DATA COMMUNICATIONS & NETWORKING

B.SC (COMPUTER SCIENCE) – III YEAR

SMCS52

By
Dr.N.N.Krishna Veni,
Assistant Professor,
Department of Computer Science,
Holy Cross Home Science College,
Thoothukudi

UNIT - I
INTRODUCTION

1.1 DATA COMMUNICATIONS

- 1.1.1 Components
- 1.1.2 Data Representation
- 1.1.3 Data Flow

1.2 NETWORKS

- 1.2.1 Distributed Processing
- 1.2.2 Network Criteria
- 1.2.3 Physical Structures
- 1.2.4 Network Models

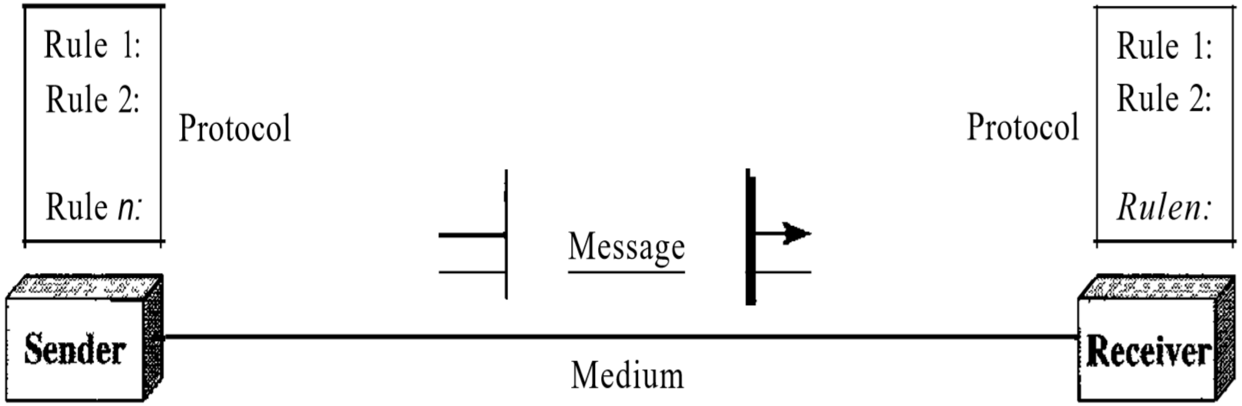
1.1 DATA COMMUNICATIONS

- Data communications are the exchange of data between two devices via some form of transmission medium such as a wire cable
- Four fundamental characteristics
  - **Delivery:** The system must deliver data to the correct destination.
  - **Accuracy:** The system must deliver the data accurately. Data that have been altered in transmission and left uncorrected are unusable.
  - **Timeliness:** The system must deliver data in a timely manner. Data delivered late are useless.
  - **Jitter:** Jitter refers to the variation in the packet arrival time. It is the uneven delay in the delivery of audio or video packets.

COMPONENTS

A data communications system has five components

- **Message:** The message is the information/data to be communicated
- **Sender:** The sender is the device that sends the data message.
- **Receiver:** The receiver is the device that receives the message.
- **Transmission medium:** The physical path by which a message travels from sender to receiver.
- **Protocol:** A protocol is a set of rules or agreement between the communicating devices.



DATA REPRESENTATION

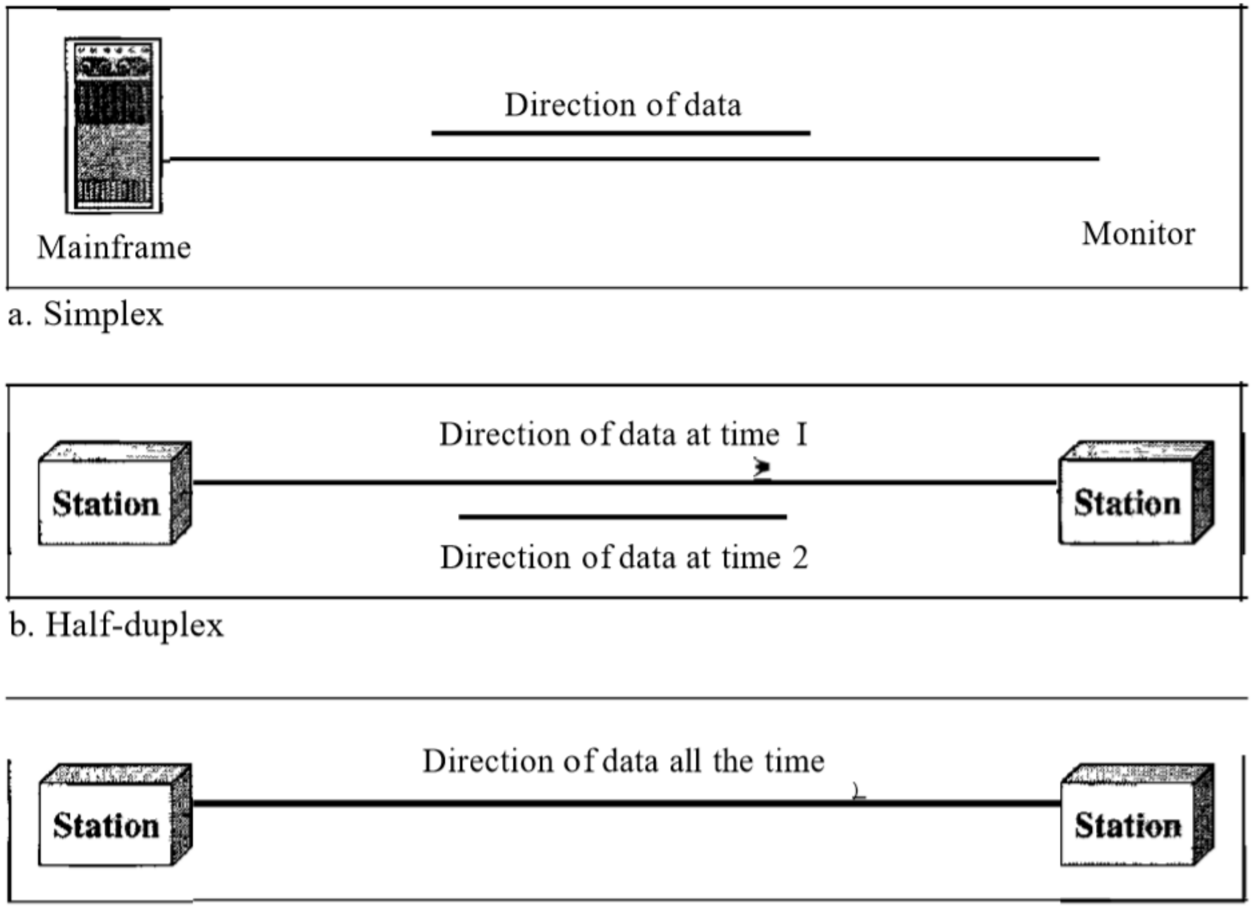
Information today comes in different forms such as text, numbers, images, audio, and video.

- **Text:** In data communications, text is represented as a bit pattern, a sequence of bits (0s or 1s).
- **Numbers:** Numbers are also represented by bit patterns.
- **Images:** Images are also represented by bit patterns.
- **Audio:** Audio refers to the recording or broadcasting of sound or music
- **Video:** Video refers to the recording or broadcasting of a picture or movie.

DATA FLOW COMPONENTS

Communication between two devices can be simplex, half-duplex, or full-duplex

- **Simplex:** In simplex mode, the communication is unidirectional.
- **Half-Duplex:** In half-duplex mode, each station can both transmit and receive, but not at the same time.
- **Full-Duplex:** In full-duplex mode, both stations can transmit and receive simultaneously


Direction of data
Mainframe
Monitor
Direction of data at time I
Station
Station
Direction of data at time 2
Direction of data all the time
Station
Station

a. Simplex

b. Half-duplex

c. Full·duplex

1.2 NETWORKS

A network is a set of devices (nodes) connected by communication links.

Distributed Processing: Most networks use distributed processing, in which a task is divided among multiple computers.

Network Criteria:

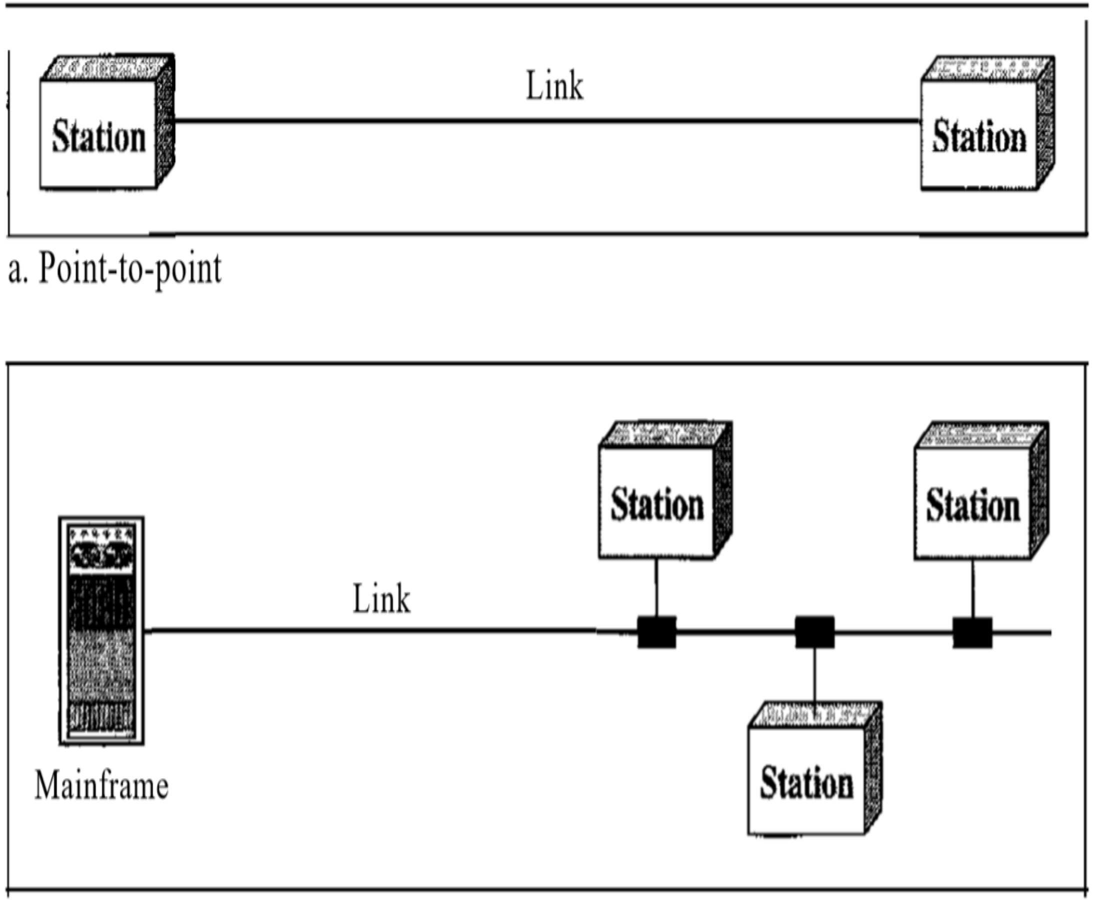
- **Performance:** Performance can be measured in many ways, including **transit time and response time**. Transit time is the amount of time required for a message to travel from one device to another. Response time is the elapsed time between an inquiry and a response.
- **Reliability:** Reliability 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:** Network 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.

PHYSICAL STRUCTURES

Type of Connection:

- **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.

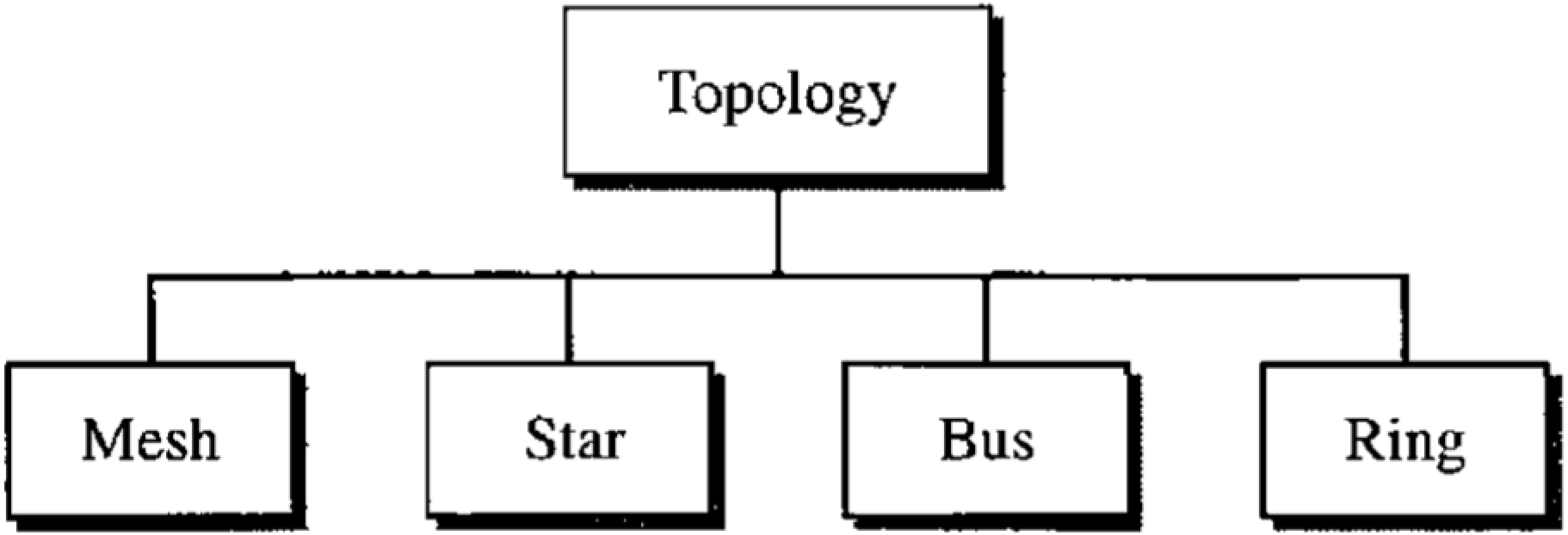
- **Multipoint**: A multipoint (also called multidrop) connection is one in which more than two specific devices share a single link . In a multipoint environment, the capacity of the channel is shared, either spatially or temporally. If several devices can use the link simultaneously, it is a spatially shared connection. If users must take turns, it is a timeshared connection.



a. Point-to-point

b. Multipoint

PHYSICAL TOPOLOGY



MESH TOPOLOGY

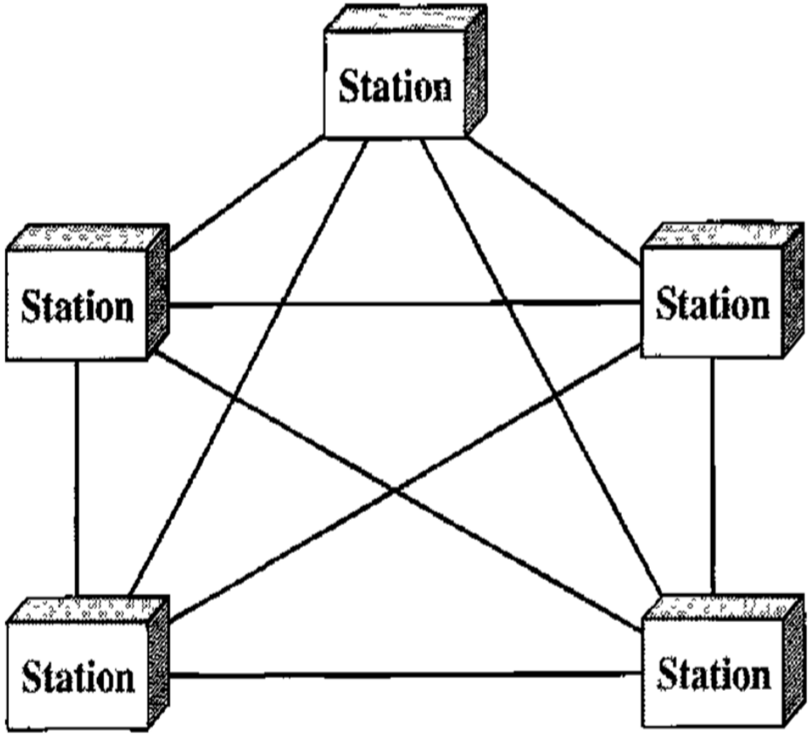
- In a mesh topology, every device has a dedicated point-to-point link to every other device.
- For n nodes we ned n-1 links

Advantages:

  - Eliminating the traffic problems
  - Robust. If one link becomes unusable, it wont affect the entire system.
  - Privacy or security.
  - Fault identification and fault isolation easy.

Disadvantages:

  - Installation and reconnection are difficult.
  - It need bulk of the wiring
  - The hardware required to connect each link can be expensive.



STAR TOPOLOGY

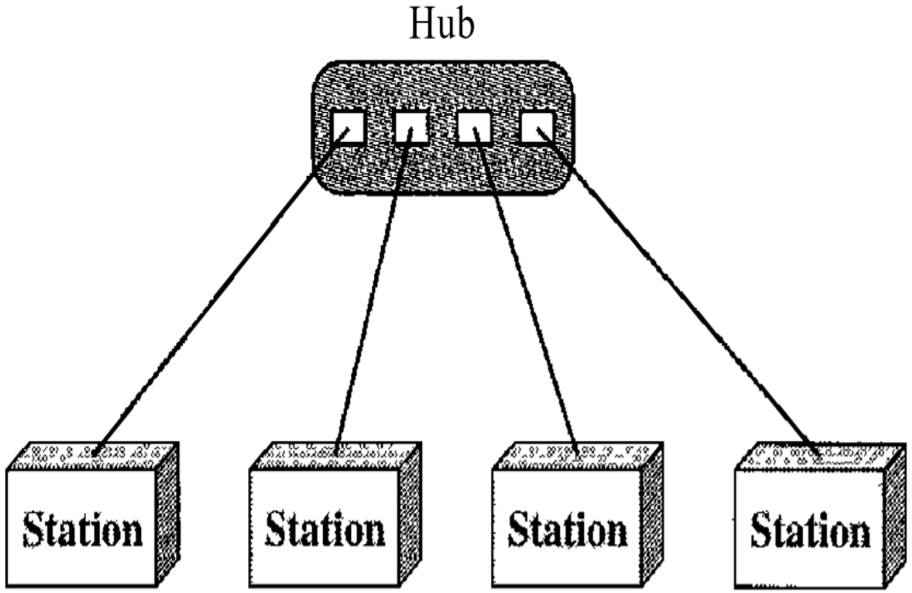
- Each device has a dedicated point-to-point link only to a central controller, usually called a hub. The devices are not directly linked to one another.

Advantages:

- Less expensive than a mesh topology.
- Each device needs only one link
- Far less cabling needed to install the networks
- Additions, moves, and deletions involve only one connection
- Robust. If one link fails, only that link is affected. All other links remain active.

Disadvantages:

- If the hub goes down, the whole system is dead.
- Although a star requires far less cable than a mesh, each node must be linked to a central hub.



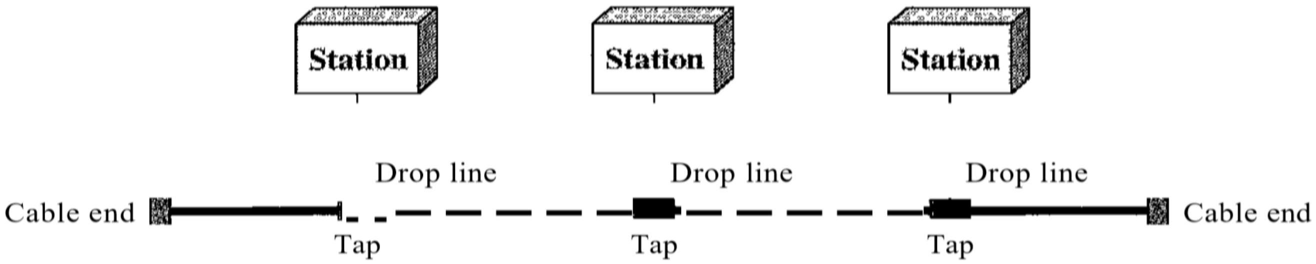
BUS TOPOLOGY

A bus topology is multipoint. One long cable acts as a backbone to link all the devices in a network

Advantages:

- Ease of installation.
- Uses less cabling than mesh or star topologies.
- In a bus, redundancy is eliminated. Only the backbone cable stretches through the entire facility.

- ***Disadvantages***:
  - Difficult reconnection and fault isolation.
  - Signal reflection at the taps can cause degradation in quality.
  - Adding new devices may require modification or replacement of the backbone.
  - A fault or break in the bus cable stops all transmission, even between devices on the same side of the problem.



RING TOPOLOGY

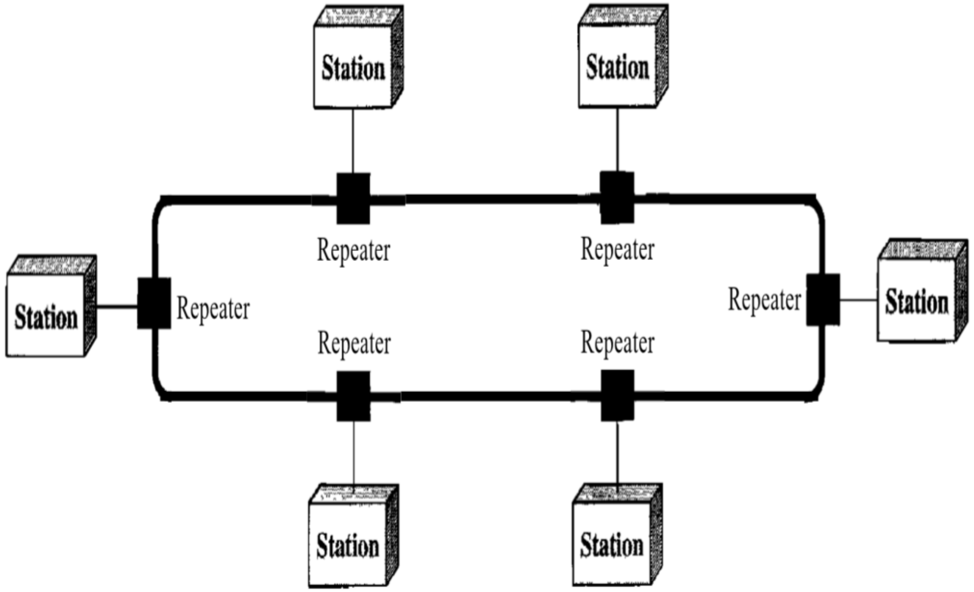
- Each device has a dedicated point-to-point connection with only the two devices on either side of it.
- A signal is passed along the ring in one direction, from device to device, until it reaches its destination

Advantages:

- A ring is relatively easy to install and reconfigure.
- To add or delete a device requires changing only two connections.
- Fault isolation is simplified.

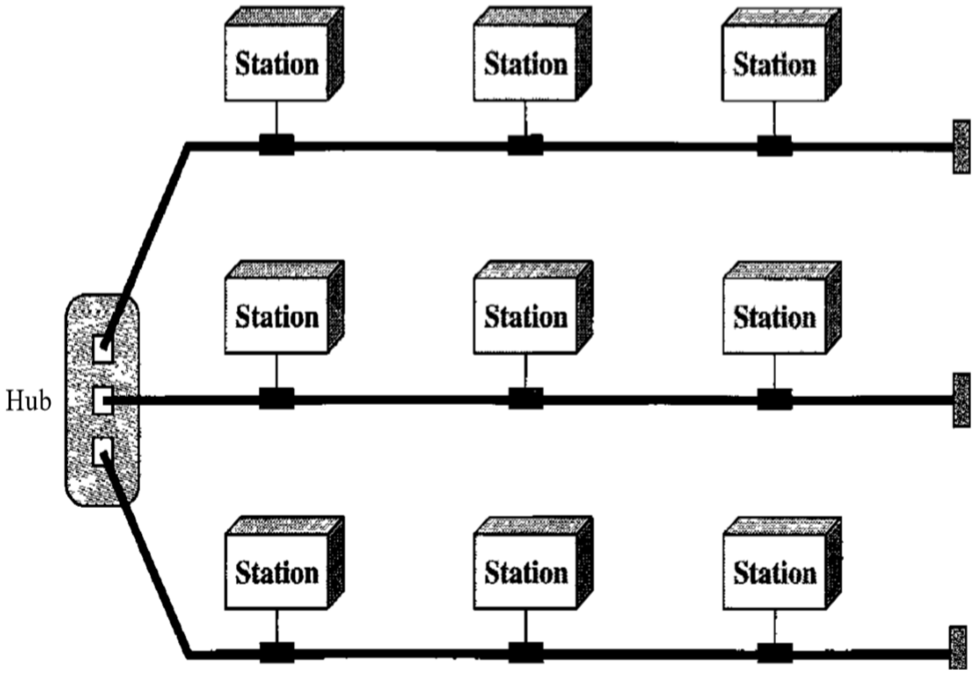
Disadvantages:

- It has unidirectional traffic.
- In a simple ring, a break in the ring can disable the entire network.



HYBRID TOPOLOGY

- A network can be hybrid. For example, we can have a main star topology with each branch connecting several stations in a bus topology



NETWORK MODELS

Categories of Networks

- ***Local Area Network:*** A local area network (LAN) is usually privately owned and links the devices in a single office, building, or campus.
- ***Wide Area Network:*** A wide area network (WAN) provides long-distance transmission of data, image, audio, and video information over large geographic areas that may comprise a country, a continent, or even the whole world.
- ***Metropolitan Area Networks:*** A metropolitan area network (MAN) is a network with a size between a LAN and a WAN. It normally covers the area inside a town or a city.
- ***Interconnection of Networks: Internetwork:*** Today, it is very rare to see a LAN, a MAN, or a LAN in isolation; they are connected to one another. When two or more networks are connected, they become an internetwork, or internet

DATA COMMUNICATIONS & NETWORKING

B.SC (COMPUTER SCIENCE) – III YEAR

SMCS52

By

Dr.N.N.Krishna Veni,
Assistant Professor,
Department of Computer Science,
Holy Cross Home Science College,
Thoothukudi

UNIT - I
INTRODUCTION

1.3 THE INTERNET

- 1.3.1 A Brief History
- 1.3.2 The Internet Today

1.4 PROTOCOLS AND STANDARDS

- 1.4.1 Protocols
- 1.4.2 Standards
- 1.4.3 Standards Organizations
- 1.4.4 Internet Standards

1.3 THE INTERNET

THE INTERNET

- The Internet is a communication system that has brought a wealth of information to our fingertips and organized it for our use

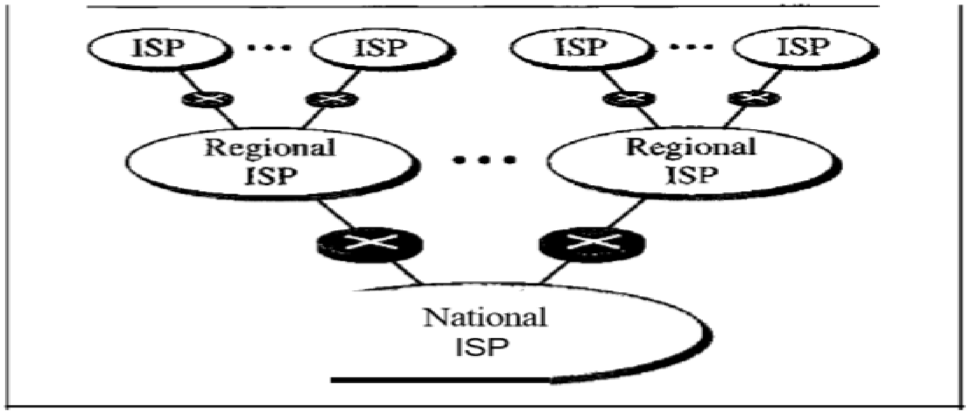
A Brief History

- An internet (note the lowercase letter i) is two or more networks that can communicate with each other.
- The most notable internet is called the Internet (uppercase letter I), a collaboration of more than hundreds of thousands of interconnected networks.
- In the mid-1960s, mainframe computers in research organizations were standalone devices. Computers from different manufacturers were unable to communicate with one another. The Advanced Research Projects Agency (ARPA) in the Department of Defense (DoD) was interested in finding a way to connect computers so that the researchers they funded could share their findings, thereby reducing costs and eliminating duplication of effort.

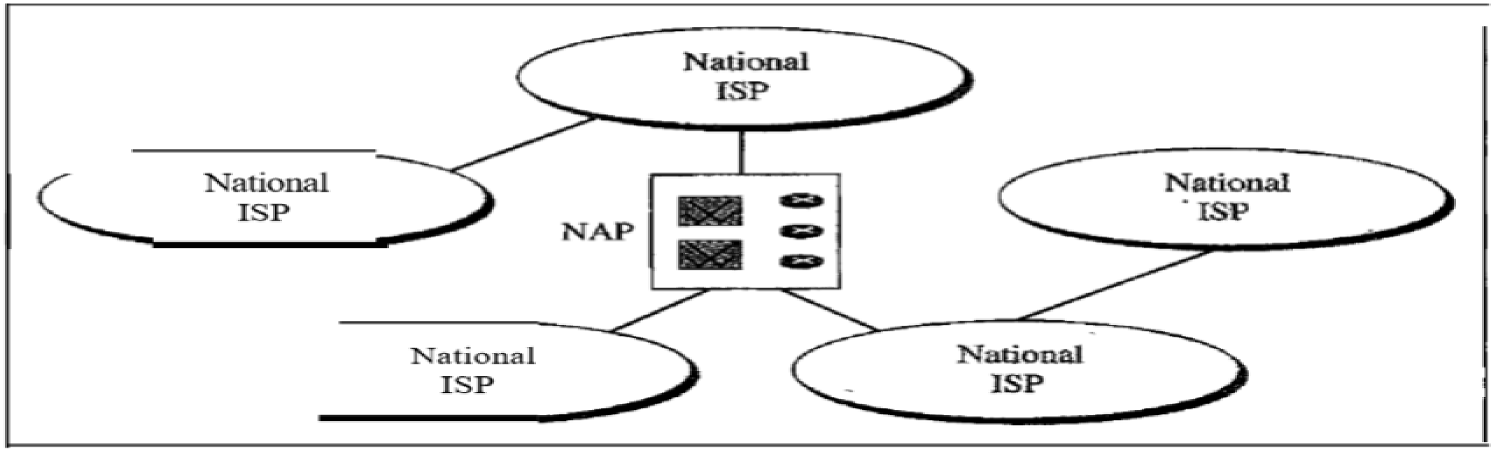
A BRIEF HISTORY

- In 1967, at an Association for Computing Machinery (ACM) meeting, ARPA presented its ideas that each host computer would be attached to a specialized computer, called an interface message processor (IMP). The IMPs, in tum, would be connected to one another. Each IMP had to be able to communicate with other IMPs as well as with its own attached host.
- By 1969, ARPANET was a reality. Four nodes, from various university of various countries, were connected via the IMPs to form a network. Software called the Network Control Protocol (NCP) provided communication between the hosts.
- In 1972, Vint Cerf and Bob Kahn, both of whom were part of the core ARPANET group, collaborated on what they called the Internetting Project.
- Cerf and Kahn's landmark 1973 paper outlined the protocols to achieve end-to-end delivery of packets. Shortly thereafter, authorities made a decision to split TCP into two protocols: Transmission Control Protocol (TCP) and Internetworking Protocol (lP). The internetworking protocol became known as TCPIIP.

THE INTERNET TODAY



a. Structure of a national ISP



b. Interconnection of national ISPs

1.4 PROTOCOLS AND STANDARDS

Protocols

- A protocol is a set of rules that govern data communications. The key elements of a protocol are
  - **Syntax:** The term syntax refers to the structure or format of the data.
  - **Semantics:** The word semantics refers to the meaning of each section of bits. How is a particular pattern to be interpreted, and what action is to be taken based on that interpretation?
  - **Timing:** The term timing refers to two characteristics: when data should be sent and how fast they can be sent.

Standards

- Standards provide guidelines to manufacturers, vendors, government agencies, and other service providers to ensure the kind of interconnectivity necessary
- Data communication standards fall into two categories: de facto (meaning "by fact" or "by convention") and de jure (meaning "by law" or "by regulation").
- **De facto:** Standards that have not been approved by an organized body but have been adopted as standards through widespread use are de facto standards..
- **De jure:** Those standards that have been legislated by an officially recognized body are de jure standards.

STANDARDS ORGANIZATIONS

- Standards are developed through the cooperation of **standards creation committees, forums, and government regulatory agencies.**
- ***Standards Creation Committees:***
  - International Organization for Standardization (ISO)
  - International Telecommunication Union-Telecommunication Standards Sector (ITU-T)
  - American National Standards Institute (ANSI)
  - Institute of Electrical and Electronics Engineers (IEEE)
  - Electronic Industries Association (EIA)
- ***Forums:*** To accommodate the need for working models and agreements and to facilitate the standardization process, many special-interest groups have developed forums made up of representatives from interested corporations.
- ***Regulatory Agencies:*** The purpose of these agencies is to protect the public interest by regulating radio, television, and wire/cable communications.

DATA COMMUNICATIONS & NETWORKING

B.SC (COMPUTER SCIENCE) – III YEAR

SMCS52

By

Dr.N.N.Krishna Veni,
Assistant Professor,
Department of Computer Science,
Holy Cross Home Science College,
Thoothukudi

UNIT - I
INTRODUCTION

1.5 LAYERED TASKS

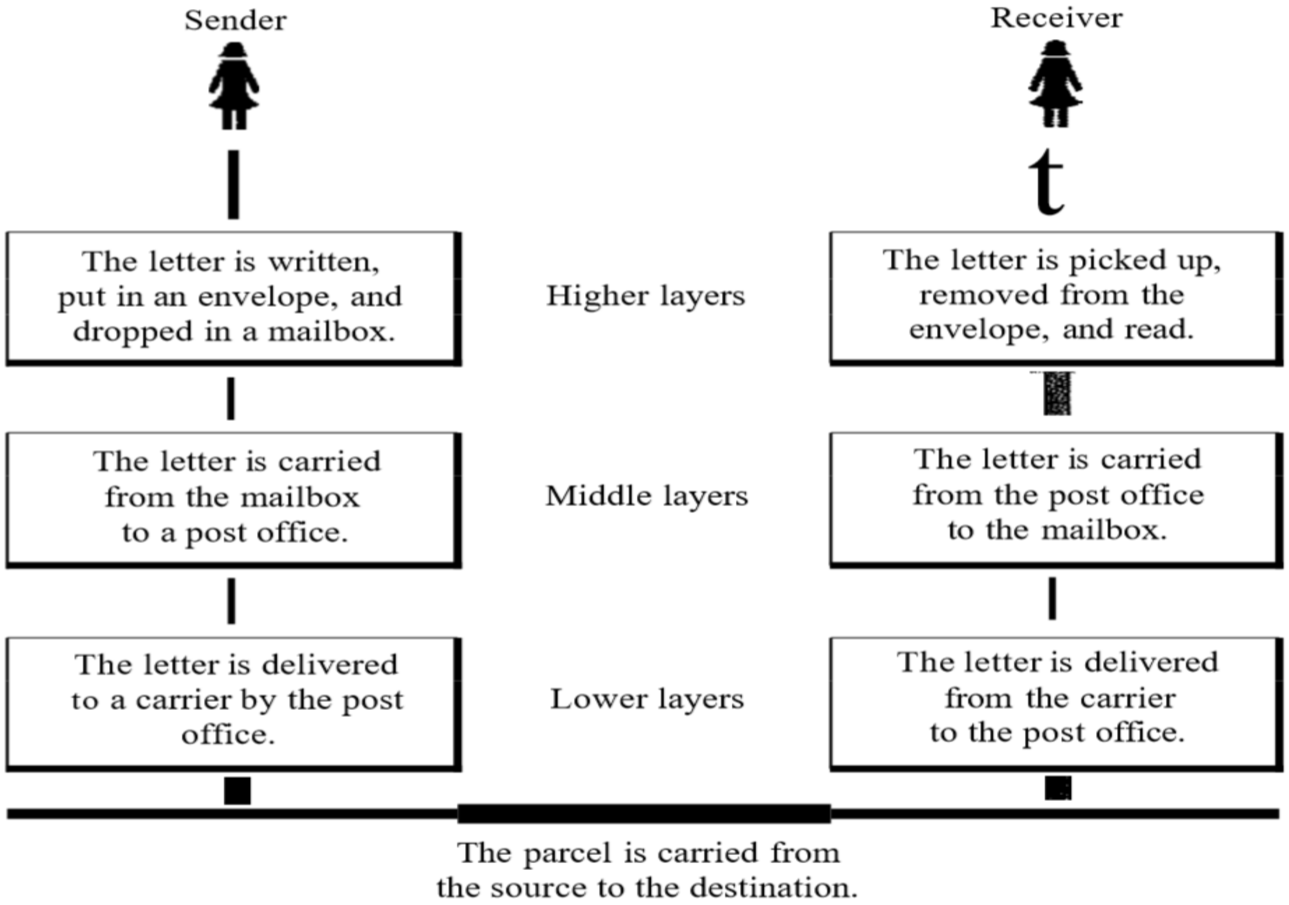
- 1.5.1 Sender, Receiver, and Carrier
- 1.5.2 Hierarchy

1.6 THE OSI MODEL

- 1.6.1 Layered Architecture
- 1.6.2 Peer-to-Peer Processes
- 1.6.3 Encapsulation

1.5 LAYERED TASKS

- We use the concept of layers in our daily life. As an example, let us consider two friends who communicate through postal mail.



Sender
Receiver
The letter is written, put in an envelope, and dropped in a mailbox.
Higher layers
The letter is picked up, removed from the envelope, and read.
The letter is carried from the mailbox to a post office.
Middle layers
The letter is carried from the post office to the mailbox.
The letter is delivered to a carrier by the post office.
Lower layers
The letter is delivered from the carrier to the post office.
The parcel is carried from the source to the destination.

1.5 LAYERED TASKS

1.5.1 SENDER, RECEIVER, AND CARRIER

At the Sender Site

- **Higher layer:** The sender writes the letter, inserts the letter in an envelope, writes the sender and receiver addresses, and drops the letter in a mailbox.
- **Middle layer**: The letter is picked up by a letter carrier and delivered to the post office.
- **Lower layer:** The letter is sorted at the post office; a carrier transports the letter.

On the Way

- The letter is then on its way to the recipient. On the way to the recipient's local post office, the letter may actually go through a central office. In addition, it may be transported by truck, train, airplane, boat, or a combination of these.

At the Receiver Site

- **Lower layer**: The carrier transports the letter to the post office.
- **Middle layer:** The letter is sorted and delivered to the recipient's mailbox.
- **Higher layer:** The receiver picks up the letter, opens the envelope, and reads it.

1.5 LAYERED TASKS

1.5.1 Hierarchy

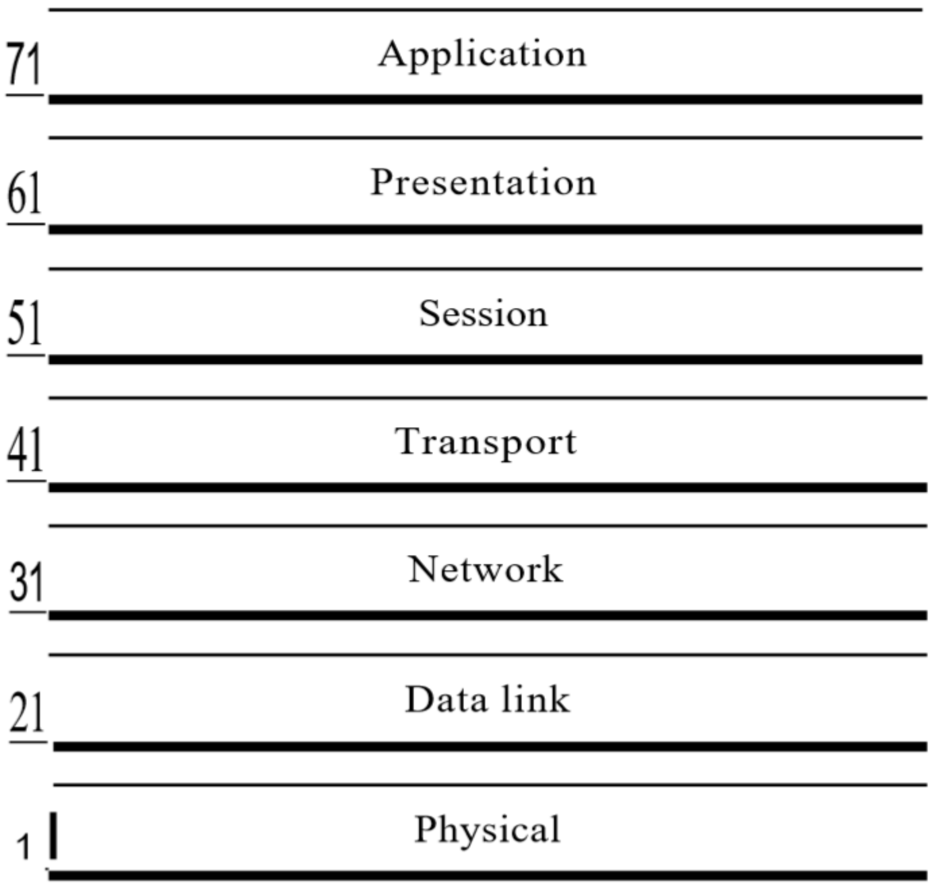
- According to our analysis, there are three different activities at the sender site and another three activities at the receiver site.
- The task of transporting the letter between the sender and the receiver is done by **the carrier**.
- The tasks must be done in the order given in the hierarchy.

Services:

- Each layer at the sending site uses the services of the layer immediately below it.
- The sender at the higher layer uses the services of the middle layer.
- The middle layer uses the services of the lower layer.
- The lower layer uses the services of the carrier.

1.6 THE OSI MODEL

- International Standards Organization (ISO) is established in 1947.
- An ISO standard that covers all aspects of network communications is the Open Systems Interconnection(OSI) model.
- It was first introduced in the late 1970s.
- 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.



7 Application

6 Presentation

5 Session

4 Transport

3 Network

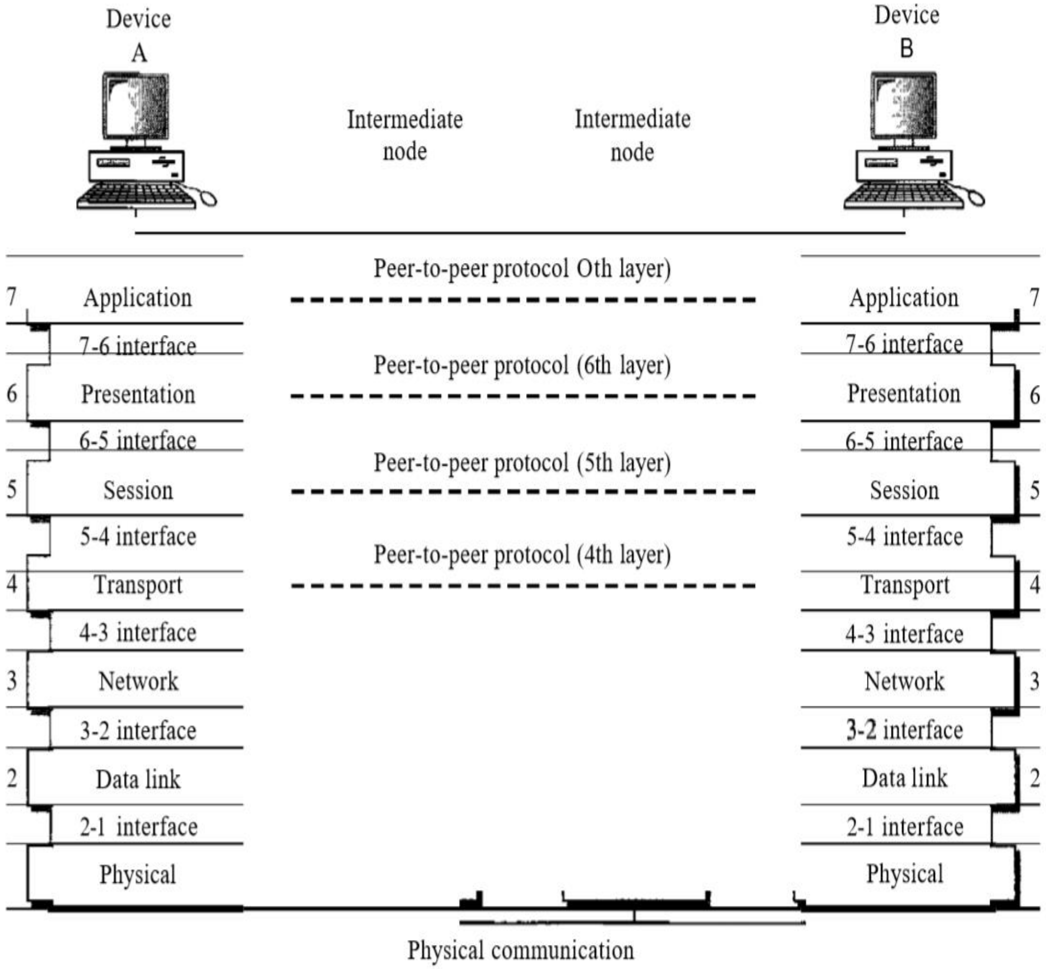
2 Data link

1 Physical

1.6 THE OSI MODEL

1.6.1 Layered Architecture

- As the message travels from A to B, it may pass through many intermediate nodes. These intermediate nodes usually involve only the first three layers of the OSI model.
- Within a **single machine,** each layer calls upon the services of the layer just below it. Layer 3, for example, uses the services provided by layer 2 and provides services for layer 4.
- **Between machines,** layer x on one machine communicates with layer x on another machine. This communication is done by **protocols.**
- The processes on each machine that communicate at a given layer are called **peer-to-peer processes**.



1.6 THE OSI MODEL

1.6.2 Peer-to-Peer Processes

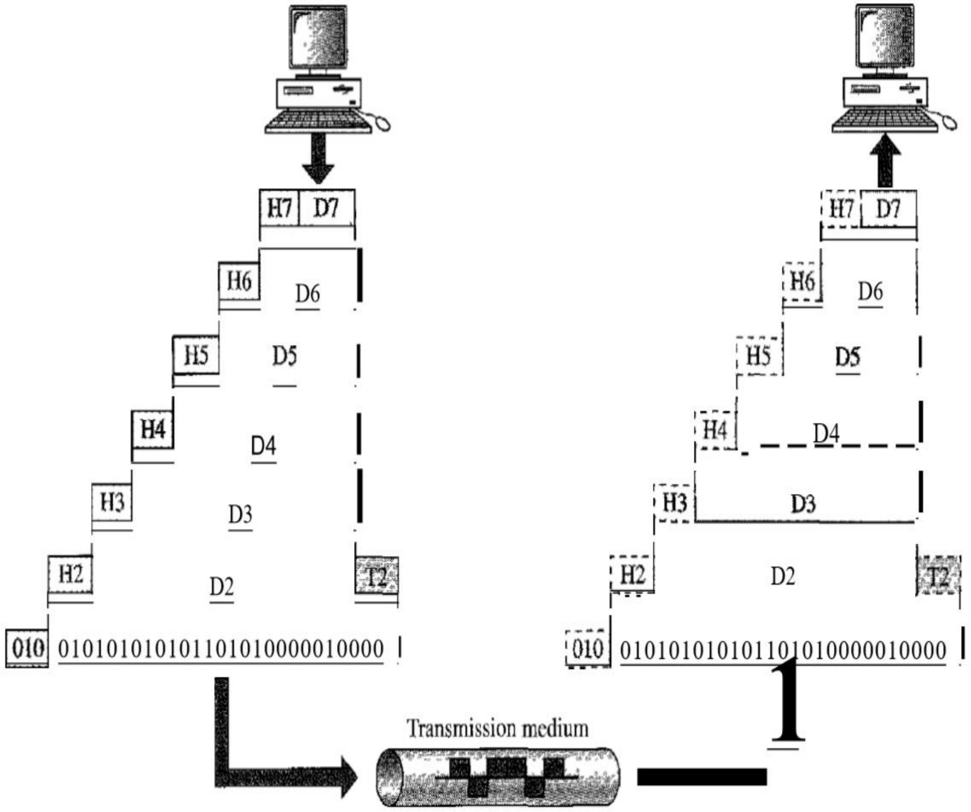
- At the physical layer, **communication is direct**. Device A sends a stream of bits to device B (through intermediate nodes).
- At the higher layers, however, communication must move down through the layers on device A, over to device B, and then back up through the layers.
- Each layer in the sending device **adds its own information** to the message it receives from the layer just above it and passes the whole package to the layer just below it.
- At layer 1 the entire package is **converted to a form that can be transmitted to the receiving device**.
- At the receiving machine, the message is unwrapped layer by layer, with each process receiving and removing the data meant for it.
- For example, layer 2 removes the data meant for it, then passes the rest to layer 3. Layer 3 then removes the data meant for it and passes the rest to layer 4, and so on.

1.6 THE OSI MODEL

Interfaces Between Layers: The passing of the data and network information through the layers of the sending device and receiving device is made possible by an interface between each pair of adjacent layers.

Organization of the Layers: The seven layers can be thought of as belonging to three subgroups.

- **The network support layers(Layers 1, 2 and 3):** They deal with the physical aspects of moving data from one device to another .
- **The user support layers(Layers 5, 6 and 7):** They allow interoperability among unrelated software systems.
- **The transport layer(Layer 4):** It links the two subgroups and ensures that what the lower layers have transmitted is in a form that the upper layers can use.



1.6 THE OSI MODEL

Encapsulation

- A packet (header and data) at level 7 is encapsulated in a packet at level 6. The whole packet at level 6 is encapsulated in a packet at level 5, and so on. In other words, the data portion of a packet at level N - 1 carries the whole packet (data and header and maybe trailer) from level N. The concept is called encapsulation; level N - 1 is not aware of which part of the encapsulated packet is data and which part is the header or trailer. For level N - 1, the whole packet coming from level N is treated as one integral unit.

DATA COMMUNICATIONS & NETWORKING

B.SC (COMPUTER SCIENCE) – III YEAR

SMCS52

By
Dr.N.N.Krishna Veni,
Assistant Professor,
Department of Computer Science,
Holy Cross Home Science College,
Thoothukudi

UNIT - I
INTRODUCTION

1.5 LAYERED TASKS

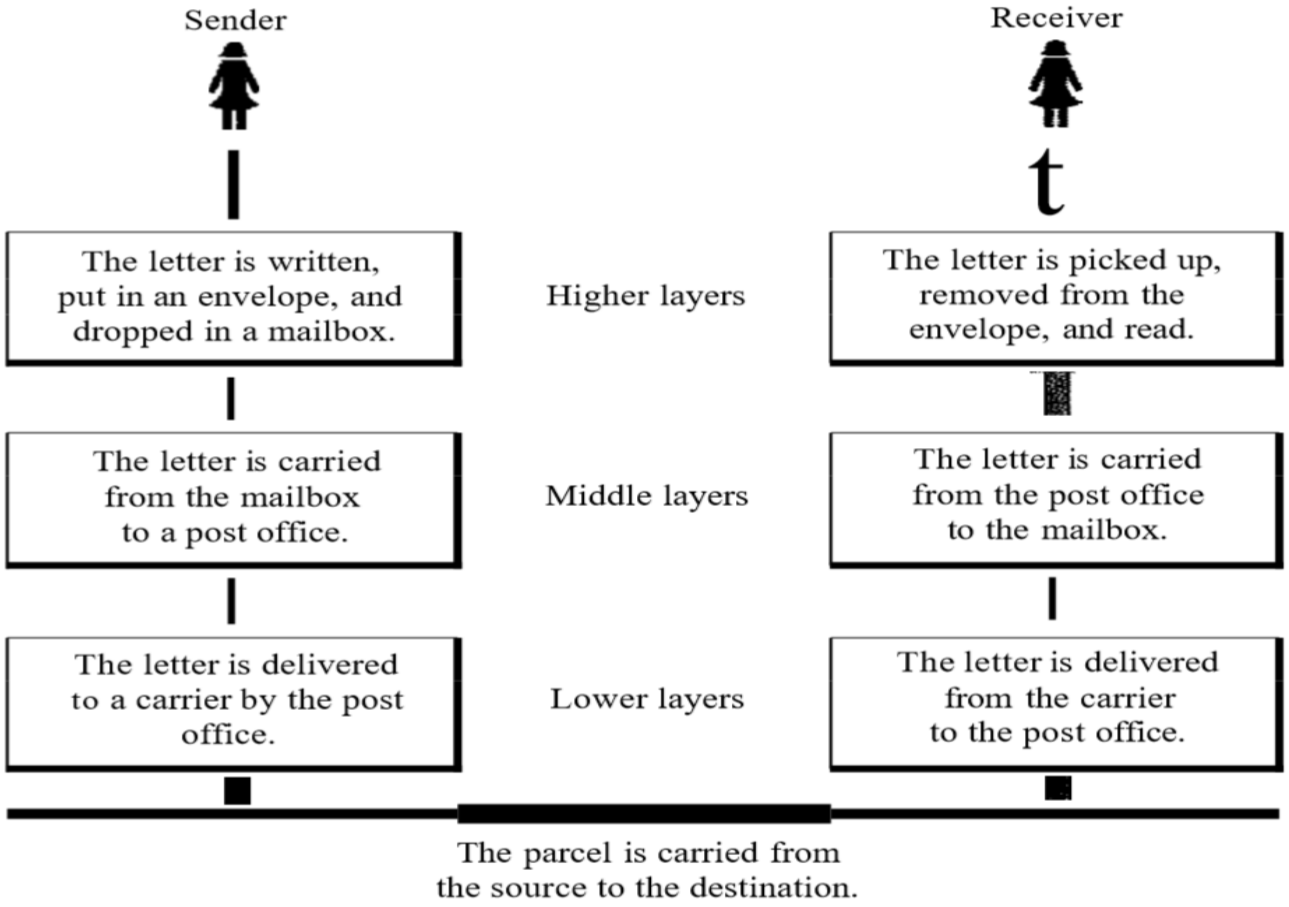
- 1.5.1 Sender, Receiver, and Carrier
- 1.5.2 Hierarchy

1.6 THE OSI MODEL

- 1.6.1 Layered Architecture
- 1.6.2 Peer-to-Peer Processes
- 1.6.3 Encapsulation

1.5 LAYERED TASKS

- We use the concept of layers in our daily life. As an example, let us consider two friends who communicate through postal mail.


Sender
Receiver
The letter is written, put in an envelope, and dropped in a mailbox.
Higher layers
The letter is picked up, removed from the envelope, and read.
The letter is carried from the mailbox to a post office.
Middle layers
The letter is carried from the post office to the mailbox.
The letter is delivered to a carrier by the post office.
Lower layers
The letter is delivered from the carrier to the post office.
The parcel is carried from the source to the destination.

1.5 LAYERED TASKS

1.5.1 SENDER, RECEIVER, AND CARRIER

At the Sender Site

- **Higher layer:** The sender writes the letter, inserts the letter in an envelope, writes the sender and receiver addresses, and drops the letter in a mailbox.
- **Middle layer**: The letter is picked up by a letter carrier and delivered to the post office.
- **Lower layer:** The letter is sorted at the post office; a carrier transports the letter.

On the Way

- The letter is then on its way to the recipient. On the way to the recipient's local post office, the letter may actually go through a central office. In addition, it may be transported by truck, train, airplane, boat, or a combination of these.

At the Receiver Site

- **Lower layer**: The carrier transports the letter to the post office.
- **Middle layer:** The letter is sorted and delivered to the recipient's mailbox.
- **Higher layer:** The receiver picks up the letter, opens the envelope, and reads it.

1.5 LAYERED TASKS

1.5.1 Hierarchy

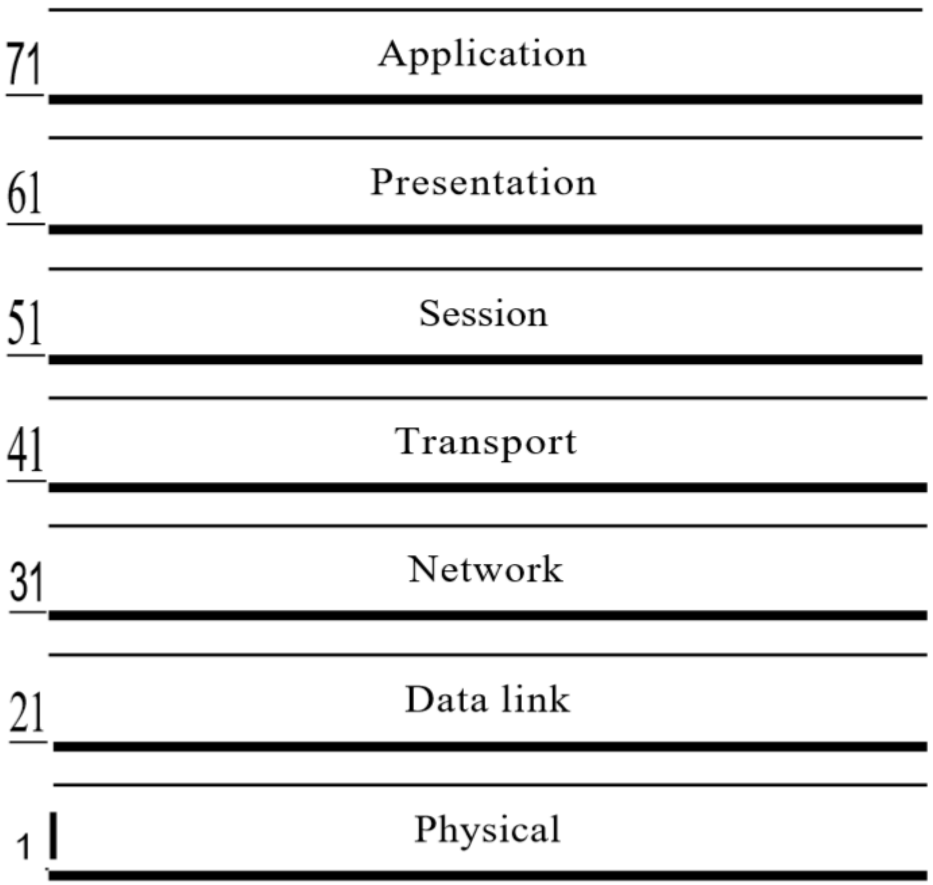
- According to our analysis, there are three different activities at the sender site and another three activities at the receiver site.
- The task of transporting the letter between the sender and the receiver is done by **the carrier**.
- The tasks must be done in the order given in the hierarchy.

Services:

- Each layer at the sending site uses the services of the layer immediately below it.
- The sender at the higher layer uses the services of the middle layer.
- The middle layer uses the services of the lower layer.
- The lower layer uses the services of the carrier.

1.6 THE OSI MODEL

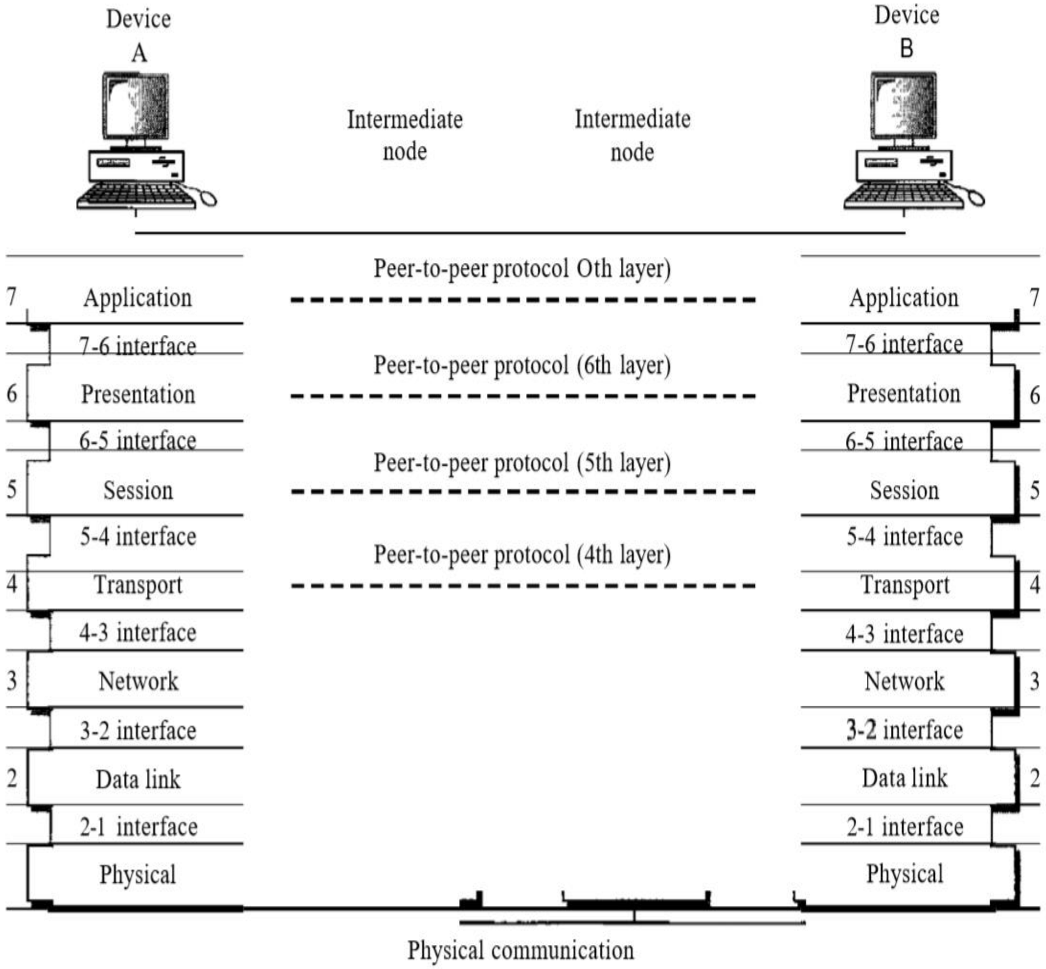
- International Standards Organization (ISO) is established in 1947.
- An ISO standard that covers all aspects of network communications is the Open Systems Interconnection(OSI) model.
- It was first introduced in the late 1970s.
- The purpose of the OSI model is to show **how to facilitate communication 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.



1.6 THE OSI MODEL

1.6.1 Layered Architecture

- As the message travels from A to B, it may pass through many intermediate nodes. These intermediate nodes usually involve only the first three layers of the OSI model.
- Within a **single machine,** each layer calls upon the services of the layer just below it. Layer 3, for example, uses the services provided by layer 2 and provides services for layer 4.
- **Between machines,** layer x on one machine communicates with layer x on another machine. This communication is done by **protocols.**
- The processes on each machine that communicate at a given layer are called **peer-to-peer processes**.



1.6 THE OSI MODEL

1.6.2 Peer-to-Peer Processes

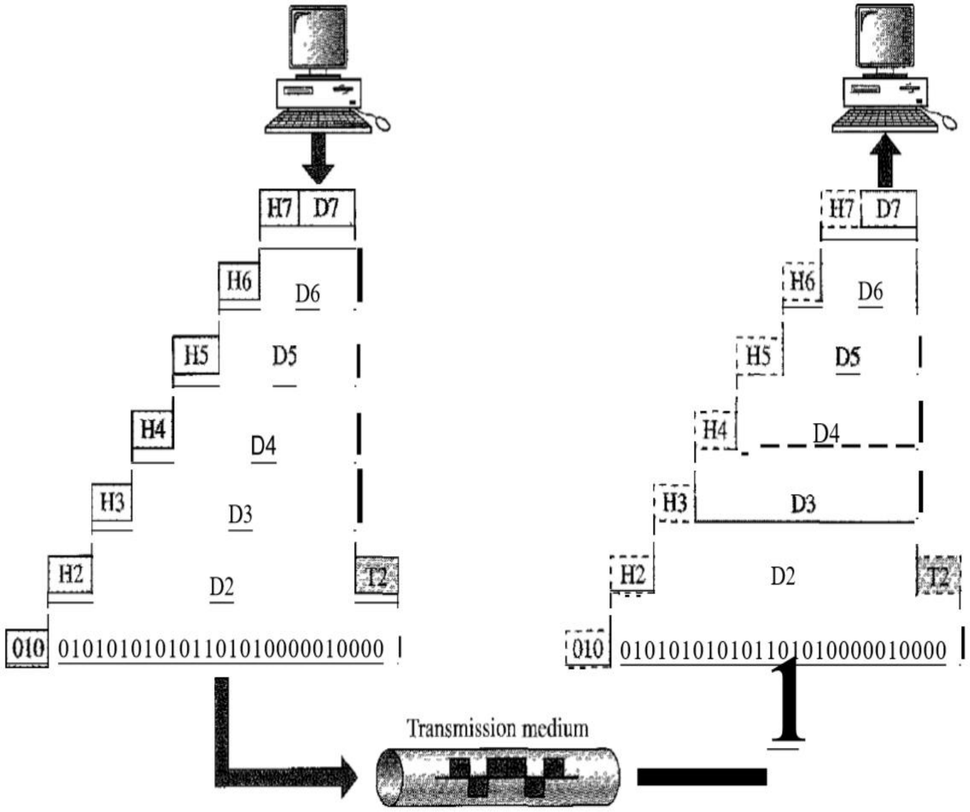
- At the physical layer, **communication is direct**. Device A sends a stream of bits to device B (through intermediate nodes).
- At the higher layers, however, communication must move down through the layers on device A, over to device B, and then back up through the layers.
- Each layer in the sending device **adds its own information** to the message it receives from the layer just above it and passes the whole package to the layer just below it.
- At layer 1 the entire package is **converted to a form that can be transmitted to the receiving device**.
- At the receiving machine, the message is unwrapped layer by layer, with each process receiving and removing the data meant for it.
- For example, layer 2 removes the data meant for it, then passes the rest to layer 3. Layer 3 then removes the data meant for it and passes the rest to layer 4, and so on.

1.6 THE OSI MODEL

Interfaces Between Layers: The passing of the data and network information through the layers of the sending device and receiving device is made possible by an interface between each pair of adjacent layers.

Organization of the Layers: The seven layers can be thought of as belonging to three subgroups.

- **The network support layers(Layers 1, 2 and 3):** They deal with the physical aspects of moving data from one device to another .
- **The user support layers(Layers 5, 6 and 7):** They allow interoperability among unrelated software systems.
- **The transport layer(Layer 4):** It links the two subgroups and ensures that what the lower layers have transmitted is in a form that the upper layers can use.



1.6 THE OSI MODEL

Encapsulation

- A packet (header and data) at level 7 is encapsulated in a packet at level 6. The whole packet at level 6 is encapsulated in a packet at level 5, and so on. In other words, the data portion of a packet at level N - 1 carries the whole packet (data and header and maybe trailer) from level N. The concept is called encapsulation; level N - 1 is not aware of which part of the encapsulated packet is data and which part is the header or trailer. For level N - 1, the whole packet coming from level N is treated as one integral unit.

DATA COMMUNICATIONS & NETWORKING

B.SC (COMPUTER SCIENCE) – III YEAR

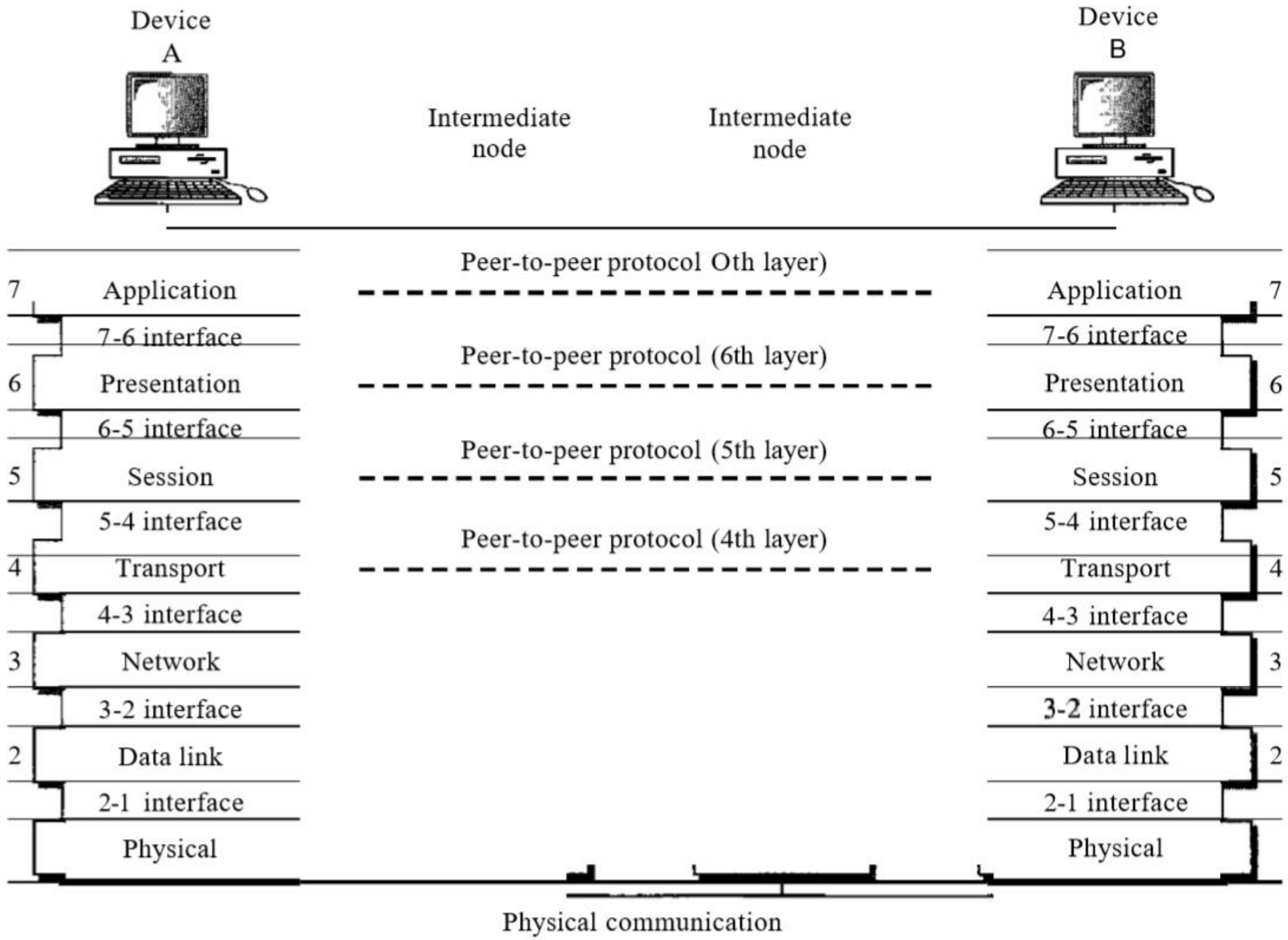
SMCS52

By
Dr.N.N.Krishna Veni,
Assistant Professor,
Department of Computer Science,
Holy Cross Home Science College,
Thoothukudi

UNIT - I
INTRODUCTION

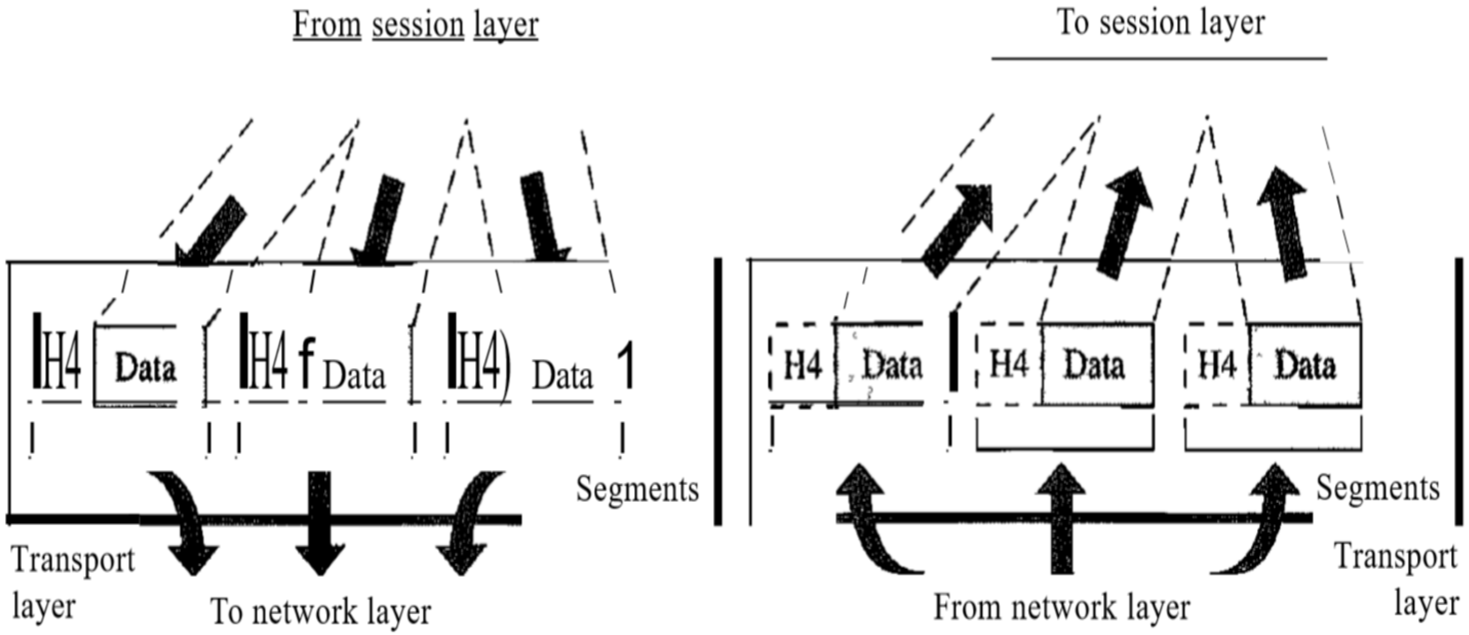
1.7 LAYERS IN THE OSI MODEL

UNIT - I
INTRODUCTION



Layers in OSI Model - Transport Layer

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



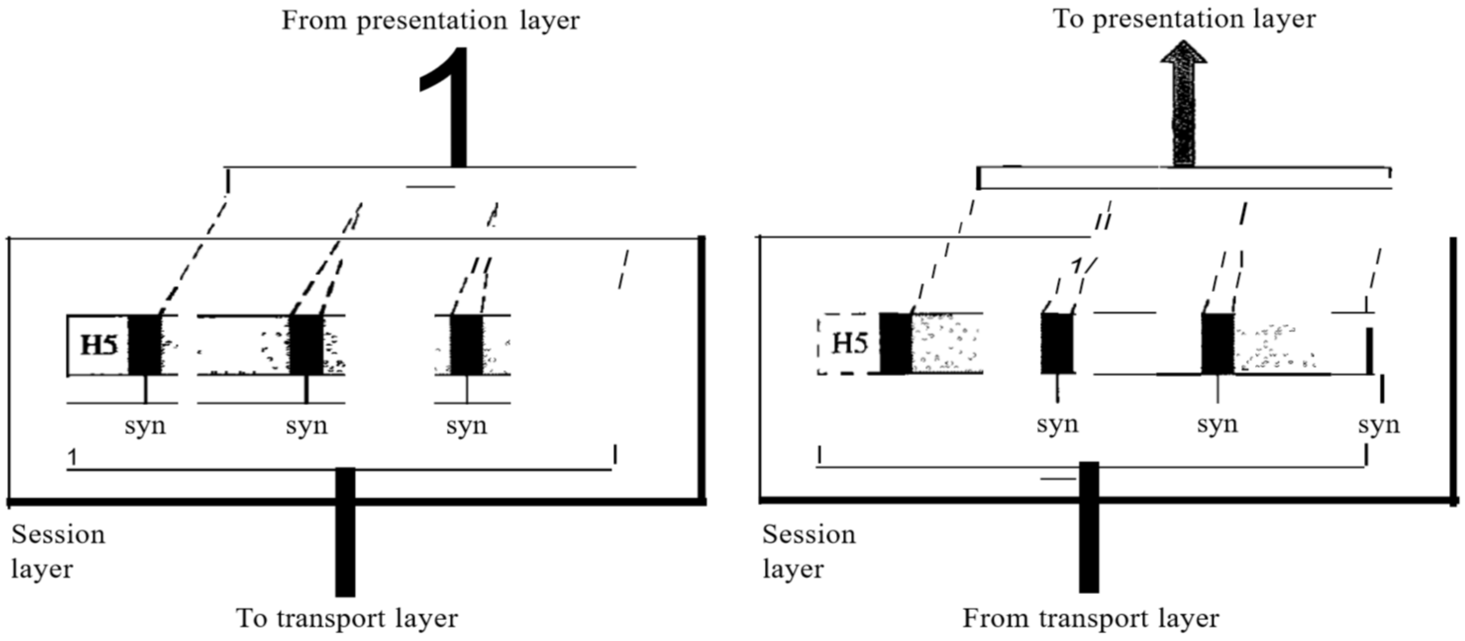
LAYERS IN OSI MODEL - TRANSPORT LAYER

Other responsibilities of the data link layer include the following:

- **Service-point addressing:** The transport layer header must a service-point address (or port address). The transport layer gets the entire message to the correct process on that computer.
- **Segmentation and reassembly**: A message is divided into transmittable segments, with each a sequence number. These numbers enable the transport layer to reassemble the message.
- **Connection control:** The transport layer can be either connectionless or connection oriented. A connectionless transport layer treats each segment as an independent packet and delivers it to the transport layer at the destination machine. A connection oriented transport layer makes a connection with the transport layer at the destination machine first before delivering the packets. After all the data are transferred, the connection is terminated.
- **Flow control:** Like the data link layer, the transport layer is responsible for flow control. However, flow control at this layer is performed end to end rather than across a single link.
- **Error control**: Like the data link layer, the transport layer is responsible for error control. However, error control at this layer is performed process-to-process rather than across a single link.

Layers in OSI Model – Session Layer

The session layer is responsible for dialog control and synchronization.

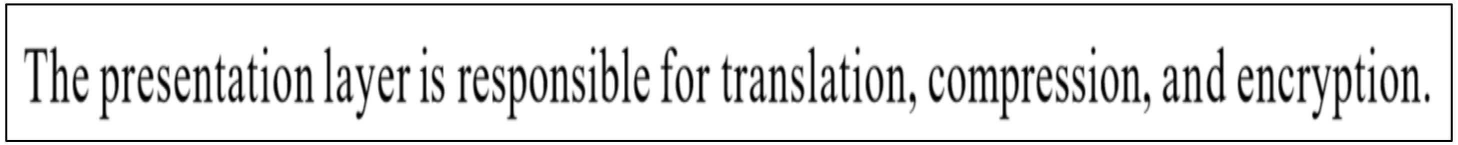


LAYERS IN OSI MODEL – SESSION LAYER

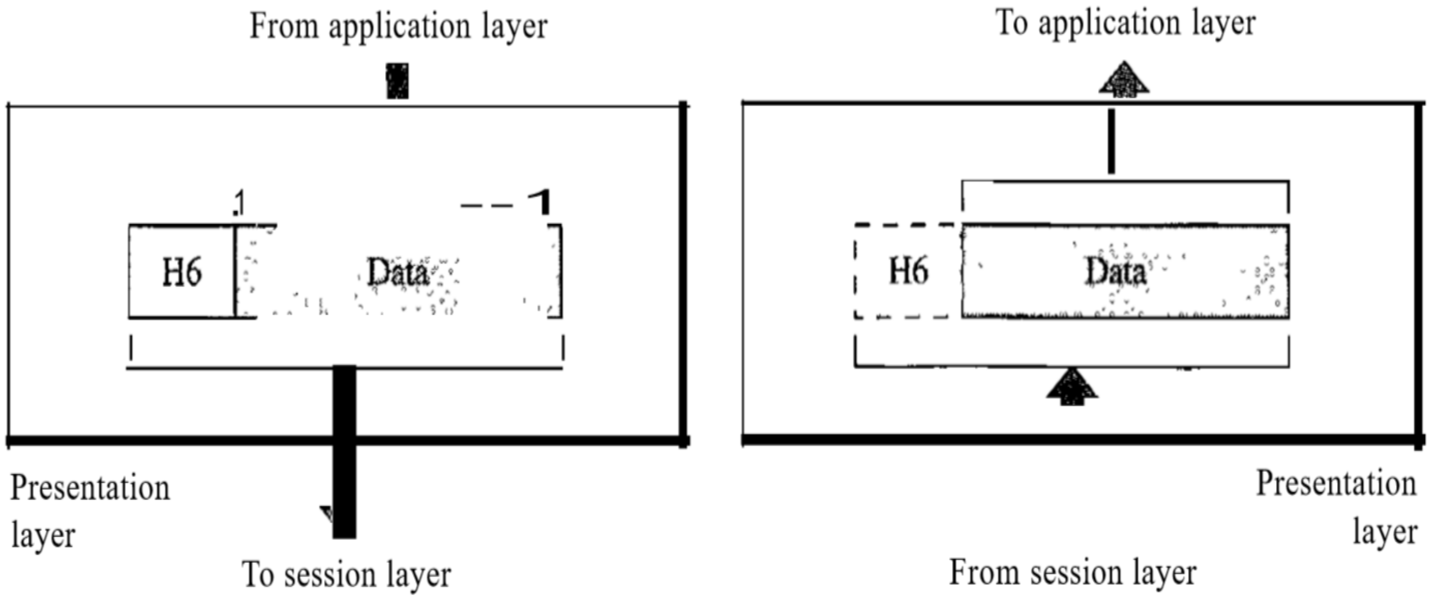
Specific responsibilities of the session layer include the following:

- **Dialog control:** The session layer allows two systems to enter into a dialog. It allows the communication between two processes to take place in either half-duplex (one way at a time) or full-duplex (two ways at a time) mode.
- **Synchronization:** The session layer allows a process to add checkpoints, or synchronization points, to a stream of data. For example, if a system is sending a file of2000 pages, it is advisable to insert checkpoints after every 100 pages to ensure that each 100-page unit is received and acknowledged independently. In this case, if a crash happens during the transmission of page 523, the only pages that need to be resent after system recovery are pages 501 to 523. Pages previous to 501 need not be resent.

LAYERS IN OSI MODEL – PRESENTATION LAYER



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



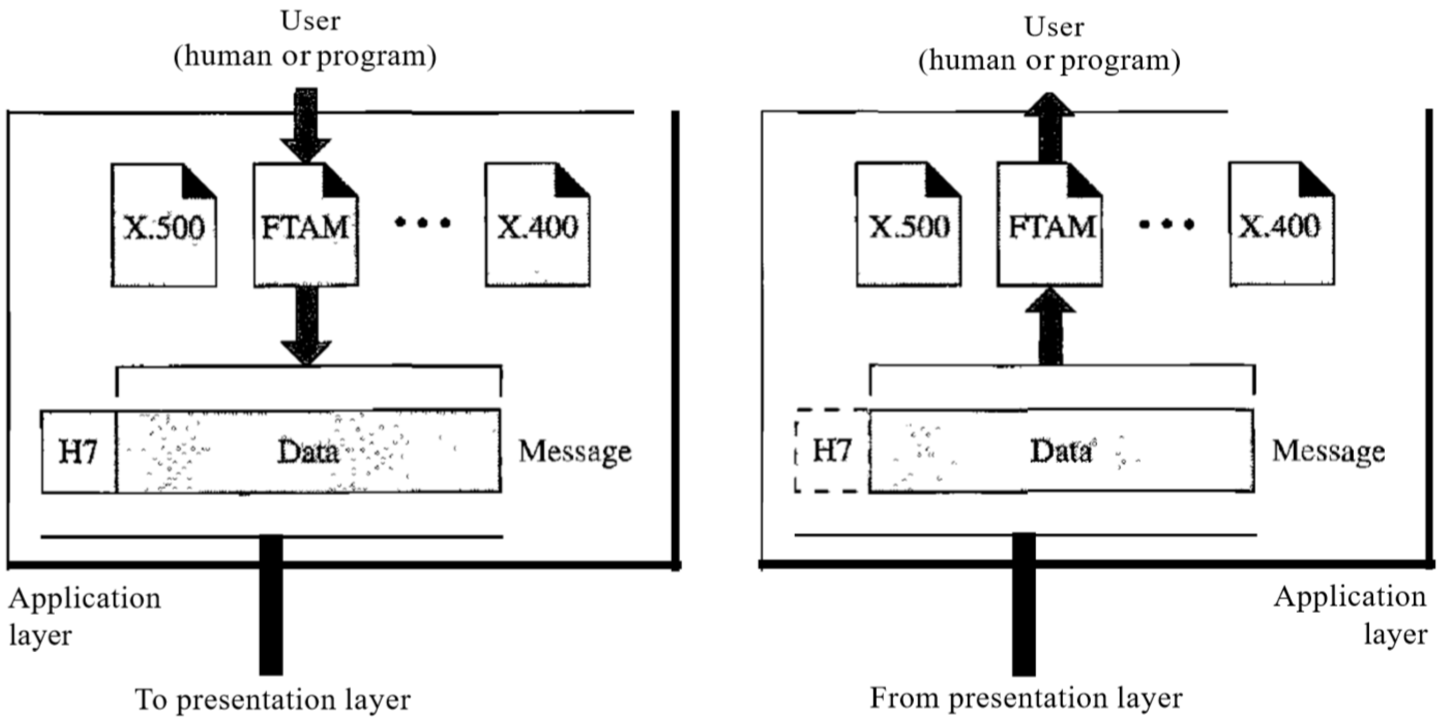
LAYERS IN OSI MODEL – PRESENTATION LAYER

Specific responsibilities of the presentation layer include the following:

- **Translation**: Because different computers use different encoding systems, the presentation layer is responsible for interoperability between these different encoding methods. The presentation layer at the sender changes the information from its sender-dependent format into a common format. The presentation layer at the receiving machine changes the common format into its receiver-dependent format.
- **Encryption:** To carry sensitive information, a system must be able to ensure privacy. Encryption means that the sender transforms the original information to another form and sends the resulting message out over the network. Decryption reverses the original process to transform the message back to its original form.
- **Compression:** Data compression reduces the number of bits contained in the information. Data compression becomes particularly important in the transmission of multimedia such as text, audio, and video.

LAYERS IN OSI MODEL – APPLICATION LAYER

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



LAYERS IN OSI MODEL – APPLICATION LAYER

Specific services provided by the application layer include the following:

- **Network virtual terminal:** A network virtual terminal is a software version of a physical terminal, and it allows a user to log on to a remote host. To do so, the application creates a software emulation of a terminal at the remote host. The user's computer talks to the software terminal which, in turn, talks to the host, and vice versa. The remote host believes it is communicating with one of its own terminals and allows the user to log on.
- **File transfer, access, and management:** This application allows a user to access files in a remote host (to make changes or read data), to retrieve files from a remote computer for use in the local computer, and to manage or control files in a remote computer locally.
- **Mail services**: This application provides the basis for e-mail forwarding and storage.
- **Directory services:** This application provides distributed database sources and access for global information about various objects and services.

DATA COMMUNICATIONS & NETWORKING

B.SC (COMPUTER SCIENCE) – III YEAR

SMCS52

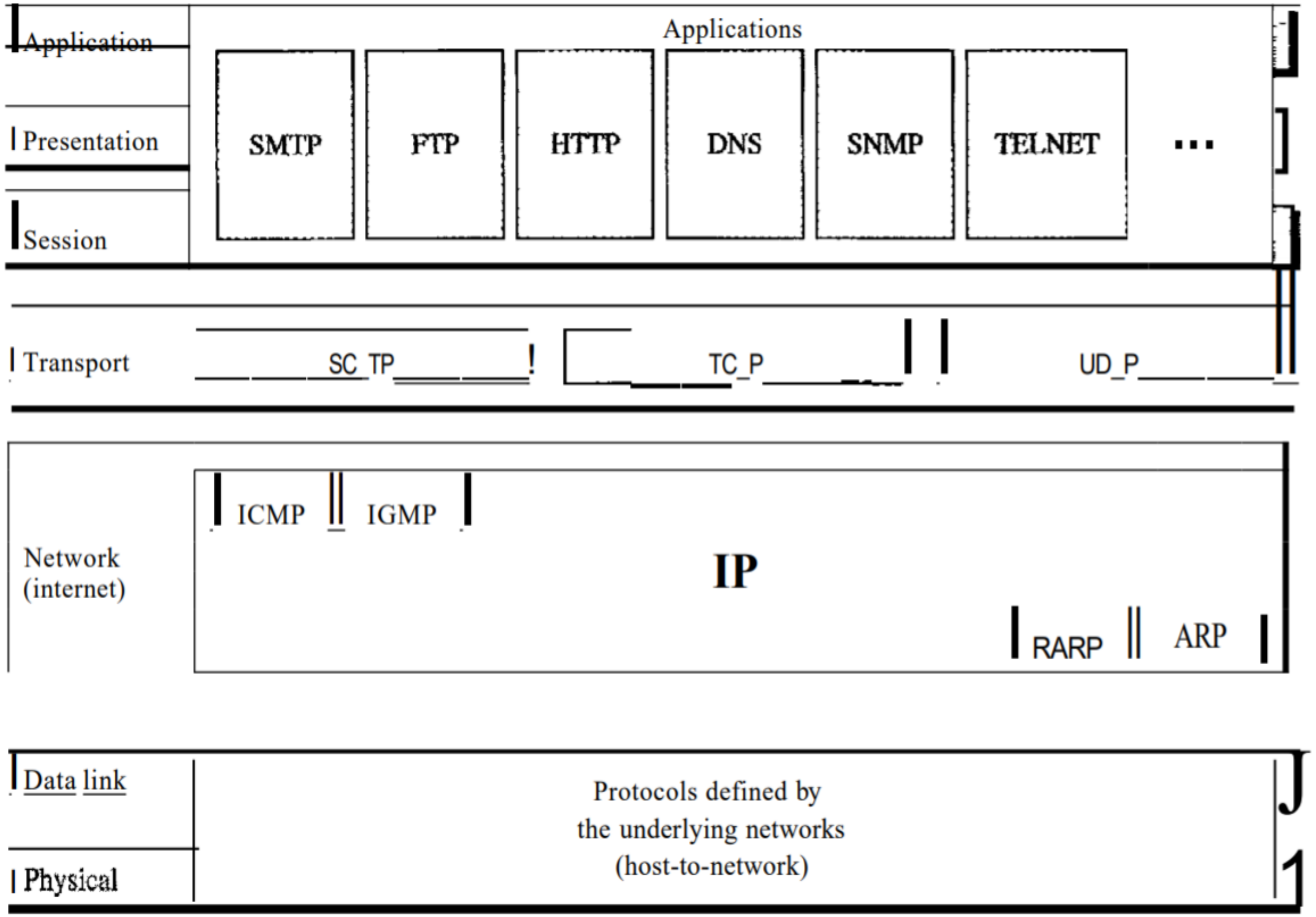
By
Dr.N.N.Krishna Veni,
Assistant Professor,
Department of Computer Science,
Holy Cross Home Science College,
Thoothukudi

UNIT - I
INTRODUCTION

1.8 TCP/IP PROTOCOL SUITE

UNIT - I
INTRODUCTION

Physical and Data Link Layers: At the physical and data link layers, TCPIIP does not define any specific protocol. It supports all the standard and proprietary protocols.



TCP/IP PROTOCOL SUITE

Network Layer: IP, in turn, uses four supporting protocols: ARP, RARP, ICMP, and IGMP

- **Internetworking Protocol (IP):** It is the transmission mechanism used by the TCP/IP protocols. It is an unreliable and connectionless protocol-a best-effort delivery service. IP transports data in packets called datagrams, each of which is transported separately. Datagrams can travel along different routes and can arrive out of sequence or be duplicated. IP does not keep track of the routes and has no facility for reordering datagrams once they arrive at their destination.
- **Address Resolution Protocol**: The Address Resolution Protocol (ARP) is used to associate a logical address with a physical address. ARP is used to find the physical address of the node when its Internet address is known.
- **Reverse Address Resolution Protocol:** It allows a host to discover its Internet address when it knows only its physical address.
- **Internet Control Message Protocol**: It is a mechanism used by hosts and gateways to send notification of datagram problems back to the sender. ICMP sends query and error reporting messages.
- **Internet Group Message Protocol**: It is used to facilitate the simultaneous transmission of a message to a group of recipients.

LAYERS IN OSI MODEL - TRANSPORT LAYER

Transport Layer : Has two protocols: TCP and UDP. UDP and TCP are transport level protocols responsible for delivery of a message from a process (running program) to another process.

- **User Datagram Protocol**: It is the simpler one. It is a process-to-process protocol that adds only port addresses, checksum error control, and length information to the data from the upper layer.
- **Transmission Control Protocol**: Provides full transport-layer services to applications. TCP is a reliable connection-oriented transport protocol.
- **Stream Control Transmission Protocol**: The Stream Control Transmission Protocol (SCTP) provides support for newer applications such as voice over the Internet.

Application Layer: The application layer in TCPIIP is equivalent to the combined session, presentation, and application layers in the OSI model Many protocols are defined at this layer.

DATA COMMUNICATIONS & NETWORKING

B.SC (COMPUTER SCIENCE) – III YEAR

SMCS52

By

Dr.N.N.Krishna Veni,
Assistant Professor,
Department of Computer Science,
Holy Cross Home Science College,
Thoothukudi

UNIT – II
INTRODUCTION

1.9 ADDRESSING

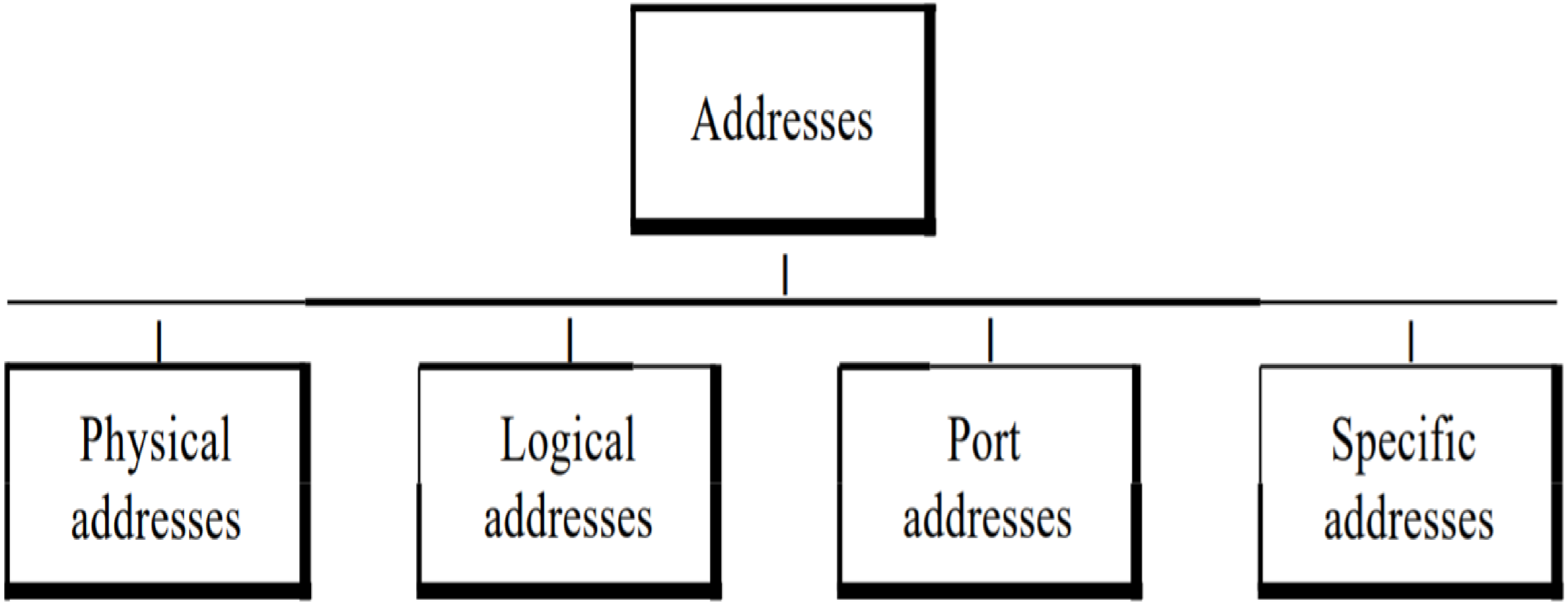
1.9.1 Physical Addresses

1.9.2 Logical Addresses

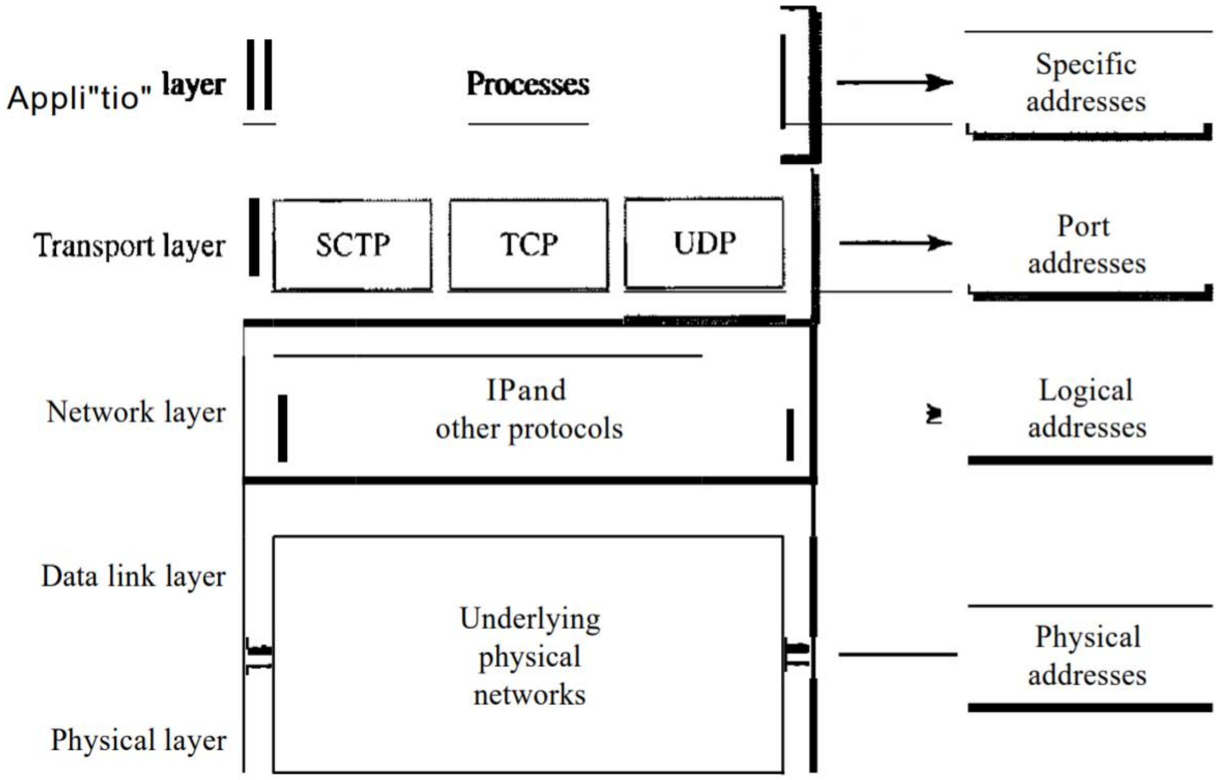
1.9.3 Port Addresses

1.9.4 Specific Addresses

ADDRESSING


Addresses
Physical addresses
Logical addresses
Port addresses
Specific addresses

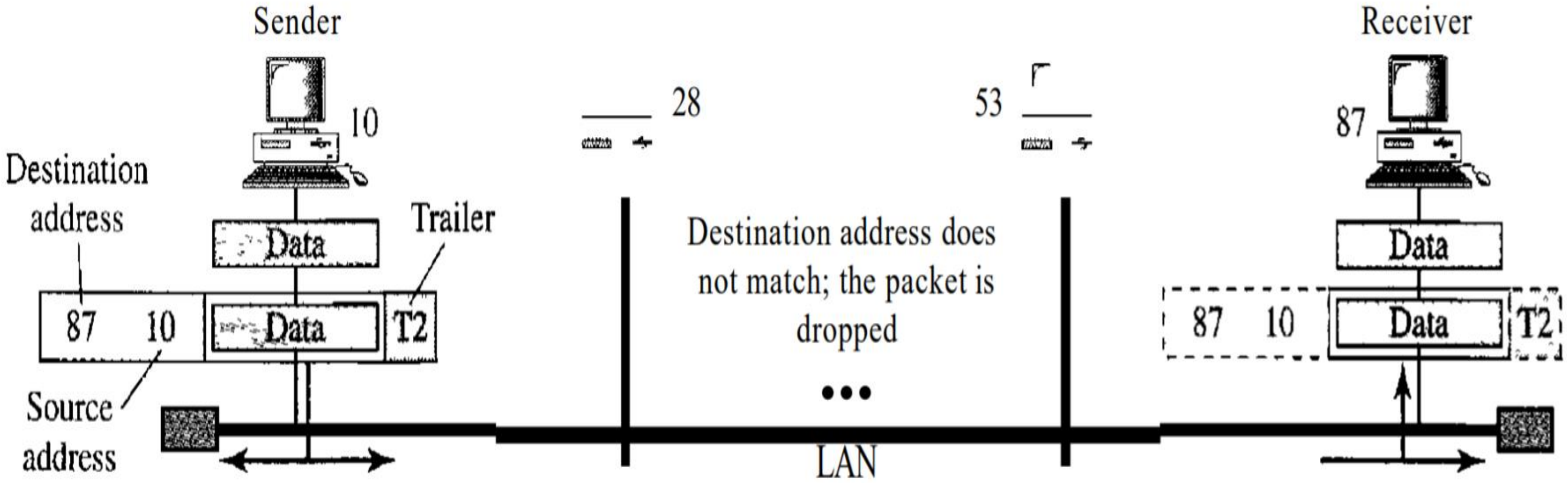
ADDRESSING


Appli"tio" layer
Processes
Specific addresses
Transport layer
SCTP
TCP
UDP
Port addresses
Network layer
IPand other protocols
Logical addresses
Data link layer
Underlying physical networks
Physical addresses
Physical layer

ADDRESSING

Physical Addresses (Within a network)

- The physical address, also known as the link address, is the address of a node as defined by its LAN or WAN. It is included in the frame used by the data link layer. It is the lowest-level address. The physical addresses have authority over the network (LAN or WAN). The size and format of these addresses vary depending on the network.



ADDRESSING

Logical addresses(Between Networks)

Logical addresses are necessary for universal communications that are independent of underlying physical networks.

Physical addresses are not adequate in an internetwork environment where different networks can have different address formats.

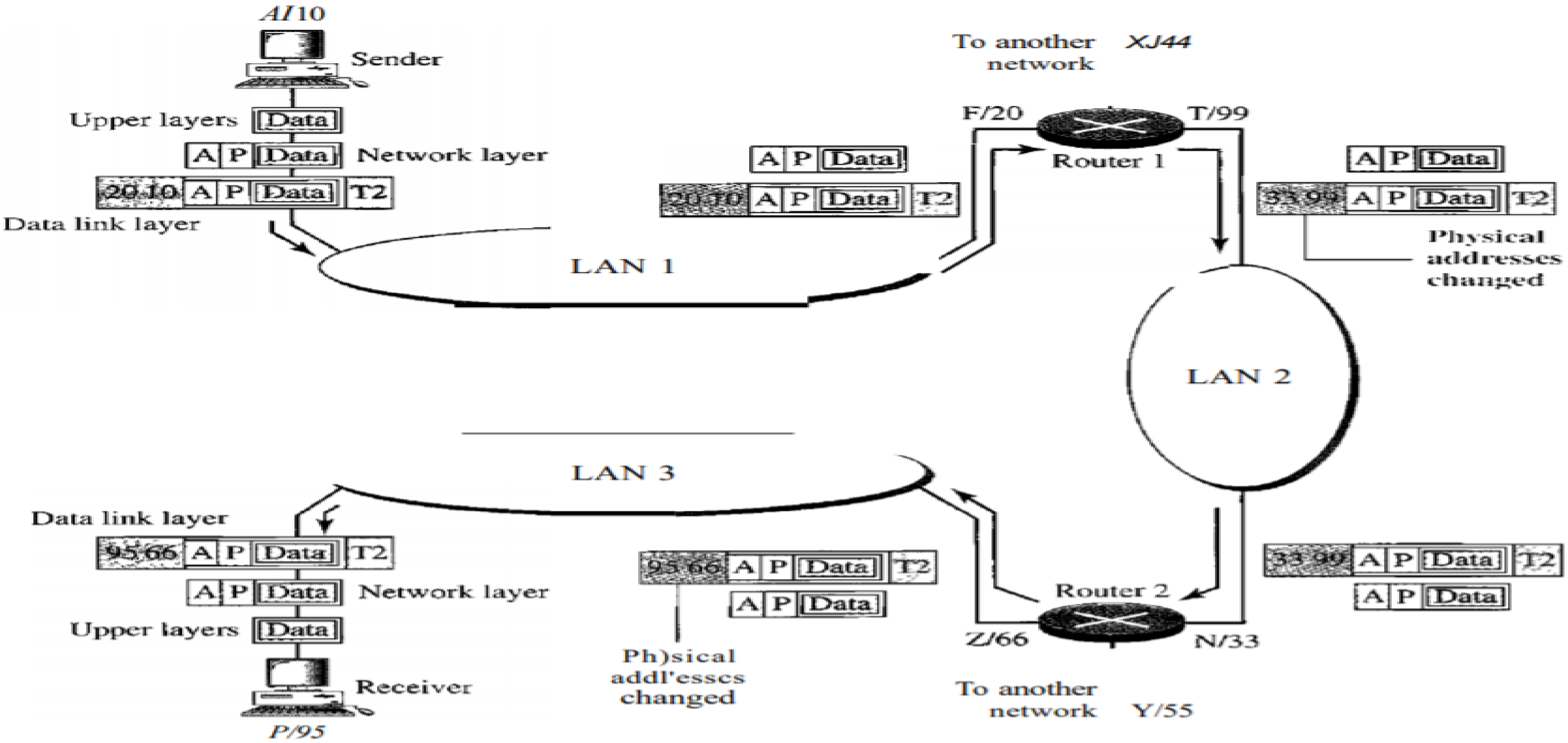
A universal addressing system is needed in which each host can be identified uniquely, regardless of the underlying physical network.

The logical addresses are designed for this purpose.

A logical address in the Internet is currently a 32-bit address that can uniquely define a host connected to the Internet.

No two publicly addressed and visible hosts on the Internet can have the same IP address.

ADDRESSING

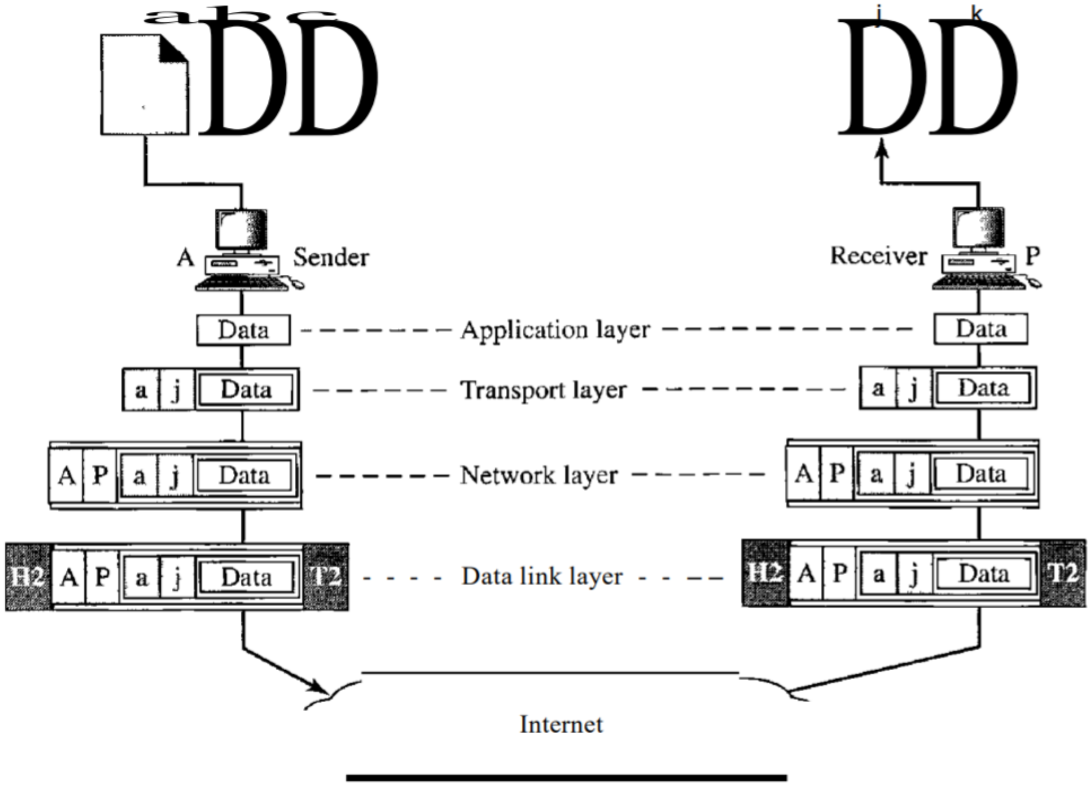


A/10
Sender
Upper layers
Data
A P Data
Network layer
20 10 A P Data T2
Data link layer
LAN 1
A P Data
20 10 A P Data T2
To another network
XJ44
F/20
T/99
Router 1
A P Data
33 99 A P Data T2
Physical addresses changed
LAN 2
LAN 3
Data link layer
95 66 A P Data T2
A P Data
Network layer
Upper layers
Data
Receiver
P/95
95 66 A P Data T2
A P Data
Ph)sical addl'esscs changed
Router 2
Z/66
N/33
To another network
Y/55
33 99 A P Data T2
A P Data

ADDRESSING

Port Addresses

- The IP address and the physical address are necessary for a quantity of data to travel from a source to the destination host. However, arrival at the destination host is not the final objective of data communications on the Internet. A system that sends nothing but data from one computer to another is not complete. Today, computers are devices that can run multiple processes at the same time. The end objective of Internet communication is a process communicating with another process. In the TCPIIP architecture, the label assigned to a process is called a port address. A port address in TCPIIP is 16 bits in length.



ADDRESSING

Specific Addresses

- Some applications have user-friendly addresses that are designed for that specific address. Examples include the e-mail address (for example, forouzan@fhda.edu) and the Universal Resource Locator (URL) (for example, www.mhhe.com). The first defines the recipient of an e-mail; the second is used to find a document on the World Wide Web. These addresses, however, get changed to the corresponding port and logical addresses by the sending computer

DATA COMMUNICATIONS & NETWORKING

B.SC (COMPUTER SCIENCE) – III YEAR

SMCS52

By
Dr.N.N.Krishna Veni,
Assistant Professor,
Department of Computer Science,
Holy Cross Home Science College,
Thoothukudi

UNIT – II
PHYSICAL LAYER

2.1 ANALOG AND DIGITAL

- 2.1.1 Analog and Digital Data
- 2.1.2 Analog and Digital Signals
- 2.1.3 Periodic and Nonperiodic Signals

2.2 TRANSMISSION IMPAIRMENT

- 2.2.1 Attenuation
- 2.2.2 Distortion
- 2.2.3 Noise

UNIT – II
ANALOG & DIGITAL

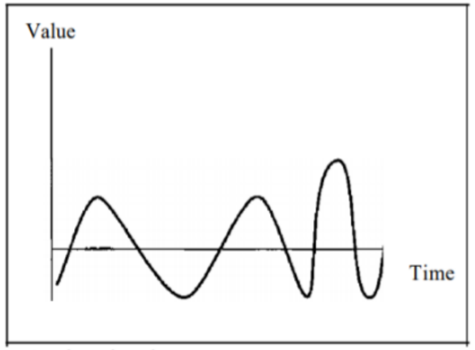
Both data and the signals that represent them can be either analog or digital in form.

2.1.1 Analog and Digital Data

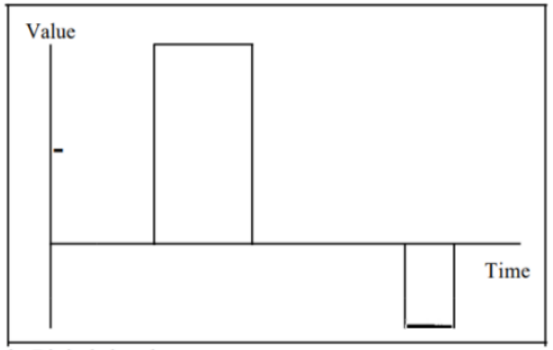
- The term analog data refers to information that is continuous. For example, an analog clock
- Digital data refers to information that has discrete states. For example, a digital clock

2.1.2 Analog and Digital Signals

- Like the data, signals can be either analog or digital.
- An analog signal has infinitely many levels of intensity over a period of time.
- A digital signal, on the other hand, can have only a limited number of defined values.



a. Analog signal



b. Digital signal

UNIT – II
ANALOG & DIGITAL

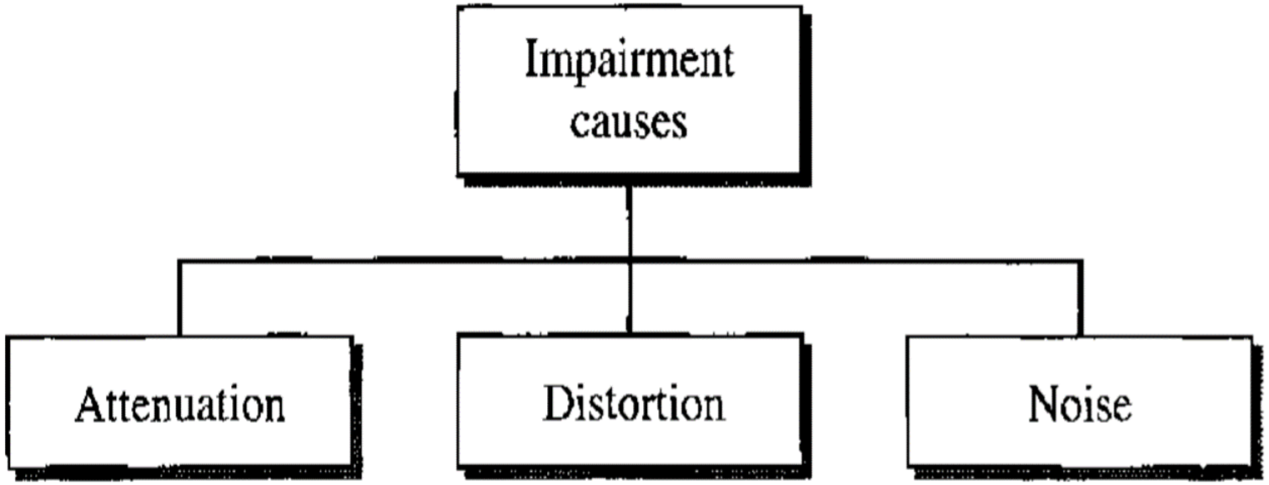
2.1.3 Periodic and Nonperiodic Signals

- Both analog and digital signals can take one of two forms: periodic or nonperiodic.
- Both analog and digital signals can be periodic or nonperiodic.
- In data communications, we commonly use periodic analog signals and nonperiodic digital signals.

- **A periodic signal** completes a pattern within a measurable time frame, called a period, and repeats that pattern over subsequent identical periods. The completion of one full pattern is called a cycle.
- **A nonperiodic signal** changes without exhibiting a pattern or cycle that repeats over time.

UNIT – II
TRANSMISSION IMPAIRMENT

- Signals travel through transmission media, which are not perfect.
- The imperfection causes signal impairment.
- The signal at the beginning of the medium is not the same as the signal at the end of the medium.
- Three causes of impairment are attenuation, distortion, and noise



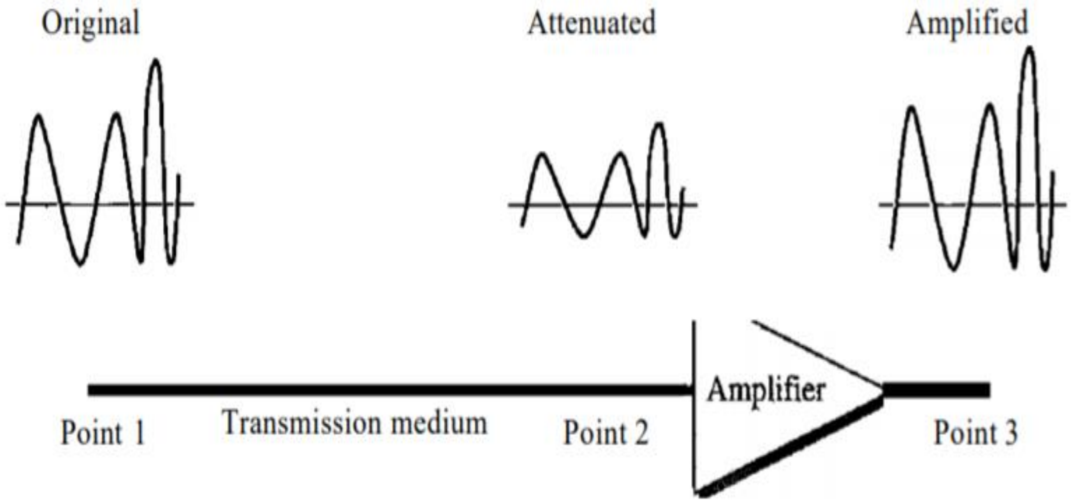
UNIT – II
TRANSMISSION IMPAIRMENT

2.2.1 Attenuation

- Attenuation means a loss of energy.
- When a signal, travels through a medium, it loses some of its energy in overcoming the resistance of the medium.
- That is why a wire carrying electric signals gets warm.
- To compensate for this loss, amplifiers are used to amplify the signal.

Decibel(dB): To show that a signal has lost or gained strength, engineers use the unit of the decibel.

- decibel is negative if a signal is attenuated
- positive if a signal is amplified.



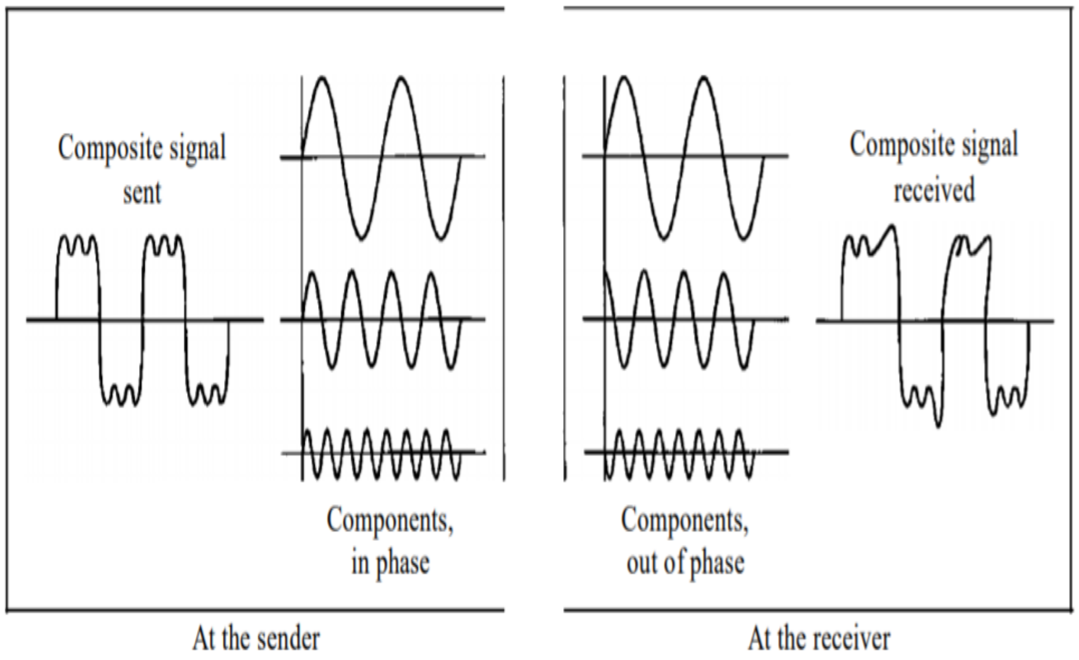
$$dB = 10 \log_{10} \frac{P_2}{P_1}$$

Variables PI and P2 are the powers of a signal at points 1 and 2, respectively

UNIT – II
TRANSMISSION IMPAIRMENT

2.2.2 Distortion

- Distortion means that the signal **changes its form or shape**.
- Each signal component has its own propagation speed through a medium and, therefore, its own delay in arriving at the final destination.
- Differences in delay may create a difference in phase if the delay is not exactly the same as the period duration.
- The shape of the composite signal is therefore not the same.



UNIT – II
TRANSMISSION IMPAIRMENT

2.2.3 Noise

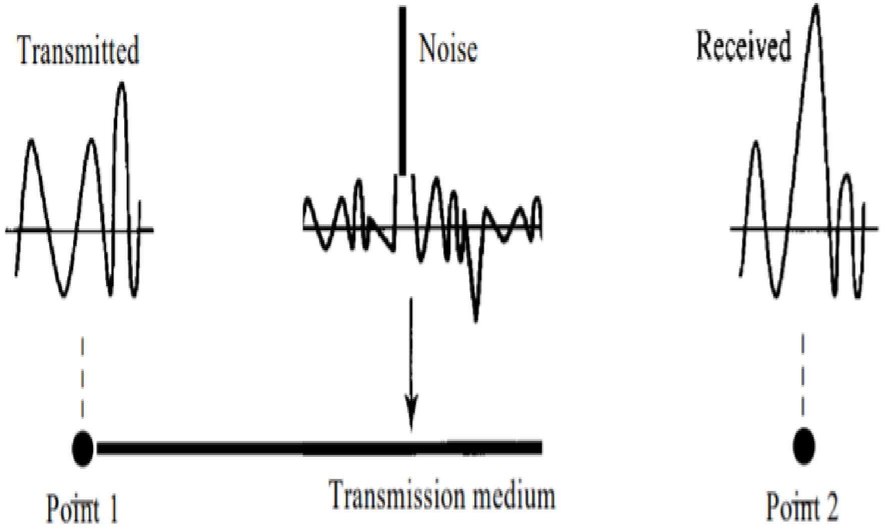
Noise is another cause of impairment. Several types of noise, such as

Thermal noise is the random motion of electrons in a wire which creates an extra signal not originally sent by the transmitter.

Induced noise comes from sources such as motors and appliances. These devices act as a sending antenna, and the transmission medium acts as the receiving antenna.

Crosstalk is the effect of one wire on the other. One wire acts as a sending antenna and the other as the receiving antenna.

Impulse noise is a spike (a signal with high energy in a very short time) that comes from power lines, lightning, and so on.



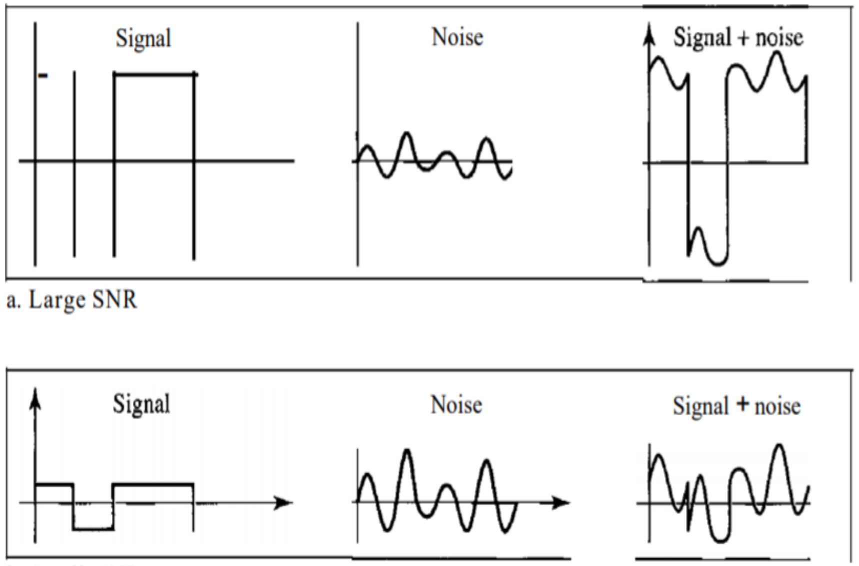
UNIT – II
TRANSMISSION IMPAIRMENT

Signal-to-Noise Ratio (SNR): The signal-to-noise ratio is defined as

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

- SNR is actually the ratio of what is wanted (signal) to what is not wanted (noise).
- A high SNR means the signal is less corrupted by noise;
- A low SNR means the signal is more corrupted by noise. Because SNR is the ratio of two powers, it is often described in decibel units, SNRdB, defined as

$$SNR_{dB} = 10\log_{10} SNR$$



a. Large SNR

b. Small SNR

DATA COMMUNICATIONS & NETWORKING

B.SC (COMPUTER SCIENCE) – III YEAR

SMCS52

By
Dr.N.N.Krishna Veni,
Assistant Professor,
Department of Computer Science,
Holy Cross Home Science College,
Thoothukudi

UNIT – II
PHYSICAL LAYER

2.3 DATA RATE LIMITS

- 2.3.1 Noiseless Channel: Nyquist Bit Rate
- 2.3.2 Noisy Channel: Shannon Capacity

2.4 PERFORMANCE

- 2.4.1 Bandwidth
- 2.4.2 Throughput
- 2.4.3 Latency (Delay)
- 2.4.4 Bandwidth-Delay Product
- 2.4.5 Jitter

UNIT – II
DATA RATE LIMITS

A very important consideration in data communications is how fast we can send data, in bits per second. over a channel

Data rate depends on **three factors**:

- The bandwidth available
- The level of the signals we use
- The quality of the channel (the level of noise)

Two theoretical formulas to calculate the data rate.

- Nyquist for a noiseless channel
- Shannon for a noisy channel.

2.3.1 Noiseless Channel: Nyquist Bit Rate

BitRate = 2 x bandwidth x log2 L

- In this formula, bandwidth is the bandwidth of the channel,
- L is the number of signal levels used to represent data, and
- BitRate is the bit rate in bits per second.
- Although the idea is theoretically correct, **practically there is a limit**.
- When we **increase the number of signal1eve1s, we impose a burden on the receiver**.
- In other words, increasing the levels of a signal reduces the reliability of the system

UNIT – II
DATA RATE LIMITS

2.3.2 Noisy Channel: Shannon Capacity

In reality, we cannot have a noiseless channel; the channel is always noisy.

In 1944, Claude Shannon introduced a formula, called the Shannon capacity, to determine the theoretical highest data rate for a noisy channel:

Capacity =bandwidth X log2 (1 +SNR)

- In this formula, bandwidth is the bandwidth of the channel,
- SNR is the signal-to-noise ratio,
- capacity is the capacity of the channel in bits per second.
- Note that in the Shannon formula there is no indication of the signal level, which means that no matter how many levels we have, we cannot achieve a data rate higher than the capacity of the channel.
- In other words, the formula defines a characteristic of the channel, not the method of transmission.

UNIT – II
PERFORMANCE

2.4.1 Bandwidth

Term bandwidth can be used in two different contexts with two different measuring values:

- **Bandwidth in Hertz**: Bandwidth in hertz is the range of **frequencies** contained in a composite signal or the range of frequencies a channel can pass.
- **Bandwidth in Bits per Seconds**: The term bandwidth can also refer to the **number of bits** per second that a channel, a link, or even a network can transmit.

Relationship: Basically, an increase in bandwidth in hertz means an increase in bandwidth in bits per second.

2.4.2 Throughput

- The throughput is a measure of how fast we can actually send data through a network.
- Although, at first glance, bandwidth in bits per second and throughput seem the same, they are different.
- A link may have a bandwidth of B bps, but we can only send T bps through this link with T always less than B.

Ex: Road - Cars

UNIT – II
PERFORMANCE

2.4.3 Latency (Delay) : The latency or delay defines how long it takes for an entire message to completely arrive at the destination.

Latency =propagation time +transmission time +queuing time + processing delay

Propagation Time

- Propagation time measures the time required for a bit to travel from the source to the destination.
- The propagation speed of electromagnetic signals depends on the medium and on the frequency of the signal
- **For example,** in a vacuum, light is propagated with a speed of 3 x 108 mfs. It is lower in air; it is much lower in cable.

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

UNIT – II
PERFORMANCE

Transmission Time:

- In data communications we don't send just 1 bit, we send a message.
- The first bit may take a time equal to the propagation time to reach its destination; the last bit also may take the same amount of time.
- The time required for transmission of a message depends on the size of the message and the bandwidth of the channel.

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

Queuing Time:

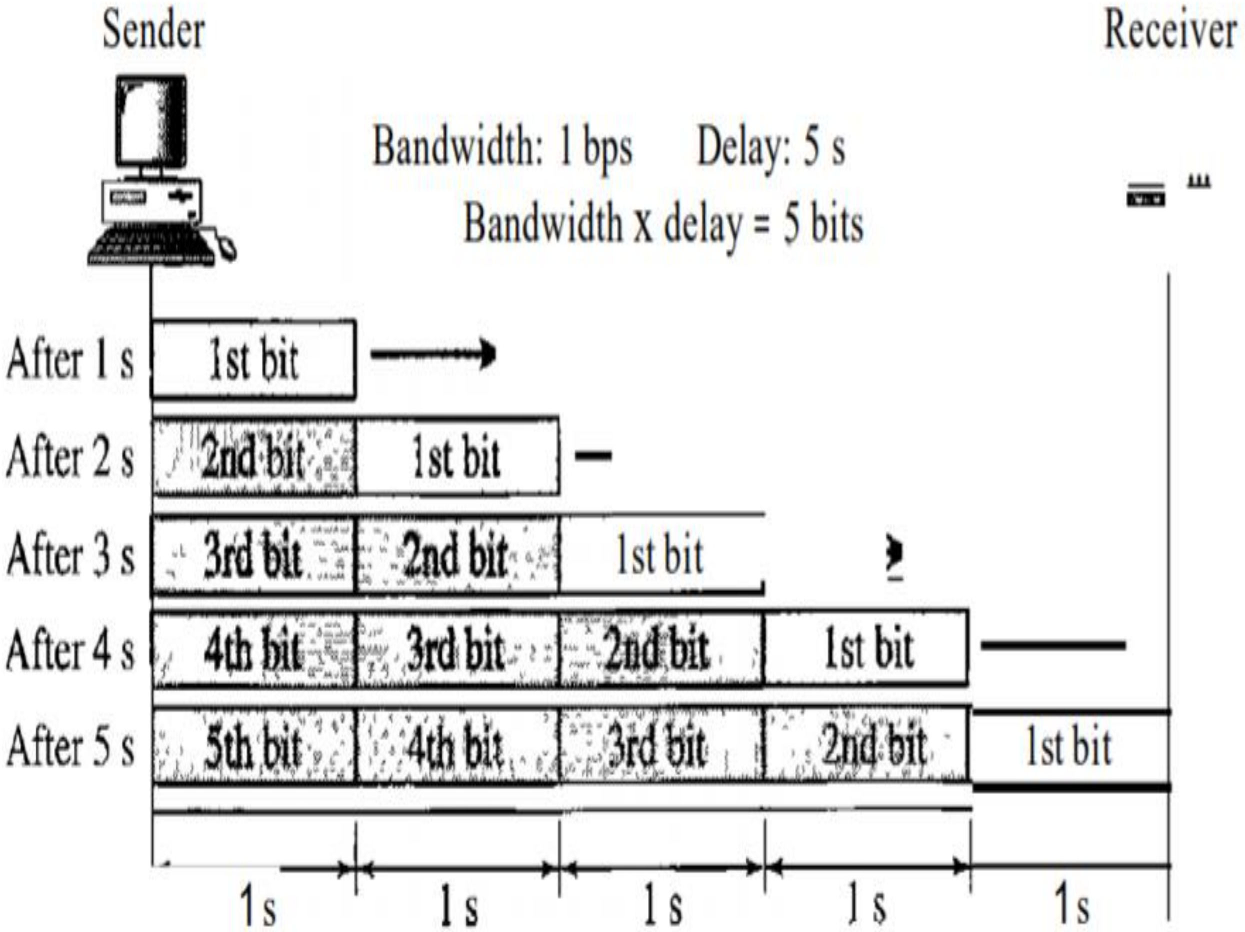
- The third component in latency is the queuing time, the time needed for each intermediate or end device to hold the message before it can be processed.
- The queuing time is not a fixed factor; it changes with the load imposed on the network.
- When there is heavy traffic on the network, the queuing time increases.

UNIT – II
PERFORMANCE

2.4.4 Bandwidth-Delay Product

Bandwidth and delay are two performance metrics of a link. However, what is very important in data communications is the product of the two, the bandwidth-delay product. Let us elaborate on this issue, using **two hypothetical cases** as examples.

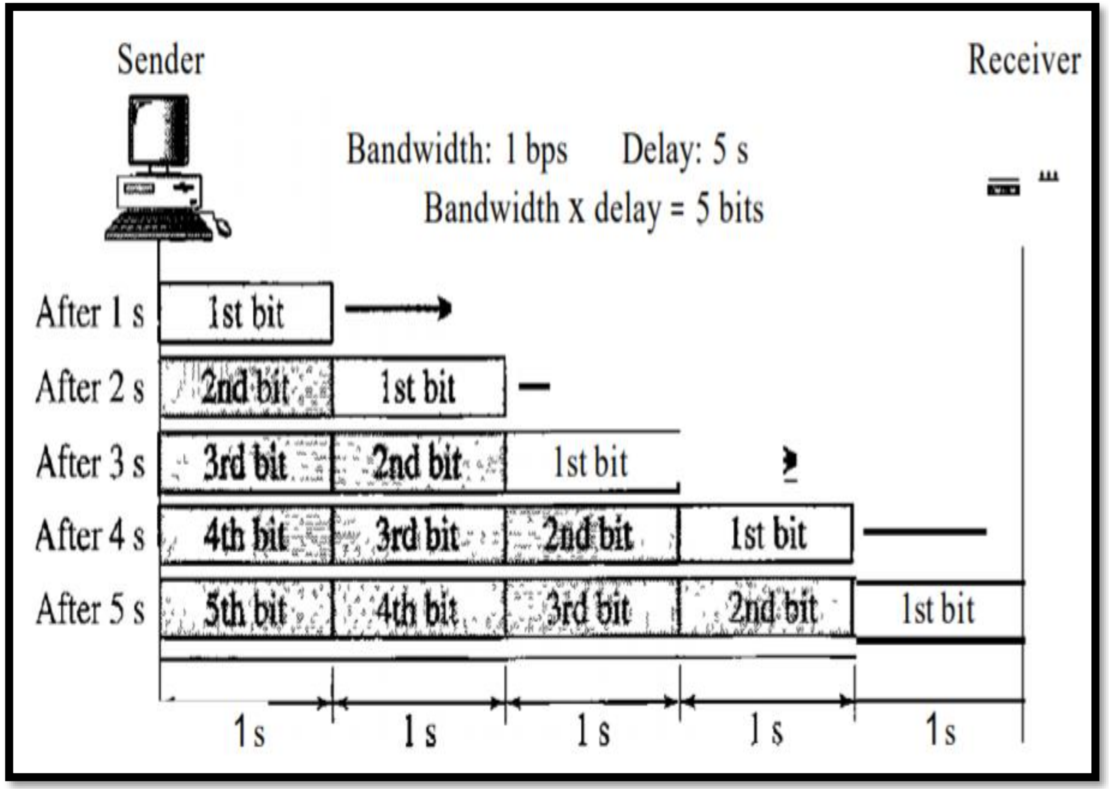
Case 1. Figure 2.7 shows case 1.



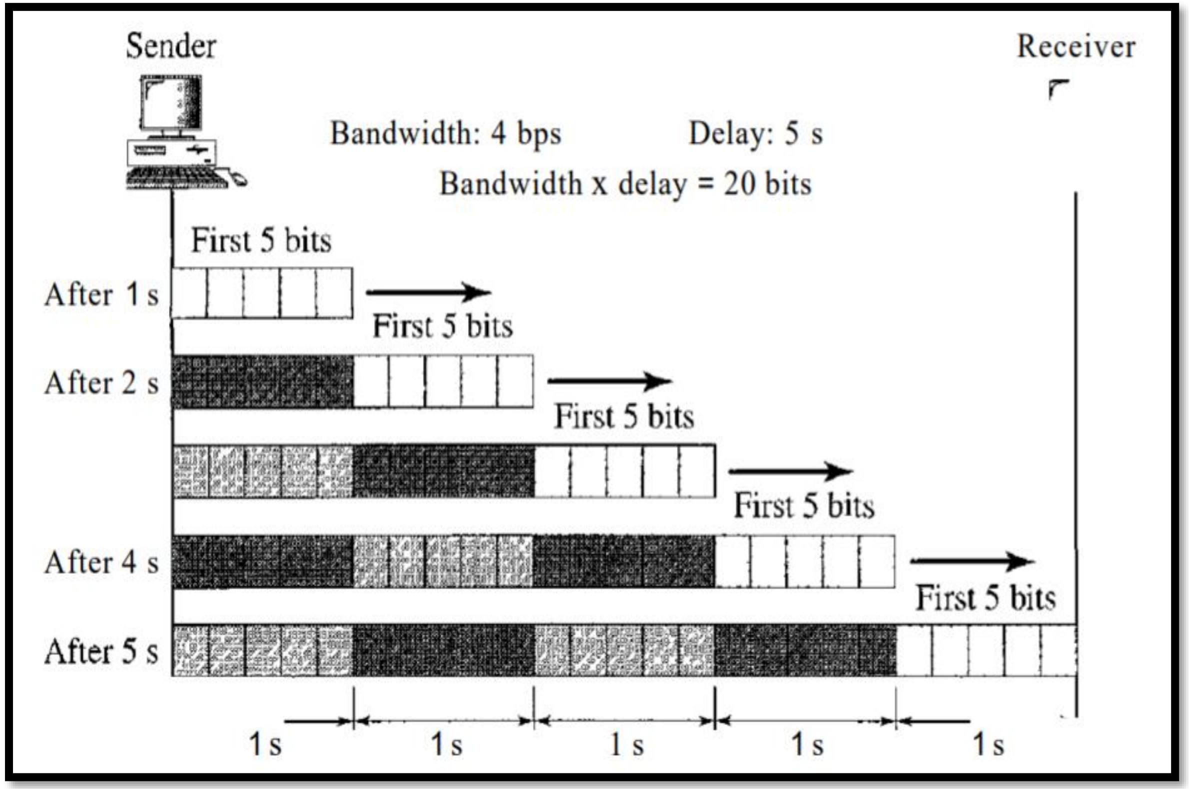
UNIT – II
PERFORMANCE

2.4.4 Bandwidth-Delay Product: Bandwidth and delay are two performance metrics of a link. Let us elaborate on this issue, using **two hypothetical cases** as examples.

Case 1



Case 2



UNIT – II
PERFORMANCE

2.4.5 Jitter

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

DATA COMMUNICATIONS & NETWORKING

B.SC (COMPUTER SCIENCE) – III YEAR

SMCS52

By
Dr.N.N.Krishna Veni,
Assistant Professor,
Department of Computer Science,
Holy Cross Home Science College,
Thoothukudi

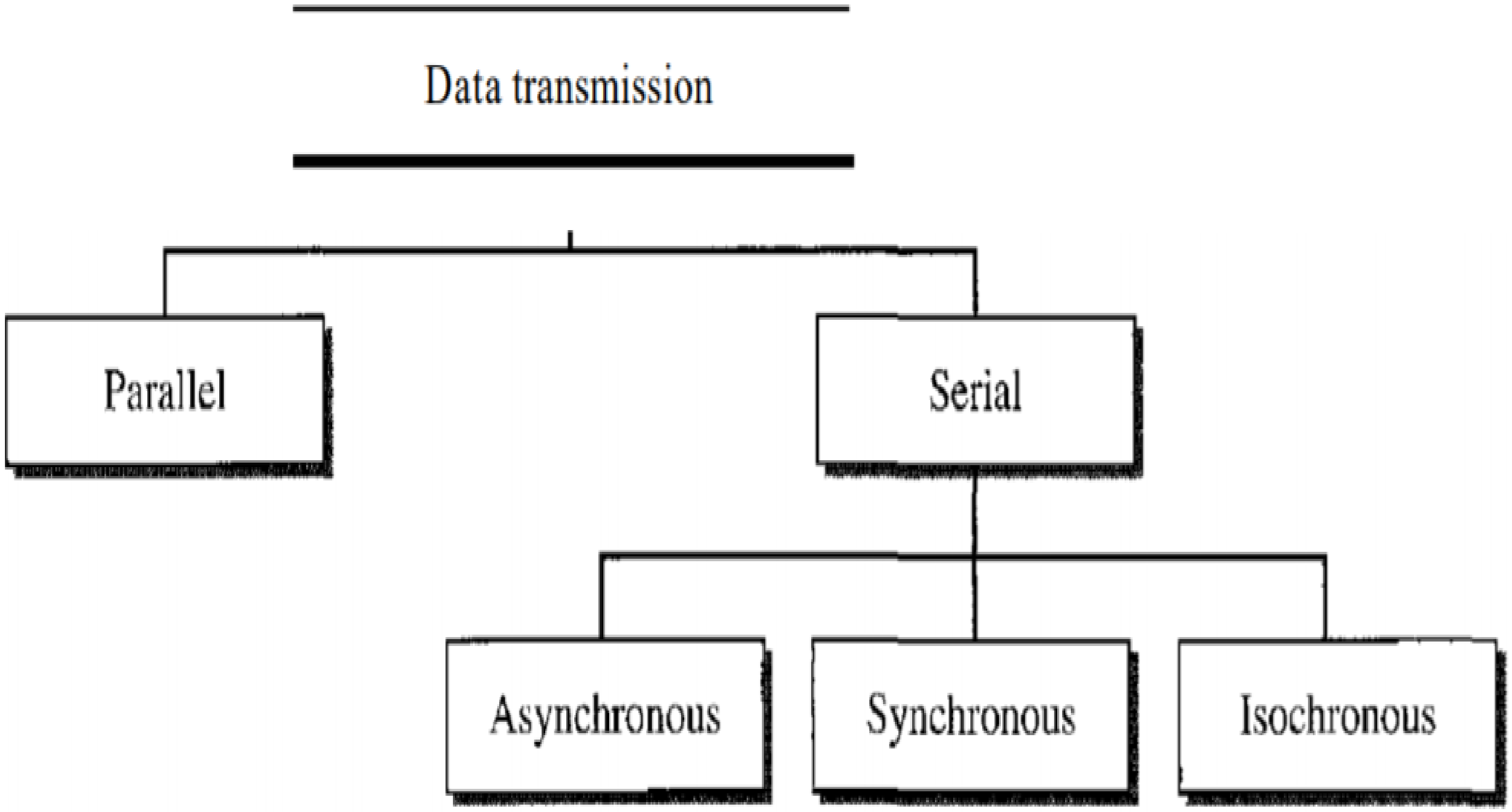
UNIT – II
PHYSICAL LAYER

2.5 TRANSMISSION MODES

2.5.1 Parallel Transmission

2.5.2 Serial Transmission

UNIT – II
TRANSMISSION MODES



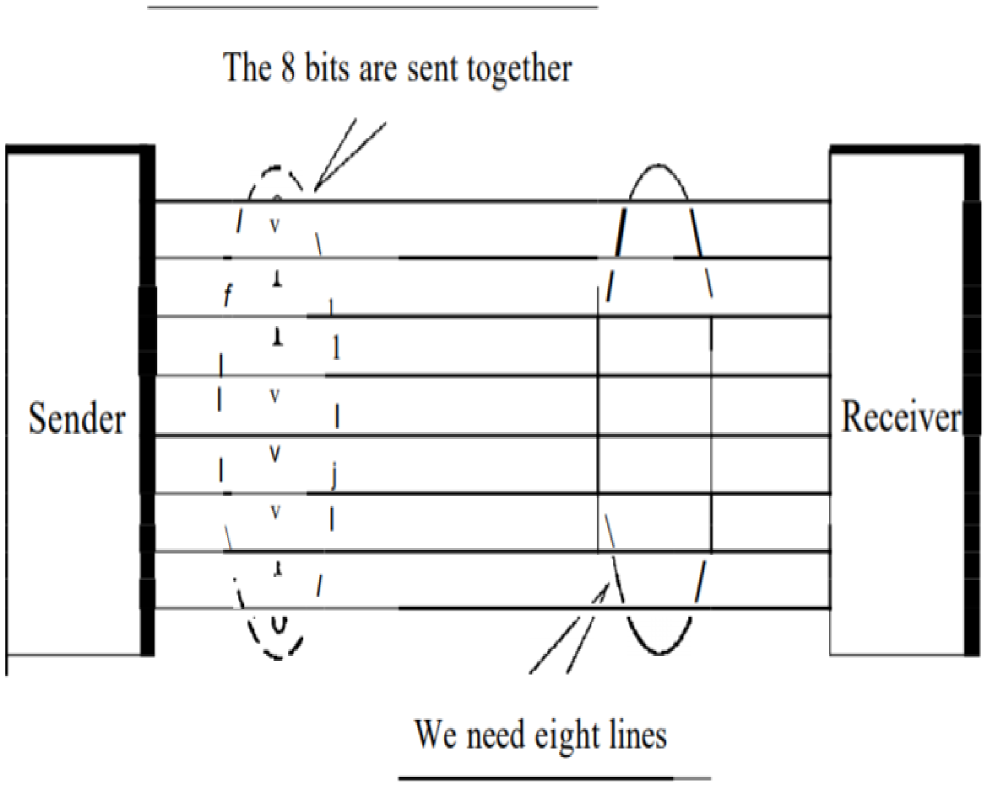
UNIT – II
TRANSMISSION MODES

2.5.1 Parallel Transmission

- Computers produce and consume data in groups of bits.
- Use n wires to send n bits at one time.
- That way each bit has its own wire, and all n bits of one group can be transmitted with each clock tick

Advantage: Main advantage of parallel transmission is speed.

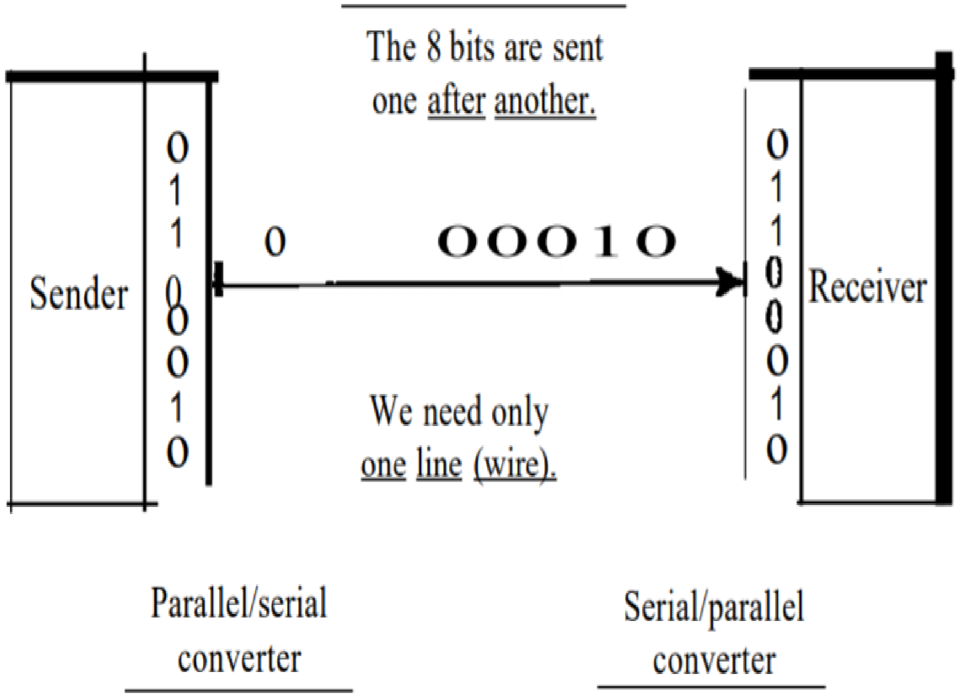
Disadvantage: Significant disadvantage is cost. Because this is expensive, parallel transmission is usually limited to short distances.



UNIT – II
TRANSMISSION MODES

2.5.2 Serial Transmission

- In serial transmission one bit follows another.
- Serial transmission occurs in one of three ways:
  - asynchronous,
  - synchronous, and
  - isochronous.



Advantage: It needs only one communication channel, reduces the cost of transmission

UNIT – II
TRANSMISSION MODES

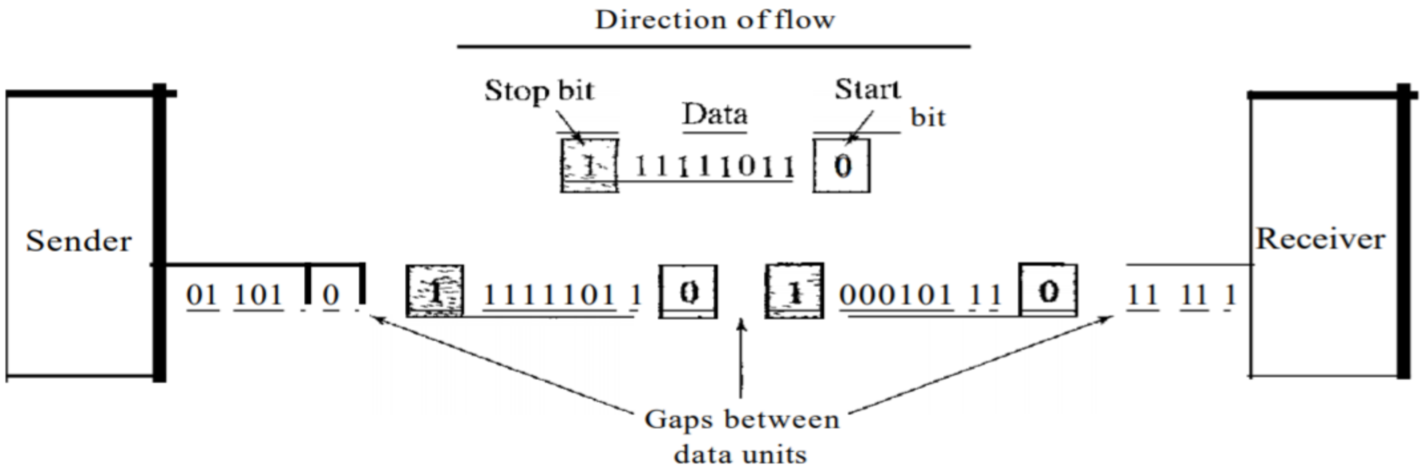
Asynchronous Transmission

- Asynchronous transmission is so named because the **timing of a signal is unimportant**.
- Instead, information is received and translated by **agreed upon patterns**.
- Patterns are based on grouping **the bit stream into bytes**.
- Each group, **usually 8 bits**, is sent along the link as a unit.
- The sending system handles each **group independently**.
- Without synchronization, the **receiver cannot use timing to predict** when the next group will arrive
- To alert the arrival of a new group**, an extra bit is added** to the beginning & end(1/more bits) of each byte.
- Beginning bit, usually a 0, is called the **start bit**. End bit, Usually 1s, are called **stop bits**
- By this method, each **byte is increased** in size to at least 10 bits, of which **8 bits is information** and **2 bits or more are signals to the receiver.**
- In addition, the transmission of each byte may then be **followed by a gap of** varying duration

UNIT – II
TRANSMISSION MODES

- At the byte level, the sender and receiver **do not have to be synchronized**.
- But within each byte, the receiver must still be **synchronized with the incoming bit stream**.
- When the receiver detects a start bit, it **sets a timer and begins counting** bits as they come in. After n bits, the receiver looks for a stop bit. As soon as it detects the stop bit, it waits until it detects the next start bit.

Advantages: It is cheap and effective, that make it an attractive choice for situations such as low-speed communication.



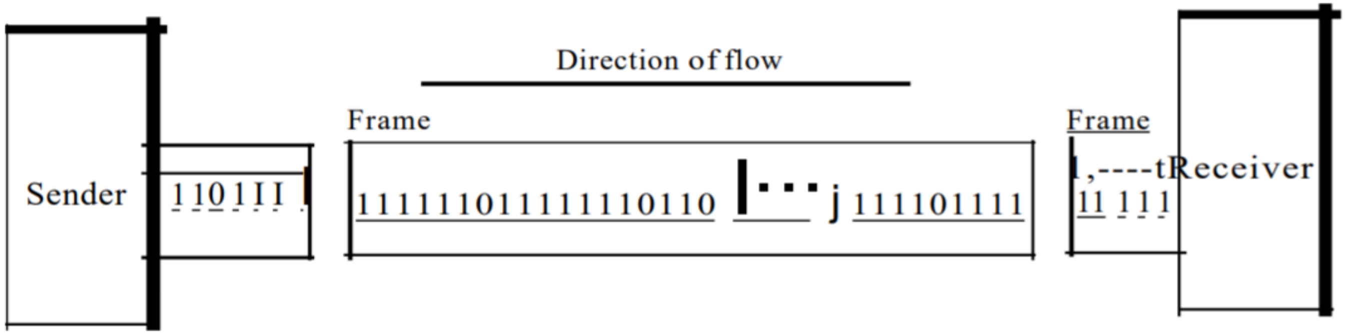
UNIT – II
TRANSMISSION MODES

Synchronous Transmission

- In synchronous transmission, the bit stream is combined into longer "**frames**," which may contain **multiple bytes**.
- Each byte, is sent onto the transmission **link without a gap** between it and the next one.
- It is left to the receiver to separate the bit stream into bytes for decoding purposes.
- If the sender wishes to send data in separate bursts, the gaps between bursts must be filled with a **special sequence of 0s and 1s** that means idle.
- The receiver counts the bits as they arrive and groups them in 8-bit units.
- The accuracy of the received information is completely dependent on the ability of the receiving device to keep an accurate count of the bits as they come in.

Advantage: Main advantage of synchronous transmission is speed.

UNIT – II
TRANSMISSION MODES



Isochronous

- In real-time audio and video, in which uneven delays between frames are not acceptable, synchronous transmission fails.
- For example, TV images are broadcast at the rate of **30 images per second**; they must be viewed at the same rate. If each image is sent by using one or more frames, there should be **no delays** between frames.
- For this type of application, synchronization **between characters is not enough; the entire stream of bits must be synchronized.**
- The isochronous transmission guarantees that the data arrive at a fixed rate.

DATA COMMUNICATIONS & NETWORKING

B.SC (COMPUTER SCIENCE) – III YEAR

SMCS52

By
Dr.N.N.Krishna Veni,
Assistant Professor,
Department of Computer Science,
Holy Cross Home Science College,
Thoothukudi

UNIT – II
PHYSICAL LAYER

2.6 MULTIPLEXING

2.6.1 Frequency-Division Multiplexing

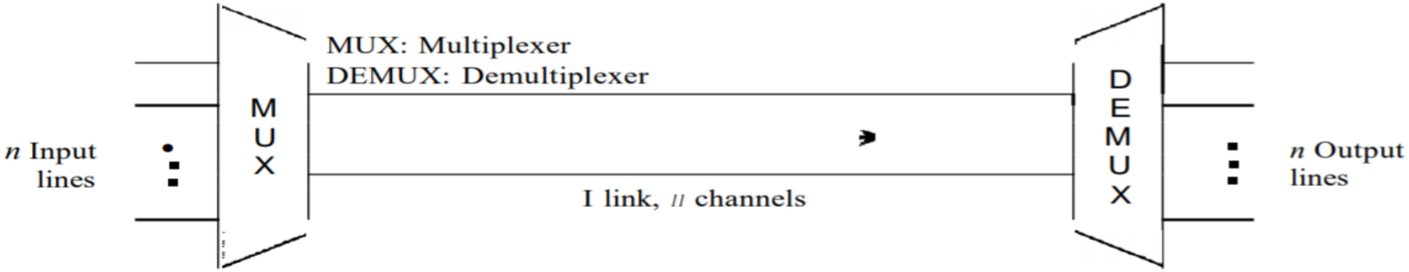
2.6.2 Wavelength-Division Multiplexing

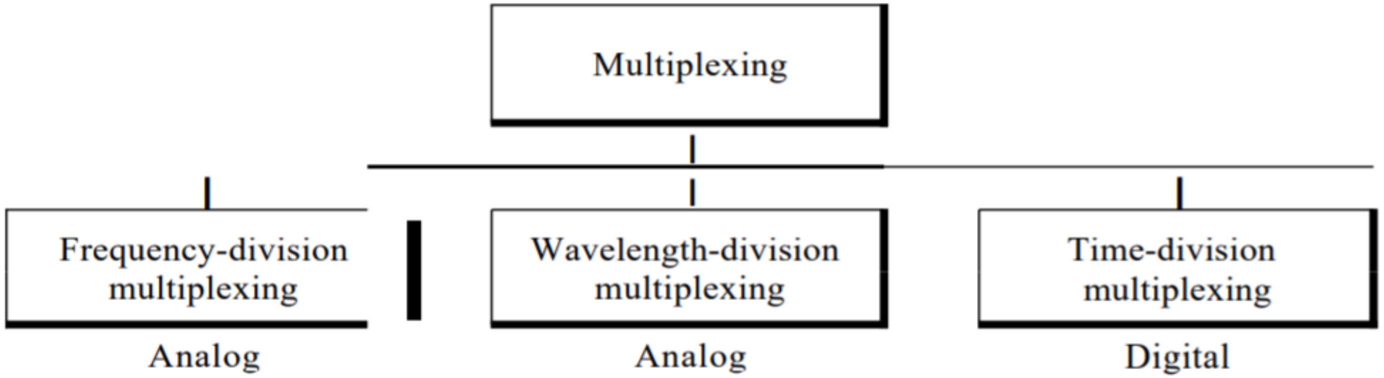
2.6.3 Synchronous Time-Division Multiplexing

2.6.4 Statistical Time-Division Multiplexing

UNIT – II
MULTIPLEXING

- Multiplexing is the set of techniques that allows the simultaneous transmission of multiple signals across a single data link.
- In a multiplexed system, n lines share the bandwidth of one link.

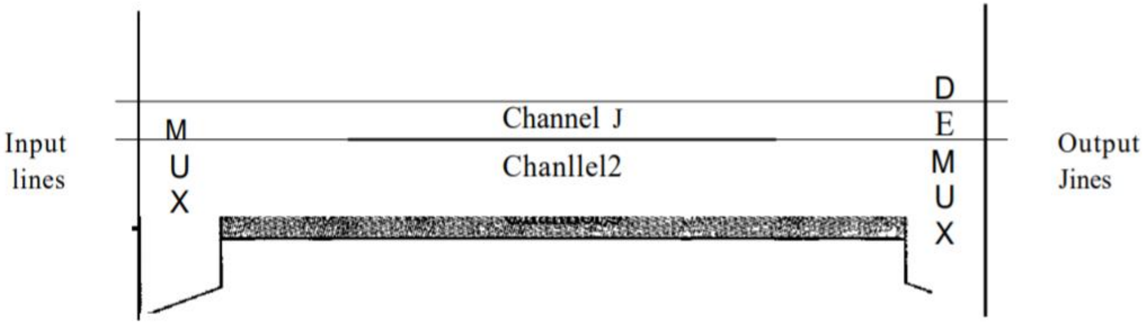



Multiplexing
Frequency-division multiplexing
Analog
Wavelength-division multiplexing
Analog
Time-division multiplexing
Digital

UNIT – II
MULTIPLEXING

2.6.1 Frequency-Division Multiplexing

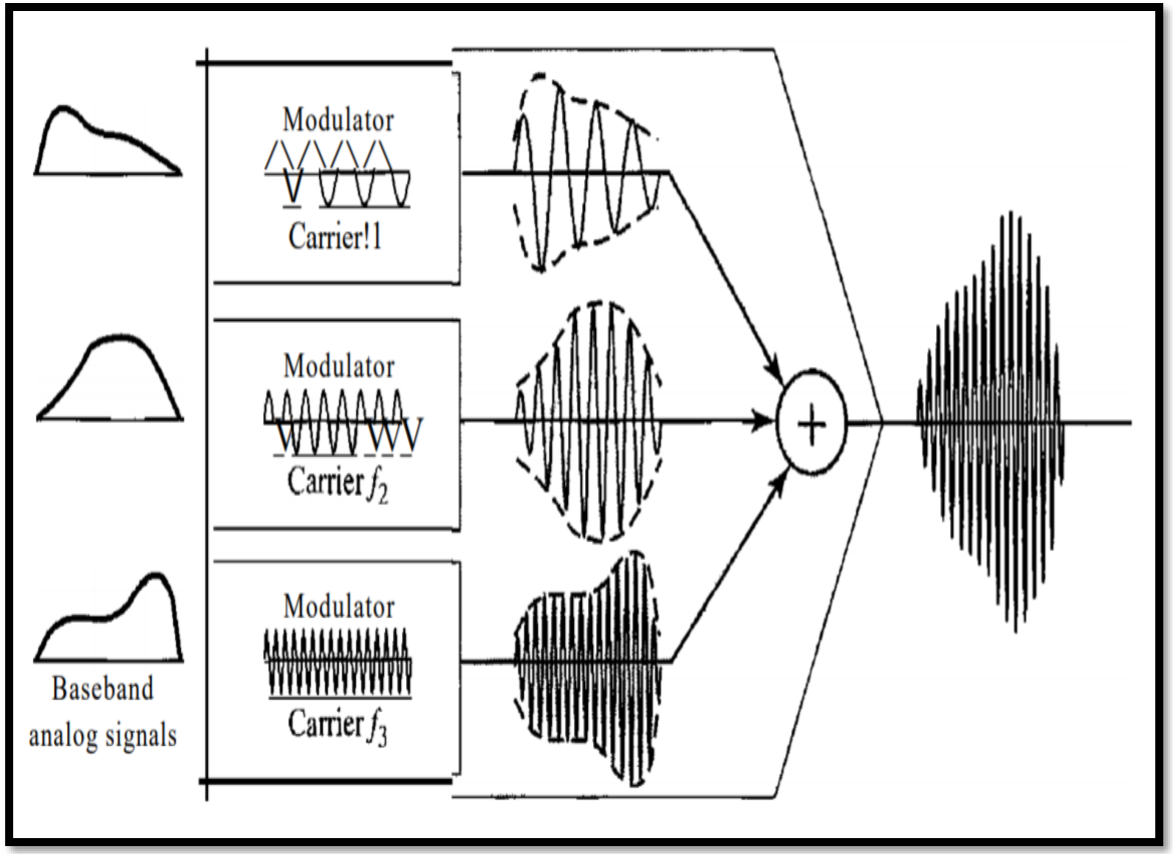
- Frequency-division multiplexing (FDM) is an analog technique that can be applied when the bandwidth of a link (in hertz) is greater than the combined bandwidths of the signals to be transmitted.
- In FDM, signals generated by each sending device modulate different **carrier frequencies**.
- These modulated signals are then **combined into a single composite signal** that can be transported by the link.
- Channels can be separated by strips of **unused bandwidth-guard-bands to** prevent signals from overlapping.
- In addition, carrier frequencies must not interfere with the original data frequencies.

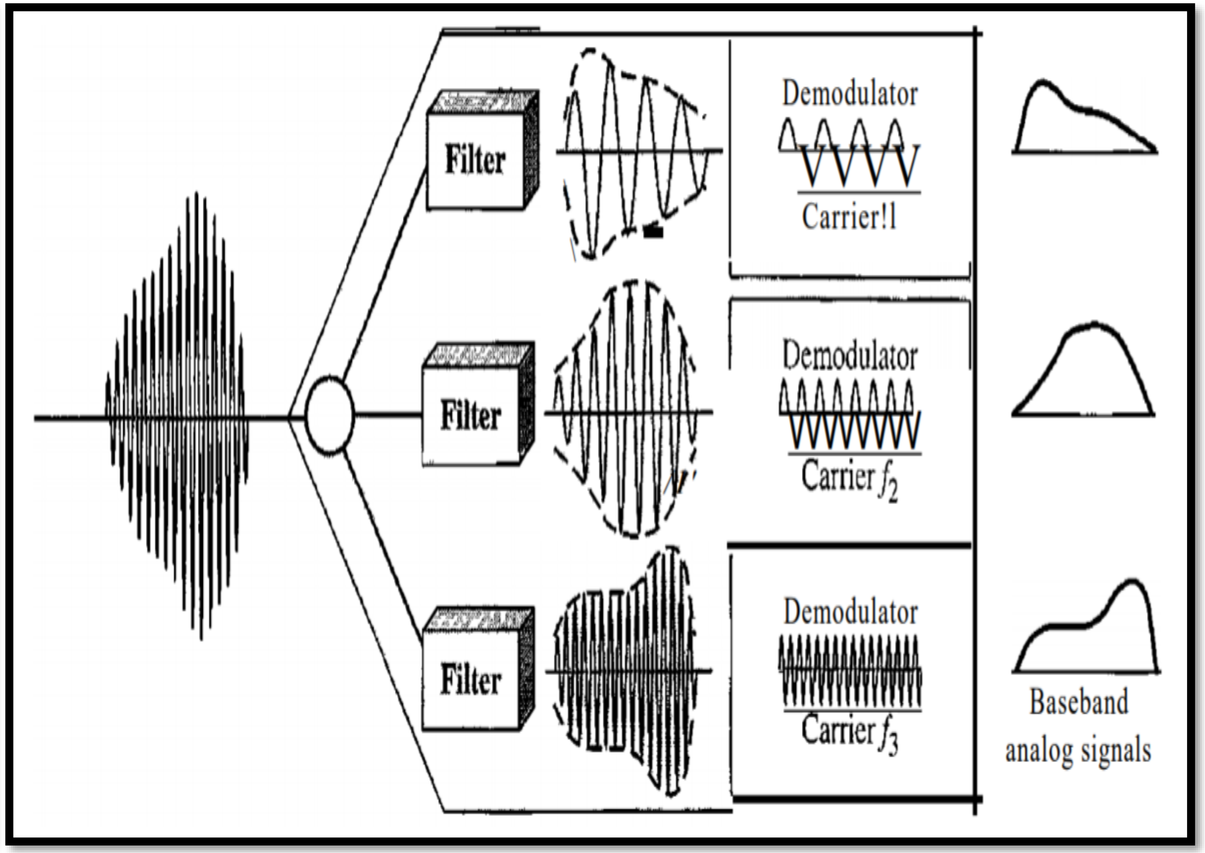


UNIT – II
MULTIPLEXING

Multiplexing Process

Demultiplexing Process

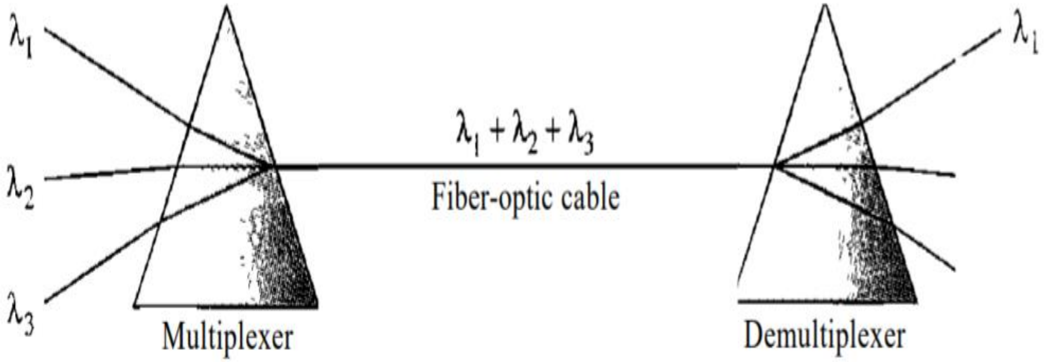




UNIT – II
MULTIPLEXING

2.6.2 Wavelength-Division Multiplexing

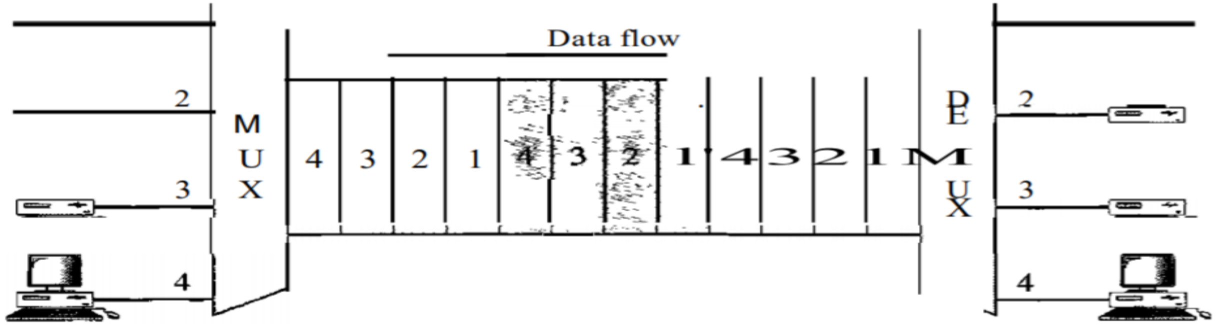
- It is designed to use the high-data-rate capability of fiber-optic cable.
- Using a fiber-optic cable for one single line wastes the available bandwidth. Multiplexing allows us to combine several lines into one.
- Although WDM technology is very complex, the basic idea is very simple.
- We want to combine multiple light sources into one single light at the multiplexer and do the reverse at the demultiplexer.
- The combining and splitting of light sources are easily handled by a prism.



UNIT – II
MULTIPLEXING

2.6.3 Time-Division Multiplexing

- It is a digital process that allows several connections to share the high bandwidth of a line by sharing the time. Each connection occupies a portion of time in the link.



- This means that all the digital data in a message from source 1 always go to one specific destination.
- The delivery is fixed and unvarying, unlike switching.
- This does not mean that the sources cannot produce analog data; analog data can be changed to digital data, and then multiplexed by using TDM.
- We can divide TDM into **two different schemes**: synchronous and statistical.

UNIT – II
MULTIPLEXING

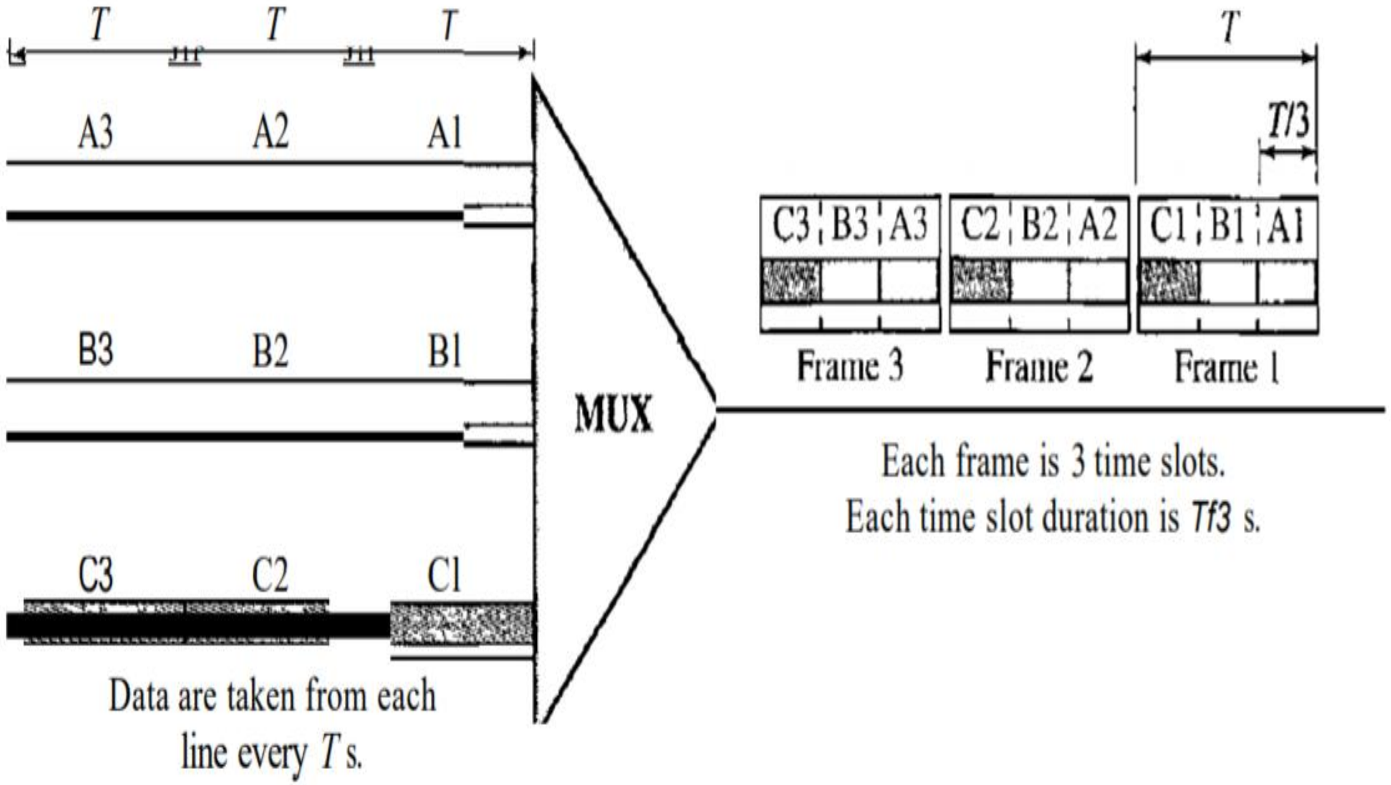
SYNCHRONOUS TIME-DIVISION MULTIPLEXING

In synchronous TDM, each input connection has an allotment in the output even if it is not sending data

- **Time Slots & Frames**
- **Interleaving**
- **Empty Slots**
- **Data Rate Management**
- **Frame Synchronizing**

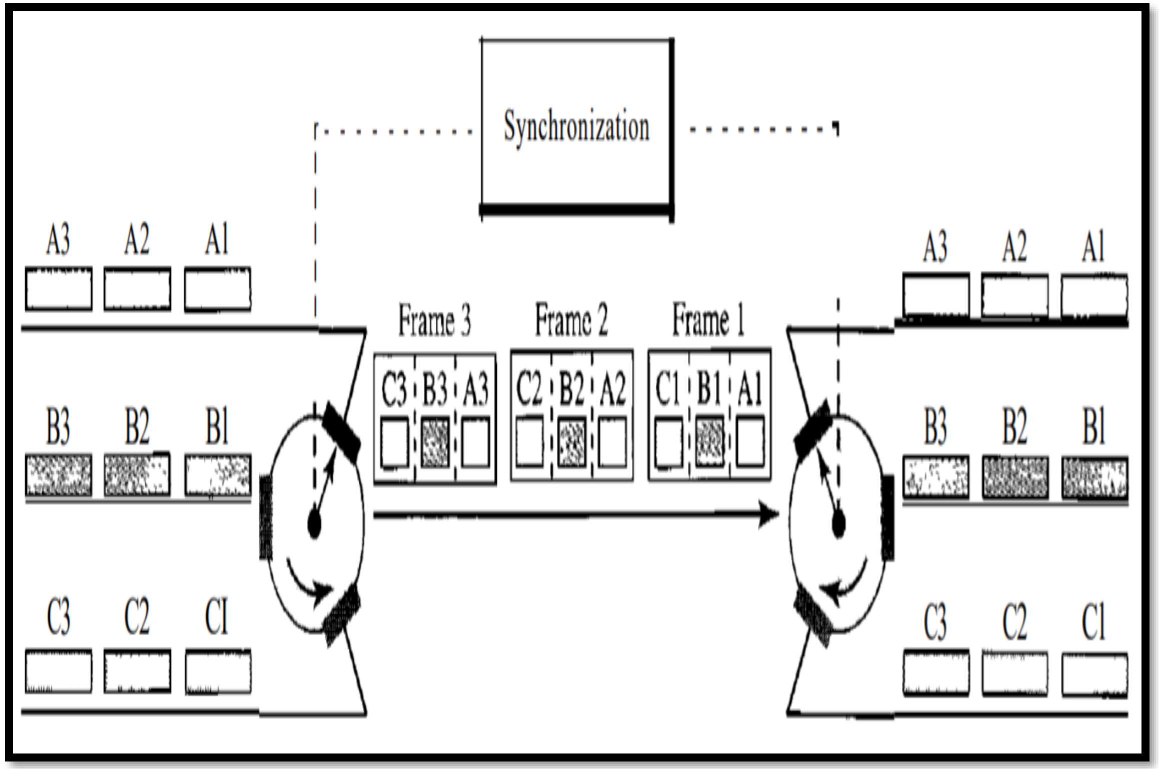
UNIT – II
MULTIPLEXING

Time Slots & Frames

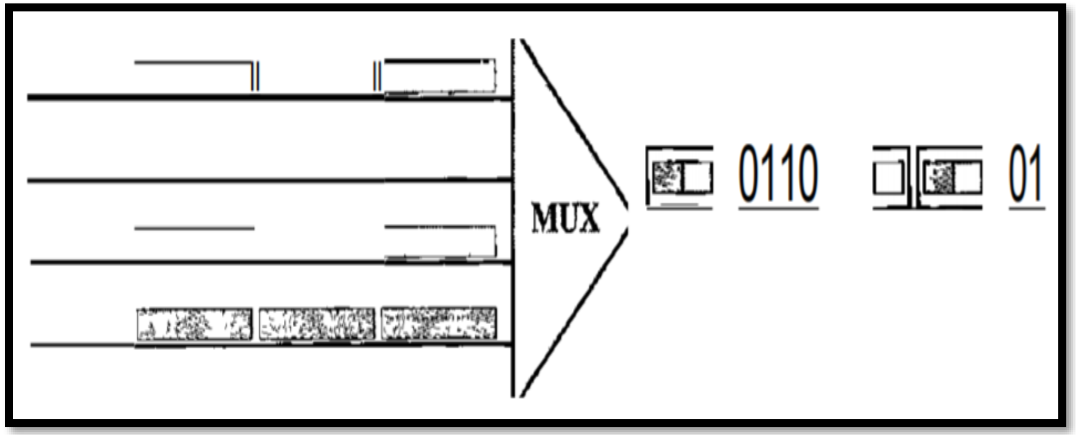

T
T
T
A3
A2
A1
B3
B2
B1
C3
C2
C1
Data are taken from each line every T s.
MUX
T
T/3
C3 B3 A3
C2 B2 A2
C1 B1 A1
Frame 3
Frame 2
Frame 1
Each frame is 3 time slots.
Each time slot duration is Tf3 s.

UNIT – II
MULTIPLEXING

Interleaving



Empty Slot

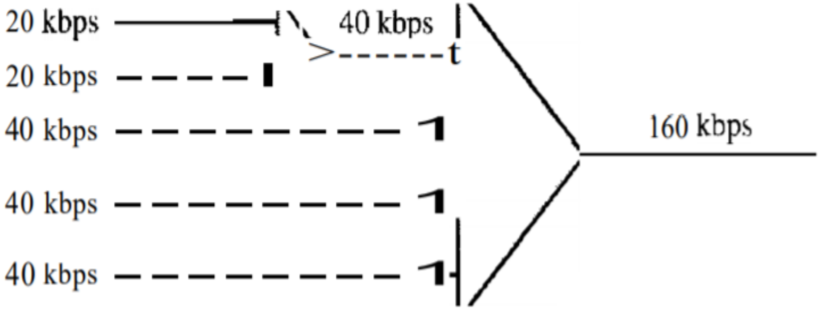


UNIT – II
MULTIPLEXING

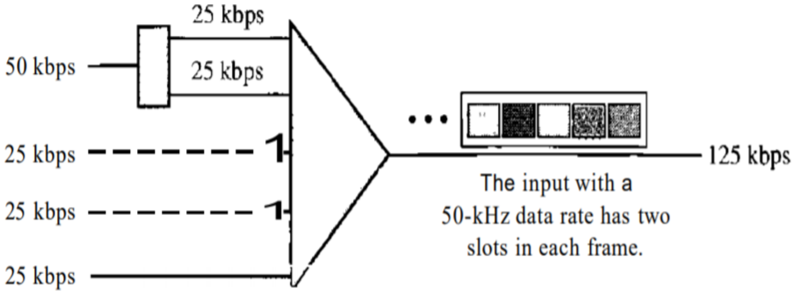
Data Rate Management

One problem with TDM is how to handle a disparity in the input data rates. In all our discussion so far, we assumed that the data rates of all input lines were the same. However, if data rates are not the sam, we call these three strategies

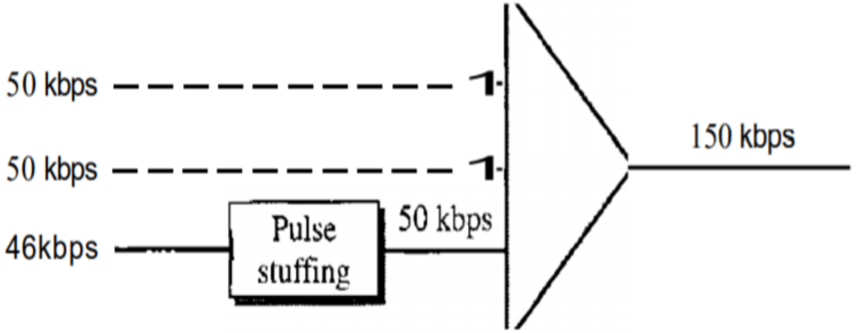
Multilevel Multiplexing



Multiple-Slot Allocation



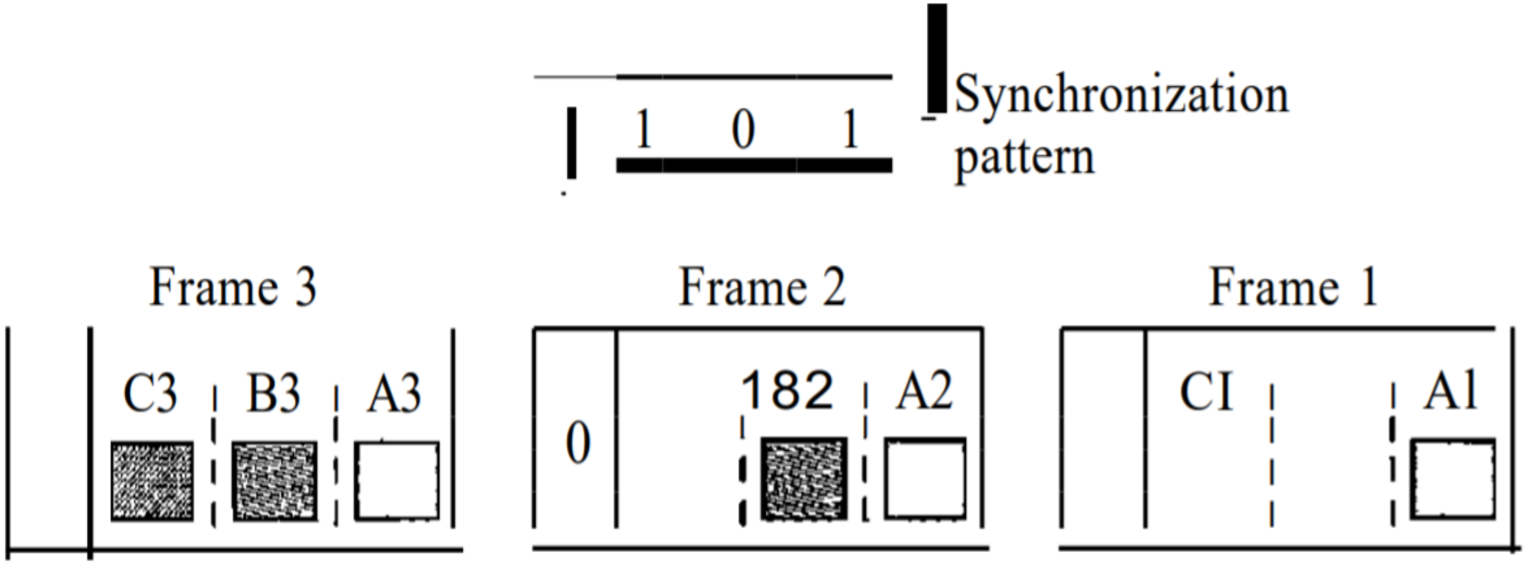
The input with a 50-kHz data rate has two slots in each frame.



Pulse Stuffing

UNIT – II
MULTIPLEXING

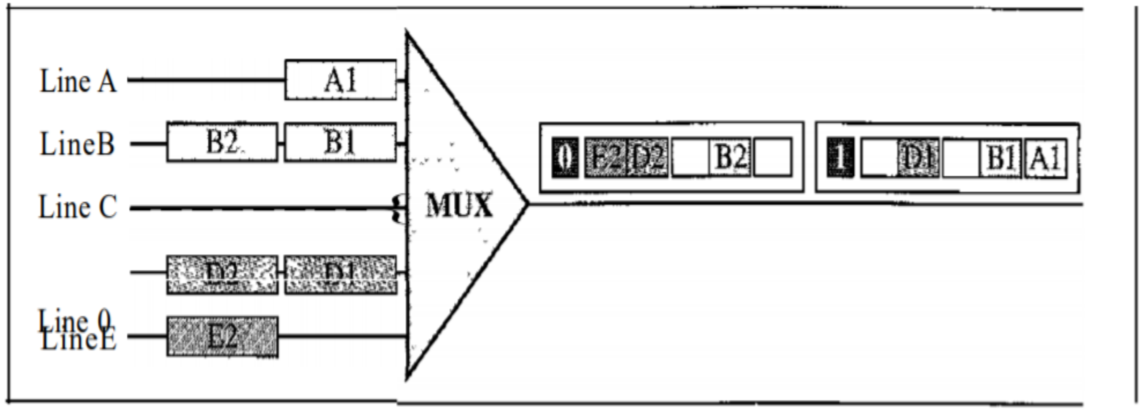
Frame Synchronizing



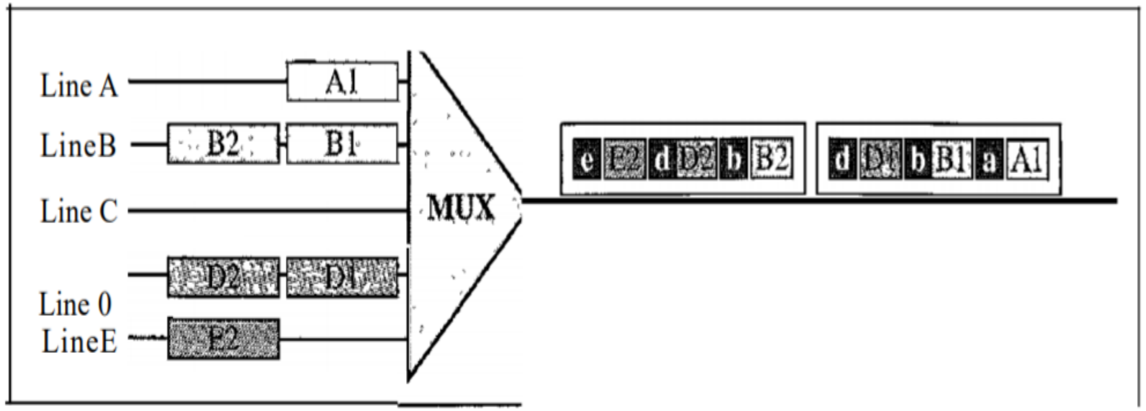
UNIT – II
MULTIPLEXING

2.6.5 Statistical Time-Division Multiplexing

- In synchronous TDM, each input has a reserved slot in the output frame. This can be inefficient if some input lines have no data to send.
- In statistical time-division multiplexing, slots are dynamically allocated to improve bandwidth efficiency.
- The multiplexer checks each input line in round robin fashion; it allocates a slot for an input line if the line has data to send; otherwise, it skips the line and checks the next line.
- **Slot Size**
- **No Synchronization**
- **Bit Bandwidth**



a. Synchronous TDM



b. Statistical TDM

DATA COMMUNICATIONS & NETWORKING

B.SC (COMPUTER SCIENCE) – III YEAR

SMCS52

By
Dr.N.N.Krishna Veni,
Assistant Professor,
Department of Computer Science,
Holy Cross Home Science College,
Thoothukudi

UNIT – III
SWITCHING

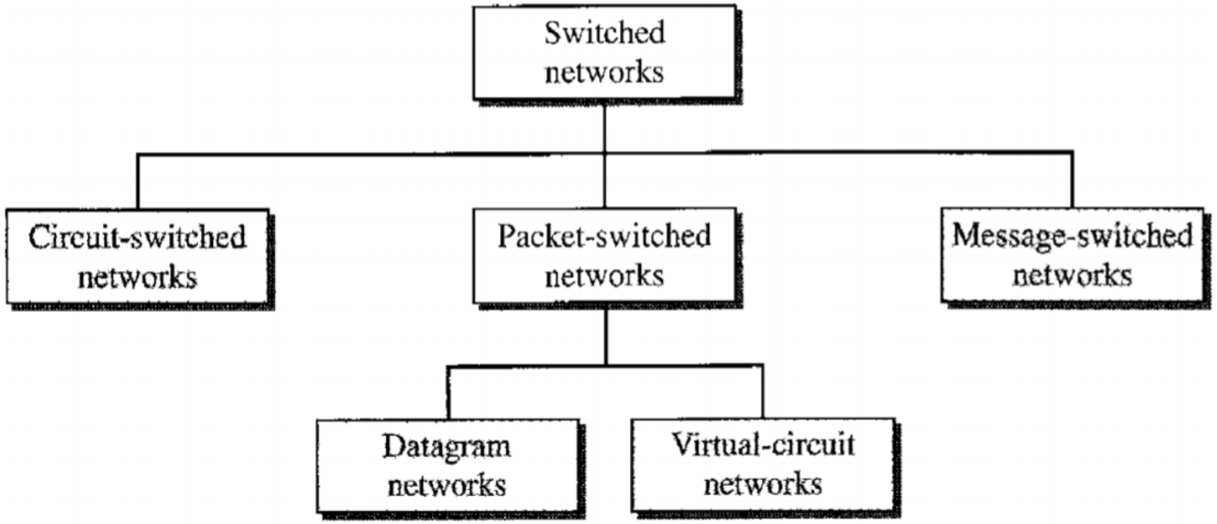
WHAT IS SWITCHING?

3.1 CIRCUIT-SWITCHED NETWORKS

- 3.1.1 Three Phases
- 3.1.2 Efficiency
- 3.1.3 Delay
- 3.1.4 Circuit-Switched Technology in Telephone Networks

UNIT – III
SWITCHING

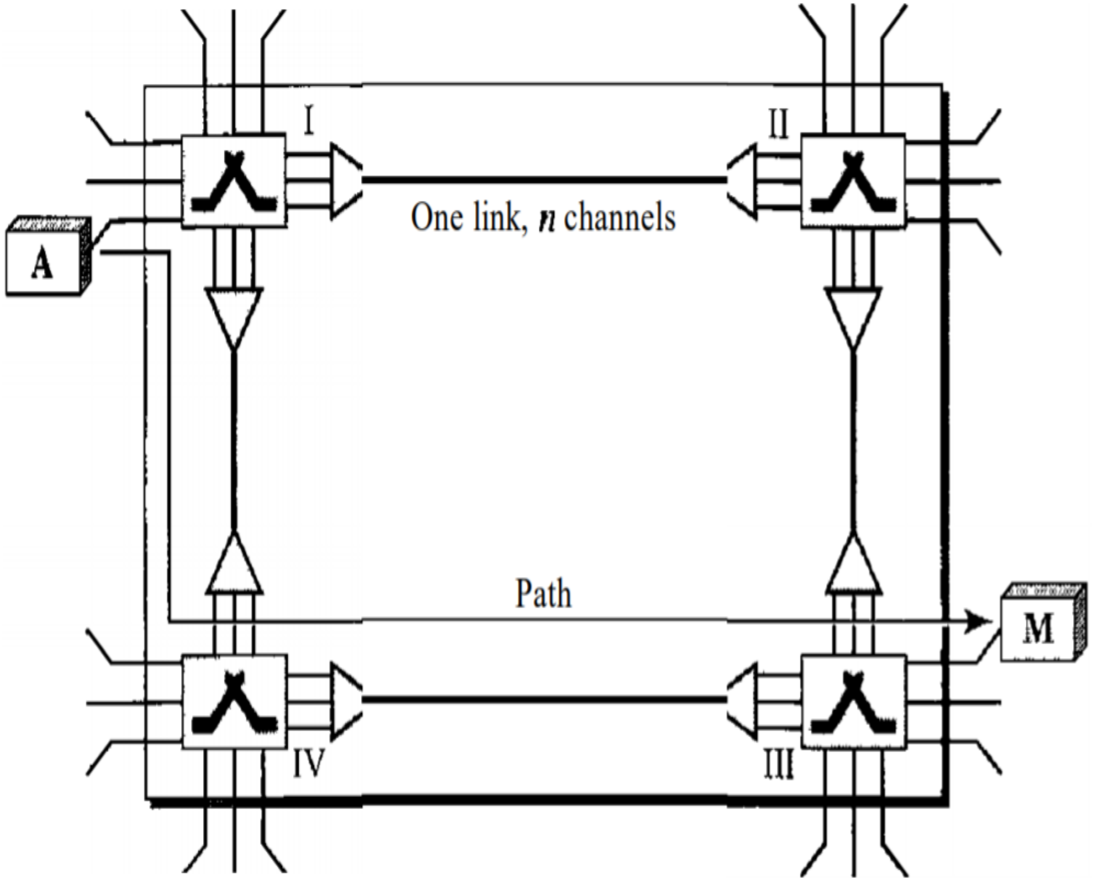
- A network is a **set of connected devices**. Whenever we have multiple devices, we have the problem of how to connect them to make one-to-one communication possible.
- A better solution is switching. A switched network consists of a **series of interlinked nodes**, called switches.
- Switches are devices capable of **creating temporary connections** between two or more devices linked to the switch.
- In a switched network, some of these nodes are connected to the end systems. Others are used only for routing.



UNIT – III
SWITCHING

3.1 CIRCUIT-SWITCHED NETWORKS

- A circuit-switched network consists of a set of switches connected by physical links.
- A connection between two stations is a dedicated path made of one or more links.
- However, each connection uses only one dedicated channel on each link.
- Circuit switching takes place at **the physical layer**.
- Before starting communication, the stations must make a **reservation for the resources till end**
- Data transferred between the two stations are **not packetized**. The data are a continuous flow sent by the source station and received by the destination station, although there may be periods of silence.
- There is **no addressing** involved during data transfer.



UNIT – III
SWITCHING

3.1.1 Three Phases

The actual communication in a circuit-switched network requires three phases:

Connection setup – Request & Acknowledge

Data transfer – Data sent

Connection teardown – Resources released

3.1.2 Efficiency

- It can be argued that circuit-switched networks are **not as efficient** as the other two types of networks because resources are allocated during the entire duration of the connection.
- These resources are unavailable to other connections.
- **In a telephone network**, people normally terminate the communication when they have finished their conversation.
- However, **in computer networks**, a computer can be connected to another computer even if there is no activity for a long time. In this case, allowing resources to be dedicated means that other connections are deprived.

UNIT – III
SWITCHING

3.1.3 Delay

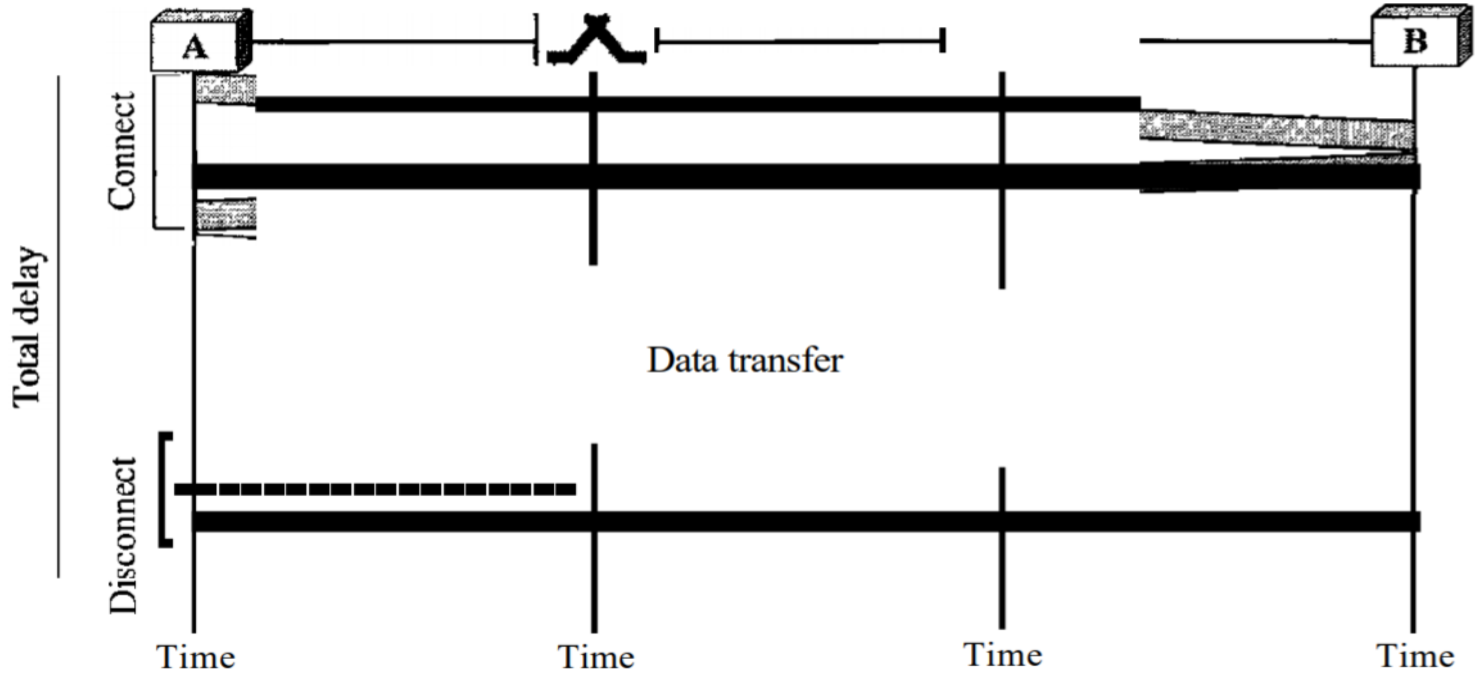
Although a circuit-switched network normally has low efficiency, **the delay in this type of network is minimal**. During data transfer the data are not delayed at each switch; the resources are allocated for the duration of the connection.

There is no waiting time at each switch.

The total delay is due to the time needed to create the connection, transfer data, and disconnect the circuit.

- **The delay caused by the setup is the sum of four parts:** the propagation time of the source computer request (slope of the first gray box), the request signal transfer time (height of the first gray box), the propagation time of the acknowledgment from the destination computer (slope of the second gray box), and the signal transfer time of the acknowledgment (height of the second gray box).
- **The delay due to data transfer is the sum of two parts:** the propagation time (slope of the colored box) and data transfer time (height of the colored box), which can be very long.
- The third box shows the time needed to tear down the circuit.

UNIT – III
SWITCHING



3.1.4 Circuit-Switched Technology in Telephone Networks

The telephone companies have previously chosen the circuit switched approach to switching in the physical layer; today the tendency is moving toward other switching techniques.

DATA COMMUNICATIONS & NETWORKING

B.SC (COMPUTER SCIENCE) – III YEAR

SMCS52

By
Dr.N.N.Krishna Veni,
Assistant Professor,
Department of Computer Science,
Holy Cross Home Science College,
Thoothukudi

UNIT – III
SWITCHING

3.1 Recall of circuit switching networks

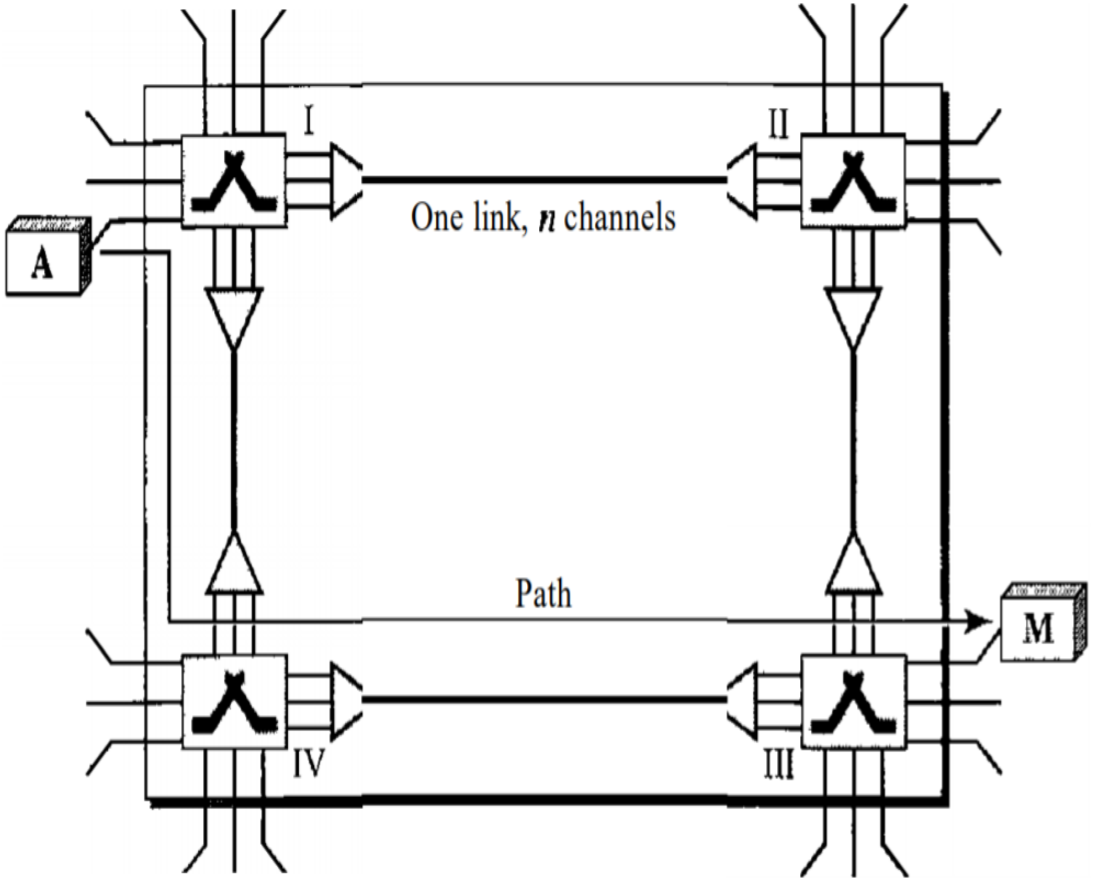
3.2 DATAGRAM NETWORKS

- 3.2.1 Routing Table
- 3.2.2 Efficiency
- 3.2.3 Delay
- 3.2.4 Datagram Networks in the Internet

UNIT – III
SWITCHING

3.1 CIRCUIT-SWITCHED NETWORKS

- A circuit-switched network consists of a set of switches connected by physical links.
- A connection between two stations is a dedicated path made of one or more links.
- However, each connection uses only one dedicated channel on each link.
- Circuit switching takes place at **the physical layer**.
- Before starting communication, the stations must make a **reservation for the resources till end**
- Data transferred between the two stations are **not packetized**. The data are a continuous flow sent by the source station and received by the destination station, although there may be periods of silence.
- There is **no addressing** involved during data transfer.



UNIT – III
SWITCHING

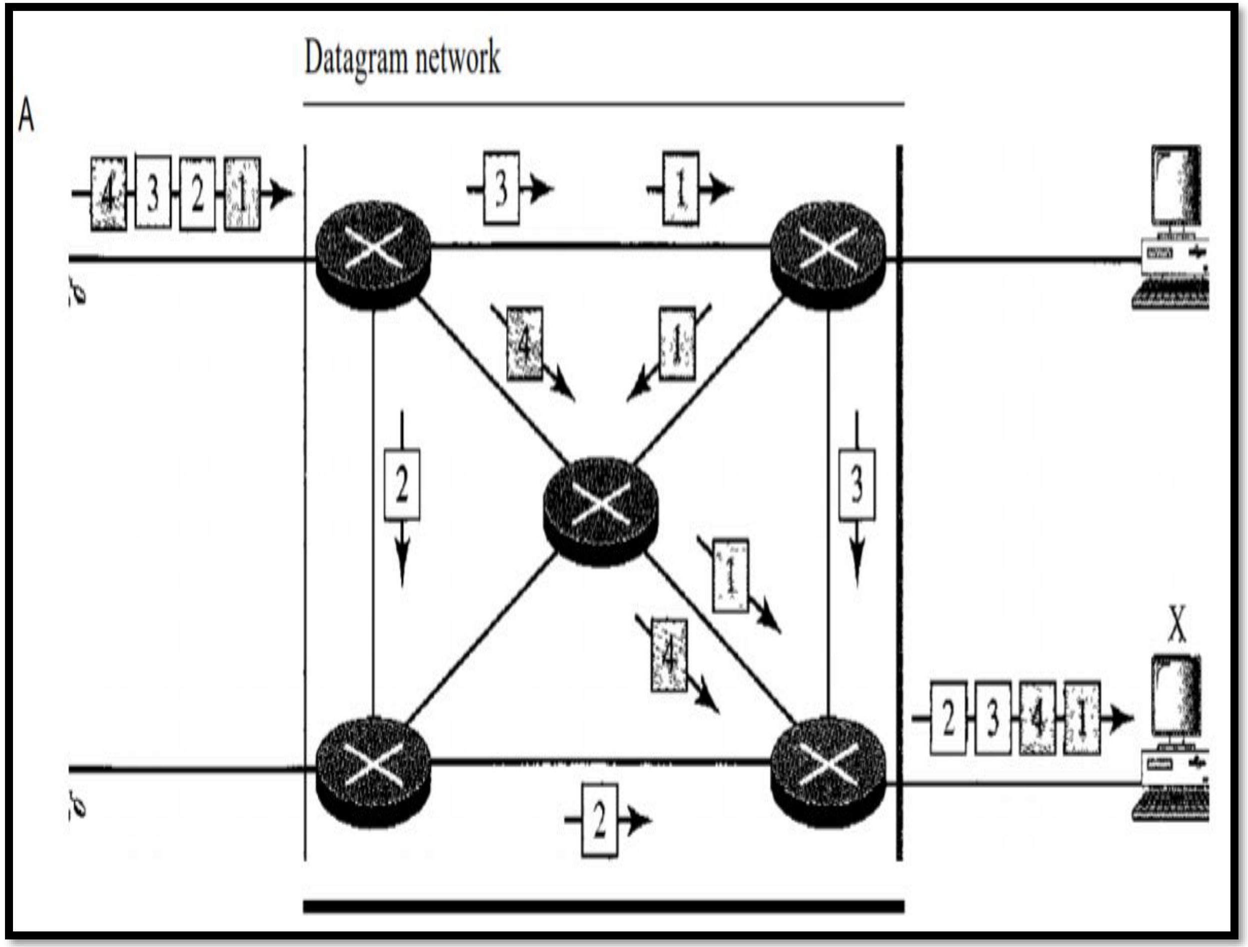
3.2 DATAGRAM NETWORKS

- In packet-switched network the message needs to be **divided into packets** of fixed or variable size.
- In packet switching, there is **no resource allocation** for a packet. Resources are allocated on demand.
- The allocation is done on a **first-come, first-served** basis.
- When a switch receives a packet, the **packet must wait** if there are other packets being processed.
- This lack of reservation may **create delay**.
- Packets in this approach are referred to as **datagrams**.
- Datagram switching is normally done at the **network layer**.
- The switches in a datagram network are traditionally **referred to as routers**.

UNIT – III
SWITCHING

3.2 DATAGRAM NETWORKS

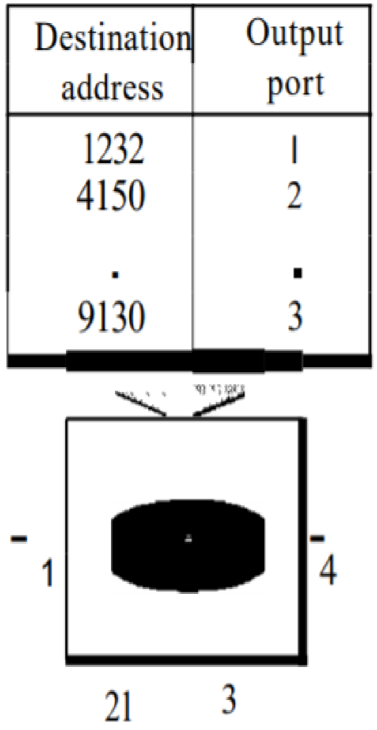
- In this example, all four packets belong to the same message, but **may travel different paths** to reach their destination.
- This approach can cause the datagrams of a transmission to arrive at their destination **out of order** with different delays between the packets.
- Packets may also be **lost or dropped** because of a lack of resources.
- The datagram networks are sometimes referred to as **connectionless** networks(does not keep information about the connection state).
- There are no **setup or teardown phases**.



UNIT – III
SWITCHING

3.2.1 Routing Table

- If there are no setup or teardown phases, **how are the packets routed** to their destinations in a datagram network?
- In this type of network, each switch **has a routing table** which is based on the destination address.
- The routing tables are dynamic and are **updated periodically**.
- The destination addresses and the corresponding forwarding output ports are **recorded in the tables**.
- This is different from the table of a circuit switched network in which each entry is created when the setup phase is completed and deleted when the teardown phase is over
- **Destination Address:** Every packet in a datagram network carries a header that contains, among other information, the destination address of the packet.

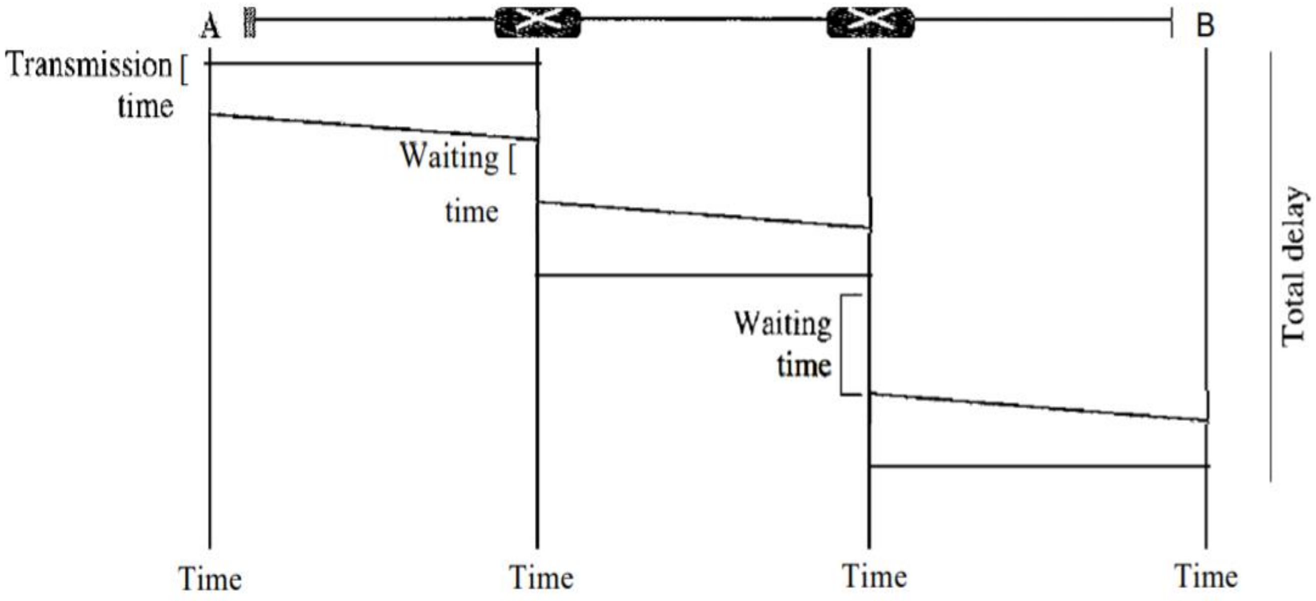


Destination address	Output port
1232	1
4150	2
.	.
9130	3

UNIT – III
SWITCHING

3.2.2 Efficiency : The efficiency of a datagram network is **better.**

3.2.3 Delay: There may be **greater delay**



$$\text{Total delay} = 3T + 3\tau + w_1 + w_2$$

3 times

3propagation delay

2 waiting time(w1&w2)

3.2.4 Datagram Networks in the Internet

The Internet has chosen the datagram approach to switching at the network layer.

DATA COMMUNICATIONS & NETWORKING

B.SC (COMPUTER SCIENCE) – III YEAR

SMCS52

By

Dr.N.N.Krishna Veni,
Assistant Professor,
Department of Computer Science,
Holy Cross Home Science College,
Thoothukudi

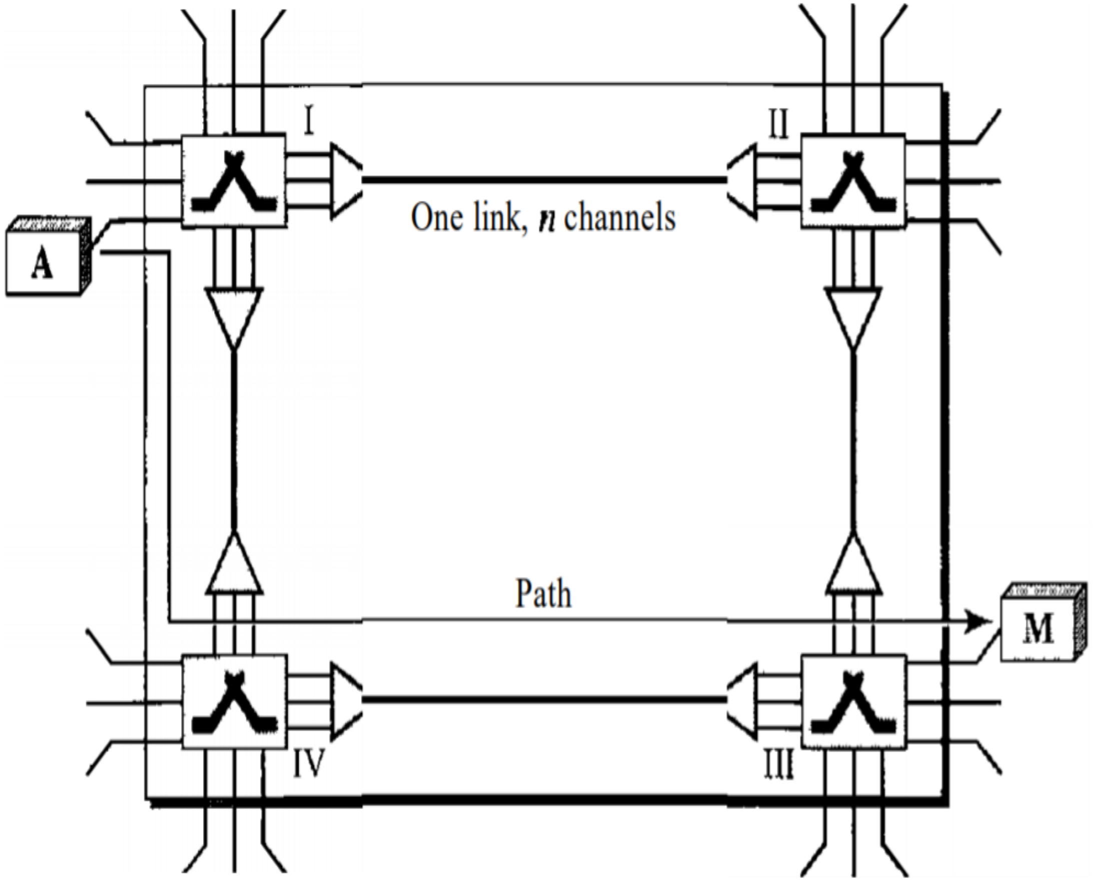
UNIT – III
SWITCHING

3.3 VIRTUAL-CIRCUIT NETWORKS

UNIT – III
SWITCHING

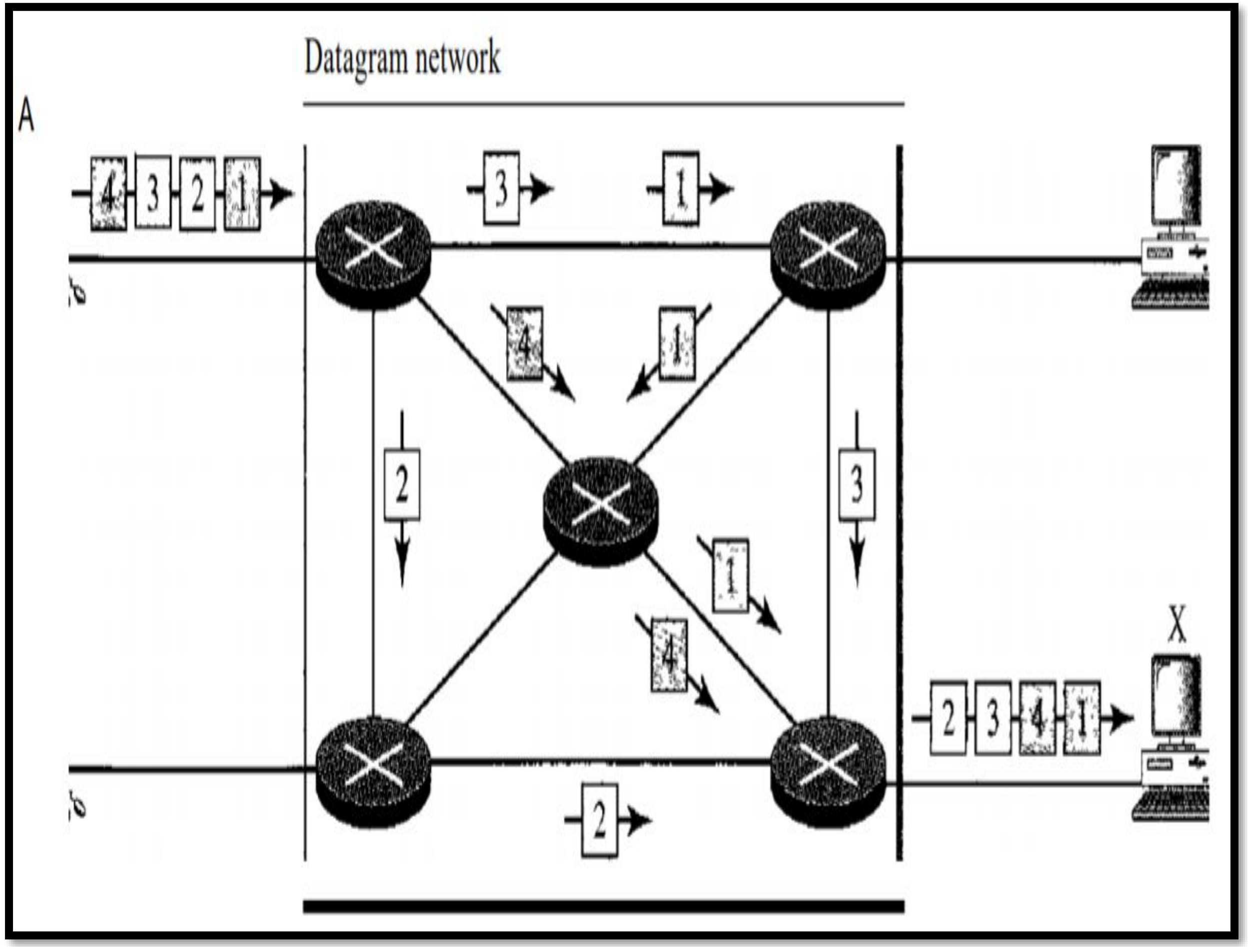
3.1 CIRCUIT-SWITCHED NETWORKS

- A circuit-switched network consists of a set of switches connected by physical links.
- A connection between two stations is a dedicated path made of one or more links.
- However, each connection uses only one dedicated channel on each link.
- Circuit switching takes place at **the physical layer**.
- Before starting communication, the stations must make a **reservation for the resources till end**
- Data transferred between the two stations are **not packetized**. The data are a continuous flow sent by the source station and received by the destination station, although there may be periods of silence.
- There is **no addressing** involved during data transfer.



UNIT – III
SWITCHING

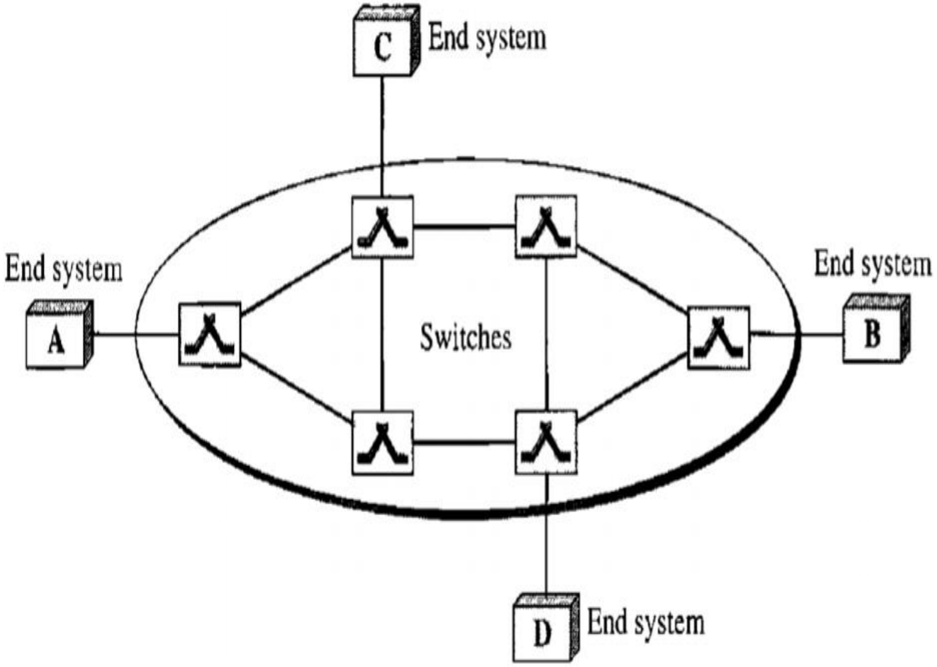
- In this example, all four packets belong to the same message, but **may travel different paths** to reach their destination.
- This approach can cause the datagrams of a transmission to arrive at their destination **out of order** with different delays between the packets.
- Packets may also be **lost or dropped** because of a lack of resources.
- The datagram networks are sometimes referred to as **connectionless** networks(does not keep information about the connection state).
- There are no **setup or teardown phases**.



UNIT – III
SWITCHING

3.3 VIRTUAL-CIRCUIT NETWORKS

- A virtual-circuit network is a **cross between** a circuit-switched and datagram network.
- It has **some characteristics of both**.
- There are setup, teardown and data transfer phase.
- Resources can be allocated during the setup phase, or on demand.
- Data are packetized and each packet carries an address in the header.
- As in a circuit-switched network, all packets follow the same path established during the connection.
- A virtual-circuit network is normally implemented in the data link layer.



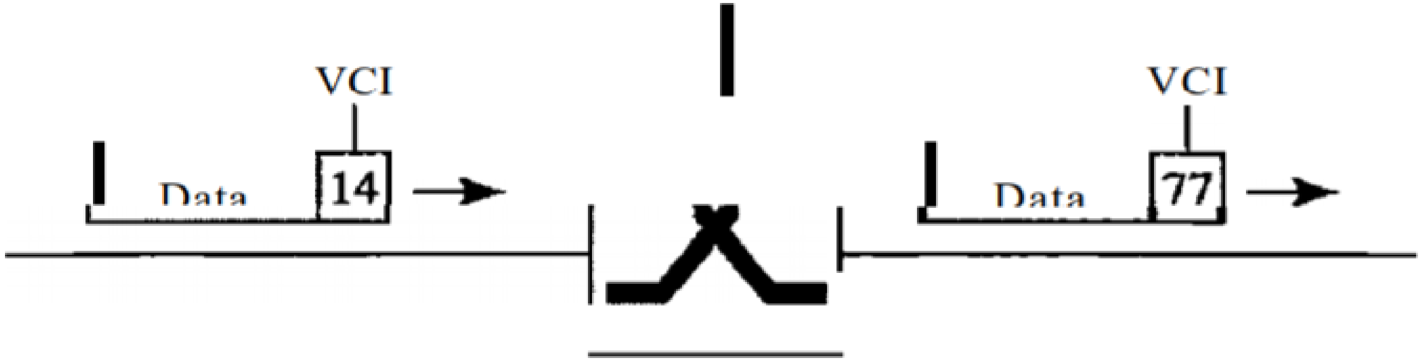
UNIT – III
SWITCHING

3.3.1 Addressing: Two types of addressing are involved:

Global Addressing: A source or a destination needs to have a global address.

Virtual-Circuit Identifier:

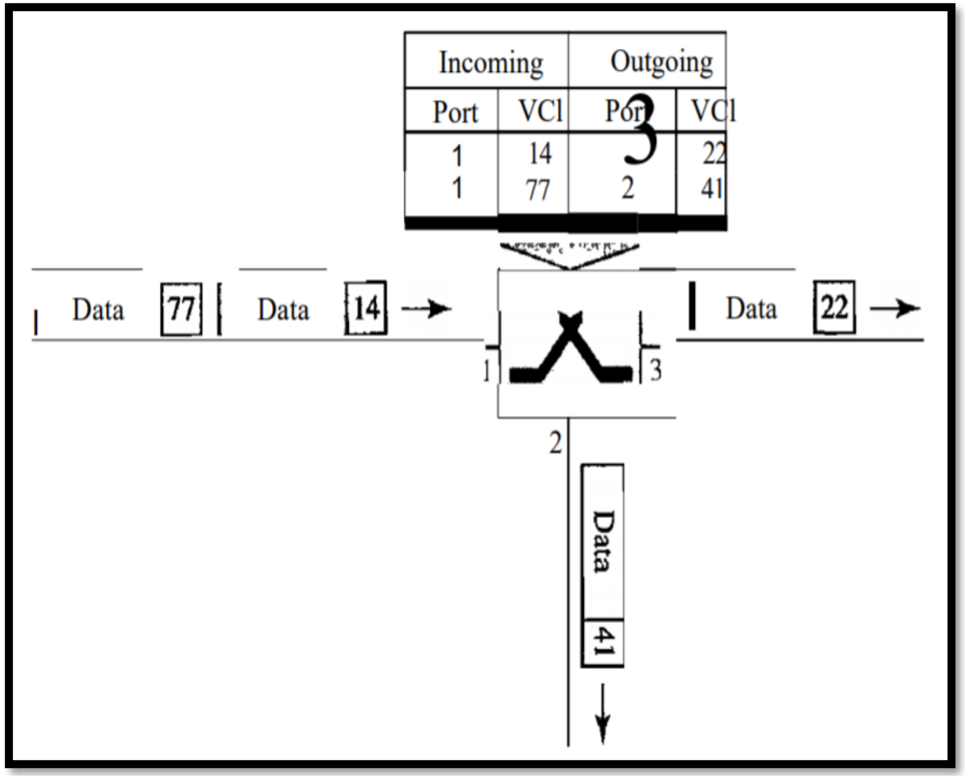
- The identifier that is actually used for data transfer is called the virtual-circuit identifier (VCI).
- A VCI, unlike a global address**, is a small number** that has only switch scope; it is used by a frame between two switches.
- When a frame arrives at a switch, it has a VCI; when it leaves, it has a different VCl.



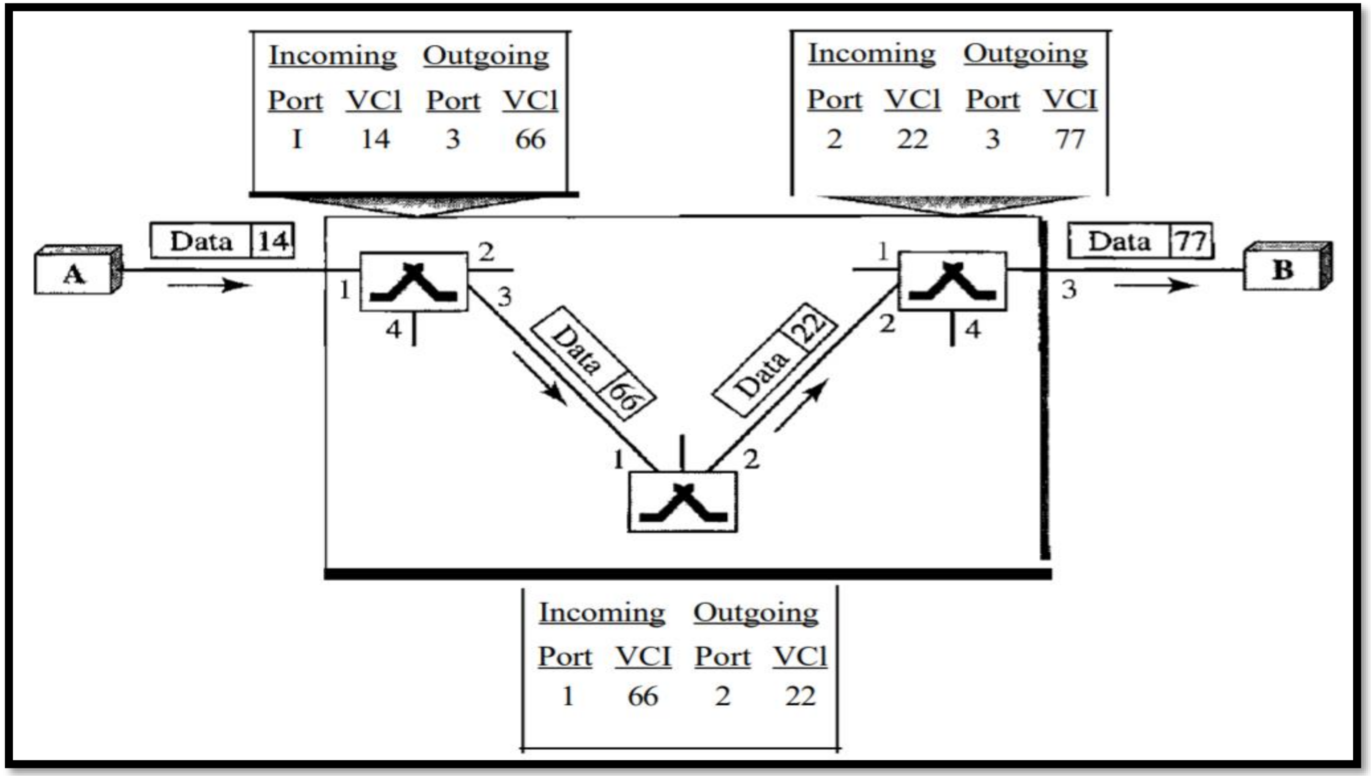
UNIT – III
SWITCHING

3.3.2 Three Phases

Data Transfer Phase: To transfer a frame from a source to its destination, all switches need to have a table entry for this virtual circuit. The routing table, has four columns.



Incoming		Outgoing	
Port	VCI	Port	VCI
1	14	3	22
1	77	2	41



Incoming		Outgoing	
Port	VCI	Port	VCI
I	14	3	66

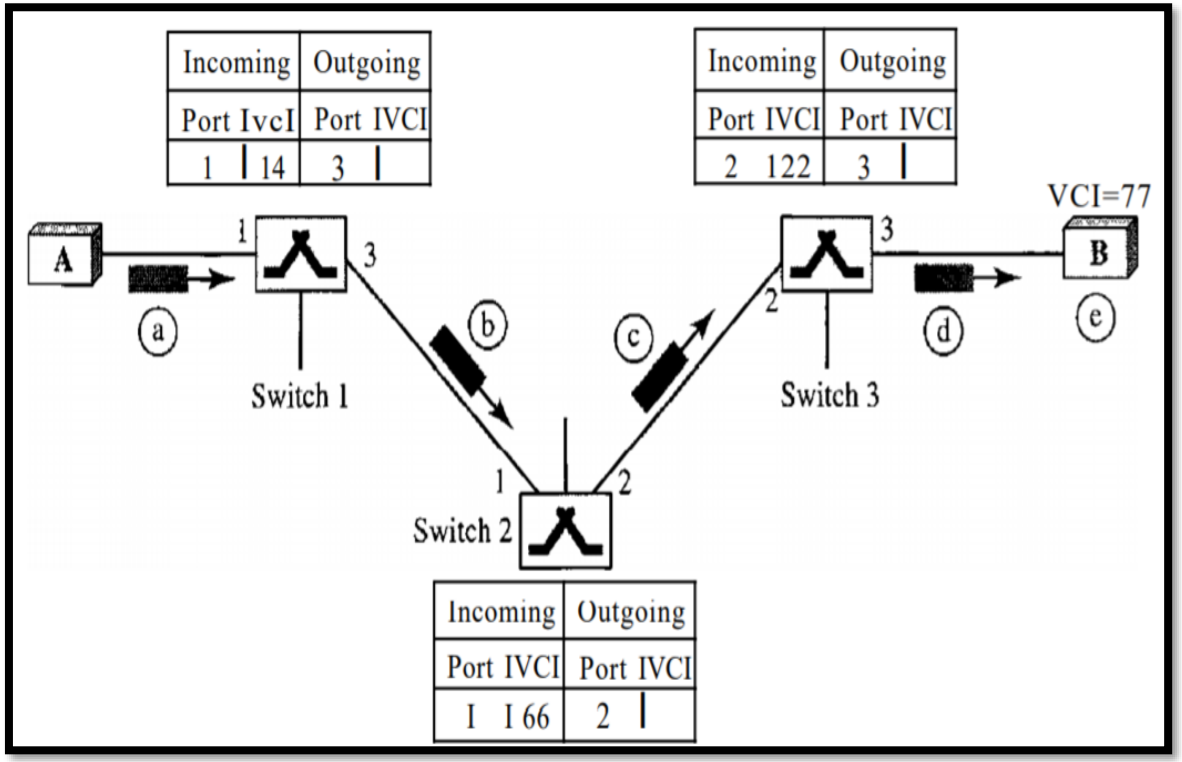
Incoming		Outgoing	
Port	VCI	Port	VCI
2	22	3	77

Incoming		Outgoing	
Port	VCI	Port	VCI
1	66	2	22

UNIT – III
SWITCHING

Setup Phase: In the setup phase, a switch creates an entry for a virtual circuit. For example, suppose source A needs to create a virtual circuit to B. Two steps are required: the setup request and the acknowledgment.

Setup Request



Switch 1:

Incoming		Outgoing	
Port	IvcI	Port	IVCI
1	14	3	

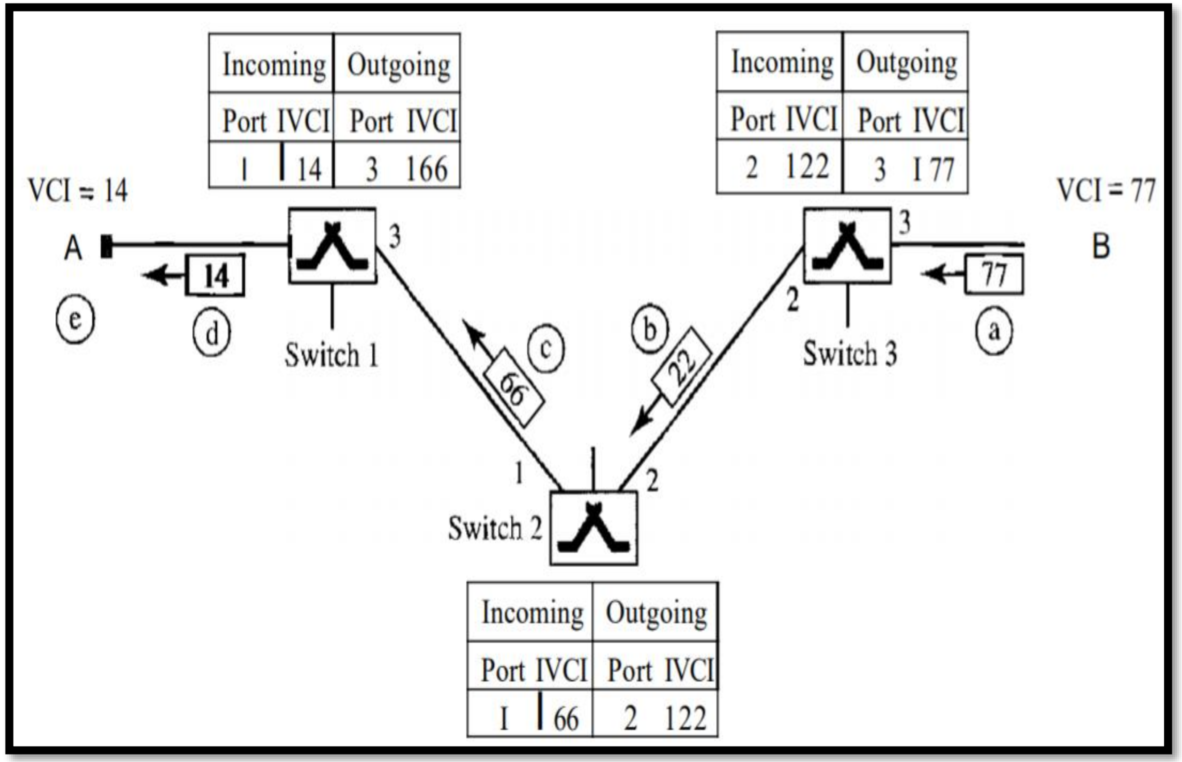
Switch 3:

Incoming		Outgoing	
Port	IVCI	Port	IVCI
2	122	3	

Switch 2:

Incoming		Outgoing	
Port	IVCI	Port	IVCI
I	I 66	2	

Acknowledgement



Switch 1:

Incoming		Outgoing	
Port	IVCI	Port	IVCI
1	14	3	166

Switch 3:

Incoming		Outgoing	
Port	IVCI	Port	IVCI
2	122	3	I 77

Switch 2:

Incoming		Outgoing	
Port	IVCI	Port	IVCI
I	66	2	122

UNIT – III
SWITCHING

- **Teardown phase:** Teardown request sent. All switches delete the corresponding entry from their tables.

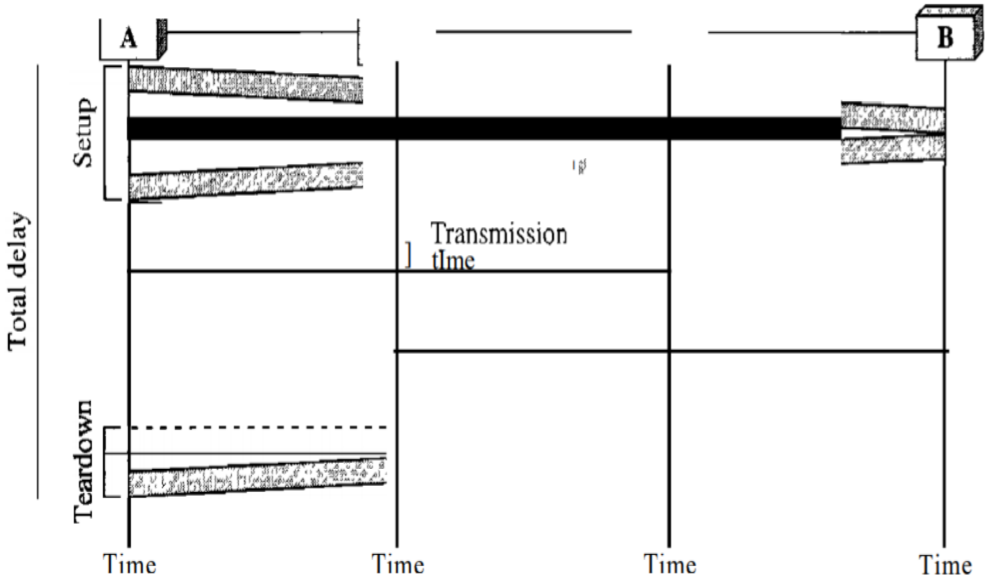
3.3.3 Efficiency

- Resource reservation in a virtual-circuit network can be made **during the setup or can be on demand** during the data transfer phase.
- In the first case, the delay for each packet is the same; in the second case, each packet may encounter different delays.

3.3.4 Delay in Virtual-Circuit Networks

- There is a one-time delay for setup and a one-time delay for teardown.
- If resources are allocated during the setup phase, there is no wait time for individual packets.

3.3.5 Circuit-Switched Technology in WANs: Used in Frame Relay and ATM networks



$$\text{Total delay} = 3T + 3\tau + \text{setup delay} + \text{teardown delay}$$

DATA COMMUNICATIONS & NETWORKING

B.SC (COMPUTER SCIENCE) – III YEAR

SMCS52

By
Dr.N.N.Krishna Veni,
Assistant Professor,
Department of Computer Science,
Holy Cross Home Science College,
Thoothukudi

UNIT – IV
DATA LINK LAYER

CHAPTER 7: Error Detection and Correction

4.1 INTRODUCTION

4.1.1Types of Errors

4.1.2 Redundancy

4.1.3 Detection Versus Correction

4.1.4 Forward Error Correction Versus Retransmission

4.1.5 Coding

4.1.6 Modular Arithmetic

UNIT – IV
ERROR DETECTION AND CORRECTION

- Networks must be able to transfer data from one device to another with acceptable accuracy.
- For most applications, a system must guarantee that the data received are identical to the data transmitted.
- Any time data are transmitted from one node to the next, they can become corrupted in passage.
- Many factors can alter one or more bits of a message.
- Some applications require a mechanism for detecting and correcting errors.

4.1 INTRODUCTION

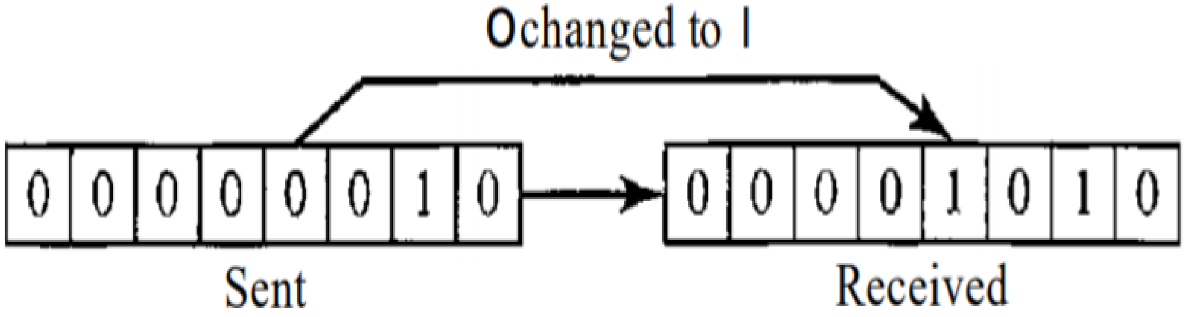
4.1.1Types of Errors

- Whenever bits flow from one point to another, they are subject to unpredictable changes because of interference.
- This interference can change the shape of the signal.
- Errors are classified into two types.

UNIT – IV
ERROR DETECTION AND CORRECTION

Single-Bit Error:

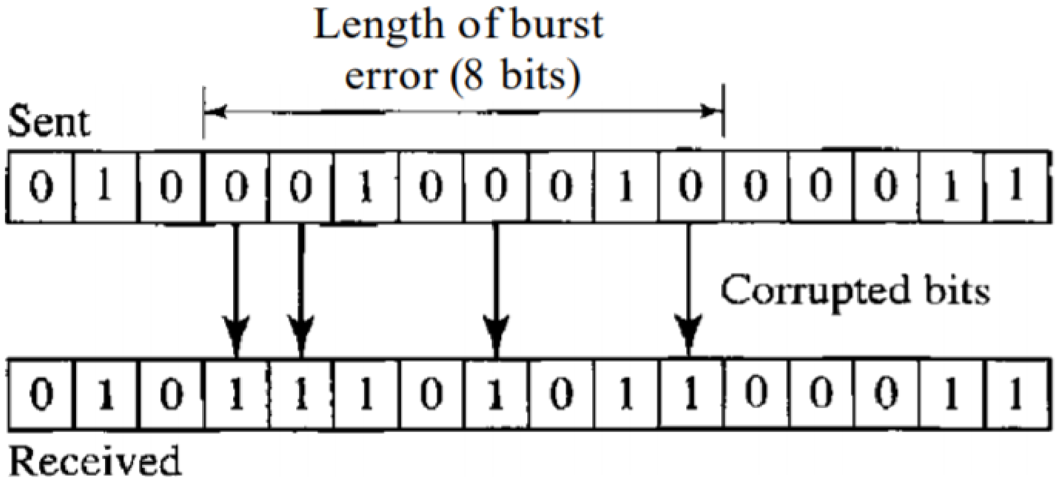
- The term single-bit error means that only 1 bit of a given data unit is changed from 1 to 0 or from 0 to 1.
- Figure 4.1 shows the effect of a single-bit error on a data unit.
- To understand the impact of the change, imagine that each group of 8 bits is an ASCII character with a 0 bit added to the left.
- In Figure,00000010 (ASCII STX) was sent, meaning start of text, but 00001010 (ASCII LF) was received, meaning line feed.



UNIT – IV
ERROR DETECTION AND CORRECTION

Burst Error:

- The term burst error means that 2 or more bits in the data unit have changed from 1 to 0 or from 0 to 1.
- Figure 4.2 shows the effect of a burst error on a data unit. In this case, 0100010001000011 was sent, but 0101110101100011 was received.
- Note that a burst error does not necessarily mean that the errors occur in consecutive bits.
- The length of the burst is measured from the first corrupted bit to the last corrupted bit.
- Some bits in between may not have been corrupted.



UNIT – IV
ERROR DETECTION AND CORRECTION

4.1.2 Redundancy

- The central concept in detecting or correcting errors is redundancy.
- To be able to detect or correct errors, we need to send some extra bits with our data.
- These redundant bits are added by the sender and removed by the receiver.
- Their presence allows the receiver to detect or correct corrupted bits.

4.1.3 Detection Versus Correction

- The correction of errors is more difficult than the detection.
- In error detection, we are looking only to see if any error has occurred. The answer is a simple yes or no. We are not even interested in the number of errors. A single-bit error is the same for us as a burst error.
- In error correction, we need to know the exact number of bits that are corrupted and more importantly, their location in the message. The number of the errors and the size of the message are important factors. If we need to correct one single error in an 8-bit data unit, we need to consider eight possible error locations; if we need to correct two errors in a data unit of the same size, we need to consider 28 possibilities. You can imagine the receiver's difficulty in finding 10 errors in a data unit of 1000 bits.

UNIT – IV
ERROR DETECTION AND CORRECTION

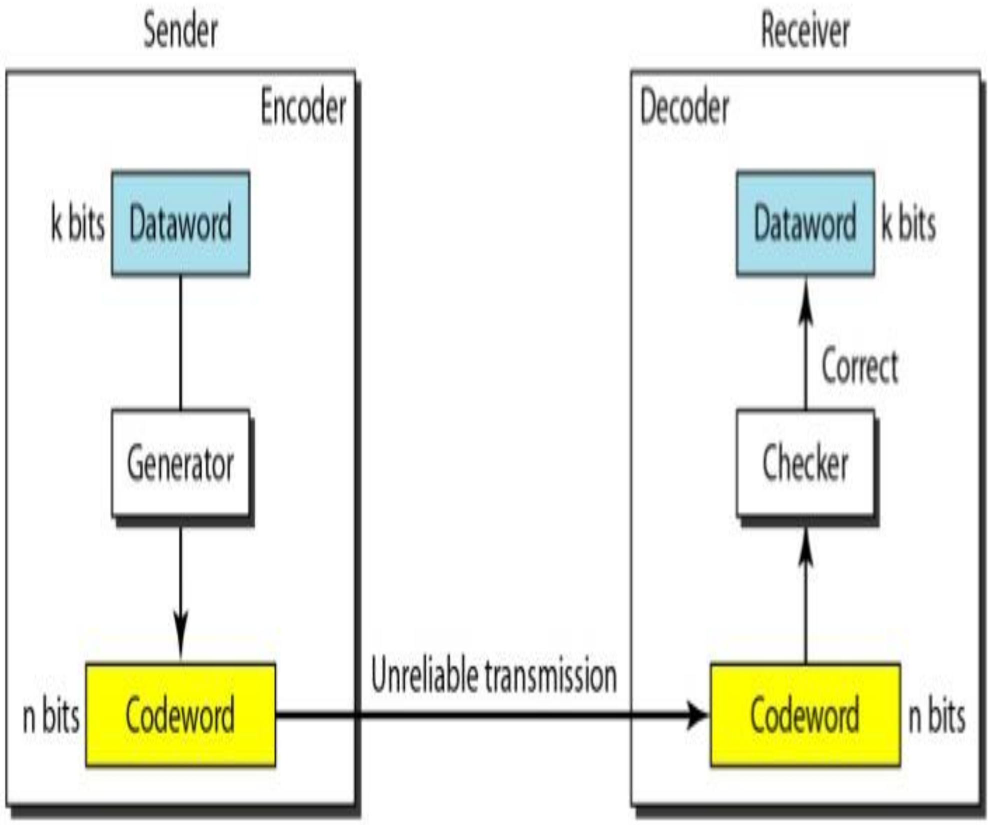
4.1.4 Forward Error Correction Versus Retransmission

There are two main methods of error correction.

- Forward error correction is the process in which the receiver tries to guess the message by using redundant bits.
- Correction by retransmission is a technique in which the receiver detects the occurrence of an error and asks the sender to resend the message.

4.1.5 Coding

- Redundancy is achieved through various coding schemes. The sender adds redundant bits through a process that creates a relationship between the redundant bits and the actual data bits. The receiver checks the relationships between the two sets of bits to detect or correct the errors.
- We can divide coding schemes into two broad categories: block coding and convolution coding.



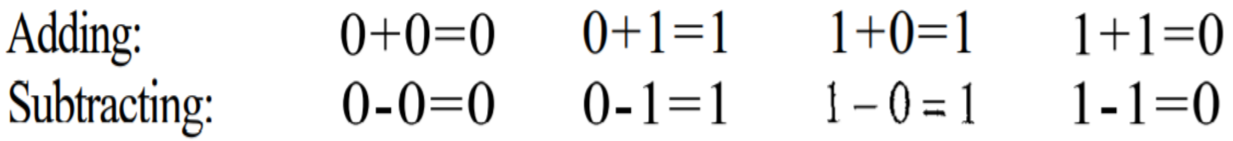
UNIT – IV
ERROR DETECTION AND CORRECTION

4.1.6 Modular Arithmetic

- In modular arithmetic, we use only a limited range of integers. We define an upper limit, called a modulus N. We then use only the integers 0 to N - 1, inclusive. This is modulo-N arithmetic.
- For example, if the modulus is 12, we use only the integers 0 to 11, inclusive.
- In a modulo-N system, if a number is greater than N, it is divided by N and the remainder is the result. If it is negative, as many Ns as needed are added to make it positive.
- Consider our clock system. If we start a job at 11 A.M. and the job takes 5 h, we can say that the job is to be finished at 16:00 if we are in the military, or we can say that it will be finished at 4 P.M. (the remainder of 16/12 is 4).
- Addition and subtraction in modulo arithmetic are simple. There is no carry when you add two digits in a column. There is no carry when you subtract one digit from another in a column.

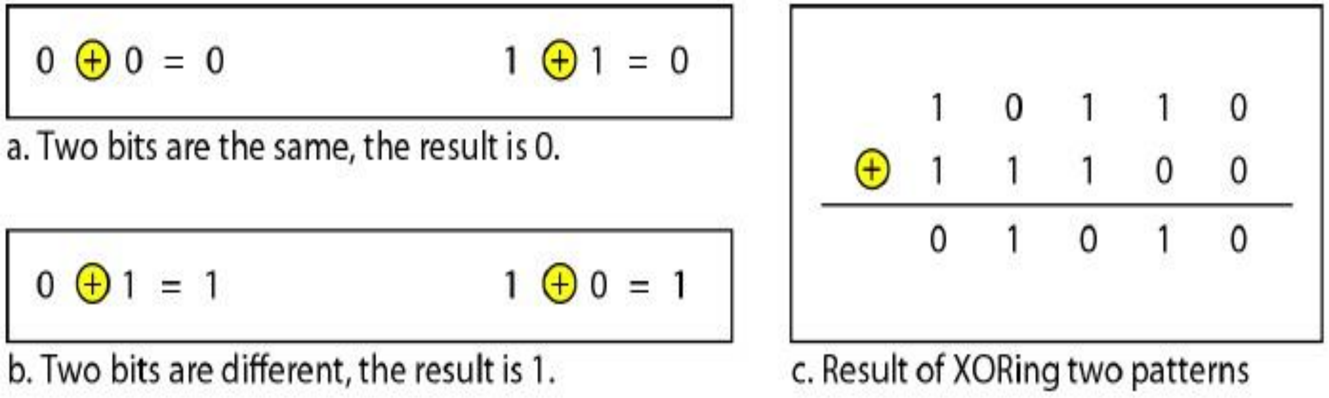
UNIT – III
ERROR DETECTION AND CORRECTION

- **Modulo-2 Arithmetic:** In this arithmetic, the modulus N is 2. We can use only 0 and 1. Operations in this arithmetic are very simple. The following shows how we can add or subtract 2 bits.



Adding:	0+0=0	0+1=1	1+0=1	1+1=0
Subtracting:	0-0=0	0-1=1	1 – 0 = 1	1-1=0

- Notice particularly that addition and subtraction give the same results. In this arithmetic we use the XOR (exclusive OR) operation for both addition and subtraction. The result of an XOR operation is 0 if two bits are the same; the result is I if two bits are different.



$0 \oplus 0 = 0$ $1 \oplus 1 = 0$

a. Two bits are the same, the result is 0.

$0 \oplus 1 = 1$ $1 \oplus 0 = 1$

b. Two bits are different, the result is 1.

$$\begin{array}{cccccc} & 1 & 0 & 1 & 1 & 0 \\ \oplus & 1 & 1 & 1 & 0 & 0 \\ \hline & 0 & 1 & 0 & 1 & 0 \end{array}$$

c. Result of XORing two patterns

UNIT – III
ERROR DETECTION AND CORRECTION

Other Modulo Arithmetic: The principle is the same; we use numbers between 0 and N - 1. If the modulus is not 2, addition and subtraction are distinct. If we get a negative result, we add enough multiples of N to make it positive.

DATA COMMUNICATIONS & NETWORKING

B.SC (COMPUTER SCIENCE) – III YEAR

SMCS52

By

Dr.N.N.Krishna Veni,
Assistant Professor,
Department of Computer Science,
Holy Cross Home Science College,
Thoothukudi

UNIT – IV
DATA LINK LAYER

CHAPTER 7: Error Detection and Correction

4.2 CHECKSUM

- 4.2.1 Idea
- 4.2.2 One's Complement
- 4.2.3 Internet Checksum

UNIT – IV
ERROR DETECTION AND CORRECTION

4.2 CHECKSUM : Several protocols still use the checksum for error detection.

4.2.1 Idea

The concept of the checksum is not difficult. Let us illustrate it with a few examples.

Example 1: Suppose our data is a list of five 4-bit numbers that we want to send to a destination. In addition to sending these numbers, we send the sum of the numbers. For example, if the set of numbers is (7, 11, 12, 0, 6), we send (7, 11, 12,0,6,36), where 36 is the sum of the original numbers. The receiver adds the five numbers and compares the result with the sum. If the two are the same, the receiver assumes no error, accepts the five numbers, and discards the sum. Otherwise, there is an error somewhere and the data are not accepted.

Example 2: In other case we can send the negative (complement) of the sum, called the checksum. In this case, we send (7, 11, 12,0,6, -36). The receiver can add all the numbers received (including the checksum). If the result is 0, it assumes no error; otherwise, there is an error

UNIT – IV
ERROR DETECTION AND CORRECTION

4.2.2 One's Complement

The previous example has one major drawback. All of our data can be written as a 4-bit word (they are less than 15) except for the checksum. One solution is to use one's complement arithmetic.

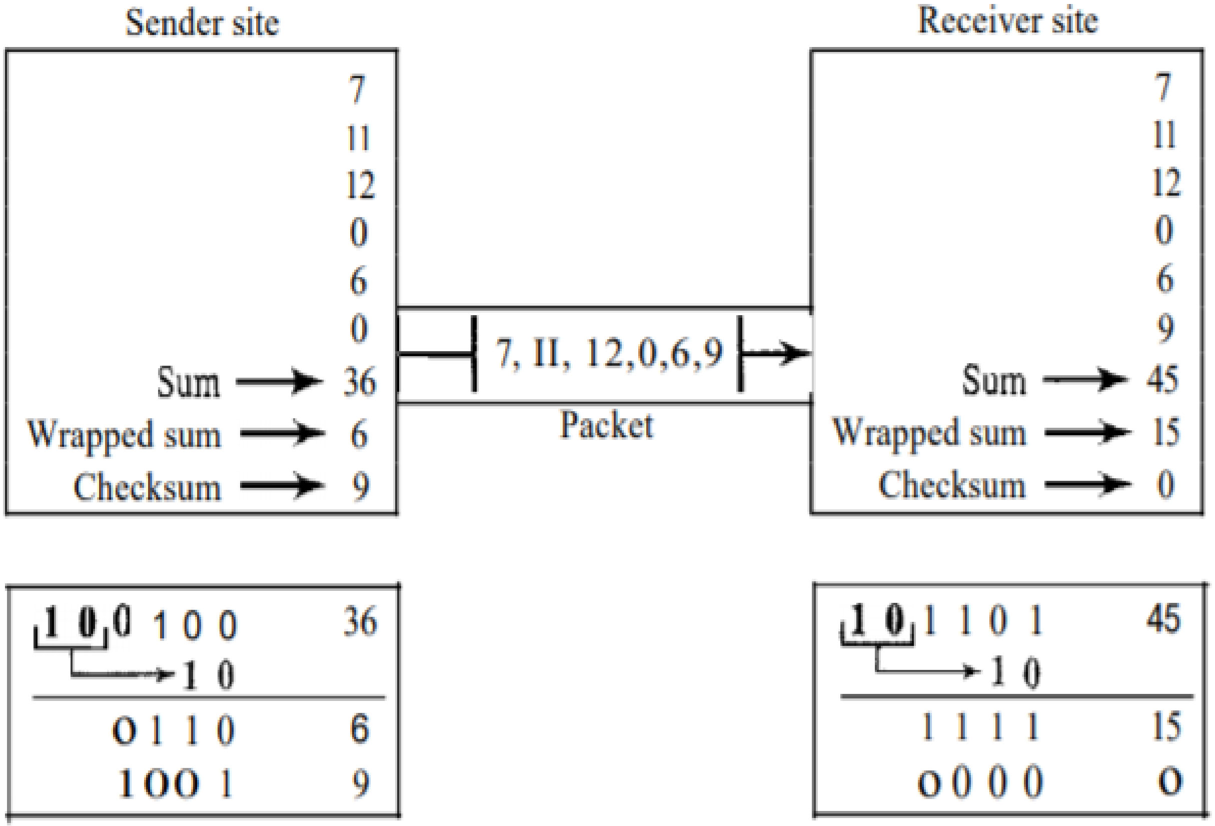
Example 3: How can we represent the number 21 in one's complement arithmetic using only four bits?

Solution: The number 21 in binary is 10101 (it needs five bits). We can wrap the leftmost bit and add it to the four rightmost bits. We have (0101 + 1) = 0110 or 6.

UNIT – IV
ERROR DETECTION AND CORRECTION

Example 4: Let us redo Example 2(7,11,12,0,6,) using one's complement arithmetic. The sender initializes the checksum to 0 and adds all data items and the checksum. The result is 36(100100) that cannot be expressed in 4 bits. The extra two bits are wrapped to create the wrapped sum value 6(0110). The wrapped sum(6 - 0110) is then complemented(1001 - 9), resulting in the checksum value 9. The sender now sends.

The receiver follows the same procedure as the sender. It adds all data items; the result is 45(101101). The sum is wrapped and becomes 15(1111). The wrapped sum is complemented(0000) and becomes 0. Since the value of the checksum is 0, this means that the data is not corrupted.



Details of wrapping and complementing

Details of wrapping and complementing

UNIT – IV
ERROR DETECTION AND CORRECTION

4.2.3 Internet Checksum

Traditionally, the Internet has been using a 16-bit checksum. The sender & receiver calculates the checksum by following these steps.

Sender site:

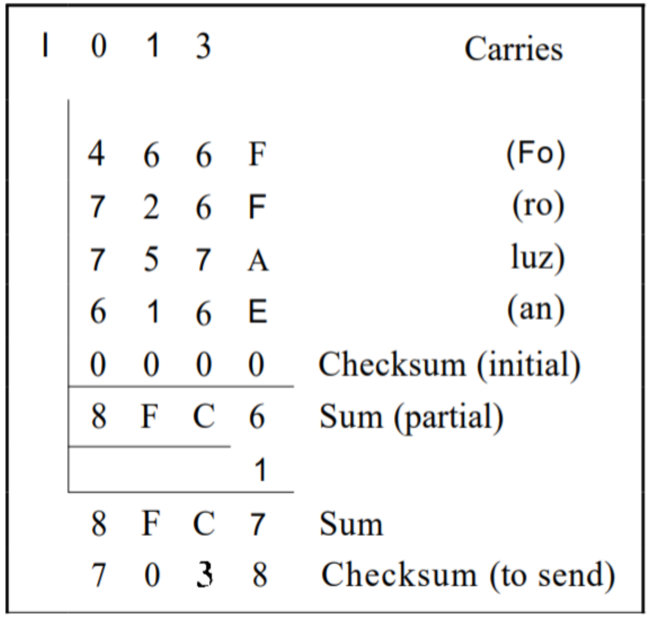
1. The message is divided into 16-bit words.
2. The value of the checksum word is set to 0.
3. All words including the checksum are added using one's complement addition.
4. The sum is complemented and becomes the checksum.
5. The checksum is sent with the data.

Receiver site:

1. The message (including checksum) is divided into 16-bit words.
2. All words are added using one's complement addition.
3. The sum is complemented and becomes the new checksum.
4. If the value of checksum is 0, the message is accepted; otherwise, it is rejected.

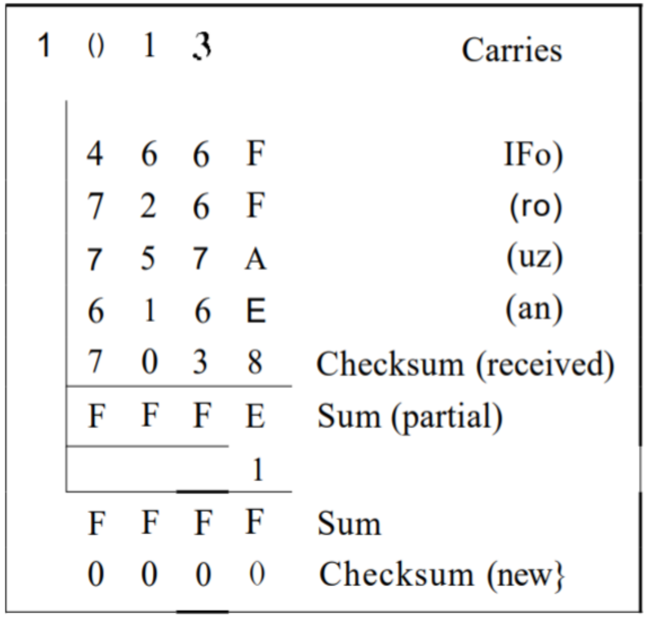
UNIT – IV
ERROR DETECTION AND CORRECTION

Example 10.23 Let us calculate the checksum for a text of 8 characters ("Forouzan"). The text needs to be divided into 2-byte (16-bit) words. We use ASCII (see Appendix A) to change each byte to a 2-digit hexadecimal number. For example, F is represented as 46 and 0 is represented as 6F. Figure 4.6 shows how the checksum is calculated at the sender and receiver sites. In part a of the figure, the value of partial sum for the first column is 36. We keep the rightmost digit (6) and insert the leftmost digit (3) as the carry in the second column. The process is repeated for each column.



I	0	1	3	Carries
4	6	6	F	(Fo)
7	2	6	F	(ro)
7	5	7	A	luz)
6	1	6	E	(an)
0	0	0	0	Checksum (initial)
8	F	C	6	Sum (partial)
			1	
8	F	C	7	Sum
7	0	3	8	Checksum (to send)

a. Checksum at the sender site



1	0	1	3	Carries
4	6	6	F	IFo)
7	2	6	F	(ro)
7	5	7	A	(uz)
6	1	6	E	(an)
7	0	3	8	Checksum (received)
F	F	F	E	Sum (partial)
			1	
F	F	F	F	Sum
0	0	0	0	Checksum (new}

b. Checksum at the receiver site

DATA COMMUNICATIONS & NETWORKING

B.SC (COMPUTER SCIENCE) – III YEAR

SMCS52

By

Dr.N.N.Krishna Veni,
Assistant Professor,
Department of Computer Science,
Holy Cross Home Science College,
Thoothukudi

UNIT – IV
DATA LINK LAYER

CHAPTER 8: Data Link Control

4.3 FRAMING

  4.3.1 Fixed-Size Framing

  4.3.2 Variable-Size Framing

UNIT – IV
FRAMING

- Data transmission in the physical layer means moving bits in the form of a signal from the source to the destination.
- The physical layer provides bit synchronization to ensure that the sender and receiver use the same bit durations and timing.
- The data link layer, on the other hand, needs to pack bits into frames, so that each frame is distinguishable from another.
- Framing in the data link layer separates a message from one source to a destination, or from other messages to other destinations, by adding a sender address and a destination address.
- Although the whole message could be packed in one frame, that is not normally done.
- One reason is that a frame can be very large, making flow and error control very inefficient.
- When a message is carried in one very large frame, even a single-bit error would require the retransmission of the whole message.
- When a message is divided into smaller frames, a single-bit error affects only that small frame.

UNIT – IV
FRAMING

4.3.1 Fixed-Size Framing

- Frames can be of fixed or variable size.
- In 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 ATM wide-area network, which uses frames of fixed size called cells

4.3.2 Variable-Size Framing

- In variable-size framing, we need a way to define the end of the 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 Protocols**
  - ➢ **Bit-Oriented Protocols**

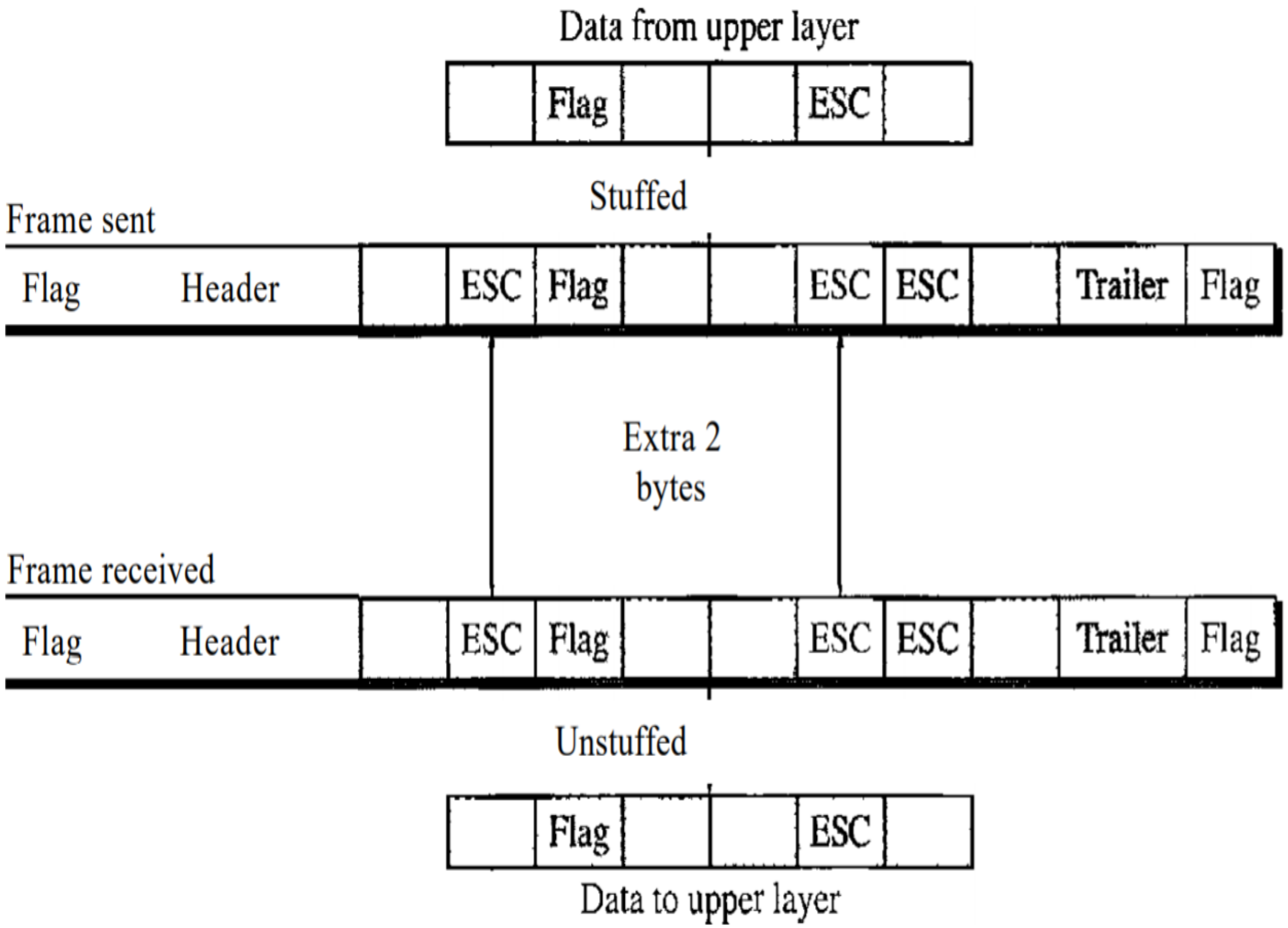
UNIT – IV
FRAMING

Character-Oriented Protocols:

- In a character-oriented protocol, data to be carried are 8-bit characters from a coding system.
- To separate one frame from the next, an 8-bit (1-byte) flag is added at the beginning and the 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.
- Any pattern 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.
- The data section is stuffed with an extra byte. This byte is usually called the **escape character** (ESC), which has a predefined bit pattern.
- Whenever the receiver encounters the ESC character, it removes it from the data section and treats the next character as data, not a delimiting flag.

UNIT – IV
FRAMING

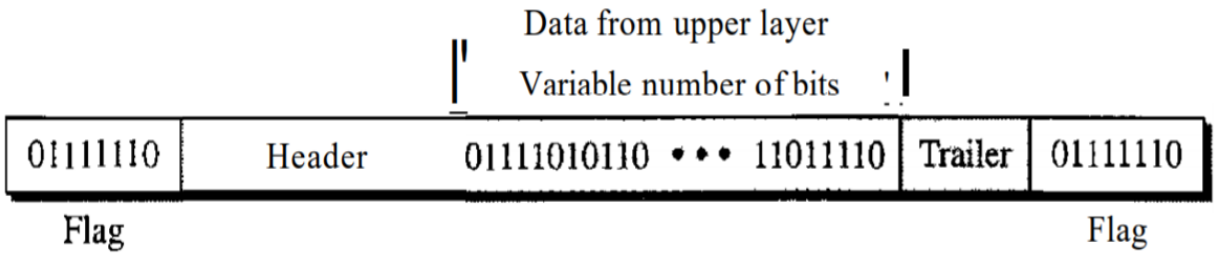
- Byte stuffing by the escape character allows the presence of the flag in the data section of the frame, but it creates another problem.
- What happens if the text contains one or more escape characters followed by a flag?
- The receiver removes the escape character, but keeps the flag, which is incorrectly interpreted as the end of the frame.
- To solve this problem, the escape characters that are part of the text must also be marked by another escape character.
- In other words, if the escape character is part of the text, an extra one is added to show that the second one is part of the text.



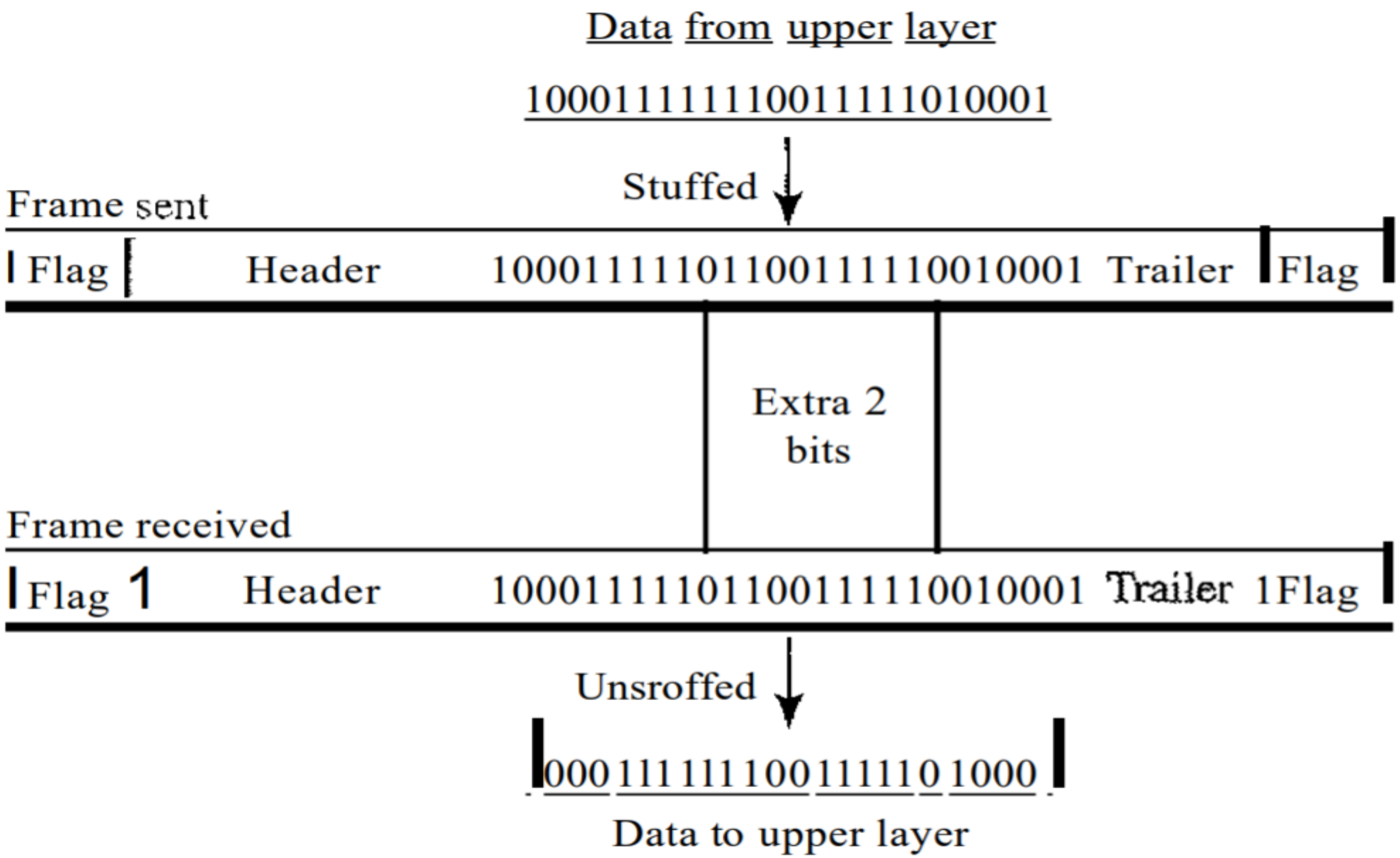
UNIT – IV
FRAMING

Bit-Oriented Protocols:

- In addition to headers (and possible trailers), we still need a delimiter to separate one frame from the other.
- Flag 01111110 as the delimiter to define the beginning and the end of the frame, as shown in Figure.
- This flag can create the same type of problem we saw in the byte-oriented protocols.
- That is, if the flag pattern appears in the data, we 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.



UNIT – IV
FRAMING



DATA COMMUNICATIONS & NETWORKING

B.SC (COMPUTER SCIENCE) – III YEAR

SMCS52

By
Dr.N.N.Krishna Veni,
Assistant Professor,
Department of Computer Science,
Holy Cross Home Science College,
Thoothukudi

UNIT – IV
DATA LINK LAYER

CHAPTER 8: Data Link Control

4.4 FLOW AND ERROR CONTROL

- 4.4.1 Flow Control
- 4.4.2 Error Control

4.5 PROTOCOLS

UNIT – IV
FLOW & ERROR CONTROL

4.4 FLOW AND ERROR CONTROL

- The most important responsibilities of the data link layer are flow control and error control.
- Collectively, these functions are known as data link control.

4.4.1 Flow Control

- Flow control coordinates the amount of data that can be sent before receiving an acknowledgment and is one of the most important duties of the data link layer.
- The flow of data must not be allowed to overwhelm the receiver.
- Any receiving device has a limited speed at which it can process incoming data and a limited amount of memory in which to store incoming data.
- The receiving device must be able to inform the sending device before those limits are reached and to request that the transmitting device send fewer frames or stop temporarily.

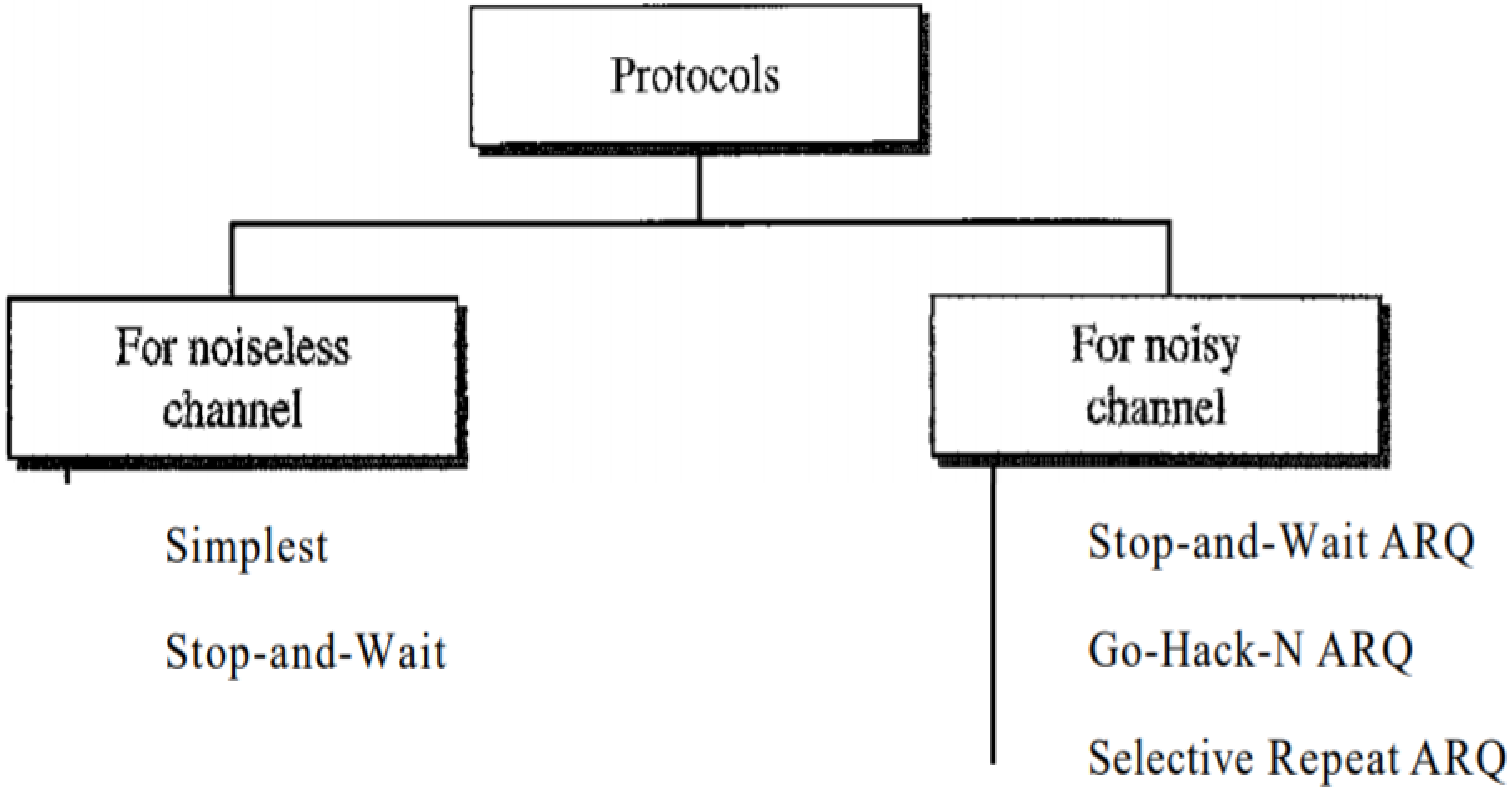
UNIT – IV
FLOW & ERROR CONTROL

- Incoming data must be checked and processed before they can be used. The rate of such processing is often slower than the rate of transmission.
- For this reason, each receiving device has a block of memory, called a buffer, reserved for storing incoming data until they are processed.
- If the buffer begins to fill up, the receiver must be able to tell the sender to halt transmission until it is once again able to receive.

4.4.2 Error Control

- Error control is both error detection and error correction.
- It allows the receiver to inform the sender of any frames lost or damaged in transmission and coordinates the retransmission of those frames by the sender.
- In the data link layer, the term error control refers primarily to methods of error detection and retransmission.
- Error control in the data link layer is often implemented simply: Any time an error is detected in an exchange, specified frames are retransmitted. This process is called automatic repeat request (ARQ).

UNIT – IV
PROTOCOLS



DATA COMMUNICATIONS & NETWORKING

B.SC (COMPUTER SCIENCE) – III YEAR

SMCS52

By
Dr.N.N.Krishna Veni,
Assistant Professor,
Department of Computer Science,
Holy Cross Home Science College,
Thoothukudi

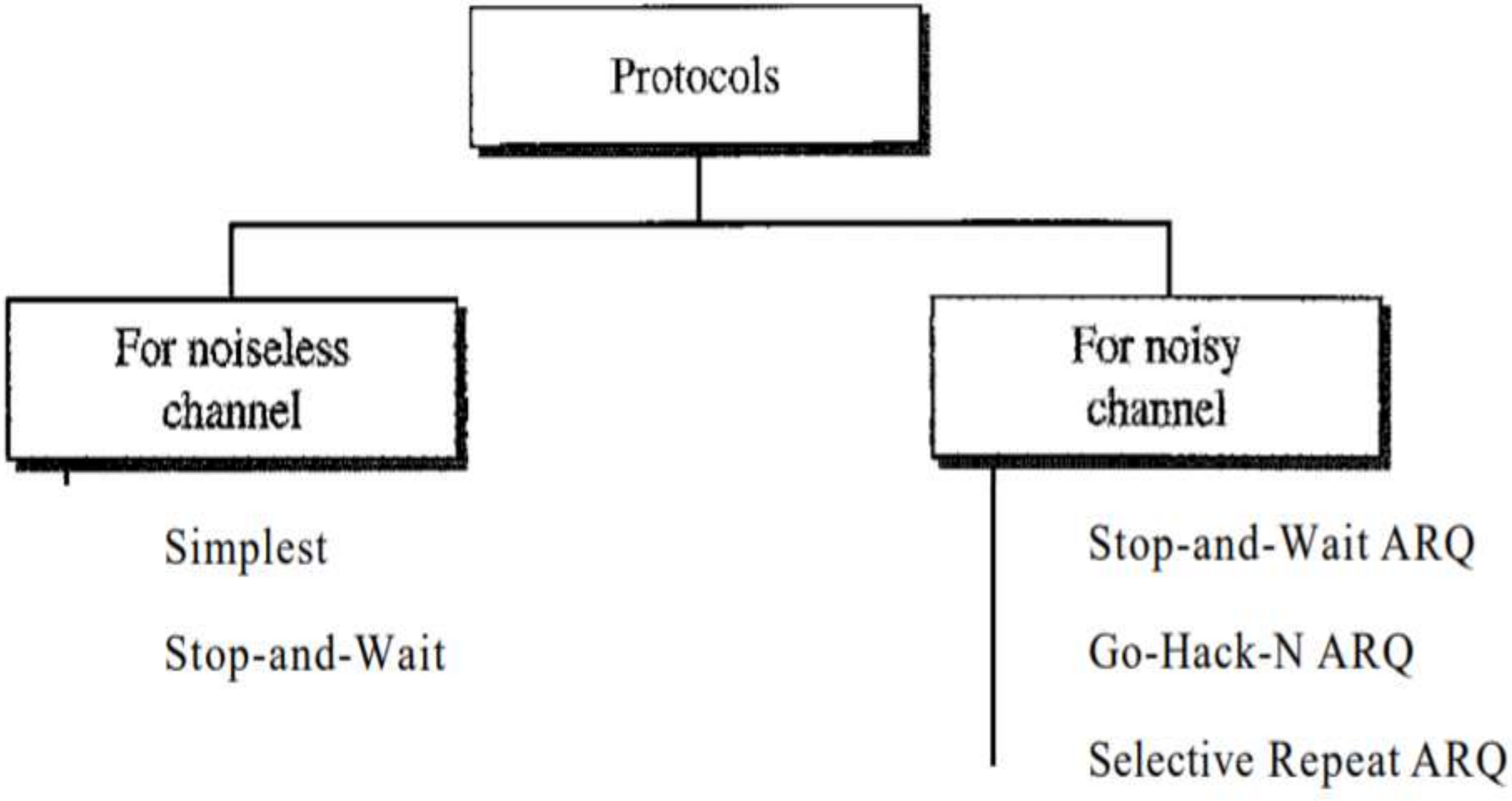
UNIT – IV
DATA LINK LAYER

CHAPTER 8: Data Link Control

4.6 NOISELESS CHANNELS

- 4.6.1 Simplest Protocol
- 4.6.2 Stop-and-Wait Protocol

UNIT – IV
DATA LINK LAYER



UNIT – IV
NOISELESS CHANNELS

4.6 NOISELESS CHANNELS

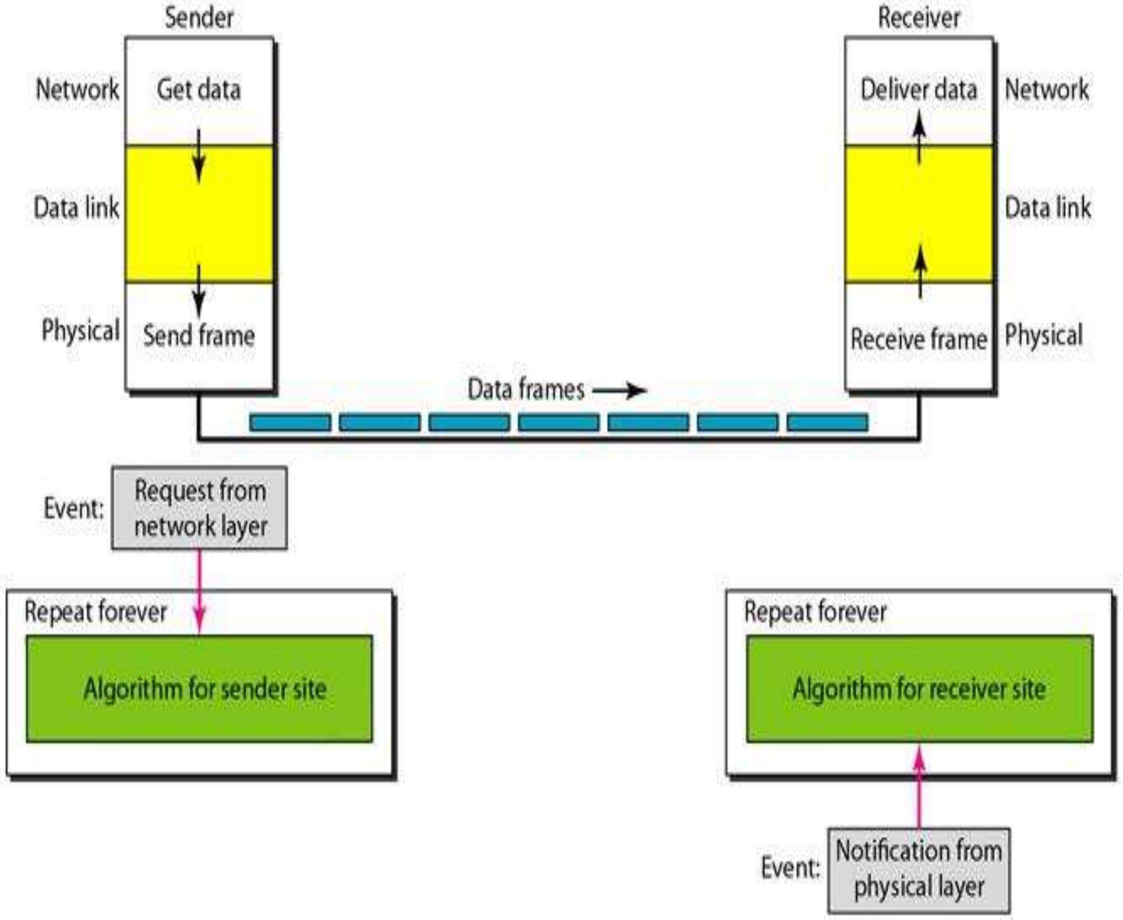
4.6.1 Simplest Protocol

- Simplest Protocol is one that has **no flow error control**, it is a **unidirectional protocol**
- The data link layer of the receiver immediately removes the header from the frame and hands the data packet to its network layer, which can also accept the packet immediately.
- In other words, the receiver can never be overwhelmed with incoming frames.
    - Design
    - Sender Site Algorithm
    - Receiver Site Algorithm
    - Example

UNIT – IV
NOISELESS CHANNELS

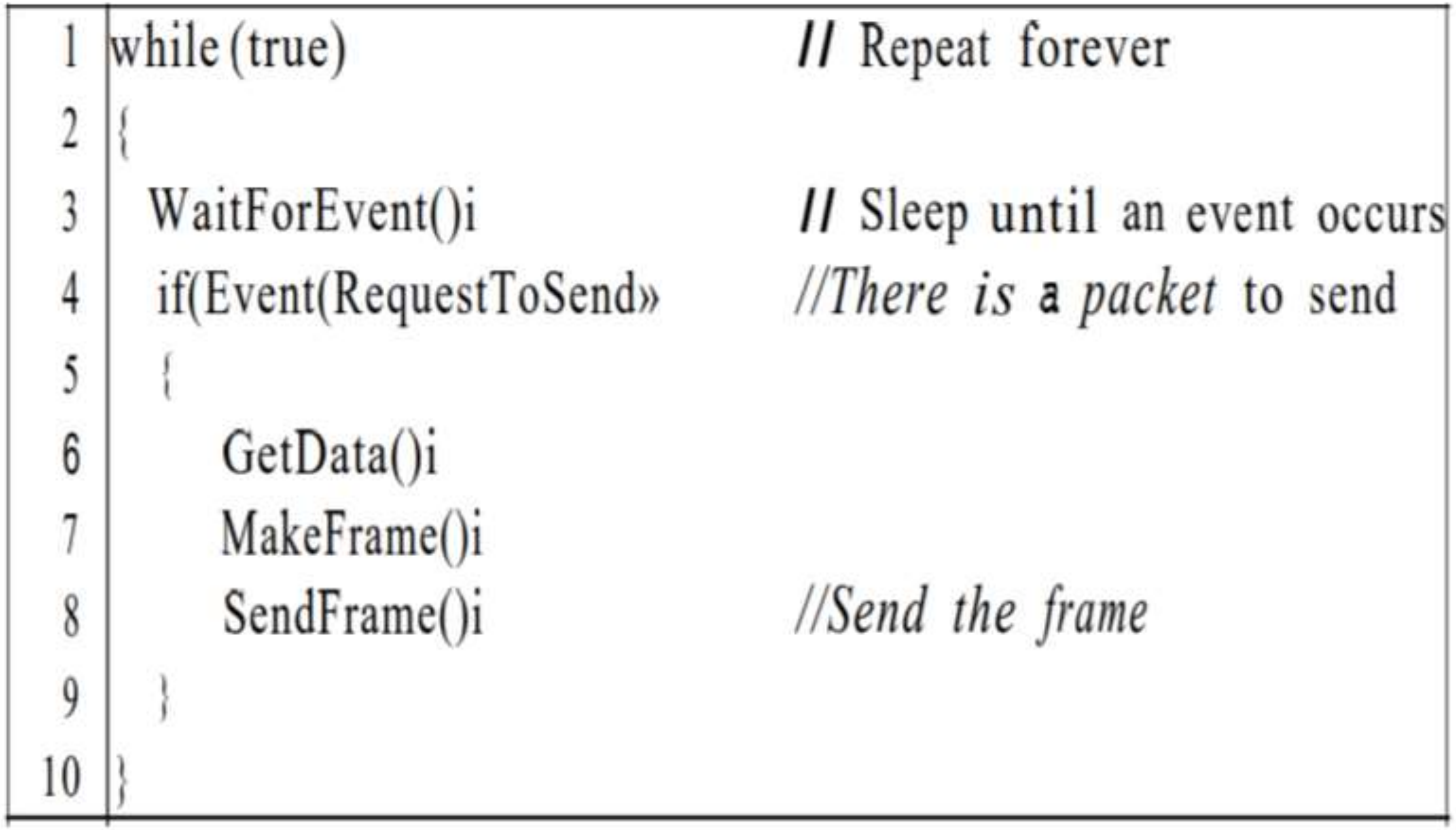
Design

- The sender site cannot send a frame until its network layer has a data packet to send.
- The receiver site cannot deliver a data packet to its network layer until a frame arrives.
- The procedure at the sender site is **constantly running;** there is no action until there is a request from the network layer.
- The procedure at the receiver site is also constantly running, but there is no action until notification from the physical layer arrives.
- Both procedures are constantly running because they do not know when the corresponding events will occur.



UNIT – IV
NOISELESS CHANNELS

Sender-site algorithm for the simplest protocol:

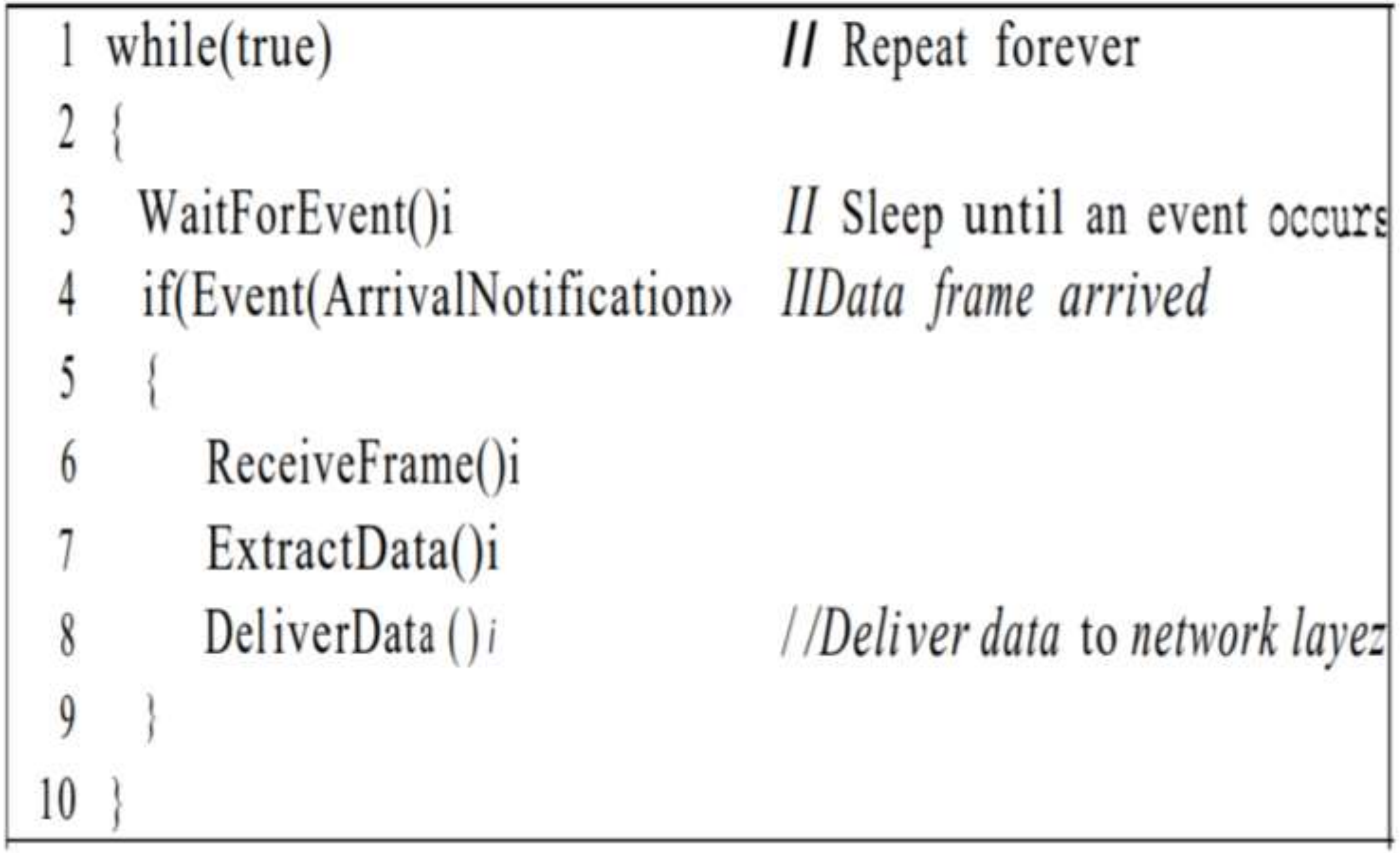


```
while(true)                      // Repeat forever
{
  WaitForEvent()i                // Sleep until an event occurs
  if(Event(RequestToSend»        //There is a packet to send
  {
     GetData()i
     MakeFrame()i
     SendFrame()i                //Send the frame
  }
}
```

Analysis: GetData() takes a data packet from the network layer, MakeFram() adds a header and delimiter flags to the data packet to make a frame, and SendFrame() delivers the frame to the physical layer for transmission.

UNIT – IV
NOISELESS CHANNELS

Receiver-site algorithm for the simplest protocol:



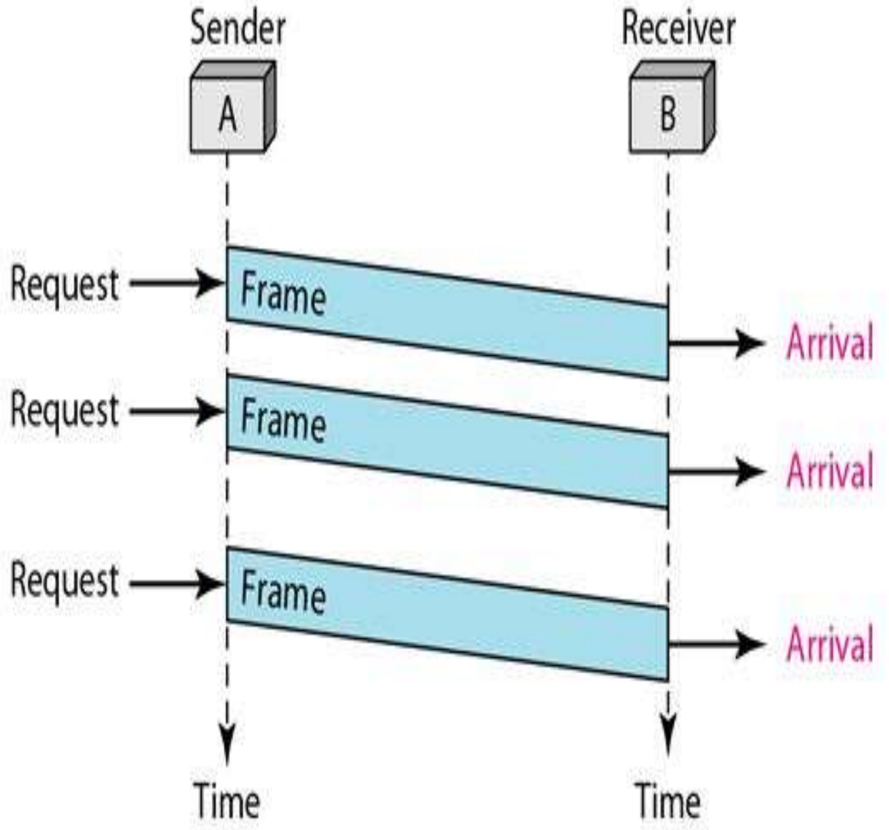
```
while(true)                              // Repeat forever
{
  WaitForEvent()i                        II Sleep until an event occurs
  if(Event(ArrivalNotification»          IIData frame arrived
  {
     ReceiveFrame()i
     ExtractData()i
     DeliverData ()i                     //Deliver data to network layez
  }
}
```

Analysis: The event here is the arrival of a data frame. After the event occurs, the data link layer receives the frame from the physical layer using the ReceiveFrame() process, extracts the data from the frame using the ExtractDat() process, and delivers the data to the network layer using the DeliverDat() process. Here, we also have an event-driven algorithm because the algorithm never knows when the data frame will arrive.

UNIT – IV
NOISELESS CHANNELS

Example4.1:

- Figure 4.13 shows an example of communication using this protocol. It is very simple.
- The sender sends a sequence of frames without even thinking about the receiver.
- To send three frames, three events occur at the sender site and three events at the receiver site.
- Note that the data frames are shown by tilted boxes; the height of the box defines the transmission time difference between the first bit and the last bit in the frame.



UNIT – IV
NOISELESS CHANNELS

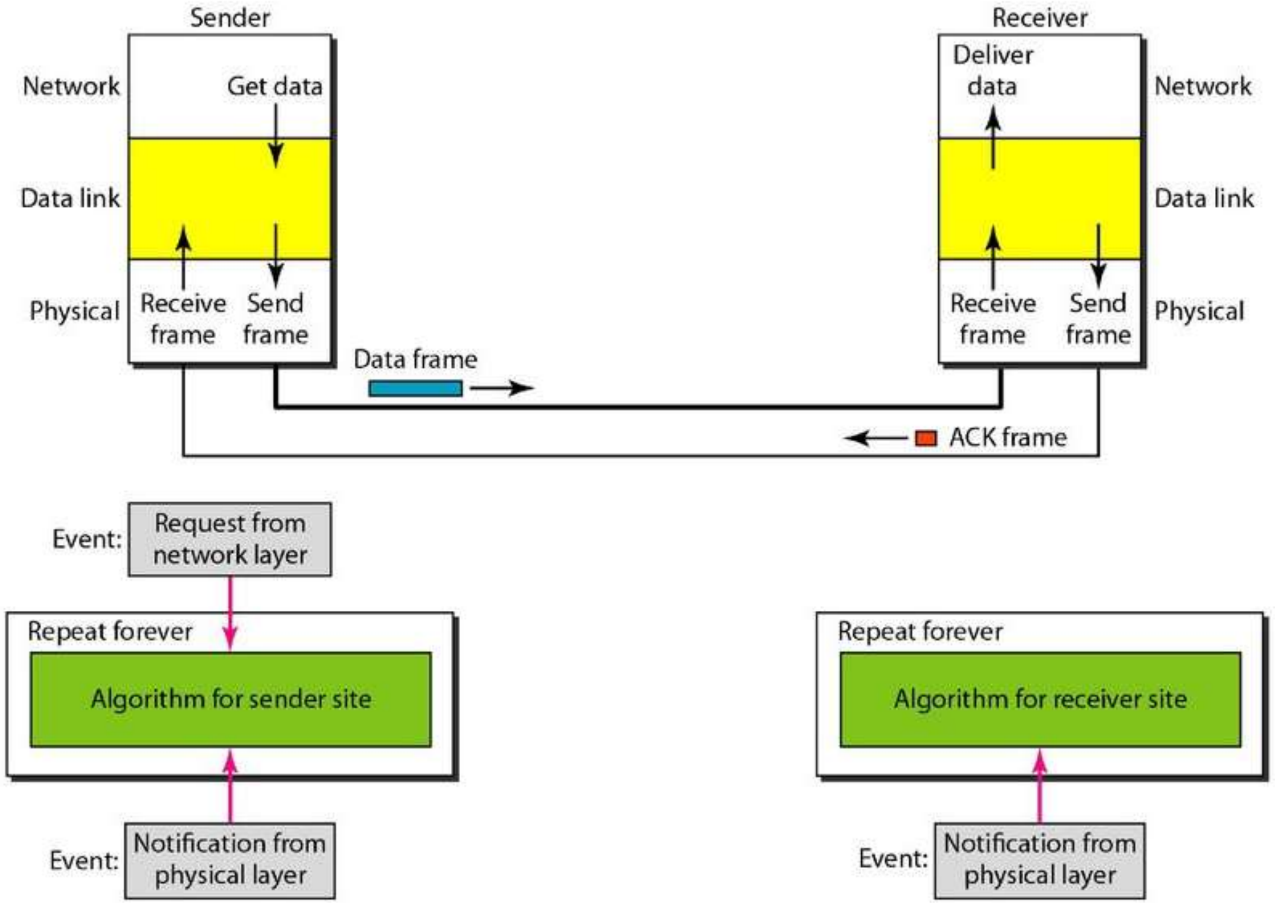
4.6.2 Stop-and-Wait Protocol

- If data frames arrive at the receiver site faster than they can be processed, the frames **must be stored** until their use.
- To prevent the receiver from **becoming overwhelmed** with frames, we somehow need to tell the sender to slow down.
- In Stop-and-Wait Protocol because the sender sends one frame, **stops until it receives** confirmation from the receiver, and then sends the next frame.
- We still have unidirectional communication for data frames, but auxiliary **ACK frames** travel from the other direction.
    - ➢ Design
    - ➢ Sender Site Algorithm
    - ➢ Receiver Site Algorithm
    - ➢ Example

UNIT – IV
NOISELESS CHANNELS

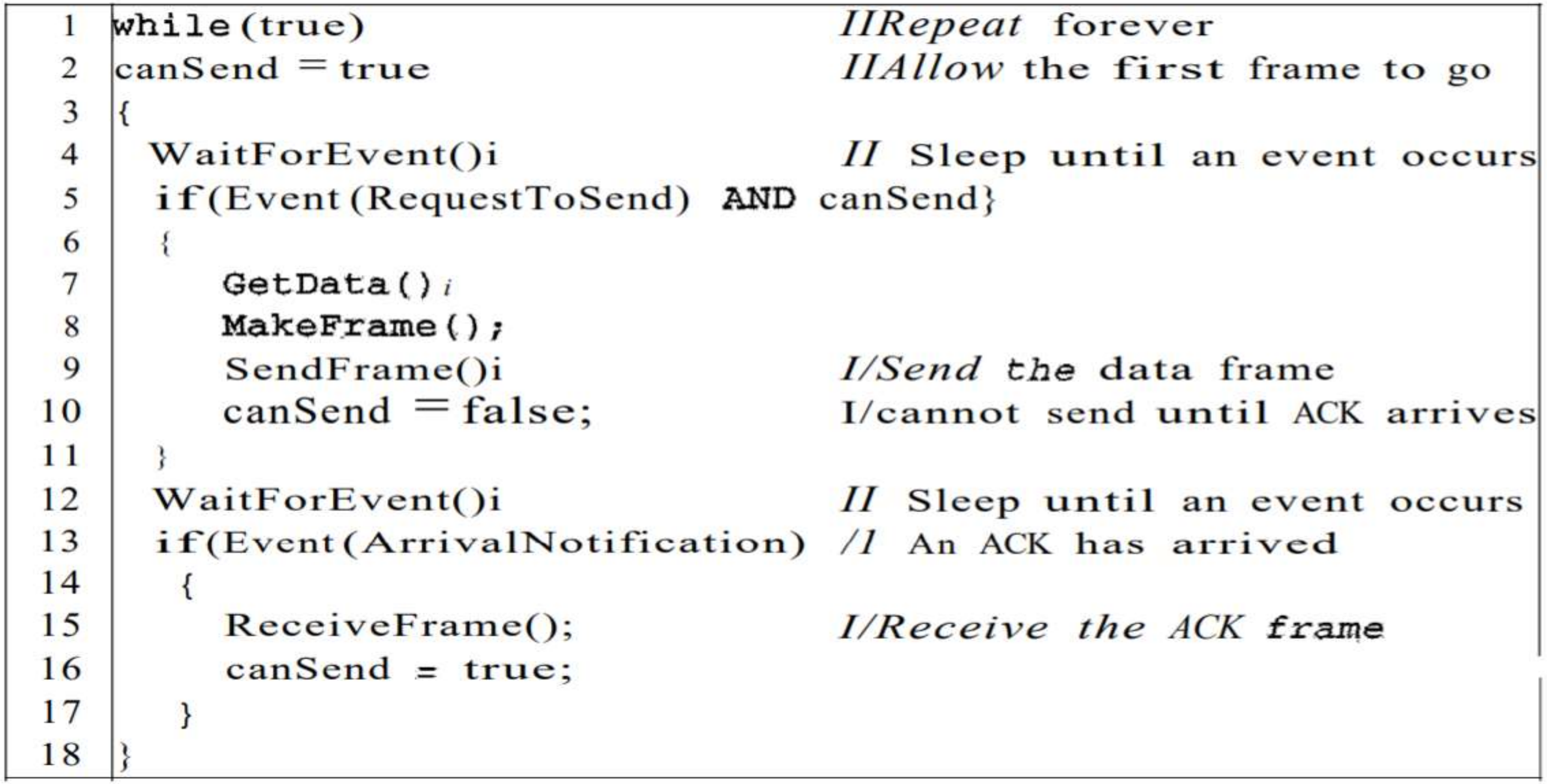
Design

- Figure 4.14 illustrates the mechanism. Comparing this figure with Figure 4.12, we can see the traffic on the forward channel (from sender to receiver) and the reverse channel.
- At any time, there is either one data frame on the forward channel or one ACK frame on the reverse channel. We therefore need a half-duplex link.



UNIT – IV
NOISELESS CHANNELS

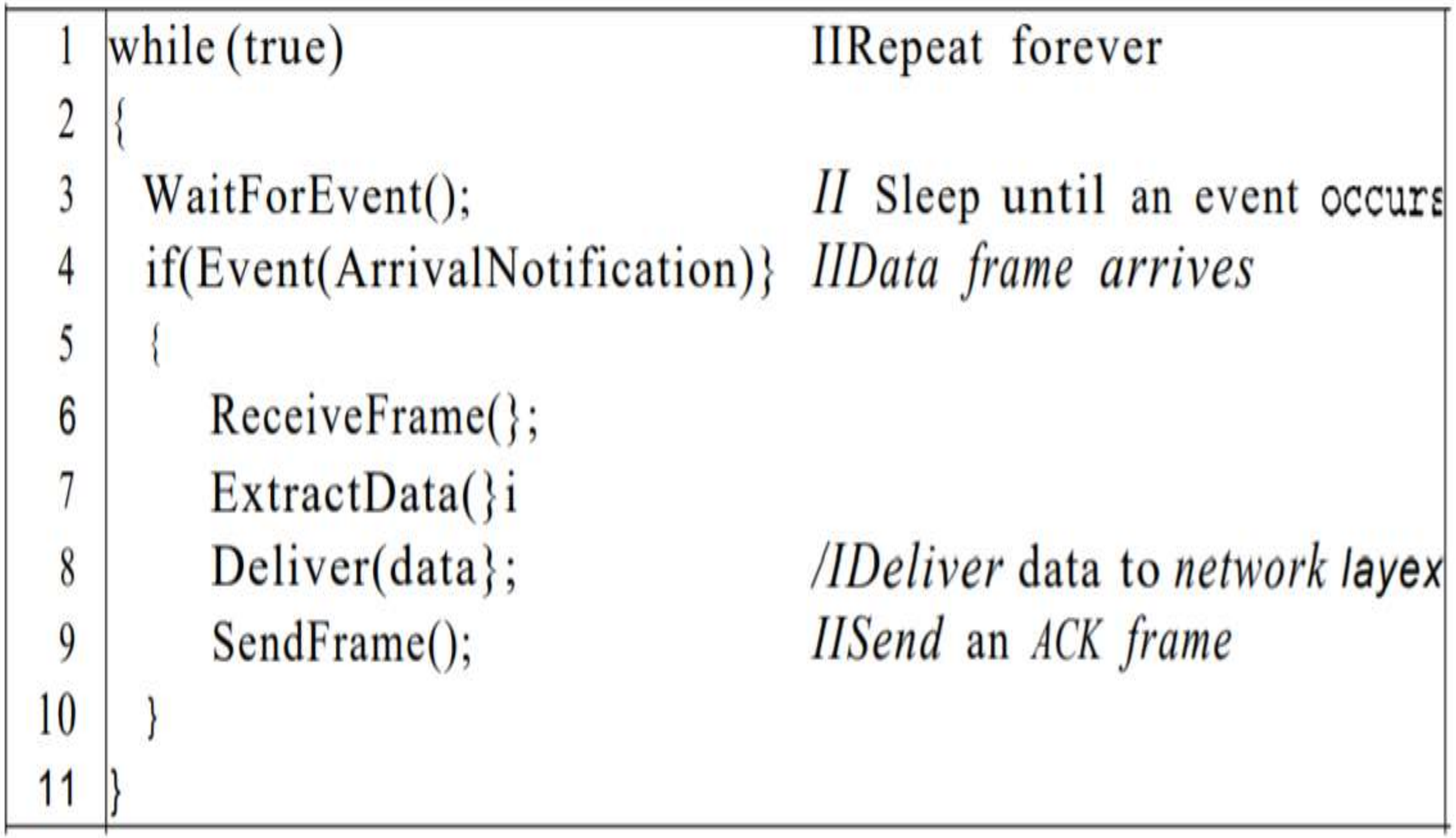
Sender-site algorithm for the stop & Wait protocol:



```
while(true)                                 IIRepeat forever
canSend = true                              IIAllow the first frame to go
{
  WaitForEvent()i                           II Sleep until an event occurs
  if(Event(RequestToSend) AND canSend}
  {
     GetData()i
     MakeFrame();
     SendFrame()i                           I/Send the data frame
     canSend = false;                       I/cannot send until ACK arrives
  }
  WaitForEvent()i                           II Sleep until an event occurs
  if(Event(ArrivalNotification)             /1 An ACK has arrived
   {
     ReceiveFrame();                        I/Receive the ACK frame
     canSend = true;
   }
}
```

UNIT – IV
NOISELESS CHANNELS

Receiver-site algorithm for the stop & Wait protocol:

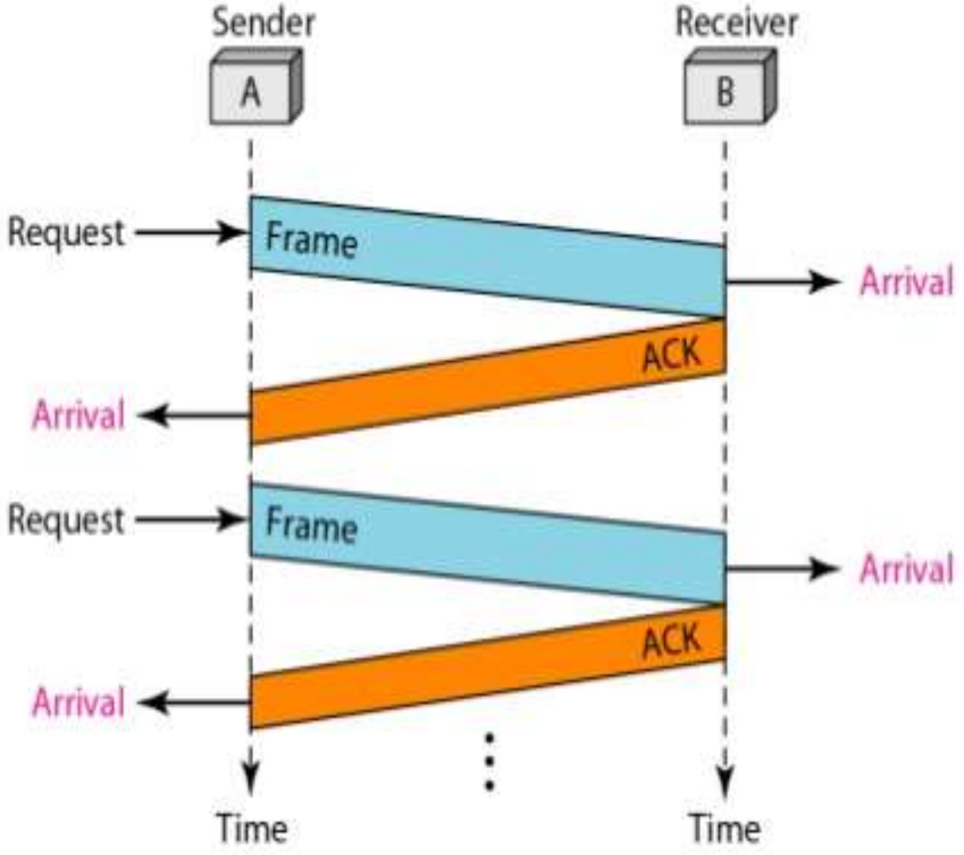


```
while (true)                          IIRepeat  forever
{
  WaitForEvent();                     II Sleep until an event occurs
  if(Event(ArrivalNotification)}      IIData  frame  arrives
  {
      ReceiveFrame(};
      ExtractData(}i
      Deliver(data};                  /IDeliver data to network layex
      SendFrame();                    IISend  an  ACK  frame
  }
}
```

UNIT – IV
NOISELESS CHANNELS

Example 4.2:

- Figure 4.15 shows an example of communication using this protocol.
- It is still very simple.
- The sender sends one frame and waits for feedback from the receiver.
- When the ACK arrives, the sender sends the next frame.
- Note that sending two frames in the protocol involves the sender in four events and the receiver in two events.



DATA COMMUNICATIONS & NETWORKING

B.SC (COMPUTER SCIENCE) – III YEAR

SMCS52

By
Dr.N.N.Krishna Veni,
Assistant Professor,
Department of Computer Science,
Holy Cross Home Science College,
Thoothukudi

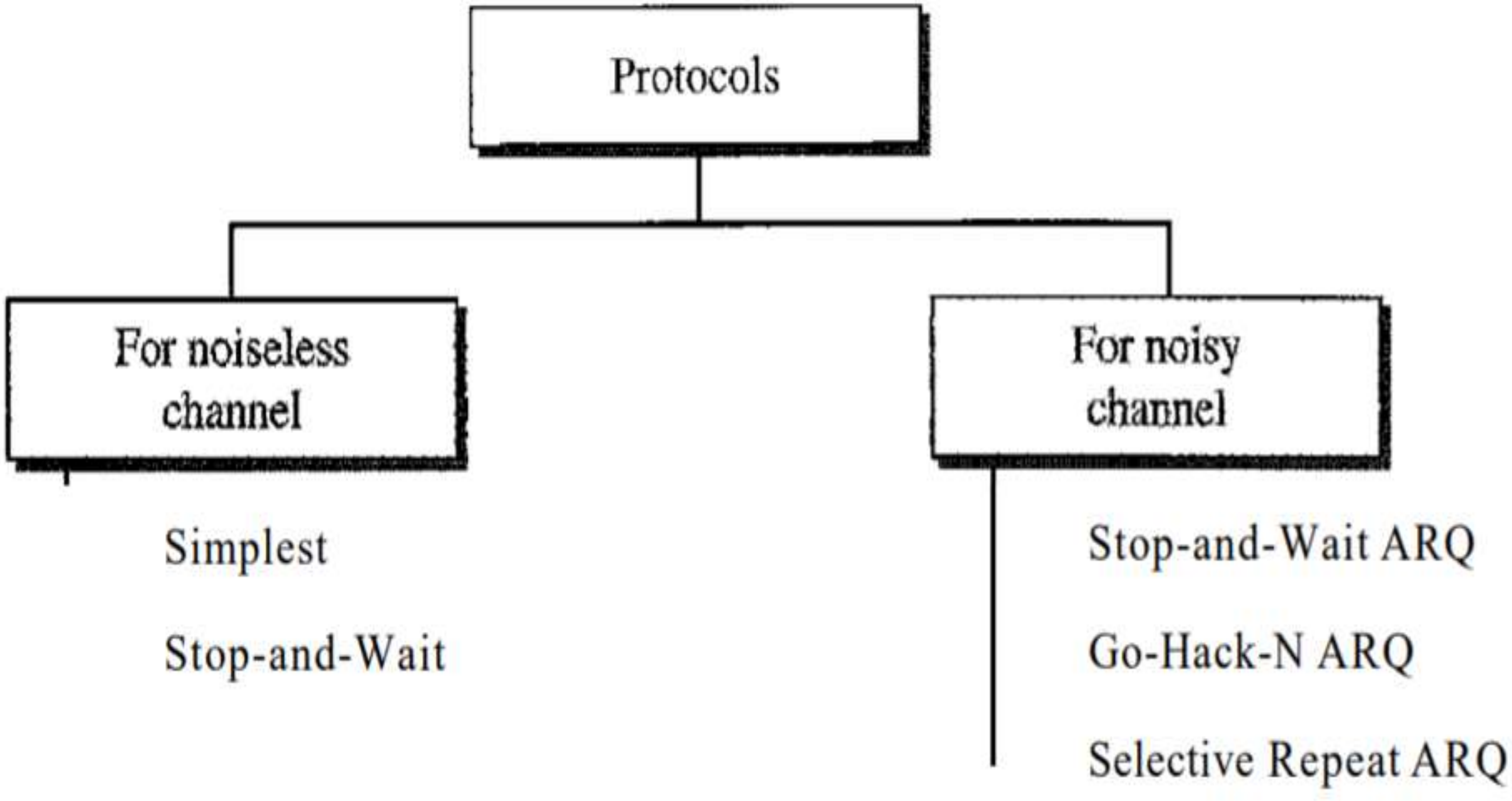
UNIT – IV
DATA LINK LAYER

CHAPTER 8: Data Link Control

4.7 NOISY CHANNELS

4.7.1 Stop-and-Wait Automatic Repeat Request

UNIT – IV
DATA LINK LAYER



Protocols

For noiseless channel

- Simplest
- Stop-and-Wait

For noisy channel

- Stop-and-Wait ARQ
- Go-Hack-N ARQ
- Selective Repeat ARQ

4.7 NOISY CHANNELS

4.7.1 Stop-and-Wait Automatic Repeat Request

- It adds a simple **error control mechanism** to the Stop-and-Wait Protocol.
- Lost frames are more difficult to handle than corrupted ones.
- The received frame could be the correct one, or a duplicate, or a frame out of order.
- The **solution is to number the frames**.
- When the receiver receives a data frame that is out of order, this means that frames were either lost or duplicated.
- The completed and lost frames need to be resent in this protocol. If the receiver does not respond when there is an error, how can the sender know which frame to resend?
- To remedy this problem, the sender keeps a copy of the sent frame.
- At the same time, it starts a timer.

UNIT – IV
NOISELESS CHANNELS

- If the timer expires and there is no ACK for the sent frame, the frame is resent, the copy is held, and the timer is restarted.
- Since an **ACK frame can also be corrupted and lost,** it too needs redundancy bits and a sequence number.
- The ACK frame for this protocol **has a sequence number field**.

SEQUENCE NUMBERS

- A field is added to the data frame to hold the sequence number of that frame.
- Since we want to minimize the frame size, we look for the smallest range
- For example, if we decide that the field is m bits long, the sequence numbers start from 0, go to 2m - 1, and then are repeated.
- Assume we have used x as a sequence number; we only need to use x + 1 after that.
- There is no need for x + 2.

UNIT – IV
NOISELESS CHANNELS

To show this, assume that the sender has sent the frame numbered x. Three things can happen.

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

we can let x = 0 and x + 1 == 1. This means that the sequence is 0, 1, 0, 1, 0, and so on

UNIT – IV
NOISELESS CHANNELS

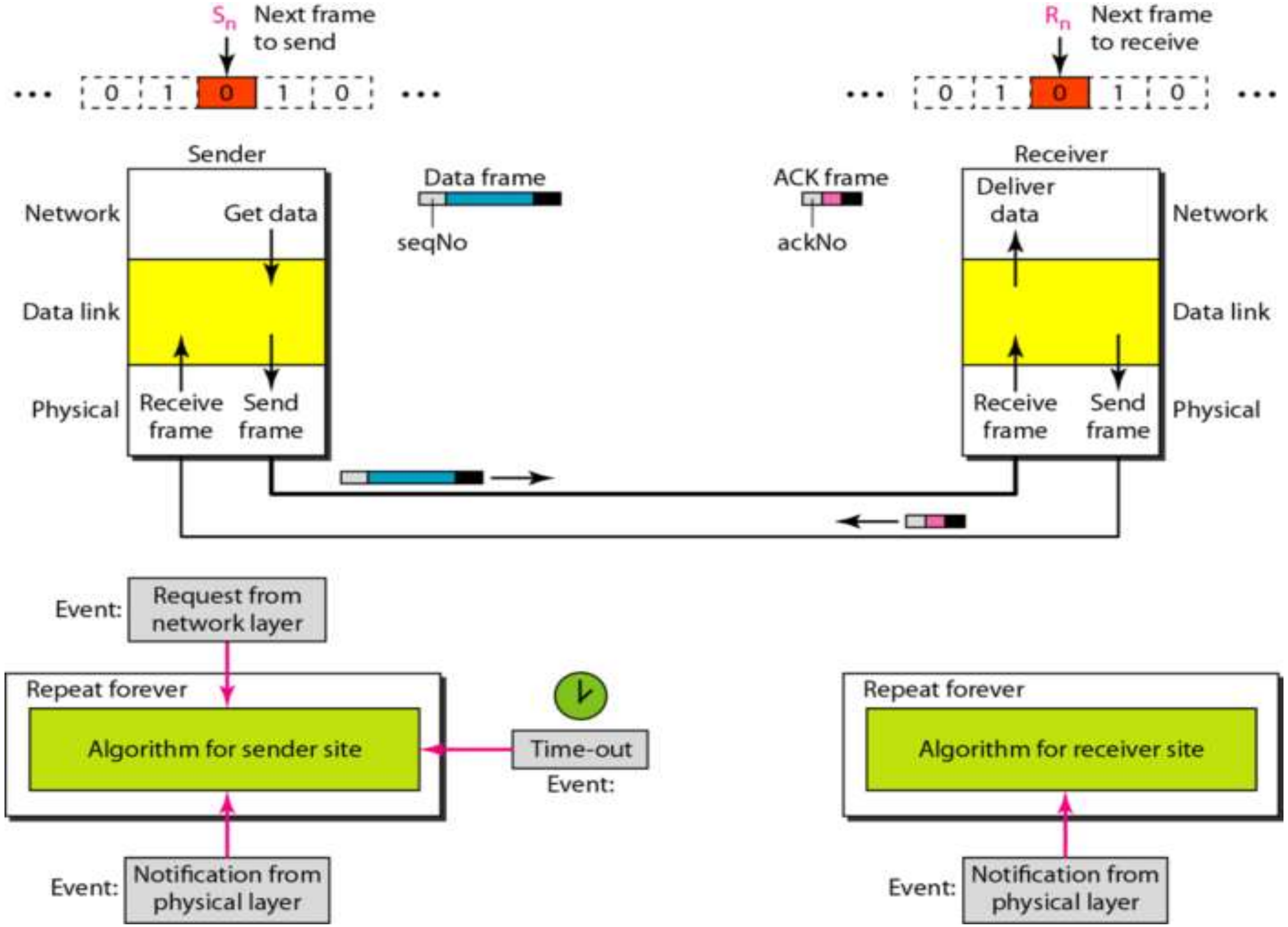
ACKNOWLEDGMENT NUMBERS

For example, if frame 0 has arrived safe and sound, the receiver sends an ACK frame with acknowledgment 1. If frame 1 has arrived safe and sound, the receiver sends an ACK frame with acknowledgment 0.

- Design
- Sender Site Algorithm
- Receiver Site Algorithm
- Example

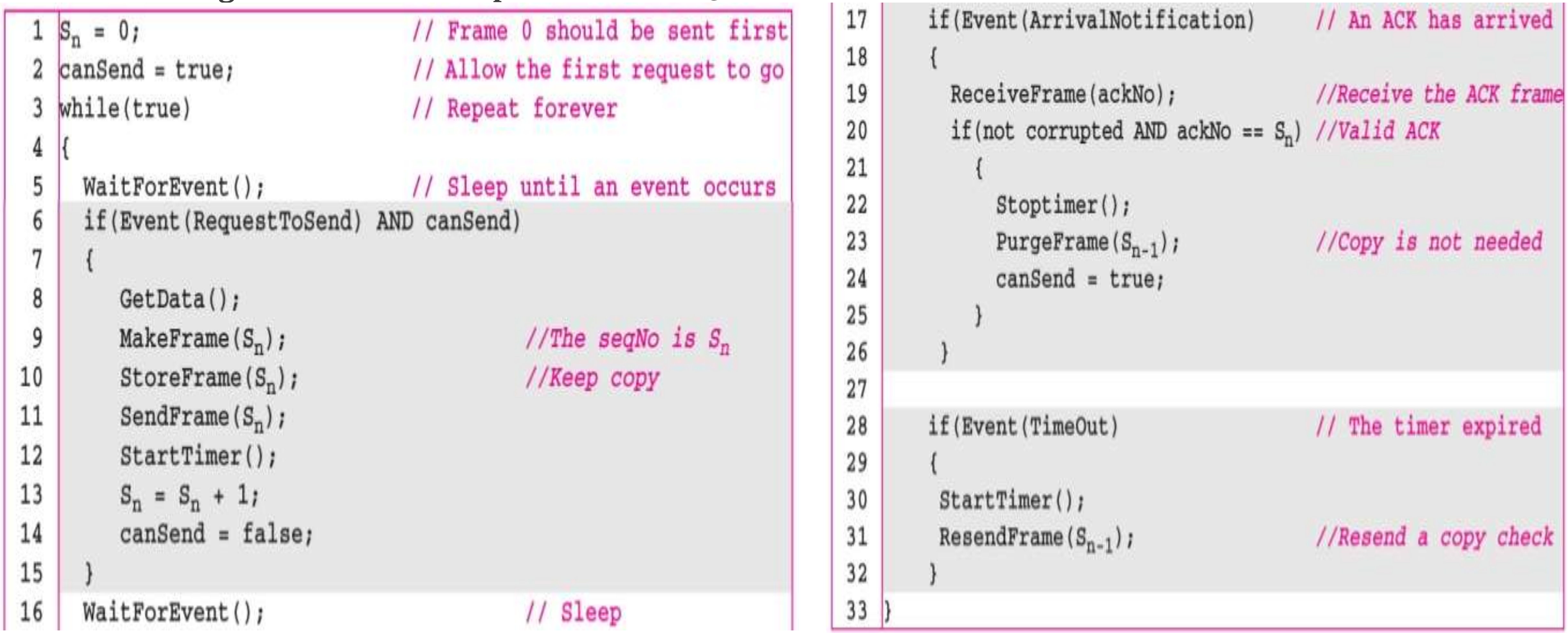
UNIT – IV
NOISELESS CHANNELS

Design



UNIT – IV
NOISELESS CHANNELS

Sender-site algorithm for the Stop-and-Wait ARQ:

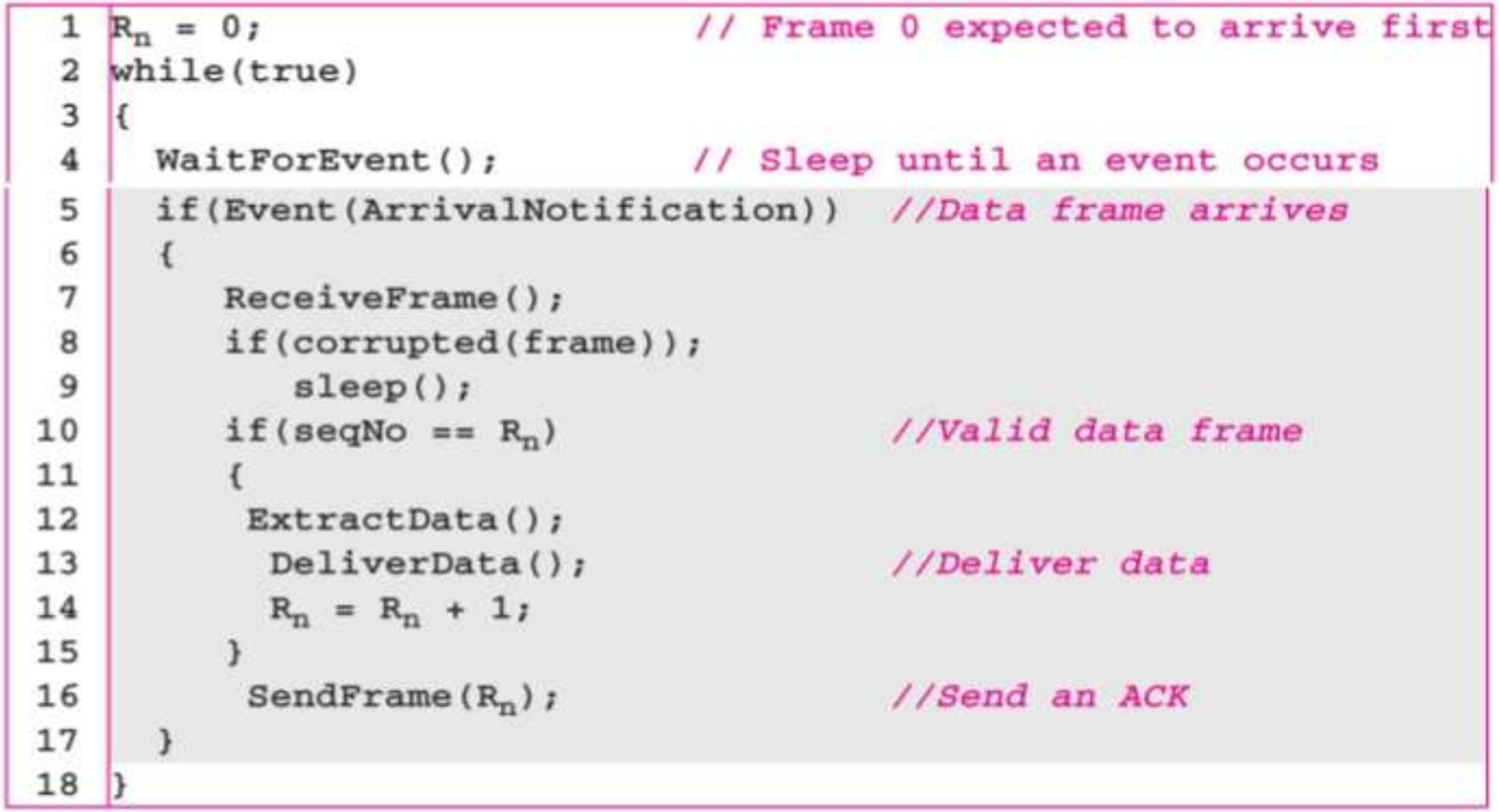


```
$S_n$ = 0;                        // Frame 0 should be sent first
canSend = true;                   // Allow the first request to go
while(true)                       // Repeat forever
{
  WaitForEvent();                 // Sleep until an event occurs
  if(Event(RequestToSend) AND canSend)
  {
     GetData();
     MakeFrame($S_n$);                   //The seqNo is $S_n$
     StoreFrame($S_n$);                  //Keep copy
     SendFrame($S_n$);
     StartTimer();
     $S_n$ = $S_n$ + 1;
     canSend = false;
  }
  WaitForEvent();                        // Sleep
  if(Event(ArrivalNotification)     // An ACK has arrived
  {
    ReceiveFrame(ackNo);            //Receive the ACK frame
    if(not corrupted AND ackNo == $S_n$) //Valid ACK
      {
        Stoptimer();
        PurgeFrame($S_{n-1}$);          //Copy is not needed
        canSend = true;
      }
   }

  if(Event(TimeOut)                 // The timer expired
  {
   StartTimer();
   ResendFrame($S_{n-1}$);          //Resend a copy check
  }
}
```

UNIT – IV

NOISELESS CHANNELS

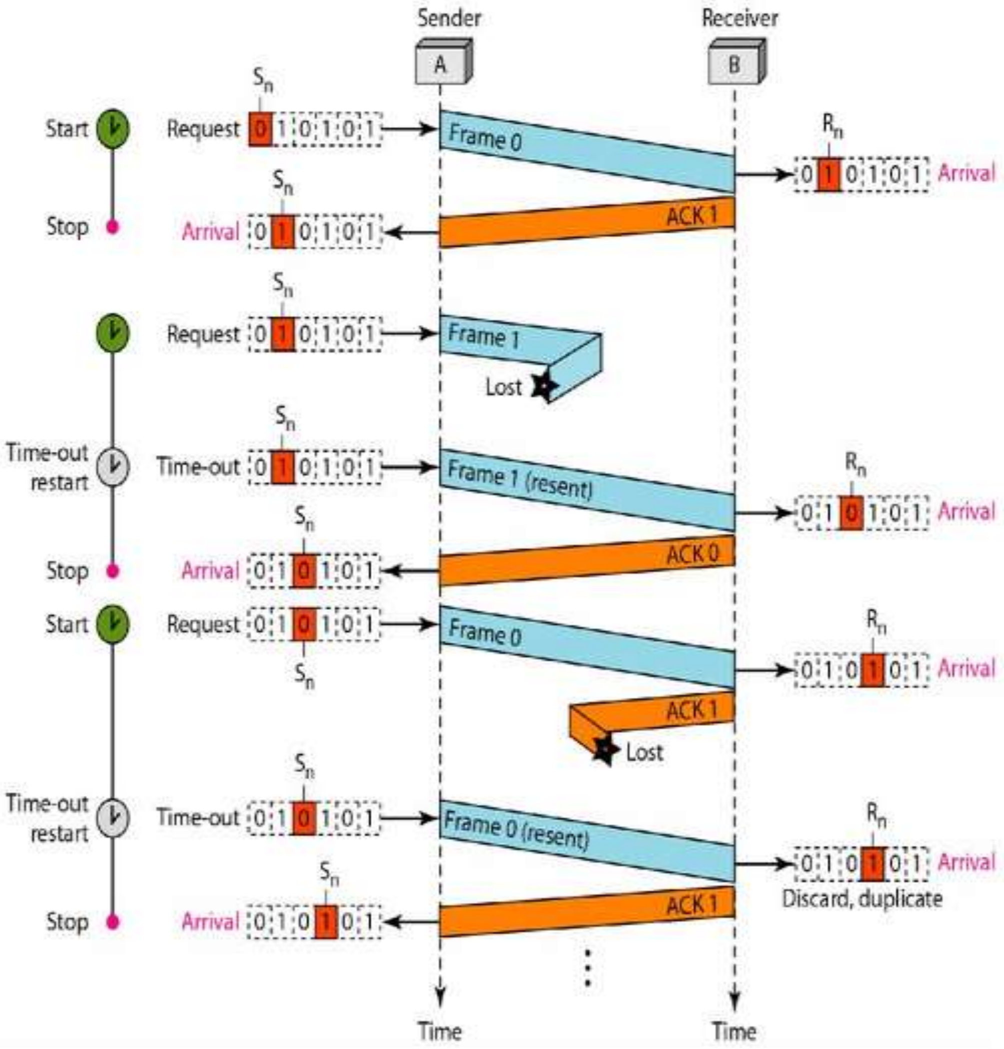
Sender-site algorithm for the Stop-and-Wait ARQ:



```
Rn = 0;                        // Frame 0 expected to arrive first
while(true)
{
  WaitForEvent();              // Sleep until an event occurs
  if(Event(ArrivalNotification))  //Data frame arrives
  {
     ReceiveFrame();
     if(corrupted(frame));
        sleep();
     if(seqNo == Rn)                //Valid data frame
     {
      ExtractData();
       DeliverData();               //Deliver data
       Rn = Rn + 1;
     }
      SendFrame(Rn);                //Send an ACK
  }
}
```

UNIT – IV
NOISELESS CHANNELS

Example



DATA COMMUNICATIONS & NETWORKING

B.SC (COMPUTER SCIENCE) – III YEAR

SMCS52

By

Dr.N.N.Krishna Veni,
Assistant Professor,
Department of Computer Science,
Holy Cross Home Science College,
Thoothukudi

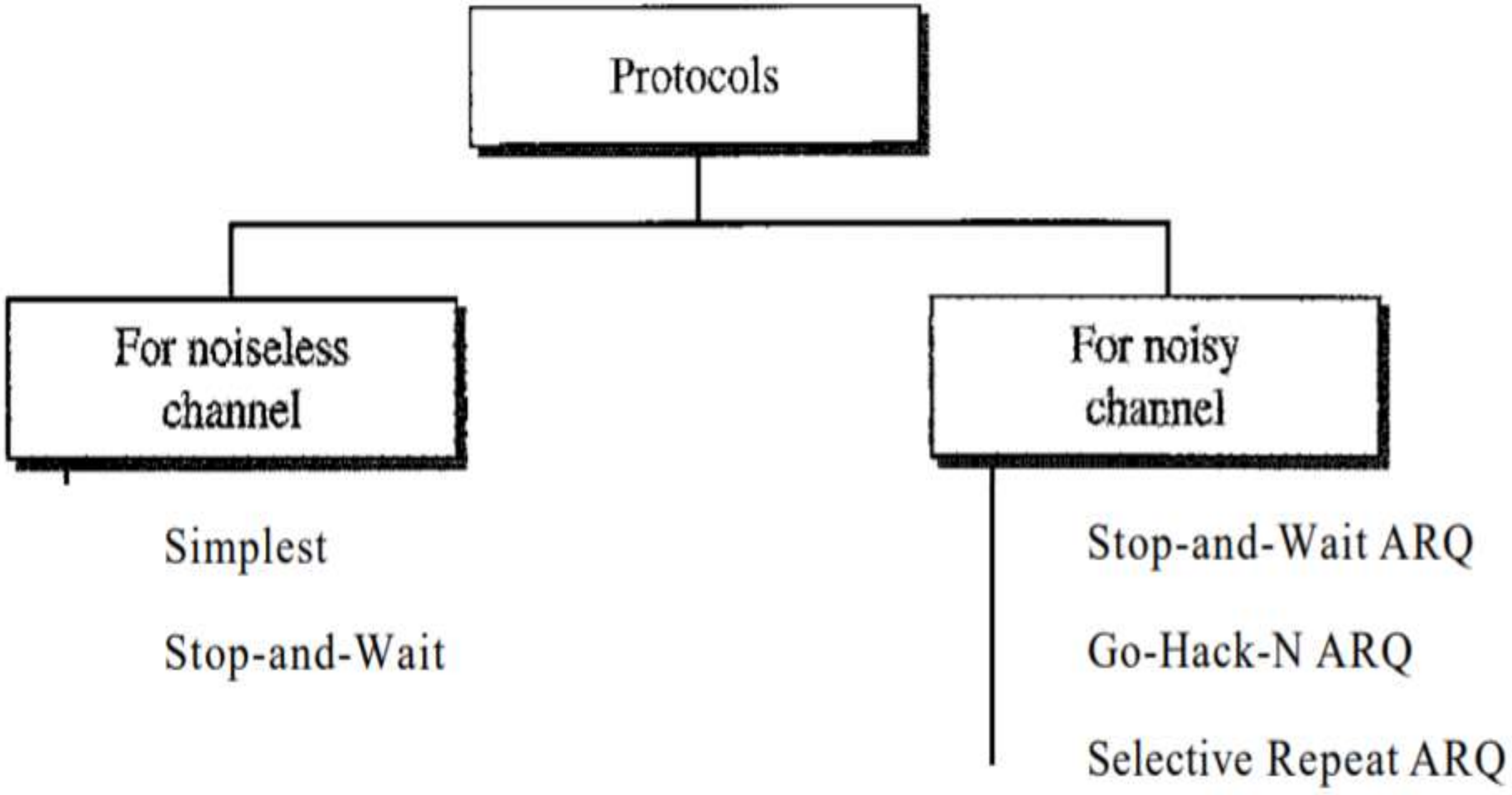
UNIT – IV
DATA LINK LAYER

CHAPTER 8: Data Link Control

4.7 NOISY CHANNELS

4.7.2 Go-Back-N Automatic Repeat Request

UNIT – IV
DATA LINK LAYER



Protocols

For noiseless channel

Simplest

Stop-and-Wait

For noisy channel

Stop-and-Wait ARQ

Go-Hack-N ARQ

Selective Repeat ARQ

UNIT – IV
NOISELESS CHANNELS

4.7 NOISY CHANNELS

4.7.2 Go-Back-N Automatic Repeat Request

- To improve the efficiency of transmission, multiple frames must be in transition while waiting for acknowledgment.
- In this protocol we can send several frames before receiving acknowledgments; we keep a copy of these frames until the acknowledgments arrive.

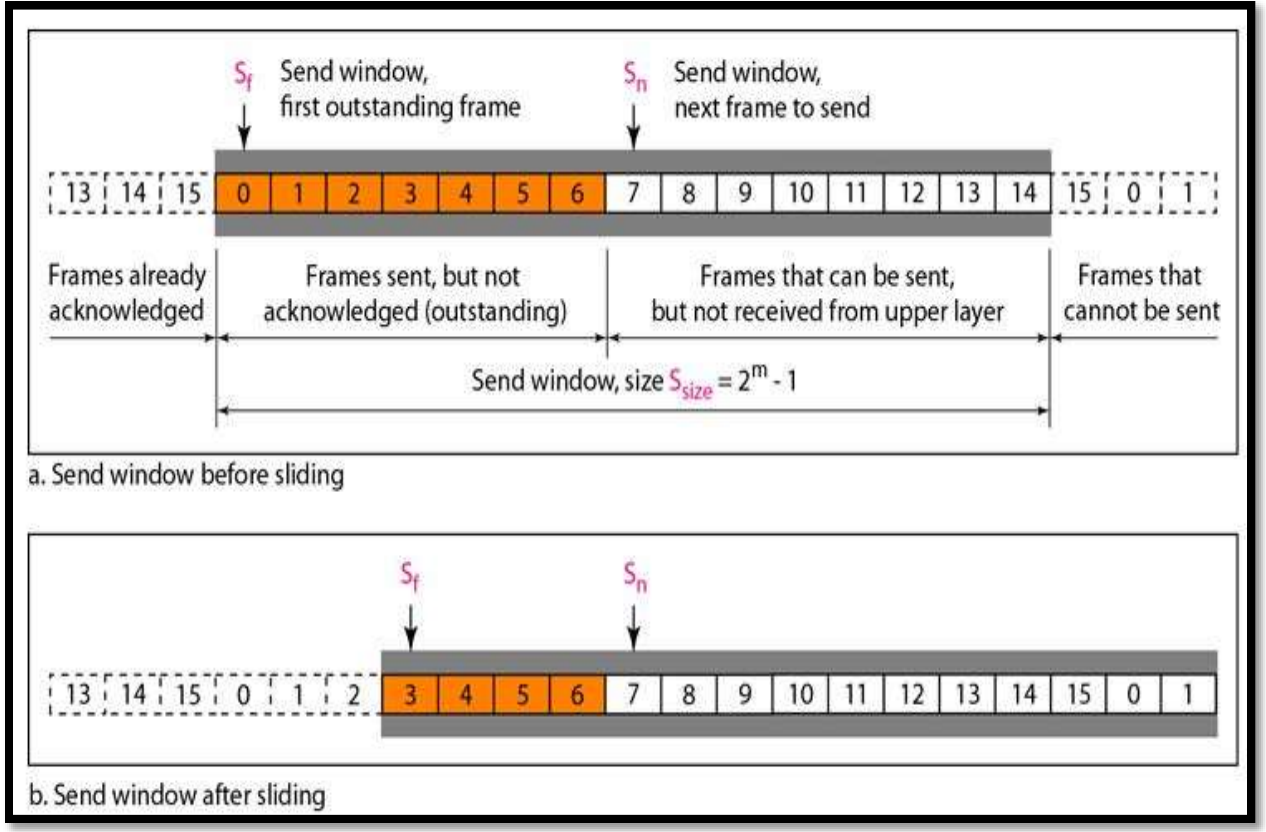
Sequence Numbers: Frames from a sending station are numbered sequentially. If the header of the frame allows m bits for the sequence number, the sequence numbers range from 0 to 2^m - 1.

- For example, if m is 4, the only sequence numbers are 0 through 15 inclusive. However, we can repeat the sequence. So the sequence numbers are

0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15, 0,1,2,3,4,5,6,7,8,9,10,11, ...

UNIT – IV
NOISELESS CHANNELS

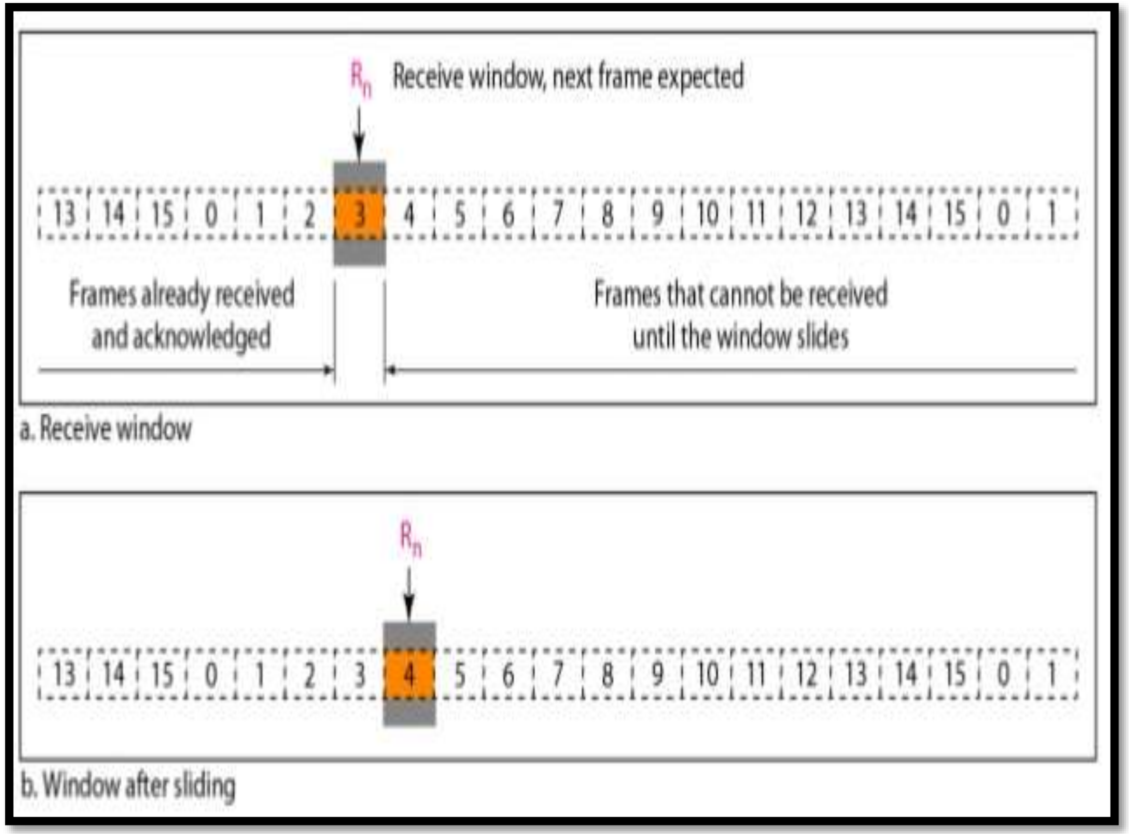
Sliding Window: The sliding window is the range of sequence numbers that is the concern of the sender and receiver. The range which is the concern of the sender is called the send sliding window; the range that is the concern of the receiver is called the receive sliding window.



S_f Send window, first outstanding frame
S_n Send window, next frame to send
13 14 15 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 0 1
Frames already acknowledged
Frames sent, but not acknowledged (outstanding)
Frames that can be sent, but not received from upper layer
Frames that cannot be sent
Send window, size $S_{size} = 2^m - 1$
S_f
S_n
13 14 15 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 0 1

a. Send window before sliding

b. Send window after sliding



R_n Receive window, next frame expected
13 14 15 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 0 1
Frames already received and acknowledged
Frames that cannot be received until the window slides
R_n
13 14 15 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 0 1

a. Receive window

b. Window after sliding

UNIT – IV
NOISELESS CHANNELS

- **Timers:** Although there can be a timer for each frame that is sent, in our protocol we use only one. The reason is that the timer for the first outstanding frame always expires first; we send all outstanding frames when this timer expires.

- **Acknowledgment:** The receiver sends a positive acknowledgment if a frame has arrived safe and sound and in order. If a frame is damaged or is received out of order, the receiver is silent and will discard all subsequent frames until it receives the one it is expecting. The silence of the receiver causes the timer of the unacknowledged frame at the sender site to expire. This, in turn, causes the sender to go back and resend all frames, beginning with the one with the expired timer. The receiver does not have to acknowledge each frame received. It can send one cumulative acknowledgment for several frames.

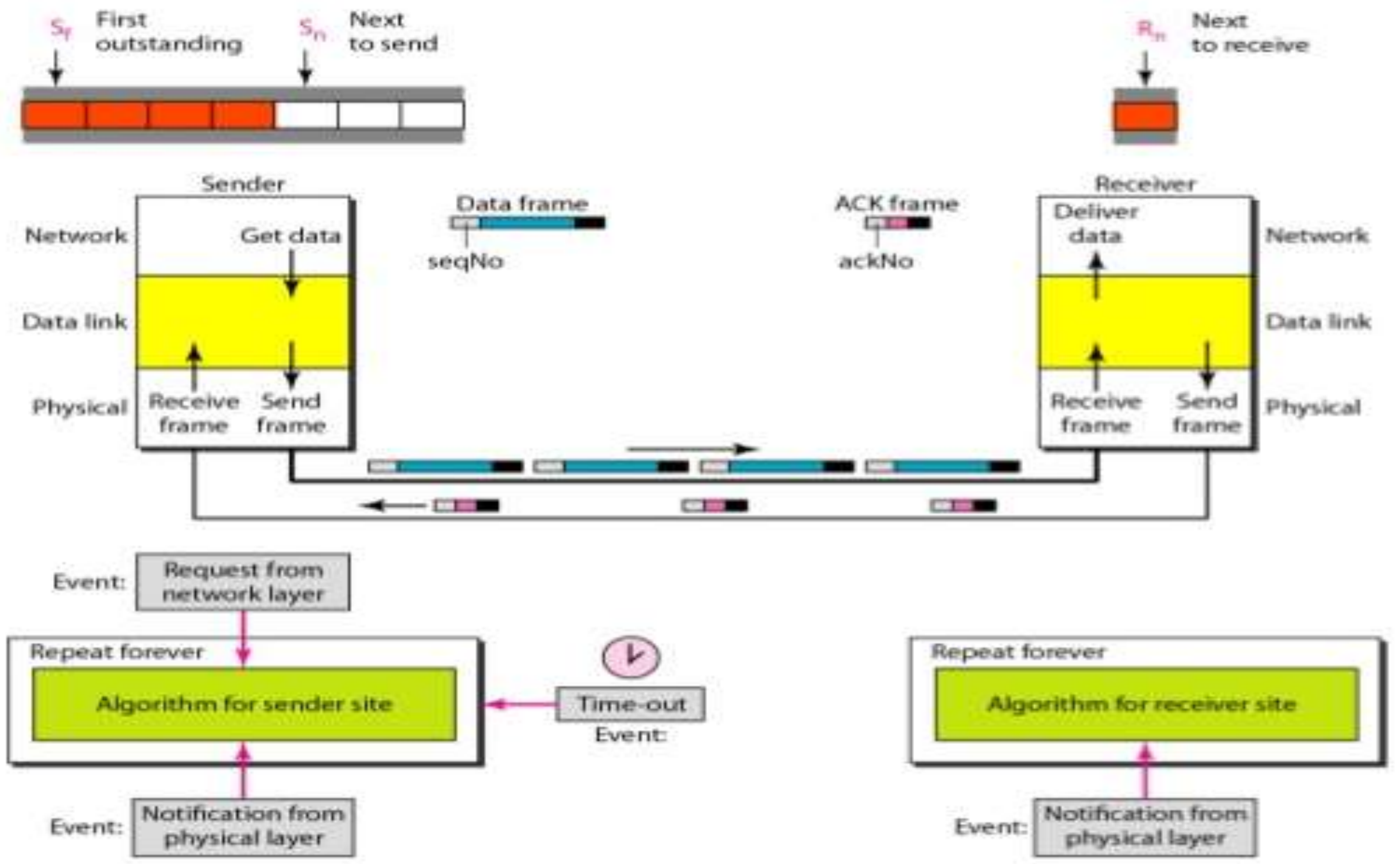
- **Resending a Frame** When the timer expires, the sender resends all outstanding frames. For example, suppose the sender has already sent frame 6, but the timer for frame 3 expires. This means that frame 3 has not been acknowledged; the sender goes back and sends frames 3, 4,5, and 6 again. That is why the protocol is called Go-Back-N ARQ.

UNIT – IV
NOISELESS CHANNELS

- Design
- Sender Site Algorithm
- Receiver Site Algorithm
- Example

UNIT – IV
NOISELESS CHANNELS

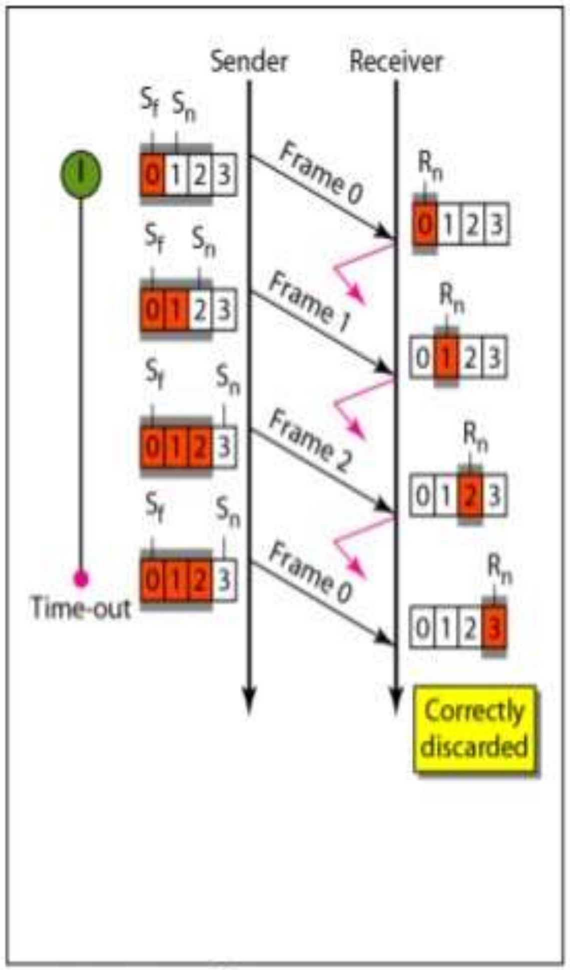
Design



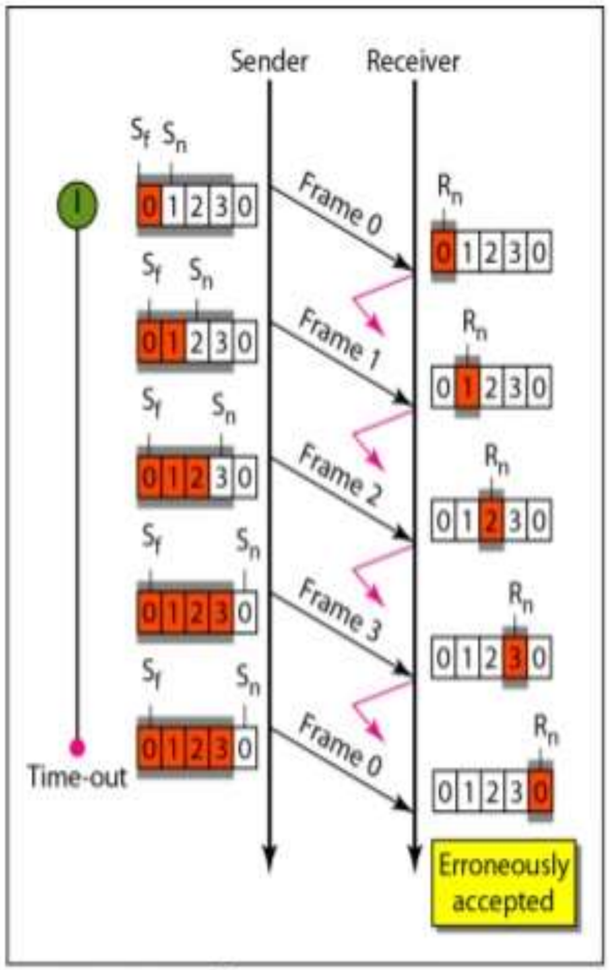
UNIT – IV
NOISELESS CHANNELS

Window size for Go-Back-N ARQ

why the size of the send window must be less than 2^m



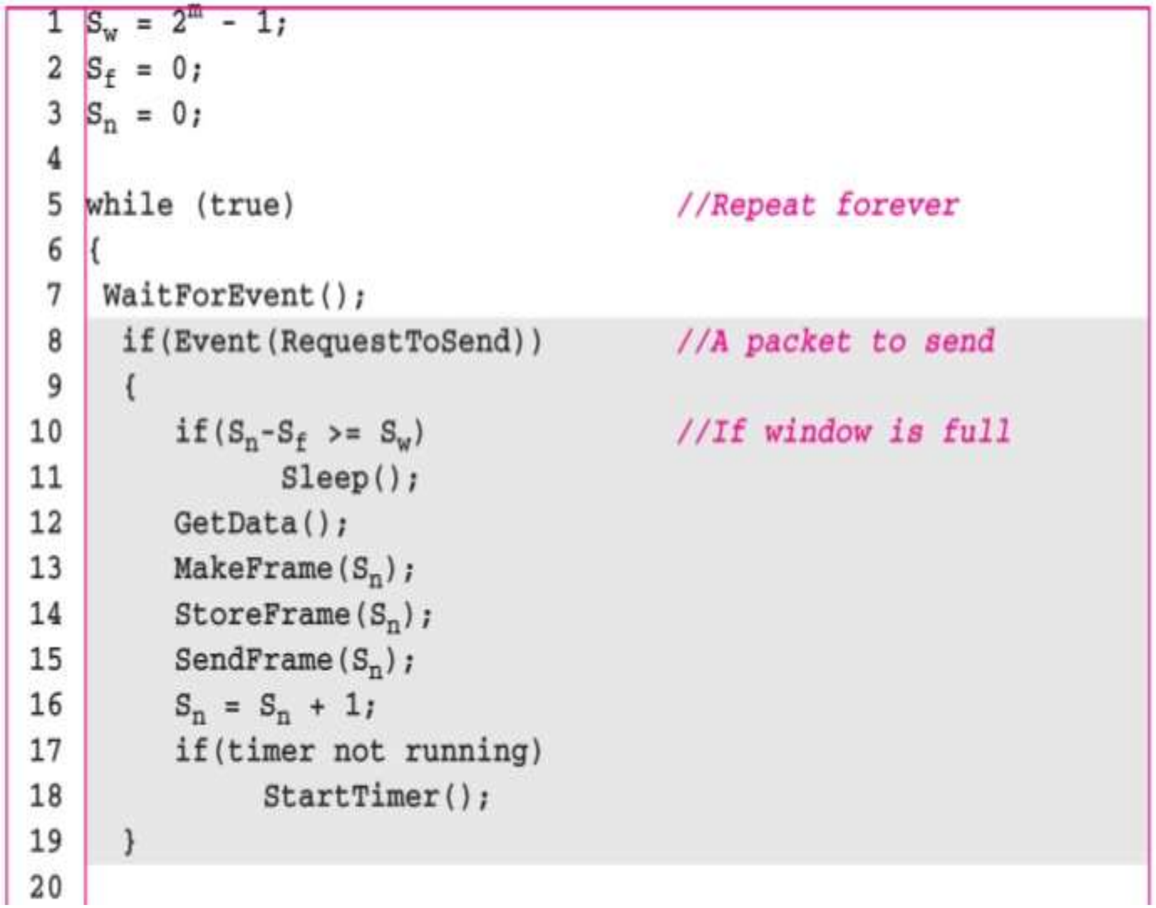
a. Window size < 2^m



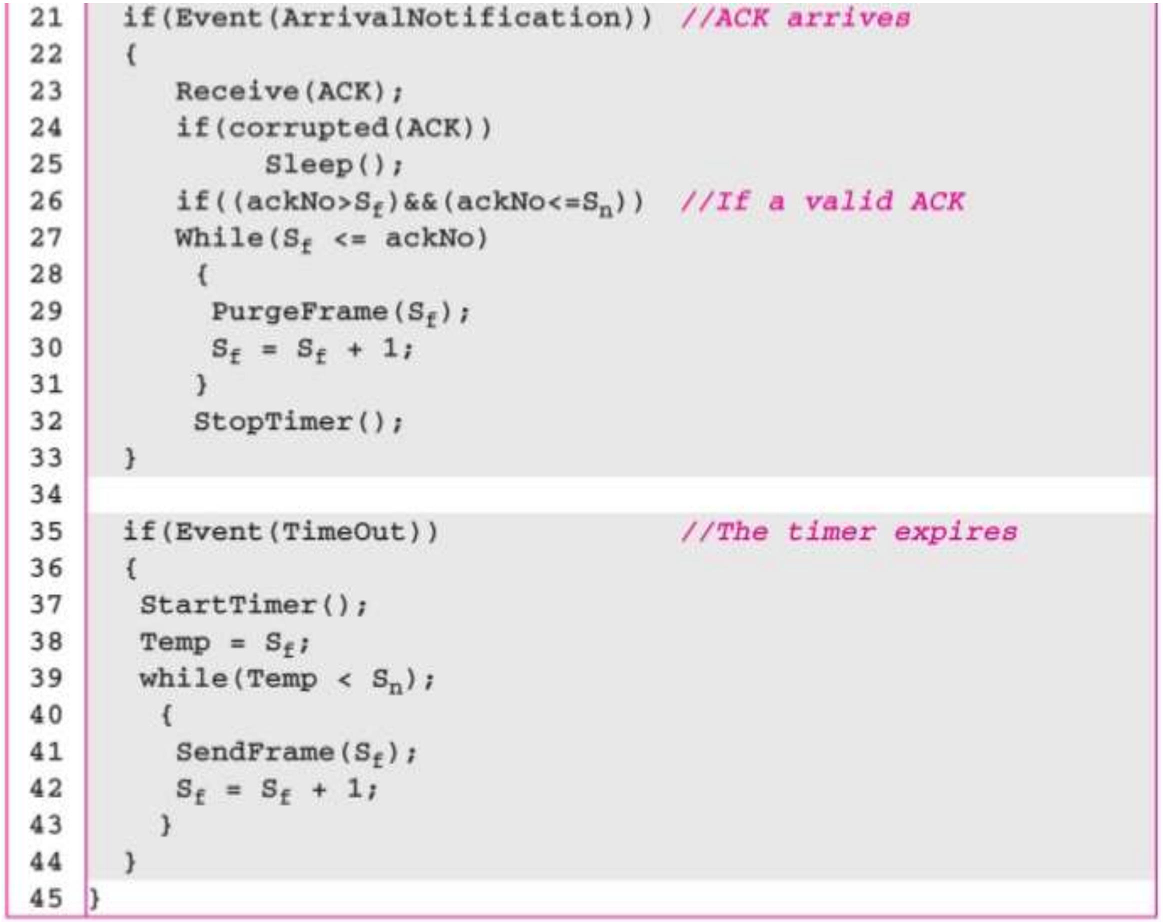
b. Window size = 2^m

UNIT – IV
NOISELESS CHANNELS

Go-Back-N sender algorithm



```
$S_w = 2^m - 1$;
$S_f = 0$;
$S_n = 0$;

while (true)                          //Repeat forever
{
 WaitForEvent();
  if(Event(RequestToSend))            //A packet to send
  {
     if($S_n$-$S_f$ >= $S_w$)             //If window is full
          Sleep();
     GetData();
     MakeFrame($S_n$);
     StoreFrame($S_n$);
     SendFrame($S_n$);
     $S_n = S_n + 1$;
     if(timer not running)
         StartTimer();
  }

```

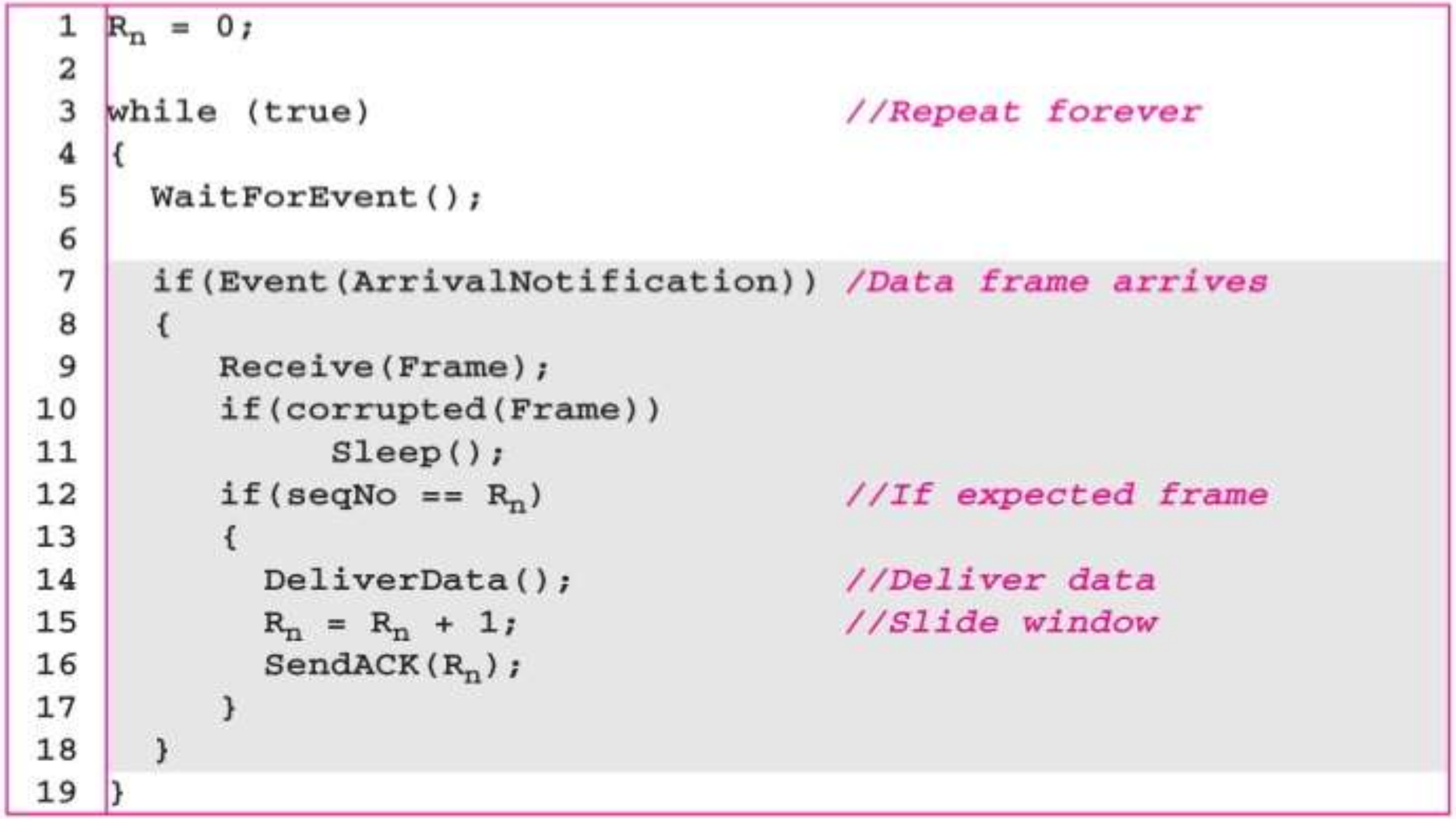


```
  if(Event(ArrivalNotification))  //ACK arrives
  {
     Receive(ACK);
     if(corrupted(ACK))
          Sleep();
     if((ackNo>$S_f$)&&(ackNo<=$S_n$))  //If a valid ACK
     While($S_f$ <= ackNo)
      {
       PurgeFrame($S_f$);
       $S_f = S_f + 1$;
      }
      StopTimer();
  }

  if(Event(TimeOut))               //The timer expires
  {
   StartTimer();
   Temp = $S_f$;
   while(Temp < $S_n$);
    {
     SendFrame($S_f$);
     $S_f = S_f + 1$;
    }
  }
}
```

UNIT – IV

NOISELESS CHANNELS

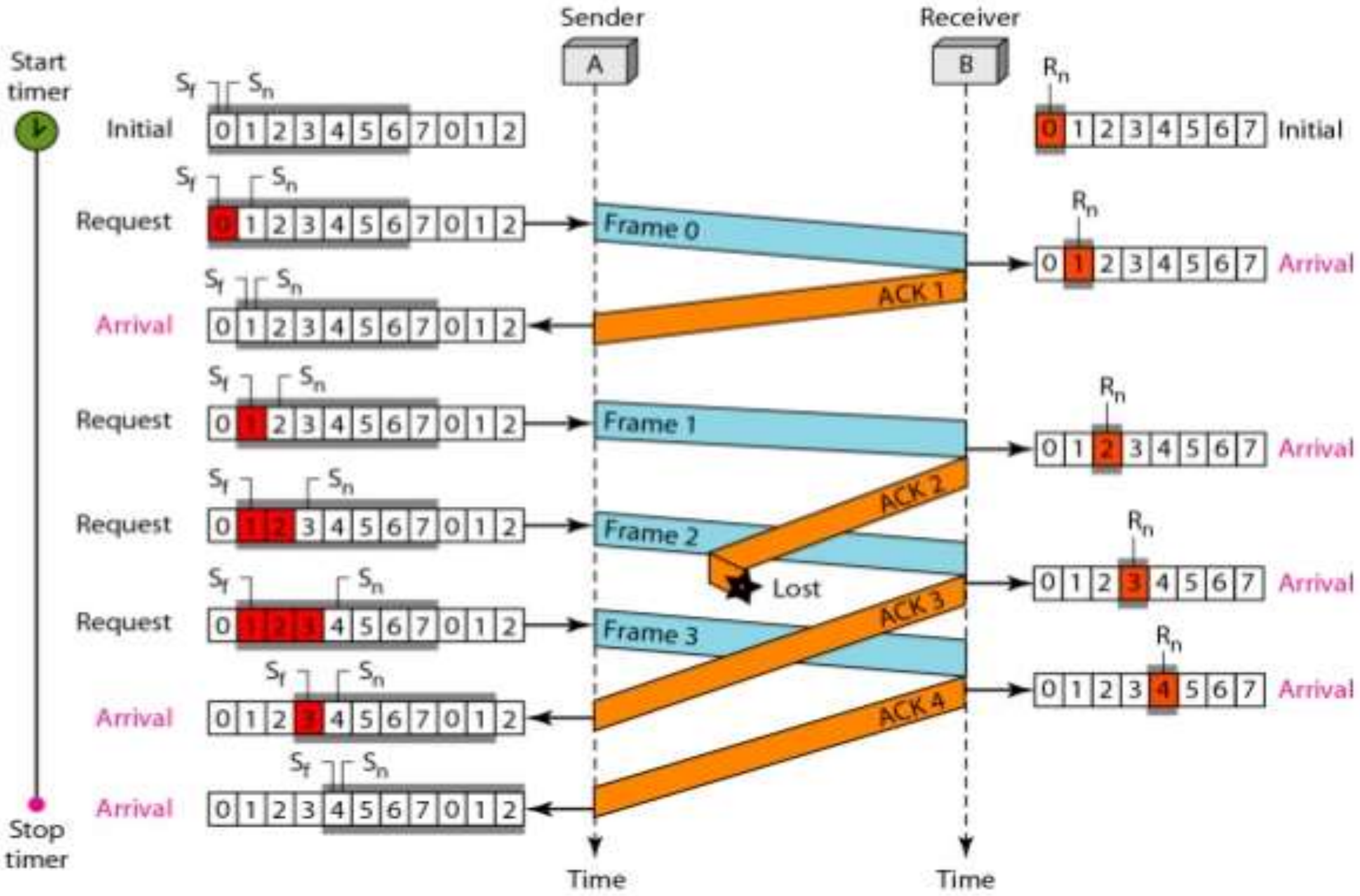
- **Go-Back-N receiver algorithm**



```
R_n = 0;

while (true)                                //Repeat forever
{
  WaitForEvent();

  if(Event(ArrivalNotification))            /Data frame arrives
  {
     Receive(Frame);
     if(corrupted(Frame))
          Sleep();
     if(seqNo == R_n)                       //If expected frame
     {
       DeliverData();                       //Deliver data
       R_n = R_n + 1;                       //Slide window
       SendACK(R_n);
     }
  }
}
```

UNIT – IV
NOISELESS CHANNELS

Example



DATA COMMUNICATIONS & NETWORKING

B.SC (COMPUTER SCIENCE) – III YEAR

SMCS52

By
Dr.N.N.Krishna Veni,
Assistant Professor,
Department of Computer Science,
Holy Cross Home Science College,
Thoothukudi

UNIT – IV
DATA LINK LAYER

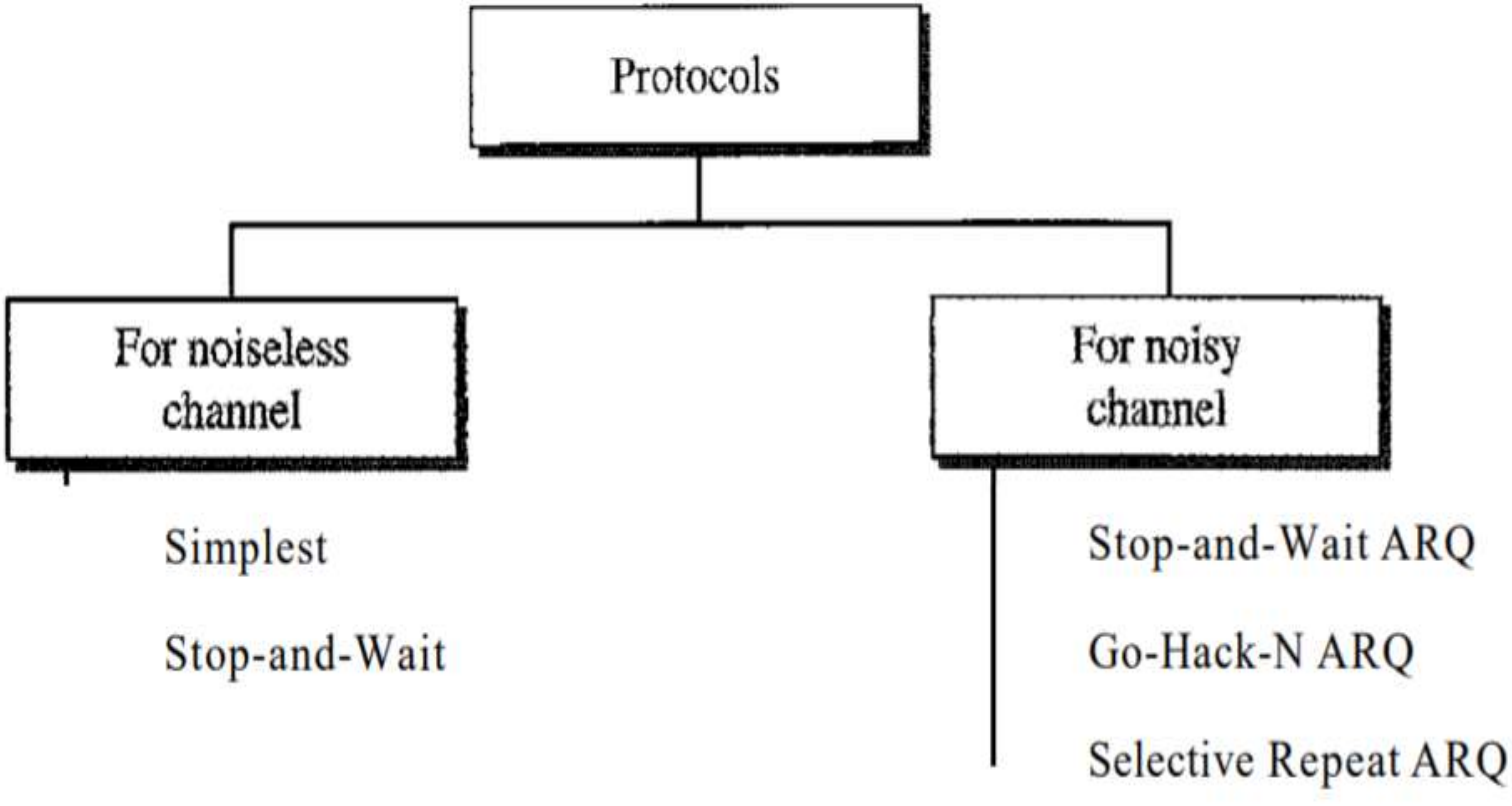
CHAPTER 8: Data Link Control

4.7 NOISY CHANNELS

4.7.3 Selective Repeat Automatic Repeat Request

4.7.4 Piggybacking

UNIT – IV
DATA LINK LAYER



UNIT – IV
NOISELESS CHANNELS

4.7 NOISY CHANNELS

4.7.3 Selective Repeat Automatic Repeat Request

- Go-Back-N ARQ simplifies the process at the receiver site.
- The receiver keeps track of only one variable, and there is no need to buffer out-of-order frames; they are simply discarded.
- However, this protocol is very inefficient for a noisy link.
- In a noisy link a frame has a higher probability of damage, which means the resending of multiple frames.
- This resending uses up the bandwidth and slows down the transmission.
- For noisy links, there is another mechanism that does not resend N frames when just one frame is damaged; only the damaged frame is resent.
- This mechanism is called Selective RepeatARQ.
- It is more efficient for noisy links, but the processing at the receiver is more complex.

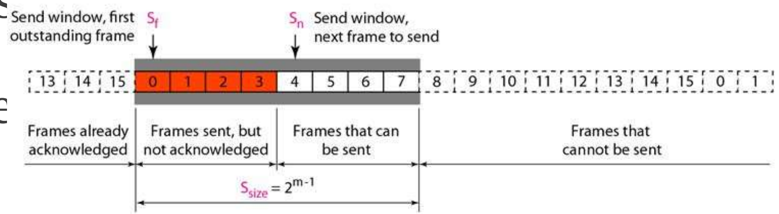
UNIT – IV
NOISELESS CHANNELS

Windows

- The Selective Repeat Protocol also uses two windows: a send window and a receive window. First, the size of the send window is much smaller; it is 2^{m-1}

- Second, the receive window is the

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

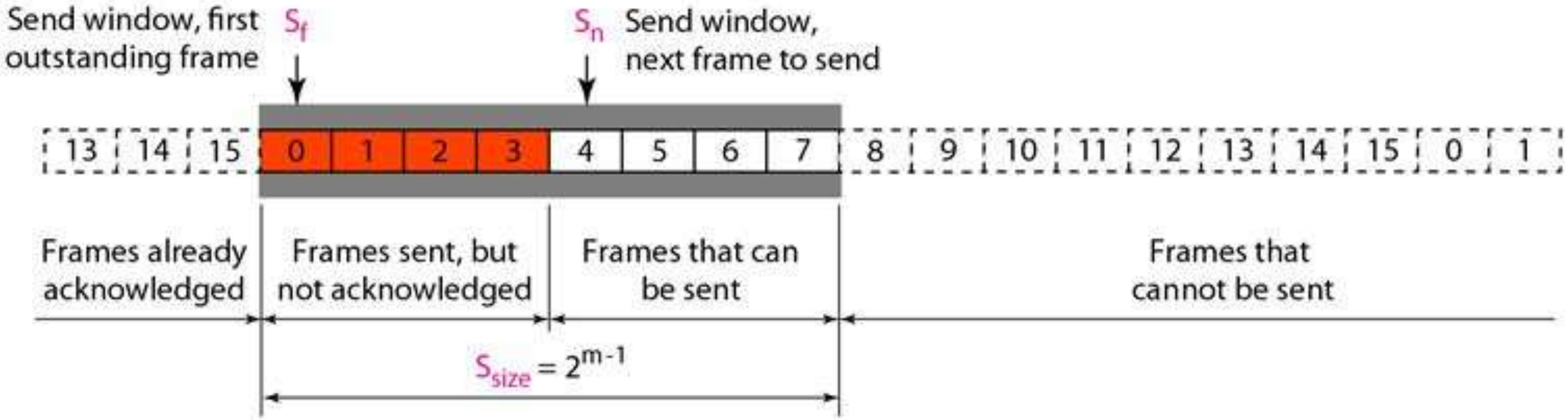


Send window, first outstanding frame
S_f
S_n
Send window, next frame to send
13 14 15 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 0 1
Frames already acknowledged
Frames sent, but not acknowledged
Frames that can be sent
Frames that cannot be sent
$S_{size} = 2^{m-1}$

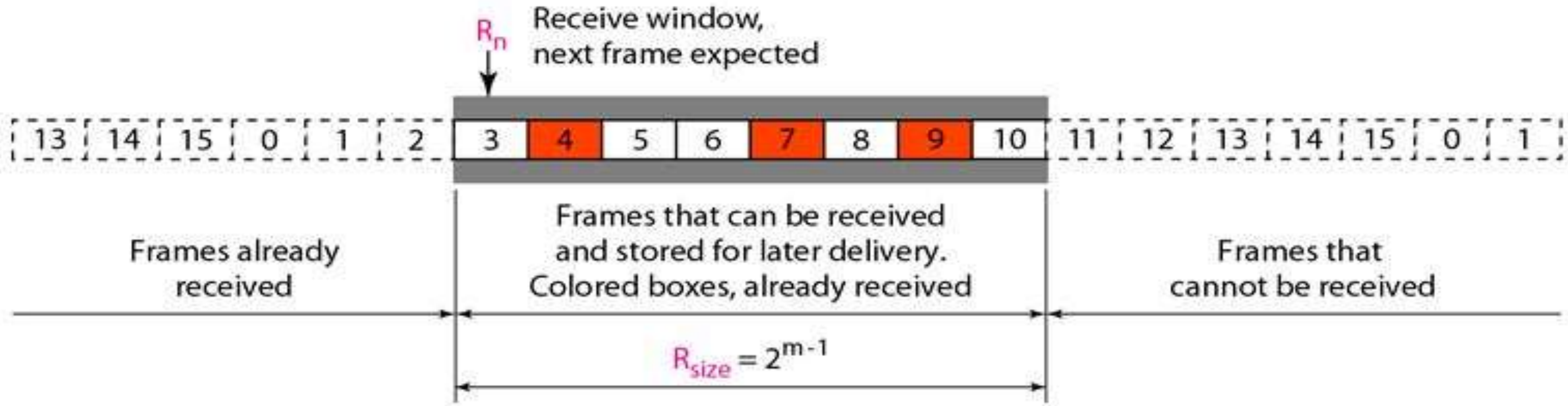
- The smaller window size means less efficiency in filling the pipe, but the fact that there are fewer duplicate frames can compensate for this.

UNIT – IV
NOISELESS CHANNELS

Sender side Windows


Send window, first outstanding frame
S_f
S_n
Send window, next frame to send
13 14 15 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 0 1
Frames already acknowledged
Frames sent, but not acknowledged
Frames that can be sent
Frames that cannot be sent
$S_{size} = 2^{m-1}$

Receiver Side Windows

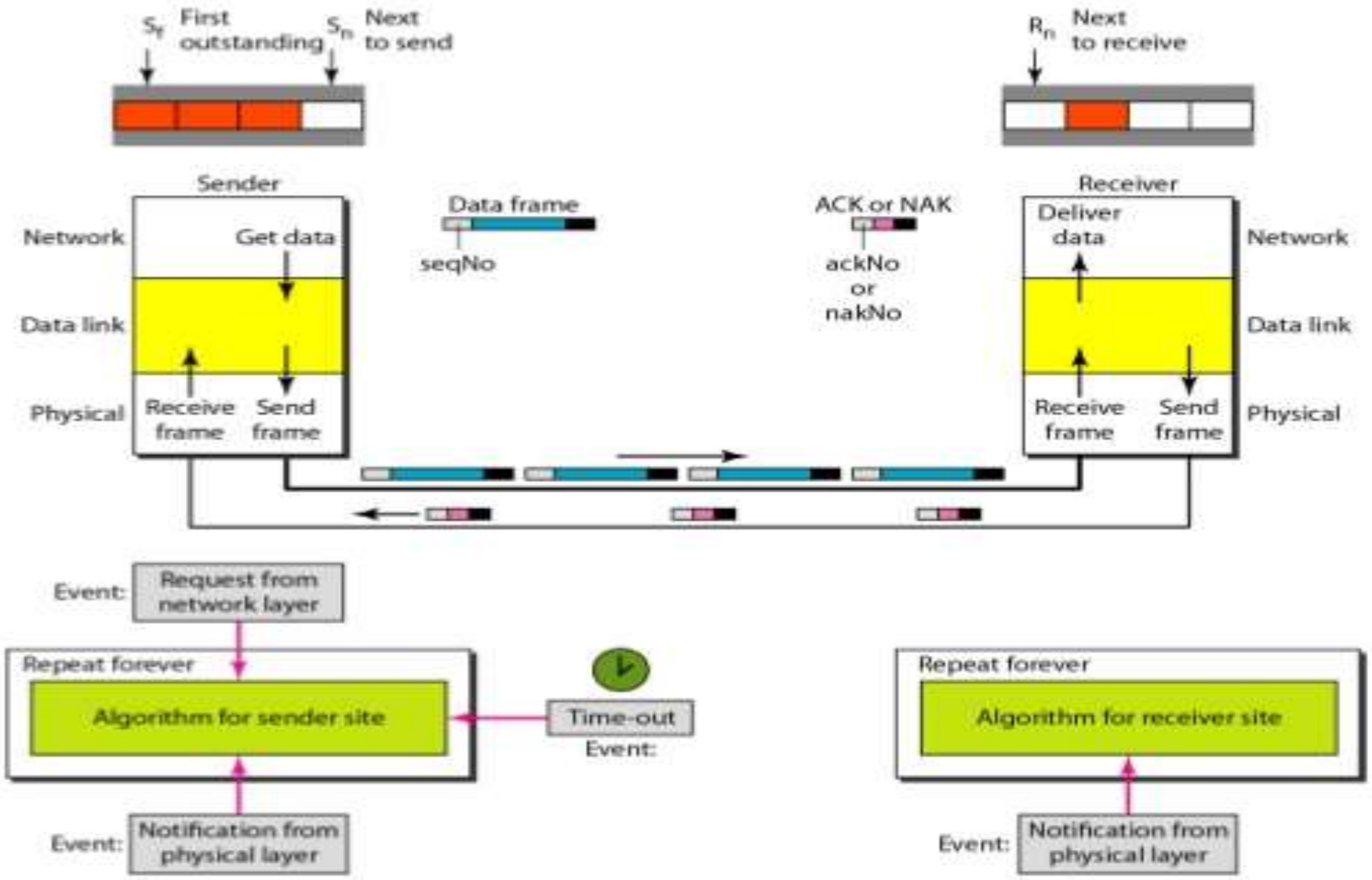

R_n
Receive window, next frame expected
13 14 15 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 0 1
Frames already received
Frames that can be received and stored for later delivery. Colored boxes, already received
Frames that cannot be received
$R_{size} = 2^{m-1}$

UNIT – IV
NOISELESS CHANNELS

- Design
- Sender Site Algorithm
- Receiver Site Algorithm
- Example

UNIT – IV
NOISELESS CHANNELS

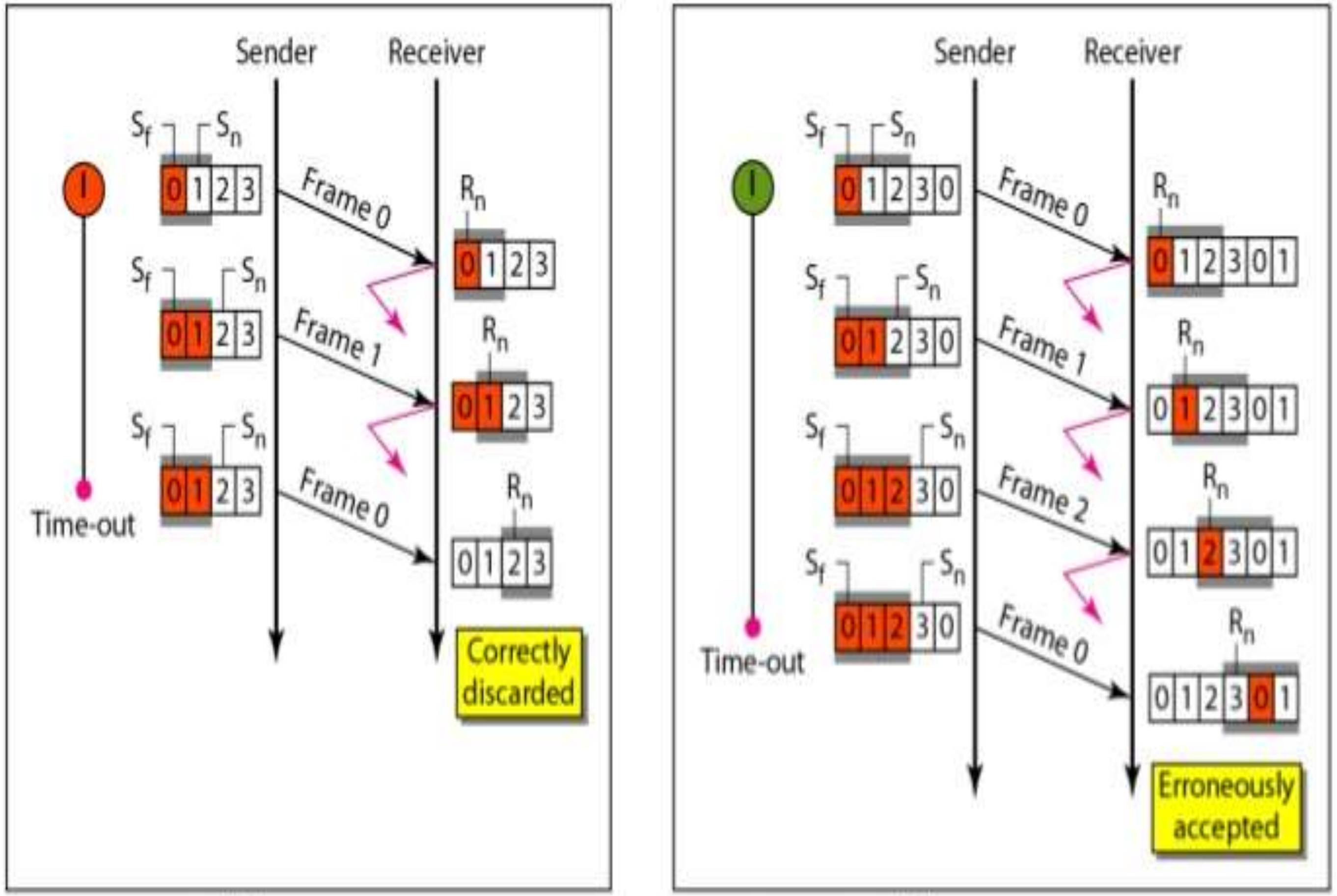
Design



UNIT – IV
NOISELESS CHANNELS

Window size for Selective Repeat Automatic Repeat Request

We can now show why the size of the sender and receiver windows must be at most one half of 2^m

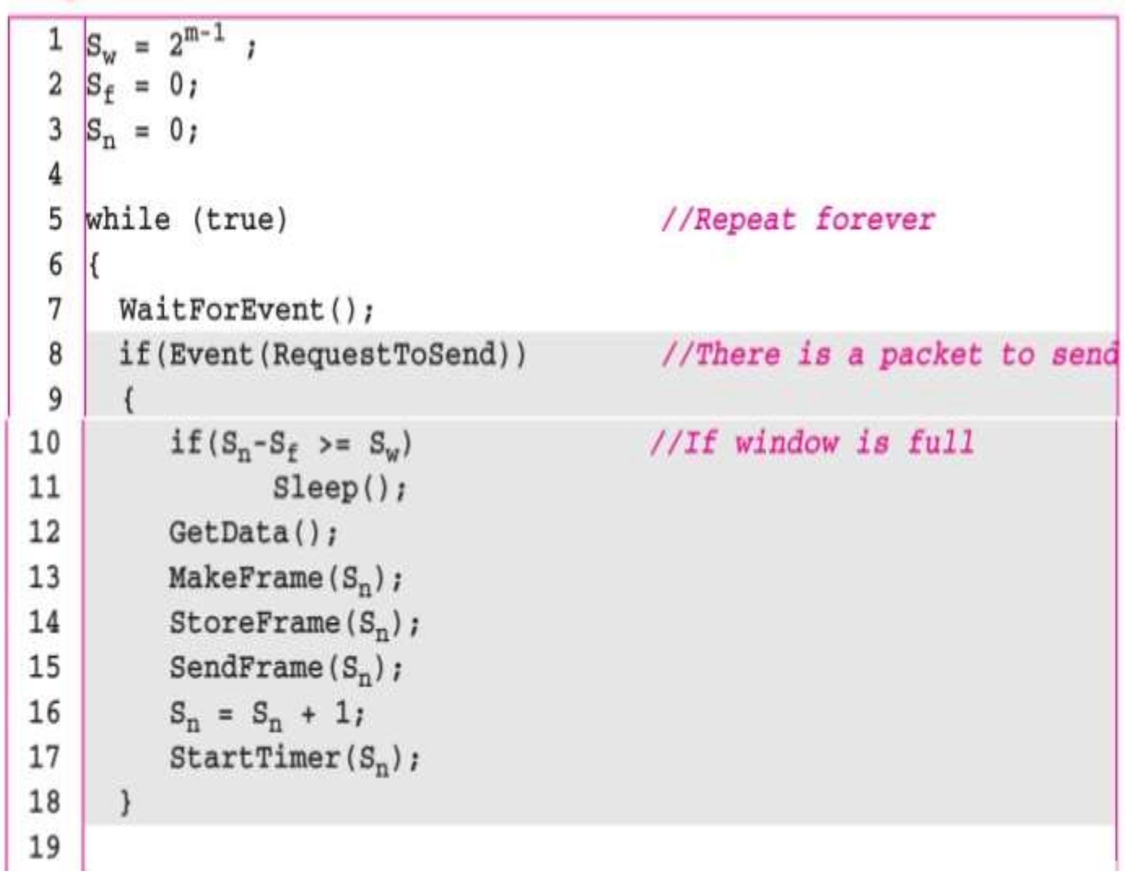


a. Window size = 2^{m-1}

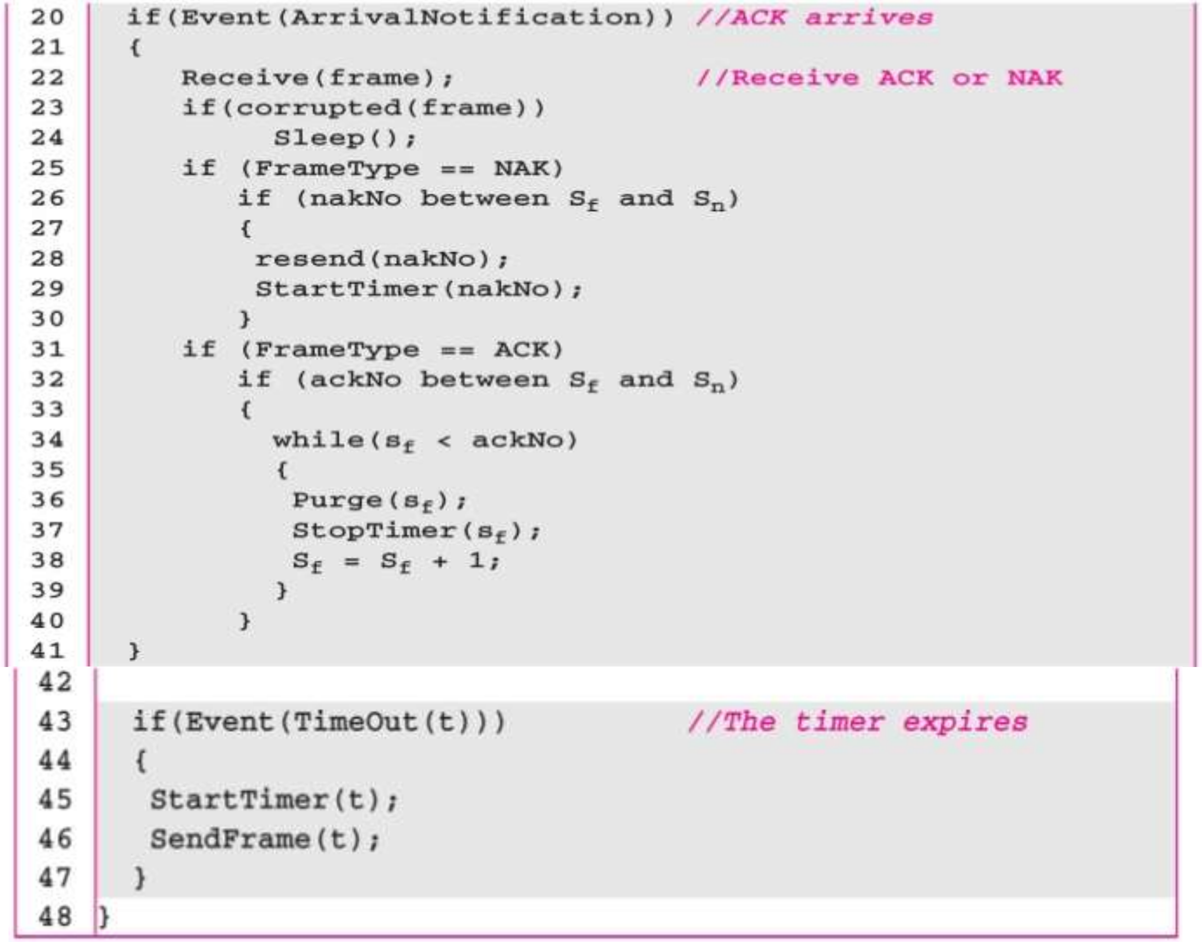
b. Window size > 2^{m-1}

UNIT – IV
NOISELESS CHANNELS

Sender algorithm



```
S_w = 2^(m-1) ;
S_f = 0;
S_n = 0;

while (true)                              //Repeat forever
{
  WaitForEvent();
  if(Event(RequestToSend))                //There is a packet to send
  {
     if(S_n-S_f >= S_w)                   //If window is full
            Sleep();
     GetData();
     MakeFrame(S_n);
     StoreFrame(S_n);
     SendFrame(S_n);
     S_n = S_n + 1;
     StartTimer(S_n);
  }

```

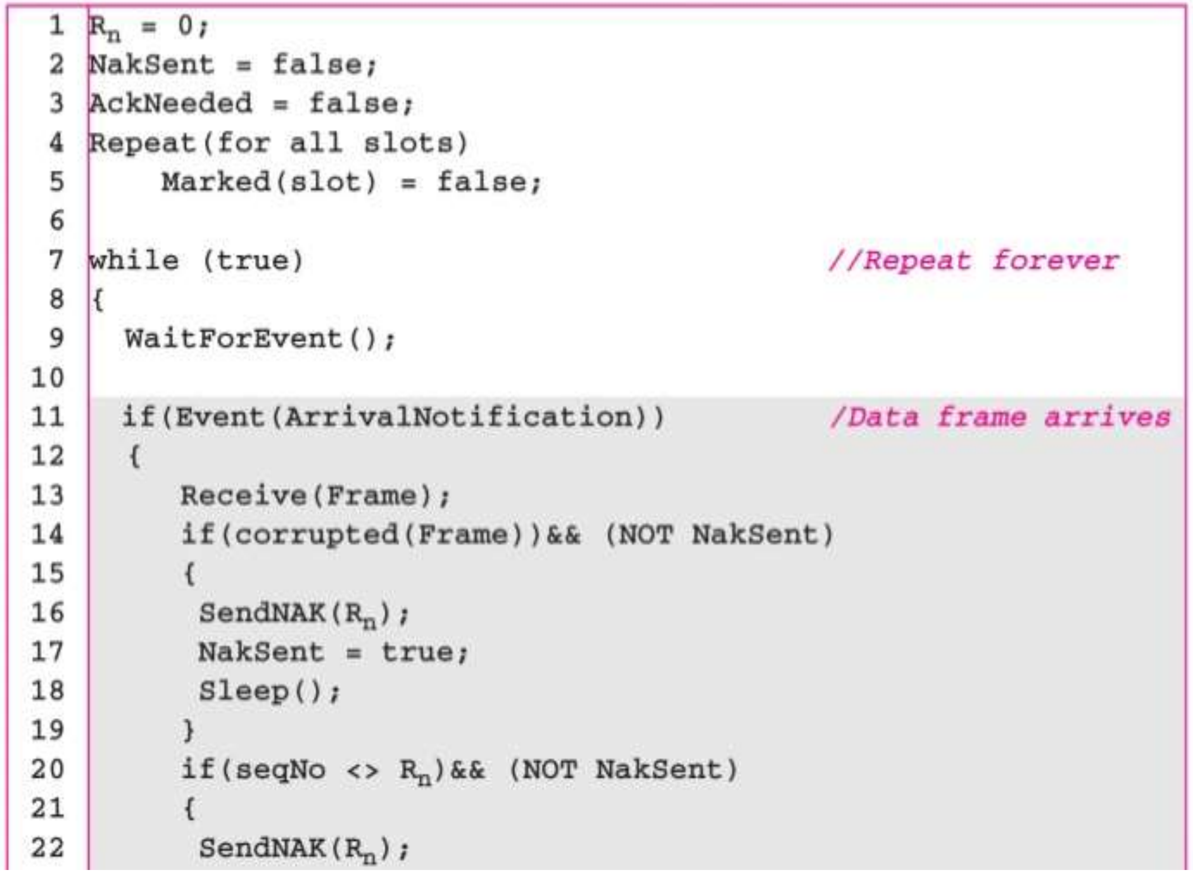


```
  if(Event(ArrivalNotification)) //ACK arrives
  {
     Receive(frame);             //Receive ACK or NAK
     if(corrupted(frame))
          Sleep();
     if (FrameType == NAK)
        if (nakNo between S_f and S_n)
        {
         resend(nakNo);
         StartTimer(nakNo);
        }
     if (FrameType == ACK)
        if (ackNo between S_f and S_n)
        {
          while(s_f < ackNo)
          {
           Purge(s_f);
           StopTimer(s_f);
           S_f = S_f + 1;
          }
        }
  }

  if(Event(TimeOut(t)))          //The timer expires
  {
   StartTimer(t);
   SendFrame(t);
  }
}
```

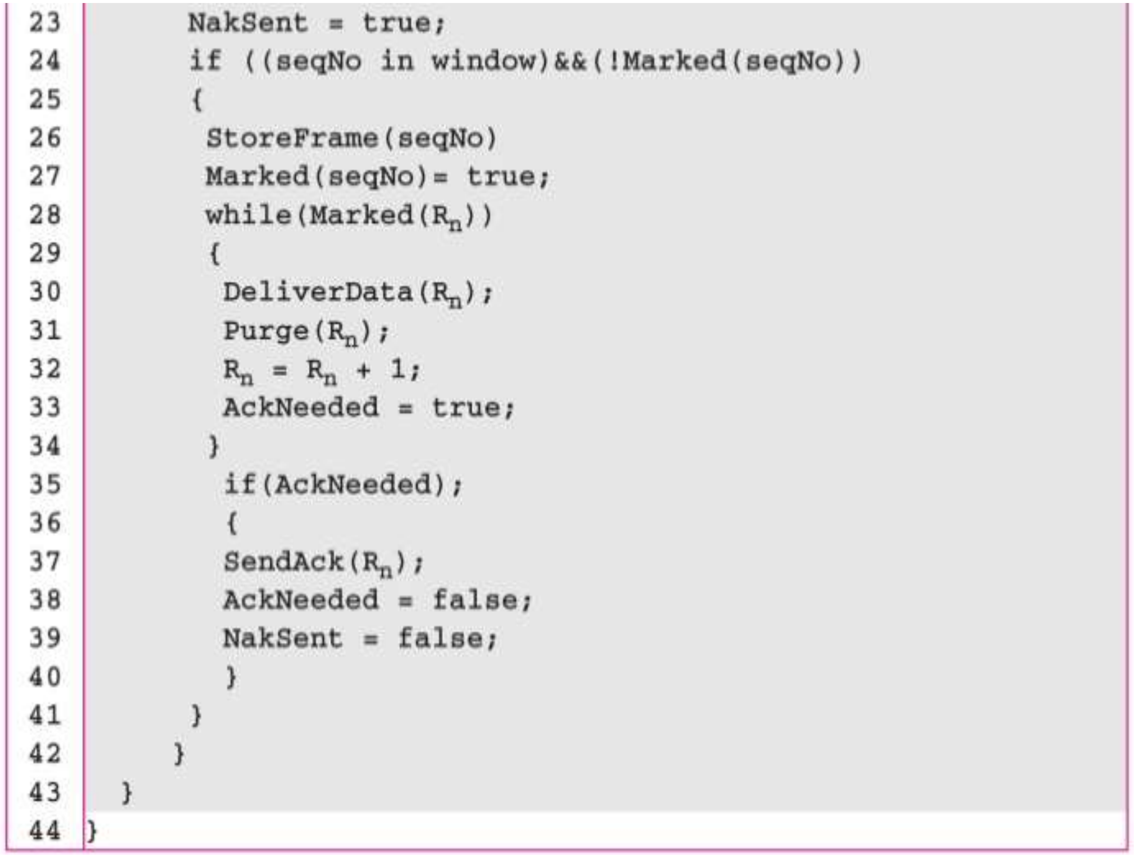
UNIT – IV

NOISELESS CHANNELS

Receiver algorithm



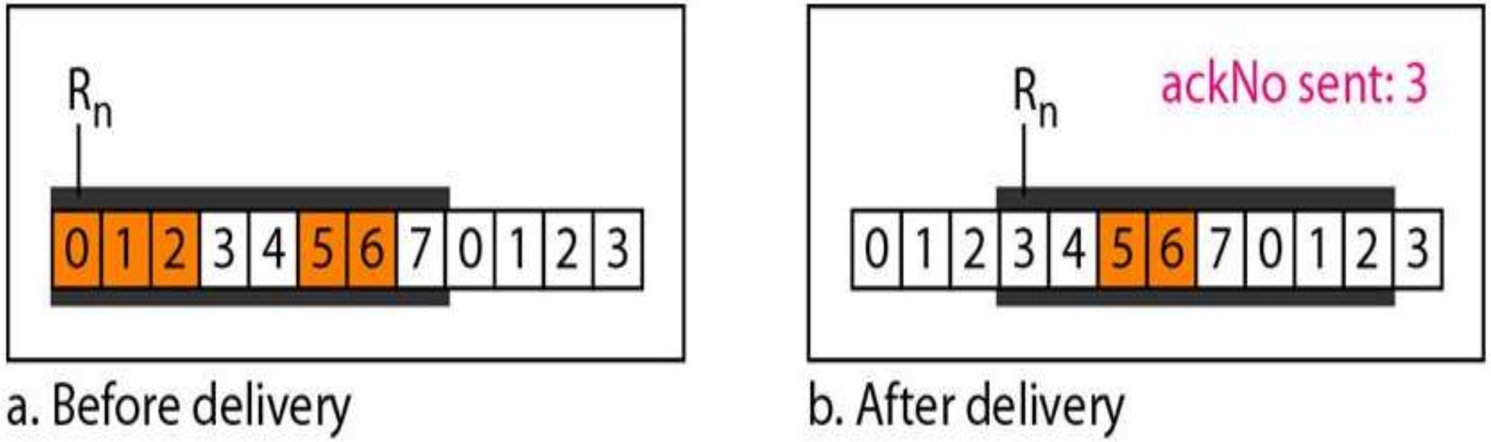
```
R_n = 0;
NakSent = false;
AckNeeded = false;
Repeat(for all slots)
    Marked(slot) = false;

while (true)                                    //Repeat forever
{
  WaitForEvent();

  if(Event(ArrivalNotification))             /Data frame arrives
  {
     Receive(Frame);
     if(corrupted(Frame))&& (NOT NakSent)
     {
      SendNAK(R_n);
      NakSent = true;
      Sleep();
     }
     if(seqNo <> R_n)&& (NOT NakSent)
     {
      SendNAK(R_n);
```



```
      NakSent = true;
      if ((seqNo in window)&&(!Marked(seqNo))
      {
       StoreFrame(seqNo)
       Marked(seqNo)= true;
       while(Marked(R_n))
       {
        DeliverData(R_n);
        Purge(R_n);
        R_n = R_n + 1;
        AckNeeded = true;
       }
        if(AckNeeded);
        {
        SendAck(R_n);
        AckNeeded = false;
        NakSent = false;
        }
      }
     }
  }
}
```

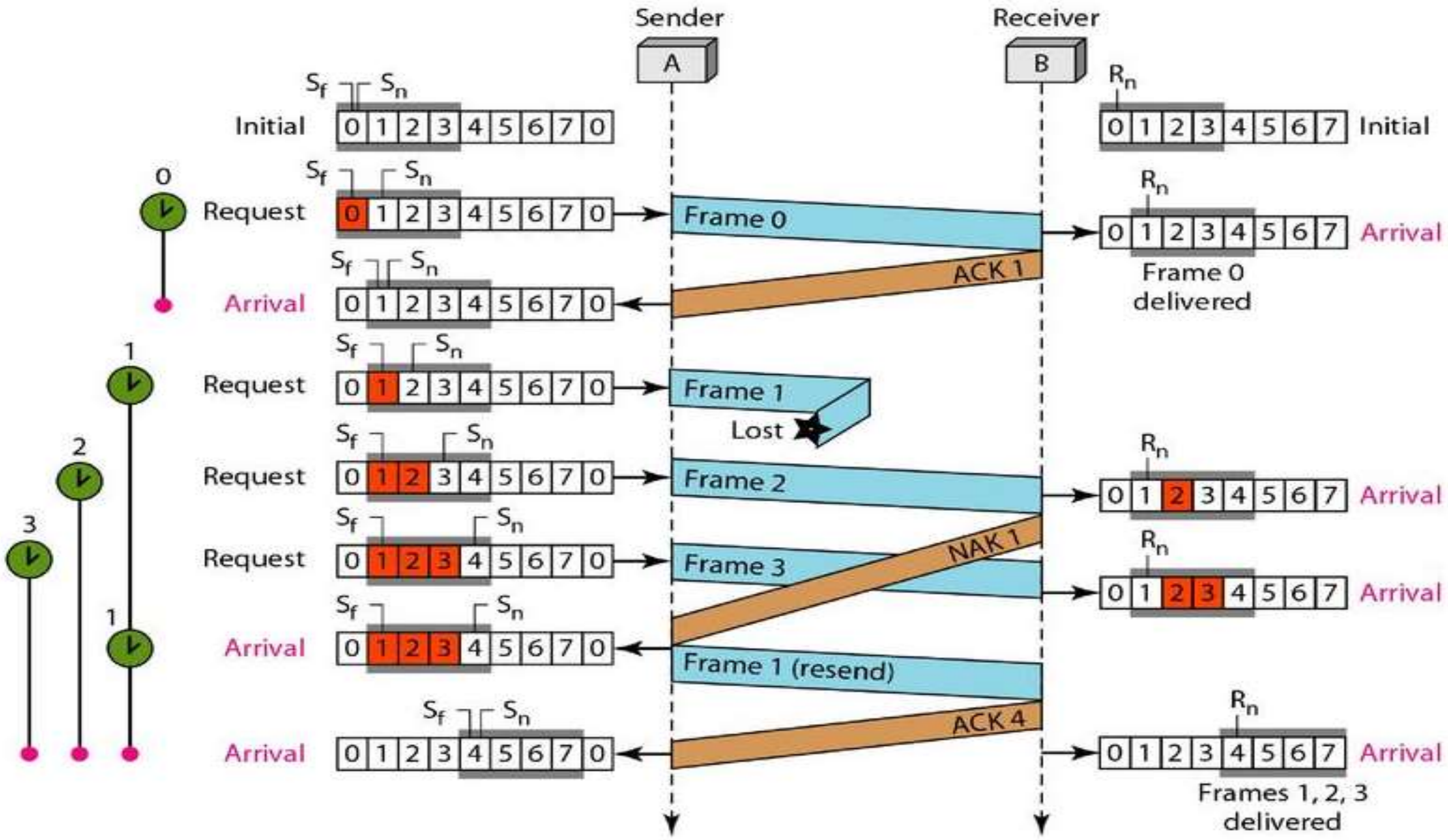
UNIT – IV
NOISELESS CHANNELS

Delivery of data



UNIT – IV
NOISELESS CHANNELS

Example



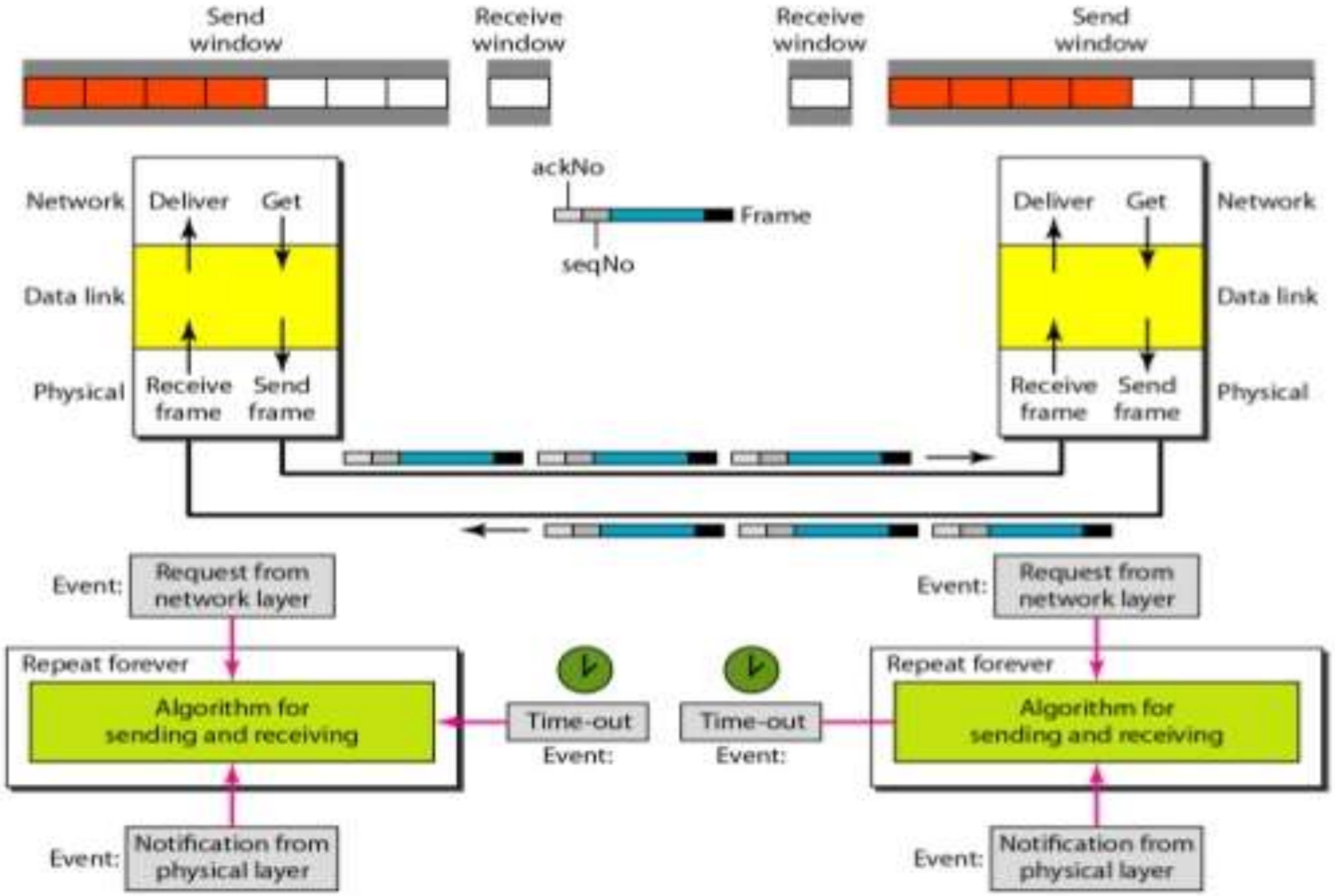
UNIT – IV
NOISELESS CHANNELS

4.7.4 Piggybacking

- The three protocols we discussed in this section are all unidirectional: data frames flow in only one direction although control information such as ACK and NAK frames can travel in the other direction.
- In real life, data frames are normally flowing in both directions: from node A to node B and from node B to node A. This means that the control information also needs to flow in both directions.
- A technique called piggybacking is used to improve the efficiency of the bidirectional protocols.
- When a frame is carrying data from A to B, it can also carry control information about arrived (or lost) frames from B; when a frame is carrying data from B to A, it can also carry control information about the arrived (or lost) frames from A.

UNIT – IV
NOISELESS CHANNELS

Design of piggybacking in Go-Back-N ARQ



DATA COMMUNICATIONS & NETWORKING

B.SC (COMPUTER SCIENCE) – III YEAR

SMCS52

By
Dr.N.N.Krishna Veni,
Assistant Professor,
Department of Computer Science,
Holy Cross Home Science College,
Thoothukudi

UNIT – IV
DATA LINK LAYER

CHAPTER 9: Wired LANs: Ethernet

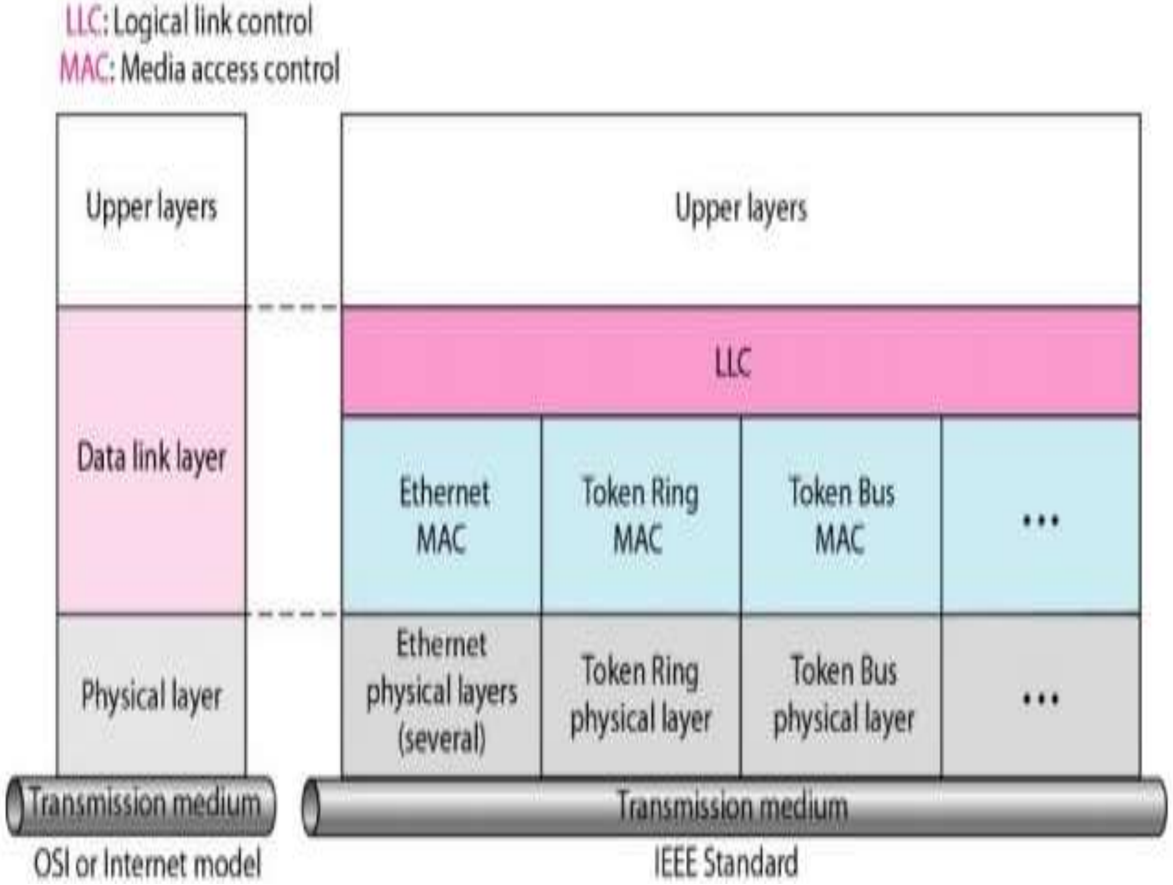
4.8 IEEE STANDARDS

UNIT – IV
DATA LINK LAYER

CHAPTER 9: Wired LANs: Ethernet

4.8 IEEE STANDARDS

- In 1985, the Computer Society of the IEEE started a project, called Project 802, to set standards to enable intercommunication among equipment from a variety of manufacturers.
- The relationship of the 802 Standard to the traditional OSI model has subdivided the data link layer into two sublayers: logical link control (LLC) and media access control (MAC).
- IEEE has also created several physical layer standards for different LAN protocols.



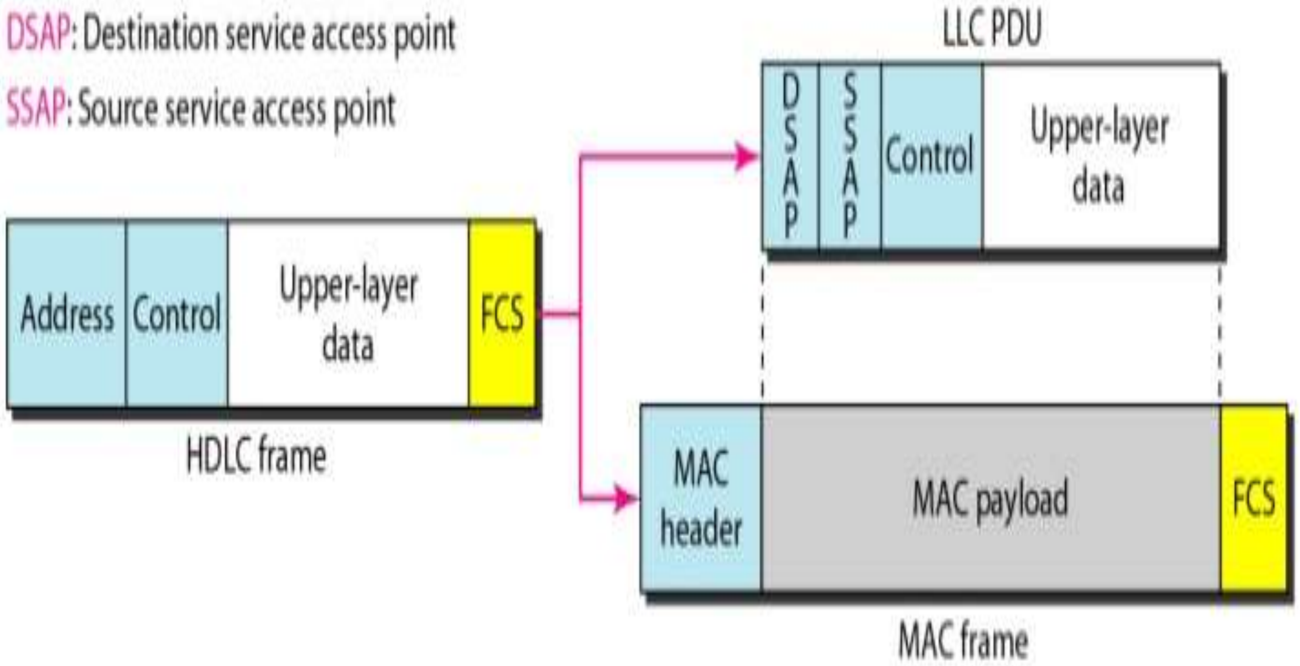
UNIT – IV
NOISELESS CHANNELS

4.8.1 Data Link Layer

The data link layer in the IEEE standard is divided into two sublayers: LLC and MAC.

In IEEE Project 802, flow control, error control, and part of the framing duties are collected into one sublayer called the logical link control.

Framing is handled in both the LLC sublayer and the MAC sublayer. The LLC provides one single data link control protocol for all IEEE LANs.


DSAP: Destination service access point
SSAP: Source service access point
LLC PDU
DSAP
SSAP
Control
Upper-layer data
Address
Control
Upper-layer data
FCS
HDLC frame
MAC header
MAC payload
FCS
MAC frame