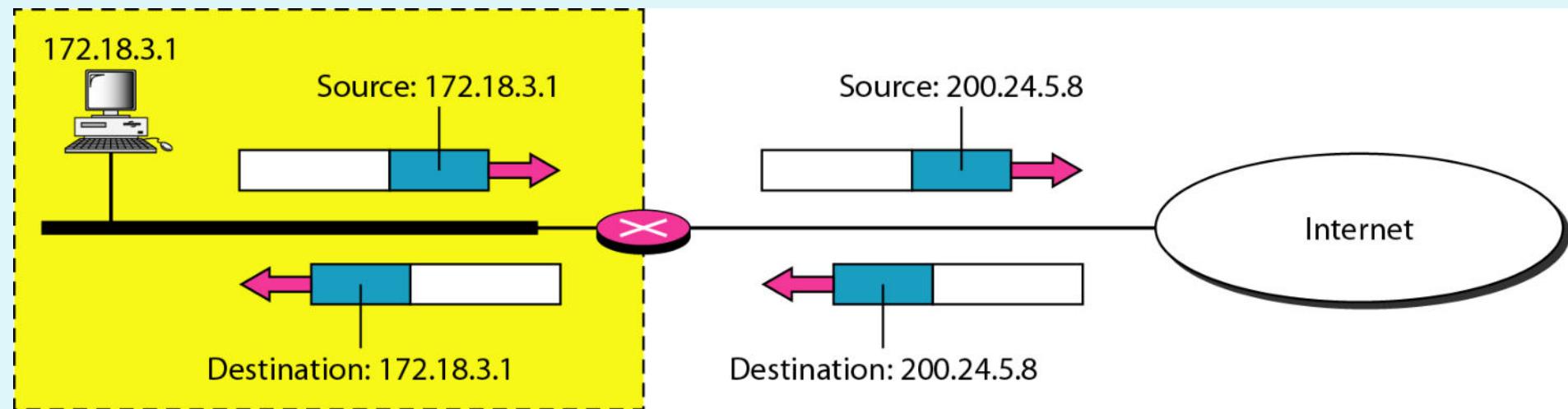


Figure 12 NAT address translation

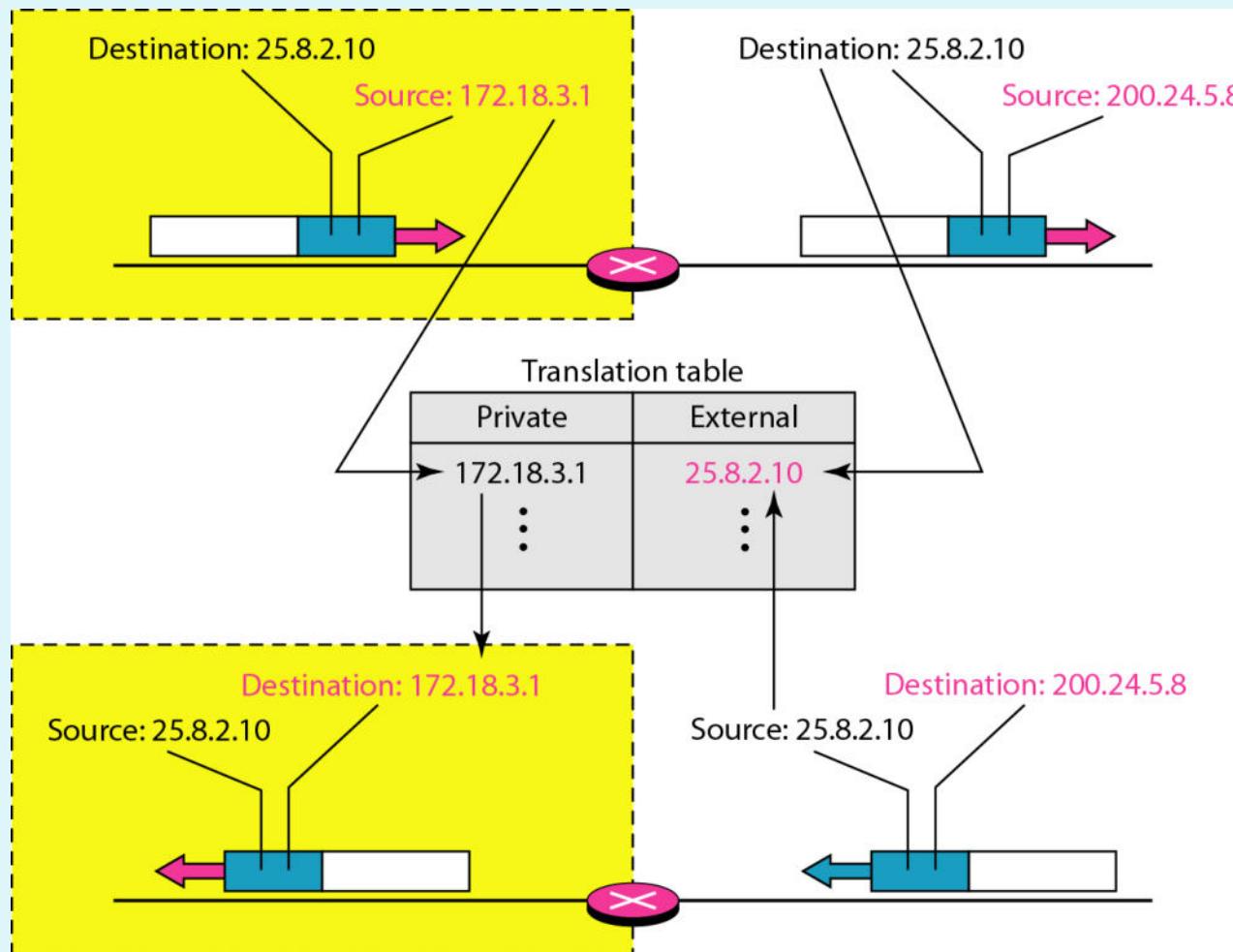
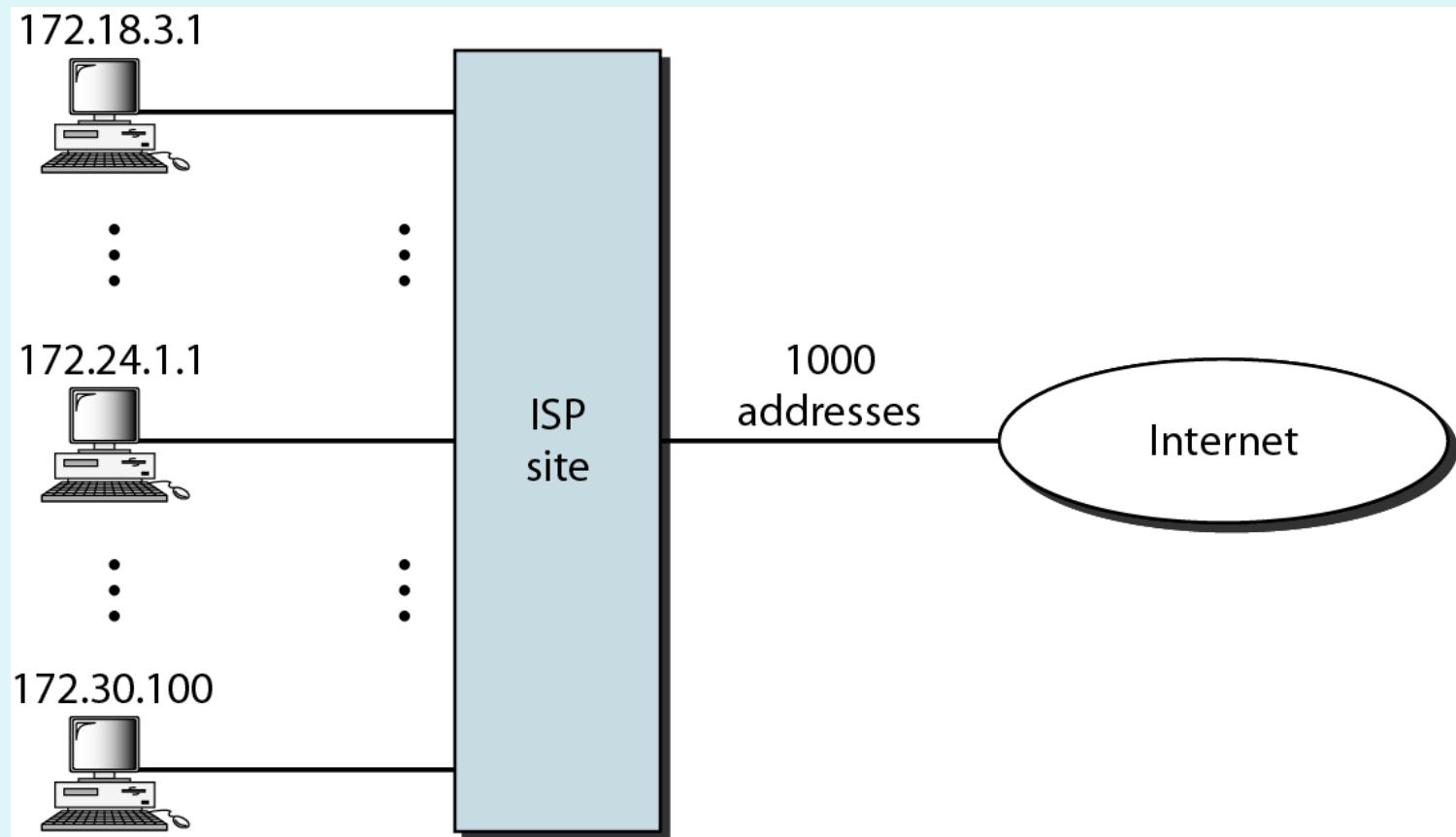


Table 4 Five-column translation table

<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 13 An ISP and NAT



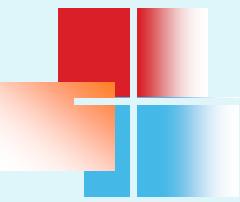
IPv6 ADDRESSES

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 14 IPv6 address in binary and hexadecimal colon notation

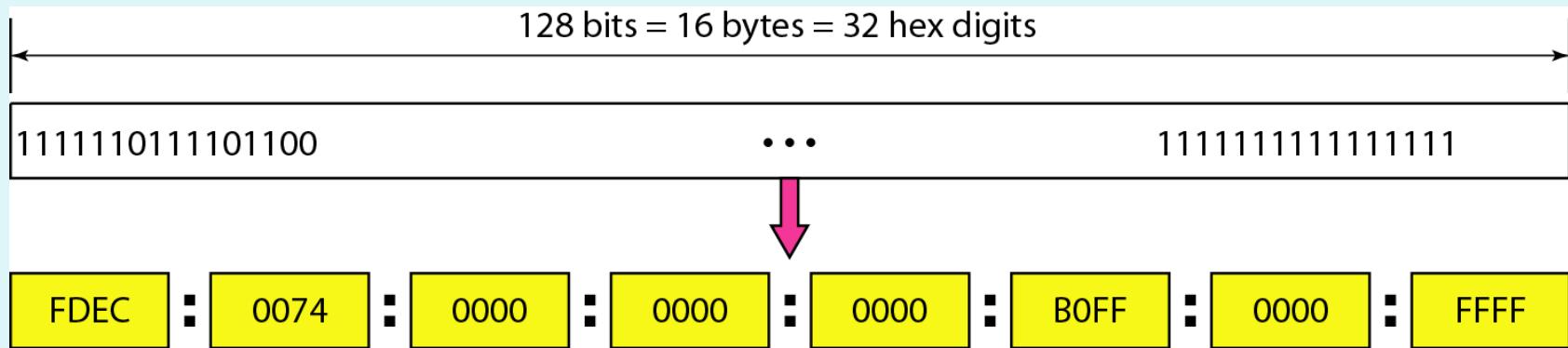
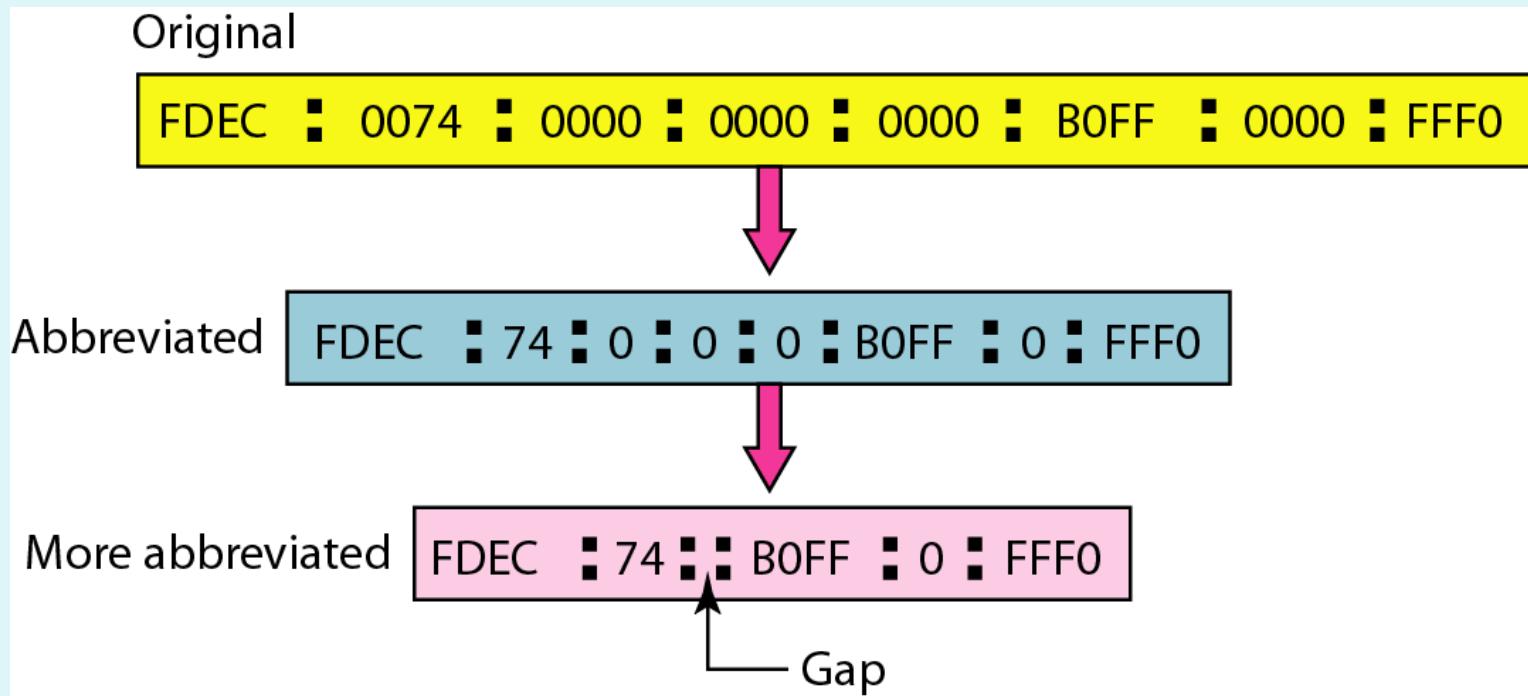


Figure 15 Abbreviated IPv6 addresses



Example 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:XXXX

0: 15: : 1: 12:1213

This means that the original address is.

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

Table 19.5 *Type prefixes for IPv6 addresses*

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 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 16 Prefixes for provider-based unicast address

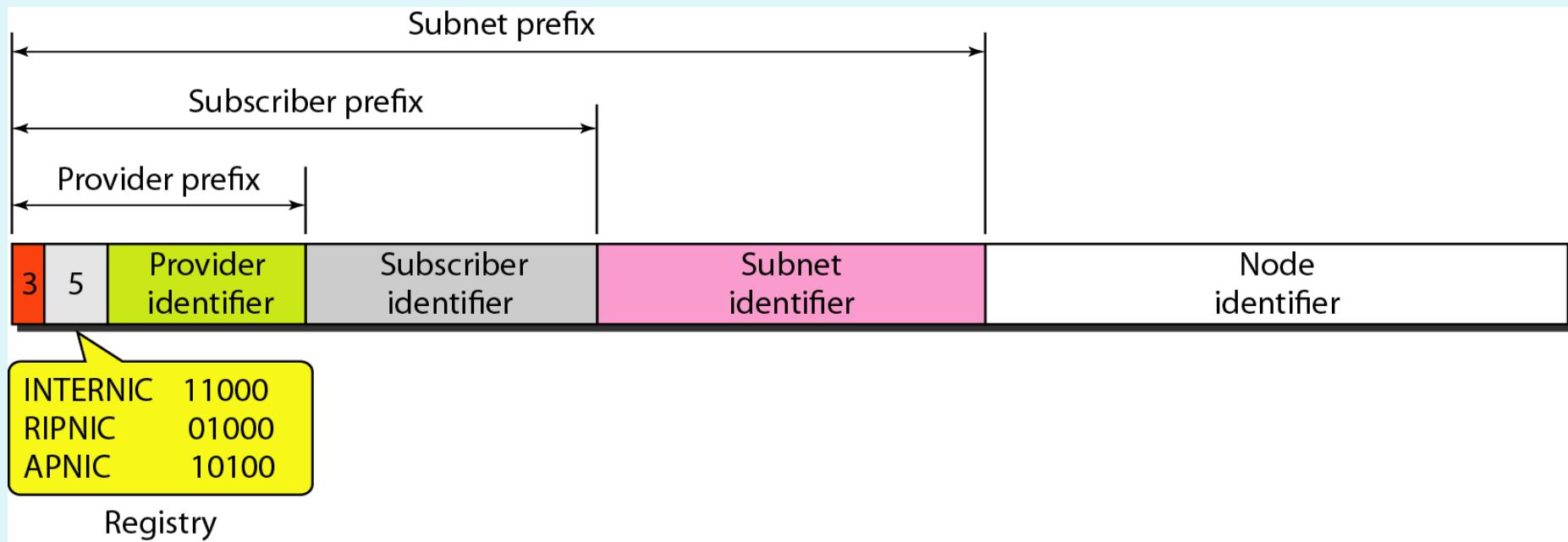


Figure 17 Multicast address in IPv6

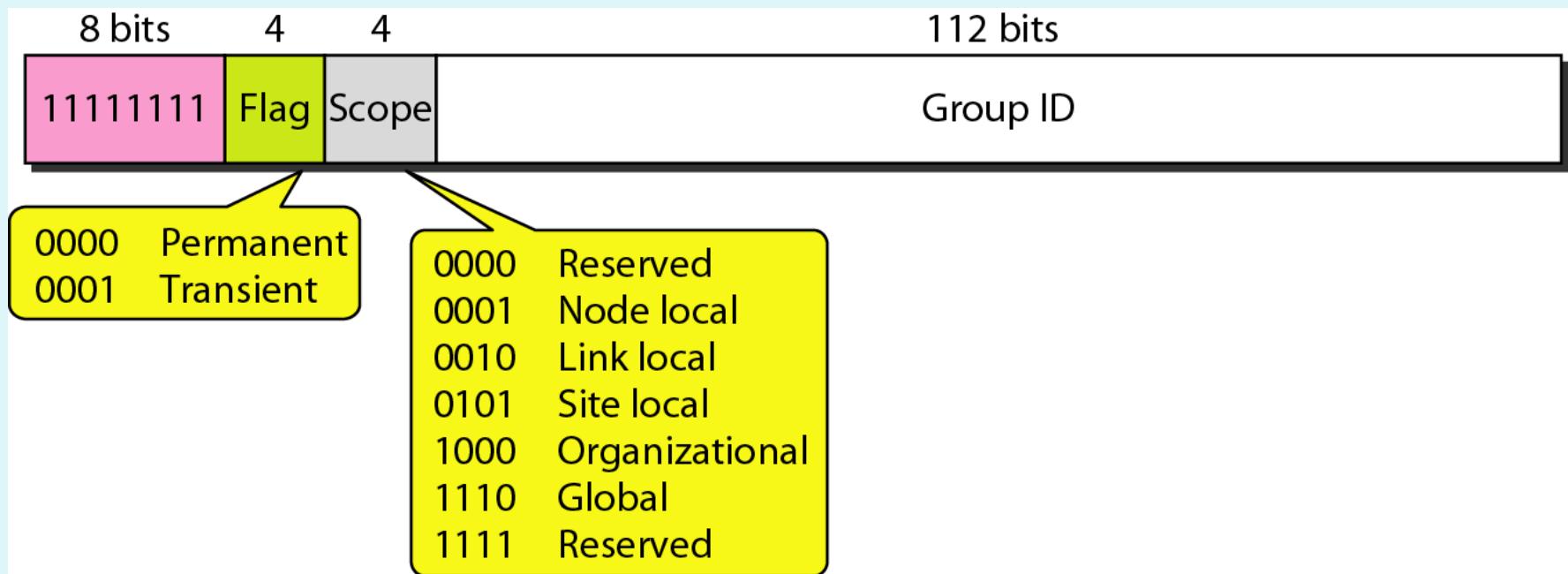


Figure 18 *Reserved addresses in IPv6*

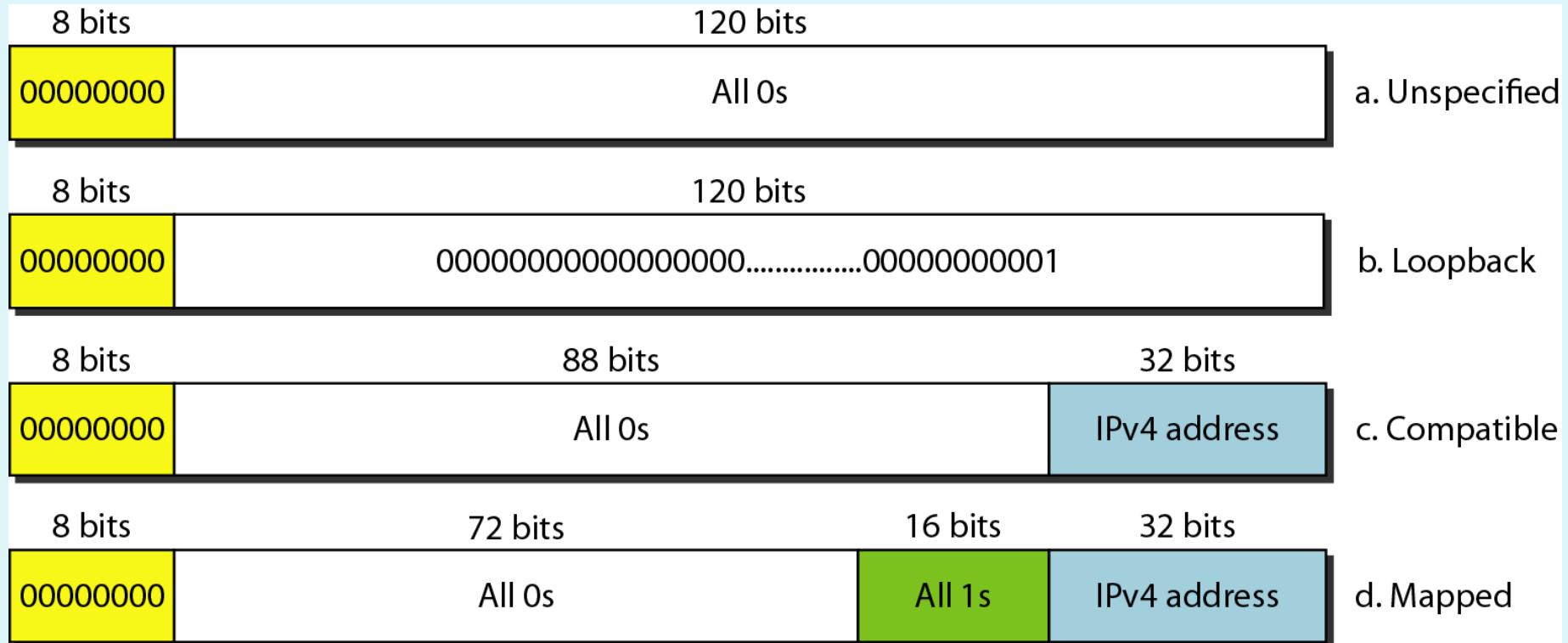
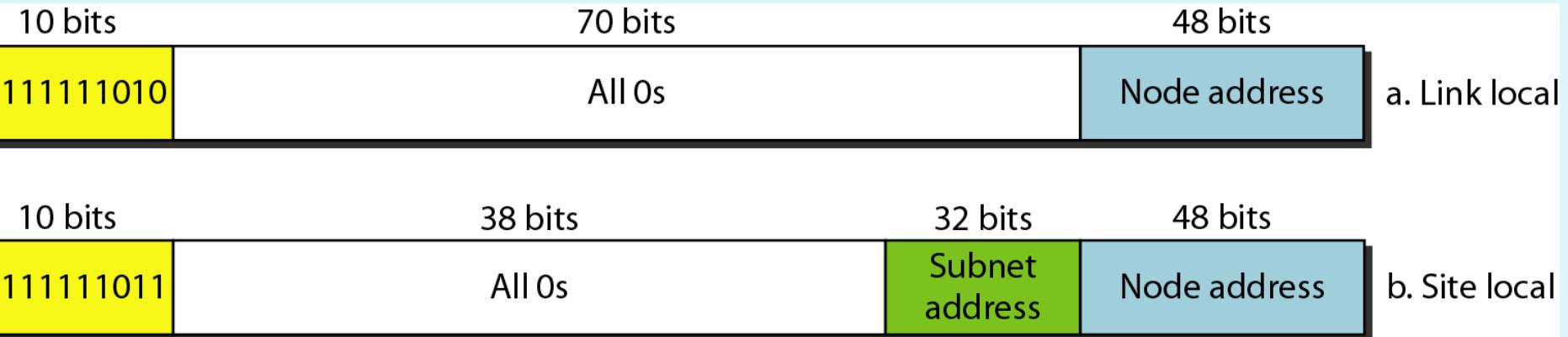
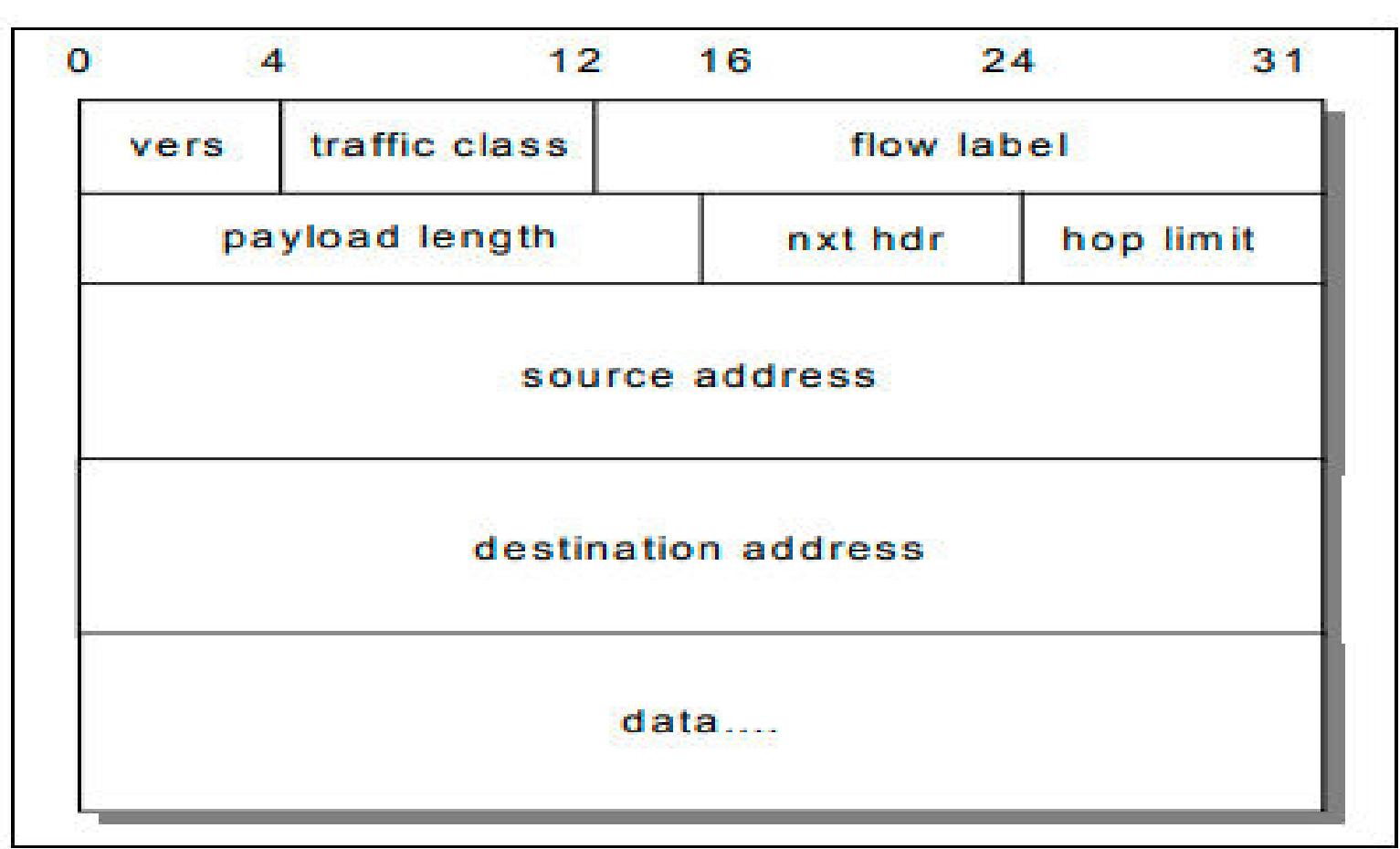
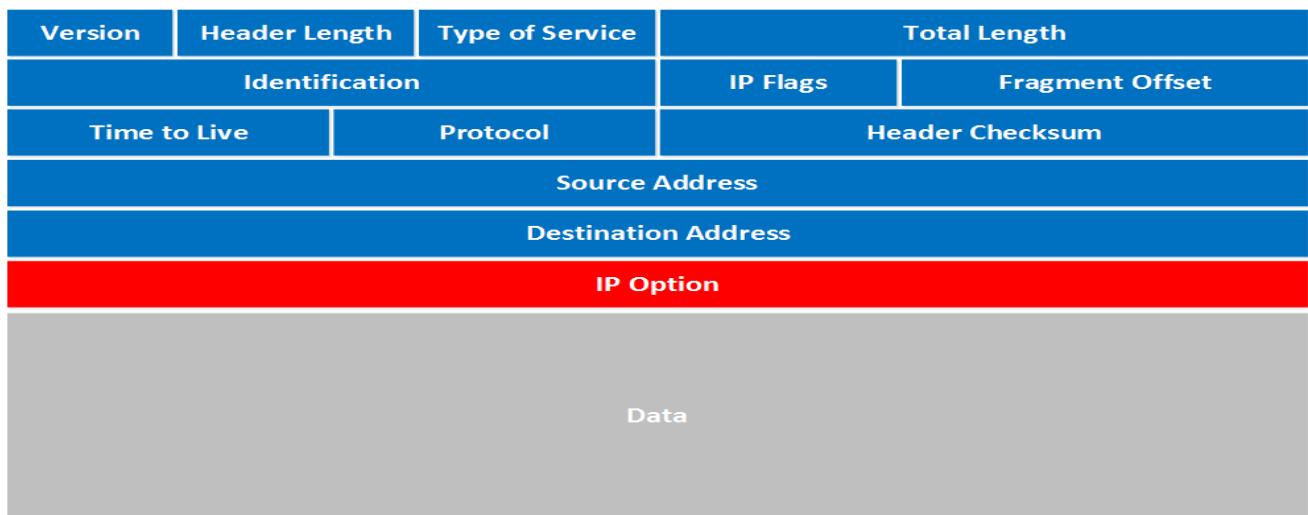


Figure 19 Local addresses in IPv6



IPv6 header format





Module 5

Routing Protocols

Dr PUNITHA K

Overview

- Routing - Link State and Distance Vector Routing Protocols- Implémentation - Performance Analyses - Packet Tracer

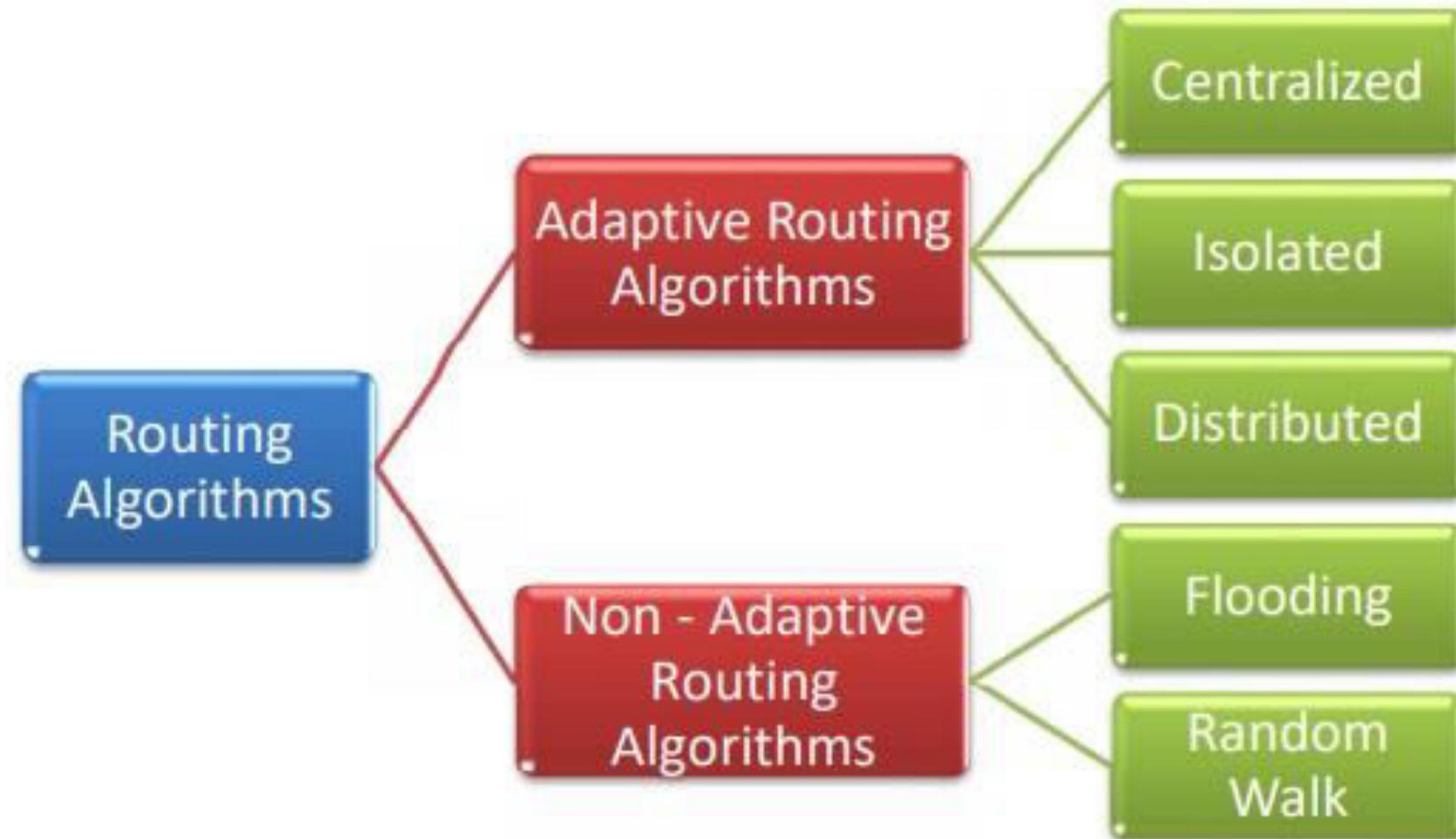
Introduction

- Routing is a process which is performed by layer 3 (or network layer) devices in order to deliver the packet by choosing an optimal path from one network to another.

Routing

- Process of establishing the routes that data packets to reach the destination.
- In this process, a routing table is created which contains information regarding routes which data packets follow.
- Various routing algorithm are used for the purpose of deciding which route an incoming data packet needs to be transmitted on to reach destination efficiently.
- Routing algorithm mathematically computes the best path, i.e. “least – cost path” that the packet can be routed through

Types of Routing Algorithms



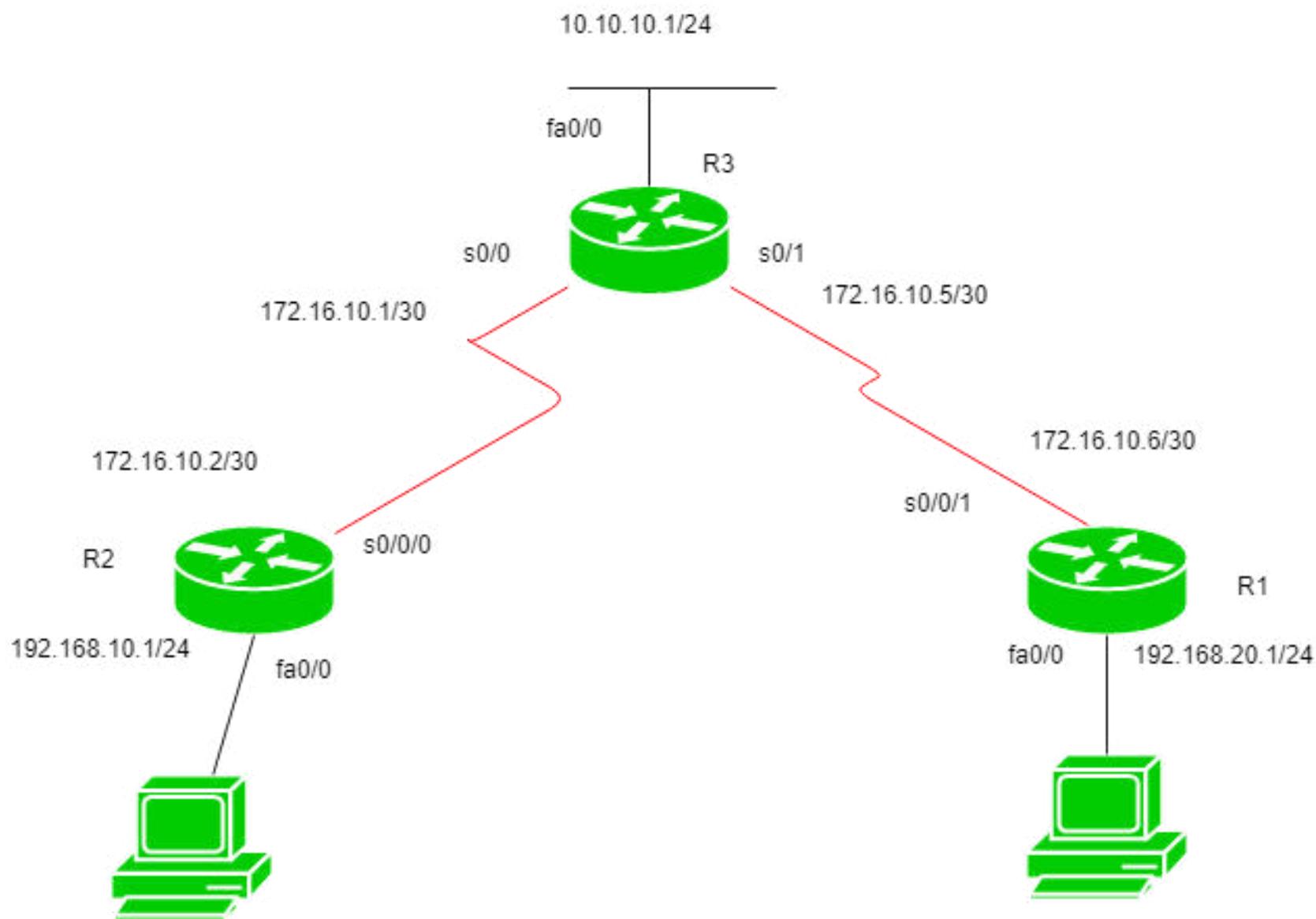
Adaptive Routing Algorithms

- Dynamic routing algorithms, makes routing decisions dynamically depending on the network conditions.
- Routing table is constructs depending on network traffic and topology.
- Try to compute the optimized route depending upon the hop count, transit time and distance.
- Types of adaptive routing algorithms are –
- **Centralized algorithm** – finds the least-cost path between source and destination nodes by using global knowledge about the network. (Global routing algorithm)
- **Isolated algorithm** – procures routing information by using local information instead of gathering information from other nodes.
- **Distributed algorithm** – decentralized algorithm that computes the least-cost path between source and destination iteratively in a distributed manner.

Non – Adaptive Routing Algorithms

- **Static routing algorithms**, construct a static routing table to determine the path through which packets are to be sent.
- Static routing table is constructed based upon the routing information stored in the routers when the network is booted up.
- Types of non – adaptive routing algorithms are –
- **Flooding** – when a data packet arrives at a router, it is sent to all the outgoing links except the one it has arrived on. Flooding may be uncontrolled, controlled or selective flooding.
- **Random walks (probabilistic algorithm)** where a data packet is sent by the router to any one of its neighbors randomly.

Configuration



Routing v/s Flooding

Routing	Flooding
--> Routing table is required.	--> No routing table is required.
--> May give shortest path.	--> Always gives shortest path.
--> Less reliable.	--> More reliable.
--> Traffic is less.	--> Traffic is high.
--> No duplicate packets.	--> Duplicate packets are present

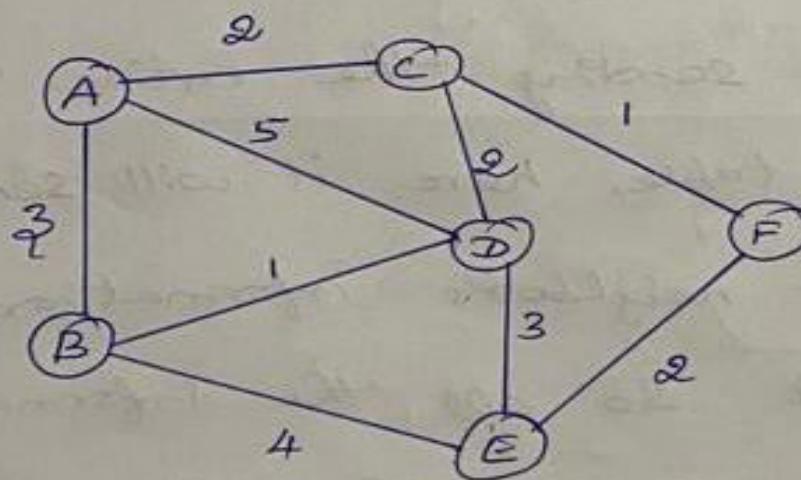
Link State Routing

⇒ To find the shortest path from one node to all other nodes.

Two phases:

- 1) Reliable Flooding: After finding the link state, that will be flooded to all the nodes in the network.
- 2) Route calculation: Each node uses Dijkstra's algorithm to find the optimal path.

Example 6:



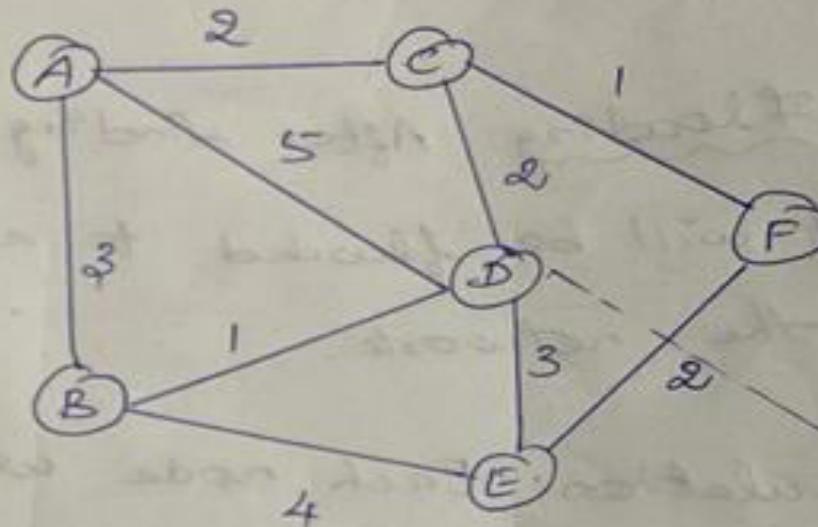
Find the link state: is finding the information regarding the node, that is the neighboring node and the cost to reach.

Link state table:

node	cost
B	3
C	2

4 / 5

node	cost
A	2
D	2
F	1



node	cost
A	3
E	4
D	1

node	cost
C	1
E	2

node	cost
B	4
D	3
F	2

node	cost
A	5
B	1
C	2
E	3

Instead of sending the entire routing table, here it will send only the nearest neighbor's information across the network. So all the information will be share across the network. For example, Node A will get the information about node C, B, D in which their not neighbor node detail are available. That Node A gets all the neighboring node information. Flooding of link state information is flooded to all the nodes in the network.

Finding the root cost:

By using Dijkstra's algorithm shortest path is find.

Dijkstra Algorithm:

```
Tree = {root}
for (y = 1 to N)
    if (y is the root)
        D(y) = 0
    else if (y is the neighbor)
        D(y) = c[root][y]
    else
        D(y) = ∞
```

}

repeat

{

 find a node with $D[w]$ minimum

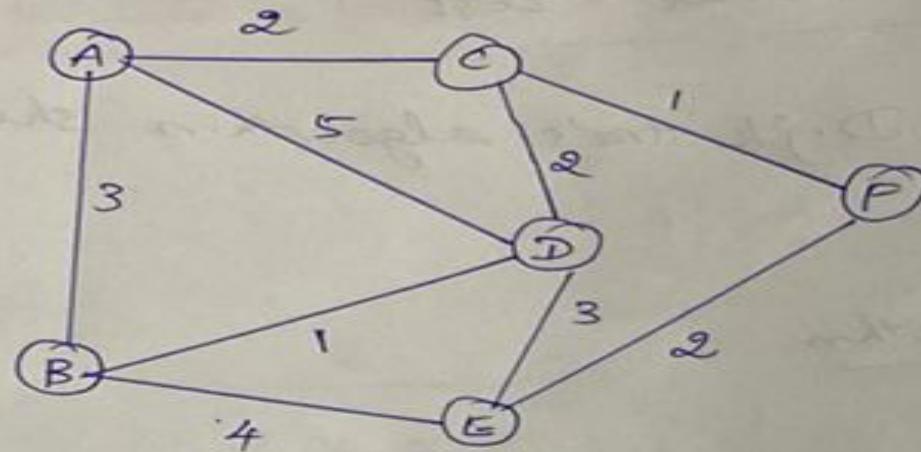
$$\text{Tree} = \text{Tree} \cup \{w\}$$

 for (every node x , which is a neighbor of w)

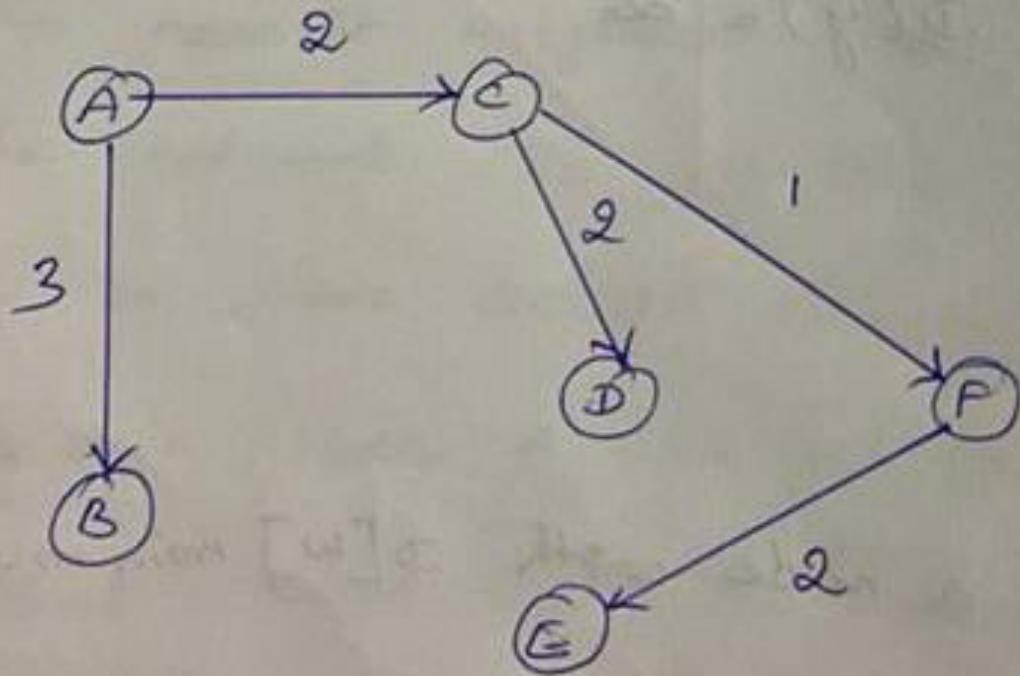
$$D[x] = \min \{ D[x], D[w] + c[w, x] \}$$

} update all nodes

}



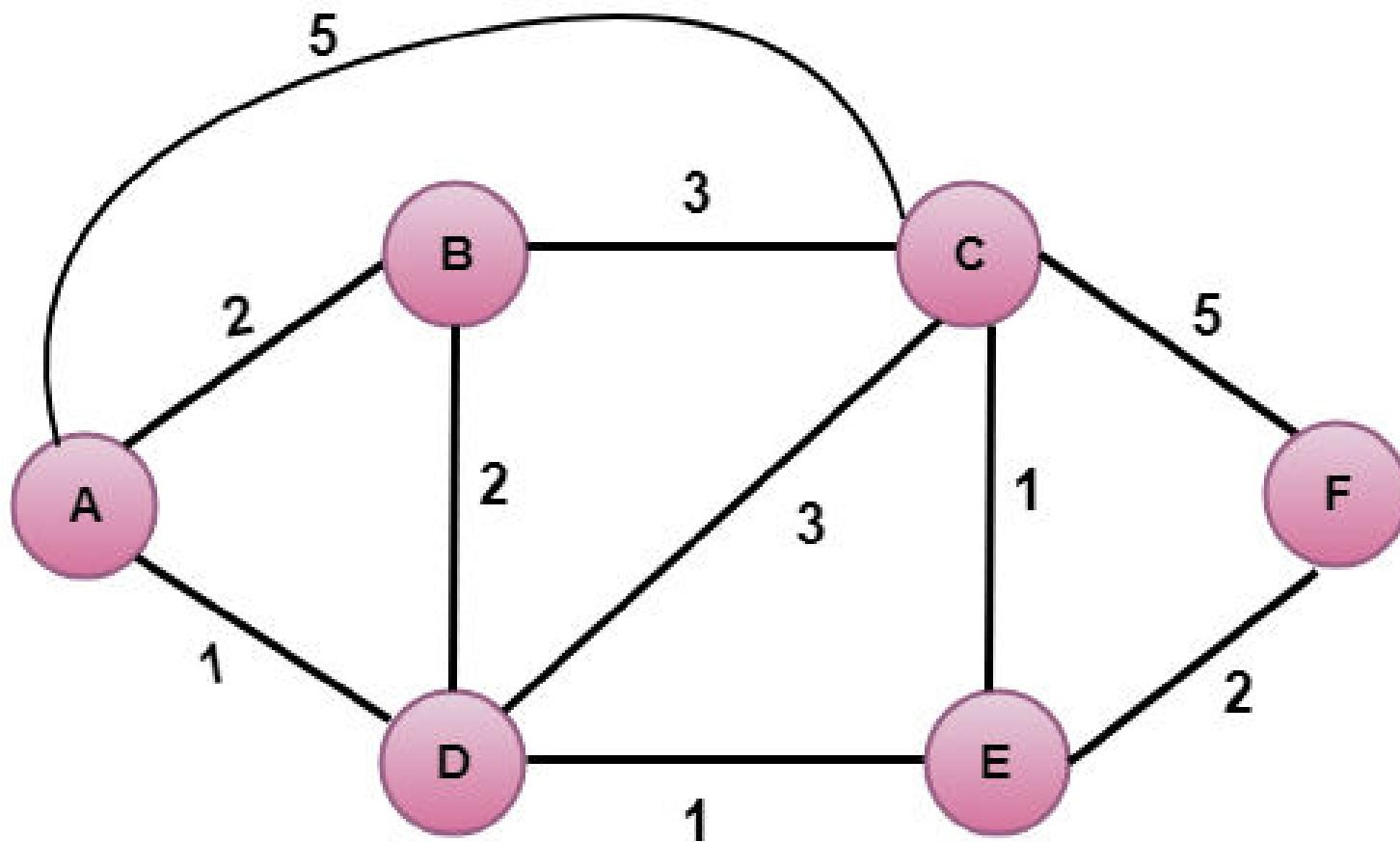
Iterations	Tree	B	C	D	E	F
Initial	$\{A\}$	3	2	5	∞	∞
1	$\{A, C\}$	3	-	4, 4	∞	3
2	$\{A, B, C\}$	-	-	4	7	3
3	$\{A, B, C, F\}$	-	-	4	5	-
4	$\{A, B, C, D, F\}$	-	-	-	5	-
5	$\{A, B, C, D, E, F\}$	-	-	-	-	-



Shortest path Graph.

Note: If you have multiple route to reach a particular node, choose the minimum cost or distance which can be calculate by using Bellman Ford algorithm.

Practice problem



Distance Vector Routing Algorithm

- Distance vector algorithm is iterative, asynchronous and distributed.
 - **Distributed:** It is distributed in that each node receives information from one or more of its directly attached neighbors, performs calculation and then distributes the result back to its neighbors.
 - **Iterative:** It is iterative in that its process continues until no more information is available to be exchanged between neighbors.
 - **Asynchronous:** It does not require that all of its nodes operate in the lock step with each other.
- The Distance vector algorithm is a dynamic algorithm.
- It is mainly used in ARPANET, and RIP.
- Each router maintains a distance table known as **Vector**.

Three Keys to understand the working of Distance Vector Routing Algorithm

- **Knowledge about the whole network:** Each router shares its knowledge through the entire network. The Router sends its collected knowledge about the network to its neighbors.
- **Routing only to neighbors:** The router sends its knowledge about the network to only those routers which have direct links. The information is received by the router and uses the information to update its own routing table.
- **Information sharing at regular intervals:** Within 30 seconds, the router sends the information to the neighboring routers.

Distance Vector Routing Algorithm

- Let $d_x(y)$ be the cost of the least-cost path from node x to node y . The least costs are related by Bellman-Ford equation,

$$d_x(y) = \min_v \{c(x,v) + d_v(y)\}$$

- Where the \min_v is the equation taken for all x neighbors. After traveling from x to v , if we consider the least-cost path from v to y , the path cost will be $c(x,v)+d_v(y)$. The least cost from x to y is the minimum of $c(x,v)+d_v(y)$ taken over all neighbors.

Distance Vector Routing Algorithm

- A router transmits its distance vector to each of its neighbors in a routing packet.
- Each router receives and saves the most recently received distance vector from each of its neighbors.
- A router recalculates its distance vector when:
 - It receives a distance vector from a neighbor containing different information than before.
 - It discovers that a link to a neighbor has gone down.

DV calculation is based on minimizing the cost to each destination

$D_x(y)$ = Estimate of least cost from x to y

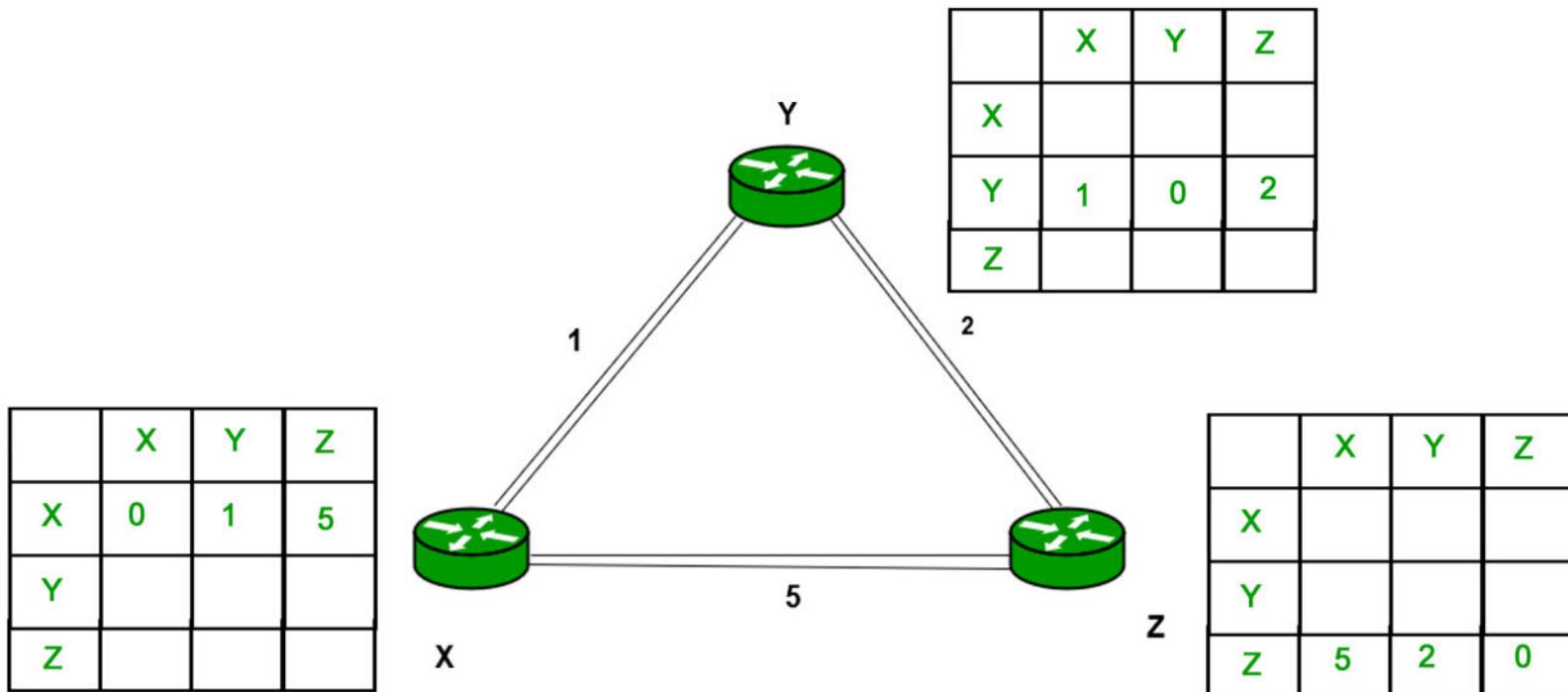
$C(x,v)$ = Node x knows cost to each neighbor v

$D_x = [D_x(y):y \in N]$ = Node x maintains distance vector

Node x also maintains its neighbors' distance vectors

– For each neighbor v , x maintains $D_v = [D_v(y): y \in N]$

Consider 3-routers X, Y and Z as shown in figure. Each router have their routing table. Every routing table will contain distance to the destination nodes

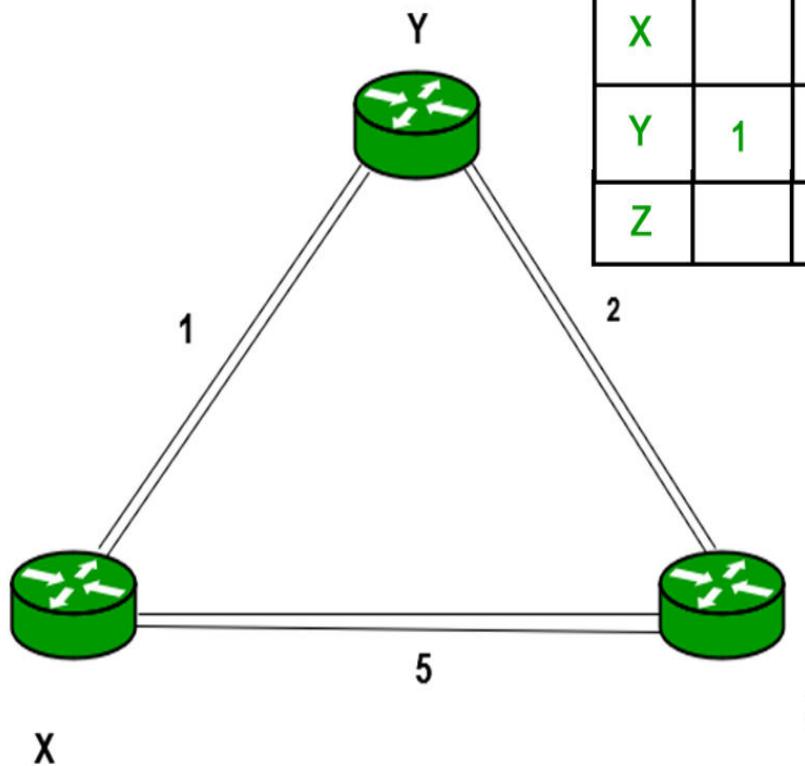


- Consider router X , X will share its routing table to neighbors and neighbors will share its routing table to it to X and distance from node X to destination will be calculated using bellmen- ford equation.

$$D_x(y) = \min \{ C(x,v) + D_v(y) \} \text{ for each node } y \in N$$

- As we can see that distance will be less going from X to Z when Y is intermediate node(hop) so it will be updated in routing table X.

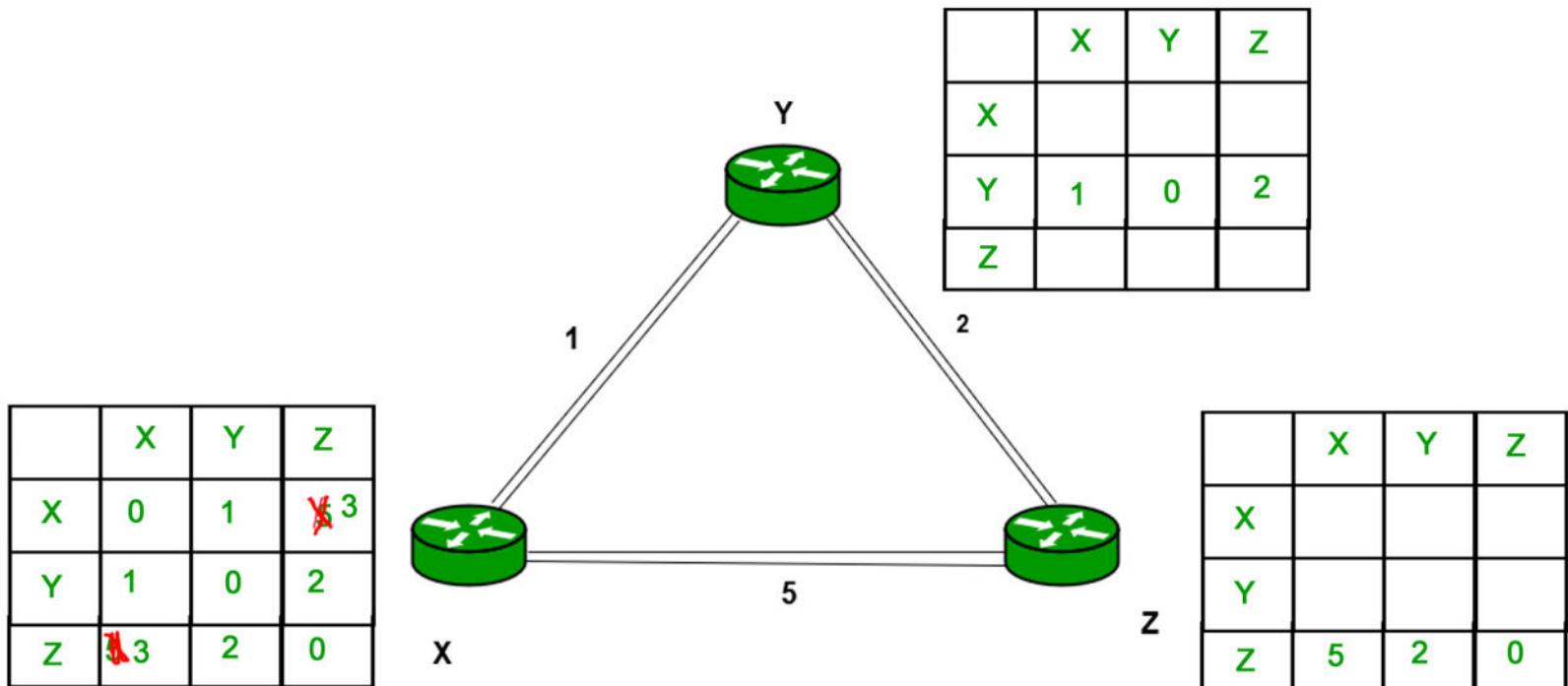
	X	Y	Z
X	0	1	3
Y	1	0	2
Z			



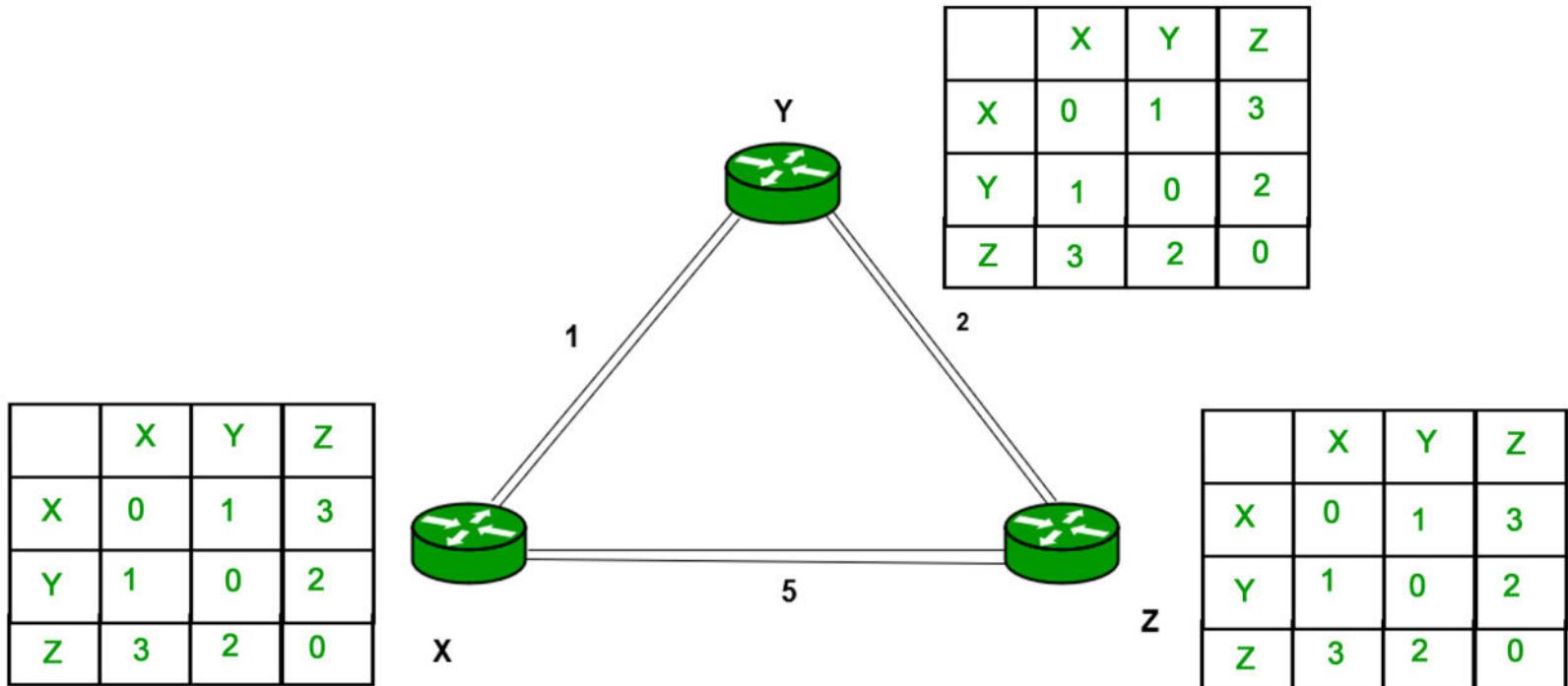
	X	Y	Z
X			
Y	1	0	2
Z			

	X	Y	Z
X			
Y			
Z	5	2	0

Similarly for Z also



Finally the routing table for all



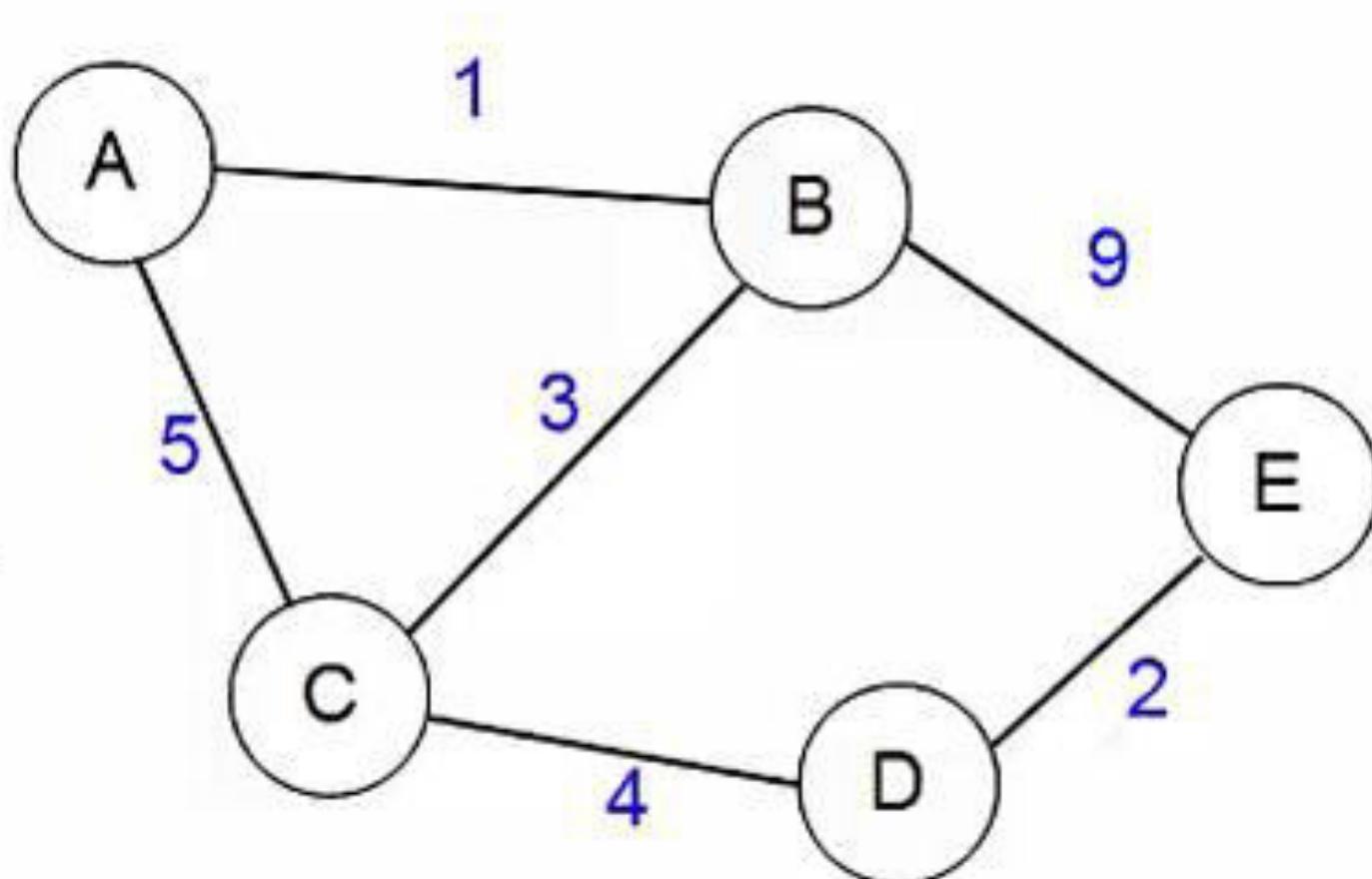
Advantages of Distance Vector routing

- It is simpler to configure and maintain than link state routing

Disadvantages of Distance Vector routing

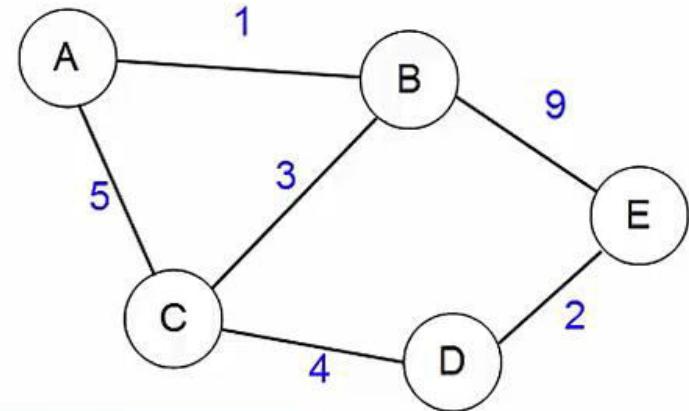
- It is slower to converge than link state.
- It is at risk from the count-to-infinity problem.
- It creates more traffic than link state since a hop count change must be propagated to all routers and processed on each router. Hop count updates take place on a periodic basis, even if there are no changes in the network topology, so bandwidth-wasting broadcasts still occur.
- For larger networks, distance vector routing results in larger routing tables than link state since each router must know about all other routers. This can also lead to congestion on WAN links

Practice Problem



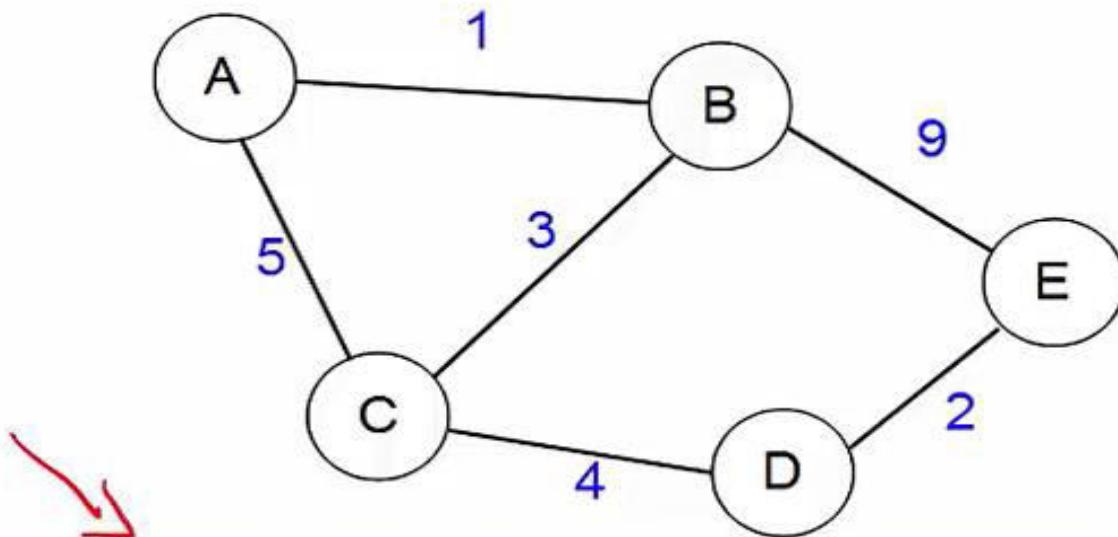
State Maintained

- Each node maintains a routing table (distance vector)
 - Destination
 - Estimated cost to destination
 - Next hop via which to reach destination
- Initial state: Cost to neighbors



Dest	Cost	Next Hop
A	1	A
C	3	C
E	9	E

Initial Routing table at B



<u>Dest</u>	<u>Cost</u>	<u>Next Hop</u>
A	1	A
C	3	C
E	9	E

<u>Dest</u>	<u>Cost</u>	<u>Next Hop</u>
A	1	A
C	3	C
D	7	C
E	9	E

Initial Routing table at B

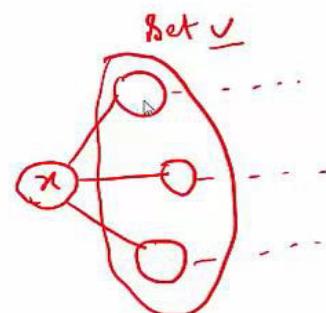
Final Routing table at B

Message Content

- Each node exchanges with all its neighbors “Routing Table” info
 - Destination and ‘Estimated’ cost to destination
 - Next hop information is not shared

Action at a router

- Bellman-Ford equation
 - $d_x(y) = \min_v \{c(x,v) + d_v(y)\}$
 - $\underline{d_x(y)}$ – least cost path from node x to y
 - \min_v – apply above eq. over all of x’s neighbors



Action at a router

- On receiving a message from a neighbor v,
 - Update cost (estimate) to destinations based on above Bellman-ford equation; change next hop accordingly
 - For each y (destination in routing table of the received message)
 - $D_x(y) = \min\{\text{current estimate}, c(x,v) + D_v(y)\}$
 - Estimated costs finally converge to optimal cost after series of message exchanges

D	C	H
A	5	A
B	3	B
D	4	D

To	A
A	0
B	1
C	5

D	C	H
A	5	A
B	3	B
D	4	D

Routing Table of C
(1)

D	C	H
A	5	A
B	3	B
D	4	D

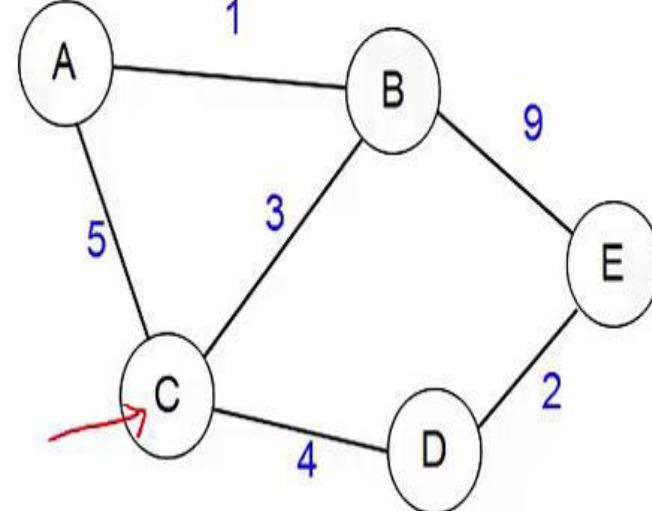
To	B
A	1
B	0
C	3
E	9

D	C	H
A	4	B
B	3	B
D	4	D
E	12	B

Routing Table of C
(2)

Message from B
C to B: C = 3

Routing Table of C



D	C	H
A	4	B
B	3	B
D	4	D
E	6	D

To	D
C	4
D	0
E	2

Routing Table of C
(3)

Message from D
C to D: C = 4

Routing Table of C

Reference Node C

Example

initial routing @ A

D	C	H
A	5	A
B	3	B
D	4	D

To	A
A	0
B	1
C	5

D	C	H
A	5	A
B	3	B
D	4	D

Routing Table of C
(1)

Message from A
C to A: C = 5

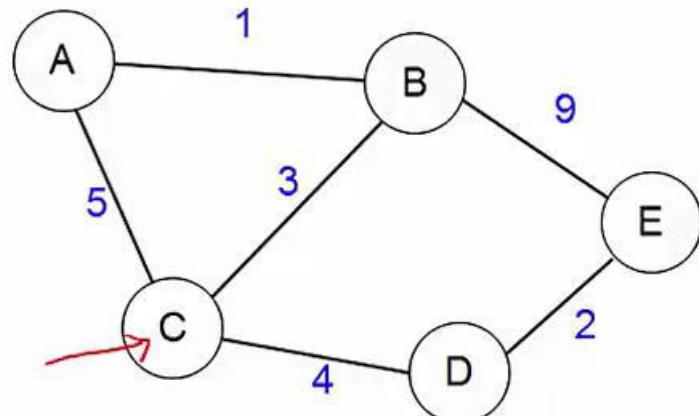
Routing Table of C

D	C	H
A	5	A
B	3	B
D	4	D

To	B
A	1
B	0
C	3
E	9

Routing Table of C
(2)

Message from B
C to B: C = 3



Module:6

Transport Layer

Overview

- TCP and UDP
- Congestion Control
- Effects of Congestion
- Traffic Management
- TCP Congestion Control
- Congestion Avoidance Mechanisms
- Queuing Mechanisms - QoS Parameter

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:

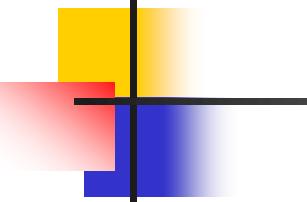
Client/Server Paradigm

Multiplexing and Demultiplexing

Connectionless Versus Connection-Oriented Service

Reliable Versus Unreliable

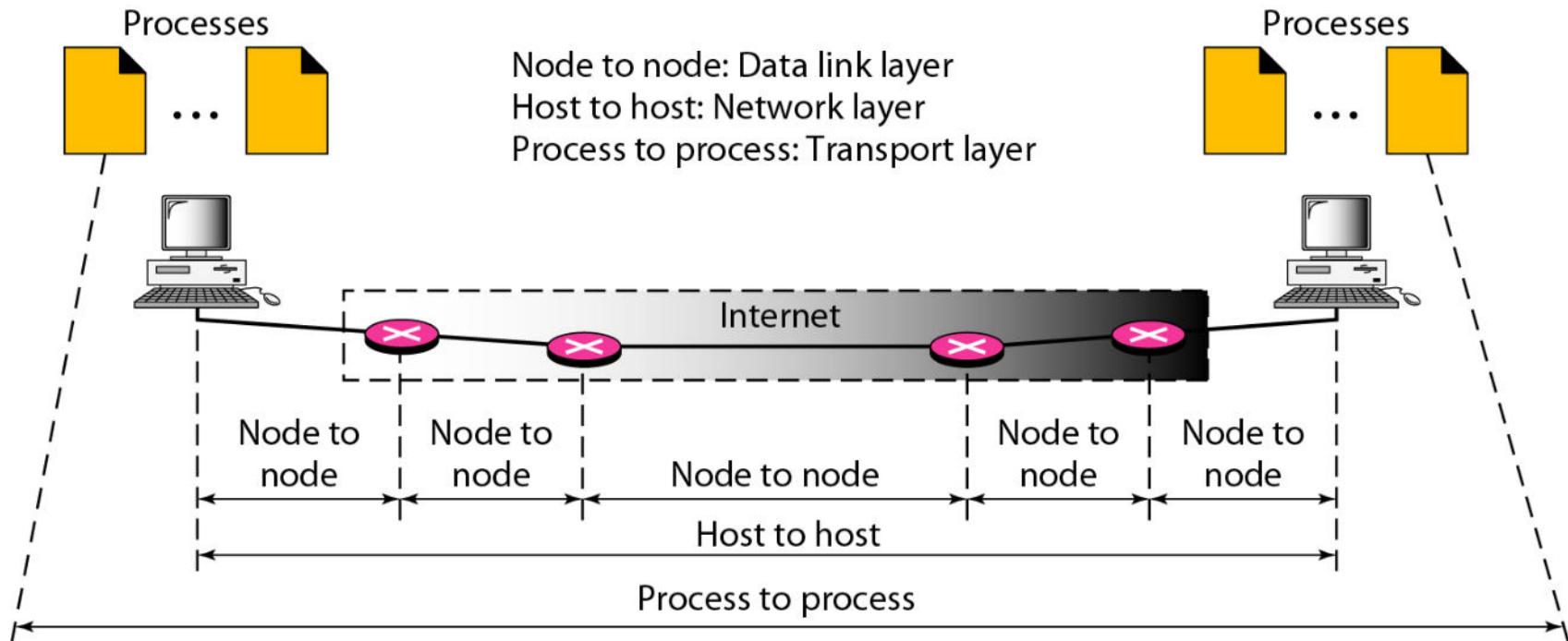
Three Protocols



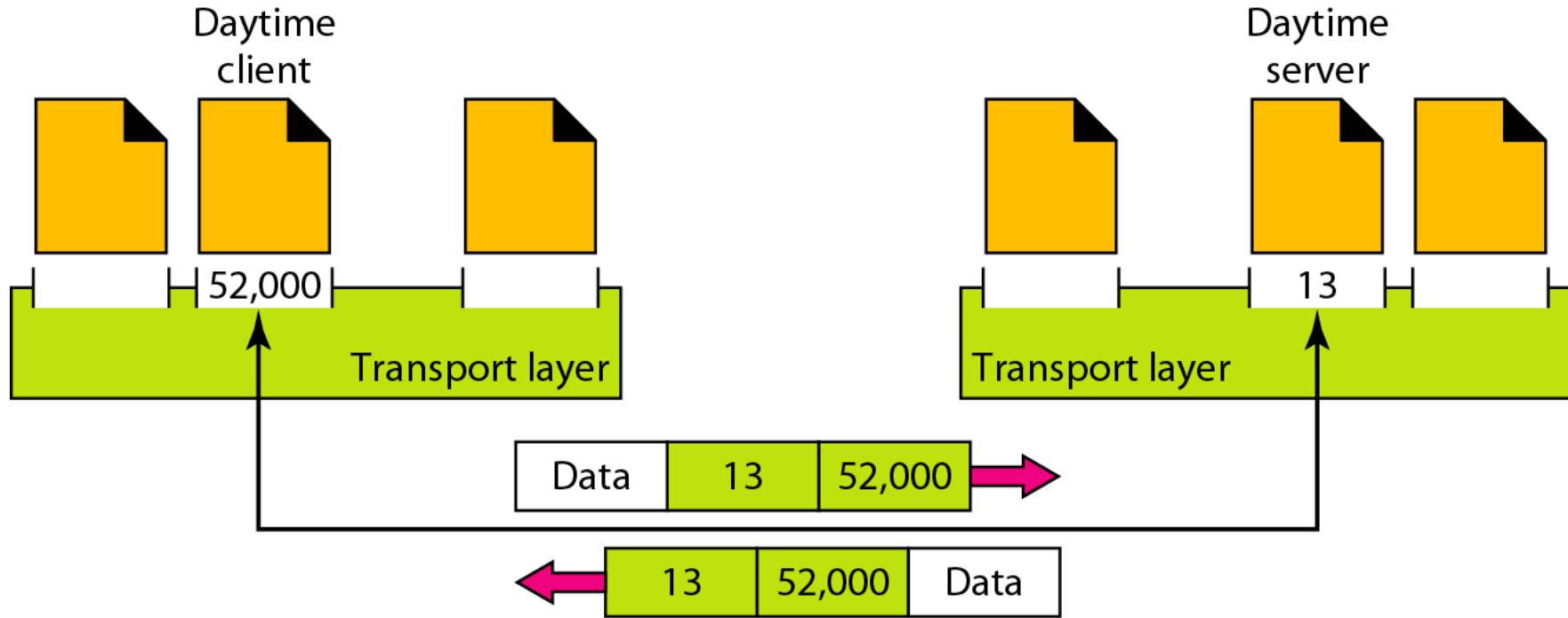
Note

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

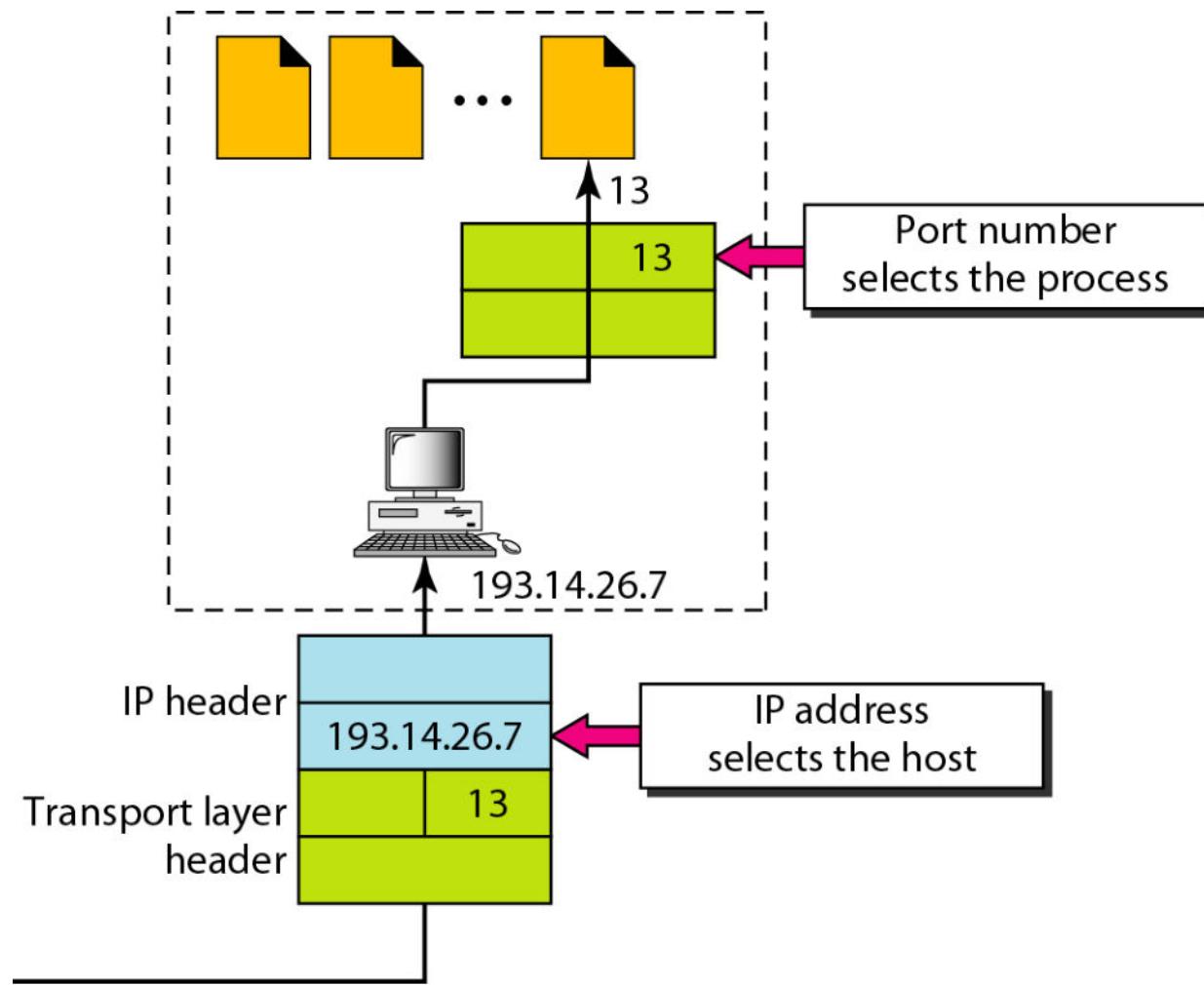
Types of data deliveries



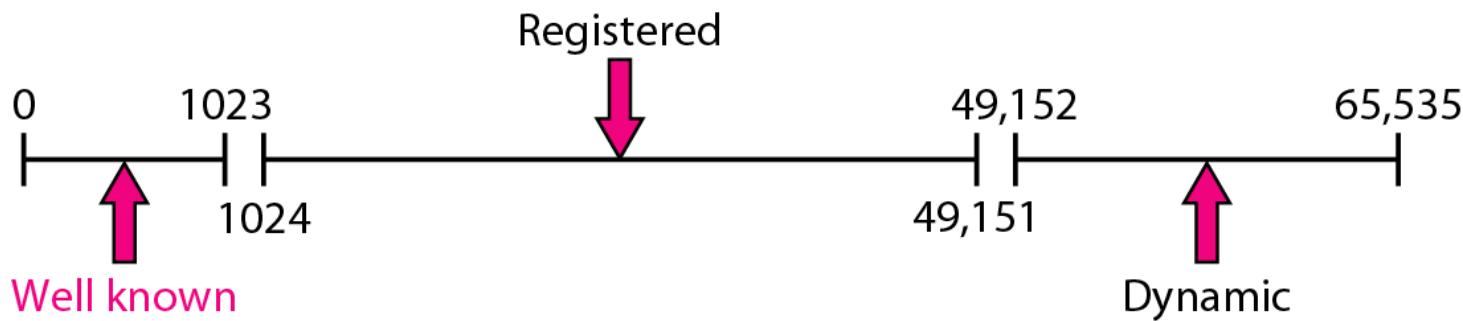
Port numbers



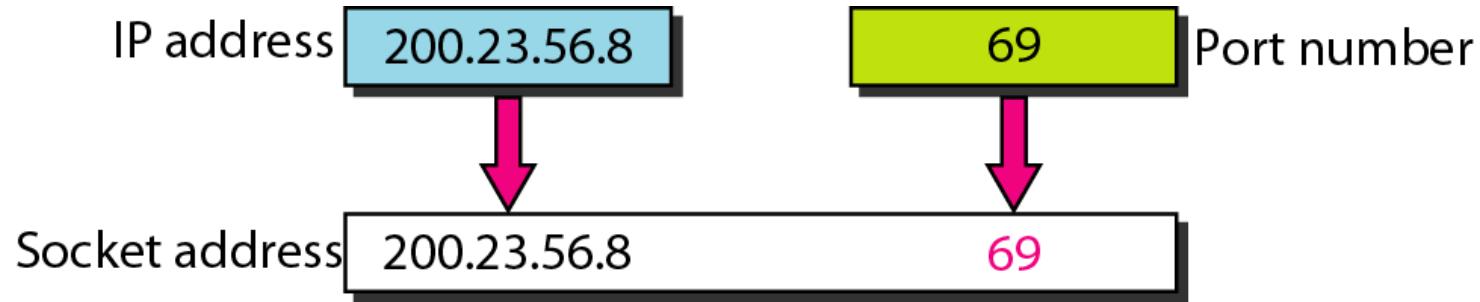
IP addresses versus port numbers



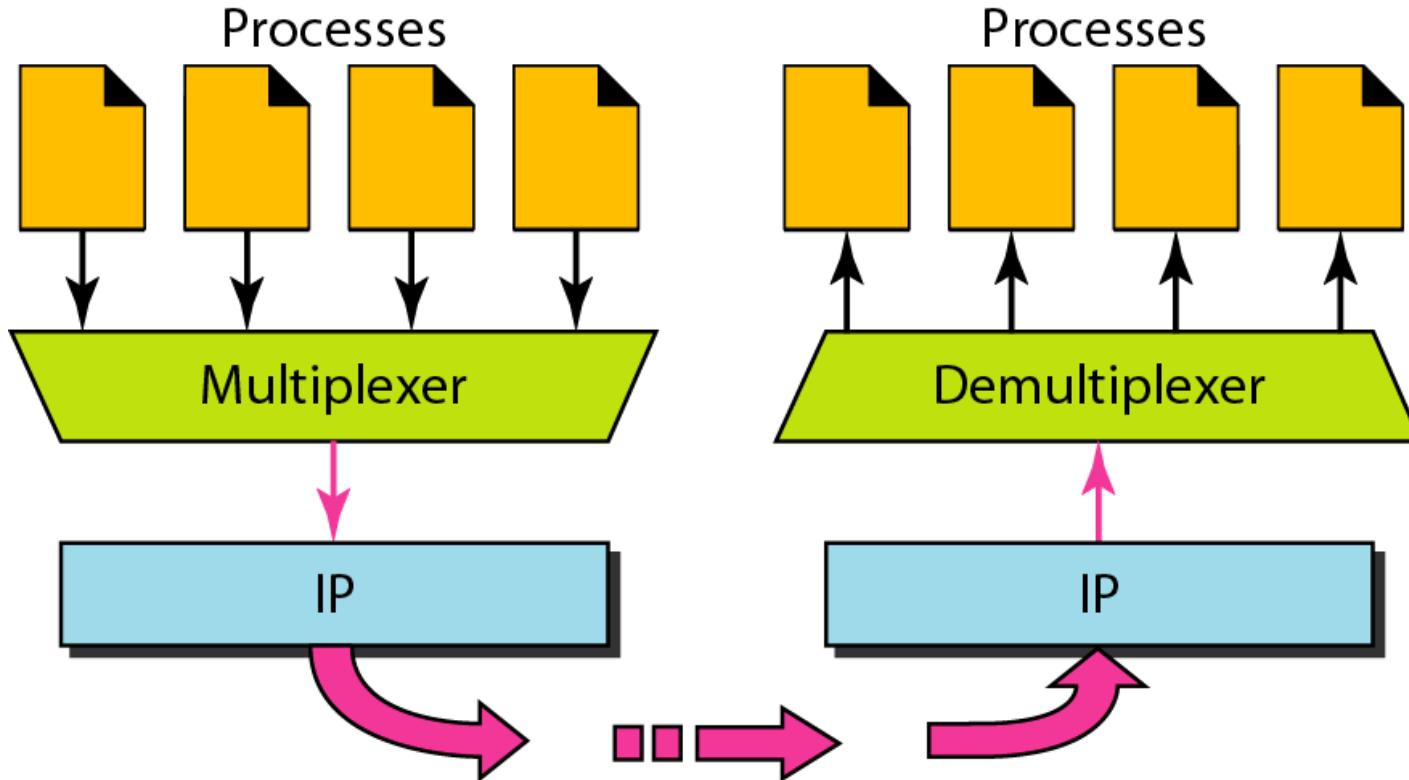
IANA Port number ranges (Internet Assigned Number Authority)



Socket address

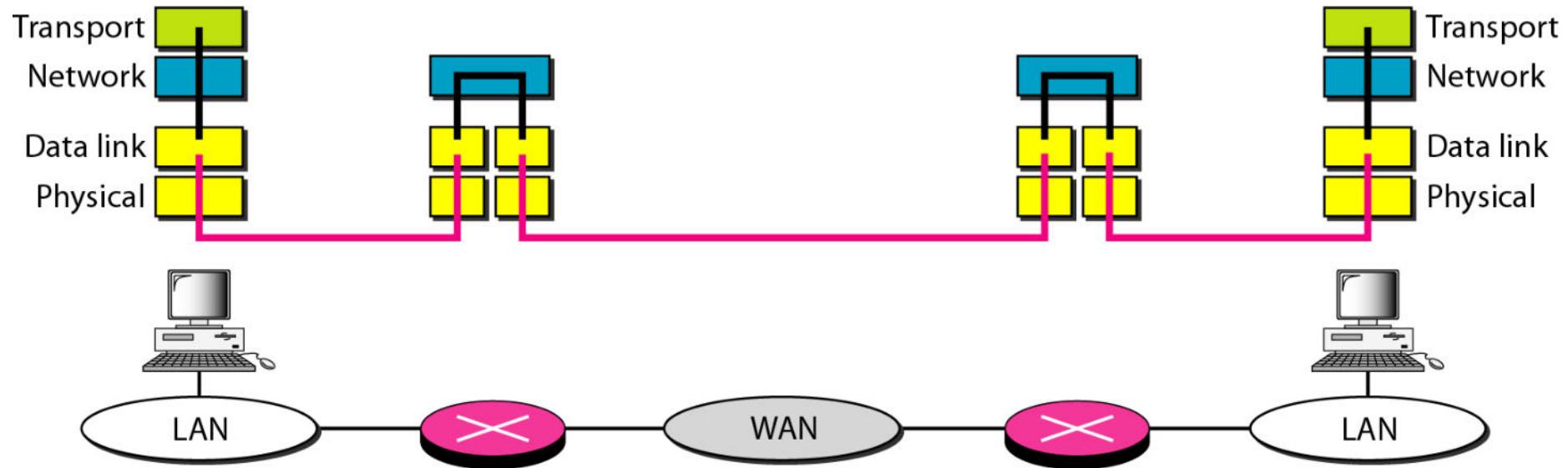


Multiplexing and demultiplexing

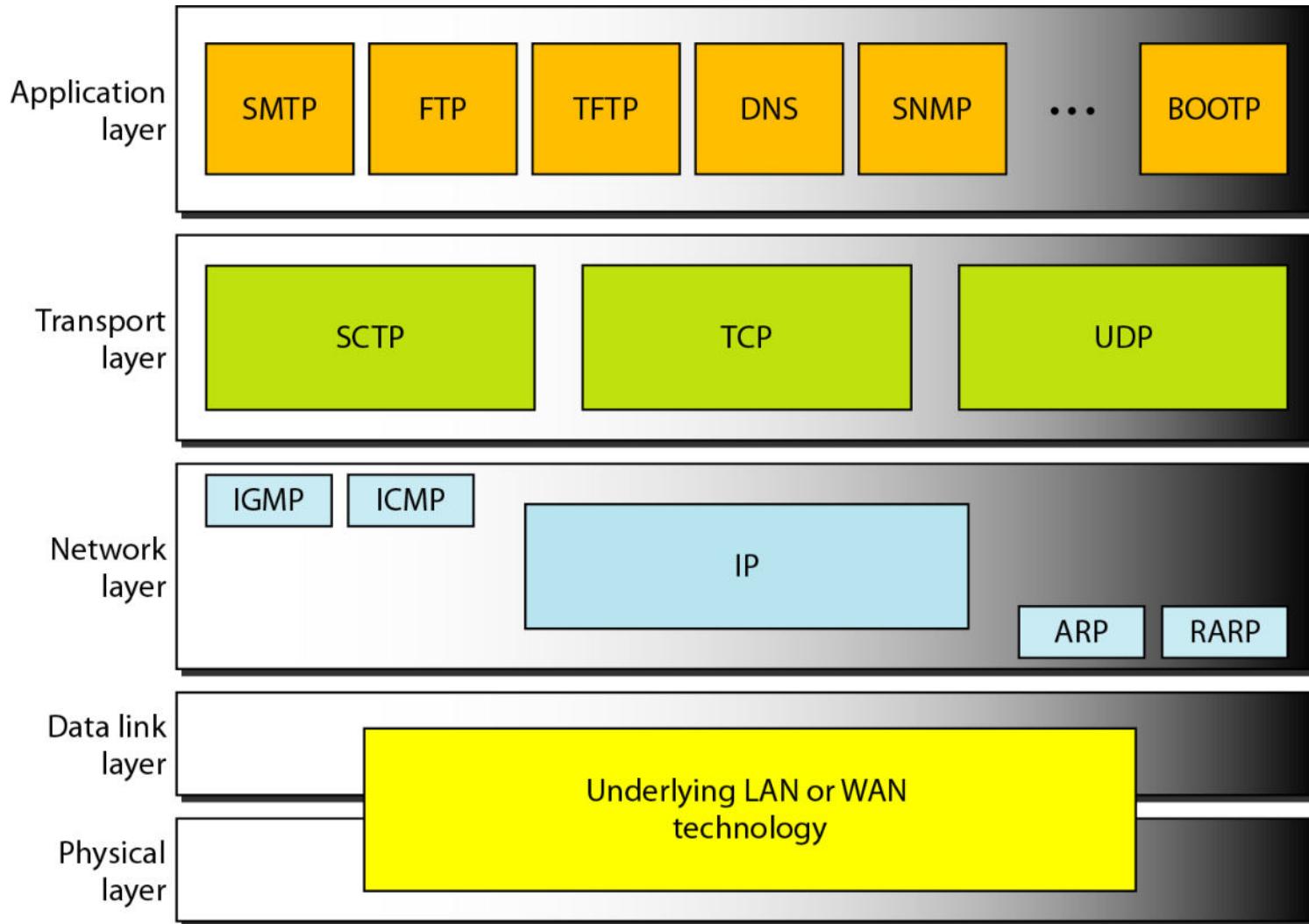


Error control

- Error is checked in these paths by the data link layer
- Error is not checked in these paths by the data link layer



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



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

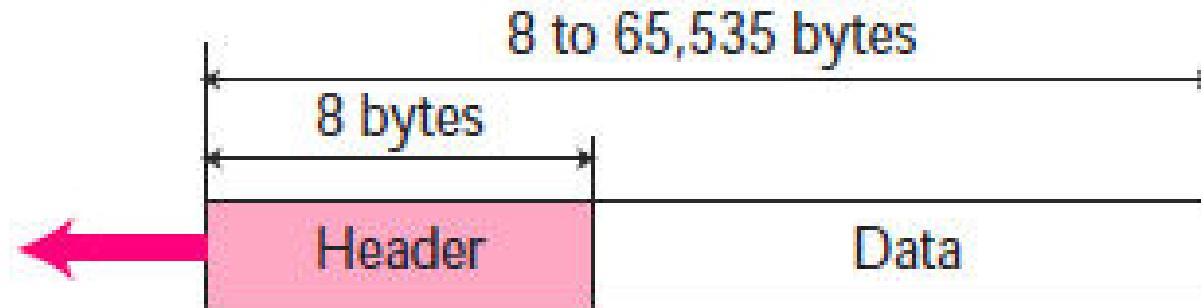
Checksum

UDP Operation

Use of UDP

- UDP is a very simple protocol using a minimum of overhead. If a process wants to send a small message and does not care much about reliability, it can use UDP. Sending a small message by using UDP takes much less interaction between the sender and receiver than using TCP or SCTP.
- User Datagram: UDP packets, called user datagrams, have a fixed-size header of 8 bytes.

User datagram



a. UDP user datagram

0	16	31
Source port number	Destination port number	
Total length		Checksum

b. Header format

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.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.*

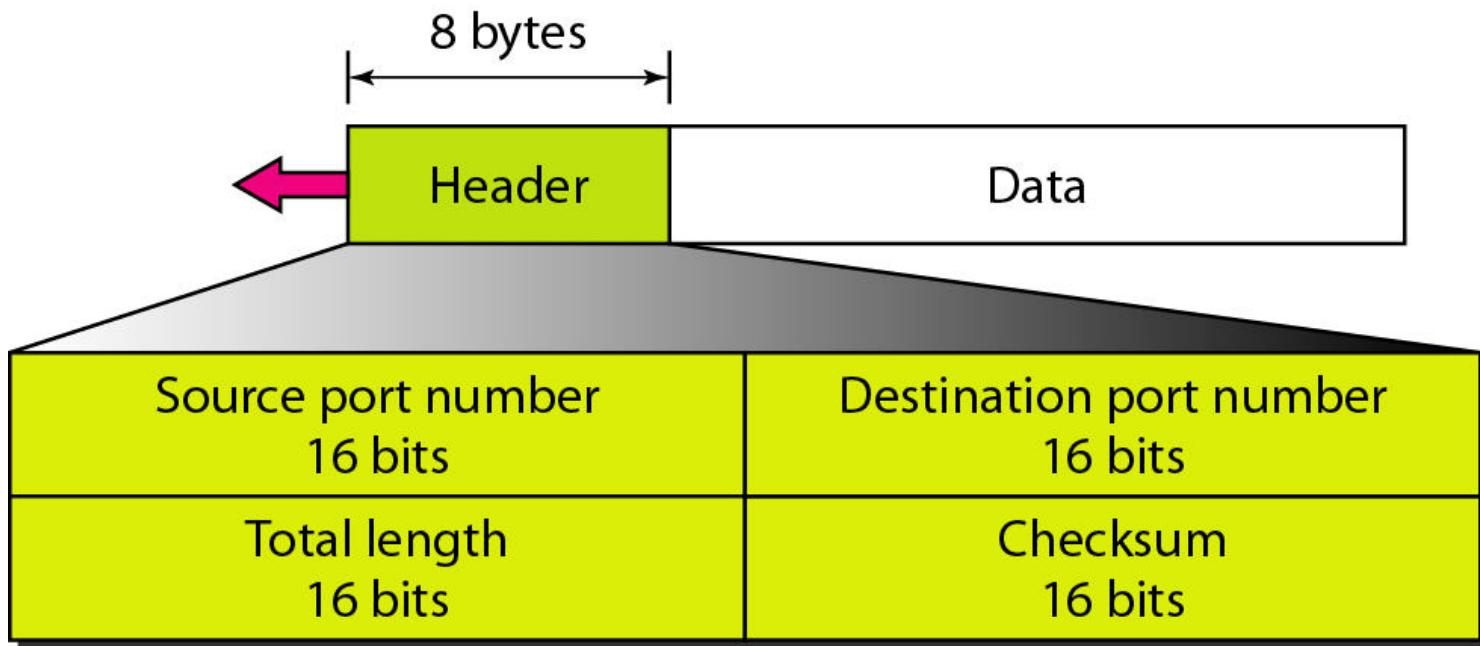
```
$ grep ftp /etc/services
ftp          21/tcp
ftp          21/udp
```

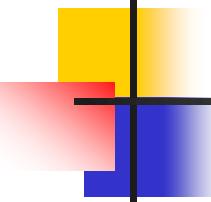
Example.1 (continued)

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
```

User datagram format





UDP length

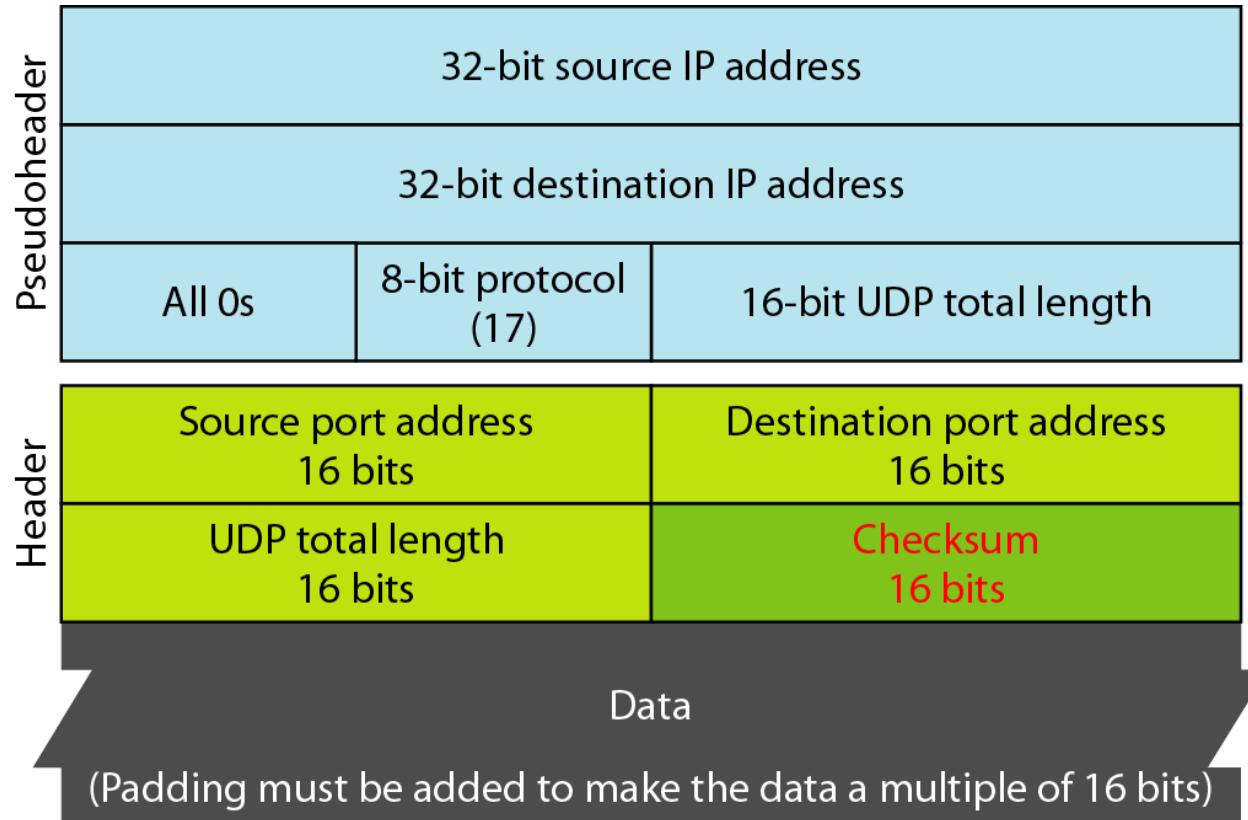
$$= \text{IP length} - \text{IP header's length}$$

A field that specifies the **length** in bytes of the **UDP** header and **UDP** data. The minimum **length** is 8 bytes because that is the **length** of the header. The field **size** sets a theoretical limit of 65,535 bytes (8 byte header + 65,527 bytes of data) for a **UDP** datagram.

Pseudo Header

to do a proper checksum, a "**pseudo header**" is included. It's "**pseudo**", because it is not actually part of the UDP datagram. It contains the most important parts of the **IP header**, that is, source and destination address, protocol number and data length

Pseudoheader for checksum calculation



Example .2

Figure 23.11 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.

153.18.8.105

171.2.14.10

All Os

17

15

1087

13

15

All Os

T

E

S

T

I

N

G

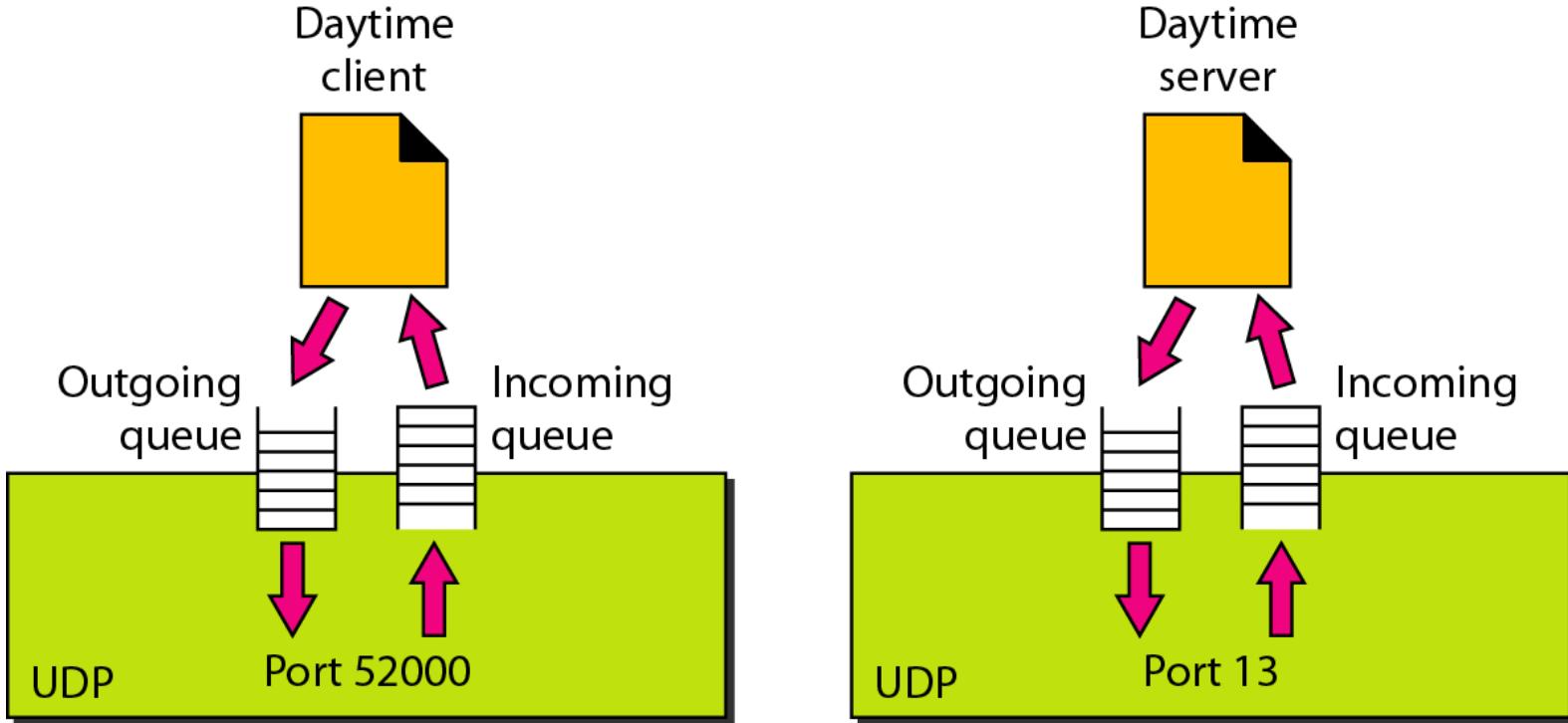
All Os

Checksum calculation of a simple UDP user datagram

153.18.8.105		
171.2.14.10		
All 0s	17	15
1087		13
15		All 0s
T	E	S
I	N	G

10011001 00010010	→ 153.18
00001000 01101001	→ 8.105
10101011 00000010	→ 171.2
00001110 00001010	→ 14.10
00000000 00010001	→ 0 and 17
00000000 00001111	→ 15
00000100 00111111	→ 1087
00000000 00001101	→ 13
00000000 00001111	→ 15
00000000 00000000	→ 0 (checksum)
01010100 01000101	→ T and E
01010011 01010100	→ S and T
01001001 01001110	→ I and N
01000111 00000000	→ G and 0 (padding)
10010110 11101011	→ Sum
01101001 00010100	→ Checksum

Queues in UDP



TCP

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

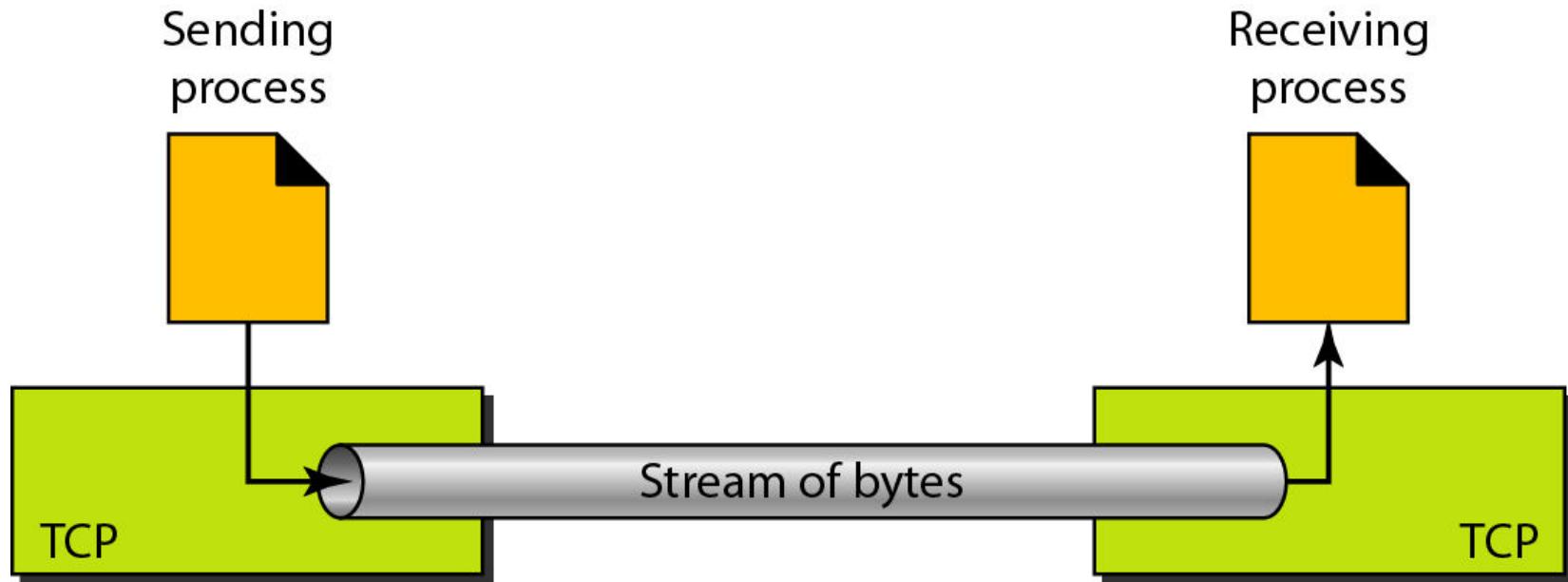
Flow Control

Error Control

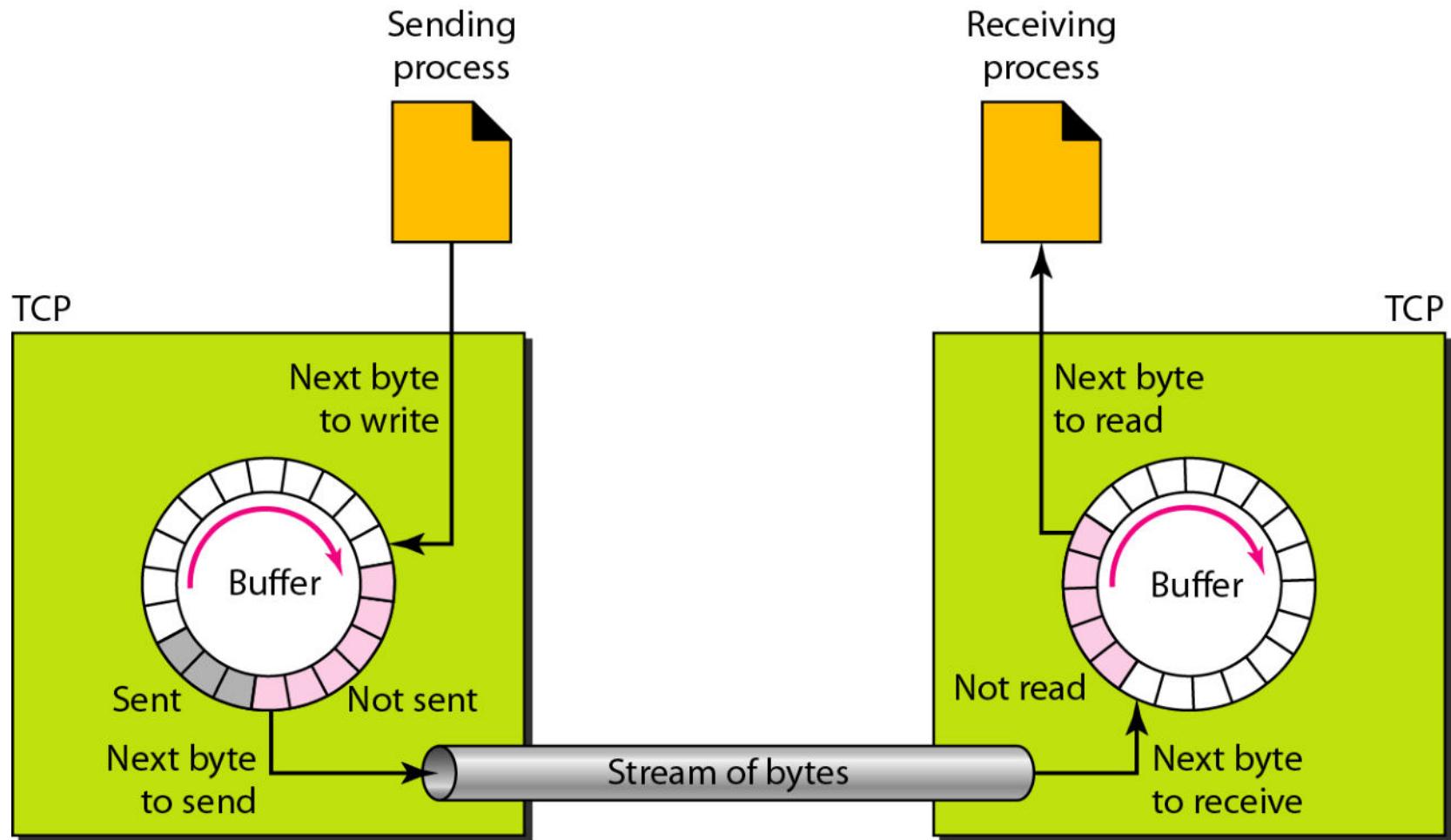
Well-known ports used by TCP

<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
20	FTP, Data	File Transfer Protocol (data connection)
21	FTP, Control	File Transfer Protocol (control connection)
23	TELNET	Terminal Network
25	SMTP	Simple Mail Transfer Protocol
53	DNS	Domain Name Server
67	BOOTP	Bootstrap Protocol
79	Finger	Finger
80	HTTP	Hypertext Transfer Protocol
111	RPC	Remote Procedure Call

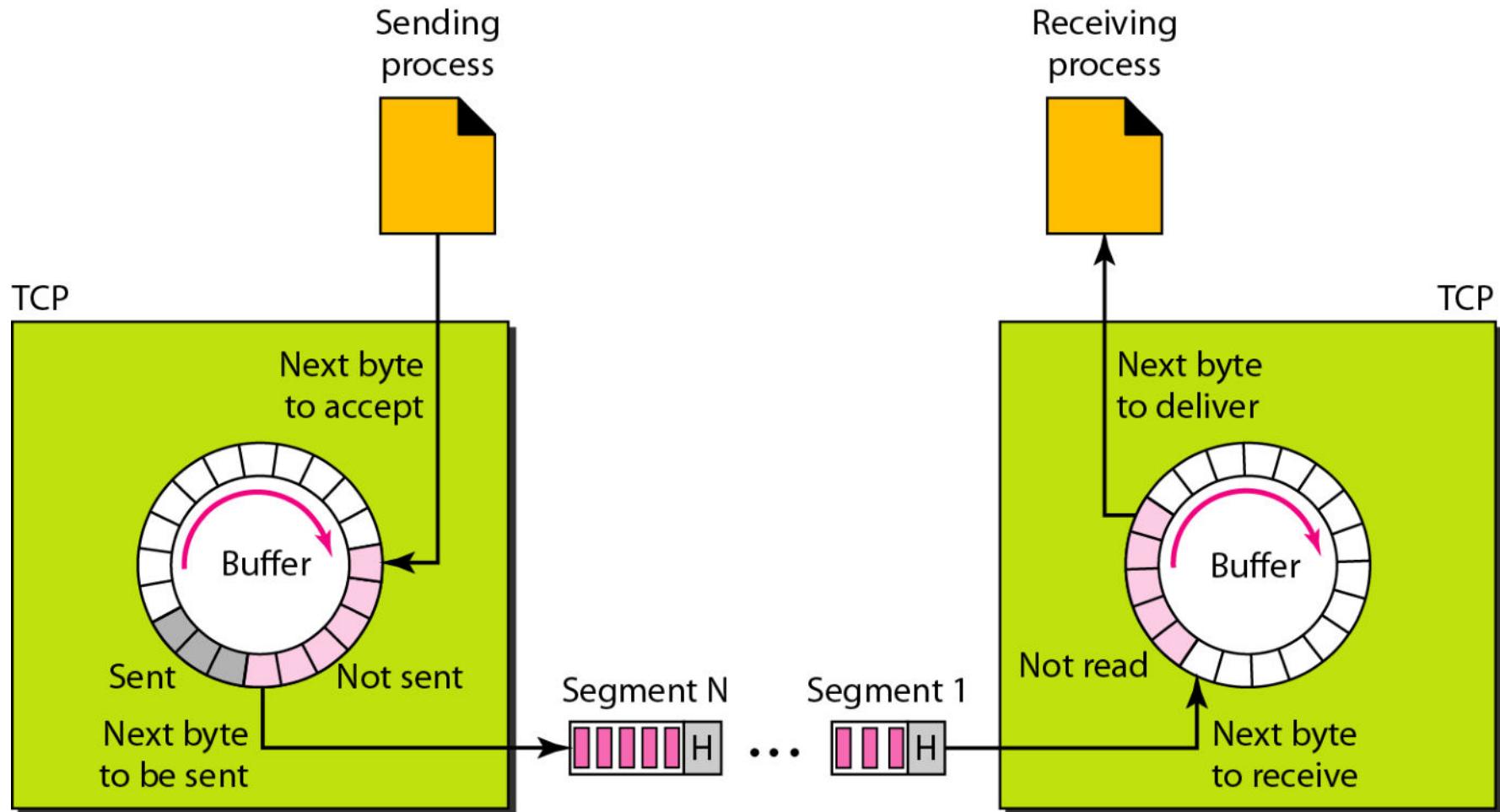
Stream delivery

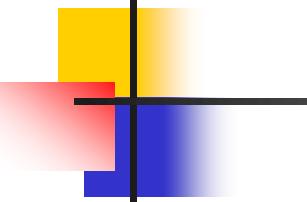


Sending and receiving buffers



TCP segments





Note

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

TCP segment format

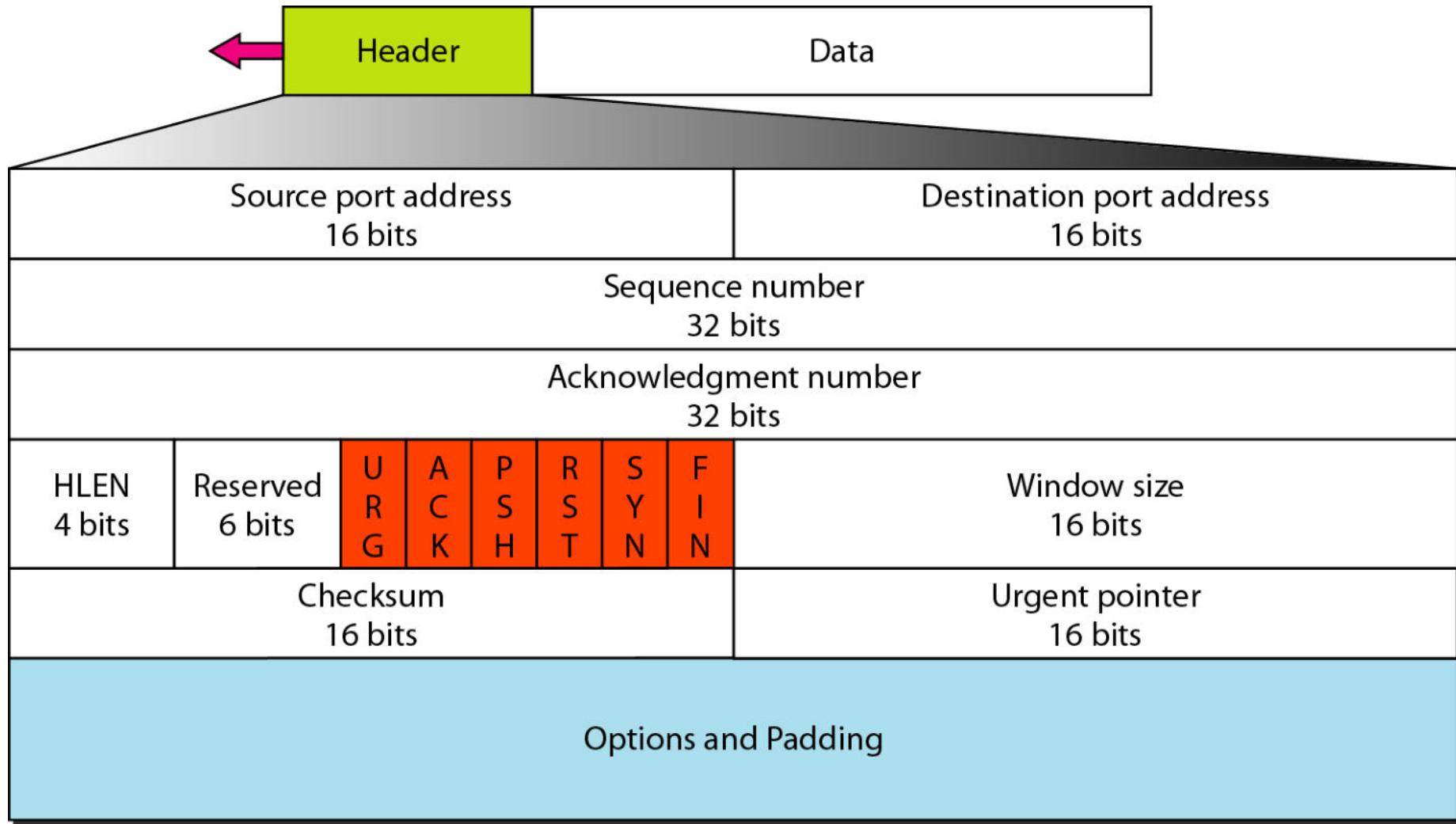


Figure 23.17 *Control field*

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

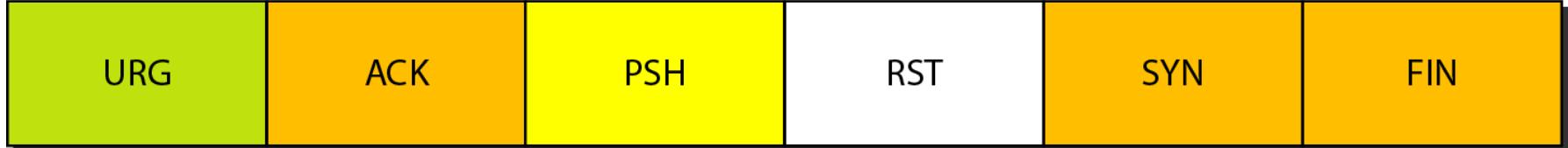
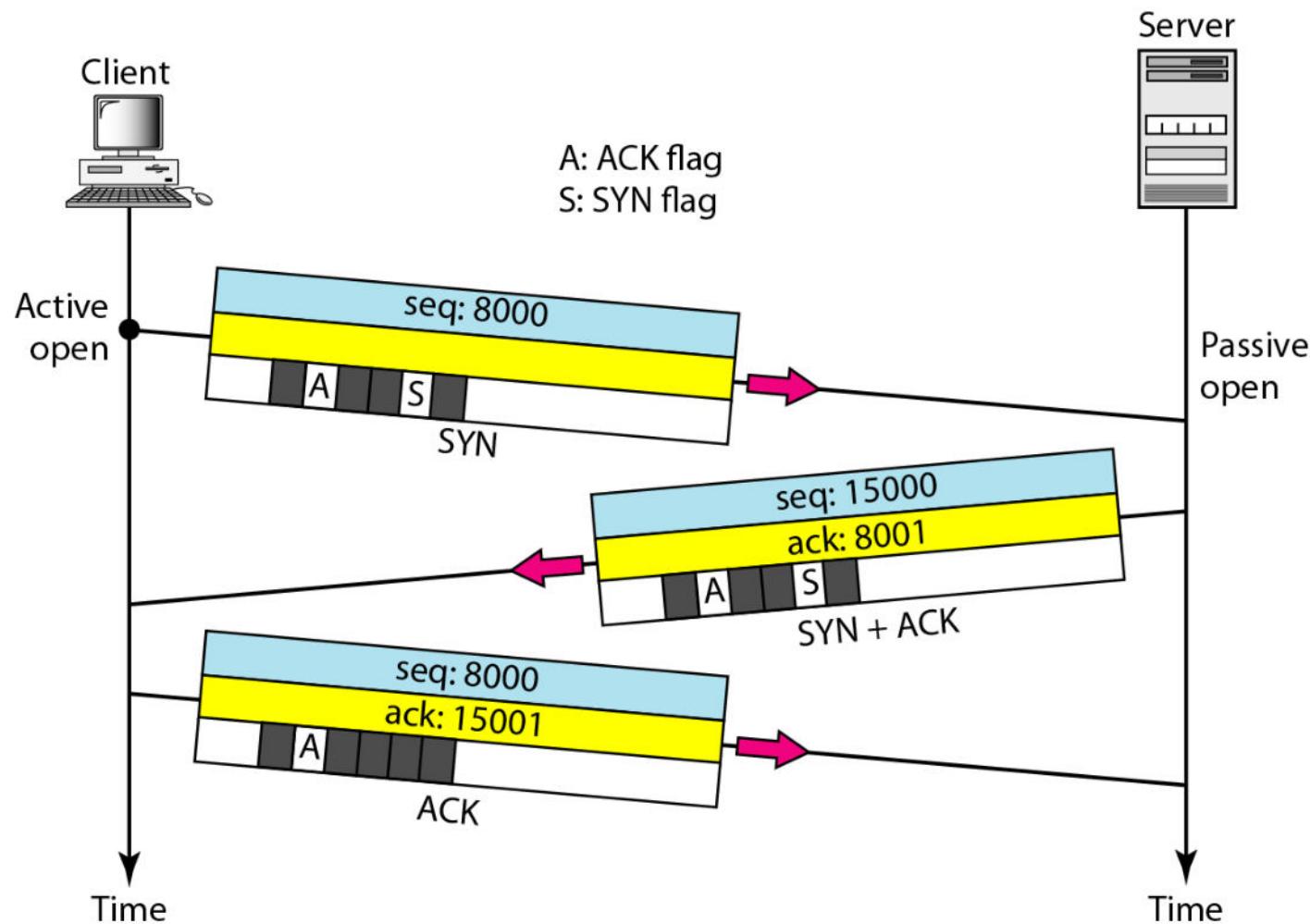


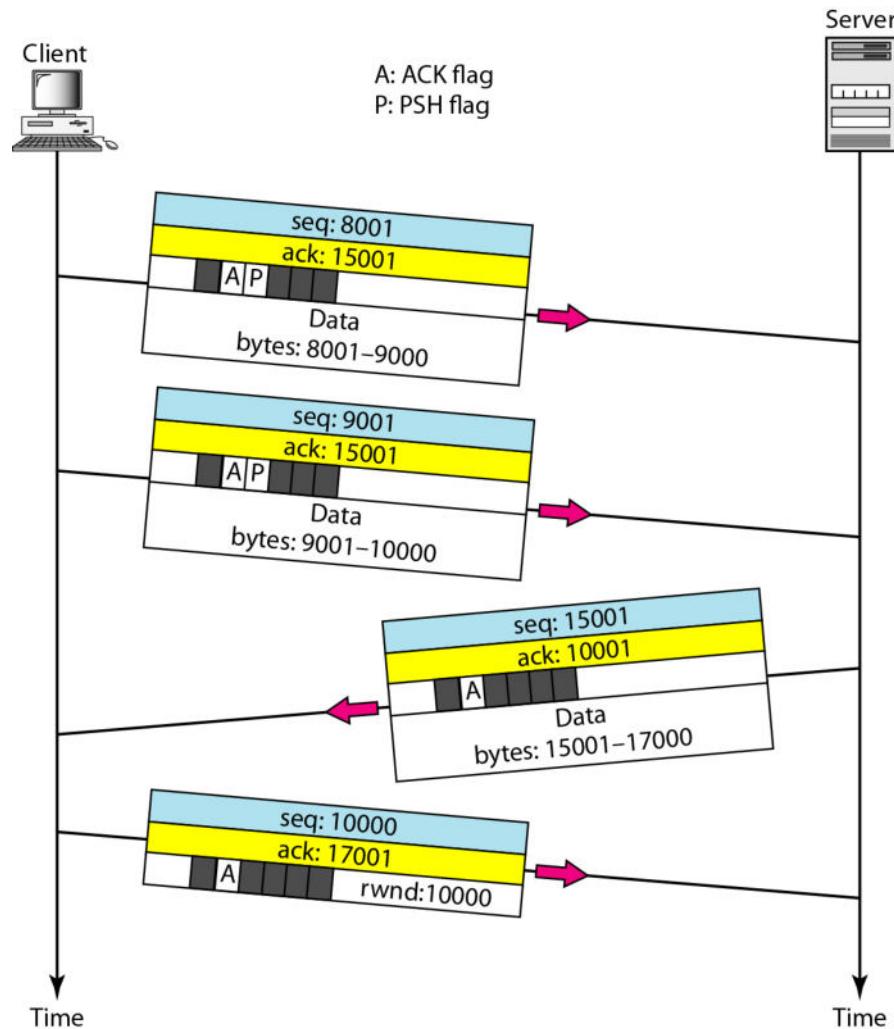
Table Description of flags in the control field

<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.

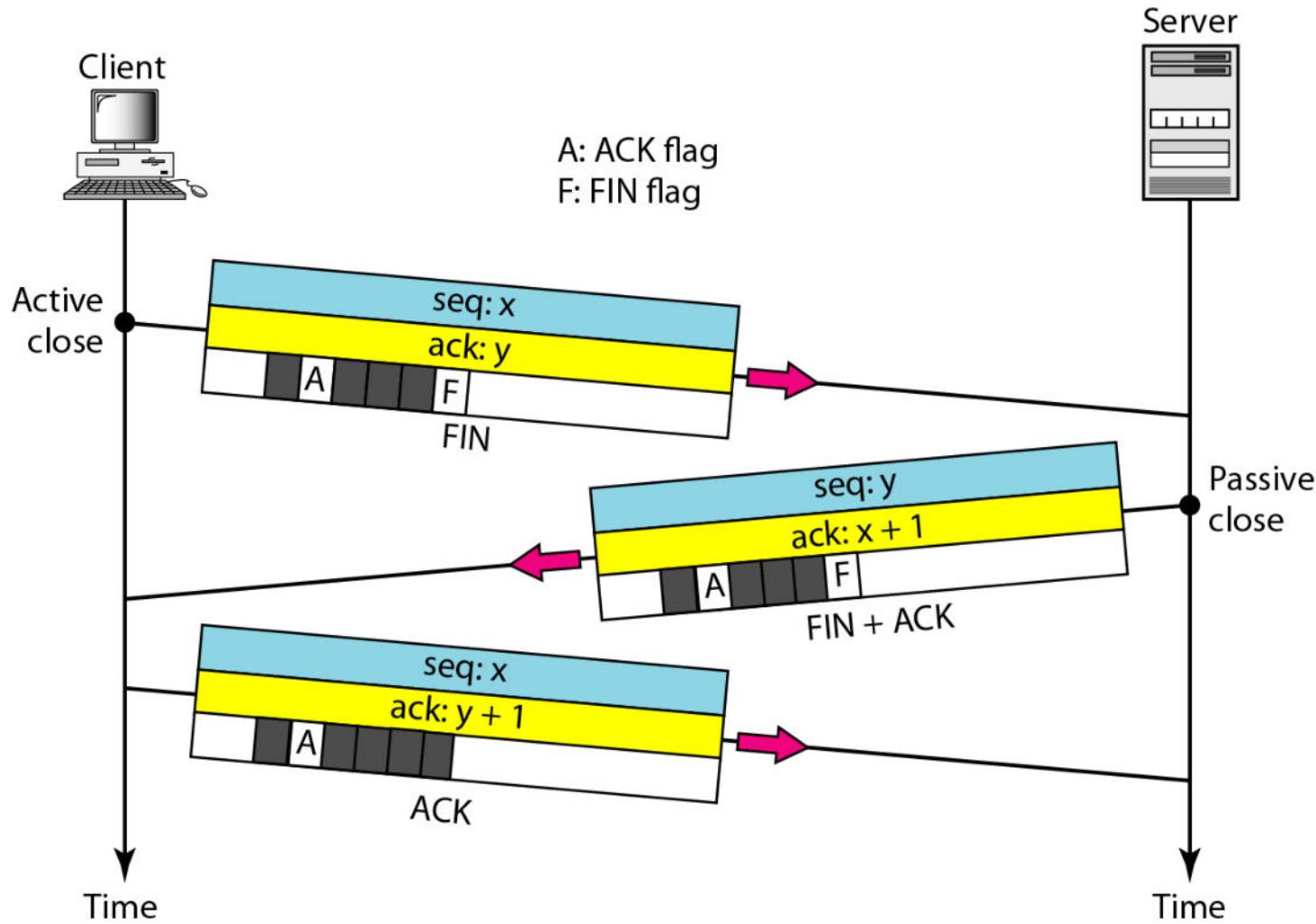
Connection establishment using three-way handshaking

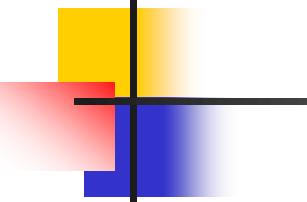


Data transfer



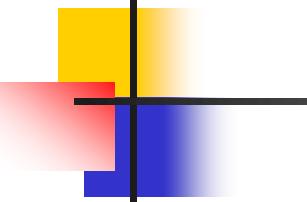
Connection termination using three-way handshaking





Note

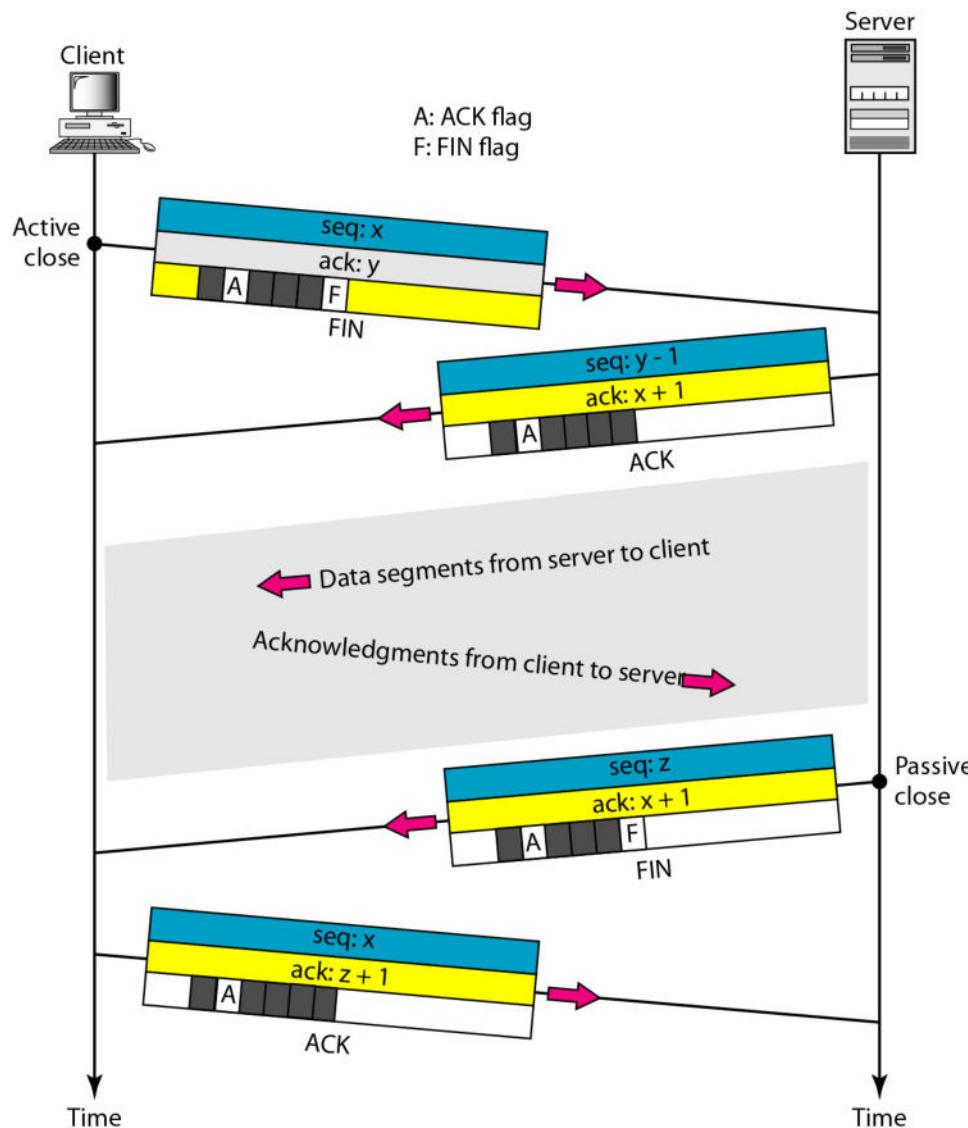
The FIN segment consumes one sequence number if it does not carry data.



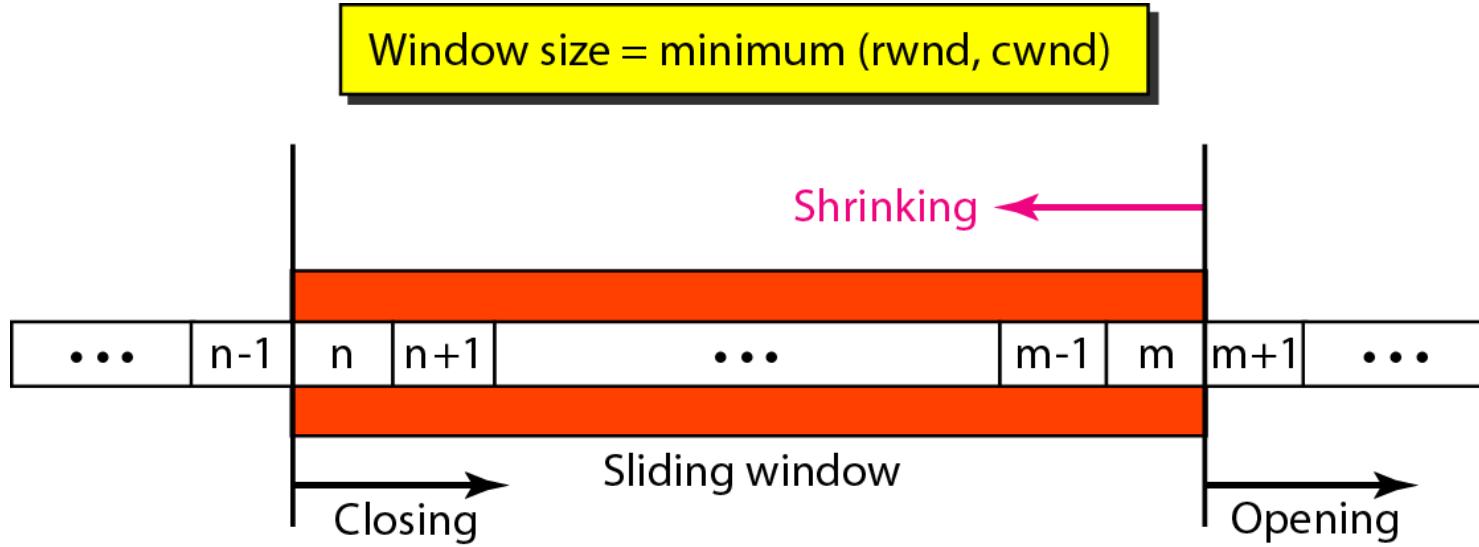
Note

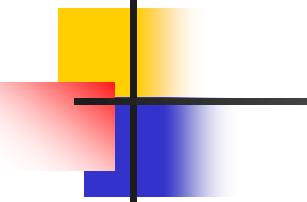
The FIN + ACK segment consumes one sequence number if it does not carry data.

Half-close



Sliding window

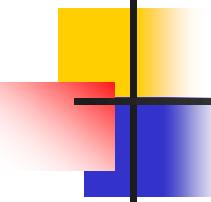




Note

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.

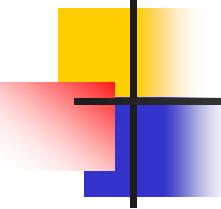


Example .4

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.

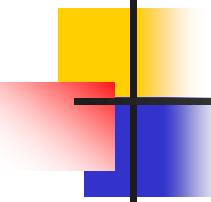


Example 5

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

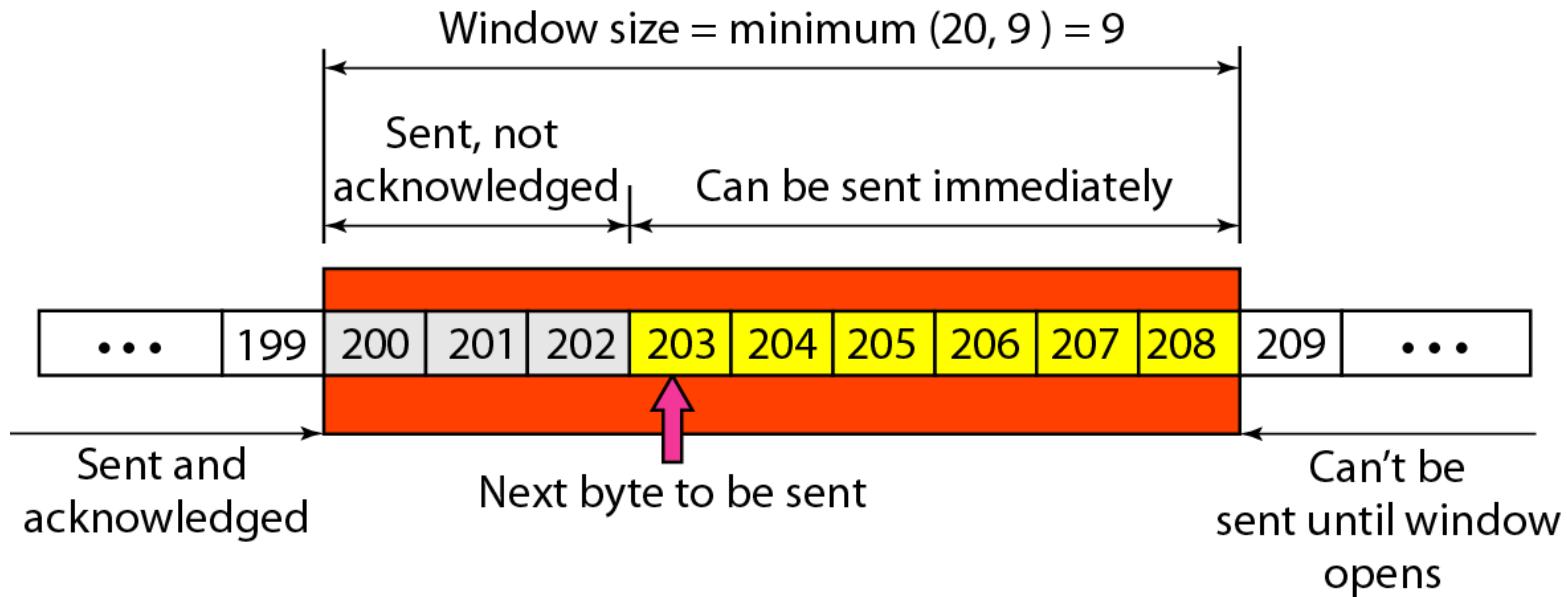
The size of the window is the smaller of rwnd and cwnd, which is 3000 bytes.



Example 6

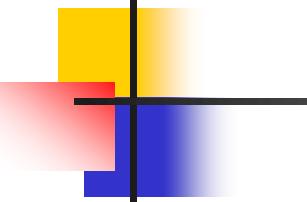
Figure 23.23 shows an unrealistic example of a sliding window. The sender has sent bytes up to 202. We assume that cwnd is 20 (in reality this value is thousands of bytes). The receiver has sent an acknowledgment number of 200 with an rwnd of 9 bytes (in reality this value is thousands of bytes). The 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 23.23 Example 23.6



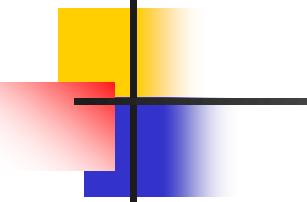
Some points about TCP sliding windows:

- ❑ 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.**



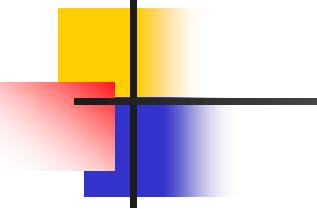
Note

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



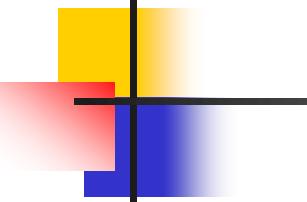
Note

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



Note

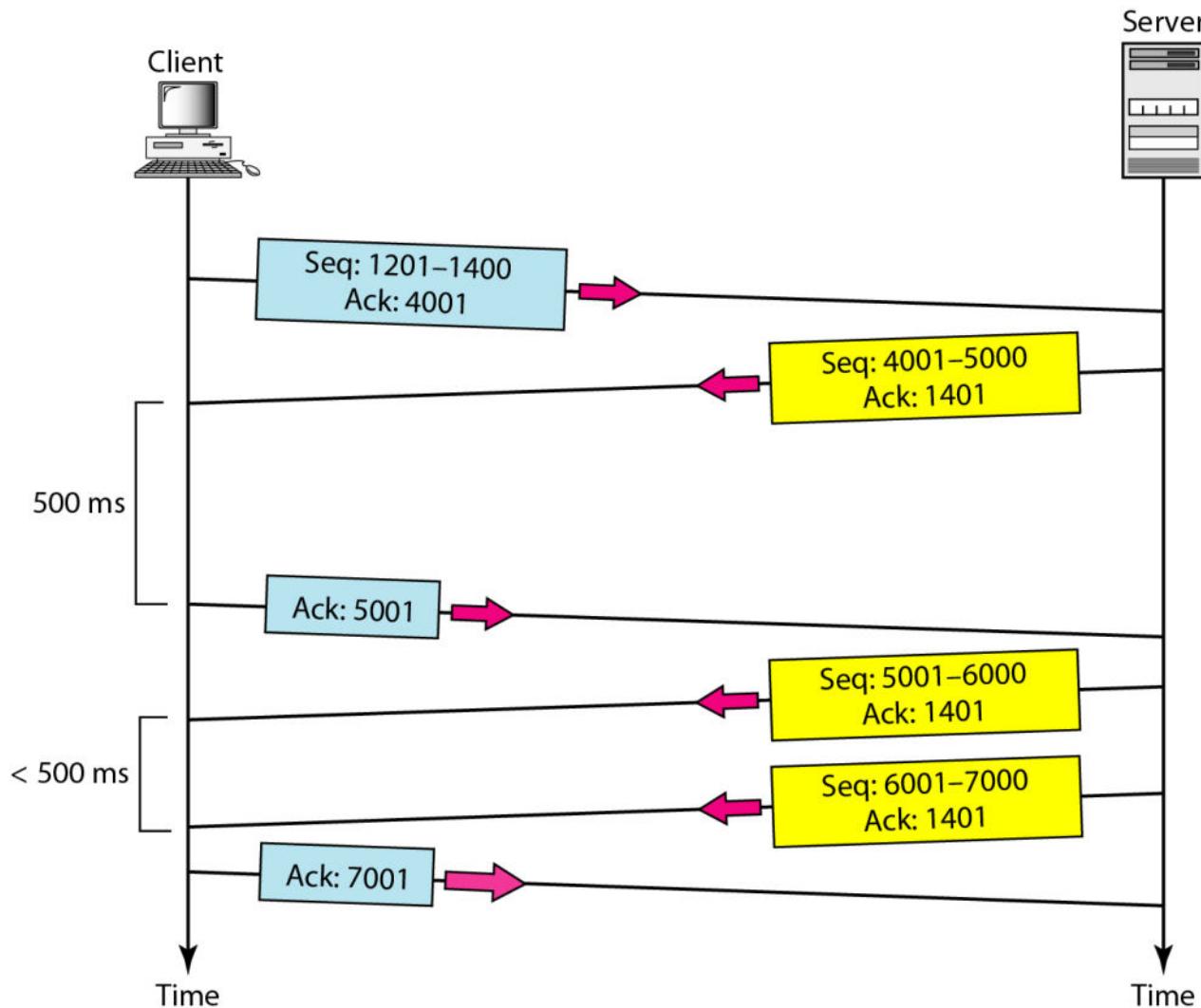
No retransmission timer is set for an ACK segment.



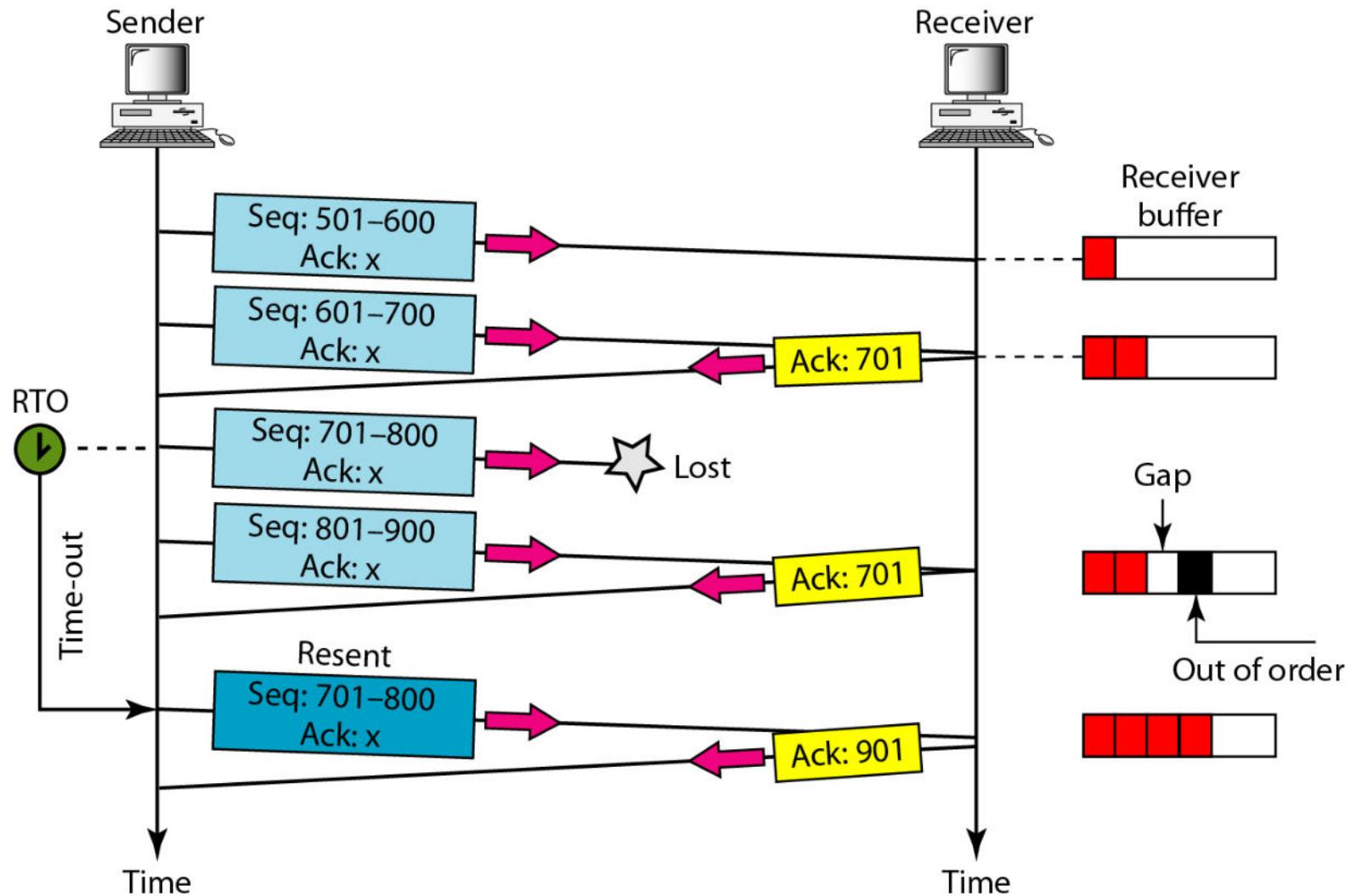
Note

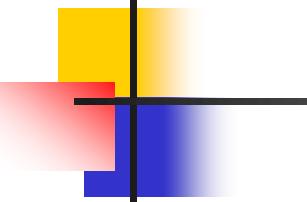
Data may arrive out of order and be temporarily stored by the receiving TCP, but TCP guarantees that no out-of-order segment is delivered to the process.

Normal operation



Lost segment





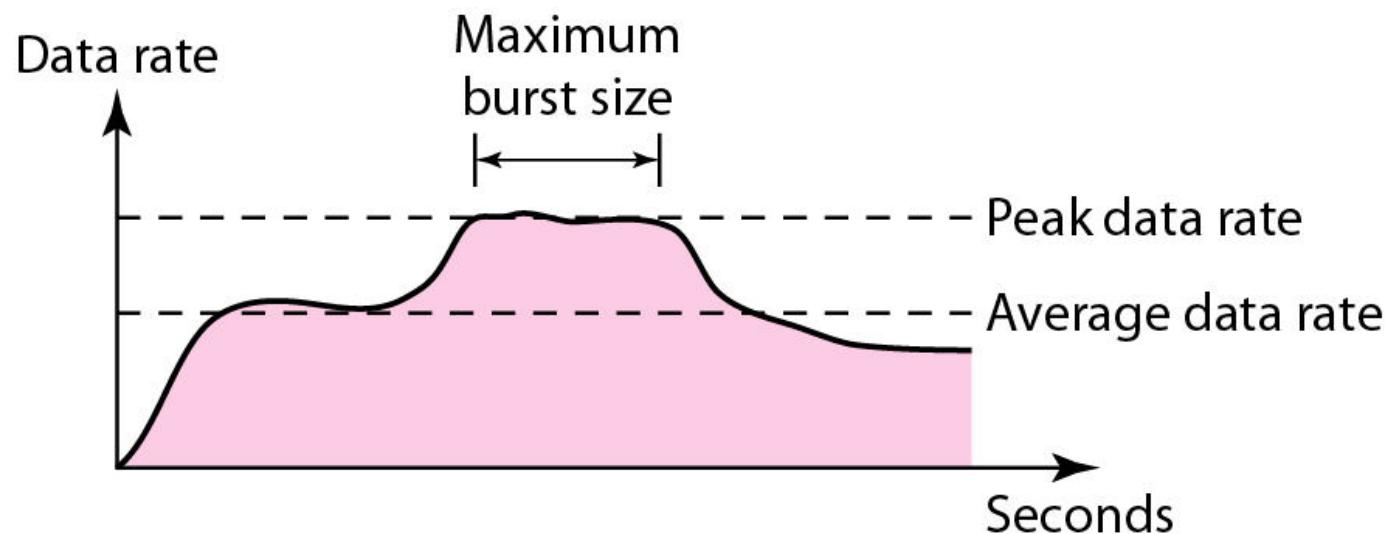
Note

The receiver TCP delivers only ordered data to the process.

Congestion Control

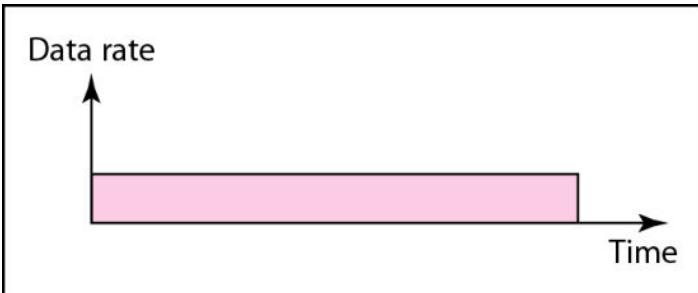
- DATA TRAFFIC:
- *The main focus of congestion control and quality of service is **data traffic**. In congestion control we try to avoid traffic congestion. In quality of service, we try to create an appropriate environment for the traffic. So, before talking about congestion control and quality of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

Traffic descriptors: data flow value

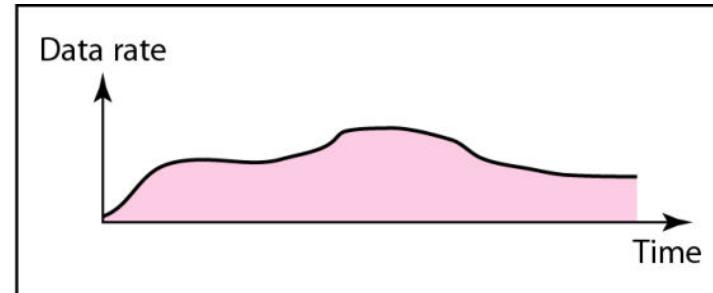


Effective bandwidth is a function of these 3 values

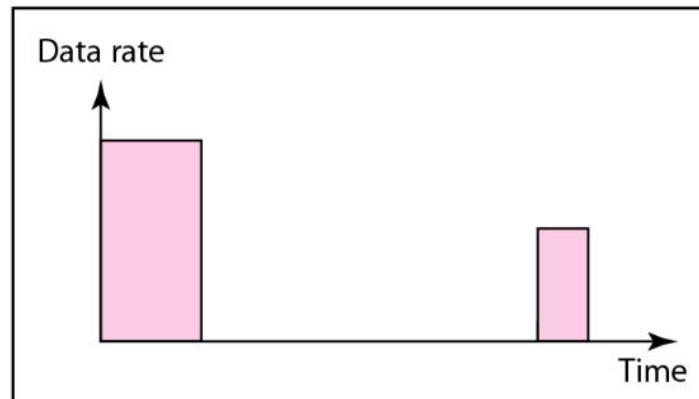
Three traffic profiles



a. Constant bit rate



b. Variable bit rate

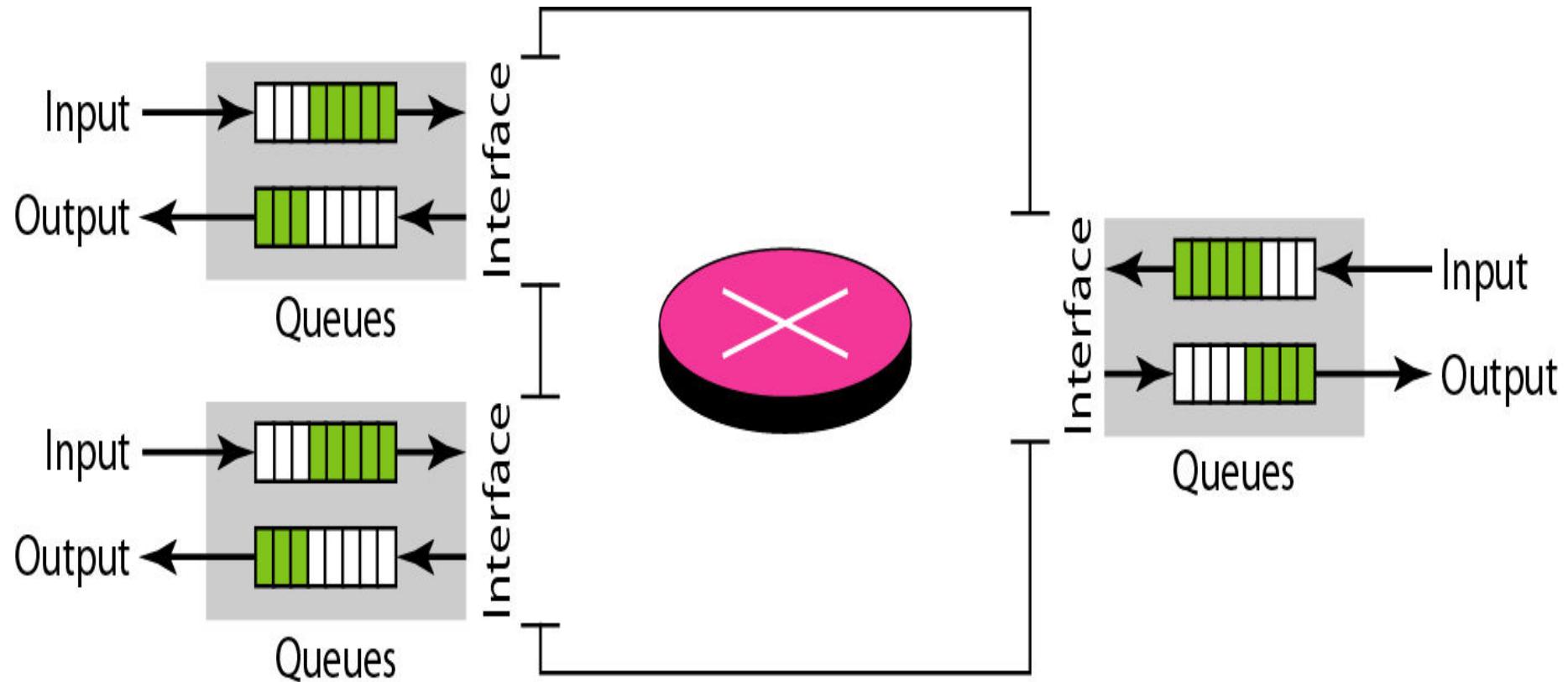


c. Bursty

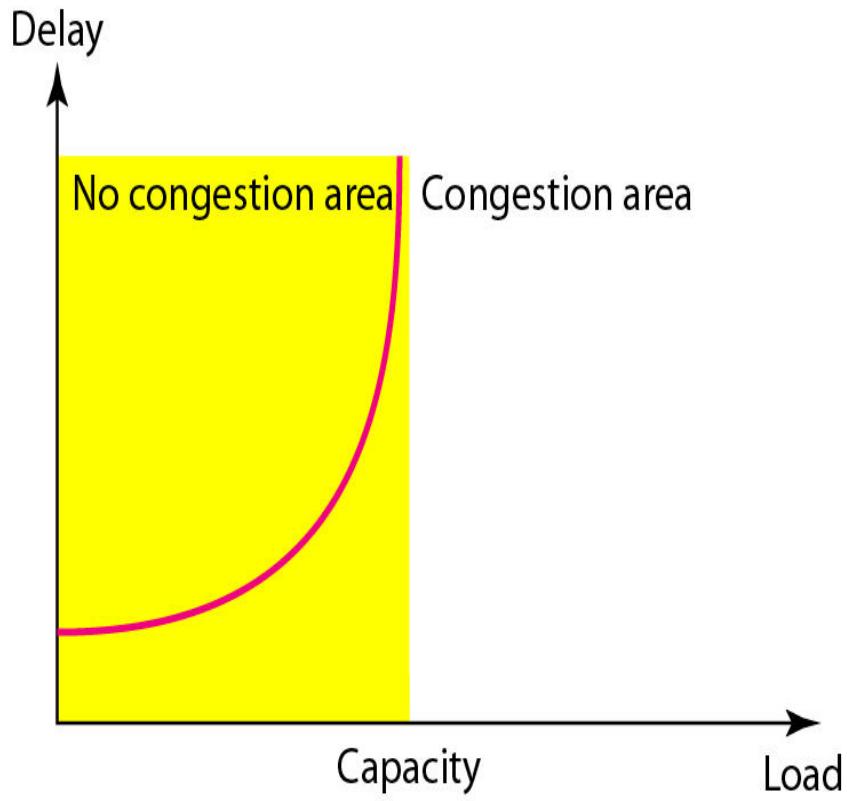
CONGESTION

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

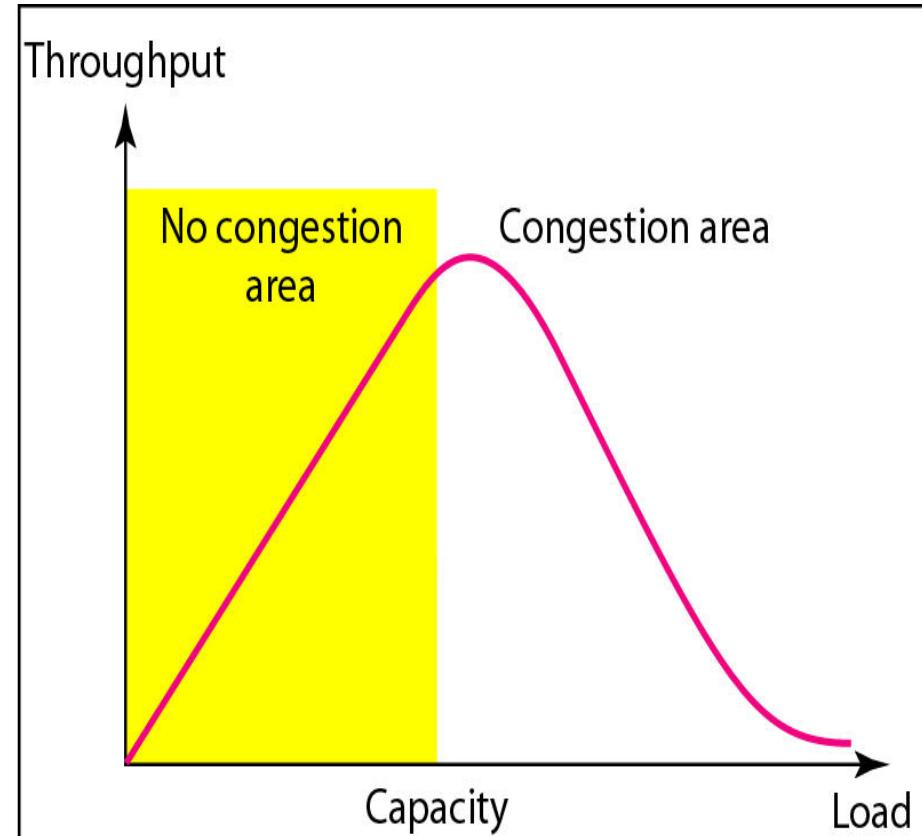
Queues in a router



Packet delay and throughput as functions of load



a. Delay as a function of load



b. Throughput as a function of load

CONGESTION CONTROL

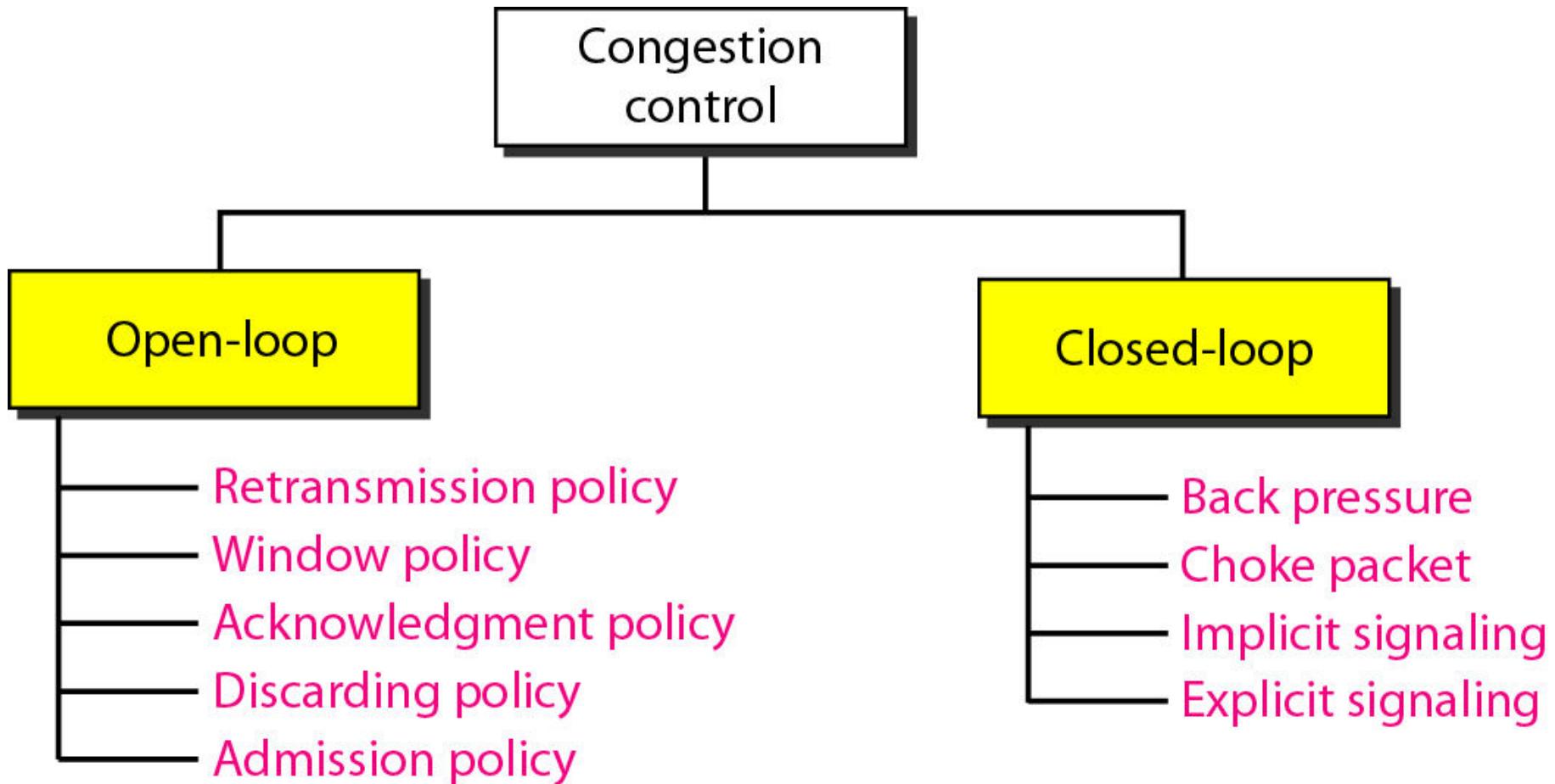
Congestion control refers to techniques and mechanisms that can either prevent congestion, before it happens, or remove congestion, after it has happened. In general, we can divide congestion control mechanisms into two broad categories: open-loop congestion control (prevention) and closed-loop congestion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

Topics discussed in this section:

Open-Loop Congestion Control

Closed-Loop Congestion Control

Congestion control categories

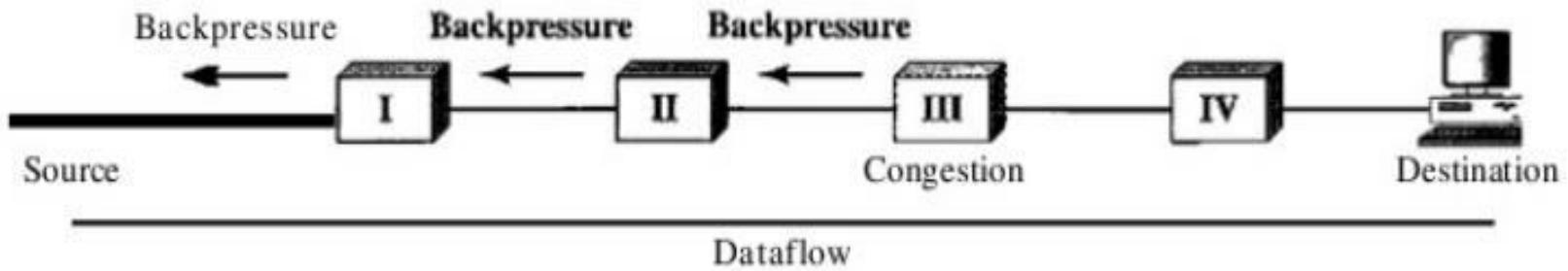


Closed Loop Congestion Control

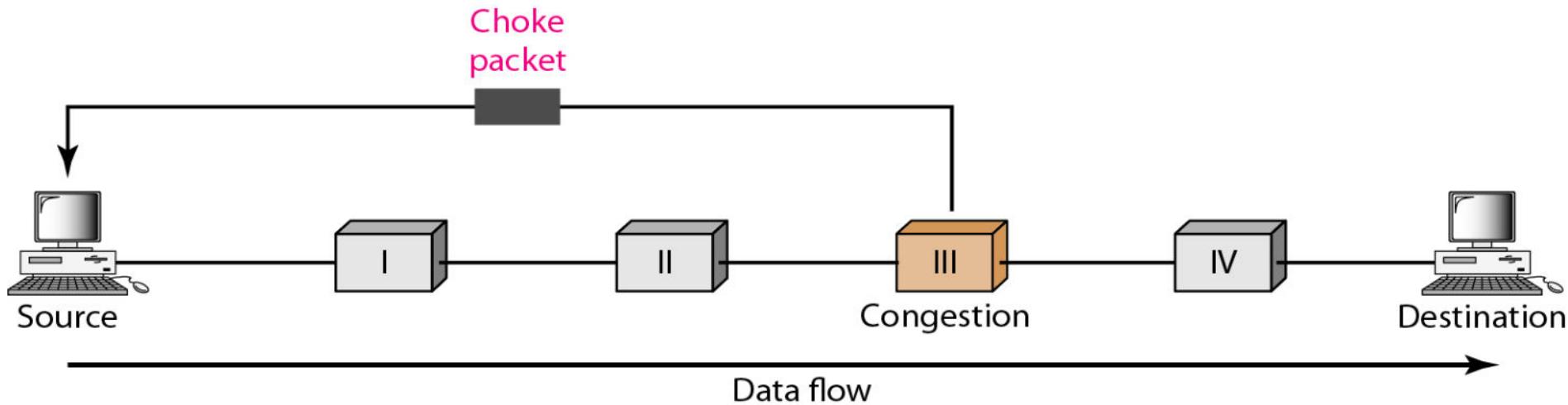
Backpressure method for alleviating congestion

Backpressure mechanism stops receiving data from the immediate upstream node or nodes. This may cause the upstream node or nodes to become congested, and they, in turn, reject data from their upstream nodes or nodes. And so on.

Backpressure is a node-to-node congestion control that starts with a node and propagates, in the opposite direction of data flow, to the source. The backpressure technique can be applied only to **virtual circuit networks**, in which each node knows the upstream node from which a flow of data is coming.



Choke packet



A choke packet is a packet sent by a node to the source to inform it of congestion.
difference between the backpressure and choke packet methods:

- In backpressure, the warning is from one node to its upstream node, although the warning may eventually reach the source station.
- In the choke packet method, the warning is from the router, which has encountered congestion, to the source station directly. The intermediate nodes through which the packet has traveled are not warned.

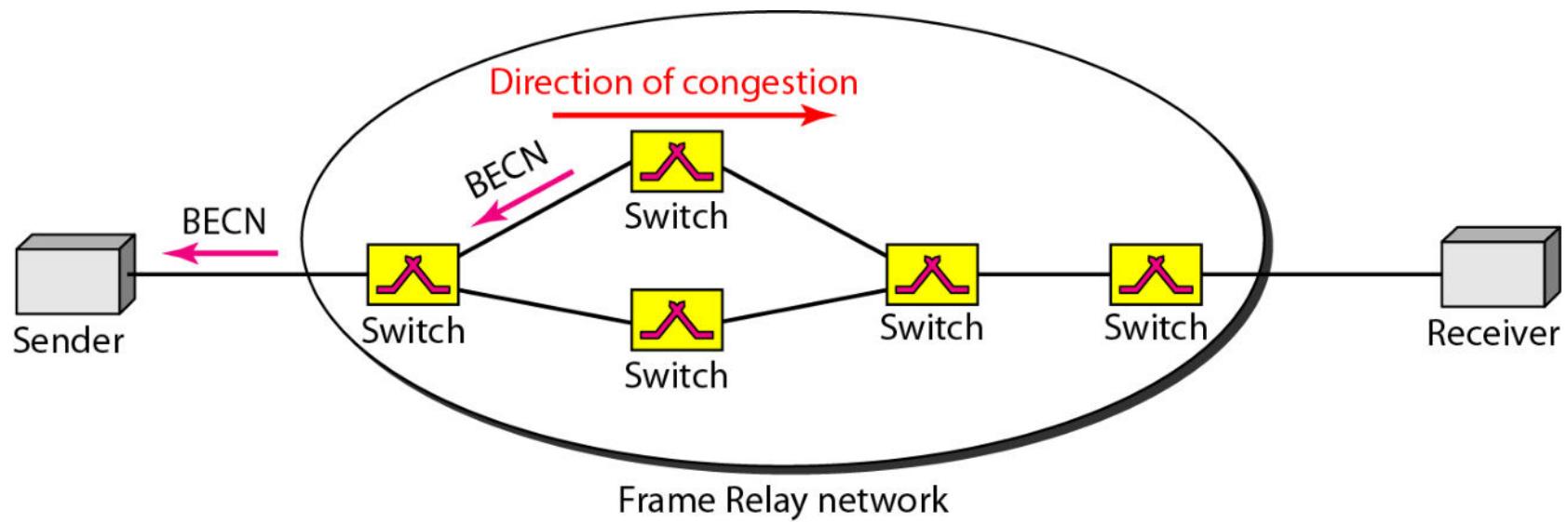
Implicit Signaling

- No communication between the congested node or nodes and the source.
- Source guesses that there is congestion somewhere in the network when it does not receive any acknowledgment. Therefore the delay in receiving an acknowledgment is interpreted as congestion in the network.
- On sensing this congestion, the source slows down.
- This type of congestion control policy is used by TCP.

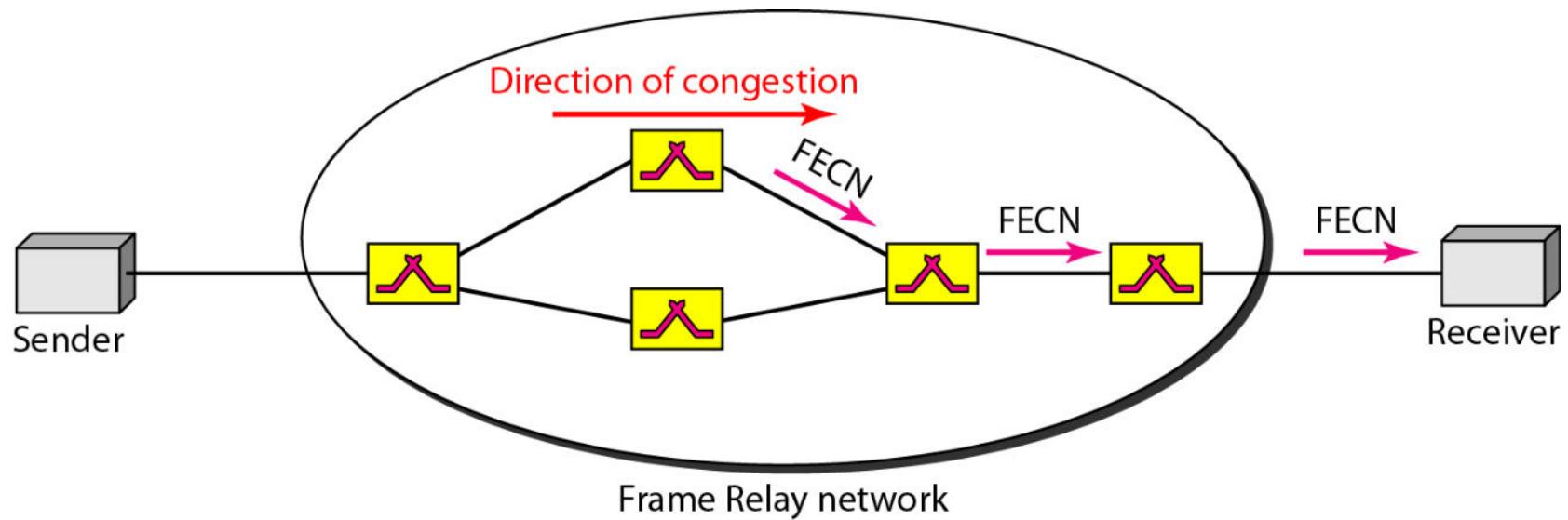
Explicit Signaling

- Different from the choke packet method:
- In the choke packet method, a separate packet is used for this purpose; in the explicit signaling method, the signal is included in the packets that carry data.
- Explicit signaling, congestion control can occur in either the forward or the backward direction.
 - Backward Signaling A bit can be set in a packet moving in the direction opposite to the congestion. This bit can warn the source that there is congestion and that it needs to slow down to avoid the discarding of packets.
 - Forward Signaling A bit can be set in a packet moving in the direction of the congestion. This bit can warn the destination that there is congestion. The receiver in this case can use policies, such as slowing down the acknowledgments, to alleviate the congestion.

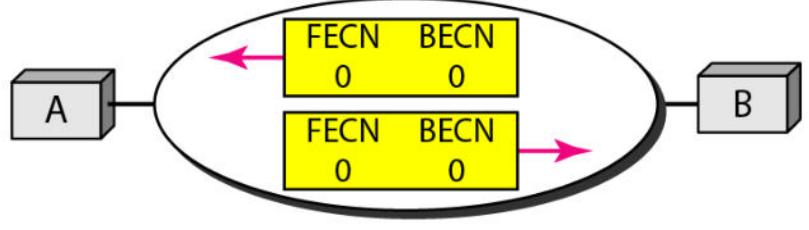
BECN



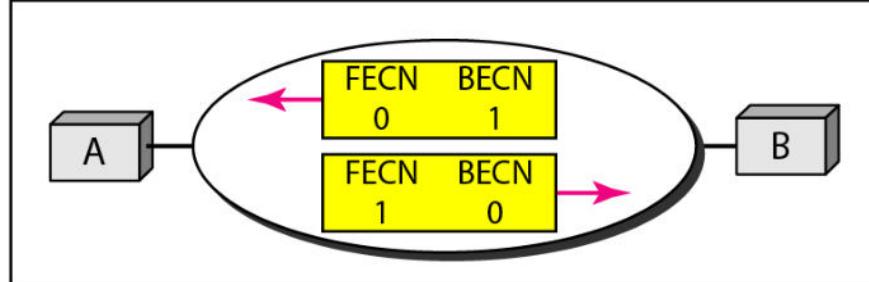
FECN



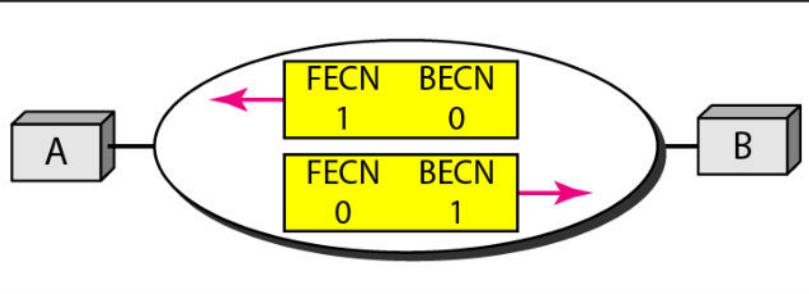
Four cases of congestion



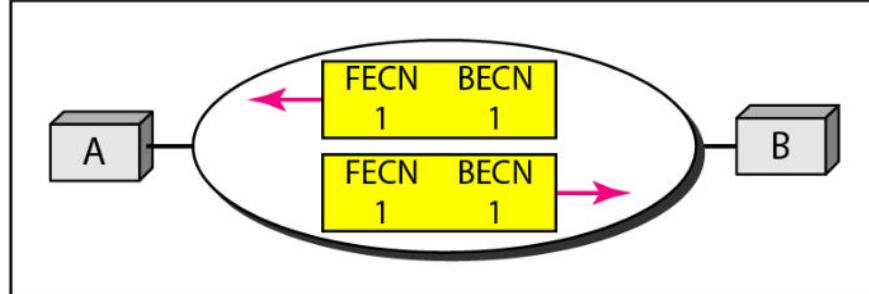
a. No congestion



b. Congestion in the direction A-B



c. Congestion in the direction B-A



d. Congestion in both directions

Congestion Control

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

Network Congestion

- Over-subscription
- Poor network design/mis-configuration
- Over-utilized devices
- Faulty devices
- Security attack

Effects of Network Congestion

- **Delay**
- **Packet Loss**
- **Timeouts**
- **Jitter**
- **Buffer Memory is Full**
- **Severe Performance Degradation**
- **Loss of Customers**

Effects of Congestion

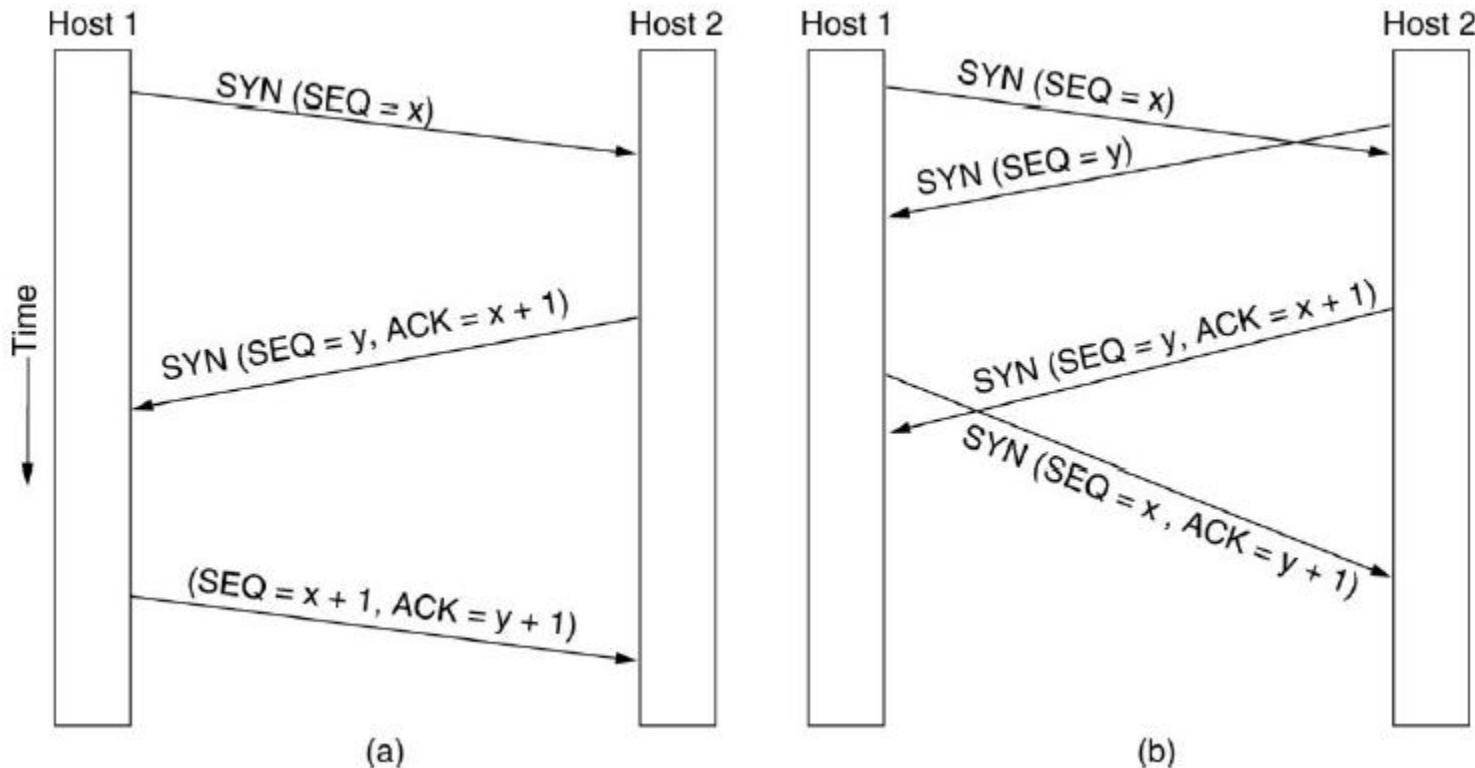
- Packets arriving are stored at input buffers
- Routing decision made
- Packet moves to output buffer
- Packets queued for output transmitted as fast as possible
 - Statistical time division multiplexing
- If packets arrive too fast to be routed, or to be output, buffers will fill
- Can discard packets
- Can use flow control
 - Can propagate congestion through network

Troubleshooting Network Congestion

- **Ping**
- **LAN Performance Tests**
- **Bandwidth Monitoring**

TCP Connection Management

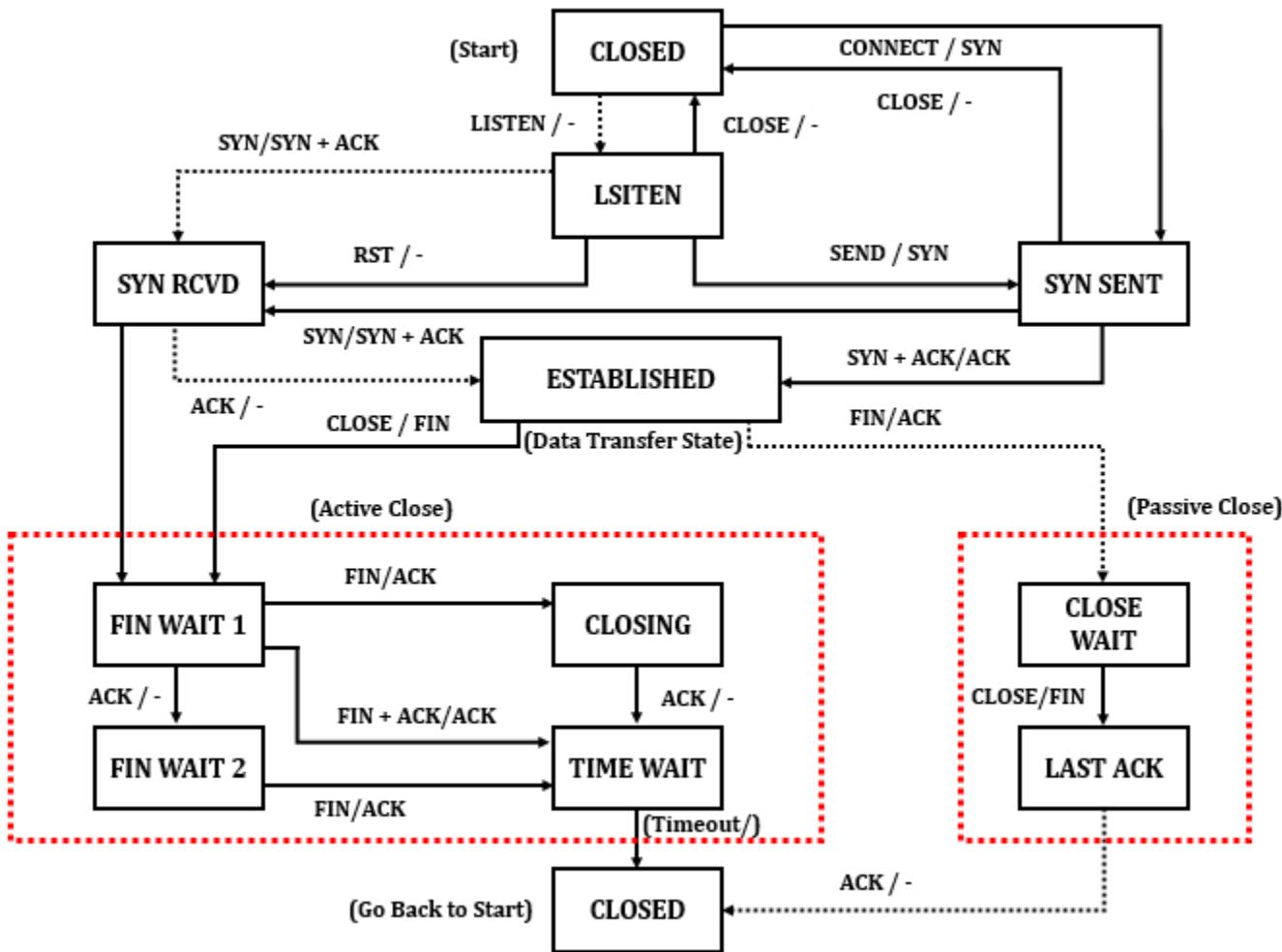
■ TCP Connection Establishment



TCP Connection Management

■ TCP Connection Release

TCP Connection Management



TWO EXAMPLES

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

Topics discussed in this section:

Congestion Control in TCP

Congestion Control in Frame Relay

TCP Congestion Control

- **Congestion policy in TCP:**
- Slow Start Phase: starts slowly increment is exponential to threshold
- Congestion Avoidance Phase: After reaching the threshold increment is by 1
- Congestion Detection Phase: Sender goes back to Slow start phase or Congestion avoidance phase.

TCP Congestion Control: Slow Start Phase : exponential increment

- In this phase after every RTT the congestion window size increments exponentially. RTT means a round-trip delay that includes queueing delays at the routers preceding the congested links.

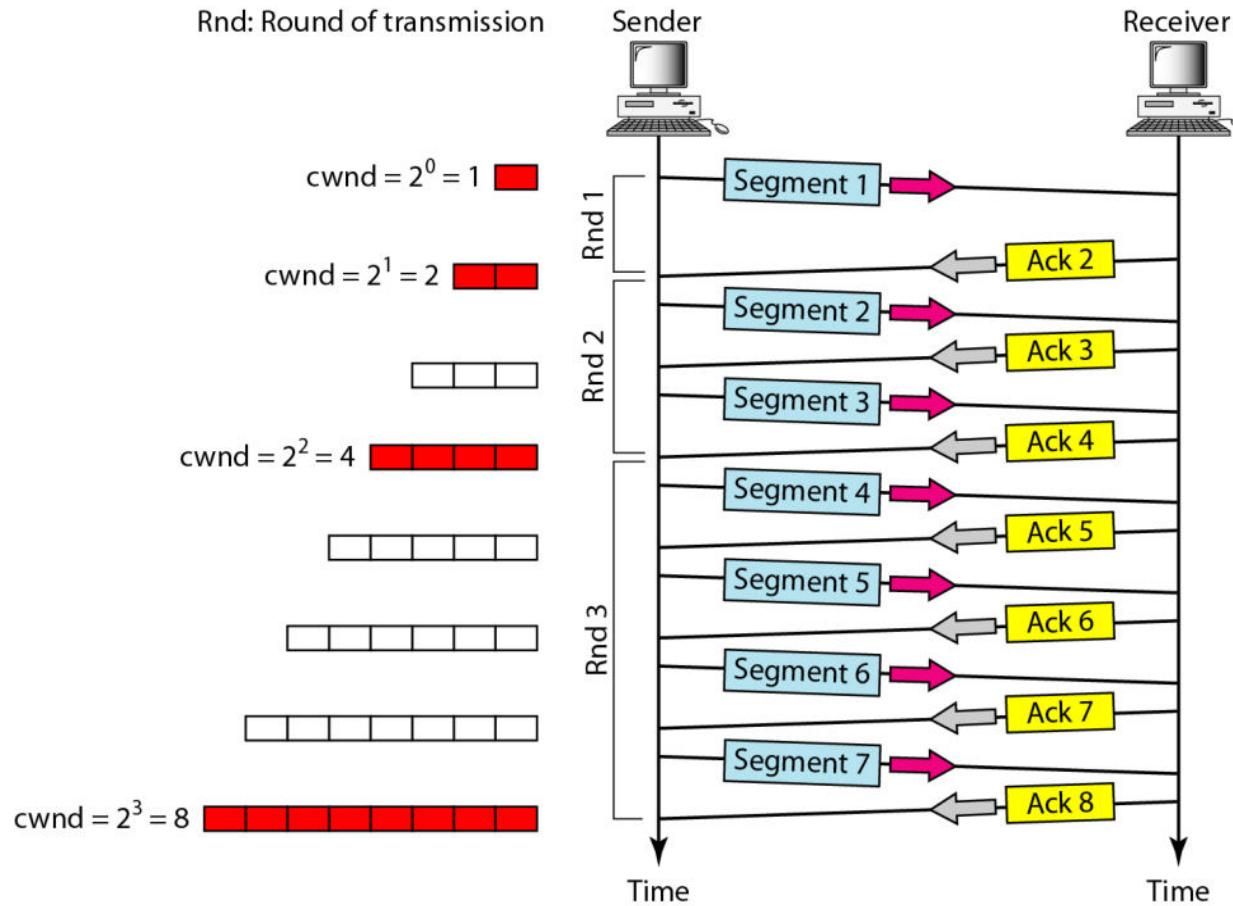
Initially cwnd = 1

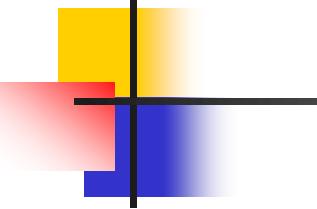
After 1 RTT, cwnd = $2^{(1)}$ = 2

2 RTT, cwnd = $2^{(2)}$ = 4

3 RTT, cwnd = $2^{(3)}$ = 8

Slow start, exponential increase





Note

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

Congestion Avoidance Mechanisms in TCP

- **additive increment** – This phase starts after the threshold value also denoted as *ssthresh*.
- The size of *cwnd* (congestion window) increases additive. After each RTT $cwnd = cwnd + 1$.

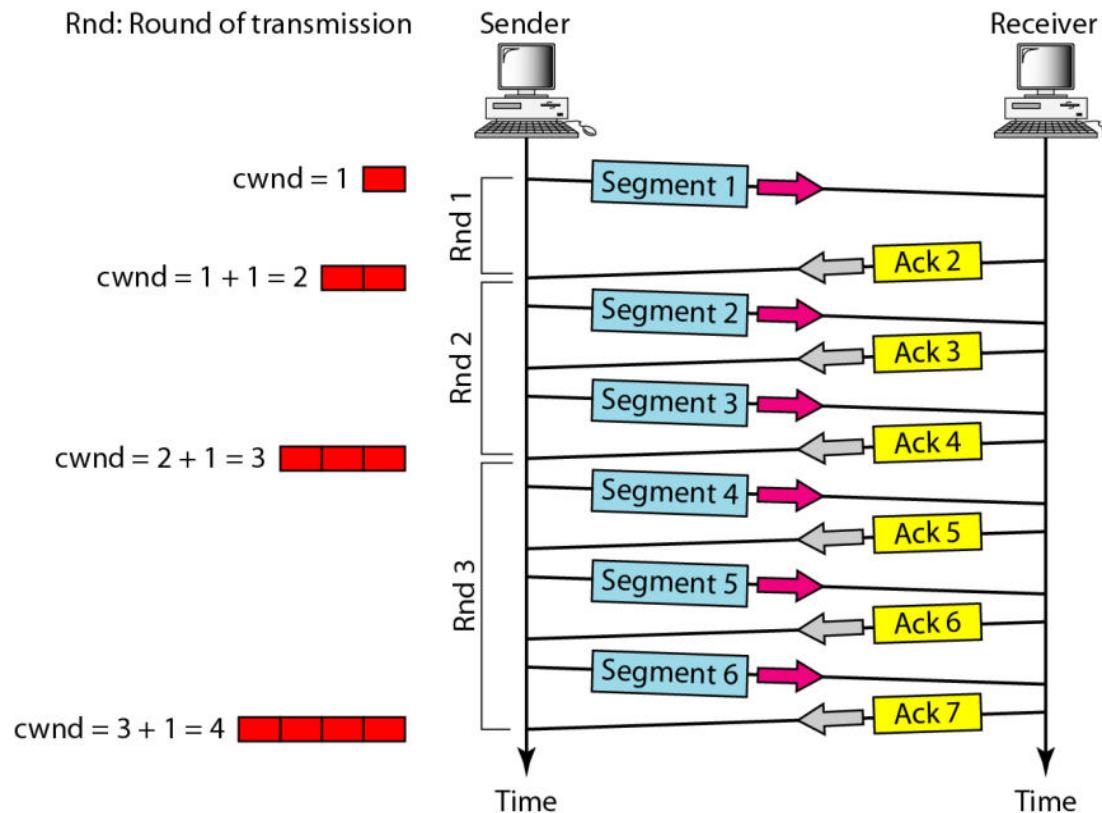
Initially $cwnd = i$

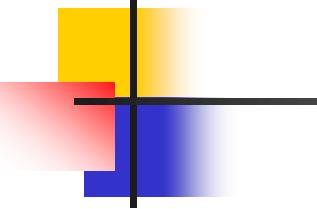
After 1 RTT, $cwnd = i+1$

2 RTT, $cwnd = i+2$

3 RTT, $cwnd = i+3$

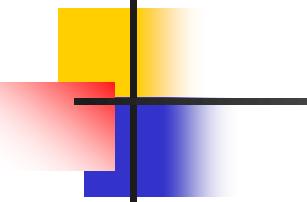
Figure Congestion avoidance, additive increase





Note

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

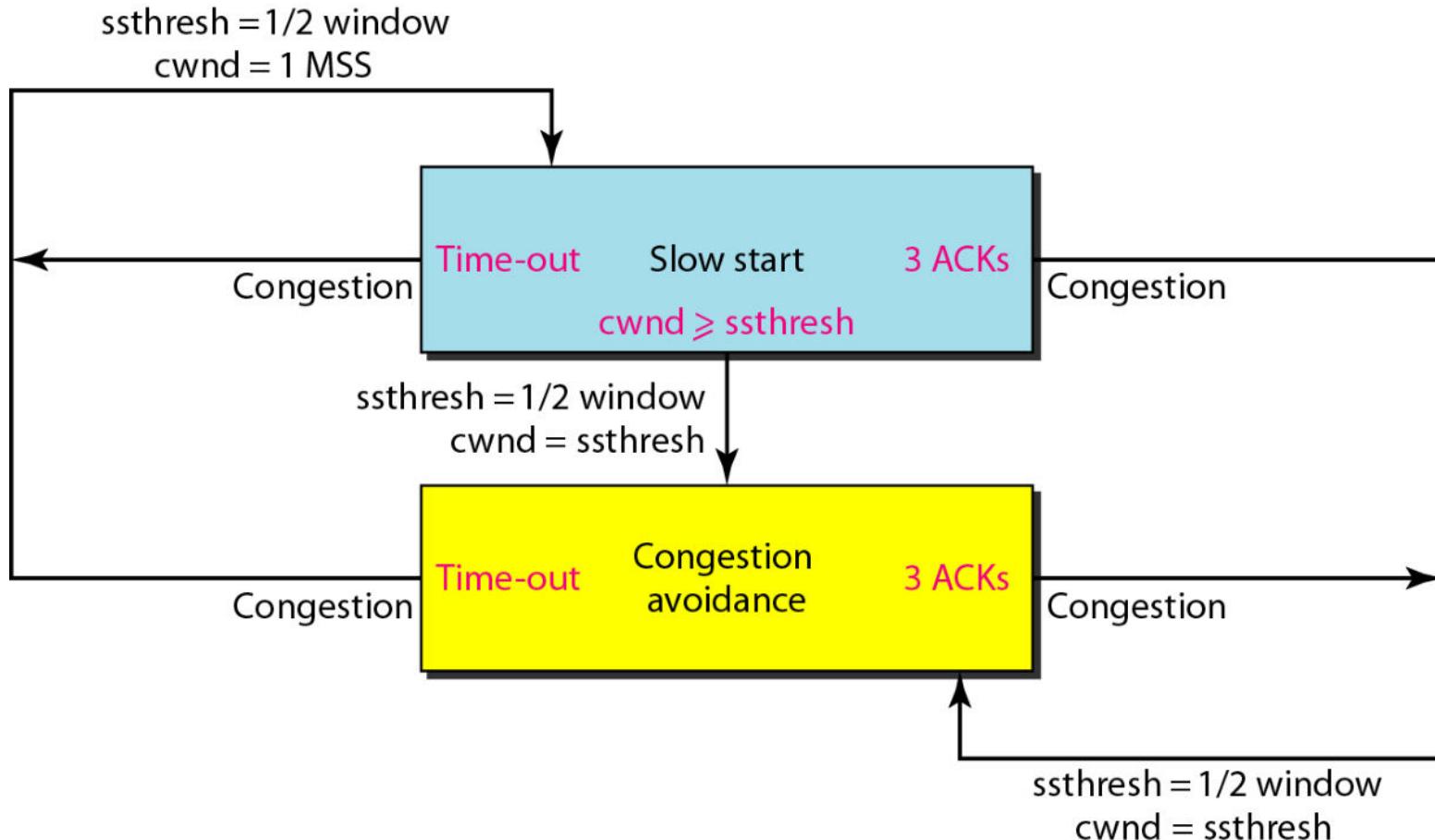


Note

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

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

TCP congestion policy summary



Congestion Detection Phase

- **Multiplicative decrement** – If congestion occurs, the congestion window size is decreased. The only way a sender can guess that congestion has occurred is the need to retransmit a segment.
- Retransmission can occur in one of two cases: when the RTO timer times out or when three duplicate ACKs are received.

Congestion Detection Phase

- **Case 1 : Retransmission due to Timeout**
- In this case congestion possibility is high.
 - (a) ssthresh is reduced to half of the current window size.
 - (b) set cwnd = 1
 - (c) start with slow start phase again.

Congestion Detection Phase

- **Case 2 : Retransmission due to 3 Acknowledgement Duplicates**
- In this case congestion possibility is less.
- (a) ssthresh value reduces to half of the current window size.
- (b) set cwnd= ssthresh
- (c) start with congestion avoidance phase

Congestion control algorithms

- **Leaky Bucket**
- **Token Bucket Algorithm**

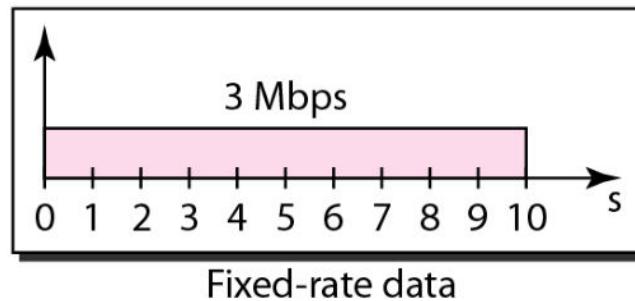
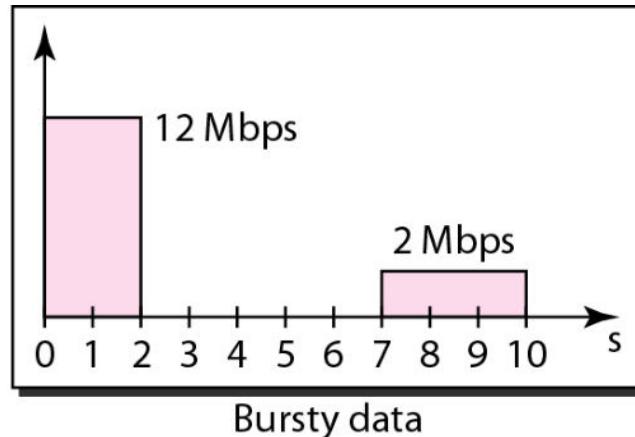
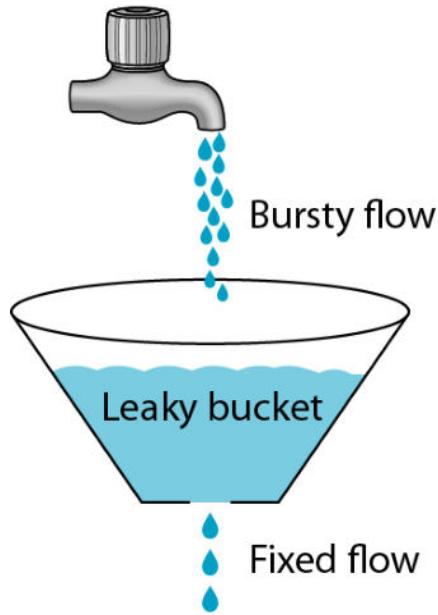
Traffic Shaping

- Traffic shaping is a mechanism to control the amount and the rate of the traffic sent to the network.
- Two techniques can shape traffic:
 - Leaky bucket
 - Token bucket.

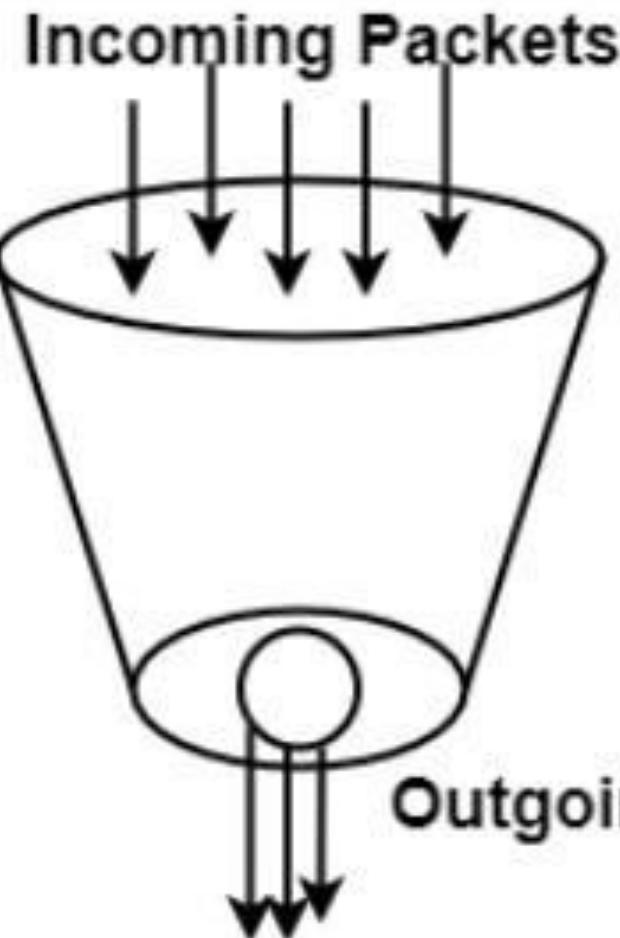
Leaky Bucket

- If a bucket has a small hole at the bottom, the water leaks from the bucket at a constant rate as long as there is water in the bucket.
- The rate at which the water leaks does not depend on the rate at which the water is input to the bucket unless the bucket is empty.
- The input rate can vary, but the output rate remains constant. Similarly, in networking, a technique called **leaky bucket** can smooth out bursty traffic.
- Bursty chunks are stored in the bucket and sent out at an average rate. Figure shows a leaky bucket and its effects.

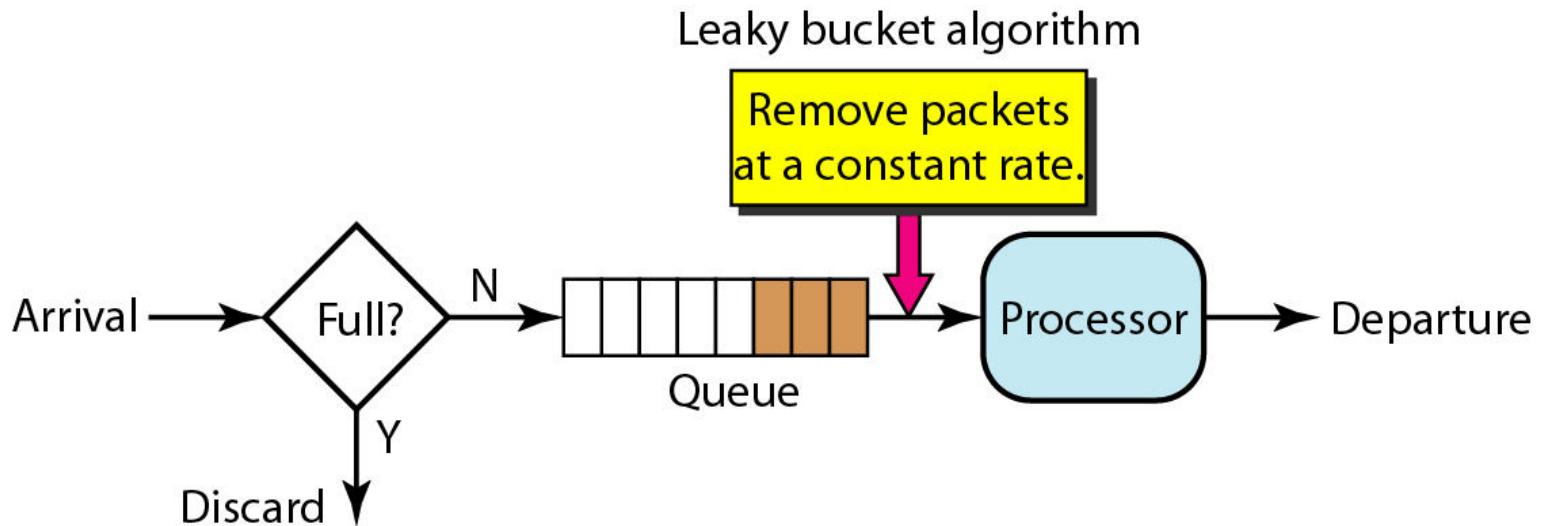
Leaky bucket

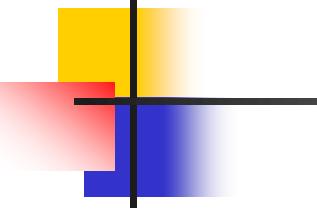


Leaky Bucket Algorithm



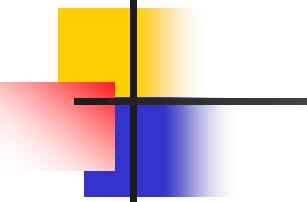
Leaky bucket implementation





Note

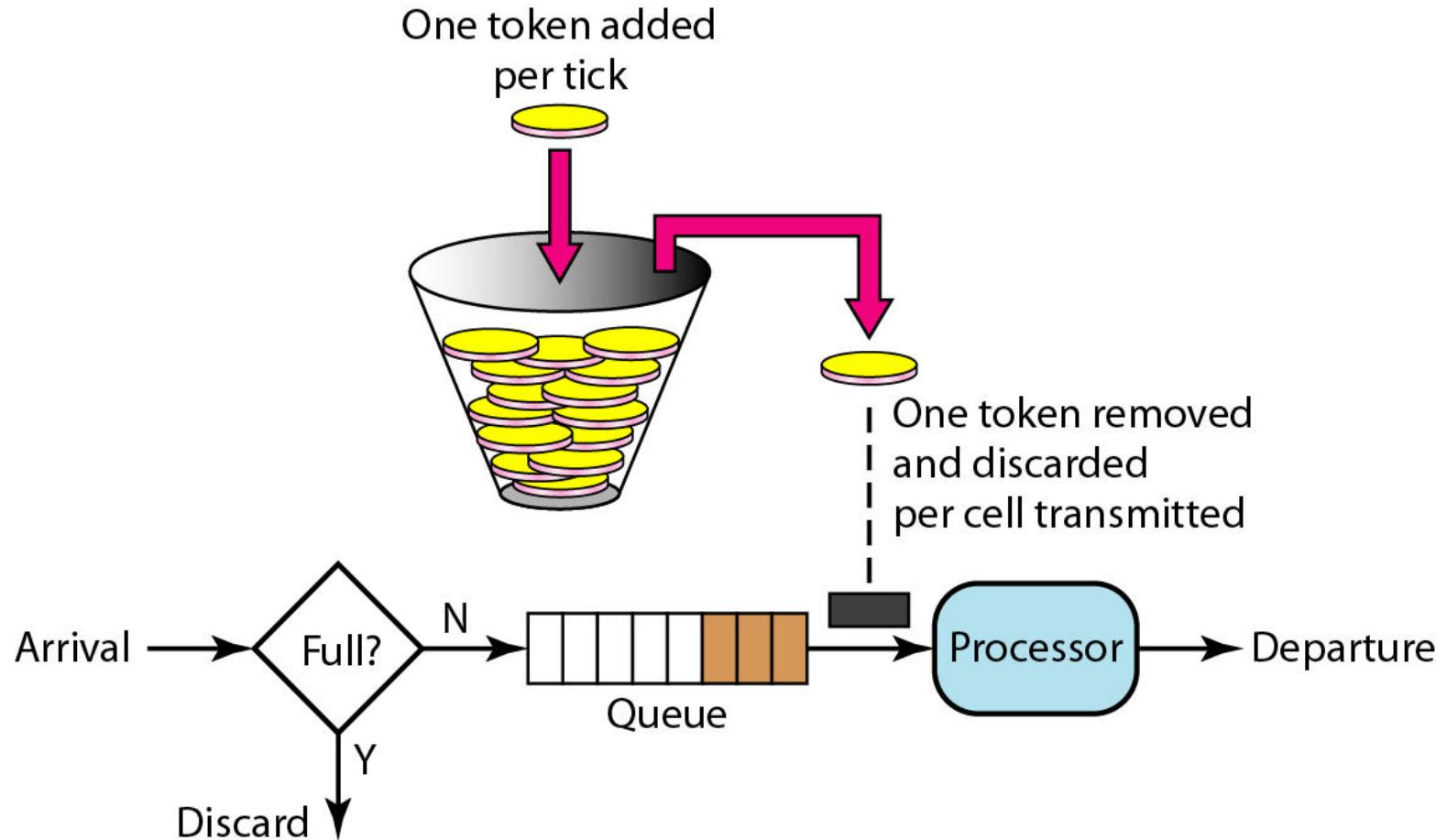
A leaky bucket algorithm shapes bursty traffic into fixed-rate traffic by averaging the data rate. It may drop the packets if the bucket is full.



Note

The token bucket allows bursty traffic at a regulated maximum rate.

Token bucket



Combining Token bucket and Leaky bucket

QUALITY OF SERVICE

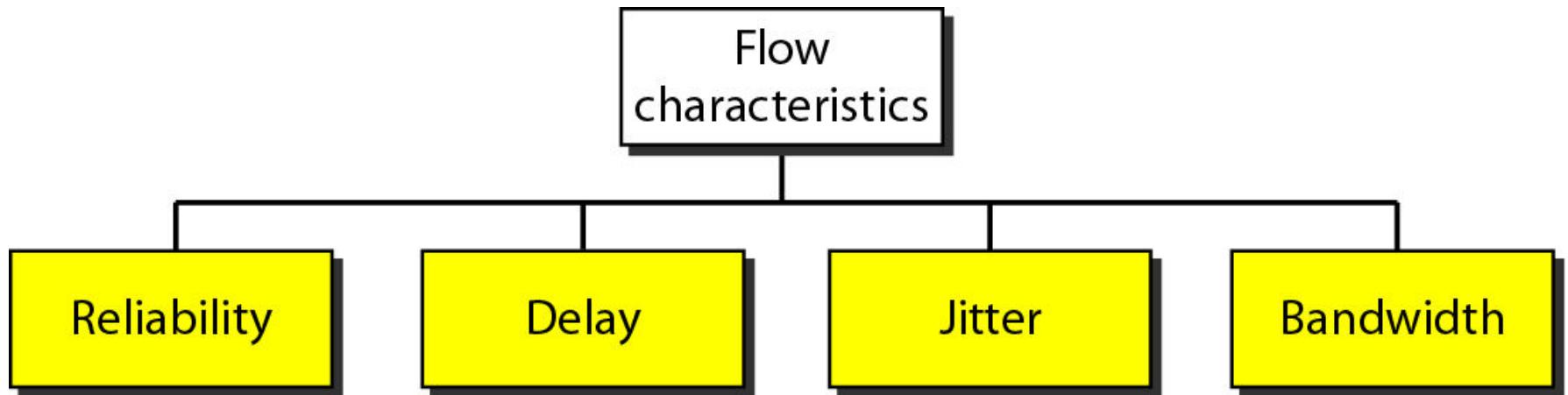
Quality of service (QoS) is an internetworking issue that has been discussed more than defined. We can informally define quality of service as something a flow seeks to attain.

Topics discussed in this section:

Flow Characteristics

Flow Classes

Flow characteristics



- **Delay:** Is the amount of time data(signal) takes to reach the destination. Now a higher **delay** generally means congestion of some sort of breaking of the communication link.
- **Jitter:** Is the variation of **delay** time. This caused by network congestion, timing drift, or route changes.

TECHNIQUES TO IMPROVE QoS

*Some techniques can be used to improve the quality of service.
The four common methods: scheduling, traffic shaping,
admission control, and resource reservation.*

Topics discussed in this section:

Scheduling

Traffic Shaping

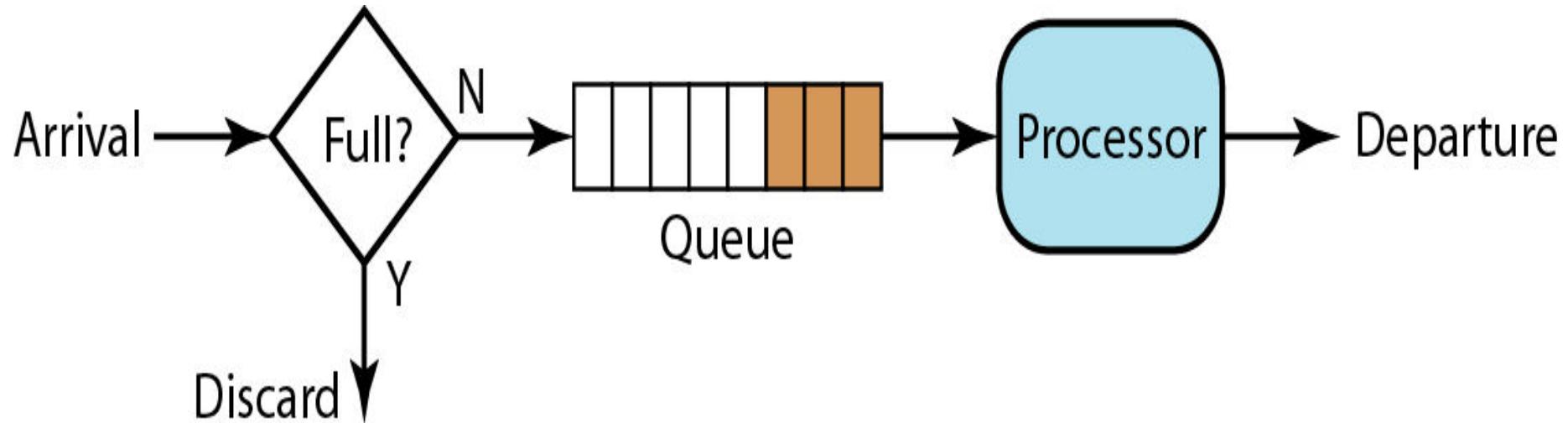
Resource Reservation

Admission Control

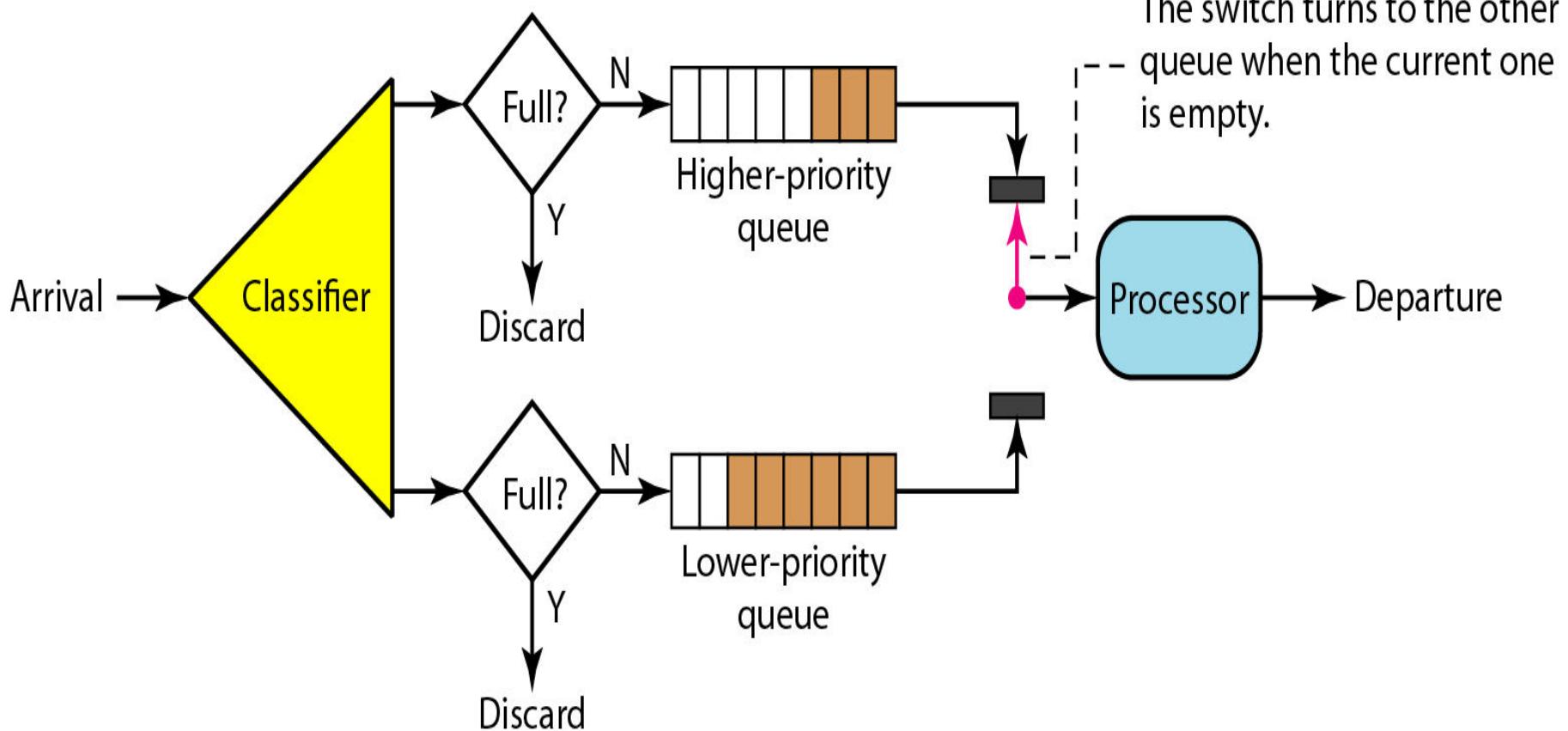
Scheduling

- Packets from different flows arrive at a switch or router for processing.
- A good scheduling technique treats the different flows in a fair and appropriate manner.
- Several scheduling techniques are designed to improve the quality of service.
 - FIFO queuing
 - Priority queuing
 - Weighted fair queuing.

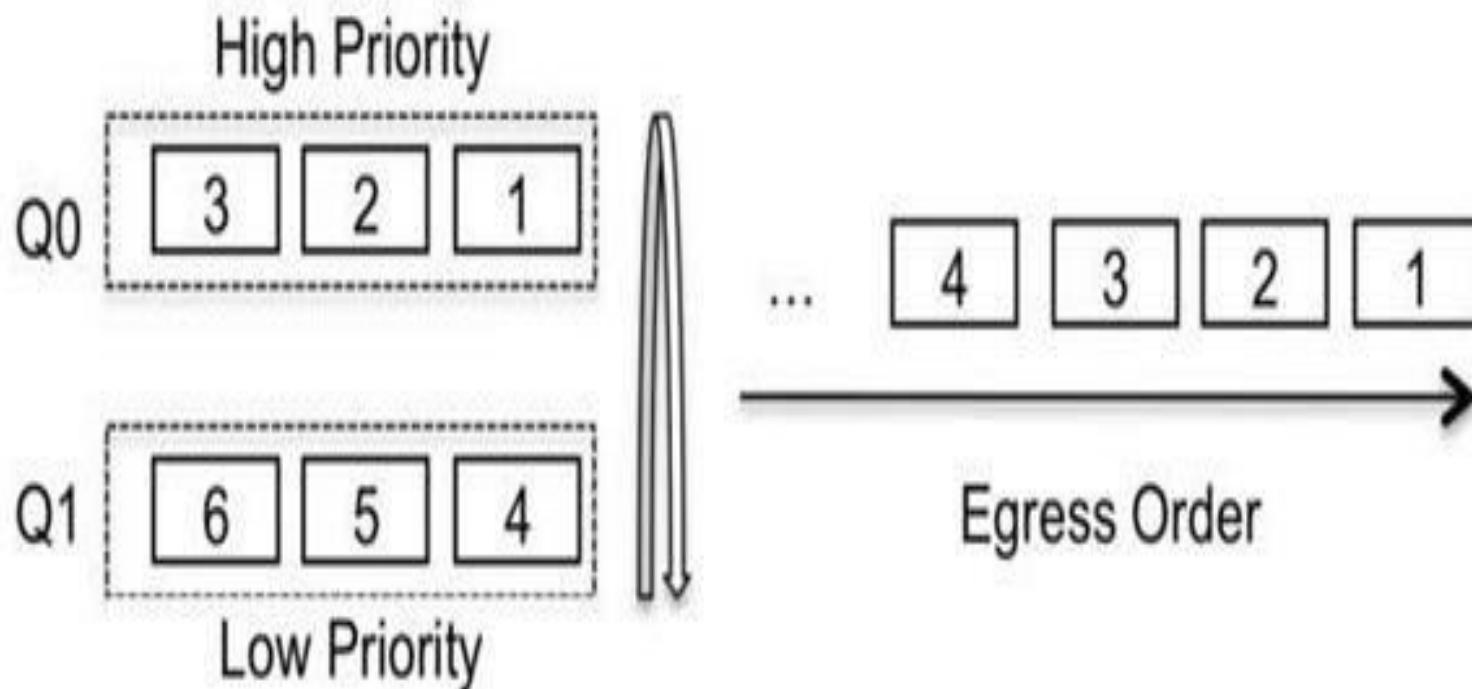
FIFO queue



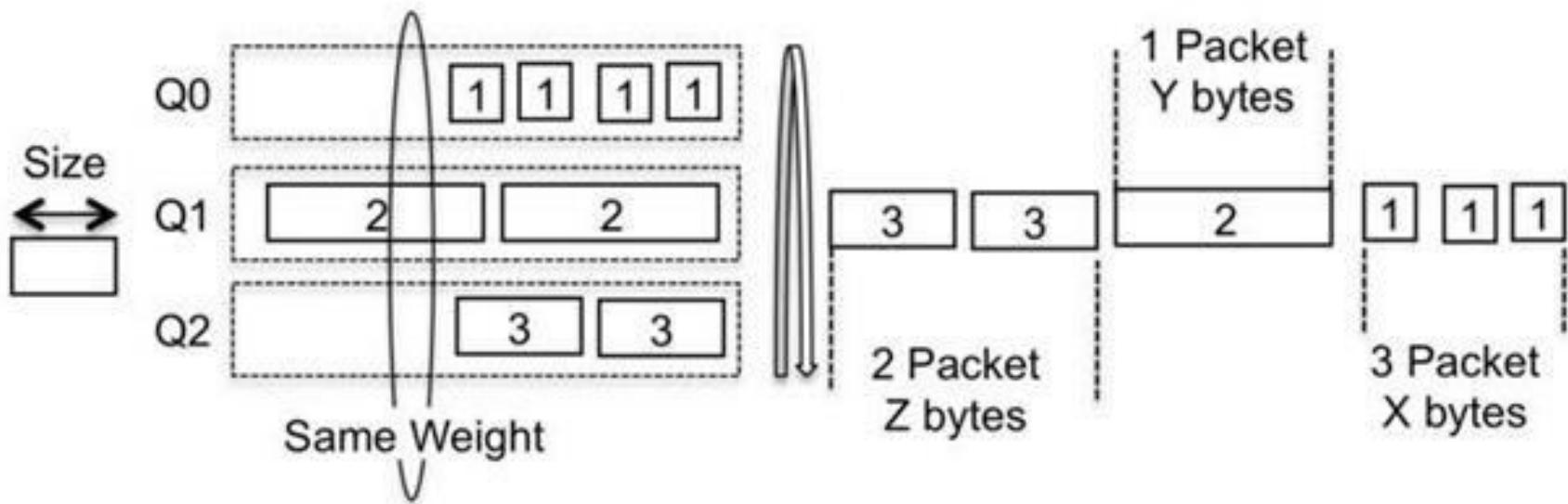
Priority queuing



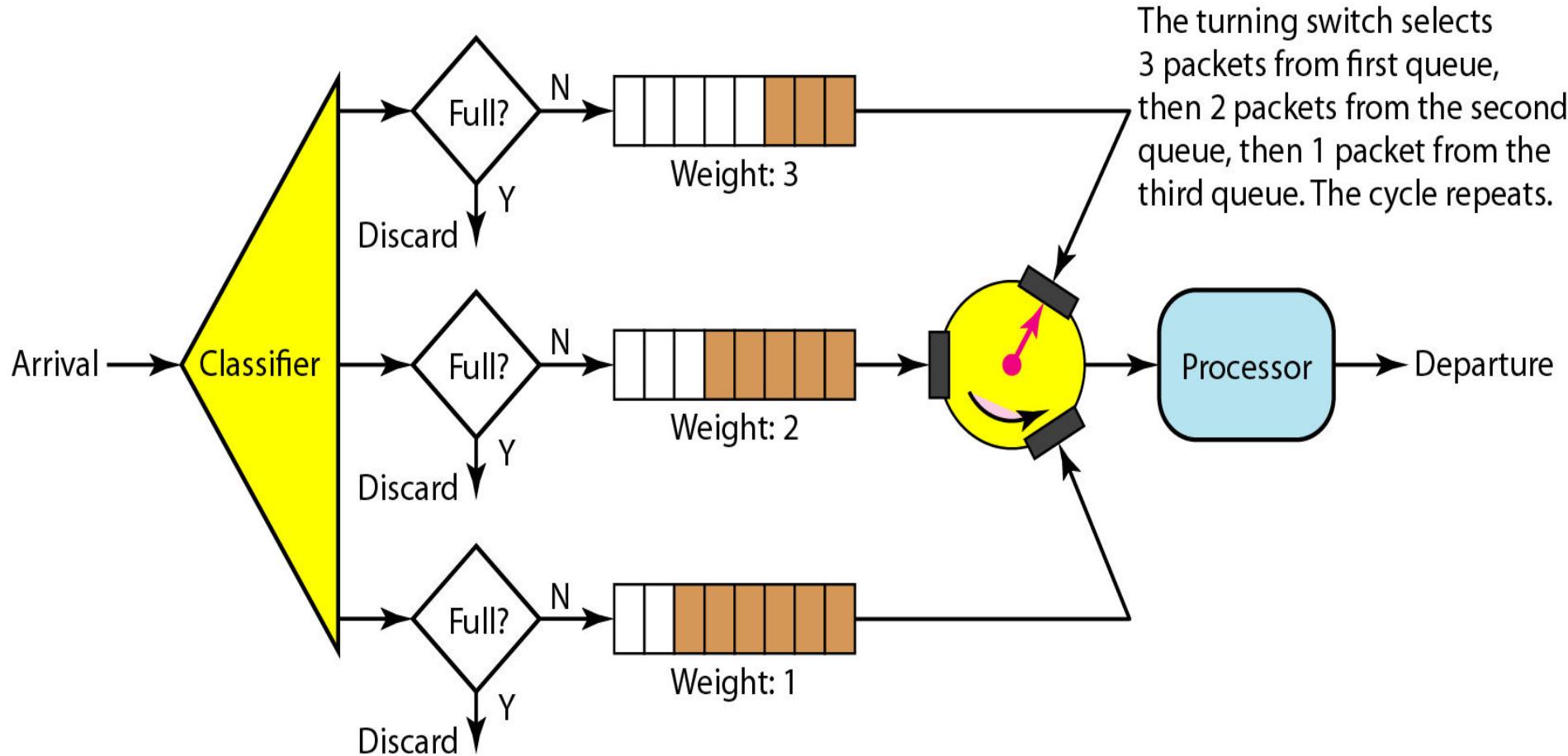
Priority queuing



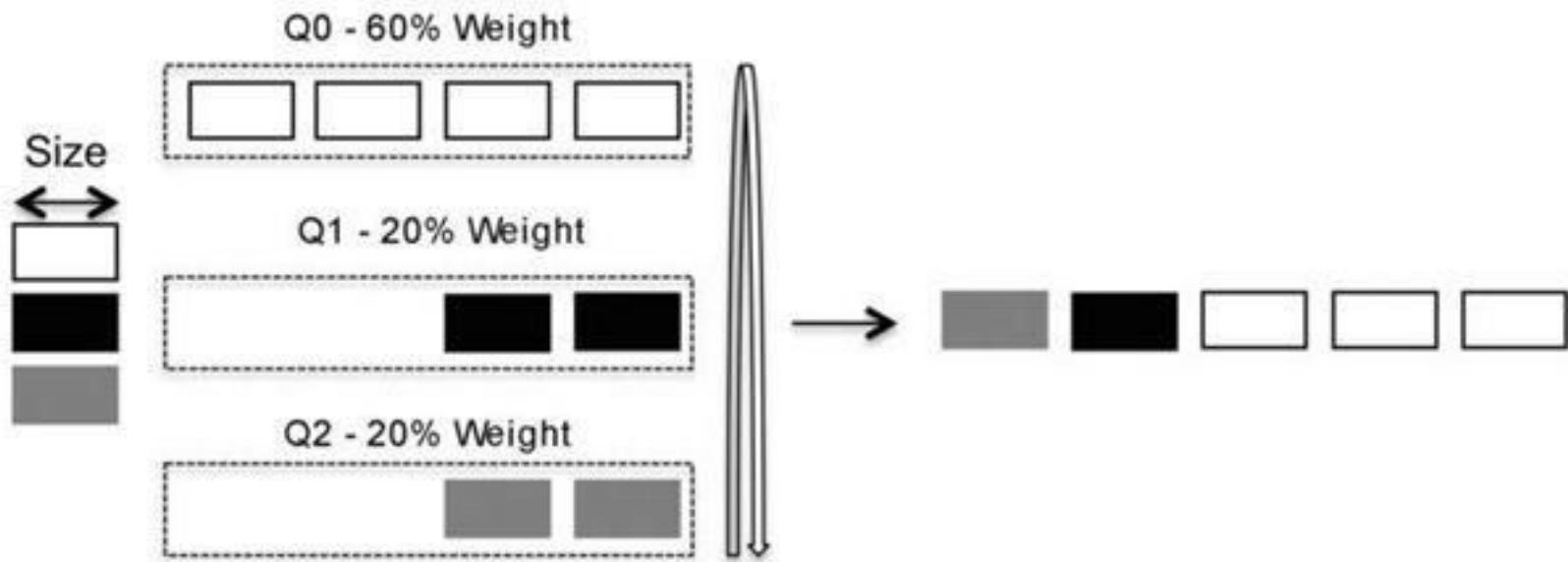
Weighted fair queuing



Weighted fair queuing



Weighted Round Robin



Deficit Weighted Round Robin

- **Key elements and parameters in implementing DWRR**
- Weight
- The quantum translates the weight value into bytes
- **The value of credits can be** positive or negative
- **The deficit counter**, which provides bandwidth fairness, is the sum of the quantum and the credits

Resource Reservation

- Buffer
- Bandwidth
- CPU time

Admission control

- Router / switch
 - Accept or reject flow based on flow specification.

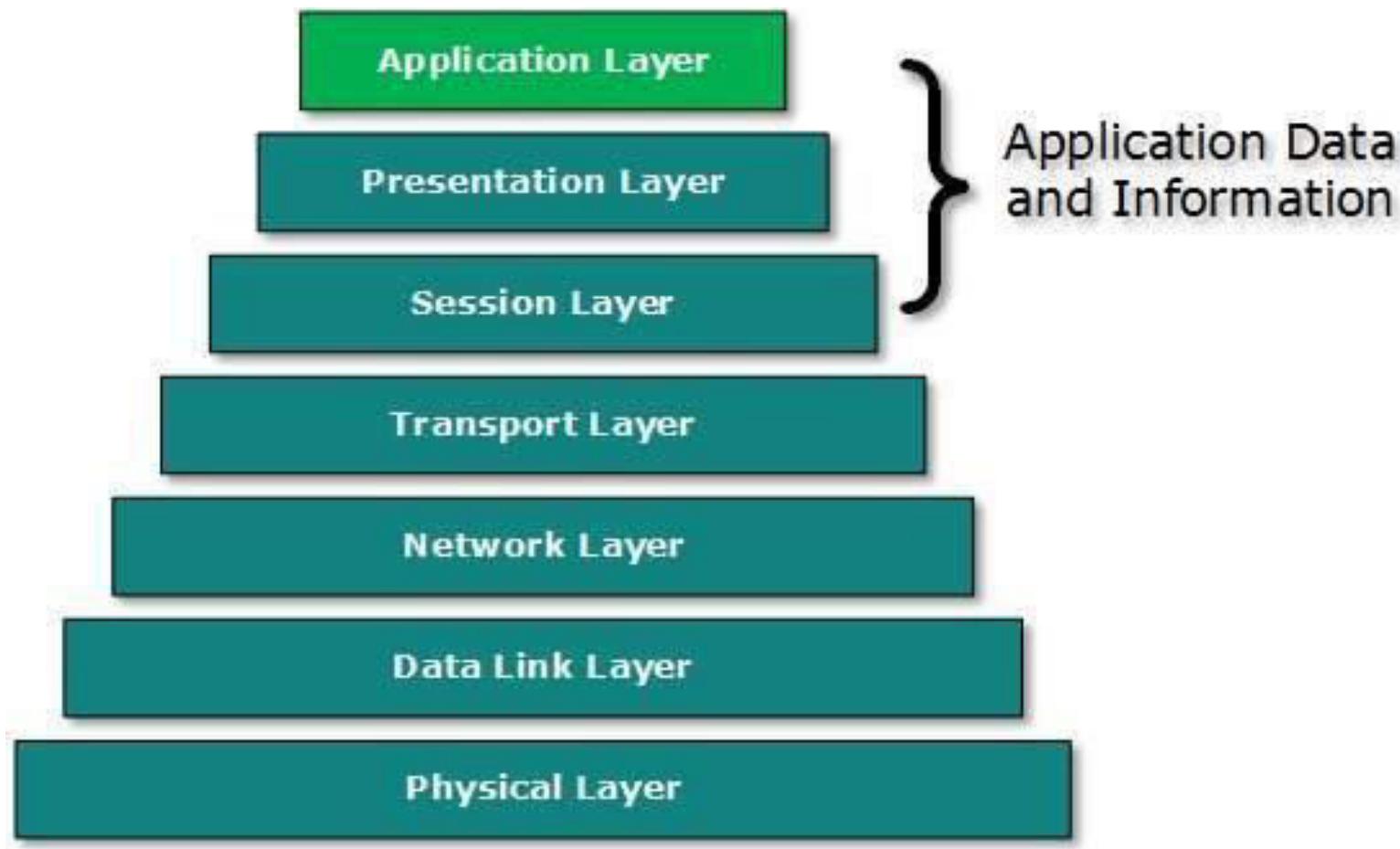
Module 7

Application layer

Application layer

- Is the top most layer in OSI and TCP/IP layered model. This layer exists in both layered Models because of its significance, of interacting with user and user applications.
- This layer is for applications which are involved in communication system. A user may or may not directly interacts with the applications.

Application layer



- Two remote application
- Peer-to-peer:** Both remote processes are executing at same level and they exchange data using some shared resource.
- Client-Server:** One remote process acts as a Client and requests some resource from another application process acting as Server.

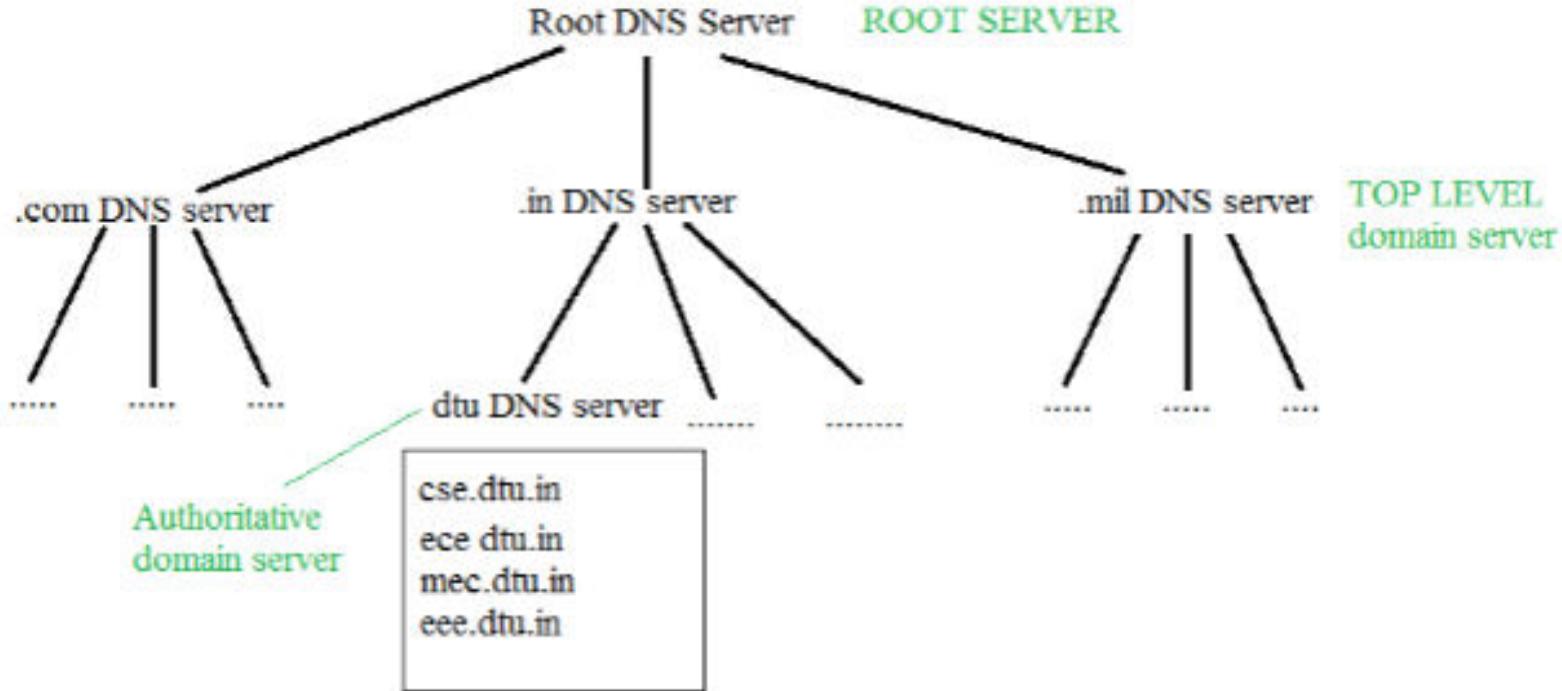
Communication

- Two processes in client-server model can interact in various ways:
- Sockets
- Remote Procedure Calls (RPC)

Domain Name System (DNS)

- DNS is a host name to IP address translation service. DNS is a distributed database implemented in a hierarchy of name servers. It is an application layer protocol for message exchange between clients and servers.
- DNS Basics
- Requirement
- Domain:
 - ❖ Generic domain
 - ❖ Country domain
 - ❖ Inverse domain

Organization of Domain



DNS record
Namespace
Name server

Name to Address Resolution

A host wants the IP address of cse.dtu.in



Hierarchy of Name Servers

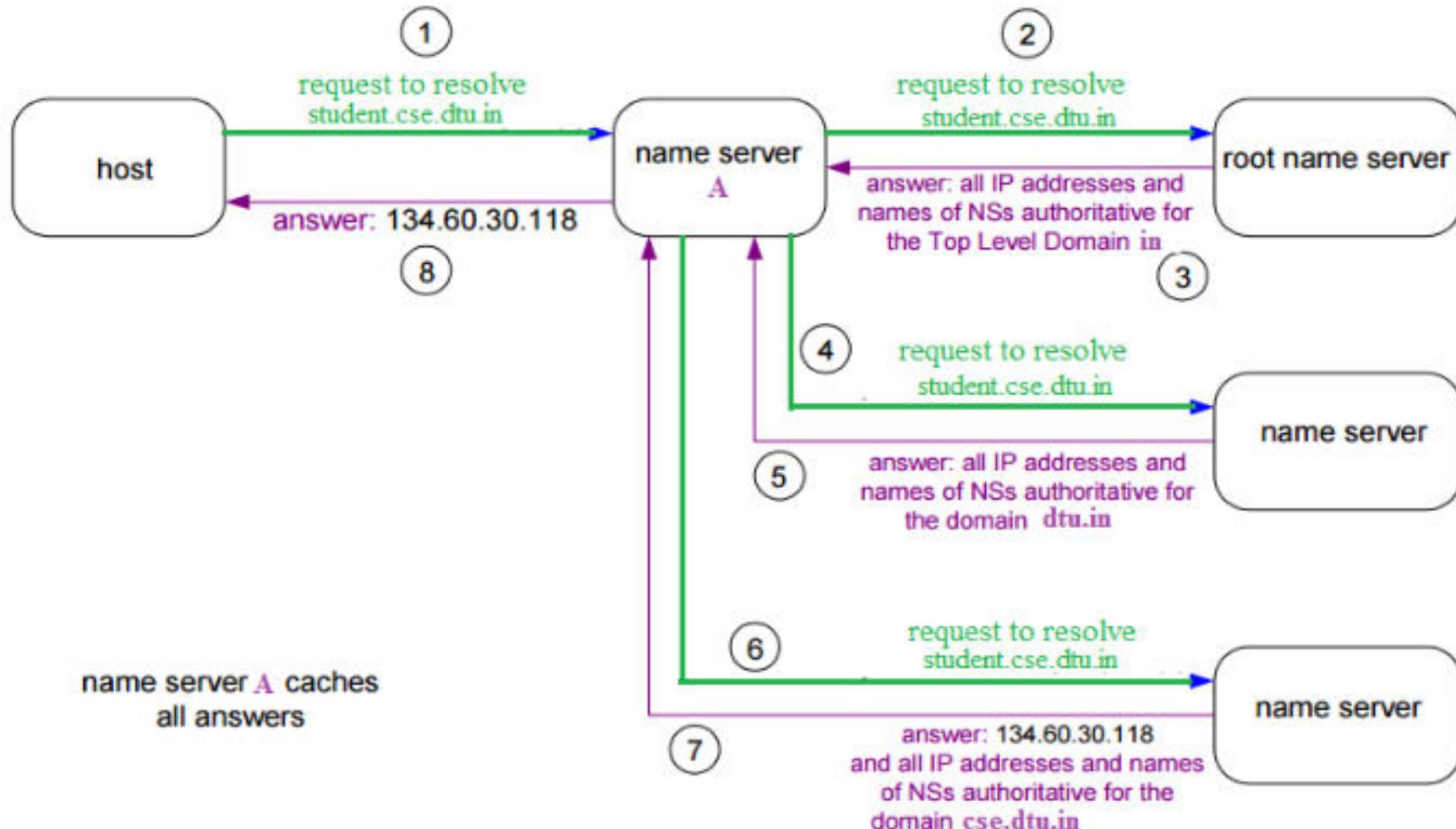
- Root name servers
- Top level server
- Authoritative name servers

- ❑ How DNS Servers Work
- ❑ DNS servers and IP addresses
- ❑ The DNS Lookup Process

Steps in a DNS lookup:

- A user enters a domain name
- DNS recursive resolver, sends a query to the root DNS nameserver (.)
- Root server returns to the resolver the address of the top-level domain
- Resolver then sends the information request to the Top-Level Domain server
- TLD name server responds to the resolver with the targeted IP address of the domain's nameserver.
- NS recursive resolver sends the query to the domain's DNS server.
- Domain's DNS server then returns the IP address to the DNS resolver for the requested domain
- Finally, the DNS resolver returns the IP address of the requested domain to the requesting web browser. The browser sends the HTTPS request to the targeted IP address, and the server with that address returns the webpage, which renders in the user's browser.

Domain Name Server



Types of DNS Services

- Recursive DNS Server**
- Authoritative DNS Server**

Application Layer - Processes

Communicating

Process:

program running within a host

Client process:

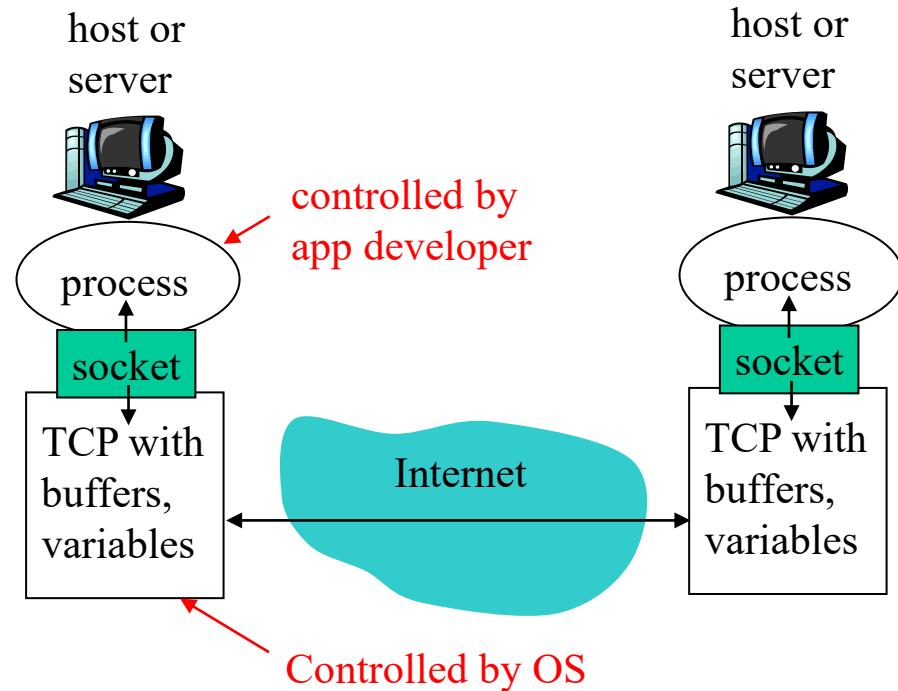
initiates communication

Server process:

waits to be contacted

process sends/receives messages to/from its **socket**

identifier includes both **IP address** and **port numbers** associated with process on host.



App-layer protocol defines

- Types of messages exchanged,
 - ❖ e.g., request, response
- Message syntax:
 - ❖ what fields in messages & how fields are delineated
- Message semantics
 - ❖ meaning of information in fields
- Rules for when and how processes send & respond to messages

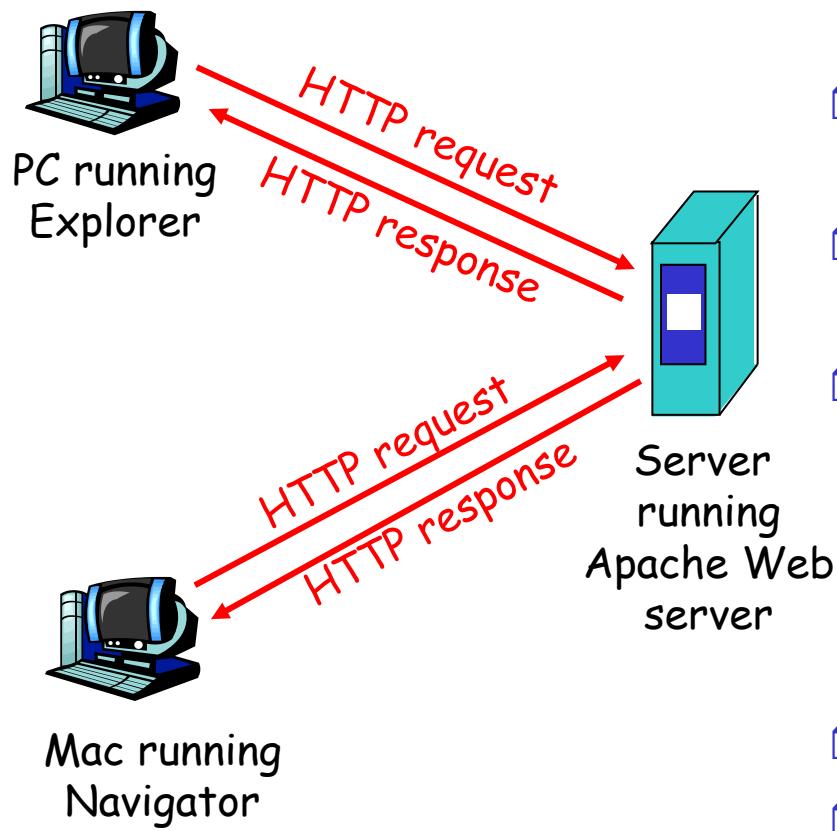
Public-domain protocols:

- ◆ defined in RFCs
- ◆ allows for interoperability
- ◆ e.g., HTTP, SMTP

Proprietary protocols:

- ◆ e.g., Skype

HTTP (Hypertext Transfer Protocol)



- Web page consists of **base HTML**-file which includes several referenced **objects**
- Each object is addressable by a Uniform Resource Locator (**URL**)
- HTTP is used as the webpage application layer protocol
- client/server model
 - ❖ **client**: browser that requests, receives, "displays" Web objects
 - ❖ **server**: Web server sends objects in response to requests
- uses TCP
- assures inter-operability

HTTP connections

Nonpersistent HTTP

- At most one object is sent over a TCP connection.

Persistent HTTP

- Multiple objects can be sent over single TCP connection between client and server.

Non-Persistent HTTP: Response time

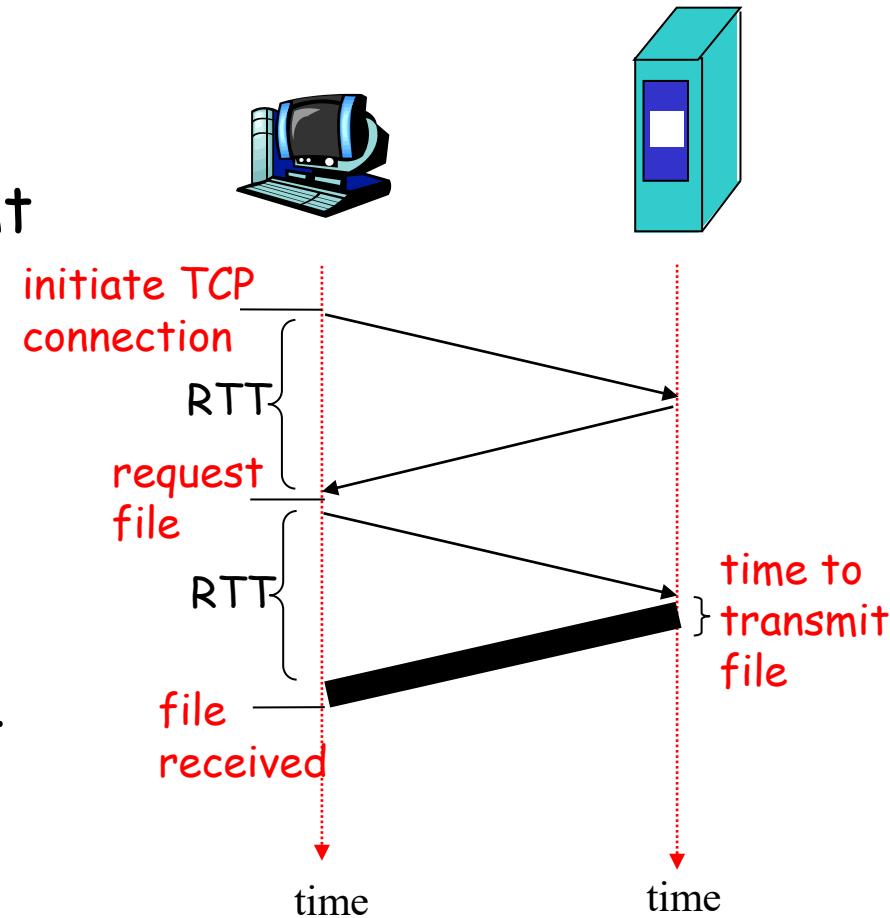
Definition of Round Trip

Time: time for a small packet to travel from client to server and back.

Response time:

- one RTT to initiate TCP connection
- one RTT for HTTP request and first few bytes of HTTP response to return
- file transmission time

$$\text{total} = 2\text{RTT} + \text{transmit time}$$



Persistent HTTP

Nonpersistent HTTP issues:

- requires 2 RTTs per object
- OS overhead for each TCP connection
- browsers often open parallel TCP connections to fetch referenced objects

Persistent HTTP

- server leaves connection open after sending response
- subsequent HTTP messages between same client/server sent over open connection
- client sends requests as soon as it encounters a referenced object
- as little as one RTT for all the referenced objects

Cookies: Keeping state

What cookies can bring:

- shopping carts
- recommendations
- user session state (Web e-mail)

aside

Cookies and privacy:

- cookies permit sites to learn a lot about you
- you may supply name and e-mail to sites

How to keep "state":

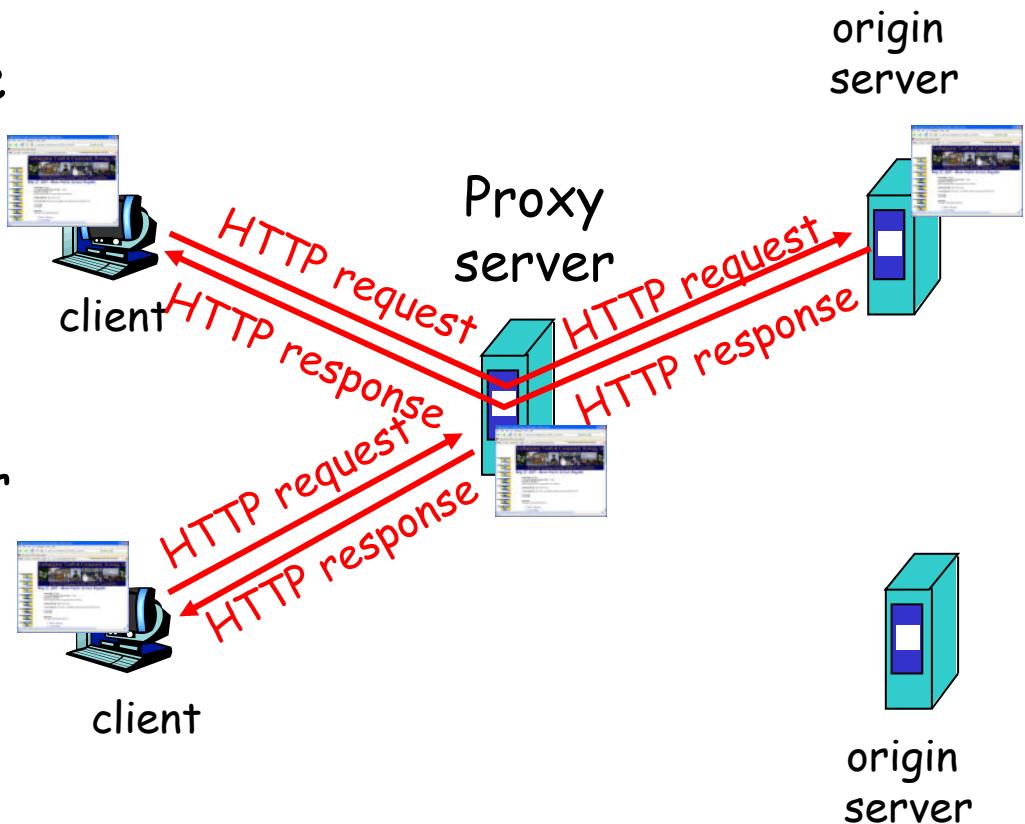
- protocol endpoints: maintain state at sender/receiver over multiple transactions
- cookies: http messages carry state

Web caches (proxy server)

Goal: satisfy client request without involving origin server

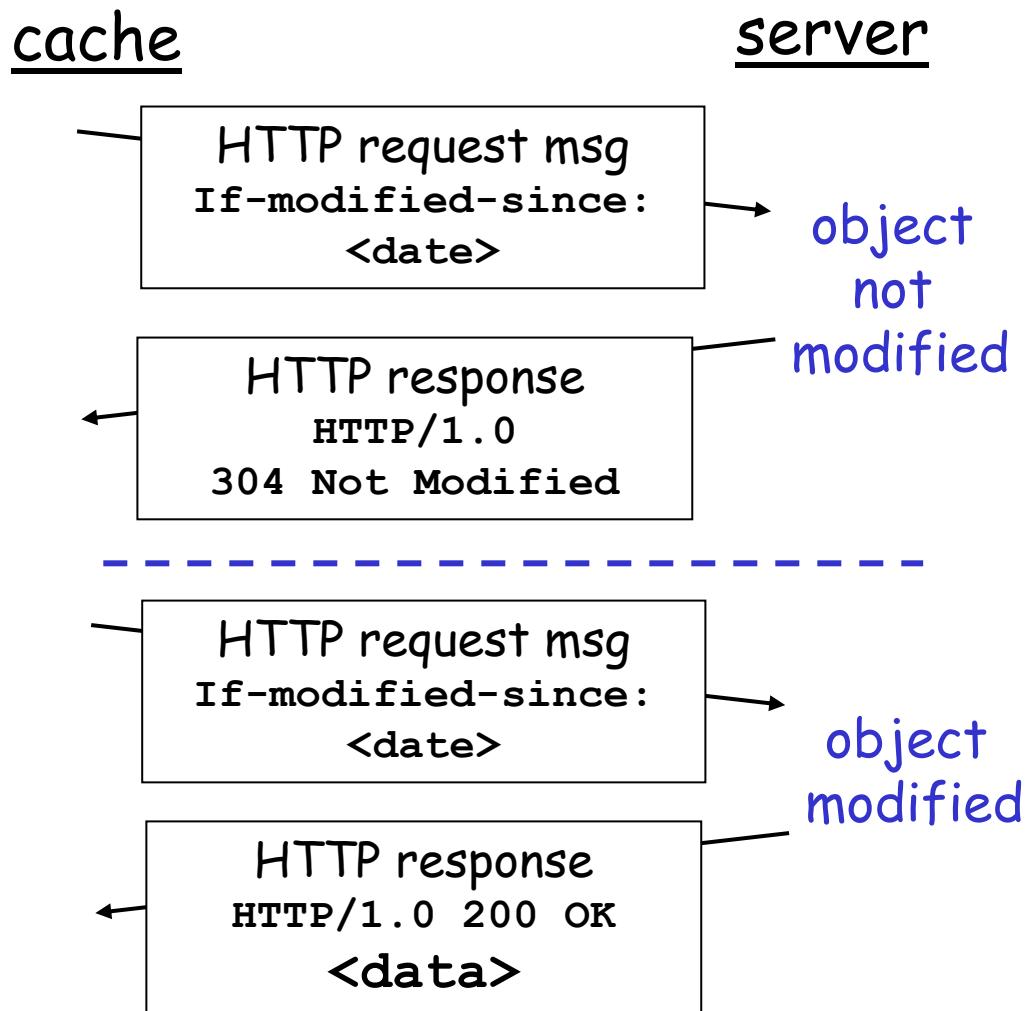
- user sets browser:
Web accesses via cache
- browser sends all HTTP
requests to cache

- Why Web caching?
 - ❖ reduce response time for
client request
 - ❖ reduce traffic on an
institution's access link.
 - ❖ enables "poor" content
providers to effectively
deliver content

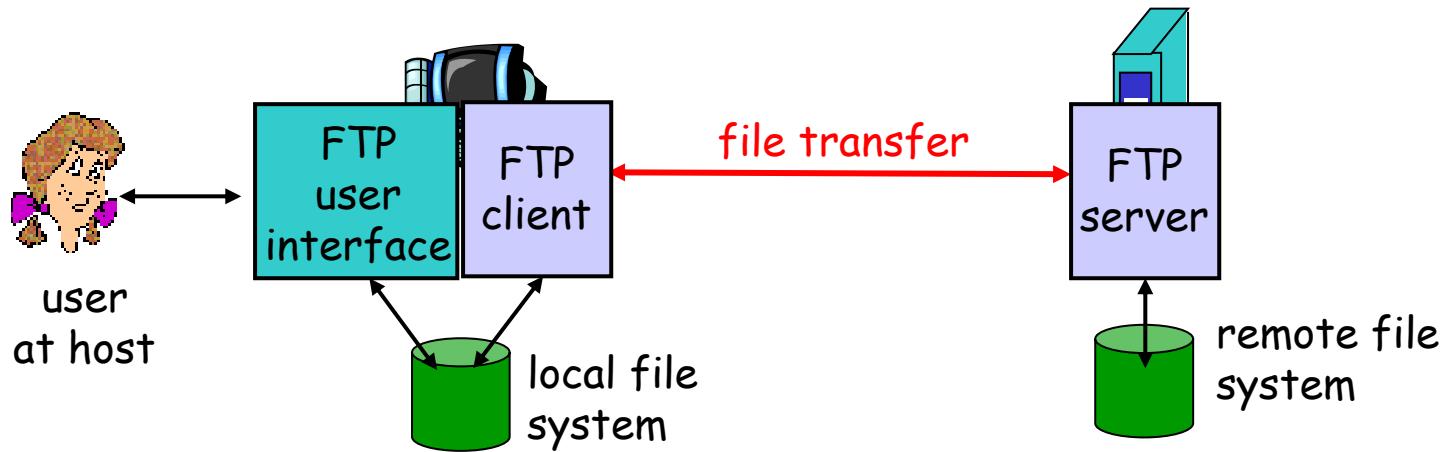


Conditional GET

- **Goal:** don't send object if cache has up-to-date cached version
- **cache:** specify date of cached copy in HTTP request
If-modified-since: <date>
- **server:** response contains no object if cached copy is up-to-date:
HTTP/1.0 304 Not Modified



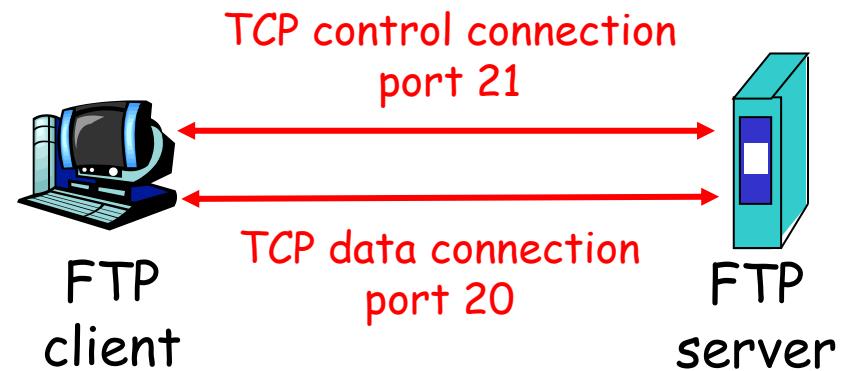
FTP: the file transfer protocol



- transfer file to/from remote host
- client/server model
 - ❖ *client*: side that initiates transfer (either to/from remote)
 - ❖ *server*: remote host

FTP: separate control, data connections

- FTP client contacts FTP server at port 21
- client authorized over control connection
- client browses remote directory by sending commands over control connection.
- when server receives file transfer command, server opens 2nd TCP connection (for file) to client
- after transferring one file, server closes data connection.
- server opens another TCP data connection to transfer another file.
- FTP server maintains "state": current directory, earlier authentication



FTP issues

- Multiple connections are used
 - ❖ for each directory listing and file transmission
- No integrity check at receiver
- Messages are sent in clear text
 - ❖ including **Passwords** and file contents
 - ❖ can be sniffed by eavesdroppers
- Solution
 - ❖ Secure FTP (SSH FTP)
 - allows a range of operations on remote files
 - ❖ FTPS (FTP over Secure Sockets Layer (SSL))
 - ❖ Transport Layer Security (TLS) encryption

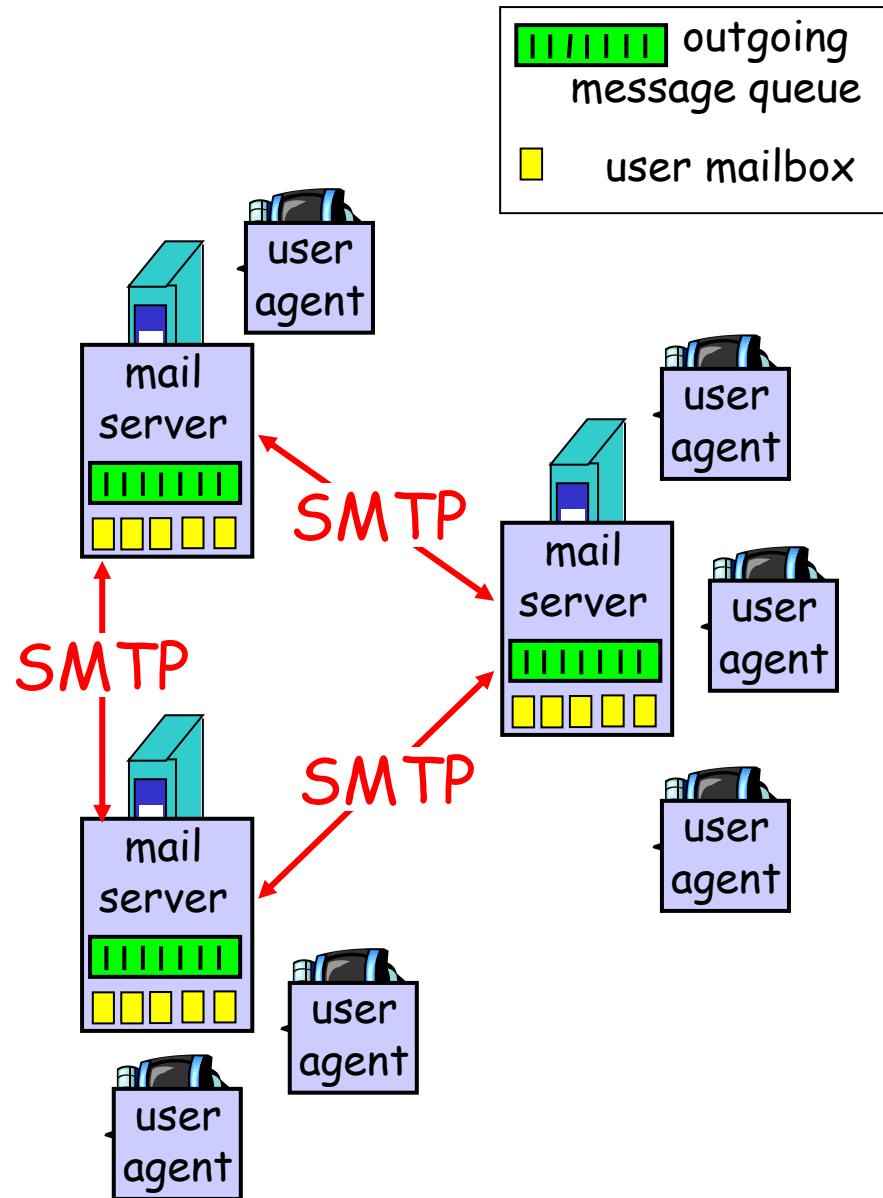
Electronic Mail

Three major components:

- user agents
- mail servers
- simple mail transfer protocol: SMTP

User Agent

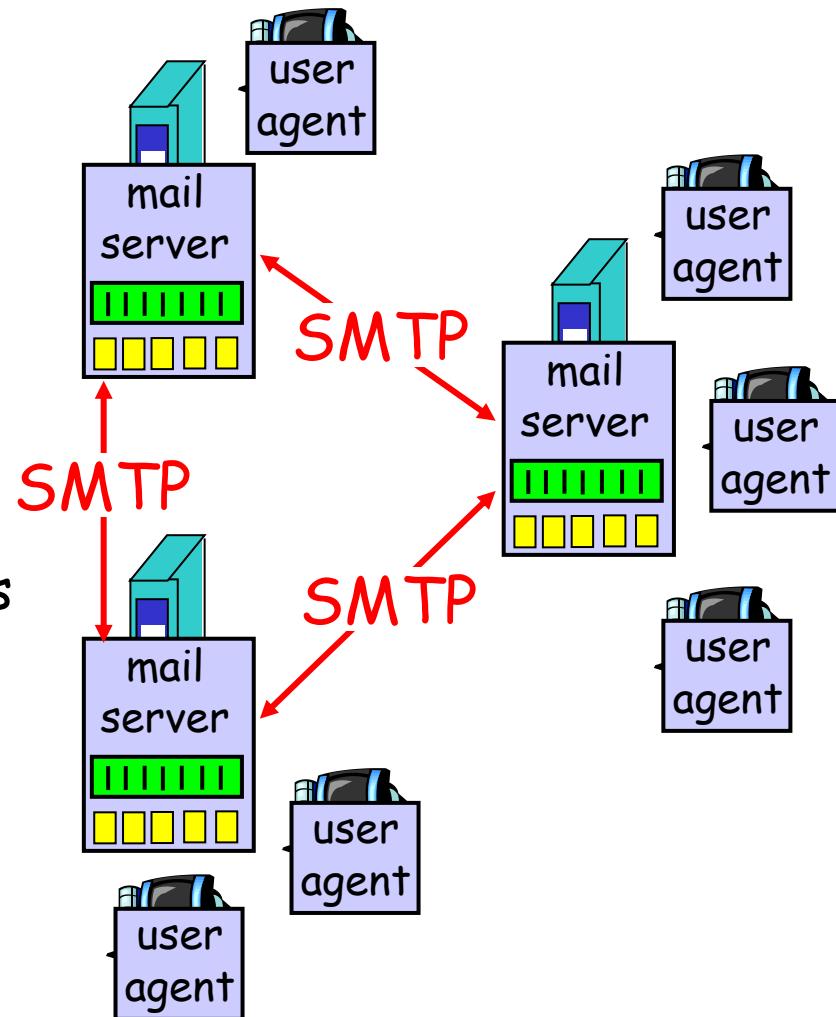
- "mail reader"
- composing, editing, reading mail messages
- e.g., Eudora, Outlook, elm, Mozilla Thunderbird
- outgoing, incoming messages stored on server



Electronic Mail: mail servers

Mail Servers

- **mailbox** contains incoming messages for user
- **message queue** of outgoing (to be sent) mail messages
- **SMTP protocol** between mail servers to send email messages
 - ❖ client: sending mail server
 - ❖ "server": receiving mail server

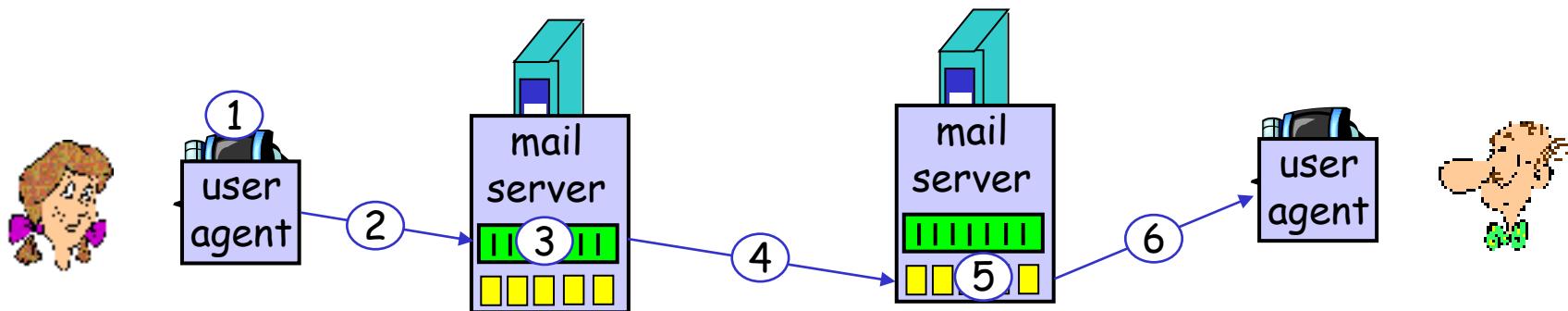


Electronic Mail: SMTP [RFC 2821]

- uses TCP to reliably transfer email message from client to server (port 25)
- direct transfer: sending server to receiving server
- three phases of transfer
 - ❖ handshaking (greeting)
 - ❖ transfer of messages
 - ❖ closure
- command/response interaction
 - ❖ commands: ASCII text
 - ❖ response: status code and phrase
- messages must be in 7-bit ASCII

Scenario: Alice sends message to Bob

- 1) Alice uses User Agent (UA) to compose message and send to bob@someschool.edu
- 2) Alice's UA sends message to her mail server; message placed in message queue
- 3) Client side of SMTP opens TCP connection with Bob's mail server
- 4) SMTP client sends Alice's message over the TCP connection
- 5) Bob's mail server places the message in Bob's mailbox
- 6) Bob invokes his user agent to read message



SMTP

- SMTP uses persistent connections
- SMTP requires message (header & body) to be in 7-bit ASCII
- SMTP server uses CRLF.CRLF to determine end of message

Mail message format

SMTP: protocol for exchanging email msgs

RFC 822: standard for text message format:

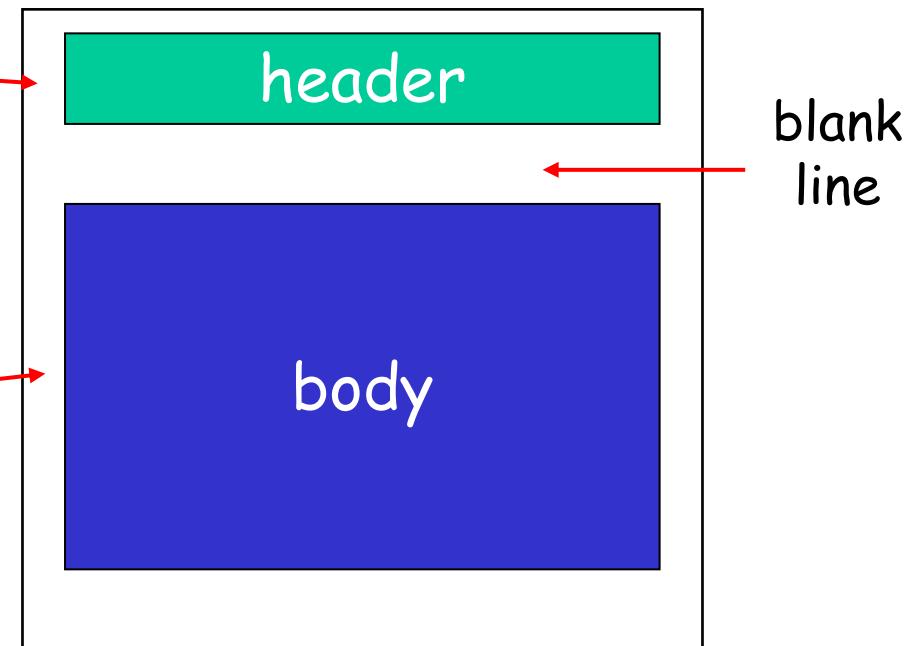
- header lines, e.g.,

- ❖ To:
- ❖ From:
- ❖ Subject:

different from SMTP commands!

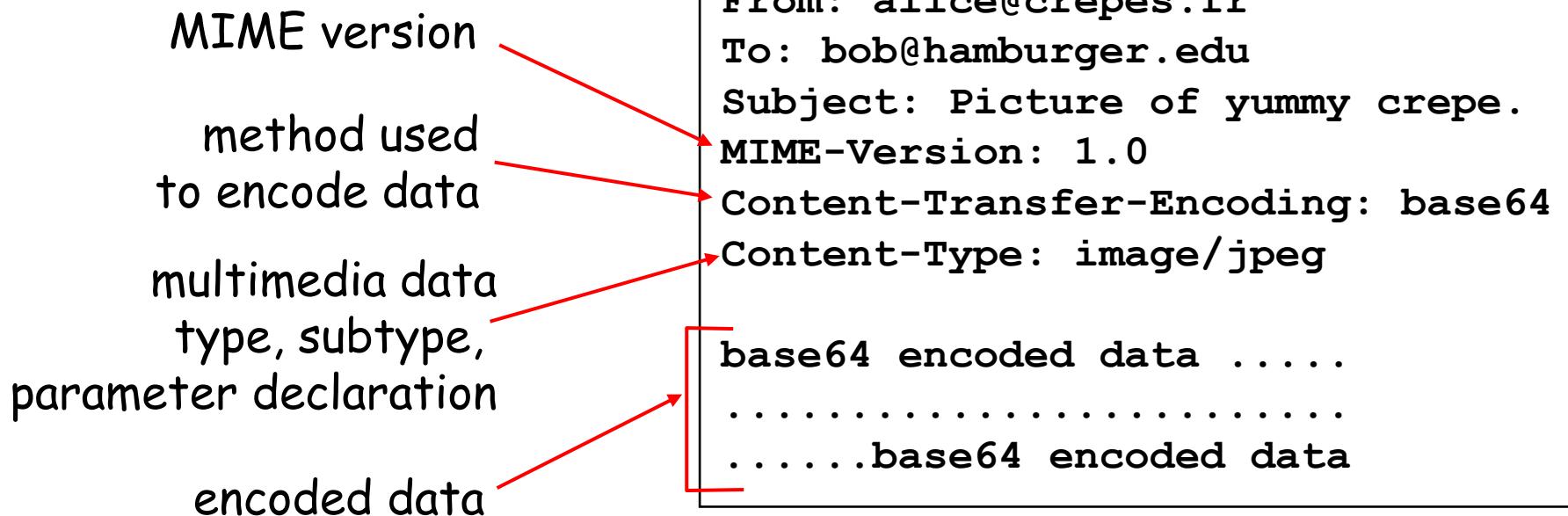
- body

- ❖ the "message",
ASCII characters only

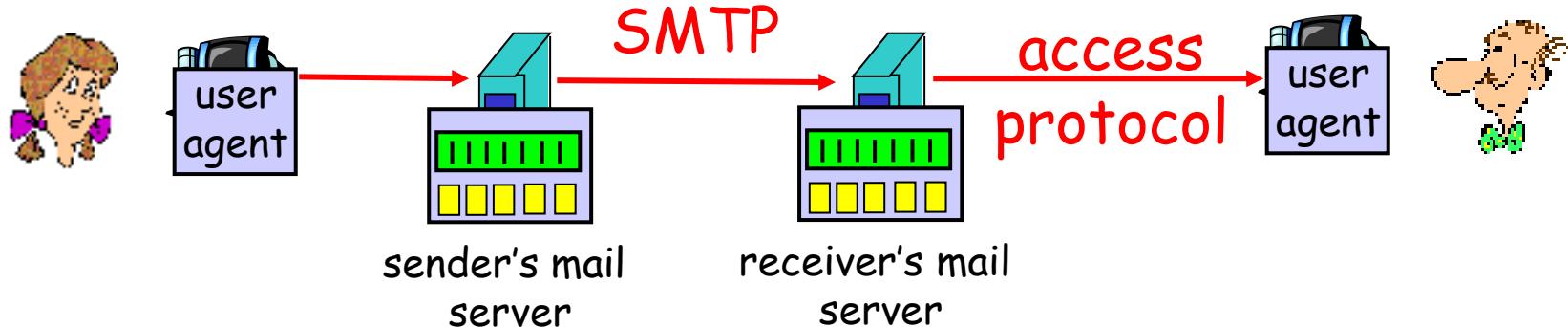


Message format: multimedia extensions

- ❑ MIME: multimedia mail extension, RFC 2045, 2056
- ❑ additional lines in msg header declare MIME content type



Mail access protocols



- SMTP: delivery/storage to receiver's server
- Mail access protocol: retrieval from server
 - ❖ POP: Post Office Protocol [RFC 1939]
 - authorization (agent <-->server) and download
 - ❖ IMAP: Internet Mail Access Protocol [RFC 1730]
 - more features (more complex)
 - manipulation of stored msgs on server
 - ❖ HTTP: gmail, Hotmail, Yahoo! Mail, etc.

DNS: Domain Name System

People: many identifiers:

- ❖ name, passport #

Internet hosts, routers:

- ❖ IP address (32 bit) - used for addressing datagrams
- ❖ "name", e.g., www.yahoo.com - used by humans

Domain Name System:

- ❑ *distributed database* implemented in hierarchy of many *name servers*
- ❑ *application-layer protocol* host, routers, name servers to communicate to *resolve* names (address/name translation)
 - ❖ note: core Internet function, implemented as application-layer protocol
 - ❖ complexity at network's "edge"

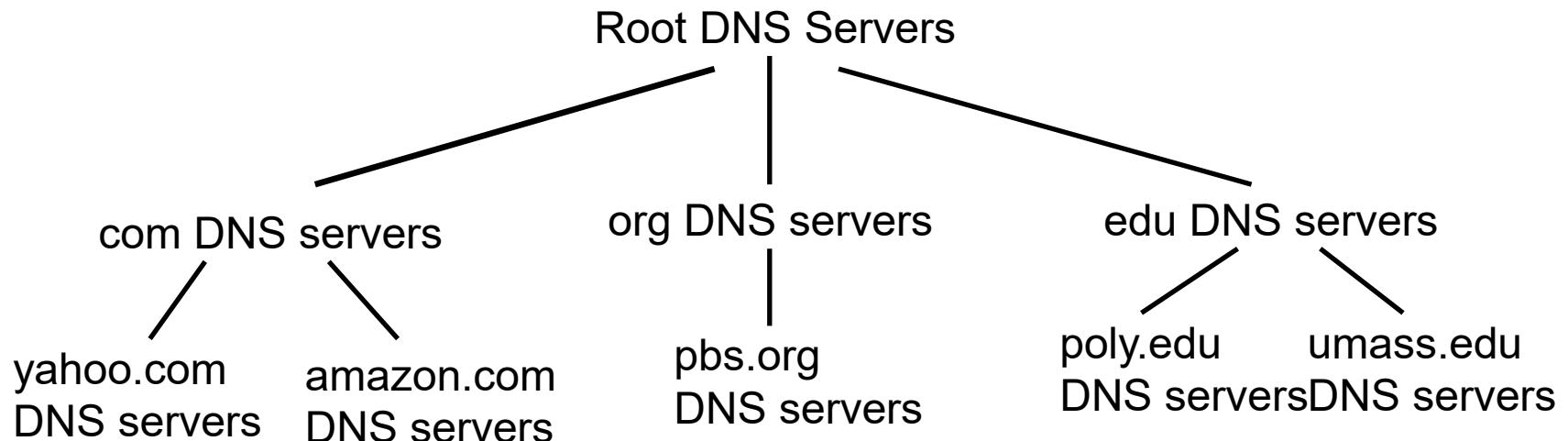
DNS services

- hostname to IP address translation
- host aliasing
 - ❖ Canonical, alias names
- load distribution
 - ❖ replicated Web servers: set of IP addresses for one canonical name

Why not centralize DNS?

- single point of failure
- traffic volume
- distant centralized database
- maintenance issues

Distributed, Hierarchical Database



Client wants IP for www.amazon.com; 1st approx:

- client queries a **root server** to find **com DNS server**
- client queries **com DNS server** to get **amazon.com DNS server**
- client queries **amazon.com DNS server** to get **IP address** for **www.amazon.com**