APS College of Engineering,

(Affiliated to Visvesvaraya Technological University and Approved by AICTE, NAAC Accredited)

Somanahalli, Kanakapura Main Road, Bengaluru-560116

Department of Electronics and Communication Engineering

Subject Name: Computer Communication Networks

Subject Name: 21EC73

Semester : 5th

Academic Year: ODD 2023-2024

Module-1

Faculty Name: Dr. Naik D C
Assistant Professor,
Dept., of ECE.

Module-1

- Introduction: Data Communications, Network, Uses of Networks, Types of Networks,
- Reference Models: TCP/IP Model, The OSI Model, Comparison of the OSI and TCP/IP reference model.
- Introduction to Data-link Layer, Link-Layer Addressing.

NETWORKS

- A network is a set of devices (often referred to as nodes) connected by communication links.
- A node can be a computer, printer, or any other device capable of sending and/or receiving data generated by other nodes on the network. “Computer network” to mean a collection of autonomous computers interconnected by a single technology.
- Two computers are said to be interconnected if they are able to exchange information.
- The connection need not be via a copper wire; fibre optics, microwaves, infrared, and communication satellites can also be used.
- Networks come in many sizes, shapes and forms.
- They are usually connected together to make larger networks, with the Internet being the most well-known example of a network of networks.

USES OF COMPUTER NETWORKS

- Business Applications:
  - To distribute information throughout the company (resource sharing). sharing physical resources such as printers, and tape backup systems, is sharing information
  - Communication medium among employees.email (electronic mail), which employees generally use for a great deal of daily communication.
  - Telephone calls between employees may be carried by the computer network instead of by the phone company. This technology is called IP telephony or Voice over IP (VoIP) when Internet technology is used.

- **Home Applications**
  - Peer-to-peer communication,
  - Person-to-person communication,
  - Electronic commerce,
  - Entertainment.(game playing,)
- **Mobile Users**
  - Text messaging or texting
  - Smart phones,
  - GPS (Global Positioning System),
  - m-commerce,
  - NFC (Near Field Communication)
- **Social Issues**
  - With the good comes the bad, as this new-found freedom brings with it many unsolved social, political, and ethical issues.

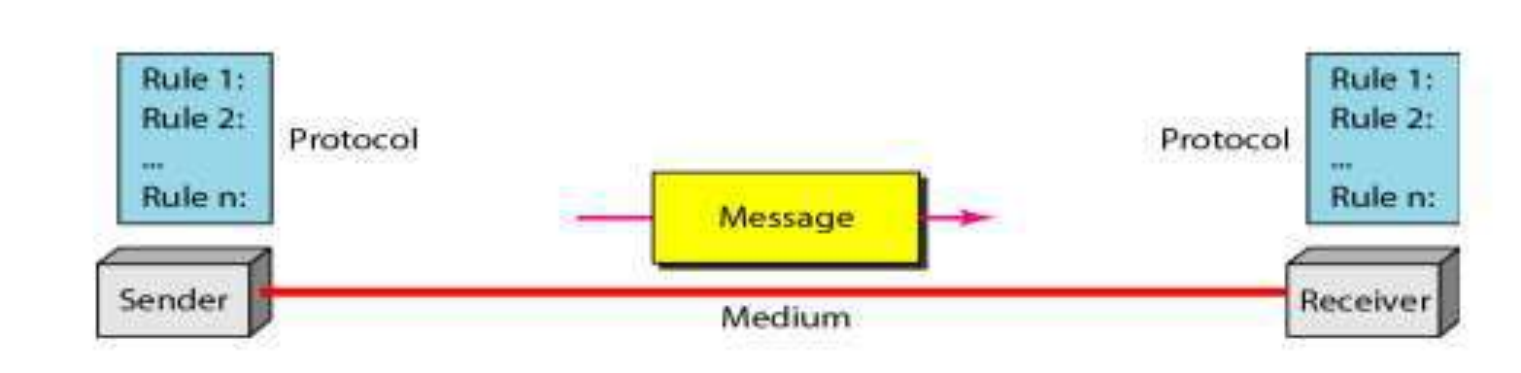
Data communications

- When we communicate, we are sharing information. This sharing can be local or remote.
- Between individuals, local communication usually occurs face to face while remote communication takes place over distance.
- The term telecommunication, which includes telephony, telegraphy, and television, means communication at a distance .
- The word data refers to information presented in whatever form is agreed upon by the parties creating and using the data.
- **Data communications** are the exchange of data between two devices via some form of transmission medium such as a wire cable. For data communications to occur, the communicating devices must be part of a communication system made up of a combination of hardware (physical equipment) and software (programs).

The effectiveness of a data communications system depends on,

1. **Delivery**-The system must deliver data to the correct destination. Data must be received by the intended device or user and only by that device or user.
2. **Accuracy**- The system must deliver the data accurately. Data that have been altered in transmission and left uncorrected are unusable.
3. **Timeliness**. The system must deliver data in a timely manner. Data delivered late are useless. In the case of video and audio, timely delivery means delivering data as they are produced, in the same order that they are produced, and without significant delay. This kind of delivery is called real-time transmission.
4. **Jitter**. Jitter refers to the variation in the packet arrival time. It is the uneven delay in the delivery of audio or video packets. For example, let us assume that video packets are sent every 30 ms. If some of the packets arrive with 30-msdelay and others with 40-msdelay, an uneven quality in the video is the result.

A data communications system has **five components**



1. **Message-**. The message is the information (data) to be communicated. Popular forms of information include text, numbers, pictures, audio, and video.
2. **Sender-**. The sender is the device that sends the data message. It can be a computer, workstation, telephone handset, video camera, and so on.
3. **Receiver-**. The receiver is the device that receives the message. It can be a computer, workstation, telephone handset, television, and so on.
4. **Transmission medium-**. The transmission medium is the physical path by which a message travels from sender to receiver. Some examples of transmission media include twisted-pair wire, coaxial cable, fibre-optic cable, and radio waves.
5. **Protocol-.** A protocol is a set of rules that govern data communications. It represents an agreement between the communicating devices. Without a protocol, two devices may be connected but not communicating, just as a person speaking French cannot be understood by a person who speaks only Japanese.

Data Representation

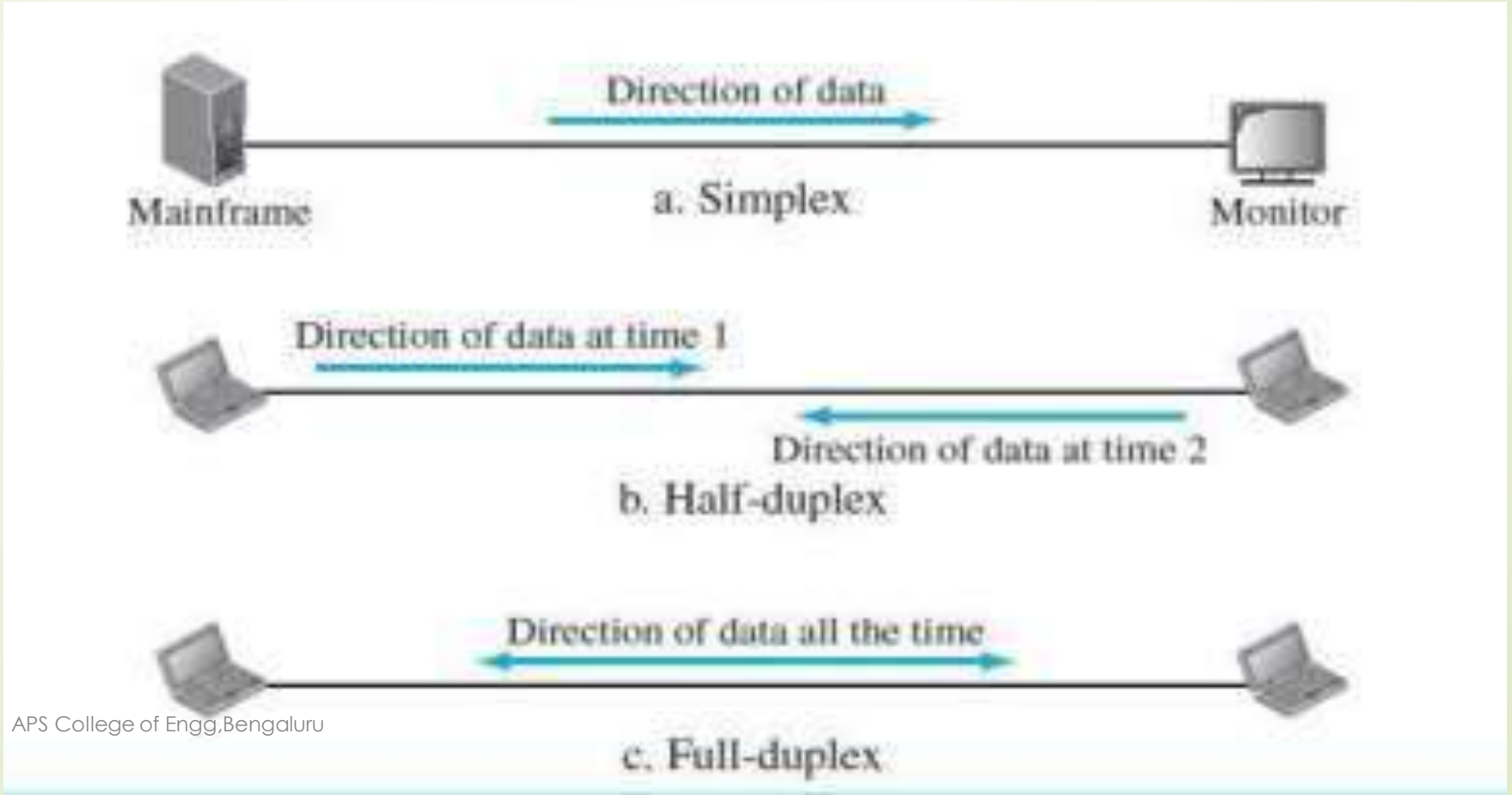
- **Text**-In data communications, text is represented as a bit pattern, a sequence of bits (0s or 1s). Different sets of bit patterns have been designed to represent text symbols. Each set is called a code, and the process of representing symbols is called coding.
- **Numbers-** are also represented by bit patterns. However, a code such as ASCII is not used to represent numbers; the number is directly converted to a binary number to simplify mathematical operations.
- **Images-** are also represented by bit patterns. In its simplest form, an image is composed of a matrix of pixels (picture elements), where each pixel is a small dot. The size of the pixel depends on the resolution.

- **Audio-** Audio refers to the recording or broadcasting of sound or music. Audio is by nature different from text, numbers, or images. It is continuous, not discrete. Even when we use a microphone to change voice or music to an electric signal, we create a continuous signal.
- **Video-** Video refers to the recording or broadcasting of a picture or movie.

  Video can either be produced as a continuous entity (e.g., by a TV camera), or it can be a combination of images, each a discrete entity, arranged to convey the idea of motion.

Data Flow

- Communication between two devices can be simplex, half-duplex, or full-duplex as shown in Figure



Simplex mode:

- The communication is unidirectional, as on a one-way street.
- Only one of the two devices on a link can transmit; the other can only receives.
- example-Keyboards and traditional monitors are examples of simplex devices. The keyboard can only introduce input; the monitor can only accept output.

Half-Duplex:

- In half-duplex mode, each station can both transmit and receive, but not at the same time.
- When one device is sending, the other can only receive, and vice versa (see Figure b). The half-duplex mode is like a one-lane road with traffic allowed in both directions.
- example-Walkie-talkies and radios are both half-duplex systems.

Full-Duplex:

- In full-duplex mode (also called duplex), both stations can transmit and receive simultaneously.
- The full-duplex mode is like a two-way street with traffic flowing in both directions at the same time.
- In full-duplex mode, signals going in one direction share the capacity of the link with signals going in the other direction.

NETWORKS

- A network is the interconnection of a set of devices capable of communication. a device can be a host such as a large computer, desktop, laptop, workstation, cellular phone, or security system.
- A device can also be a connecting device such as a router, which connects the network to other networks, a switch, which connects devices together, a modem (modulator-demodulator), which changes the form of data, and so on.
- These devices in a network are connected using wired or wireless transmission media such as cable or air.

Network Criteria

- A network must be able to meet a certain number of criteria. The most important of these are performance, reliability, and security.

Performance:

- Performance can be measured in many ways, including transit time and response time.
- Transit time is the amount of time required for a message to travel from one device to another.
- Response time is the elapsed time between an inquiry and a response.
- The performance of a network depends on a number of factors, including the number of users, the type of transmission medium, the capabilities of the connected hardware, and the efficiency of the software
- Performance is often evaluated by two networking metrics: throughput and delay.

Reliability

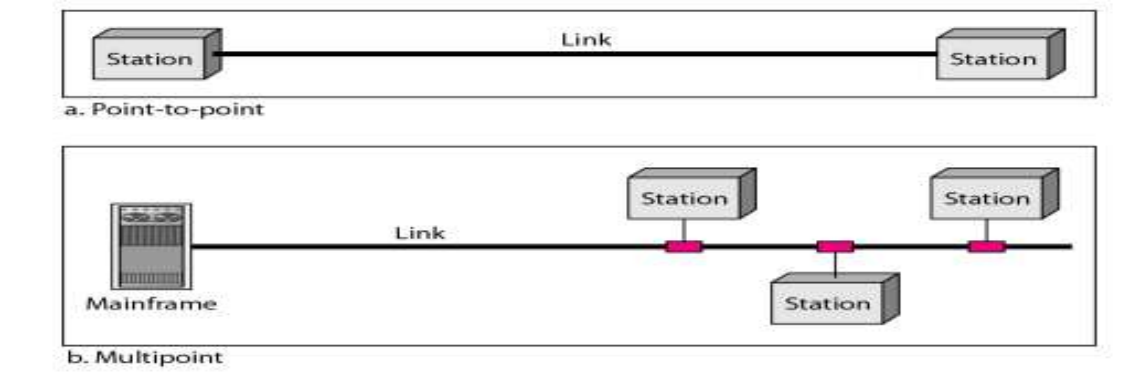
- Network reliability is measured by the frequency of failure, the time it takes a link to recover from a failure, and the network's robustness in a catastrophe.

Security

- Security issues include protecting data from unauthorized access, protecting data from damage and development, and implementing policies and procedures for recovery from breaches and data losses.

Type of Connection

- A network is two or more devices connected through links. A link is a communications pathway that transfers data from one device to another.



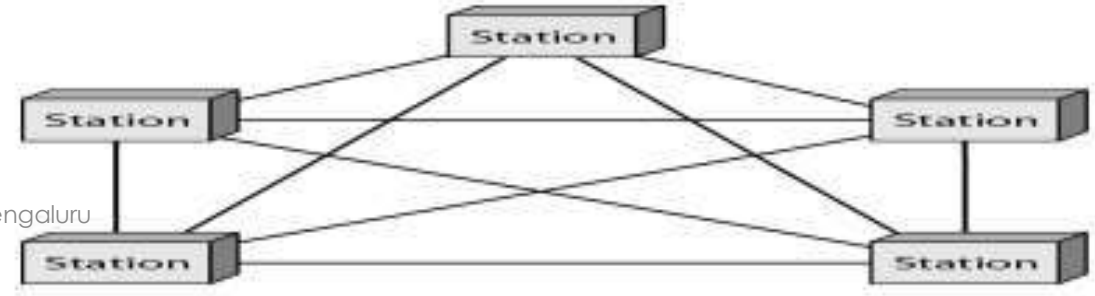
- **Point-to-Point**: A point-to-point connection provides a dedicated link between two devices. The entire capacity of the link is reserved for transmission between those two devices.
- **Multipoint**: A multipoint (also called multi-drop) connection is one in which more than two specific devices share a single link. In a multipoint environment, the capacity of the channel is shared, either spatially or temporally.
- If several devices can use the link simultaneously, it is a spatially shared connection. If users must take turns, it is a timeshared connection.

Physical Topology

- The term physical topology refers to the way in which a network is laid out physically.
- Two or more devices connect to a link; two or more links form a topology.
- The topology of a network is the geometric representation of the relationship of all the links and linking devices to one another.
- There are four basic topologies possible:
  - Mesh,
  - Star,
  - Bus, and
  - Ring.

MESH:

- A mesh topology is the one where every node is connected to every other node in the network.
- A mesh topology can be a full mesh topology or a partially connected mesh topology.
- In a full mesh topology, every computer in the network has a connection to each of the other computers in that network.
- The number of connections in this network can be calculated using the following formula (n is the number of computers in the network): n(n-1)/2
- In a partially connected mesh topology, at least two of the computers in the network have connections to multiple other computers in that network.



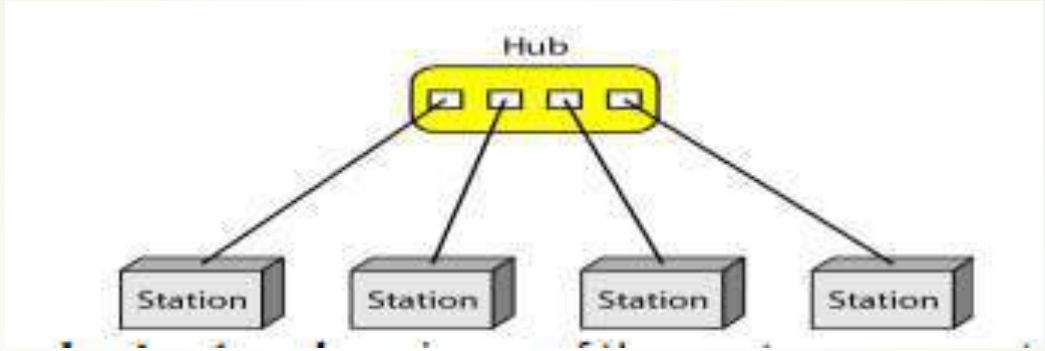
Advantages of a mesh topology

- Can handle high amounts of traffic, because multiple devices can transmit data simultaneously.
- A failure of one device does not cause a break in the network or transmission of data.
- Adding additional devices does not disrupt data transmission between other devices.

Disadvantages of a mesh topology.

- The cost to implement is higher than other network topologies, making it a less desirable option.
- Building and maintaining the topology is difficult and time consuming.
- The chance of redundant connections is high, which adds to the high costs and potential for reduced efficiency.

STAR:



- A star network, star topology is one of the most common network setups. In this configuration, every node connects to a central network device, like a hub, switch, or computer.
- The central network device acts as a server and the peripheral devices act as clients.

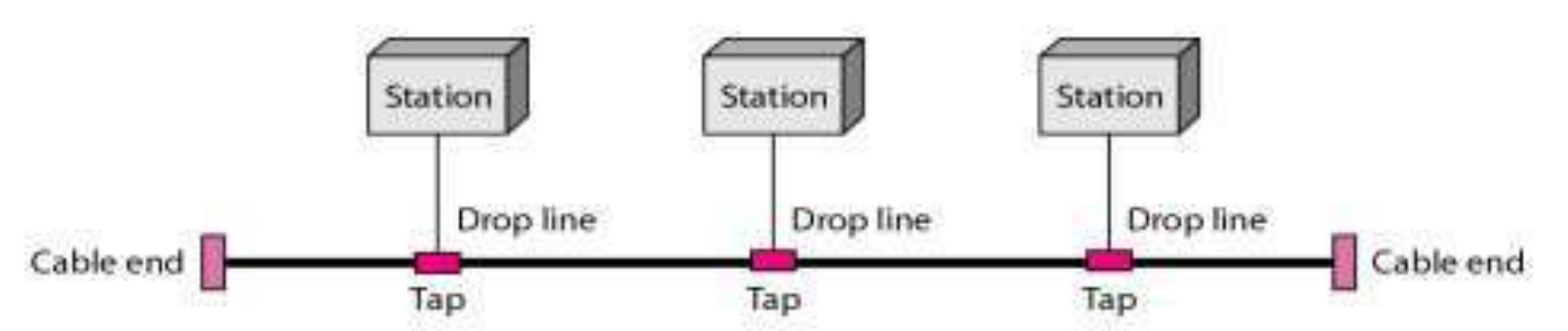
Advantages of star topology

- Centralized management of the network, through the use of the central computer, hub, or switch.
- Easy to add another computer to the network.
- If one computer on the network fails, the rest of the network continues to function normally.
- The star topology is used in local-area networks (LANs), High-speed LANs often use a star topology with a central hub.

Disadvantages of star topology

- Can have a higher cost to implement, especially when using a switch or router as the central network device.
- The central network device determines the performance and number of nodes the network can handle.
- If the central computer, hub, or switch fails, the entire network goes down and all computers are disconnected from the network

BUS:



- a line topology, a bus topology is a network setup in which each computer and network device are connected to a single cable or backbone.

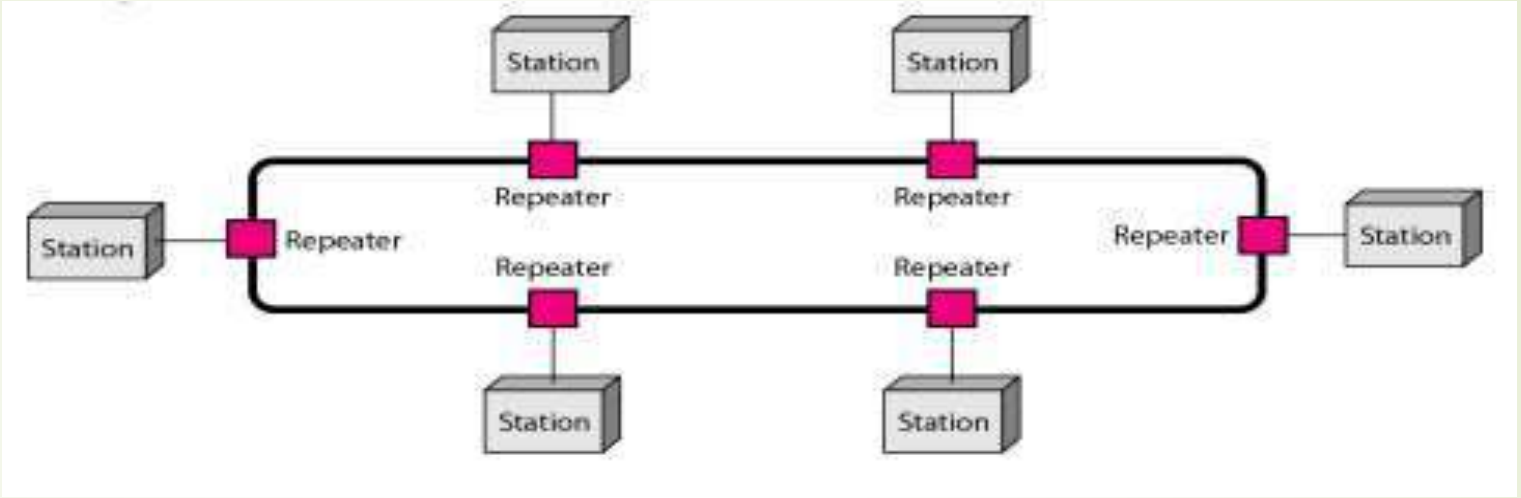
Advantages of bus topology

- It works well when you have a small network.
- It's the easiest network topology for connecting computers or peripherals in a linear fashion.

Disadvantages of bus topology

- It can be difficult to identify the problems if the whole network goes down.
- Bus topology is not great for large networks.

RING:



- A ring topology is a network configuration in which device connections create a circular data path.
- In a ring network, packets of data travel from one device to the next until they reach their destination.
- Most ring topologies allow packets to travel only in one direction, called a unidirectional ring network. Others permit data to move in either direction, called bidirectional.

Advantages of ring topology

- All data flows in one direction, reducing the chance of packet collisions.
- Data can transfer between workstations at high speeds.
- Additional workstations can be added without impacting performance of the network.

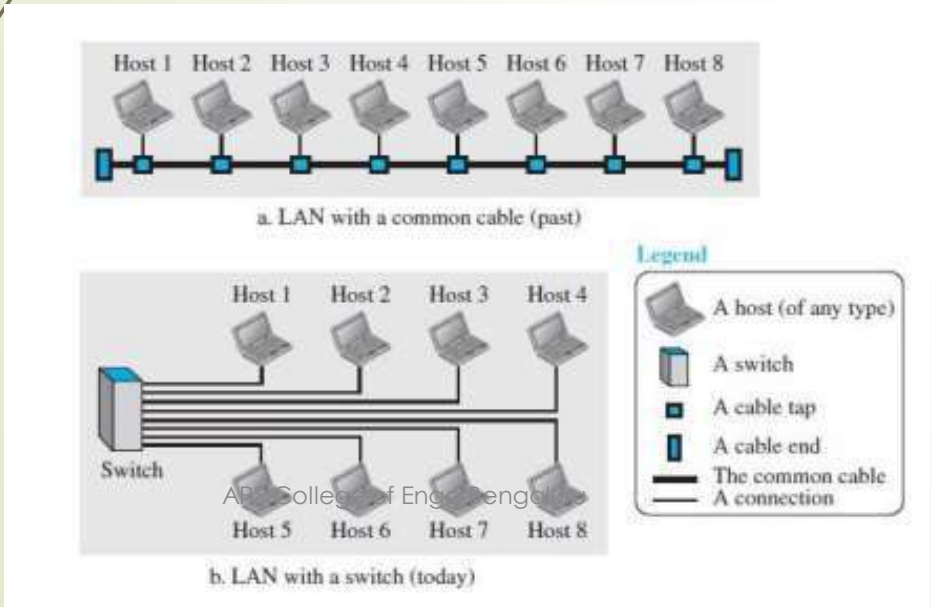
Disadvantages of ring topology

- All data being transferred over the network must pass through each workstation on the network, which can make it slower than a star topology.
- The entire network will be impacted if one workstation shuts down.

Types of Network based on size

- The types of network are classified based upon the size, the area it covers and its physical architecture.
- The three primary network categories are LAN, WAN and MAN.
- Each network differs in their characteristics such as distance, transmission speed, cables and cost.

Local Area Network(LAN)



- Group of interconnected computers within a small area. (room, building, campus).
- Two or more pc's can from a LAN to share files, folders, printers, applications and other devices.

MAN (Metropolitan Area Network)

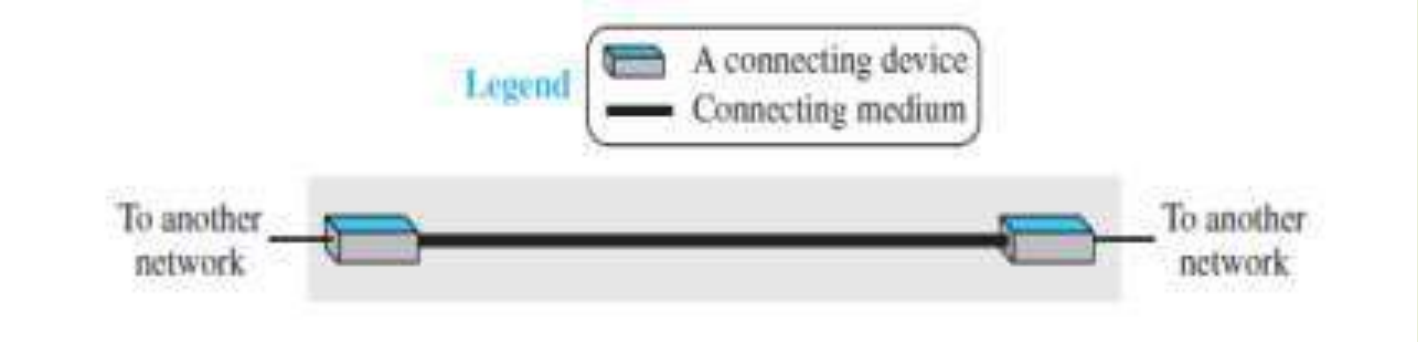
- Design to extend over a large area.
- Connecting number of LAN's to form larger network, so that resources can be shared.
- Networks can be up to 5 to 50 km
- Data transfer rate is low compare to LAN.

WAN (Wide Area Network)

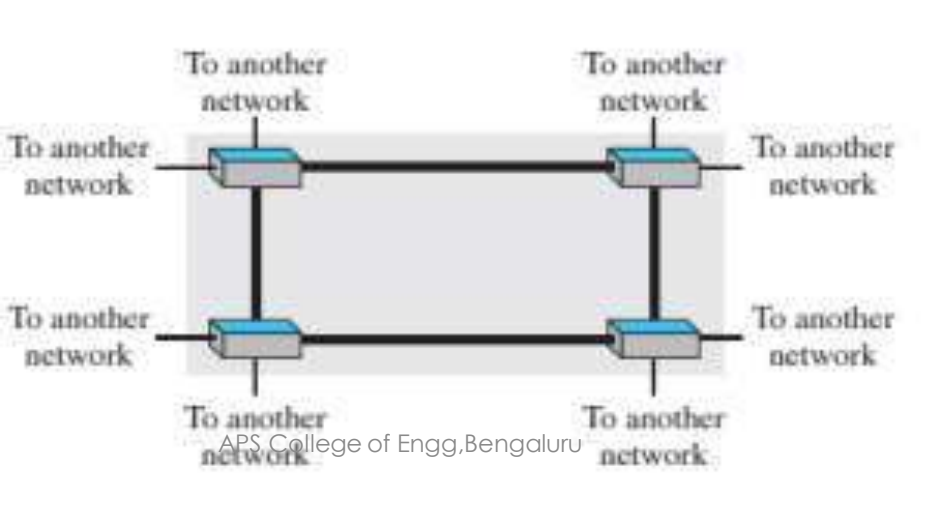
- A wide area network (WAN) is also an interconnection of devices capable of communication.
- WAN has a wider geographical range, spanning a town, a state, a country, or even the world
- Data transfer rate depends upon the ISP provider and varies over the location. Best example is the internet.
- We see two distinct examples of WANs today: point-to-point WANs and switched WANs.

Point-to-Point WAN

- A point-to-point WAN is a network that connects two communicating devices through a transmission media(cable or air).



Switched WAN



- A switched WAN is a network with more than two ends. A switched WAN, is used in the backbone of global communication today.
- We can say that a switched WAN is a combination of several point-to-point WANs that are connected by switches..

Internetwork

- Today, it is very rare to see a LAN or a WAN in isolation; they are connected to one another. When two or more networks are connected, they make an internetwork, or internet.

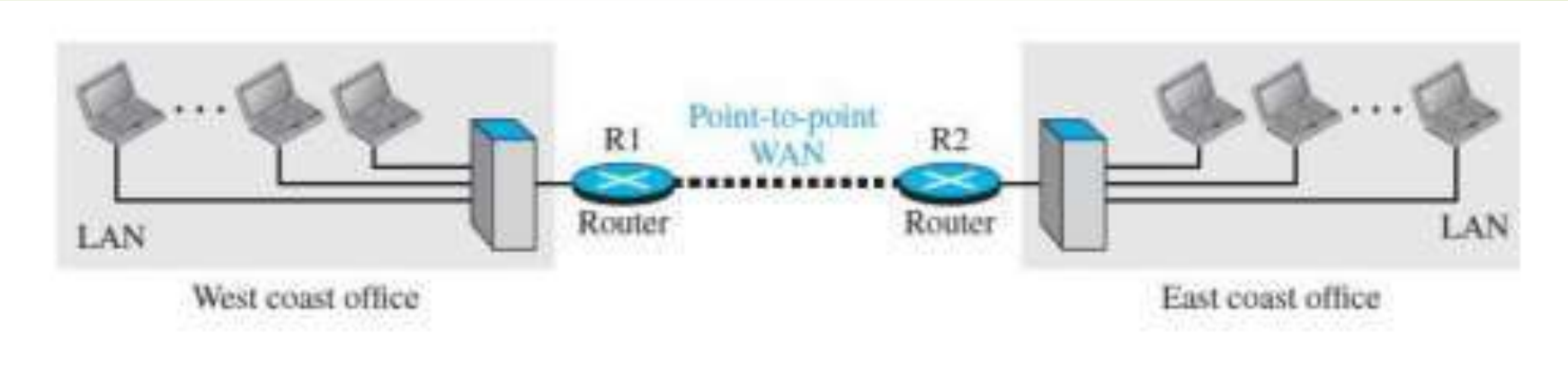


fig: An network made of two LAN and point-to-point dedicated WAN

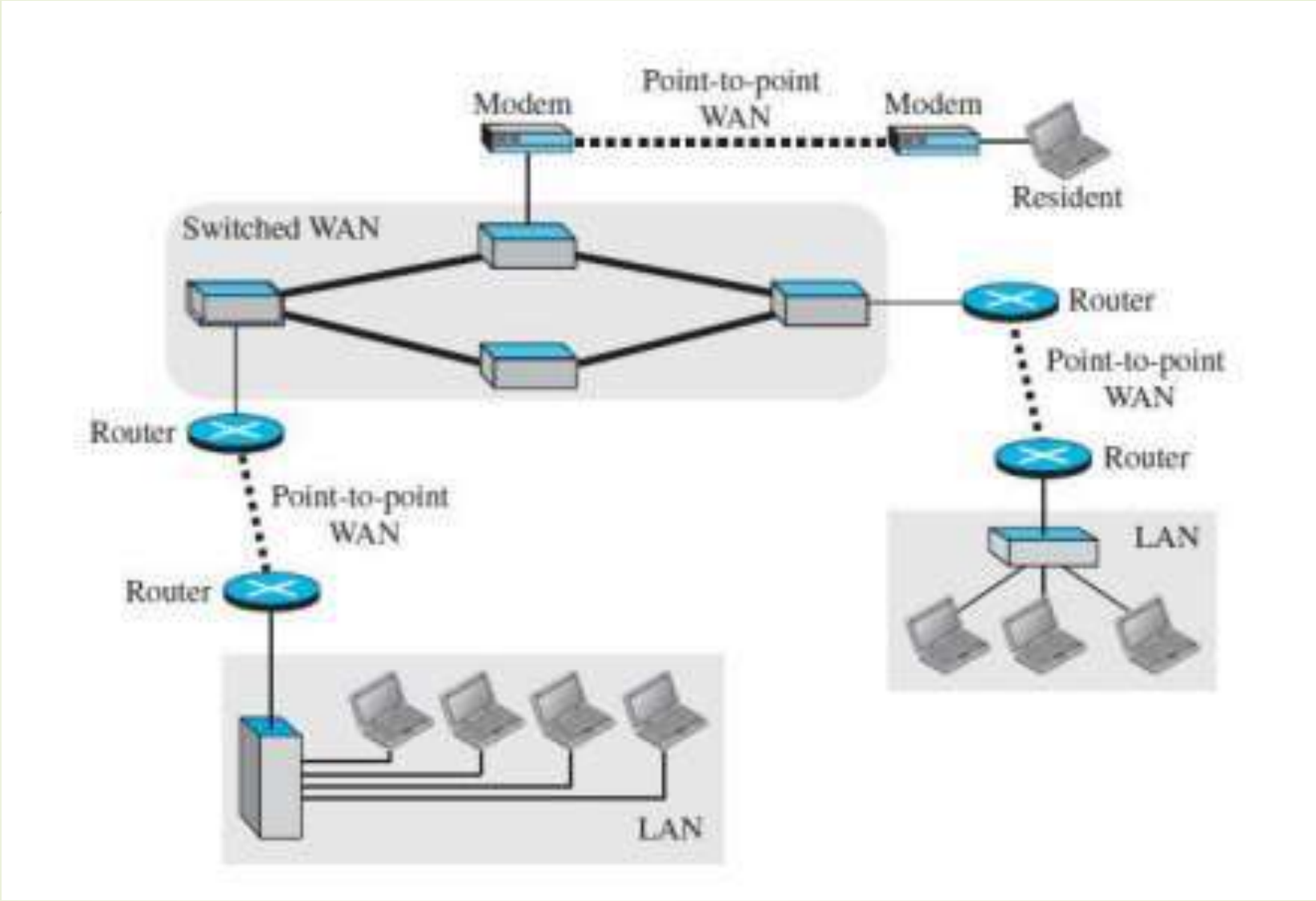


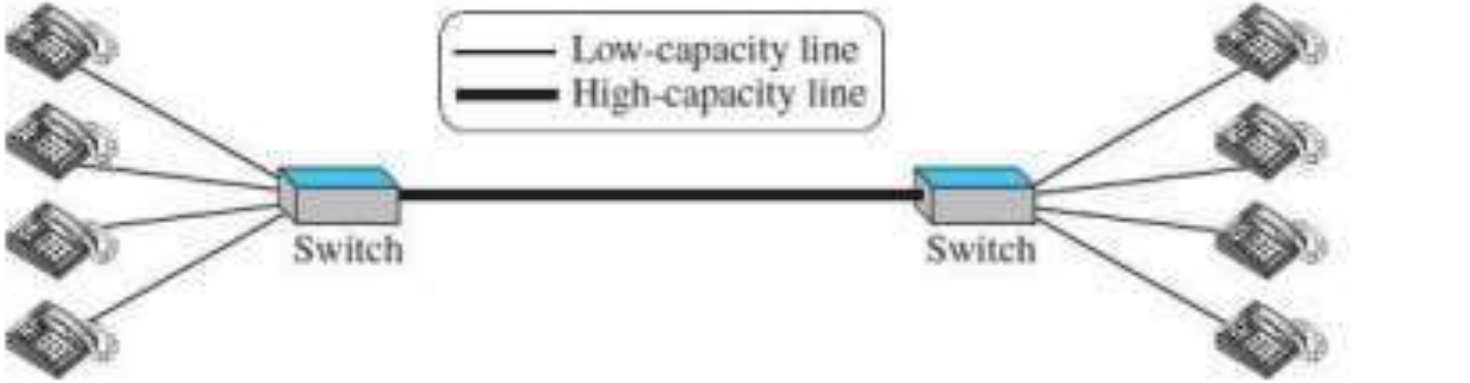
fig: A heterogeneous network made of four WANs and three LANs

Switching

- An internet is a switched network in which a switch connects at least two links together. A switch needs to forward data from a network to another network when required.
- The two most common types of switched networks are
  - Circuit-switched
  - Packet-switched networks.

Circuit-Switched Network

- In a circuit-switched network, a dedicated connection, called a circuit, is always available between the two end systems; the switch can only make it active or inactive.
- FIG shows a very simple switched network that connects four telephones to each end. We have used telephone sets instead of computers as an end system because circuit switching was very common in telephone networks in the past,



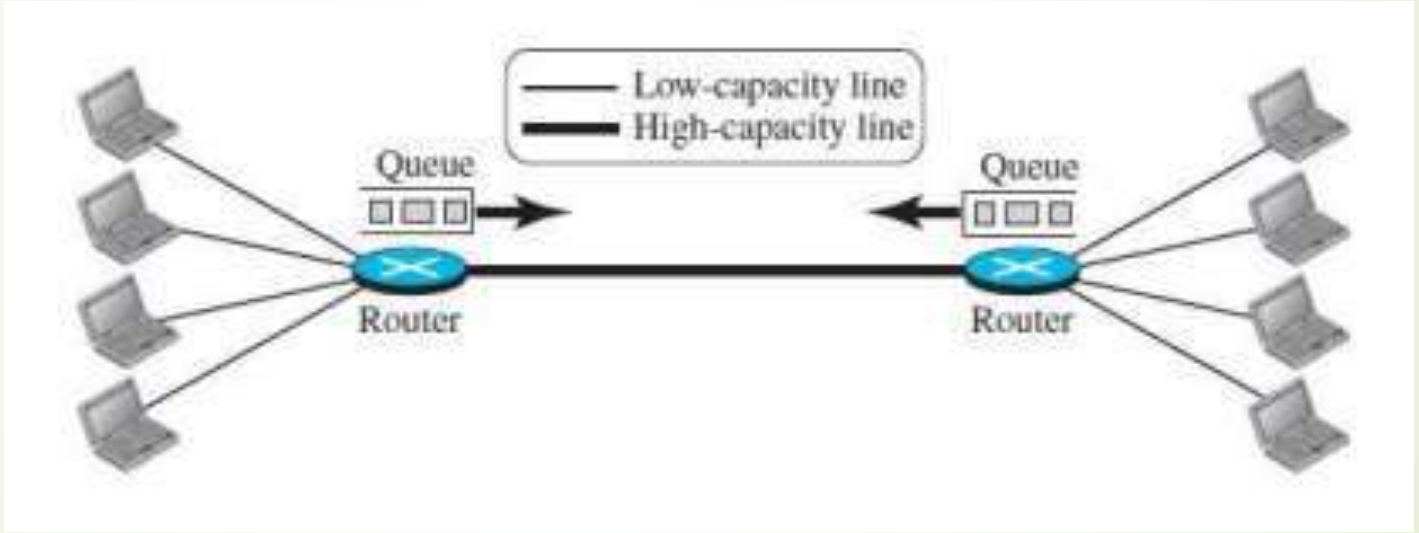
- The thick line connecting two switches is a high-capacity communication line that can handle four voice communications at the same time; the capacity can be shared between all pairs of telephone sets. The switches used in this example have forwarding tasks.

Let us look at two cases.

- In the **first case**, all telephone sets are busy; four people at one site are talking with four people at the other site; the capacity of the thick line is fully used.
- In the **second case**, only one telephone set at one side is connected to a telephone set at the other side; only one-fourth of the capacity of the thick line is used.
- This means that a circuit- switched network is efficient only when it is working at its full capacity; most of the time, it is inefficient because it is working at partial capacity.

Packet-Switched Network

- In a computer network, the communication between the two computers is done in blocks of data called packets.
- This allows switches to function for both storing and forwarding because a packet is an independent entity that can be stored and sent later.



- A router in a packet-switched network has a queue that can store and forward the packet.

Network Models

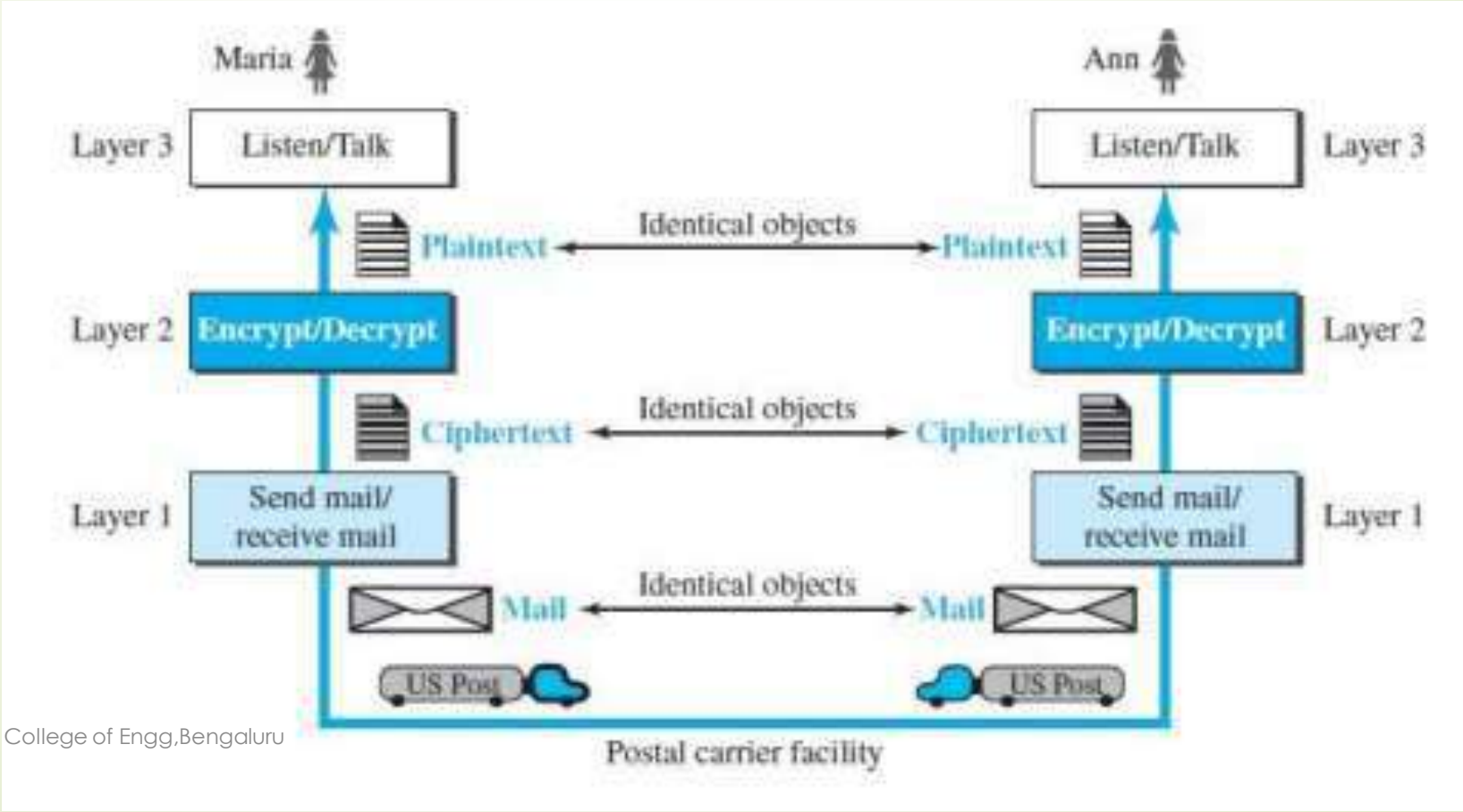
- Protocol Layering Protocol defines the rules that both the sender and receiver and all intermediate devices need to follow to be able to communicate effectively.
- When communication is simple, we may need only one simple protocol; when the communication is complex, we may need to divide the task between different layers, in which case we need a protocol at each layer, or protocol layering.

First Scenario

In the first scenario, communication is so simple that it can occur in only one layer. Assume Maria and Ann are neighbours with a lot of common ideas. Communication between Maria and Ann takes place in one layer, face to face, in the same language, as shown in Figure.

Second Scenario

- Now we can say that the communication between Maria and Ann takes place in three layers, as shown in Figure . We assume that Ann and Maria each have three machines(or robots) that can perform the task at each layer.



Need for protocol layering

- Protocol layering enables us to divide a complex task into several smaller and simpler tasks.
- A layer (module) can be defined as a black box with inputs and outputs, without concern about how inputs are changed to outputs. If two machines provide the same outputs when given the same inputs, they can replace each other.

Advantages

- Protocol layering allows to separate the services from the implementation. Lower layer give the services to the upper layer.
- Protocol layering in the Internet, is that communication does not always use only two end systems; there are intermediate systems that need only some layers, but not all layers. If we did not use protocol layering, we would have to make each intermediate system as complex as the end systems, which makes the whole system more expensive.

Principles of Protocol Layering

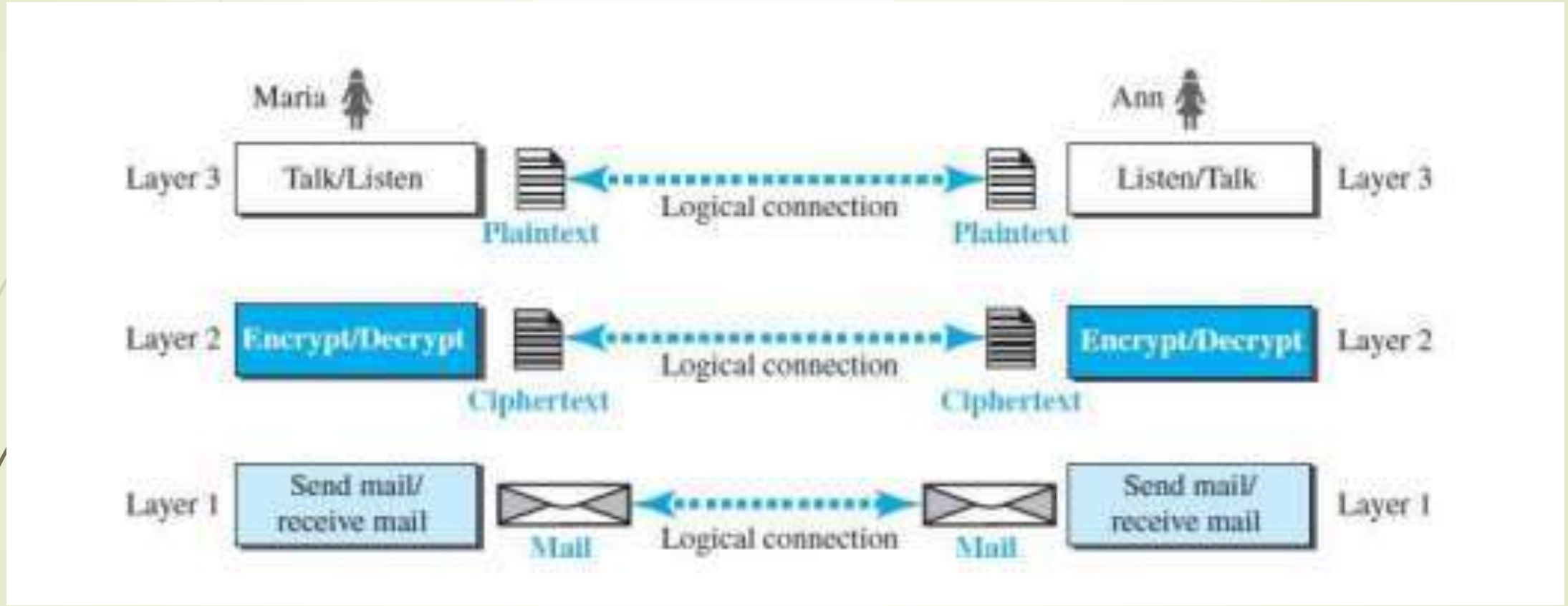
First Principle

- The first principle dictates that if we want bidirectional communication,
- we need to make each layer so that it is able to perform two opposite tasks, one in each direction. For example, the third layer task is to listen (in one direction) and talk (in the other direction). The second layer needs to be able to encrypt and decrypt. The first layer needs to send and receive mail.

Second Principle

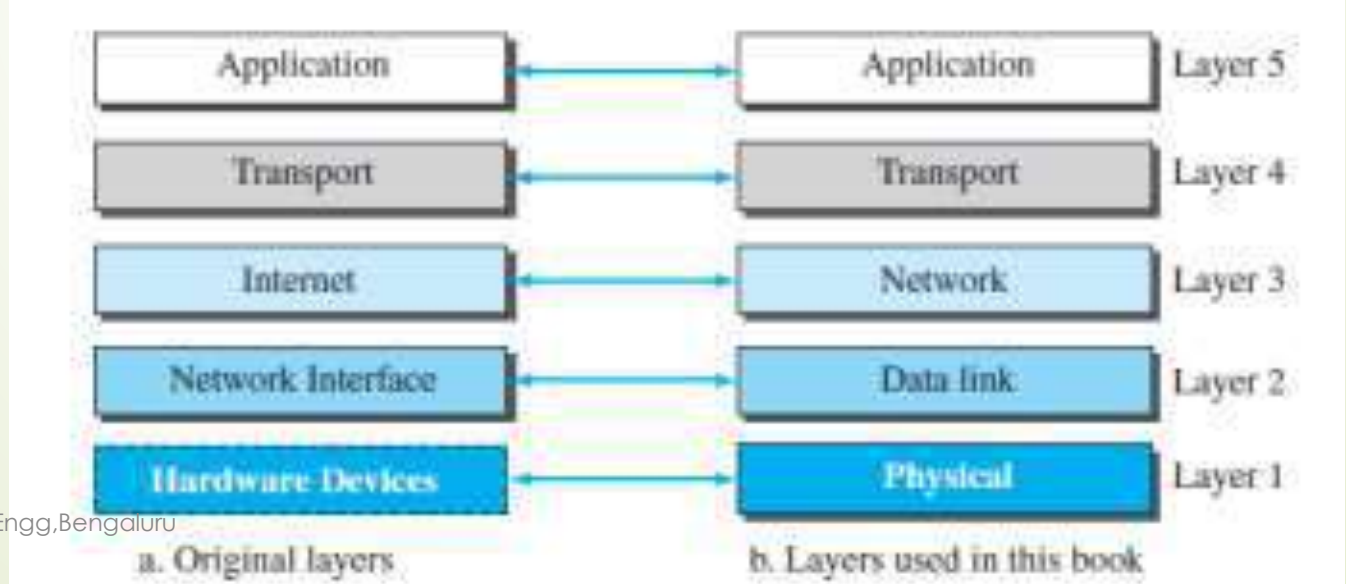
- The second principle that we need to follow in protocol layering is that the two objects under each layer at both sites should be identical.
- For example, the object under layer 3 at both sites should be a plaintext letter. The object under layer 2 at both sites should be a cipher text letter. The object under layer 1 at both sites should be a piece of mail.

Logical Connections



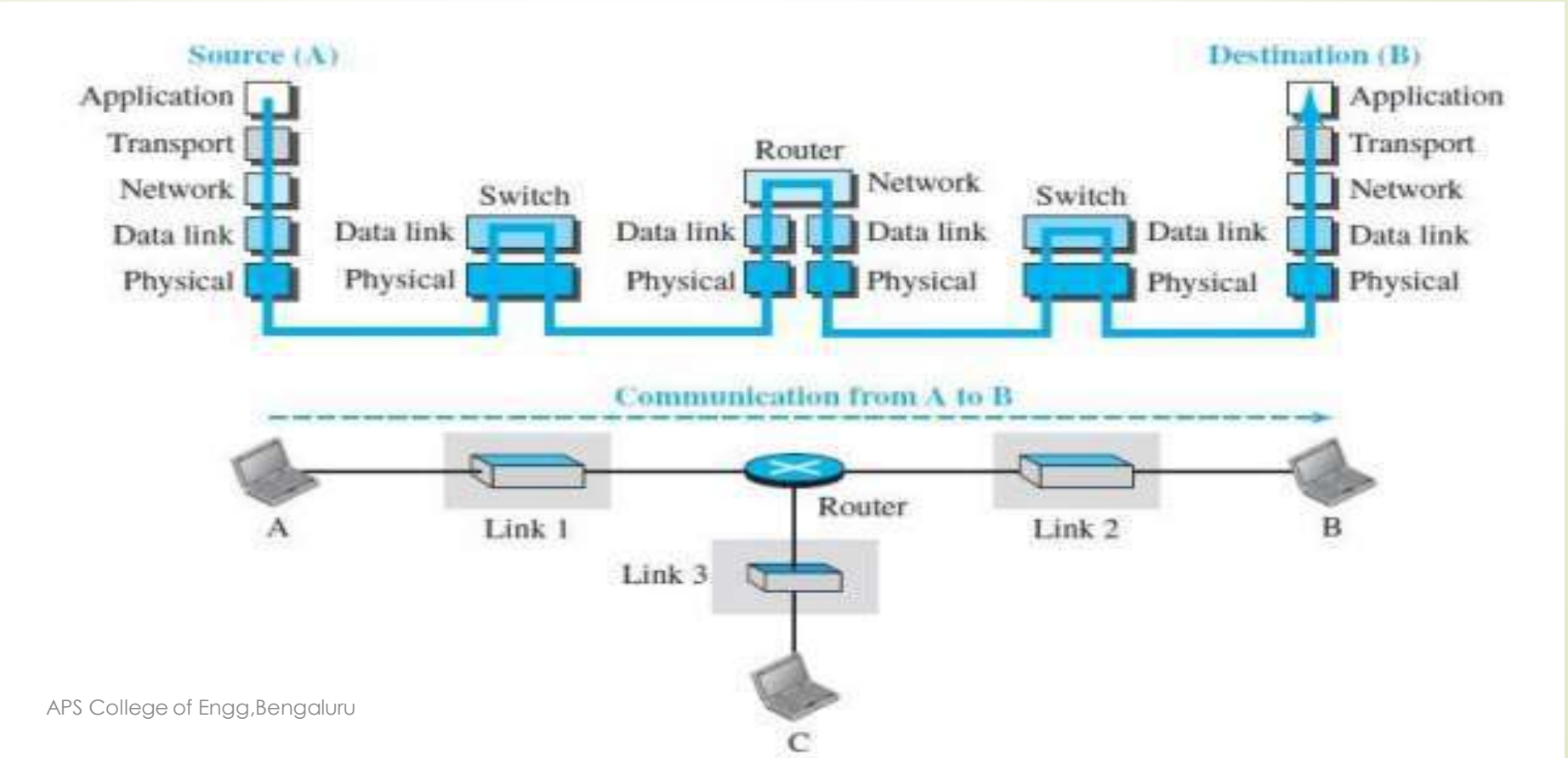
TCP/IP PROTOCOL SUITE

- TCP/IP is a protocol suite (a set of protocols organized in different layers) used in the Internet today.
- It is a hierarchical protocol made up of interactive modules, each of which provides a specific functionality.
- The term hierarchical means that each upper level protocol is supported by the services provided by one or more lower level protocols.



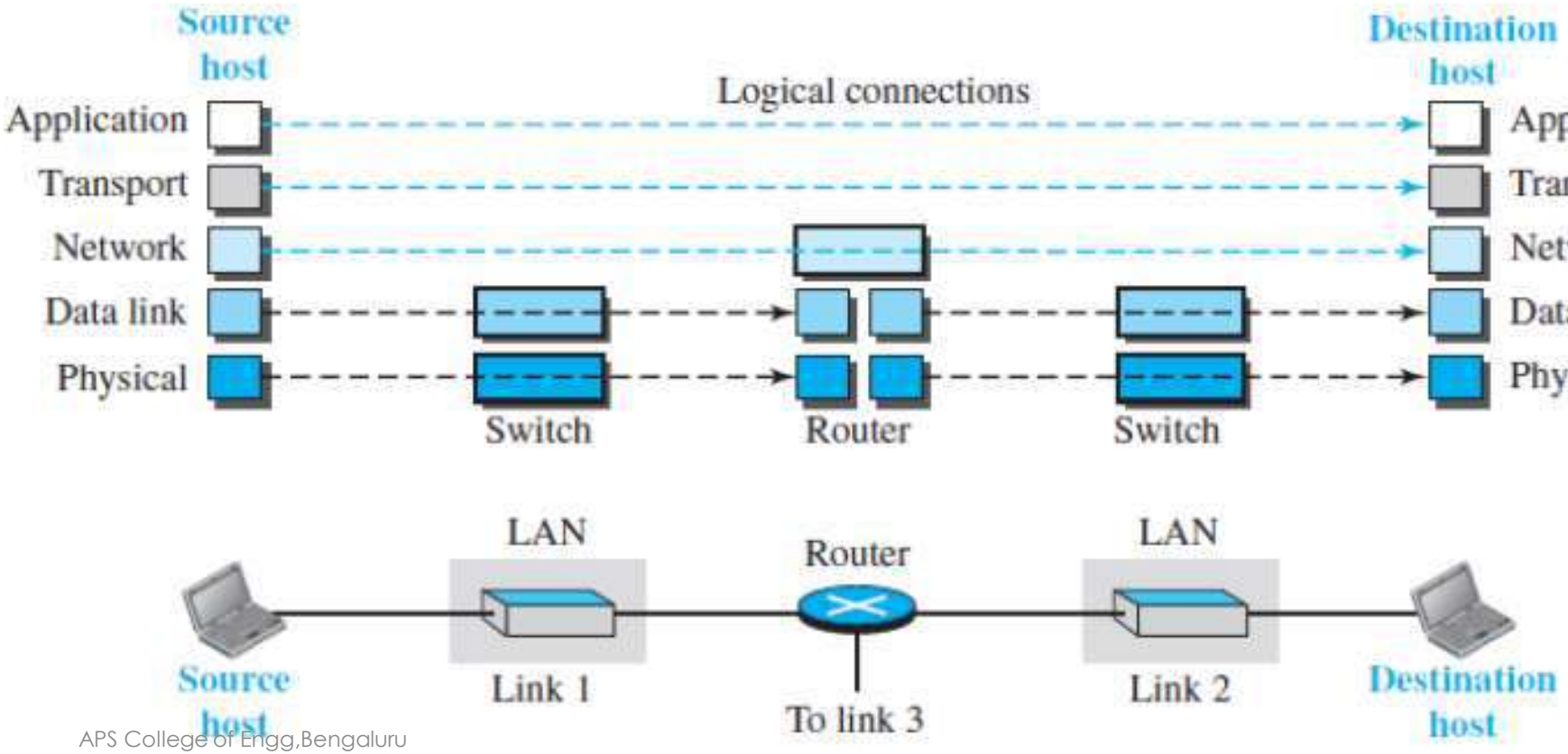
Layered Architecture

To show how the layers in the TCP/IP protocol suite are involved in communication between two hosts, we assume that we want to use the suite in a small internet made up of three LANs.



Layers in the TCP/IP Protocol Suite

To better understand the duties of each layer, we need to think about the logical connections between layers.



- Using logical connections makes it easier to think about the duty of each layer.
- As the figure shows, the duty of the application, transport, and network layers is end-to-end.
- However, the duty of the data-link and physical layers is hop-to-hop, in which a hop is a host or router. In other words, the domain of duty of the top three layers is the internet, and the domain of duty of the two lower layers is the link

Description of Each Layer

Physical Layer

- Physical layer is responsible for carrying individual bits in a frame across the link. Although the physical layer is the lowest level in the TCP/IP protocol suite, the communication between two devices at the physical layer is still a logical communication.
- Two devices are connected by a transmission medium (cable or air). Transmission medium does not carry bits, it carries electrical or optical signals. So the bits received in a frame from the data-link layer are transformed and sent through the transmission media.

The following are the main responsibilities of the physical layer

- Definition of Hardware Specifications, Encoding and Signalling, Data Transmission and Reception, Topology and Physical Network Design

Data-link Layer

- Internet is made up of several links(LANs and WANs) connected by routers. The data-link layer is responsible for taking the datagram and moving it across the link.(node to node communication)
- In each case, the data-link layer is responsible for moving the packet through the link. TCP/IP does not define any specific protocol for the data-link layer.
- It supports all the standard and proprietary protocols. The data-link layer takes a datagram and encapsulates it in a packet called a frame.
- Each link-layer protocol provide a different service like framing, Flow control, Error control and congestion control.

Network Layer

- The network layer is responsible for creating a connection between the source computer and the destination computer.
- The communication at the network layer is host-to-host. However, since there can be several routers from the source to the destination, the routers in the path are responsible for choosing the best route for each packet.
- The network layer is responsible packetizing and routing and forwarding the packet through possible routes.
- The network layer in the Internet includes the main protocol, Internet Protocol (IP), that defines the format of the packet, called a datagram at the network layer.
- IP is also responsible for routing a packet from its source to its destination, which is achieved by each router forwarding the datagram to the next router in its path.

- The network layer also includes unicast (one-to-one) and multicast (one-to-many) routing protocols. A routing protocol does not take part in routing but it creates forwarding tables for routers to help them in the routing process.
- The Internet Control Message Protocol(ICMP) helps IP to report some problems when routing a packet.
- The Internet Group Management Protocol(IGMP) is another protocol that helps IP in multitasking.
- The Dynamic Host Configuration Protocol (DHCP) helps IP to get the network-layer address for a host.
- The Address Resolution Protocol (ARP) is a protocol that helps IP to find the link-layer address of a host or a router when its network-layer address is given.

Transport Layer

- The transport layer at the source host gets the message from the application layer, encapsulates it in a transport layer packet (called a segment or a user datagram in different protocols) and sends it, through the logical (imaginary) connection, to the transport layer at the destination host.
- The transport layer is responsible for giving services to the application layer: to get a message from an application program running on the source host and deliver it to the corresponding application program on the destination host.
- The main protocol, **Transmission Control Protocol** (TCP), is a connection-oriented protocol that first establishes a logical connection between transport layers at two hosts before transferring data.

- **TCP** provides
  - Flow control (matching the sending data rate of the source host with the receiving data rate of the destination host to prevent overwhelming the destination),
  - Error control (to guarantee that the segments arrive at the destination without error and resending the corrupted ones),
  - Congestion control to reduce the loss of segments due to congestion in the network.
- **User Datagram Protocol (UDP)**, is a connectionless protocol that transmits user datagrams without first creating a logical connection.
- In UDP, each user datagram is an independent entity without being related to the previous or the next one.
- UDP is a simple protocol that does not provide flow, error, or congestion control.

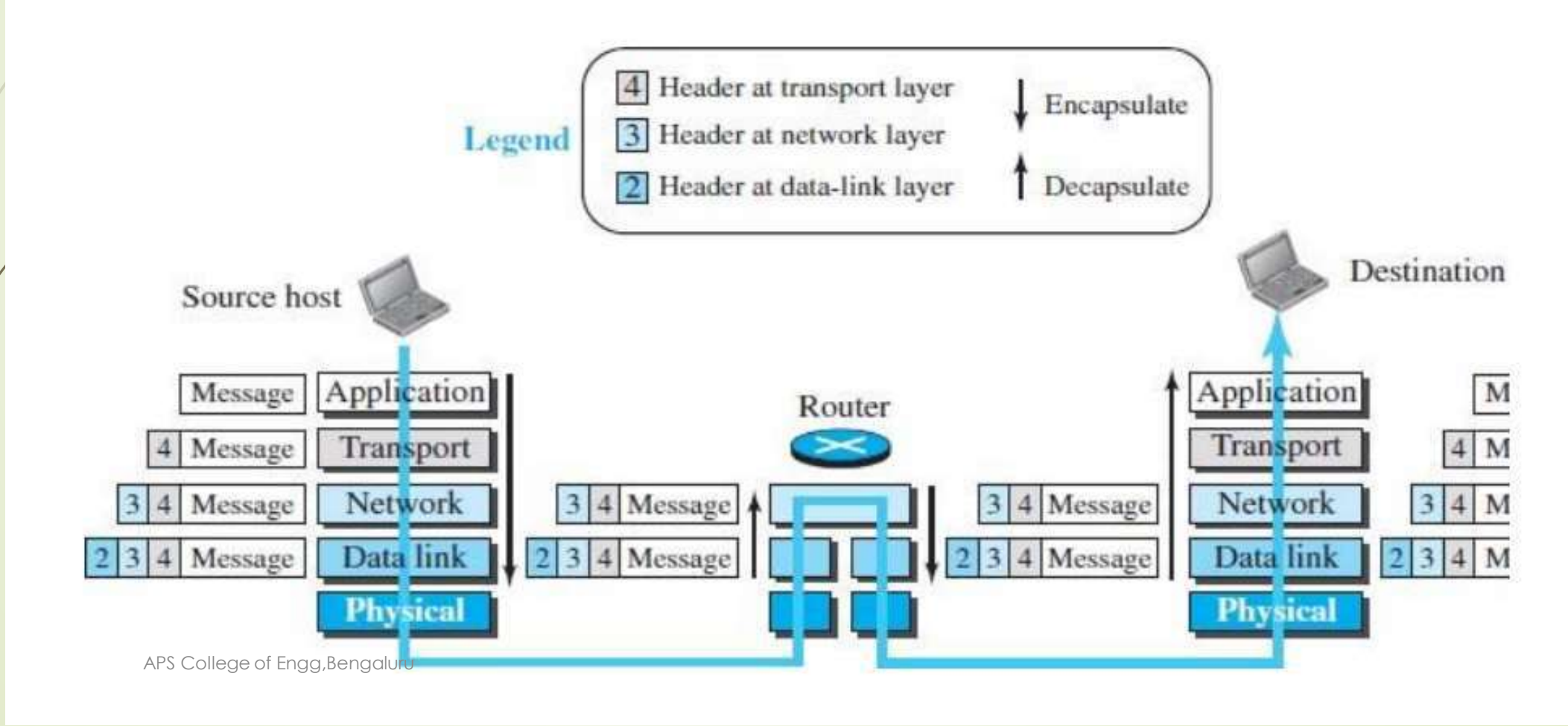
Application Layer

- The two application layers exchange messages between each other as though there were a bridge between the two layers. However, communication is done through all the layers.
- To communicate, a process sends a request to the other process and receives a response. Process-to-process communication is the duty of the application layer.
- The application layer in the Internet includes many predefined protocols.

1. The Hypertext Transfer Protocol (HTTP) is a vehicle for accessing the World Wide Web (WWW).
2. The Simple Mail Transfer Protocol (SMTP) is the main protocol used in electronic mail (e- mail)service.
3. The File Transfer Protocol (FTP) is used for transferring files from one host to another.
4. The Terminal Network (TELNET) and Secure Shell (SSH) are used for accessing a site remotely.
5. The Simple Network Management Protocol (SNMP) is used by an administrator to manage the Internet at global and local levels.
6. The Domain Name System (DNS) is used by other protocols to find the network-layer address of a computer.
7. The Internet Group Management Protocol (IGMP) is used to collect membership in a group.

Encapsulation and Decapsulation

One of the important concepts in protocol layering in the Internet is encapsulation/ decapsulation..



Encapsulation at the Source Host

1. At the application layer, the data to be exchanged is referred to as a message. A message normally does not contain any header or trailer, but if it does, we refer to the whole as the message. The message is passed to the transport layer.
2. The transport layer takes the message as the payload, the load that the transport layer should take care of. It adds the transport layer header to the payload, which contains the identifiers of the source and destination application programs that want to communicate plus some more information that is needed for the end-to end delivery of the message, such as information needed for flow, error control, or congestion control.

3. The network layer takes the transport-layer packet as data or payload and adds its own header to the payload. The header contains the addresses of the source and destination hosts and some more information used for error checking of the header, fragmentation information, and so on.

4. The data-link layer takes the network-layer packet as data or payload and adds its own header, which contains the link-layer addresses of the host or the next hop (the router). The result is the link-layer packet, which is called a frame.

- The frame is passed to the physical layer for transmission.

Decapsulation and Encapsulation at the Router

At the router, we have both decapsulation and encapsulation because the router is connected to two or more links.

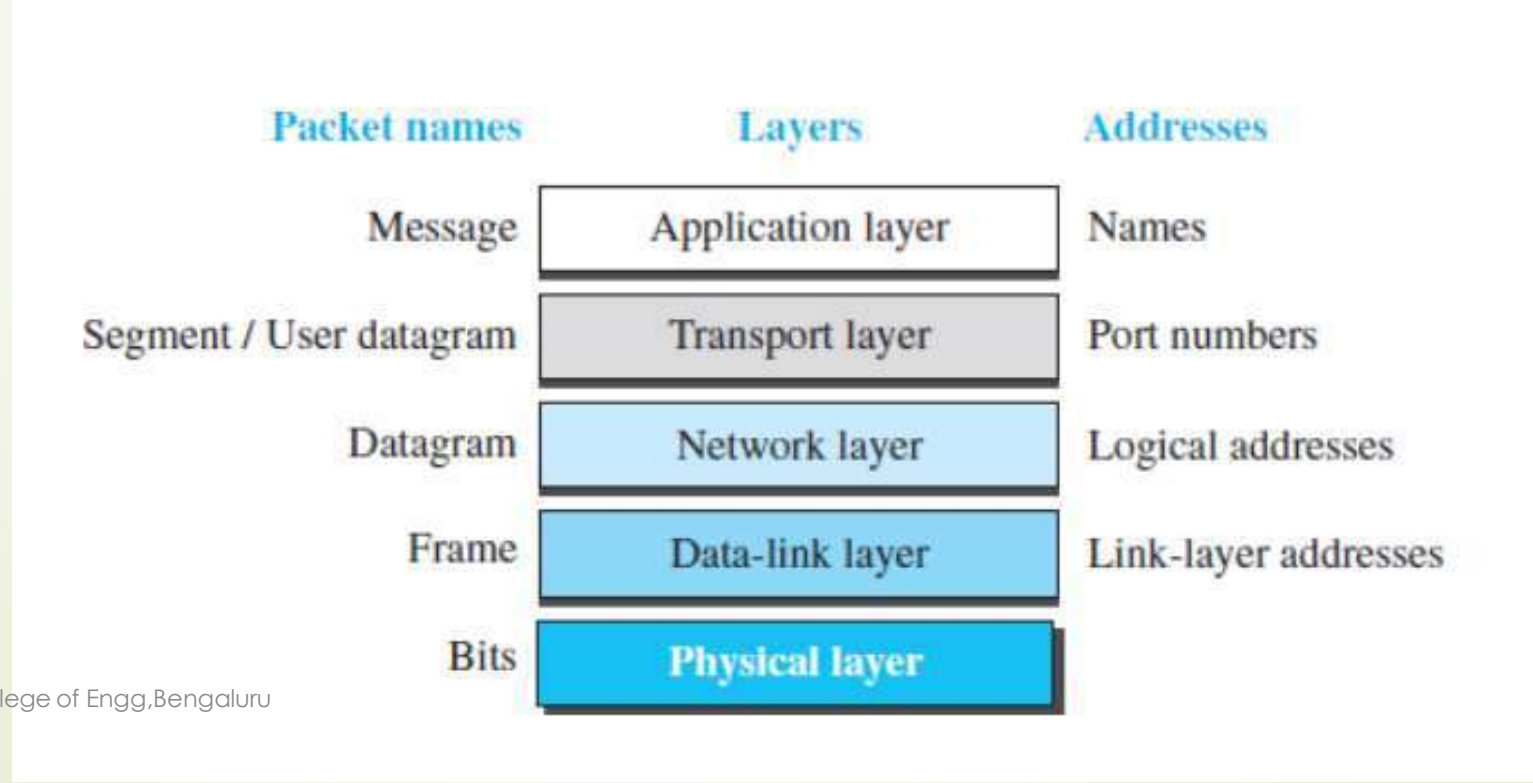
1. After the set of bits are delivered to the data-link layer, this layer decapsulates the datagram from the frame and passes it to the network layer.
2. The network layer only inspects the source and destination addresses in the datagram header and consults its forwarding table to find the next hop to which the datagram is to be delivered.
3. The data-link layer of the next link encapsulates the datagram in a frame and passes it to the physical layer for transmission.

Decapsulation at the Destination Host

- At the destination host, each layer only decapsulates the packet received, removes the payload, and delivers the payload to the next-higher layer protocol until the message reaches the application layer.

Addressing

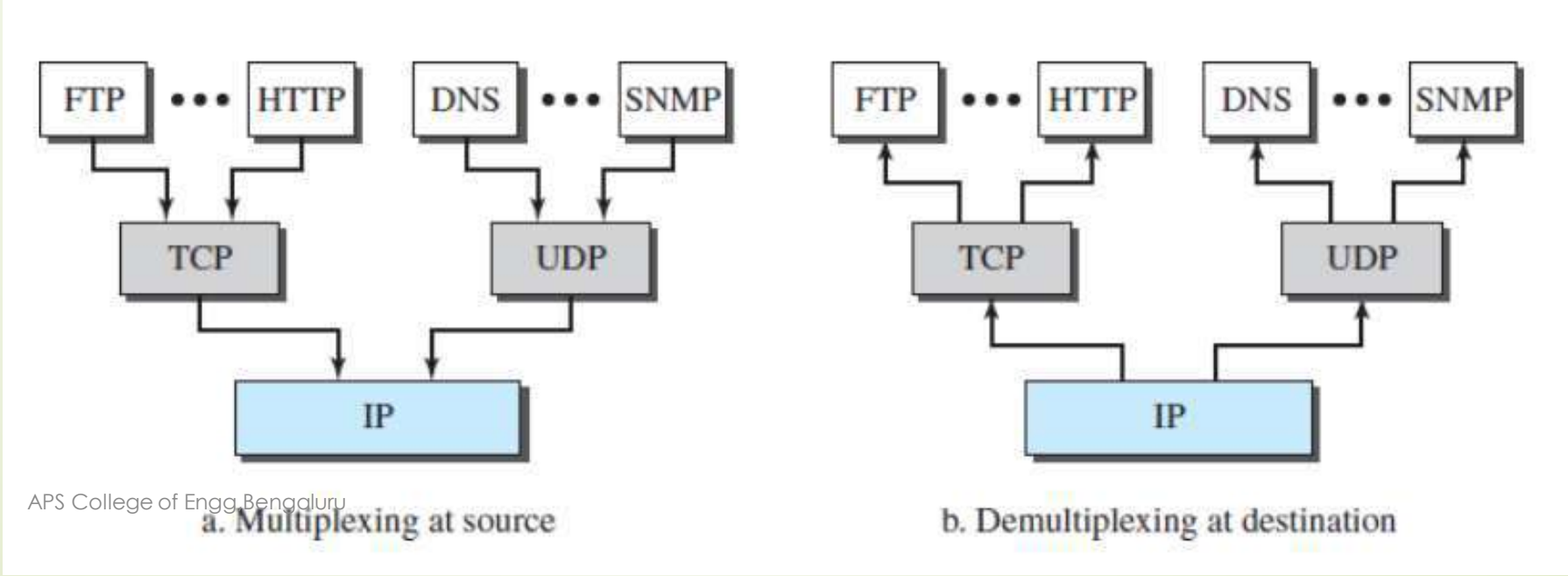
- Although it looks as if we need five pairs of addresses, one pair per layer, we normally have only four because the physical layer does not need addresses; the unit of data exchange at the physical layer is a bit, which definitely cannot have an address.



- At the transport layer, addresses are called port numbers, and these define the application- layer programs at the source and destination. Port numbers are local addresses that distinguish between several programs running at the same time.
- At the network-layer, the addresses are global, with the whole Internet as the scope. A network-layer address uniquely defines the connection of a device to the Internet.
- The link-layer addresses, sometimes called MAC addresses, are locally defined addresses, each of which defines a specific host or router in a network (LAN or WAN).

Multiplexing and Demultiplexing

- TCP/IP protocol suite uses several protocols at some layers, we have multiplexing at the source and demultiplexing at the destination.
- Multiplexing means that a protocol at a layer can encapsulate a packet from several next- higher layer protocols (one at a time).
- Demultiplexing means that a protocol can decapsulate and deliver a packet to several next-higher layer protocols (one at a time).



- At the transport layer, either UDP or TCP can accept a message from several application-layer protocols.
- At the network layer, IP can accept a segment from TCP or a user datagram from UDP. IP can also accept a packet from other protocols such as ICMP, IGMP, and so on.
- At the data-link layer, a frame may carry the payload coming from IP.

THE OSI MODEL

- An ISO standard that covers all aspects of network communications is the Open Systems Interconnection (OSI)model.
- An open system is a set of protocols that allows any two different systems to communicate regardless of their underlying architecture.
- The purpose of the OSI model is to show how to facilitate communication between different systems without requiring changes to the logic of the underlying hardware and software.
- The OSI model is not a protocol; it is a model for understanding and designing a network architecture that is flexible, robust, and interoperable.
- The OSI model is a layered framework for the design of network systems that allows communication between all types of computer systems.

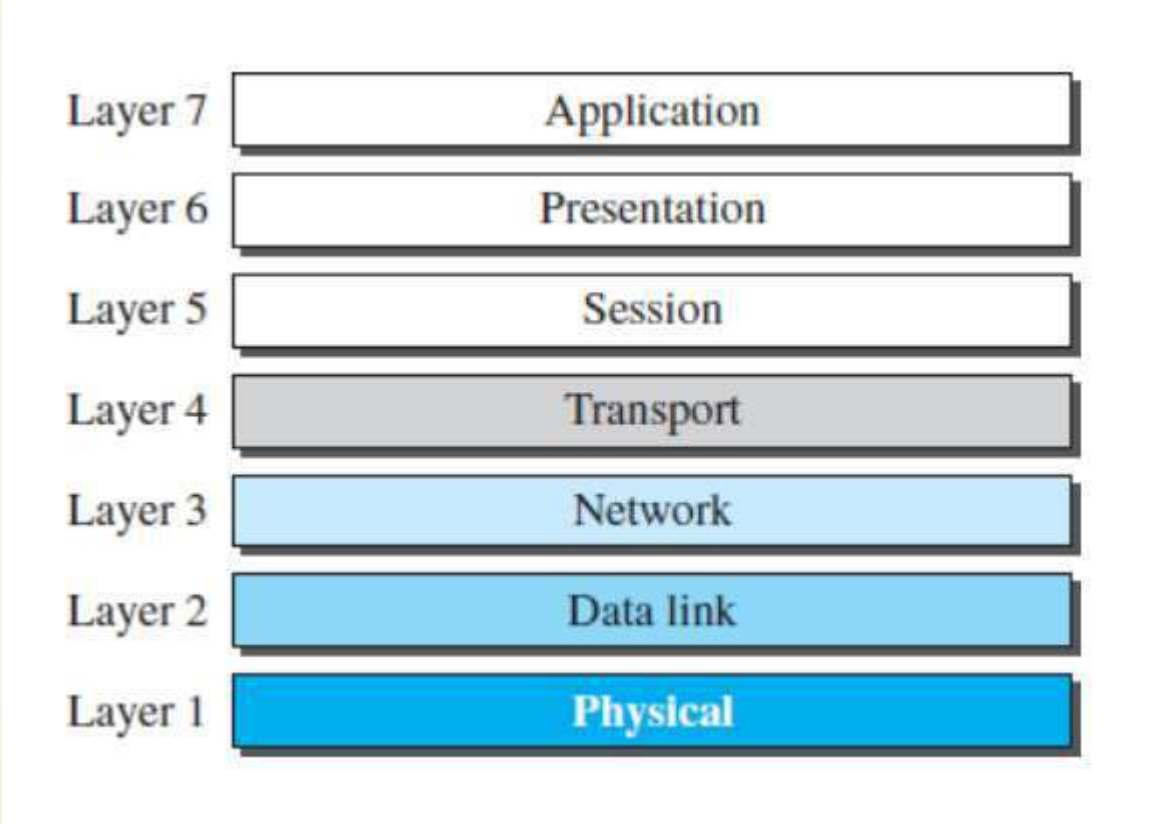


Fig: OSI model

OSI versus TCP/IP

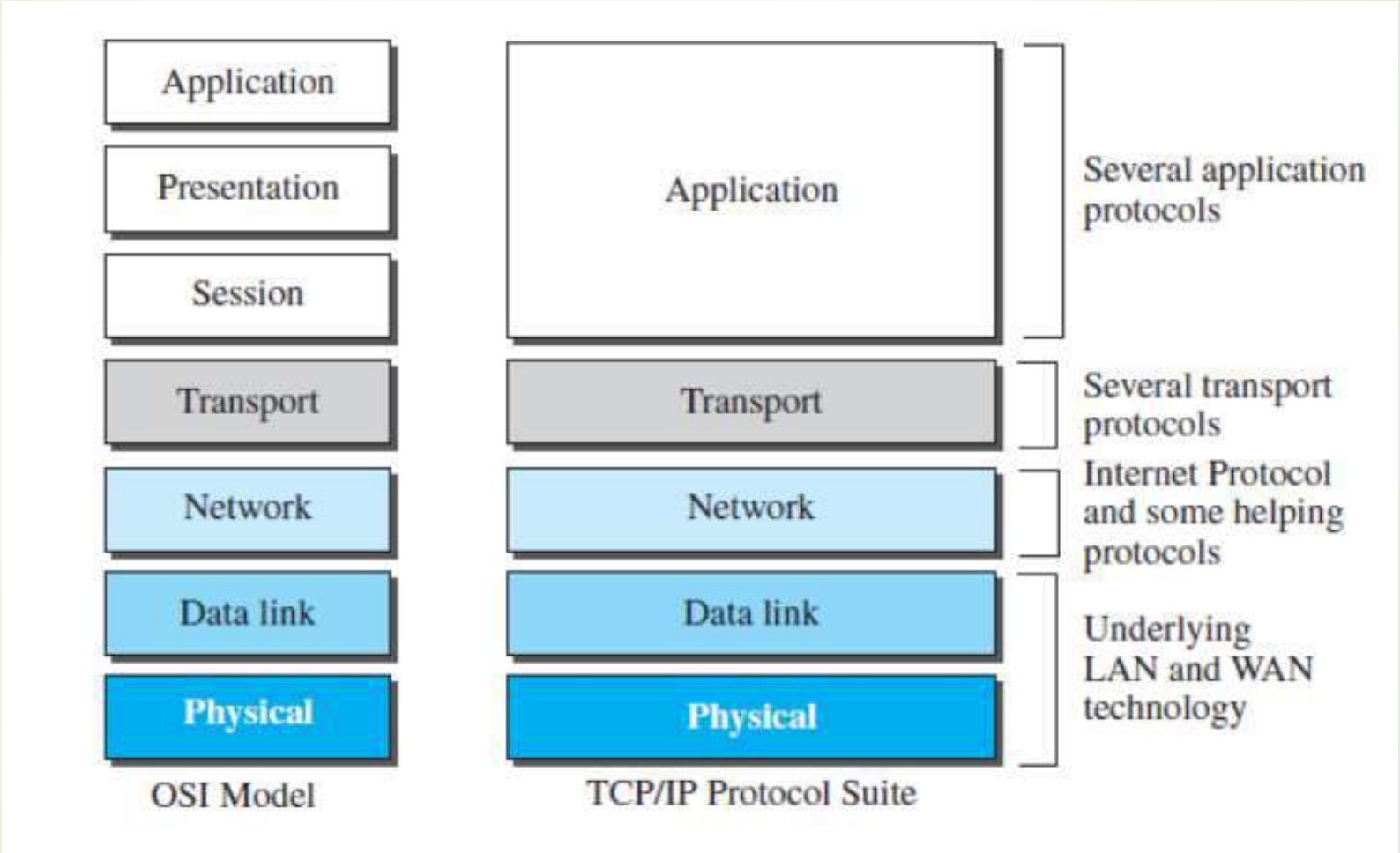


Fig: TCP/IP and OSI mode

- When we compare the two models, we find that two layers, session and presentation, are missing from the TCP/IP protocol suite.
- These two layers were not added to the TCP/IP protocol suite after the publication of the OSI model.
- The application layer in the suite is usually considered to be the combination of three layers in the OSI model.
- First, TCP/IP has more than one transport-layer protocol. Some of the functionalities of the session layer are available in some of the transport- layer protocols.
- Second, the application layer is not only one piece of software. Many applications can be developed at this layer

DATA-LINK LAYER

- The Internet is a combination of networks glued together by connecting devices (routers or switches).
- If a packet is to travel from a host to another host, it needs to pass-through these networks.
- Communication at the data-link layer is made up of five separate logical connections between the data-link layers in the path.

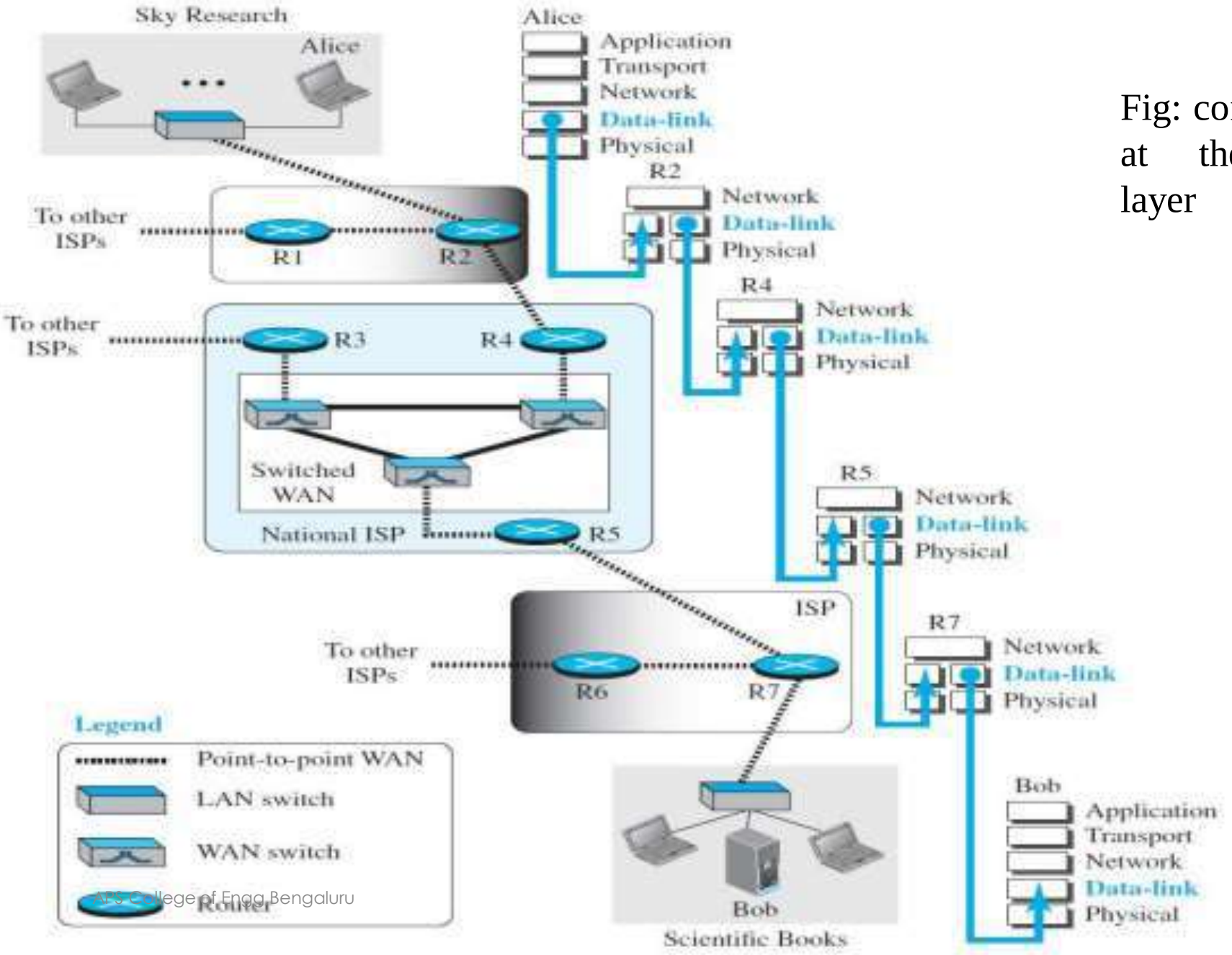
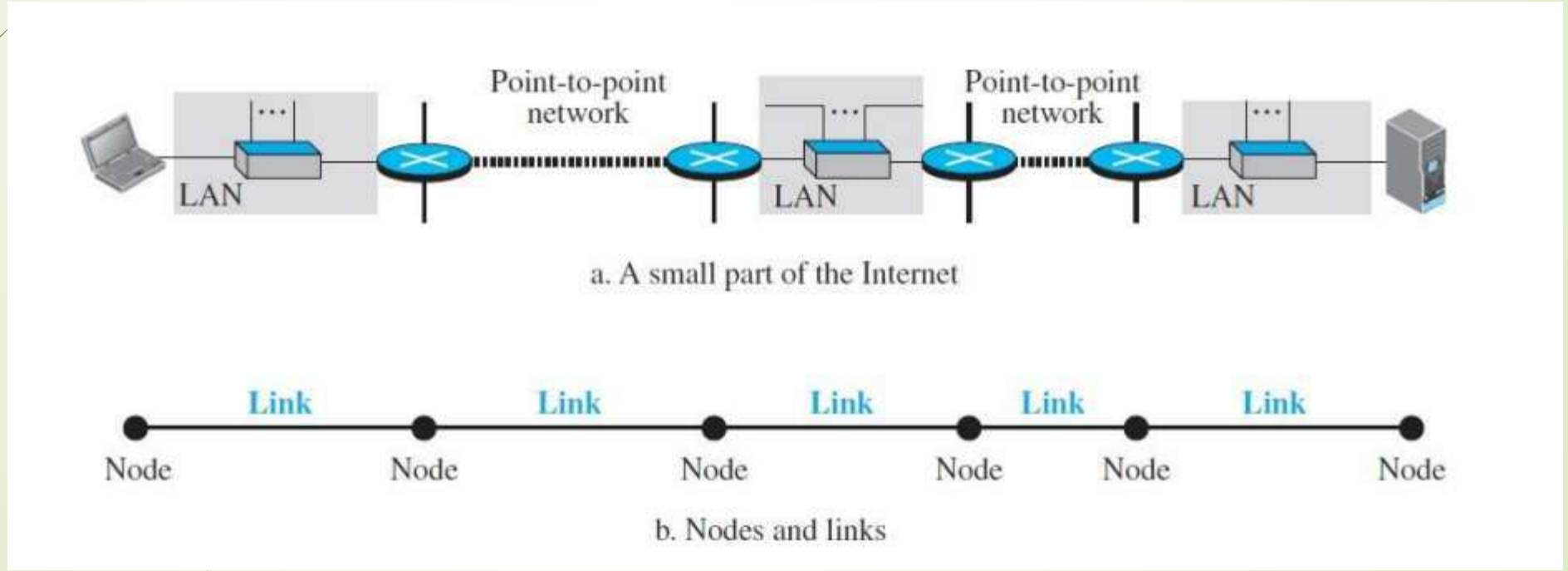


Fig: communication at the data-link layer

- The data-link layer at Alice's computer communicates with the data-link layer at router R2.
- The data-link layer at router R2 communicates with the data-link layer at router R4, and so on.
- Finally, the data-link layer at router R7 communicates with the data-link layer at Bob's computer.
- Only one data-link layer is involved at the source or the destination, but two data-link layers are involved at each router. The reason is that Alice's and Bob's computers are each connected to a single network, but each router takes input from one network and sends output to another network.

Nodes and Links

- Communication at the data-link layer is node-to-node.
- A data unit from one point in the Internet needs to pass through many networks (LANs and WANs) to reach another point.
- Theses LANs and WANs are connected by routers.



Services provided by the data-link layer.

- When a packet is travelling in the Internet, the data-link layer of a node (host or router) is responsible for delivering a datagram to the next node in the path.
- For this purpose, the data-link layer of the sending node needs to encapsulate the datagram received from the network in a frame, and the data-link layer of the receiving node needs to decapsulate the datagram from the frame.
- One may ask why we need encapsulation and decapsulation at each intermediate node.
- The reason is that each link may be using a different protocol with a different frame format.
- Even if one link and the next are using the same protocol, encapsulation and decapsulation are needed because the link-layer addresses are normally different.

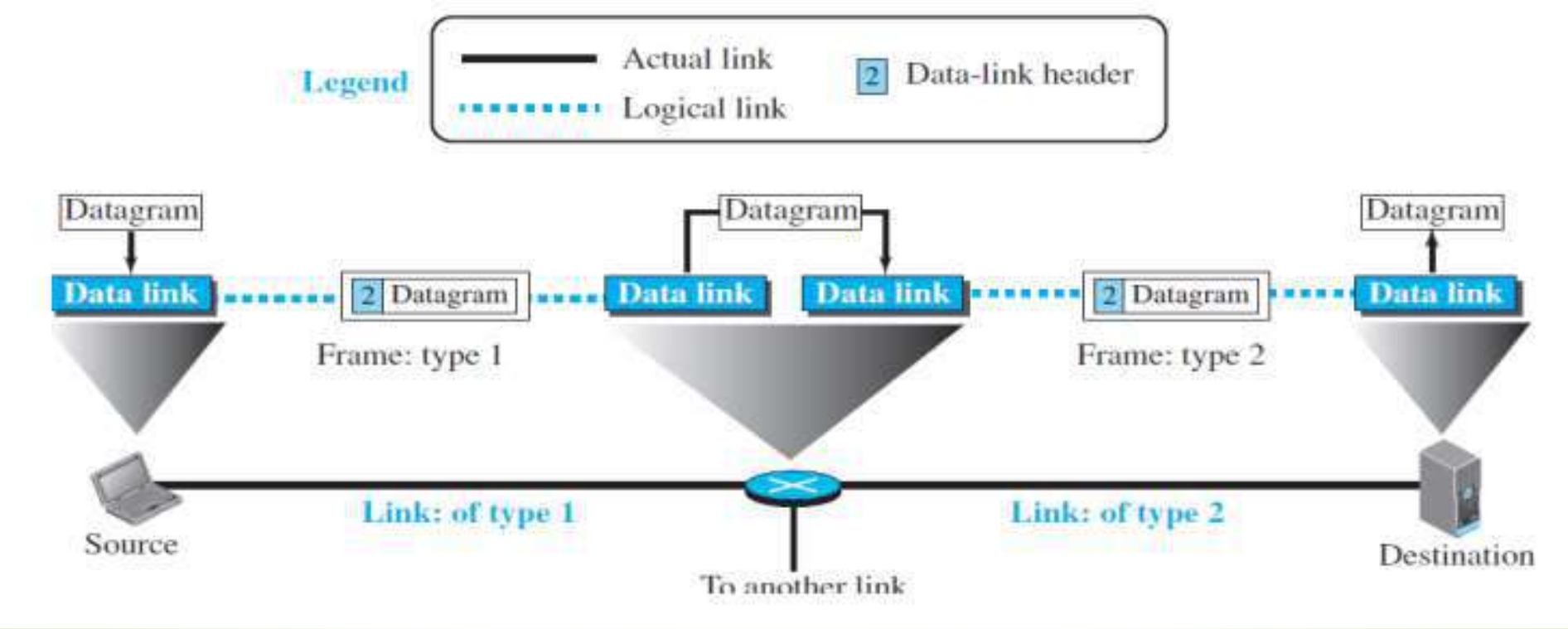


Fig: communication with only three nodes

Framing

- The first service provided by the data-link layer is framing. The data-link layer at each node needs to encapsulate the datagram (packet received from the network layer) in a frame before sending it to the next node. The node also needs to decapsulate the datagram from the frame received on the logical channel.

Flow Control

- If the rate of produced frames is higher than the rate of consumed frames, frames at the receiving end need to be buffered while waiting to be consumed (processed).

- Definitely, we cannot have an unlimited buffer size at the receiving side. We have two choices
- The first choice is to let the receiving data-link layer drop the frames if its buffer is full.
- The second choice is to let the receiving data-link layer send a feedback to the sending data-link layer to ask it to stop or slow down.
- Different data-link-layer protocols use different strategies for flow control.

Error Control

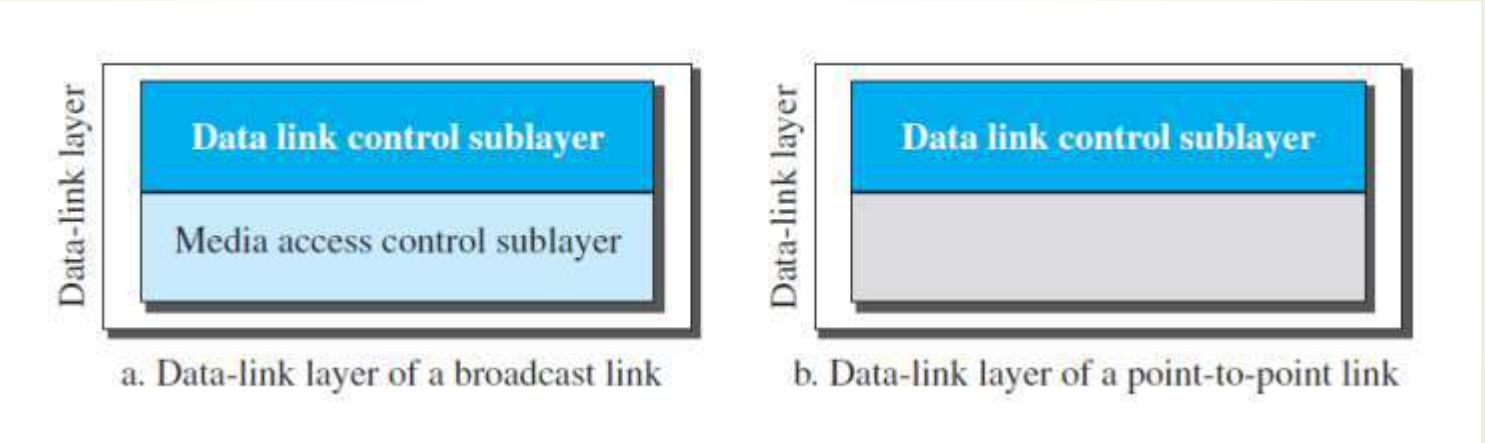
- At the sending node, a frame in a data-link layer needs to be changed to bits, transformed to electromagnetic signals, and transmitted through the transmission media.
- At the receiving node, electromagnetic signals are received, transformed to bits, and put together to create a frame.
- The error needs first to be detected. After detection, it needs to be either corrected at the receiver node or discarded and retransmitted by the sending node.
- Since error detection and correction is an issue in every layer(node-to node or host-to-host)

Congestion Control

- Although a link may be congested with frames, which may result in frame loss, most data-link- layer protocols do not directly use a congestion control to alleviate congestion, although some wide-area networks do.
- In general, congestion control is considered an issue in the network layer or the transport layer because of its end-to-end nature.

Two Categories of Link

- In a point-to-point link, the link is dedicated to the two devices;
- in a broadcast link, the link is shared between several pairs of devices.
- The data-link layer is divided into two sublayers: data link control (DLC) and media access control (MAC).



a. Data-link layer of a broadcast link

b. Data-link layer of a point-to-point link

- The data link control sublayer deals with all issues common to both point-to-point and broadcast links.
- The media access control sublayer deals only with issues specific to broadcast links.

LINK-LAYER ADDRESSING

- In a connectionless internetwork such as the Internet we cannot make a datagram reach its destination using only IP addresses.
- The reason is that each datagram in the Internet, from the same source host to the same destination host, may take a different path.
- The source and destination IP addresses define the two ends but cannot define which links the datagram should pass through. We need to remember that the IP addresses in a datagram should not be changed.
- The above discussion shows that we need another addressing mechanism in a connectionless internetwork: the link-layer addresses of the two nodes. A link-layer address is sometimes called a link address, sometimes a physical address, and sometimes a MAC address.

- When a datagram passes from the network layer to the data-link layer, the datagram will be encapsulated in a frame and two data-link addresses are added to the frame header. These two addresses are changed every time the frame

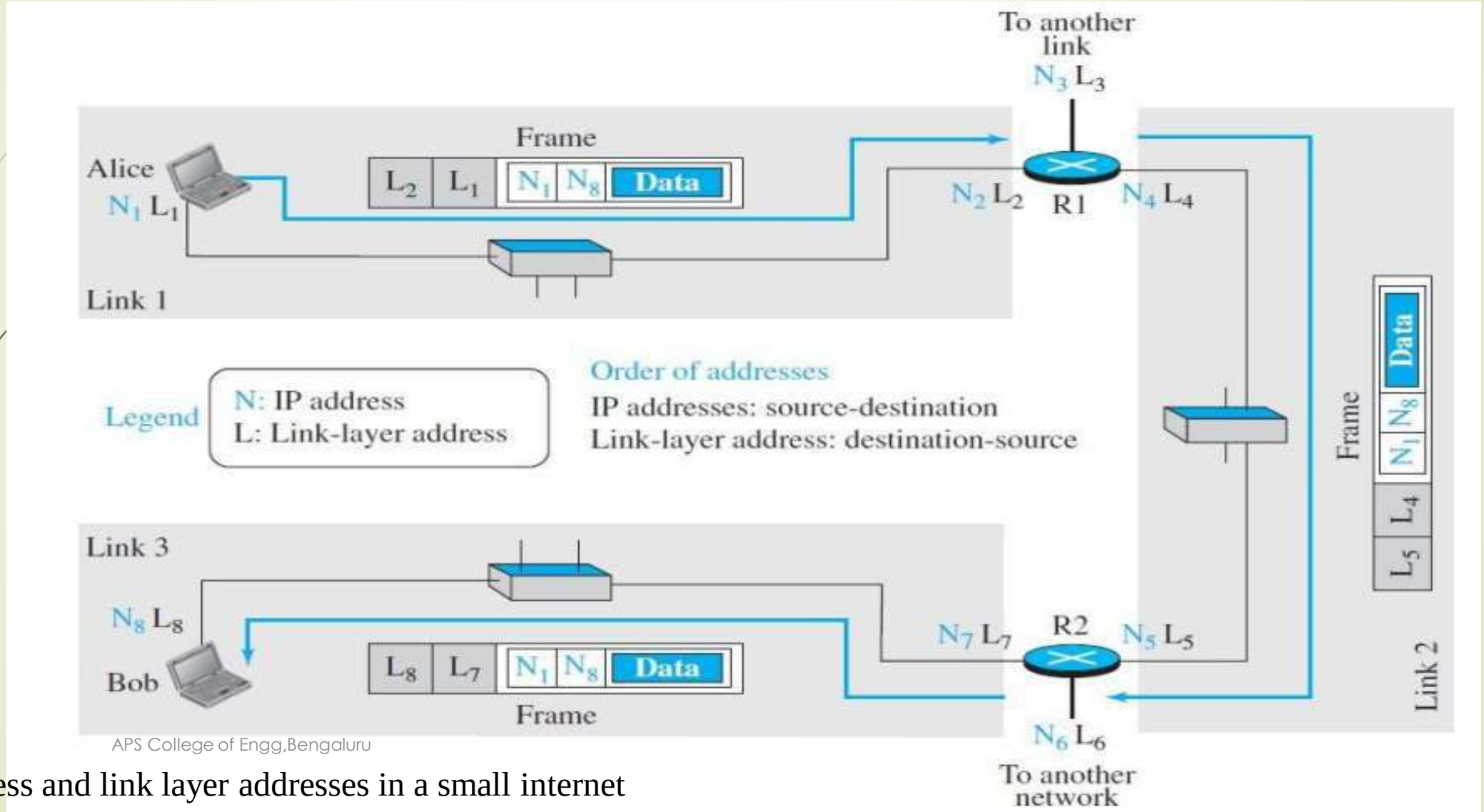


Fig: IP address and link layer addresses in a small internet

- In link 1, the link-layer addresses are L1 and L2. In link 2, they are L4 and L5. In link 3, they are L7 and L8.
- Note that the IP addresses and the link-layer addresses are not in the same order. For IP addresses, the source address comes before the destination address; for link-layer addresses, the destination address comes before the source.

Different link-layer protocols.

- Three Types of addresses in link-layer protocols are:
  - Unicast,
  - Multicast, and
  - Broadcast.

- **Unicast Address**: Each host or each interface of a router is assigned a unicast address. Unicasting means one-to-one communication. A frame with a unicast address destination is destined only for one entity in the link.
- **Multicast Address:** Some link-layer protocols define multicast addresses. Multicasting means one-to many communication.
- **Broadcast Address:** Some link-layer protocols define a broadcast address. Broadcasting means one-to-all communication.

Address Resolution Protocol(ARP)

- Anytime a node has an IP datagram to send to another node in a link, it has the IP address of the receiving node.
- The source host knows the IP address of the default router. Each router except the last one in the path gets the IP address of the next router by using its forwarding table.
- The last router knows the IP address of the destination host.
- The ARP protocol is one of the auxiliary protocols defined in the network layer.
- ARP accepts an IP address from the IP protocol, maps the address to the corresponding link-layer address, and passes it to the data-link layer.

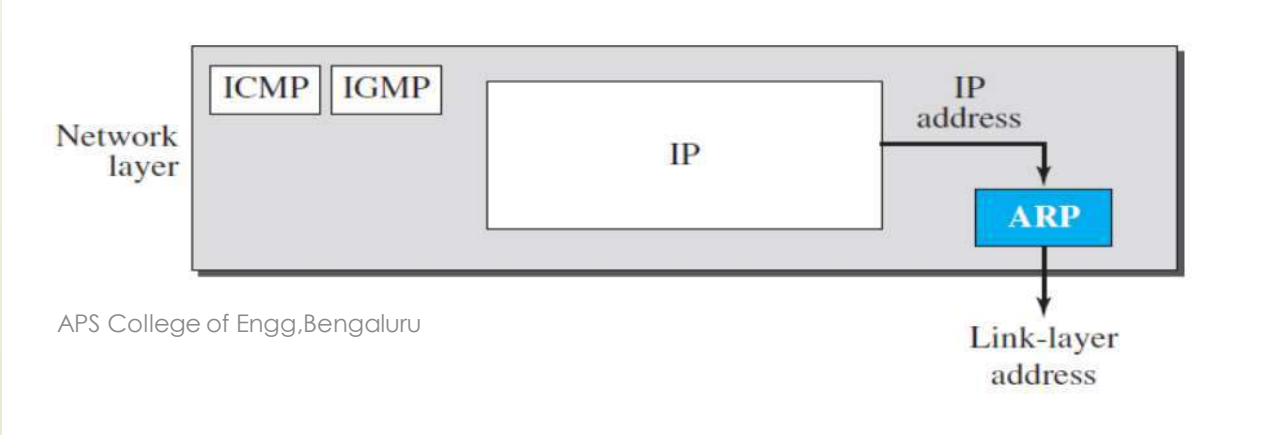


Fig: Position of ARP in TCP/IP protocol suite

- Anytime a host or a router needs to find the link-layer address of another host or router in its network, it sends an ARP request packet.
- The packet includes the link-layer and IP addresses of the sender and the IP address of the receiver.

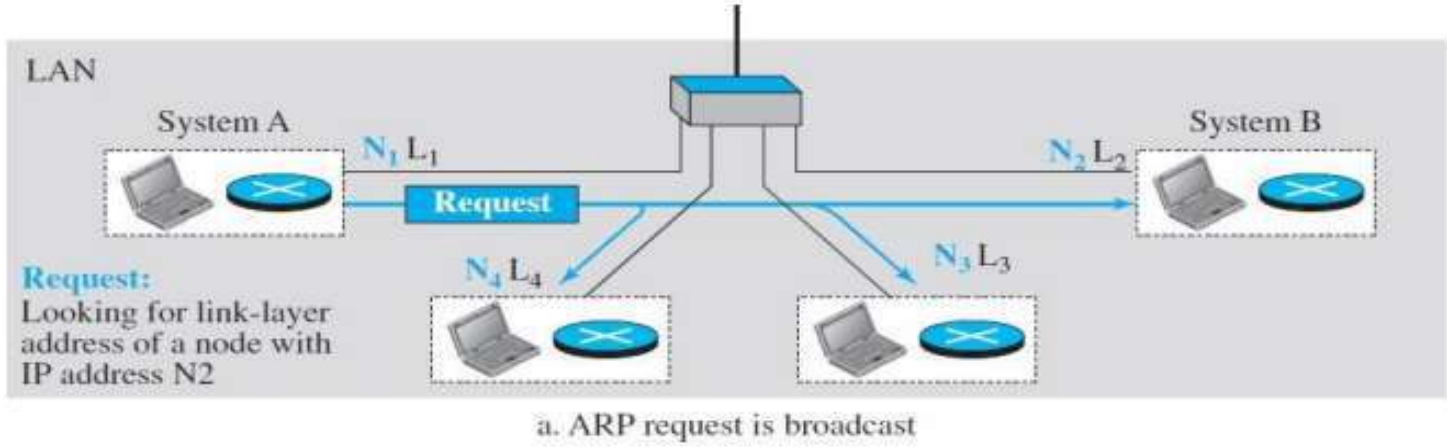


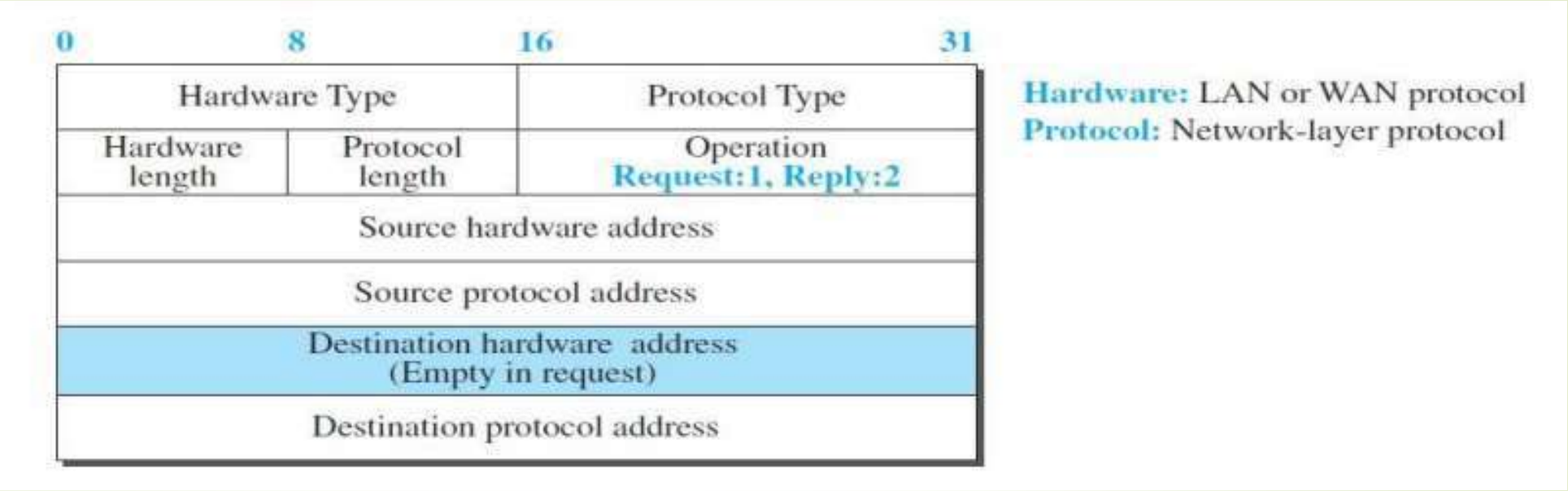
Fig: ARP operation

- Every host or router on the network receives and processes the ARP request packet, but only the intended recipient recognizes its IP address and sends back an ARP response packet. The response packet contains the recipient's IP and link-layer addresses.
- The packet is unicast directly to the node that sent the request packet. In Figure (a) the system on the left (A) has a packet that needs to be delivered to another system (B) with IP address N2.
- It uses the services of ARP by asking the ARP protocol to send a broadcast ARP request packet to ask for the physical address of a system with an IP address of N2.
- This packet is received by every system on the physical network, but only system B will answer it.
- System B sends an ARP reply packet that includes its physical address. Now system A can send all the packets it has for this destination using the physical address it received.

Caching

- Let us assume that there are 20 systems connected to the network (link):system A, system B, and 18 other systems.
- We also assume that system A has10 datagrams to send to system in one second.
- Without using ARP, system A needs to send 10 broadcast frames. Each of the 18 other systems need to receive the frames, decapsulate the frames, remove the datagram and pass it to their network-layer to find out the datagrams do not belong to them. This means processing and discarding180 broadcast frames.
- Using ARP, system A needs to send only one broadcast frame. Each of the 18 other systems need to receive the frames, decapsulate the frames, remove the ARP message and pass the message to their ARP protocol to find that the frame must be discarded. This means processing and discarding only 18 (instead of 180) broadcast frames. After system B responds with its own data-link address, system A can store the link-layer address in its cache memory. The rest of the nine frames are only unicast. Since processing broadcast frames is expensive(time consuming), the first method is preferable.

ARP Packet Format



- The source hardware and source protocol addresses are variable-length fields defining the link- layer and network-layer addresses of the sender.
- The destination hardware address and destination protocol address fields define the receiver link-layer and network-layer addresses.
- An ARP packet is encapsulated directly into a data-link frame. The frame needs to have a field to show that the payload belongs to the ARP and not to the network-layer datagram.

Thank You

APS College of Engineering,

(Affiliated to Visvesvaraya Technological University and Approved by AICTE, NAAC Accredited)

Somanahalli, Kanakapura Main Road, Bengaluru-560116

Department of Electronics and Communication Engineering

Subject Name: Computer Communication Networks

Subject Name: 21EC73

Semester : 5th

Academic Year: ODD 2023-2024

Module-2

Faculty Name: Dr. Naik D C
Assistant Professor,
Dept., of ECE.

Module-2

- Data Link Control services: Framing, Flow and Error Control
- Media Access Control: Random Access: ALOHA, CSMA, CSMA/CD
- Connecting Devices: Hubs, Switches, Virtual LANs: Membership, Configuration, Communication between switches, Advantages.
- Wired and Wireless LANs: Ethernet Protocol, Standard Ethernet.
- Introduction to wireless LAN: Architectural Comparison, Characteristics, Access Control.

- DLC SERVICES The data link control (DLC) deals with procedures for communication between two adjacent nodes—node-to-node communication—no matter whether the link is dedicated or broadcast.
- Data link control functions include framing and flow and error control.

Framing Data

- Transmission in the physical layer means moving bits in the form of a signal from the source to the destination.
- Framing in the data-link layer separates a message from one source to a destination by adding a sender address and a destination address.
- The destination address defines where the packet is to go; the sender address helps the recipient acknowledge the receipt.
- When a message is carried in one very large frame, even a single-bit error would require the retransmission of the whole frame. When a message is divided into smaller frames, a single-bit error affects only that small frame.

Frame Size

- Frames can be of fixed or variable size.
- **Fixed-size framing**: there is no need for defining the boundaries of the frames; the size itself can be used as a delimiter.
- An example of this type of framing is the (Asynchronous transfer mode)ATM WAN, which uses frames of fixed size called cells.
- **Variable-size framing**: prevalent in local-area networks. Invariable-size framing, we need a way to define the end of one frame and the beginning of the next.

- Two approaches were used for this purpose:
  - a character-oriented approach and
  - a bit-oriented approach

Character-Oriented Framing

- In character-oriented (or byte-oriented) framing, data to be carried are 8-bit characters from a coding system such as ASCII.

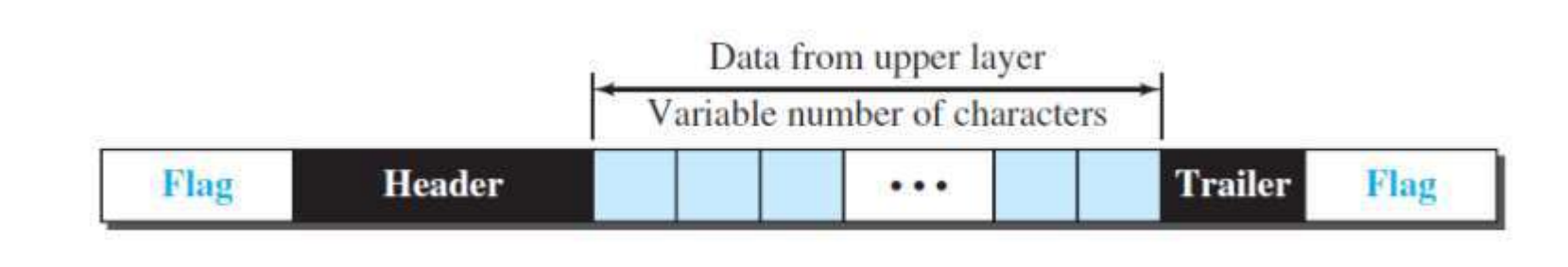


Fig: A frame in a character-oriented protocol

- The header, which normally carries the source and destination addresses and other control information, and the trailer, which carries error detection redundant bits, are also multiples of 8 bits.

- To separate one frame from the next, an 8-bit (1-byte) flag is added at the beginning and the end of a frame.
- The flag, composed of protocol-dependent special characters, signals the start or end of a frame.
- Character-oriented framing was popular when only text was exchanged by the data-link layers.
- The flag could be selected to be any character not used for text communication. Now, however, we send other types of information such as graphs, audio, and video.
- Any character used for the flag could also be part of the information. If this happens, the receiver, when it encounters this pattern in the middle of the data, thinks it has reached the end of the frame.

- To fix this problem, a byte-stuffing strategy was added to character-oriented framing. In byte stuffing (or character stuffing), a special byte is added to the data section of the frame when there is a character with the same pattern as the flag.
- The data section is stuffed with an extra byte. This byte is usually called the escape character (ESC) and has a predefined bit pattern.

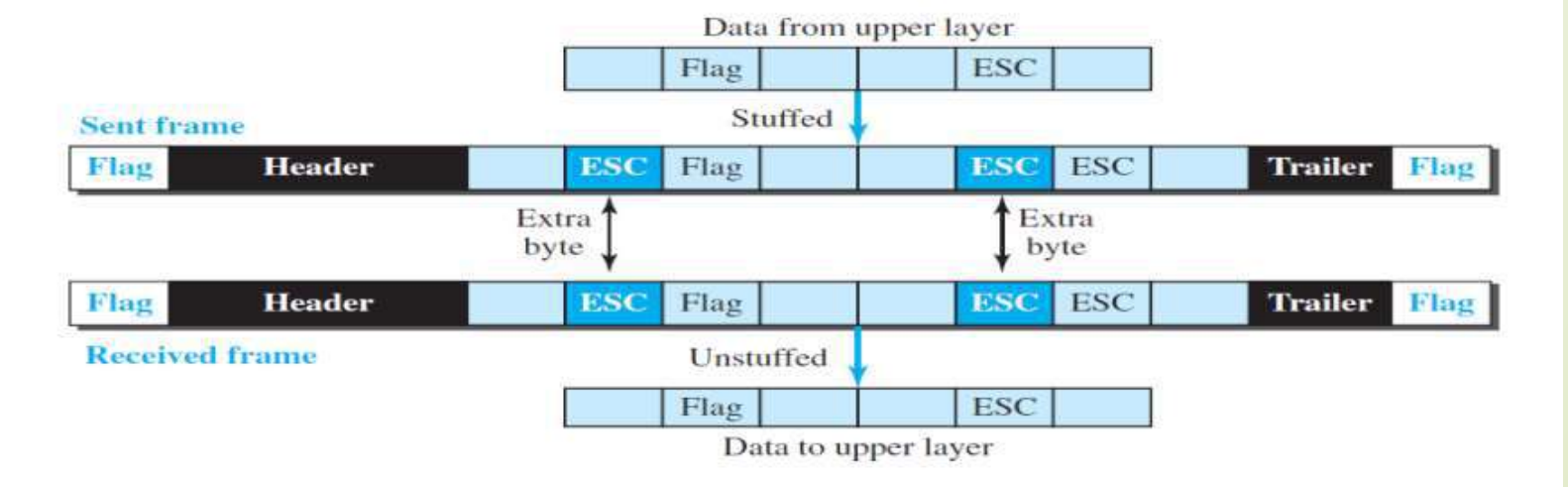


Fig: Byte stuffing and unstuffing

- Whenever the receiver encounters the ESC character, it removes it from the data section and Byte stuffing by the escape character allows the presence of the flag in the data section of the frame, but it creates another problem.

Ques: What happens if the text contains one or more escape characters followed by a byte with the same pattern as the flag?

Ans: The receiver removes the escape character, but keeps the next byte, which is incorrectly interpreted as the end of the frame.

Bit-Oriented Framing

- In bit-oriented framing, the data section of a frame is a sequence of bits to be interpreted by the upper layer as text, graphic, audio, video, and so on.
- However, in addition to headers (and possible trailers), we still need a delimiter to separate one frame from the other. Most protocols use a special 8-bit pattern flag, 01111110, as the delimiter to define the beginning and the end of the frame.

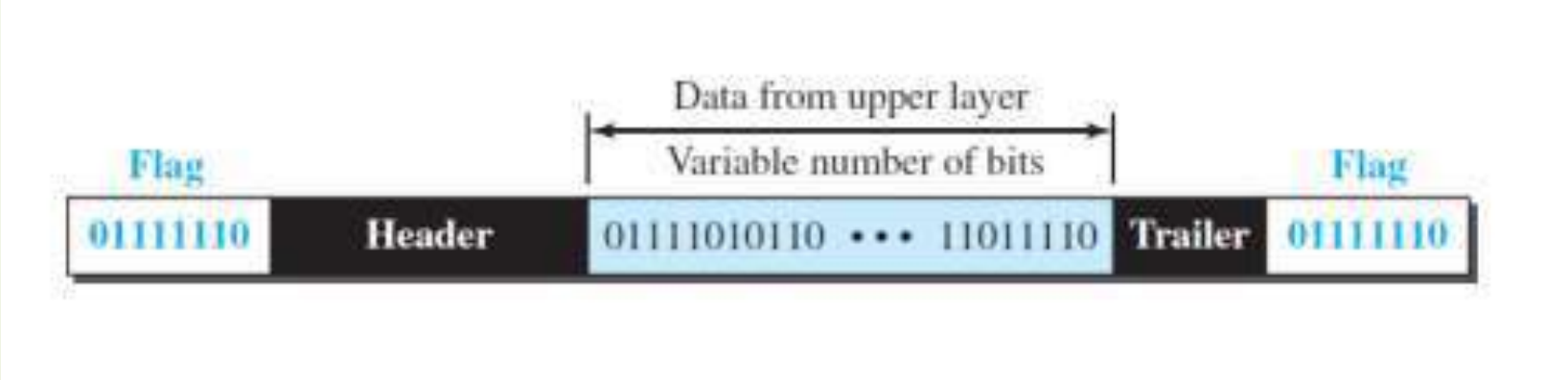


Fig: A frame in bit-oriented protocol

- If the flag pattern appears in the data, need to somehow inform the receiver that this is not the end of the frame.
- We do this by stuffing 1 single bit (instead of 1 byte) to prevent the pattern from looking like a flag.
- The strategy is called bit stuffing. In bit stuffing, if a 0 and five consecutive 1 bits are encountered, an extra 0 is added.
- This extra stuffed bit is eventually removed from the data by the receiver.
- Bit stuffing is the process of adding one extra 0 whenever five consecutive 1s follow a 0 in the data, so that the receiver does not mistake the pattern 0111110 for a flag.

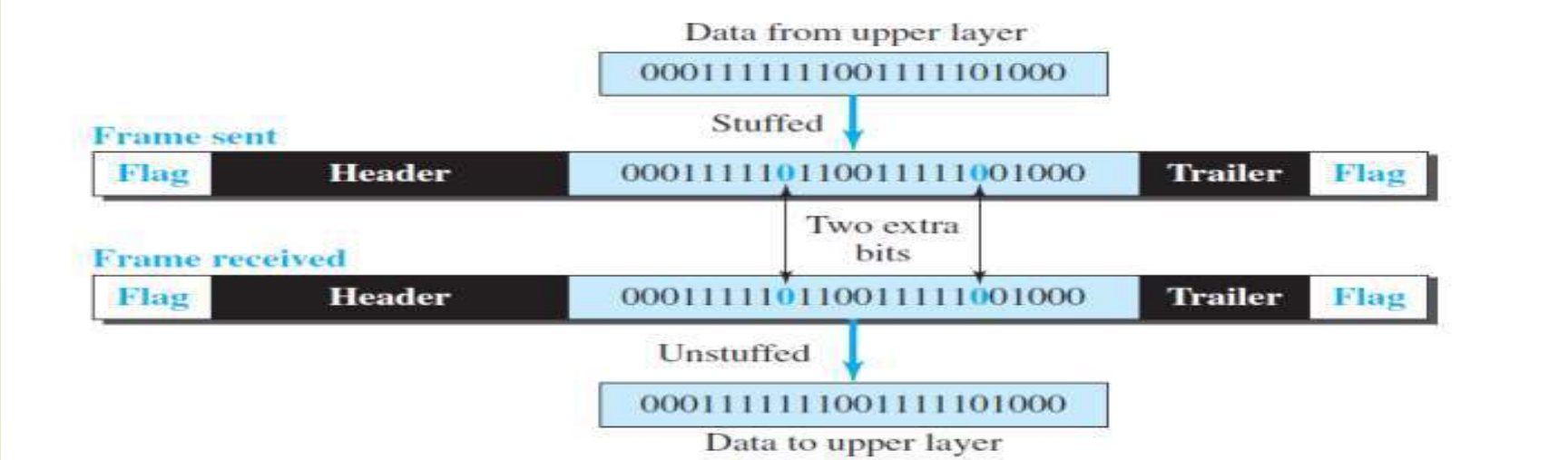


Fig: Bit stuffing and unstuffing

Flow Control

- Whenever an entity produces items and another entity consumes them, there should be a balance between production and consumption rates.
- If the items are produced faster than they can be consumed, the consumer can be overwhelmed and may need to discard some items.
- If the items are produced more slowly than they can be consumed, the consumer must wait, and the system becomes less efficient.

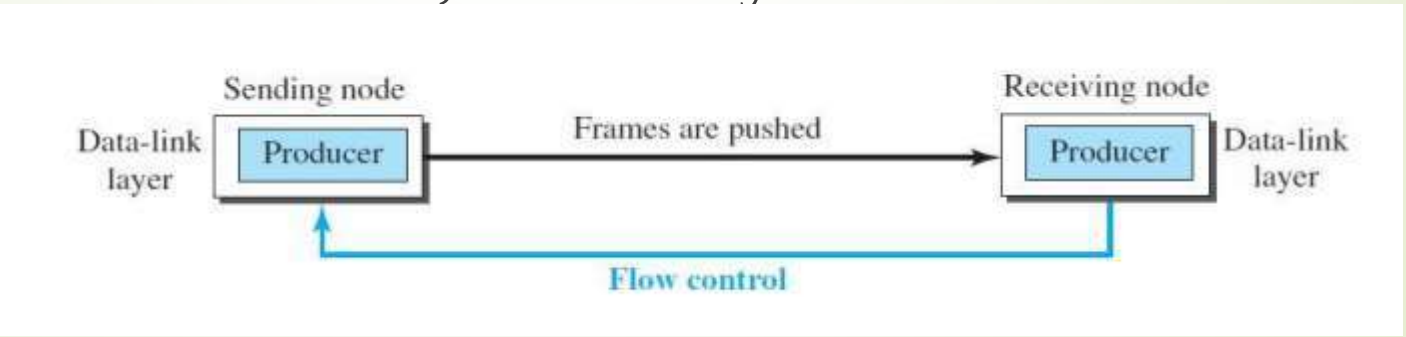


Fig: Flow control at the data link layer

Buffers

- Although flow control can be implemented in several ways, one of the solutions is normally to use two buffers; one at the sending data-link layer and the other at the receiving data-link layer.
- A buffer is a set of memory locations that can hold packets at the sender and receiver.
- The flow control communication can occur by sending signals from the consumer to the producer.
- When the buffer of the receiving data-link layer is full, it informs the sending data- link layer to stop pushing frames.

Error Control

- we need to implement error control at the data-link layer to prevent the receiving node from delivering corrupted packets to its network layer.
- Error control at the data-link layer is normally very simple and implemented using one of the following methods.
- In the method, a CRC is added to the frame header by the sender and checked by the receiver.
- In the first method, if the frame is corrupted, it is silently discarded; if it is not corrupted, the packet is delivered to the network layer. This method is used mostly in wired LANs such as Ethernet.

Combination of Flow and Error Control

- Flow and error control can be combined.
- In a simple situation, the acknowledgment that is sent for flow control can also be used for error control to tell the sender the packet has arrived uncorrupted.
- The lack of acknowledgment means that there is a problem in the sent frame.
- A frame that carries an acknowledgment is normally called an ACK to distinguish it from the data frame.

Connectionless and Connection-Oriented

- A DLC protocol can be either connection less or connection-oriented.

Connectionless Protocol

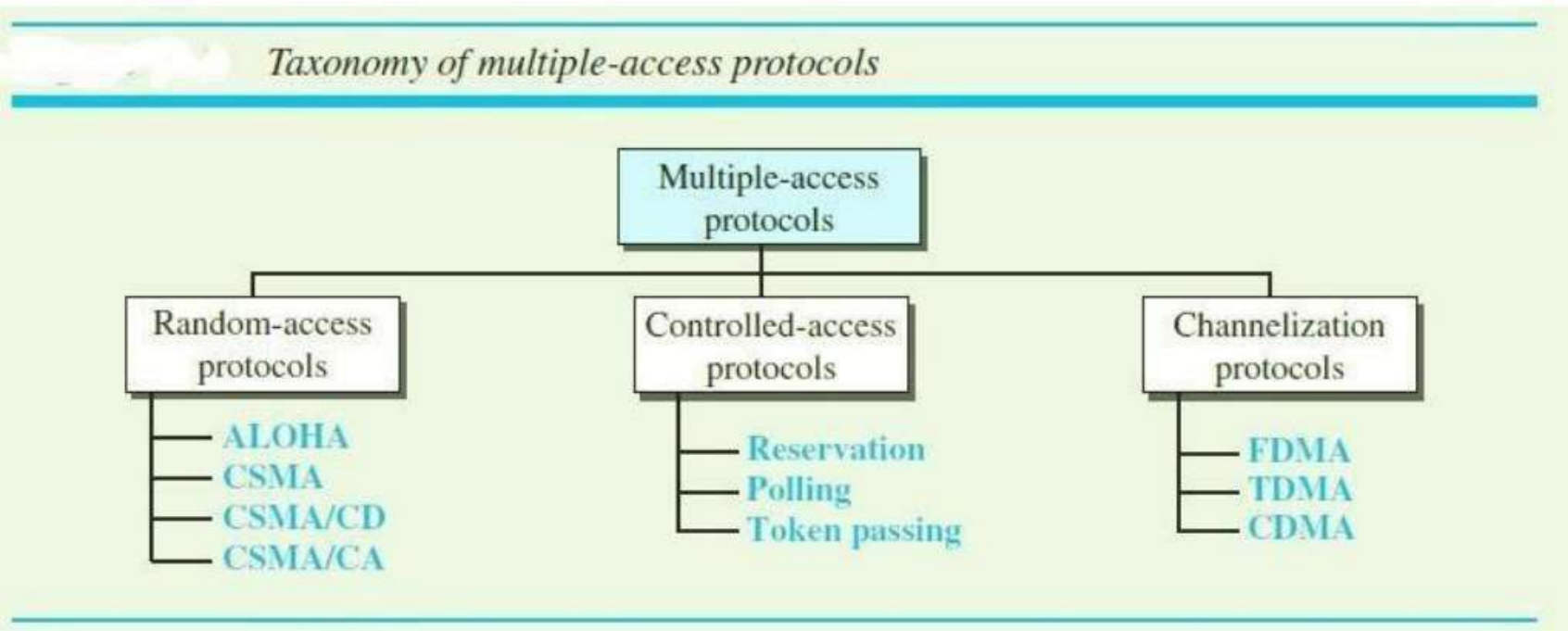
- In a connectionless protocol, frames are sent from one node to the next without any relationship between the frames; each frame is independent.
- Note that the term connectionless here does not mean that there is no physical connection (transmission medium) between the nodes; it means that there is no connection between frames.
- The frames are not numbered and there is no sense of ordering. Most of the data-link protocols for LANs are connectionless protocols.

Connection-Oriented Protocol

- In a connection-oriented protocol, a logical connection should first be established between the two nodes (setup phase).
- After all frames that are somehow related to each other are transmitted (transfer phase), the logical connection is terminated (teardown phase).
- In this type of communication, the frames are numbered and sent in order.
- If they are not received in order, the receiver needs to wait until all frames belonging to the same set are received and then deliver them in order to the network layer.
- Connection oriented protocols are rare in wired LANs, but we can see them in some point-to-point protocols, some wireless LANs, and some WANs.

Media Access Control (MAC)

- 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.
- Many protocols have been devised to handle access to a shared link. All of these protocols belong to a sub layer in the data-link layer called media access control (MAC).



- 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). In other words, each station can transmit when it desires on the condition that it follows the predefined procedure, including testing the state of the medium.
- Two features give this method its name.
  - First, there is no scheduled time for a station to transmit. Transmission is random among the stations. That is why these methods are called random access.
  - Second, no rules specify which station should send next. Stations compete with one another to access the medium. That is why these methods are also called contention methods.

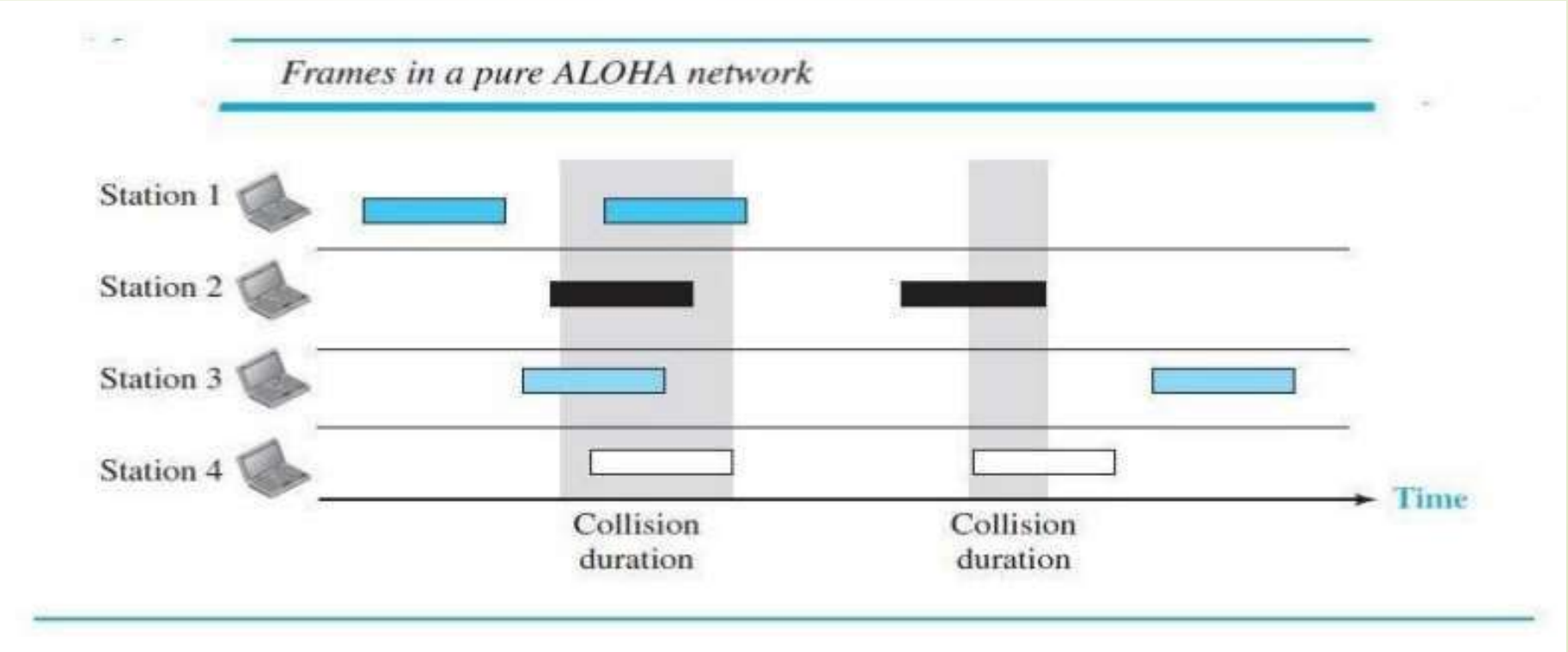
- In a random-access method, each station has the right to the medium without being controlled by any other station. However, if more than one station tries to send, there is an access conflict “collision” and the frames will be either destroyed or modified.
- The random-access methods have evolved from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

ALOHA

- ALOHA, the earliest random access method, was developed at the University of Hawaii in early 1970. It was designed for a radio (wireless) LAN, but it can be used on any shared medium.
- 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.

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 (multiple access) there is only one channel to share, there is the possibility of collision between frames from different stations.

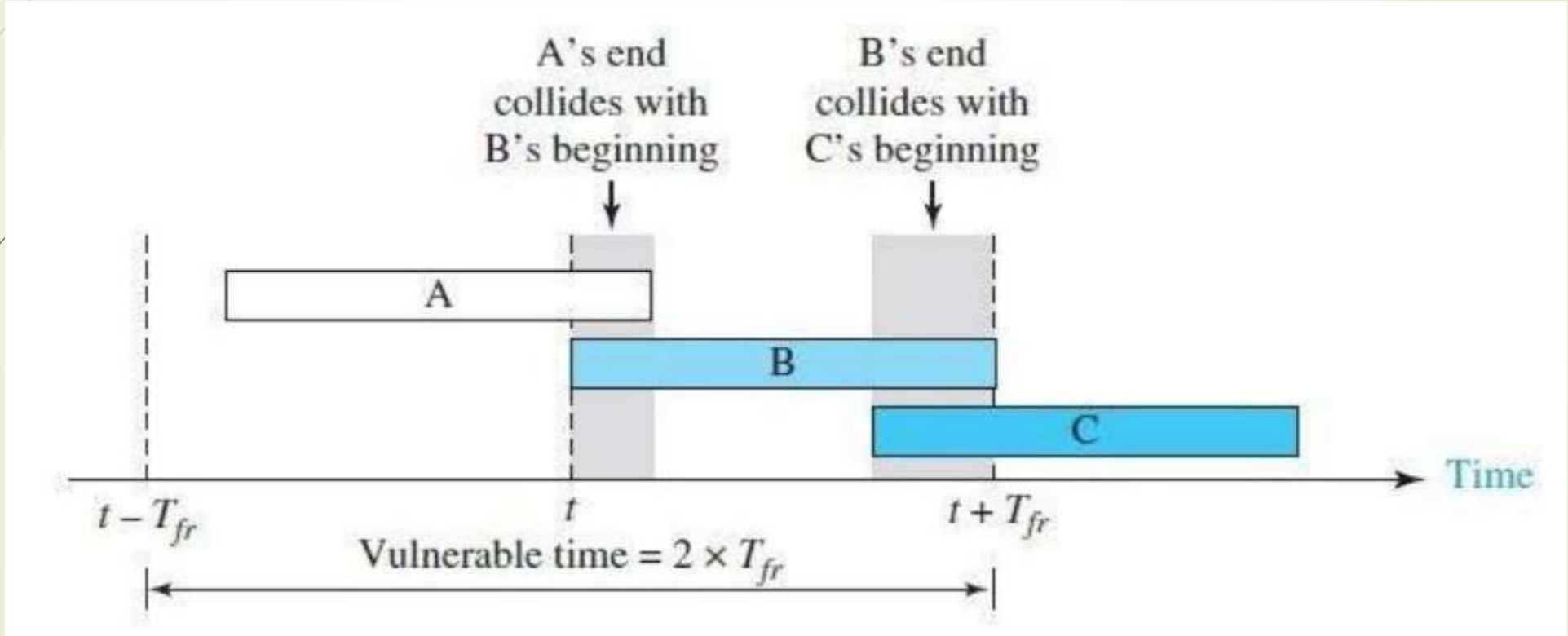


- There are four stations (unrealistic assumption) that contend with one another for access to the shared channel. The above figure shows that each station sends two frames, there are a total of eight frames on the shared medium.
- Some of these frames collide because multiple frames are in contention for the shared channel.
- If one bit of a frame coexists on the channel with one bit from another frame, there is a collision and both will be destroyed. It is obvious that the frames have to be resend that have been destroyed during transmission.
- The pure ALOHA protocol relies on acknowledgments from the receiver. When a station sends a frame, it expects the receiver to send an acknowledgment. If the acknowledgment does not arrive after a time-out period, the station assumes that the frame (or the acknowledgment) has been destroyed and resends the frame.
- A collision involves two or more stations. If all these stations try to resend their frames after the time-out, the frames will collide again

- Pure ALOHA dictates that when the time-out period passes, each station waits a random amount of time before resending its frame. The randomness will help avoid more collisions. This time is called as the back off time TB.
- Pure ALOHA has a second method to prevent congesting the channel with retransmitted frames. After a maximum number of retransmission attempts K_{max}, a station must give up and try later.
- The time-out period is equal to the maximum possible round-trip propagation delay, which is twice the amount of time required to send a frame between the two most widely separated stations ($2 \times Tp$).
- The backoff time TB is a random value that normally depends on K (the number of attempted unsuccessful transmissions).
- In this method, for each retransmission, a multiplier R = 0 to 2 K is randomly chosen and multiplied by Tp (maximum propagation time) or T_{fr} (the average time required to send out a frame) to find TB

Vulnerable time

The length of time in which there is a possibility of collision. The stations send fixed-length frames with each frame taking T_{fr} seconds to send.



- Station B starts to send a frame at time t. Imagine station A has started to send its frame after t − Tfr . This leads to a collision between the frames from station B and station A.
- On the other hand, suppose that station C starts to send a frame before time t + Tfr. There is also a collision between frames from station B and station C.

PROBLEM: 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 Tfr is 200 bits/200 kbps or 1 ms. The vulnerable time is 2 × 1 ms = 2 ms. This means no station should send later than 1 ms before this station starts transmission and no station should start sending during the period (1 ms) that this station is sending.

Throughput

- G = the average number of frames generated by the system during one frame transmission time (T_{fr})
- S= the average number of successfully transmitted frames for pure ALOHA. And is given by,

$$S = G \times e^{-2G}. \text{ ----------(1)}$$

- Differentiate equation (1) with respect to G and equate it to 0, we get G = 1/2. Substitute G=1/2 in equation (1) to get Smax.

- The maximum throughput Smax = 0.184.
- If one-half a frame is generated during one frame transmission time (one frame during two frame transmission times), then 18.4 percent of these frames reach their destination successfully.
- G is set to G = 1/2 to produce the maximum throughput because the vulnerable time is 2 times the frame transmission time. Therefore, if a station generates only one frame in this vulnerable time (and no other stations generate a frame during this time), the frame will reach its destination successfully.

NOTE:

- The throughput for pure ALOHA is $S = G \times e^{-2G}$.
- The maximum throughput Smax = 1/(2e) = 0.184 when G = (1/2).

PROBLEM

- 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 T_{fr} is 200/200 kbps or 1 ms.

  (a) If the system creates 1000 frames per second, or 1 frame per millisecond (1s = 1000ms) then G = 1 (because G= number of frames generated for one T_{fr}).

  S = G × e^{-2G} = 0.135 (13.5 percent). This means that the throughput is

  1000×0.135 = 135 frames. Only 135 frames out of 1000 will probably survive.

(b) If the system creates 500 frames per second, or 1/2 frame per millisecond (1s = 1000ms) then G = 1/2 (because G= number of frames generated for one T_{fr}).

S = G × e^{-2G} = 0.184 (18.4 percent). This means that the throughput is 500 ×0.184 = 92 frames. Only 92 frames out of 500 will probably survive.

(c) If the system creates 250 frames per second, or 1/4 frame per millisecond (1s = 1000ms) then G = 1/4 (because G= number of frames generated for one T_{fr}).

S = G × e^{-2G} = 0.152 (15.2 percent). This means that the throughput is 250 ×0.152 = 135 frames. Only 38 frames out of 250 will probably survive.

Slotted ALOHA

- Pure ALOHA has a vulnerable time of $2 \times T_{fr}$. This is so because there is no rule that defines when the station can send.
- A station may send soon after another station has started or just before another station has finished. Slotted ALOHA was invented to improve the efficiency of pure ALOHA.
- In slotted ALOHA we divide the time into slots of T_{fr} seconds and force the station to send only at the beginning of the time slot.

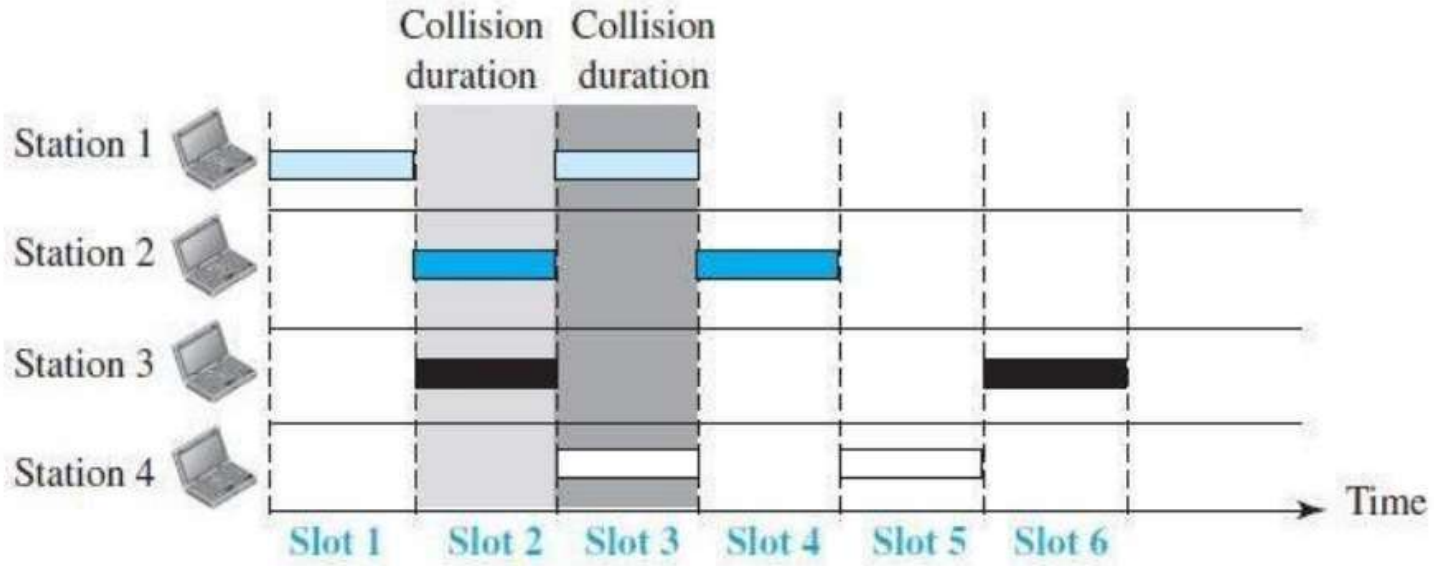


Figure: Frames in Slotted ALOHA network

- A station is allowed to send only at the beginning of the synchronized time slot, if a station misses this moment, it must wait until the beginning of the next time slot.
- This means that the station which started at the beginning of this slot has already finished sending its frame.
- There is still the possibility of collision if two stations try to send at the beginning of the same time slot. However, the vulnerable time is now reduced to one-half, equal to T_{fr}.

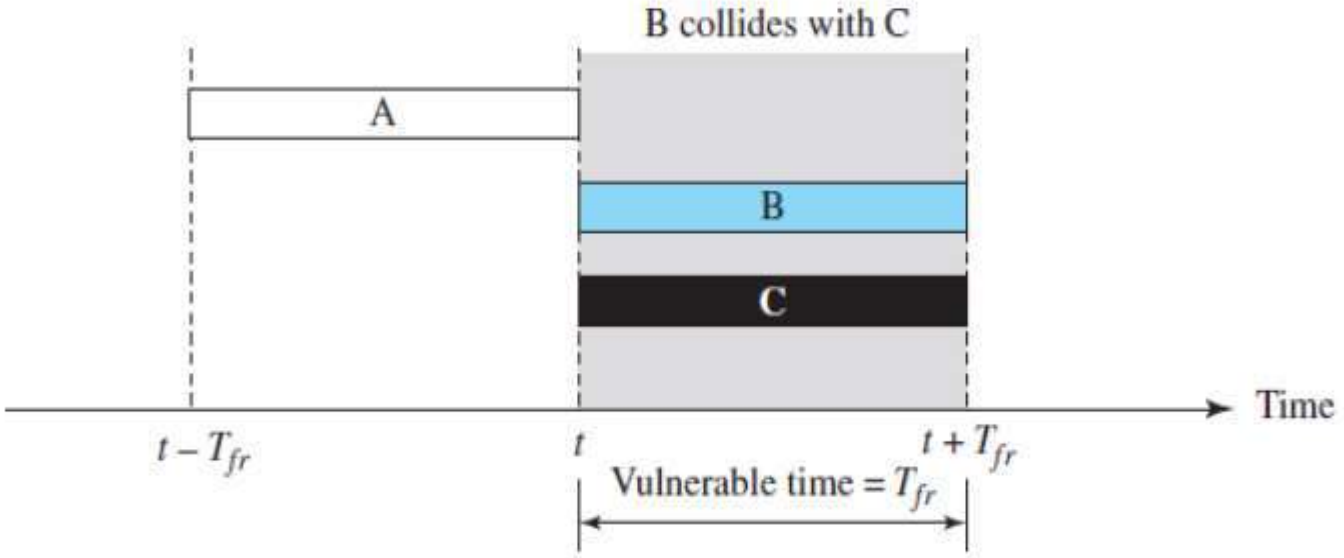


Figure: Vulnerable time for slotted ALOHA protocol

Throughput

- G = the average number of frames generated by the system during one frame transmission time (T_{fr})
- S= the average number of successfully transmitted frames for Slotted ALOHA.
- And is given by, $S = G \times e^{-G}$. ----------(1)
- Differentiate equation (1) with respect to G and equate it to 0, we get G = 1. Substitute G=1 in equation (1) to get Smax.
- The maximum throughput Smax = 0.368

- If one frame is generated during one frame transmission time then 36.8 percent of these frames reach their destination successfully.
- G is set to G = 1 to produce the maximum throughput because the vulnerable time is equal to the frame transmission time. Therefore, if a station generates only one frame in this vulnerable time (and no other stations generate a frame during this time), the frame will reach its destination successfully.

NOTE:

- The throughput for Slotted ALOHA is S = G × e^{-G} .
- The maximum throughput Smax = 1/(e) = 0.368 when G = 1.

PROBLEM

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

a. 1000 frames per second?

b. 500 frames per second?

c. 250 frames per second?

Solution: The frame transmission time T_{fr} is 200/200 kbps or 1 ms.

(a) If the system creates 1000 frames per second, or 1 frame per millisecond (1s = 1000ms) then G = 1 (because G= number of frames generated for one T_{fr}). $S = G \times e^{-G}$ = 0.368 (36.8 percent). This means that the throughput is 1000 ×0.368 = 368 frames. Only 368 frames out of 1000 will probably survive.

(b) If the system creates 500 frames per second, or 1/2 frame per millisecond (1s = 1000ms) then G = 1/2 (because G= number of frames generated for one Tfr).

$S = G \times e^{-G}$ = 0.303 (30.3 percent). This means that the throughput is 500 ×0.303 = 151 frames. Only 151 frames out of 500 will probably survive.

(c) If the system creates 250 frames per second, or 1/4 frame per millisecond (1s = 1000ms) then G = 1/4 (because G= number of frames generated for one Tfr).

$S = G \times e^{-G}$ = 0.195 (19.5 percent). This means that the throughput is 250 ×0.195 = 49 frames. Only 49 frames out of 250 will probably survive.

CSMA

- To minimize the chance of collision and, therefore, increase the performance, the CSMA method was developed. The chance of collision can be reduced if a station senses the medium before trying to use it.
- Carrier sense multiple access (CSMA) requires that each station first listen to the medium (or check the state of the medium) before sending.
- 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 reason for this is shown in Figure, a space and time model of a CSMA network. Stations are connected to a shared channel.

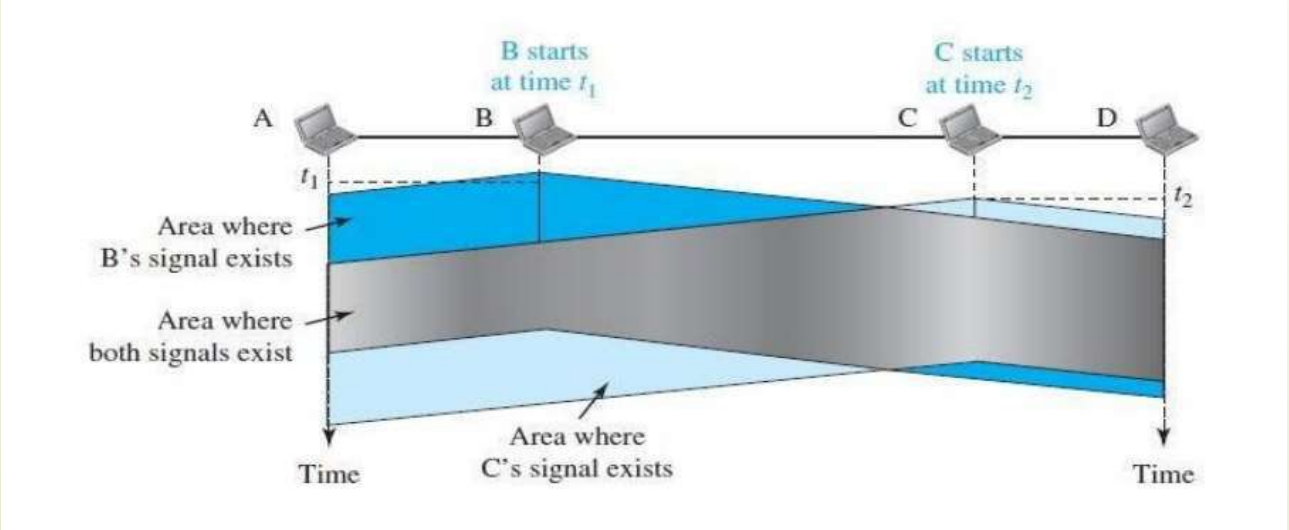
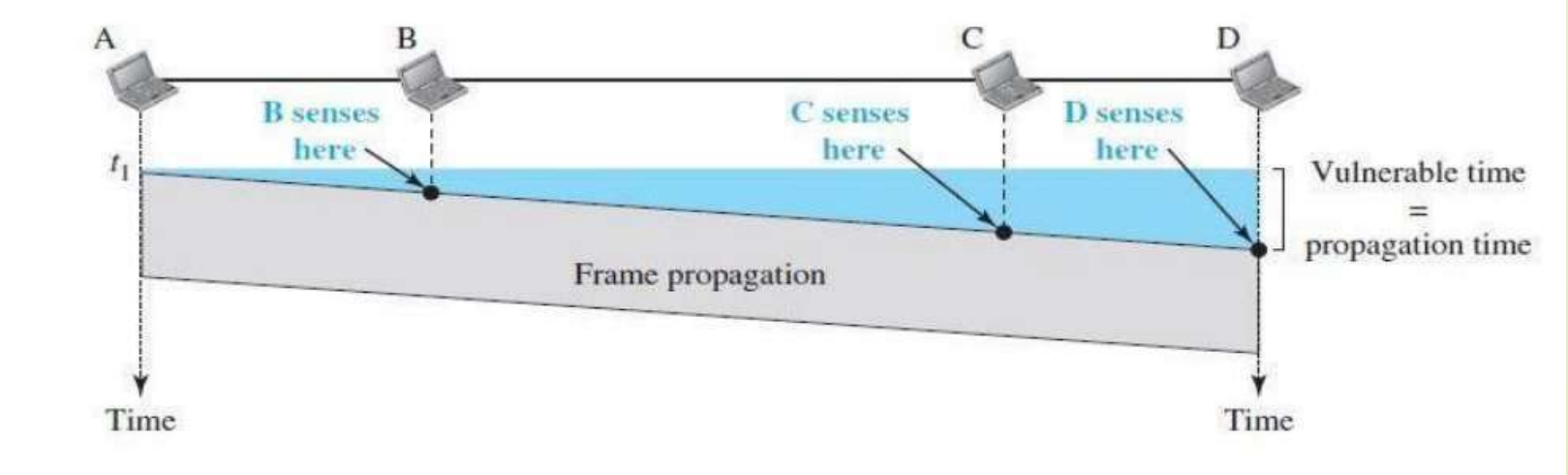


Figure: Space/time model of a collision in CSMA

- The possibility of collision still exists because of propagation delay; when a station sends a frame, it still takes time (although very short) for the first bit to reach every station and for every station to sense it.
- A station may sense the medium and find it idle, only because the first bit sent by another station has not yet been received.
- At time t1, station B senses the medium and finds it idle, so it sends a frame. At time t2 (t2 > t1), station C senses the medium and finds it idle because, at this time, the first bits from station B have not reached station C. Station C also sends a frame. The two signals collide and both frames are destroyed.

Vulnerable Time

- The vulnerable time for CSMA is the propagation time Tp. This is the time needed for a signal to propagate from one end of the medium to the other.
- When a station sends a frame and any other station tries to send a frame during this time, a collision will result.
- But if the first bit of the frame reaches the end of the medium, every station will already have heard the bit and will refrain from sending.
- Figure below shows the worst case. The leftmost station, A, sends a frame at time t1, which reaches the rightmost station, D, at time t1 + Tp. The gray area shows the vulnerable area in time and space.



Persistence Methods

Persistence method is developed to determine what the station has to do whenever it encounters the channel is idle or busy. There are 3 persistent methods

1. 1-persistent method
2. Non persistent method, and
3. p-Persistent method.

1-Persistent

- The 1-persistent method is simple and straightforward.
- After the station finds the line idle, it sends its frame immediately (with probability 1).
- This method has the highest chance of collision because two or more stations may find the line idle and send their frames immediately

Non persistent

- In the non persistent method, a station that has a frame to send senses the line. If the line is idle, it sends immediately. If the line is not idle, it waits a random amount of time and then senses the line again.
- The non persistent approach reduces the chance of collision because it is unlikely that two or more stations will wait the same amount of time and retry to send simultaneously.
- This method reduces the efficiency of the network because the medium remains idle when there may be stations with frames to send.

p-Persistent

- The p-persistent method is used if the channel has time slots with a slot duration equal to or greater than the maximum propagation time.
- The p-persistent approach combines the advantages of the other two strategies. It reduces the chance of collision and improves efficiency. In this method, after the station finds the line idle it follows these steps:

1. With probability p, the station sends its frame.
2. With probability q = 1 − p, the station waits for the beginning of the next time slot and checks the line again.

(a) If the line is idle, it goes to step 1.

(b) If the line is busy, it acts as though a collision has occurred and uses the backoff procedure.

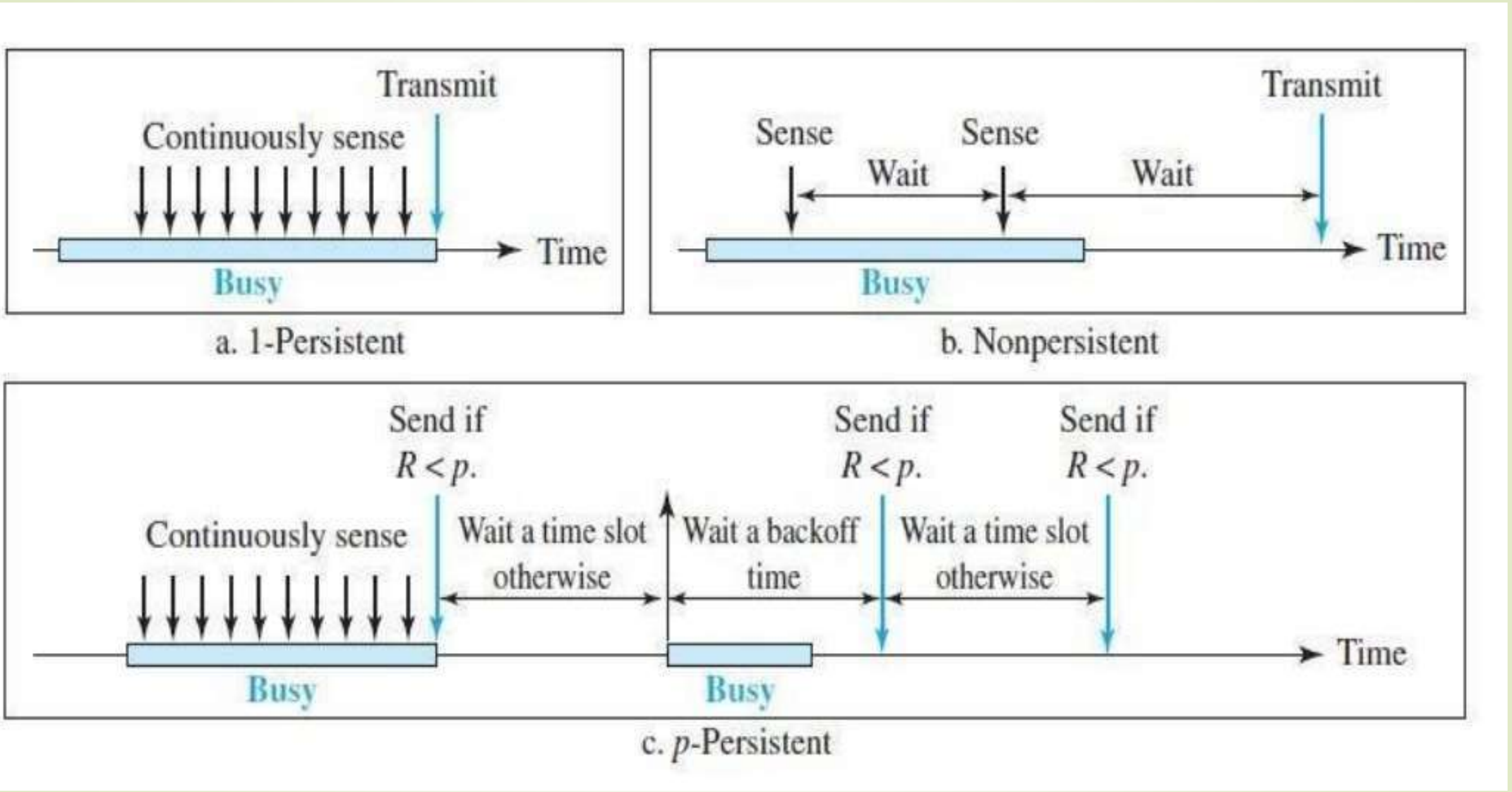


Figure: Behaviour of three persistence methods

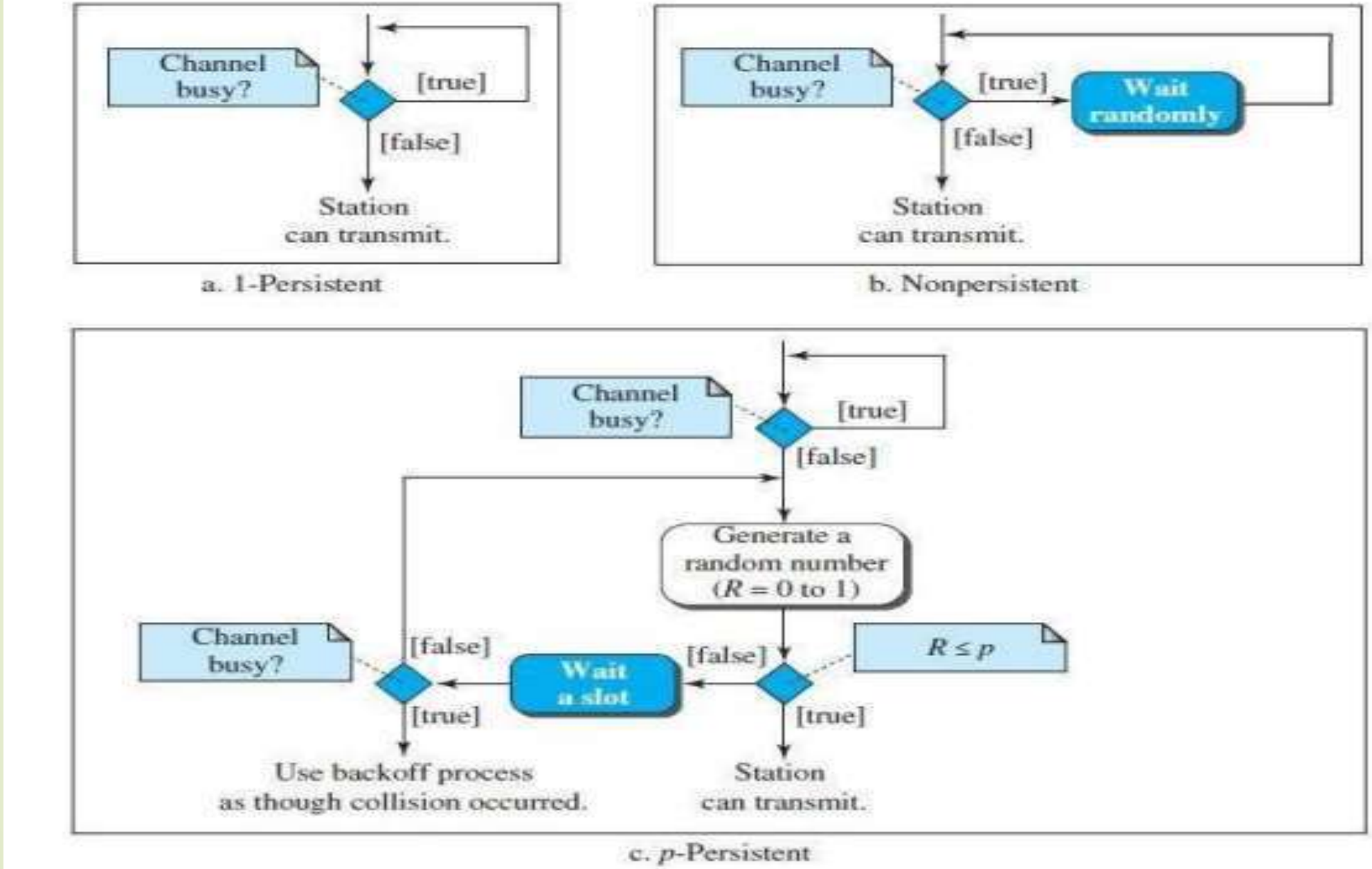


Figure: Flow diagram for three persistence methods

CSMA/CD

- The CSMA method does not specify the procedure following a collision. Carrier sense multiple access with collision detection (CSMA/CD) augments the algorithm to handle the collision.
- Station monitors the medium after it sends a frame to see if the transmission was successful.
- The first bits transmitted by the two stations involved in the collision. Although each station continues to send bits in the frame until it detects the collision.

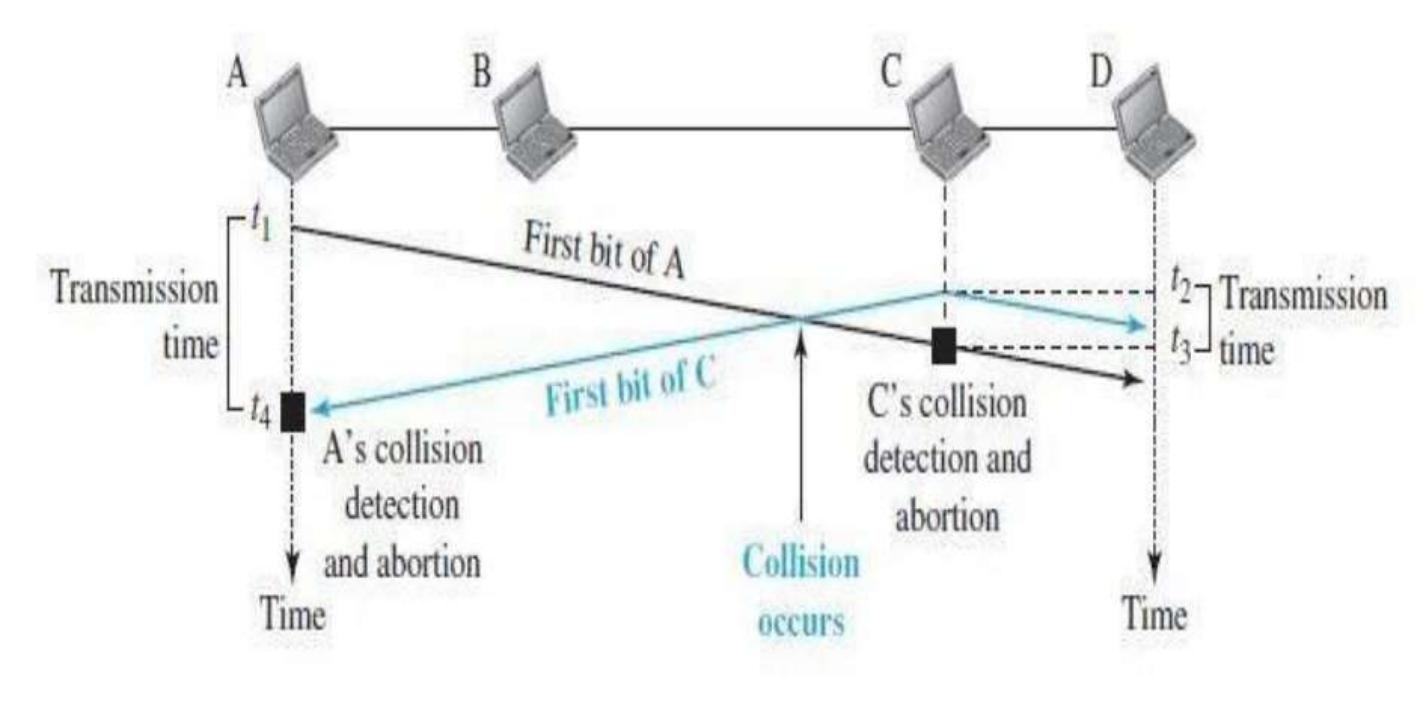


Figure: Collision of the first bits in CSMA/CD

- At time t1, station A has executed its persistence procedure and starts sending the bits of its frame. At time t2, station C has not yet sensed the first bit sent by A.
- Station C executes its persistence procedure and starts sending the bits in its frame, which propagate both to the left and to the right.
- The collision occurs sometime after time t2. Station C detects a collision at time t3 when it receives the first bit of A's frame. Station C immediately aborts transmission.
- Station A detects collision at time t4 when it receives the first bit of C's frame, it also immediately aborts transmission.

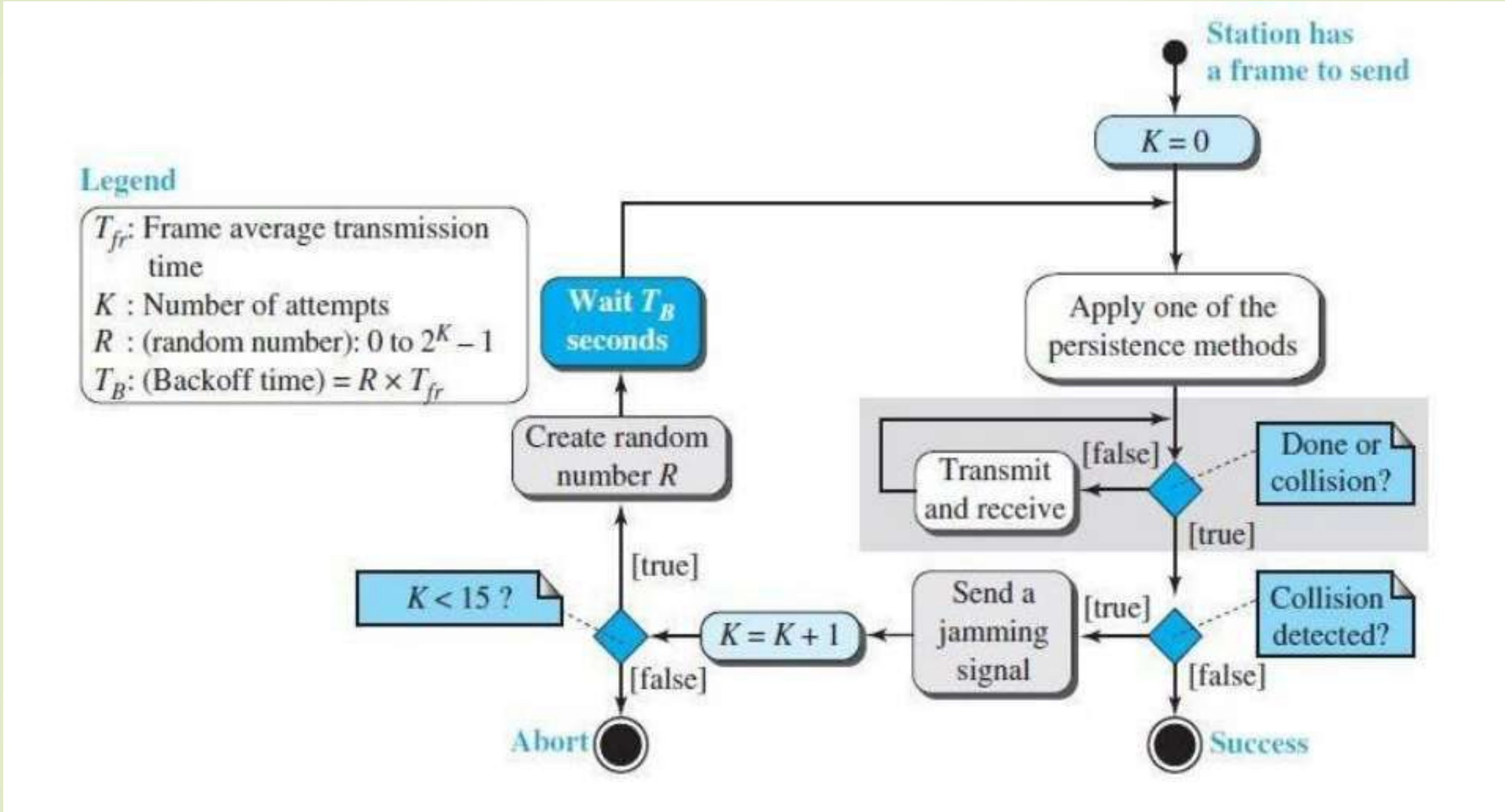


Figure: Flow diagram for the CSMA/CD

The flow diagram for CSMA/CD is as shown in Figure. It is similar to the one for the ALOHA protocol, but there are differences.

1. The first difference is the addition of the persistence process. It is required to sense the channel before sending the frame by using one of the persistence processes (non persistent, 1 persistent, or p-persistent).
2. The second difference is the frame transmission. In ALOHA, there is transmission of the entire frame and then wait for an acknowledgment. In CSMA/CD, transmission and collision detection are continuous processes.

- It is not like the entire frame is sent and then look for a collision. The station transmits and receives continuously and simultaneously (using two different ports or a bidirectional port).

- Loop is used to show that transmission is a continuous process. It is constantly monitored in order to detect one of two conditions: either transmission is finished or a collision is detected.
- Either event stops transmission. When it comes 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.

3. The third difference is the sending of a short jamming signal to make sure that all other stations become aware of the collision.

PROBLEM:

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) is 25.6 µs, what is the minimum size of the frame?

Solution:

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

Throughput

- The throughput of CSMA/CD is greater than that of pure or slotted ALOHA.
- The maximum throughput occurs at a different value of G and is based on the persistence method and the value of p in the p-persistent approach.
- For the 1-persistent method, the maximum throughput is around 50 percent when G = 1. For the non persistent method, the maximum throughput can go up to 90 percent when G is between 3 and 8.

CSMA/CA

- Carrier sense multiple access with collision avoidance (CSMA/CA) was invented for wireless networks.
- Collisions are avoided through the use of CSMA/CA's three strategies: the inter frame space, the contention window, and acknowledgments.

Inter frame Space (IFS):

- When an idle channel is found, the station does not send immediately. It waits for a period of time called the inter frame space or IFS.
- Even though the channel may appear idle when it is sensed, a distant station may have already started transmitting.
- The distant station's signal has not yet reached this station. The IFS time allows the front of the transmitted signal by the distant station to reach this station.
- After waiting an IFS time, if the channel is still idle, the station can send, but it still needs to wait a time equal to the contention window

Contention Window

- The contention window is an amount of time divided into slots. A station that is ready to send chooses a random number of slots as its wait time.
- The number of slots in the window changes according to the binary exponential back off strategy. This means that it is set to one slot the first time and then doubles each time the station cannot detect an idle channel after the IFS time.
- This is very similar to the p-persistent method except that a random outcome defines the number of slots taken by the waiting station.
- One interesting point about the contention window is that the station needs to sense the channel after each time slot.

Acknowledgement

- Even with all the precautions considered, there still may be a collision resulting in destroyed data. In addition, the data may be corrupted during the transmission.
- The positive acknowledgment and the time-out timer can help guarantee that the receiver has received the frame.

CONTROLLED ACCESS

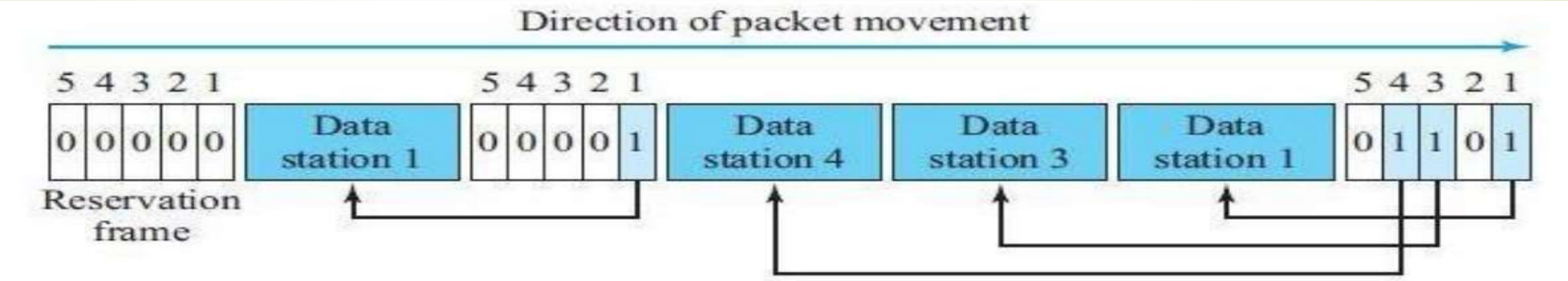
In controlled access, the stations consult one another to find which station has the right to send. A station cannot send unless it has been authorized by other stations.

There are three controlled access methods,

1. Reservation.
2. Polling.
3. Token passing.

1. Reservation.

In the reservation method, a station needs to make a reservation before sending data.



- Time is divided into intervals. In each interval, a reservation frame precedes the data frames sent in that interval.
- If there are N stations in the system, there are exactly N reservation mini slots in the reservation frame. Each mini slot belongs to a station. When a station needs to send a data frame, it makes a reservation in its own mini slot. The stations that have made reservations can send their data frames after the reservation frame.
- Above Figure shows a situation with five stations and a five-mini slot reservation frame. In the first interval, only stations 1, 3, and 4 have made reservations. In the second interval, only station 1 has made a reservation.

2. Polling

- Polling works with topologies in which one device is designated as a primary station and the other devices are secondary stations.
- All data exchanges must be made through the primary device even when the ultimate destination is a secondary device. The primary device controls the link; the secondary devices follow its instructions.
- The primary device determines which device is allowed to use the channel at a given time. The primary device, therefore, is always the initiator of a session.
- This method uses poll and select functions to prevent collisions. However, the drawback is if the primary station fails, the system goes down.

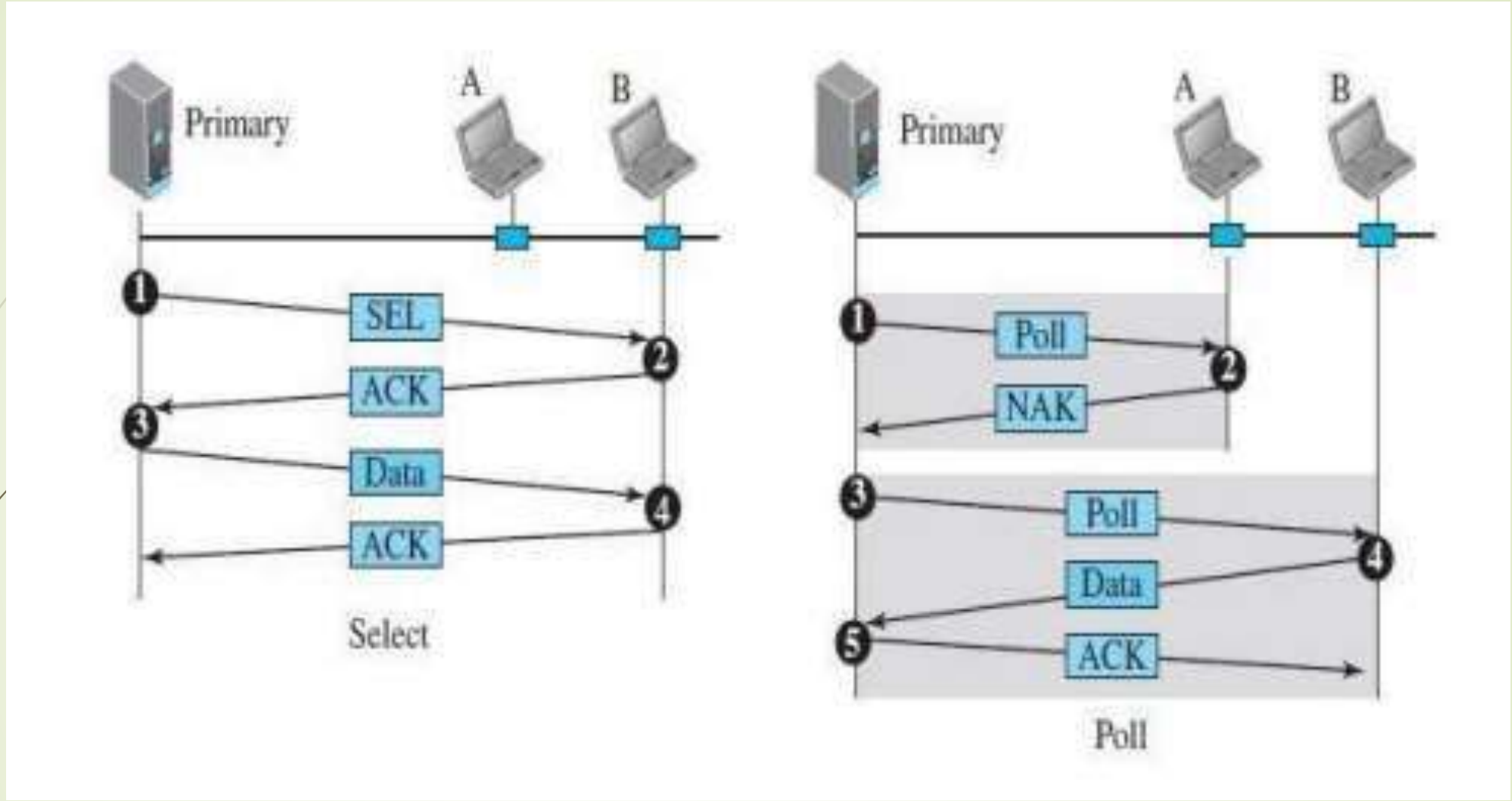


Figure: Select and poll functions in polling-access method

Select

- The select function is used whenever the primary device has something to send. Since the primary controls the link. If it is neither sending nor receiving data, it knows the link is available.
- If it has something to send, the primary device sends it. The primary station has to confirm whether the target device is prepared to receive.
- The primary must alert the secondary to the upcoming transmission and wait for an acknowledgment of the secondary's ready status. Before sending data, the primary creates and transmits a select (SEL) frame, one field of which includes the address of the intended secondary.

Poll

- The poll function is used by the primary device to solicit transmissions from the secondary devices
- When the primary is ready to receive data, it must ask (poll) each device in turn if it has anything to send. When the first secondary is approached, it responds either with a NAK frame if it has nothing to send or with data (in the form of a data frame) if it does.
- If the response is negative (a NAK frame), then the primary polls the next secondary in the same manner until it finds one with data to send.
- When the response is positive (a data frame), the primary reads the frame and returns an acknowledgment (ACK frame), verifying its receipt poll function is used by the primary device.

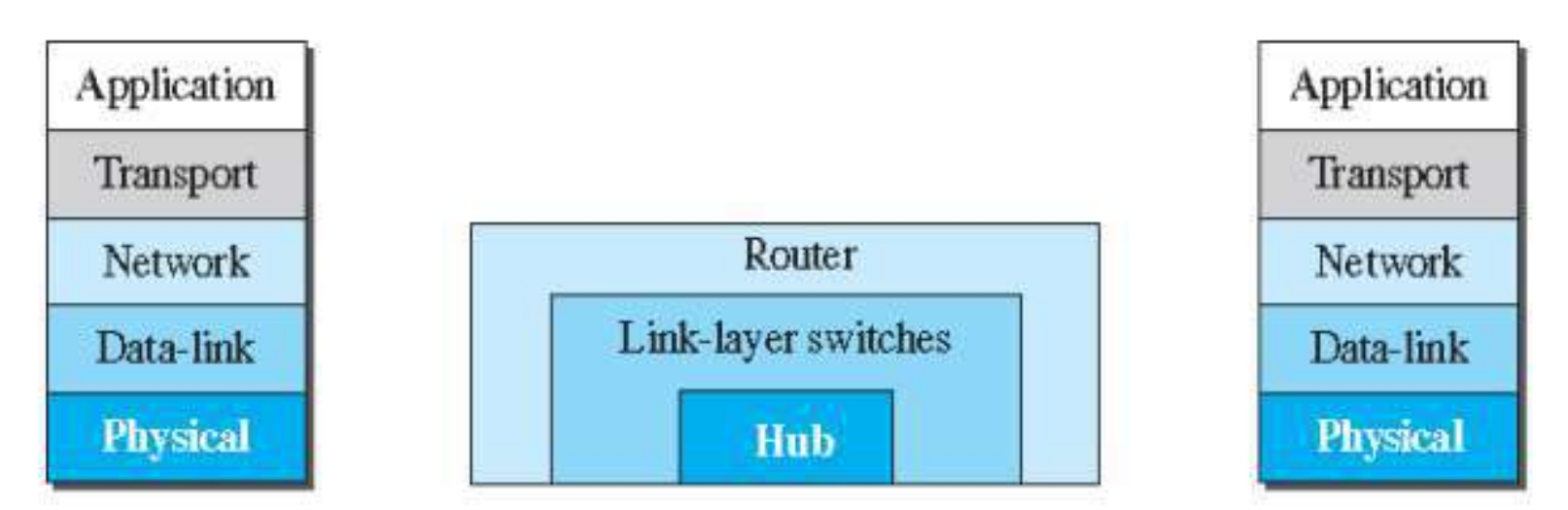
3. Token Passing

- In the token-passing method, the stations in a network are organized in a logical ring. For each station, there is a predecessor and a successor.
- The predecessor is the station which is logically before the station in the ring; the successor is the station which is after the station in the ring. The current station is the one that is accessing the channel now.
- The right to this access has been passed from the predecessor to the current station. The right will be passed to the successor when the current station has no more data to send.
- In this method, a special packet called a token circulates through the ring. The possession of the token gives the station the right to access the channel and send its data.
- When a station has some data to send, it waits until it receives the token from its predecessor. It then holds the token and sends its data.

- When the station has no more data to send, it releases the token, passing it to the next logical station in the ring.
- The station cannot send data until it receives the token again in the next round. In this process, when a station receives the token and has no data to send, it just passes the data to the next station.
- Token management is needed for this access method. Stations must be limited in the time they can have possession of the token.
- The token must be monitored to ensure it has not been lost or destroyed. For example, if a station that is holding the token fails, the token will disappear from the network.
- Another function of token management is to assign priorities to the stations and to the types of data being transmitted.
- And finally, token management is needed to make low-priority stations release the token to high-priority stations.

CONNECTING DEVICES

- We use connecting devices to connect hosts together to make a network or to connect networks together to make an internet.
- Connecting devices can operate in different layers of the Internet model.
- There are 3 kinds of connecting devices: 1. Hubs, 2. Link-layer switches, and 3.Routers.



Hubs

- A hub is a device that operates only in the physical layer.
- Signals that carry information within a network can travel a fixed distance before attenuation.
- A repeater receives a signal and, before it becomes too weak or corrupted, regenerates and retimes the original bit pattern and then sends the refreshed signal.
- Hub can be used to serve as the connecting point and at the same time function as a repeater.
- When a packet from station A to station B arrives at the hub, the hub forwards the packet to all outgoing ports except the one from which the signal was received.

- Today, Ethernet LANs use star topology.
- In a star topology, a repeater is a multipoint device, often called a hub, that can be used to serve as the connecting point and at the same time function as a repeater.
- A hub or repeater is a physical layer device. They do not have a link-layer address and they do no check the link layer address of the received frame. They just regenerates the corrupted bits and send them out from every port.

Link-Layer Switches

- A link-layer switch (or switch) operates in both the physical and the data-link layers. As a physical- layer device, it regenerates the signal it receives.
- As a link-layer device, the link-layer switch can check the MAC addresses (source and destination) contained in the frame.
- What is the difference in functionality is between a link-layer switch and a hub.?
- Answer: A link-layer switch has filtering capability.
- A Link layer switch can check the destination address of a frame and can decide from which outgoing port the frame should be sent.

Filtering

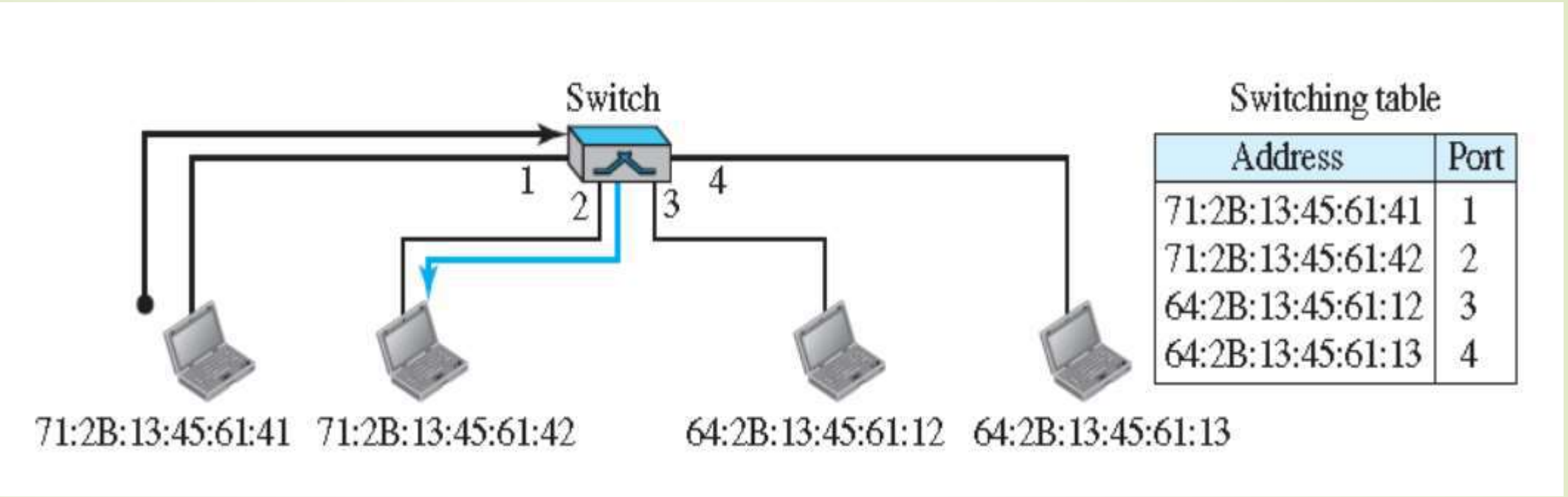


Figure: Link Layer Switch

- If a frame destined for station 71:2B:13:45:61:42 arrives at port 1,
- The link-layer switch consults its table to find the departing port.
- According to its table, frames for 71:2B:13:45:61:42 should be sent out only through port 2;
- Therefore, there is no need for forwarding the frame through other ports.

Transparent Switches

- A transparent switch is a switch in which the stations are completely unaware of the switch’s existence.
- If a switch is added or deleted from the system, reconfiguration of the stations is unnecessary.
- According to the IEEE 802.1d specification, a system equipped with transparent switches must meet three criteria:
- Frames must be forwarded from one station to another.
- The forwarding table is automatically made by learning frame movements in the network.
- Loops in the system must be prevented.

Learning

- The earliest switches had switching tables that were static. The system administrator would manually enter each table entry during switch setup.
- Although the process was simple, it was not practical. If a station was added or deleted, the table had to be modified manually.
- A better solution to the static table is a dynamic table that maps addresses to ports (interfaces) automatically.
- To make a table dynamic, we need a switch that gradually learns from the frames movements.
- To do this, the switch inspects both the destination and the source addresses in each frame that passes through the switch.
- The destination address is used for the forwarding decision (table lookup);the source address is used for adding entries to the table and for updating purposes.

Figure 17.4 *Learning switch*

Gradual building of table

Address	Port

a. Original

Address	Port
71:2B:13:45:61:41	1

b. After A sends a frame to D

Address	Port
71:2B:13:45:61:41	1
64:2B:13:45:61:13	4

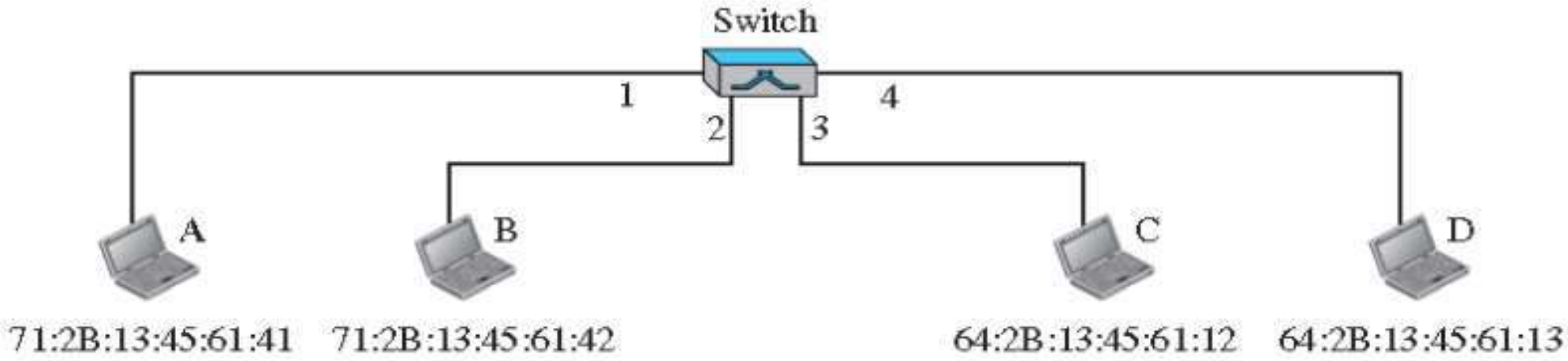
c. After D sends a frame to B

Address	Port
71:2B:13:45:61:41	1
64:2B:13:45:61:13	4
71:2B:13:45:61:42	2

d. After B sends a frame to A

Address	Port
71:2B:13:45:61:41	1
64:2B:13:45:61:13	4
71:2B:13:45:61:42	2
64:2B:13:45:61:12	3

e. After C sends a frame to D

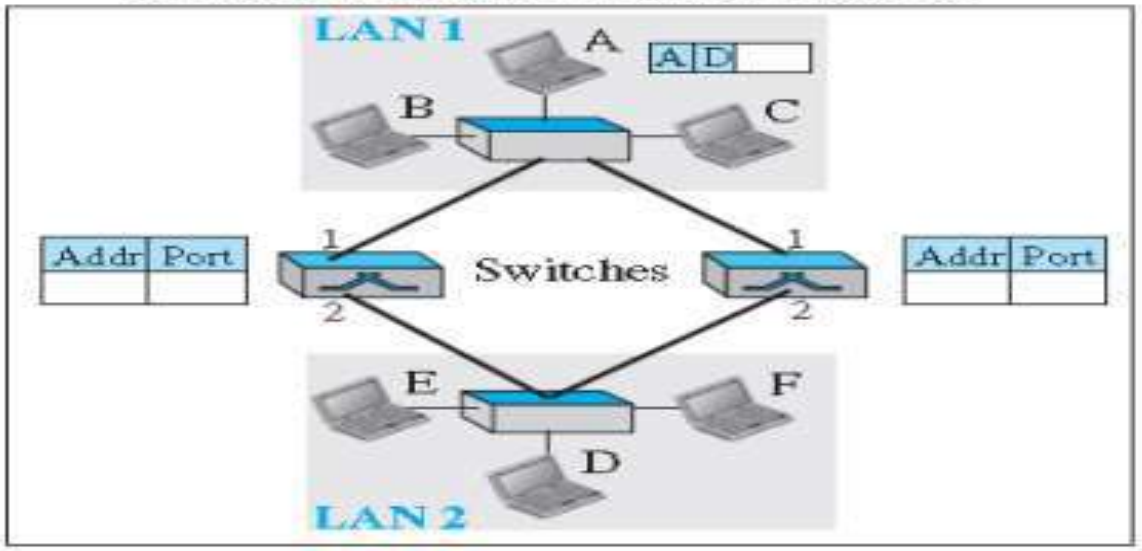


- When station A sends a frame to station D, the frame goes out from all three ports; the frame floods the network.
- However, by looking at the source address, the switch learns that station A must be connected to port1.
- This means that frames destined for A, in the future, must be sent out through port 1.
- The switch adds this entry to its table.
- When station D sends a frame to station B, the switch has no entry for B, it adds one more entry to the table related to station D.
- The learning process continues until the table has information about every port.
- However, the learning process may take a long time. For example, if a station does not send out a frame (a rare situation), the station will never have an entry in the table

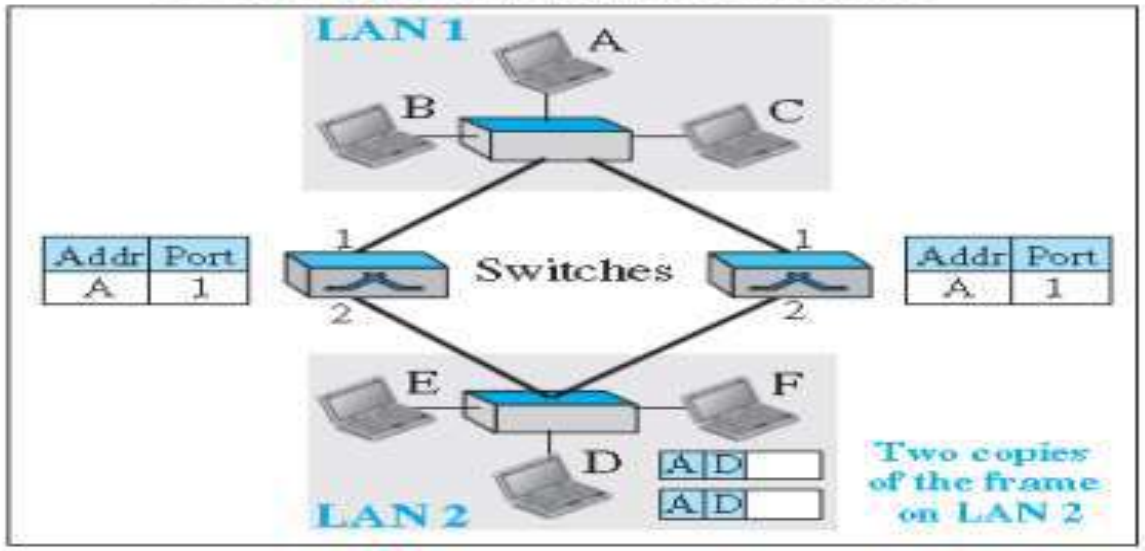
Loop Problem

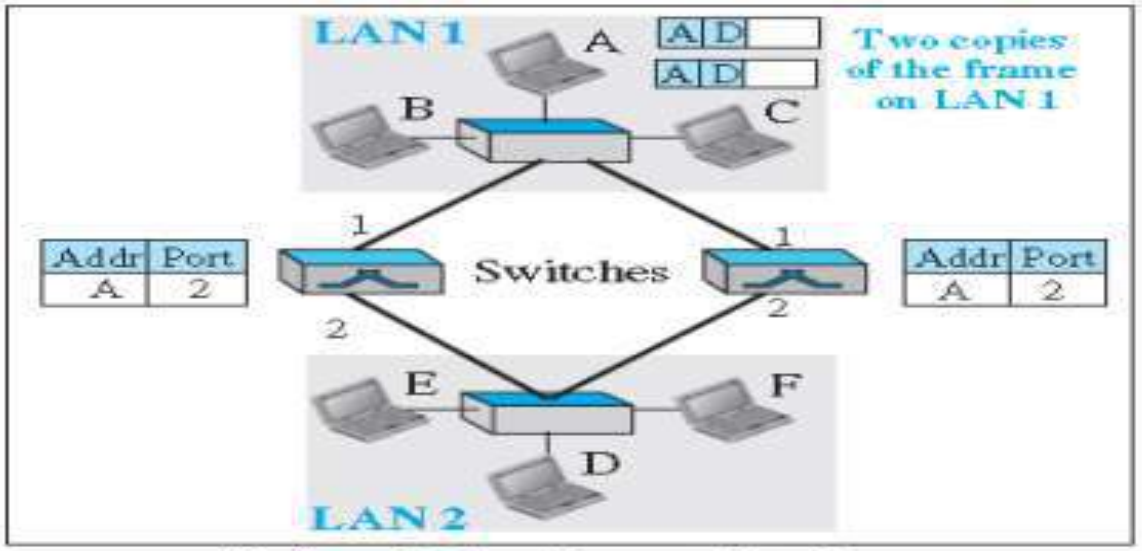
- Transparent switches work fine as long as there are no redundant switches in the system.
- Systems administrators, however, like to have redundant switches(more than one switch between a pair of LANs) to make the system more reliable.
- Redundancy can create loops in the system, which is very undesirable. Loops can be created only when two or more broadcasting LANs (those using hubs, for example) are connected by more than one switch.

a. Station A sends a frame to station D

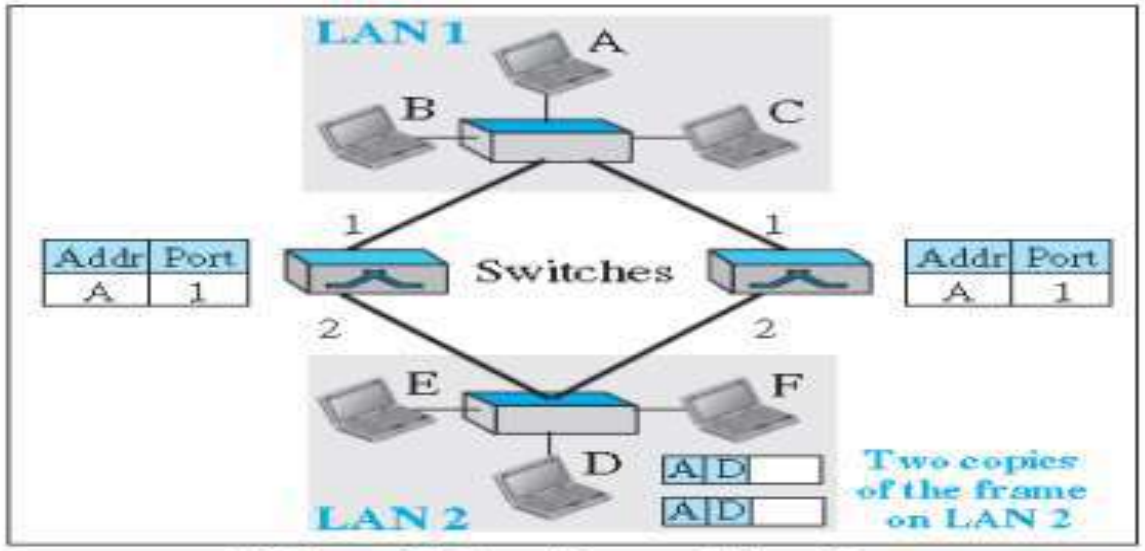


b. Both switches forward the frame





c. Both switches forward the frame



c. Both switches forward the frame

Spanning Tree Algorithm

- To solve the looping problem, the IEEE specification requires that switches use the spanning tree algorithm to create a loopless topology.
- In a switched LAN, this means creating a topology in which each LAN can be reached from any other LAN through one path only (noloop).
- To find the spanning tree, we need to assign a cost (metric) to each arc.
- The interpretation of the cost is left up to the systems administrator.
- We have chosen the minimum hops. However, the hop count is normally 1 from a switch to the LAN and 0 in the reverse direction.

- Steps to find Spanning tree

1. Every switch has a built-in ID (normally the serial number, which is unique). Each switch broadcasts this ID so that all switches know which one has the smallest ID. The switch with the smallest ID is selected as the root switch (root of the tree). We assume that switch S1 has the smallest ID. It is, therefore, selected as the root switch.
2. The algorithm tries to find the shortest path (a path with the shortest cost) from the root switch to every other switch or LAN. The shortest path can be found by examining the total cost from the root switch to the destination.
3. The combination of the shortest paths creates the shortest tree.
4. Based on the spanning tree, we mark the ports that are part of it, the forwarding ports, which forward a frame that the switch receives. We also mark those ports that are not part of the spanning tree, the blocking ports, which block the frames received by the switch

Advantages of Switches

1. **Collision Elimination:** A link-layer switch eliminates the collision. This means increasing the average bandwidth available to a host in the network. In a switched LAN, there is no need for carrier sensing and collision detection; each host can transmit at any time.
2. **Connecting Heterogeneous Devices:** A link-layer switch can connect devices that use different protocols at the physical layer and different transmission media.

As long as the format of the frame at the data-link layer does not change, a switch can receive a frame from a device that uses twisted-pair cable and sends data at 10 Mbps and deliver the frame to another device that uses fiber-optic cable and can receive data at 100 Mbps

Routers

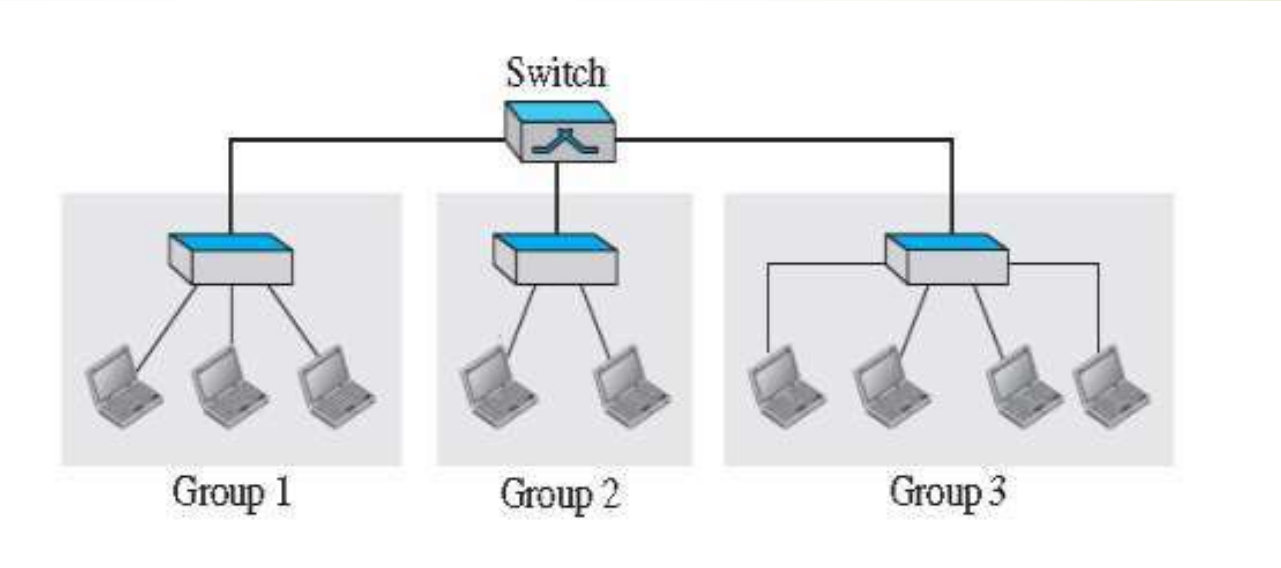
- A router is a three-layer device; it operates in the physical, data-link, and network layers.
- As a physical-layer device, it regenerates the signal it receives. As a link-layer device, the router checks the physical addresses (source and destination) contained in the packet.
- As a network-layer device, a router checks the network-layer addresses.
- A router can connect networks. In other words, a router is an internetworking device; It connects independent networks to form an internetwork.

- Differences between a router and a switch

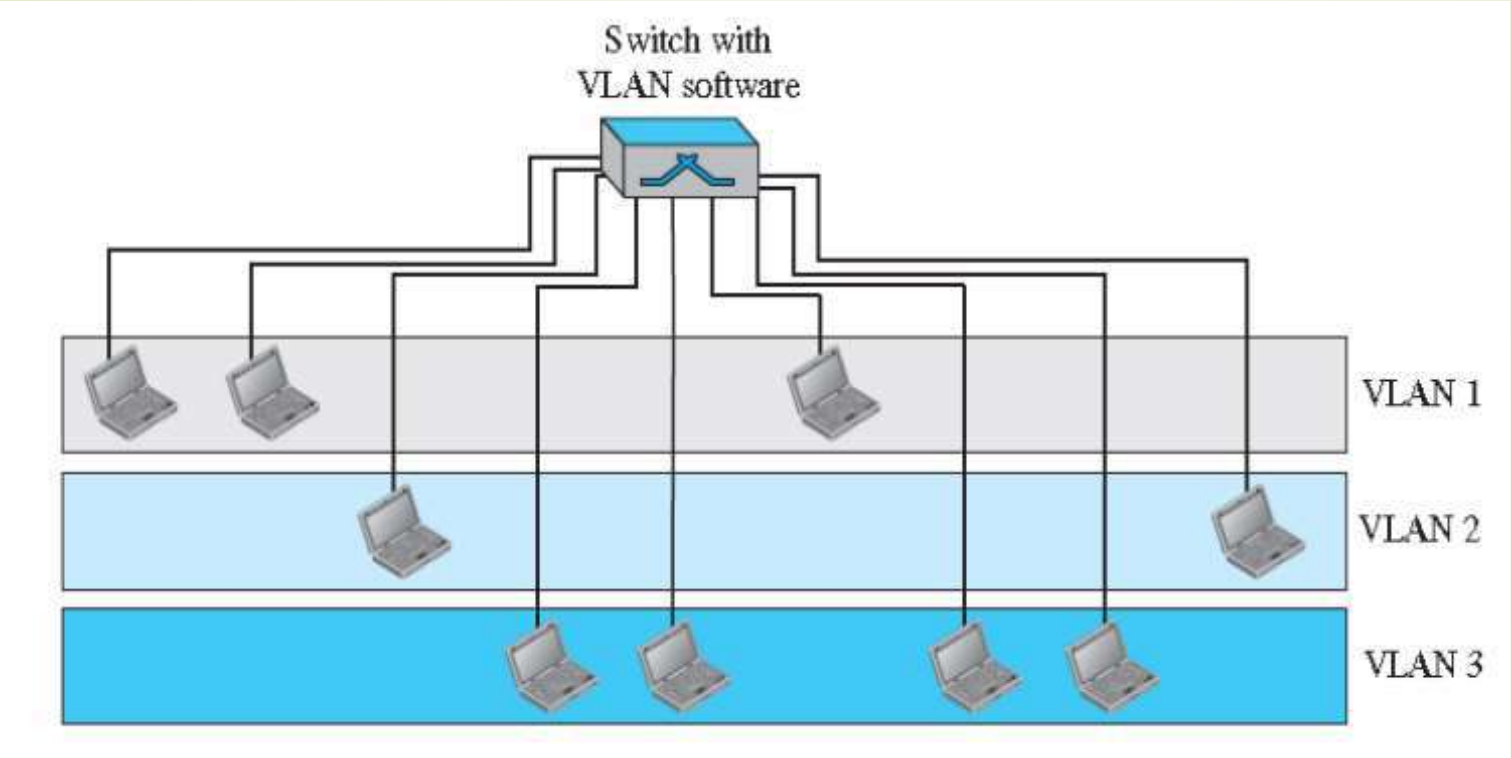
1. A router has a physical and logical (IP) address for each of its interfaces.
2. A router acts only on those packets in which the link-layer destination address matches the address of the interface at which the packet arrives.
3. A router changes the link-layer address of the packet (both source and destination) when it forwards the packet.

VIRTUAL LANs

- A station is considered part of a LAN if it physically belongs to that LAN.
- What happens if we need a virtual connection between two stations belonging to two different physical LANs? We can roughly define a virtual local area network (VLAN) as a local area network configured by software, not by physical wiring.



A Switch connecting three LANs



A Switch using VLAN software

- The whole idea of VLAN technology is to divide a LAN into logical, instead of physical, segments.
- A LAN can be divided into several logical LANs, called VLANs. Each VLAN is a work group in the organization.
- If a person moves from one group to another, there is no need to change the physical configuration.
- Any station can be logically moved to another VLAN. All members belonging to a VLAN can receive broadcast messages sent to that particular VLAN.

- VLAN technology even allows the grouping of stations connected to different switches in a VLAN.
- Stations from switches A and B belong to each VLAN. This is a good configuration for a company with two separate buildings. Each building can have its own switched LAN connected by a backbone.
- People in the first building and people in the second building can be in the same work group even though they are connected to different physical LANs.

Configuration

- How are the stations grouped into different VLANs? Stations are configured in one of three ways:

1. **Manual Configuration:** In a manual configuration, the network administrator uses the VLAN software to manually assign the stations into different VLANs at setup. Later migration from one VLAN to another is also done manually. Note that this is not a physical configuration; it is a logical configuration. The term manually here means that the administrator types the port numbers, the IP addresses, or other characteristics, using the VLAN software.
2. **Automatic Configuration:** In an automatic configuration, the stations are automatically connected or disconnected from a VLAN using criteria defined by the administrator. For example, the administrator can define the project number as the criterion for being a member of a group. When a user changes projects, he or she automatically migrates to a new VLAN.
3. **Semiautomatic Configuration:** A semiautomatic configuration is somewhere between a manual configuration and an automatic configuration. Usually, the initializing is done manually, with migrations done automatically.

Communication between Switches

- In a multi-switched backbone, each switch must know not only which station belongs to which VLAN, but also the membership status of stations connected to other switches.
- Three methods have been devised for this purpose: they are

1. **Table Maintenance** In this method, when a station sends a broadcast frame to its group members, the switch creates an a table and records station membership. The switches send their tables to one another periodically for updating.
2. **Frame Tagging** In this method, when a frame is traveling between switches, an extra header is added to the MAC frame to define the destination VLAN. The frame tag is used by the receiving switches to determine the VLANs to be receiving the broad cast message.

3. Time-Division Multiplexing (TDM) In this method, the connection between switches is divided into time-shared channels. For example, if the total number of VLANs in a backbone is five, each trunk is divided into five channels. The traffic destined for VLAN1 travels in channel1,the traffic destined for VLAN 2 travels in channel 2, and so on. The receiving switch determines the destination VLAN by checking the channel from which the frame arrived.

Advantages of VLAN

1. **Cost and Time Reduction** VLANs can reduce the migration cost of stations going from one group to another. Physical reconfiguration takes time and is costly. Instead of physically moving one station to another segment or even to another switch, it is much easier and quicker to move it by using software.

2. **Creating Virtual Work Groups** VLANs can be used to create virtual work groups. For example, in a campus environment, professors working on the same project can send broadcast messages to one another without the necessity of belonging to the same department. This can reduce traffic if the multicasting capability of IP was previously used.

3. **Security** VLANs provide an extra measure of security. People belonging to the same group can send broadcast messages with the guaranteed assurance that users in other groups will not receive these messages.

Wired LANs: Ethernet

ETHERNET PROTOCOL

- A local area network (LAN) is a computer network that is designed for a limited geographic area such as a building or a campus.
- Although a LAN can be used as an isolated network to connect computers in an organization for the sole purpose of sharing resources, most LANs today are also linked to a wide area network (WAN) or the Internet.
- Almost every LAN except Ethernet has disappeared from the marketplace because Ethernet was able to update itself to meet the needs of the time.

IEEE Project 802

- In 1985, the Computer Society of the IEEE started a project, called Project 802, to set standards to enable intercommunication among equipment from a variety of manufacturers.
- Project 802 does not seek to replace any part of the OSI model or TCP/IP protocol suite. Instead, it is a way of specifying functions of the physical layer and the data-link layer of major LAN protocols.
- The relationship of the 802 Standard to the TCP/IP protocol suite is shown

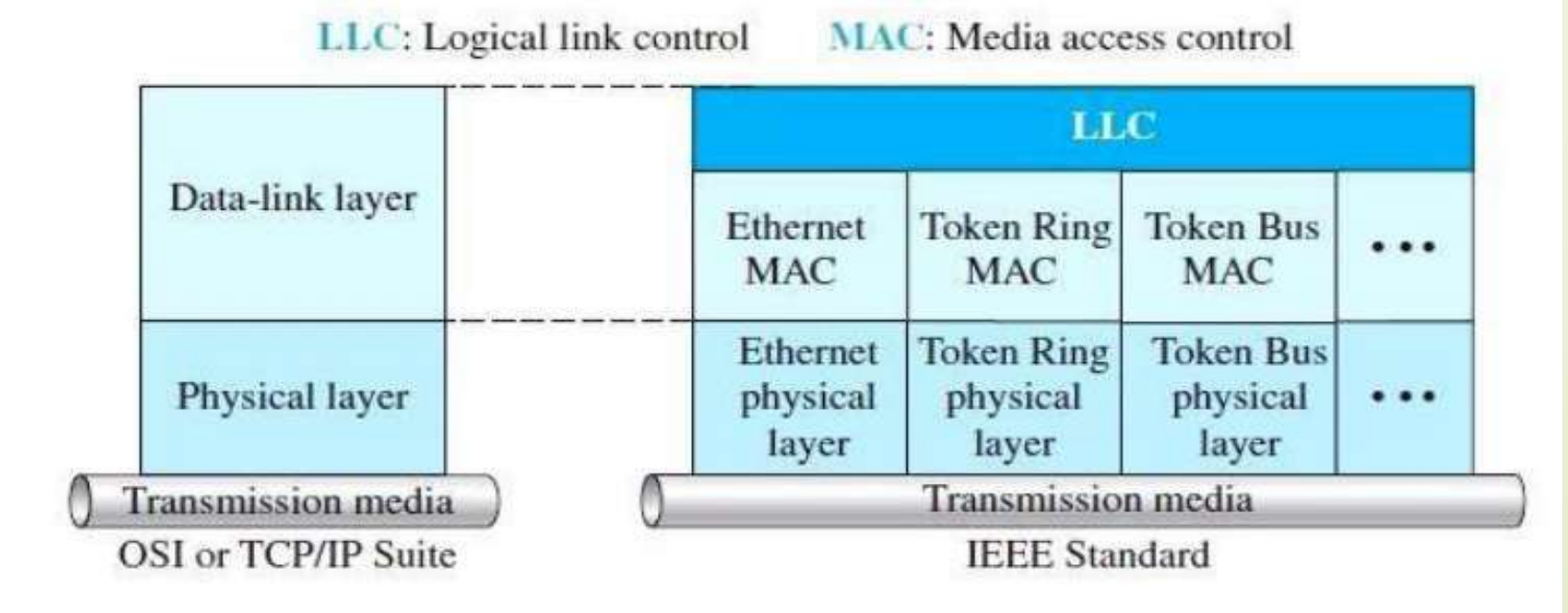


Figure IEEE standard for LANs

The IEEE has subdivided the data-link layer into two sub layers:

1. Logical link control (LLC)
2. Media access control (MAC)

Logical Link Control (LLC)

- In IEEE Project 802, flow control, error control, and part of the framing duties are collected into one sub layer called the logical link control (LLC). Framing is handled in both the LLC sublayer and the MAC sublayer.
- The LLC provides a single link-layer control protocol for all IEEE LANs. This means LLC protocol can provide interconnectivity between different LANs because it makes the MAC sub layer transparent.

Media Access Control (MAC)

- IEEE Project 802 has created a sublayer called media access control that defines the specific access method for each LAN. For example, it defines CSMA/CD as the media access method for Ethernet LANs and defines the token-passing method for Token Ring and Token Bus LANs.
- Part of the framing function is also handled by the MAC layer.

Ethernet Evolution

The Ethernet LAN was developed in the 1970s by Robert Metcalfe and David Boggs.

The four generations of Ethernet are :

1. Standard Ethernet (10 Mbps)
2. Fast Ethernet (100 Mbps)
3. Gigabit Ethernet (1 Gbps) and
4. 10 Gigabit Ethernet (10 Gbps)

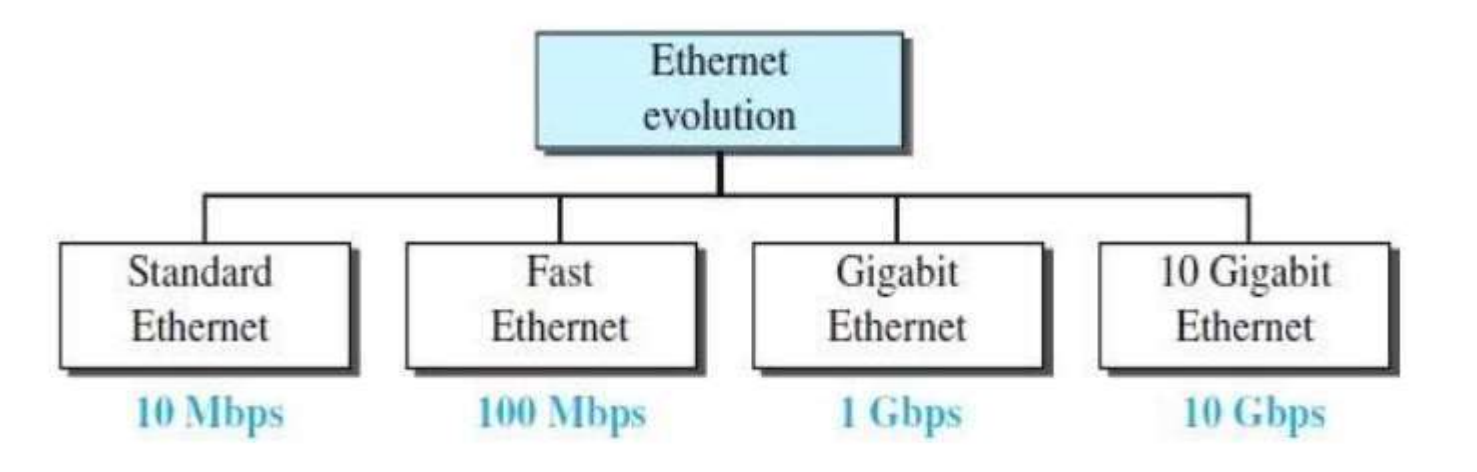


Figure: Ethernet evolution through four generations

STANDARD ETHERNET

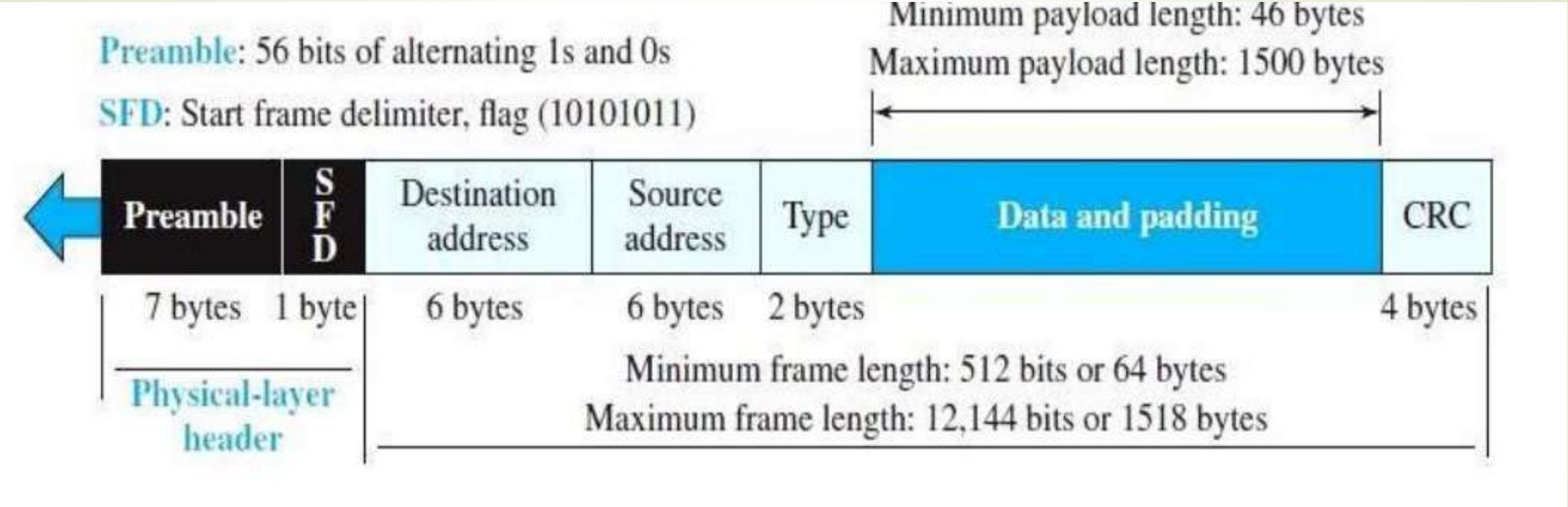
Let us discuss some characteristics of the standard Ethernet.

1. **Connectionless and Unreliable Service**

- Ethernet provides a connectionless service, which means each frame sent is independent of the previous or next frame. Ethernet has no connection establishment or connection termination phases.
- The sender sends a frame whenever it has, the receiver may or may not be ready for it. The sender may overwhelm the receiver with frames, which may result in dropping frames. If a frame drops, the sender will not know about it. Since IP, which is using the service of Ethernet, is also connectionless, it will not know about it either.
- Ethernet is also unreliable like IP and UDP. If a frame is corrupted during transmission and the receiver finds out about the corruption, which has a high level of probability of happening because of the CRC-32, the receiver drops the frame silently. It is the duty of high-level protocols to find out about it.

2. Frame Format

The Ethernet frame contains seven fields, as shown in Figure



- **Preamble**. This field contains 7 bytes (56 bits) of alternating 0s and 1s that alert the receiving system to the coming frame and enable it to synchronize its clock if it's out of synchronization. The pattern provides only an alert and a timing pulse. The 56-bit pattern allows the stations to miss some bits at the beginning of the frame. The preamble is actually added at the physical layer and is not part of the frame.
- **Start frame delimiter (SFD).** This field (1 byte: 10101011) signals the beginning of the frame. The SFD warns the station or stations that this is the last chance for synchronization. The last 2 bits are $(11)_2$ and alert the receiver that the next field is the destination address. This field is actually a flag that defines the beginning of the frame, an Ethernet frame is a variable-length frame. It needs a flag to define the beginning of the frame. The SFD field is also added at the physical layer.

- **Destination address (DA).** This field is six bytes (48 bits) and contains the link layer address of the destination station or stations to receive the packet. When the receiver sees its own link-layer address, or a multicast address for a group that the receiver is a member of, or a broadcast address, it decapsulates the data from the frame and passes the data to the upper layer protocol defined by the value of the type field.
- **Source address (SA)**. This field is also six bytes and contains the link-layer address of the sender of the packet.
- **Type**. This field defines the upper-layer protocol whose packet is encapsulated in the frame. This protocol can be IP, ARP, and so on. In other words, it serves the same purpose as the protocol field in a datagram and the port number in a segment or user datagram. It is used for multiplexing and demultiplexing.

- **Data**. This field carries data encapsulated from the upper-layer protocols. It is a minimum of 46 bytes and a maximum of 1500 bytes.
- If the data coming from the upper layer is more than 1500 bytes, it should be fragmented and encapsulated in more than one frame. If it is less than 46 bytes, it needs to be padded with extra 0s.
- A padded data frame is delivered to the upper-layer protocol as it is (without removing the padding), which means that it is the responsibility of the upper layer to remove or, in the case of the sender, to add the padding.
- The upper-layer protocol needs to know the length of its data. For example, a datagram has a field that defines the length of the data.
- **CRC.** The last field contains error detection information, in this case a CRC-32. The CRC is calculated over the addresses, types, and data field.
- If the receiver calculates the CRC and finds that it is not zero (corruption in transmission), it discards the frame.

3. Frame Length

- Ethernet has imposed restrictions on both the minimum and maximum lengths of a frame. The minimum length restriction is required for the correct operation of CSMA/CD.
- An Ethernet frame needs to have a minimum length of 512 bits or 64 bytes. Part of this length is the header and the trailer.
- If we count 18 bytes of header and trailer (6 bytes of source address, 6 bytes of destination address, 2 bytes of length or type, and 4 bytes of CRC), then the minimum length of data from the upper layer is 64 − 18 = 46 bytes.
- If the upper-layer packet is less than 46 bytes, padding is added to make up the difference.

- The standard defines the maximum length of a frame (without preamble and SFD field) as 1518 bytes. If we subtract the 18 bytes of header and trailer, the maximum length of the payload is 1500 bytes.
- The maximum length restriction has two historical reasons.
  - First, memory was very expensive when Ethernet was designed; a maximum length restriction helped to reduce the size of the buffer.
  - Second, the maximum length restriction prevents one station from monopolizing the shared medium, blocking other stations that have data to send.

NOTE:

- Minimum frame length: 64 bytes
- Maximum frame length: 1518 bytes
- Minimum data length: 46 bytes
- Maximum data length: 1500 bytes

Addressing

- Each station on an Ethernet network (such as a PC, workstation, or printer) has its own network interface card (NIC).
- The NIC fits inside the station and provides the station with a link-layer address.
- The Ethernet address is 6 bytes (48 bits), normally written in hexadecimal notation, with a colon between the bytes. For example, the following shows an Ethernet MAC address:

4A:30:10:21:10:1A

Transmission of Address Bits

- The way the addresses are sent out online is different from the way they are written in hexadecimal notation.
- The transmission is left to right, byte by byte; however, for each byte, the least significant bit is sent first and the most significant bit is sent last.
- This means that the bit that defines an address as unicast or multicast arrives first at the receiver. This helps the receiver to immediately know if the packet is unicast or multicast.

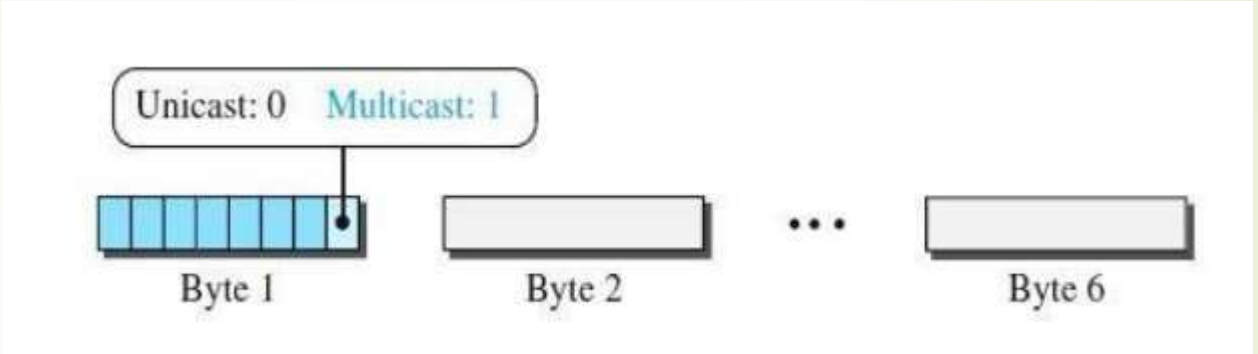
Example

Show how the address 47:20:1B:2E:08:EE is sent out online. Solution: The address is sent left to right, byte by byte; for each byte, it is sent right to left, bit by bit, as shown below

Hexadecimal	47	20	1B	2E	08	EE
Binary	01000111	00100000	00011011	00101110	00001000	11101110
Transmitted ←	11100010	00000100	11011000	01110100	00010000	01110111

Unicast, Multicast, and Broadcast Addresses

- A source address is always a unicast address, the frame comes from only one station. The destination address, however, can be unicast, multicast, or broadcast.



- If the least significant bit of the first byte in a destination address is 0, the address is unicast; otherwise, it is multicast.
- With the way the bits are transmitted, the unicast/multicast bit is the first bit which is transmitted or received.
- The broadcast address is a special case of the multicast address: the recipients are all the stations on the LAN. A broadcast destination address is forty-eight 1s.

Example

Define the type of the following destination addresses

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

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

a. This is a unicast address because A in binary is 1010 (even).
b. This is a multicast address because 7 in binary is 0111 (odd).
c. This is a broadcast address because all digits are Fs in hexadecimal.

Efficiency of Standard Ethernet

- The efficiency of the Ethernet is defined as the ratio of the time used by a station to send data to the time the medium is occupied by this station. The practical efficiency of standard Ethernet has been measured to be,

$$Efficiency = \frac{1}{(1 + 6.4Xa)}$$

- **a=** the number of frames that can fit on the medium.
- **a= (propagation delay/transmission delay)**
- The transmission delay is the time it takes a frame of average size to be sent out and the propagation delay is the time it takes to reach the end of the medium. As the value of parameter decreases, the efficiency increases. This means that if the length of the media is shorter or the frame size longer, the efficiency increases. In the ideal case, a= 0 and the efficiency is 1.

Example

In the Standard Ethernet with the transmission rate of 10 Mbps, we assume that the length of the medium is 2500 m and the size of the frame is 512 bits. The propagation speed of a signal in a cable is normally 2×10^8 m/s.

Solution:

Propagation delay $= 2500/(2 \times 10^8) = 12.5\ \mu s$ **Transmission delay** $= 512/(10^7) = 51.2\ \mu s$

$a = 12.5/51.2 = 0.24$ **Efficiency = 39%**

The example shows that a = 0.24, which means only 0.24 of a frame occupies the whole medium in this case. The efficiency is 39 percent, which is considered moderate; it means that only 61 percent of the time the medium is occupied but not used by a station.

Implementation

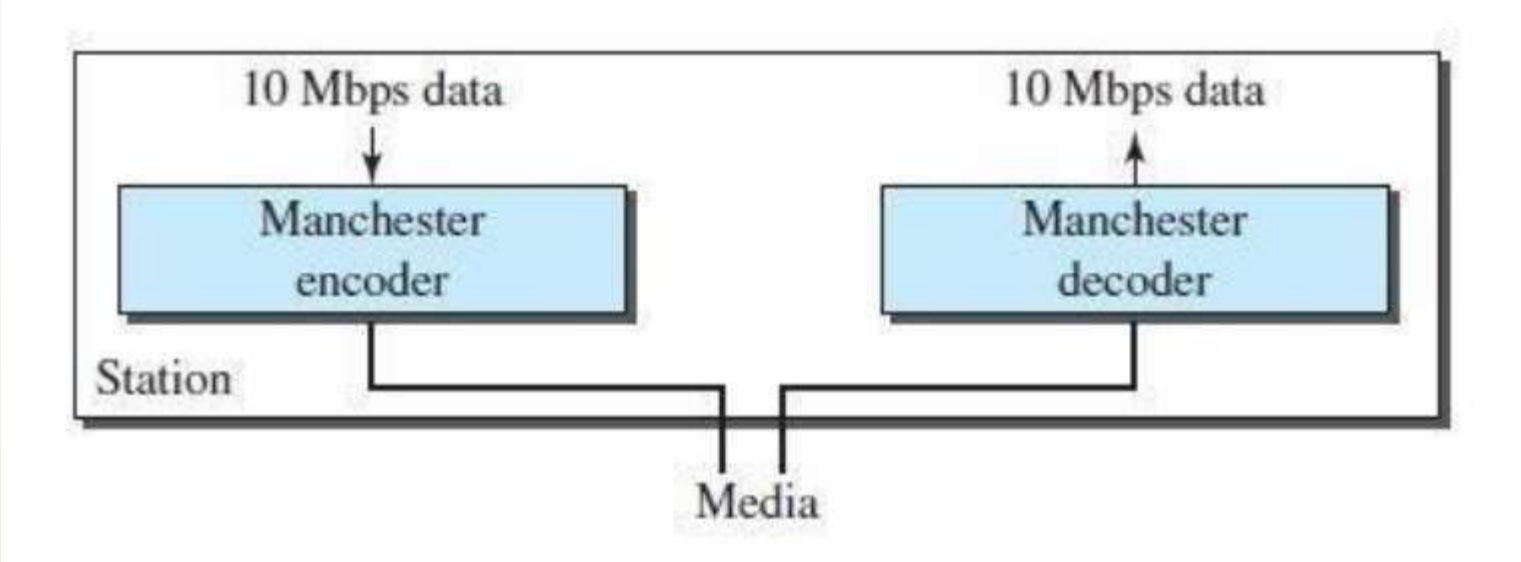
The Standard Ethernet defined several implementations, but only four of them became popular during the 1980s. Table below shows a summary of Standard Ethernet implementations.

Implementation	*Medium*	*Medium Length*	*Encoding*
10Base5	Thick coax	500 m	Manchester
10Base2	Thin coax	185 m	Manchester
10Base-T	2 UTP	100 m	Manchester
10Base-F	2 Fiber	2000 m	Manchester

In the nomenclature 10BaseX, the number defines the data rate (10 Mbps), the term Base means baseband (digital) signal, and X approximately defines either the maximum size of the cable in 100 meters (for example 5 for 500 or 2 for 185 meters) or the type of cable, T for unshielded twisted pair cable (UTP) and F for fiber-optic. The standard Ethernet uses a baseband signal, which means that the bits are changed to a digital signal and directly sent on the line.

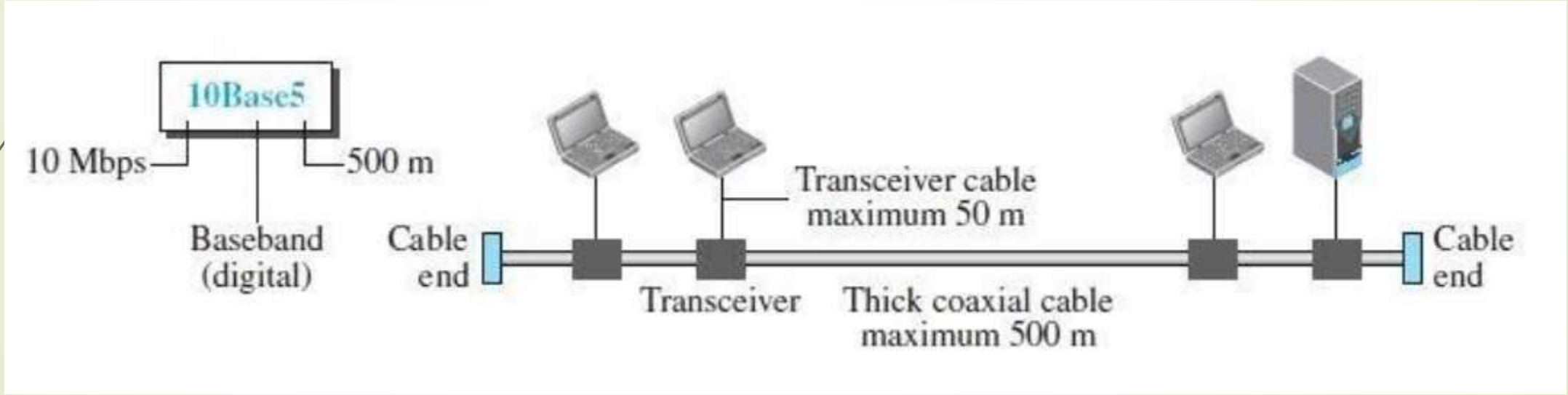
Encoding and Decoding

All standard implementations use digital signalling (baseband) at 10 Mbps. At the sender, data are converted to a digital signal using the Manchester scheme; at the receiver, the received signal is interpreted as Manchester and decoded into data. Manchester encoding is self-synchronous, providing a transition at each bit interval.



10Base5: Thick Ethernet

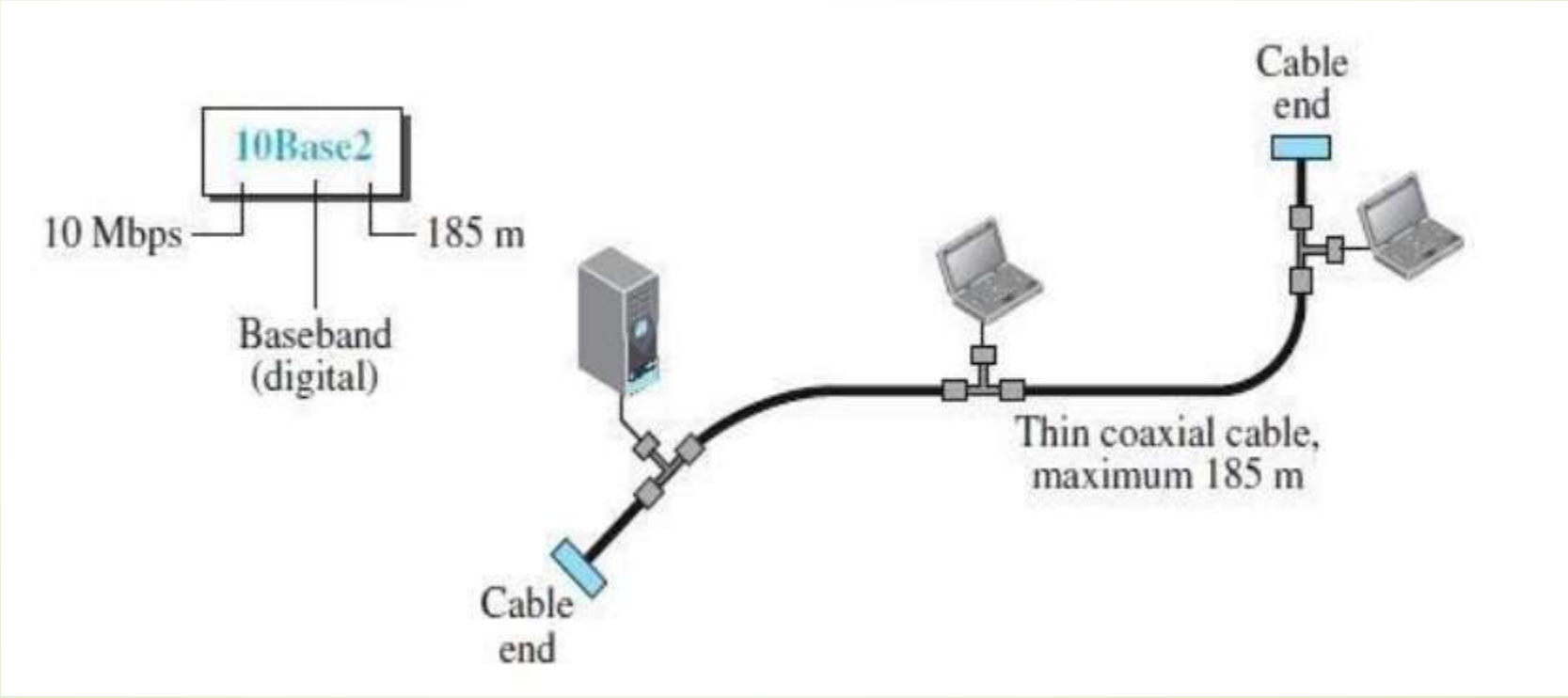
The first implementation is called 10Base5, thick Ethernet, or Thicknet. The nickname derives from the size of the cable, which is roughly the size of a garden hose and too stiff to bend with your hands. 10Base5 was the first Ethernet specification to use a bus topology with an external transceiver (transmitter/receiver) connected via a tap to a thick coaxial cable.



- The transceiver is responsible for transmitting, receiving, and detecting collisions. The transceiver is connected to the station via a transceiver cable that provides separate paths for sending and receiving.
- This means that collision can only happen in the coaxial cable.
- The maximum length of the coaxial cable must not exceed 500 m, otherwise, there is excessive degradation of the signal.
- If a length of more than 500 m is needed, up to five segments, each a maximum of 500 meters, can be connected using repeaters.

10Base2: Thin Ethernet

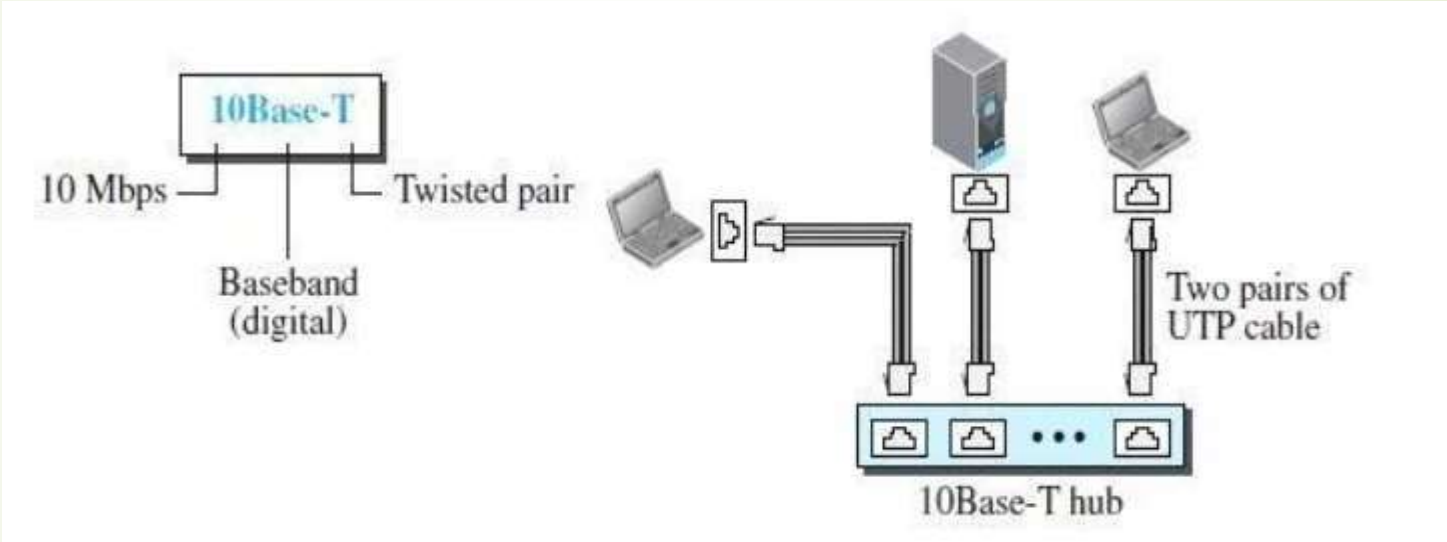
- The second implementation is called 10Base2, thin Ethernet, or Cheapernet. 10Base2 also uses a bus topology, but the cable is much thinner and more flexible.
- In this case, the transceiver is normally part of the network interface card (NIC), which is installed inside the station.



- The collision here occurs in the thin coaxial cable. This implementation is more cost effective than 10Base5 because thin coaxial cable is less expensive than thick coaxial and the tee connections are much cheaper than taps.
- Installation is simpler because the thin coaxial cable is very flexible.
- However, the length of each segment cannot exceed 185 m (close to 200 m) due to the high level of attenuation in thin coaxial cable.

10Base-T: Twisted-Pair Ethernet

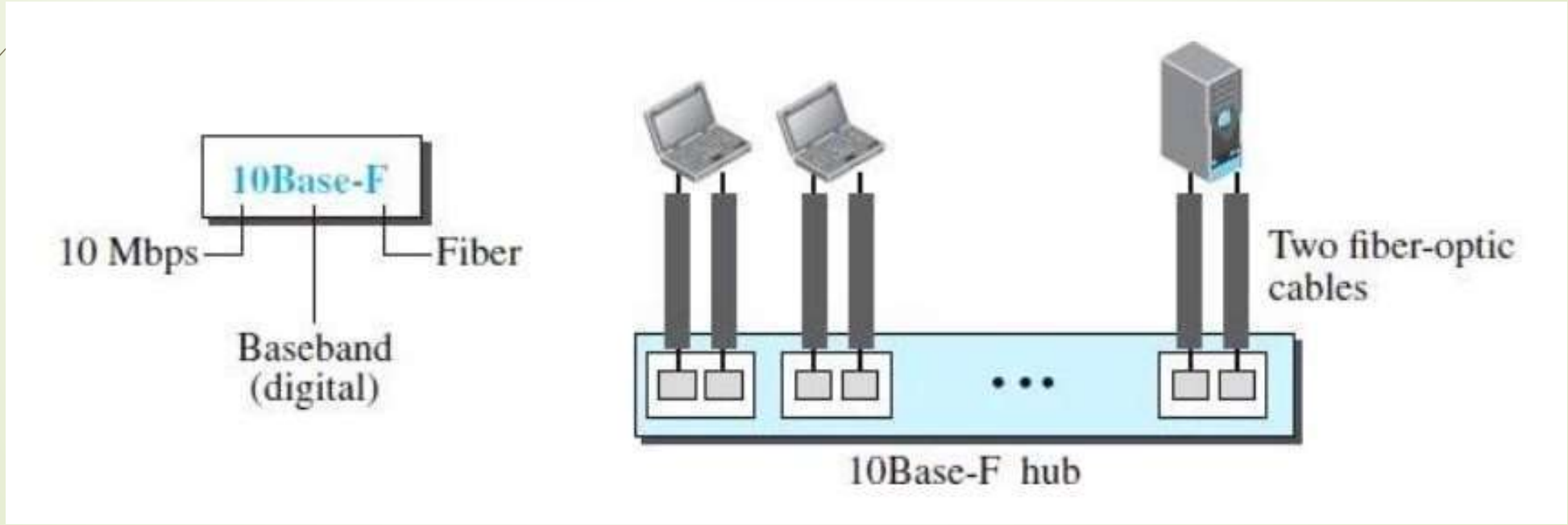
- The third implementation is called 10Base-T or twisted-pair Ethernet. 10Base-T uses a physical star topology. The stations are connected to a hub via two pairs of twisted cable



- Two pairs of twisted cable create two paths (one for sending and one for receiving) between the station and the hub.
- The maximum length of the twisted cable here is defined as 100 m, to minimize the effect of attenuation in the twisted cable.

10Base-F: Fiber Ethernet

Although there are several types of optical fiber 10-Mbps Ethernet, the most common is called 10Base-F. 10Base-F uses a star topology to connect stations to a hub. The stations are connected to the hub using two fiber-optic cables.



Wireless LANs

- Wireless communication is one of the fastest-growing technologies. The demand for connecting devices without the use of cables is increasing everywhere.
- Wireless LANs can be found on college campuses, in office buildings, and in many public areas.

Architectural Comparison

1. **Medium**
2. **Hosts**

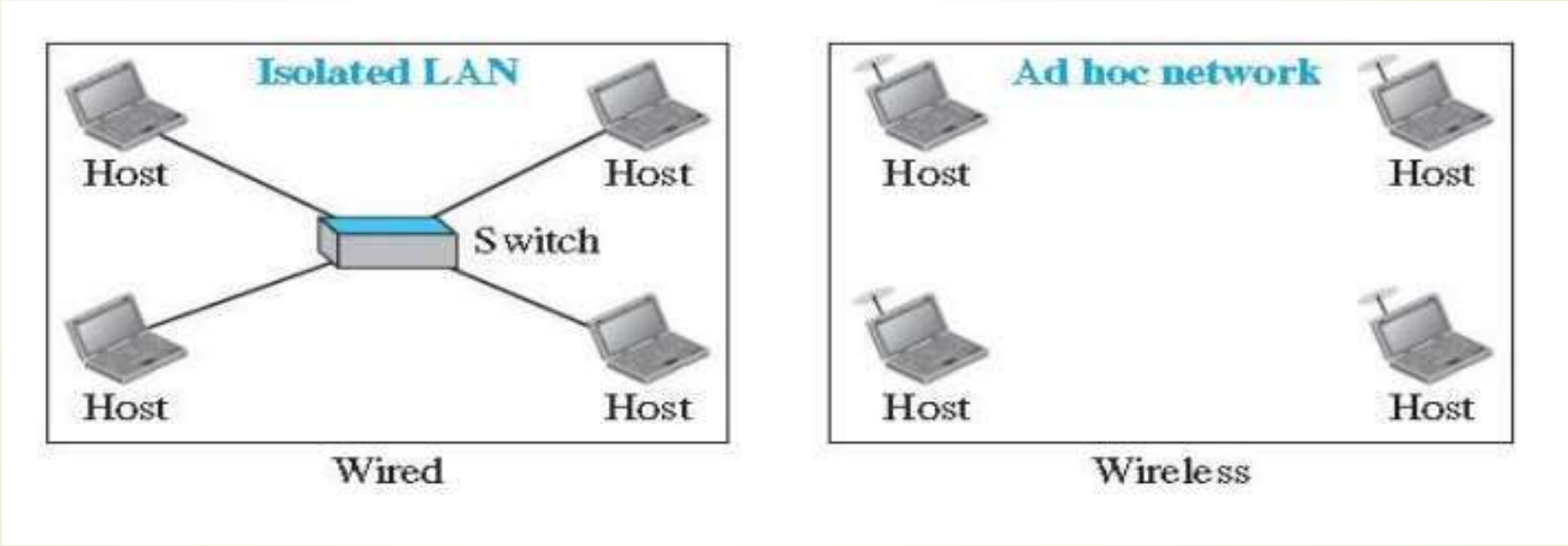
- In a wireless LAN, the medium is air, the signal is generally broadcast. When hosts in a wireless LAN communicate with each other, they are sharing the same medium (multiple access).
- In a very rare situation, we may be able to create a point-to-point communication between two wireless hosts by using a very limited bandwidth and two-directional antennas

2. Hosts

- In a wired LAN, a host is always connected to its network at a point with a fixed link layer address related to its network interface card (NIC).Of course, a host can move from one point in the Internet to another point.
- In this case, its link-layer address remains the same, but its network-layer address will change.
- However, before the host can use the services of the Internet, it needs to be physically connected to the Internet.
- In a wireless LAN, a host is not physically connected to the network; it can move freely and can use the services provided by the network.

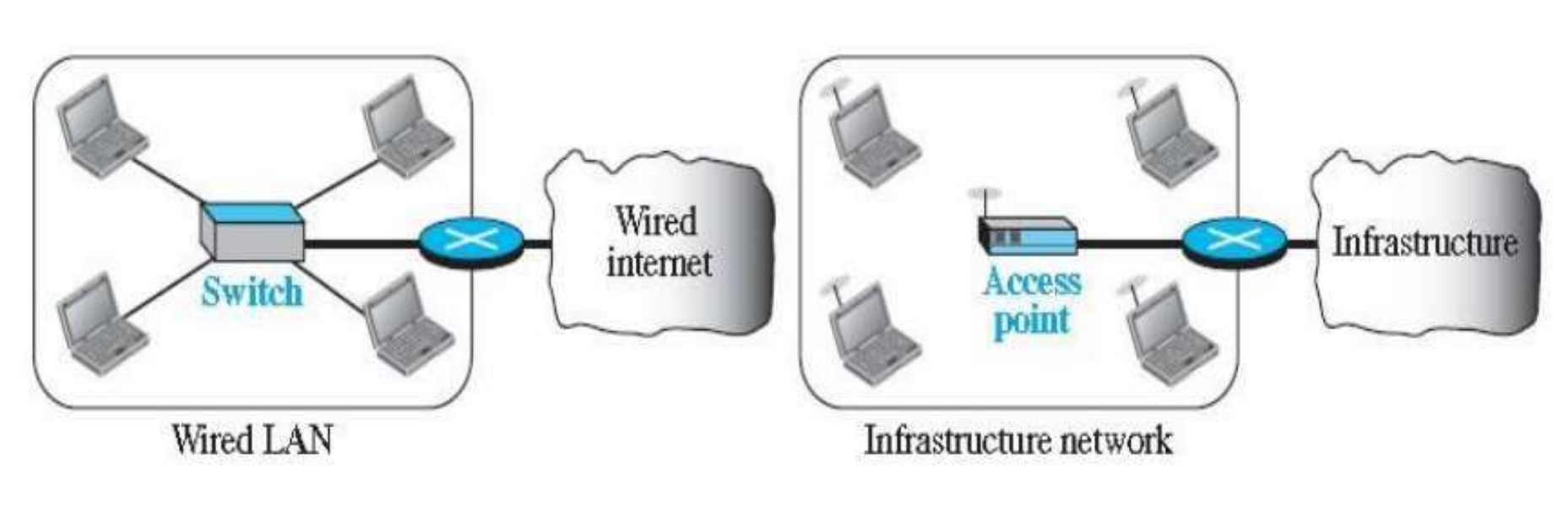
3. Isolated LANs

- A wired isolated LAN is a set of hosts connected via a link-layer switch.
- A wireless isolated LAN, called an ad hoc network in wireless LAN terminology, is a set of hosts that communicate freely with each other.
- The concept of a link-layer switch does not exist in wireless LANs.



4. Connection to Other Networks

- A wired LAN can be connected to another network or an internetwork such as the Internet using a router.
- A wireless LAN may be connected to a wired infrastructure network, to a wireless infrastructure network, or to another wireless LAN.



5. Moving between Environments

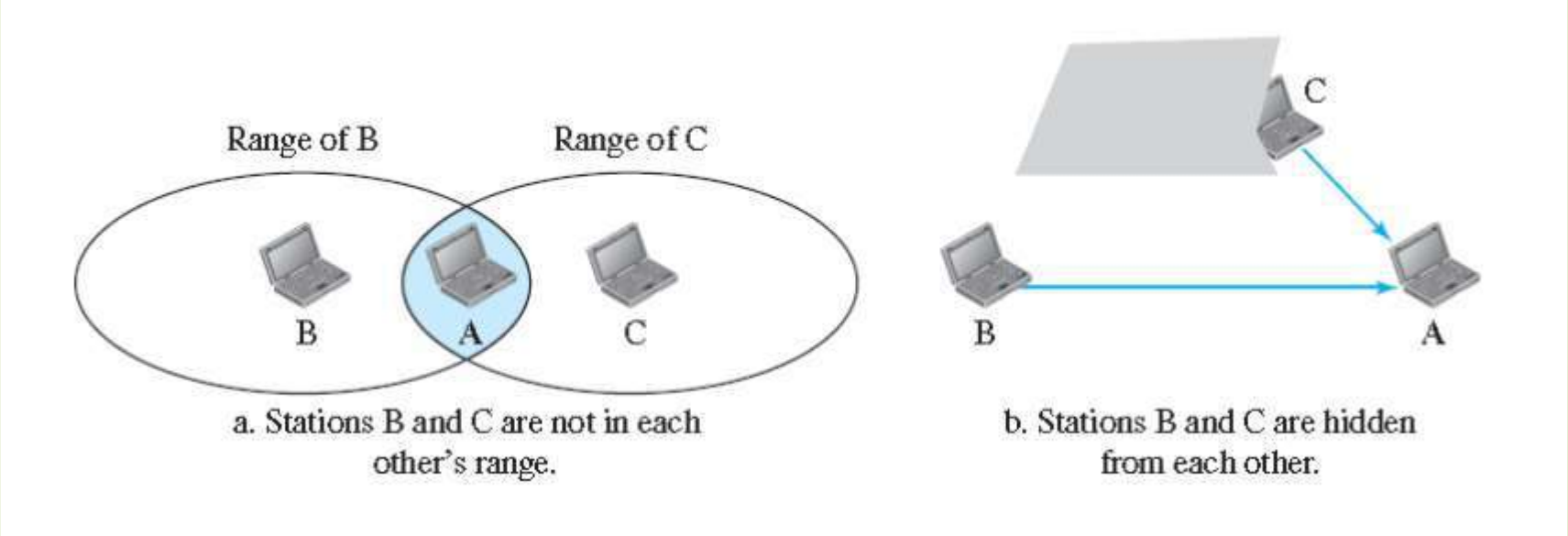
- In order to move from the wired environment to a wireless environment we need to change the network interface cards designed for wired environments to the ones designed for wireless environments.
- We replace the link-layer switch with an access point. In this change, the link-layer addresses will change but the network-layer addresses (IP addresses) will remain the same; we are moving from wired links to wireless links.

Access Control

The most important issue we need to discuss in a wireless LAN is access control. The CSMA/CD algorithm does not work in wireless LANs for three reasons:

1. To detect a collision, a host needs to send and receive at the same time (sending the frame and receiving the collision signal), which means the host needs to work in a duplex mode. Wireless hosts do not have enough power to do so (the power is supplied by batteries). They can only send or receive at onetime.

2. Hidden station problem, In this a station may not be aware of another station's transmission due to some obstacles or range problems, collision may occur but not be detected. Hidden stations can reduce the capacity of the network because of the possibility of collision.



3. Since the distance between stations can be great. Signal fading could prevent a station at one end from hearing a collision at the other end.

- To overcome the above three problems, Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) was invented for wireless LANs.

Thank You

APS College of Engineering,

(Affiliated to Visvesvaraya Technological University and Approved by AICTE, NAAC Accredited)

Somanahalli, Kanakapura Main Road, Bengaluru-560116

Department of Electronics and Communication Engineering

Subject Name: Computer Communication Networks

Subject Name: 21EC73

Semester : 5th

Academic Year: ODD 2023-2024

Module-3

Faculty Name: Dr. Naik D C
Assistant Professor,
Dept., of ECE.

MODULE-3

Network Layer: Introduction, Network layer services: Packetizing, Routing and Forwarding, other services, Packet Switching: Datagram Approach, Virtual-circuit Approach, IPV4 Addresses, Forwarding of IP Packets

Network-Layer Protocols: Internet Protocol (IP) **Unicast Routing**: Introduction, Routing Algorithms: Distance-Vector Routing, Link-State Routing, Path-Vector Routing.

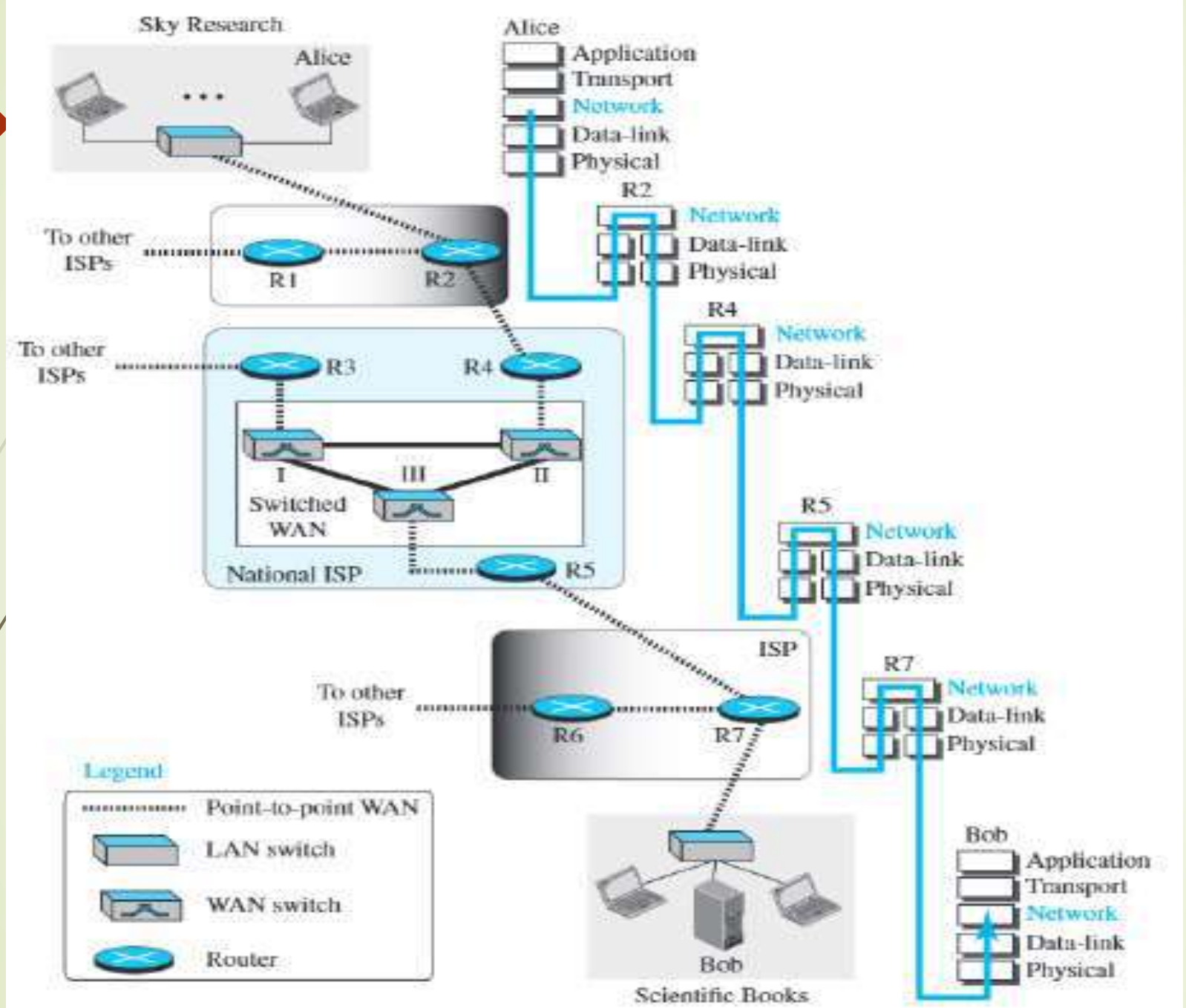


Figure: shows the communication between Alice and Bob at the network layer.

- The figure shows that the Internet is made of many networks (or links) connected through the connecting devices. In other words, the Internet is an internetwork, a combination of LANs and WANs.
- To better understand the role of the network layer (or the internetwork layer), we need to think about the connecting devices (routers or switches) that connect the LANs and WANs.
- As the figure shows, the network layer is involved at the source host, destination host, and all routers in the path (R2, R4, R5, and R7).
- At the source host (Alice), the network layer accepts a packet from a transport layer, encapsulates the packet in a datagram, and delivers the packet to the data-link layer.
- At the destination host (Bob), the datagram is decapsulated, and the packet is extracted and delivered to the corresponding transport layer.

- Although the source and destination hosts are involved in all five layers of the TCP/IP suite, the routers use three layers if they are routing packets only; however, they may need the transport and application layers for control purposes.
- A router in the path is normally shown with two data-link layers and two physical layers, because it receives a packet from one network and delivers it to another network.

Packetizing

- The first duty of the network layer is definitely packetizing: encapsulating the payload (data received from upper layer) in a network-layer packet at the source and decapsulating the payload from the network-layer packet at the destination.
- In other words, one duty of the network layer is to carry a payload from the source to the destination with-out changing it or using it.
- The source host receives the payload from an upper-layer protocol, adds a header that contains the source and destination addresses and some other information that is required by the network-layer protocol and delivers the packet to the data-link layer.
- The destination host receives the network-layer packet from its data-link layer, decapsulates the packet, and delivers the payload to the corresponding upper-layer protocol

Routing and Forwarding

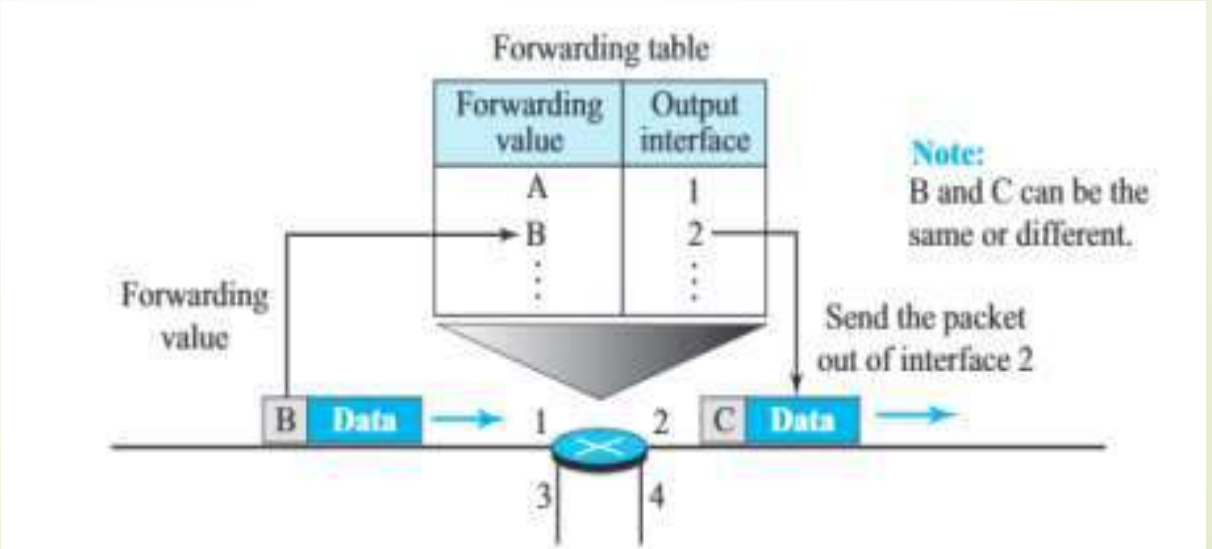
- Other duties of the network layer, which are as important as the first, are routing and forwarding, which are directly related to each other.

Routing

- The network layer is responsible for routing the packet from its source to the destination.
- A physical network is a combination of networks (LANs and WANs) and routers that connect them.
- The network layer needs to have some specific strategies for defining the best route
- In the Internet today, this is done by running some routing protocols to help the routers coordinate their knowledge about the neighbourhood and to come up with consistent tables to be used when a packet arrives.

Forwarding

- If routing is applying strategies and running some routing protocols to create the decision-making tables for each router, forwarding can be defined as the action applied by each router when a packet arrives at one of its interfaces.
- When a router receives a packet from one of its attached networks, it needs to forward the packet to another attached network or to some attached networks.
- To make this decision, the router uses a piece of information in the packet header, which can be the destination address or a label, to find the corresponding output interface number in the forwarding table.



Other Service

Some of the other services expected from the network layer are.

- Error Control
- Flow Control
- Congestion Control
- Quality of Service
- Security

Error Control

- Although error control also can be implemented in the network layer, the designers of the network layer in the Internet ignored this issue for the data being carried by the network layer.
- One reason for this decision is the fact that the packet in the network layer may be fragmented at each router, which makes error checking at this layer inefficient.

Flow Control

- Flow control regulates the amount of data a source can send with out overwhelming the receiver.
- If the upper layer at the source computer produces data faster than the upper layer at the destination computer can consume it, the receiver will be overwhelmed with data.
- To control the flow of data, the receiver needs to send some feedback to the sender to inform the latter that it is overwhelmed with data.
- A few reasons for the lack of flow control in the design of the network layer can be mentioned
  - First, since there is no error control in this layer, the job of the network layer at the receiver is so simple that it may rarely be overwhelmed.
  - Second, the upper layers that use the service of the network layer can implement buffers to receive data from the network layer as they are ready and do not have to consume the data as fast as it is received.
  - Third, flow control is provided for most of the upper-layer protocols that use the services of the network layer, so another level of flow control makes the network layer more complicated and the whole system less efficient.

Congestion Control

- Congestion in the network layer is a situation in which too many datagrams are present in an area of the Internet.
- Congestion may occur if the number of datagrams sent by source computers is beyond the capacity of the network or routers.
- In this situation, some routers may drop some of the datagrams.
- If the congestion continues, sometimes a situation may reach a point where the system collapses and no datagrams are delivered.

Quality of Service

- As the Internet has allowed new applications such as multimedia communication (in particular real-time communication of audio and video), the quality of service (QoS) of the communication has become more and more important.

Security

- Security was not a concern when the Internet was originally designed because it was used by a small numbers of users at universities for research activities.
- Today, however, security is a big concern, to provide security for a connectionless network layers, we need to have another virtual level that changes the connectionless service to a connection-oriented service.

Packet Switching

- A router, in fact, is a switch that creates a connection between an input port and an output port, just as an electrical switch connects the input to the output to let electricity flow.
- Although in data communication switching techniques are divided into two broad categories, **circuit switching** and **packet switching**.
- Only packet switching is used at the network layer because the unit of data at this layer is a packet.
- Circuit switching is mostly used at the physical layer;
- At the network layer, a message from the upper layer is divided into manageable packets and each packet is sent through the network.

- The source of the message sends the packets one by one; the destination of the message receives the packets one by one.
- The connecting devices in a packet-switched network still need to decide how to route the packets to the final destination.
- Today, a packet-switched network can use two different approaches to route the packets: **the datagram approach** and **the virtual circuit approach.**

Datagram Approach: Connectionless Service

- When the Internet started, to make it simple, the network layer was designed to provide a connectionless service in which the network-layer protocol treats each packet independently, with each packet having no relationship to any other packet.
- The idea was that the network layer is only responsible for delivery of packets from the source to the destination. In this approach, the packets in a message may or may not travel the same path to their destination.

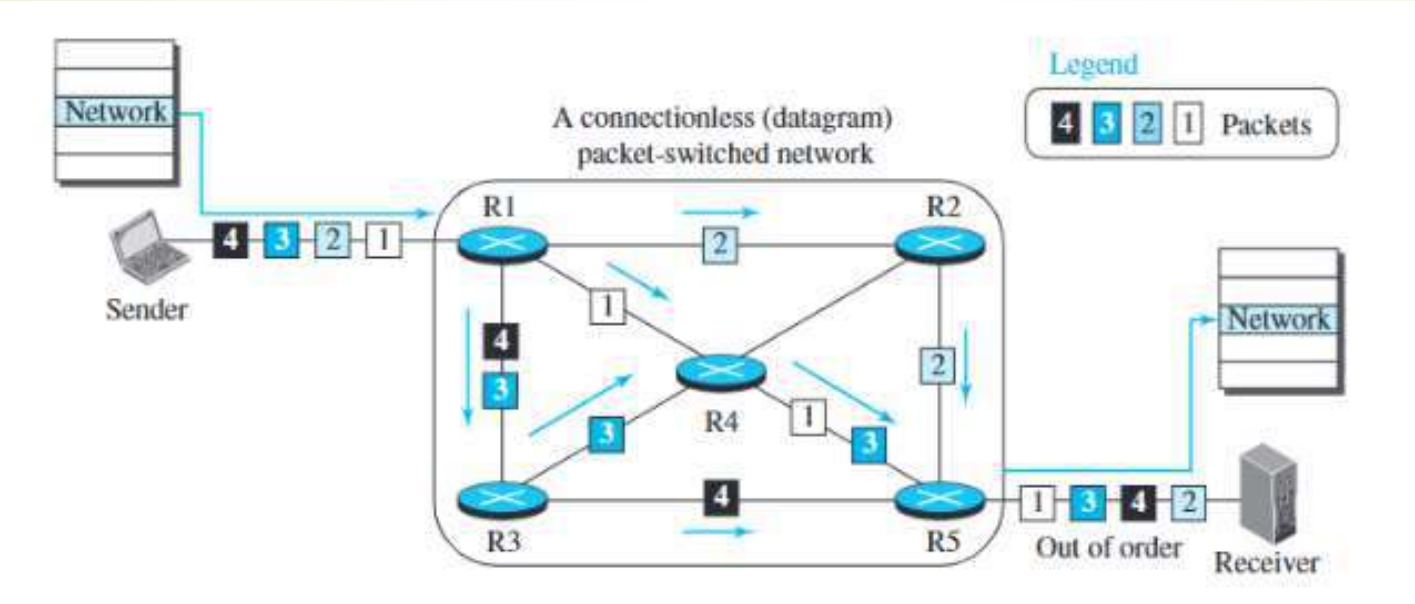


Figure: A connectionless packet-switched network

- When the network layer provides a connectionless service, each packet traveling in the Internet is an independent entity; there is no relationship between packets belonging to the same message.
- The switches in this type of network are called routers.
- A packet belonging to a message may be followed by a packet belonging to the same message or to a different message. A packet may be followed by a packet coming from the same or from a different source.
- Each packet is routed based on the information contained in its header: source and destination addresses.
- The router in this case routes the packet based only on the destination address. The source address may be used to send an error message to the source if the packet is discarded.

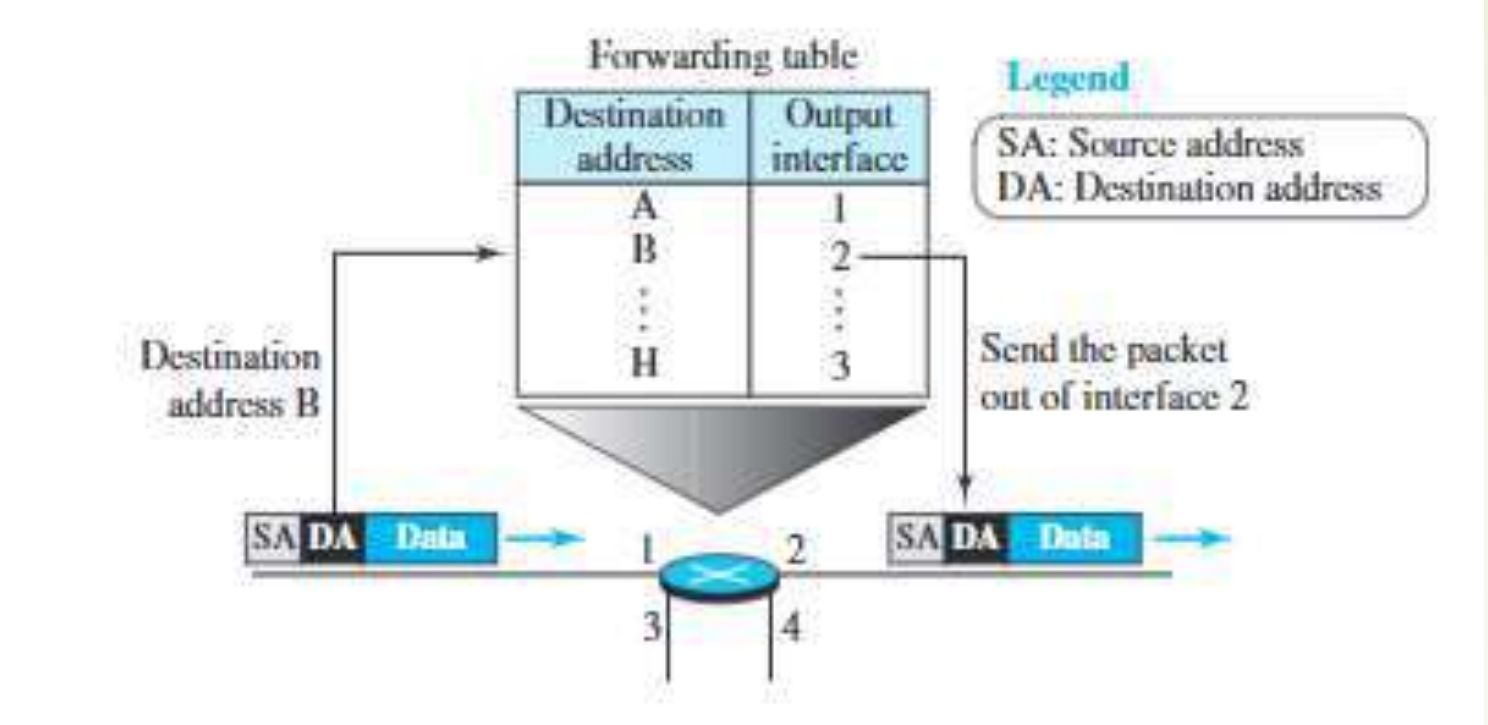


Figure: Forwarding process in a router when used in a connectionless network

- In the datagram approach, the forwarding decision is based on the destination address of the packet.

Virtual-Circuit Approach: Connection-Oriented Service

- In a connection-oriented service (also called virtual-circuit approach), there is a relationship between all packets belonging to a message.
- Before all datagrams in a message can be sent, a virtual connection should be set up to define the path for the datagrams. After connection setup, the datagrams can all follow the same path.
- In this type of service, not only must the packet contain the source and destination addresses, it must also contain a flow label, a virtual circuit identifier that defines the virtual path the packet should follow.

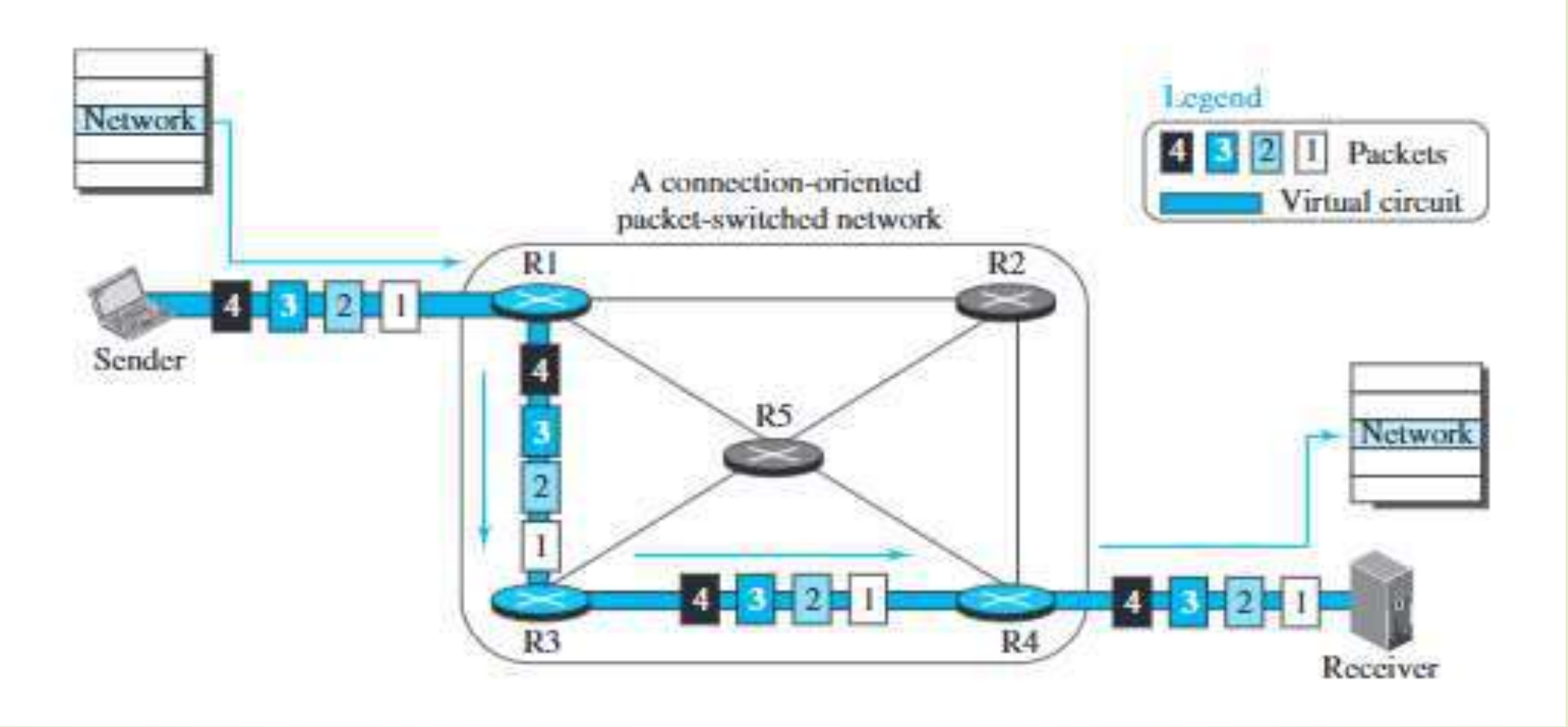
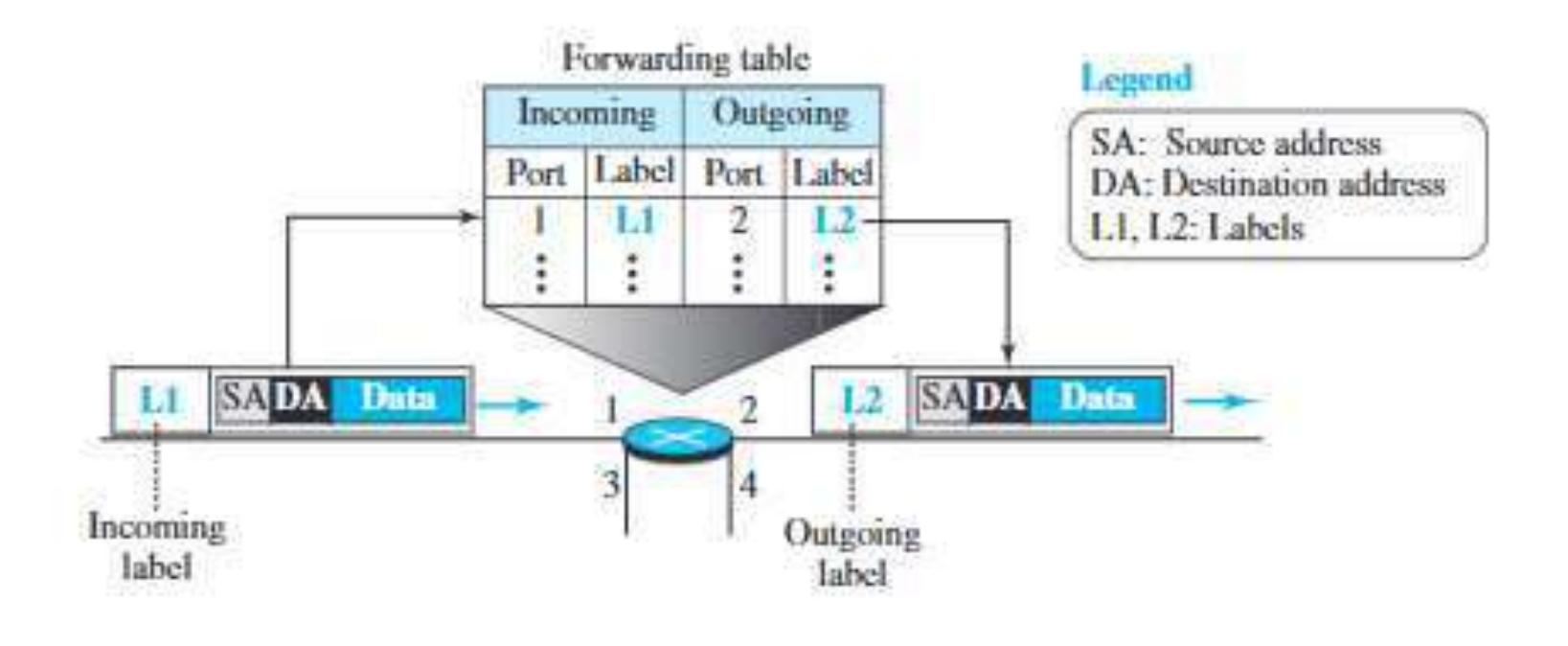


Figure: A virtual-circuit packet-switched network

- Each packet is forwarded based on the label in the packet. To follow the idea of connection-oriented design to be used in the Internet, we assume that the packet has a label when it reaches the router.

- Figure shows the idea. In this case, the forwarding decision is based on the value of the label, or virtual circuit identifier, as it is sometimes called.



- In the virtual-circuit approach, the forwarding decision is based on the label of the packet.

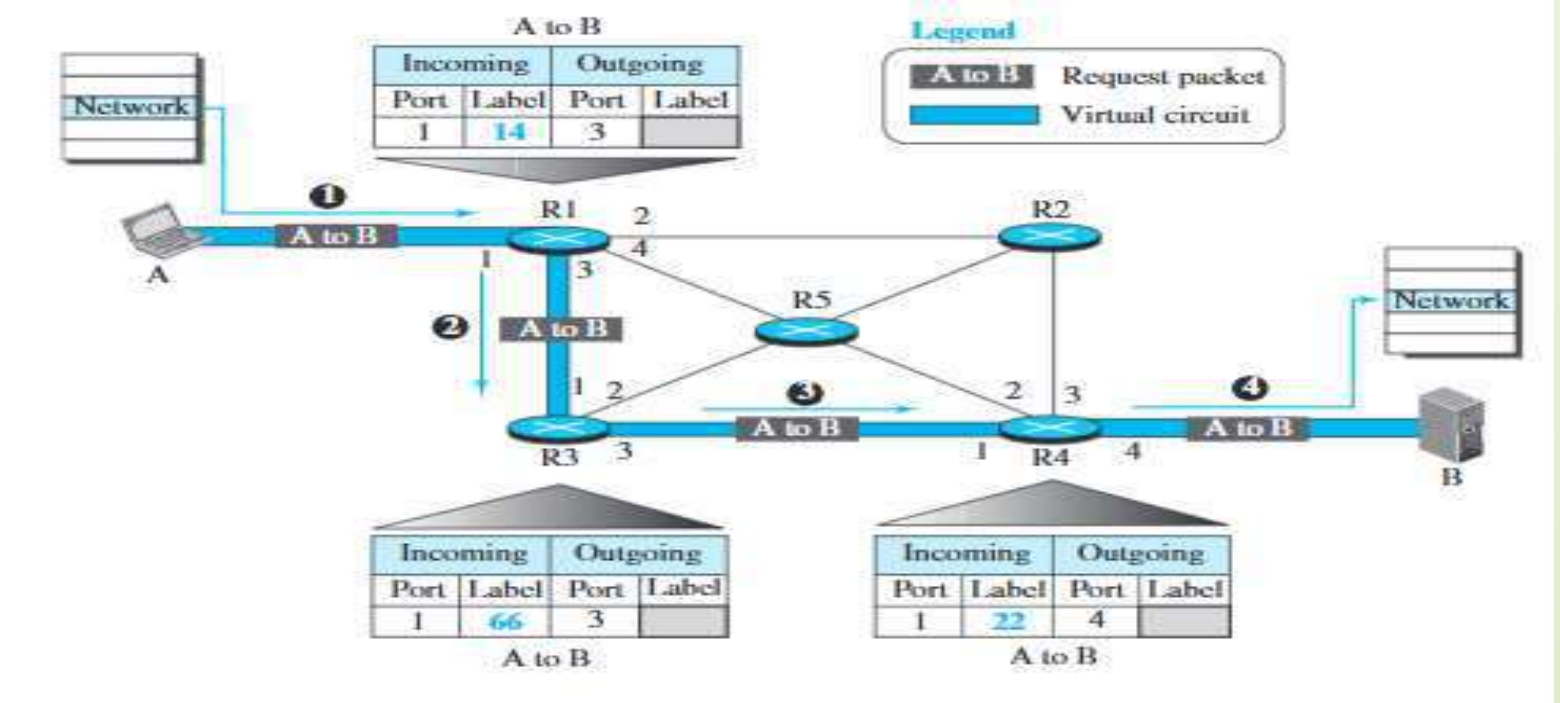
- To create a connection-oriented service, a three-phase process is used:
  - setup,
  - data transfer, and
  - Teardown
- In the setup phase, the source and destination addresses of the sender and receiver are used to make table entries for the connection-oriented service.
- In the teardown phase, the source and destination inform the router to delete the corresponding entries.
- Data transfer occurs between these two phases.

Setup Phase

- In the setup phase, a router creates an entry for a virtual circuit. For example, suppose source A needs to create a virtual circuit to destination B.
- Two auxiliary packets need to be exchanged between the sender and the receiver: the request packet and the acknowledgment packet.

Request packet

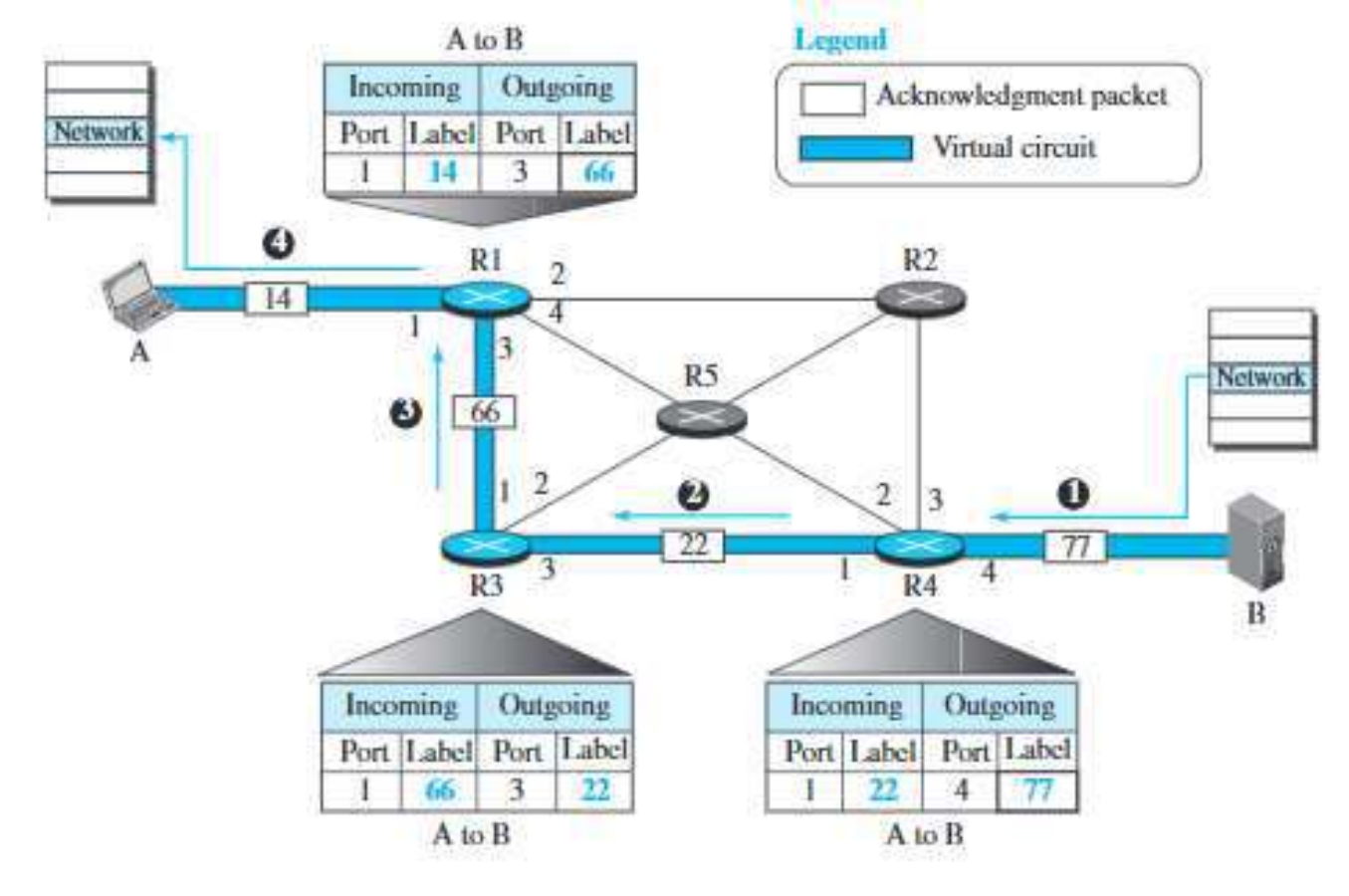
- A request packet is sent from the source to the destination. This auxiliary packet carries the source and destination addresses. Figure shows the process.



1. Source A sends a request packet to router R1.
2. Router R1 receives the request packet. It knows that a packet going from A to B goes out through port 3. How the router has obtained this information is a point covered later. For the moment, assume that it knows the output port. The router creates an entry in its table for this virtual circuit, but it is only able to fill three of the four columns. The router assigns the incoming port (1) and chooses an available incoming label (14) and the outgoing port (3). It does not yet know the outgoing label, which will be found during the acknowledgment step. The router then forwards the packet through port 3 to router R3.
3. Router R3 receives the setup request packet. The same events happen here as at router R1; three columns of the table are completed: in this case, incoming port (1), incoming label (66), and outgoing port (3).
4. Router R4 receives the setup request packet. Again, three columns are completed: incoming port (1), incoming label (22), and outgoing port (4).
5. Destination B receives the setup packet, and if it is ready to receive packets from A, it assigns a label to the incoming packets that come from A, in this case 77, as shown in Below Figure. This label lets the destination know that the packets come from A, and not from other sources.

Acknowledgment Packet

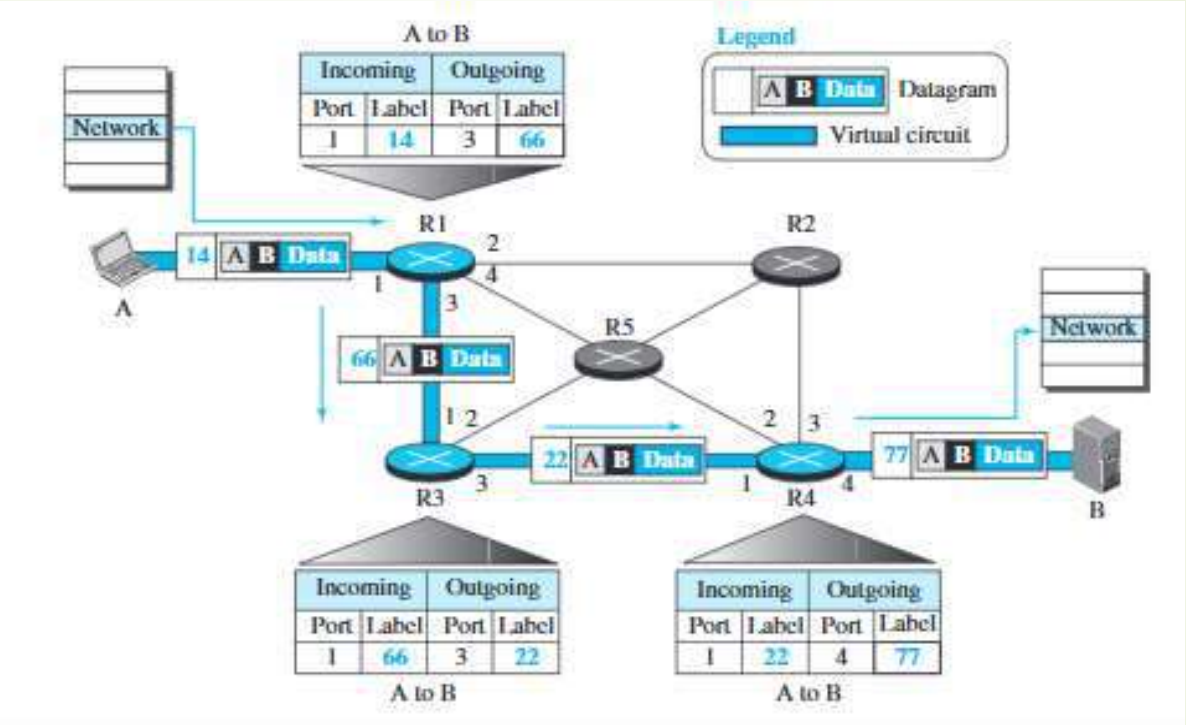
- A special packet, called the acknowledgment packet, completes the entries in the switching tables. Figure shows the process.



1. The destination sends an acknowledgment to router R4. The acknowledgment carries the global source and destination addresses so the router knows which entry in the table is to be completed. The packet also carries label 77, chosen by the destination as the incoming label for packets from A. Router R4 uses this label to complete the outgoing label column for this entry. Note that 77 is the incoming label for destination B, but the outgoing label for router R4.
2. Router R4 sends an acknowledgment to router R3 that contains its incoming label in the table, chosen in the setup phase. Router R3 uses this as the outgoing label in the table.
3. Router R3 sends an acknowledgment to router R1 that contains its incoming label in the table, chosen in the setup phase. Router R1 uses this as the outgoing label in the table.
4. Finally router R1 sends an acknowledgment to source A that contains its incoming label in the table, chosen in the setup phase.
5. The source uses this as the outgoing label for the data packets to be sent to destination B.

Data-Transfer Phase

- The second phase is called the data-transfer phase. After all routers have created their forwarding table for a specific virtual circuit, then the network-layer packets belonging to one message can be sent one after another.



- The source computer uses the label 14, which it has received from router R1 in the setup phase.
- Router R1 forwards the packet to router R3, but changes the label to 66. Router R3 forwards the packet to router R4, but changes the label to 22.
- Finally, router R4 delivers the packet to its final destination with the label 77. All the packets in the message follow the same sequence of labels, and the packets arrive in order at the destination.

Teardown Phase

- In the teardown phase, source A, after sending all packets to B, sends a special packet called a teardown packet. Destination B responds with a confirmation packet. All routers delete the corresponding entries from their tables.

IPV4 ADDRESSES

- The identifier used in the IP layer of the TCP/IP protocol suite to identify the connection of each device to the Internet is called the Internet address or IP address.
- An IPv4 address is a 32-bit address that uniquely and universally defines the connection of a host or a router to the Internet.
- The IP address is the address of the connection, not host or the router, because if the device is moved to another network, the IP address may be changed.
- IPv4 addresses are unique in the sense that each address defines one, and only one, connection to the Internet.
- If a device has two connections to the Internet, via two networks, it has two IPv4 addresses.
- IPv4 addresses are universal in the sense that the addressing system must be accepted by any host that wants to be connected to the Internet

Address Space

- A protocol like IPv4 that defines addresses has an address space. An address space is the total number of addresses used by the protocol.
- If a protocol uses b bits to define an address, the address space is 2^b because each bit can have two different values (0 or 1).
- IPv4 uses 32-bit addresses, which means that the address space is 2^{32} or 4,294,967,296 (more than four billion). If there were no restrictions, more than 4 billion devices could be connected to the Internet.

Notation

- There are three common notations to show an IPv4 address: binary notation (base 2), dotted-decimal notation (base 256), and hexadecimal notation (base 16). In binary notation, an IPv4 address is displayed as 32 bits.

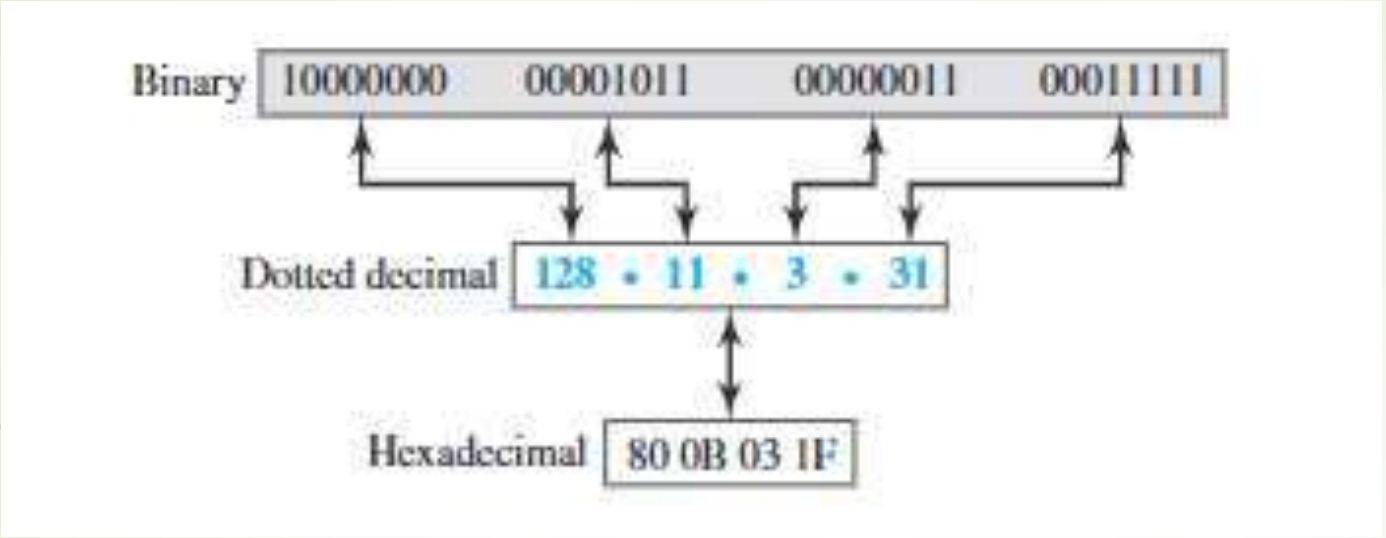


Figure: Three different notations in IPv4 addressing

Hierarchy in Addressing

- In any communication network that involves delivery, such as a telephone network or a postal network, the addressing system is hierarchical.
- Similarly, a telephone number is divided into the country code, area code, local exchange, and the connection.
- A 32-bit IPv4 address is also hierarchical, but divided only into two parts. The first part of the address, called the prefix defines the network; the second part of the address, called the suffix, defines the node (connection of a device to the Internet).

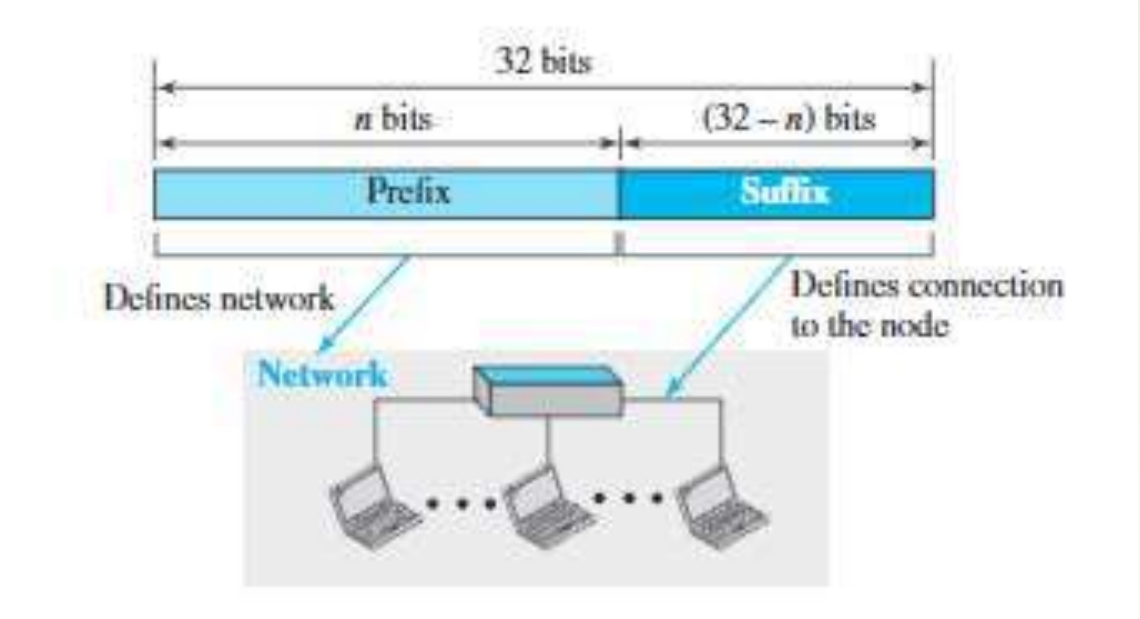
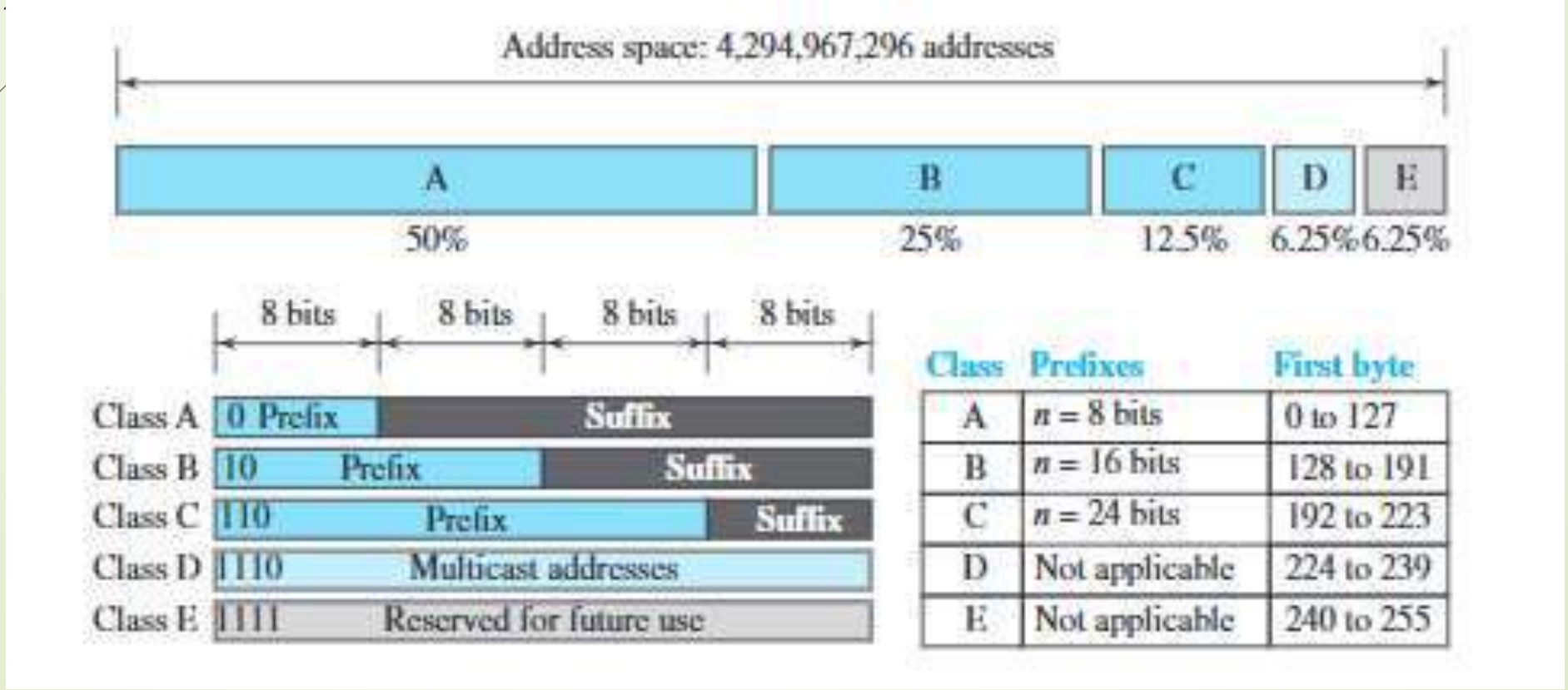


Figure: Hierarchy in addressing

- A prefix can be fixed length or variable length.
- The network identifier in the IPv4 was first designed as a fixed-length prefix. This scheme is referred to as classful addressing.
- The new scheme, which is referred to as classless addressing, uses a variable-length network prefix.

Classful Addressing

- When the Internet started, an IPv4 address was designed with a fixed-length prefix, but to accommodate both small and large networks, three fixed-length prefixes were designed instead of one (n=8,n=16, and n=24).
- The whole address space was divided into five classes (class A, B, C, D, and E).



Class	Prefixes	First byte
A	$n = 8$ bits	0 to 127
B	$n = 16$ bits	128 to 191
C	$n = 24$ bits	192 to 223
D	Not applicable	224 to 239
E	Not applicable	240 to 255

- In class A, the network length is 8 bits, but since the first bit, which is 0, defines the class, we can have only seven bits as the network identifier.
- This means there are only $2^7 = 128$ networks in the world that can have a class A address.
- In class B, the network length is 16 bits, but since the first two bits, which are $(10)_2$, define the class, we can have only 14 bits as the network identifier. This means there are only 2^{14} =16,384 networks in the world that can have a class B address.
- All addresses that start with $(110)_2$ belong to class C. In class C, the network length is 24 bits, but since three bits define the class, we can have only 21 bits as the network identifier. This means there are 2^{21}=2,097,152 networks in the world that can have a class C address.
- Class D is not divided into prefix and suffix. It is used for multicast addresses. All addresses that start with 1111 in binary belong to class E. As in Class D, Class E is not divided into prefix and suffix and is used as reserve.

Address Depletion

- Since the addresses were not distributed properly, the Internet was faced with the problem of the addresses being rapidly used up, resulting in no more addresses available for organizations and individuals that needed to be connected to the Internet.
- To understand the problem, let us think about class A. This class can be assigned to only 128 organizations in the world, but each organization needs to have a single network.
- Class B addresses were designed for midsize organizations, but many of the addresses in this class also remained unused.
- Class C addresses have a completely different flaw in design. The number of addresses that can be used in each network (256) was so small that most companies were not comfortable using a block in this address class.
- Class E addresses were almost never used, wasting the whole class

Subnetting and Supernetting

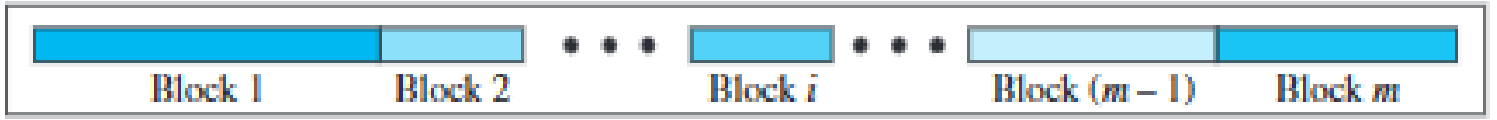
- To alleviate address depletion, two strategies were proposed and, to some extent, implemented: subnetting and supernetting. In subnetting, a class A or class B block is divided into several subnets.
- Each subnet has a larger prefix length than the original network. For example, if a network in class A is divided into four subnets, each subnet has a prefix of $n_{sub} = 10$. At the same time, if all of the addresses in a network are not used, subnetting allows the addresses to be divided among several organizations.
- This idea did not work because most large organizations were not happy about dividing the block and giving some of the unused addresses to smaller organizations.
- While subnetting was devised to divide a large block into smaller ones, supernetting was devised to combine several class C blocks into a larger block to be attractive to organizations that need more than the 256 addresses available in a class C block. This idea did not work either because it makes the routing of packets more difficult.

Advantage of Classful Addressing

- Although classful addressing had several problems and became obsolete, it had one advantage: Given an address, we can easily find the class of the address and, since the prefix length for each class is fixed, we can find the prefix length immediately.
- In other words, the prefix length in classful addressing is inherent in the address; no extra information is needed to extract the prefix and the suffix.

Classless Addressing

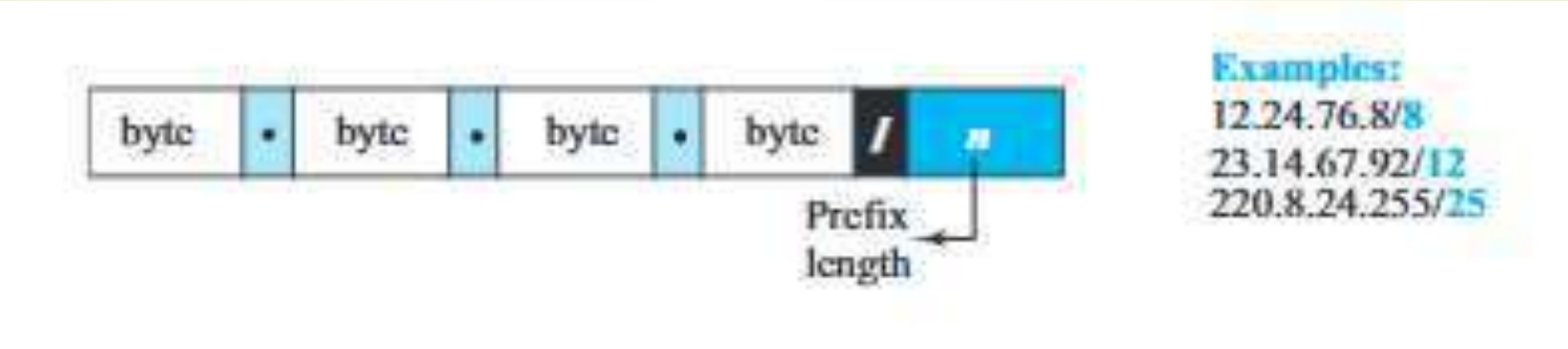
- Subnetting and supernetting in classful addressing did not really solve the address depletion problem.
- The larger address space, however, requires that the length of IP addresses also be increased, which means the format of the IP packets needs to be changed.
- The short-term solution still uses IPv4 addresses, but it is called classless addressing.
- In other words, the class privilege was removed from the distribution to compensate for the address depletion.
- In 1996, the Internet authorities announced a new architecture called classless addressing. In classless addressing, variable-length blocks are used that belong to no classes. We can have a block of 1 address, 2 addresses, 4 addresses, 128 addresses, and so on.



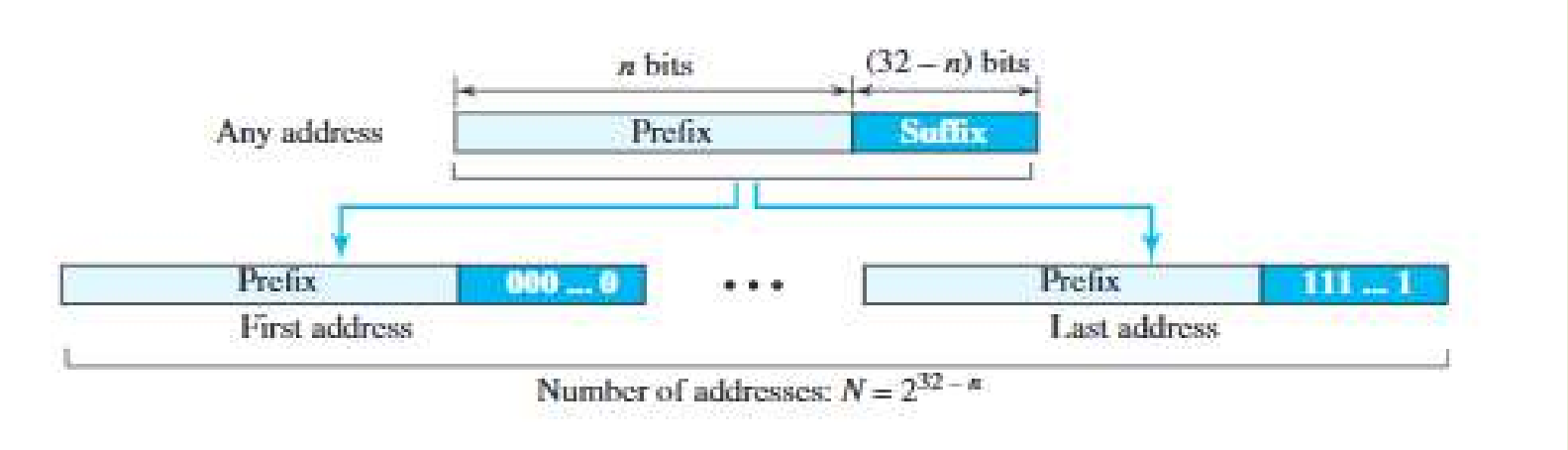
- Unlike classful addressing, the prefix length in classless addressing is variable. We can have a prefix length that ranges from 0 to 32.
- The size of the network is inversely proportional to the length of the prefix. A small prefix means a larger network; a large prefix means a smaller network.
- The idea of classless addressing can be easily applied to classful addressing. An address in class A can be thought of as a classless address in which the prefix length is 8.
- An address in class B can be thought of as a classless address in which the prefix is 16, and so on. In other words, classful addressing is a special case of classless addressing.

Prefix Length: Slash Notation

- In classless addressing is how to find the prefix length if an address is given.
- Since the prefix length is not inherent in the address, we need to separately give the length of the prefix. In this case, the prefix length, n, is added to the address, separated by a slash.
- The notation is informally referred to as slash notation and formally as classless interdomain routing or CIDR strategy. An address in classless addressing can then be represented as shown



Extracting Information from an Address



- Given any address in the block, we normally like to know three pieces of information about the block to which the address belongs: the number of addresses, the first address in the block, and the last address. Since the value of prefix length, n, is given, we can easily find these three pieces of information.

1. The number of addresses in the block is found as $N = 2^{32-n}$.
2. To find the first address, we keep the n leftmost bits and set the (32 −n) rightmost bits all to 0s.
3. To find the last address, we keep the n leftmost bits and set the (32 −n) rightmost bits all to 1s.

Example:

- A classless address is given as 167.199.170.82/27. We can find the above three pieces of information as follows. The number of addresses in the network is $2^{32-n} = 2^5 = 32$ addresses.

Solution:

The first address can be found by keeping the first 27 bits and changing the rest of the bits to 0s.

Address: 167.199.170.82/27	10100111	11000111	10101010	01010010
First address: 167.199.170.64/27	10100111	11000111	10101010	01000000

- The last address can be found by keeping the first 27 bits and changing the rest of the bits to 1s.

Address: 167.199.170.82/27	10100111	11000111	10101010	01011111
Last address: 167.199.170.95/27	10100111	11000111	10101010	01011111

Address Mask

- Another way to find the first and last addresses in the block is to use the address mask.
- The address mask is a 32-bit number in which the n leftmost bits are set to 1s and the rest of the bits (32 −n) are set to 0s. A computer can easily find the address mask because it is the complement of $(2^{32-n} - 1)$.
- The reason for defining a mask in this way is that it can be used by a computer program to extract the information in a block, using the three bit-wise operations NOT, AND, and OR.

1. The number of addresses in the block N =NOT(mask) +1.
2. The first address in the block =(Any address in the block) AND(mask).
3. The last address in the block =(Any address in the block) OR[(NOT(mask)].

Example

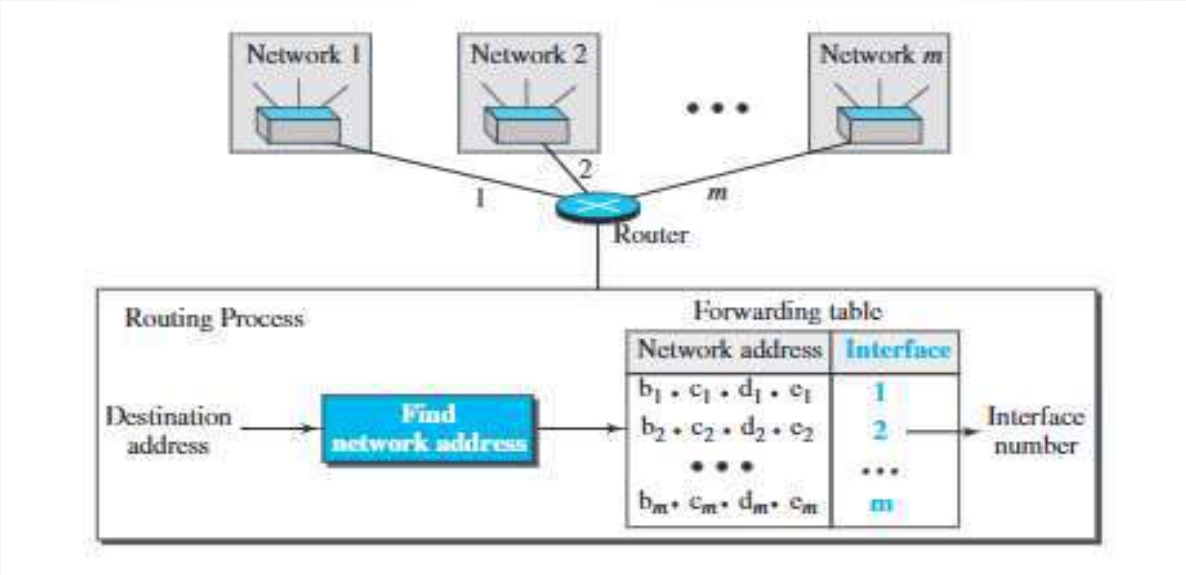
- The address is given as 167.199.170.82/27 using the mask. The mask in dotted-decimal notation is 256.256.256.224. The AND, OR, and NOT operations can be applied to individual bytes using calculators and applets at the book website.

Solution:

Number of addresses in the block:	N = NOT (mask) + 1 = 0.0.0.31 + 1 = 32 addresses
First address:	First = (address) AND (mask) = 167.199.170.82
Last address:	Last = (address) OR (NOT mask) = 167.199.170.255

Network Address

- The first address, the network address, is particularly important because it is used in routing a packet to its destination network. For the moment, let us assume that an internet is made of m networks and a router with m interfaces.
- When a packet arrives at the router from an y source host, the router needs to know to which network the packet should be sent: from which interface the packet should be sent out.



- After the network address has been found, the router consults its forwarding table to find the corresponding interface from which the packet should be sent out.
- The network address is actually the identifier of the network; each network is identified by its network address.

Block Allocation

- The next issue in classless addressing is block allocation. How are the blocks allocated?
- The ultimate responsibility of block allocation is given to a global authority called the Internet Corporation for Assigned Names and Numbers (ICANN).
- However, ICANN does not normally allocate addresses to individual Internet users. It assigns a large block of addresses to an ISP (or a larger organization that is considered an ISP in this case).
- For the proper operation of the CIDR(Classless Inter-Domain Routing), two restrictions need to be applied to the allocated block.

1. The number of requested addresses, N, needs to be a power of 2. The reason is that $N = 2^{32-n}$ or $n = 32 - \log_2 N$. If N is not a power of 2, we cannot have an integer value for n.
2. The requested block needs to be allocated where there is an adequate number of contiguous addresses available in the address space. However, there is a restriction on choosing the first address in the block. The first address needs to be divisible by the number of addresses in the block. The reason is that the first address needs to be the prefix followed by $(32 - n)$ number of 0s. The decimal value of the first address is then

$$\text{first address} = (\text{prefix in decimal}) \times 2^{32-n} = (\text{prefix in decimal}) \times N.$$

Special Addresses

- Five special addresses that are used for special purposes:

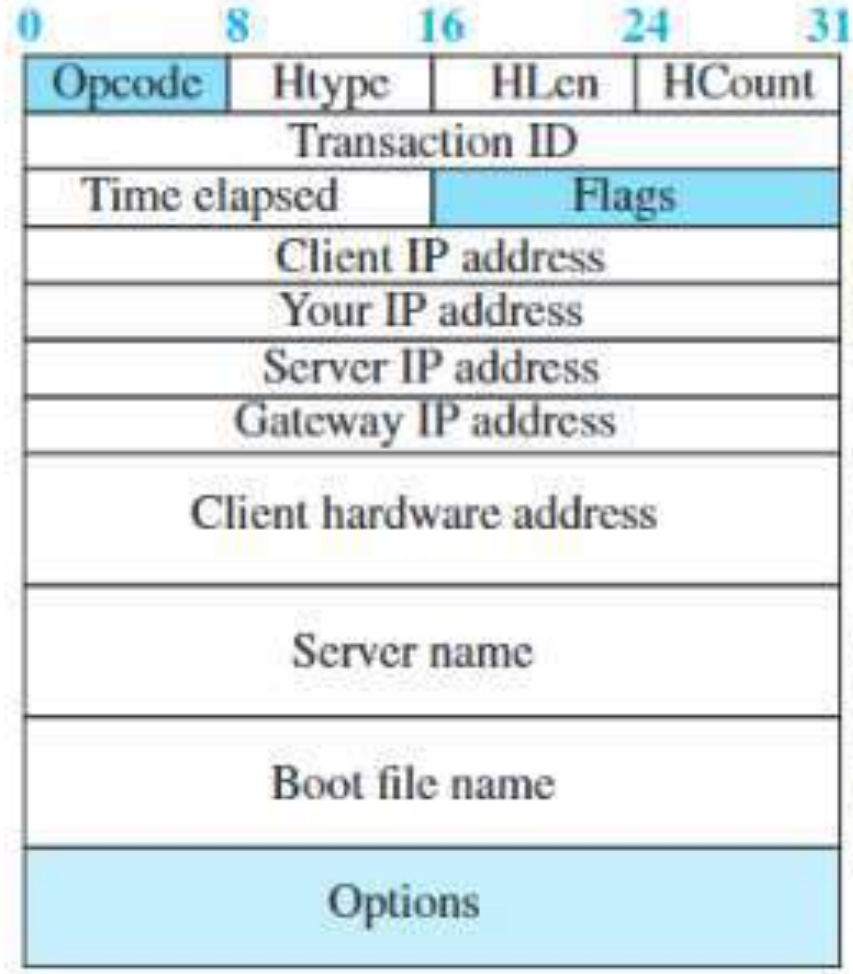
1. **This-host address:** The only address in the block 0.0.0.0/32 is called the this-host address. It is used when-ever a host needs to send an IP datagram but it does not know its own address to use as the source address.
2. **Limited-broadcast address:** The only address in the block 255.255.255.255/32 is called the limited-broadcast address. It is used whenever a router or a host needs to send a datagram to all devices in a network
3. **Loopback address:** The block 127.0.0.0/8 is called the loopback address. A packet with one of the addresses in this block as the destination address never leaves the host; it will remain in the host. Any address in the block is used to test a piece of software in the machine.
4. **Private addresses:** Four blocks are assigned as private addresses: 10.0.0.0/8, 172.16.0.0/12, 92.168.0.0/16,and 169.254.0.0/16.
5. **Multicast addresses:** The block 224.0.0.0/4 is reserved for multicast addresses

Dynamic Host Configuration Protocol (DHCP)

- After a block of addresses are assigned to an organization, the network administration can manually assign addresses to the individual hosts or routers.
- However, address assignment in an organization can be done automatically using the Dynamic Host Configuration Protocol (DHCP).
- DHCP is an application-layer program, using the client-server paradigm, that actually helps TCP/IP at the network layer.
- DHCP has found such widespread use in the Internet that it is often called a plug- and-play protocol.
- A network manager can configure DHCP to assign permanent IP addresses to the host and routers. DHCP can also be configured to provide temporary, on demand, IP addresses to hosts.

DHCP Message Format

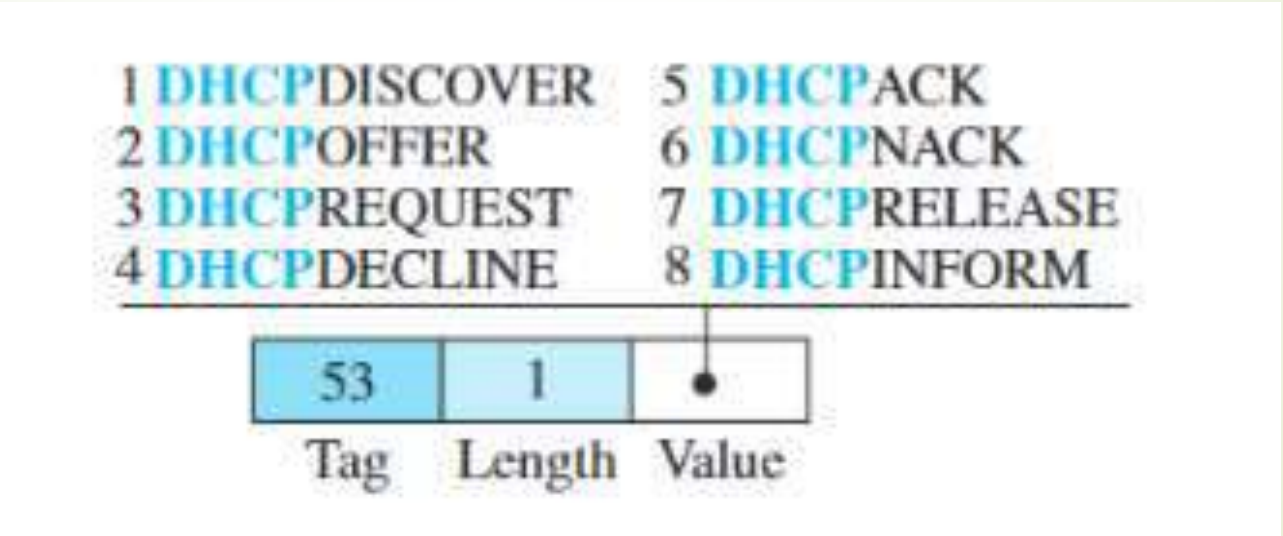
- DHCP is a client-server protocol in which the client sends a request message and the server returns a response message.



Fields:

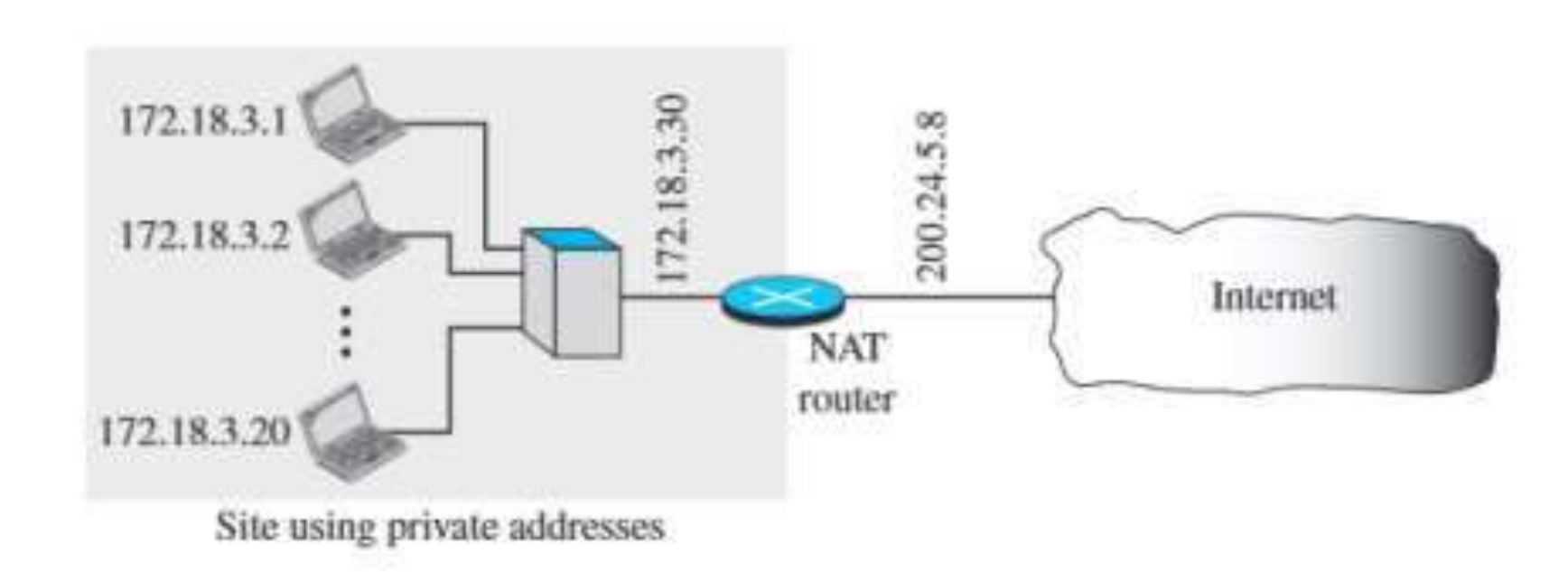
Opcode: Operation code, request (1) or reply (2)
Htype: Hardware type (Ethernet, ...)
HLen: Length of hardware address
HCount: Maximum number of hops the packet can travel
Transaction ID: An integer set by the client and repeated by the server
Time elapsed: The number of seconds since the client started to boot
Flags: First bit defines unicast (0) or multicast (1); other 15 bits not used
Client IP address: Set to 0 if the client does not know it
Your IP address: The client IP address sent by the server
Server IP address: A broadcast IP address if client does not know it
Gateway IP address: The address of default router
Server name: A 64-byte domain name of the server
Boot file name: A 128-byte file name holding extra information
Options: A 64-byte field with dual purpose described in text

- The 64-byte option field has a dual purpose. It can carry either additional information or some specific vendor information. The server uses a number, called a magic cookie, in the format of an IP address with the value of 99.130.83.99.
- An option is composed of three fields: a 1-byte tag field, a 1-byte length field, and a variable-length value field.



Network Address Translation (NAT)

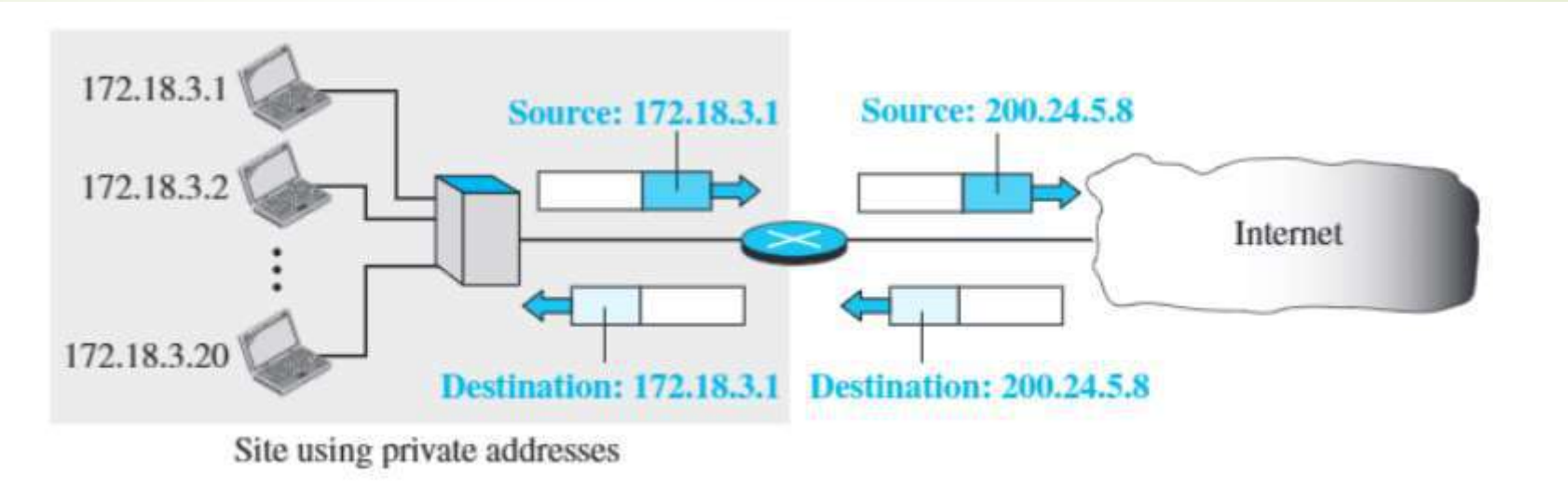
- The distribution of addresses through ISPs has created a new problem. Assume that an ISP has granted a small range of addresses to a small business or a household.
- If the business grows or the household needs a larger range, the ISP may not be able to grant the demand because the addresses before and after the range may have already been allocated to other networks.
- The technology allows a site to use a set of private addresses for internal communication and a set of global Internet addresses (at least one) for communication with the rest of the world.
- The site must have only one connection to the global Internet through a NAT-capable router that runs NAT software.



- As the figure shows, the private network uses private addresses. The router that connects the network to the global address uses one private address and one global address.
- The private network is invisible to the rest of the Internet; the rest of the Internet sees only the NAT router with the address 200.24.5.8.

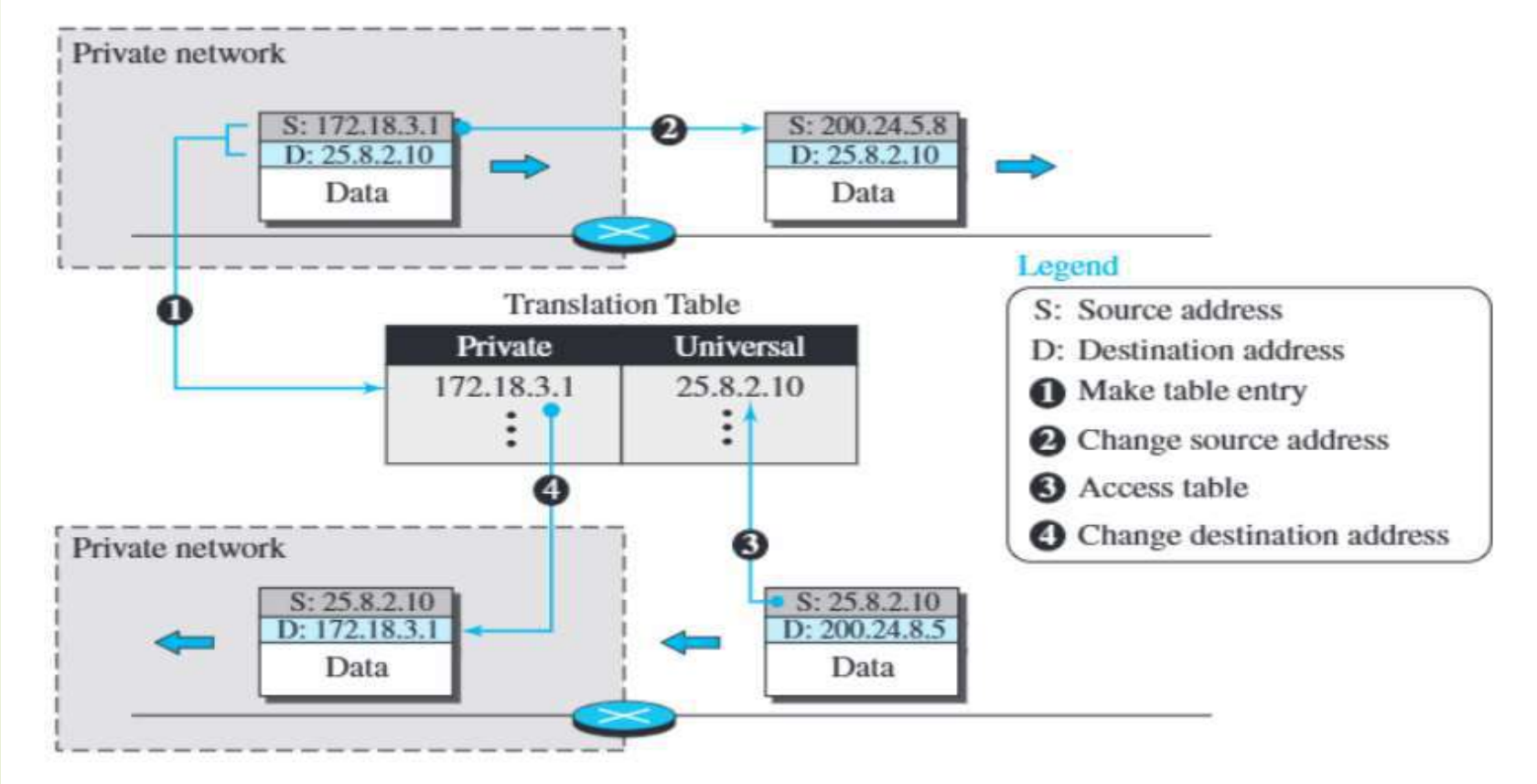
Address Translation

- All of the outgoing packets go through the NAT router, which replaces the source address in the packet with the global NAT address.
- All incoming packets also pass through the NAT router, which replaces the destination address in the packet (the NAT router global address) with the appropriate private address.



Translation Table

- The reader may have noticed that translating the source addresses for an outgoing packet is straightforward. But how does the NAT router know the destination address for a packet coming from the Internet?
- There may be tens or hundreds of private IP addresses, each belonging to one specific host. The problem is solved if the NAT router has a translation table.



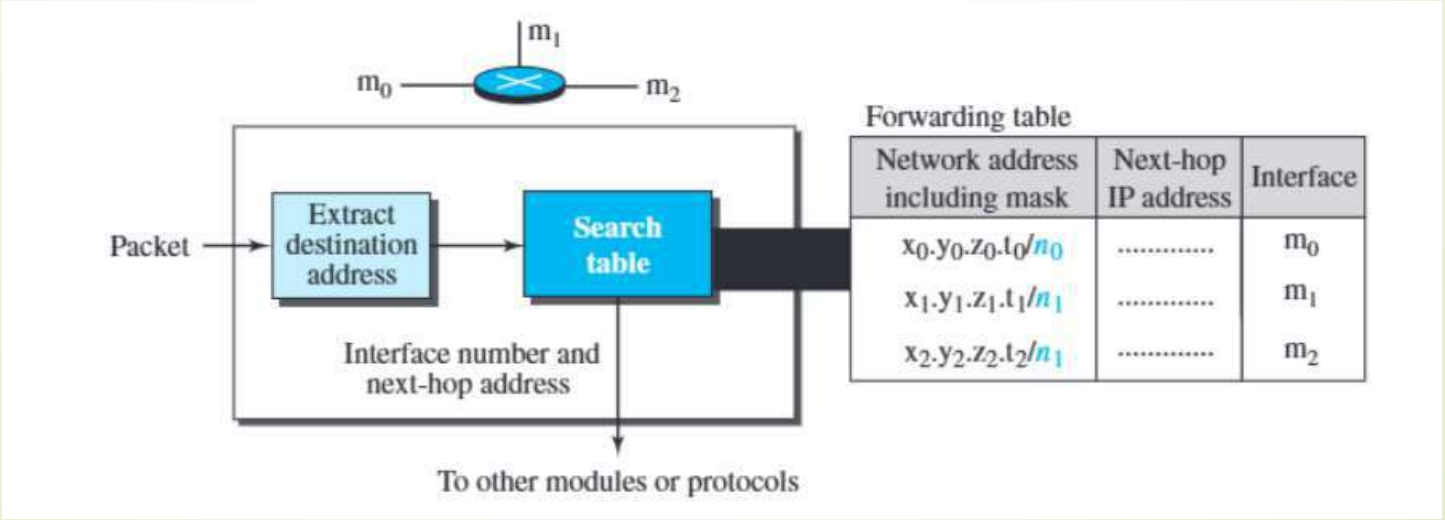
- In its simplest form, a translation table has only two columns: the private address and the external address (destination address of the packet). When the router translates the source address of the outgoing packet, it also makes note of the destination address.
- where the packet is going. When the response comes back from the destination, the router uses the source address of the packet (as the external address) to find the private address of the packet.

FORWARDING OF IP PACKETS

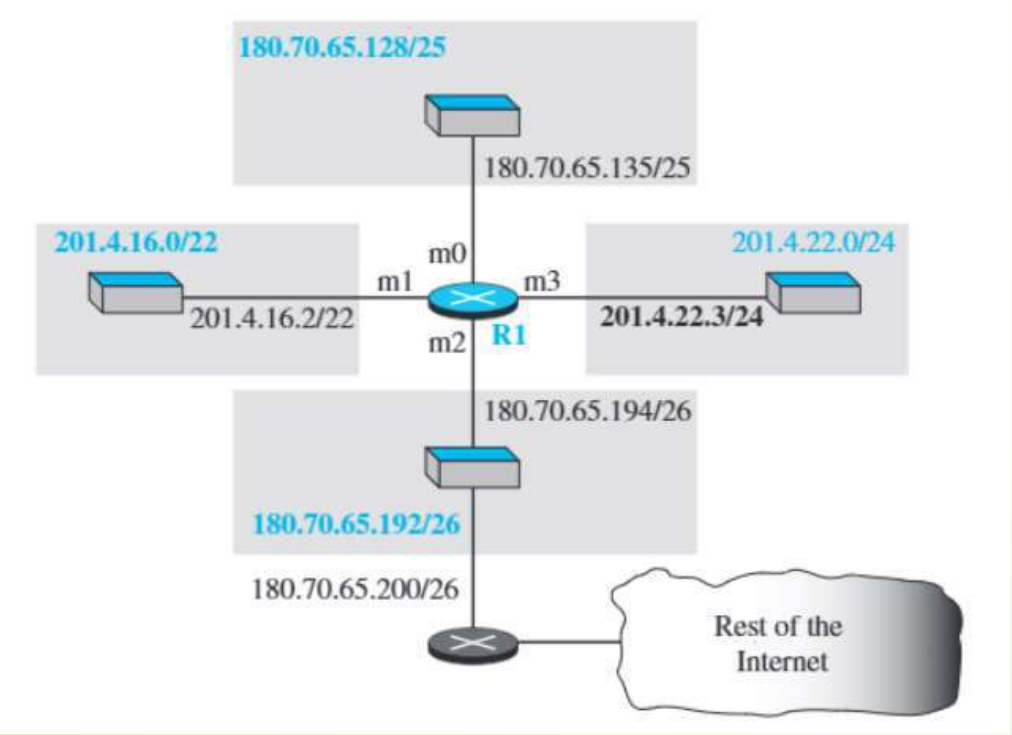
- Forwarding means to place the packet in its route to its destination.
- Since the Internet today is made of a combination of links (networks), forwarding means to deliver the packet to the next hop (which can be the final destination or the intermediate connecting device).
- Although the IP protocol was originally designed as a connectionless protocol, today the tendency is to change it to a connection-oriented protocol.
- When IP is used as a connectionless protocol,
  - forwarding is based on the destination address of the IP datagram;
  - when the IP is used as a connection-oriented protocol, forwarding is based on the label attached to an IP datagram.

Forwarding Based on Destination Address

- In this case, forwarding requires a host or a router to have a forwarding table.
- When a host has a packet to send or when a router has received a packet to be forwarded, it looks at this table to find the next hop to deliver the packet to.
- The table needs to be searched based on the network address (first address in the block). Unfortunately, the destination address in the packet gives no clue about the network address.
- To solve the problem, we need to include the mask (/n) in the table. In other words, a classless forwarding table needs to include four pieces of information: the mask, the network address, the interface number, and the IP address of the next router.



- The job of the forwarding module is to search the table, row by row. In each row, the n leftmost bits of the destination address (prefix) are kept and the rest of the bits (suffix) are set to 0s.
- If the resulting address (which we call the network address), matches with the address in the first column, the information in the next two columns is extracted; otherwise the search continues.
- Example: Make a forwarding table for router R1 using the configuration in Figure.
-



Forwarding table for router R1 in Figure

Network address/mask	Next hop	Interface
180.70.65.192/26	—	m2
180.70.65.128/25	—	m0
201.4.22.0/24	—	m3
201.4.16.0/22	—	m1
Default	180.70.65.200	m2

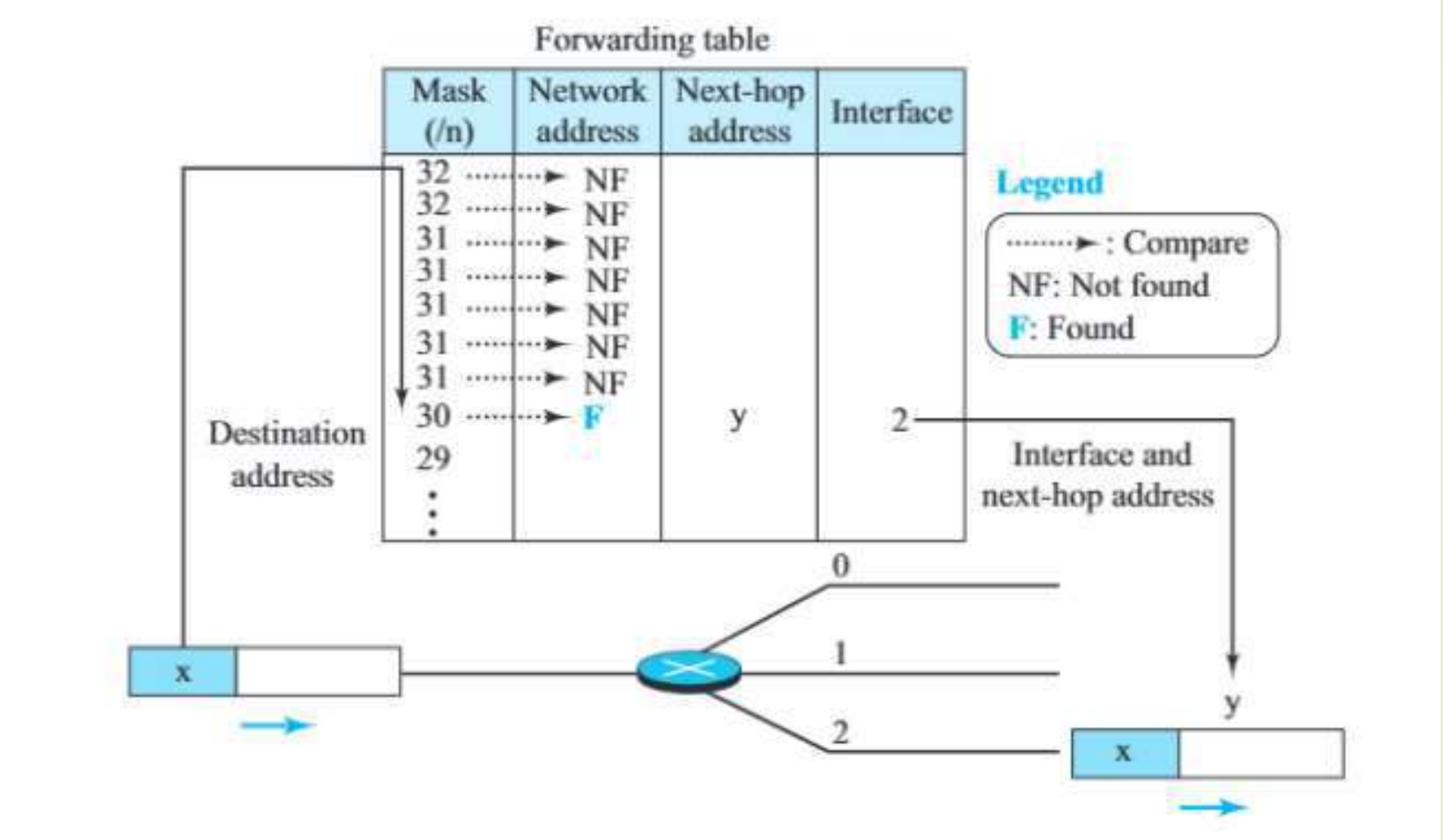
Forwarding table for router R1 in Figure using prefix bits

Leftmost bits in the destination address	*Next hop*	*Interface*
10110100 01000110 01000001 11	—	m2
10110100 01000110 01000001 1	—	m0
11001001 00000100 00011100	—	m3
11001001 00000100 000100	—	m1
Default	180.70.65.200	m2

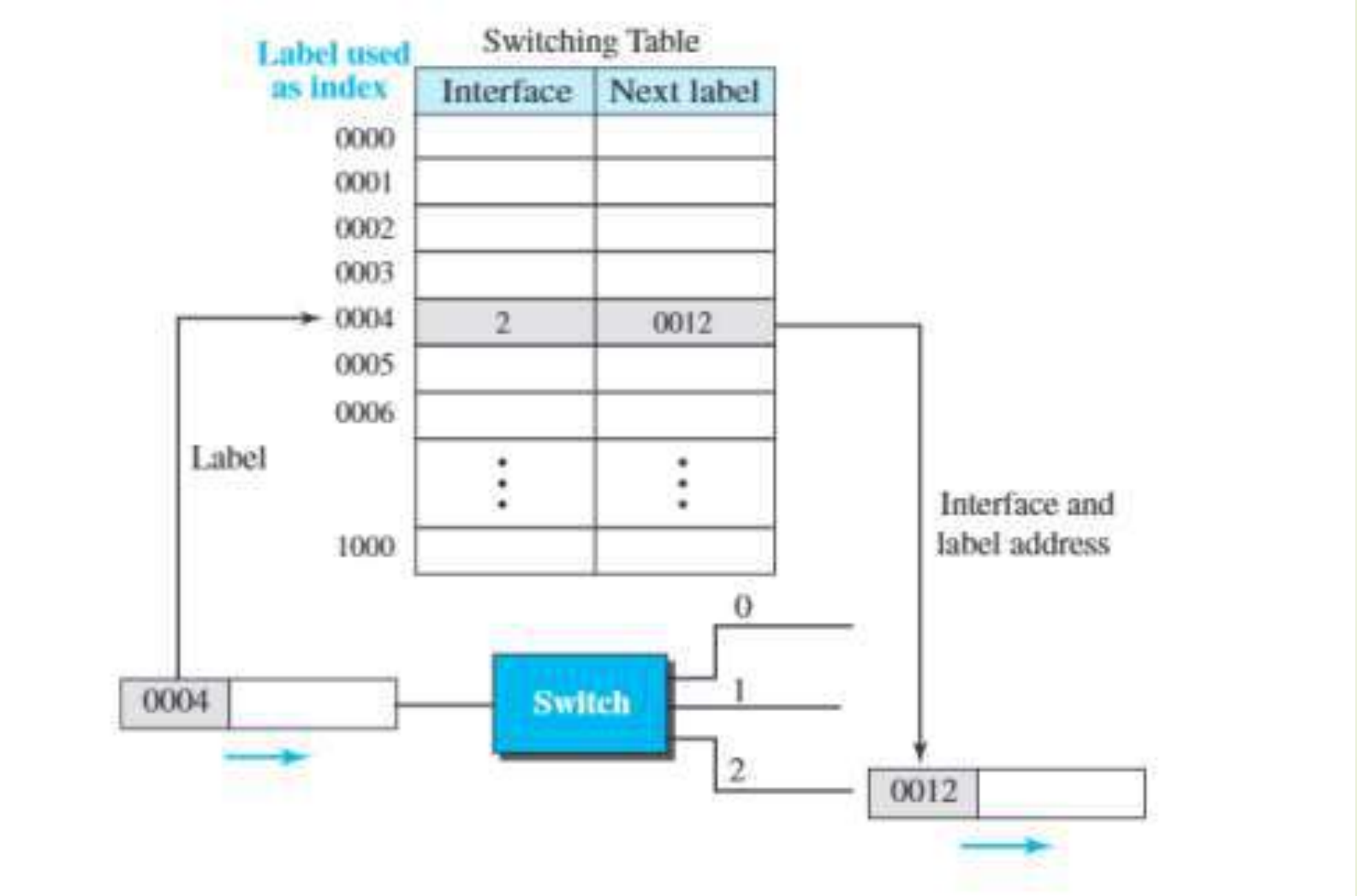
- When a packet arrives whose leftmost 26 bits in the destination address match the bits in the first row, the packet is sent out from interface m2.
- When a packet arrives whose leftmost 25 bits in the address match the bits in the second row, the packet is sent out from interface m0, and so on.
- The table clearly shows that the first row has the longest prefix and the fourth row has the shortest prefix. The longer prefix means a smaller range of addresses; the shorter prefix means a larger range of addresses.

Forwarding Based on Label

- In a connectionless network (datagram approach), a router forwards a packet based on the destination address in the header of the packet.
- On the other hand, in a connection-oriented network (virtual-circuit approach), a switch forwards a packet based on the label attached to the packet.
- Routing is normally based on searching the contents of a table; switching can be done by accessing a table using an index. In other words, routing involves searching; switching involves accessing.



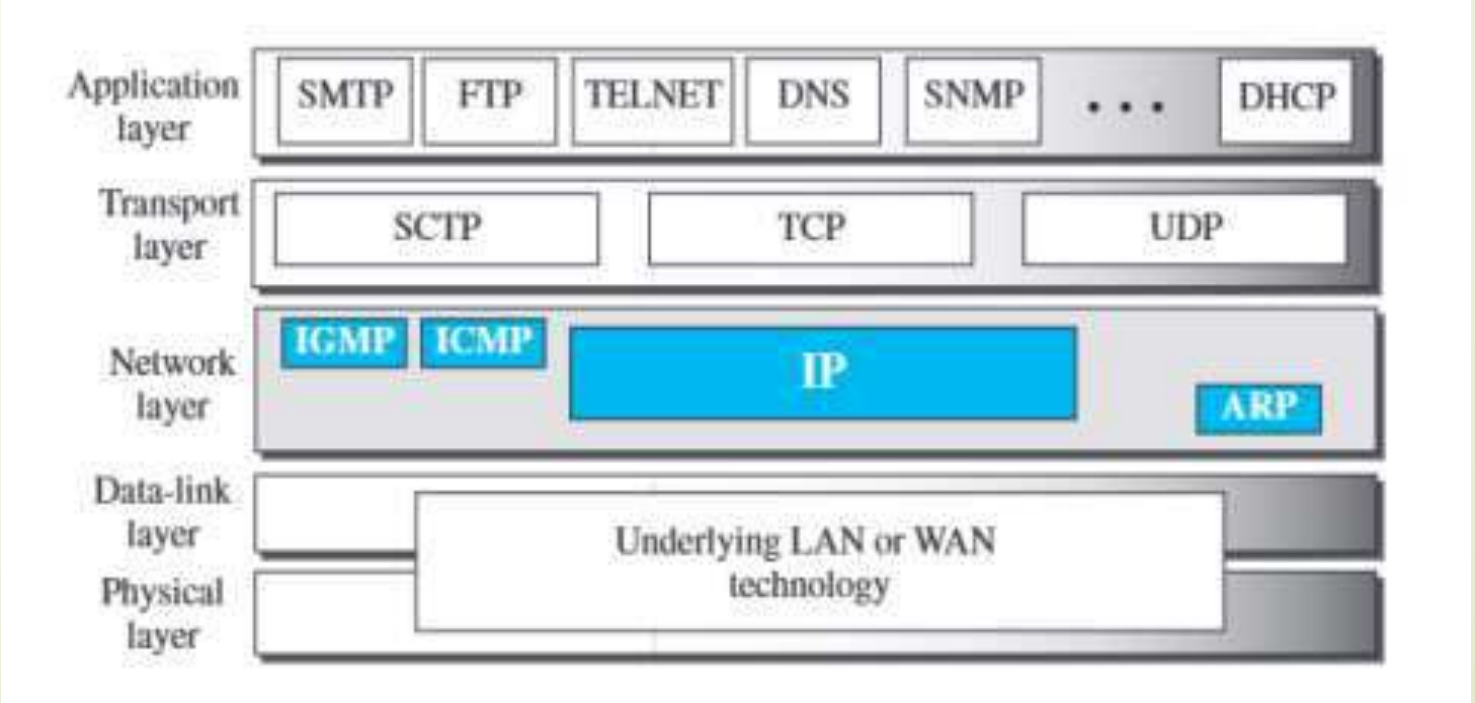
Forwarding based on destination address



Forwarding based on label

Network Layer Protocols

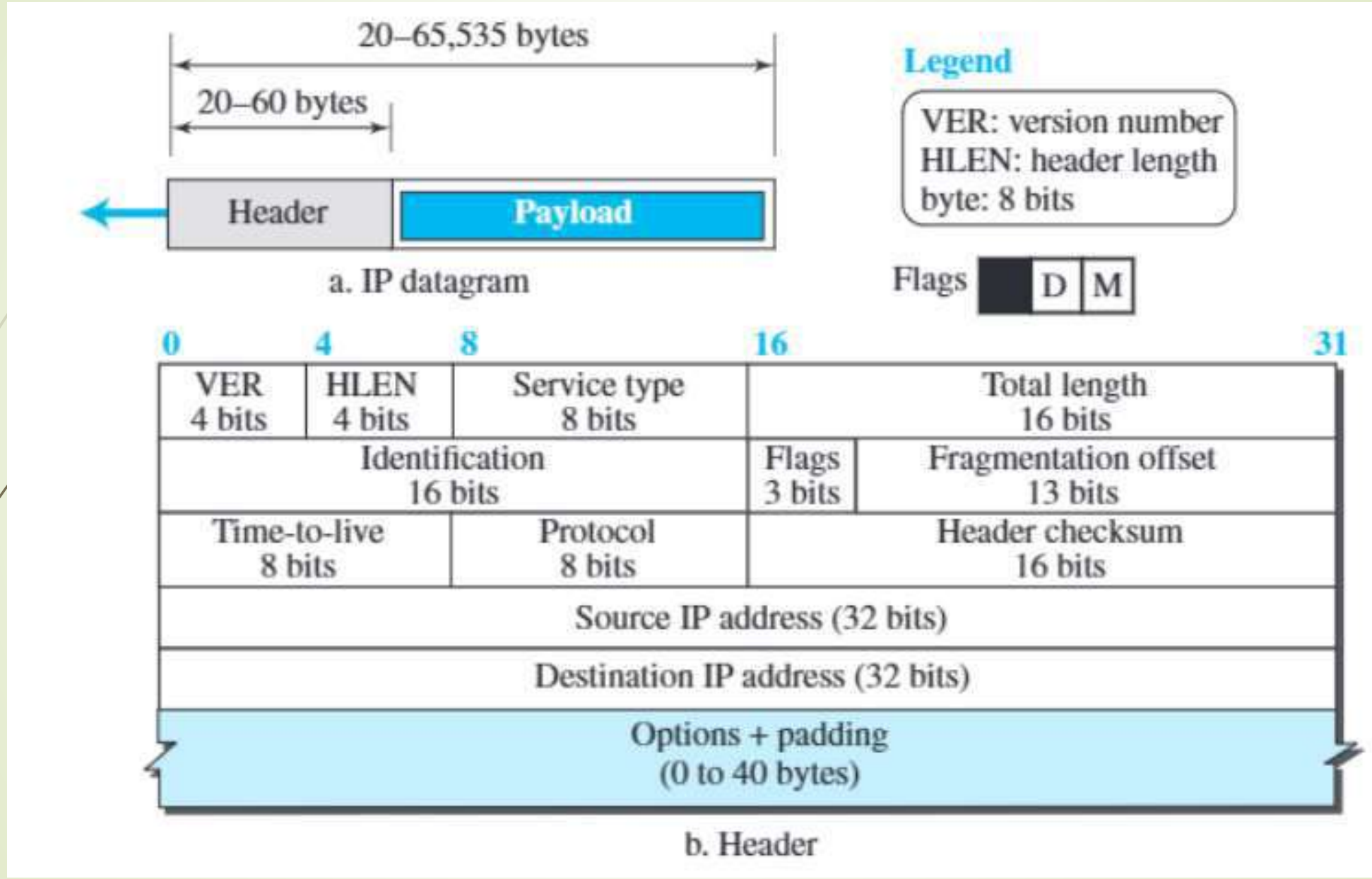
- Communication at the network layer is host-to-host (computer-to-computer); a computer somewhere in the world needs to communicate with another computer somewhere else in the world through the Internet.
- The packet transmitted by the sending computer may pass through several LANs or WANs before reaching the destination computer.
- The term IP address refers to the logical address in the network layer of the TCP/IP protocol suite.



- The main protocol, Internet Protocol version 4 (IPv4), is responsible for packetizing, forwarding, and delivery of a packet at the network layer.
- The Internet Control Message Protocol version 4 (ICMPv4) helps IPv4 to handle some errors that may occur in the network-layer delivery.
- The Internet Group Management Protocol (IGMP) is used to help IPv4 in multicasting.
- The Address Resolution Protocol (ARP) is used to glue the network and data-link layers in mapping network-layer addresses to link-layer addresses.

Datagram Format

- The Internet Protocol version 4 (IPv4) is the delivery mechanism used by the TCP/IP protocols.
- Packets used by the IP are called datagrams.
- A datagram is a variable-length packet consisting of two parts: header and payload (data).
- The header is 20 to 60 bytes in length and contains information essential to routing and delivery.


20–65,535 bytes
20–60 bytes
Header
Payload
a. IP datagram
Legend
VER: version number
HLEN: header length
byte: 8 bits
Flags
D
M
0
4
8
16
31
VER
4 bits
HLEN
4 bits
Service type
8 bits
Total length
16 bits
Identification
16 bits
Flags
3 bits
Fragmentation offset
13 bits
Time-to-live
8 bits
Protocol
8 bits
Header checksum
16 bits
Source IP address (32 bits)
Destination IP address (32 bits)
Options + padding
(0 to 40 bytes)
b. Header

A brief description of each field is in order:

1. **Version Number**: The 4-bit version number (VER) field defines the version of the IPv4 protocol, which, obviously, has the value of 4.
2. **Header Length**: The 4-bit header length (HLEN) field defines the total length of the datagram header in 4-byte words. The IPv4 datagram has a variable-length header. When a device receives a datagram, it needs to know when the header stops and the data, which is encapsulated in the packet, starts. The total length is divided by 4 and the value is inserted in the field. The receiver needs to multiply the value of this field by 4 to find the total length.
3. **Service Type**: In the original design of the IP header, this field was referred to as type of service (TOS), which defined how the datagram should be handled.

4. Total Length: This 16-bit field defines the total length (header plus data) of the IP datagram in bytes.

- A 16-bit number can define a total length of up to 65,535 (when all bits are 1s).
- However, the size of the datagram is normally much less than this. This field helps the receiving device to know when the packet has completely arrived.
- To find the length of the data coming from the upper layer, subtract the header length from the total length.
- The header length can be found by multiplying the value in the HLEN field by 4.

$$\text{Length of data} = \text{total length} - (\text{HLEN}) \times 4$$

5. Identification, Flags, and Fragmentation Offset: These three fields are related to the fragmentation of the IP datagram when the size of the datagram is larger than the underlying network.

6. **Time-to-live:** Due to some malfunctioning of routing protocols a datagram may be circulating in the Internet, visiting some networks over and over without reaching the destination.

7. Protocol: In TCP/IP, the data section of a packet, called the payload, carries the whole packet from another protocol. A datagram, for example, can carry a packet belonging to any transport-layer protocol such as UDP or TCP. A datagram can also carry a packet from other protocols that directly use the service of the IP, such as some routing protocols or some auxiliary protocols.

8.Header checksum: IP is not a reliable protocol; it does not check whether the payload carried by a datagram is corrupted during the transmission. IP puts the burden of error checking of the payload on the protocol that owns the payload, such as UDP or TCP

9. Source and Destination Addresses: These 32-bit source and destination address fields define the IP address of the source and destination respectively. The source host should know its IP address. The destination IP address is either known by the protocol that uses the service of IP or is provided by the DNS.

10. Options: A datagram header can have up to 40 bytes of options. Options can be used for network testing and debugging. Although options are not a required part of the IP header, option processing is required of the IP software.

11. Payload: Payload, or data, is the main reason for creating a datagram. Payload is the packet coming from other protocols that use the service of IP.

Example 19.1

An IPv4 packet has arrived with the first 8 bits as $(01000010)_2$. The receiver discards the packet. Why?

Solution

There is an error in this packet. The 4 leftmost bits $(0100)_2$ show the version, which is correct. The next 4 bits $(0010)_2$ show an invalid header length $(2 \times 4 = 8)$. The minimum number of bytes in the header must be 20. The packet has been corrupted in transmission.

Example 19.2

In an IPv4 packet, the value of HLEN is $(1000)_2$. How many bytes of options are being carried by this packet?

Solution

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

Example 19.3

In an IPv4 packet, the value of HLEN is 5, and the value of the total length field is $(0028)_{16}$. How many bytes of data are being carried by this packet?

Solution

The HLEN value is 5, which means the total number of bytes in the header is 5×4, or 20 bytes (no options). The total length is $(0028)_{16}$ or 40 bytes, which means the packet is carrying 20 bytes of data $(40 - 20)$.

Fragmentation

- A datagram can travel through different networks.
- Each router decapsulates the IP datagram from the frame it receives, processes it, and then encapsulates it in another frame.
- The format and size of the received (or sent) frames depend on the protocol used by the physical network through which the frame has just travelled (or going to travel).
- For example, if a router connects a LAN to a WAN, it receives a frame in the LAN format and sends a frame in the WAN format.

Maximum transfer unit (MTU)

- Each link-layer protocol has its own frame format.
- One of the features of each format is the maximum size of the payload that can be encapsulated.
- In other words, when a datagram is encapsulated in a frame, the total size of the datagram must be less than this maximum size, which is defined by the restrictions imposed by the hardware and software used in the network.

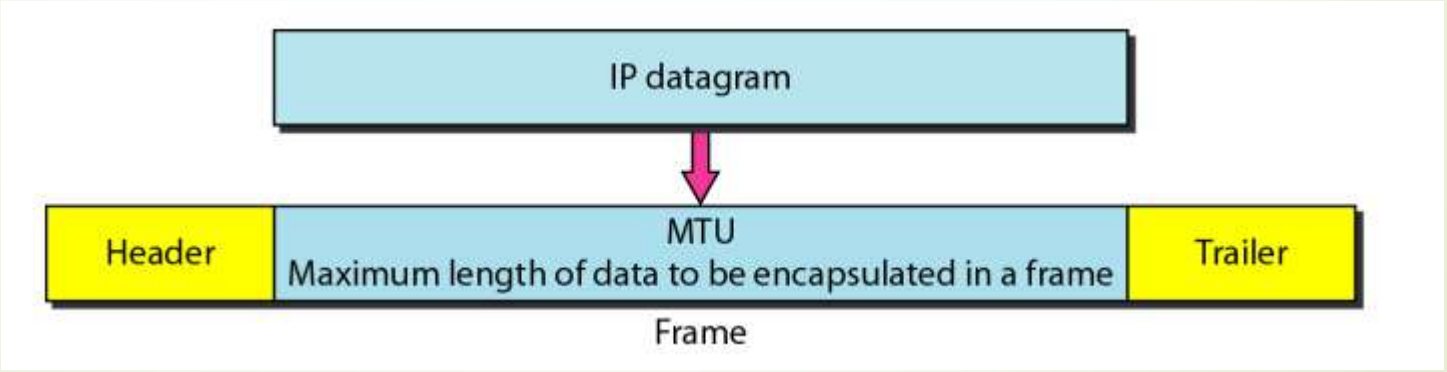


Figure: Maximum transfer unit (MTU)

- In order to make the IP protocol independent of the physical network, the designers decided to make the maximum length of the IP datagram equal to 65,535 bytes.
- This makes transmission more efficient if one day we use a link-layer protocol with an MTU of this size. However, for other physical networks, we must divide the datagram to make it possible for it to pass through these networks. This is called fragmentation.
- A fragmented datagram may itself be fragmented if it encounters a network with an even smaller MTU.
- In other words, a datagram may be fragmented several times before it reaches the final destination.

Fields Related to Fragmentation

- Three fields in an IP datagram are related to fragmentation:
  - Identification,
  - Flags, and
  - Fragmentation offset.
- **Identification**,
- When a datagram is fragmented, the value in the identification field is copied into all fragments.
- In other words, all fragments have the same identification number, which is also the same as the original datagram. The identification number helps the destination in reassembling the datagram.
- It knows that all fragments having the same identification value should be assembled into one datagram

Flags used in fragmentation

- When the payload of the IP datagram is fragmented, most parts of the header, with the exception of some options, must be copied by all fragments.
- The host or router that fragments a datagram must change the values of three fields: flags, fragmentation offset, and total length.
- The value of the checksum must be recalculated regardless of fragmentation.

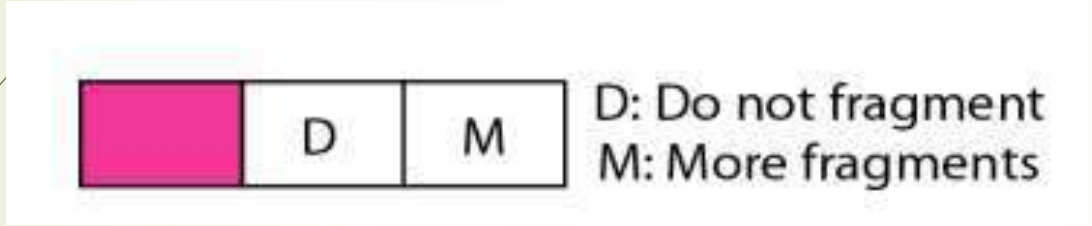


Figure: Flags in fragmentation

- If its value is 1, the machine must not fragment the datagram and If its value is 0, the datagram can be fragmented if necessary.
- The third bit is called the more fragment bit. If its value is 1, it means the datagram is not the last fragment; there are more fragments after this one. If its value is 0, it means this is the last or only fragment.

Fragmentation offset

- The 13-bit fragmentation offset field shows the relative position of this fragment with respect to the whole datagram.
- It is the offset of the data in the original datagram measured in units of 8 bytes.

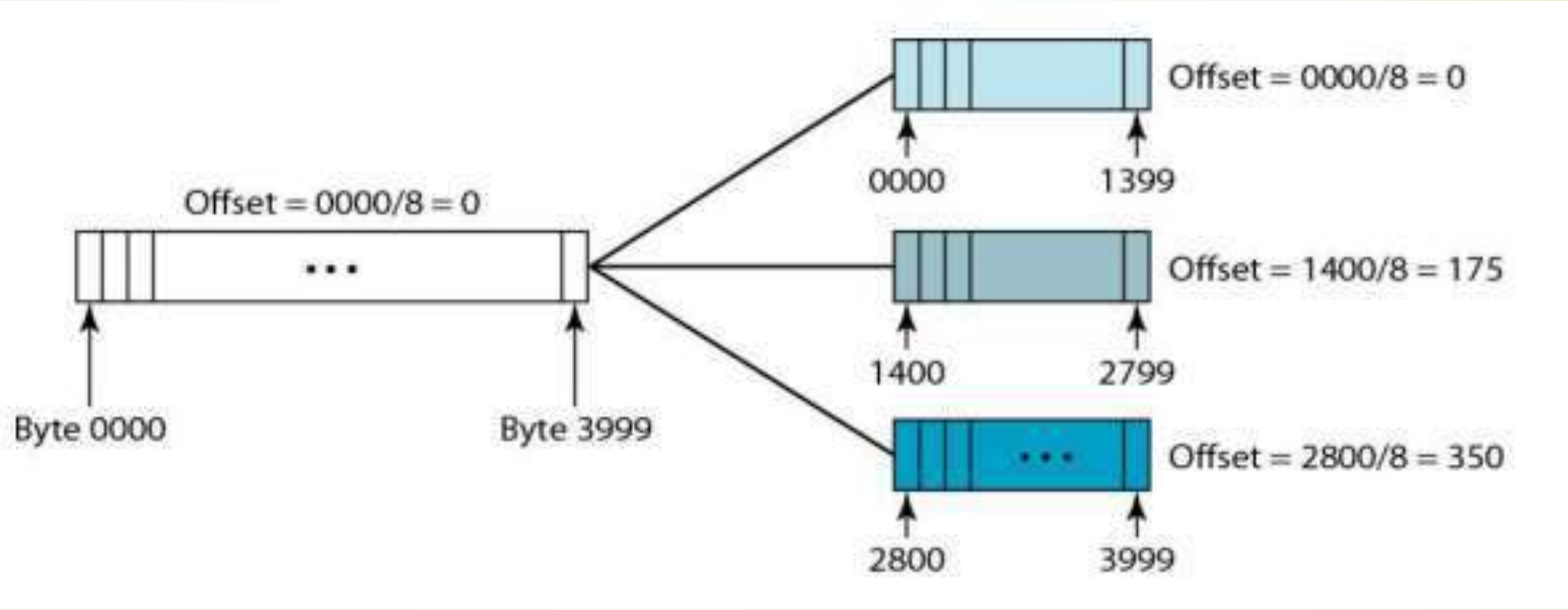


Figure: A fragmentation example.

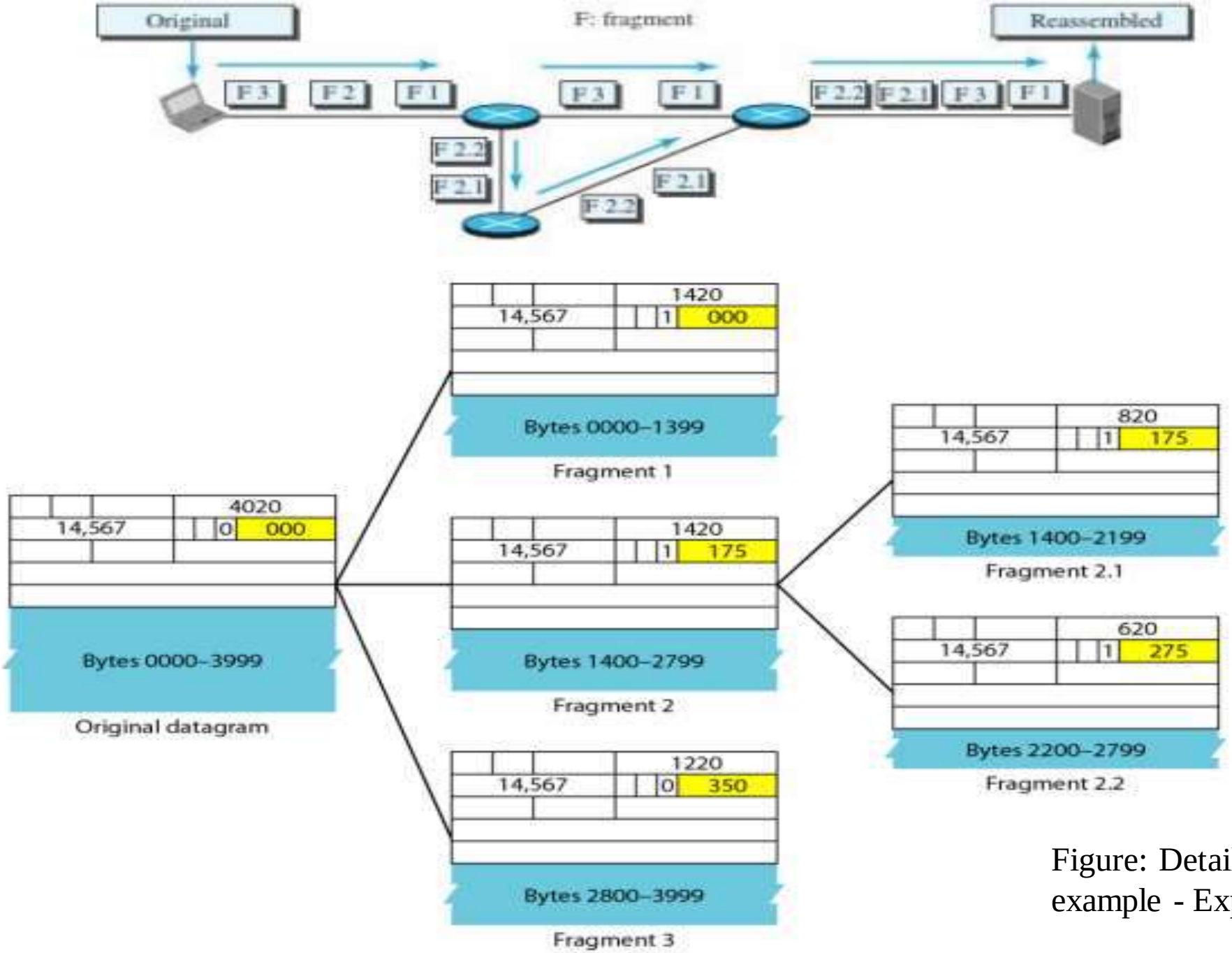


Figure: Detailed fragmentation example - Expanded view

Options

- Options, as the name implies, are not required for a datagram. They can be used for network testing and debugging.

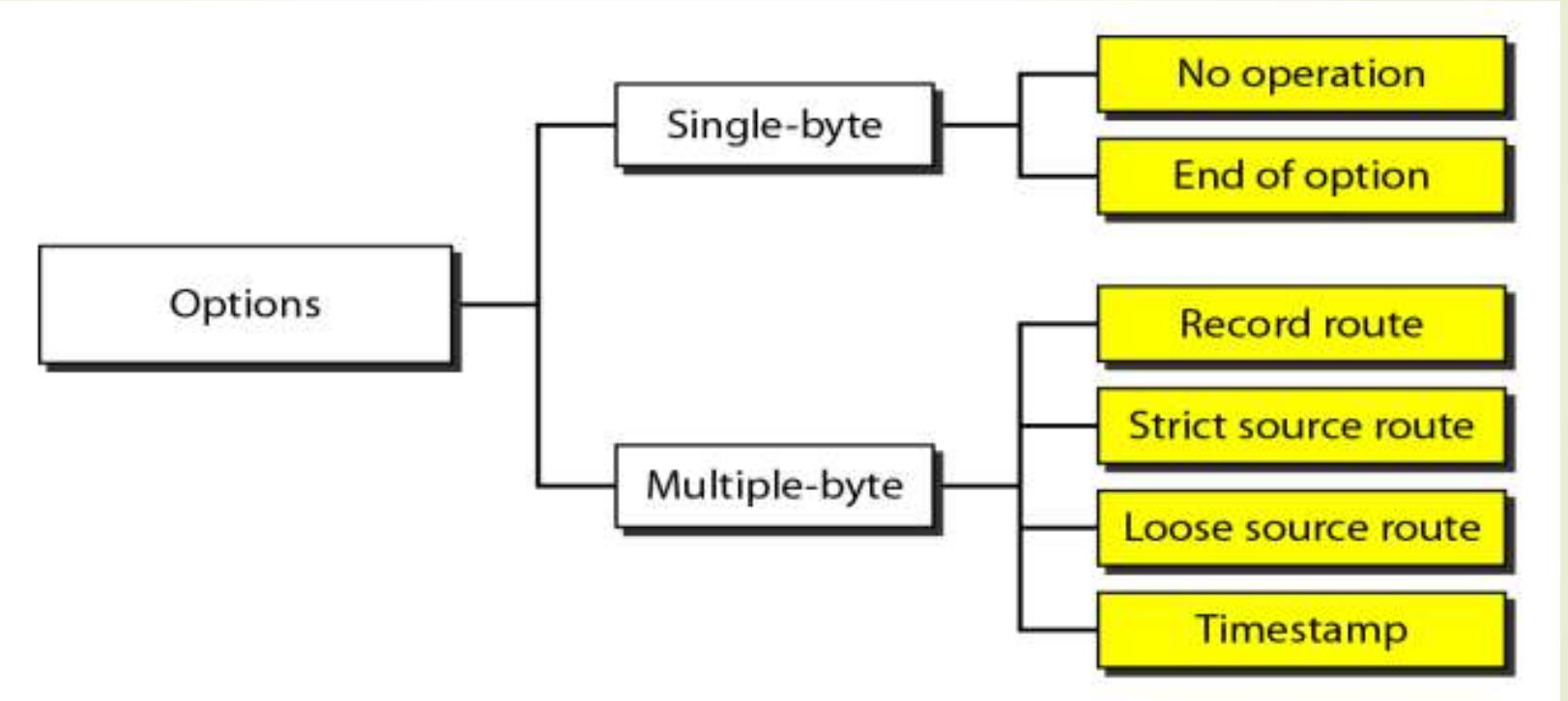


Figure: Classification of Options

- **Single-Byte Options**: There are two single-byte options. No Operation A no-operation option is a 1- byte option used as a filler between options. End of Option An end-of-option option is a 1-byte option used for padding at the end of the option field. It, however, can only be used as the last option.
- **Record Route** A record route option is used to record the Internet routers that handle the datagram. It can list up to nine router addresses. It can be used for debugging and management purposes.
- **Strict Source Route** A strict source route option is used by the source to predetermine a route for the datagram as it travels through the Internet. Here, the sender can choose a route with a specific type of service, such as minimum delay or maximum throughput. Alternatively, it may choose a route that is safer or more reliable for the sender's purpose.
- **Loose Source Route**: A loose source route option is similar to the strict source route, but it is less rigid. Each router in the list must be visited, but the datagram can visit other routers as well.
- **Timestamp**: A timestamp option is used to record the time of datagram processing by a router. The time is expressed in milliseconds from midnight, Universal time or Greenwich mean time. Knowing the time a datagram is processed can help users and managers track the behavior of the routers in the Internet

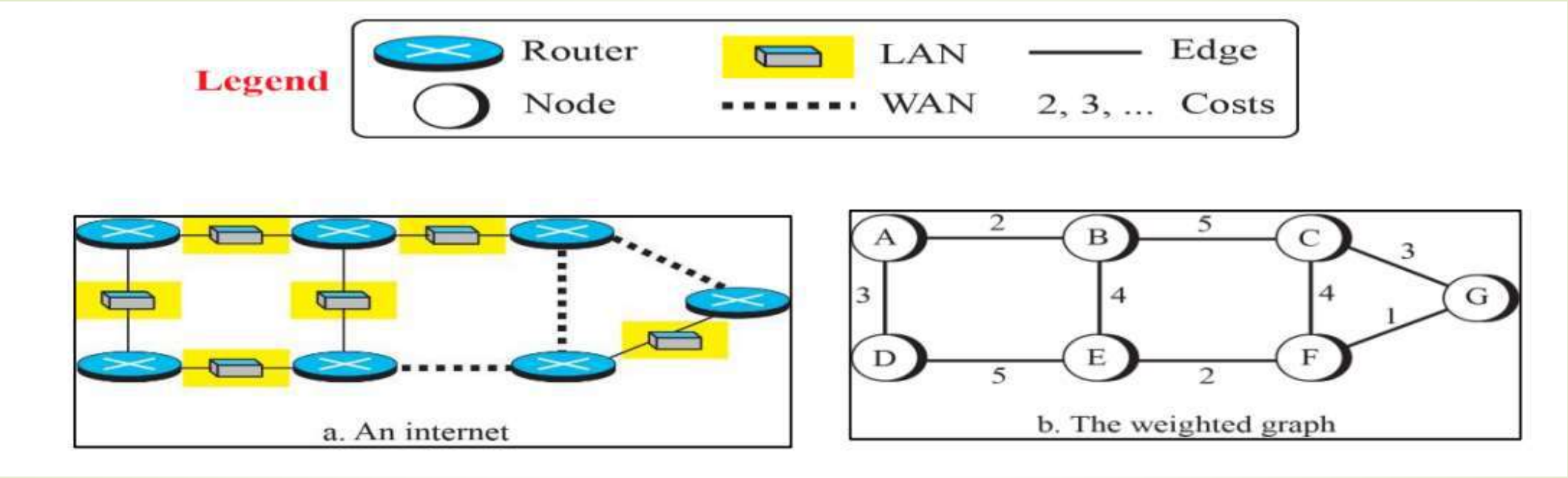
Security of IPv4 Datagrams

There are three security issues that are particularly applicable to the IP protocol:

- **Packet Sniffing**: An intruder may intercept an IP packet and make a copy of it. Packet sniffing is a passive attack, in which the attacker does not change the contents of the packet. This type of attack is very difficult to detect because the sender and the receiver may never know that the packet has been copied. Although packet sniffing cannot be stopped, encryption of the packet can make the attacker's effort useless. The attacker may still sniff the packet, but the content is not detectable.
- **Packet Modification**: The second type of attack is to modify the packet. The attacker intercepts the packet, changes its contents, and sends the new packet to the receiver. The receiver believes that the packet is coming from the original sender. This type of attack can be detected using a data integrity mechanism. The receiver, before opening and using the contents of the message, can use this mechanism to make sure that the packet has not been changed during the transmission.
- **IP Spoofing** : An attacker can masquerade as somebody else and create an IP packet that carries the source address of another computer. An attacker can send an IP packet to a bank pretending that it is coming from one of the customers.

Network Layer Protocols and Unicast Routing

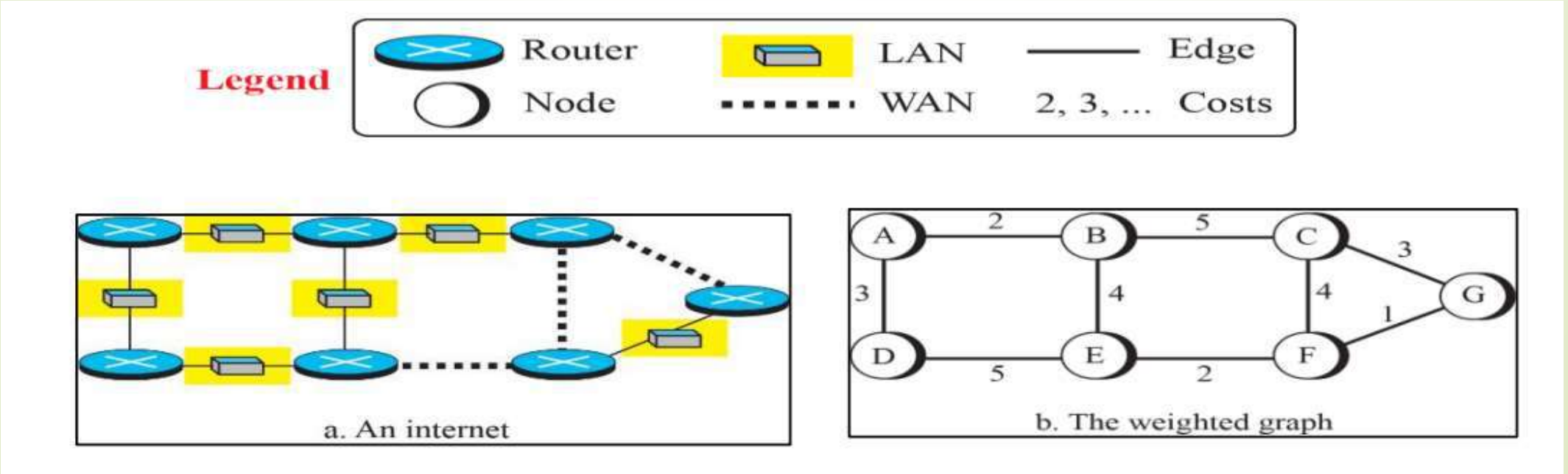
- Unicast routing in the Internet, with a large number of routers and a huge number of hosts, can be done only by using hierarchical routing: routing in several steps using different routing algorithms.
- In unicast routing, a packet is routed, hop by hop, from its source to its destination by the help of forwarding tables.
- The source host needs no forwarding table because it delivers its packet to the default router in its local network.
- The destination host needs no forwarding table either because it receives the packet from its default router in its local network.
- This means that only the routers that glue together the networks in the internet need forwarding tables. So, routing a packet from its source to its destination means routing the packet from a source router to a destination router.



- To find the best route, an internet can be modelled as a graph.
- To model an internet as a graph, we can think of each router as a node and each network between a pair of routers as an edge. An internet is, in fact, modelled as a weighted graph, in which each edge is associated with a cost.
- If a weighted graph is used to represent a geographical area, the nodes can be cities and the edges can be roads connecting the cities; the weights, in this case, are distances between cities.
- In routing, however, the cost of an edge has a different interpretation in different routing protocols. For the moment, we assume that there is a cost associated with each edge. If there is no edge between the nodes, the cost is infinity.

Least-Cost Routing

- When an internet is modeled as a weighted graph, one of the ways to interpret the best route from the source router to the destination router into find the least cost between the two.
- In other words, the source router chooses a route to the destination router in such a way that the total cost for the route is the least cost among all possible routes.
- This means that each router needs to find the least-cost route between itself and all the other routers to be able to route a packet using this criteria.



Least-Cost Trees

- If there are N routers in an internet, there are (N− 1) least-cost paths from each router to any other router.
- This means we need N × (N − 1) least-cost paths for the whole internet.
- If we have only 10 routers in an internet, we need 90 least-cost paths. A better way to see all of these paths is to combine them in a least-cost tree.
- A least-cost tree is a tree with the source router as the root that spans the whole graph (visits all other nodes) and in which the path between the root and any other node is the shortest.
- In this way, we can have only one shortest-path tree for each node; we have N least-cost trees for the whole internet.

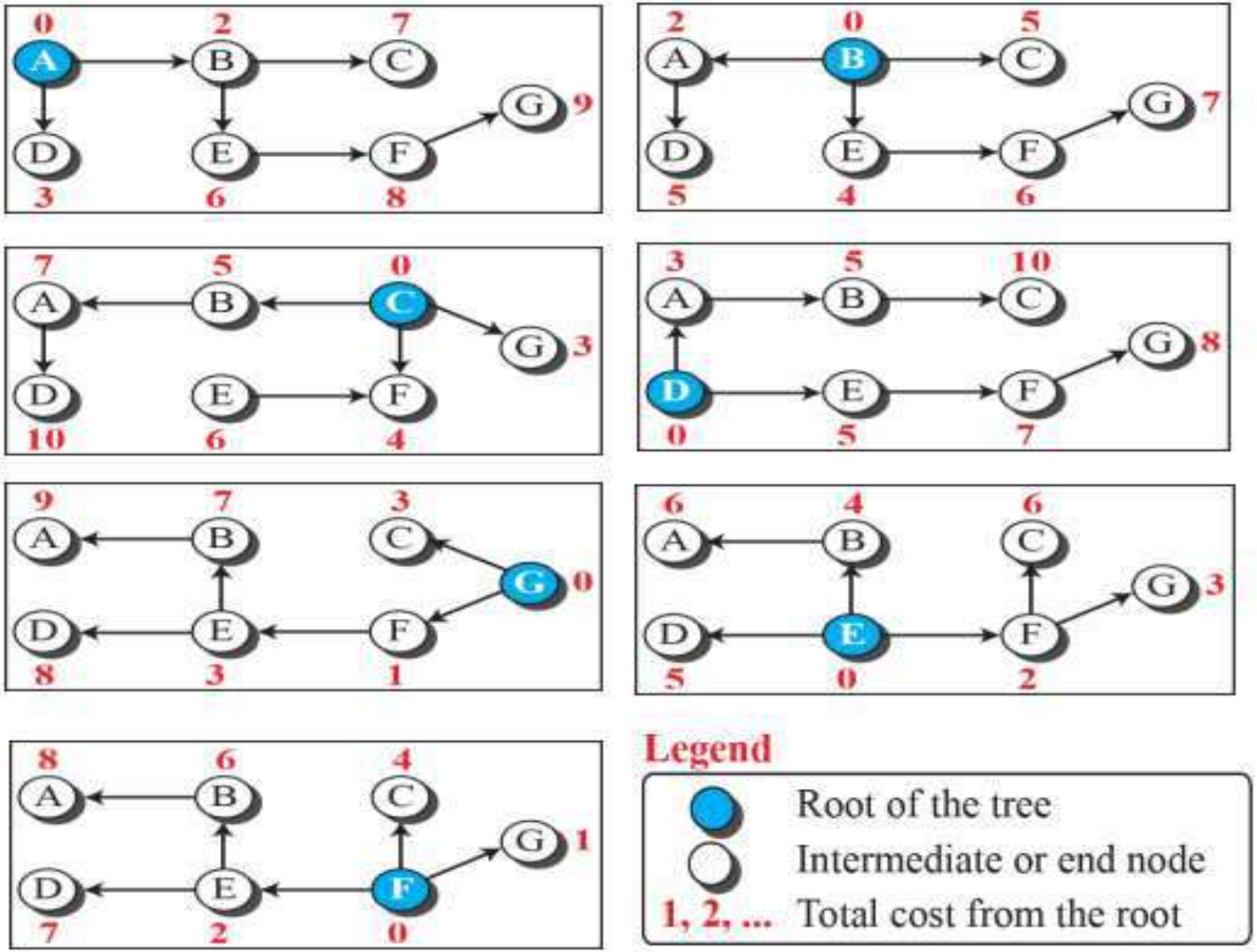


Figure: Least-cost trees for nodes in the internet of Figure.

- The least-cost trees for a weighted graph can have several properties if they are created using consistent criteria 1.

1. The least-cost route from X to Y in X's tree is the inverse of the least-cost route from Y to Xin Y's tree; the cost in both directions is the same. For example, in above Figure, the route from A to F in A's tree is (A→B→E→F),but the route from F to Ain F's tree is(F→E→B→A), which is the inverse of the first route. The cost is 8 in each case.
2. Instead of travelling from X to Z using X's tree, we can travel from X to Y using X's tree and continue from Y to Z using Y's tree. For example, in above Figure, we can go from A to G in A's tree using the route (A→B→E→F→G). We can also go from A to E in A's tree (A→B→E) and then continue in E's tree using the route (E→F→G). The combination of the two routes.

ROUTING ALGORITHMS

Several routing algorithms have been designed in the past. The differences between these methods are in the way they interpret the least cost and the way they create the least-cost tree for each node.

Distance-Vector Routing

- In distance-vector routing, the first thing each node creates is its own least-cost tree with the rudimentary information it has about its immediate neighbors.
- The incomplete trees are exchanged between immediate neighbors to make the trees more and more complete and to represent the whole internet.
- In distance-vector routing, a router continuously tells all of its neighbors what it knows about the whole internet (although the knowledge can be incomplete).

Bellman-Ford Equation

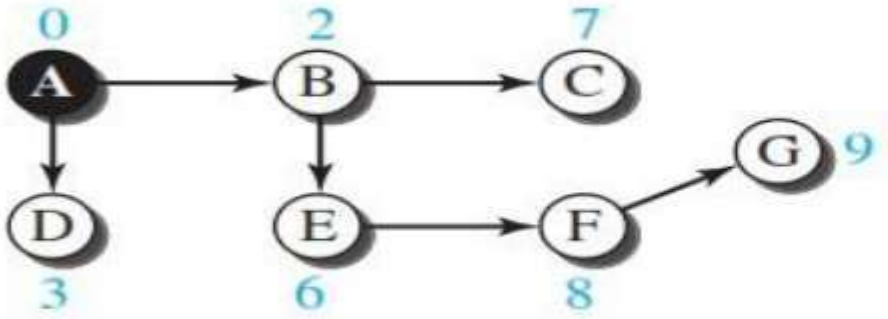
- The heart of distance-vector routing is the famous Bellman-Ford equation. This equation is used to find the least cost (shortest distance) between a source node and a destination node, through some intermediary nodes, when the costs between the source and the intermediary nodes and the least costs between the intermediary nodes and the destination are given.
- The following shows the general case in which D_{ij} is the shortest distance and c_{ij} is the cost between nodes i and j.

$$D_{xy} = \min\{(c_{xa} + D_{ay}), (c_{xb} + D_{by}), (c_{xc} + D_{cy}), \dots\}$$

- In distance-vector routing, normally we want to update an existing least cost with a least cost through an intermediary node.

Distance Vectors

- The concept of a distance vector is the rationale for the name distance-vector routing.
- A least-cost tree is a combination of least-cost paths from the root of the tree to all destinations. These paths are graphically glued together to form the tree.
- Distance-vector routing unglues these paths and creates a distance vector, a one dimensional array to represent the tree.



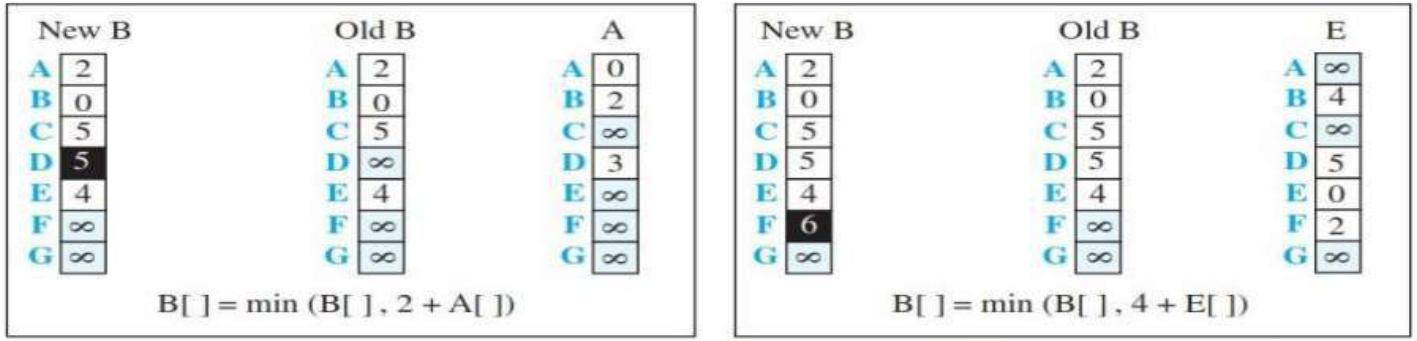
a. Tree for node A

	A
A	0
B	2
C	7
D	3
E	6
F	8
G	9

b. Distance vector for node A

- Shows the tree for node A in the internet in Figure 4.1 and the corresponding distance vector.
- Note that the name of the distance vector defines the root, the indexes define the destinations, and the value of each cell defines the least cost from the root to the destination.
- A distance vector does not give the path to the destinations as the least-cost tree does; it gives only the least costs to the destinations

- These rudimentary vectors cannot help the internet to effectively forward a packet. For example, node A thinks that it is not connected to node G because the corresponding cell shows the least cost of infinity.
- To improve these vectors, the nodes in the internet need to help each other by exchanging information.
- After each node has created its vector, it sends a copy of the vector to all its immediate neighbors. After a node receives a distance vector from a neighbor, it updates its distance vector using the Bellman-Ford equation (second case).
- However, we need to understand that we need to update, not only one least cost, but N of them in which N is the number of the nodes in the internet. If we are using a program, we can do this using a loop;



a. First event: B receives a copy of A's vector.

b. Second event: B receives a copy of E's vector.

Note:
X[ ]: the whole vector

Figure: Updating distance vectors

```
1   Distance_Vector_Routing ( )
2   {
3       // Initialize (create initial vectors for the node)
4       D[myself] = 0
5       for (y = 1 to N)
6       {
7           if (y is a neighbor)
8               D[y] = c[myself][y]
9           else
10              D[y] = ∞
11      }
12      send vector {D[1], D[2], ..., D[N]} to all neighbors
13      // Update (improve the vector with the vector received from a neighbor)
14      repeat (forever)
15      {
16          wait (for a vector D_w from a neighbor w or any change in the link)
17          for (y = 1 to N)
18          {
19              D[y] = min [D[y], (c[myself][w] + D_w[y])]      // Bellman-Ford equation
20          }
21          if (any change in the vector)
22              send vector {D[1], D[2], ..., D[N]} to all neighbors
23      }
24  } // End of Distance Vector
```

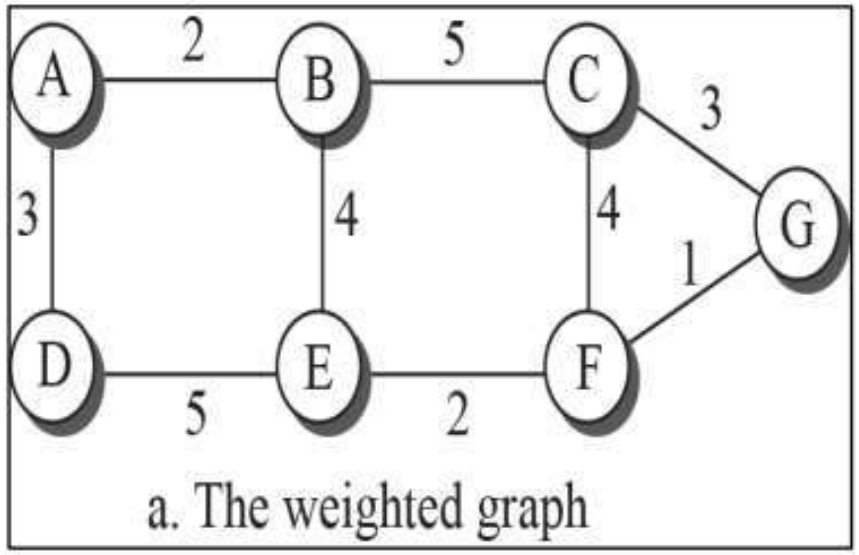
Figure: Distance-Vector Routing Algorithm for a Node

Link-State Routing

- This method uses the term link-state to define the characteristic of a link (an edge)that represents a network in the internet.
- In this algorithm the cost associated with an edge defines the state of the link. Links with lower costs are preferred to links with higher costs; if the cost of a link is infinity, it means that the link does not exist or has been broken.

Link-State Database (LSDB)

- To create a least-cost tree with this method, each node needs to have a complete map of the network, which means it needs to know the state of each link.
- The collection of states for all links is called the link-state database (LSDB). There is only one LSDB for the whole internet; each node needs to have a duplicate of it to be able to create the least-cost tree.
- The LSDB can be represented as a two- dimensional array (matrix) in which the value of each cell defines the cost of the corresponding link.



a. The weighted graph

	A	B	C	D	E	F	G
A	0	2	∞	3	∞	∞	∞
B	2	0	5	∞	4	∞	∞
C	∞	5	0	∞	∞	4	3
D	3	∞	∞	0	5	∞	∞
E	∞	4	∞	5	0	2	∞
F	∞	∞	4	∞	2	0	1
G	∞	∞	3	∞	∞	1	0

b. Link state database

Figure: Example of a link-state database

- **By flooding**: Each node send LS packet (LSP) to all its immediate neighbors (each interface) to collect two pieces of information for each neighboring node:

1. The identity of the node.
2. The cost of the link.

- When a node receives an LSP, it compares the LSP with the copy it may already have. The node checks the sequence number in both LSP to know which one is old and discards it, and keep the new one.
- Then the node sends a copy of it out of each interface except the one from which the packet arrived. This guarantees that flooding stops somewhere in the network (where a node has only one interface).
- Each node creates the comprehensive LSDB. This LSDB is the same for each node and shows the whole map of the internet. In other words, a node can make the whole map if it needs to, using this LSDB.

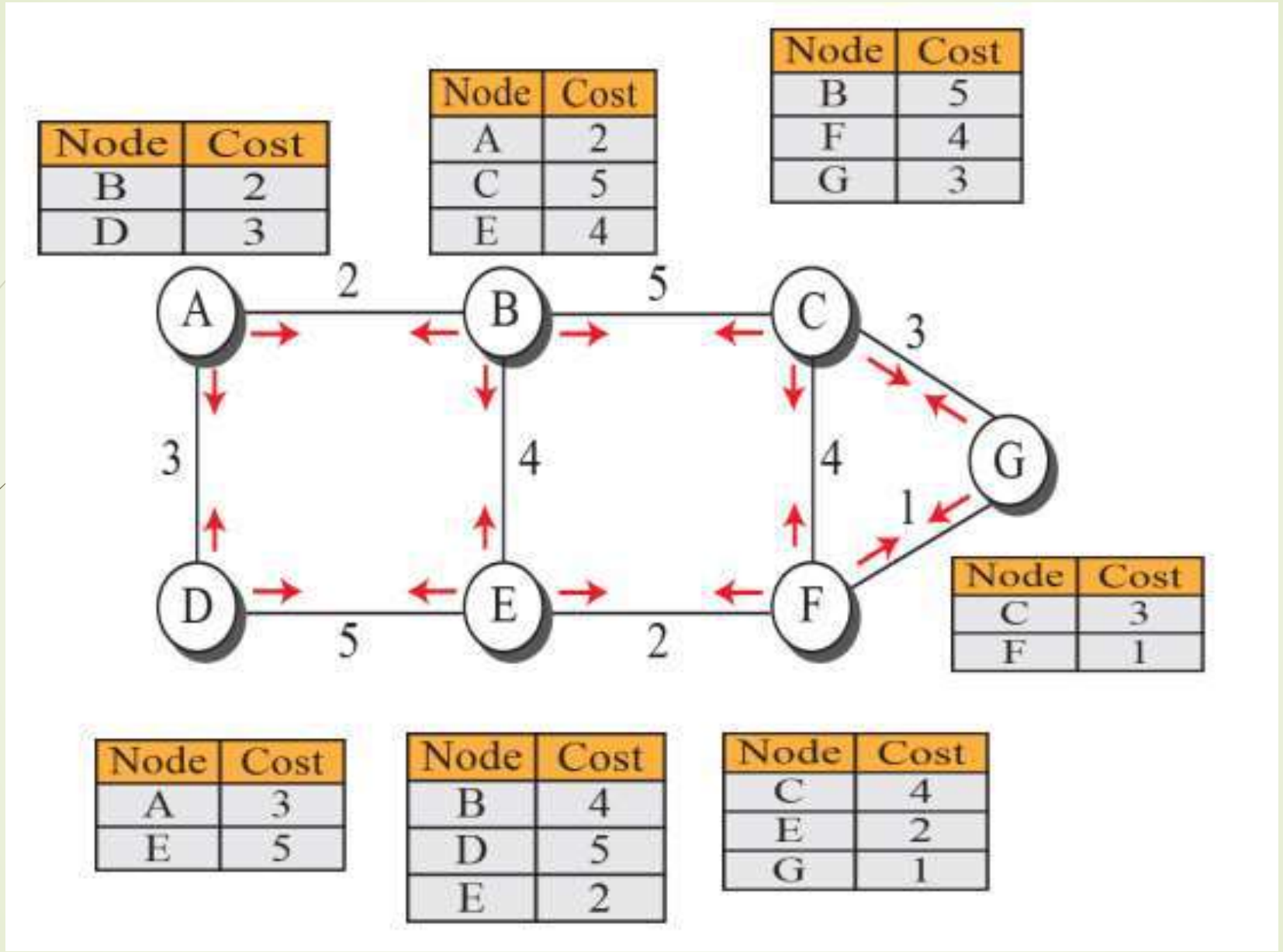


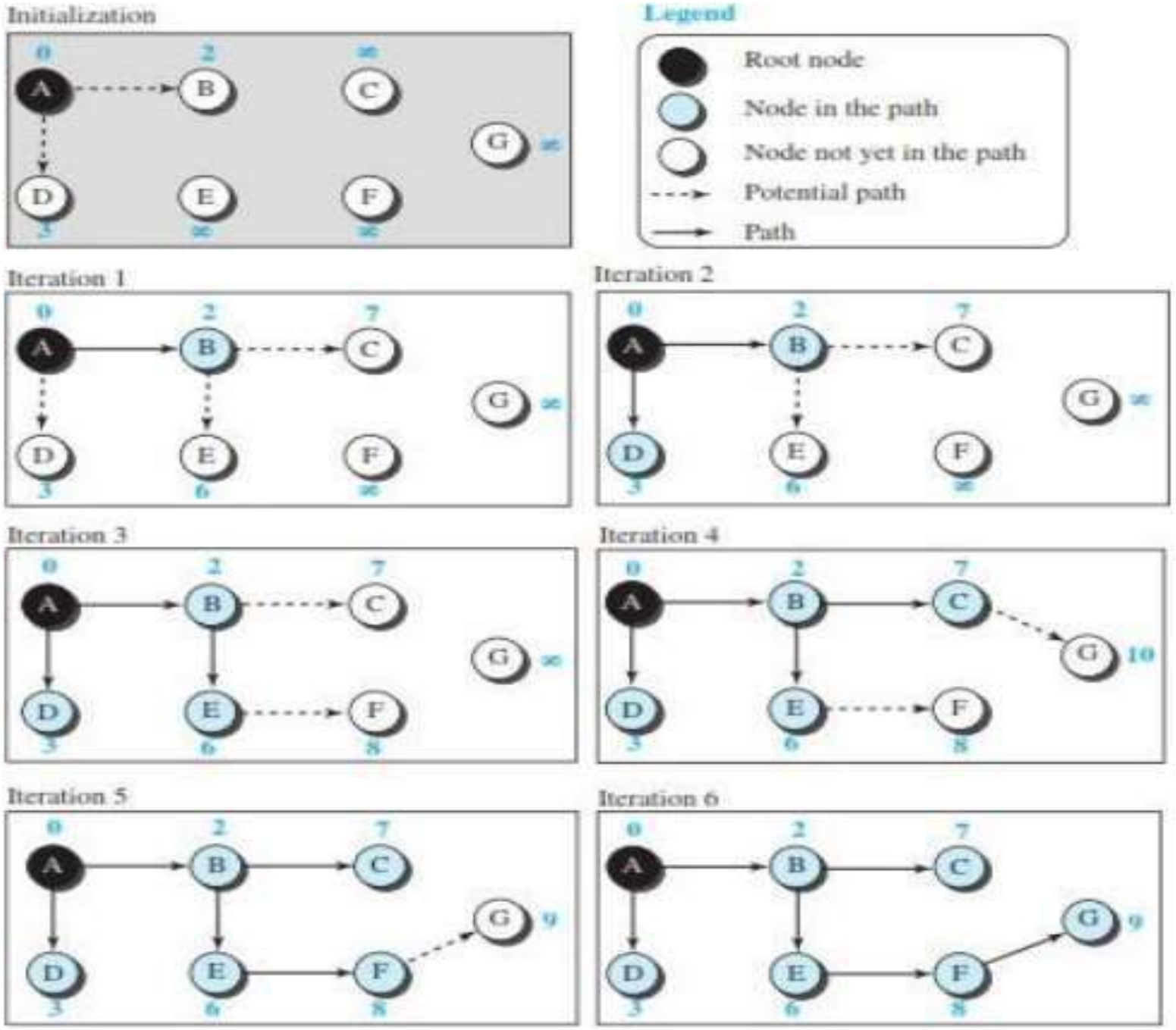
Figure : LSPs created and sent out by each node to build LSDB

- Formation of Least-Cost Trees-To create a least-cost tree for itself, using the shared LSDB, each node needs to run the famous Dijkstra Algorithm. This iterative algorithm uses the following steps:

1. The node chooses itself as the root of the tree, creating a tree with a single node, and sets the total cost of each node based on the information in the LSDB.
2. The node selects one node, among all nodes not in the tree, which is closest to the root, and adds this to the tree. After this node is added to the tree, the cost of all other nodes not in the tree needs to be updated because the paths may have been changed.
3. The node repeats step 2 until all nodes are added to the tree. We need to convince ourselves that the above three steps finally create the least-cost tree.

```
1  Dijkstra's Algorithm ( )
2  {
3      // Initialization
4      Tree = {root}                    // Tree is made only of the root

5      for (y = 1 to N)                 // N is the number of nodes
6      {
7          if (y is the root)
8              D[y] = 0                 // D[y] is shortest distance from root to node y
9          else if (y is a neighbor)
10             D[y] = c[root][y]        // c[x][y] is cost between nodes x and y in LSDB
11         else
12             D[y] = ∞
13     }
14     // Calculation
15     repeat
16     {
17         find a node w, with D[w] minimum among all nodes not in the Tree
18         Tree = Tree ∪ {w}            // Add w to tree
19         // Update distances for all neighbors of w
20         for (every node x, which is a neighbor of w and not in the Tree)
21         {
22             D[x] = min {D[x], (D[w] + c[w][x])}
23         }
24     } until (all nodes included in the Tree)
25 } // End of Dijkstra
```


Initialization
Legend
Root node
Node in the path
Node not yet in the path
Potential path
Path
Iteration 1
Iteration 2
Iteration 3
Iteration 4
Iteration 5
Iteration 6

Path-Vector Routing

- Both link-state and distance-vector routing are based on the least-cost goal. However, there are instances where this goal is not the priority.
- In other words, the least-cost goal, applied by LS or DV routing, does not allow a sender to apply specific policies to the route a packet may take.
- To respond to these demands, a third routing algorithm, called path-vector (PV) routing has been devised.
- Path-vector routing does not have the drawbacks of Ls or DV routing as described above because it is not based on least-cost routing. The best route is determined by the source using the policy it imposes on the route.
- In other words, the source can control the path. Although path-vector routing is not actually used in an internet, and is mostly designed to route a packet between ISPs.

Spanning Trees

- In path-vector routing, the path from a source to all destinations is also determined by the best spanning tree. The best spanning tree, however, is not the least-cost tree; it is the tree determined by the source when it imposes its own policy.
- . If there is more than one route to a destination, the source can choose the route that meets its policy best. A source may apply several policies at the same time.

- Each source has created its own spanning tree that meets its policy. The policy imposed by all sources is to use the minimum number of nodes to reach a destination.
- The spanning tree selected by A and E is such that the communication does not pass through D as a middle node. Similarly, the spanning tree selected by B is such that the communication does not past through C as a middle node.

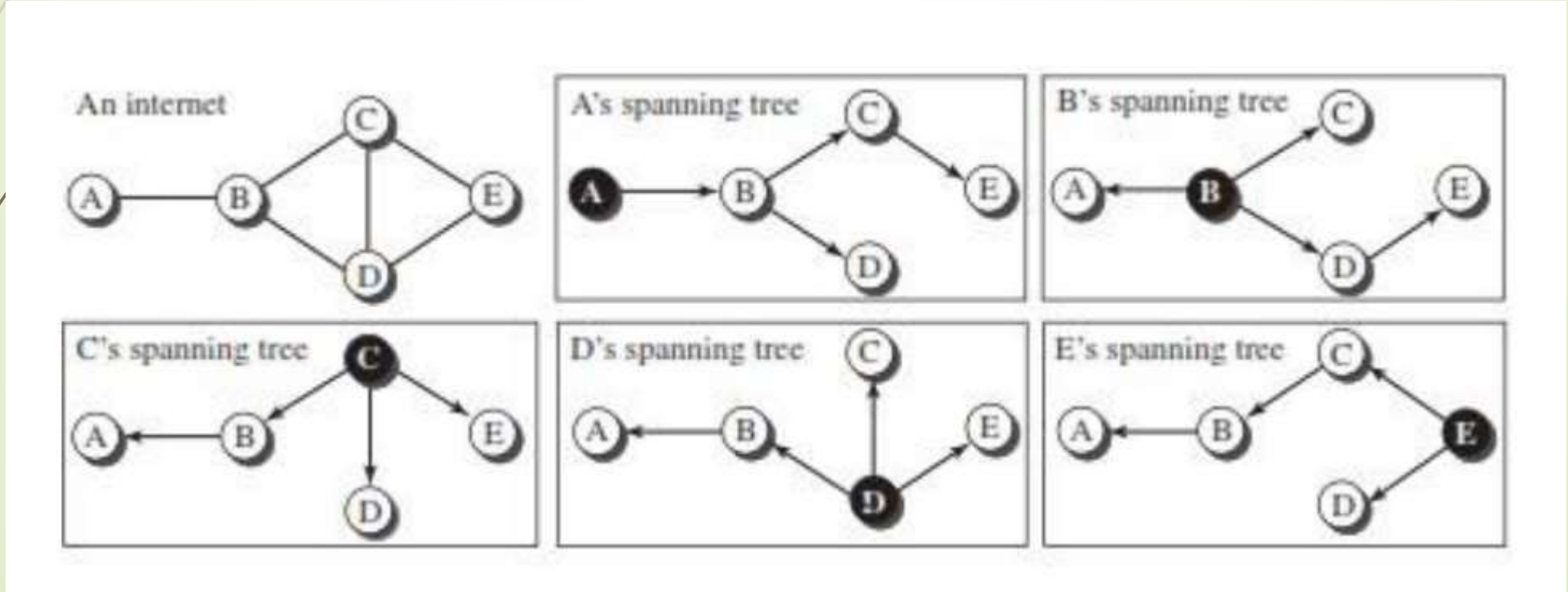


Figure: Spanning trees in path-vector routing

Creation of Spanning Trees

- The spanning trees are made, gradually and asynchronously, by each node. When anode is booted, it creates a path vector based on the information it can obtain about its immediate neighbor. Anode sends greeting messages to its immediate neighbors to collect these pieces of information.
- All of these tables are not created simultaneously; they are created when each node is booted. The figure also shows how these path vectors are sent to immediate neighbors after they have been created (arrows).
- Each node, after the creation of the initial path vector, sends it to all its immediate neighbors. Each node, when it receives a path vector from aneighbor, updates its path vector using an equation similar to the Bellman-Ford, but applying its own policy instead of looking for the least cost.

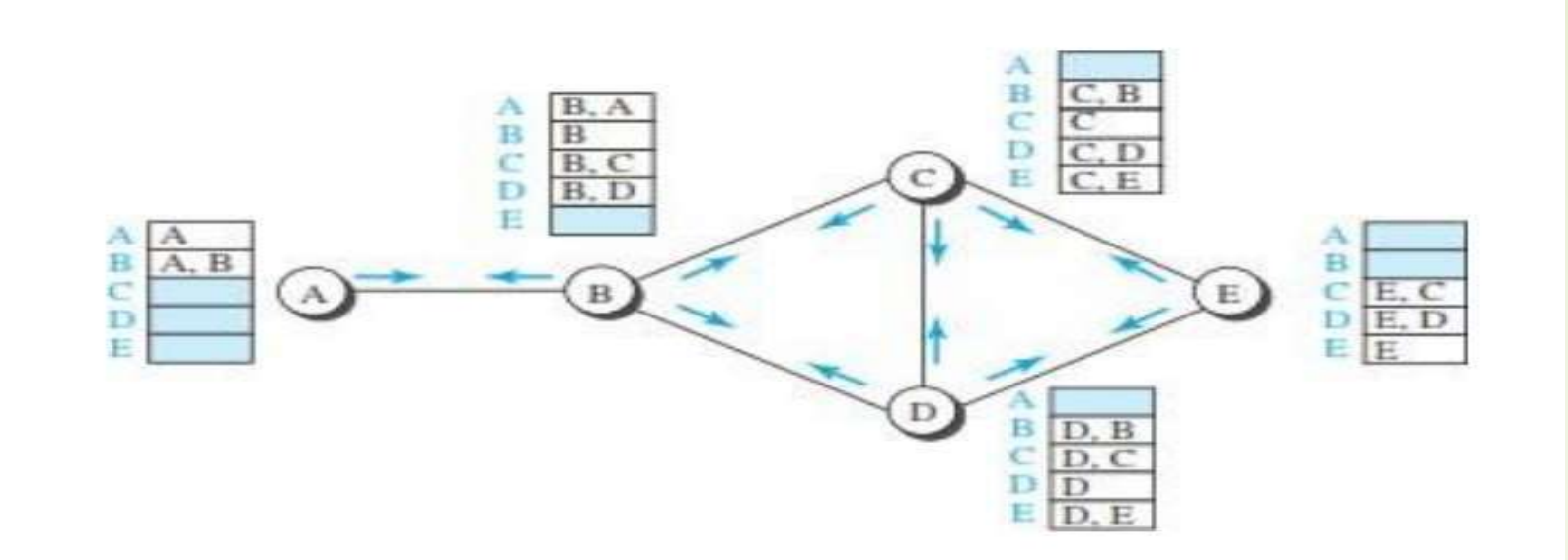
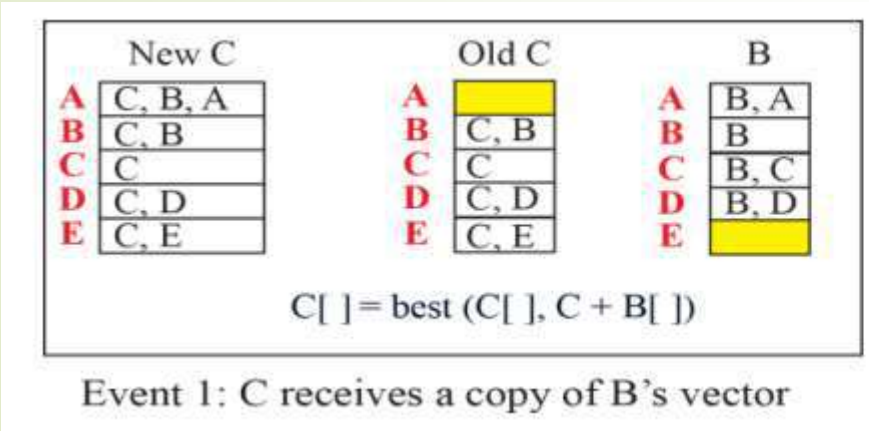
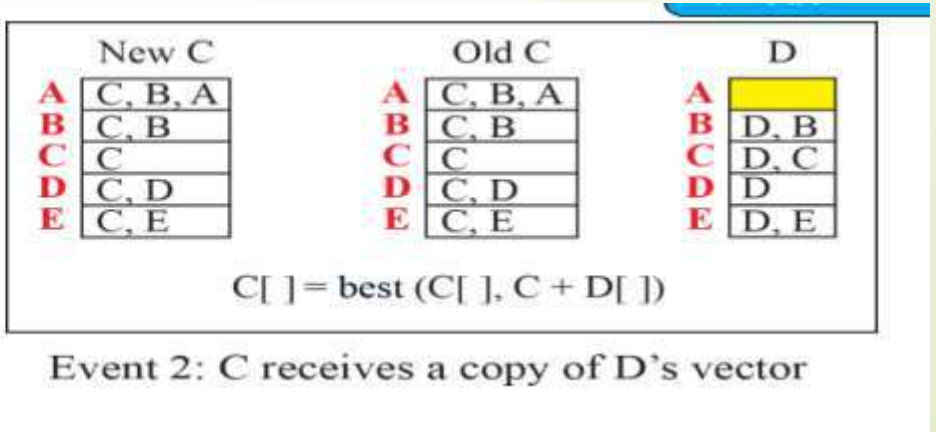


Figure: Path vectors made at booting time

In the first event, node C receives a copy of B's vector, which improves its vector: now it knows how to reach node A. In the second event, node C receives a copy of D's vector, which does not change its vector. The vector for node C after the first event is stabilized and serves as its forwarding table.



Event 1: C receives a copy of B's vector



Event 2: C receives a copy of D's vector

Figure: Updating path vectors

Path-Vector Algorithm

```
1  Path_Vector_Routing ( )
2  {
3     // Initialization
4     for (y = 1 to N)
5     {
6        if (y is myself)
7           Path[y] = myself
8        else if (y is a neighbor)
9           Path[y] = myself + neighbor node
10       else
11          Path[y] = empty
12    }
13    Send vector {Path[1], Path[2], ..., Path[y]} to all neighbors
14    // Update
15    repeat (forever)
16    {
17       wait (for a vector Path_w from a neighbor w)
18       for (y = 1 to N)
19       {
20          if (Path_w includes myself)
21             discard the path                    // Avoid any loop
22          else
23             Path[y] = best {Path[y], (myself + Path_w[y])}
24       }
25       If (there is a change in the vector)
26          Send vector {Path[1], Path[2], ..., Path[y]} to all neighbors
27    }
28 } // End of Path Vector
```

IPV6

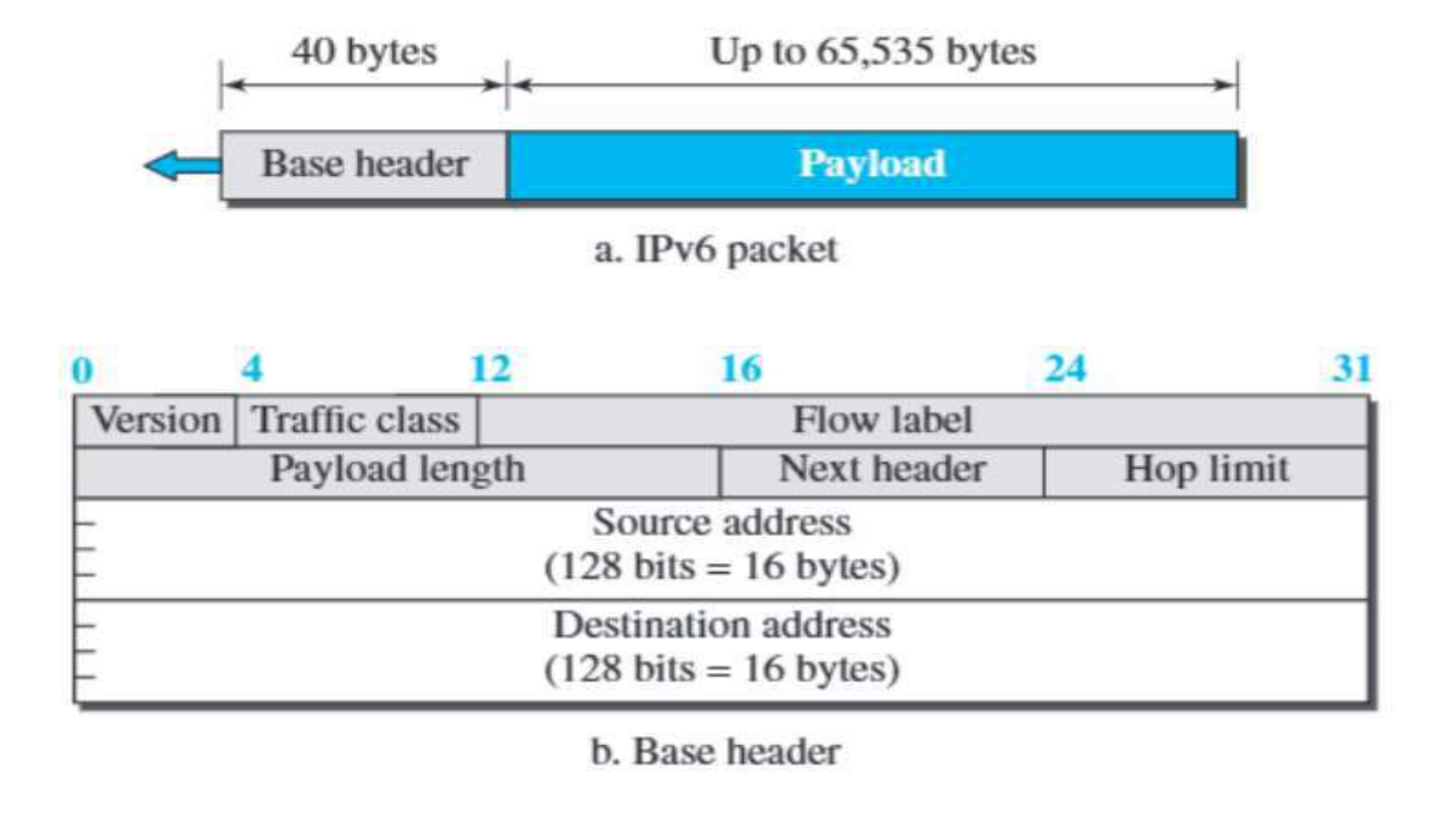
- The network layer protocol in the TCPIIP protocol suite is currently IPv4 (Internet working Protocol, version 4).
- IPv4 provides the host-to-host communication between systems in the Internet. Although IPv4 is well designed, data communication has evolved since the inception of IPv4 in the 1970s.
- IPv4 has some deficiencies (listed below) that make it unsuitable for the fast-growing Internet. o Despite all short-term solutions, such as subnetting, classless addressing, and NAT, address depletion is still a long-term problem in the Internet.
- The Internet must accommodate real-time audio and video transmission. This type of transmission requires minimum delay strategies and reservation of resources not provided in the IPv4 design.
- The Internet must accommodate encryption and authentication of data for some applications. No encryption or authentication is provided by IPv4.
- In IPv6, the Internet protocol was extensively modified to accommodate the unforeseen growth of the Internet. The format and the length of the IP address were changed along with the packet format. Related protocols, such as ICMP, were also modified. Other protocols in the network layer, such as ARP, RARP, and IGMP, were either deleted or included in the ICMPv6 protocol.

Advantages of IPv6 over IPv4

- **Larger address space**. An IPv6 address is 128 bits long. Compared with the 32-bit address of IPv4, this is a huge (296) increase in the address space.
- **Better header format**. IPv6 uses a new header format in which options are separated from the base header and inserted, when needed, between the base header and the upper-layer data. This simplifies and speeds up the routing process because most of the options do not need to be checked by routers.
- **New options**. IPv6 has new options to allow for additional functionalities.
- **Allowance for extension**. IPv6 is designed to allow the extension of the protocol if required by new technologies or applications.
- **Support for resource allocation**. In IPv6, the type-of-service field has been removed, but a mechanism (called low label) has been added to enable the source to request special handling of the packet. This mechanism can be used to support traffic such as real-time audio and video.
- **Support for more security**. The encryption and authentication options in IPv6 provide confidentiality and integrity of the packet.

Packet Format

Each packet is composed of a mandatory base header followed by the payload. The payload consists of two parts: optional extension headers and data from an upper layer. The base header occupies 40 bytes, whereas the extension headers and data from the upper layer contain up to 65,535 bytes of information.



- **Version.** This 4-bit field defines the version number of the IP. For IPv6, the value is 6.
- **Priority**. The 4-bit priority field defines the priority of the packet with respect to traffic congestion.
- **Flow label**. The flow label is a 3-byte (24-bit) field that is designed to provide special handling for a particular flow of data.
- **Payload length**. The 2-byte payload length field defines the length of the IP datagram excluding the base header.
- **Next header**. The next header is an 8-bit field defining the header that follows the base header in the datagram. The next header is either one of the optional extension headers used by IP or the header of an encapsulated packet such as UDP or TCP. Each extension header also contains this field.
- **Hop limit**. This 8-bit hop limit field serves the same purpose as the TIL field in IPv4.
- **Source address**. The source address field is a 16-byte (128-bit) Internet address that identifies the original source of the datagram.
- **Destination address**. The destination address field is a 16-byte (128-bit) Internet address that usually identifies the final destination of the datagram. However, if source routing is used, this field contains the address of the next router.

Comparison between IPv4 and IPv6 packet headers

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

Comparison between IPv4 options and IPv6 extension headers

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

Thank You

APS College of Engineering,

(Affiliated to Visvesvaraya Technological University and Approved by AICTE, NAAC Accredited)

Somanahalli, Kanakapura Main Road, Bengaluru-560116

Department of Electronics and Communication Engineering

Subject Name: Computer Communication Networks

Subject Name: 21EC73

Semester : 5th

Academic Year: ODD 2023-2024

Module-4

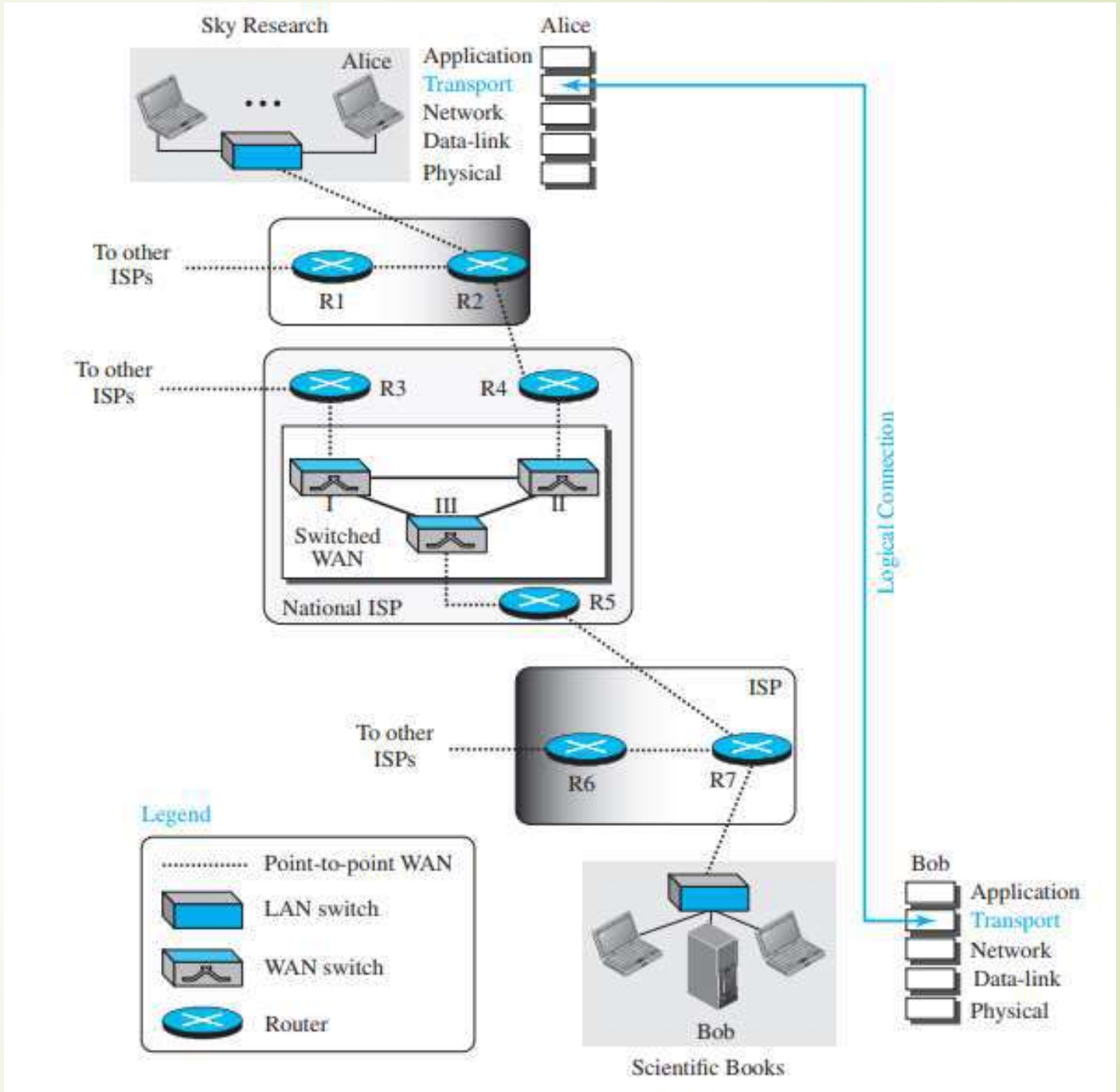
Faculty Name: Dr. Naik D C
Assistant Professor,
Dept., of ECE.

Module-4

INTRODUCTION, Transport-Layer Services , Connectionless and Connection-Oriented Protocols, TRANSPORT-LAYER PROTOCOLS, Simple Protocol, Stop-and-Wait Protocol, Go-Back-N Protocol (GBN), Selective-Repeat Protocol, Bidirectional Protocols: Piggybacking.

USER DATAGRAM PROTOCOL, User Datagram, UDP Services, UDP Applications, TRANSMISSION CONTROL PROTOCOL, TCP Services, TCP Features, Segment, A TCP Connection, State Transition Diagram, Windows in TCP, Flow Control, Error Control, TCP Congestion Control.

- The transport layer is responsible for providing services to the application layer; it receives services from the network layer.
- The transport layer is located between the application layer and the network layer. It provides a process-to-process communication between two application layers, one at the local host and the other at the remote host.
- Communication is provided using a logical connection, which means that the two application layers, which can be located in different parts of the globe, assume that there is an imaginary direct connection through which they can send and receive messages.



Transport-Layer Services

Process-to-Process Communication

- The first duty of a transport-layer protocol is to provide process-to-process communication. A process is an application-layer entity that uses the services of the transport layer.
- The network layer is responsible for communication at the computer level (host-to-host communication).
- A network-layer protocol can deliver the message only to the destination computer. However, this is an incomplete delivery. The message still needs to be handed to the correct process.
- This is where a transport-layer protocol takes over. A transport-layer protocol is responsible for delivery of the message to the appropriate process.

Addressing: Port Numbers

- Although there are a few ways to achieve process-to-process communication, the most common is through the client-server paradigm.
- A process on the local host, called a client, needs services from a process usually on the remote host, called a server.
- A remote computer can run several server programs at the same time, just as several local computers can run one or more client programs at the same time.
- For communication, we must define the local host, local process, remote host, and remote process. The local host and the remote host are defined using IP addresses.
- To define the processes, we need second identifiers, called port numbers. In the TCP/IP protocol suite, the port numbers are integers between 0 and 65,535 (16 bits)

- The server process must also define itself with a port number. This port number, however, cannot be chosen randomly. If the computer at the server site runs a server process and assigns a random number as the port number, the process at the client site that wants to access that server and use its services will not know the port number.
- Of course, one solution would be to send a special packet and request the port number of a specific server, but this creates more overhead. TCP/IP has decided to use universal port numbers for servers; these are called well-known port numbers.

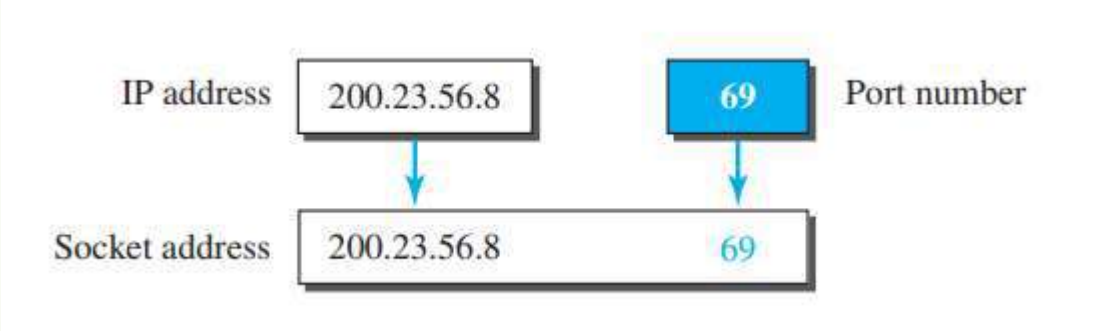
ICANN Ranges

- ICANN has divided the port numbers into three ranges: well-known, registered, and dynamic (or private), as shown in Figure

- **Well-known ports**. The ports ranging from 0 to 1023 are assigned and controlled by ICANN. These are the well-known ports.
- **Registered ports.** The ports ranging from 1024 to 49,151 are not assigned or controlled by ICANN. They can only be registered with ICANN to prevent duplication.
- **Dynamic ports**. The ports ranging from 49,152 to 65,535 are neither controlled nor registered. They can be used as temporary or private port numbers.

Socket Addresses

- A transport-layer protocol in the TCP suite needs both the IP address and the port number, at each end, to make a connection.
- The combination of an IP address and a port number is called a socket address.
- The client socket address defines the client process uniquely just as the server socket address defines the server process uniquely.



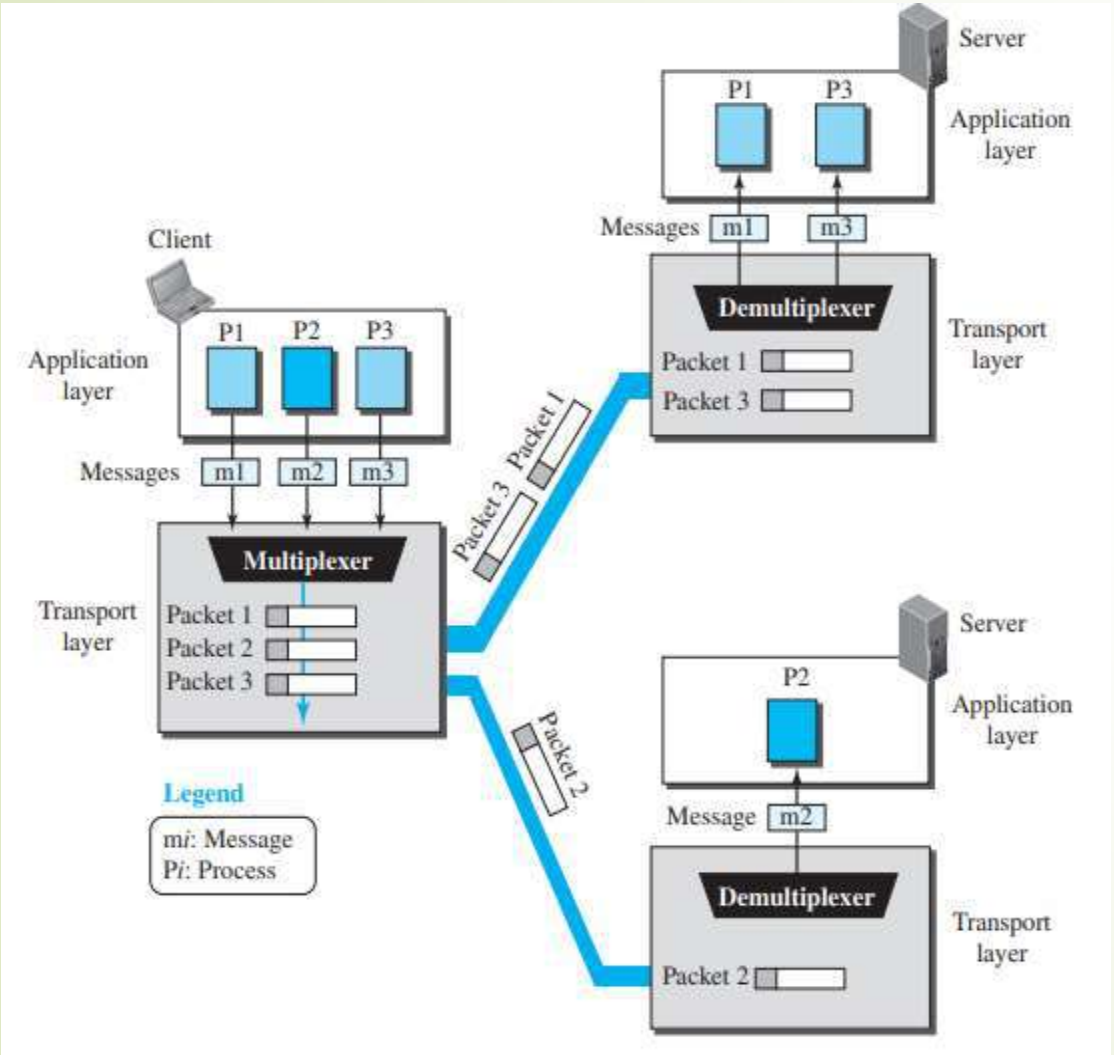
- To use the services of the transport layer in the Internet, we need a pair of socket addresses: the client socket address and the server socket address.
- These four pieces of information are part of the network-layer packet header and the transport-layer packet header. The first header contains the IP addresses; the second header contains the port numbers.

Encapsulation and Decapsulation

- To send a message from one process to another, the transport-layer protocol encapsulates and decapsulates messages.
- When a process has a message to send, it passes the message to the transport layer along with a pair of socket addresses and some other pieces of information, which depend on the transport-layer protocol.
- The transport layer receives the data and adds the transport-layer header. The packets at the transport layer in the Internet are called user datagrams, segments, or packets, depending on what transport-layer protocol we use.
- Decapsulation happens at the receiver site. When the message arrives at the destination transport layer, the header is dropped and the transport layer delivers the message to the process running at the application layer.
- The sender socket address is passed to the process in case it needs to respond to the message received.

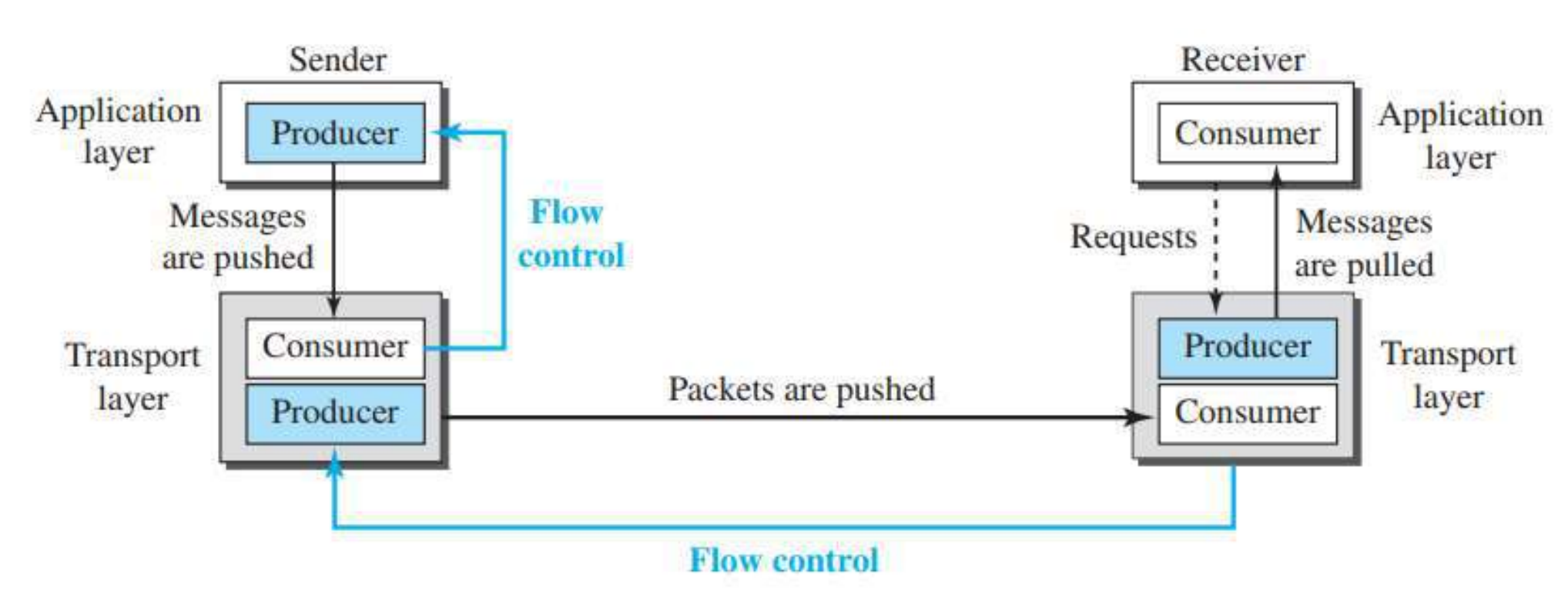
Multiplexing and Demultiplexing

- Whenever an entity accepts items from more than one source, this is referred to as multiplexing (many to one); whenever an entity delivers items to more than one source, this is referred to as demultiplexing (one to many).



- Three client processes are running at the client site, P1, P2, and P3. The processes P1 and P3 need to send requests to the corresponding server process running in a server.
- The client process P2 needs to send a request to the corresponding server process running at another server.
- The transport layer at the client site accepts three messages from the three processes and creates three packets. It acts as a multiplexer.
- The packets 1 and 3 use the same logical channel to reach the transport layer of the first server. When they arrive at the server, the transport layer does the job of a demultiplexer and distributes the messages to two different processes.
- The transport layer at the second server receives packet 2 and delivers it to the corresponding process.

Figure: Flow control at the transport layer



Buffers

- Although flow control can be implemented in several ways, one of the solutions is normally to use two buffers: one at the sending transport layer and the other at the receiving transport layer.
- A buffer is a set of memory locations that can hold packets at the sender and receiver.
- The flow control communication can occur by sending signals from the consumer to the producer. When the buffer of the sending transport layer is full, it informs the application layer to stop passing chunks of messages; when there are some vacancies, it informs the application layer that it can pass message chunks again.
- When the buffer of the receiving transport layer is full, it informs the sending transport layer to stop sending packets. When there are some vacancies, it informs the sending transport layer that it can send packets again.

Error Control

- In the Internet, since the underlying network layer (IP) is unreliable, we need to make the transport layer reliable if the application requires reliability.
- Reliability can be achieved to add error control services to the transport layer.
- Error control at the transport layer is responsible for

1. Detecting and discarding corrupted packets.
2. Keeping track of lost and discarded packets and resending them.
3. Recognizing duplicate packets and discarding them.
4. Buffering out-of-order packets until the missing packets arrive.

- Error control, unlike flow control, involves only the sending and receiving transport layers.
- We are assuming that the message chunks exchanged between the application and transport layers are error free.

Sequence Numbers

- Error control requires that the sending transport layer knows which packet is to be resent and the receiving transport layer knows which packet is a duplicate, or which packet has arrived out of order.
- This can be done if the packets are numbered. We can add a field to the transport-layer packet to hold the sequence number of the packet.
- When a packet is corrupted or lost, the receiving transport layer can somehow inform the sending transport layer to resend that packet using the sequence number.
- The receiving transport layer can also detect duplicate packets if two received packets have the same sequence number.
- The out-of-order packets can be recognized by observing gaps in the sequence numbers.
- Packets are numbered sequentially. However, because we need to include the sequence number of each packet in the header, we need to set a limit. If the header of the packet allows m bits for the sequence number, the sequence numbers range from 0 to $2^m - 1$.

Sliding Window

- Since the sequence numbers use modulo 2m, a circle can represent the sequence numbers from 0 to $2^m - 1$.
- The buffer is represented as a set of slices, called the sliding window, that occupies part of the circle at any time.
- The buffer is represented as a set of slices, called the sliding window, that occupies part of the circle at any time.
- At the sender site, when a packet is sent, the corresponding slice is marked. When all the slices are marked, it means that the buffer is full and no further messages can be accepted from the application layer.
- When an acknowledgment arrives, the corresponding slice is unmarked. If some consecutive slices from the beginning of the window are unmarked, the window slides over the range of the corresponding sequence numbers to allow more free slices at the end of the window.
- The sequence numbers are in modulo 16 (m = 4) and the size of the window is 7. Note that the sliding window is just an abstraction: the actual situation uses computer variables to hold the sequence numbers of the next packet to be sent and the last packet sent.

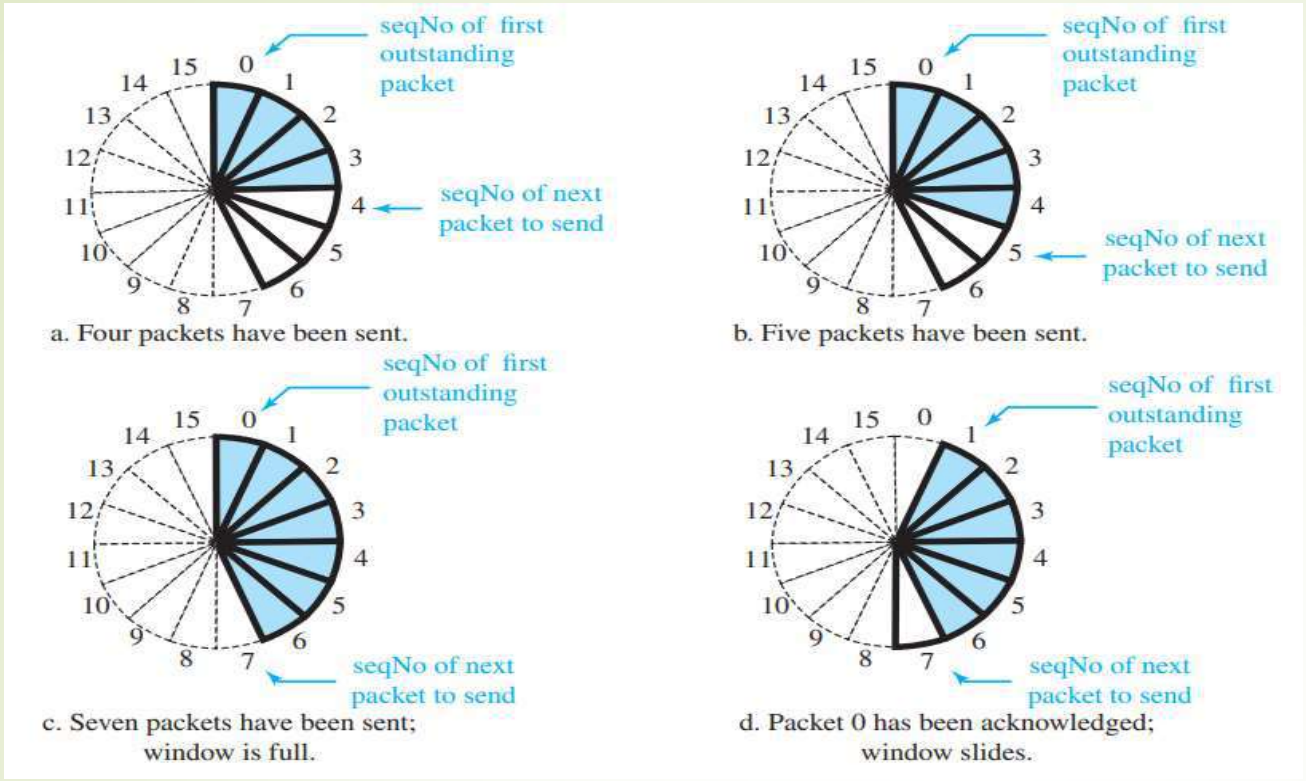


Figure: Sliding window in circular format

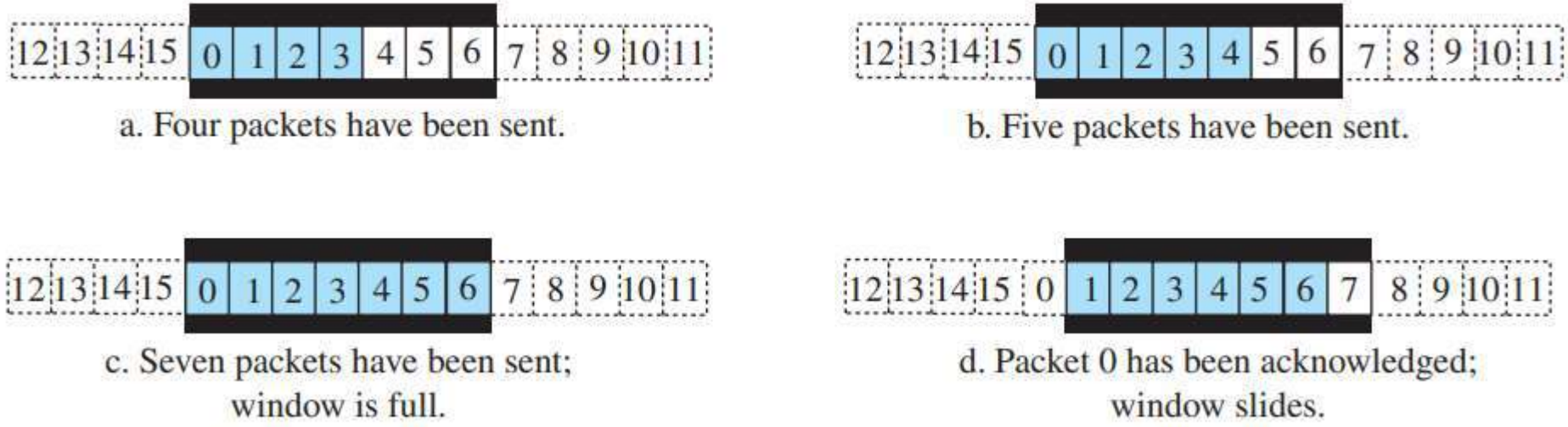


Figure: Sliding window in linear format

Congestion Control

- 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 that control the congestion and keep the load below the capacity.
- Congestion in a network or internetwork occurs because routers and switches have queues—buffers that hold the packets before and after processing.
- If a router cannot process the packets at the same rate at which they arrive, the queues become overloaded and congestion occurs.
- Congestion at the transport layer is actually the result of congestion at the network layer, which manifests itself at the transport layer.

Connectionless and Connection-Oriented Protocols

- A transport-layer protocol, like a network-layer protocol, can provide two types of services: connectionless and connection-oriented.
- The nature of these services at the transport layer, however, is different from the ones at the network layer.
- At the network layer, a connectionless service may mean different paths for different datagrams belonging to the same message.
- At the transport layer, we are not concerned about the physical paths of packets (we assume a logical connection between two transport layers).
- Connectionless service at the transport layer means independency between packets; connection-oriented means dependency.

Connectionless Service

- In a connectionless service, the source process (application program) needs to divide its message into chunks of data of the size acceptable by the transport layer and deliver them to the transport layer one by one.
- The transport layer treats each chunk as a single unit without any relation between the chunks.
- When a chunk arrives from the application layer, the transport layer encapsulates it in a packet and sends it.
- The chunks are handed over to the connectionless transport protocol in order. However, since there is no dependency between the packets at the transport layer, the packets may arrive out of order at the destination and will be delivered out of order to the server process.
- We can say that no flow control, error control, or congestion control can be effectively implemented in a connectionless service.

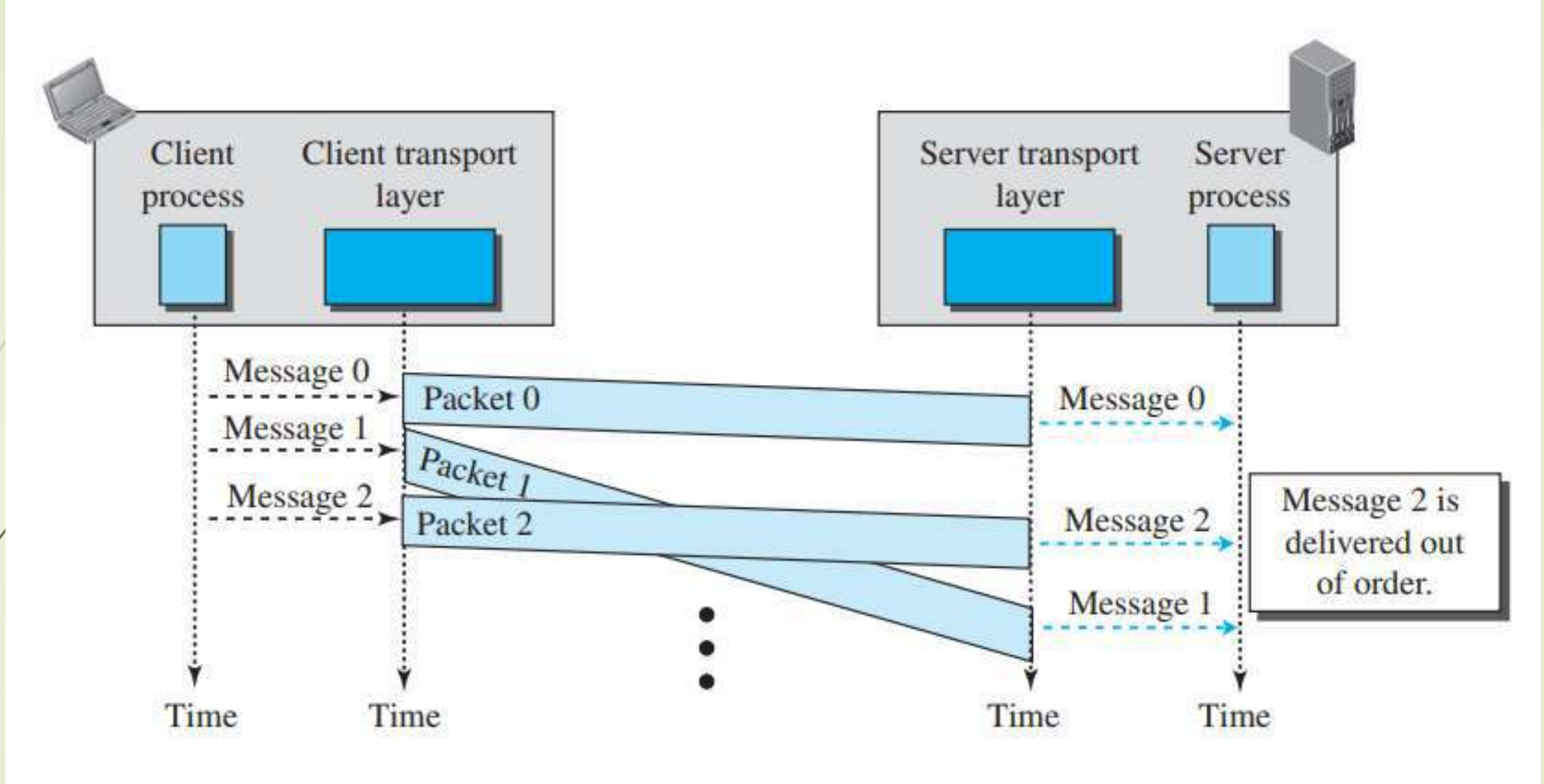


Figure: Connectionless service

Connection-Oriented Service

In a connection-oriented service, the client and the server first need to establish a logical connection between themselves. The data exchange can only happen after the connection establishment. After data exchange, the connection needs to be torn down.

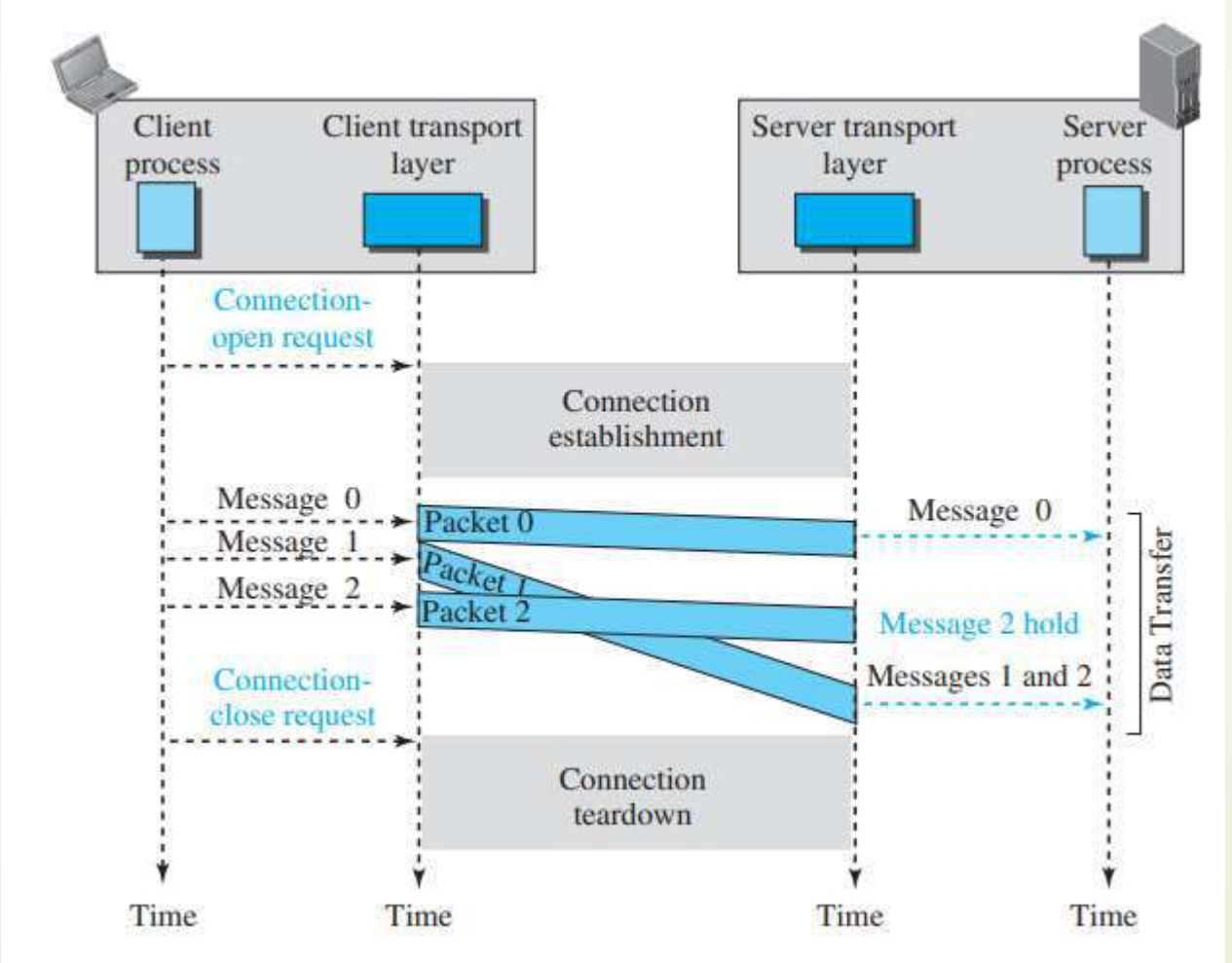
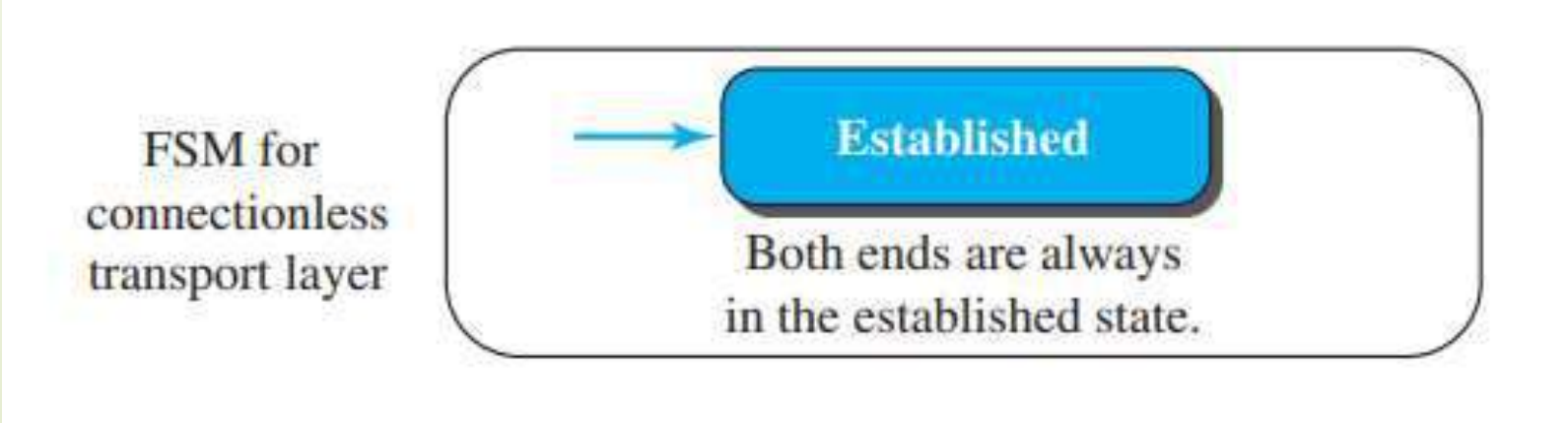


Figure: Connection-oriented service

Finite State Machine

- Using this tool, each transport layer (sender or receiver) is taught as a machine with a finite number of states.
- The machine is always in one of the states until an event occurs. Each event is associated with two reactions: defining the list (possibly empty) of actions to be performed and determining the next state (which can be the same as the current state).
- One of the states must be defined as the initial state, the state in which the machine starts when it turns on.



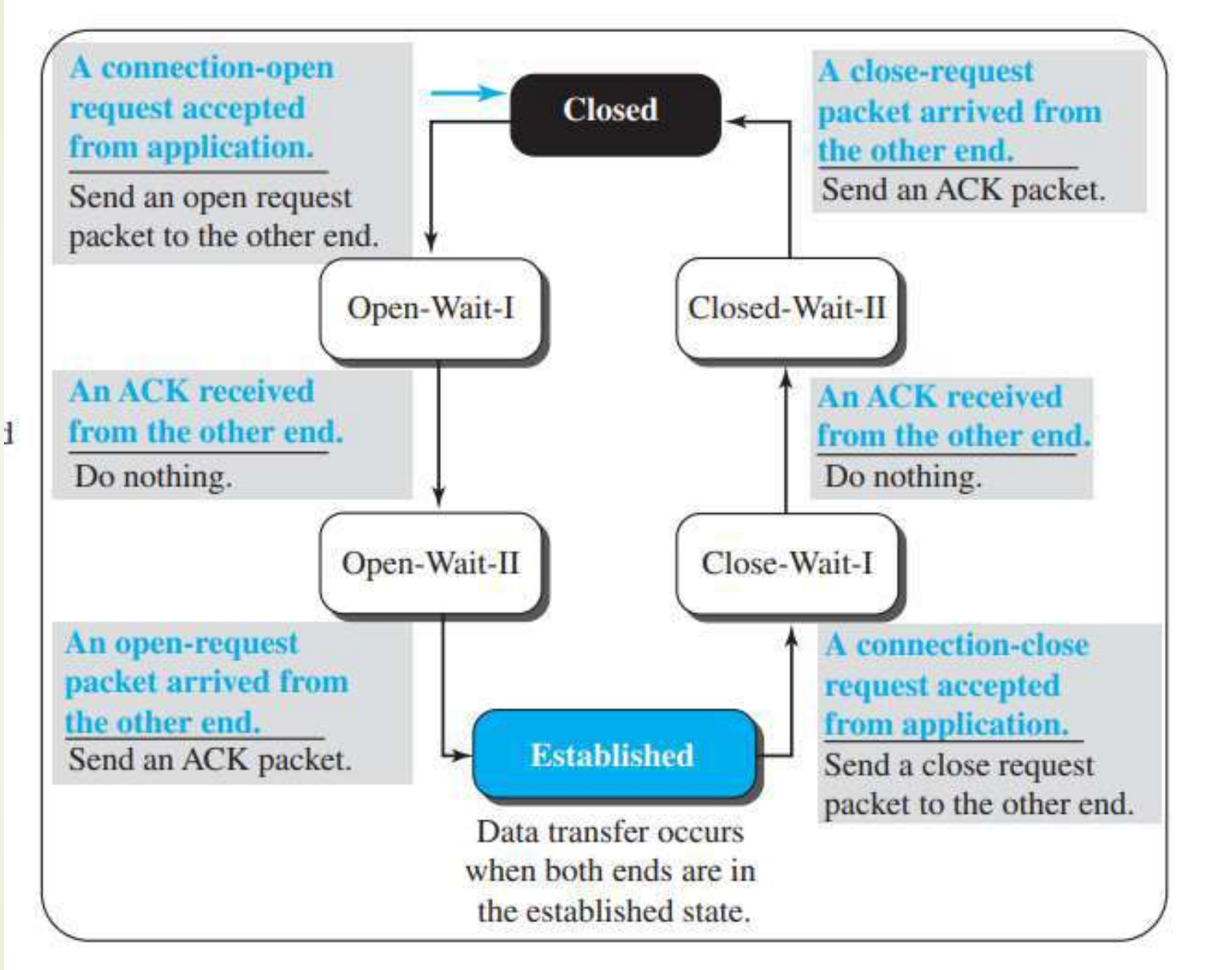


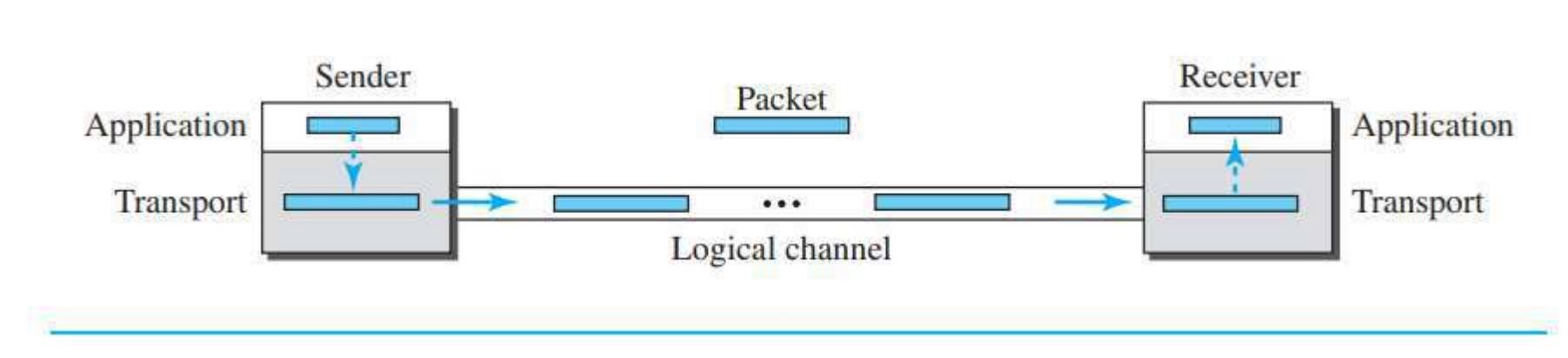
Figure: FSM for connection-oriented transport layer

TRANSPORT-LAYER PROTOCOLS

- Simple Protocol
- Stop-and-Wait Protocol
- Go-Back-N Protocol (GBN)
- Selective-Repeat Protocol
- Bidirectional Protocols: Piggybacking

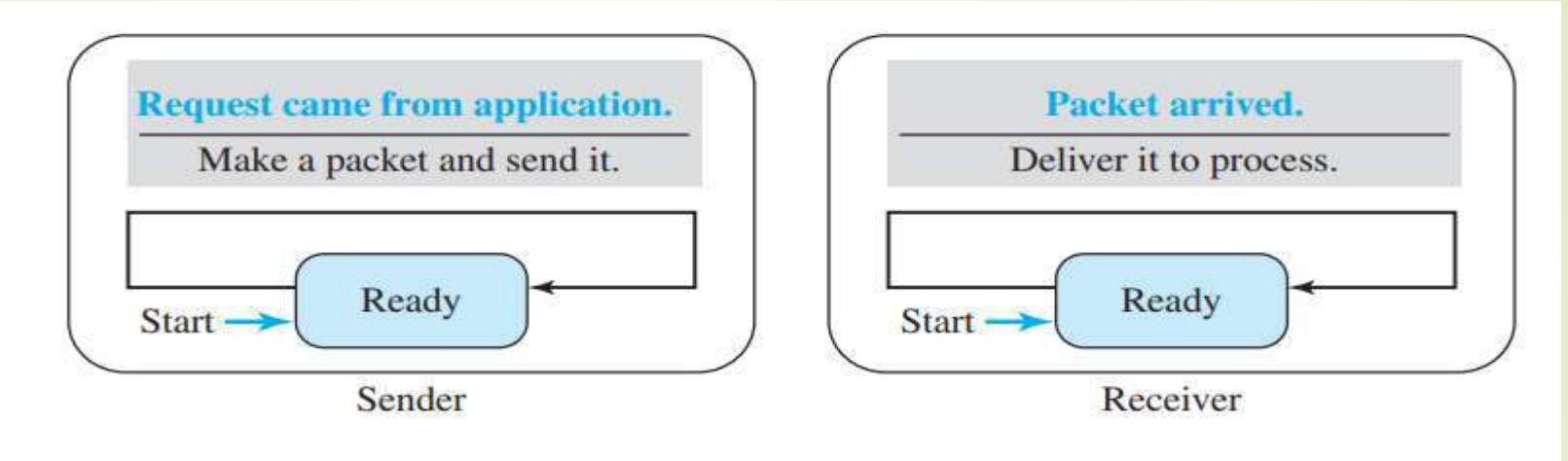
Simple Protocol

- We assume that the receiver can immediately handle any packet it receives. In other words, the receiver can never be overwhelmed with incoming packets.



- The transport layer at the sender gets a message from its application layer, makes a packet out of it, and sends the packet.
- The transport layer at the receiver, receives a packet from its network layer, extracts the message from the packet, and delivers the message to its application layer.
- The transport layers of the sender and receiver provide transmission services for their application layers.

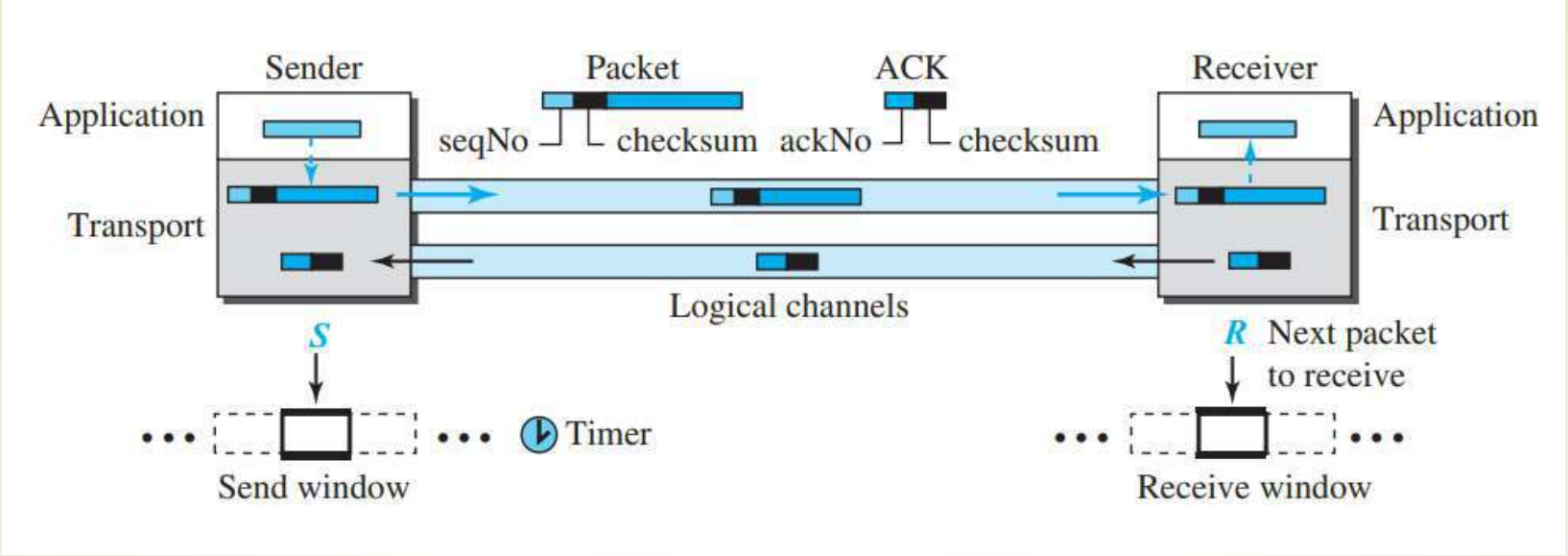
FSMs



- We can show these requirements using two FSMs. Each FSM has only one state, the ready state.
- The sending machine remains in the ready state until a request comes from the process in the application layer.
- When this event occurs, the sending machine encapsulates the message in a packet and sends it to the receiving machine.
- The receiving machine remains in the ready state until a packet arrives from the sending machine.
- When this event occurs, the receiving machine decapsulates the message out of the packet and delivers it to the process at the application layer.

Stop-and-Wait Protocol

- Both the sender and the receiver use a sliding window of size 1.
- The sender sends one packet at a time and waits for an acknowledgment before sending the next one.
- To detect corrupted packets, we need to add a checksum to each data packet. When a packet arrives at the receiver site, it is checked. If its checksum is incorrect, the packet is corrupted and silently discarded.
- Every time the sender sends a packet, it starts a timer. If an acknowledgment arrives before the timer expires, the timer is stopped and the sender sends the next packet (if it has one to send).
- If the timer expires, the sender resends the previous packet, assuming that the packet was either lost or corrupted.
- This means that the sender needs to keep a copy of the packet until its acknowledgment arrives.



Stop-and-Wait protocol

Sequence Numbers

To show this, assume that the sender has sent the packet with sequence number x. Three things can happen.

1. The packet arrives safe and sound at the receiver site; the receiver sends an acknowledgment. The acknowledgment arrives at the sender site, causing the sender to send the next packet numbered $x + 1$.
2. The packet is corrupted or never arrives at the receiver site; the sender resends the packet (numbered x) after the time-out. The receiver returns an acknowledgment.
3. The packet arrives safe and sound at the receiver site; the receiver sends an acknowledgment, but the acknowledgment is corrupted or lost. The sender resends the packet (numbered x) after the time-out. Note that the packet here is a duplicate. The receiver can recognize this fact because it expects packet $x + 1$ but packet x was received.

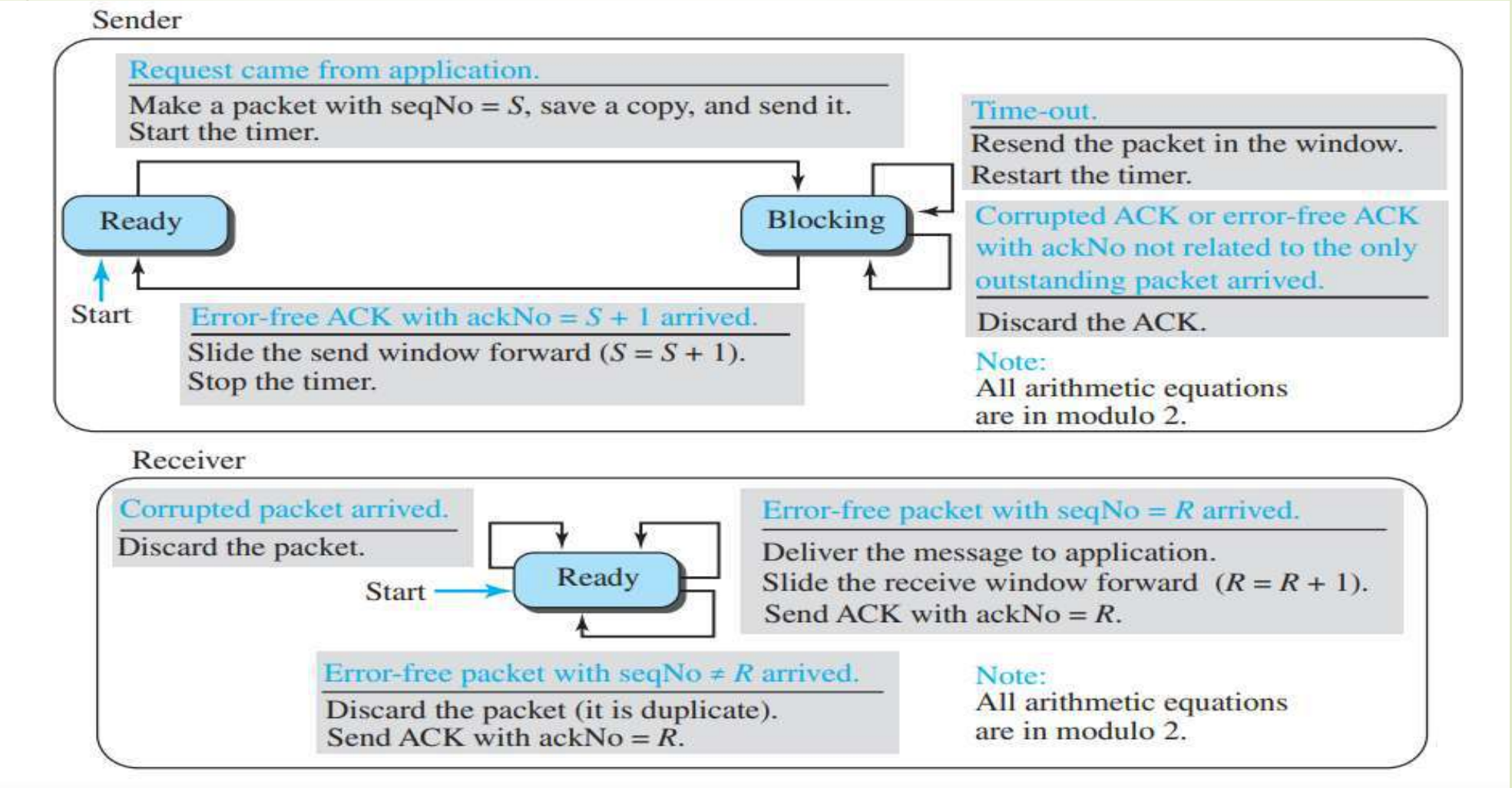
Acknowledgment Numbers

- Since the sequence numbers must be suitable for both data packets and acknowledgments, we use this convention: The acknowledgment numbers always announce the sequence number of the next packet expected by the receiver.
- For example, if packet 0 has arrived safe and sound, the receiver sends an ACK with acknowledgment 1 (meaning packet 1 is expected next). If packet 1 has arrived safe and sound, the receiver sends an ACK with acknowledgment 0 (meaning packet 0 is expected).

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

FSMs

Figure shows the FSMs for the Stop-and-Wait protocol. Since the protocol is a connection-oriented protocol, both ends should be in the established state before exchanging data packets. The states are actually nested in the established state.



Sender

The sender is initially in the ready state, but it can move between the ready and blocking state. The variable S is initialized to 0.

- **Ready state.** When the sender is in this state, it is only waiting for one event to occur. If a request comes from the application layer, the sender creates a packet with the sequence number set to S. A copy of the packet is stored, and the packet is sent. The sender then starts the only timer. The sender then moves to the blocking state.
- **Blocking state.** When the sender is in this state, three events can occur:
  a. If an error-free ACK arrives with the ackNo related to the next packet to be sent, which means ackNo = (S + 1) modulo 2, then the timer is stopped. The window slides, S = (S + 1) modulo 2. Finally, the sender moves to the ready state.
  b. If a corrupted ACK or an error-free ACK with the ackNo ≠ (S + 1) modulo 2 arrives, the ACK is discarded.
  c. If a time-out occurs, the sender resends the only outstanding packet and restarts the timer.

Receiver

The receiver is always in the ready state. Three events may occur:

a. If an error-free packet with seqNo = R arrives, the message in the packet is delivered to the application layer. The window then slides, R = (R + 1) modulo 2. Finally an ACK with ackNo = R is sent.

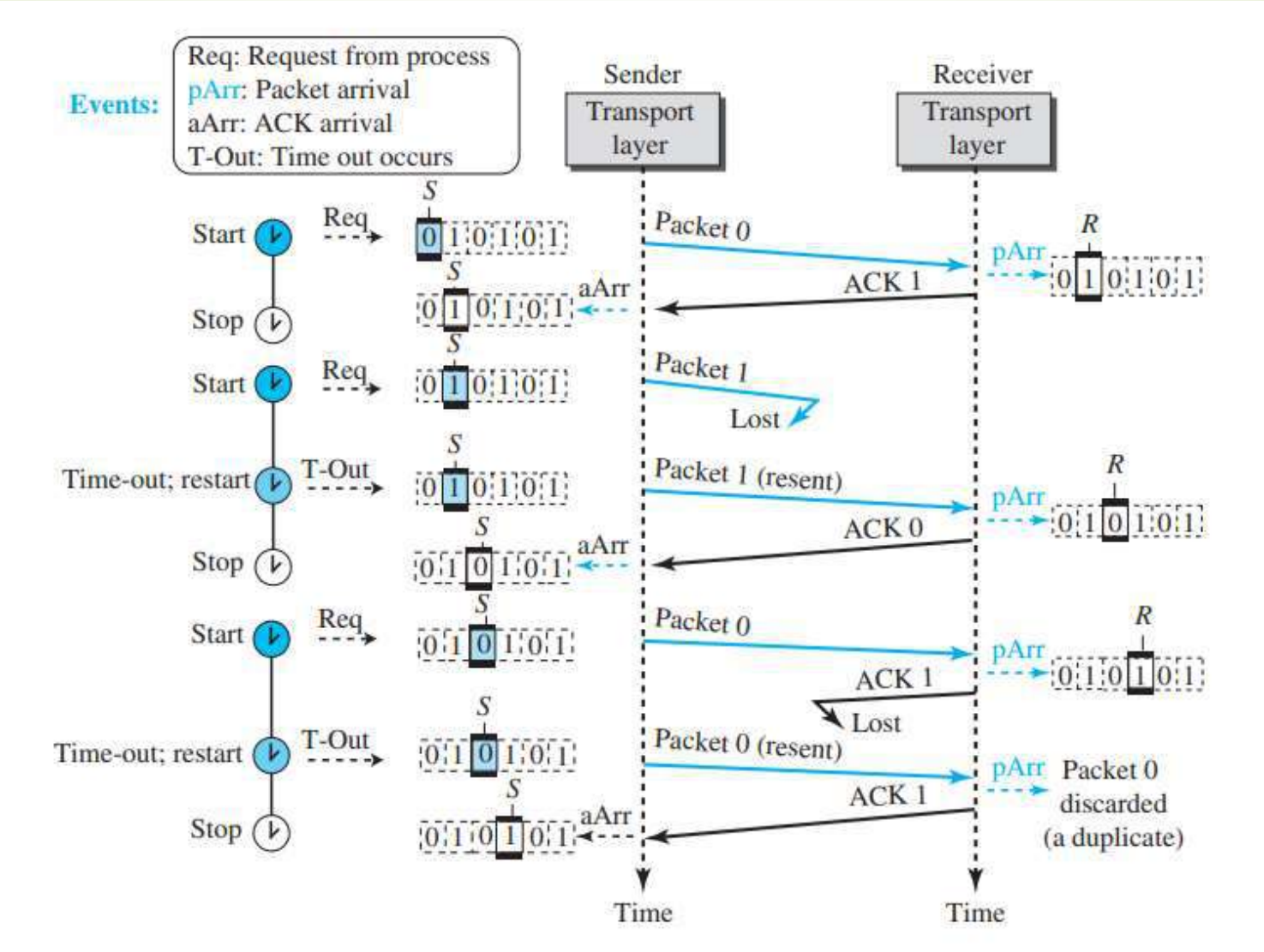
b. If an error-free packet with seqNo ≠ R arrives, the packet is discarded, but an ACK with ackNo = R is sent.

c. If a corrupted packet arrives, the packet is discarded.

Efficiency

- The Stop-and-Wait protocol is very inefficient if our channel is thick and long.
- By thick, we mean that our channel has a large bandwidth (high data rate); by long, we mean the round-trip delay is long. The product of these two is called the bandwidth-delay product.

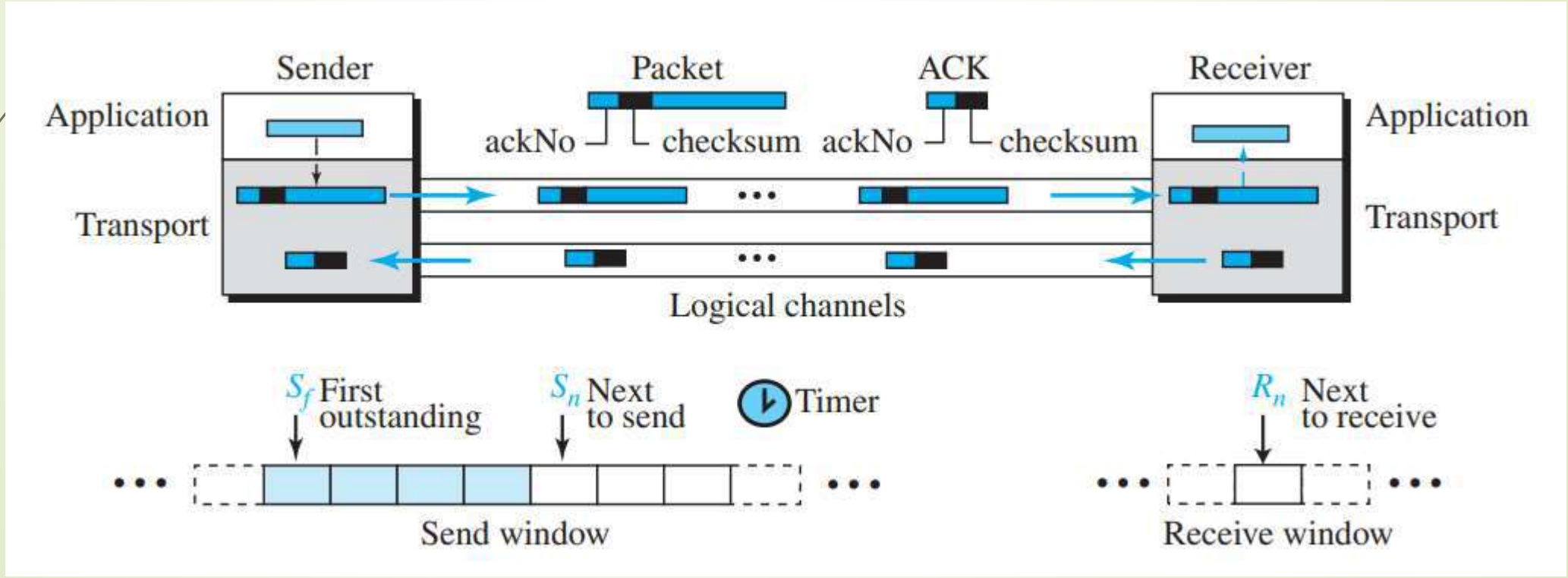
Figure shows an example of the Stop-and-Wait protocol. Packet 0 is sent and acknowledged. Packet 1 is lost and resent after the time-out. The resent packet 1 is acknowledged and the timer stops. Packet 0 is sent and acknowledged, but the acknowledgment is lost. The sender has no idea if the packet or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



Flow diagram for Example

Go-Back-N Protocol (GBN)

- In other words, we need to let more than one packet be outstanding to keep the channel busy while the sender is waiting for acknowledgment.
- The key to Go-back-N is that we can send several packets before receiving acknowledgments, but the receiver can only buffer one packet.
- We keep a copy of the sent packets until the acknowledgments arrive.



Sequence Numbers

As we mentioned before, the sequence numbers are modulo 2^m, where m is the size of the sequence number field in bits.

Acknowledgment Numbers

- An acknowledgment number in this protocol is cumulative and defines the sequence number of the next packet expected.
- For example, if the acknowledgment number (ackNo) is 7, it means all packets with sequence number up to 6 have arrived, safe and sound, and the receiver is expecting the packet with sequence number 7.

Send Window

- The send window is an imaginary box covering the sequence numbers of the data packets that can be in transit or can be sent.
- In each window position, some of these sequence numbers define the packets that have been sent; others define those that can be sent.

Figure shows a sliding window of size 7 (m = 3) for the Go-Back-N protocol.

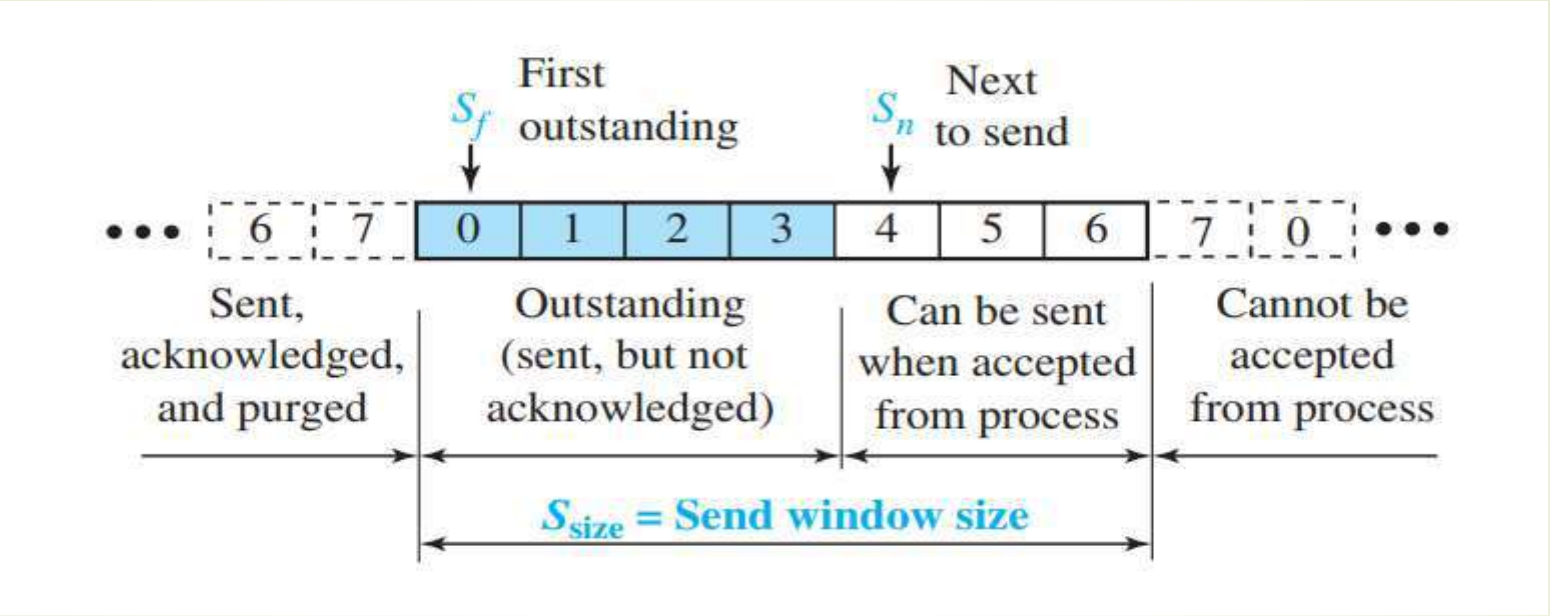


Figure: Send window for Go-Back-N

- The send window at any time divides the possible sequence numbers into four regions.
- The first region, left of the window, defines the sequence numbers belonging to packets that are already acknowledged. The sender does not worry about these packets and keeps no copies of them.
- The second region, colored, defines the range of sequence numbers belonging to the packets that have been sent, but have an unknown status. The sender needs to wait to find out if these packets have been received or were lost. We call these outstanding packets.
- The third range, white in the figure, defines the range of sequence numbers for packets that can be sent; however, the corresponding data have not yet been received from the application layer.
- Finally, the fourth region, right of the window, defines sequence numbers that cannot be used until the window slides.

Receive Window

- The receive window makes sure that the correct data packets are received and that the correct acknowledgments are sent. In Go-Back-N, the size of the receive window is always 1.
- The receiver is always looking for the arrival of a specific packet. Any packet arriving out of order is discarded and needs to be resent.
- The sequence numbers to the left of the window belong to the packets already received and acknowledged; the sequence numbers to the right of this window define the packets that cannot be received.
- Any received packet with a sequence number in these two regions is discarded. Only a packet with a sequence number matching the value of R_n is accepted and acknowledged.
- The receive window also slides, but only one slot at a time. When a correct packet is received, the window slides, $R_n = (R_n + 1)$ modulo 2^m.

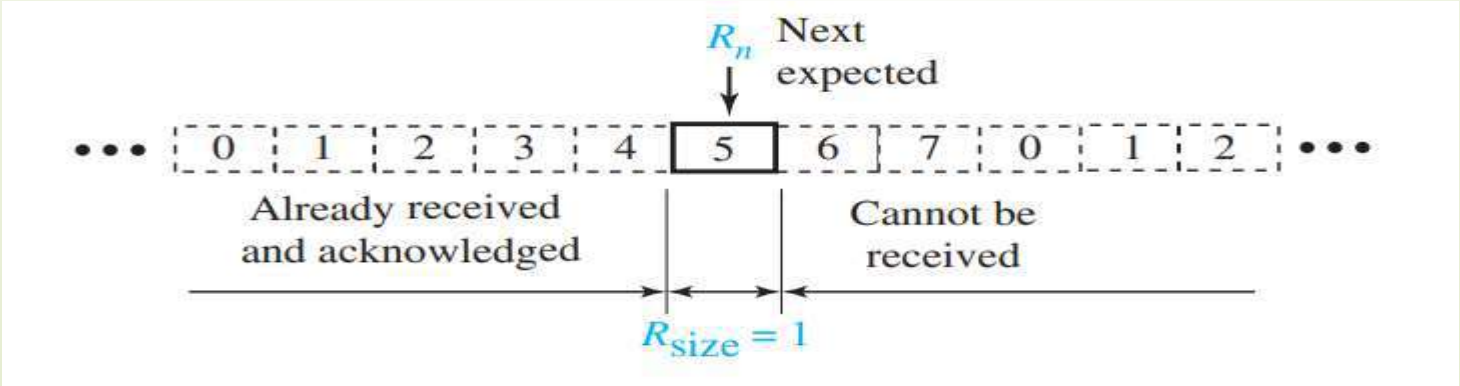


Figure Receive window for Go-Back-N

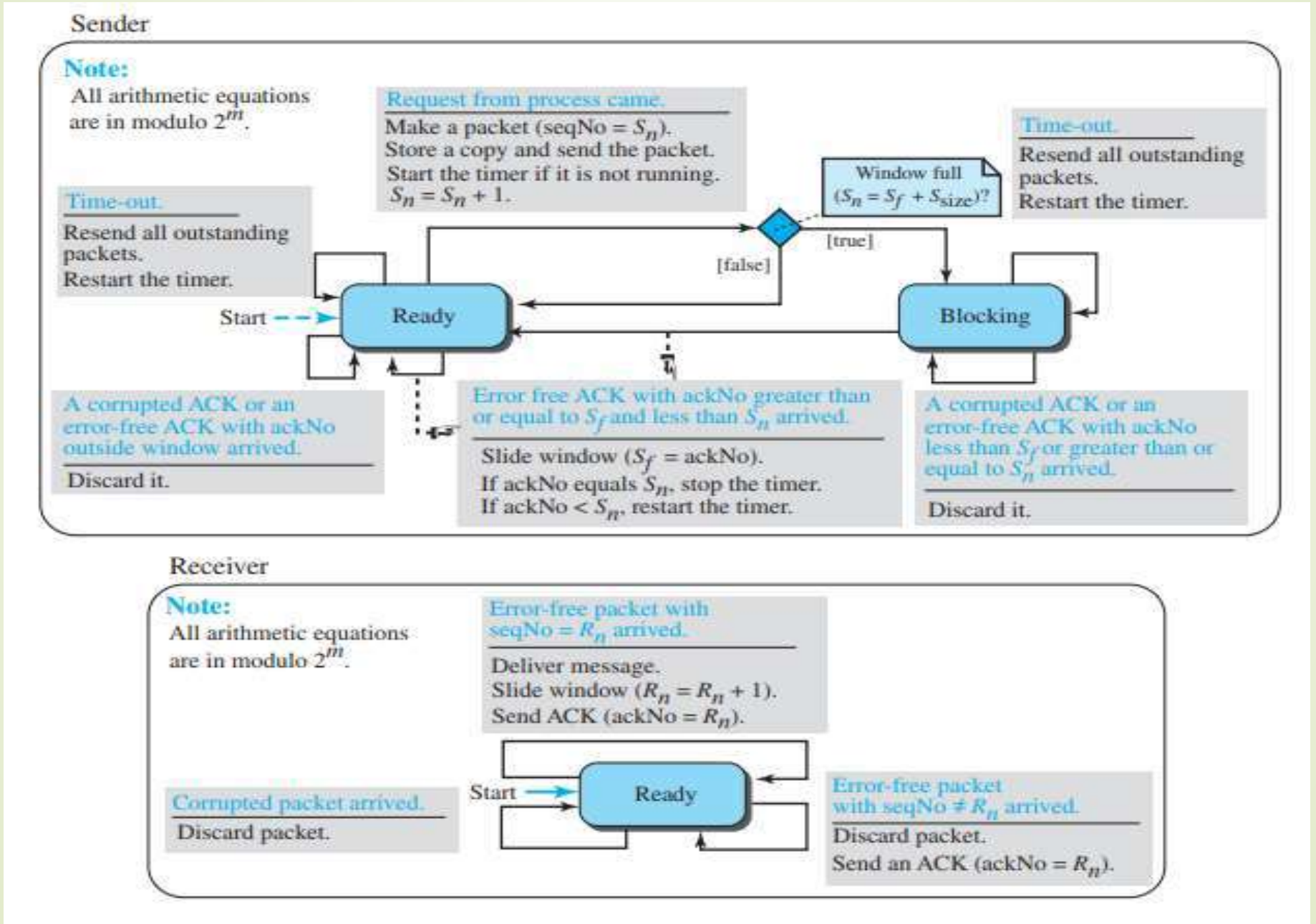
Timers

- Although there can be a timer for each packet that is sent, in our protocol we use only one. The reason is that the timer for the first outstanding packet always expires first. We resend all outstanding packets when this timer expires.

Resending packets

- When the timer expires, the sender resends all outstanding packets.
- For example, suppose the sender has already sent packet 6 (Sn = 7), but the only timer expires. If Sf = 3, this means that packets 3, 4, 5, and 6 have not been acknowledged; the sender goes back and resends packets 3, 4, 5, and 6. That is why the protocol is called Go-Back-N. On a time-out, the machine goes back N locations and resends all packets.

FSMs : Figure shows the FSMs for the GBN protocol.



Sender

The sender starts in the ready state, but thereafter it can be in one of the two states: ready or blocking. The two variables are normally initialized to 0 (Sf = Sn = 0).

Ready state. Four events may occur when the sender is in ready state.

a. If a request comes from the application layer, the sender creates a packet with the sequence number set to Sn. A copy of the packet is stored, and the packet is sent. The sender also starts the only timer if it is not running. The value of Sn is now incremented, (Sn = Sn + 1) modulo 2m. If the window is full, Sn = (Sf + Ssize) modulo 2m, the sender goes to the blocking state.

b. If an error-free ACK arrives with ackNo related to one of the outstanding packets, the sender slides the window (set Sf = ackNo), and if all outstanding packets are acknowledged (ackNo = Sn), then the timer is stopped. If all outstanding packets are not acknowledged, the timer is restarted.

c. If a corrupted ACK or an error-free ACK with ackNo not related to the outstanding packet arrives, it is discarded.

d. If a time-out occurs, the sender resends all outstanding packets and restarts the timer.

Blocking state. Three events may occur in this case:

a. If an error-free ACK arrives with ackNo related to one of the outstanding packets, the sender slides the window (set Sf = ackNo) and if all outstanding packets are acknowledged (ackNo = Sn), then the timer is stopped. If all outstanding packets are not acknowledged, the timer is restarted. The sender then moves to the ready state.

b. If a corrupted ACK or an error-free ACK with the ackNo not related to the outstanding packets arrives, the ACK is discarded.

c. If a time-out occurs, the sender sends all outstanding packets and restarts the timer.

Receiver

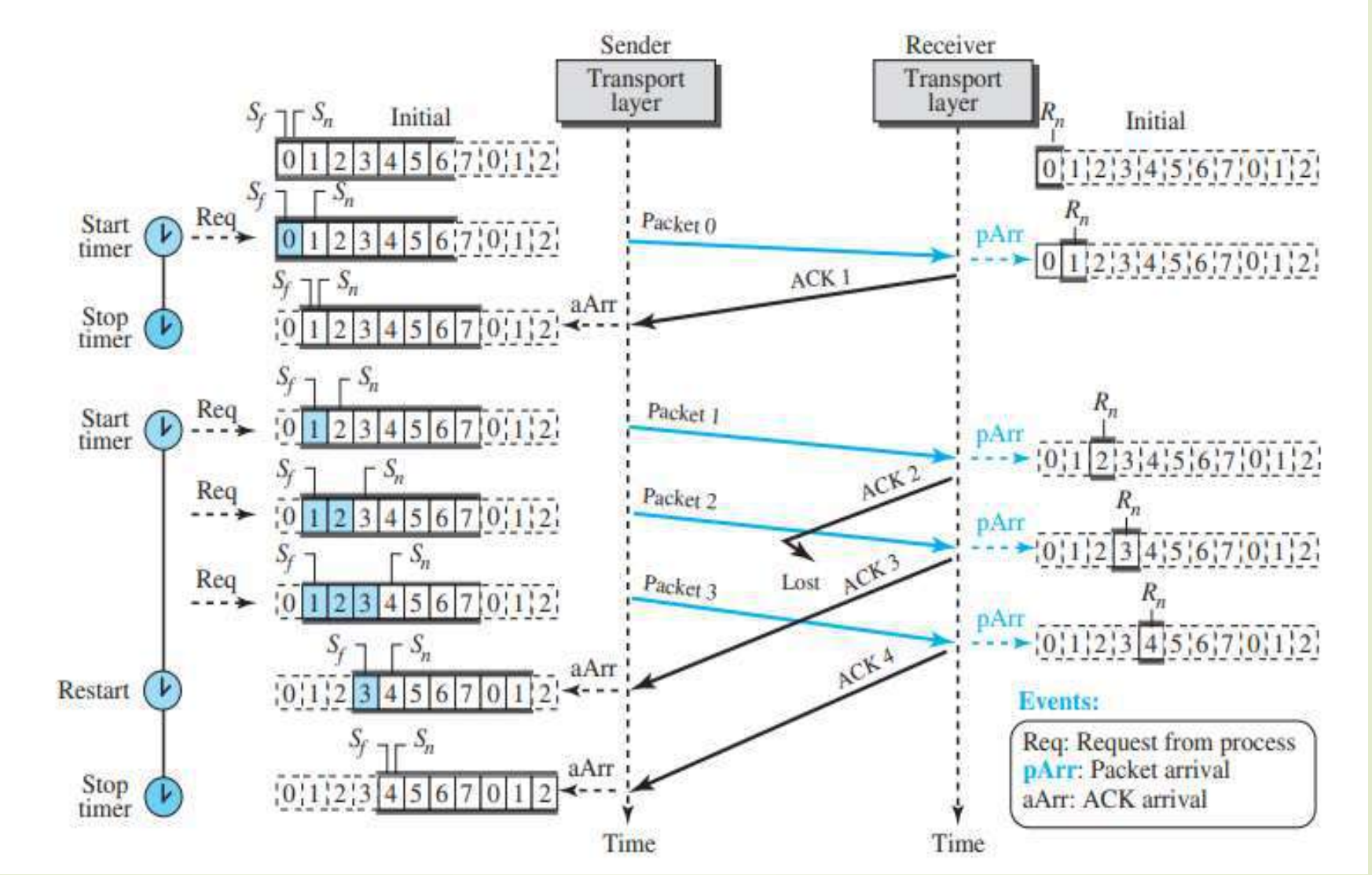
The receiver is always in the ready state. The only variable, Rn, is initialized to 0. Three events may occur:

a. If an error-free packet with seqNo = Rn arrives, the message in the packet is delivered to the application layer. The window then slides, Rn = (Rn + 1) modulo 2m. Finally an ACK is sent with ackNo = Rn.

b. If an error-free packet with seqNo outside the window arrives, the packet is discarded, but an ACK with ackNo = Rn is sent.

c. If a corrupted packet arrives, it is discarded.

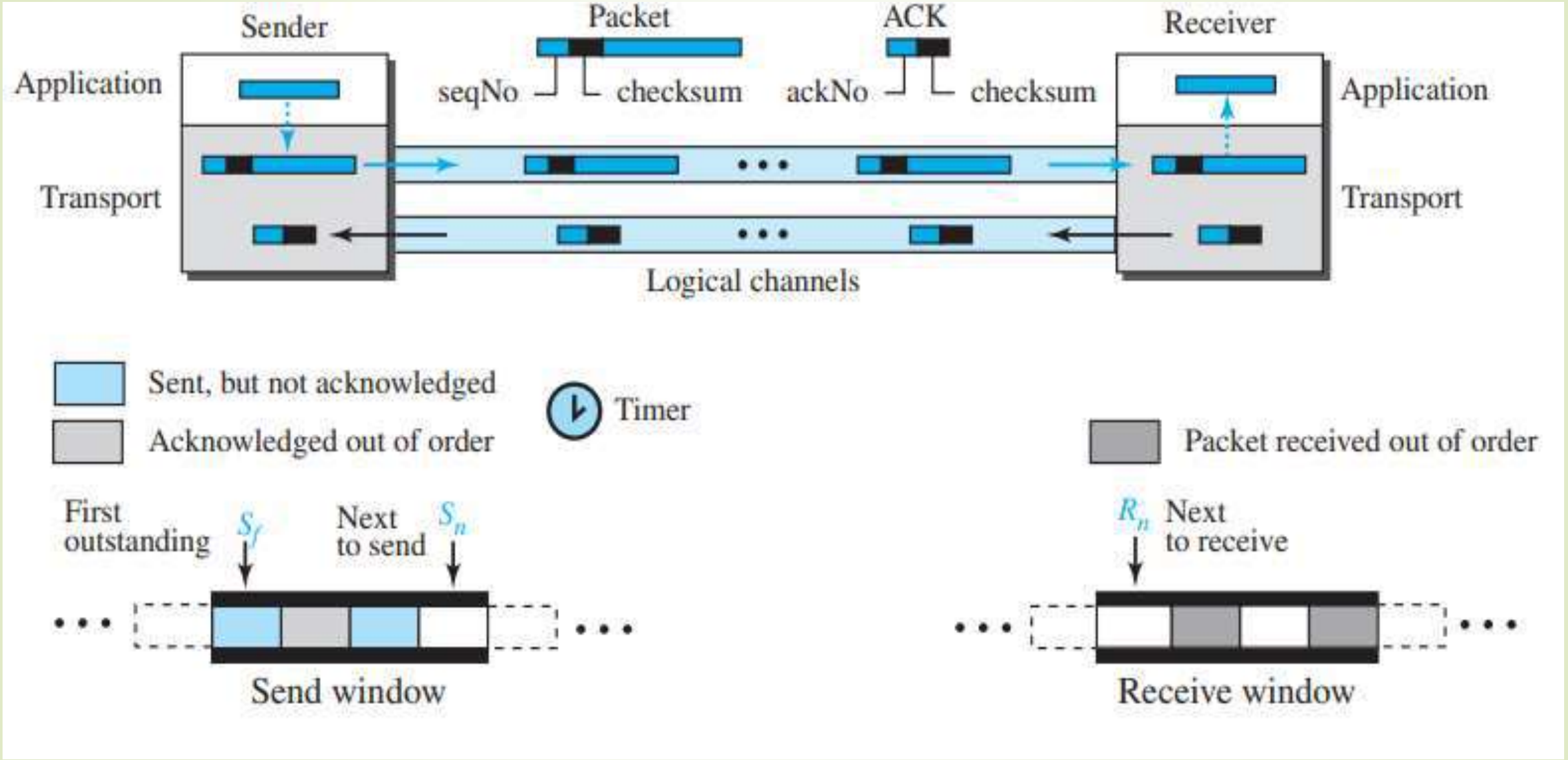
Example: Figure shows an example of Go-Back-N. This is an example of a case where the forward channel is reliable, but the reverse is not. No data packets are lost, but some ACKs are delayed and one is lost. The example also shows how cumulative acknowledgments can help if acknowledgments are delayed or lost.



Selective-Repeat Protocol

- The Go-Back-N protocol simplifies the process at the receiver. The receiver keeps track of only one variable, and there is no need to buffer out-of-order packets; they are simply discarded.
- However, this protocol is inefficient if the underlying network protocol loses a lot of packets. Each time a single packet is lost or corrupted, the sender resends all outstanding packets, even though some of these packets may have been received safe and sound but out of order.
- If the network layer is losing many packets because of congestion in the network, the resending of all of these outstanding packets makes the congestion worse, and eventually more packets are lost.
- Selective-Repeat (SR) protocol, has been devised, which, as the name implies, resends only selective packets, those that are actually lost.

Figure: Outline of Selective-Repeat



Windows

The Selective-Repeat protocol also uses two windows: a send window and a receive window. However, there are differences between the windows in this protocol and the ones in Go-Back-N.

- First, the maximum size of the send window is much smaller; it is $2^m - 1$.
- The Second, the receive window is the same size as the send window.

The send window maximum size can be $2^m - 1$. For example, if m = 4, the sequence numbers go from 0 to 15, but the maximum size of the window is just 8 (it is 15 in the Go-Back-N Protocol).

- The receive window in Selective-Repeat is totally different from the one in Go-Back-N.
- The size of the receive window is the same as the size of the send window (maximum $2^m - 1$).
- The Selective-Repeat protocol allows as many packets as the size of the receive window to arrive out of order and be kept until there is a set of consecutive packets to be delivered to the application layer.
- Because the sizes of the send window and receive window are the same, all the packets in the send packet can arrive out of order and be stored until they can be delivered.

Figure: Send window for Selective-Repeat protocol

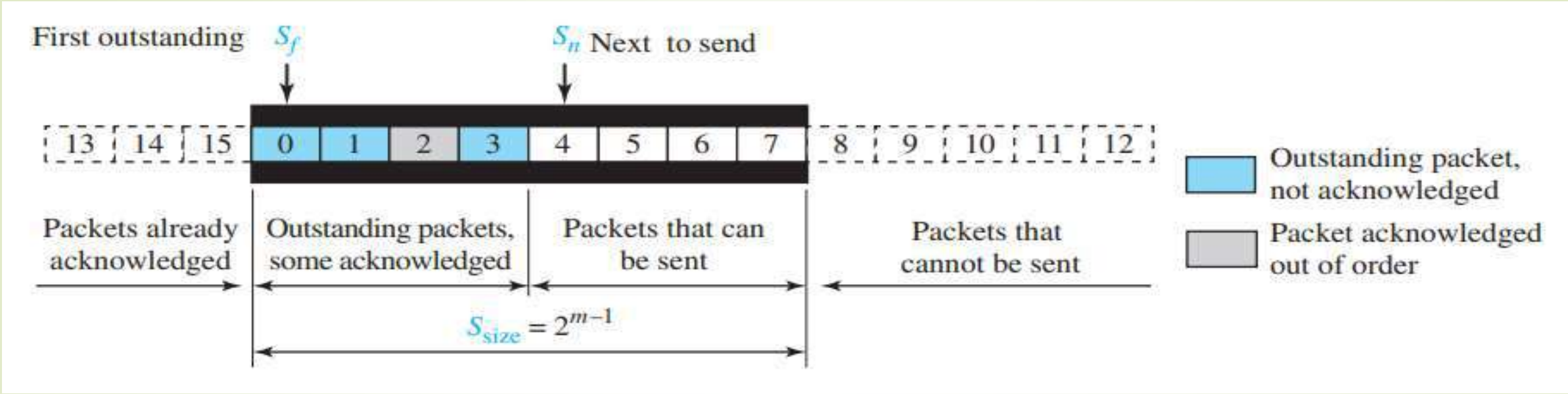
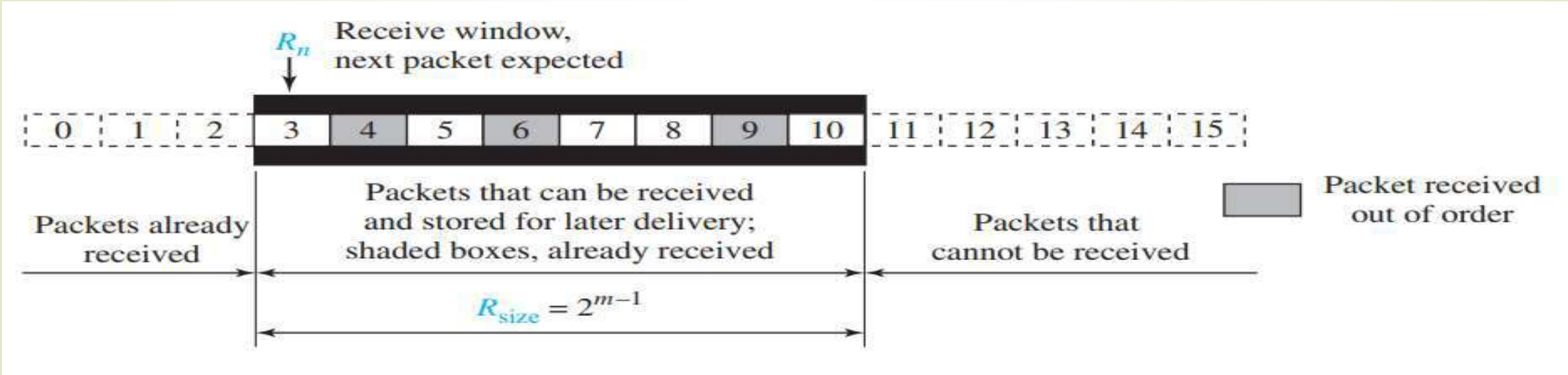


Figure: Receive window for Selective-Repeat protocol



Timer

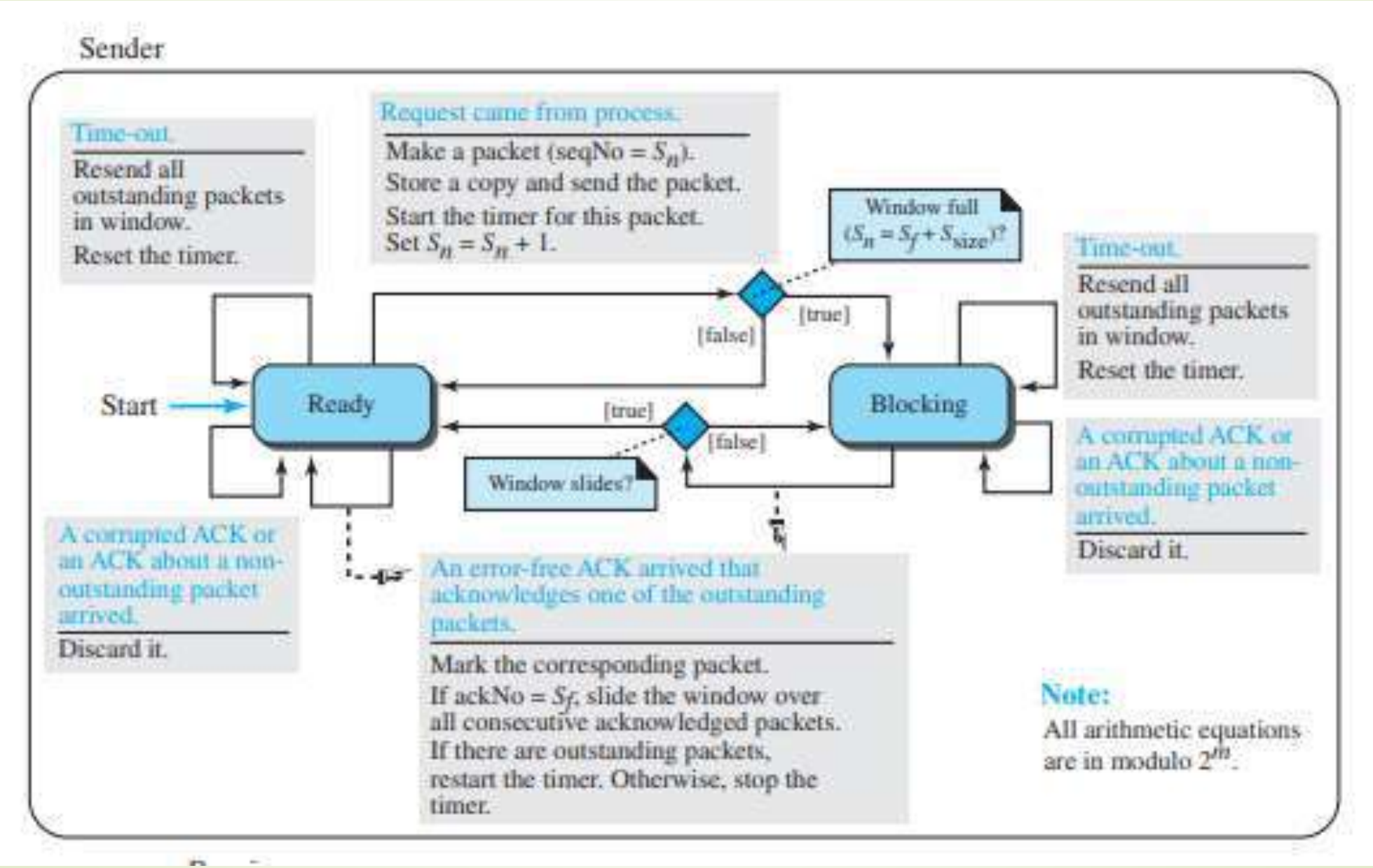
Theoretically, Selective-Repeat uses one timer for each outstanding packet. When a timer expires, only the corresponding packet is resent. In other words, GBN treats outstanding packets as a group; SR treats them individually. However, most transport-layer protocols that implement SR use only a single timer.

Acknowledgments

- There is yet another difference between the two protocols. In GBN an ackNo is cumulative; it defines the sequence number of the next packet expected, confirming that all previous packets have been received safe and sound.
- The semantics of acknowledgment is different in SR. In SR, an ackNo defines the sequence number of a single packet that is received safe and sound; there is no feedback for any other.

In the Selective-Repeat protocol, an acknowledgment number defines the sequence number of the error-free packet received.

FSMs: Figure shows the FSMs for the Selective-Repeat protocol.



Sender

The sender starts in the ready state, but later it can be in one of the two states: ready or blocking. The following shows the events and the corresponding actions in each state.

Ready state. Four events may occur in this case:

a. If a request comes from the application layer, the sender creates a packet with the sequence number set to Sn. A copy of the packet is stored, and the packet is sent. If the timer is not running, the sender starts the timer. The value of Sn is now incremented, Sn = (Sn + 1) modulo 2m. If the window is full, Sn = (Sf + Ssize) modulo 2m, the sender goes to the blocking state.

b. If an error-free ACK arrives with ackNo related to one of the outstanding packets, that packet is marked as acknowledged. If the ackNo = Sf , the window slides to the right until the Sf points to the first unacknowledged packet (all consecutive acknowledged packets are now outside the window). If there are outstanding packets, the timer is restarted; otherwise, the timer is stopped.

c. If a corrupted ACK or an error-free ACK with ackNo not related to an outstanding packet arrives, it is discarded.

d. If a time-out occurs, the sender resends all unacknowledged packets in the window and restarts the timer.

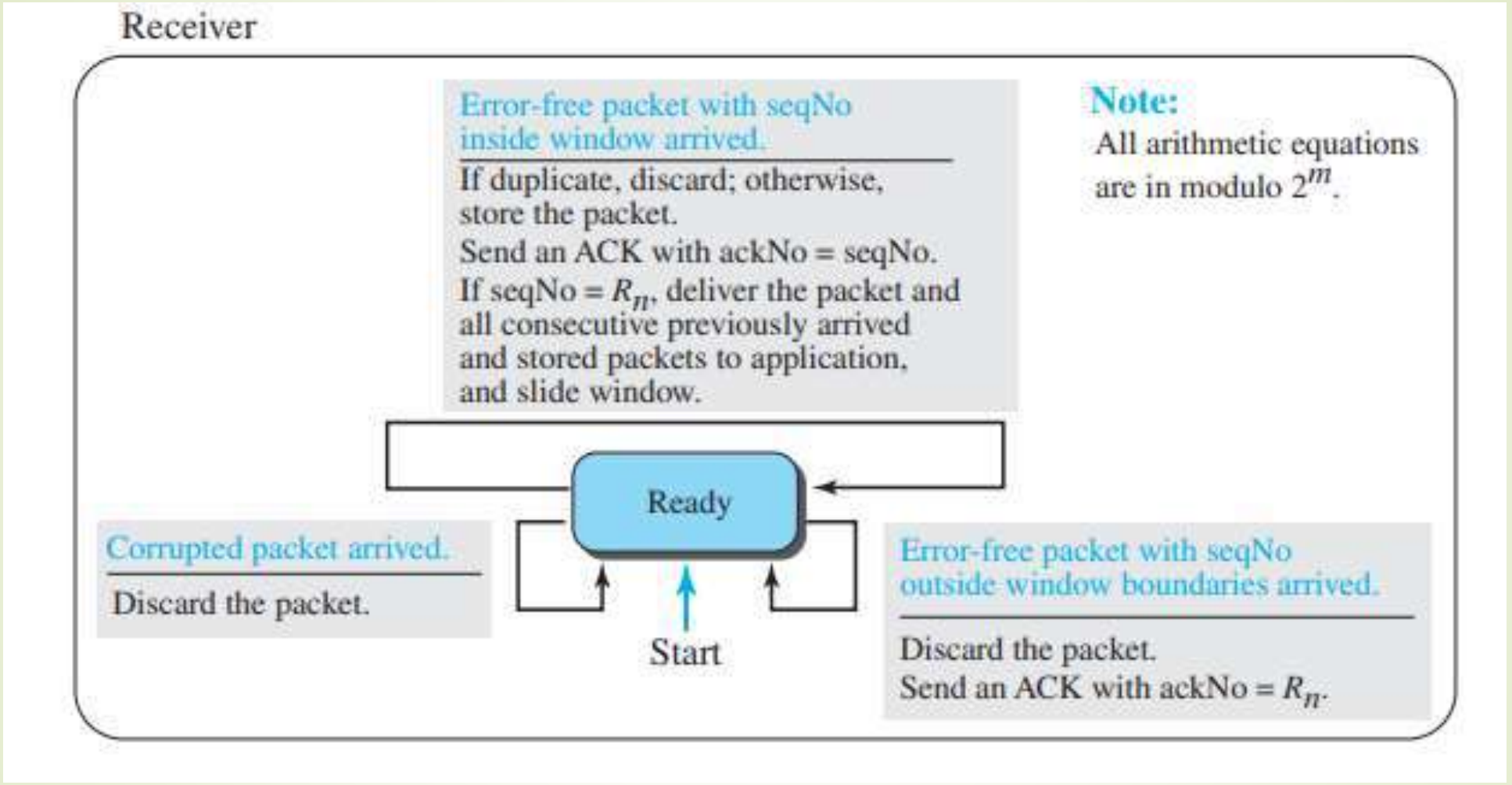
Blocking state

Three events may occur in this case:

a. If an error-free ACK arrives with ackNo related to one of the outstanding packets, that packet is marked as acknowledged. In addition, if the ackNo = Sf , the window is slid to the right until the Sf points to the first unacknowledged packet (all consecutive acknowledged packets are now outside the window). If the window has slid, the sender moves to the ready state.

b. If a corrupted ACK or an error-free ACK with the ackNo not related to outstanding packets arrives, the ACK is discarded.

c. If a time-out occurs, the sender resends all unacknowledged packets in the window and restarts the timer.


Receiver
Error-free packet with seqNo inside window arrived.
If duplicate, discard; otherwise, store the packet.
Send an ACK with ackNo = seqNo.
If seqNo = R_n, deliver the packet and all consecutive previously arrived and stored packets to application, and slide window.
Note:
All arithmetic equations are in modulo 2^m.
Ready
Start
Corrupted packet arrived.
Discard the packet.
Error-free packet with seqNo outside window boundaries arrived.
Discard the packet.
Send an ACK with ackNo = R_n.

Receiver

The receiver is always in the ready state. Three events may occur:

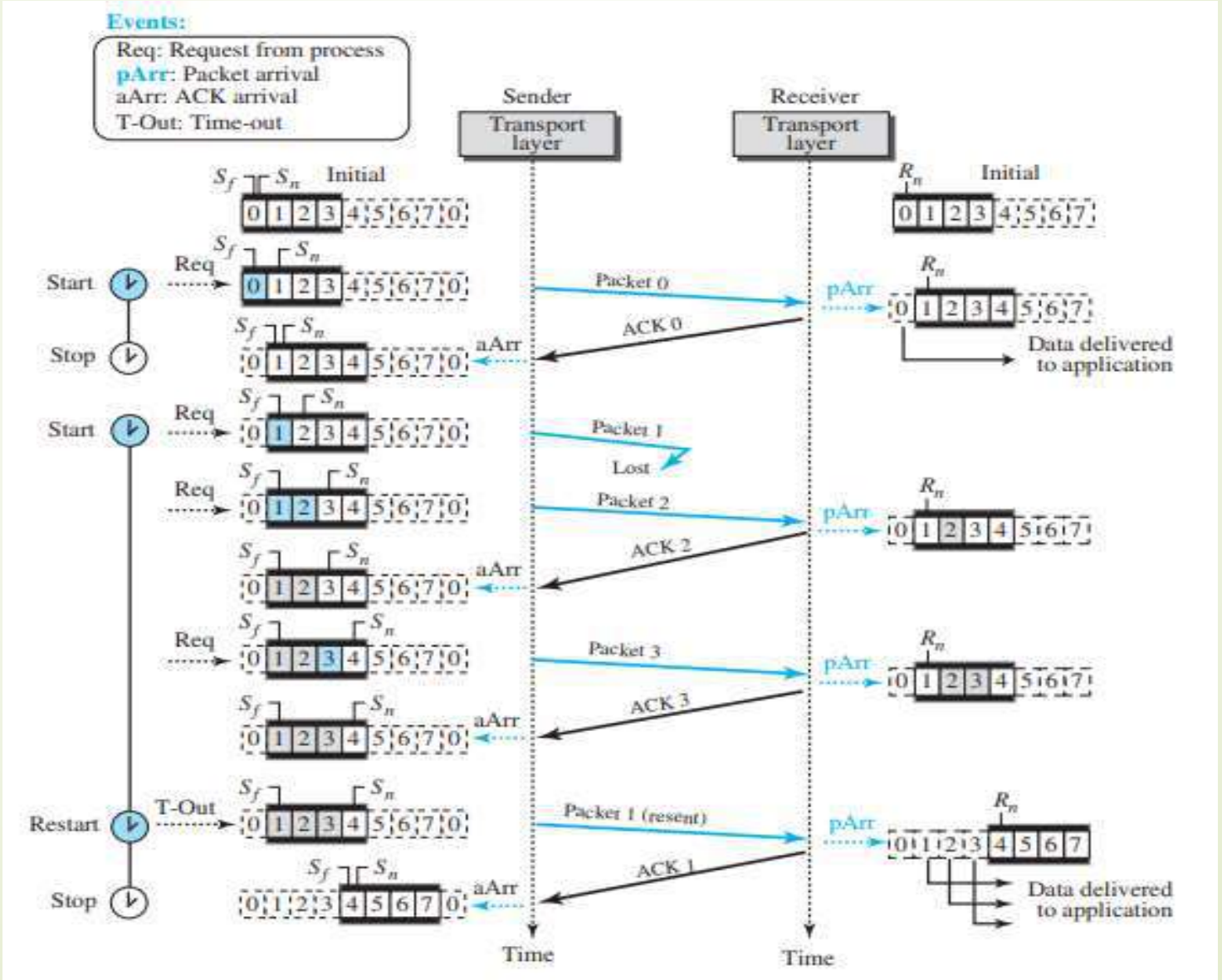
a. If an error-free packet with seqNo in the window arrives, the packet is stored and an ACK with ackNo = seqNo is sent. In addition, if the seqNo = Rn, then the packet and all previously arrived consecutive packets are delivered to the application layer and the window slides so that the Rn points to the first empty slot.

b. If an error-free packet with seqNo outside the window arrives, the packet is discarded, but an ACK with ackNo = Rn is returned to the sender. This is needed to let the sender slide its window if some ACKs related to packets with seqNo < Rn were lost.

c. If a corrupted packet arrives, the packet is discarded.

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

Example: Packets 0, 1, 2, and 3 are sent. However, packet 1 is lost. The receiver receives packets 2 and 3, but they are discarded because they are received out of order (packet 1 is expected). When the receiver receives packets 2 and 3, it sends ACK1 to show that it expects to receive packet 1. However, these ACKs are not useful for the sender because the ackNo is equal to Sf, not greater than Sf. So the sender discards them. When the time-out occurs, the sender resends packets 1, 2, and 3, which are acknowledged.



Bidirectional Protocols: Piggybacking

- The four protocols we discussed earlier in this section are all unidirectional: data packets flow in only one direction and acknowledgments travel in the other direction.
- In real life, data packets are normally flowing in both directions: from client to server and from server to client. This means that acknowledgments also need to flow in both directions.
- A technique called piggybacking is used to improve the efficiency of the bidirectional protocols.
- When a packet is carrying data from A to B, it can also carry acknowledgment feedback about arrived packets from B; when a packet is carrying data from B to A, it can also carry acknowledgment feedback about the arrived packets from A.
- The client and server each use two independent windows: send and receive.

Figure: Design of piggybacking in Go-Back-N

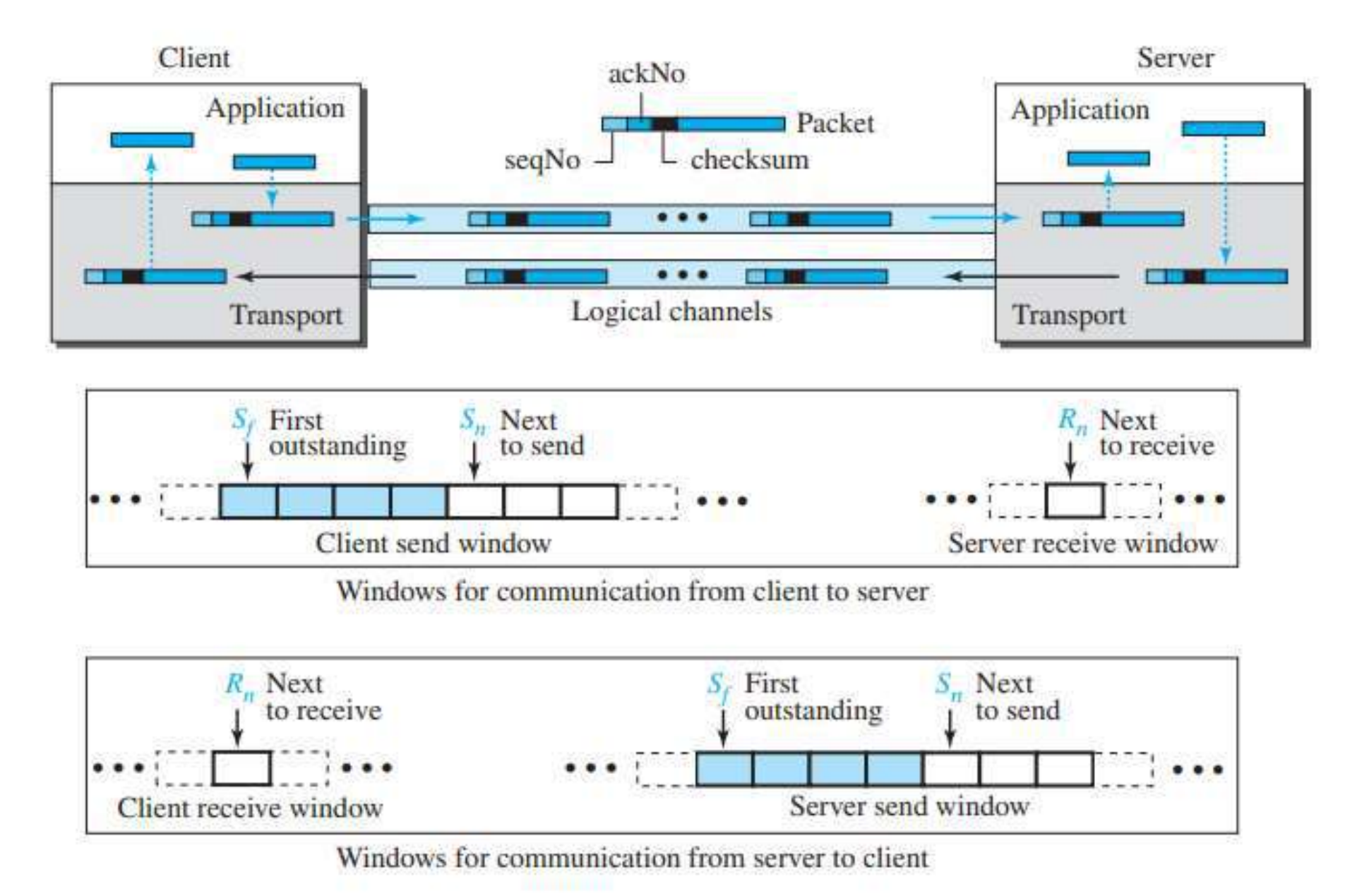
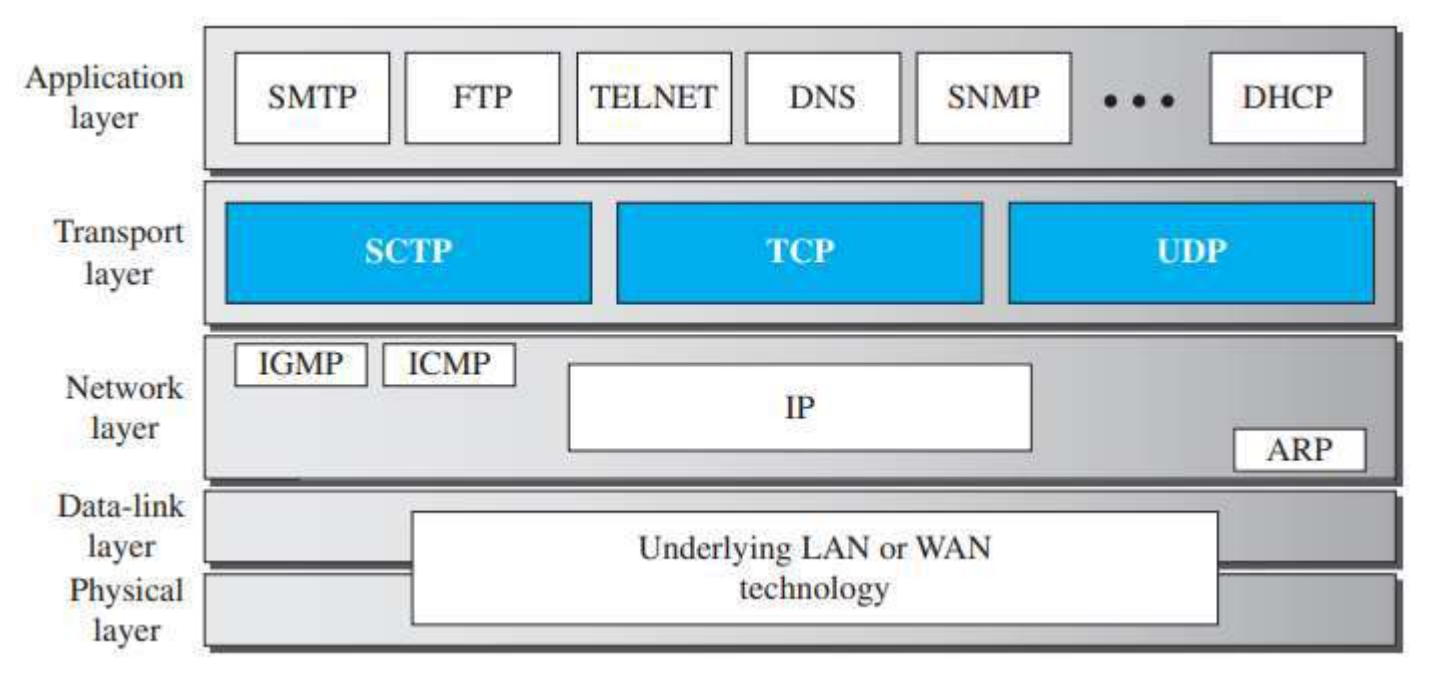


Figure: Position of transport-layer protocols in the TCP/IP protocol suite



Services

- Each protocol provides a different type of service and should be used appropriately.
- **UDP** is an unreliable connectionless transport-layer protocol used for its simplicity and efficiency in applications where error control can be provided by the application-layer process.
- **TCP** is a reliable connection-oriented protocol that can be used in any application where reliability is important.
- **SCTP** is a new transport-layer protocol that combines the features of UDP and TCP.

USER DATAGRAM PROTOCOL

- The User Datagram Protocol (UDP) is a connectionless, unreliable transport protocol. It does not add anything to the services of IP except for providing process-to-process communication instead of host-to-host communication.
- If UDP is so powerless, why would a process want to use it? With the disadvantages come some advantages. 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 using UDP takes much less interaction between the sender and receiver than using TCP.

User Datagram

UDP packets, called user datagrams, have a fixed-size header of 8 bytes made of four fields, each of 2 bytes (16 bits).

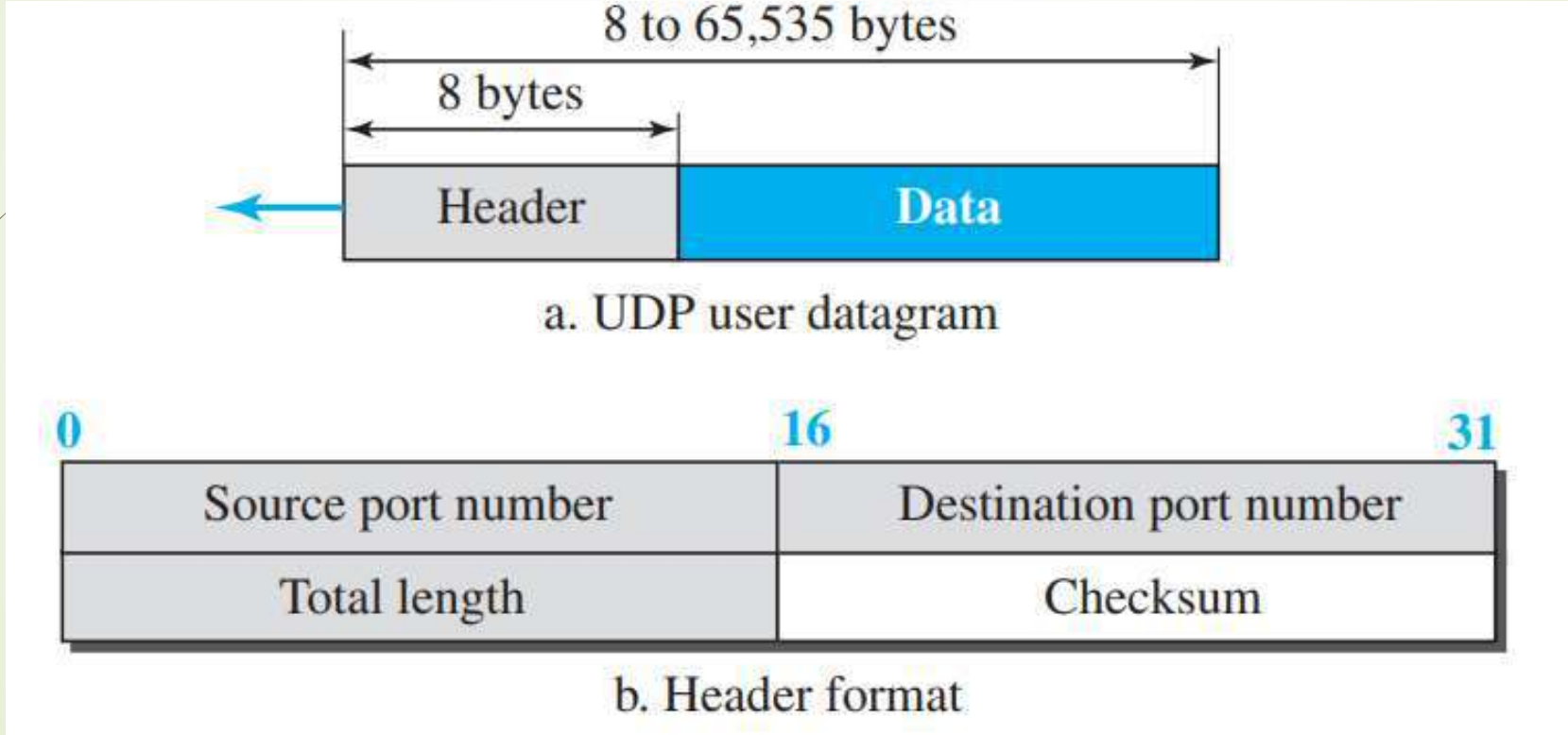


Figure: User datagram packet format

Example

The following is the content of a UDP header in hexadecimal format.

CB84000D001C001C

a. What is the source port number?
b. What is the destination port number?
c. What is the total length of the user datagram?
d. What is the length of the data?
e. Is the packet directed from a client to a server or vice versa?
f. What is the client process?

Solution

a. The source port number is the first four hexadecimal digits $(CB84)_{16}$, which means that the source port number is 52100.
b. The destination port number is the second four hexadecimal digits $(000D)_{16}$, which means that the destination port number is 13.
c. The third four hexadecimal digits $(001C)_{16}$ define the length of the whole UDP packet as 28 bytes.
d. The length of the data is the length of the whole packet minus the length of the header, or $28 - 8 = 20$ bytes.
e. Since the destination port number is 13 (well-known port), the packet is from the client to the server.
f. The client process is the Daytime (see Table 24.1).

UDP Services

Process-to-Process Communication

UDP provides process-to-process communication using socket addresses, a combination of IP addresses and port numbers.

Connectionless Services

- UDP provides a connectionless service. This means that each user datagram sent by UDP is an independent datagram. There is no relationship between the different user datagrams even if they are coming from the same source process and going to the same destination program.
- The user datagrams are not numbered. Also, unlike TCP, there is no connection establishment and no connection termination. This means that each user datagram can travel on a different path.

Flow Control

- There is no flow control, and hence no window mechanism. The receiver may overflow with incoming messages. The lack of flow control means that the process using UDP should provide for this service, if needed.

Error Control

- There is no error control mechanism in UDP except for the checksum. This means that the sender does not know if a message has been lost or duplicated.
- When the receiver detects an error through the checksum, the user datagram is silently discarded. The lack of error control means that the process using UDP should provide for this service, if needed.

Congestion

- Control Since UDP is a connectionless protocol, it does not provide congestion control. UDP assumes that the packets sent are small and sporadic and cannot create congestion in the network.

The following shows some typical applications that can benefit more from the services of UDP than from those of TCP.

- UDP is suitable for a process that requires simple request-response communication with little concern for flow and error control. It is not usually used for a process such as FTP that needs to send bulk data.
- UDP is suitable for a process with internal flow- and error-control mechanisms. For example, the Trivial File Transfer Protocol (TFTP) process includes flow and error control. It can easily use UDP.
- UDP is a suitable transport protocol for multicasting. Multicasting capability is embedded in the UDP software but not in the TCP software.
- UDP is used for some route updating protocols such as Routing Information Protocol (RIP).
- UDP is normally used for interactive real-time applications that cannot tolerate uneven delay between sections of a received message.

TRANSMISSION CONTROL PROTOCOL

- Transmission Control Protocol (TCP) is a connection-oriented, reliable protocol. TCP explicitly defines connection establishment, data transfer, and connection teardown phases to provide a connection-oriented service.
- TCP uses a combination of GBN and SR protocols to provide reliability.
- TCP uses checksum (for error detection), retransmission of lost or corrupted packets, cumulative and selective acknowledgments, and timers.

TCP Services

- Process-to-Process Communication
- Stream Delivery Service
- Sending and Receiving Buffers
- Segments
- Full-Duplex Communication
- Multiplexing and Demultiplexing
- Reliable Service

Process-to-Process Communication

- TCP provides process-to-process communication using port numbers.

Stream Delivery Service

- TCP, on the other hand, allows the sending process to deliver data as a stream of bytes and allows the receiving process to obtain data as a stream of bytes.
- TCP creates an environment in which the two processes seem to be connected by an imaginary "tube" that carries their bytes across the Internet.

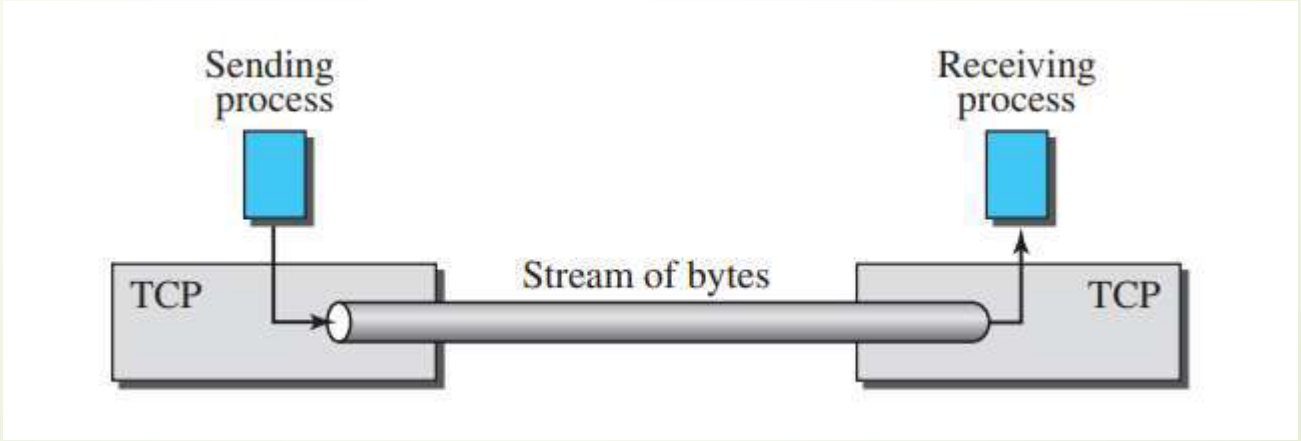


Figure: Stream delivery

Sending and Receiving Buffers

- Because the sending and the receiving processes may not necessarily write or read data at the same rate, TCP needs buffers for storage.
- There are two buffers, the sending buffer and the receiving buffer, one for each direction.
- One way to implement a buffer is to use a circular array of 1-byte locations as shown in Figure

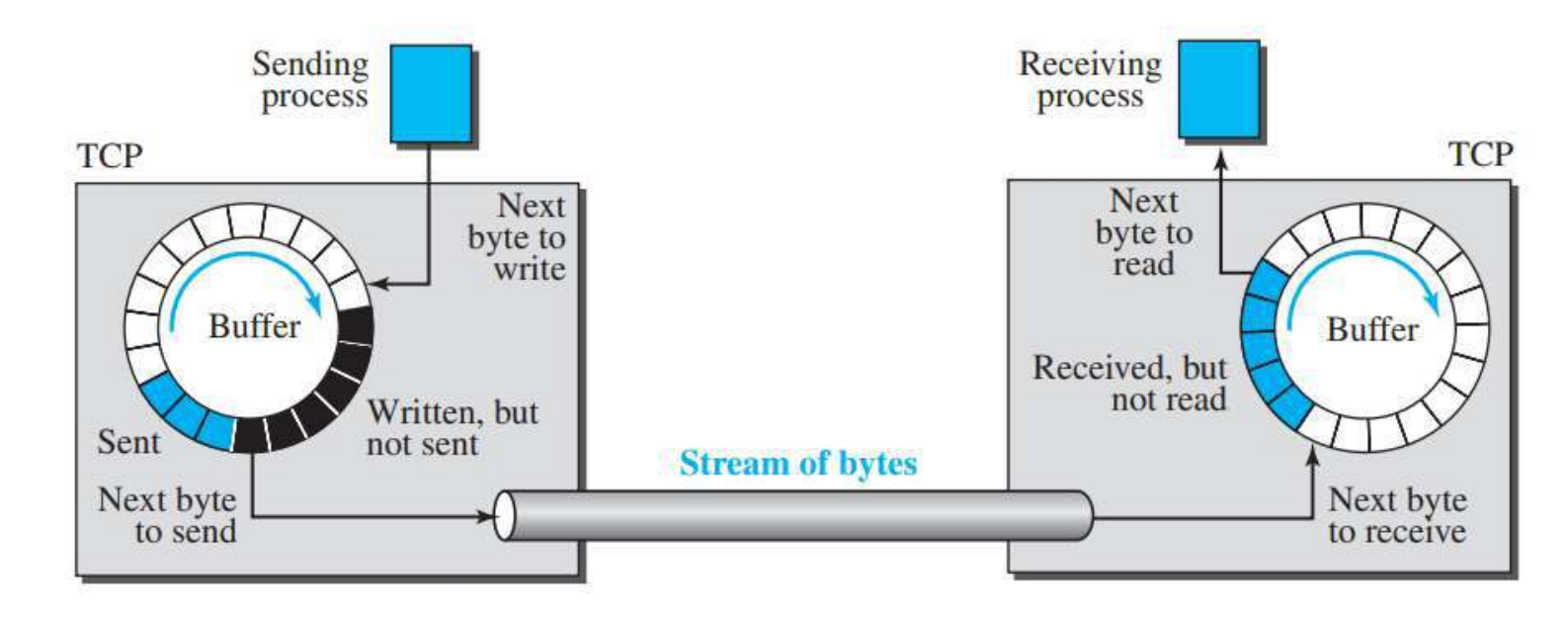
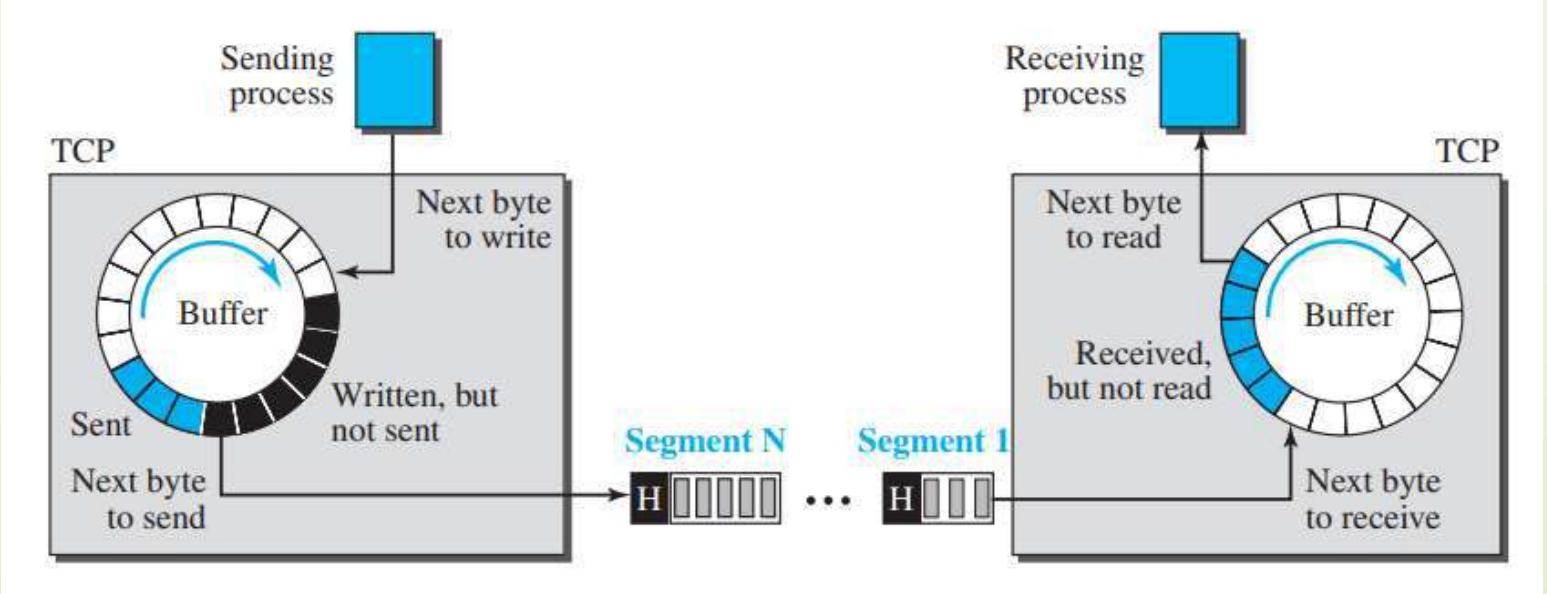


Figure: Sending and receiving buffers

Segments

- Although buffering handles the disparity between the speed of the producing and consuming processes, we need one more step before we can send data.
- The network layer, as a service provider for TCP, needs to send data in packets, not as a stream of bytes.
- At the transport layer, TCP groups a number of bytes together into a packet called a segment. TCP adds a header to each segment (for control purposes) and delivers the segment to the network layer for transmission.
- The segments are encapsulated in an IP datagram and transmitted. This entire operation is transparent to the receiving process.



Full-Duplex Communication

TCP offers full-duplex service, where data can flow in both directions at the same time. Each TCP endpoint then has its own sending and receiving buffer, and segments move in both directions.

Multiplexing and Demultiplexing

TCP performs multiplexing at the sender and demultiplexing at the receiver. However, since TCP is a connection-oriented protocol, a connection needs to be established for each pair of processes.

Reliable Service

TCP is a reliable transport protocol. It uses an acknowledgment mechanism to check the safe and sound arrival of data. We will discuss this feature further in the section on error control.

TCP Features

Numbering System

- Although the TCP software keeps track of the segments being transmitted or received, there is no field for a segment number value in the segment header.
- Instead, there are two fields, called the sequence number and the acknowledgment number. These two fields refer to a byte number and not a segment number.

Byte Number

- TCP numbers all data bytes (octets) that are transmitted in a connection. Numbering is independent in each direction.
- When TCP receives bytes of data from a process, TCP stores them in the sending buffer and numbers them.
- The numbering does not necessarily start from 0. Instead, TCP chooses an arbitrary number between 0 and $2^{32} - 1$ for the number of the first byte.

Sequence Number

- After the bytes have been numbered, TCP assigns a sequence number to each segment that is being sent. The sequence number, in each direction, is defined as follows:

1. The sequence number of the first segment is the ISN (initial sequence number), which is a random number.
2. The sequence number of any other segment is the sequence number of the previous segment plus the number of bytes (real or imaginary) carried by the previous segment.

Segment

A packet in TCP is called a segment.

Format

The segment consists of a header of 20 to 60 bytes, followed by data from the application program. The header is 20 bytes if there are no options and up to 60 bytes if it contains options.

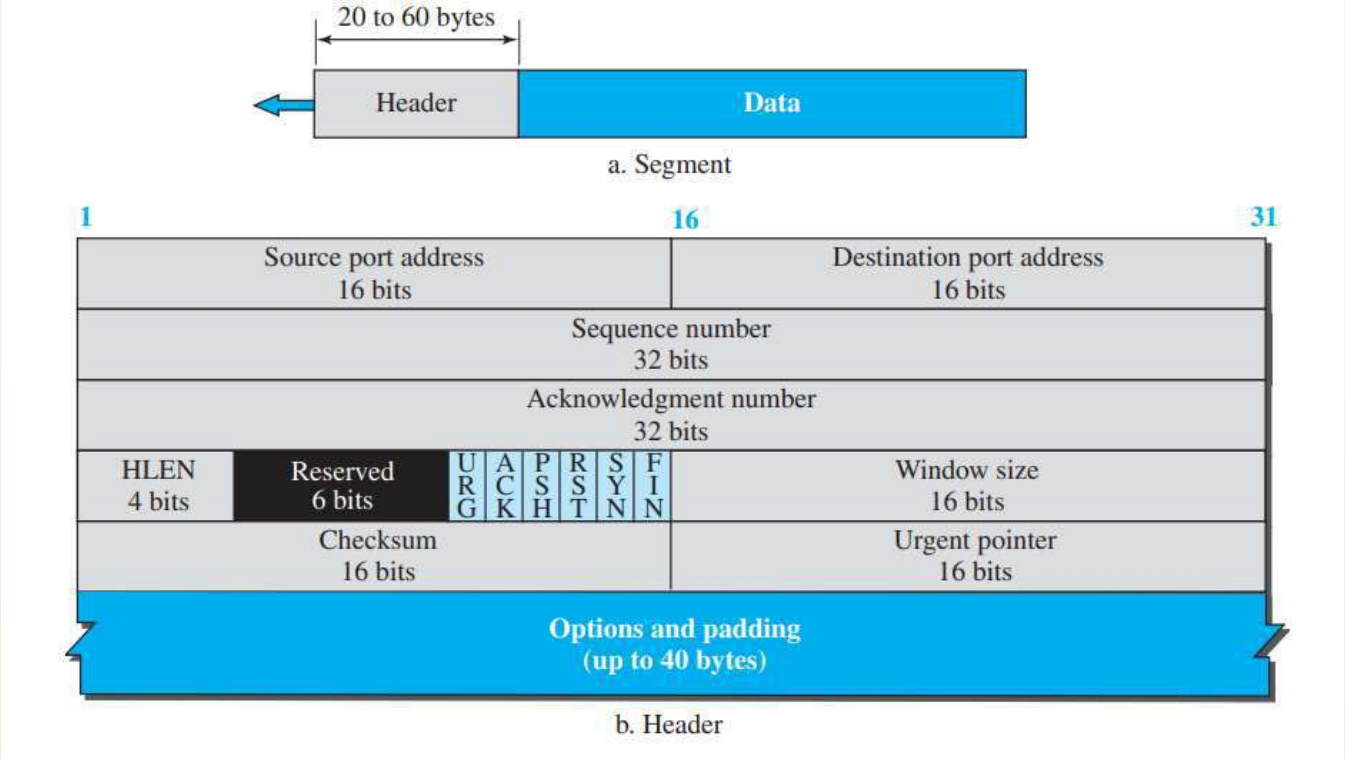


Figure: TCP segment format

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

Acknowledgment number. This 32-bit field defines the byte number that the receiver of the segment is expecting to receive from the other party.

- If the receiver of the segment has successfully received byte number x from the other party, it returns x + 1 as the acknowledgment number. Acknowledgment and data can be piggybacked together.

Header length. This 4-bit field indicates the number of 4-byte words in the TCP header. The length of the header can be between 20 and 60 bytes. Therefore, the value of this field is always between 5 (5 × 4 = 20) and 15 (15 × 4 = 60).

Control. This field defines 6 different control bits or flags, One or more of these bits can be set at a time. These bits enable flow control, connection establishment and termination, connection abortion, and the mode of data transfer in TCP.

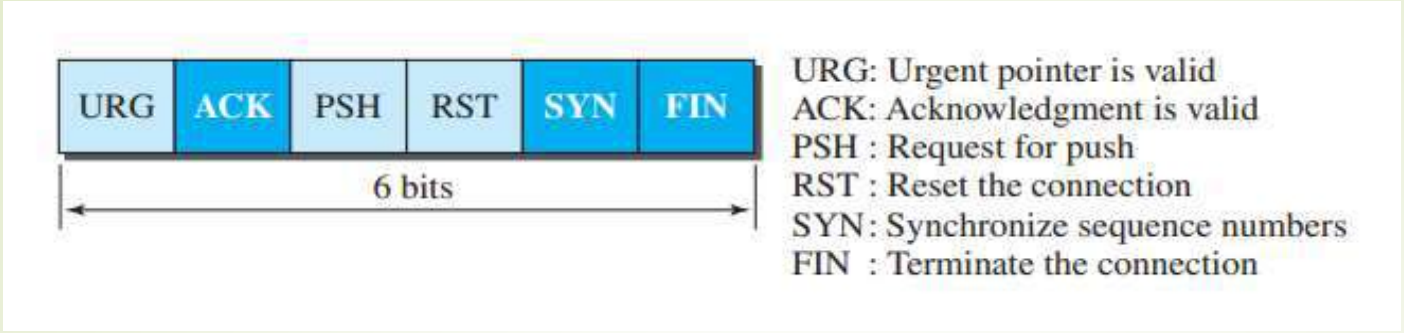


Figure: Control field

Window size. This field defines the window size of the sending TCP in bytes. Note that the length of this field is 16 bits, which means that the maximum size of the window is 65,535 bytes. This value is normally referred to as the receiving window (rwnd) and is determined by the receiver. The sender must obey the dictation of the receiver in this case.

Checksum. This 16-bit field contains the checksum. The calculation of the checksum for TCP follows the same procedure as the one described for UDP. However, the use of the checksum in the UDP datagram is optional, whereas the use of the checksum for TCP is mandatory.

Urgent pointer. This 16-bit field, which is valid only if the urgent flag is set, is used when the segment contains urgent data. It defines a value that must be added to the sequence number to obtain the number of the last urgent byte in the data section of the segment.

Options. There can be up to 40 bytes of optional information in the TCP header.

A TCP Connection

- All of the segments belonging to a message are then sent over this logical path. Using a single logical pathway for the entire message facilitates the acknowledgment process as well as retransmission of damaged or lost frames.
- The point is that a TCP connection is logical, not physical. TCP operates at a higher level. TCP uses the services of IP to deliver individual segments to the receiver, but it controls the connection itself. If a segment is lost or corrupted, it is retransmitted.
- Unlike TCP, IP is unaware of this retransmission. If a segment arrives out of order, TCP holds it until the missing segments arrive; IP is unaware of this reordering.
- In TCP, connection-oriented transmission requires three phases: connection establishment, data transfer, and connection termination.

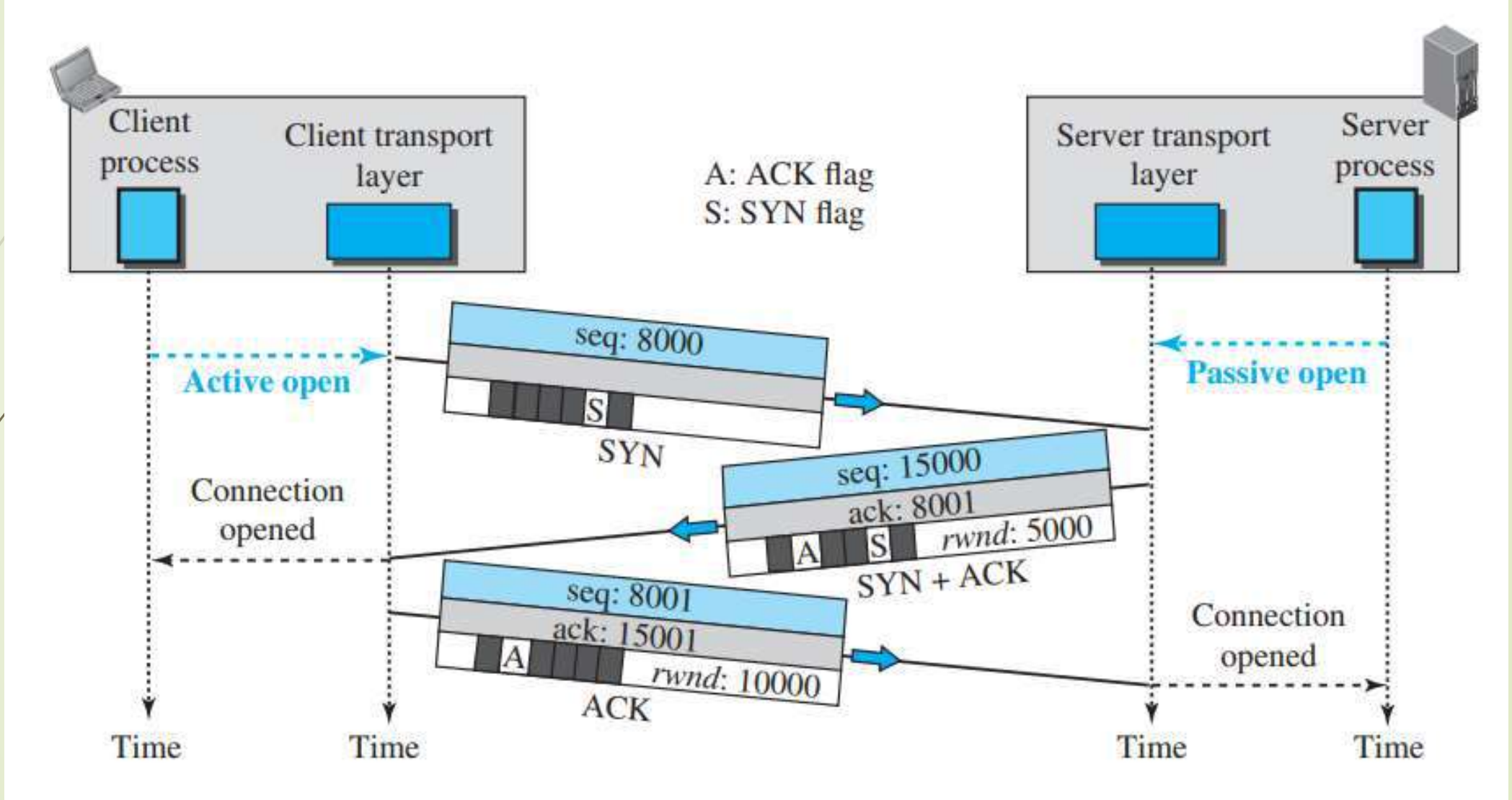
Connection Establishment

- TCP transmits data in full-duplex mode. When two TCPs in two machines are connected, they are able to send segments to each other simultaneously.
- This implies that each party must initialize communication and get approval from the other party before any data are transferred.

Three-Way Handshaking

- The connection establishment in TCP is called three-way handshaking.
- The process starts with the server. The server program tells its TCP that it is ready to accept a connection. This request is called a passive open.
- Although the server TCP is ready to accept a connection from any machine in the world, it cannot make the connection itself. The client program issues a request for an active open.
- A client that wishes to connect to an open server tells its TCP to connect to a particular server.

Figure Connection establishment using three-way handshaking



The three steps in this phase are as follows.

1. The client sends the first segment, a SYN segment, in which only the SYN flag is set.

- This segment is for synchronization of sequence numbers. The client in our example chooses a random number as the first sequence number and sends this number to the server.
- This sequence number is called the initial sequence number (ISN). Note that this segment does not contain an acknowledgment number. It does not define the window size either; a window size definition makes sense only when a segment includes an acknowledgment.
- Note that the SYN segment is a control segment and carries no data. However, it consumes one sequence number because it needs to be acknowledged. We can say that the SYN segment carries one imaginary byte.

2. The server sends the second segment, a SYN + ACK segment with two flag bits set as: SYN and ACK. This segment has a dual purpose.

- First, it is a SYN segment for communication in the other direction. The server uses this segment to initialize a sequence number for numbering the bytes sent from the server to the client.
- The server also acknowledges the receipt of the SYN segment from the client by setting the ACK flag and displaying the next sequence number it expects to receive from the client.
- Because the segment contains an acknowledgment, it also needs to define the receive window size, rwnd (to be used by the client), as we will see in the flow control section. Since this segment is playing the role of a SYN segment, it needs to be acknowledged. It, therefore, consumes one sequence number.

3. The client sends the third segment. This is just an ACK segment. It acknowledges the receipt of the second segment with the ACK flag and acknowledgment number field.

- Note that the ACK segment does not consume any sequence numbers if it does not carry data, but some implementations allow this third segment in the connection phase to carry the first chunk of data from the client.
- In this case, the segment consumes as many sequence numbers as the number of data bytes.

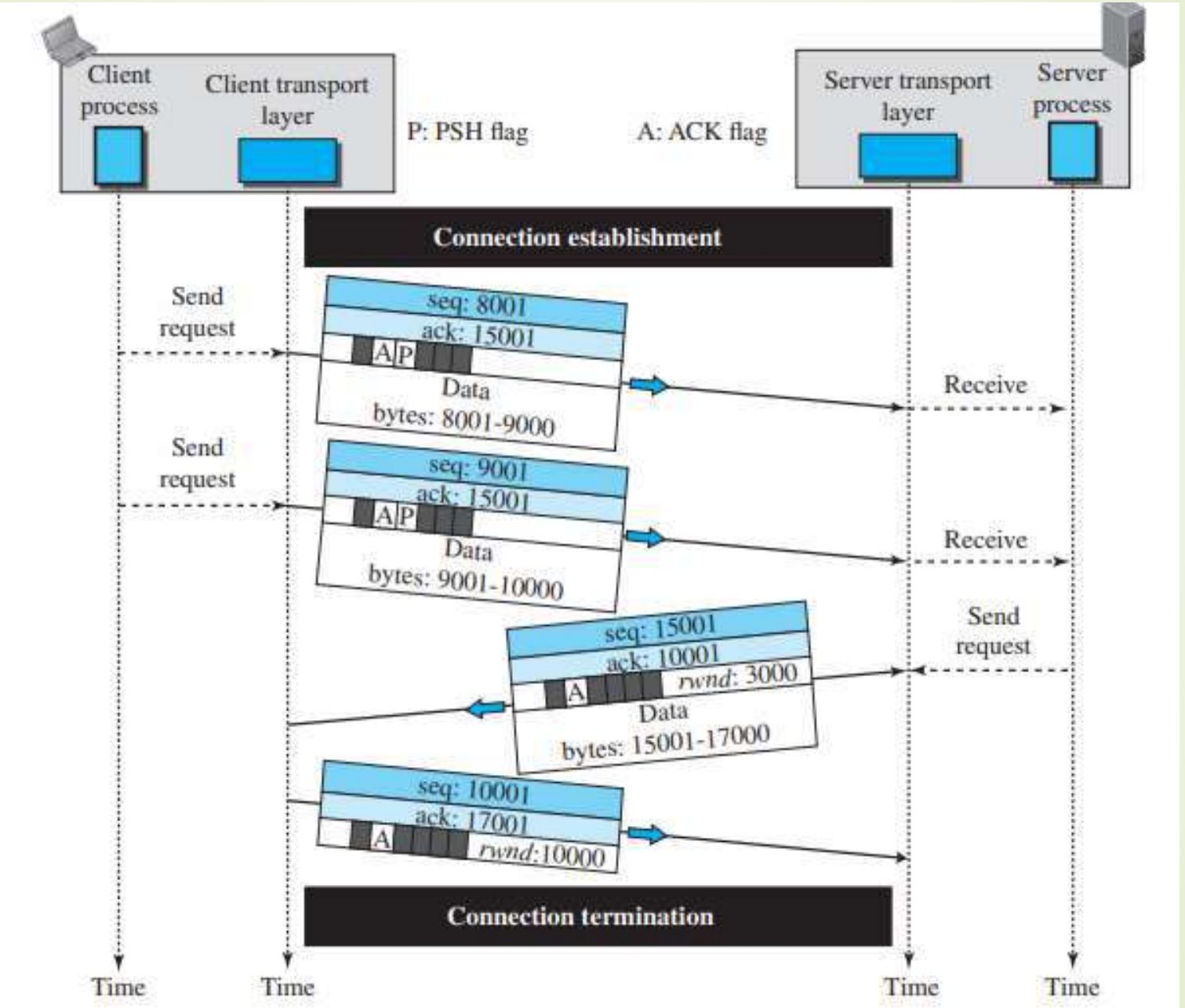
SYN Flooding Attack

- The connection establishment procedure in TCP is susceptible to a serious security problem called SYN flooding attack.
- This happens when one or more malicious attackers send a large number of SYN segments to a server pretending that each of them is coming from a different client by faking the source IP addresses in the datagrams.
- The server, assuming that the clients are issuing an active open, allocates the necessary resources, such as creating transfer control block (TCB) tables and setting timers. The TCP server then sends the SYN + ACK segments to the fake clients, which are lost.
- When the server waits for the third leg of the handshaking process, however, resources are allocated without being used.
- If, during this short period of time, the number of SYN segments is large, the server eventually runs out of resources and may be unable to accept connection requests from valid clients.

Data Transfer

- After connection is established, bidirectional data transfer can take place. The client and server can send data and acknowledgments in both directions.
- In this example, after a connection is established, the client sends 2,000 bytes of data in two segments.
- The server then sends 2,000 bytes in one segment. The client sends one more segment.
- The first three segments carry both data and acknowledgment, but the last segment carries only an acknowledgment because there is no more data to be sent.
- Note the values of the sequence and acknowledgment numbers. The data segments sent by the client have the PSH (push) flag set so that the server TCP knows to deliver data to the server process as soon as they are received.

Figure: Data transfer



Pushing Data

- We saw that the sending TCP uses a buffer to store the stream of data coming from the sending application program.
- The sending TCP can select the segment size. The receiving TCP also buffers the data when they arrive and delivers them to the application program when the application program is ready or when it is convenient for the receiving TCP. This type of flexibility increases the efficiency of TCP.
- The application program at the sender can request a push operation. This means that the sending TCP must not wait for the window to be filled. It must create a segment and send it immediately.
- The sending TCP must also set the push bit (PSH) to let the receiving TCP know that the segment includes data that must be delivered to the receiving application program as soon as possible and not to wait for more data to come.
- This means to change the byte-oriented TCP to a chunk-oriented TCP, but TCP can choose whether or not to use this feature.

Connection Termination

- Either of the two parties involved in exchanging data (client or server) can close the connection, although it is usually initiated by the client.
- Most implementations today allow two options for connection termination: three-way handshaking and four-way handshaking with a half-close option.

Three-Way Handshaking

- Most implementations today allow three-way handshaking for connection termination,

1. In this situation, the client TCP, after receiving a close command from the client process, sends the first segment, a FIN segment in which the FIN flag is set.

- Note that a FIN segment can include the last chunk of data sent by the client or it can be just a control segment as shown in the figure. If it is only a control segment, it consumes only one sequence number because it needs to be acknowledged.

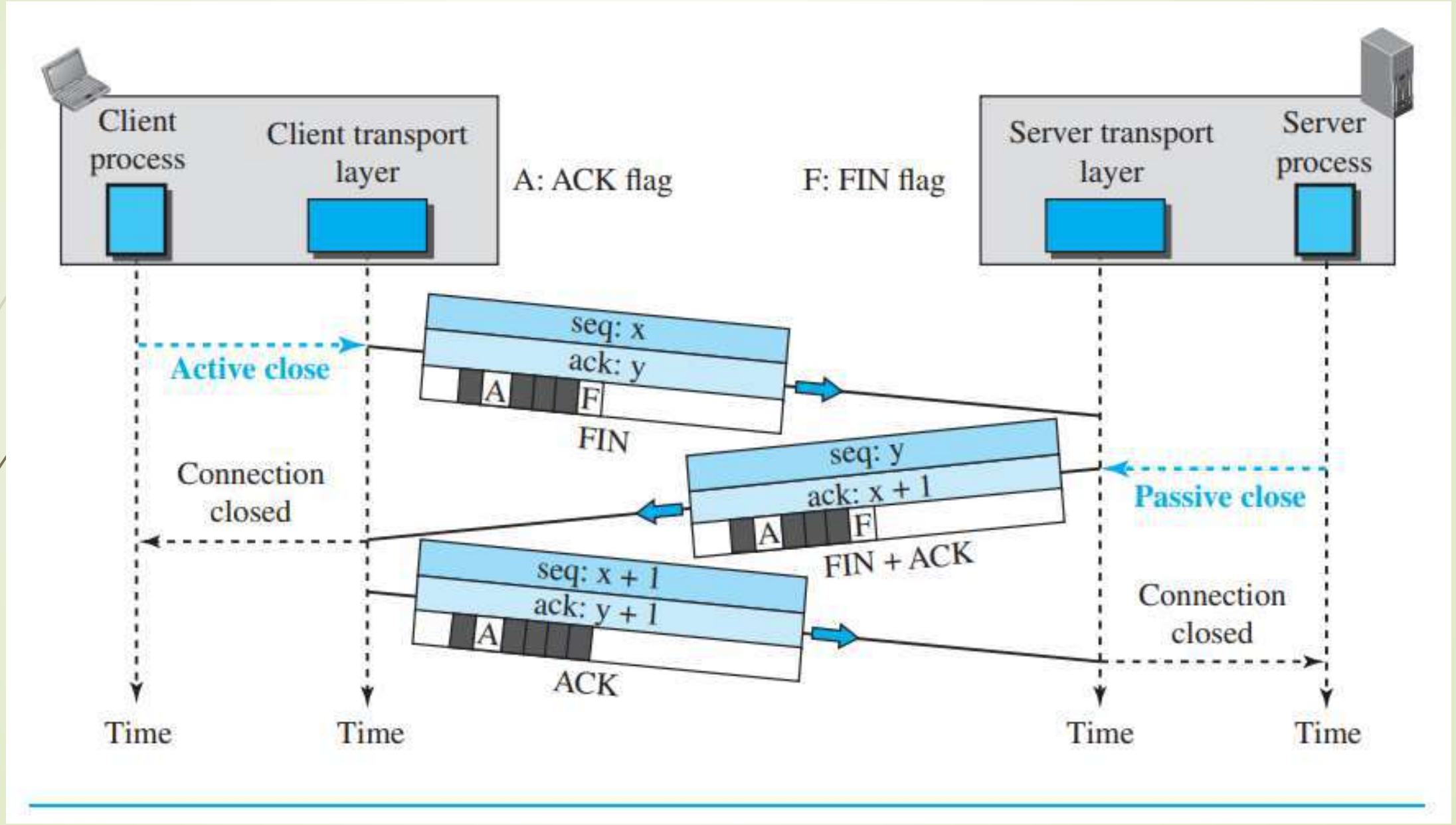
2. The server TCP, after receiving the FIN segment, informs its process of the situation and sends the second segment, a FIN + ACK segment, to confirm the receipt of the FIN segment from the client and at the same time to announce the closing of the connection in the other direction.

- This segment can also contain the last chunk of data from the server. If it does not carry data, it consumes only one sequence number because it needs to be acknowledged.

3. The client TCP sends the last segment, an ACK segment, to confirm the receipt of the FIN segment from the TCP server.

- This segment contains the acknowledgment number, which is one plus the sequence number received in the FIN segment from the server. This segment cannot carry data and consumes no sequence numbers.

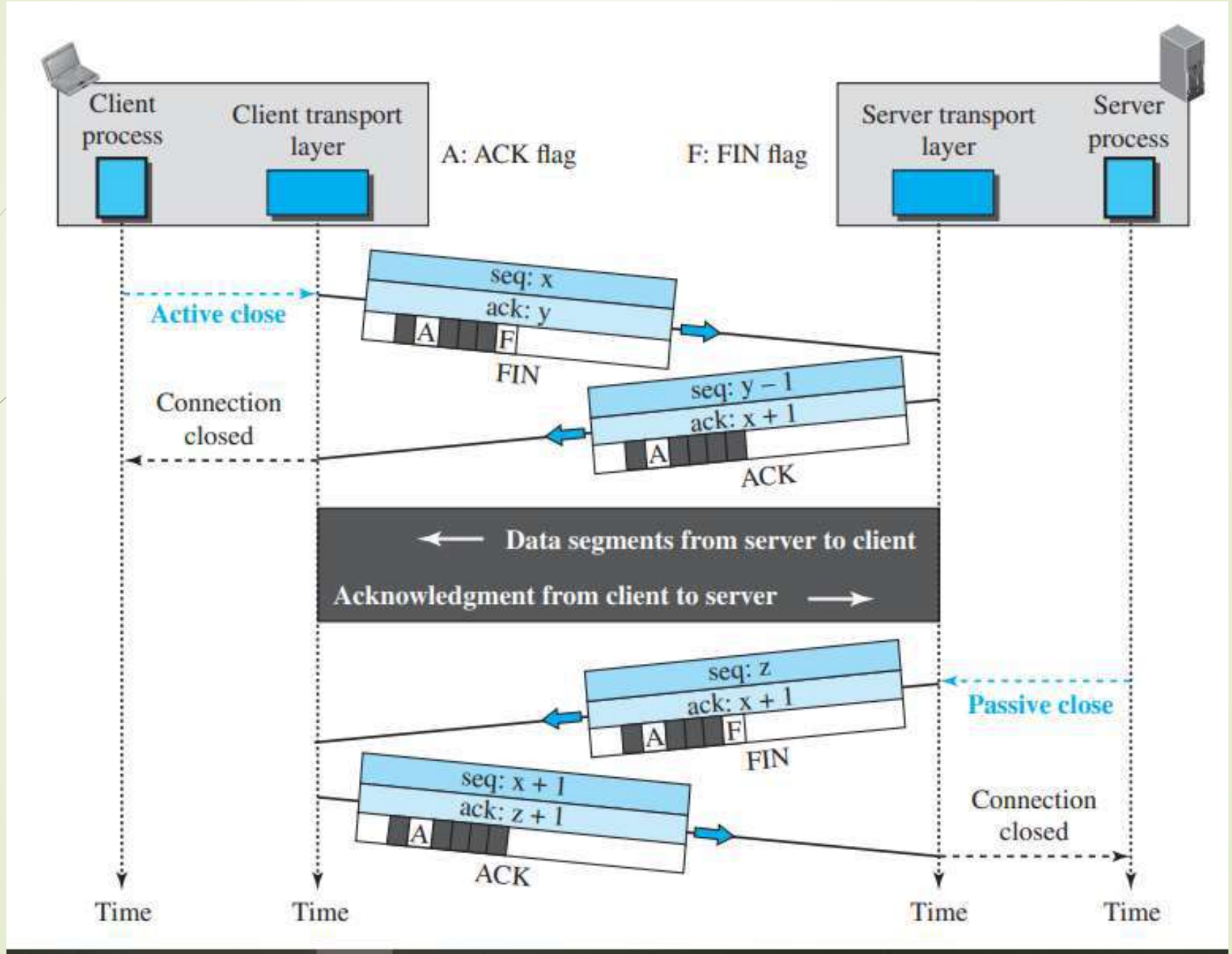
Figure: Connection termination using three-way handshaking



Half-Close

- In TCP, one end can stop sending data while still receiving data. This is called a half-close.
- Either the server or the client can issue a half-close request. It can occur when the server needs all the data before processing can begin.
- However, the server-to-client direction must remain open to return the sorted data.
- The server, after receiving the data, still needs time for sorting; its outbound direction must remain open.
- The data transfer from the client to the server stops. The client half-closes the connection by sending a FIN segment.
- The server accepts the half-close by sending the ACK segment. The server, however, can still send data.
- When the server has sent all of the processed data, it sends a FIN segment, which is acknowledged by an ACK from the client.
- After half-closing the connection, data can travel from the server to the client and acknowledgments can travel from the client to the server. The client cannot send any more data to the server.

Figure: Half-close



Windows in TCP

- TCP uses two windows (send window and receive window) for each direction of data transfer, which means four windows for a bidirectional communication.

Send Window

The window size is 100 bytes, the send window size is dictated by the receiver (flow control) and the congestion in the underlying network (congestion control).

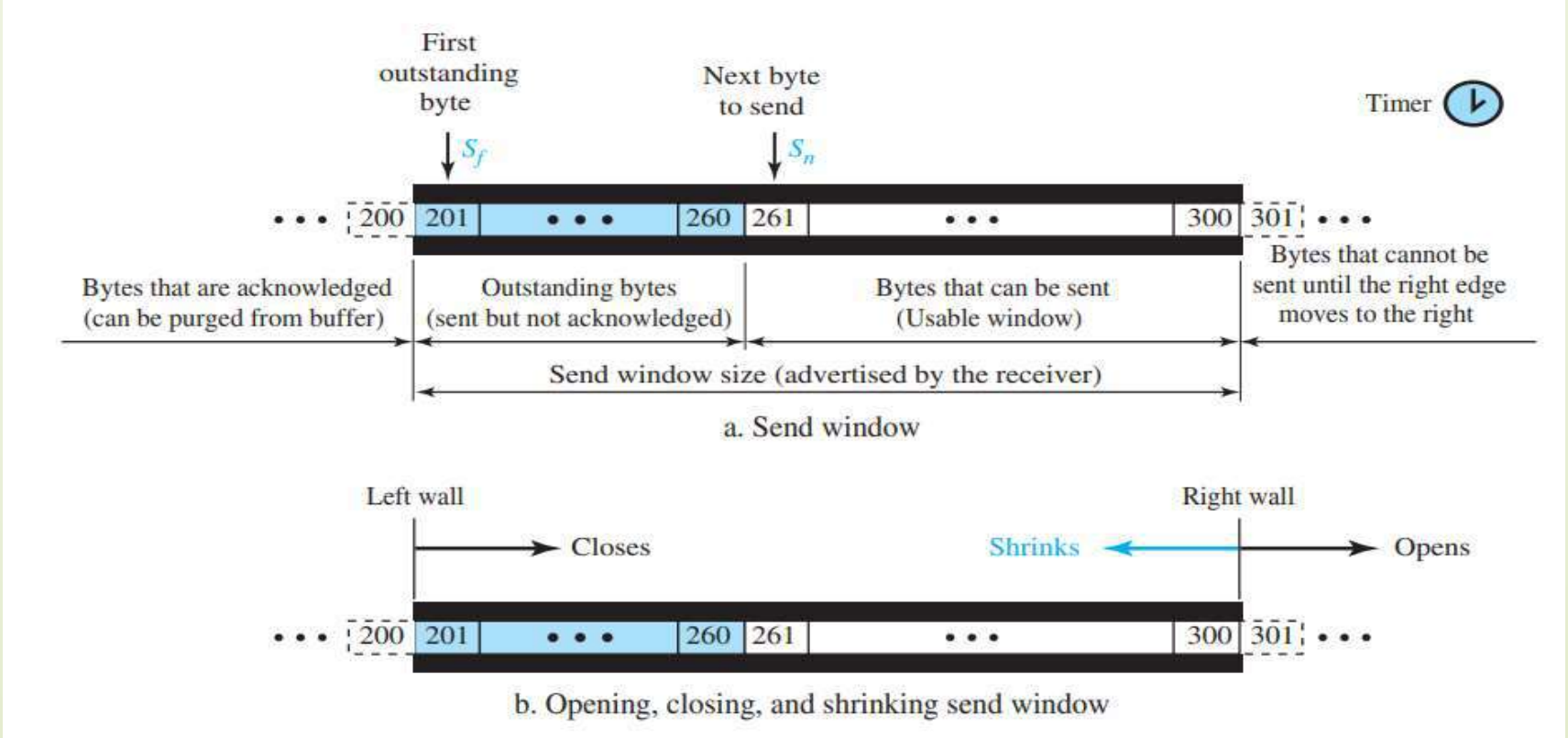


Figure: Send window in TCP

The send window in TCP is similar to the one used with the Selective-Repeat protocol, but with some differences:

1. One difference is the nature of entities related to the window. The window size in SR is the number of packets, but the window size in TCP is the number of bytes. Although actual transmission in TCP occurs segment by segment, the variables that control the window are expressed in bytes.
2. The second difference is that, in some implementations, TCP can store data received from the process and send them later, but we assume that the sending TCP is capable of sending segments of data as soon as it receives them from its process.
3. Another difference is the number of timers. The theoretical Selective-Repeat protocol may use several timers for each packet sent, but as mentioned before, the TCP protocol uses only one timer.

Receive Window

The window size is 100 bytes. The figure also shows how the receive window opens and closes; in practice, the window should never shrink.

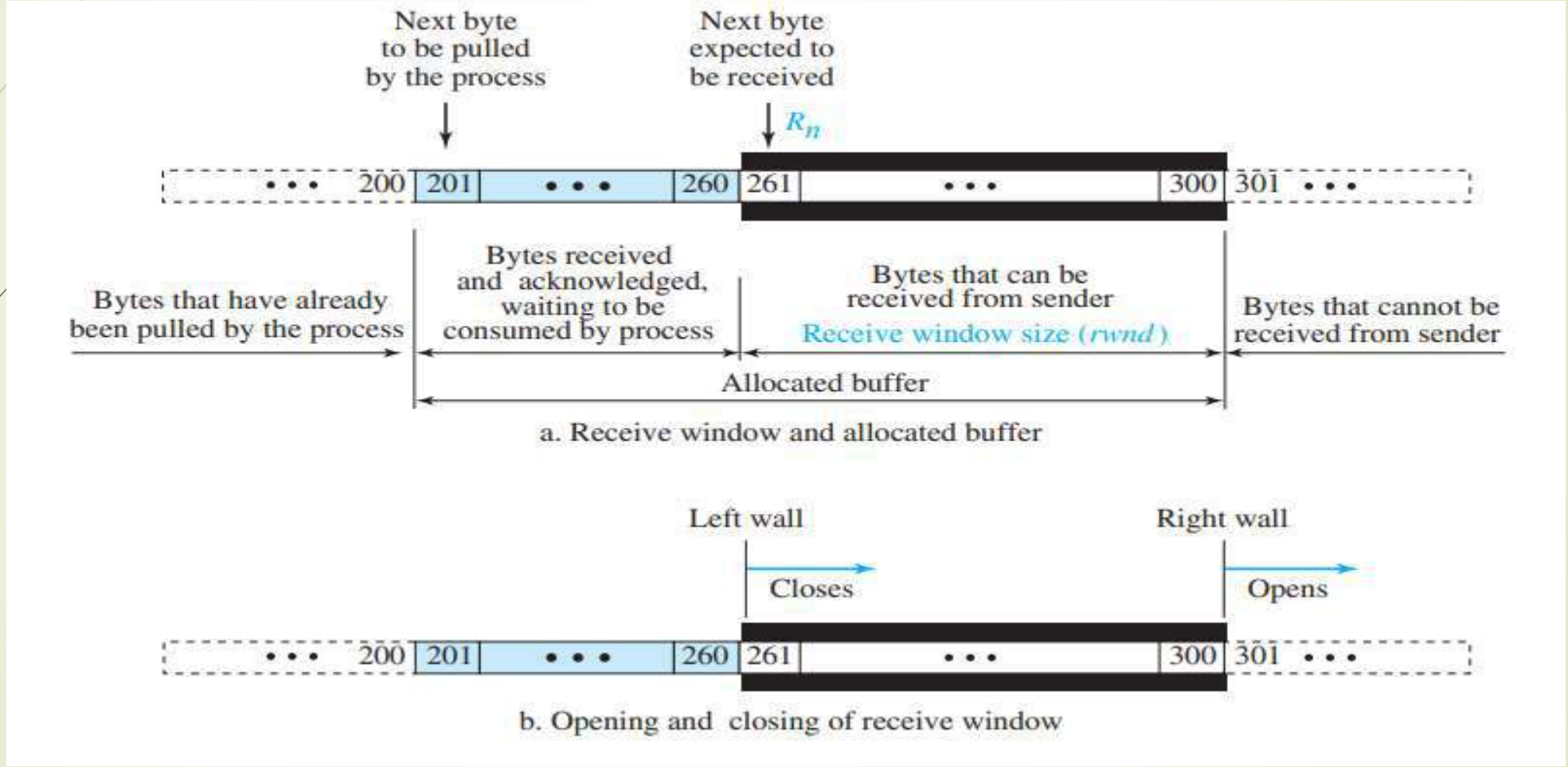


Figure: Receive window in TCP

There are two differences between the receive window in TCP and the one we used for SR.

1. The first difference is that TCP allows the receiving process to pull data at its own pace. This means that part of the allocated buffer at the receiver may be occupied by bytes that have been received and acknowledged, but are waiting to be pulled by the receiving process. The receive window size is then always smaller than or equal to the buffer size, as shown in Figure 24.18. The receive window size determines the number of bytes that the receive window can accept from the sender before being overwhelmed (flow control). In other words, the receive window size, normally called rwnd, can be determined as:

   rwnd 5 buffer size 2 number of waiting bytes to be pulled

2. The second difference is the way acknowledgments are used in the TCP protocol. Remember that an acknowledgement in SR is selective, defining the uncorrupted packets that have been received. The major acknowledgment mechanism in TCP is a cumulative acknowledgment announcing the next expected byte to receive (in this way TCP looks like GBN, discussed earlier). The new version of TCP, however, uses both cumulative and selective acknowledgments; we will discuss these options on the book website.

Error Control

- TCP is a reliable transport-layer protocol. This means that an application program that delivers a stream of data to TCP relies on TCP to deliver the entire stream to the application program on the other end in order, without error, and without any part lost or duplicated.
- TCP provides reliability using error control. Error control includes mechanisms for detecting and resending corrupted segments, resending lost segments, storing out-of-order segments until missing segments arrive, and detecting and discarding duplicate segments. Error control in TCP is achieved through the use of three simple tools:
  - checksum,
  - acknowledgment, and
  - time-out.

Checksum

- Each segment includes a checksum field, which is used to check for a corrupted segment. If a segment is corrupted, as detected by an invalid checksum, the segment is discarded by the destination TCP and is considered as lost. TCP uses a 16-bit checksum that is mandatory in every segment.

Acknowledgment

- TCP uses acknowledgments to confirm the receipt of data segments. Control segments that carry no data, but consume a sequence number, are also acknowledged. ACK segments are never acknowledged.
- Acknowledgment Type In the past, TCP used only one type of acknowledgment: cumulative acknowledgment. Today, some TCP implementations also use selective acknowledgment.

- **Cumulative Acknowledgment (ACK):** TCP was originally designed to acknowledge receipt of segments cumulatively. The receiver advertises the next byte it expects to receive, ignoring all segments received and stored out of order. This is sometimes referred to as positive cumulative acknowledgment, or ACK. The word positive indicates that no feedback is provided for discarded, lost, or duplicate segments. The 32-bit ACK field in the TCP header is used for cumulative acknowledgments, and its value is valid only when the ACK flag bit is set to 1.
- **Selective Acknowledgment (SACK):** More and more implementations are adding another type of acknowledgment called selective acknowledgment, or SACK. A SACK does not replace an ACK, but reports additional information to the sender. A SACK reports a block of bytes that is out of order, and also a block of bytes that is duplicated, i.e., received more than once. However, since there is no provision in the TCP header for adding this type of information, SACK is implemented as an option at the end of the TCP header.

Generating Acknowledgments

When does a receiver generate acknowledgments? During the evolution of TCP, several rules have been defined and used by several implementations. We give the most common rules here. The order of a rule does not necessarily define its importance.

1. When end A sends a data segment to end B, it must include (piggyback) an acknowledgment that gives the next sequence number it expects to receive. This rule decreases the number of segments needed and therefore reduces traffic.
2. When the receiver has no data to send and it receives an in-order segment (with expected sequence number) and the previous segment has already been acknowledged, the receiver delays sending an ACK segment until another segment arrives or until a period of time (normally 500 ms) has passed. In other words, the receiver needs to delay sending an ACK segment if there is only one outstanding in-order segment. This rule reduces ACK segments.
3. When a segment arrives with a sequence number that is expected by the receiver, and the previous in-order segment has not been acknowledged, the receiver immediately sends an ACK segment. In other words, there should not be more than two in-order unacknowledged segments at any time. This prevents the unnecessary retransmission of segments that may create congestion in the network.

4. When a segment arrives with an out-of-order sequence number that is higher than expected, the receiver immediately sends an ACK segment announcing the sequence number of the next expected segment. This leads to the fast retransmission of missing segments (discussed later).

5. When a missing segment arrives, the receiver sends an ACK segment to announce the next sequence number expected. This informs the receiver that segments reported missing have been received.

6. If a duplicate segment arrives, the receiver discards the segment, but immediately sends an acknowledgment indicating the next in-order segment expected. This solves some problems when an ACK segment itself is lost.

Retransmission

- The heart of the error control mechanism is the retransmission of segments. When a segment is sent, it is stored in a queue until it is acknowledged. When the retransmission timer expires or when the sender receives three duplicate ACKs for the first segment in the queue, that segment is retransmitted.

Retransmission after RTO

- The sending TCP maintains one retransmission time-out (RTO) for each connection. When the timer matures, i.e. times out, TCP resends the segment in the front of the queue (the segment with the smallest sequence number) and restarts the timer. Note that again we assume Sf < Sn. The value of RTO is dynamic in TCP and is updated based on the round-trip time (RTT) of segments. RTT is the time needed for a segment to reach a destination and for an acknowledgment to be received.

Retransmission after Three Duplicate ACK Segments

- The previous rule about retransmission of a segment is sufficient if the value of RTO is not large. To expedite service throughout the Internet by allowing senders to retransmit without waiting for a time out, most implementations today follow the three duplicate ACKs rule and retransmit the missing segment immediately. This feature is called fast retransmission. In this version, if three duplicate acknowledgments (i.e., an original ACK plus three exactly identical copies) arrive for a segment, the next segment is retransmitted without waiting for the time-out.

Out-of-Order Segments

- TCP implementations today do not discard out-of-order segments. They store them temporarily and flag them as out-of-order segments until the missing segments arrive.
- Note, however, that out-of-order segments are never delivered to the process. TCP guarantees that data are delivered to the process in order.

TCP Congestion Control

- TCP uses different policies to handle the congestion in the network.

Congestion Window

- The size of the send window is controlled by the receiver using the value of rwnd, which is advertised in each segment traveling in the opposite direction.
- The use of this strategy guarantees that the receive window is never overflowed with the received bytes (no end congestion). This, however, does not mean that the intermediate buffers, buffers in the routers, do not become congested.
- A router may receive data from more than one sender. No matter how large the buffers of a router may be, it may be overwhelmed with data, which results in dropping some segments sent by a specific TCP sender. In other words, there is no congestion at the other end, but there may be congestion in the middle.
- TCP needs to worry about congestion in the middle because many segments lost may seriously affect the error control. More segment loss means resending the same segments again, resulting in worsening the congestion, and finally the collapse of the communication.

- TCP is an end-to-end protocol that uses the service of IP. The congestion in the router is in the IP territory and should be taken care of by IP.
- TCP cannot ignore the congestion in the network; it cannot aggressively send segments to the network.
- TCP cannot be very conservative, either, sending a small number of segments in each time interval, because this means not utilizing the available bandwidth of the network. TCP needs to define policies that accelerate the data transmission when there is no congestion and decelerate the transmission when congestion is detected.
- To control the number of segments to transmit, TCP uses another variable called a congestion window, cwnd, whose size is controlled by the congestion situation in the network.
- The cwnd variable and the rwnd variable together define the size of the send window in TCP. The first is related to the congestion in the middle (network); the second is related to the congestion at the end. The actual size of the window is the minimum of these two.

Actual window size 5 minimum (rwnd, cwnd)

Congestion Detection

- The TCP sender uses the occurrence of two events as signs of congestion in the network: time-out and receiving three duplicate ACKs. The first is the time-out.
- If a TCP sender does not receive an ACK for a segment or a group of segments before the time-out occurs, it assumes that the corresponding segment or segments are lost and the loss is due to congestion.
- Recall that when a TCP receiver sends a duplicate ACK, it is the sign that a segment has been delayed, but sending three duplicate ACKs is the sign of a missing segment, which can be due to congestion in the network.
- When a receiver sends three duplicate ACKs, it means that one segment is missing, but three segments have been received. The network is either slightly congested or has recovered from the congestion.
- A very interesting point in TCP congestion is that the TCP sender uses only one feedback from the other end to detect congestion: ACKs.

Congestion Policies

TCP's general policy for handling congestion is based on three algorithms:

- slow start,
- congestion avoidance, and
- fast recovery.

Slow Start: Exponential Increase

- The slow-start algorithm is based on the idea that the size of the congestion window (cwnd) starts with one maximum segment size (MSS), but it increases one MSS each time an acknowledgment arrives.
- We assume that rwnd is much larger than cwnd, so that the sender window size always equals cwnd. We also assume that each segment is of the same size and carries MSS bytes.
- The sender starts with cwnd = 1. This means that the sender can send only one segment.
- After the first ACK arrives, the acknowledged segment is purged from the window, which means there is now one empty segment slot in the window.

- The size of the congestion window is also increased by 1 because the arrival of the acknowledgment is a good sign that there is no congestion in the network.
- The size of the window is now 2. After sending two segments and receiving two individual acknowledgments for them, the size of the congestion window now becomes 4, and so on.
- In other words, the size of the congestion window in this algorithm is a function of the number of ACKs arrived and can be determined as follows.

If an ACK arrives, cwnd = cwnd + 1.

- If we look at the size of the cwnd in terms of round-trip times (RTTs), we find that the growth rate is exponential in terms of each round trip time, which is a very aggressive approach:

Start	→	$cwnd = 1 \rightarrow 2^0$
After 1 RTT	→	$cwnd = cwnd + 1 = 1 + 1 = 2 \rightarrow 2^1$
After 2 RTT	→	$cwnd = cwnd + 2 = 2 + 2 = 4 \rightarrow 2^2$
After 3 RTT	→	$cwnd = cwnd + 4 = 4 + 4 = 8 \rightarrow 2^3$

- A slow start cannot continue indefinitely. There must be a threshold to stop this phase. The sender keeps track of a variable named ssthresh (slow-start threshold). When the size of the window in bytes reaches this threshold, slow start stops and the next phase starts.

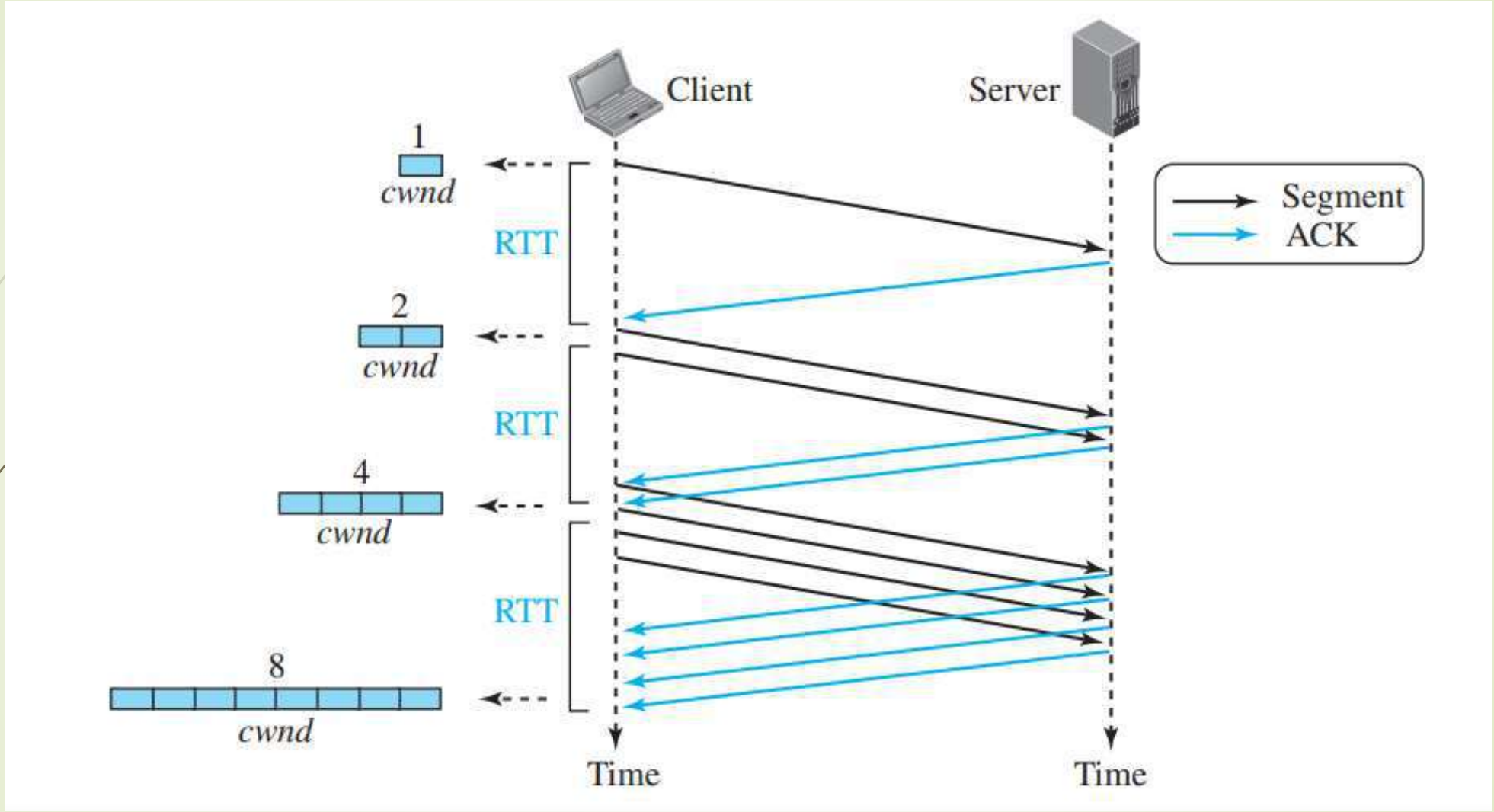


Figure: Slow start, exponential increase

Congestion Avoidance: Additive Increase

- If we continue with the slow-start algorithm, the size of the congestion window increases exponentially. To avoid congestion before it happens, we must slow down this exponential growth.
- TCP defines another algorithm called congestion avoidance, which increases the cwnd additively instead of exponentially.
- When the size of the congestion window reaches the slow-start threshold in the case where cwnd = i, the slow-start phase stops and the additive phase begins.
- In this algorithm, each time the whole “window” of segments is acknowledged, the size of the congestion window is increased by one. A window is the number of segments transmitted during RTT.
- The sender starts with cwnd = 4. This means that the sender can send only four segments. After four ACKs arrive, the acknowledged segments are purged from the window, which means there is now one extra empty segment slot in the window.

- The size of the congestion window is also increased by 1. The size of window is now 5.
- After sending five segments and receiving five acknowledgments for them, the size of the congestion window now becomes 6, and so on. In other words, the size of the congestion window in this algorithm is also a function of the number of ACKs that have arrived and can be determined as follows:

If an ACK arrives, cwnd 5 cwnd 1 (1/cwnd).

- The size of the window increases only 1/cwnd portion of MSS (in bytes). In other words, all segments in the previous window should be acknowledged to increase the window 1 MSS bytes.
- If we look at the size of the cwnd in terms of round-trip times (RTTs), we find that the growth rate is linear in terms of each round-trip time, which is much more conservative than the slow-start approach.

Start	→	$cwnd = i$
After 1 RTT	→	$cwnd = i + 1$
After 2 RTT	→	$cwnd = i + 2$
After 3 RTT	→	$cwnd = i + 3$

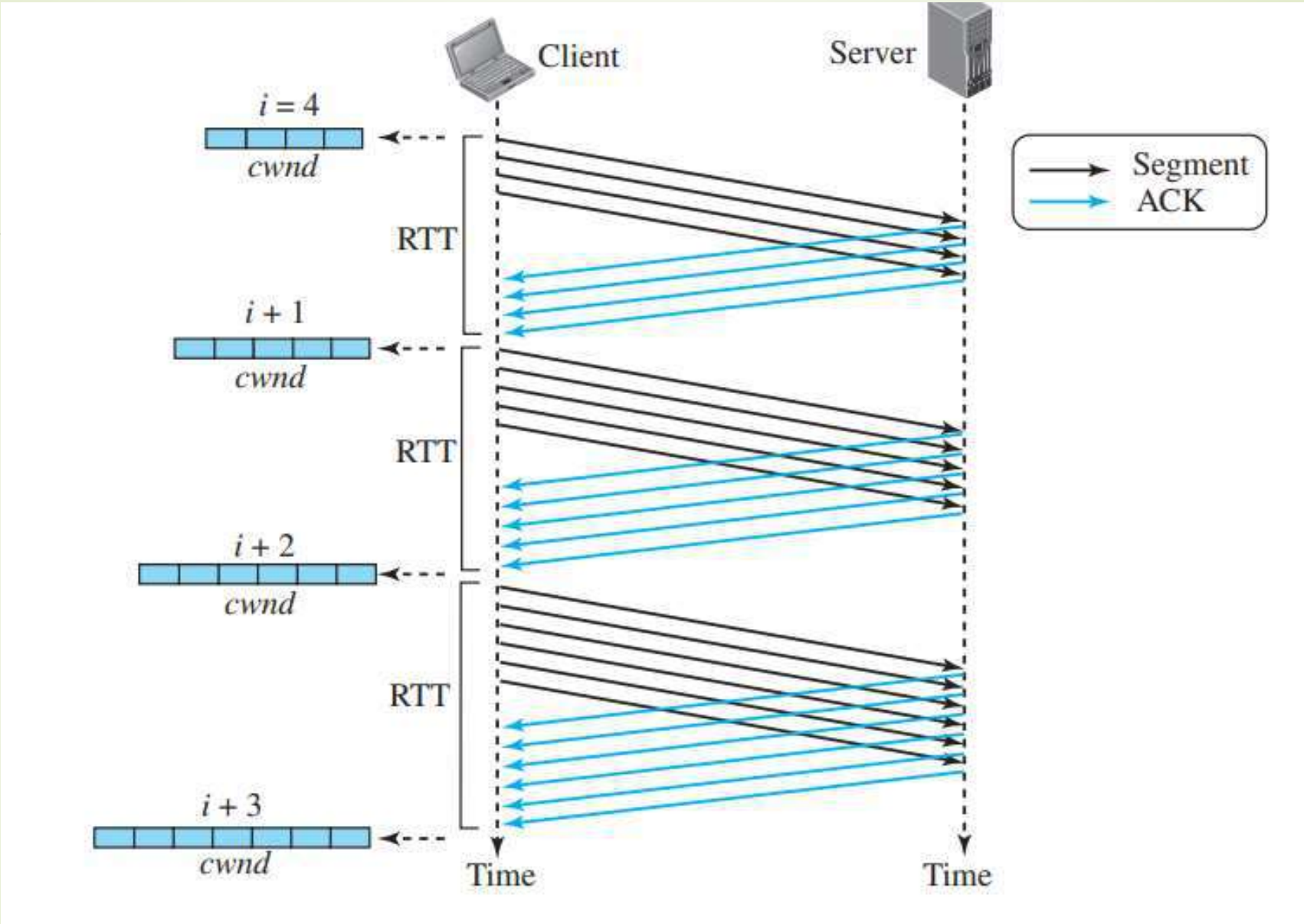


Figure: Congestion avoidance, additive increase

Fast Recovery

- The fast-recovery algorithm is optional in TCP.
- It starts when three duplicate ACKs arrive, which is interpreted as light congestion in the network.
- Like congestion avoidance, this algorithm is also an additive increase, but it increases the size of the congestion window when a duplicate ACK arrives (after the three duplicate ACKs that trigger the use of this algorithm).
- We can say

If a duplicate ACK arrives, cwnd 5 cwnd 1 (1 / cwnd).

Policy Transition

- We discussed three congestion policies in TCP. Now the question is when each of these policies is used and when TCP moves from one policy to another.
- To answer these questions, we need to refer to three versions of TCP:
  - Taho TCP,
  - Reno TCP, and
  - New Reno TCP.

Taho TCP,

- The early TCP, known as Taho TCP, used only two different algorithms in their congestion policy: slow start and congestion avoidance.
- We use Below Figure to show the FSM for this version of TCP. However, we need to mention that we have deleted some small trivial actions, such as incrementing and resetting the number of duplicate ACKs, to make the FSM less crowded and simpler.

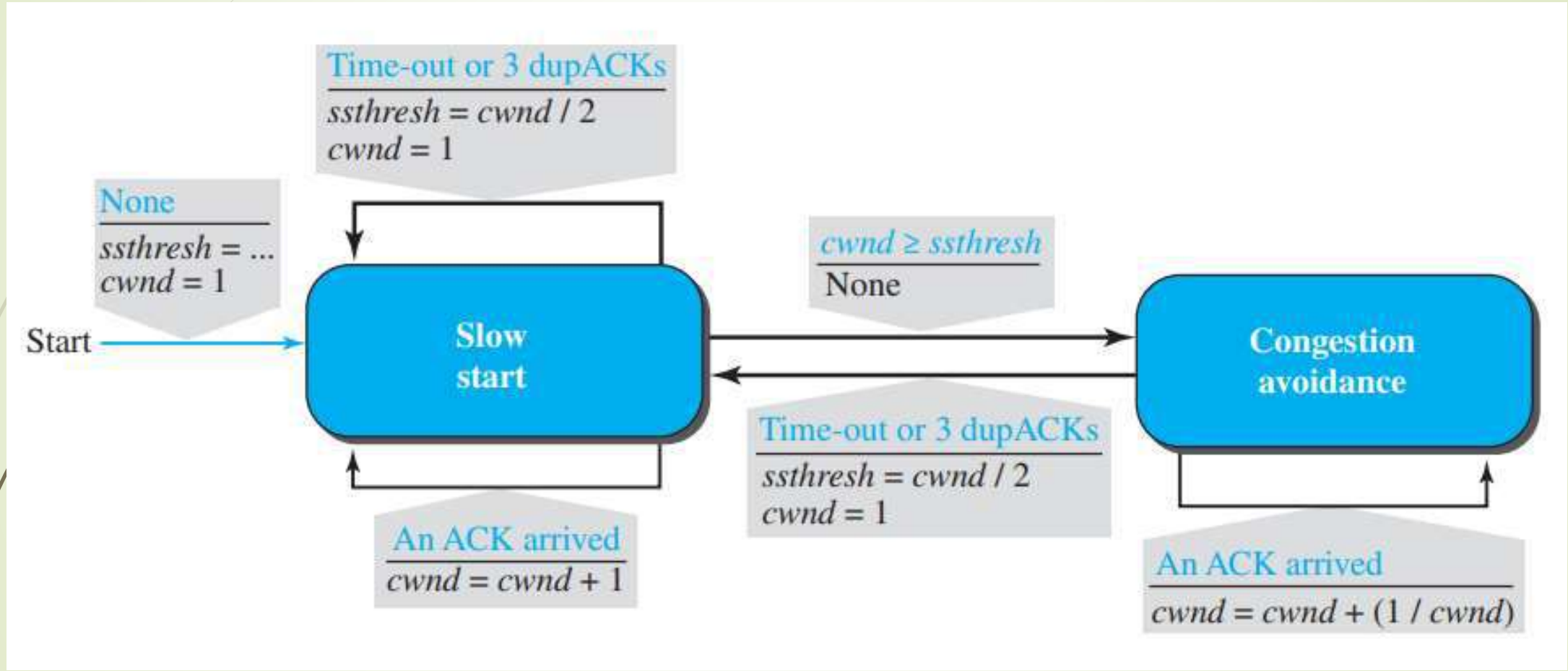


Figure: FSM for Taho TCP

- Taho TCP treats the two signs used for congestion detection, time-out and three duplicate ACKs, in the same way.
- In this version, when the connection is established, TCP starts the slow-start algorithm and sets the ssthresh variable to a pre-agreed value (normally a multiple of MSS) and the cwnd to 1 MSS.
- In this state, as we said before, each time an ACK arrives, the size of the congestion window is incremented by 1.
- We know that this policy is very aggressive and exponentially increases the size of the window, which may result in congestion.
- If congestion is detected (occurrence of time-out or arrival of three duplicate ACKs), TCP immediately interrupts this aggressive growth and restarts a new slow start algorithm by limiting the threshold to half of the current cwnd and resetting the congestion window to 1.
- If no congestion is detected while reaching the threshold, TCP learns that the ceiling of its ambition is reached; it should not continue at this speed. It moves to the congestion avoidance state and continues in that state.
- In the congestion-avoidance state, the size of the congestion window is increased by 1 each time a number of ACKs equal to the current size of the window has been received.

Reno TCP

- A newer version of TCP, called Reno TCP, added a new state to the congestion-control FSM, called the fast-recovery state. This version treated the two signals of congestion, time-out and the arrival of three duplicate ACKs, differently.
- In this version, if a time-out occurs, TCP moves to the slow-start state (or starts a new round if it is already in this state); on the other hand, if three duplicate ACKs arrive, TCP moves to the fast-recovery state and remains there as long as more duplicate ACKs arrive.
- The fast-recovery state is a state somewhere between the slow-start and the congestion-avoidance states. It behaves like the slow start, in which the cwnd grows exponentially, but the cwnd starts with the value of ssthresh plus 3 MSS (instead of 1).
- When TCP enters the fast-recovery state, three major events may occur. If duplicate ACKs continue to arrive, TCP stays in this state, but the cwnd grows exponentially.
- If a time-out occurs, TCP assumes that there is real congestion in the network and moves to the slow-start state. If a new (nonduplicate) ACK arrives, TCP moves to the congestion-avoidance state, but deflates the size of the cwnd to the ssthresh value, as though the three duplicate ACKs have not occurred, and transition is from the slow-start state to the congestion-avoidance state.

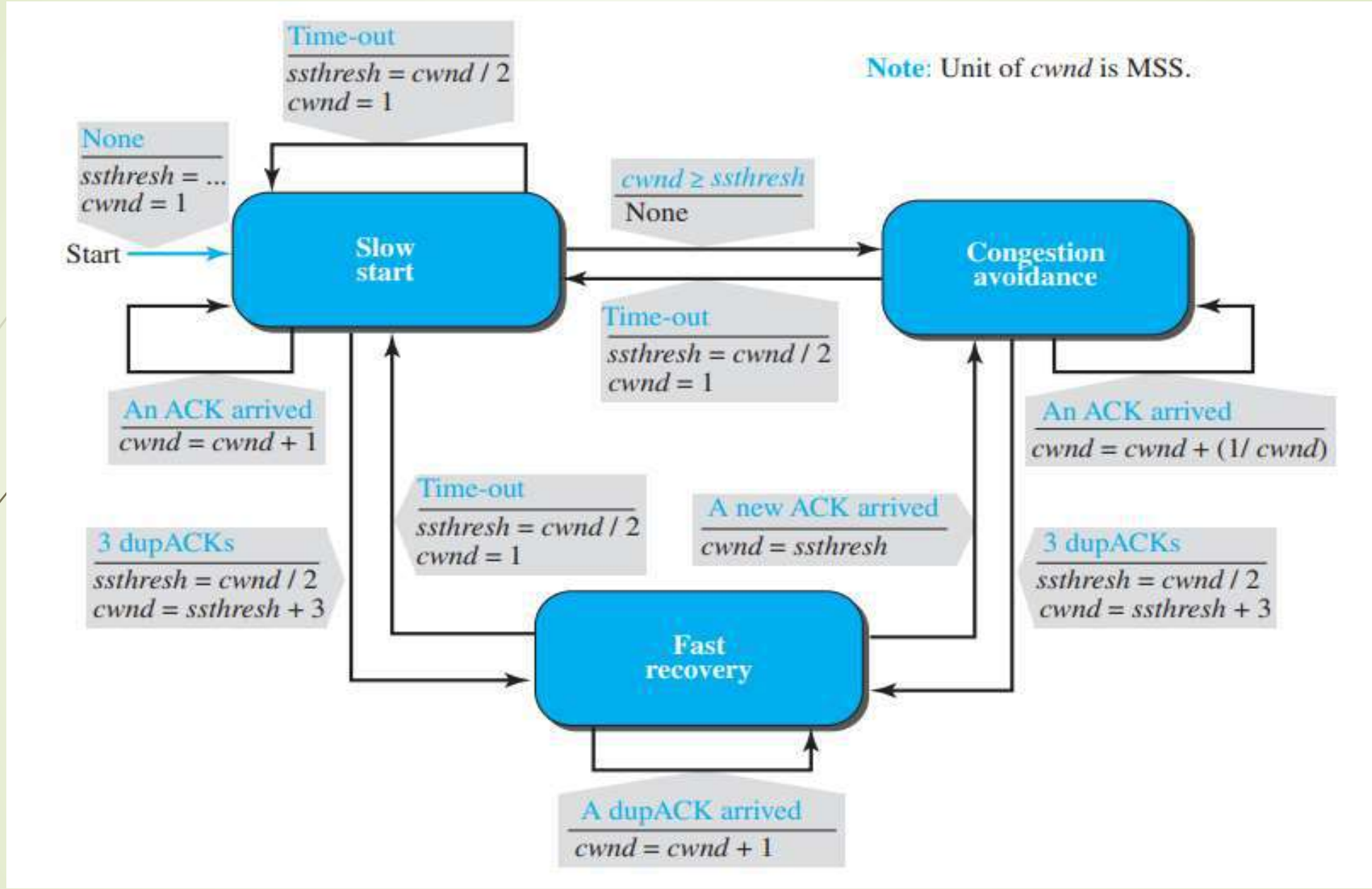


Figure: FSM for Reno TCP

NewReno TCP

- A later version of TCP, called NewReno TCP, made an extra optimization on the Reno TCP.
- In this version, TCP checks to see if more than one segment is lost in the current window when three duplicate ACKs arrive.
- When TCP receives three duplicate ACKs, it retransmits the lost segment until a new ACK (not duplicate) arrives. If the new ACK defines the end of the window when the congestion was detected, TCP is certain that only one segment was lost.
- However, if the ACK number defines a position between the retransmitted segment and the end of the window, it is possible that the segment defined by the ACK is also lost.
- NewReno TCP retransmits this segment to avoid receiving more and more duplicate ACKs for it.

Thank You