

UNIT-1

B SAI BABA,M.Tech(Ph.D),VIT,Bhimavaram

What is **Computer Network?**

Resource Sharing

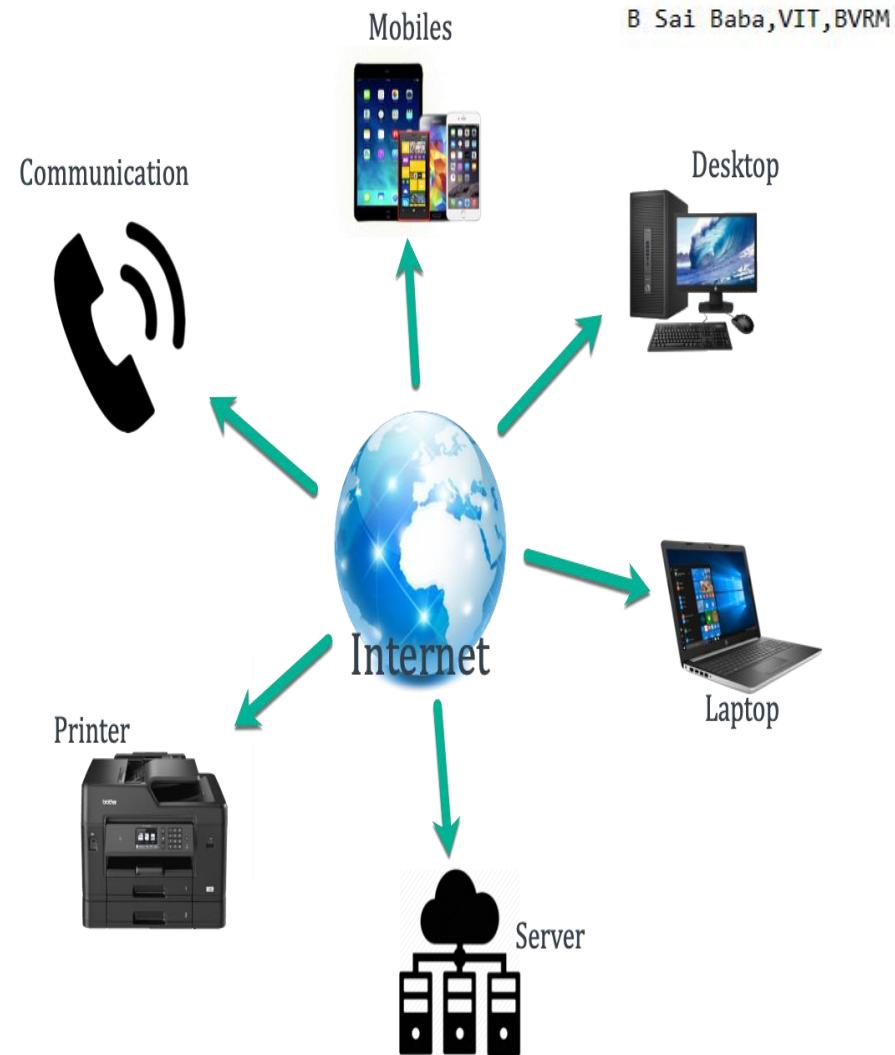


- A computer network is a collection of interconnected **computers, devices, and other hardware components** that are linked together to facilitate **communication, data sharing, and resource sharing**.
- It allows multiple computers and devices to exchange information, access shared resources, and collaborate effectively.

(or)

- A computer network is a number of computers linked together to allow them to “talk” to each other and share resources. Networked computers can **share hardware, software and data**.
- Connecting computers to form computer networks and the internet has had a huge impact on our lives.

What is **Internet?**



- Internet is **a global network** that connects billions of computers across the world with each other and to the World Wide Web.
- It uses standard **internet protocol suite** (TCP/IP) to connect billions of computer users worldwide.
- At present, internet is the fastest mean of sending or exchanging information and data between computers across the world.

(or)

- “**Network of networks**”

Where does the Internet come from?



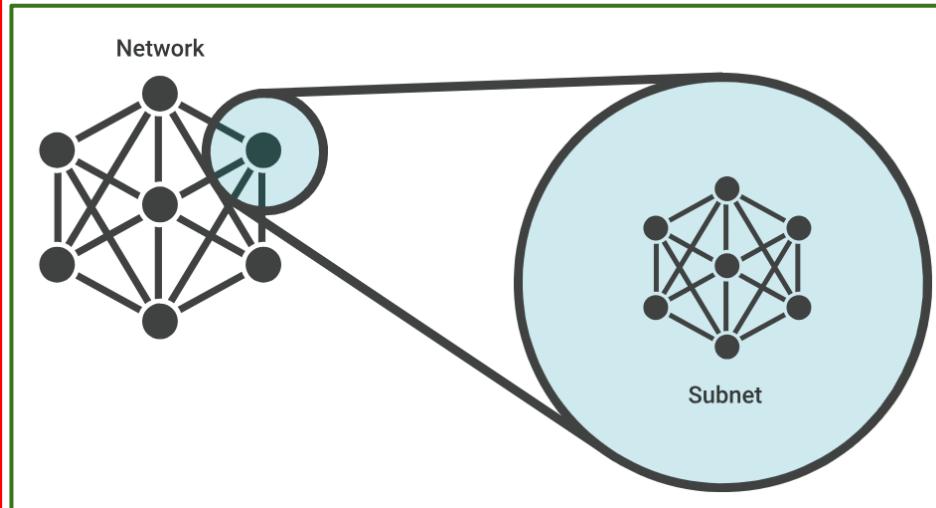
What is **WWW?**

- The World Wide Web, commonly known as **the web**, is a system of interconnected **documents and resources** that are accessed via the internet.
- World Wide Web is an information system that allows users to access and navigate websites and web pages on the internet
- It is **a collection of websites, web pages, multimedia content, and hyperlinks** that allow users to navigate and access information.
- It is a common prefix used in **Uniform Resource Locators or URLs**(internet addresses) to identify a web server that hosts a website. **Eg: www.google.com**
- The web is just one of the many services and applications that utilize the internet infrastructure

Network Elements

Subnet ?

- A subnet, short for subnetwork, is a portion of a larger network that is divided or segmented into smaller logical networks.
- It involves dividing a single network into multiple smaller networks to improve network performance, security, and manageability.





Network Interface Card (NIC)



Internal Network Cards



USB based NIC



Wireless NIC

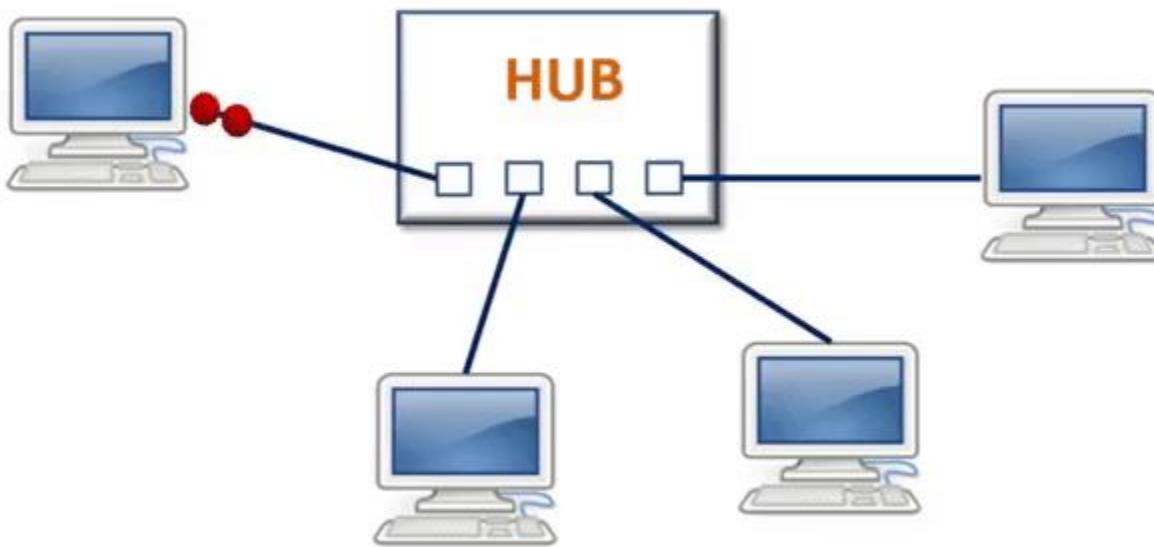
External Network Cards

NIC?

- A Network Interface Card (NIC) is a hardware component **without which a computer cannot be connected over a network.**
- It is a circuit board installed in a computer that provides a dedicated network connection to the computer.
- It is also called **network interface controller, network adapter or LAN adapter.**
- It can support a transfer rate of 10,100 to 1000 Mb/s.

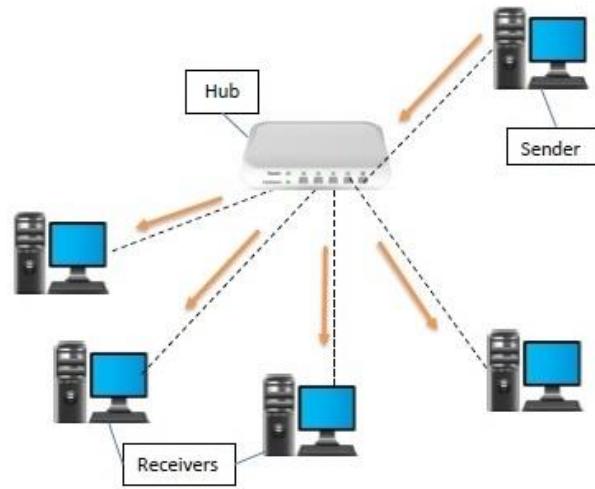


Hub?

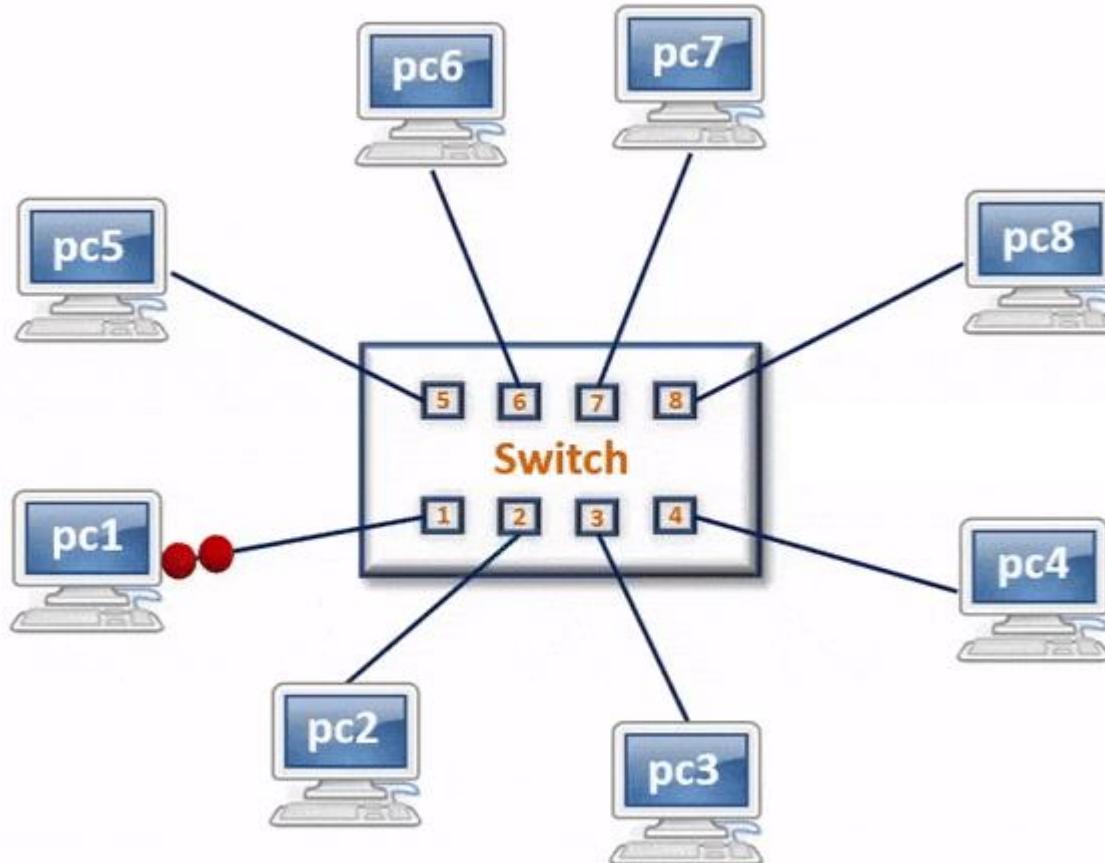


Hub?

- A hub is a physical layer networking device which is used to **connect multiple devices** in a network. They are generally used to connect computers in a LAN.
- A hub has many ports in it. A computer which intends to be connected to the network is plugged in to one of these ports.
- When a data frame arrives at a port, **it is broadcast to every other port**, without considering whether it is destined for a particular destination or not.

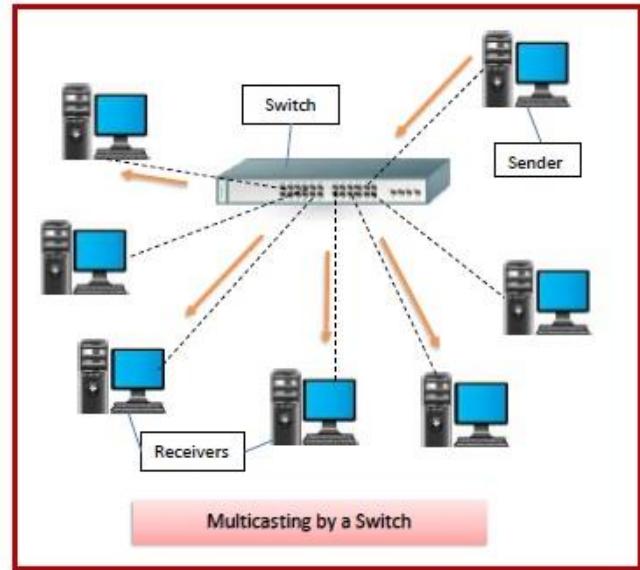


Switch?

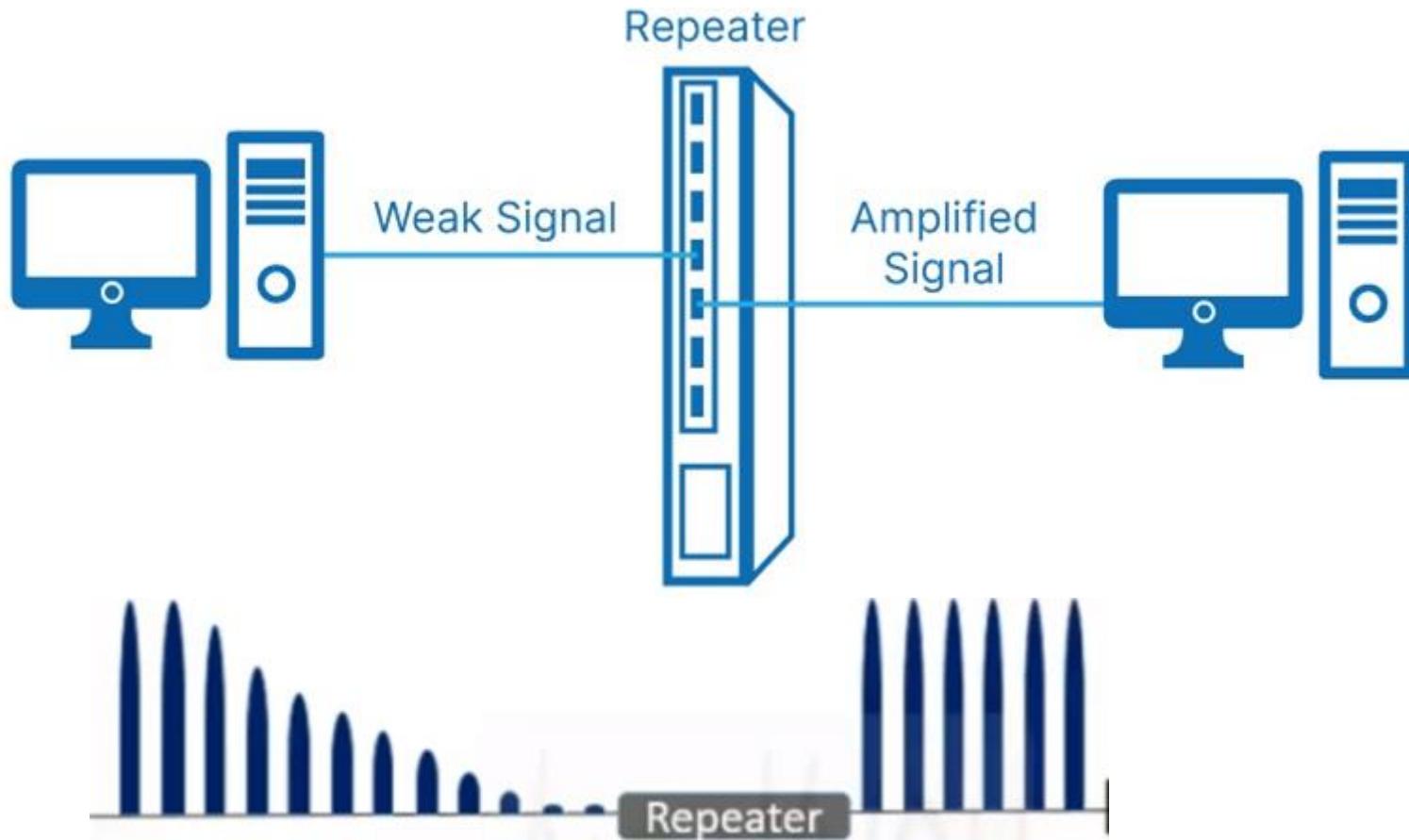


Switch?

- A switch is **a data link layer hardware device** that connects multiple devices on a computer network.
- A Switch contains **more advanced features than Hub.**
- Switch delivers the message to the correct destination based on the **physical address** present in the incoming message.
- **A Switch does not broadcast the message to the entire network like the Hub. It determines the device to whom the message is to be transmitted.**



Repeater ?



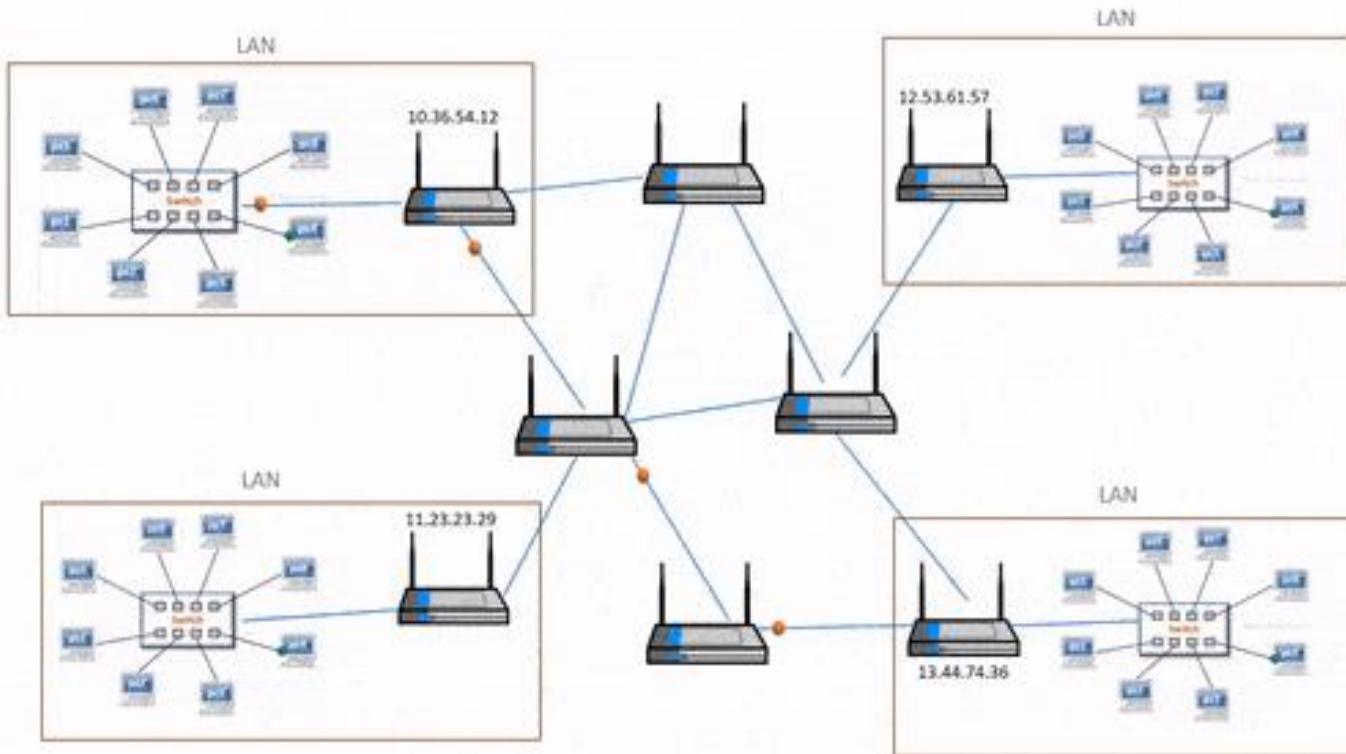
- Repeaters are used at the Physical layer of OSI model.
- A repeater is a powerful network hardware device that **regenerates an incoming signal from the sender before retransmitting it to the receiver.**
- It is also known as **a signal booster**, The primary function of a repeater is **to receive the weakened signal, amplify it, and then retransmit it at its original strength.**
- In a network, as data travels over cables (such as copper or fiber-optic cables), it tends to **weaken** or attenuate over long distances.
- This attenuation can lead to **data loss and degradation**, affecting the overall performance of the network.

Router?

- A router is a network device that operates at **the network layer** (Layer 3) of the OSI model.
- Its primary function is **to forward data packets between different computer networks**, such as local area networks (LANs) and wide area networks (WANs).
- A router acts as **a central point of connection** for multiple devices and directs network traffic based on destination IP addresses.



Router Device



Network Topologies

“ A topology is the **layout of how a network communicate with different devices.**”

(or)

“A Network Topology **is the arrangement with which computer systems or network devices are connected to each other.** Topologies may define both physical and logical aspect of the network.”

The various network topologies are:

- Point to Point Topology
- Mesh Topology
- Star Topology
- Bus Topology
- Ring Topology
- Tree Topology
- Hybrid Topology

Point to Point Topology

- Point-to-point topology is a network configuration where two devices or nodes are directly connected to each other without any intermediate devices.
- It is the simplest communication between two nodes, in which one is the sender and the other one is the receiver.
- In this type of topology, data can flow directly from one point to another along a **dedicated communication channel**.



Point to Point Topology

Mesh Topology

- In a mesh topology, every device is connected to another device via a particular channel. In Mesh Topology, the protocols used are **AHCP** (Ad Hoc Configuration Protocols), **DHCP** (Dynamic Host Configuration Protocol), etc.
- Every device is connected to another via dedicated channels. These channels are known as **links**.
- Suppose, the N number of devices are connected with each other in a mesh topology, the total number of ports that are required by each device is N-1.
- In Figure , there are 5 devices connected to each other, hence the total number of ports required by each device is 4. The total number of ports required = $N * (N-1)$.



Mesh Topology

Advantages of Mesh Topology

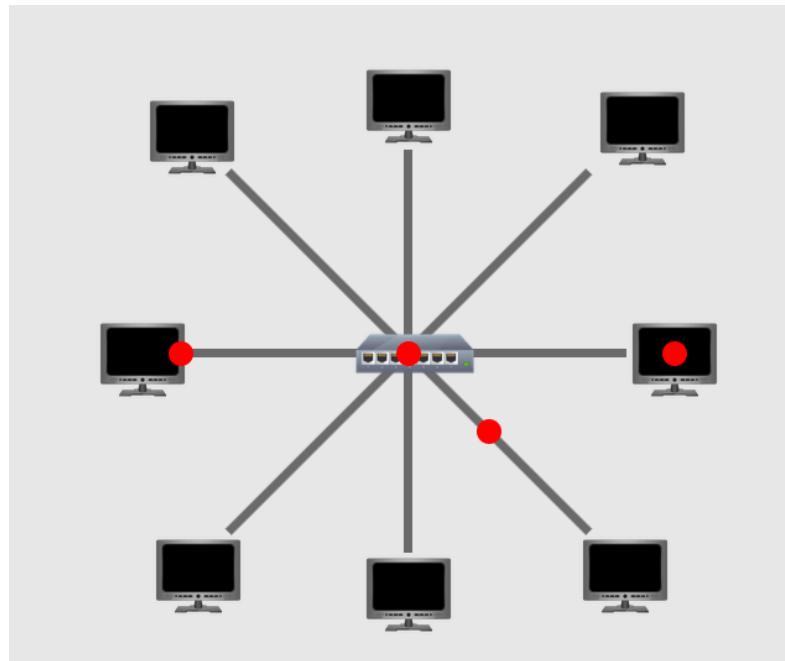
- Communication is very fast between the nodes.
- Mesh Topology is robust.
- **Fault Tolerance:** *Mesh topology provides high redundancy and fault tolerance. If one link or connection fails, there are alternative paths available for data to reach its destination, ensuring network reliability.*

Drawbacks of Mesh Topology

- **Installation and configuration are difficult.**
- The cost of cables is high as bulk wiring is required, hence suitable for less number of devices.
- The cost of maintenance is high

Star Topology

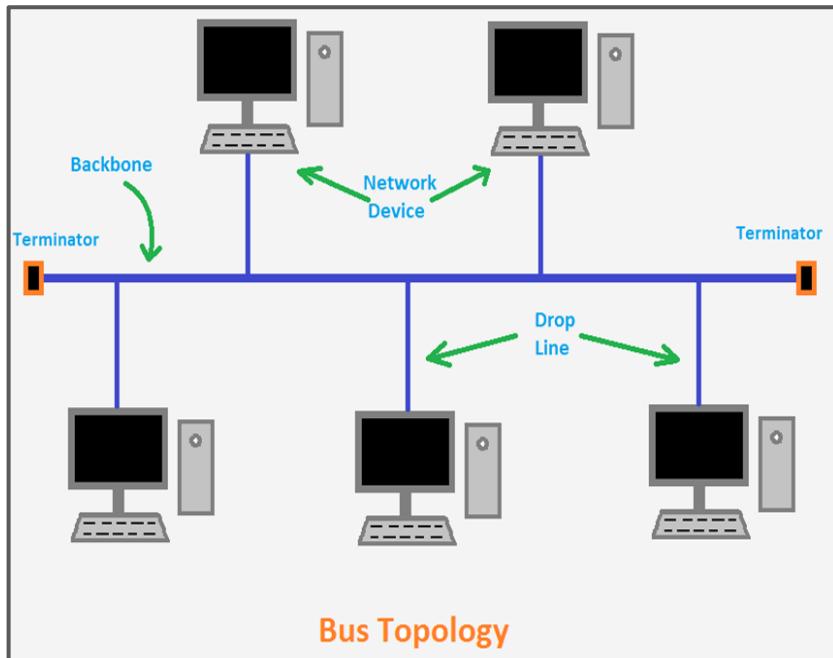
- Star topology is a network configuration in which all devices in the network are connected to **a central hub or switch**.
- In this arrangement, each device communicates directly with **the central hub and not with other devices** in the network. The hub acts as a central point of communication, managing data traffic between the connected device
- Star topology is very popular because the startup costs are low. It is also easy to create new nodes to the network.
- **If one device fails**, it does not affect the rest of the network, as they are not directly connected to each other.
- **If the central hub fails** throughout the network goes down.



Star Topology

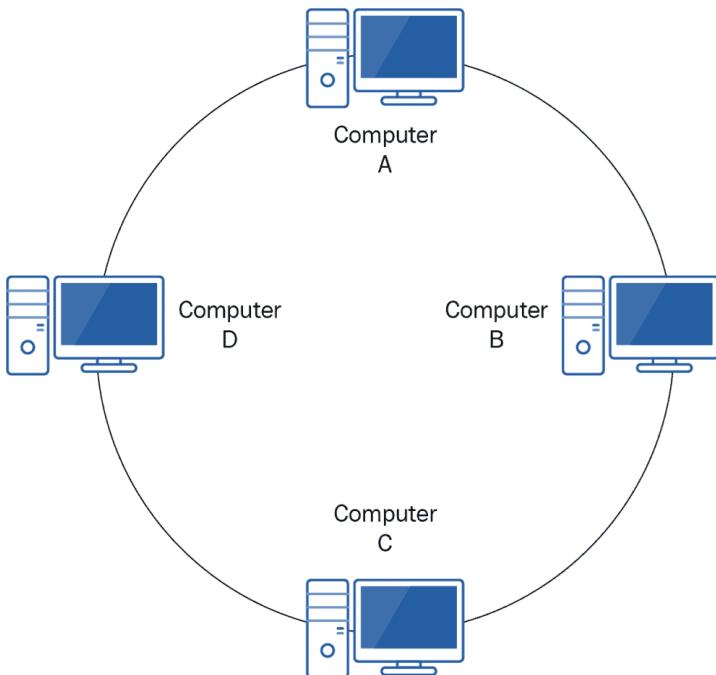
Bus Topology

- The bus topology is designed in such a way that all the stations are connected through a **single cable** known as **a backbone cable**.
- When a node wants to send a message over the network, it puts a message over the network. All the stations available in the network will receive the message whether it has been addressed or not.
- Data is transmitted in **both directions along the bus**.
- The **backbone cable** is considered as a "**single lane**" through which the message is broadcast to all the stations.
- It is a multi-point connection and a non-robust topology because **if the backbone fails the topology crashes**.



- In a ring topology, devices are connected in a **closed loop**, with each device having exactly **two neighbors for communication**.
- The node that receives the message from the previous computer will retransmit to the next node.
- **The data flows in one direction, i.e., it is unidirectional.**
- The data flows in a **single loop** continuously known as an endless loop.
- It has no terminated ends, i.e., each node is connected to other node and having no termination point.
- The data in a ring topology flow in a **clockwise direction**.
- **Failure of any device** or connection in the ring can cause the **entire network to fail**, making it less fault-tolerant.

Ring Topology



Ring Topology

Tree Topology

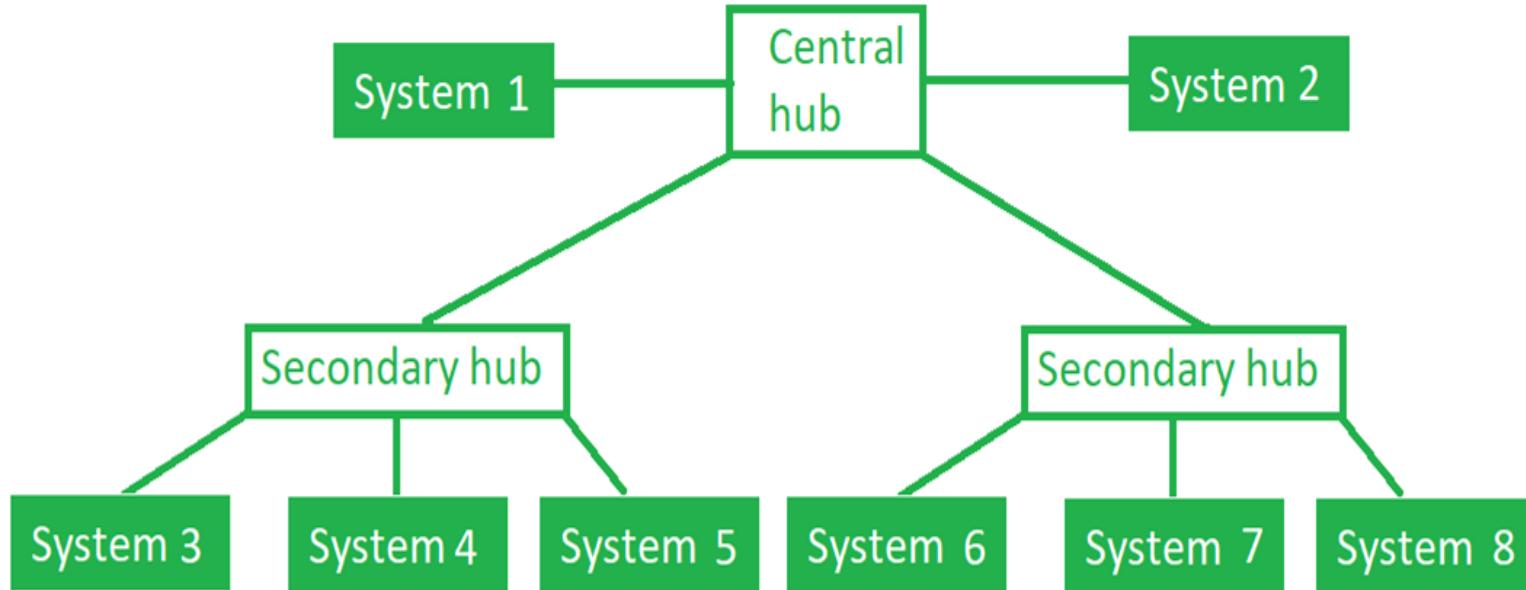


Figure : In this, the various secondary hubs are connected to the central hub which contains the repeater. This data flow from top to bottom i.e. from the central hub to the secondary and then to the devices or from bottom to top i.e. devices to the secondary hub and then to the central hub.

- Tree topology, also known as **hierarchical topology**, is a network configuration that combines characteristics of both **bus and star topologies**.
- In a tree network, all devices are connected to a central hub, which acts as the root of the tree. From this central hub, branches extend out to other hubs or end devices, creating a hierarchical structure.
- It is a multi-point connection and a non-robust topology because **if the backbone fails** the topology crashes.

Key features of a tree topology:

1. **Central hub:** The central hub is the primary element of the tree topology and is responsible for connecting all the branches and end devices in the network. It acts as the main communication point for all connected devices.
2. **Secondary hub:** It is the intermediate levels of the hierarchy. Each **Secondary hub** is connected to the central hub or other intermediate hubs. These branches can be further extended into sub-branches, creating a multi-level hierarchy.

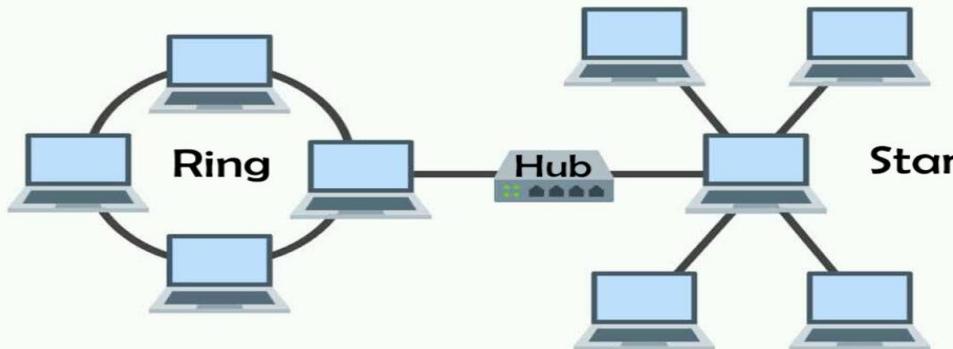
3. End devices: The end devices are the devices located at the leaves of the tree. These devices can be computers, printers, switches, or any other networked devices.

4. Hierarchical structure: The tree topology exhibits a hierarchical arrangement of devices, which makes it easy to manage and scale the network. Information generally flows from the top (root) of the tree down to the leaves and vice versa.

Advantages of tree topology:

- 1. Scalability:** Tree topologies can be easily scaled by adding more branches or connecting additional end devices to the existing branches.
- 2. Centralized control:** The central hub provides a single point of control and management for the entire network.
- 3. Fault isolation:** If a branch or an end device fails, only the devices in that branch or connected to that branch are affected, leaving the rest of the network intact.

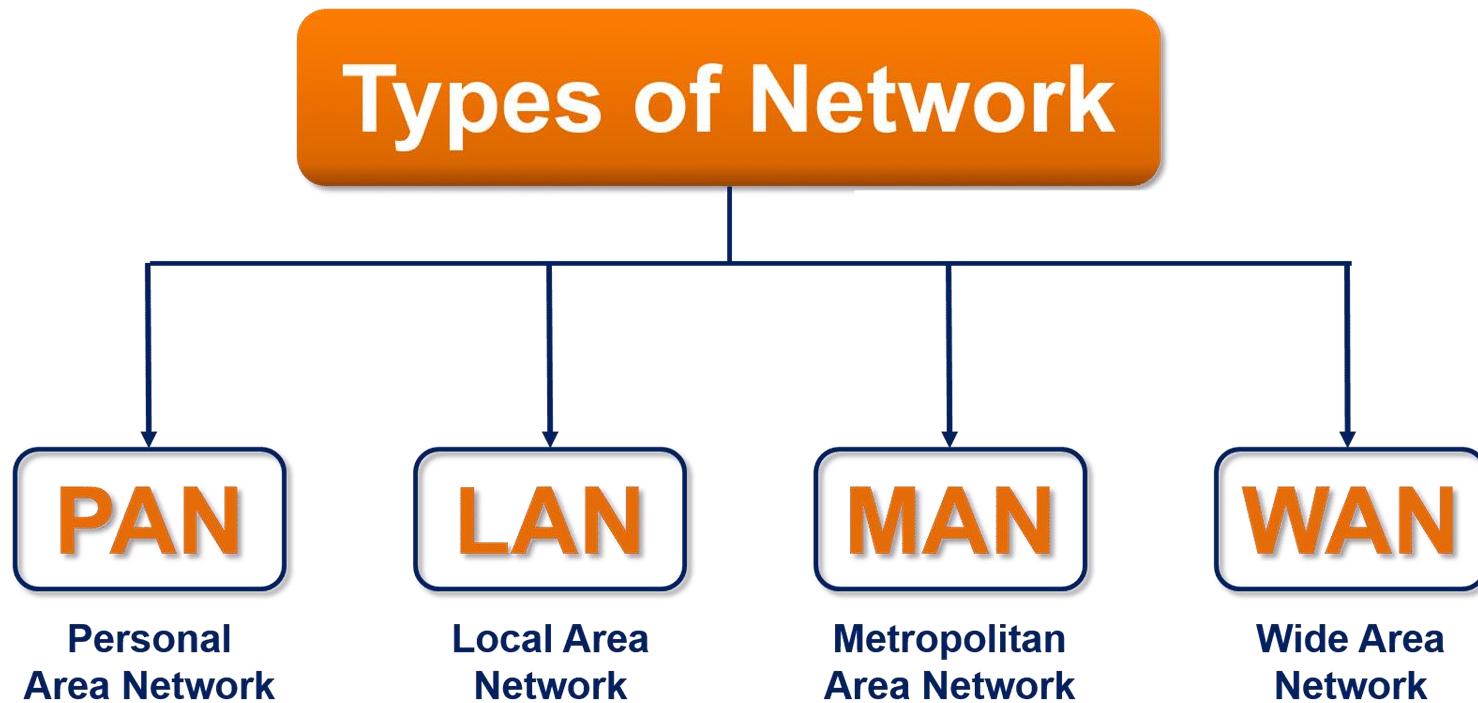
Hybrid Topology

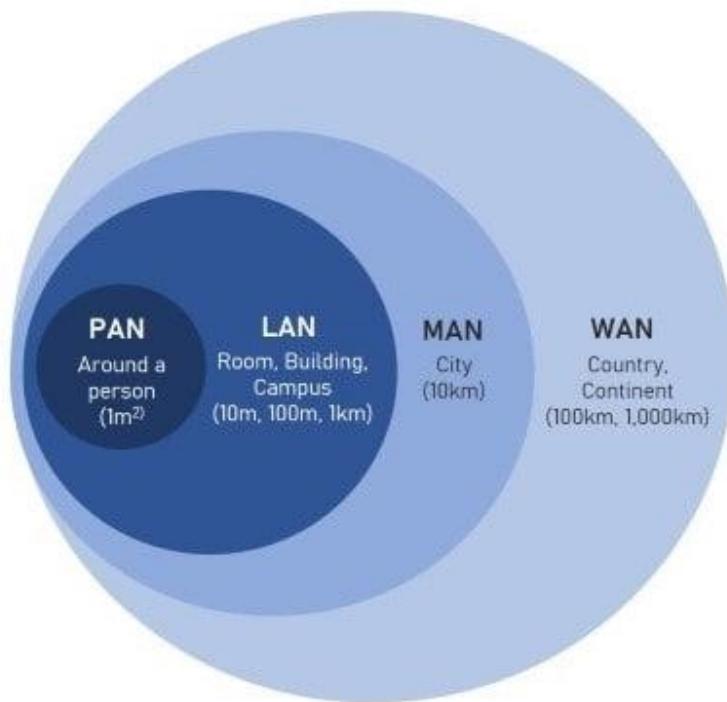


- This topological technology is the combination of all the various types of topologies we have studied above.
- Hybrid Topology is used when the nodes are free to take any form.
- It means these can be individuals such as Ring or Star topology or can be a combination of various types of topologies seen above.

Types of **N**etworks

- Networks can be categorized into various types based on their size, geographical coverage, and the way they are structured.





Interprocessor distance	Processors located in same	Example
1 m	Square meter	Personal area network
10 m	Room	
100 m	Building	
1 km	Campus	Local area network
10 km	City	
100 km	Country	
1000 km	Continent	
10,000 km	Planet	The Internet

PAN (Personal Area Network)

- PANs (Personal Area Networks) let devices communicate **over the range of a person.**
- A Personal Area Network (PAN) is a type of network used for connecting personal devices in close proximity to an individual.
- PANs are designed to facilitate communication and data exchange between devices such as **smartphones, tablets, laptops, personal computers, wireless headphones, smartwatches, and other wearable devices.**
- **Example: Bluetooth devices**
 - In the simplest form, Bluetooth networks use the master-slave paradigm of **Fig. 1-7.**
 - The system unit (the PC) is normally the **master**, talking to the mouse, keyboard, etc., as **slaves**.
 - The master tells the slaves what addresses to use, when they can broadcast, how long they can transmit, what frequencies they can use, and so on.

Characteristics of a PAN are:

Limited Coverage: PANs have a very limited coverage area, typically within a range of a few meters or up to about **10 meters (33 feet)**. They are intended to connect devices that are physically close to each other.

Wireless Technology: PANs typically use wireless communication technologies for connectivity.

- **Bluetooth** is one of the most common wireless technologies used in PANs due to its **low power consumption and short-range capabilities**.

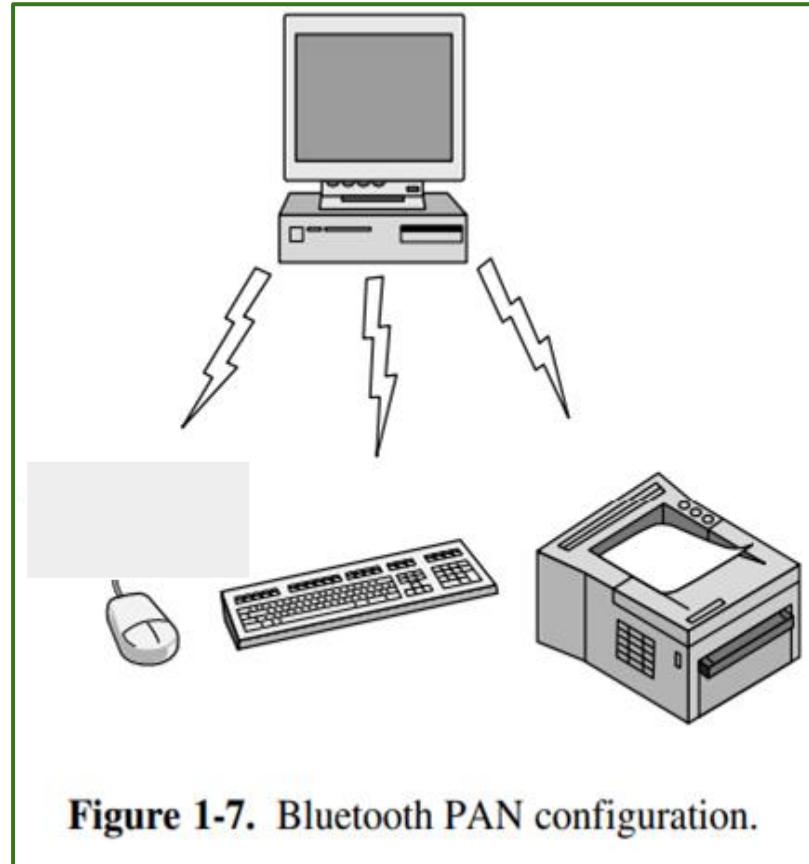


Figure 1-7. Bluetooth PAN configuration.

Examples of PAN Applications:

- ★ Pairing wireless **headphones or speakers** with a smartphone or computer.
- ★ Connecting **a wireless keyboard and mouse** to a laptop or desktop computer.
- ★ Synchronizing data between **a smartphone and a fitness tracker or smartwatch**.
- ★ Transferring files between two smartphones via Bluetooth.

LAN (Local Area Network)

- Local Area Network (LAN) is a type of network that connects devices within a limited geographical area, such as **a single building like a home, office or factory.**
- LANs are widely used to connect personal computers and consumer electronics to let them share resources (e.g., printers) and exchange information.
- Types of LAN networks:
 - a. **Wired LAN**
 - b. **Wireless LAN**

Wired LAN:

- **Ethernet(IEEE 802.3)** is the most common and widely used LAN technology.
- It uses **twisted-pair or fiber-optic cables** to connect devices in a star topology.
- Wired LANs run at speeds of **100 Mbps to 1 Gbps**, have low delay(microseconds or nanoseconds), and make very few errors.

Wireless LAN:

- There is a standard for **wireless LANs called IEEE 802.11**, popularly known as **WiFi**, to connect devices without the need for physical cables.
- WLANs use **radio waves** to transmit data over the airwaves, providing wireless connectivity and mobility to devices within the network.
- It runs at speeds anywhere from **10 to hundreds of Mbps**.
- **Access Points:**
 - Access points are sometimes called **base stations**. The access points connect to the wired network, and all communication between clients goes through an access point
 - Access points provide the wireless connectivity and allow devices to connect to the WLAN.
- WLANs are widely used in homes, offices, public spaces, airports, hotels, educational institutions, and various other environments to provide wireless internet access and facilitate device connectivity.

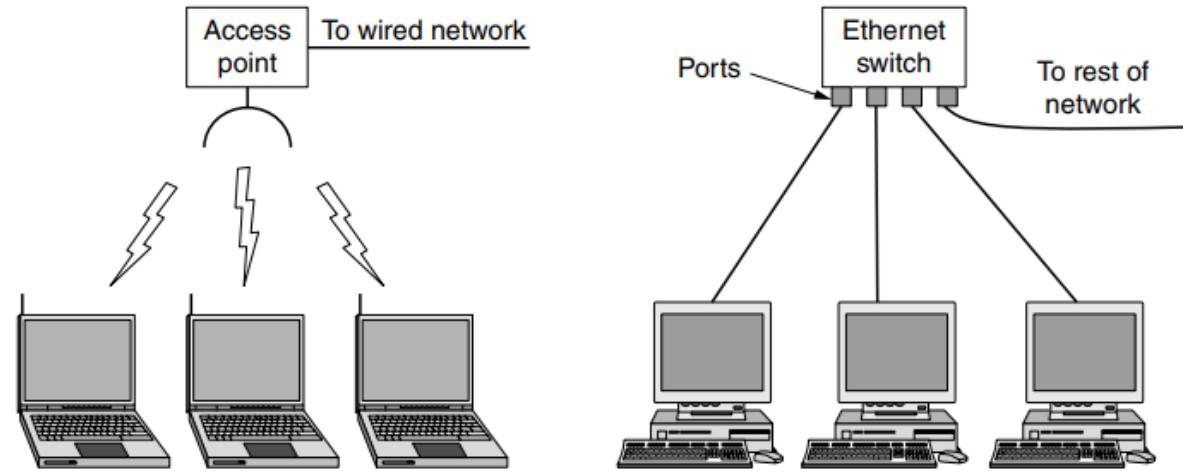
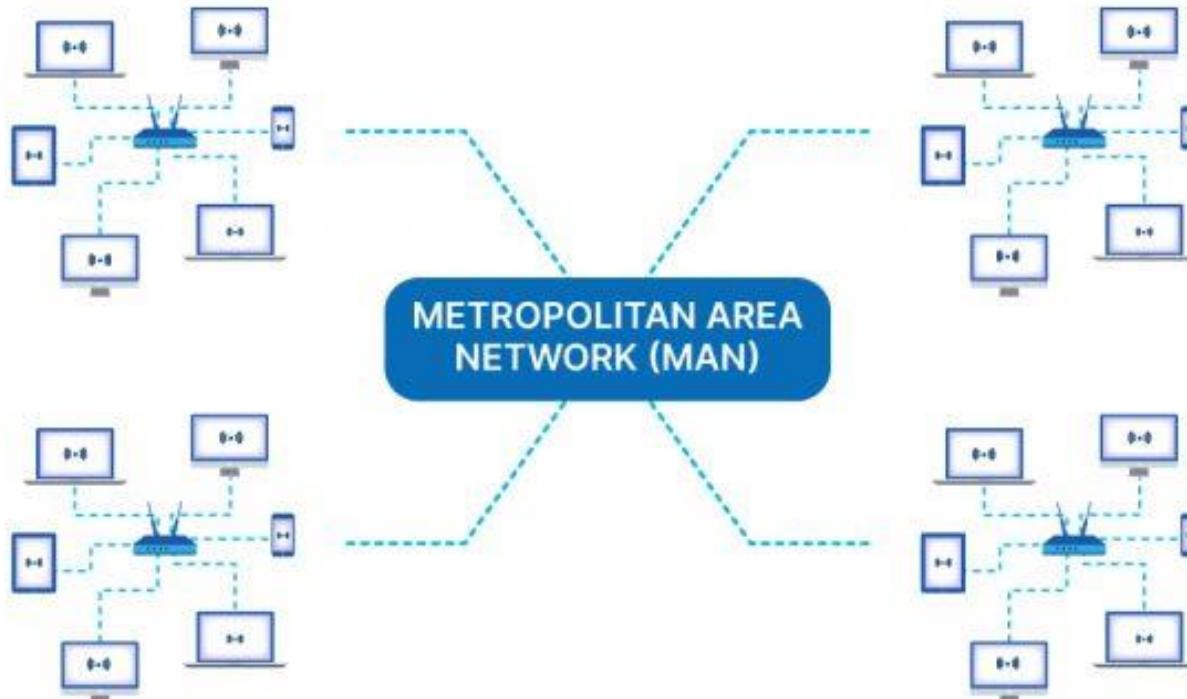


Figure 1-8. Wireless and wired LANs. (a) 802.11. (b) Switched Ethernet.

- Fig. 1-8(a & b) shows a sample topology of **WLAN** and **Switched Ethernet**.
- Each computer speaks the Ethernet protocol and connects to a box called **a switch** with a point-to-point link.
- Hence the name. A switch has multiple ports, each of which can connect to one computer.
- The job of the switch is to relay packets between computers that are attached to it, using **the address in each packet** to determine which computer to send it to.

MAN (Metropolitan Area Network)



METROPOLITAN AREA NETWORK (MAN)

- MAN covers a larger geographical area than a Local Area Network (LAN) but is smaller than a Wide Area Network (WAN).
- A MAN typically spans **a city or a metropolitan region**, connecting multiple LANs and data centers within the area.
- The best-known examples of MANs are **the cable television networks** available in many cities.
- These systems grew from earlier community **antenna systems** used in areas with poor over-the-air television reception.
- In those early systems, a large antenna was placed on top of a nearby hill and a signal was then piped to the subscribers' houses.
- When the Internet began attracting a mass audience, the cable TV network operators began to realize that with some changes to the system, they could provide two-way Internet service in unused parts of the spectrum.
- At that point, the cable TV system began to morph from simply a way to distribute television to a metropolitan area network.
- a MAN might look something like the system shown in Fig. 1-9.
- In this figure we see both television signals and Internet being fed into **the centralized cable headend** for subsequent distribution to people's homes.

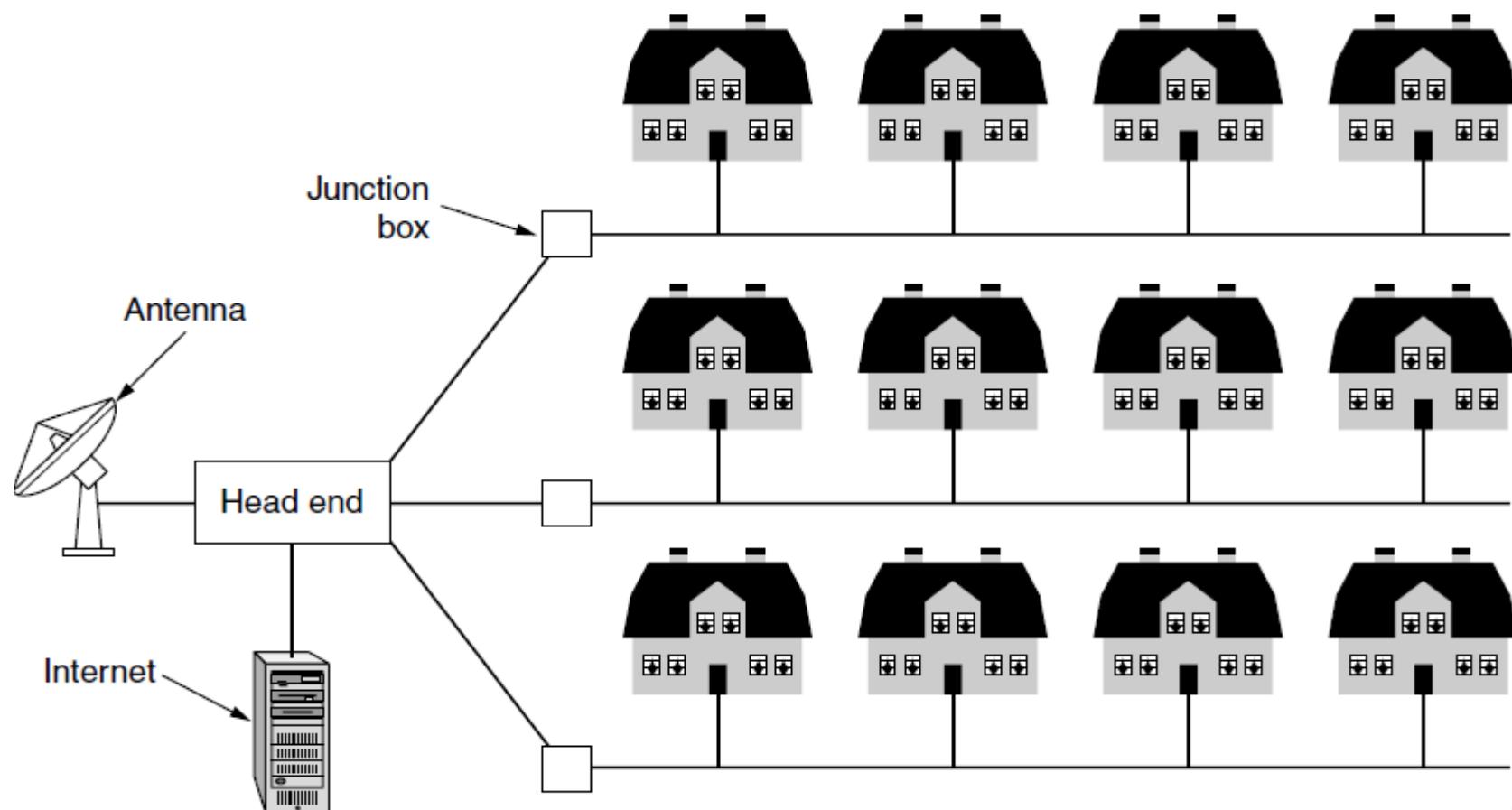
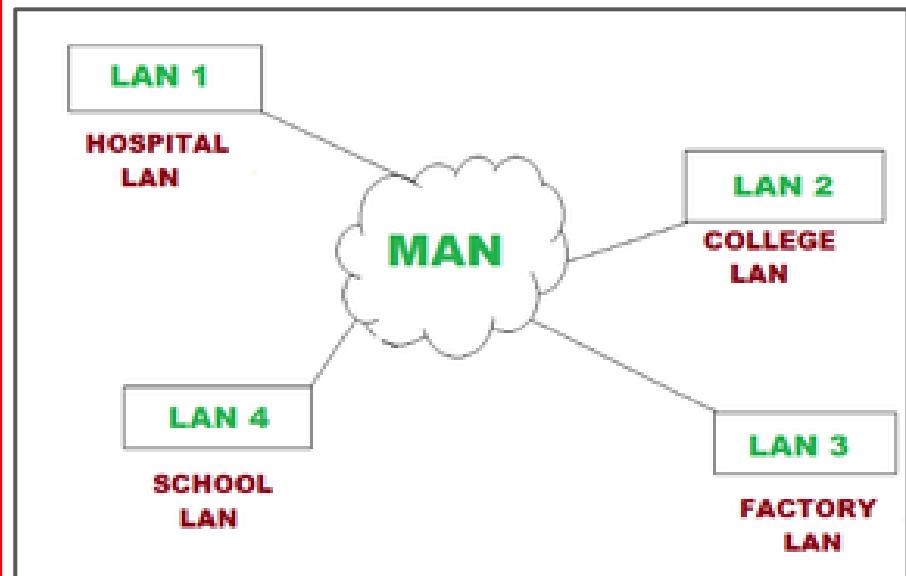


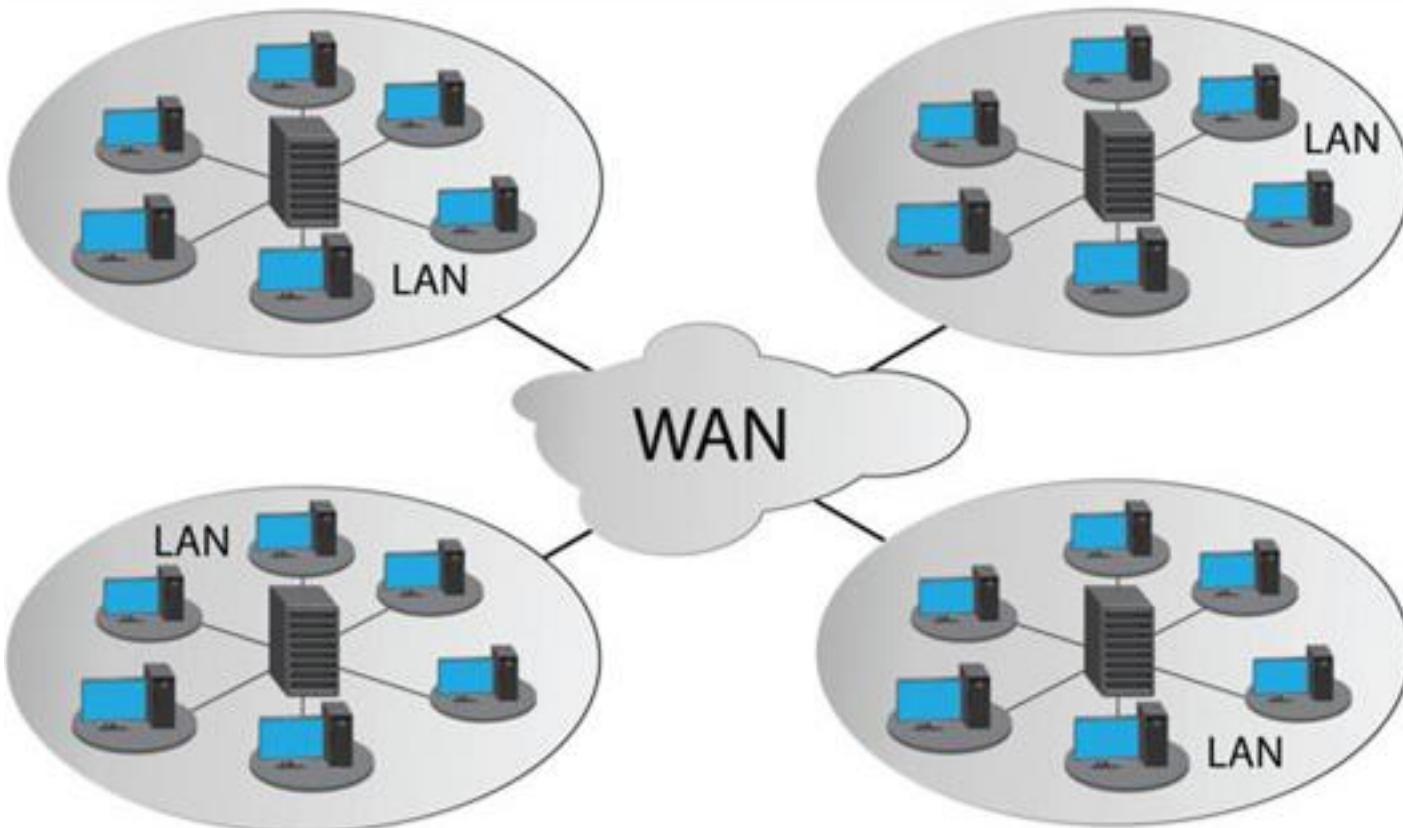
Figure 1-9. A metropolitan area network based on cable TV.

Q:: How does the MAN network work?

- MAN's primary goal is to establish a communication link between two independent LAN nodes in order to connect geographically dispersed LANs.
- To accomplish this, the Metropolitan Area Network typically uses **optical fiber** as a transmission medium, and the network is built with the help of routers and switches.
- Eg:
 - **Cable TV network**
 - **Telephone networks**



WAN (Wide Area Network)



- WAN is a type of computer network that spans a large geographical area, typically connecting multiple Local Area Networks (LANs) or Metropolitan Area Networks (MANs) across **cities, states, countries, or even continents**.
- We will begin our discussion with wired WANs, using the example of a company with branch offices in different cities.
- The WAN in Fig. 1-10 is a network that connects offices in Perth, Melbourne, and Brisbane. Each of these offices contains computers intended for running user (i.e., application) programs.
- We will follow traditional usage and call these machines hosts. The rest of the network that connects these hosts is then called the communication subnet, or just subnet for short.
- The job of the subnet is to carry messages from host to host, just as the telephone system carries words (really just sounds) from speaker to listener.

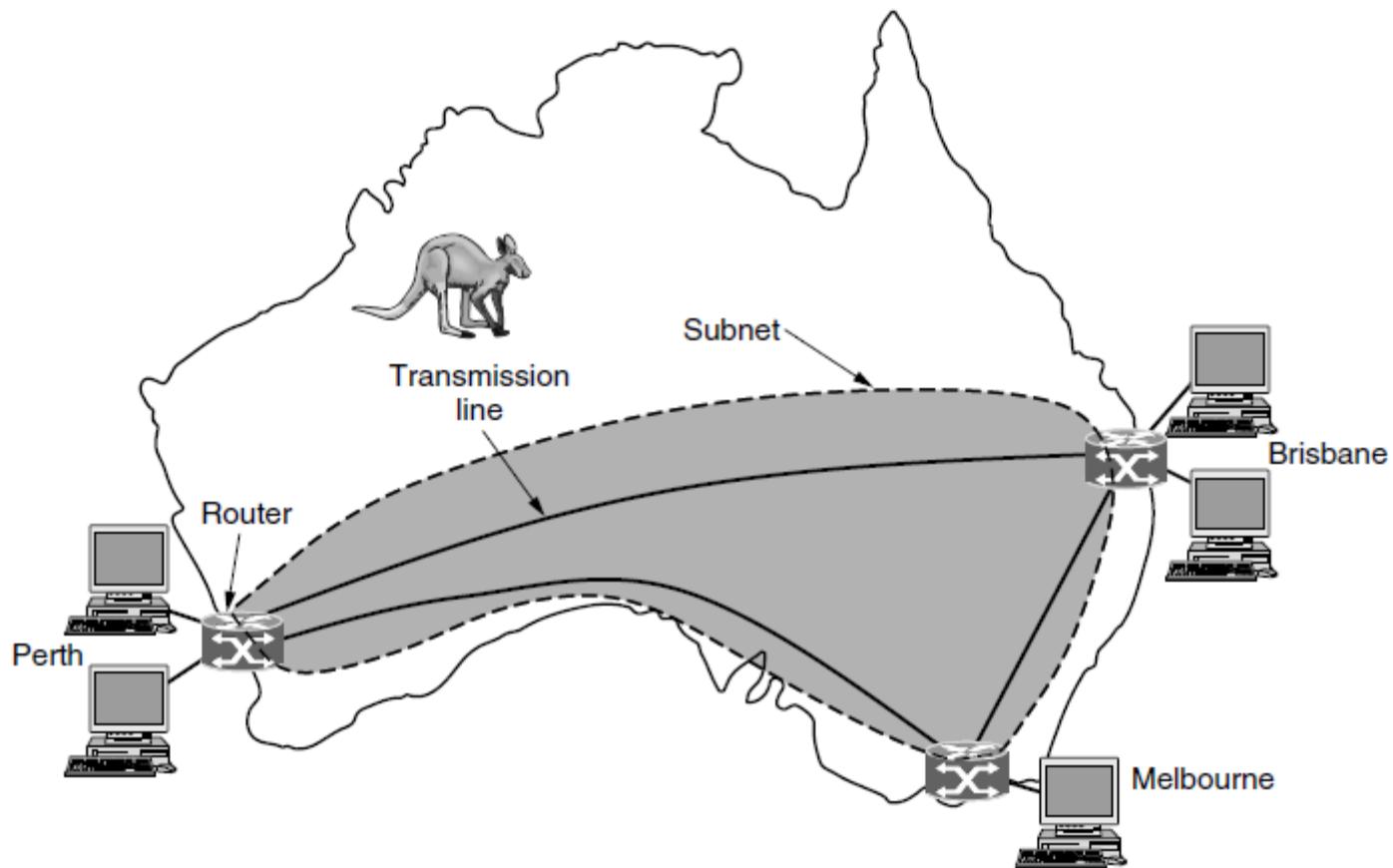


Figure 1-10. WAN that connects three branch offices in Australia.

- In most WANs, the subnet consists of two distinct components:
 - **Transmission Lines**
 - **Switching elements.**
- **Transmission lines** move bits between machines. They can be made of copper wire, optical fiber, or even radio links.
- **Switching elements**, or just switches, are specialized computers that connect two or more transmission lines.
- When data arrive on an incoming line, the switching element must choose an outgoing line on which to forward them.
- These switching computers have been called by various names in the past; the name **router** is now most commonly used.

Types of Computer Networks



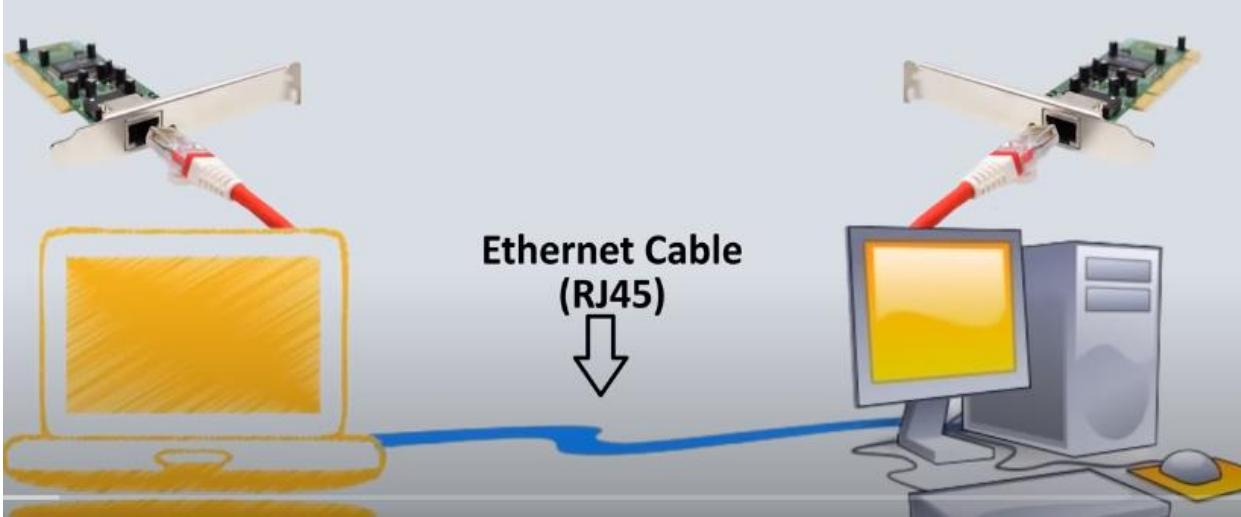
Reference Models

1. The OSI Reference Model

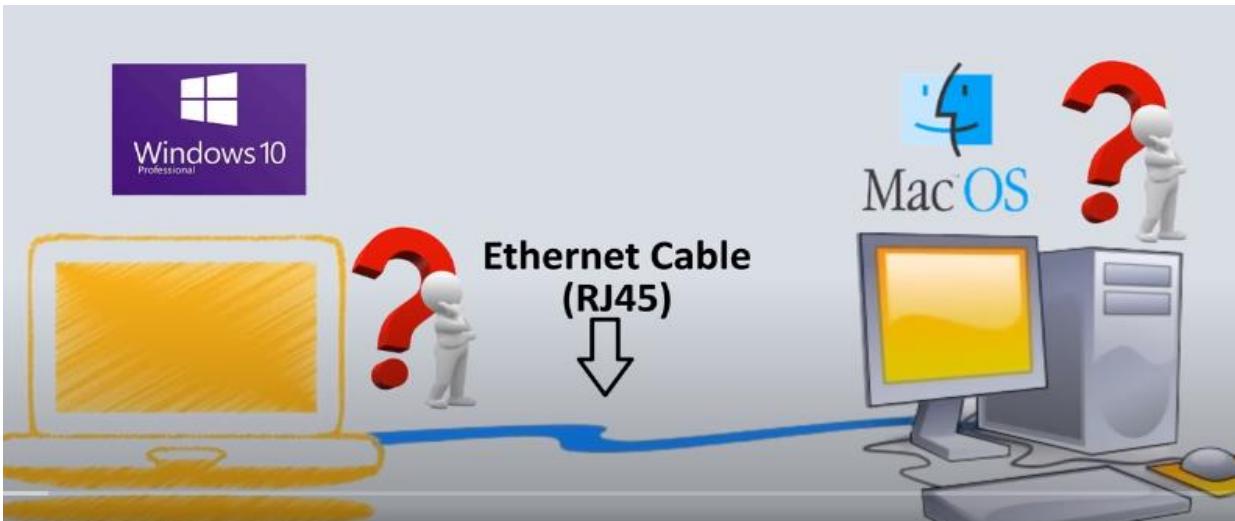
2. The TCP/IP Reference Model

[1]

Why Reference Models?



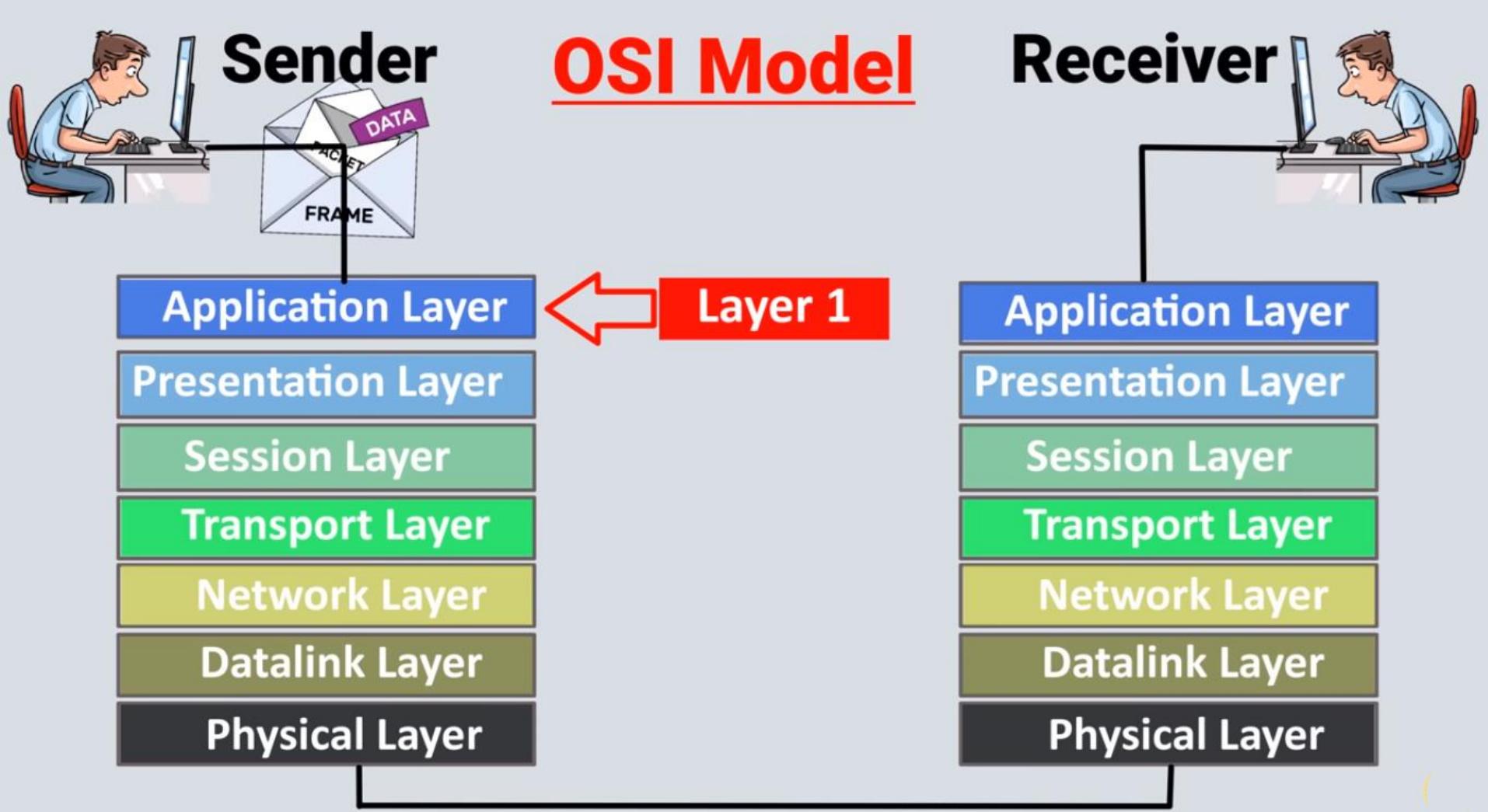
[2]

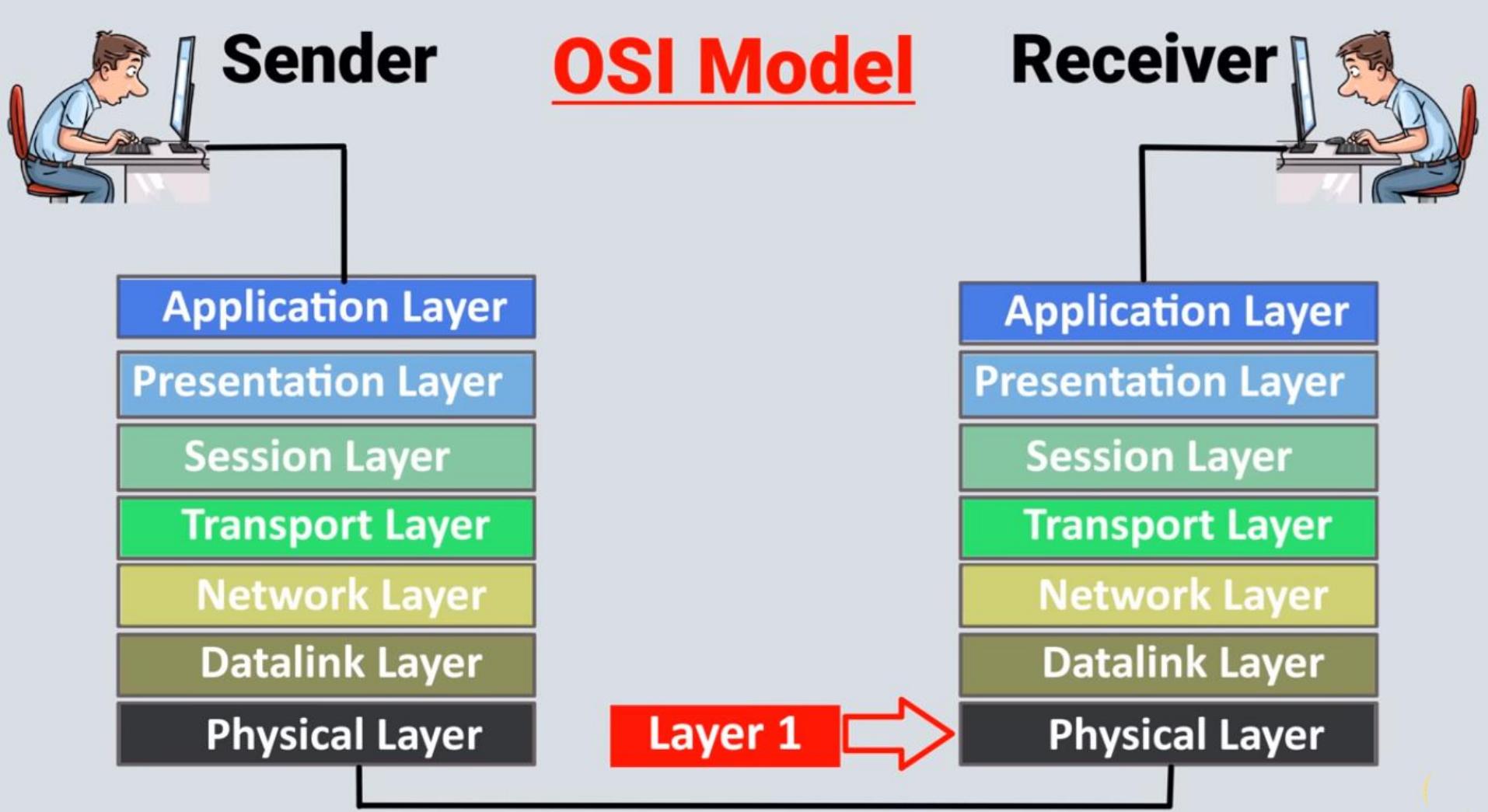


- ★ The OSI (Open Systems Interconnection) model and the TCP/IP (Transmission Control Protocol/Internet Protocol) model are both **conceptual frameworks** used to **understand and describe how data communication occurs over a network.**
- ★ They both break down the communication process into **layers**, but there are some key differences between the two models.
- ★ Both models serve as important tools for understanding network communication, and their concepts are used in the design and implementation of modern networks and protocols.
- ★ The TCP/IP model's practical implementation and widespread use have made it the dominant reference model in today's networking landscape.

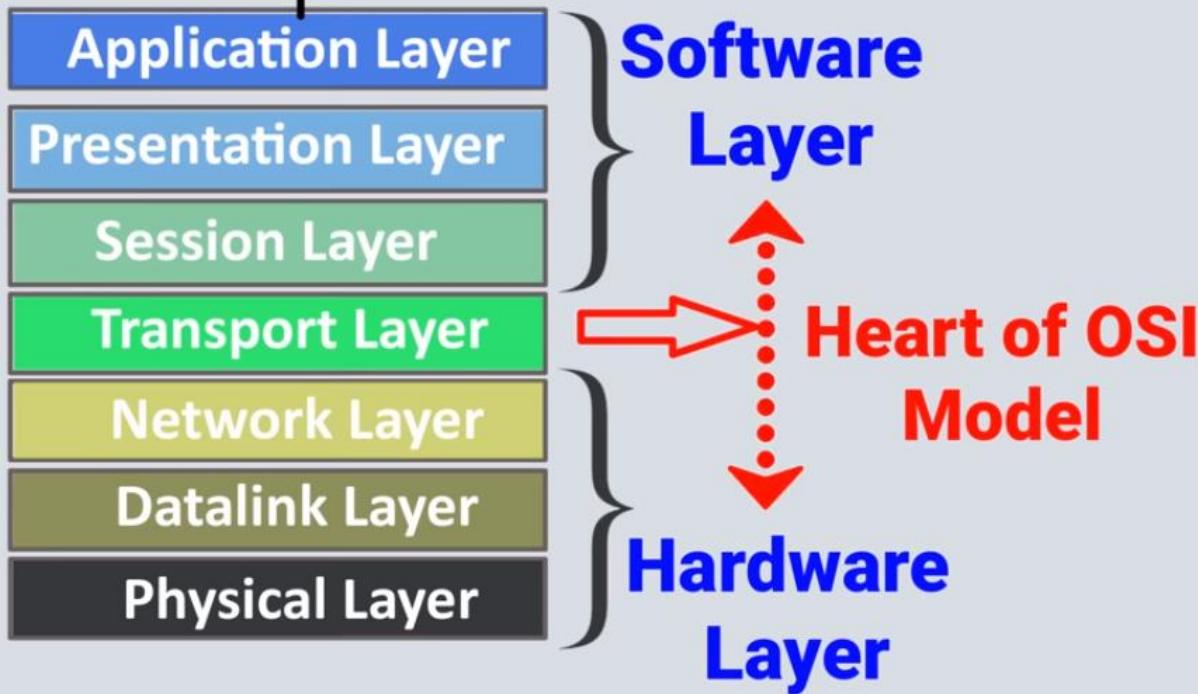
OSI Reference Model

- OSI stands for **Open Systems Interconnection**. It has been developed by ISO – ‘**International Organization for Standardization**‘, in the year **1984**.
- The OSI (Open Systems Interconnection) reference model is **a conceptual framework** used to understand and standardize how different networking protocols and technologies interact and communicate with each other.
- The purpose of OSI reference model is to guide technology vendors(Microsoft,CISCO) and developers , so their H/W and S/W can interoperate and define common framework.
- It is a **7-layer architecture** with each layer having specific functionality to perform.
- All these 7 layers work collaboratively to transmit the data from one person to another across the globe.





OSI Model



Application Layer

- ★ At the very top of the OSI Reference Model stack of layers.
- ★ It provides **interface to end user**. The application Layer is also called **Desktop Layer**
- ★ Its **primary focus** is to enable communication between software applications running on different devices across a network.
- ★ All applications and Softwares we use in daily life like **Google chrome, Firefox,Drop box , Yahoo..etc**, that all work with the help of Application layer protocols.
- ★ Some examples of Application layer protocols include **HTTP** (Hypertext Transfer Protocol) for web browsing, **SMTP** (Simple Mail Transfer Protocol) for sending email, **FTP** (File Transfer Protocol) for file transfers, and **DNS** (Domain Name System) for translating domain names to IP addresses.



HTTP , HTTPS for Web Browsing



SMTP (Simple Mail Transfer Protocol) for sending email



Dropbox



Xender

FTP for File Transfer

Presentation Layer



www.Google.com

Encryption

Converting

110100010110

!@#\$%0&1@!

Decryption

- The presentation layer is also called the **Translation layer**.
- The data from the application layer is extracted here and manipulated as per the required format to transmit over the network.
 - i.e Converts Application layer data into Machine understandable binary format (1's and 0's).
- **The Functions of the Presentation Layer are**
 - **Translation:** The Presentation Layer translates data from the application layer's format into a common format that can be understood by both the sender and receiver. This translation is necessary when different systems use different data formats.
 - **Encryption/ Decryption:** Data encryption translates the data into another form or code. The encrypted data is known as the ciphertext and the decrypted data is known as plain text. This ensures that data remains confidential and protected from unauthorized access.
 - **Compression:** Reduces the number of bits that need to be transmitted on the network.



Session Layer

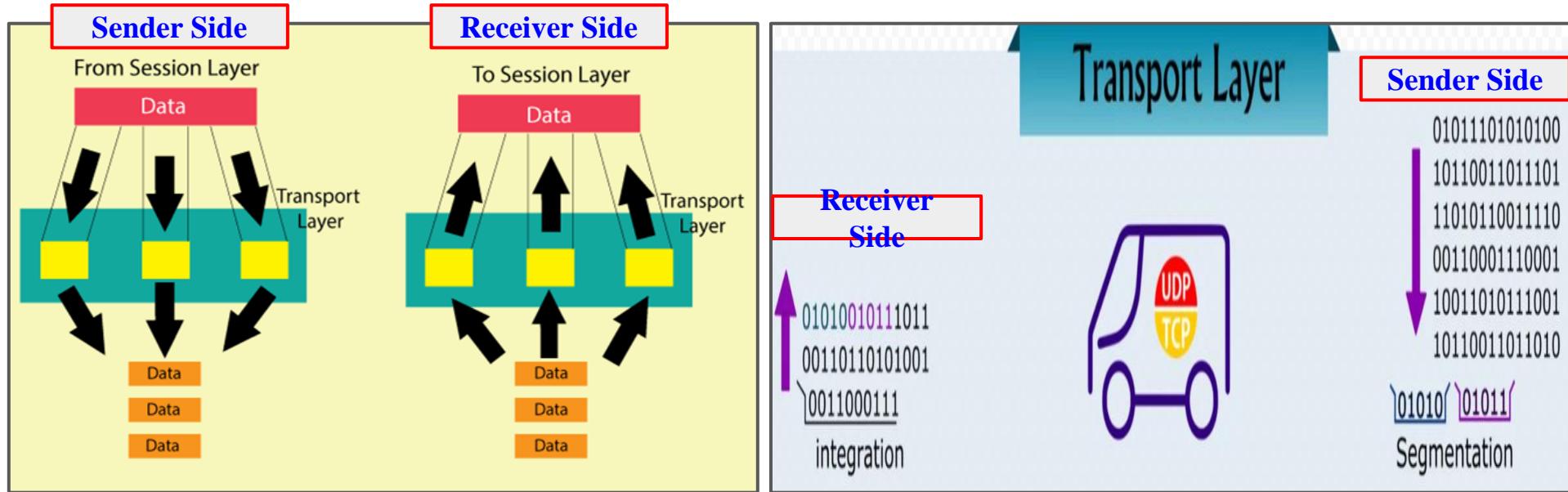
- Session Layer is responsible for establishing, maintaining, and terminating sessions or connections between two communicating devices on a network. Its primary goal is to provide synchronization and coordination between the two endpoints to enable reliable data exchange.
- Key functions of the Session Layer include
 - ◆ **Session Establishment:** The Session Layer is responsible for setting up and initiating communication sessions between two devices. It establishes a logical connection between the sender and receiver before data exchange begins.
 - ◆ **Session Maintenance:** Once the session is established, the Session Layer ensures its stability and integrity during the data transfer. It manages the session and monitors its status to handle any issues that may arise.
 - ◆ **Session Termination:** When the communication between the devices is complete, the Session Layer terminates the session in an organized manner, releasing any allocated resources and freeing up system memory.



Transport Layer

- The Transport layer is responsible for the transmission of data across network connections.
- The transport layer provides services to the application layer and takes services from the network layer.
- The transport layer is called as **Heart of the OSI model.**
- The data in the transport layer is referred to as **Segments.**
- It is responsible for the **“End to End Delivery” of the complete message.**
- The transport layer also provides the acknowledgment of the successful data transmission and re-transmits the data if an error is found.
- The Functions of the Transport Layer are :
 - **Segmentation and Reassembly**
 - **Flow Control**
 - **Error control**

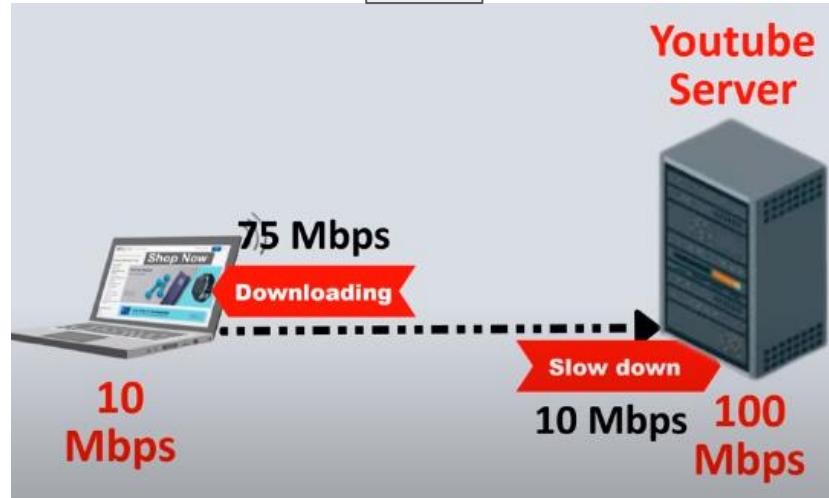
1. Segmentation and Reassembly



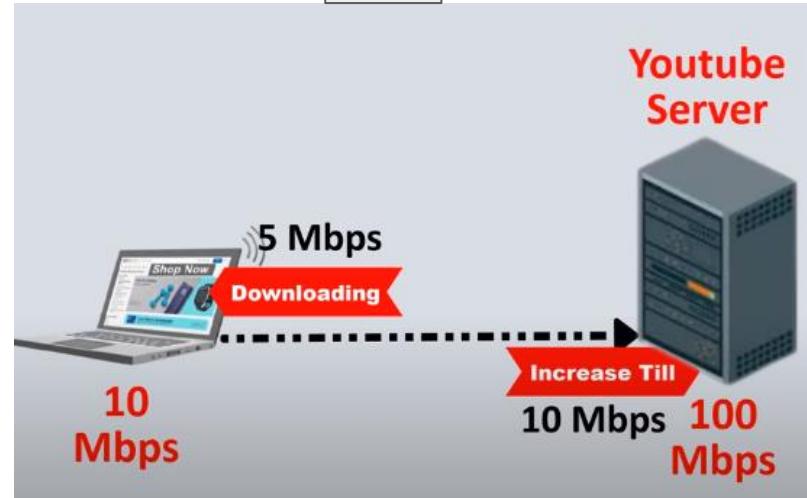
- Transport layer **breaking up messages into smaller segments**, then transmitting them over the network.
- The transport layer also **reassembles the segments into the original message** when they reach their destination.

2. Flow Control

1



2



- The Transport Layer manages the flow of data between the sender and receiver to prevent overwhelming the receiver with data.
- It uses flow control mechanisms to regulate the rate of data transmission and ensure that the receiver can handle the incoming data..

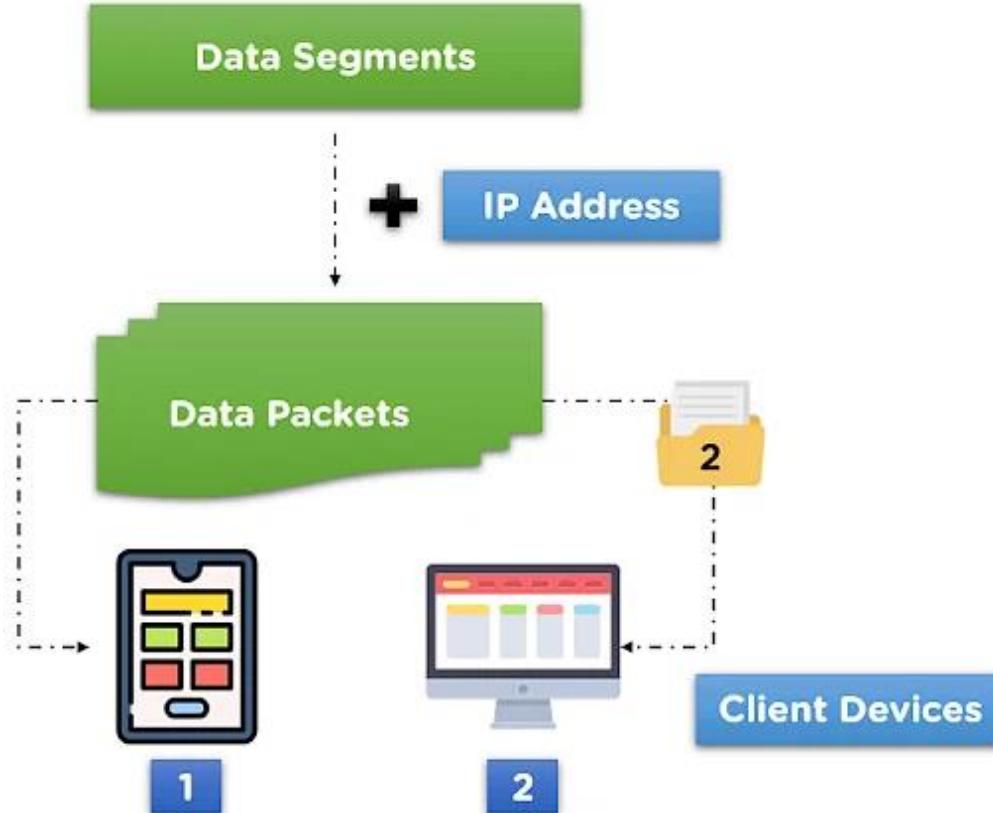
- **Error Control:** This layer is responsible for error detection and correction. It checks for errors in the received data and requests retransmission of any lost or corrupted segments to ensure the data's integrity.

Transport layer protocols

- Protocols in the transport layer of the OSI model provide **communication between applications on different hosts**.
- The two main **Transport layer protocols** are:
 - **Transmission Control Protocol [TCP]**
 - The Transmission Control Protocol (TCP) is **connection-oriented**, meaning that before exchanging data, the two applications must establish a connection between them.
 - After the connection is established, packets are sent and received reliably.
 - It provides **reliable** communication between two hosts.
 - **User Datagram Protocol [UDP]**
 - The User Datagram Protocol (UDP) is **connectionless**, meaning that data is exchanged without establishing a connection.
 - Packets are sent and received without any guarantees about their order or delivery(**Unreliable**).

Network Layer

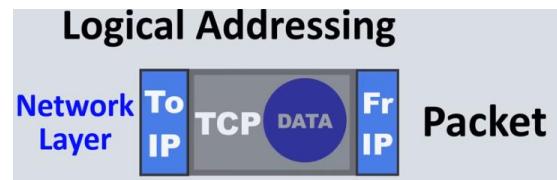
Logical Addressing:



- The Network layer operates above the data link layer (Layer 2) and below the transport layer (Layer 4) in the OSI model.
- It is responsible for delivery of data from original source to destination.

Services provided by the Network layer are:

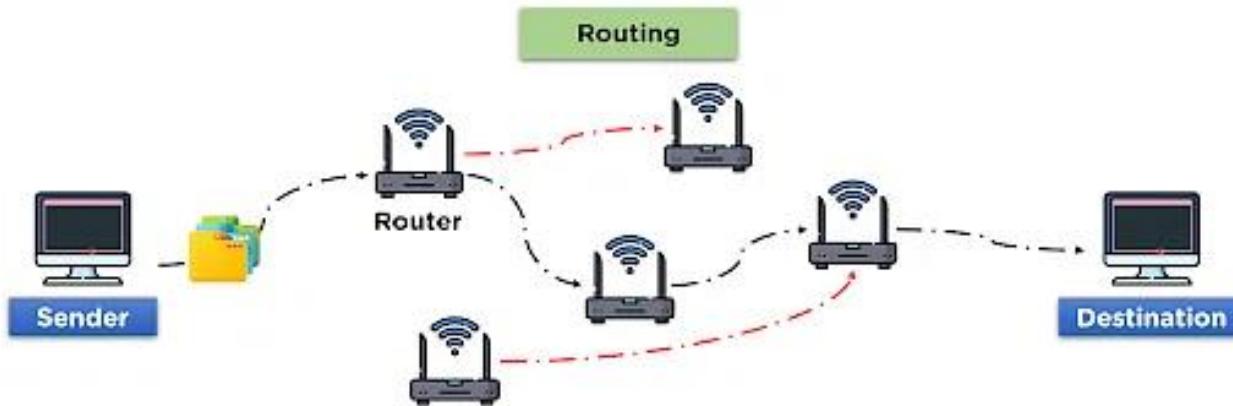
Logical Addressing:



- It is responsible for breaking down the **data segments** into **data packets** and is tasked with **reassembling** them on the receiver side.
- Devices at the network layer are identified using logical addresses, such as **IP (Internet Protocol)** addresses. The network layer adds the source and destination IP addresses to the data packets to ensure proper delivery.

Routing:

- The Network layer determines the **most efficient path** for data packets to travel from the source to the destination across multiple networks.
- It uses **routing algorithms** and protocols to make forwarding decisions based on the network topology and destination address.



Congestion:

- The network layer can detect and respond to network congestion, either by slowing down the rate of packet transmission or by using other congestion control mechanisms.

What is Congestion:

- Congestion in the network layer is a situation where **the network is overloaded with traffic**, and this can lead to a number of problems, including:
 - **Packet loss**
 - **Increased delay**

Data Link Layer

- ★ The data link layer is responsible for the **node-to-node delivery of the message.**

Services provided by Data Link Layer:

1. Framing:

Physical Addressing



- ★ The Data Link Layer takes the **packets** received from the Network Layer and encapsulates them into **frames** for transmission over the physical medium. At the receiving end, it extracts the data from the frames and passes it up to the Network Layer.
- ★ When a **packet** arrives in a network, it is the responsibility of the DLL to transmit it to the Host using its **MAC address**(**Media Access Control**).
- ★ MAC addresses are used for addressing and delivering data frames to the correct destination on the same network.

2. Error Detection and Correction: The data link layer is responsible for **error detection** at the link level. It checks for transmission errors in received frames using error-checking mechanisms like **CRC (Cyclic Redundancy Check)**. If errors are detected, the layer can request **retransmission** of the corrupted frames.

3. Flow Control: The Data Link Layer manages the flow of data between devices to avoid overwhelming the receiving end. It ensures that data is sent at a rate that the receiving device can handle.

4. Access control :It ensures that only authorized devices can access the network.

Note:

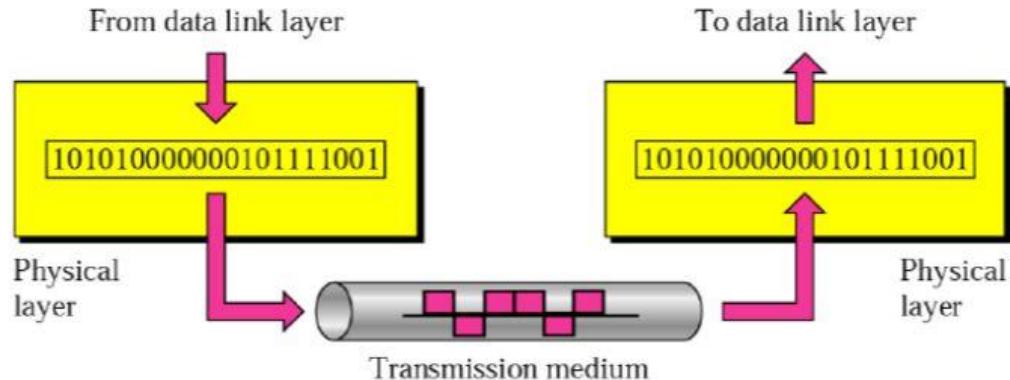
Switch & Bridge are Data Link Layer devices.



Physical Layer



- It deals with the **physical transmission** of **data bits** over a physical medium, such as **copper cables, fiber optics, or wireless channels**.
- The primary function of the Physical Layer is to establish and maintain a physical link between network devices and transmit **raw binary data** as electrical or optical signals.
- It is also responsible for converting the **data frames** received from the Data-link layer into **data bits of 1's and 0's (raw bits)** for transmission over the network.



- The physical layer consists of **three** main components:
 - Transmission media
 - Transceivers
 - Connectors
- **Transmission media** includes wires, cables, and optical fibers.
- **Transceivers** convert electrical signals into optical or radio signals.
- **Connectors** attach transmission media to devices such as computers, hub, repeaters, modems .

Note:

Hub, Repeater, Modem, and Cables are Physical Layer devices.

Data Format	Layer	Function
Data	Application Layer	Applications access network services
Data	Presentation Layer	Translations ,Encryption and decryption of data
Data	Session Layer	Connection management b/w networks
Segment	Transport Layer	Segmentation,Flow control, responsible for End-to-end delivery
Packet	Network Layer	Adding IP add, Determine the path for data transfer
Frame	Data Link Layer	Framing, Error detection and correction
Raw Bits	Physical Layer	Transfer raw bits using physical media

TCP/IP Model

OSI model acts as a reference model and is not implemented on the Internet because of its late invention. **The current model being used is the TCP/IP model.**

- TCP/IP was designed and developed by the **Department of Defense (DoD)** in the 1960s and is based on standard protocols.
- It stands for **Transmission Control Protocol/Internet Protocol**.
- The TCP/IP model is a concise version of the OSI model. It contains **four layers**, unlike the seven layers in the OSI model.

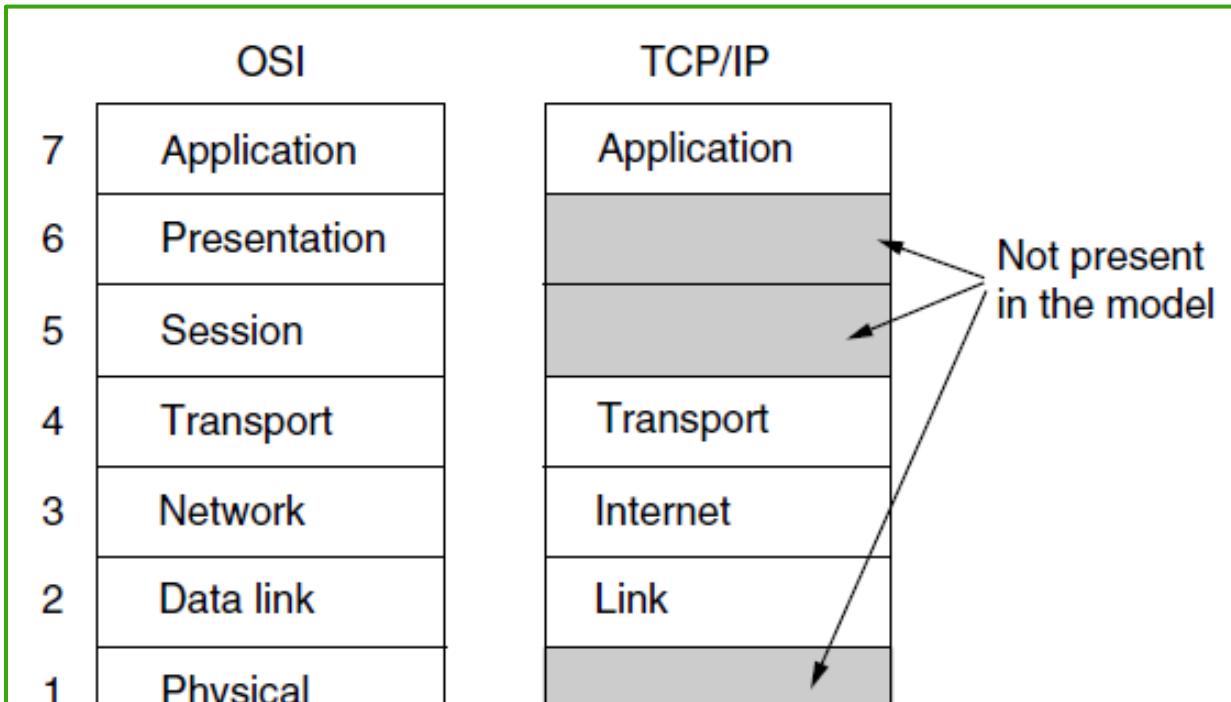


Figure 1-21. The TCP/IP reference model.

The Link Layer [Network Access Layer]

- A Network Access Layer is the lowest layer of the TCP/IP model.
- A Network Access Layer is the combination of the **Physical layer and Data Link layer** defined in the OSI reference model.
- The Link Layer is the lowest layer of the TCP/IP model and handles **the physical transmission** of data over the network medium.
- It is responsible for defining hardware-specific details such as **MAC (Media Access Control)** addresses for devices, **error detection**.
- The functions carried out by this layer are encapsulating the **IP datagram into frames** transmitted by the network and mapping of IP addresses into physical addresses.
- The **protocols** used by this layer are **ethernet, token ring, FDDI, X.25, frame relay**.

The Internet Layer

- An Internet layer is the second layer of the TCP/IP model.
- An Internet layer is also known as **the network layer**.
- The Internet layer is responsible for **Logical Addressing, Routing, and Congestion Control** data packets across different networks.

The main protocols residing at this layer are as follows:

1. **IP (Internet Protocol)**
2. **ICMP (Internet Control Message Protocol)**
3. **ARP (Address Resolution Protocol)**

1. IP (Internet Protocol) :

- ★ It is responsible for delivering packets from the source host to the destination host by looking at the **IP addresses** in the packet headers.
- ★ IP has 2 versions: **IPv4 and IPv6**.
- ★ IPv4 is the one that most websites are using currently. But IPv6 is growing as the number of IPv4 addresses is limited in number when compared to the number of users.

1. ICMP (Internet Control Message Protocol):

- ★ It is encapsulated within IP Packets and is responsible for providing hosts with information about **network conditions or to report errors**.

1. ARP (Address Resolution Protocol):



- ★ Its job is **to find the hardware address of a host from a known IP address**. ARP has several types: Reverse ARP, Proxy ARP, Gratuitous ARP, and Inverse ARP.

The Transport Layer

1. The Transport Layer is responsible for managing **end-to-end communication** and ensures **reliable data delivery** between applications running on different devices.
2. It handles the **segmentation and reassembly** of data, as well as **flow control and error recovery**.
3. The TCP/IP transport layer protocols exchange data receipt acknowledgments and retransmit missing packets to ensure that packets arrive in order and without error.
4. It ensures that data is delivered in the correct order, without loss or duplication. The most commonly used protocols at this layer are the
 - a. **Transmission Control Protocol (TCP)**, which is connection-oriented and guarantees reliable delivery.
 - b. **User Datagram Protocol (UDP)**, which offers a connectionless and faster but less reliable delivery option.

The Application Layer

- ★ The TCP/IP model does not have session or presentation layers.
- ★ The Application layer is responsible for supporting end-user applications and their protocols.

It contains all the higher-level protocols.

- ★ **TELNET:** It is an abbreviation for Terminal Network. It establishes the connection between the local computer and remote computer in such a way that the local terminal appears to be a terminal at the remote system.
- ★ **FTP:** FTP stands for File Transfer Protocol. FTP is a standard internet protocol used for transmitting the files from one computer to another computer.

The Application Layer

- ★ **HTTP:** HTTP stands for Hypertext transfer protocol. This protocol allows us to access the data over the **world wide web**. It transfers the data in the form of plain text, audio, video. i.e. HTTP takes care of Web Browsers and Websites.
- ★ **SMTP:** SMTP stands for Simple mail transfer protocol. The TCP/IP protocol that supports the e-mail is known as a Simple mail transfer protocol. This protocol is used to send the data to another e-mail address.
- ★ **SNMP:** Simple Network Management Protocol is used for **managing and monitoring network devices and systems**. It is commonly used in network management systems to collect information, configure devices, and monitor network performance.

- ★ **DNS:** The DNS (Domain Name System) protocol is an application-layer protocol used to translate human-readable domain names, such as "**www.google.com**," into their corresponding IP addresses.
- ★ **RTP:** The RTP (Real-time Transport Protocol) for delivering real-time media such as voice or movies over networks.

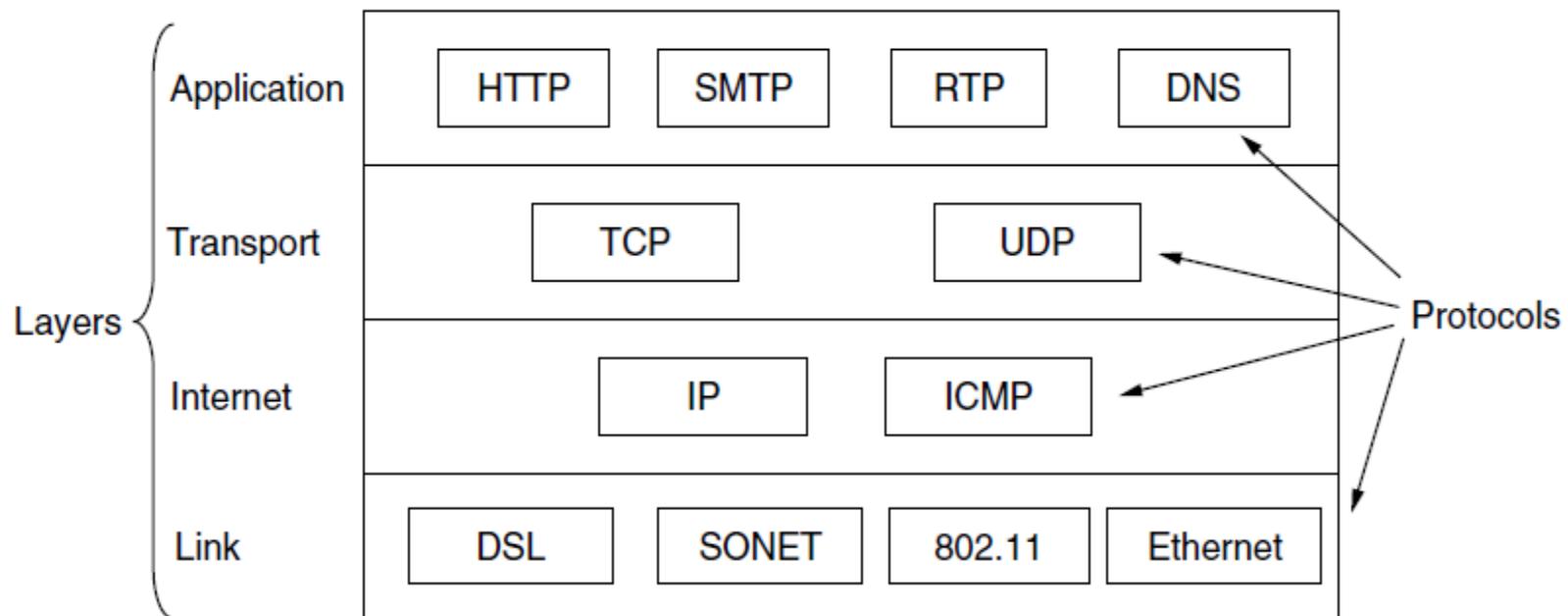


Figure 1-22. The TCP/IP model with some protocols we will study.

OSI vs TCP/IP Reference Model

Assignment Topic

UNIT-1::Part-2

Physical Layer

B SAI BABA,M.Tech(Ph.D),VIT,Bhimavaram

Syllabus

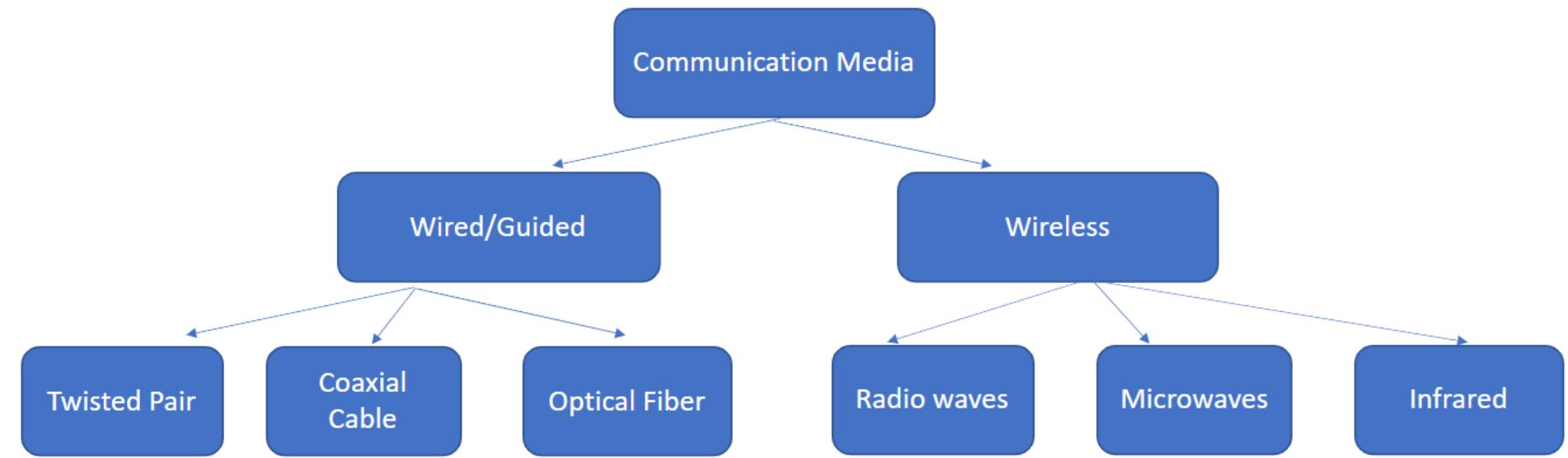
Physical Layer:

- **Guided Transmission Media**
- **Wireless transmission**
- **Mobile telephone system.**

Guided Transmission Media

Transmission Media

- Transmission media, or communication channels or communication lines, refer to the **physical media through which data is transmitted from one device to another.**
- They are used to establish communication between two or more devices.
- It allows them to exchange information and data.
- Types of Transmission Media
 - **Guided Transmission Media (Wired Transmission)**
 - **Unguided Transmission Media (Wireless Transmission)**



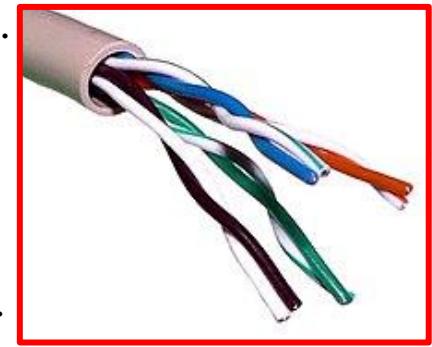
Guided Transmission Media

- Guided Transmission Media, also known as **Wired or Bounded transmission media**, is the physical medium through which the signals are transmitted.
- The transmitted signals are directed and confined in a narrow pathway using physical links.
- It provides us with features like **higher speeds, and better security** and is used preferably for comparatively **shorter distances**.
- There are three types of Guided Transmission Media:
 - **Twisted Pair cable**
 - **Coaxial cable**
 - **Fibre Optic Cable**

Twisted-Pair Cables

Twisted-Pair Cables

- Twisted-Pair Cables are cables consisting of **two insulated conductor wires** (typically copper) wound and twisted together arranged in a regular spiral pattern.
 - ◆ **One wire** carries the signal to the receiver.
 - ◆ **Other Wire** is used as a ground reference.
- A twisted pair cable is **cheap** as compared to other transmission media.
- Installation of the twisted pair cable is easy, and it is a lightweight cable.
- Types:
 - ◆ **Unshielded Twisted Pair Cable**
 - ◆ **Shielded Twisted Pair Cable**

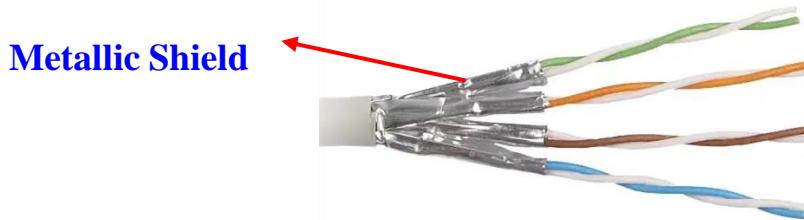


1. Unshielded Twisted Pair Cable



- UTP consists of 4 pairs of color-coded wires twisted around each other.
- The wires are twisted to prevent electromagnetic interference (EMI) and crosstalk between adjacent pairs.
- UTP is a type of copper cable commonly used for **networking and telecommunications**.
- The term "**unshielded**" means that UTP cables *do not have an overall metallic shield or foil layer to protect the twisted pairs from external interference.*
- Instead, each pair of wires is individually insulated, and the twisting of the pairs helps to cancel out electromagnetic interference.

2. Shielded Twisted Pair Cable

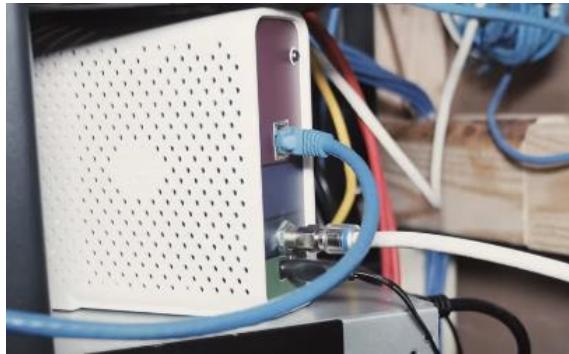


- Shielded twisted pair (STP) is a type of copper cable that is used for **networking and telecommunications**.
- It is similar to unshielded twisted pair (UTP) but *includes an additional metallic shield or foil layer around the individual twisted pairs*.
- The primary purpose of the shielding is to provide protection against electromagnetic interference (EMI) or Cross talks that can degrade the quality of data transmission.

Ethernet Categories

Category	Shielding	Max Transmission Speed
Cat 3	Unshielded	10 Mbps
Cat 5	Unshielded	100 Mbps
Cat 5e	Unshielded	1 Gbps
Cat 6	Unshielded or Shielded	1 Gbps
Cat 6a	Shielded	10 Gbps
Cat 7	Shielded	10 Gbps
Cat 8	Shielded	Upto 40 Gbps

- The difference between these categories is **the maximum speed** they can handle without having any crosstalk(interference).
- **The number represents the tightness of the twists** that are applied to the wire



Coaxial Cable



- Coaxial cable is a type of cable commonly used in networks, particularly for
 - **Cable television (CATV)**
 - **Broadband Internet**
- It consists of *a central conductor wire surrounded by a dielectric insulating material, which is further enclosed by a braided metal shield and an outer protective sheath.*
- The combination of these layers makes coaxial cable suitable for transmitting **high-frequency signals with minimal interference**
- Two kinds of coaxial cable are widely used.
 - One kind, **50-ohm cable**, is commonly used when it is intended for **digital transmission**.
 - The other kind, **75-ohm cable**, is commonly used for **analog transmission and cable television**.
- In the mid-1990s, cable TV operators began to provide Internet access over cable, which has made 75-ohm cable more important for data communication.

- A coaxial cable consists of a stiff copper wire as **the core**, surrounded by **an insulating material**.
- The insulator is encased by a **cylindrical conductor**, often as a closely woven **braided mesh**.
- The **outer conductor** is covered in a protective **plastic sheath**.

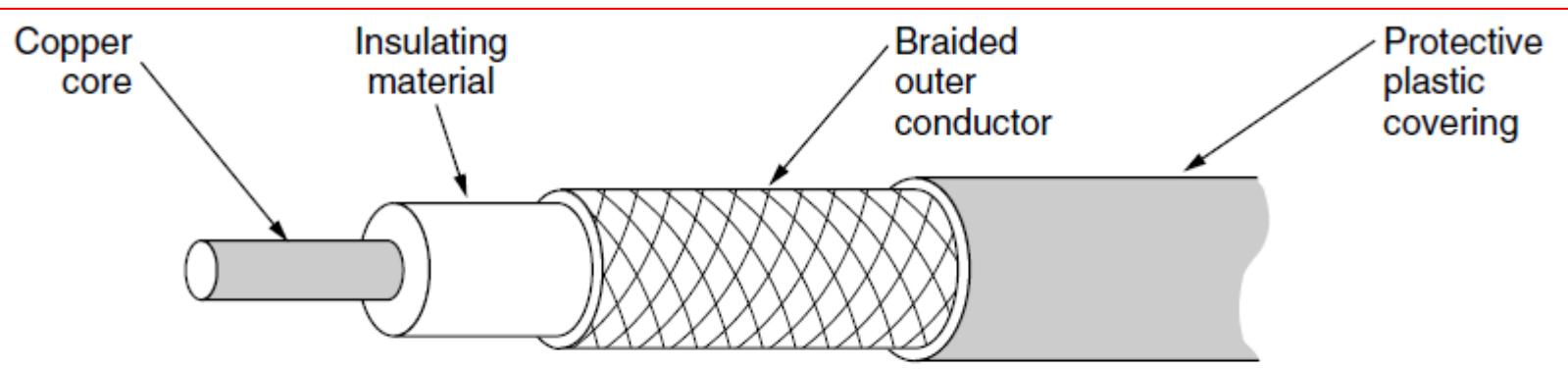
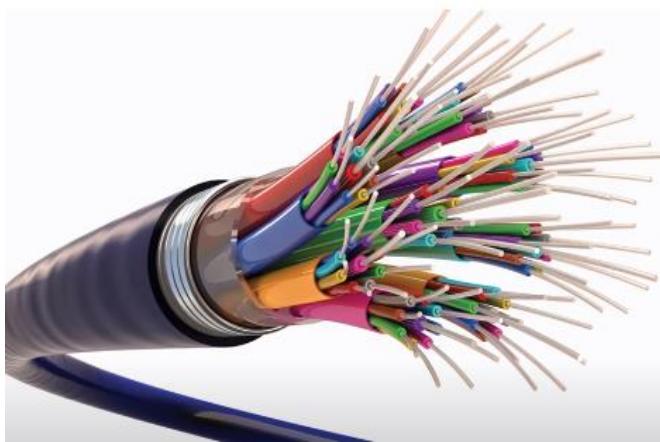
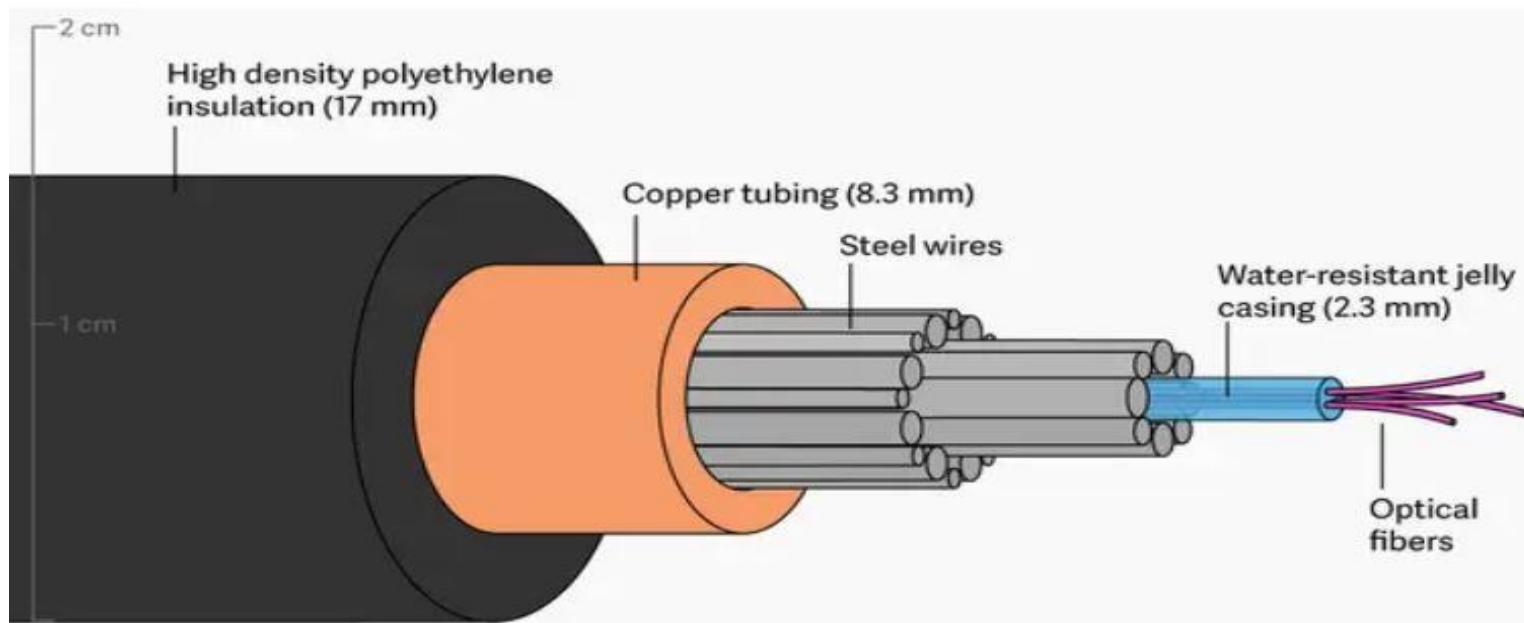


Figure 2-4. A coaxial cable.



Fiber Optics

- Fiber optic cables are a type of transmission media that use **thin strands of glass or plastic fibers** to transmit data as pulses of light.
- They provide **high-speed, long-distance, and secure communication** for various applications, including telecommunications, networking, and data transmission.



- Fiber optic cables are similar to coax, except without the braid. Figure 2-8(a) shows a single fiber viewed from the side.
- At the center is the glass core through which the light propagates.
- In **multimode fibers**, the core is typically **50 microns in diameter**, about the thickness of a human hair. In **single-mode fibers**, the core is **8 to 10 microns**.

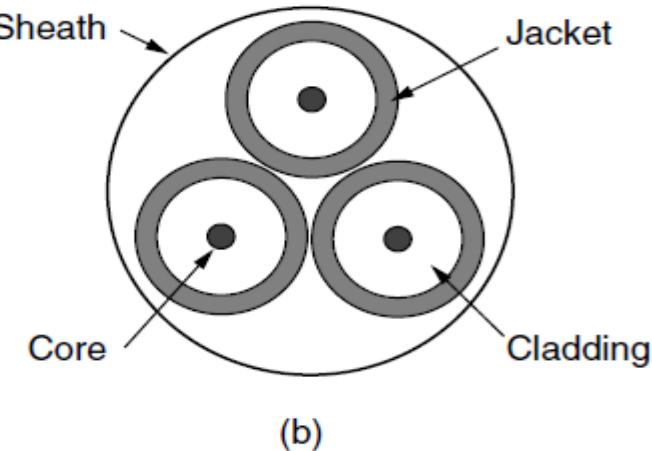
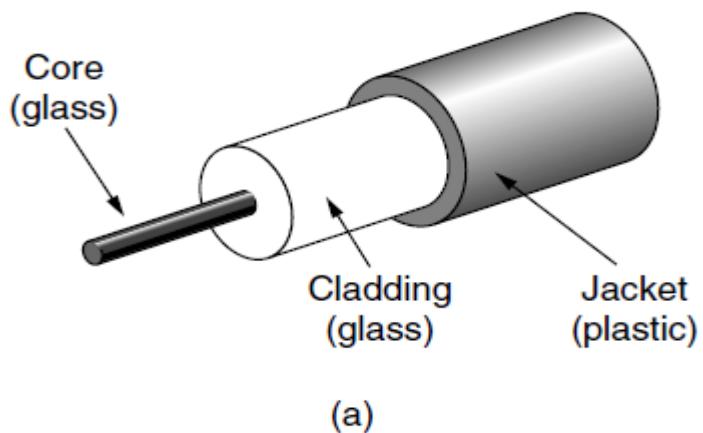
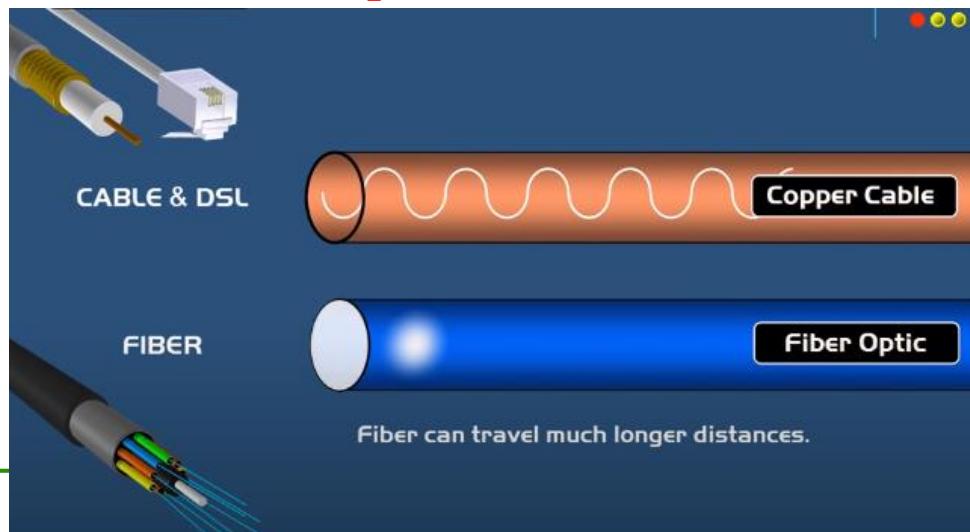


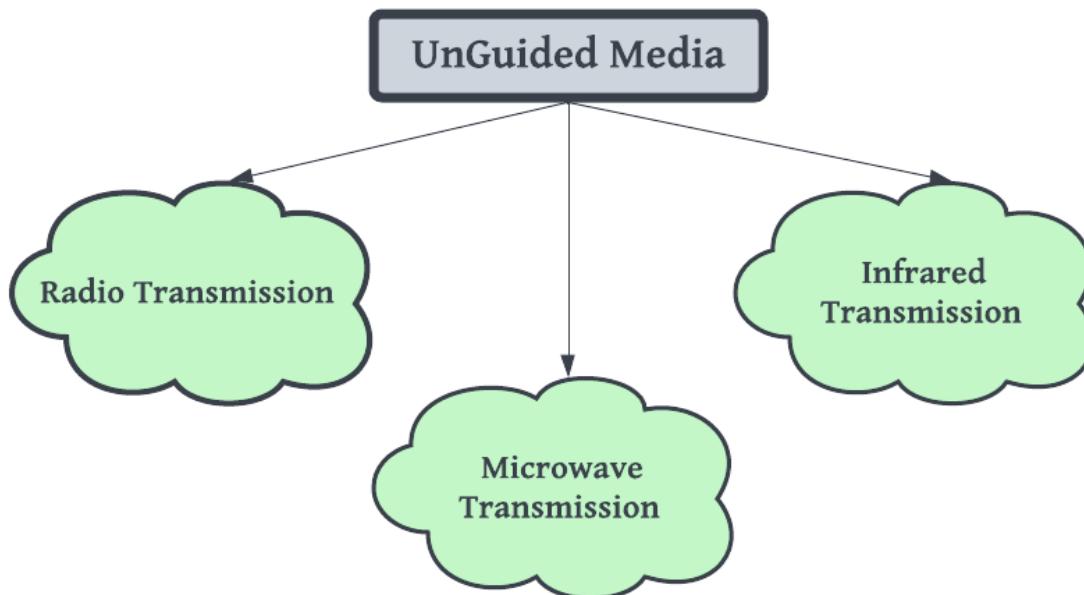
Figure 2-8. (a) Side view of a single fiber. (b) End view of a sheath with three fibers.

- The core is surrounded by a glass cladding with a lower index of refraction than the core, to keep all the light in the core.
- Next comes a thin plastic jacket to protect the cladding.
- Fibers are typically grouped in bundles, protected by an outer sheath. Figure 2-8(b)(Previous Slide) shows a sheath with three fibers.

Differences between cables and Fiber Optics



Wireless transmission medium



- Wireless transmission media, also known as unguided media, enable **the transmission of data without the use of physical cables.**
- These media use **electromagnetic waves or light** to carry signals through **the air or space.**
- Wireless signals are spread over in the air and are received and interpreted by **appropriate antennas.**
- When an antenna is attached to electrical circuit of a computer or wireless device, it converts the digital data into wireless signals and spread all over within its frequency range.

Features:

- ★ The signal is broadcasted through air
- ★ Used for larger distances
- ★ Unguided signals can travel in several ways:
 - **Ground propagation**
 - **Sky propagation**
 - **Line-of-sight propagation**

Radio Transmission

- Radio frequency (RF) waves are easy to generate, can travel long distances, and can penetrate buildings easily.
- So they are widely used for communication, both indoors and outdoors.
- Radio waves also are **omnidirectional**, meaning that they travel in all directions from the source
- Radio waves can have
 - **Wavelength from 1 mm – 1,00,000 km**
 - **Frequency ranging from 3 Hz (Extremely Low Frequency) to 300 GHz (Extremely High Frequency).**
- Lower frequencies such as **VLF(Very Low Frequency), LF(Low Frequency), MF(Medium Frequency) bands** can travel **on the ground up to 1000 kilometers, over the earth's surface.**

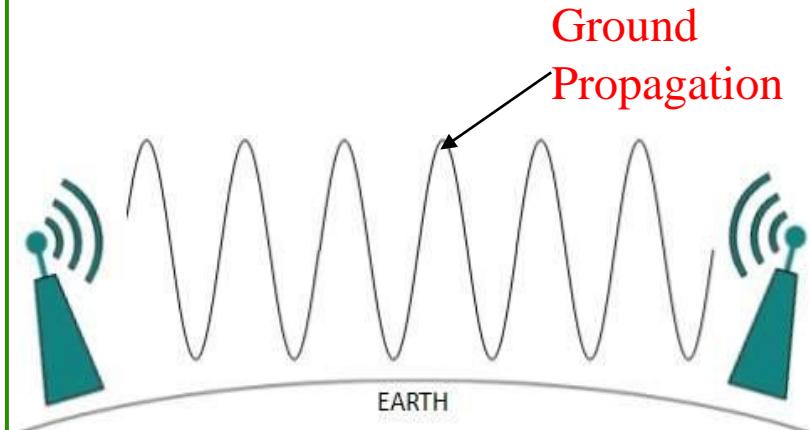


Fig: In the VLF, LF, MF bands, Radio waves follows the curvature of the Earth

Radio Transmission

- Radio waves of **high frequencies** are prone to be absorbed by rain and other obstacles.
- They use Ionosphere of earth atmosphere.
- High frequency radio waves such as **HF(High Frequency) and VHF(Very High Frequency) bands** are spread upwards.
- When they reach Ionosphere, they are refracted back to the earth.

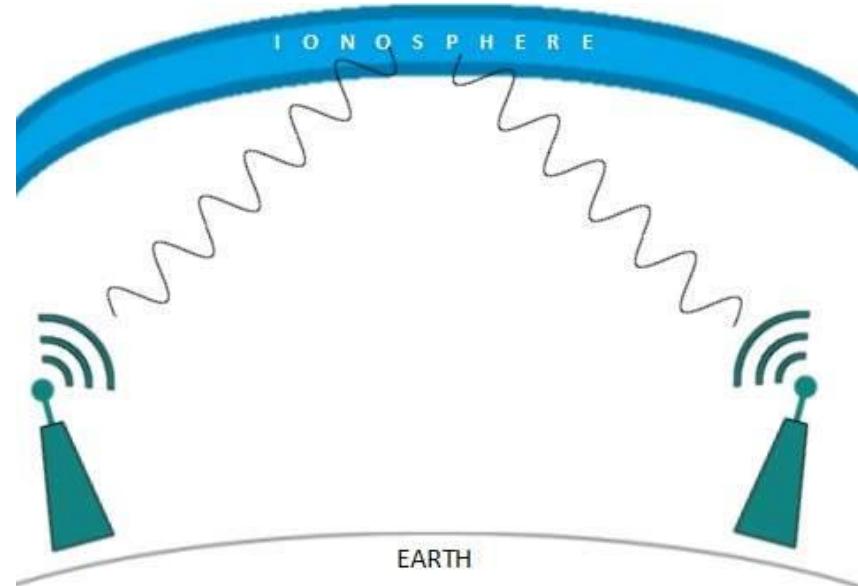


Fig: In HF,VHF bands, Radio Waves bounce off the Ionosphere

Advantages of Radio Waves

- Radio waves are **omnidirectional** (propagated in all directions).
- It can penetrate walls.

Radio Waves Uses:

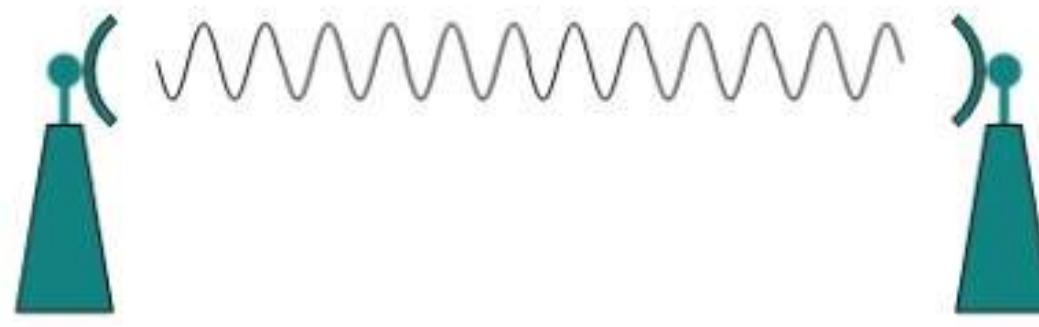
- FM radio
- Television
- Cellular Phones
- Wi-Fi



Omnidirectional

Microwave Transmission

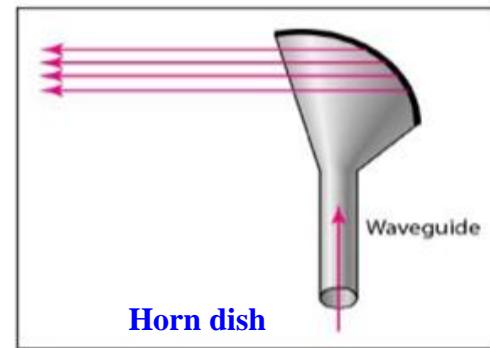
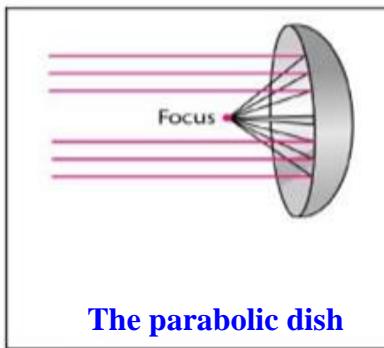
- Microwaves have **higher frequencies and shorter wavelengths** than **radio waves**.
- Microwaves are a form of electromagnetic radiation with wavelengths ranging from about **one millimeter to one meter (1mm to 1 m)**.
- Electromagnetic waves from frequencies between **1 GHz to 300 GHz** are called microwaves.



Unidirectional

Microwave Transmission

- Microwaves are **unidirectional**. The sending and receiving antennas need to be aligned.
- Microwaves need unidirectional antennas that send out signals in **one direction**.
- Two types of antennas are used for microwave communications:
 - The parabolic dish
 - The horn



- Microwave transmission uses **a line of sight propagation**.

Line-of-sight (LOS) transmission in the context of microwave communication refers to a direct and unobstructed path between the transmitting and receiving antennas.

For a successful line-of-sight communication, there should be a clear and straight line between the two antennas without any physical barriers or obstacles.

Advantages of Microwaves:

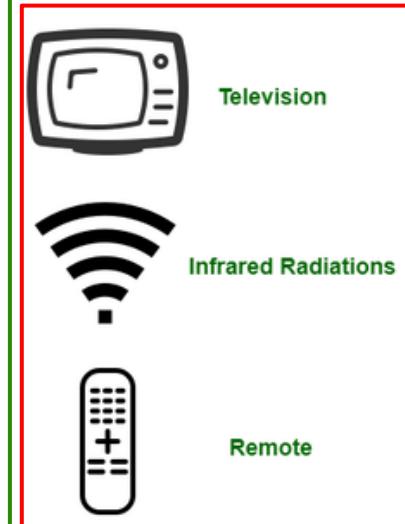
- Microwaves are unidirectional (sending and receiving antennas need to be aligned).
- Its propagation is line-of-sight (the sending and receiving antennas need to be properly aligned with each other.)

Example of Microwaves

- Cellular phones
- Satellite networks
- Wireless LAN

Infrared Transmission

- Infrared waves are used for **very short distance communication**.
- Infrared waves, **having high frequencies, cannot penetrate walls**.
- This advantageous characteristic **prevents interference** between one system and another; a short range communication system in one room cannot be affected by another system in the next room
- Frequency Range: 300 GHz – 400 THz.
- **It is used in TV remotes, wireless mouse, keyboard, printer, etc.**
- Infrared radiation (IR), is electromagnetic radiation (EMR) with longer wavelengths than those of visible light, and **invisible to the human eye**.



Advantages of Infrared

- Infrared waves is used for short distance communication having high frequencies.
- Cannot penetrate walls.

Examples of Infrared Waves

- Burning charcoal
- Heat from an electric heater

Applications of Infrared

- Infrared Data Association (IrDA) is used for communication between devices such as PCs, keyboards, mice, and printers. IrDA port allows wireless keyboard to communicate with a computer.

Mobile telephone system

Assignment Topic

UNIT-2

Data Link Layer

B SAI BABA,M.Tech(Ph.D),VIT,Bhimavaram

Syllabus

The Data Link Layer:

- Design issues
- Error detection and correction
- Elementary data link protocols
- Sliding window protocols
- HDLC
- The data link layer in the internet.

Syllabus

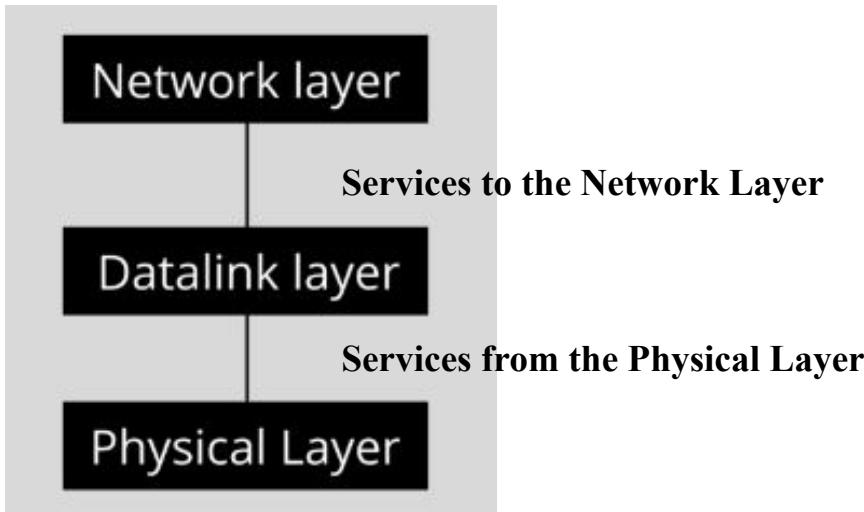
- The Media Access Sub Layer
 - Channel allocation problem
 - Multiple access protocols.

Data Link Layer Design Issues

- In the OSI model, the data link layer is a 6th layer from the top and 2nd layer from the bottom.
- The data link layer is responsible for maintaining the data link between two hosts or nodes.
- **The main functions and the design issues of this layer are**
 - **Providing Services to The Network Layer**
 - **Framing**
 - **Addressing**
 - **Error Control**
 - **Flow Control**
 - **Access Control**

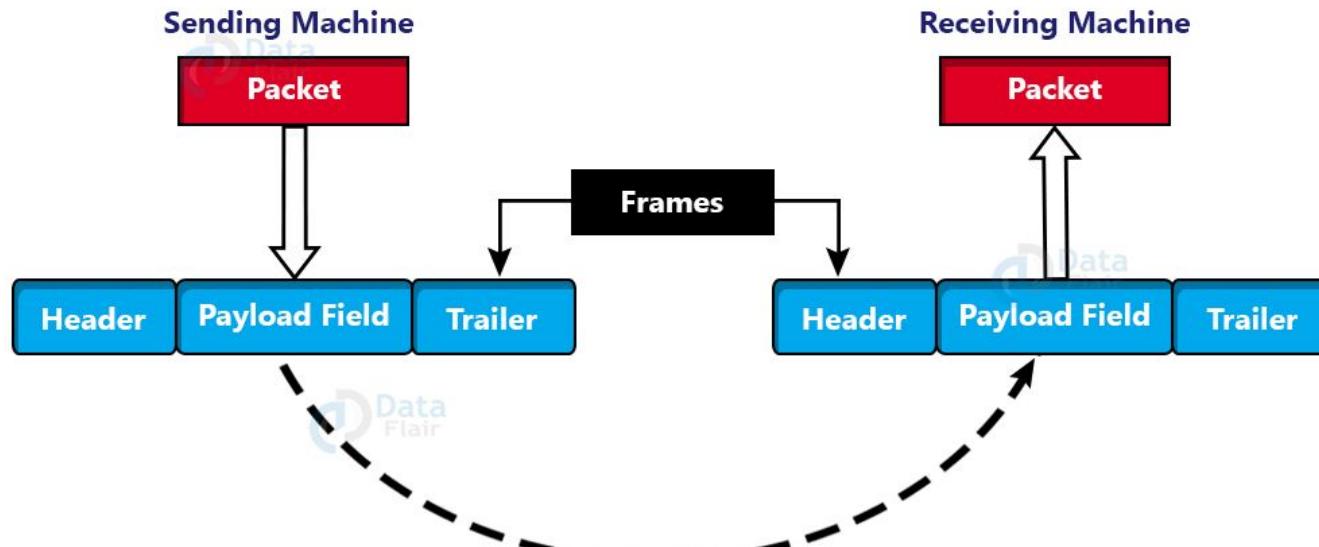
Providing Services to The Network Layer

- In the OSI Model, each layer uses the services of the layer below it and provides services to the layer above it.
- The data link layer uses the services offered by the physical layer.
- The primary function of this layer is to provide a well defined service interface to network layer above it.



Framing

- To provide service to the network layer, the data link layer must use the service provided to it by the physical layer.
- What the physical layer does is accept a raw bit stream and attempt to deliver it to the destination.
- Physical layers only just accept and transfer stream of bits without any regard to meaning or structure. Therefore it is up to data link layer to simply develop and recognize frame boundaries.
- Framing purpose is to divide the stream of data from the network layer into manageable frames that can be transmitted over the physical medium.
- The data link layer encapsulates Network Layer **Packets** into **Frames** by adding header and trailer information.



- 1. Frame Header:** It contains **the source and the destination addresses** of the frame and the control bytes.
- 2. Payload field :** It contains the **data packet from network layer**.
- 3. Trailer:** It contains the **error detection and error correction bits**. It is also called a **Frame Check Sequence (FCS)**

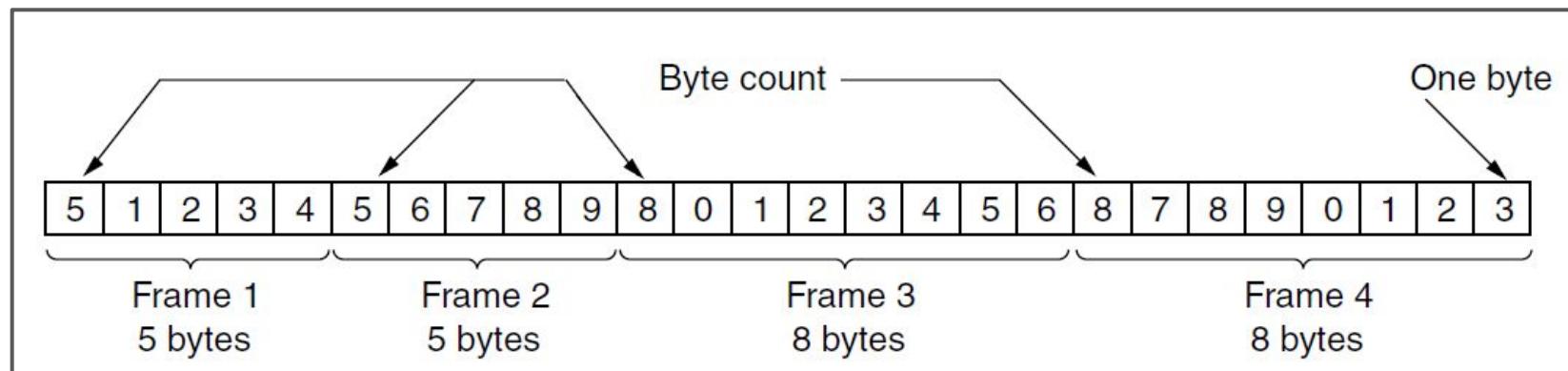
Framing Methods



1. Byte count
2. Flag bytes with byte stuffing
3. Flag bits with bit stuffing

1. Byte count

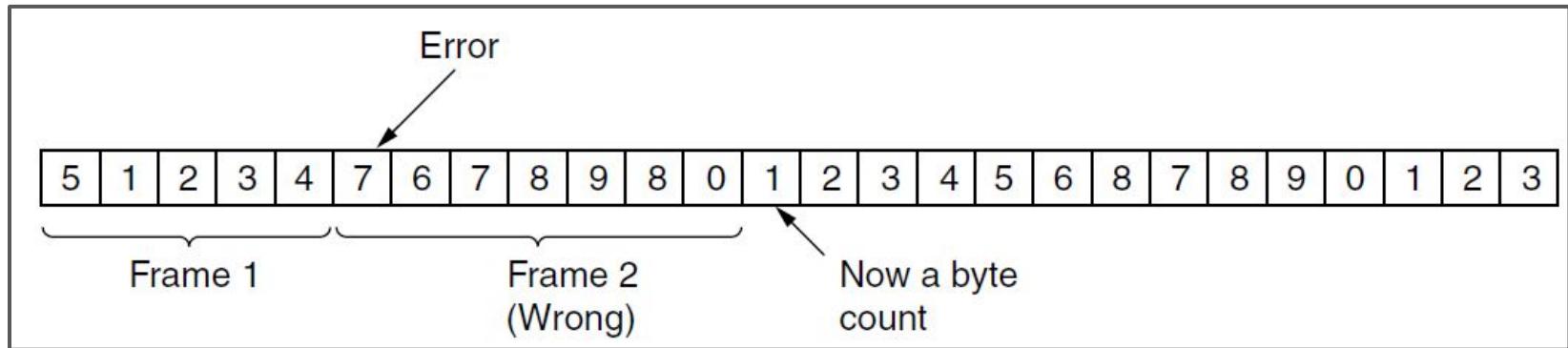
- This method is **rarely used** and is generally required to count total number of Bytes that are present in frame. This is be done by using field in header.
- Byte count method ensures data link layer at the receiver about total number of Bytes that follow, and about where the frame ends.



: Without errors:

1. Byte count:

: With errors:



- There is disadvantage also of using this method i.e., if anyhow Byte count is disturbed or distorted by an error occurring during transmission, then destination or receiver might lose synchronization.
- The destination or receiver might also be not able to locate or identify beginning of next frame.

2. Flag bytes with byte stuffing

- The second framing method gets around the problem of resynchronization after an error by having each frame start and end with special bytes.
- Often the same byte, called a flag byte, is used as both the starting and ending delimiter.
- This byte is shown in Fig. 3-4(a) as FLAG.
- Two consecutive flag bytes indicate the end of one frame and the start of the next. Thus, if the receiver ever loses synchronization it can just search for two flag bytes to find the end of the current frame and the start of the next frame.

FLAG	Header	Payload field	Trailer	FLAG
------	--------	---------------	---------	------

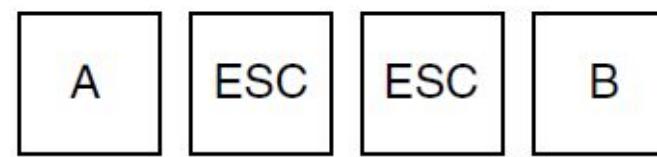
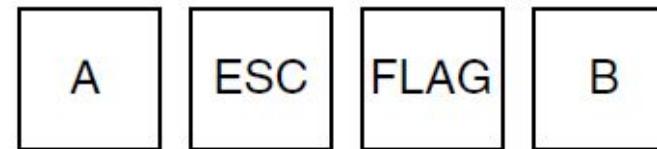
A Frame delimited by flag bytes

Four examples of byte sequences before and after byte stuffing

Original bytes



After stuffing



- It may happen that the **flag byte occurs in the data**. One way to solve this problem is to have the sender's data link layer insert a special **escape byte (ESC)** just before each “accidental” flag byte in the data.
- Thus, a framing flag byte can be distinguished from one in the data by the absence or presence of an escape byte before it.
- The data link layer on the receiving end removes the escape bytes before giving the data to the network layer. This technique is called **byte stuffing**.

3. Flag bits with bit stuffing

- Framing can be also be done at **the bit level**, so frames can contain an arbitrary number of bits made up of units of any size.
- It was developed for the once very popular **HDLC** (Highlevel Data Link Control) protocol.
- **Each frame begins and ends with a special bit pattern, **01111110****



A Frame delimited by a flag byte 01111110

- Whenever **the sender's data link layer encounters five consecutive 1s in the data**, it automatically stuffs a 0 bit into the outgoing bit stream.

(a) 0110111111111111110010

(b) 01101111011111011111010010

Stuffed bits

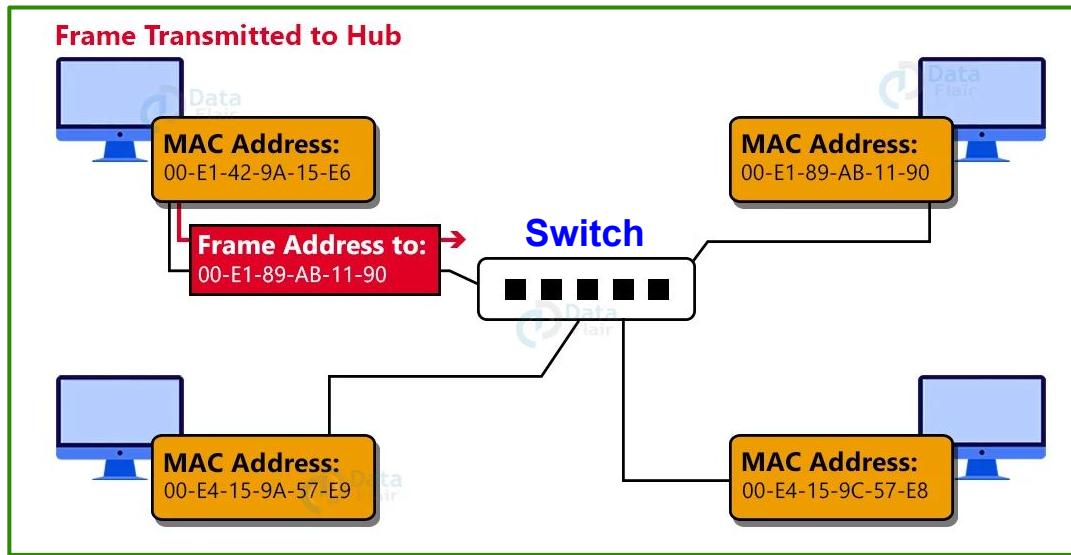
(c) 0110111111111111110010

Figure 3-5. Bit stuffing. (a) The original data. (b) The data as they appear on the line. (c) The data as they are stored in the receiver's memory after destuffing.

- When the receiver sees five consecutive incoming 1 bits, followed by a 0 bit, it automatically destuffs (i.e., deletes) the 0 bit.
 - Just as byte stuffing is completely transparent to the network layer in both computers, so is bit stuffing.
 - If the user data contain the flag pattern, 01111110, this flag is transmitted as 011111010 but stored in the receiver's memory as 01111110. Figure 3-5 gives an example of bit stuffing.

Data Link Layer Addressing

- In the data link layer of the network protocol stack, addressing is used to identify network nodes (devices) within a local network or a LAN (Local Area Network).
- There are **two main types of a** the data link layer:
 - **Media Access Control (MAC) Layer**
 - **Logical Link Control (LLC) Layer**



1. MAC Address (Media Access Control Address)

- MAC addresses are **unique identifiers** assigned to **Network Interface Cards (NICs)** at the factory.
- They consist of **6 bytes (48 bits)** and are usually represented in 12-digit hexadecimal notation (e.g., 00:1A:2B:3C:4D:5E).
- MAC addresses are globally unique, allowing each network device to have a unique identifier.
- The source and destination MAC addresses are included in the **frame header** of data frames at the data link layer.
- This allows devices to identify the intended recipient of the frame.

2. Logical Link Control Layer

- The LLC layer is responsible for **managing communication** between devices on the same network segment.
- It provides **a standardized interface** to the Network Layer (Layer 3) above it and the MAC layer below it.
- The LLC layer helps to ensure that upper-layer protocols can communicate efficiently over various types of data link technologies.

Key functions of the LLC layer include:

Error Control: Detecting and handling errors in the data link layer by using mechanisms like acknowledgments and retransmissions.

Flow Control: The Data Link Layer manages the flow of data between devices to prevent congestion and ensure efficient data transmission.

Link Management: Establishing, maintaining, and terminating logical links between devices.

Error Control

- The data link layer ensures error free link for data transmission. The issues it caters to with respect to error control are –
 - **Dealing with transmission errors**
 - **Sending acknowledgement frames in reliable connections**
 - **Retransmitting lost frames**
 - **Identifying duplicate frames and deleting them**
 - **Controlling access to shared channels in case of broadcasting.**

Flow Control

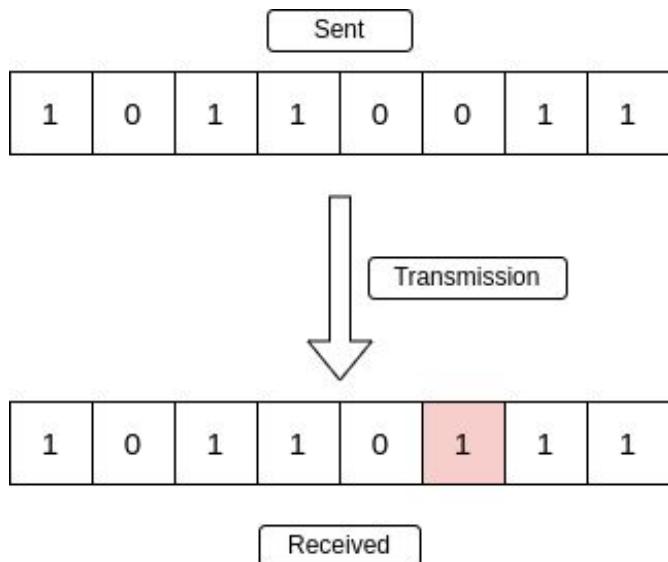
- The data link layer regulates flow control so that a fast sender does not drown a slow receiver.
- When the sender sends frames at very high speeds, a slow receiver may not be able to handle it. There will be frame losses even if the transmission is error-free.
- The two common approaches for flow control are –
 - **Feedback based flow control**
 - **Rate based flow control**

Error detection and correction

- **Error** is a condition **when the receiver's information does not match the sender's information.**
- During transmission, digital signals suffer from **noise** that can introduce errors in the binary bits traveling from sender to receiver. That means a 0 bit may change to 1 or a 1 bit may change to 0.
- Data (Implemented either at the Data link layer or Transport Layer of the OSI Model) may get scrambled by noise or get corrupted whenever a message is transmitted.
- To prevent such errors, **error-detection codes are added as extra data to digital messages.**
- This helps in detecting any errors that may have occurred during message transmission.
- Types of Errors
 - **Single Bit Error**
 - **Multiple Bit Error**
 - **Burst Error**

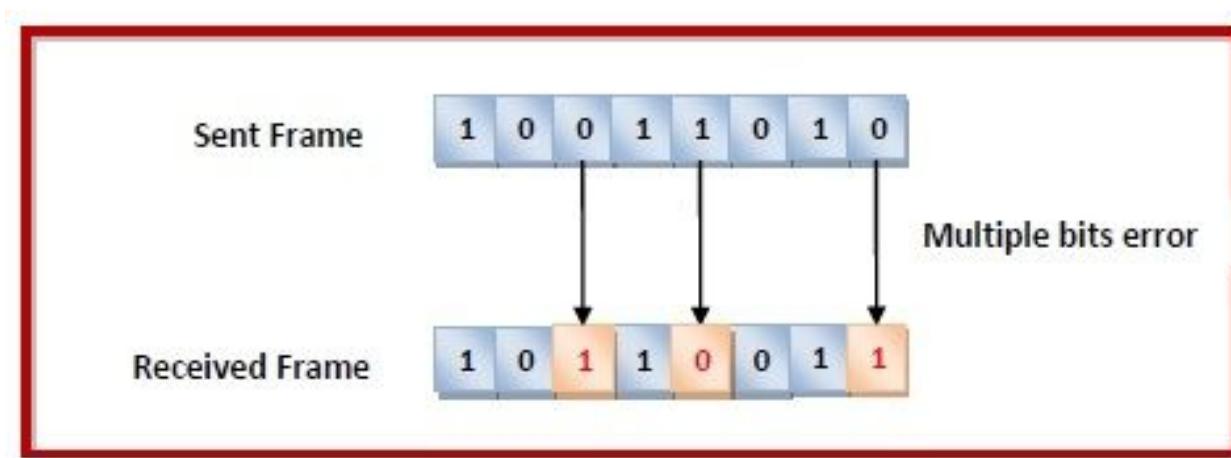
Single Bit Error

- A single-bit error refers to a type of data transmission error that occurs when **one bit** (i.e., a single binary digit) of a transmitted data unit is altered during transmission, resulting in an incorrect or corrupted data unit.



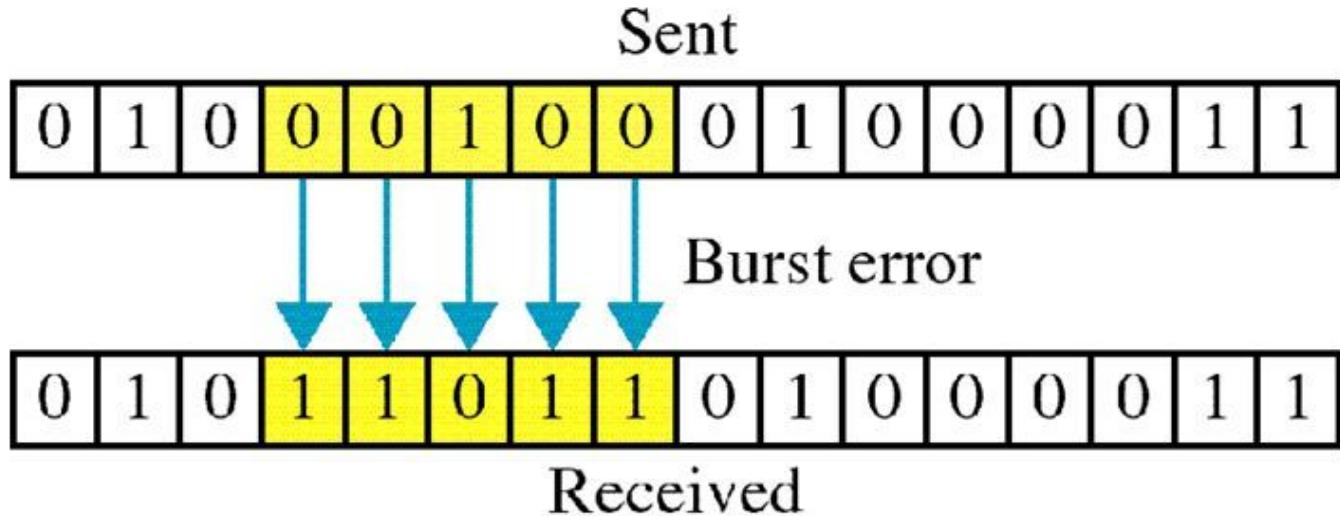
Multiple-Bit Error

- A multiple-bit error is an error type that arises when more than one bit in a data transmission is affected.
- Although multiple-bit errors are relatively rare when compared to single-bit errors, they can still occur, particularly in high-noise or high-interference digital environments.



Burst Error

- More than one consecutive bit is corrupted in the received frame.



Error Control

- Error control can be done in two ways
 - **Error detection** – Error detection involves checking whether any error has occurred or not. The number of error bits and the type of error does not matter.
 - **Error correction** – Error correction involves ascertaining the exact number of bits that has been corrupted and the location of the corrupted bits.

Error Detection Methods:

- ★ Parity
- ★ Checksums
- ★ Cyclic Redundancy Checks (CRCs)

Error Correction Methods:

- ★ Hamming codes
- ★ Binary convolutional codes
- ★ Reed-Solomon codes
- ★ Low-Density Parity Check codes

Error Detection Methods:

1. Parity Check:

- One extra bit is transmitted in addition to the original bits to make **the number of 1s even in the case of even parity or odd in the case of odd parity.**
- While creating a frame, the sender counts the number of 1s in it and adds the parity bit in the following way
 - **In case of even parity:** If a number of 1s is even then parity bit value is 0. If the number of 1s is odd then parity bit value is 1.
 - **In case of odd parity:** If a number of 1s is odd then parity bit value is 0. If a number of 1s is even then parity bit value is 1.

On receiving a frame, the receiver counts the number of 1s in it. In case of even parity check, if the count of 1s is even, the frame is accepted, otherwise, it is rejected. A similar rule is adopted for odd parity check.

Example of Simple Even Parity Check

SENDER

1 0 0 0 1 1

Compute parity bit

1 0 0 0 1 1 | 1

Transmission Media

RECEIVER

Reject Data

Even

N

Y

Accept Data

Compute parity bit

1 0 0 0 1 1 | 1



Disadvantage:

- Only single-bit error is detected by this method, **it fails in multi-bit error detection .**
- It can not detect an error in case of **an error in two bits.**

* **Two-Dimensional Parity Check:**

- For each row, parity check bits are calculated, which is identical to a basic parity check bit.
- For each column, parity check bits are computed and transmitted together with the data.
- These are compared with the parity bits calculated on the received data at the receiving end.

Two-Dimensional Parity Check

Original Data

10011001

11100010

00100100

10000100

Row Parities

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

Column Parities ➡

100110010

111000100

001001000

100001000

110110110

Data to be Sent

2. Checksum:

Checksum is a error detection which detects the error by dividing the data into the segments of equal size and then use 1's complement to find the sum of the segments and then sum is transmitted with the data to the receiver and same process is done by the receiver and at the receiver side, all zeros in the sum indicates the correctness of the data.

- First of all **data** is divided into **k segments** in a checksum error detection scheme and **each segment has m bits**.
- For finding out the sum at the sender's side, all segments are added through **1's complement** arithmetic. And for determining the checksum **we complement the sum**.
- Along with **data segments, the checksum segments** are also transferred.
- All the segments that are received on the receiver's side are added through 1's complement arithmetic to determine the sum. Then complement the sum also.
- The received data is **accepted only** on the condition that the **result is found to be 0**. And if the **result is not 0** then it will be **discarded**.

Original Data

10011001	11100010	00100100	10000100
1	2	3	4

$k=4, m=8$

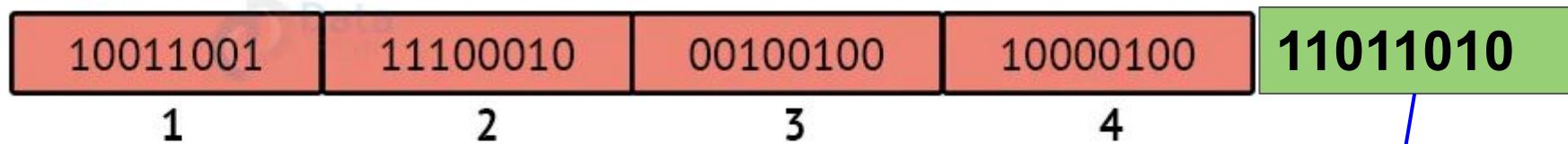
Original Data

10011001	11100010	00100100	10000100
1	2	3	4

k=4 , m=8

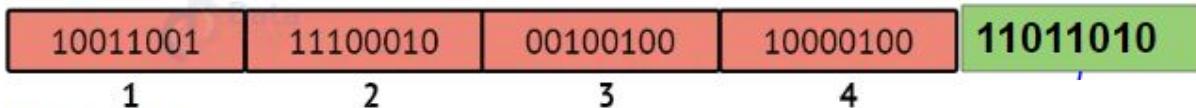
SENDER	
1	10011001
2	11100010
	<u>101111011</u>
	1
	<u>01111100</u>
3	00100100
	<u>10100000</u>
4	10000100
	<u>100100100</u>
	1
Sum: <u>00100101</u>	
CheckSum: 11011010	

Original Data



$k=4, m=8$

Check Sum

**RECIEVER**

1 10011001
 2 11100010

 101111011
 |
 01111100
 3 00100100

 10100000
 4 10000100

 100100100
 |
 00100101
 11011010

Sum: 11111111

Complement: 00000000

Conclusion: Accept Data

Eg: 10110011 10101011 01011010 11010101, K=4 & m=8

Sender Side:

Sum = ?

CheckSum = ?

Receiver Side:

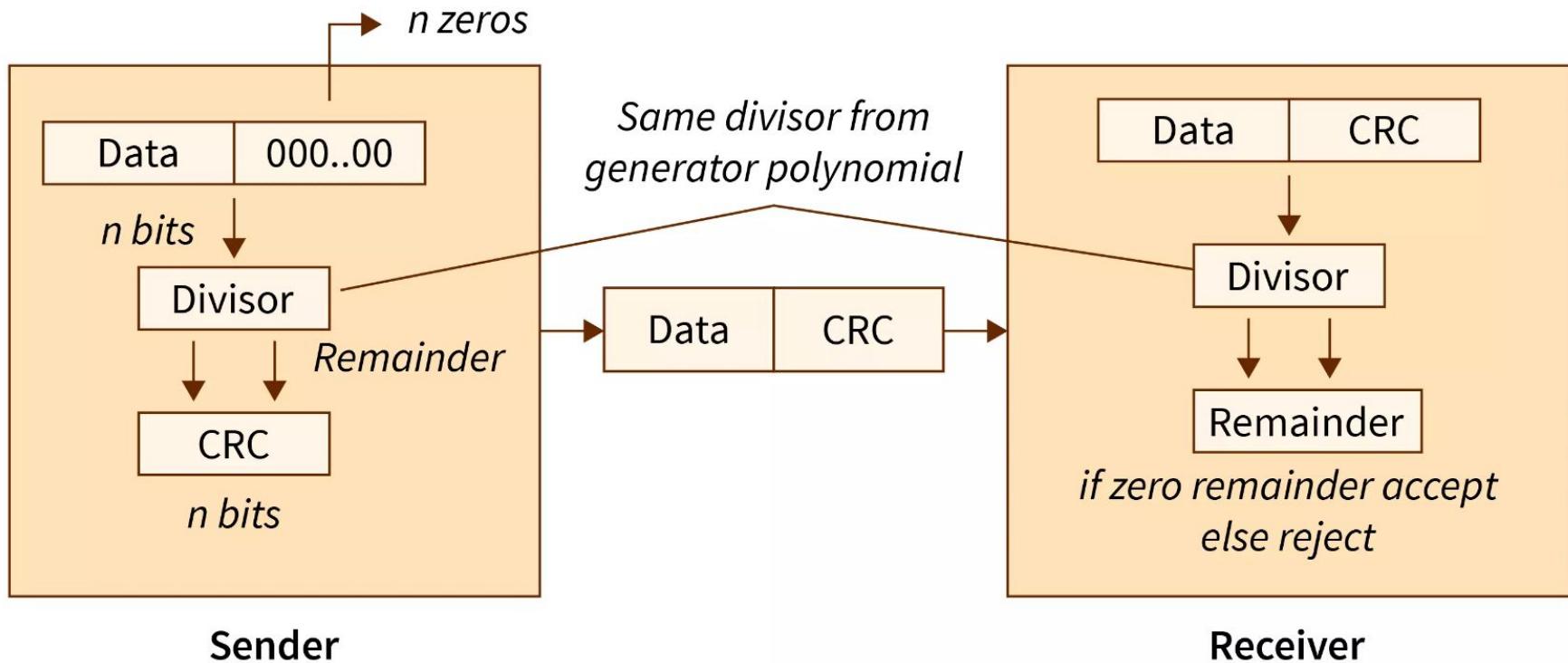
Sum = ?

CheckSum = ?

3. Cyclic Redundancy Checks (CRCs) or polynomial code:

- Unlike the checksum scheme, which is based on addition, **CRC is based on binary division.**
- In CRC, a sequence of redundant bits, called cyclic redundancy check bits, are appended to the end of the data unit so that the resulting data unit becomes exactly divisible by a second, predetermined binary number.
- **The sender** divides the bits that are being transferred and calculates the remainder.
- **The sender inserts the remainder at the end of the original bits before sending the actual bits.**
- **A codeword** is made up of the **actual data bits plus the remainder**. The transmitter sends data bits in the form of codewords.
- The receiver, on the other hand, divides the codewords using the same CRC divisor.
- **If the remainder consists entirely of zeros**, the data bits are **validated**; otherwise, it is assumed that some data corruption happened during transmission.

Cyclic Redundancy Checks



Eg:

A bit stream 1010000 is transmitted using the standard CRC method. The generator polynomial is x^3+1 . What is the actual bit string transmitted? Conclude whether the receiver receive original message or error message.

original message
1010000

@ means X-OR

Sender

$$\begin{array}{r}
 1001 \boxed{101000000} \\
 @1001 \\
 \hline
 0011000000 \\
 @1001 \\
 \hline
 01010000 \\
 @1001 \\
 \hline
 0011000 \\
 @1001 \\
 \hline
 01010 \\
 @1001 \\
 \hline
 0011
 \end{array}$$

Message to be transmitted
 $101000000 + 011$
 $\hline 101000011$

Generator polynomial
 x^3+1
 $1.x^3+0.x^2+0.x^1+1.x^0$
 CRC generator
1001 4-bit

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

$$\begin{array}{r}
 1001 \boxed{101000011} \\
 @1001 \\
 \hline
 001100011 \\
 @1001 \\
 \hline
 01010011 \\
 @1001 \\
 \hline
 0011011 \\
 @1001 \\
 \hline
 01001 \\
 @1001 \\
 \hline
 0000
 \end{array}$$

Receiver

Zero means data is accepted

Examples:

1. A bit stream **1101011011** is transmitted using the standard CRC method. The generator polynomial is x^4+x+1 . What is the actual bit string transmitted?

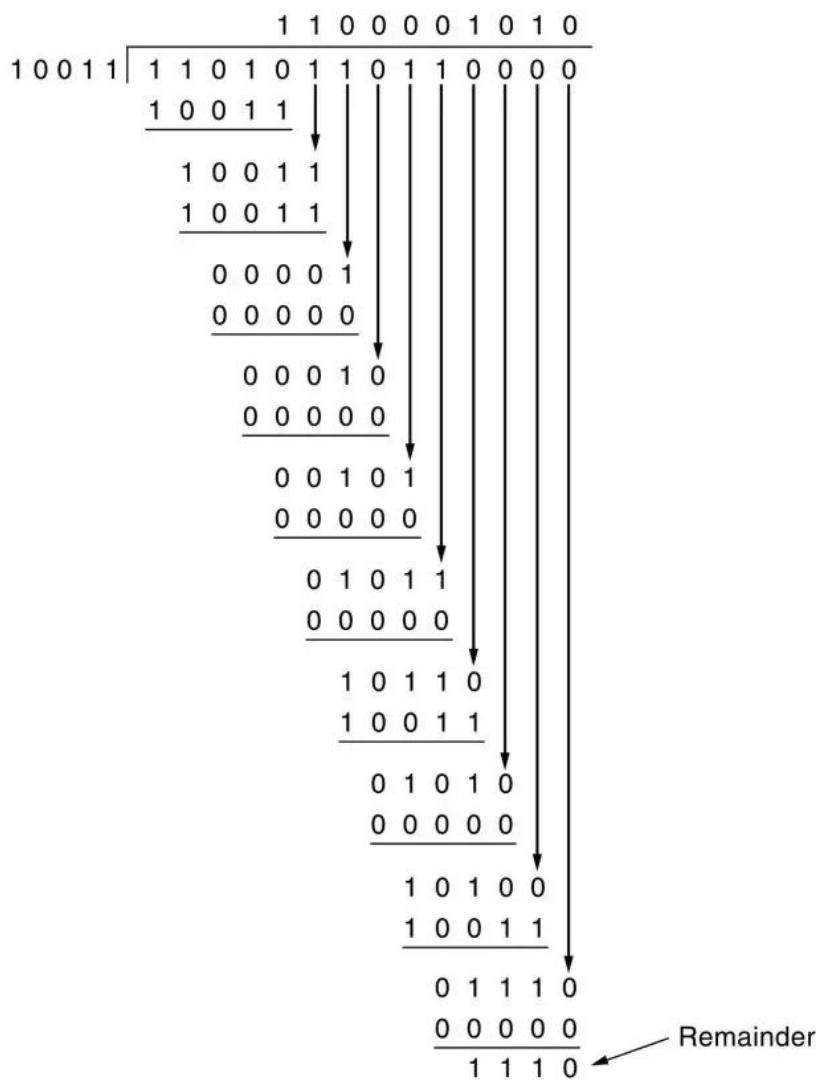
2. What is the remainder obtained by dividing $x^7 + x^5 + 1$ by the generator polynomial $x^3 + 1$?

Examples:

1. A bit stream 1101011011 is transmitted using the standard CRC method. The generator polynomial is $x^4 + x + 1$. What is the actual bit string transmitted?

Solution:

- The generator polynomial $G(x) = x^4 + x + 1$ is encoded as **10011**.
- Clearly, the generator polynomial consists of **5 bits**.
- So, a string of **4 zeroes** is appended to the bit stream to be transmitted.
- The resulting bit stream is **11010110110000**.



From here, CRC = **1110**.

Now,

- The code word to be transmitted is obtained by replacing the last 4 zeroes of 11010110110000 with the CRC.
- Thus, the code word transmitted to the receiver = **11010110111110**.

Error Correction Methods

- ★ Hamming codes
- ★ Binary convolutional codes
- ★ Reed-Solomon codes
- ★ Low-Density Parity Check codes

★ Error Correction codes are used to detect and correct errors that occur during data transmission from the transmitter to the receiver.

★ There are two approaches to error correction:

1. Backward Error Correction:

When a backward mistake is detected, the receiver requests that the sender **retransmit** the complete data unit.

2. Forward Error Correction:

In this scenario, the error-correcting code is used by the receiver, which automatically corrects the mistakes.

★ A single extra bit can identify but **not correct the errors.**

★ To correct the mistakes, the specific location of the error must be known.

★ If we wish to compute a single-bit mistake, for example, the **error correcting algorithm** will identify which one of seven bits is incorrect. We will need to add some more redundant bits to do this.

★ The number of redundant bits is calculated using the following formula: $2^r >= d+r+1$

★ The above formula is used to compute the value of r. For example, if the value of d is 4, the least possible number that fulfils the above relation is 3.

Hamming codes

- ★ It is a block code that is capable of **detecting up to two simultaneous bit errors and correcting single-bit errors.**
- ★ **Parity bits:** A bit that is added to the original binary data to make sure the total number of 1s is even or odd (in case of even or odd parity respectively).
 - **Even parity:** To check for even parity, if the total number of 1s is even, the parity bit value is 0. If the total number of occurrences of 1s is odd, the parity bit value is 1.
 - **Odd Parity:** To test for odd parity, if the total number of 1s is even, the parity bit value is 1. If the total number of 1s is odd, the parity bit value is 0.

Hamming codes

Algorithm of Hamming code:

1. An information of 'd' bits are added to the redundant bits 'r' to form $d+r$.
2. The location of each of the $(d+r)$ digits is assigned a decimal value.
3. The 'r' bits are placed in the positions $1, 2, \dots, 2k-1$.
4. At the receiving end, the parity bits are recalculated. The decimal value of the parity bits determines the position of an error.

Relationship b/w Error position & binary number:

Error Position	Binary Number
0	000
1	001
2	010
3	011
4	100
5	101
6	110
7	111

Let's understand the concept of Hamming code through an example:

- Suppose the original data is **1010** which is to be sent.

Total number of data bits 'd' = 4

Number of redundant bits r : $2^r \geq d+r+1$

$$2^r \geq 4+r+1$$

Therefore, the value of r is 3 that satisfies the above relation.

Total number of bits = $d+r = 4+3 = 7$;

- Determining the position of the redundant bits

- The number of redundant bits is 3. The three bits are represented by r_1, r_2, r_4 .
- The position of the redundant bits is calculated with corresponds to the raised **power of 2**.

Therefore, their corresponding positions are $1, 2^1, 2^2$.

The position of $r_1 = 1$

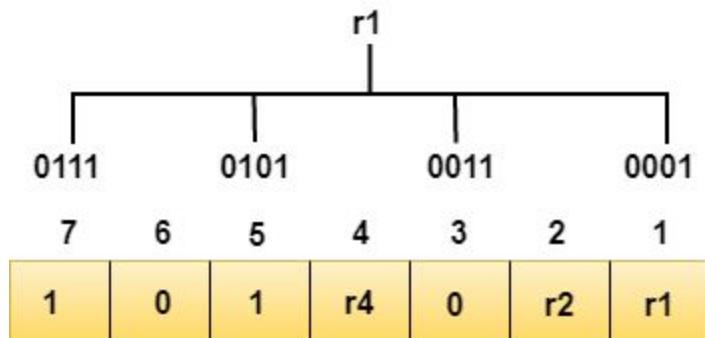
The position of $r_2 = 2$

The position of $r_4 = 4$

Determining the Parity bits

- Determining the r1 bit

- The **r1** bit is calculated by performing a parity check on the bit positions whose binary representation includes 1 in the first position.

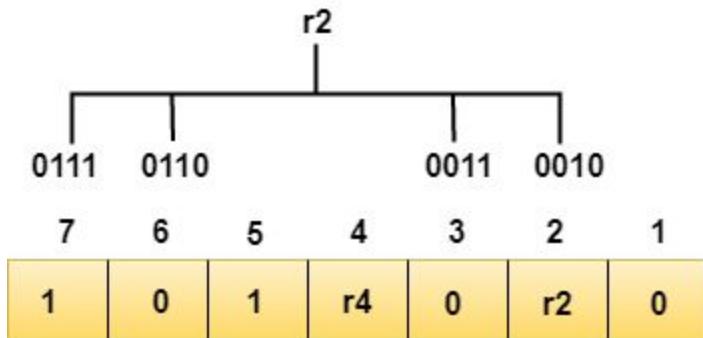


- We observe from the above figure that the bit positions that includes 1 in the first position are 1, 3, 5, 7. Now, we perform the even-parity check at these bit positions.
- The total number of 1 at these bit positions corresponding to r1 is even, therefore, **the value of the r1 bit is 0.**

Determining the Parity bits

- Determining the r2 bit

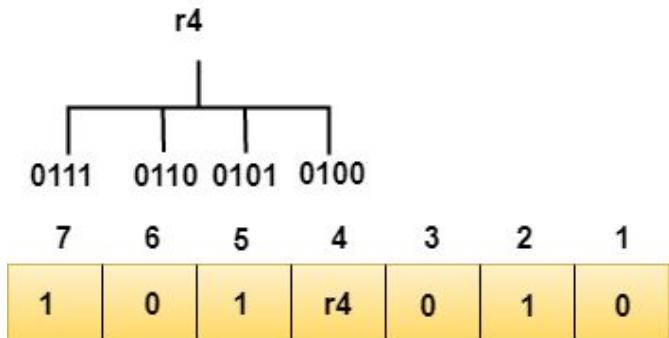
- The **r2** bit is calculated by performing a parity check on the bit positions whose binary representation includes 1 in the second position.



- We observe from the above figure that the bit positions that includes 1 in the second position are 2, 3, 6, 7. Now, we perform the even-parity check at these bit positions.
- The total number of 1 at these bit positions corresponding to r2 is odd, therefore, **the value of the r2 bit is 1.**

Determining the r4 bit

- The r4 bit is calculated by performing a parity check on the bit positions whose binary representation includes 1 in the third position.



- We observe from the above figure that the bit positions that include 1 in the third position are 4, 5, 6, 7. Now, we perform the even-parity check at these bit positions.
- The total number of 1 at these bit positions corresponding to r4 is even, therefore, **the value of the r4 bit is 0**.

* Data transferred is given below:

7	6	5	4	3	2	1
1	0	1	0	0	1	0

Example: If the data to be transmitted is **1011001**

Flow Control

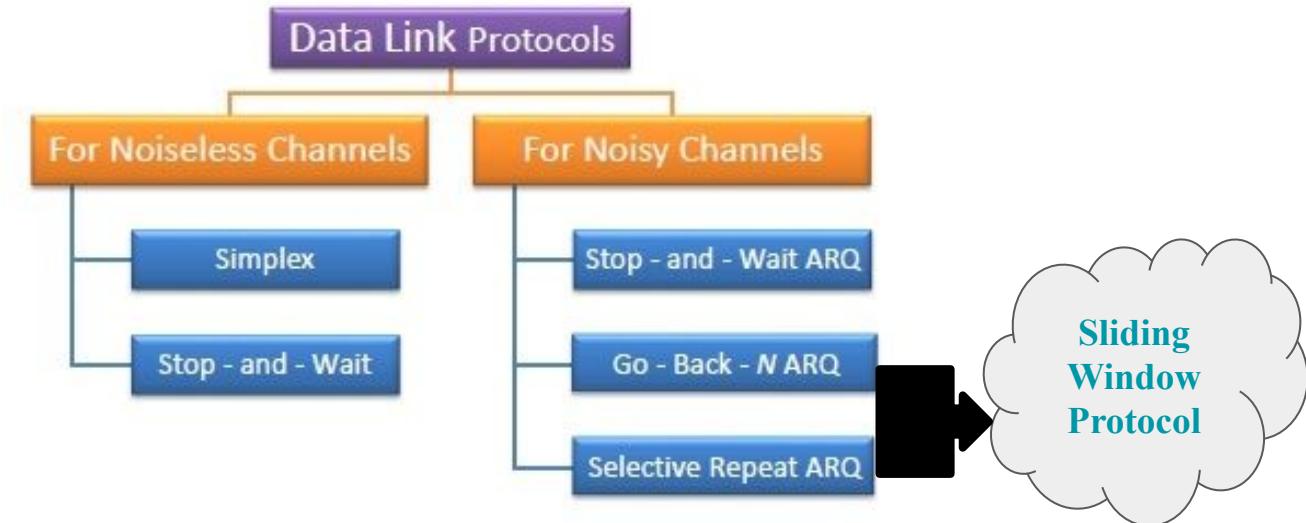
Elementary data link protocols

Flow control

- ★ Flow control is a **Speed Matching Mechanism**.
- ★ Flow control in the data link layer is a mechanism used **to manage the rate of data transmission between the sender and the receiver**.
- ★ It ensures that the sender **does not overwhelm** the receiver with more data than it can handle, **preventing data loss or buffer overflow**.
- ★ Flow control is set of procedures that tells the sender how much data it can transmit before it must wait for an acknowledgement from the receiver.
- ★ Receiver must inform the sender before the limits are reached and request that the sender to send fewer frames or stop temporarily.
- ★ There are different **protocols** involved in Flow Control.

What is **Protocol?**

- ★ In computer networks, a protocol refers to **a set of rules and procedures** that govern the communication and interaction between devices or systems.
- ★ It defines how data is transmitted, formatted, addressed, routed, and processed in a network.
- ★ Protocols ensure that devices can understand and interpret the information exchanged, allowing for reliable and standardized communication.



Stop and wait Protocol

- ★ It is a basic protocol used for reliable data transmission between a sender and a receiver when there is **no noise or errors** in the channel.

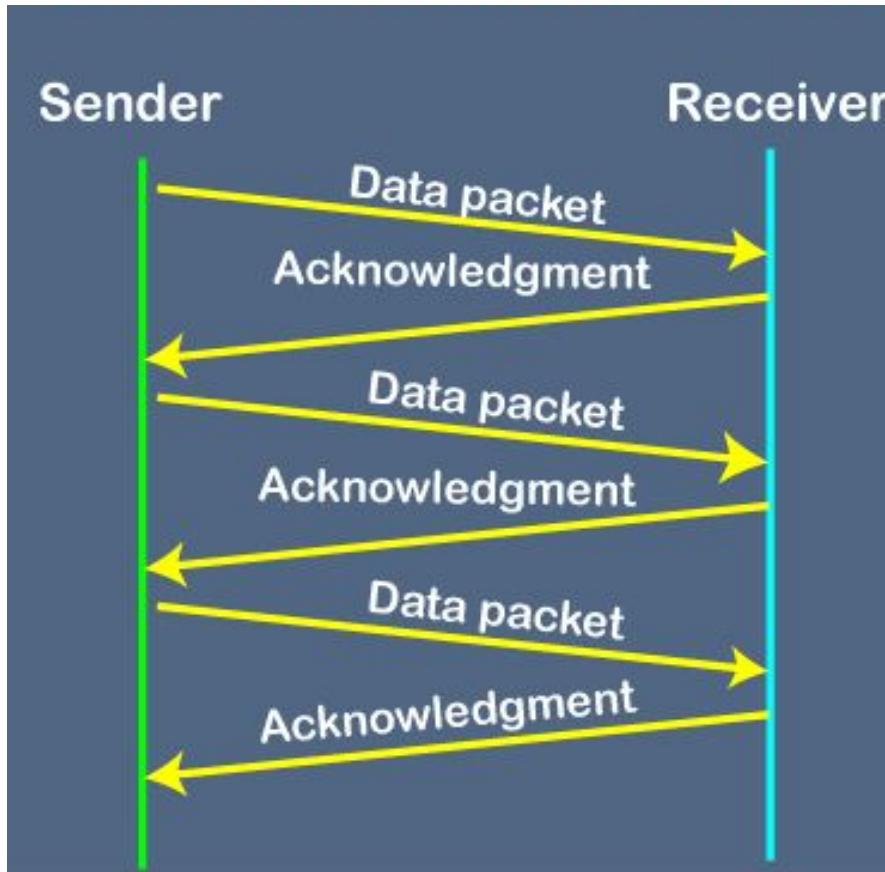
Sender:

- The sender sends a single frame to the receiver.
- After sending the frame, it waits for an acknowledgment (ACK) from the receiver.
- If the sender receives the ACK, it assumes that the frame was successfully received by the receiver and proceeds to send the next frame.
- If the sender does not receive the ACK within a specified time, it assumes that the frame was lost and retransmits the same frame.

Receiver:

- The receiver receives a frame from the sender.
- If the frame is error-free, the receiver sends an ACK back to the sender indicating successful reception.
- If the frame contains errors, the receiver discards the frame and does not send an ACK.
- The receiver waits for the sender to retransmit the frame if it does not receive a valid frame.

Flow diagram for Stop & Wait Protocol



Problems of Stop and Wait Protocol

1. Problems due to Lost Data.

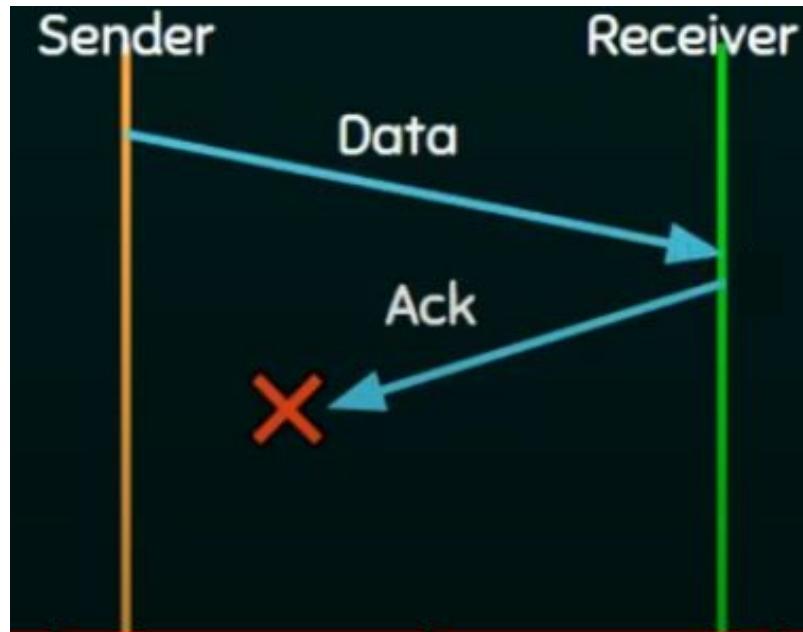
- a. **Sender** waits for **Ack** for an infinite amount of time.
- b. **Receiver** waits for **Data** for an infinite amount of time.



Problems of Stop and Wait Protocol

2. Problems due to Lost Ack.

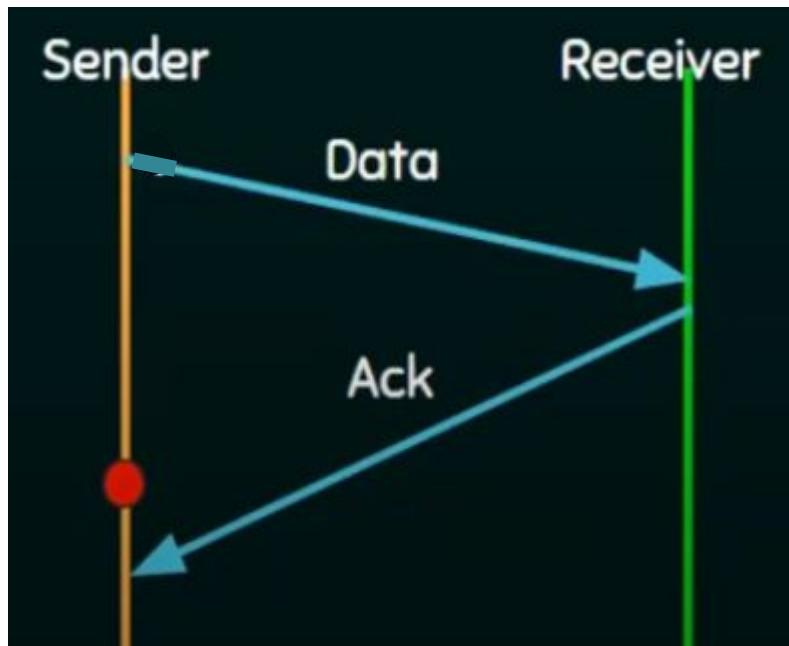
- a. Sender waits for an infinite amount of time for Ack.

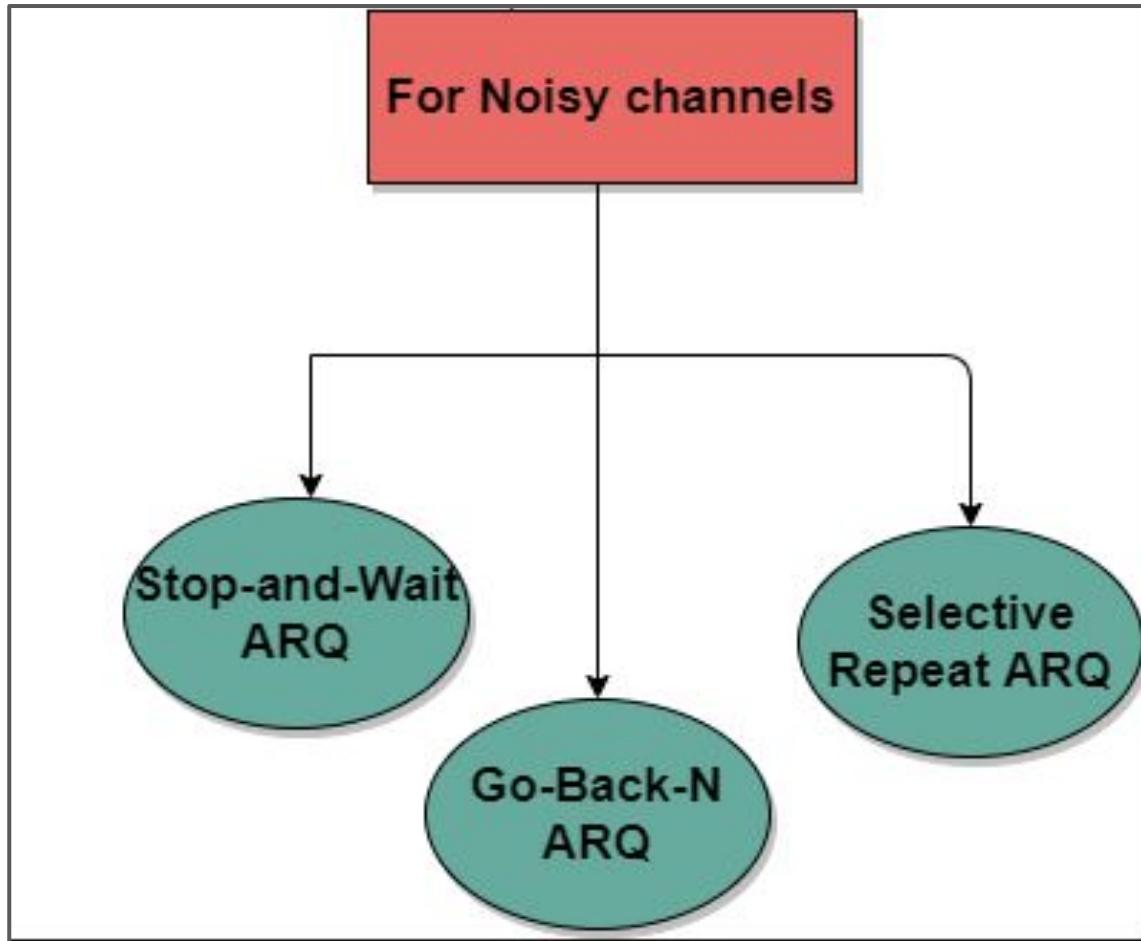


Problems of Stop and Wait Protocol

3. Problems due to Delayed Ack / Data.

- a. After **timeout** on sender side, a **delayed Ack** might be wrongly considered as an Ack of some other data packet.





Stop and Wait ARQ [Stop-and-Wait Automatic Repeat Request]

- ★ Stop-and-Wait ARQ is a specific variation of the Stop-and-Wait protocol that incorporates error detection and retransmission mechanisms to ensure reliable data transmission in the presence of **errors or noise in the channel**.

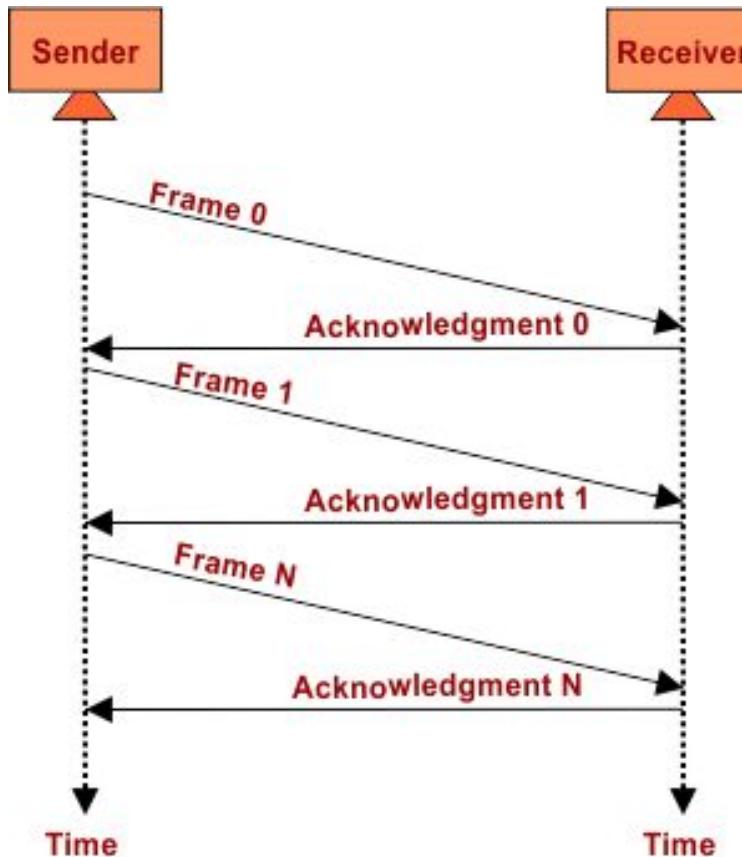
Sender:

- The sender sends a frame to the receiver. After sending the frame, it starts a timer.
- If the sender receives an acknowledgment (ACK) from the receiver within the timeout period, it assumes that the frame was successfully received and proceeds to send the next frame.
- If the timer expires before receiving the ACK, the sender assumes that the frame was lost or damaged and retransmits the same frame.

Receiver:

- The receiver receives a frame from the sender.
- If the frame contains errors or is damaged, the receiver discards the frame and does not send an ACK.
- If the frame is error-free and successfully received, the receiver sends an ACK back to the sender indicating successful reception.
- The receiver waits for the sender to retransmit the frame if it does not receive a valid frame.

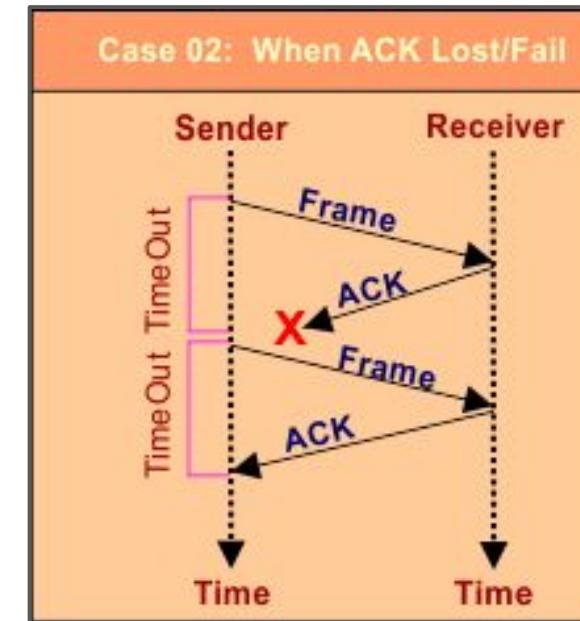
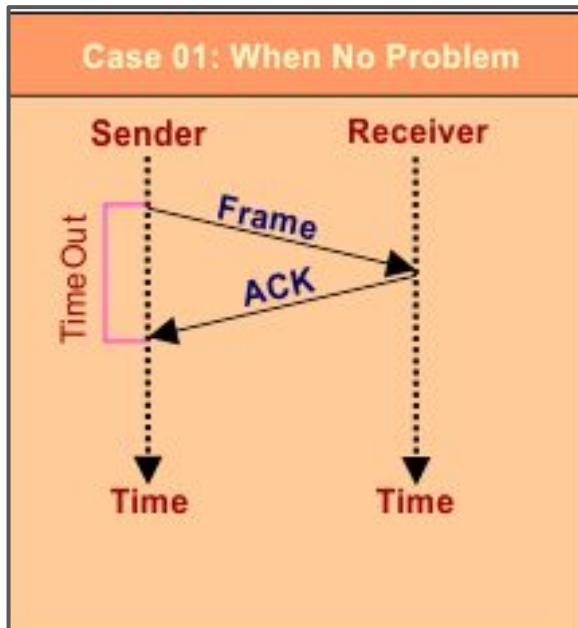
- Stop and wait ARQ = Stop and wait + Timeout Timer + Sequence Number



All possible scenarios of this protocol are explained under

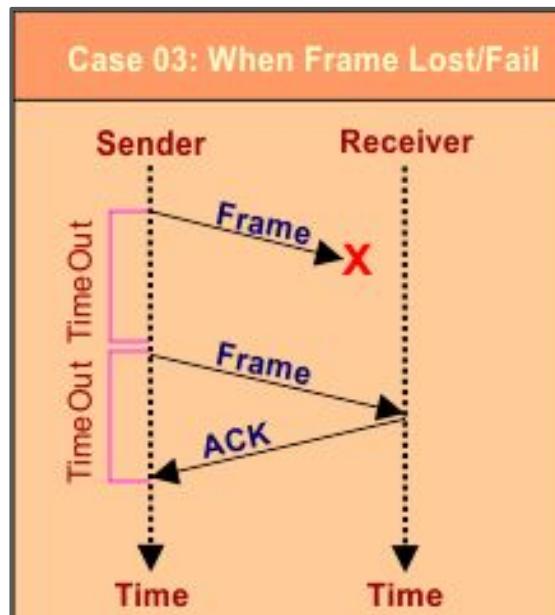
[1] The Ack is received before the timer Expires

[2] The Ack is lost

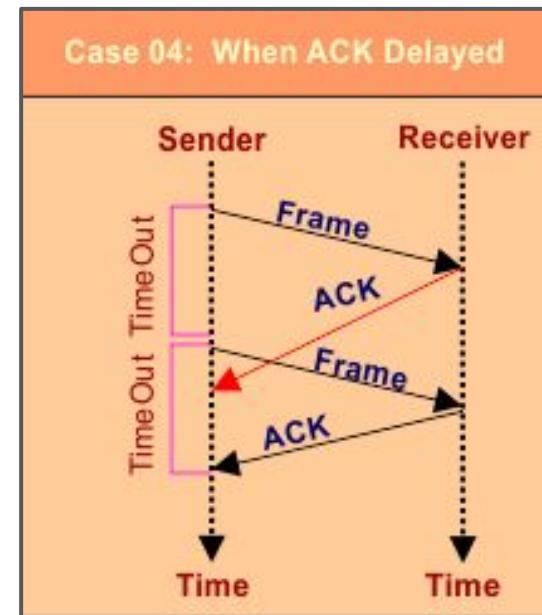


All possible scenarios of this protocol are explain under

[3] When Frame Lost



[4] The Ack is Delayed



Sliding Window Protocol

- ★ Go-Back-N ARQ
- ★ Selective Repeat ARQ



Sliding Window Protocol

- ★ The Sliding Window Protocol is a flow control protocol used in data communication to allow **efficient and reliable transmission of data between a sender and a receiver**.
- ★ It is an **extension of the Stop-and-Wait protocol** and overcomes its limitations by allowing **multiple frames** to be in transit **without waiting for individual acknowledgments**.
- ★ The number of frames to be sent based on the **Window Size**.

★ Sender:

- The sender maintains a "**window**" that represents **the range of acceptable sequence numbers for the frames it can send**.
- The sender can transmit **multiple frames** within the window **without waiting for acknowledgments**.
- As frames are sent, the sender **slides the window forward**.
- The sender starts **a timer** for the first frame in the window.

Sliding Window Protocol

★ Receiver:

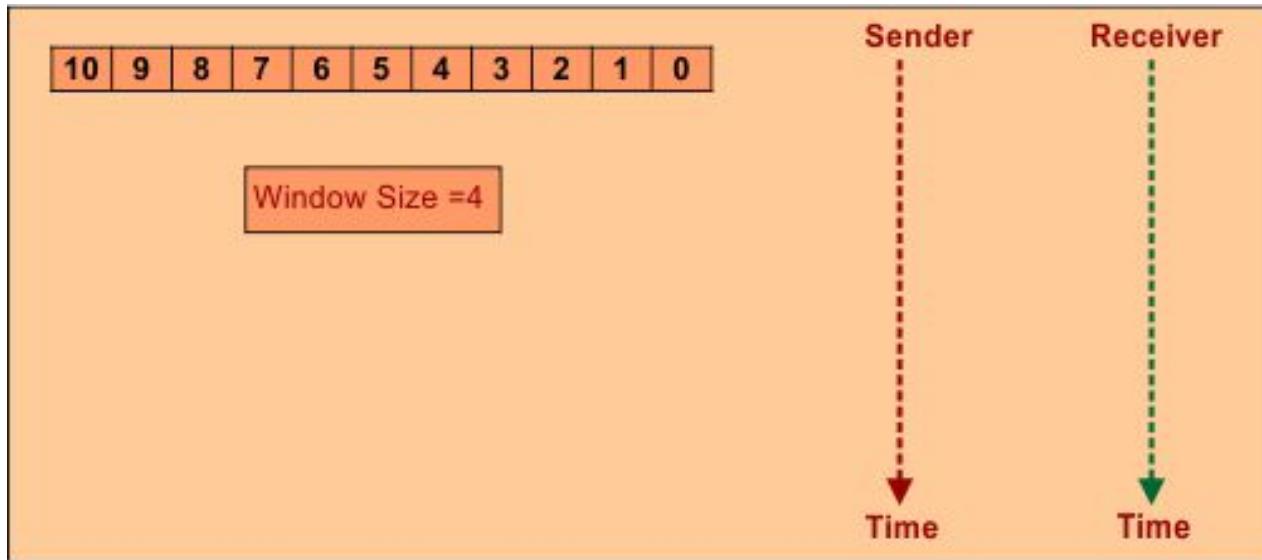
- The receiver maintains a corresponding "window" that represents the range of acceptable sequence numbers for the frames it expects to receive.
- The receiver acknowledges the frames it successfully receives, indicating the next expected sequence number.
- If a frame is received out of order or contains errors, the receiver discards the frame and does not send an acknowledgment.
- The receiver can accept frames within its window and slides the window forward as it receives valid frames.

Working of Sliding Window Protocol

Example :

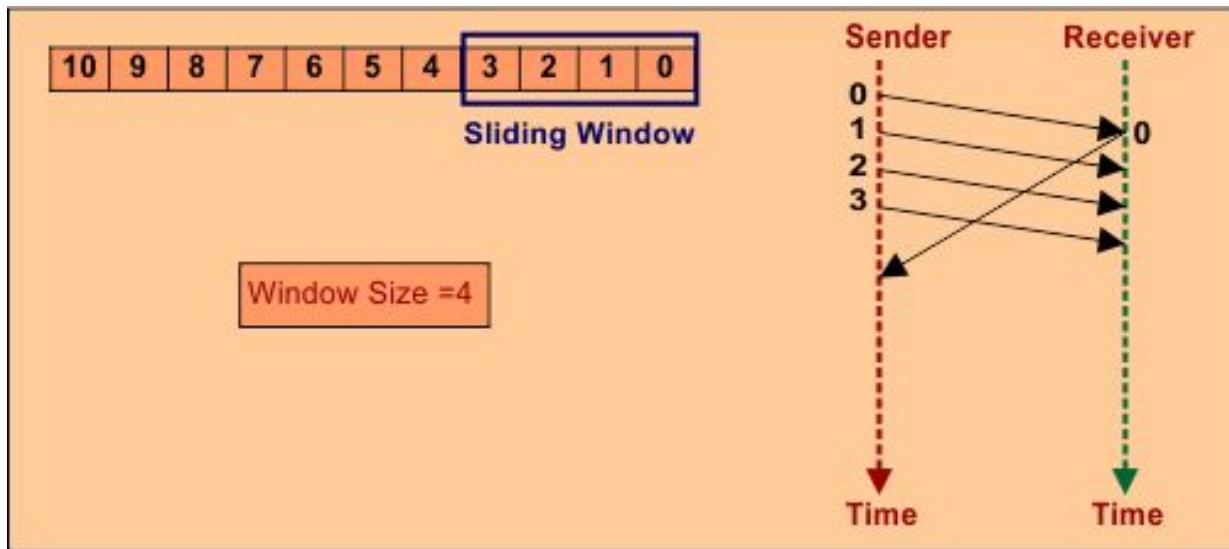
For example, suppose there are 11 frames required transmitting and sender window size is 4 then the sequence number will be 0,1,2,3, 0,1,2,3, 0, 1 and 2.

Step 1: 11 frames (0-10), window size, sender and receiver are shown below



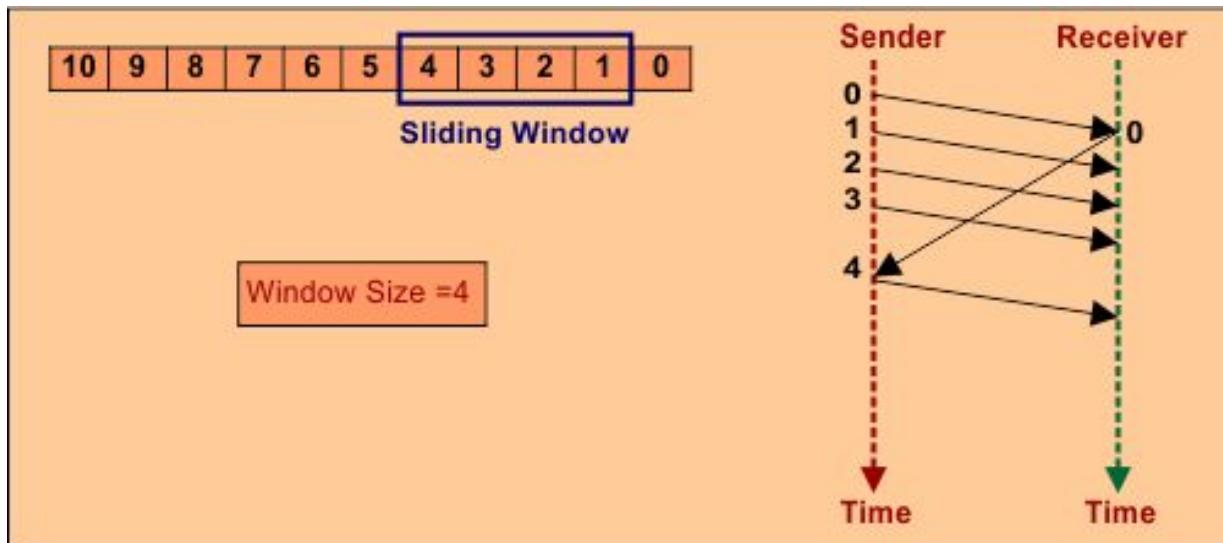
Working of Sliding Window Protocol

Step 2: After sending all frames equal to window size, Sender wait for ACK from receiver of first frame (i.e. 0).



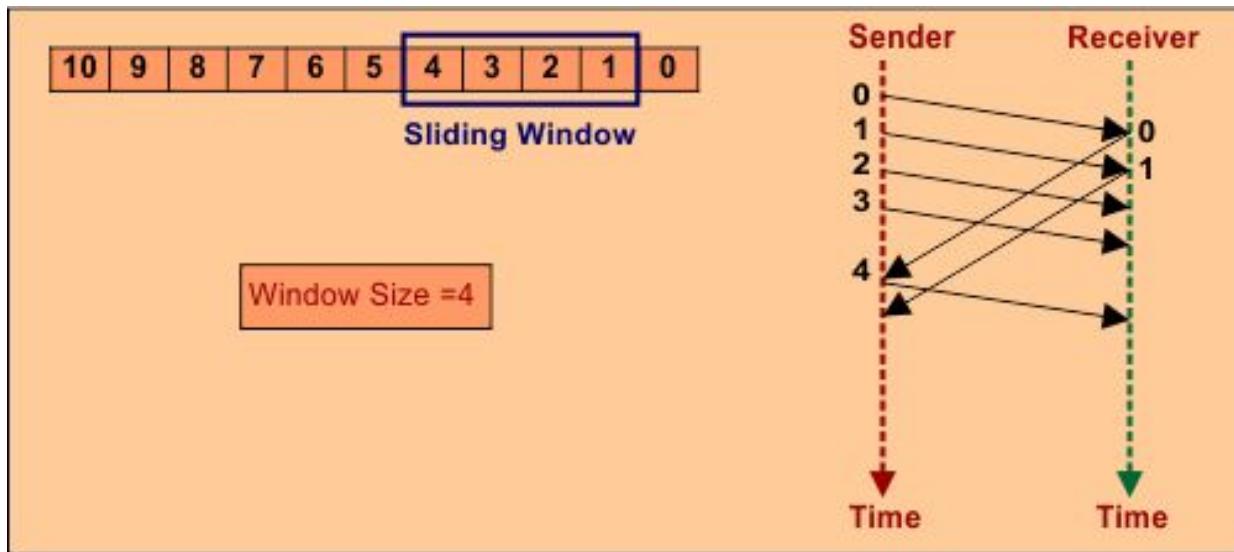
Working of Sliding Window Protocol

Step 3: After receiving the ACK the Sliding window moves one position next and transmit frame 4 and frame No.4 is transmitted.



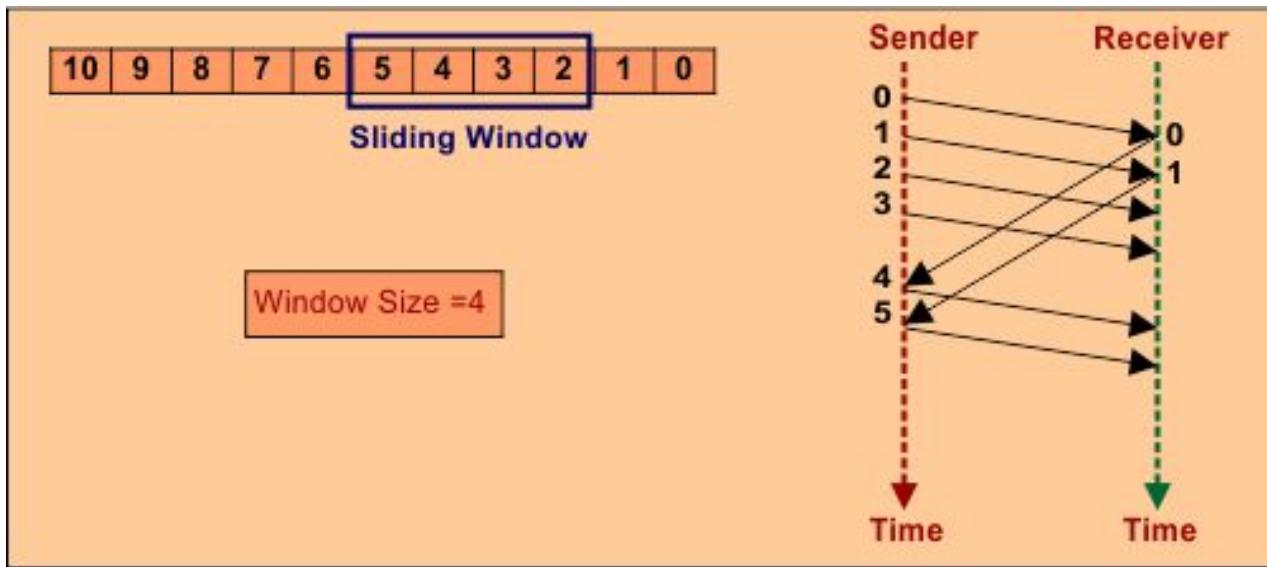
Working of Sliding Window Protocol

Step 4: After transmission of frame No.4, Sender waits for ACK from receiver of Second frame (i.e. 1).

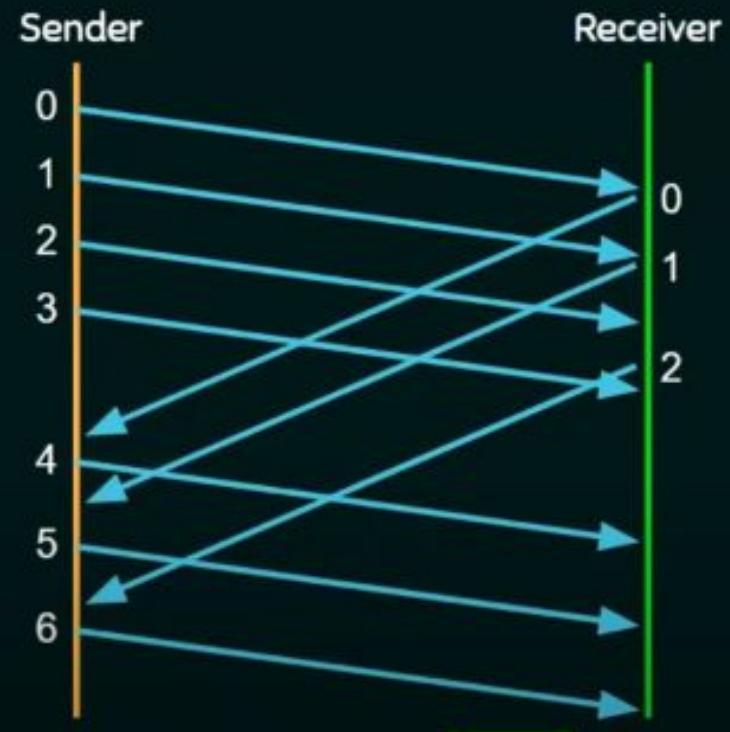
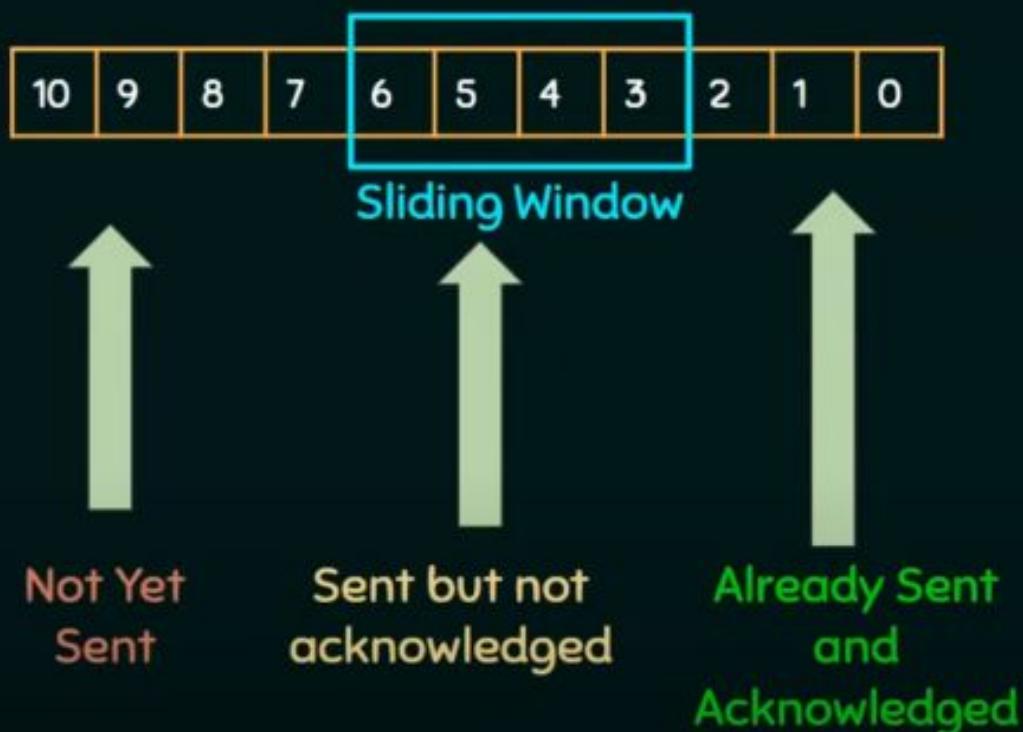


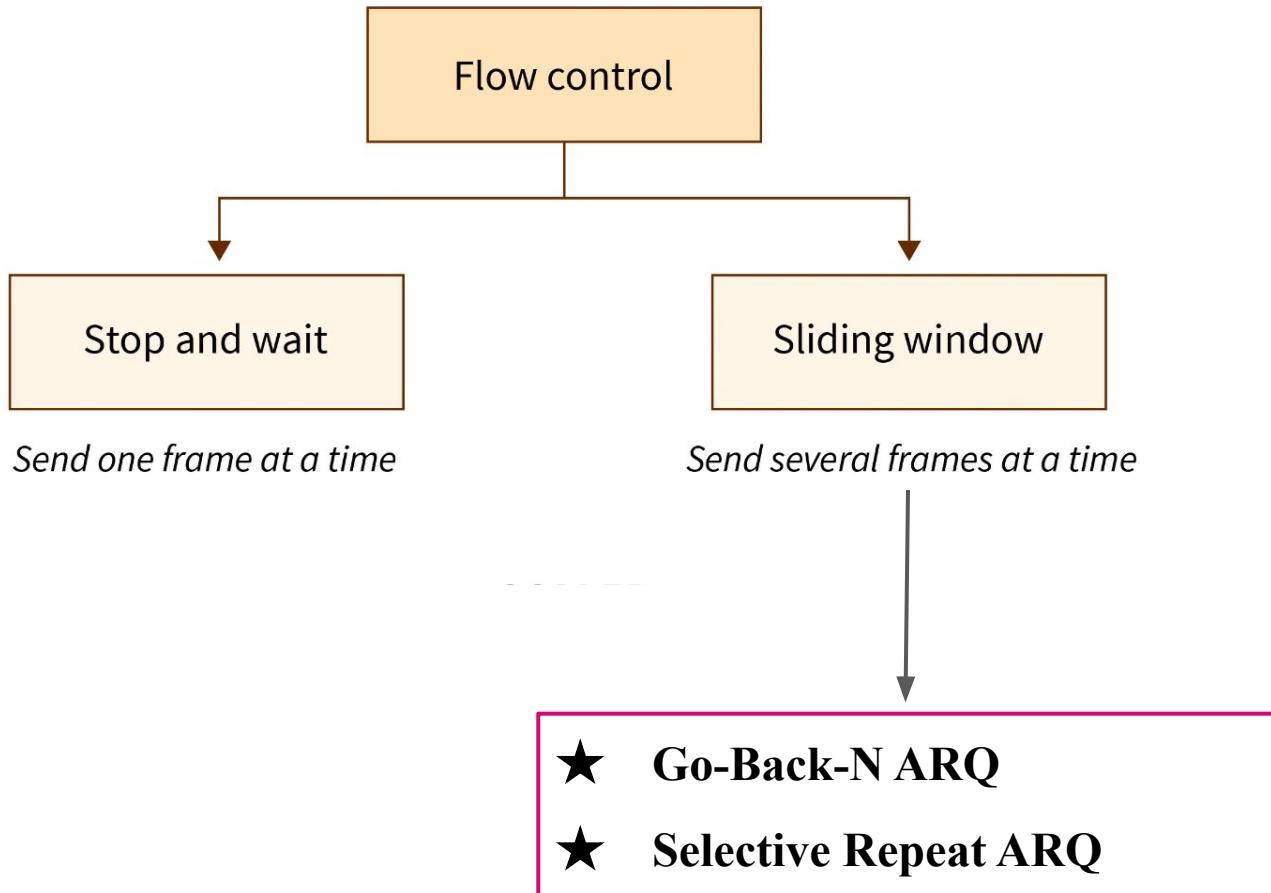
Working of Sliding Window Protocol

Step 5: After receiving ACK for Frame No.1 Sender transmit the Next frame No.5 by moving the sliding window one position next and so on.



Working of Sliding Window Protocol





Go-Back-N ARQ

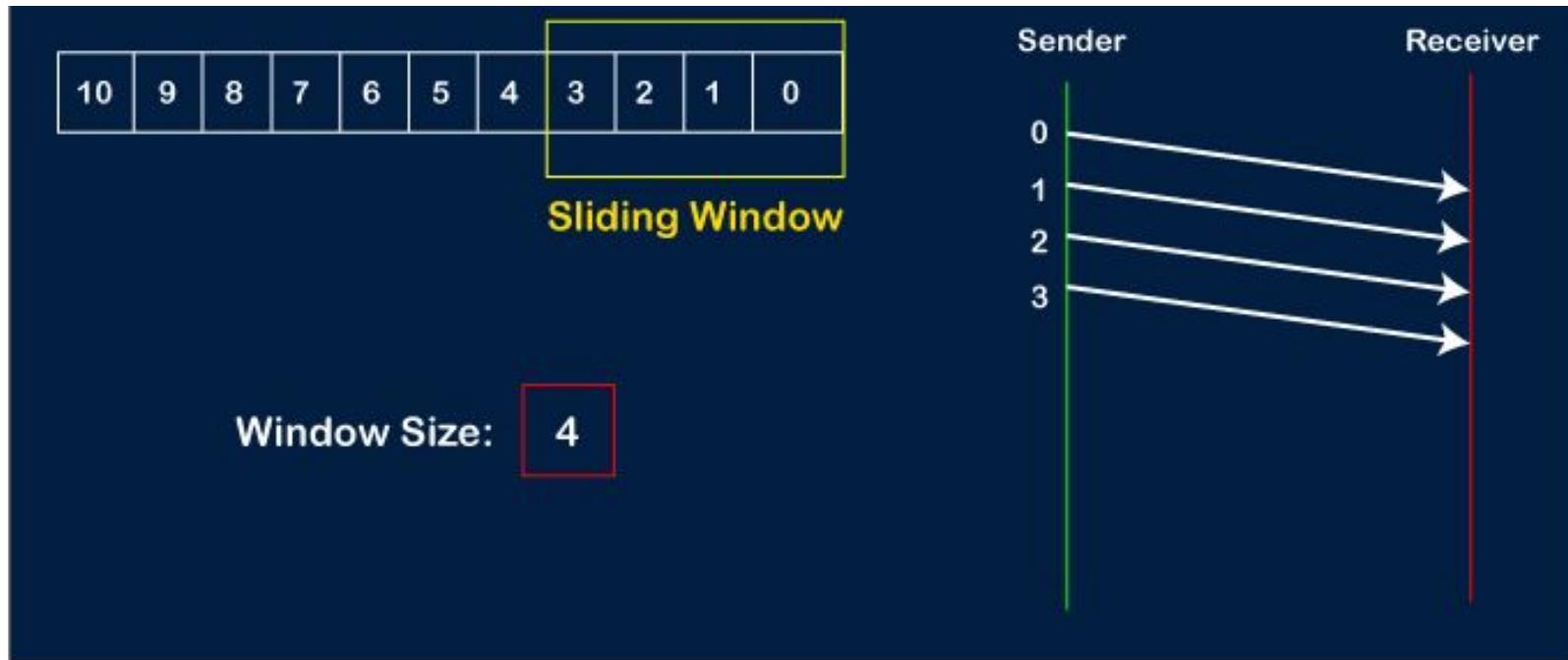
- In Go-Back-N ARQ, N is the sender's window size. Suppose we say that **Go-Back-3**, which means that the **three frames can be sent at a time before expecting the acknowledgment from the receiver**.
- It uses the principle of protocol **pipelining** in which the multiple frames can be sent before receiving the acknowledgment of the first frame. If we have **five frames** and the concept is **Go-Back-3**, which means that **the three frames can be sent**, i.e., **frame no 1, frame no 2, frame no 3** can be sent before expecting the acknowledgment of **frame no 1**.
- In Go-Back-N ARQ, the frames are numbered sequentially as Go-Back-N ARQ sends the multiple frames at a time that requires **the numbering approach** to distinguish the frame from another frame, and these numbers are known as **the sequential numbers**.

- The number of frames that can be sent at a time totally depends on the size of the sender's window.
- So, we can say that 'N' is the number of frames that can be sent at a time before receiving the acknowledgment from the receiver.
- If the acknowledgment of a frame is not received within an agreed-upon time period, then all the frames available in the current window will be retransmitted.
 - Suppose we have sent the frame no 5, but we didn't receive the acknowledgment of frame no 5, and the current window is holding three frames, then these three frames will be retransmitted.

Working of Go-Back-N ARQ

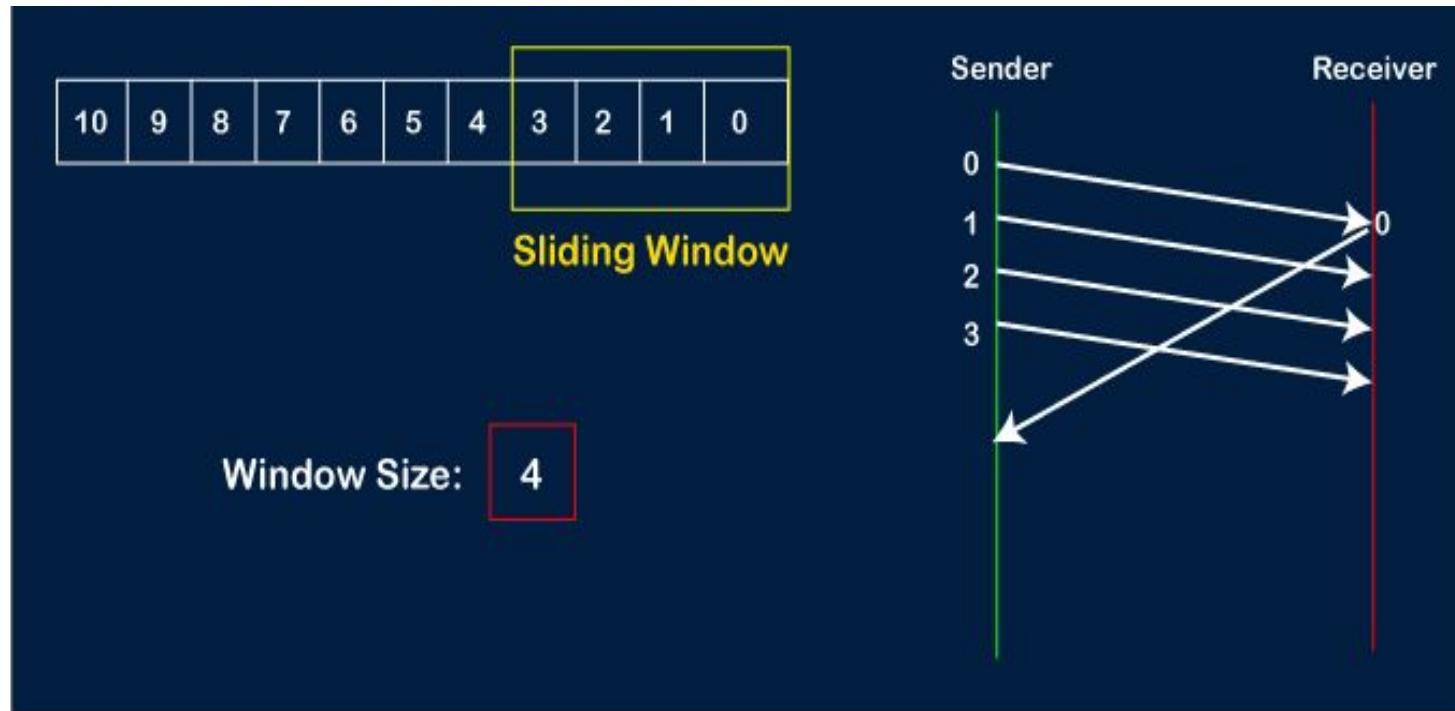
Example : For example, suppose there are 11 frames required transmitting and sender window size is 4

Step 1: Step 1: Firstly, the sender will send the first four frames to the receiver, i.e., 0,1,2,3, and now the sender is expected to receive the acknowledgment of the 0th frame.



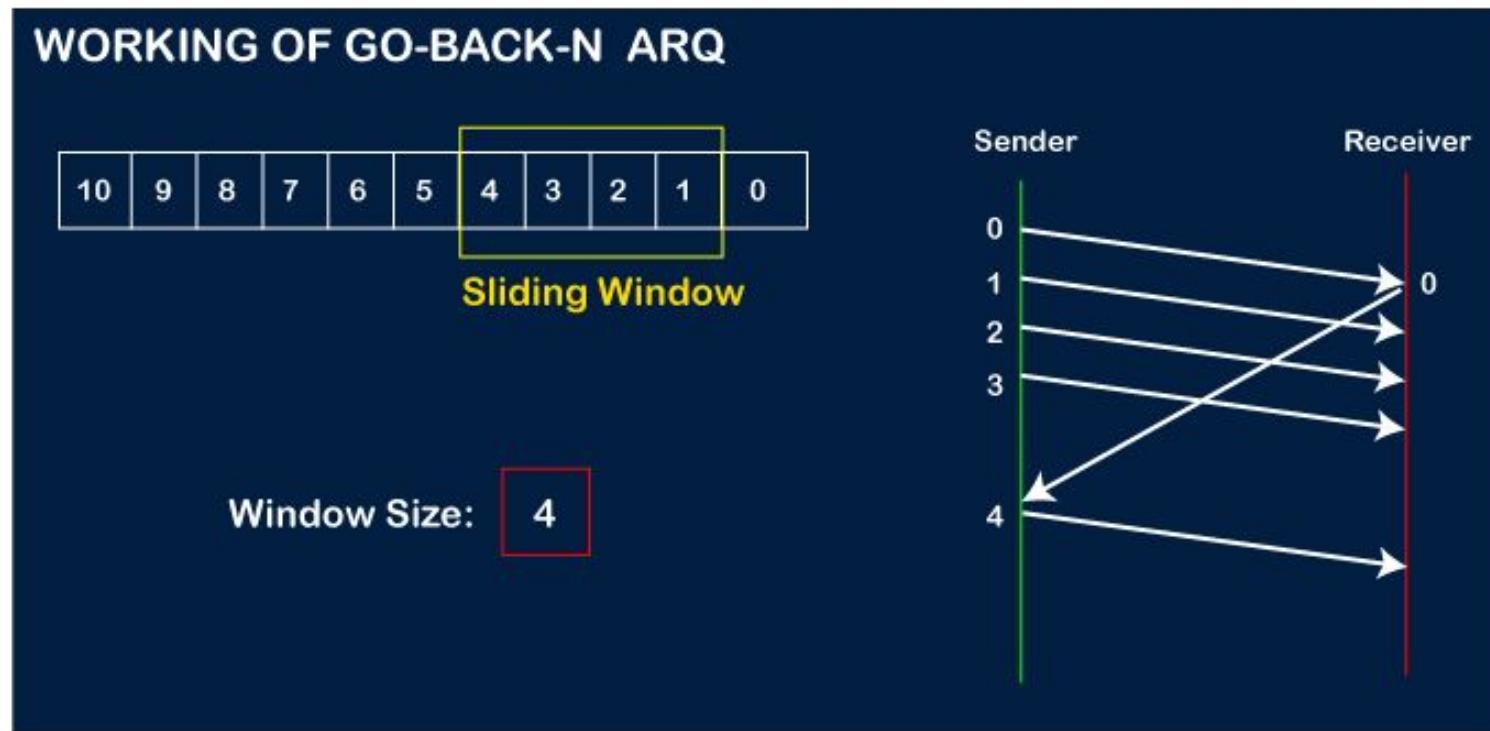
Working of Go-Back-N ARQ

Step 2: Let's assume that the receiver has sent the acknowledgment for the 0 frame, and the receiver has successfully received it.



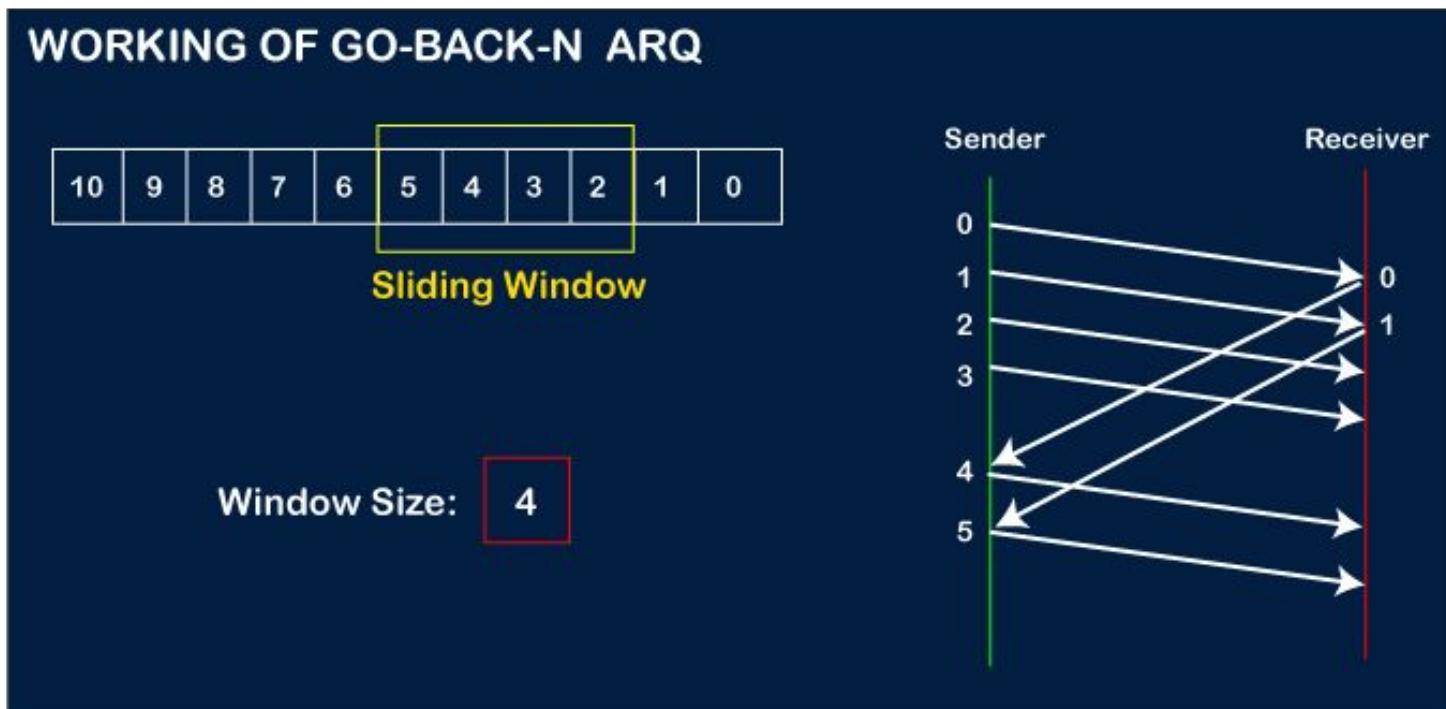
Working of Go-Back-N ARQ

Step 3: The sender will then send the next frame, i.e., 4, and the window slides containing four frames (1,2,3,4).



Working of Go-Back-N ARQ

Step 4: The receiver will then send the acknowledgment for the frame no 1. After receiving the acknowledgment, the sender will send the next frame, i.e., frame no 5, and the window will slide having four frames (2,3,4,5).



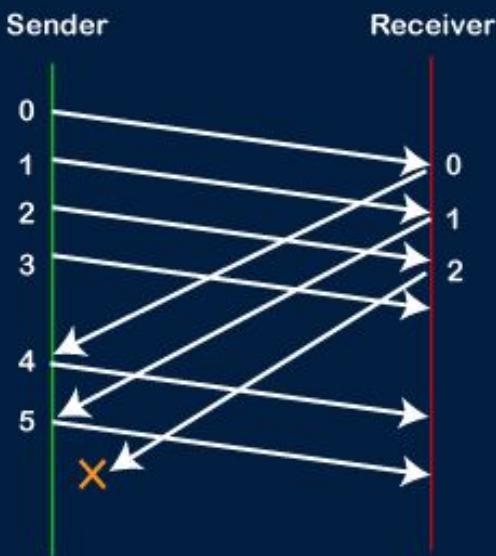
Step 5: Now, let's assume that the receiver is not acknowledging the frame no 2, either the frame is lost, or the acknowledgment is lost. Instead of sending the frame no 6, the sender Go-Back to 2, which is the first frame of the current window, retransmits all the frames in the current window, i.e., 2,3,4,5.

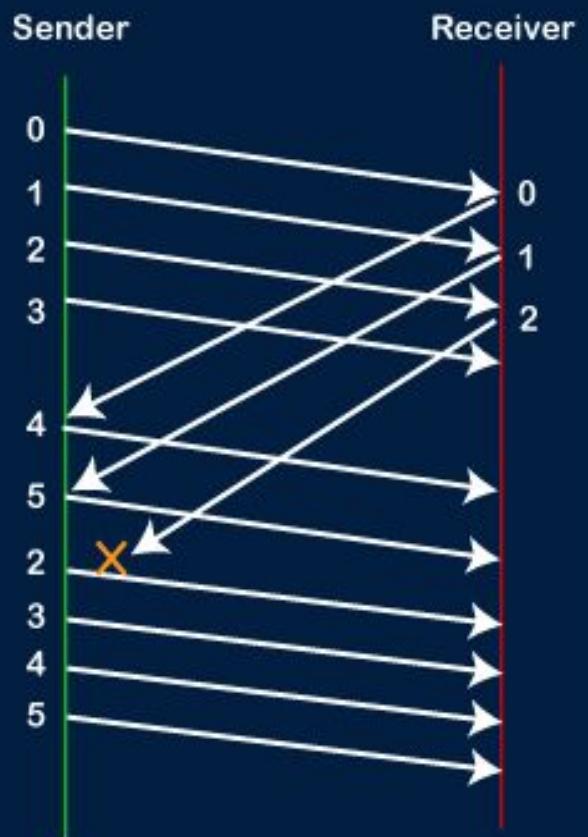
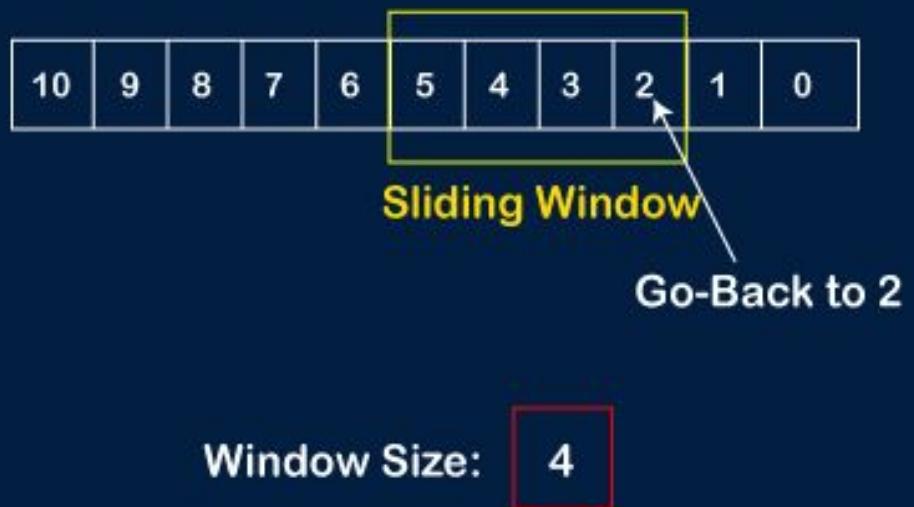
WORKING OF GO-BACK-N ARQ



Window Size:

4



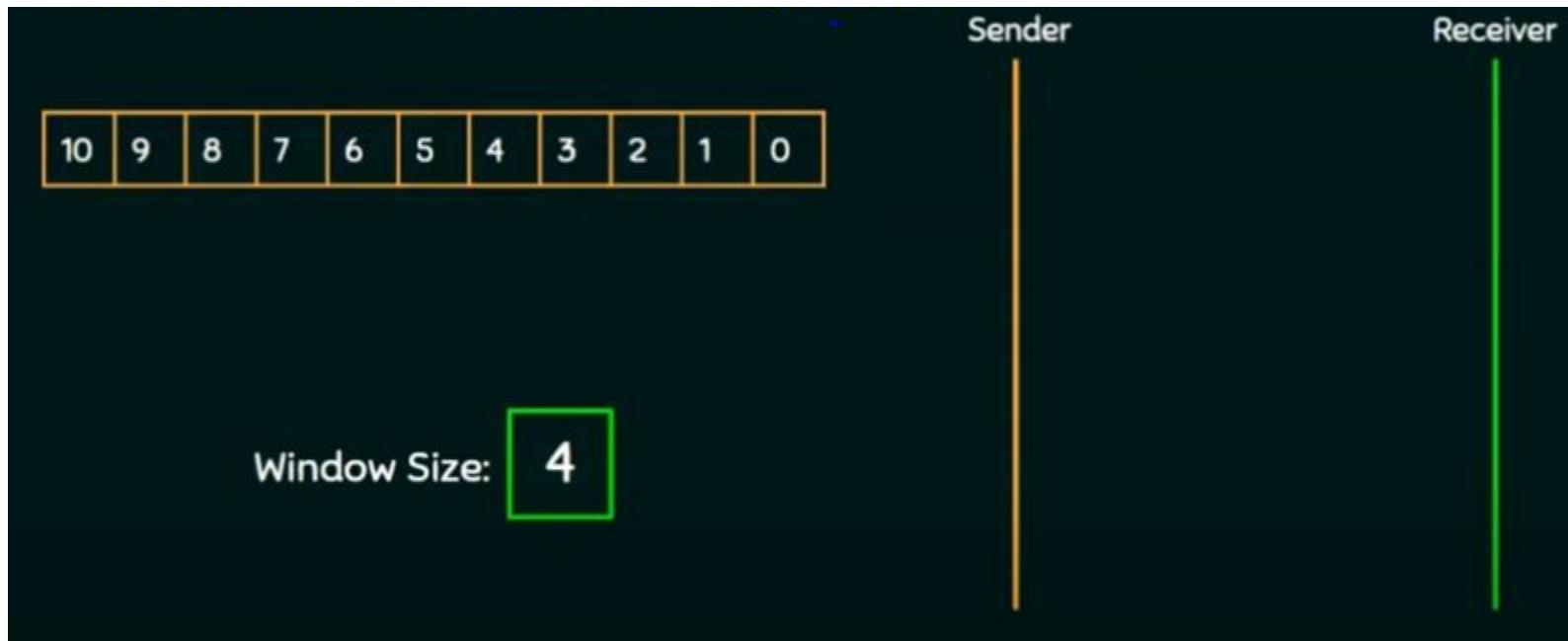


Selective Repeat ARQ

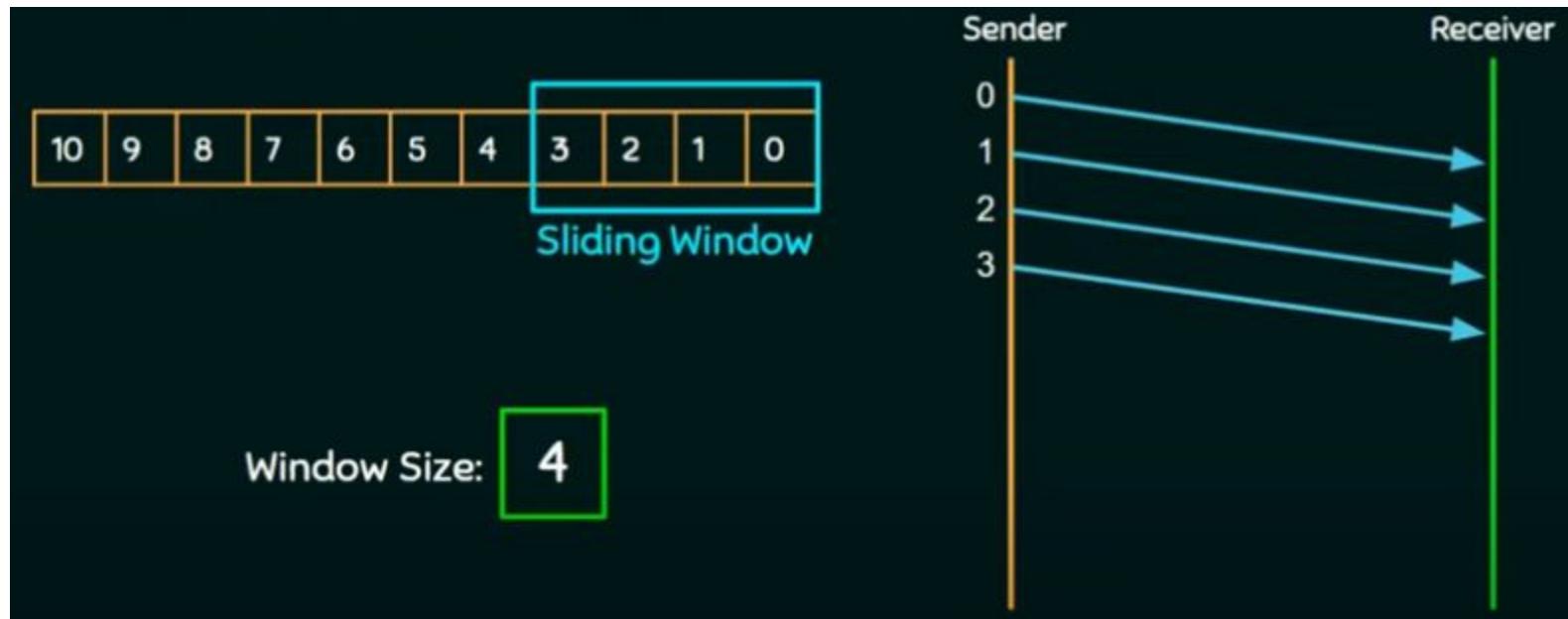
- In selective repeat, both the sender and receiver maintain a window. The sender can transmit multiple packets within the window, and the receiver acknowledges each correctly received packet.
- **If the sender does not receive an acknowledgment for a specific packet within a timeout period, it retransmits only that packet, instead of retransmitting the entire window.**
- Selective repeat reduces retransmissions and improves efficiency by eliminating unnecessary retransmissions of already received packets.

Working of Selective Repeat ARQ

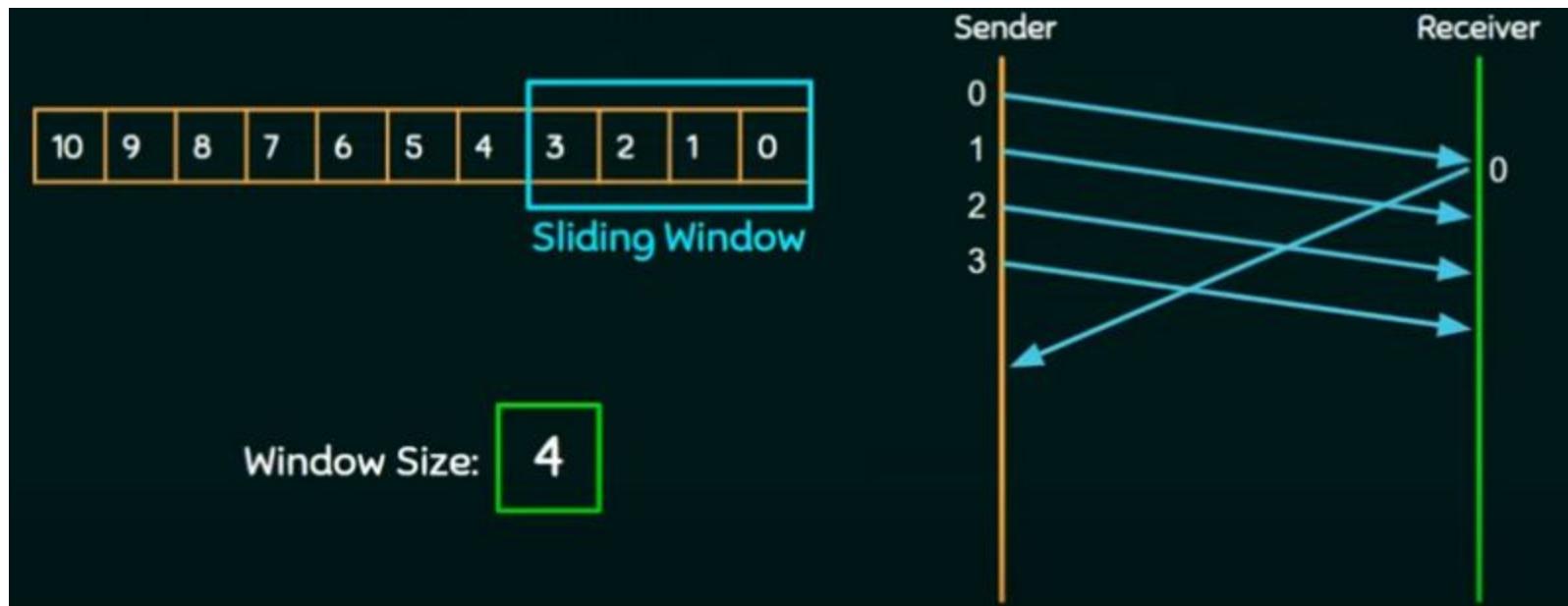
Example : Suppose there are 11 frames required transmitting and sender **window size is 4**



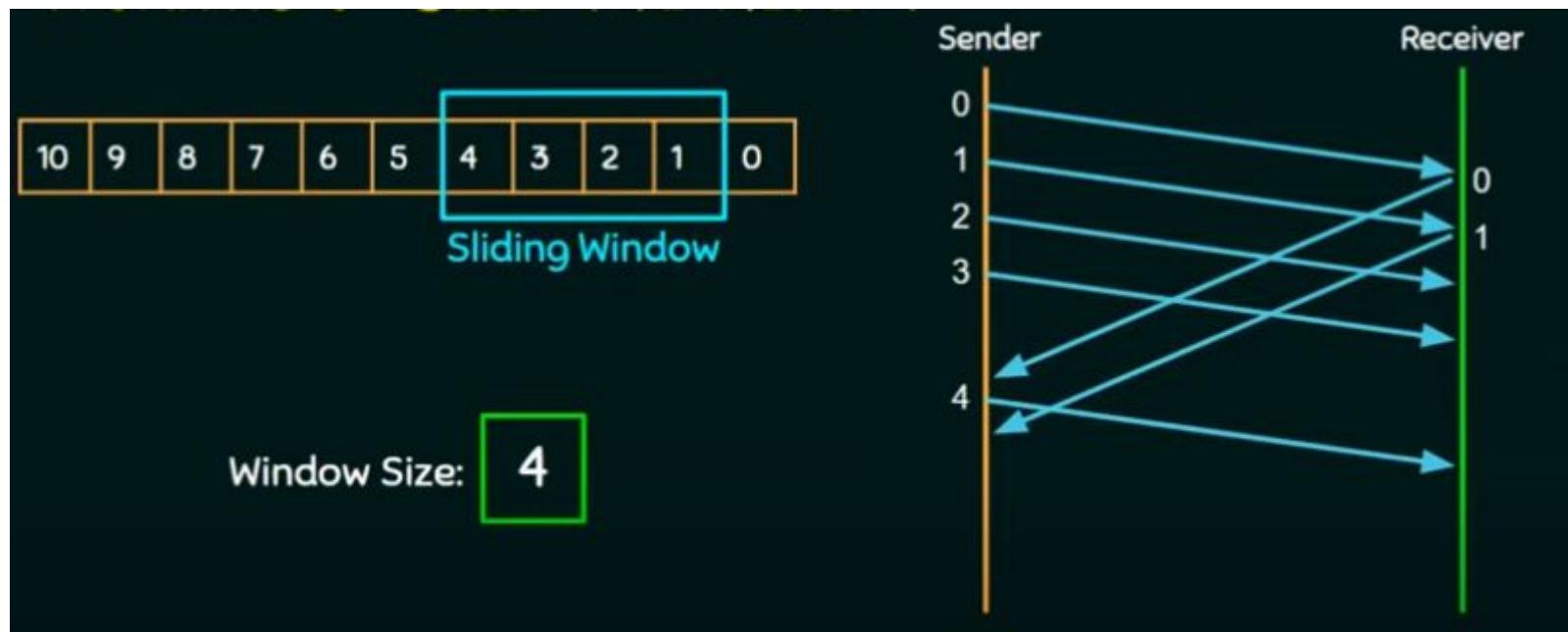
Step 1: Step 1: Firstly, the sender will send the first four frames to the receiver, i.e., 0,1,2,3, and now the sender is expected to receive the acknowledgment of the 0th frame.



Step 2: Let's assume that the receiver has sent the acknowledgment for the 0 frame, and the receiver has successfully received it.



Step 3: The sender will then send the next frame, i.e., 4, and the window slides containing four frames (1,2,3,4).

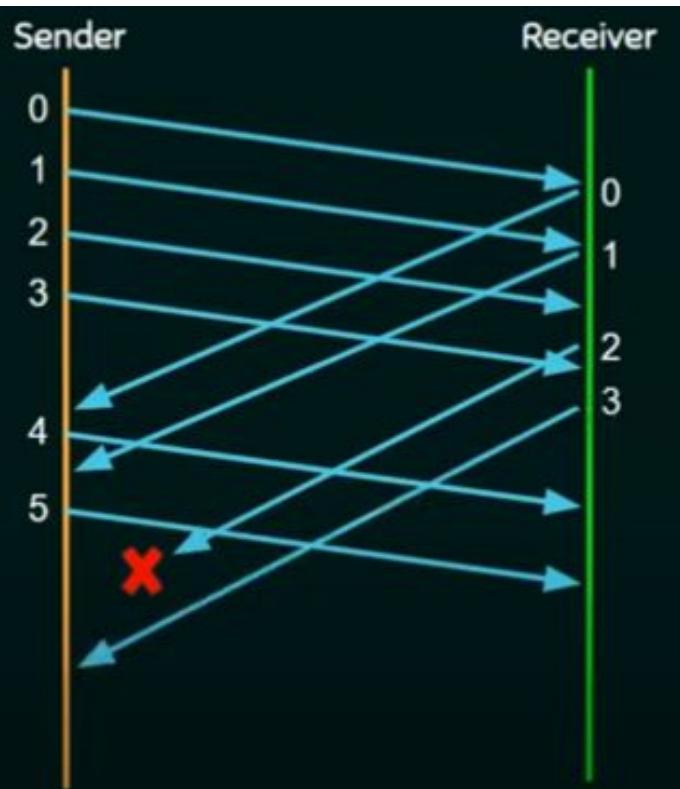
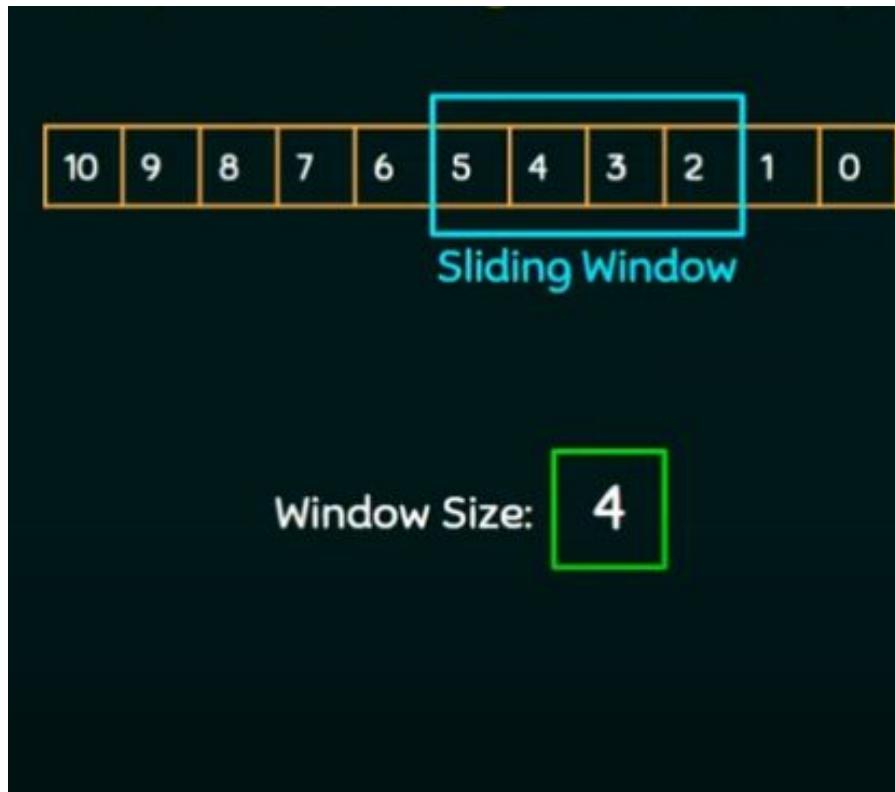


Step 4: The receiver will then send the acknowledgment for the frame no 1. After receiving the acknowledgment, the sender will send the next frame, i.e., frame no 5, and the window will slide having four frames (2,3,4,5).

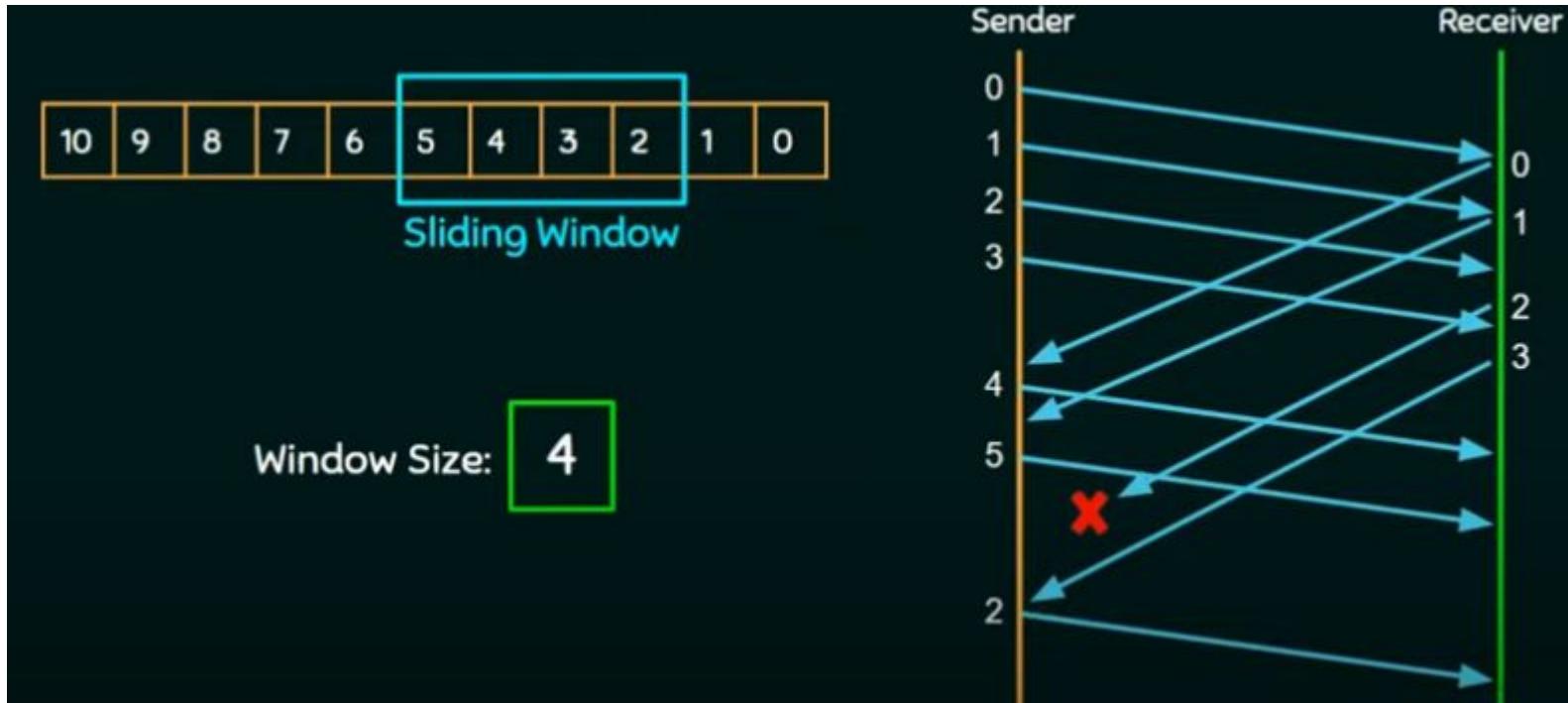


Step 5: Now, let's assume that the receiver is not acknowledging the frame no 2, either the frame is lost, or the acknowledgment is lost. In Selective Repeat ARQ only the lost or error frames are retransmitted, whereas correct frames are received and buffered.





Sender will **retransmit frame 2 alone** and as usual other frames are transmitted.



Go-Back-N ARQ

Selective Repeat ARQ

If a frame is corrupted or lost in it, all subsequent frames have to be sent again.

In this, only the frame is sent again, which is corrupted or lost.

If it has a high error rate, it wastes a lot of bandwidth.

There is a loss of low bandwidth.

It is less complex.

It is more complex because it has to do sorting and searching as well. And it also requires more storage.

It does not require sorting.

In this, sorting is done to get the frames in the correct order.

It does not require searching.

The search operation is performed in it.

It is used more.

It is used less because it is more complex.

HDLC(High-Level Data Link Control)

Outcomes:

- Understand the bit-oriented protocol.
- Importance of HDLC.
- Know the frame format of HDLC.
- Know the types of HDLC frames.

Bit -Oriented Approach:

- It simply views frames as collection of bits.

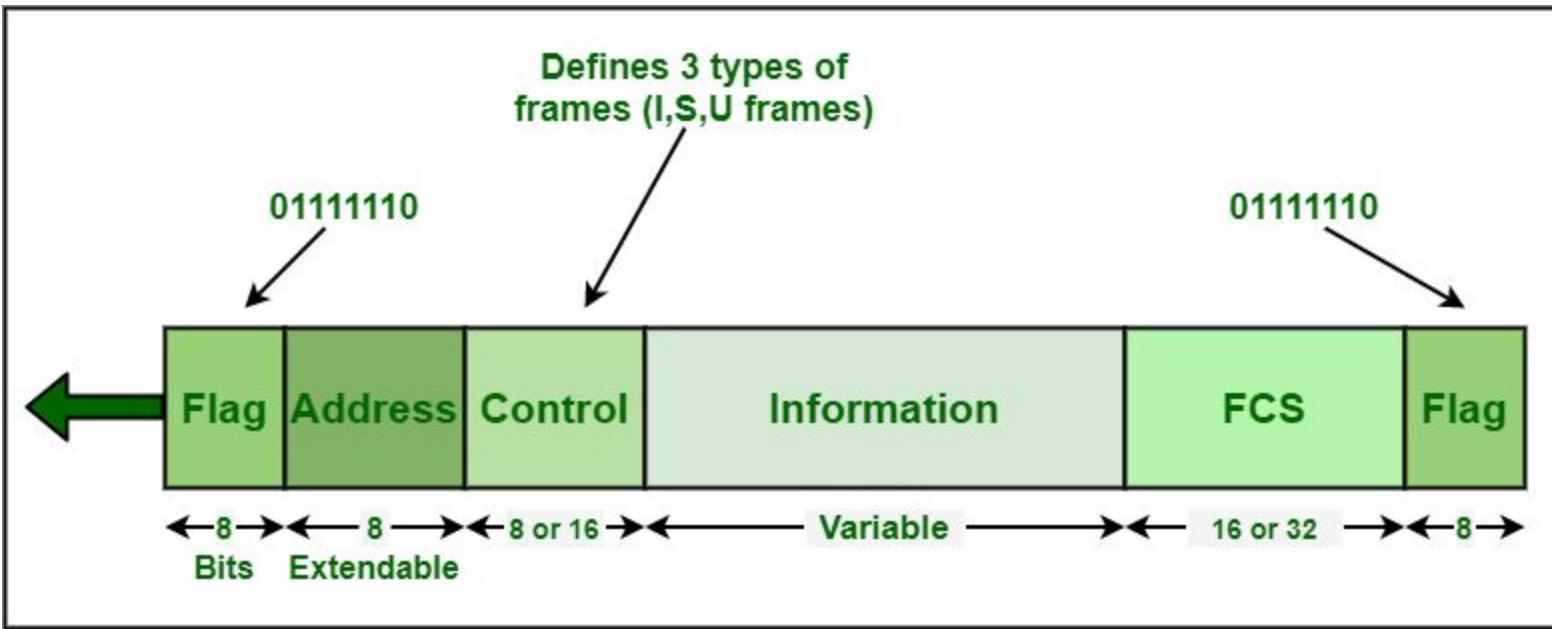
Bit -Oriented protocol:

- HDLC (High-Level Data Link Control)

What is HDLC ?

- Synchronous Data Link Control (**SDLC**) developed by **IBM** is an examples of bit-oriented protocol.
- SDLC was later standardized by **ISO** as **HDLC protocol**.
- High-Level Data Link Control (HDLC) basically provides **reliable delivery** of data frames over a network or communication link.
- HDLC provides various operations such as **framing, data transparency, error detection, and correction, and even flow control**.
- Primary stations simply transmit commands that contain address of secondary stations. The secondary station then simply transmits responses that contain its own address.

Frame Format of HDLC



Basic Frame Structure

Frame Format of HDLC

- **Flag:** The frame starts and ends with a flag delimiter, which is an 8-bit sequence. The most commonly used flag sequence is 01111110.
- **Address:** The address field is used to specify the source and destination devices. It can be 1 to 8 bits long, depending on the configuration of the HDLC variant.
- **Control:** The control field is used to indicate the frame type and control information. It is typically 8 bits long and contains various control bits for different purposes, such as indicating supervisory frames, information frames, or commands.
- **Information:** The information field carries the actual data being transmitted. Its length can vary depending on the needs of the application.
- **Cyclic Redundancy Check (CRC):** a CRC (Cyclic Redundancy Check) value, which is calculated based on the frame's data and control fields. The receiver uses the CRC value to check for errors in the received frame.
- **Flag:** The frame ends with another flag delimiter, which is the same 8-bit sequence used at the beginning of the frame.

Types of HDLC Frames

Control Field –

- HDLC generally uses this field to determine **how to control process of communication.**
- The control field is different for different types of frames in HDLC protocol.
- The types of frames can be
 - **Information frame (I-frame)**
 - **Supervisory frame (S-frame)**
 - **Un-numbered frame (U-frame).**

I-Frame	In the control field if the first bit is 0	Carrying Information
S-Frame	In the control field if the first bit is 10	Used for Error Control and Flow Control
U-Frame	In the control field if the first bit is 11	Used for Link Management

The data link layer in the internet

- I. In the Internet, the data link layer is responsible for the reliable transmission of data over a physical network link.
- II. It sits above the physical layer and below the network layer in the networking protocol stack. While the Internet Protocol (IP) primarily operates at the network layer, the data link layer is essential for the proper functioning of IP-based communication.
- III. The data link layer in the Internet encompasses various protocols and technologies, including:
 - A. Ethernet:** Ethernet is the most commonly used technology at the data link layer in local area networks (LANs). It defines the standards for wired communication, specifying the physical and data link layer protocols for transmitting data between devices connected to the same LAN.
 - B. Wi-Fi (Wireless LAN):** Wi-Fi is a wireless data link layer technology that enables devices to connect to a network wirelessly. It uses protocols such as IEEE 802.11 to establish wireless connections and handle data transmission over the air.

C. Point-to-Point Protocol (PPP): PPP is a data link layer protocol used for establishing direct point-to-point connections over various physical links, including dial-up connections and dedicated serial links. It provides authentication, error detection, and framing capabilities.

D. Asynchronous Transfer Mode (ATM): Although less commonly used in modern networks, ATM was a data link layer technology that provided high-speed transmission of fixed-size cells. It was used in wide area networks (WANs) and for certain specialized applications.

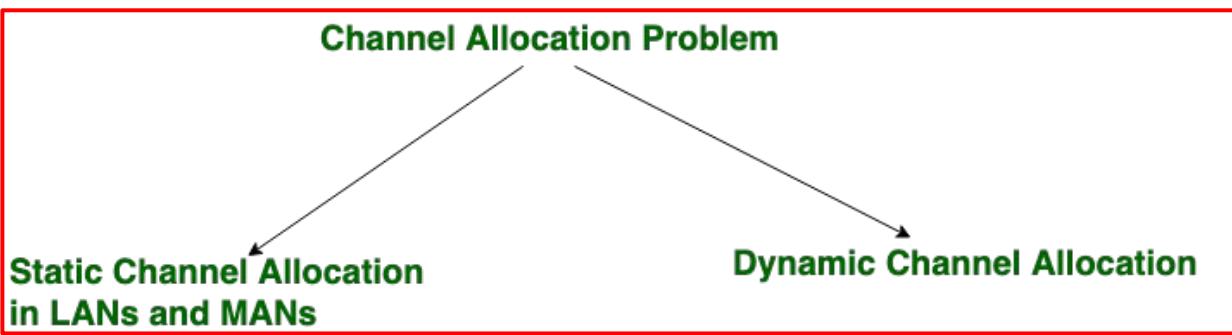
E. MPLS (Multi-Protocol Label Switching): MPLS is a protocol that operates at both the network and data link layers. It adds a label to IP packets at the data link layer, allowing routers to make forwarding decisions based on the label rather than examining the entire IP header.

Syllabus

- **The Media Access Sub Layer**
 - **Channel allocation problem**
 - **Multiple access protocols.**

Channel allocation problem

- Channel allocation is a process in which **a single channel is divided and allotted to multiple users** in order to carry user specific tasks.
- There are user's quantity may vary every time the process takes place.
- If there are N number of users and channel is divided into N equal-sized sub channels, Each user is assigned one portion.
- If the number of users are small and don't vary at times, then Frequency Division Multiplexing can be used as it is a simple and efficient channel bandwidth allocating technique.
- Solved by two schemes:



Static Channel Allocation in LANs and MANs:

- It is the classical or traditional approach of allocating a single channel among multiple competing users using Frequency Division Multiplexing (FDM).
- If there are N users, the frequency channel is divided into N equal sized portions (bandwidth), each user being assigned one portion. since each user has a private frequency band, there is no interference between users.
- It is **not** efficient to divide into **fixed number of chunks**.

$$T = 1 / (U * C - L)$$

$$T(FDM) = N * T(1/U(C/N) - L/N)$$

Where:

T = Mean time delay,

C = Capacity of channel,

L = Arrival rate of frames,

$1/U$ = bits/frame,

N = Number of sub channels,

$T(FDM)$ = Frequency Division Multiplexing Time

2. Dynamic Channel Allocation

1. Independent Traffic

- ★ The model consists of N independent stations (e.g., computers, telephones), each with a program or user that generates frames for transmission.
- ★ The expected number of frames generated in an interval of length Δt is $\lambda\Delta t$, where λ is a constant (the arrival rate of new frames).
- ★ Once a frame has been generated, the station is blocked and does nothing until the frame has been successfully transmitted.

2. Single Channel.

- ★ A single channel is available for all communication. All stations can transmit on it and all can receive from it.
- ★ The stations are assumed to be equally capable, though protocols may assign them different roles (e.g., priorities).

2. Dynamic Channel Allocation

3. Observable Collisions.

- ★ If two frames are transmitted simultaneously, they overlap in time and the resulting signal is garbled.
- ★ This event is called a collision. All stations can detect that a collision has occurred.
- ★ A collided frame must be transmitted again later. No errors other than those generated by collisions occur.

4. Continuous or Slotted Time

- ★ Time may be assumed continuous, in which case frame transmission can begin at any instant. Alternatively, time may be slotted or divided into discrete intervals (called slots).
- ★ Frame transmissions must then begin at the start of a slot.
- ★ A slot may contain 0, 1, or more frames, corresponding to an idle slot, a successful transmission, or a collision, respectively.

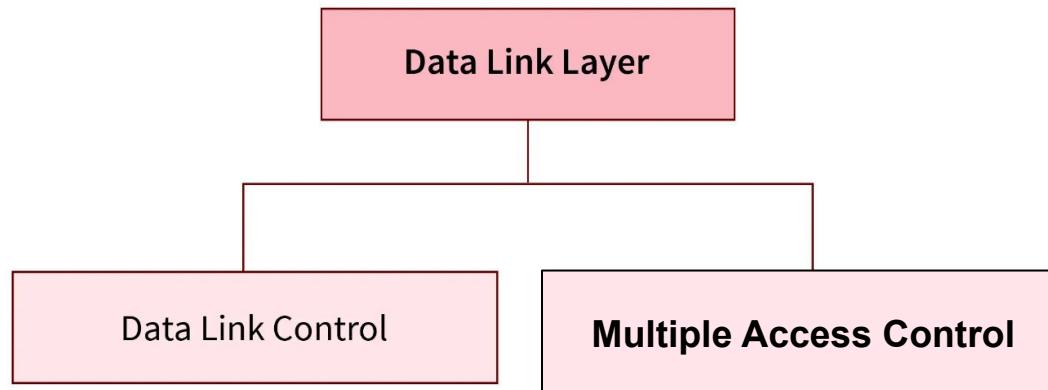
2. Dynamic Channel Allocation

5. Carrier Sense or No Carrier Sense

- ★ With the carrier sense assumption, stations can tell if the channel is in use before trying to use it.
- ★ No station will attempt to use the channel while it is sensed as busy.
- ★ If there is no carrier sense, stations cannot sense the channel before trying to use it.
- ★ They just go ahead and transmit. Only later can they determine whether the transmission was successful.

Multiple Access Protocols

- The Data Link Layer is responsible for transmission of data between two nodes. Its main functions are-



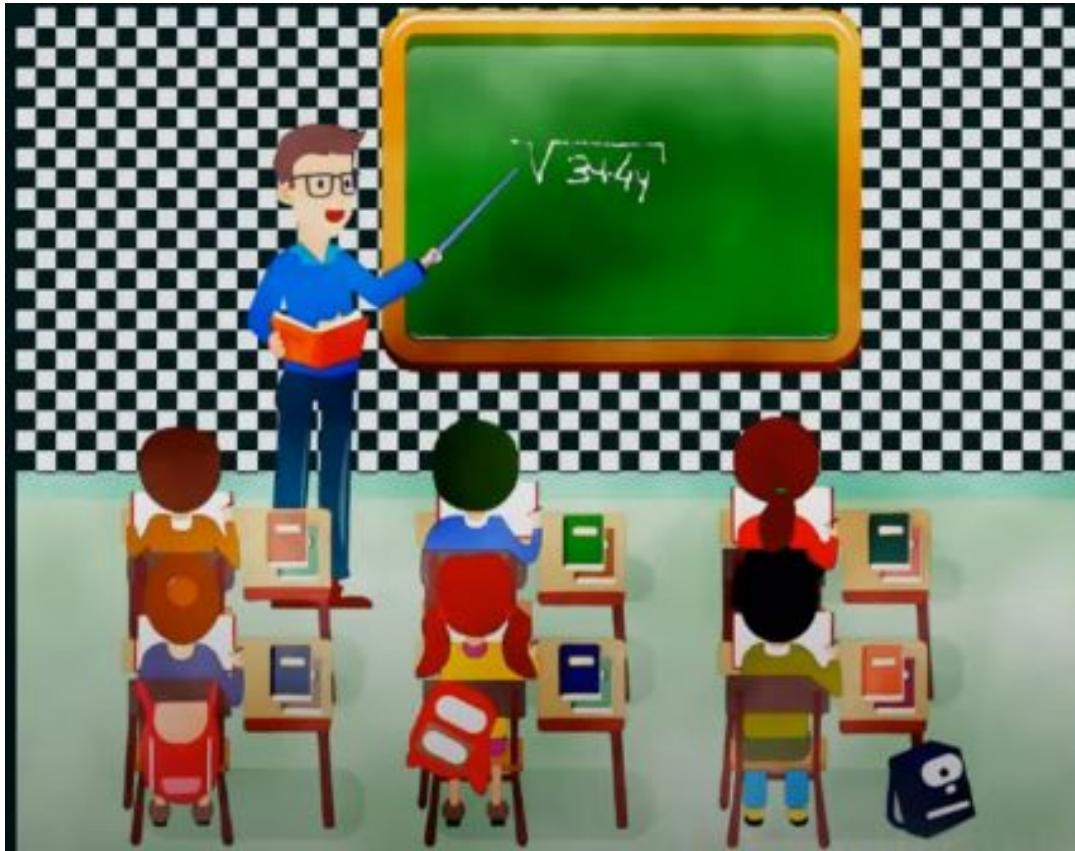
Data Link control :

- Data Link Control is a sublayer of the Data Link Layer (Layer 2) in the OSI model.
- **The data link control protocols offer reliable communication** when there is **a dedicated link** between the communicating stations.
- **If there is a dedicated link**, the data link control protocols are self-sufficient to handle the **framing, flow and error control**.

Why Multiple Access Control:

- Multiple access protocols are a set of protocols operating in the **Medium Access Control sublayer (MAC sublayer)** of the Open Systems Interconnection (OSI) model.
- If there is **a dedicated link** between the sender and the receiver then **data link control layer is sufficient**.
- However if there is **no dedicated link** present then **multiple stations** can access the channel simultaneously. Hence **multiple access protocols** are required to decrease collision and avoid crosstalk.

Analogy:



COLLISION



COLLISION



COLLISION

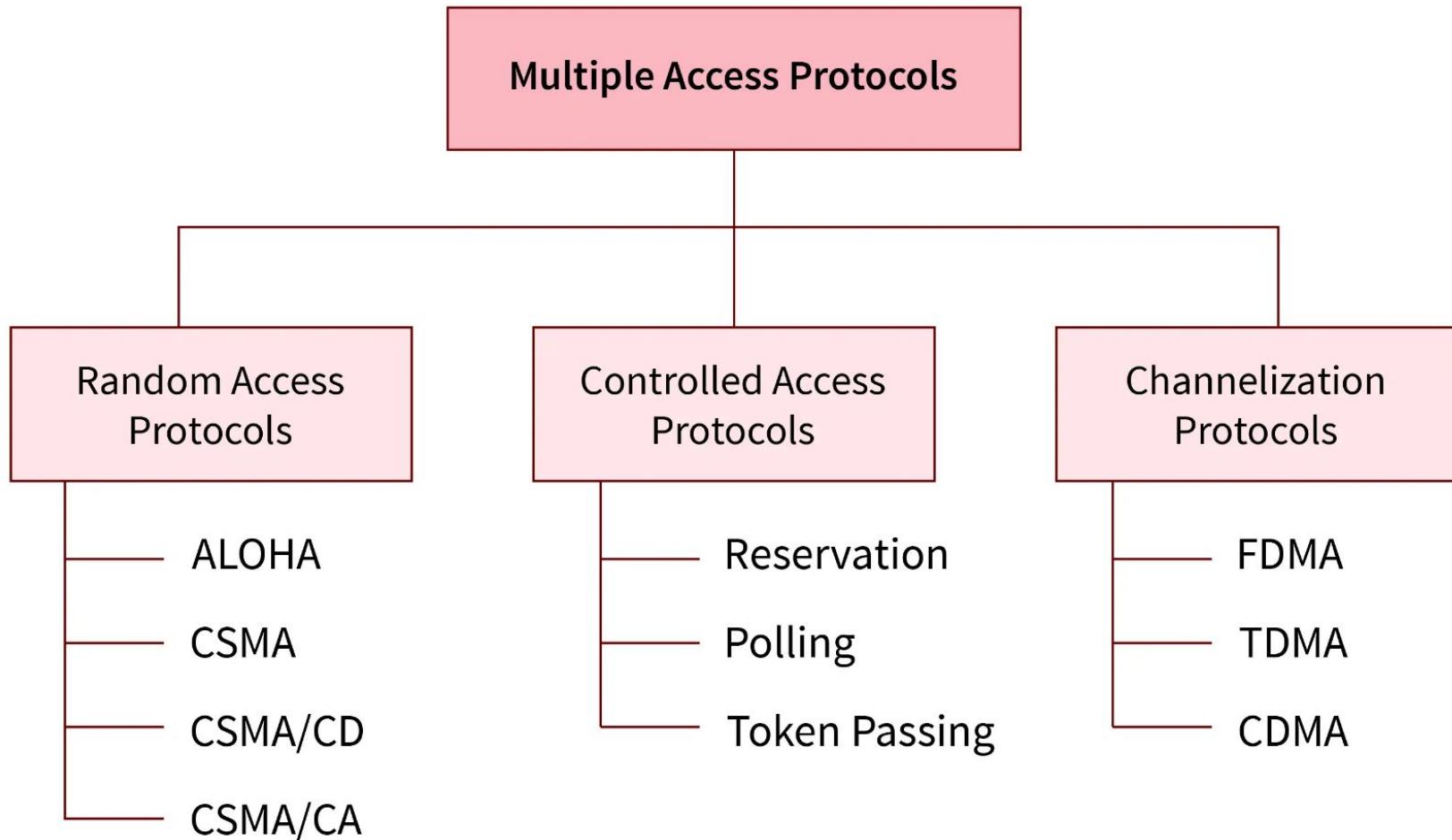


COLLISION



Example:

- [1] For example, in a classroom full of students, when a teacher asks a question and all the students (or stations) start answering simultaneously (send data at same time) then a lot of chaos is created(data overlap or data lost) then it is the job of the teacher (multiple access protocols) to manage the students and make them answer one at a time.
- [2] A Real-time example of a Medium Access Control (MAC) protocol is the Wi-Fi protocol (IEEE 802.11 standard). Wi-Fi is a widely used wireless communication technology that allows devices to connect to the internet or a local network. Within the Wi-Fi protocol, the MAC layer is responsible for controlling access to the shared wireless medium.



1. Random Access Protocol:

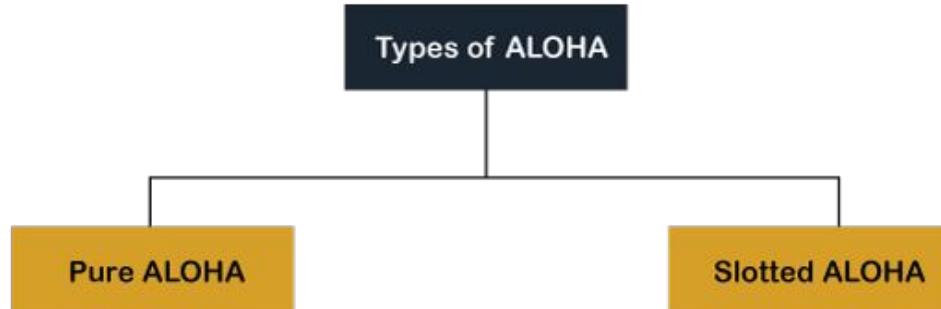
- ★ In this, **all stations** have **same superiority** that is no station has more priority than another station.
- ★ Any station can send data depending on medium's state(idle or busy).
- ★ It has two features:
 - There is **no fixed time** for sending data
 - There is **no fixed sequence of stations** sending data

The Random access protocols are further subdivided as:

1. **ALOHA**
2. **Carrier Sense Multiple Access (CSMA)**
3. **Carrier Sense Multiple Access with Collision Detection (CSMA/CD)**
4. **Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA)**

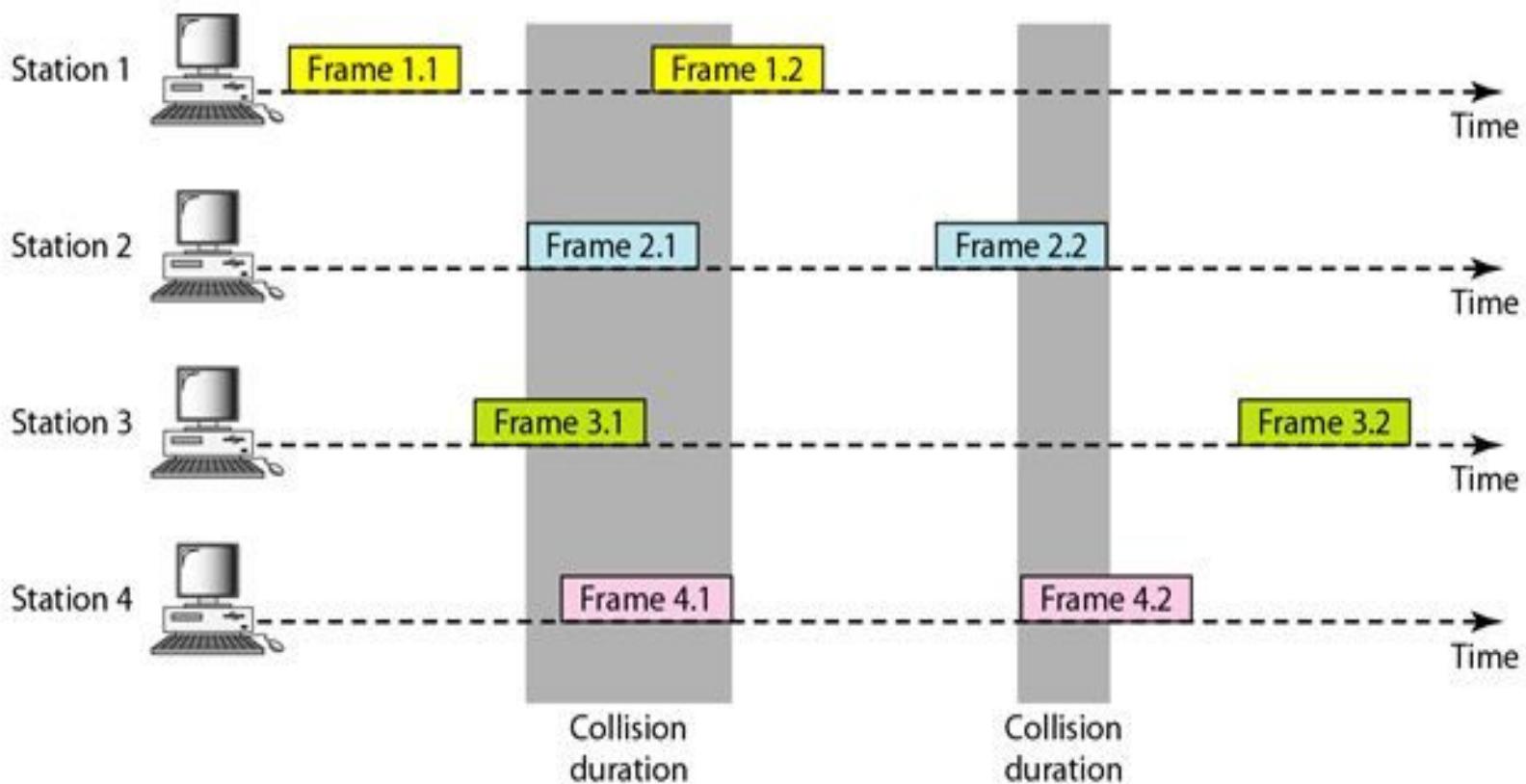
ALOHA

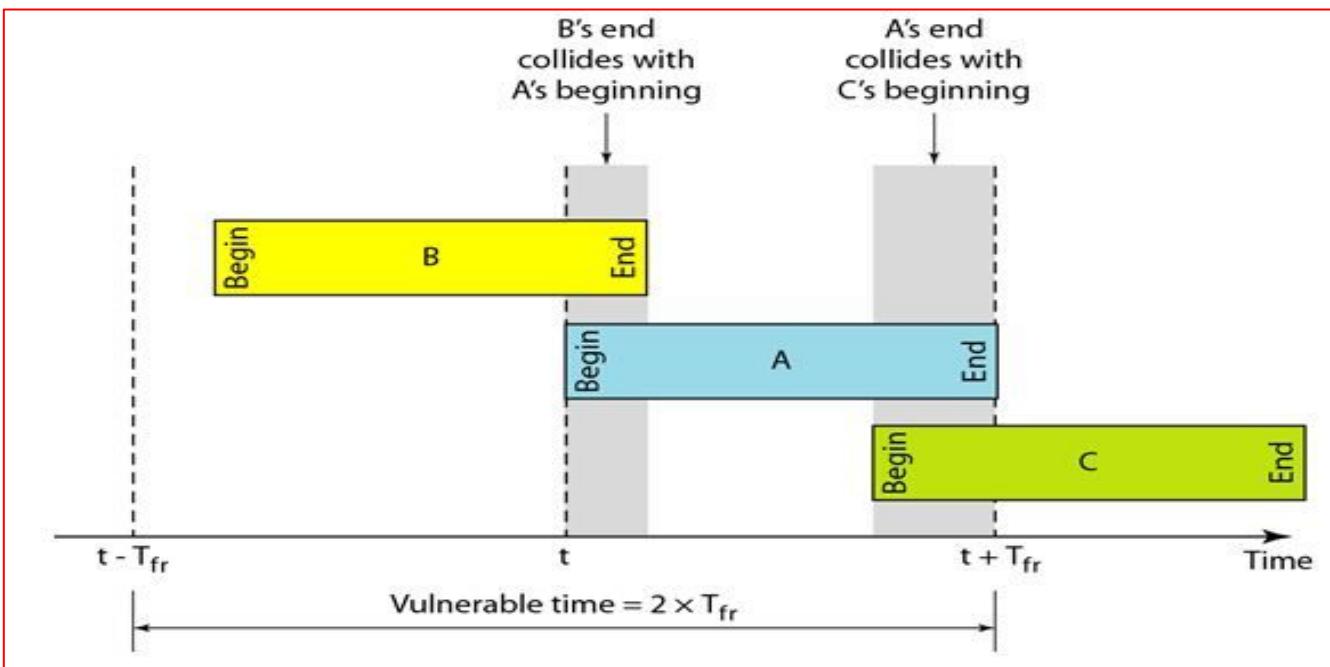
- ALOHA is an early **random access protocol** used in computer networks.
- It was first used in the ALOHAnet network at the University of Hawaii in the 1970s.
- The basic operation of the ALOHA protocol is as follows:
 - *Devices can transmit data whenever they have a message to send.*
 - *If two or more devices transmit simultaneously, their messages will collide and be corrupted.*
 - *Devices that detect a collision will wait for a random amount of time before trying to transmit again.*



PURE ALOHA

- Pure Aloha allows stations to transmit whenever they have data to be sent.
- In pure Aloha, when each station transmits data to a channel **without checking** whether the channel is **idle or busy**, the chances of collision may occur, and the data frame can be lost.
- When any station transmits the data frame to a channel, the pure Aloha waits for the receiver's acknowledgment.
- If it does not acknowledge the receiver end within the specified time, the station waits for a **random amount of time**, called **the backoff time (T_b)**. And the station may assume the frame has been lost or destroyed.
- Therefore, it retransmits the frame until all the data are successfully transmitted to the receiver.
- Since different stations wait for different amount of time, the probability of further collision decreases.
- The throughput of pure aloha is maximized when frames are of uniform length.





- Whenever two frames try to occupy the channel at the same time, there will be a collision and both will be garbled.
- If the first bit of a new frame overlaps with just the last bit of a frame almost finished, both frames will be totally destroyed and both will have to be retransmitted later.

- The time required to send a frame is called **Frame Transmission Time(T_{fr}).**
- **Vulnerable time** (the length of time in which there is a possibility of collision) for pure ALOHA is:

- **Vulnerable Time = $2 * T_{fr}$**
(T_{fr} Frame transmission time)
- **Throughput (S) = $G \times e^{-2G}$**

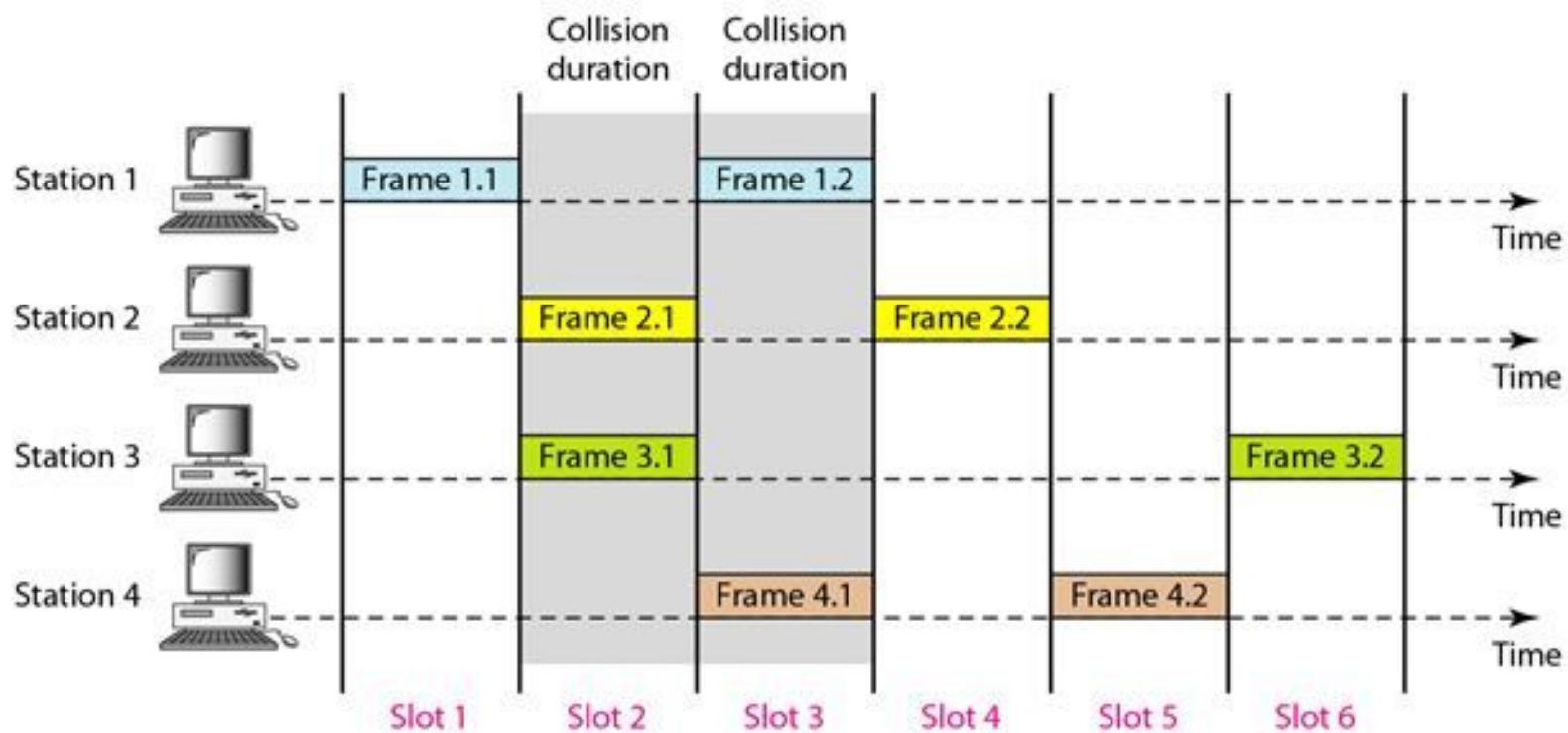
Where **G** is the average number of transmission attempts per unit of time and **e** is the mathematical constant approximately equal to 2.71828. The maximum throughput occurs when $G = 0.5$,

- **Maximum throughput = 0.184**

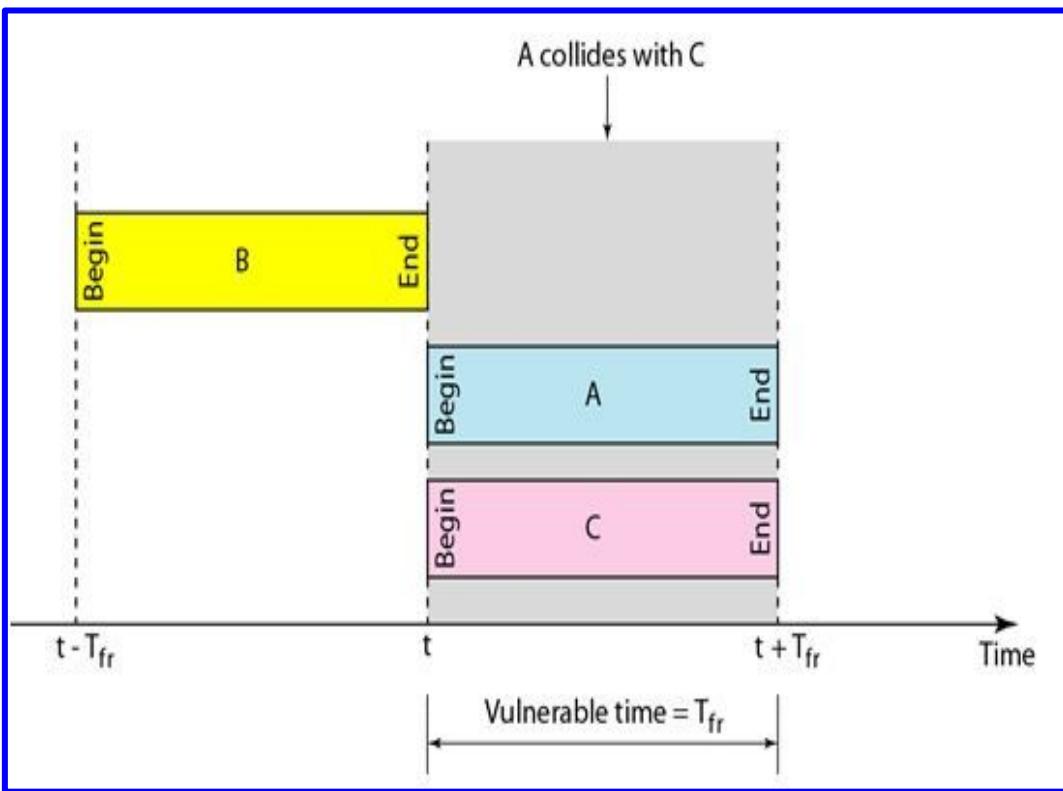
SLOTTED ALOHA

- Slotted ALOHA is an improvement over ALOHA and Pure ALOHA, introducing a **synchronized time-slot-based approach**.
- In Slotted ALOHA, **the time is divided into discrete slots**.
- Devices are allowed to transmit only **at the beginning of each time slot**.
- This synchronization **reduces the chances of collisions**.
- If a collision occurs, the devices involved wait for the **next time slot to retransmit**.
- If a station misses out the allowed time, it must wait for the next slot, This **reduces the probability of collision**.

Slotted Aloha



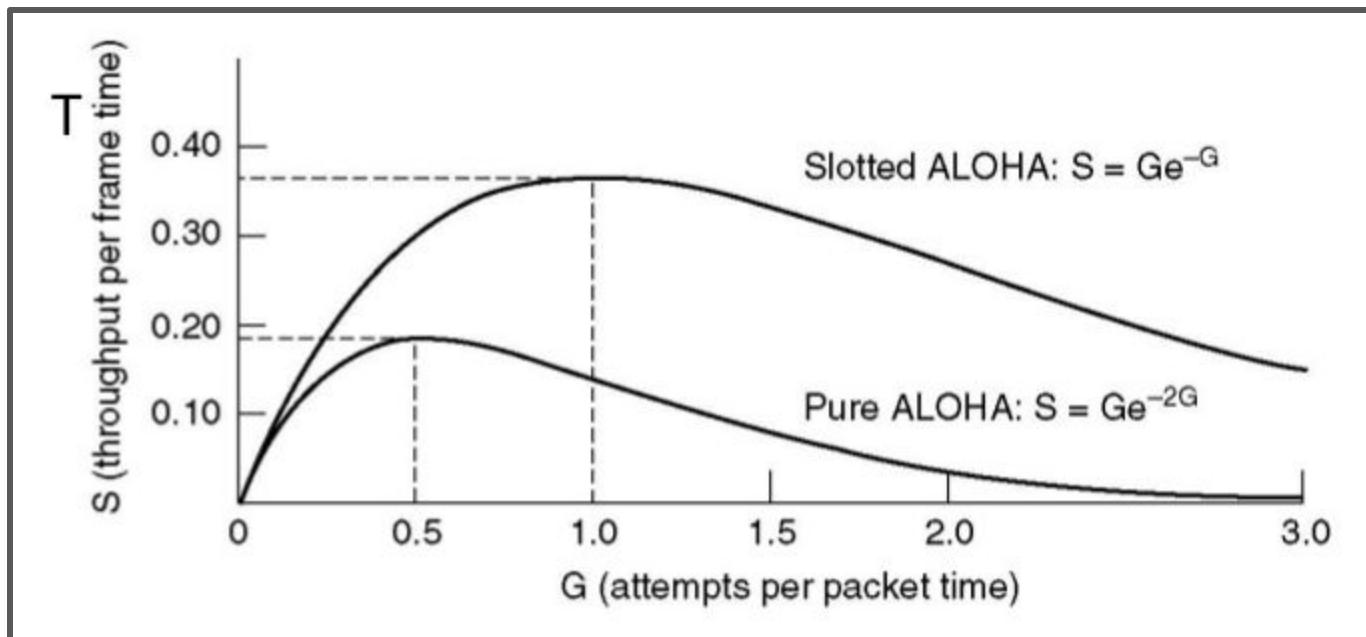
Vulnerable Time



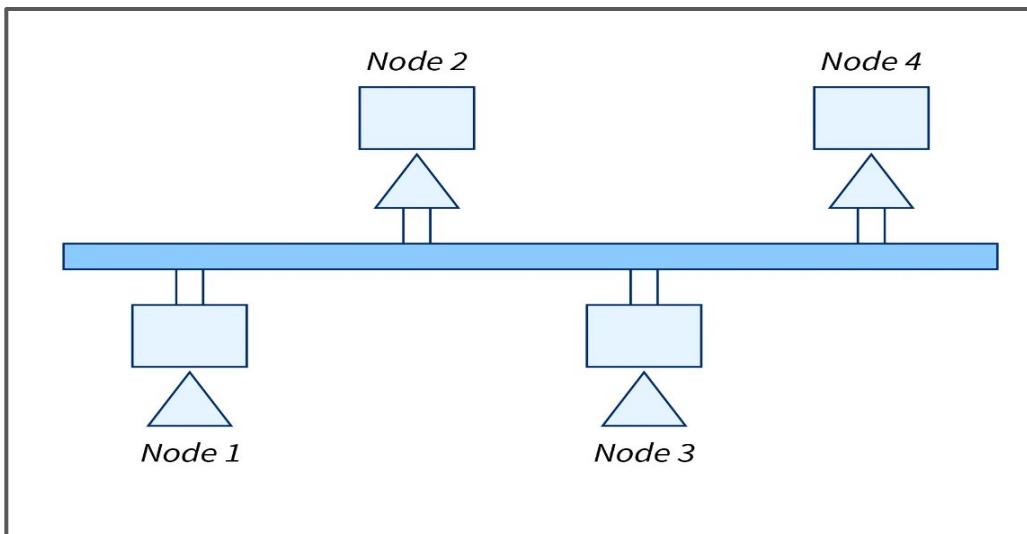
- Vulnerable Time = T_{fr}
Where $T_{fr} \rightarrow$ Frame Transmission Time
- Throughput = $S = G \times e^{-G}$
- Maximum throughput = **0.368** for $G=1$

Pure Aloha	Slotted Aloha
Any station can send data at any moment.	Any station can send data at the start of any time period.
The time is not globally synchronised and is continuous.	The time is discrete and synced worldwide.
Vulnerable time for a collision to occur = $2 \times T_t$.	Vulnerable time for a collision to occur = T_t
The biggest benefit of pure aloha is its ease of implementation.	The primary benefit of slotted aloha is that it lowers collisions by half and doubles the efficiency of pure aloha.

Throughput versus offered traffic for ALOHA System



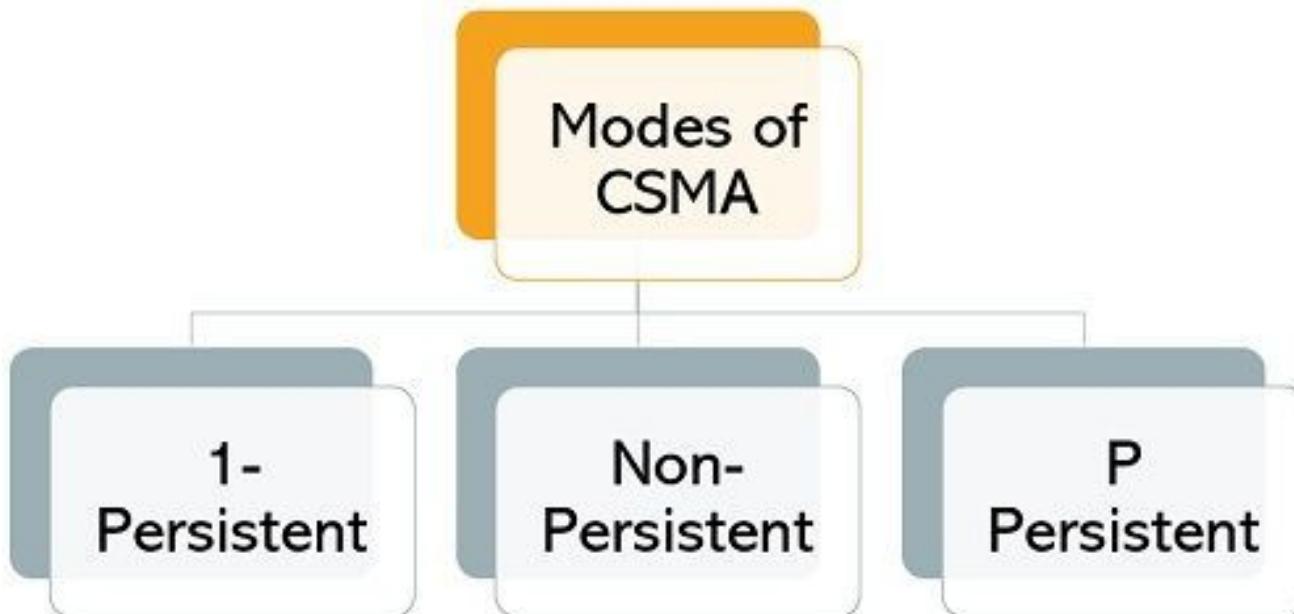
CSMA



Carrier Sense Multiple Access

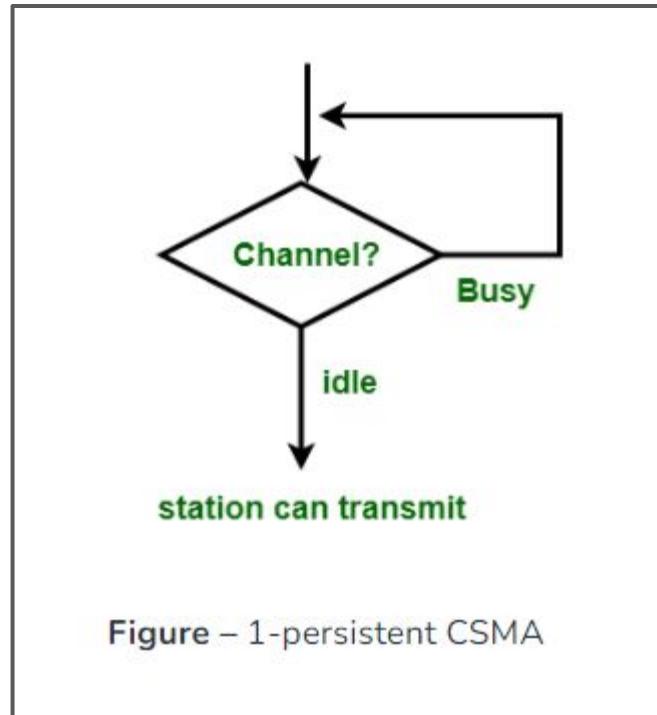
- Carrier Sense Multiple Access ensures fewer collisions as the station is required to first sense the medium (for idle or busy) before transmitting data.
- If it is idle then it sends data, otherwise it waits till the channel becomes idle.
- **Principle of CSMA :** “ Sense before transmit ” or “ Listen before talk ”.
- **Carrier Busy** = Transmission is taking place
- **Carrier Idle** = No transmission currently taking place
- However there is still chance of collision in CSMA due to **propagation delay**.
 - For example, if station A wants to send data, it will first sense the medium. If it finds the channel idle, it will start sending data. However, by the time the first bit of data is transmitted (delayed due to propagation delay) from station A, if station B requests to send data and senses the medium it will also find it idle and will also send data. This will result in collision of data from station A and B.

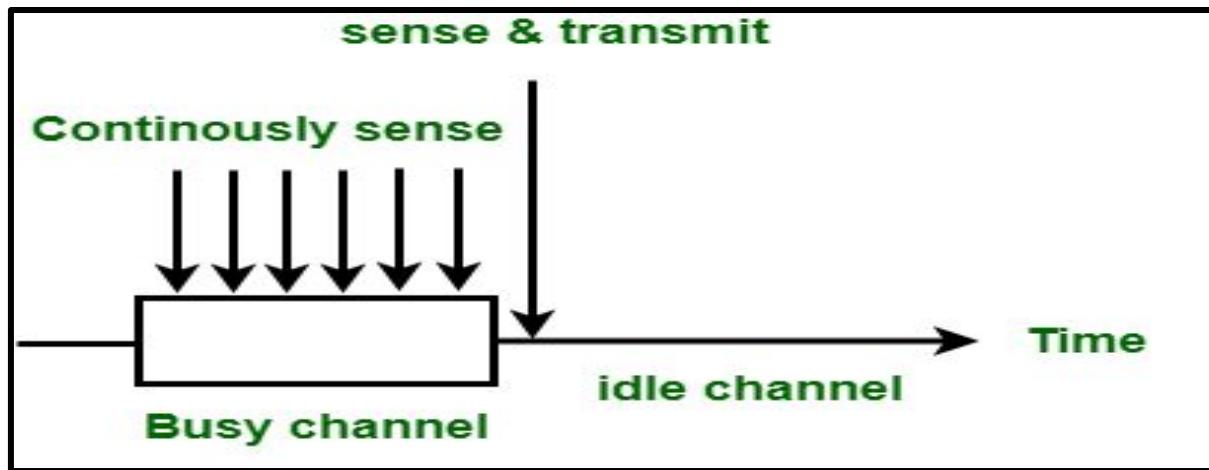
Types of CSMA



1-Persistent CSMA

- In this, if the station wants to transmit the data. Then the station first senses the medium.
- If the medium is busy then the station waits until the channel becomes idle. And the station **continuously senses the channel** until the medium becomes idle.
- If the station detected the channel as idle then the station will immediately send the data frame with 1 probability that's why the name of this method is 1-persistent.
- Refer to the below image to show the flow diagram of the 1-persistent method of CSMA



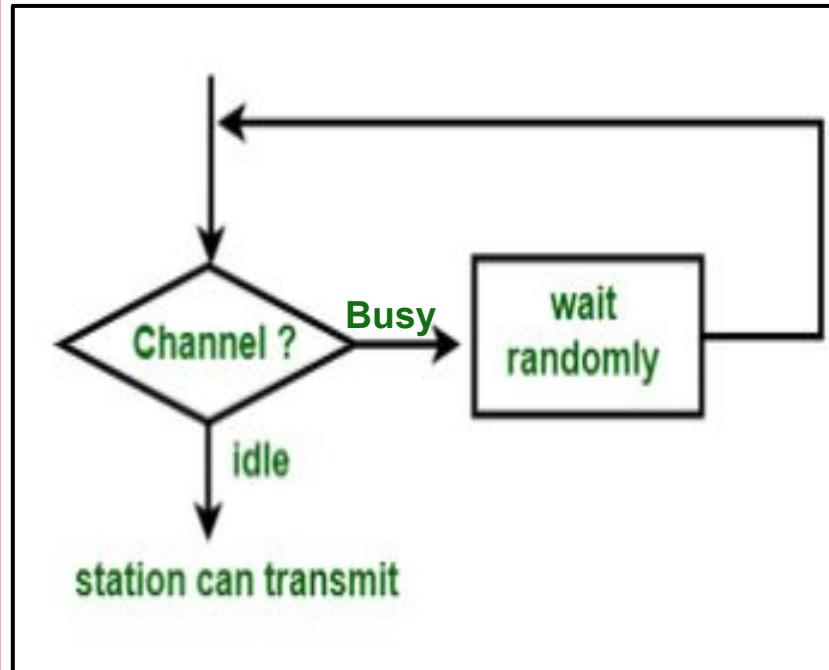


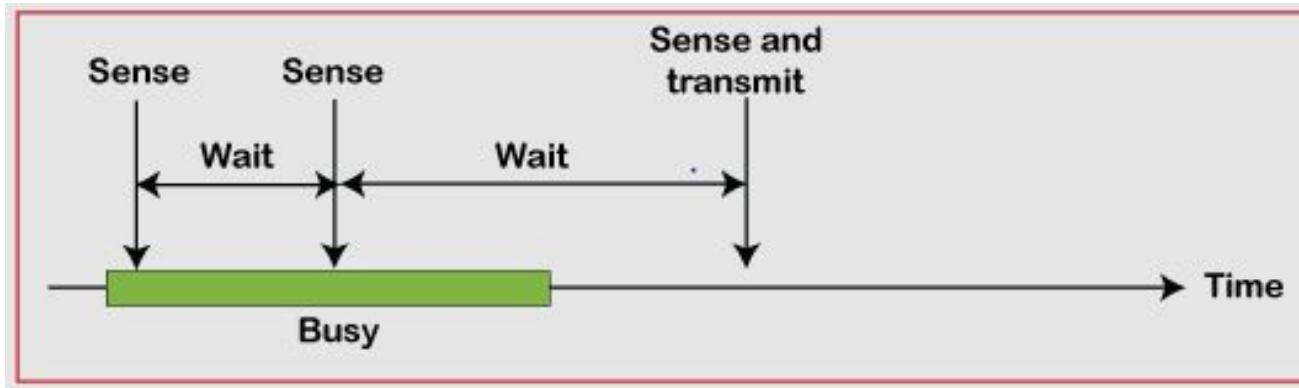
Problems :

- In this method there is **a high possibility of collision** as two or more station sense the channel idle at the same time and transmits data simultaneously which may lead to a collision.
- In this method, once the station finds that the medium is idle then it immediately sends the frame.
- By using this method there are higher chances for collision because **it is possible that two or more stations find the shared medium idle at the same time** and then they send their frames immediately.

Non-Persistent CSMA

- If the station wants to transmit the data then first of all it will sense the medium.
- If the medium is idle then the station will immediately send the data.
- Otherwise, if the medium is busy then the station **waits for a random amount of time and then again senses the channel** after waiting for a random amount of time.
- In Non-persistent there is less chance of collision in comparison to the 1-persistent method as this station will not continuously sense the channel but the channel waiting for a random amount of time.





Advantage of non-persistent

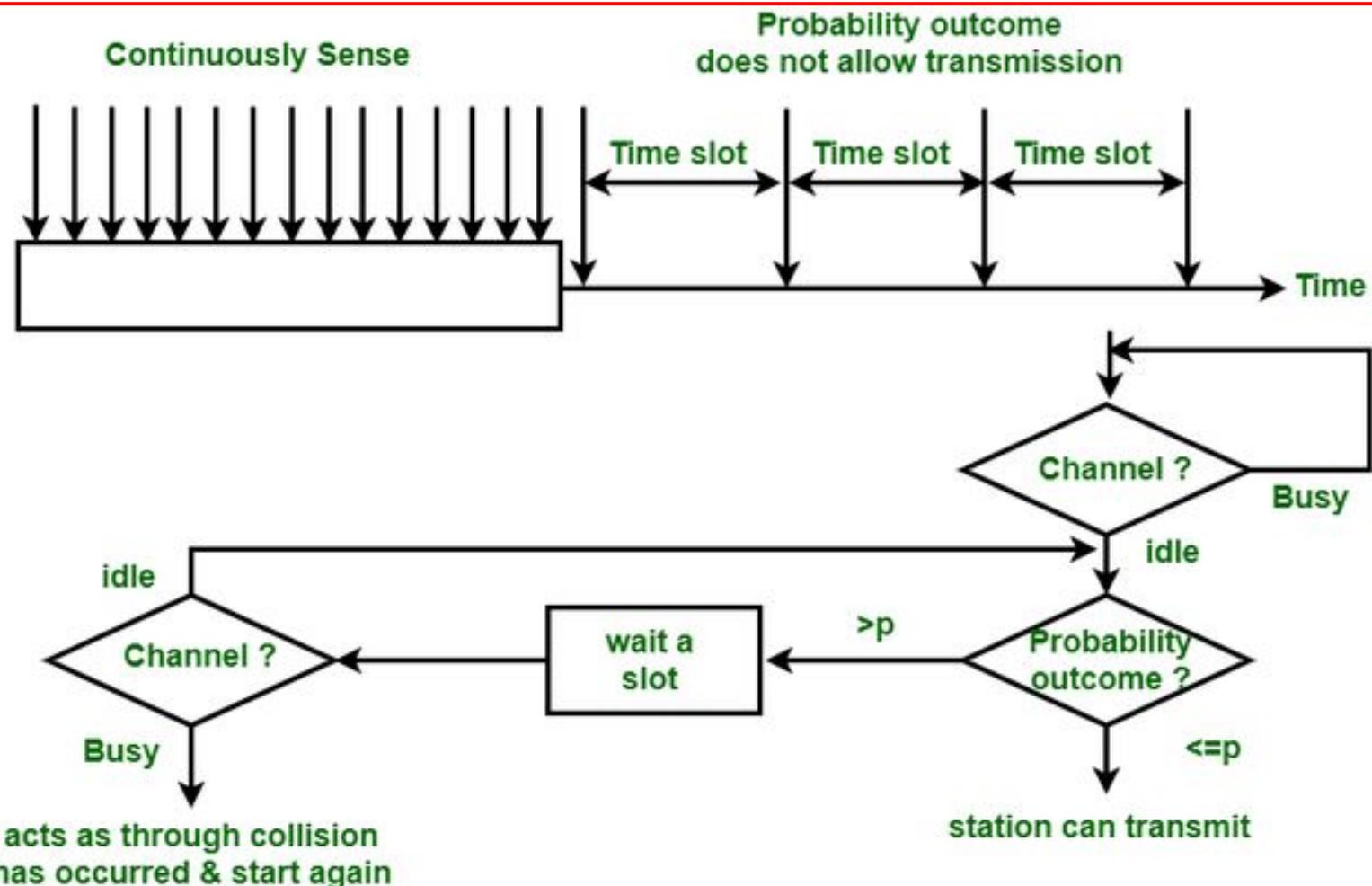
- It reduces the chance of collision because the stations wait a random amount of time.
- It is unlikely that two or more stations will wait for same amount of time and will retransmit at the same time.

Disadvantage of non-persistent

- It reduces the efficiency of network because the channel remains idle when there may be stations with frames to send.
- This is due to the fact that the stations wait a random amount of time after the collision.

P-Persistent

- It is the combination of **1-Persistent** and **Non-persistent** modes.
- This method is used when **channel has time slots** such that the time slot duration is equal to or greater than the maximum propagation delay time.
- Whenever a station becomes ready to send, it senses the channel.
- If channel is **busy**, station **waits until next slot**.
- If channel is **idle**, it **transmits with a probability p**.
- With the probability $q=1-p$, the station then waits for the beginning of the next time slot.
- If the next slot is also idle, it either transmits or waits again with probabilities p and q.
- This process is repeated till either frame has been transmitted or another station has begun transmitting.



Carrier Sense Multiple Access with Collision Detection (CSMA/CD)

CSMA/CD stands for Carrier Sense Multiple Access with Collision Detection.

It is a protocol used in **Ethernet networks** to control access to the shared transmission medium and manage collisions when multiple devices attempt to transmit data simultaneously.

Here's how CSMA/CD works:

Carrier Sense:

Before a device starts transmitting data, it listens to the network to check if the medium is idle. If it detects other devices transmitting, it waits for the network to become idle.

Multiple Access:

If the medium is **idle**, the device can start transmitting its data.

Collision Detection:

- ★ Sender transmits its data on the link. CSMA/CD does not use an ‘acknowledgment’ system.
- ★ It checks for successful and unsuccessful transmissions through collision signals.
- ★ During transmission, if a **collision signal** is received by the node, transmission is stopped.
- ★ The station then transmits a **jam signal** onto the link and waits for **random time** intervals before it re-sends the frame.
- ★ After some random time, it again attempts to transfer the data and repeats the above process.
- ★ The station’s hardware must listen to the channel while it is transmitting.
- ★ **If the signal it reads back is different from the signal it is putting out, it knows that a collision is occurring.**

Backoff and Retransmission:

After a collision is detected, the device waits for a random amount of time (backoff) before attempting to retransmit the data. The random backoff helps to reduce the probability of another collision. The device retransmits the data once the backoff time elapses and the medium is idle.

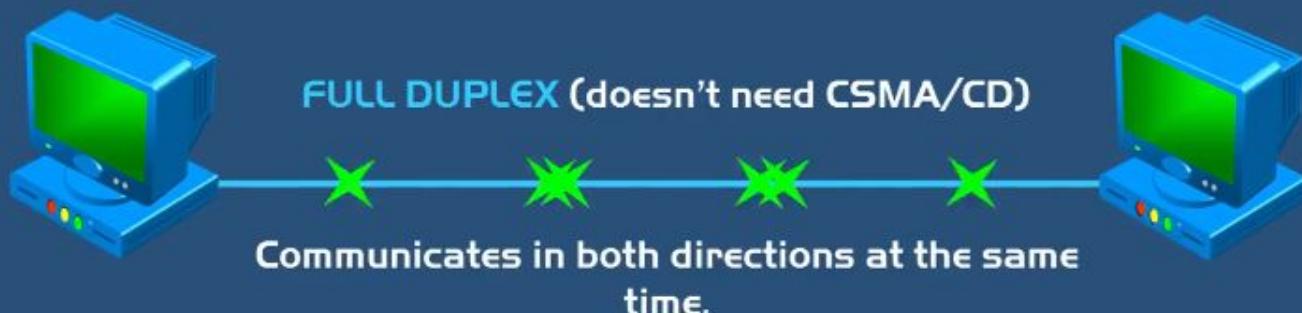
- CSMA/CD was **widely used** in early Ethernet networks based on **coaxial or twisted-pair cables**.
- However, with the advent of faster Ethernet technologies and the prevalence of switched networks, CSMA/CD has become less common.

CSMA/CD

Carrier Sense Multiple Access
with Collision Detection

CSMA/CD was used on early Ethernet networks.

Not as relevant today.



The Conceptual Model for CSMA/CD

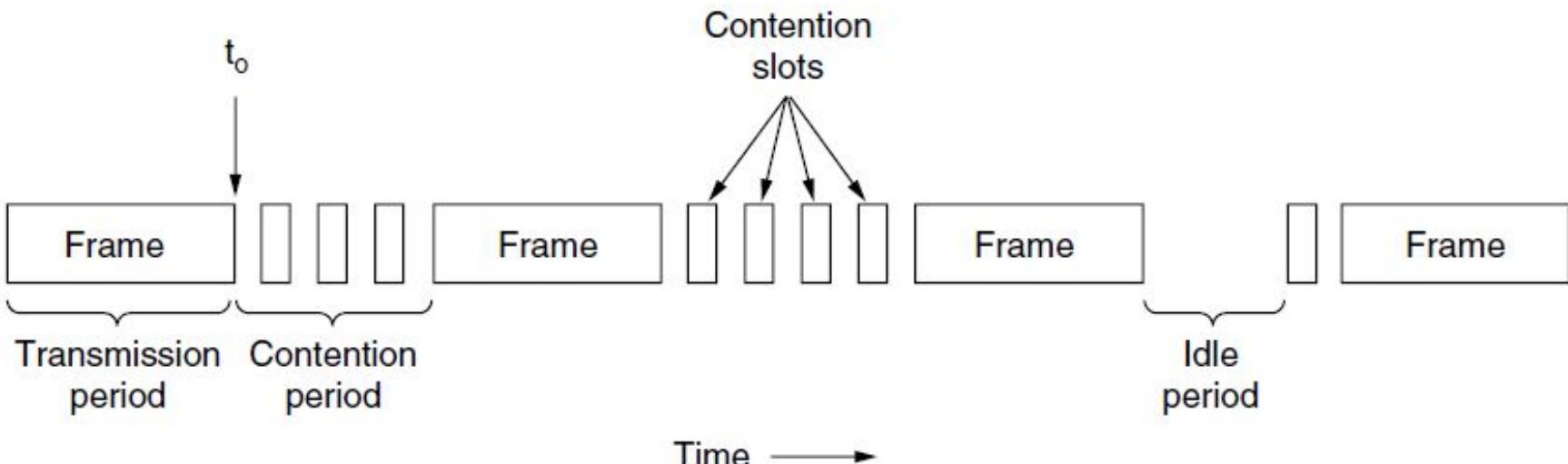


Figure 4-5. CSMA/CD can be in contention, transmission, or idle state.

- ★ CSMA/CD, as well as many other LAN protocols, uses the conceptual model of Fig. 4-5.
- ★ At the point marked t_0 , a station has finished transmitting its frame.
- ★ Any other station having a frame to send may now attempt to do so. If two or more stations decide to transmit simultaneously, there will be a collision.
- ★ If a station detects a collision, it aborts its transmission, waits a random period of time, and then tries again (assuming that no other station has started transmitting in the meantime).
- ★ Therefore, our model for CSMA/CD will consist of alternating **contention and transmission periods, with idle periods** occurring when all stations are quiet (e.g., for lack of work).

Contention Periods:

Contending devices (those wanting to transmit) use contention periods. These contention periods are time intervals during which **a device listens to the network to determine if it's clear for transmission.**

Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA)

CSMA/CA

Carrier Sense Multiple Access
with Collision Avoidance

Used on wireless networks.



- CSMA/CA stands for *Carrier Sense Multiple Access with Collision Avoidance*.
- It is a protocol used in **wireless networks**, particularly **Wi-Fi networks**, to manage access to the shared wireless medium and prevent collisions.
- In **wired networks**, if a collision has occurred then the energy of the received signal almost doubles, and the station can sense the possibility of collision.
- In the case of wireless networks, most of the energy is used for transmission, and the energy of the received signal increases by only 5-10% if a collision occurs. It can't be used by the station to sense collision. Therefore **CSMA/CA has been specially designed for wireless networks**.

Here's how CSMA/CA works:

1. **Carrier Sense:** Before a device starts transmitting data, it listens to the wireless medium to check if it is idle. If the medium is busy, the device waits for it to become idle.

2. Collision Avoidance:

Rather than relying on collision detection, CSMA/CA uses a technique called collision avoidance. The device sends **a Request to Send (RTS)** frame to the Wireless Access Point(WAP), indicating its intention to transmit data. The RTS frame includes information about the duration of the transmission.

3. Clear to Send (CTS): Upon receiving the RTS frame, the access point responds with a Clear to Send (CTS) frame, granting permission to the device to transmit data. The CTS frame also includes the duration of the transmission.

4. Transmission: Once the device receives the CTS frame, it can start transmitting its data. Other devices within range of the access point listen to the CTS frame and defer their own transmissions to avoid collisions.

UNIT-3

Network Layer

B SAI BABA,M.Tech(Ph.D),VIT,Bhimavaram

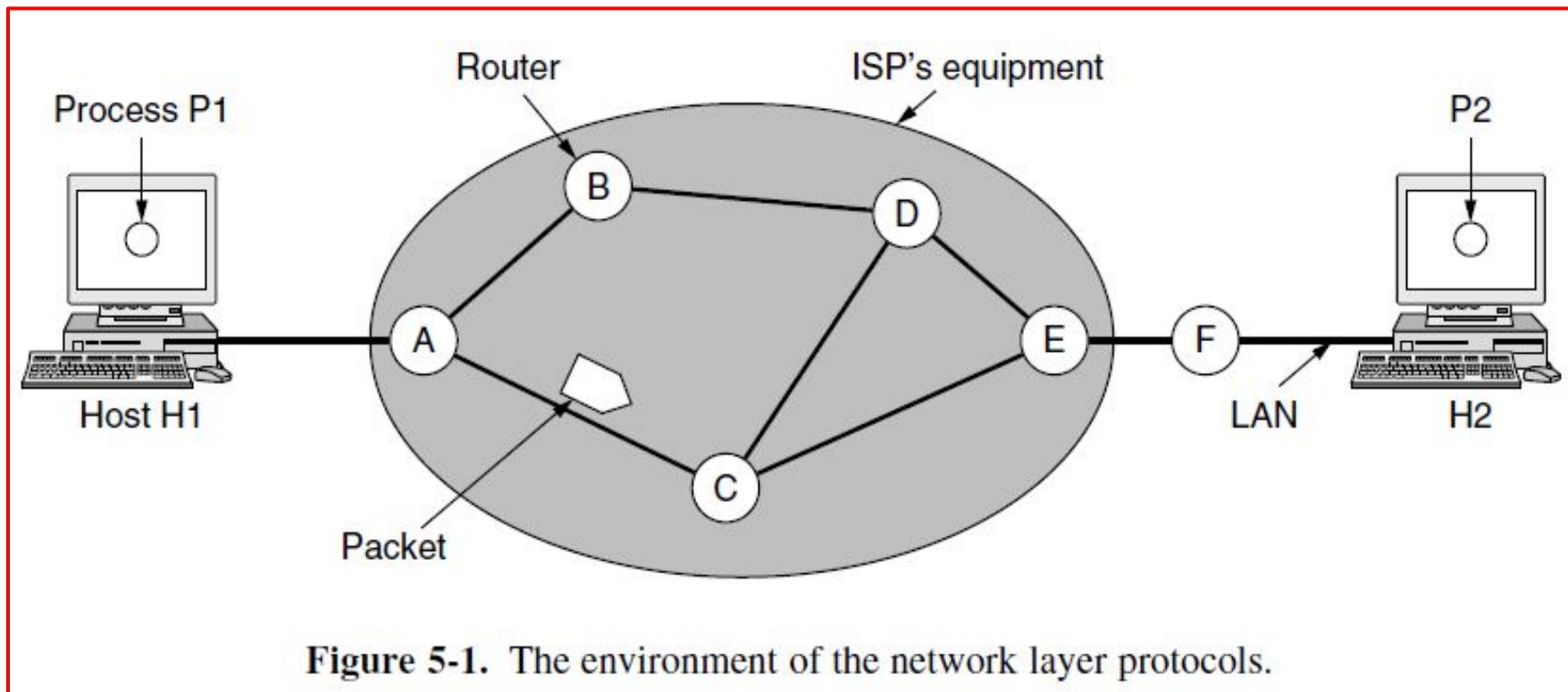
Syllabus

- ★ Network Layer:
 - Network layer design issues
- ★ Routing Algorithms
- ★ Congestion Control Algorithms
- ★ Internet working
- ★ The Network layer in the internet (IPv4 and IPv6)
- ★ Quality of Service

Network Layer Design Issues

- ★ **The Network Layer or Layer 3** of the OSI (Open Systems Interconnection) model.
- ★ The design issues can be elaborated as–
 1. **Store – and – Forward Packet Switching**
 2. **Services to Transport Layer**
 - a. **Providing Connection Oriented Service**
 - b. **Providing Connectionless Service**
 3. **Routing**
 4. **Congestion Control**
 5. **Logical Addressing(IP Address)**

Store – and – Forward Packet Switching



Store – and – Forward Packet Switching

- ★ The network layer operates in an environment that uses store and forward packet switching.
- ★ The node which has a packet to send, **delivers it to the nearest router.**
- ★ The packet is stored in the router until it has fully arrived and its checksum is verified for error detection.
- ★ Once, this is done, **the packet is forwarded to the next router.**
- ★ Since, each router needs to store the entire packet before it can forward it to the next hop, the mechanism is called **store – and – forward switching.**

Services to Transport Layer

- ★ The Network Layer provides service to its immediate upper layer, namely **Transport Layer**, through the network – transport layer interface.
- ★ **The two types of services provided are –**
 1. **Connection – Oriented Service :**

In this service, **a path is setup between the source and the destination**, and all the data packets belonging to a message are routed along this path.

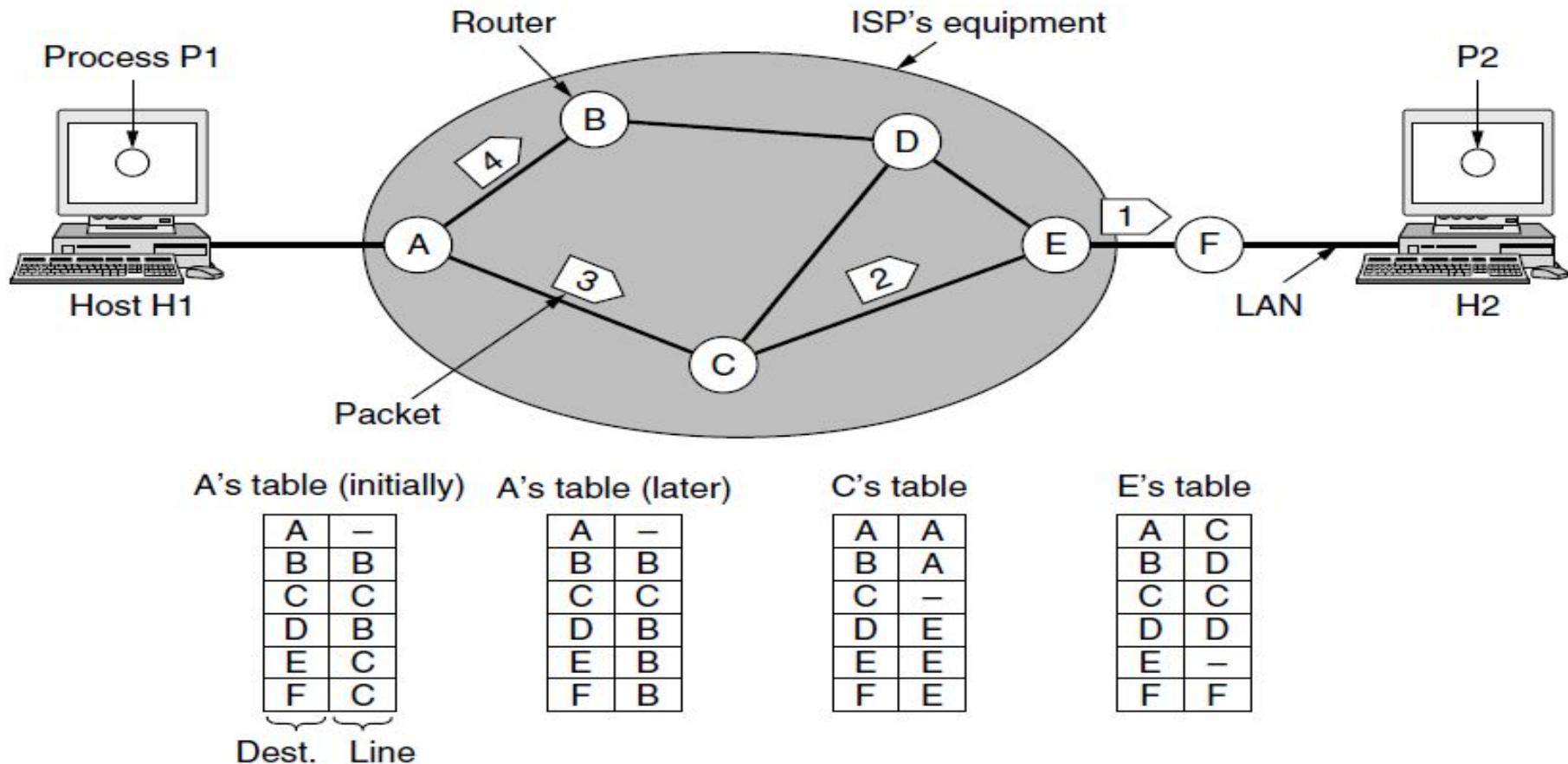
2. **Connectionless Service :**

In this service, each packet of the message is considered as **an independent entity** and is **individually routed from the source to the destination**.

Providing Connectionless Service

- ★ If connectionless service is offered, packets are injected into the network individually and routed independently of each other.**No advance setup is needed.**
- ★ In this context, the packets are frequently called **datagrams** and the network is called **a datagram network**.
- ★ In this section, we will examine datagram networks :

Figure 5-2. Routing within a datagram network



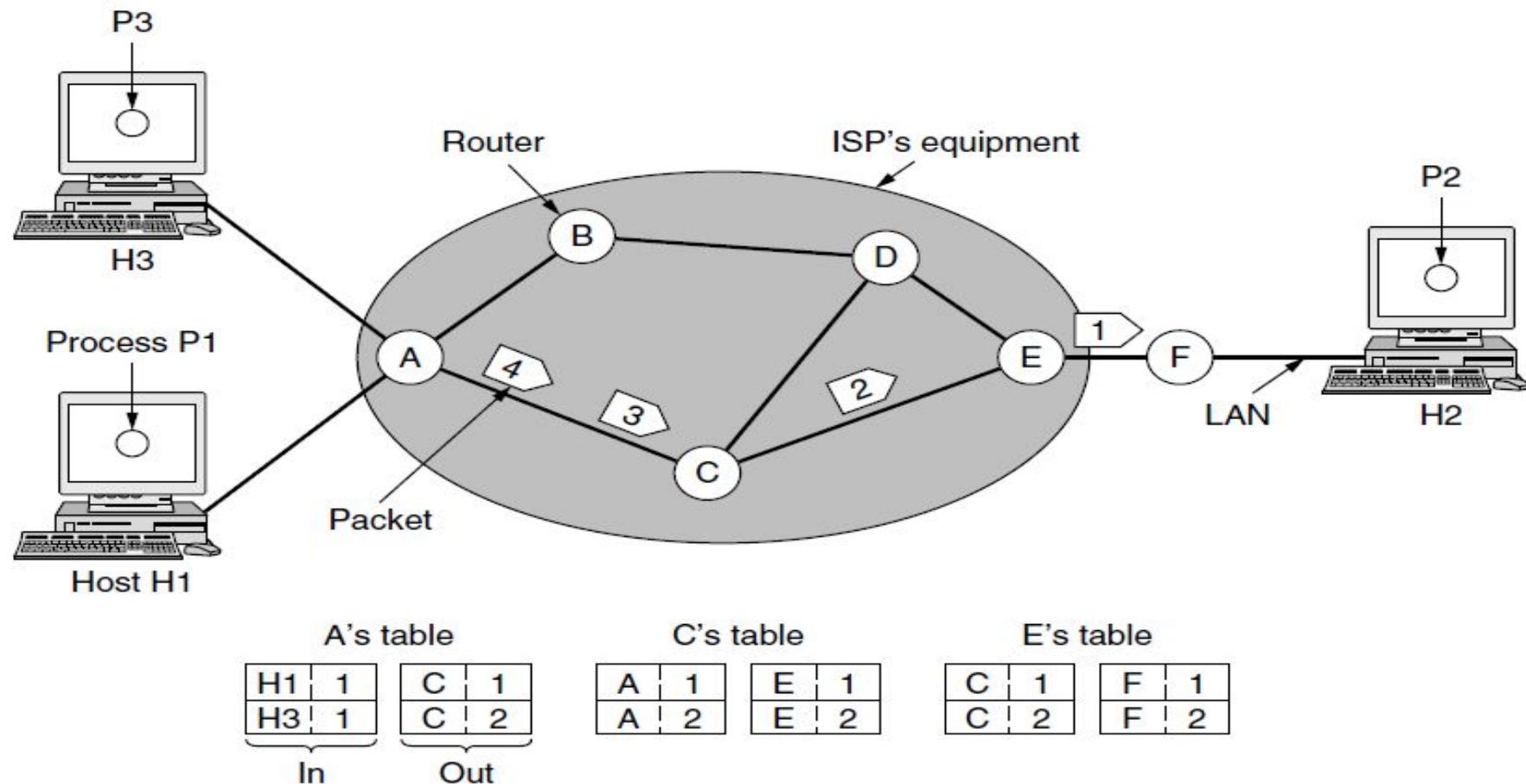
- ★ Let us assume for this example that the message is four times longer than the maximum packet size, so the network layer has to break it into four packets, 1, 2, 3, and 4, and send each of them in turn to router A using some point-to-point protocol.
- ★ At this point the ISP takes over. Every **router** has **an internal table** telling it where to send packets for each of the possible destinations.
- ★ **Each table entry is a pair** consisting of **a destination** and **the outgoing line to use for that destination**. Only directly connected lines can be used.
- ★ For example, in Fig. 5-2, **A has only two outgoing lines—to B and to C**—so every incoming packet must be sent to one of these routers, even if the ultimate destination is to some other router.
- ★ A's initial routing table is shown in the figure under the label “initially.”

- ★ At A, packets 1, 2, and 3 are stored briefly, having arrived on the incoming link and had their checksums verified. Then each packet is forwarded according to A's table, onto the outgoing link to C within a new frame.
- ★ Packet 1 is then forwarded to E and then to F. When it gets to F, **it is sent within a frame** over the LAN to H2. **Packets 2 and 3** follow the same route.
- ★ However, something different happens to **packet 4**.
- ★ When it gets to A it is sent to router B, even though it is also destined for F.
- ★ For some reason, A decided to send packet 4 via a different route than that of the first three packets. Perhaps it has learned of **a traffic jam** somewhere along **the ACE path** and updated its routing table, as shown under the label "later."
- ★ The algorithm that manages the tables and makes the routing decisions is called **the routing algorithm**.

Providing Connection Oriented Service

- ★ If connection-oriented service is used, **a path** from the source router all the way to the destination router **must be established before any data packets can be sent.**
- ★ This connection is called a **VC (virtual circuit),**
- ★ The network is called **a virtual-circuit network.**

Figure 5-3. Routing within a virtual-circuit network



- ★ The idea behind virtual circuits is **to avoid having to choose a new route** for every packet sent, as in Fig. 5-2.
- ★ Instead, when a connection is established, a route from the source machine to the destination machine is chosen as part of the connection setup and stored in tables inside the routers.
- ★ That route is used for all traffic flowing over the connection, exactly the same way that the telephone system works.
- ★ When the connection is released, the virtual circuit is also terminated.
- ★ **With connection-oriented service, each packet carries an identifier telling which virtual circuit it belongs to.**
- ★ As an example, consider the situation shown in Fig. 5-3. Here, host H1 has established connection 1 with host H2. This connection is remembered as the first entry in each of the routing tables.

- ★ The first line of A's table says that if a packet bearing connection identifier 1 comes in from H1, it is to be sent to router C and given connection identifier 1.
- ★ Similarly, the first entry at C routes the packet to E, also with connection identifier 1.
- ★ Now let us consider what happens if H3 also wants to establish a connection to H2.
- ★ It chooses connection identifier 1 (because it is initiating the connection and this is its only connection) and tells the network to establish the virtual circuit. This leads to the second row in the tables.
- ★ Note that we have a conflict here because although A can easily distinguish connection 1 packets from H1 from connection 1 packets from H3, C cannot do this. For this reason, A assigns a different connection identifier to the outgoing traffic for the second connection.
- ★ Avoiding conflicts of this kind is why routers need the ability to replace connection identifiers in outgoing packets. In some contexts, this process is called **label switching**.

Comparison of Virtual Circuit and Datagram Network

Issue	Datagram network	Virtual-circuit network
Circuit setup	Not needed	Required
Addressing	Each packet contains the full source and destination address	Each packet contains a short VC number
State information	Routers do not hold state information about connections	Each VC requires router table space per connection
Routing	Each packet is routed independently	Route chosen when VC is set up; all packets follow it
Effect of router failures	None, except for packets lost during the crash	All VCs that passed through the failed router are terminated
Quality of service	Difficult	Easy if enough resources can be allocated in advance for each VC
Congestion control	Difficult	Easy if enough resources can be allocated in advance for each VC

Routing Algorithms

Routing Algorithm

- ★ In order to transfer the packets from source to the destination, the network layer must determine **the best route** through which packets can be transmitted.
- ★ Whether the network layer provides **datagram service or virtual circuit service**, the main job of the network layer is to provide **the best route**. The routing protocol provides this job.
- ★ The routing protocol is a routing algorithm that provides the best path from the source to the destination. The best path is the path that has the "**least-cost path**" from source to the destination.
- ★ Routing is the process of forwarding the packets from source to the destination but **the best route to send the packets is determined by the routing algorithm**.

Optimality Principle

Introduction :

- ★ A general statement is made about optimal routes without regard to network topology or traffic. This statement is known as the optimality principle(Bellman,1975).

Statement of the optimality principle :

- ★ It states that if the router J is on the optimal path from router I to router K, then the optimal path from J to K also falls along the same route. Call the route from I to J, r1 and the rest of the route r2. it could be concatenated with r1 to improve the route from I to K, contradicting our statement that r1r2 is optimal only if a route better than r2 existed from J to K.

Introduction :

- ★ A general statement is made about optimal routes without regard to network topology or traffic. This statement is known as the optimality principle(Bellman,1975).

Explanation:

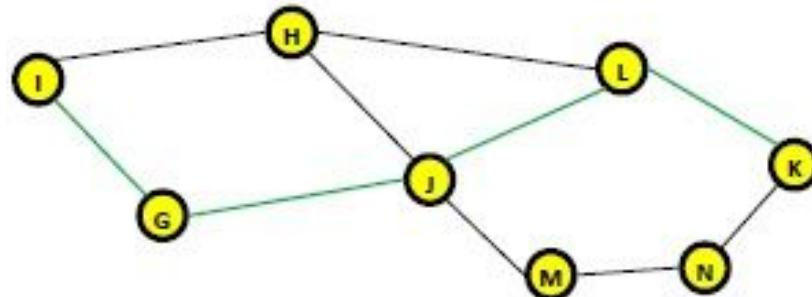
- ★ The purpose of a routing algorithm at a router is to decide which output line an incoming packet should go.
- ★ **The optimality principle** from a particular router to another may be the **least cost path, the least distance path, the least time path, the least hops path** or a combination of any of the above.

The optimality principle can be logically proved as follows –

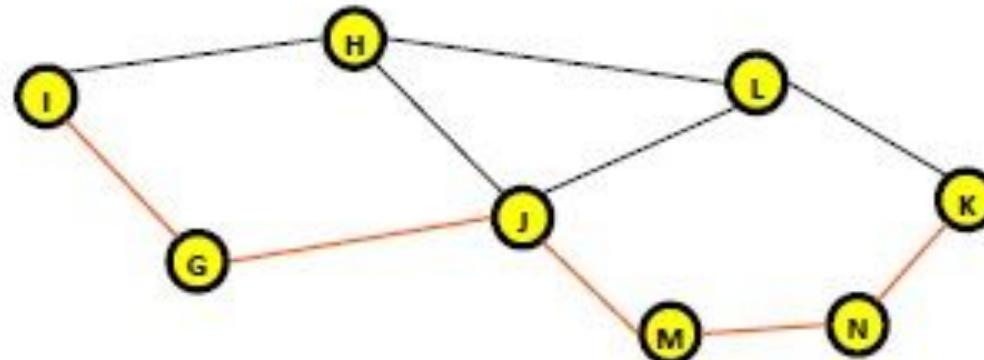
If the router J is on the optimal path from router I to router K, If a better route could be found between router J and router K, the path from router I to router K via J would be updated via this route. Thus, the optimal path from J to K will again lie on the optimal path from I to K.

Example

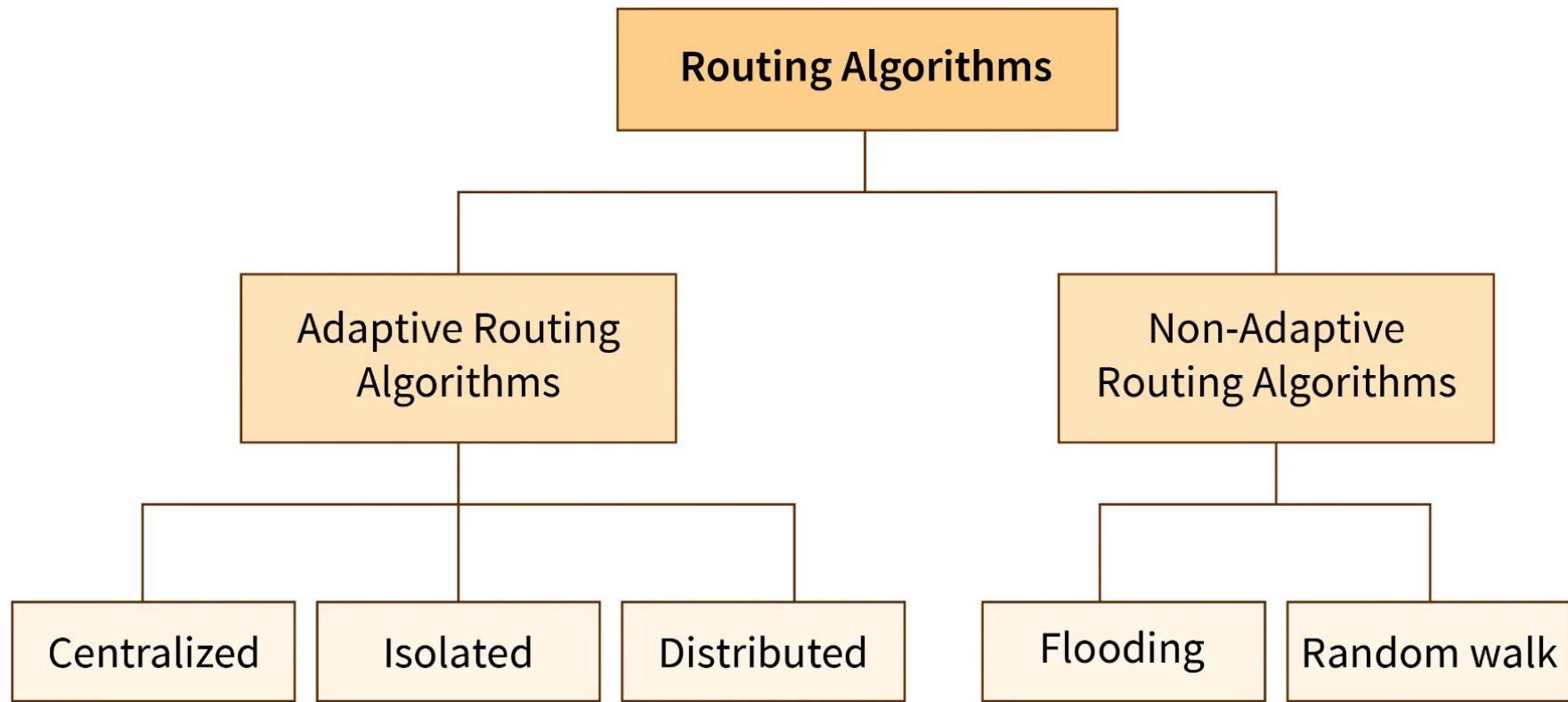
Consider a network of routers, {G, H, I, J, K, L, M, N} as shown in the figure. Let the optimal route from I to K be as shown via the green path, i.e. via the route I-G-J-L-K. According to the optimality principle, the optimal path from J to K will be along the same route, i.e. J-L-K.



Now, suppose we find a better route from J to K is found, say along J-M-N-K. Consequently, we will also need to update the optimal route from I to K as I-GJ- M-N-K, since the previous route ceases to be optimal in this situation. This new optimal path is shown line orange lines in the following figure –



Types of Routing Algorithm



Adaptive Routing Algorithm

- ★ Adaptive routing algorithm is also called **a dynamic routing algorithm**.
- ★ In this algorithm, **the routing decisions** are made based on **network traffic and topology**.
- ★ The parameters which are used in adaptive routing algorithms are **distance, hop, estimated transit time**.
- ★ The adaptive routing algorithm is of **three types** –
 - **Centralized algorithm**
 - **Isolation algorithm**
 - **Distributed algorithm**

Centralized algorithm:

- ★ In centralized routing, **one centralized node** has the total network information and takes the routing decisions.
- ★ It finds the least-cost path between source and destination nodes by using **global knowledge about the network**. So, it is also known as global routing algorithm.
- ★ The advantage of this routing is that only the central node is required to store network information and so the resource requirement of the other nodes may be less.
- ★ Eg: **Link state routing algorithm**

Isolated algorithm:

- ★ This algorithm procures the routing information by using **local information** instead of gathering information from other nodes.

Distributed algorithm:

- ★ This is a **decentralized algorithm** where each node receives information from its neighbouring nodes and takes the decision based upon the received information.
- ★ The least-cost path between source and destination is computed **iteratively in a distributed manner**.
- ★ An advantage is that each node can **dynamically change routing decisions** based upon the changes in the network.
- ★ Example : **distance vector routing algorithm.**

Non-adaptive Routing Algorithm

- ★ Non-adaptive routing algorithm is also called **a static routing algorithm**.
- ★ In a non-adaptive routing algorithm, the routing decisions are **not made based on network traffic and topology**.
- ★ This algorithm is used by static routing.
- ★ Non-adaptive routing algorithms are **simple** as compared to Adaptive routing algorithms in terms of **complexity**.
- ★ The non-adaptive routing algorithm is of **two types** –
 - **Flooding**
 - **Random walks**

Shortest Path Algorithm

Introduction:

- ★ **Dijkstra's algorithm and the Bellman-Ford algorithm** are two different algorithms used to find the shortest path in a weighted graph.

Dijkstra's algorithm	Bellman-Ford algorithm
It works only for graphs with non-negative edge weights .	It works for graphs with both non-negative and negative edge weights .

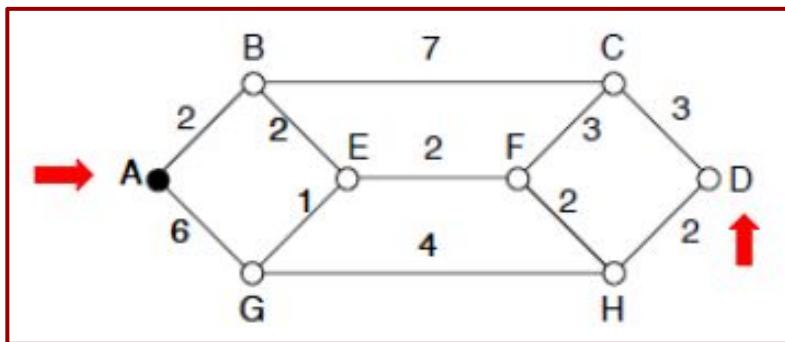
Dijkstra's algorithm:

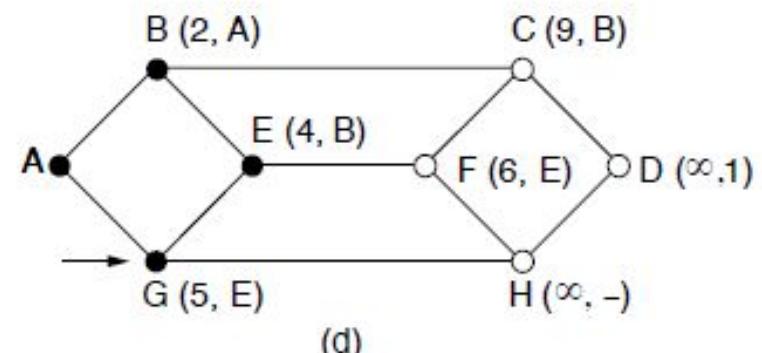
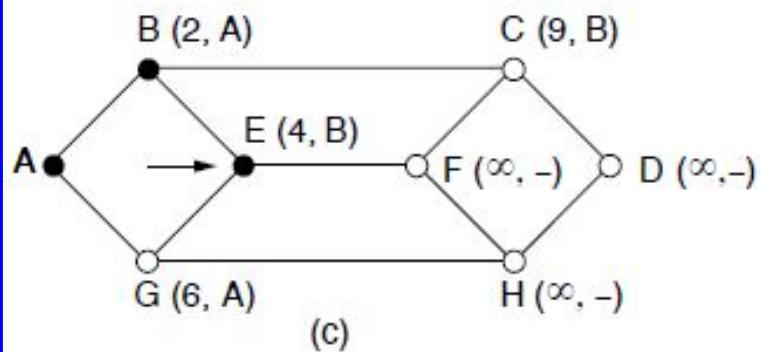
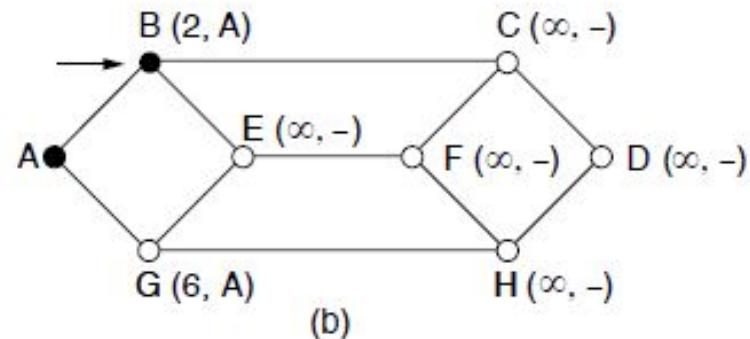
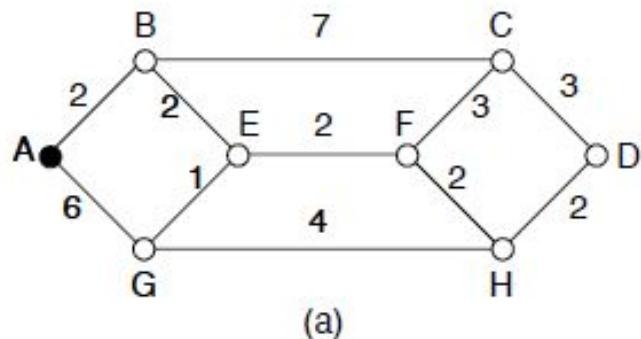
- ★ The shortest path algorithm was first introduced by **Edsger W. Dijkstra** in 1956.
- ★ This algorithm, commonly known as **Dijkstra's algorithm**, is used to find the shortest path between nodes in a graph, particularly in **weighted graphs** where each **edge has a non-negative weight**.
- ★ Dijkstra's algorithm is widely used in various applications, including **computer networking, transportation systems**, and more, to find the shortest path from a source node to all other nodes in the graph.

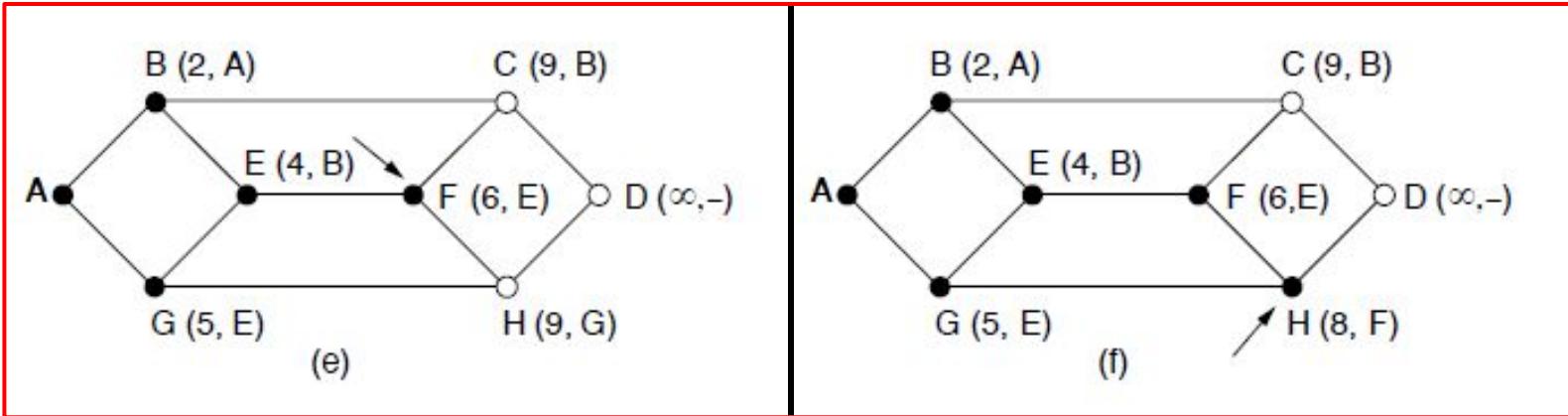
Procedure of Shortest Path Algorithm

- ★ Initially mark all nodes (except source) with infinite distance.
 - working node = source node
 - Sink node = destination node
- ★ While the working node is not equal to the sink
 1. Mark the working node as permanent.
 2. Examine all adjacent nodes in turn
 - If the sum of label on working node plus distance from working node to adjacent node is less than current labeled distance on the adjacent node, this implies a shorter path. Re label the distance on the adjacent node and label it with the node from which the probe was made.
 - 3. Examine all tentative nodes (not just adjacent nodes) and mark the node with the smallest labeled value as permanent. This node becomes the new working node.
- ★ Reconstruct the path backwards from sink to source

- The idea is to build a graph of the network, with each node of the graph representing a router and each edge of the graph representing a communication line, or link.
- To choose a route between a given pair of routers, the algorithm just finds the shortest path between them on the graph.
- The concept of a shortest path deserves some explanation.
 - One way of measuring path length is the **number of hops**. Using this metric, the paths ABC and ABE in Fig. are equally long.
 - Another metric is **the geographic distance in kilometers**, in which case ABC is clearly much longer than ABE







- The first six steps used in computing the shortest path from A to D.
- The arrows indicate the working node.

Example :2

Dijkstra Algorithm (Shortest Path Algorithm)

- How Routers Decide Shortest Path Using dijkstra Algorithm ?

Graph Step 1 2 3 4 5 6 7 8 9

```
graph LR; a((a)) ---|2| c((c)); a ---|4| b((b)); c ---|1| b; c ---|8| d((d)); c ---|10| e((e)); b ---|5| d; d ---|6| z((z)); d ---|2| e; e ---|5| z;
```

Dijkstra's Algorithm

What is the shortest path to travel from A to Z?

Step 1

Graph Step 1 2 3 4 5 6 7 8 9

```
graph LR; a((a)) ---|2| c((c)); a ---|4| b((b)); c ---|1| b; c ---|8| d((d)); c ---|10| e((e)); b ---|5| d; d ---|6| z((z)); e ---|5| z;
```

Node	Status	Shortest Distance From A	Previous Node
A	Current Node	0	
B		∞	
C		∞	
D		∞	
E		∞	
Z		∞	

Start by setting the starting node (A) as the current node.

Step 2

Graph Step 1 2 3 4 5 6 7 8 9

Node	Status	Shortest Distance From A	Previous Node
A	Current Node	0	
B		∞ 4	A
C		∞ 2	A
D		∞	
E		∞	
Z		∞	

Check all the nodes connected to A and update their "Distance from A" and set their "previous node" to "A".

Step 3

Graph Step 1 2 3 4 5 6 7 8 9

Node	Status	Shortest Distance From A	Previous Node
A	Visited Node	0	
B		∞	A
C	Current Node	2	A
D		∞	
E		∞	
Z		∞	

Set the current node (A) to "visited" and use the closest unvisited node to A as the current node (e.g. in this case: Node C).

Step 4

Graph Step 1 2 3 4 5 6 7 8 9

Node	Status	Shortest Distance From A	Previous Node
A	Visited Node	0	
B		4 $2+1=3$	C
C	Current Node	2	A
D		∞ $2+8=10$	C
E		∞ $2+10=12$	C
Z		∞	

Check all unvisited nodes connected to the current node and add the distance from A to C to all distances from the connected nodes. Replace their values only if the new distance is lower than the previous one.

C → B: $2 + 1 = 3 < 4$ – Change Node B
 C → D: $2 + 8 = 10 < \infty$ – Change Node D
 C → E: $2 + 10 = 12 < \infty$ – Change Node E

Step 5

Graph Step 1 2 3 4 5 6 7 8 9

Set the current node C status to Visited.
We then repeat the same process always picking the closest unvisited node to A as the current node.
In this case node B becomes the current node.

Node	Status	Shortest Distance From A	Previous Node
A	Visited Node	0	
B	Current Node	3	C
C	Visited Node	2	A
D		10	C
E		12	C
Z		∞	

Step 6

Graph Step 1 2 3 4 5 6 7 8 9

Node	Status	Shortest Distance From A	Previous Node
A	Visited Node	0	
B	Current Node	3	C
C	Visited Node	2	A
D		10 3+5=8	B
E		12	C
Z		∞	

B → D $3+5 = 8 < 10$ – Change Node D

Next "Current Node" will be D as it has the shortest distance from A amongst all unvisited nodes.

Step 7

Graph Step 1 2 3 4 6 6 7 8 9

Node	Status	Shortest Distance From A	Previous Node
A	Visited Node	0	
B	Visited Node	3	C
C	Visited Node	2	A
D	Current Node	8	B
E		12 $8 + 2 = 10$	D
Z		∞ $8 + 6 = 14$	D

D \rightarrow E $8+2=10 \neq 12$ – Change Node E
D \rightarrow Z $8+6=14 < \infty$ – Change Node Z

We found a path from A to Z but it may not be the shortest one yet. So we need to carry on the process.

Next "Current Node": E

Step 8

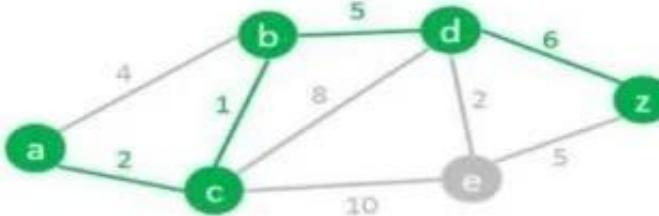
Graph Step 1 2 3 4 5 6 7 8 9

Node	Status	Shortest Distance From A	Previous Node
A	Visited Node	0	
B	Visited Node	3	C
C	Visited Node	2	A
D	Visited Node	8	B
E	Current Node	10	D
Z		14 $10 + 5 = 15$	D

$E \rightarrow Z: 10 + 5 = 15 > 14$ – We do not change node Z.

Step 9

Graph Step 1 2 3 4 5 6 7 8 9



Node	Status	Shortest Distance From A	Previous Node
A	Visited Node	0	
B	Visited Node	3	C
C	Visited Node	2	A
D	Visited Node	8	B
E	Visited Node	10	D
Z	Current Node	14	D

We found the shortest path from A to Z.
Read the path from Z to A using the previous node column:
Z > D > B > C > A
So the Shortest Path is:
A – C – B – D – Z with a length of 14

Distance Vector Routing Algorithm

“Distant vector routing algorithm also called as Bellman-Ford algorithm or Ford Fulkerson algorithm used to calculate the shortest path in the network.”

Distance Vector Routing Algorithm

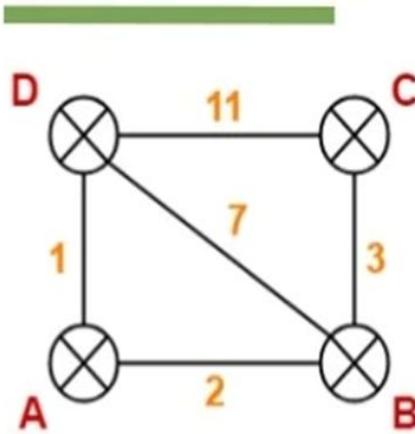
- The Distance vector algorithm is a dynamic algorithm.
- The term distance vector refers to the fact that the protocol manipulates vectors of distances to other nodes in the network.
- It is mainly used in Advanced Research Projects Agency Network (ARPANET), and Routing Information Protocol (RIP).
- Each router maintains a distance table known as **Vector**.

Distance Vector Routing Algorithm

- The Distance vector algorithm is iterative, asynchronous and distributed.
- **Distributed:** It is distributed in that each node receives information from one or more of its directly attached neighbors, performs calculation and then distributes the result back to its neighbors.
- **Iterative:** It is iterative in that its process continues until no more information is available to be exchanged between neighbors.
- **Asynchronous :**An asynchronous algorithm is one in which nodes operate independently and don't necessarily follow a synchronized clock or timing

Each router prepares its routing table. By their local knowledge, each router knows about-

- All the routers present in the network
- Distance to its neighboring routers



A Routing Table

Net Id	Cost	Next Hop
A	0	A
B	2	B
C	∞	-
D	1	D

B Routing Table

Net Id	Cost	Next Hop
A	2	A
B	0	B
C	3	C
D	7	D

C Routing Table

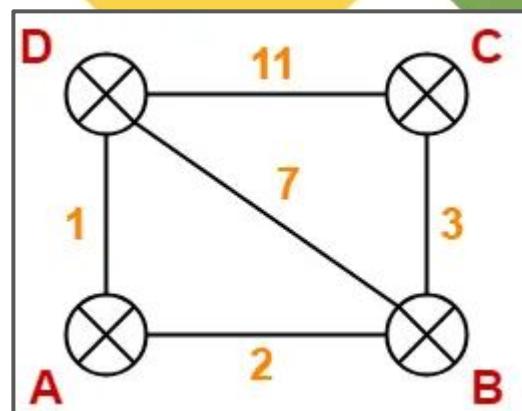
Net Id	Cost	Next Hop
A	∞	-
B	3	B
C	0	C
D	11	D

D Routing Table

Net Id	Cost	Next Hop
A	1	A
B	7	B
C	11	C
D	0	D

Updating Routing Table

1. Each router exchanges its distance vector with its neighbors.
2. After exchanging the distance vectors, each router prepares a new routing table.
3. Router A receives distance vectors from its neighbors B and D.
4. Router A prepares a new routing table.



Bellman-Ford algorithm:

The Distance Vector calculation is based on minimizing the cost to each destination

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

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

$$Dx(y) = \min \{ C(x,v) + Dv(y) \}$$

Source

Destination

Intermediate Node

At Router A:

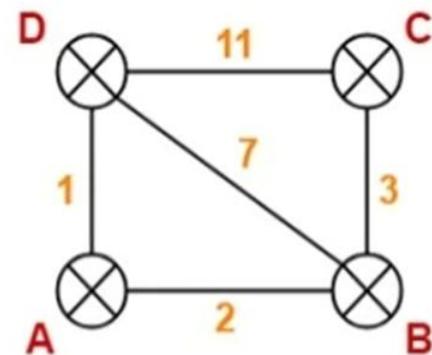
	From B	From D
A		A
B	2	B
C	0	C
D	3	7
	7	D

$$\text{Cost}(A \rightarrow B) = 2$$

$$\text{Cost}(A \rightarrow D) = 1$$

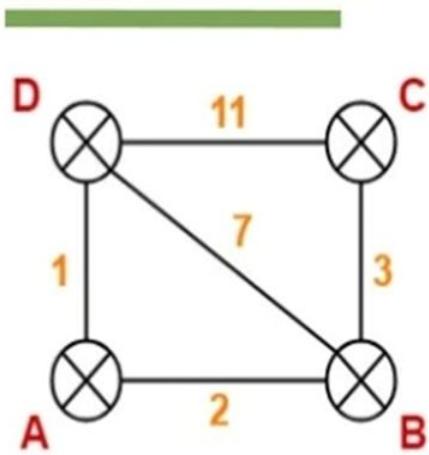
Destination	Distance	Next hop
A	0	A
B		
C		
D		

New Routing Table at Router A



- Cost of reaching destination B from router A = $\min \{ 2+0 , 1+7 \} = 2$ via B.
- Cost of reaching destination C from router A = $\min \{ 2+3 , 1+11 \} = 5$ via B.
- Cost of reaching destination D from router A = $\min \{ 2+7 , 1+0 \} = 1$ via D.

Updating “A Routing Table”



A Routing Table

Net Id	Cost	Next Hop
A	0	A
B	2	B
C	∞	-
D	1	D

New A Routing Table

Net Id	Cost	Next Hop
A	0	A
B	2	B
C	5	B
D	1	D

At Router B:

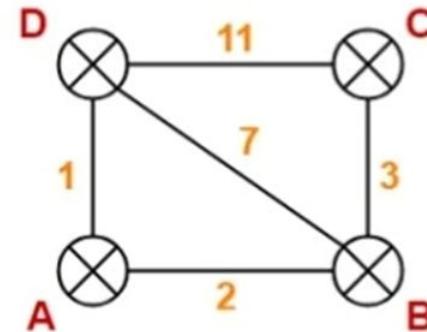
- Router B receives distance vectors from its neighbors A, C and D.
- Router B prepares a new routing table

	From A	From C	From D
A	0	A	A
B	2	B	B
C	∞	C	0
D	1	D	11

$$\text{Cost}(B \rightarrow A) = 2 \quad \text{Cost}(B \rightarrow C) = 3 \quad \text{Cost}(B \rightarrow D) = 7$$

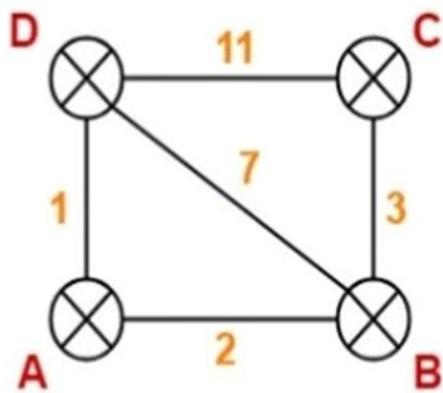
Destination	Distance	Next hop
A		
B	0	B
C		
D		

New Routing Table at Router B



- Cost of reaching destination A from router B = $\min \{ 2+0, 3+\infty, 7+1 \} = 2$ via A.
- Cost of reaching destination C from router B = $\min \{ 2+\infty, 3+0, 7+11 \} = 3$ via C.
- Cost of reaching destination D from router B = $\min \{ 2+1, 3+11, 7+0 \} = 3$ via A.

Updating “B Routing Table”



B Routing Table

Net Id	Cost	Next Hop
A	2	A
B	0	B
C	3	C
D	7	D

New B Routing Table

Net Id	Cost	Next Hop
A	2	A
B	0	B
C	3	C
D	3	A

At Router C:

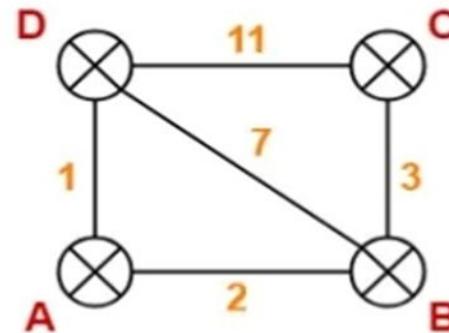
- Router C receives distance vectors from its neighbors B and D.
- Router C prepares a new routing table

	From B	From D
A	2	A
B	0	B
C	3	C
D	7	D

Cost (C→B) = 3 Cost (C→D) = 11

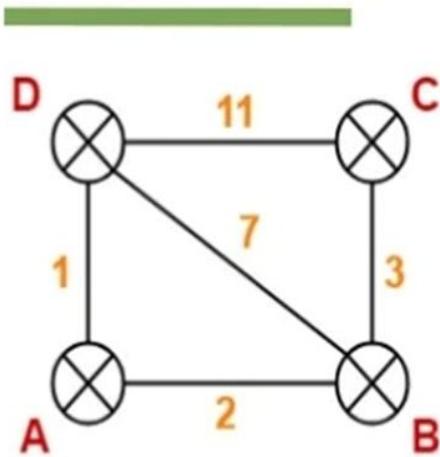
Destination	Distance	Next hop
A		
B		
C	0	C
D		

New Routing Table at Router C



- Cost of reaching destination A from router C = $\min \{ 3+2, 11+1 \} = 5$ via B.
- Cost of reaching destination B from router C = $\min \{ 3+0, 11+7 \} = 3$ via B.
- Cost of reaching destination D from router C = $\min \{ 3+7, 11+0 \} = 10$ via B.

Updating “C Routing Table”



C Routing Table

Net Id	Cost	Next Hop
A	∞	-
B	3	B
C	0	C
D	11	D

**New C
Routing Table**

Net Id	Cost	Next Hop
A	5	B
B	3	B
C	0	C
D	10	B

At Router D:

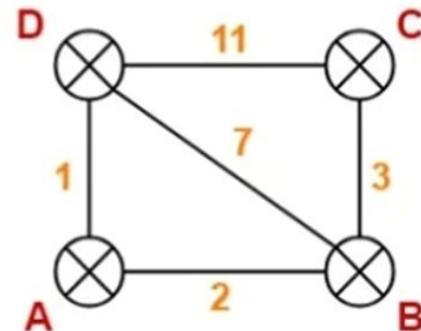
- Router D receives distance vectors from its neighbors A, B and C.
- Router D prepares a new routing table

	From A	From B	From C
A	0	A	A
B	2	B	B
C	∞	C	C
D	1	D	D

Cost (D→A) = 1 Cost (D→B) = 7 Cost (D→C) = 11

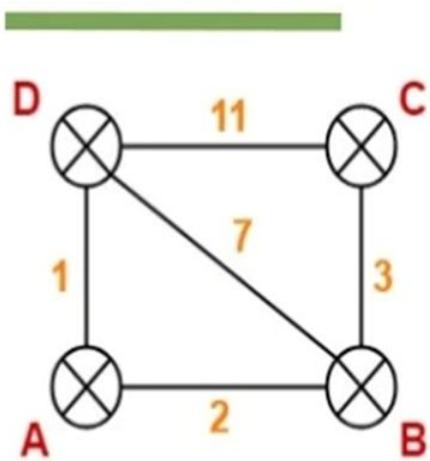
Destination	Distance	Next hop
A		
B		
C		
D	0	D

New Routing Table at Router D



- Cost of reaching destination A from router D = $\min \{ 1+0 , 7+2 , 11+\infty \} = 1$ via A.
- Cost of reaching destination B from router D = $\min \{ 1+2 , 7+0 , 11+3 \} = 3$ via A.
- Cost of reaching destination C from router D = $\min \{ 1+\infty , 7+3 , 11+0 \} = 10$ via B.

Updating “D Routing Table”



D Routing Table

Net Id	Cost	Next Hop
A	1	A
B	7	B
C	11	C
D	0	D

**New D
Routing Table**

Net Id	Cost	Next Hop
A	1	A
B	3	A
C	10	B
D	0	D

New Routing Tables:

Router A

Destination	Distance	Next Hop
A	0	A
B	2	B
C	5	B
D	1	D

Router B

Destination	Distance	Next Hop
A	2	A
B	0	B
C	3	C
D	3	A

Router C

Destination	Distance	Next Hop
A	5	B
B	3	B
C	0	C
D	10	B

Router D

Destination	Distance	Next Hop
A	1	A
B	3	A
C	10	B
D	0	D

"Still, the routing tables don't produce an optimal path. So, repeat the process, i.e., each router exchanges its tables until the table gives an optimal path."

At Router A-

- Router A receives distance vectors from its neighbors B and D.
- Router A prepares a new routing table as-

From B

A	2
B	0
C	3
D	3

Cost(A→B) = 2

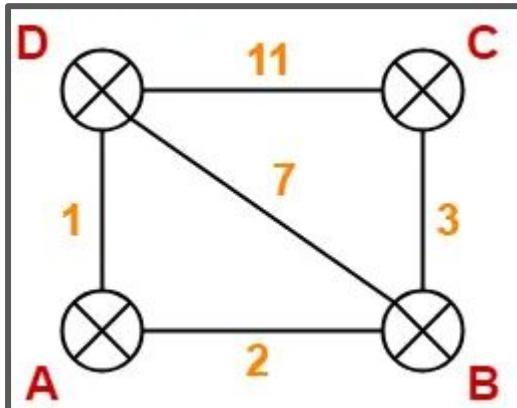
From D

A	1
B	3
C	10
D	0

Cost(A→D) = 1

Destination	Distance	Next hop
A	0	A
B		
C		
D		

New Routing Table at Router A



- Cost of reaching destination B from router A = $\min \{ 2+0 , 1+3 \} = 2$ via B.
- Cost of reaching destination C from router A = $\min \{ 2+3 , 1+10 \} = 5$ via B.
- Cost of reaching destination D from router A = $\min \{ 2+3 , 1+0 \} = 1$ via D.

The new routing table **at Router A** is-

Destination	Distance	Next Hop
A	0	A
B	2	B
C	5	B
D	1	D

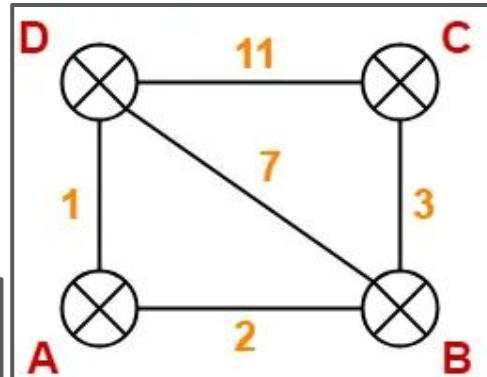
At Router B-

- Router B receives distance vectors from its neighbors A, C and D.
- Router B prepares a new routing table as-

From A		From C		From D		New Routing Table at Router B		
Destination	Distance	Destination	Distance	Destination	Distance	Destination	Distance	Next hop
A	0	A	5	A	1	A		
B	2	B	3	B	3	B	0	B
C	5	C	0	C	10	C		
D	1	D	10	D	0	D		

Cost (B→A) = 2 Cost (B→C) = 3 Cost (B→D) = 3

New Routing Table at Router B



- Cost of reaching destination A from router B = $\min \{ 2+0 , 3+5 , 3+1 \} = 2$ via A.
- Cost of reaching destination C from router B = $\min \{ 2+5 , 3+0 , 3+10 \} = 3$ via C.
- Cost of reaching destination D from router B = $\min \{ 2+1 , 3+10 , 3+0 \} = 3$ via A.

The new routing table **at Router B** is-

Destination	Distance	Next Hop
A	2	A
B	0	B
C	3	C
D	3	A

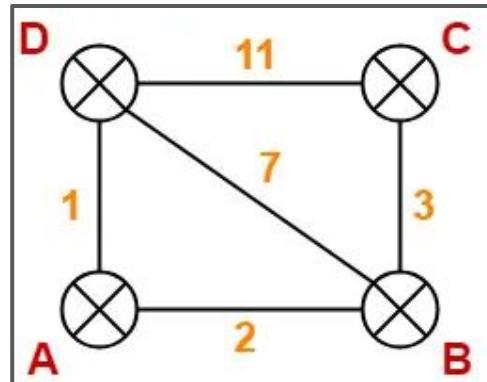
At Router C-

- Router C receives distance vectors from its neighbors B and D.
- Router C prepares a new routing table as-

From B		From D		Destination	Distance	Next hop
A	2	A	1			
B	0	B	3			
C	3	C	10			
D	3	D	0			

Cost (C→B) = 3 Cost (C→D) = 10

New Routing Table at Router C



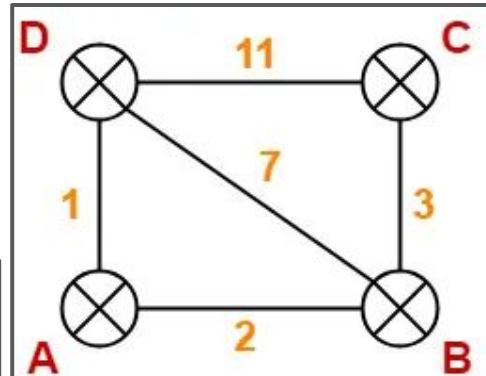
- Cost of reaching destination A from router C = $\min \{ 3+2, 10+1 \} = 5$ via B.
- Cost of reaching destination B from router C = $\min \{ 3+0, 10+3 \} = 3$ via B.
- Cost of reaching destination D from router C = $\min \{ 3+3, 10+0 \} = 6$ via B.

The new routing table **at Router C** is-

Destination	Distance	Next Hop
A	5	B
B	3	B
C	0	C
D	6	B

At Router D-

- Router D receives distance vectors from its neighbors A, B and C.
 - Router D prepares a new routing table as-



From A	From B	From C
A B C D 0 2 5 1	A B C D 2 0 3 3	A B C D 5 3 0 10

Destination	Distance	Next hop
A		
B		
C		
D	0	D

Cost (D→A) = 1 Cost (D→B) = 3 Cost (D→C) = 10

New Routing Table at Router D

- Cost of reaching destination A from router D = $\min \{ 1+0, 3+2, 10+5 \} = 1$ via A.
 - Cost of reaching destination B from router D = $\min \{ 1+2, 3+0, 10+3 \} = 3$ via A.
 - Cost of reaching destination C from router D = $\min \{ 1+5, 3+3, 10+0 \} = 6$ via A.

The new routing table **at Router D** is-

Destination	Distance	Next Hop
A	1	A
B	3	A
C	6	A
D	0	D

These will be the final routing tables at each router.

Router A

Destination	Distance	Next Hop
A	0	A
B	2	B
C	5	B
D	1	D

Router B

Destination	Distance	Next Hop
A	2	A
B	0	B
C	3	C
D	3	A

Router C

Destination	Distance	Next Hop
A	5	B
B	3	B
C	0	C
D	6	B

Router D

Destination	Distance	Next Hop
A	1	A
B	3	A
C	6	A
D	0	D

Non- Adaptive Routing Algorithm

Flooding

- In case of flooding, every incoming packet is sent to all the outgoing links except the one from it has been reached.
- The disadvantage of flooding is that node may contain several copies of a particular packet.
- To overcome this problem, sequence numbers, spanning tree & hop count are used.

Random Walks

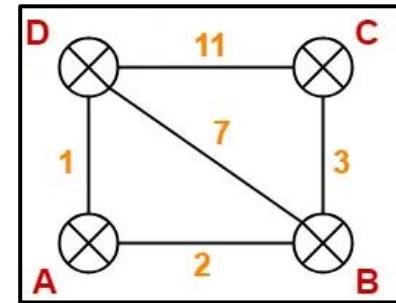
- In this type of algorithm, data packets are transmitted through the node by node or host by host randomly to one of its neighbors.
- This method is extremely strong which is frequently executed by transmitting data packets over the network link which is queued least.

Flooding

“Flooding is a simple broadcasting technique where a network node sends a message to all of its neighbors *without considering the topology or routing paths.*”

Flooding:

- Flooding requires **no information** like **topology, load condition, cost of different paths.**
- Every incoming packet to a node is sent out on every outgoing line except the one it arrived on.
- For example in the above figure
 - Incoming packet to [A] is sent out to [B], [D]
 - From [B] is sent to [C],[D] and from [D] it is sent to [A], [C].



Limitations:

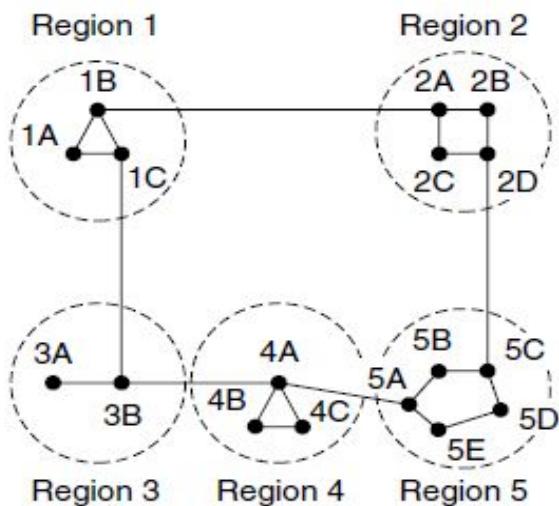
- Flooding generates **vast number of duplicate packets**.
- **Network Congestion:** Flooding causes a lot of unnecessary traffic since messages are sent to all nodes in the network, reduces overall network performance.

Advantages:

- **Reliability:** Flooding ensures that a message reaches all nodes in the network, making it highly reliable.
- **Simple Implementation:** Flooding is straightforward to implement and **does not require complex routing tables or algorithms.**
- **Fault Tolerance:** Because flooding sends messages to all nodes, it is inherently fault-tolerant. **Even if some nodes or links in the network fail, the message can still reach other nodes.**
- **Limited Network Size:** Flooding can be efficient in **very small networks** where the number of nodes is manageable.

Hierarchical Routing Algorithm

- **As networks grow in size, the router routing tables grow proportionally.**
- Not only is router memory consumed by ever-increasing tables, but more CPU time is needed to scan them and more bandwidth is needed to send status reports about them.
- At a certain point, the network may grow to the point where it is no longer feasible for every router to have an entry for every other router, so the routing will have to be done hierarchically.
- **When hierarchical routing is used**, the routers are divided into what we will call **regions**.
- Each router knows all the details about how to route packets to destinations within its own region but knows nothing about the internal structure of other regions.
- For huge networks, a two-level hierarchy may be insufficient; it may be necessary to group the **regions into clusters, the clusters into zones, the zones into groups**, and so on.



(a)

*Autonomous System (AS) ?
* Gateway Router ?

Full table for 1A

Dest.	Line	Hops
1A	-	-
1B	1B	1
1C	1C	1
2A	1B	2
2B	1B	3
2C	1B	3
2D	1B	4
3A	1C	3
3B	1C	2
4A	1C	3
4B	1C	4
4C	1C	4
5A	1C	4
5B	1C	5
5C	1B	5
5D	1C	6
5E	1C	5

(b)

Hierarchical table for 1A

Dest.	Line	Hops
1A	-	-
1B	1B	1
1C	1C	1
2	1B	2
3	1C	2
4	1C	3
5	1C	4

(c)

Figure 5-14. Hierarchical routing.

- Figure 5-14 gives a quantitative example of routing in **a two-level hierarchy** with **five regions**.
- The full routing table for router **1A** has **17 entries**, as shown in Fig. 5-14(b).
- When routing is done hierarchically, as in Fig. 5-14(c), there are entries for all the local routers, as before, but all other regions are condensed into a single router, so all traffic for region 2 goes via the 1B-2A line, but the rest of the remote traffic goes via the 1C-3B line.
- **Hierarchical routing has reduced the table from 17 to 7 entries.**
- As the ratio of the number of regions to the number of routers per region grows, the savings in table space increase.

- An **Autonomous System (AS)** is a collection of routers whose prefixes and routing policies are **under common administrative control**. This could be a network service provider, a large company, a university.
- All the routers which present in the **Autonomous System** will follow the same routing algorithm.
- Autonomous routing protocols can be categorized into **two main types** based on their **scope and purpose** within an Autonomous System (AS) or between ASes.
 - **Intra-AS Routing Protocol (Interior Routing Protocols)**
 - **Inter-AS Routing Protocol (Exterior Routing Protocols (EGPs))**

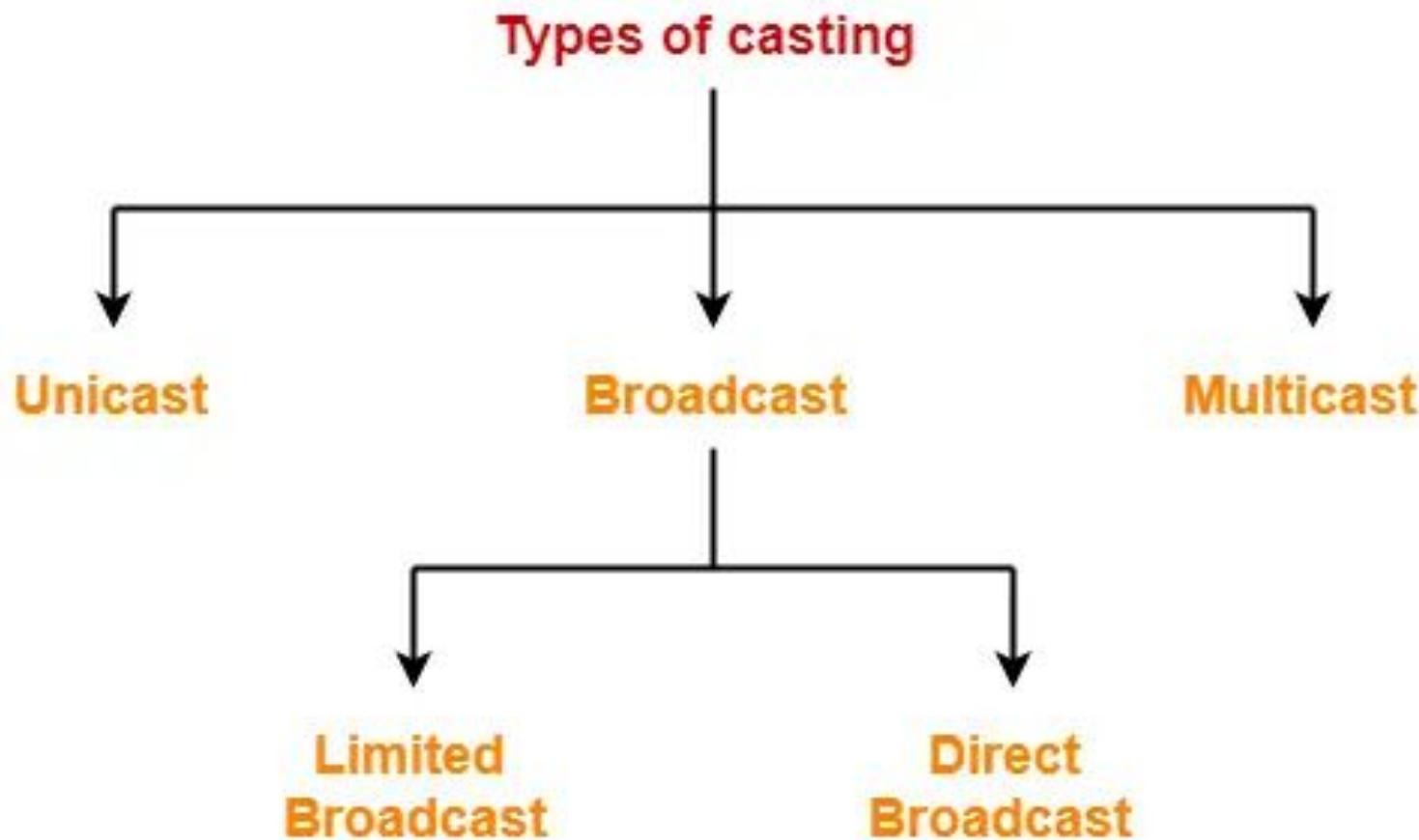
Intra-AS Routing Protocol (Interior Routing Protocols):

- Interior Routing Protocols are used for **routing within a single Autonomous System (AS)**.
- They determine how traffic is routed within the boundaries of the AS.
- Eg: **Routing Information Protocol (RIP), Open Shortest Path First (OSPF)**

Inter-AS Routing Protocol (Exterior Routing Protocols (EGPs) :

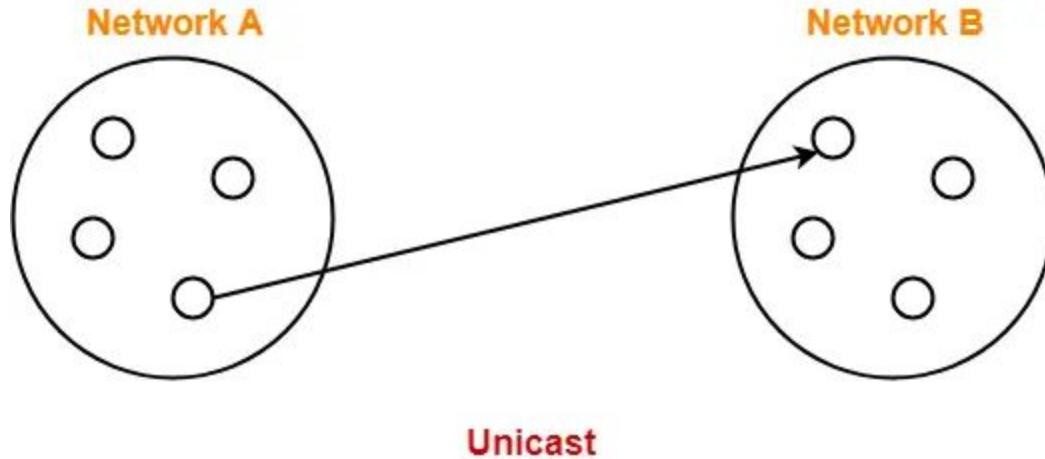
- Exterior Routing Protocols are used for **routing between different Autonomous Systems (ASes)**.
- They are responsible for inter-domain routing, facilitating the exchange of routing information .
- Eg: **Border Gateway Protocol (BGP) , Exterior Gateway Protocol (EGP)**

Broadcast and Multicast Routing



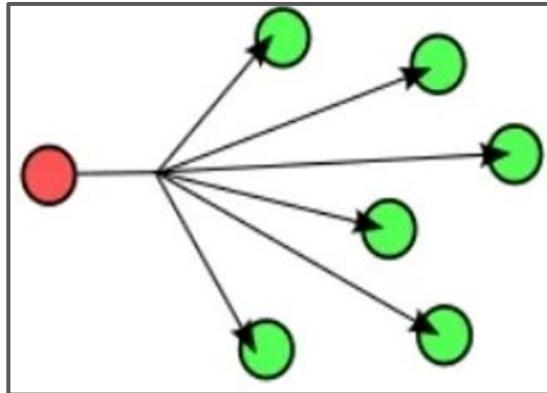
Unicast:

- Transmitting data from **one source host to one destination host** is called as unicast.
- It is a **one to one transmission**

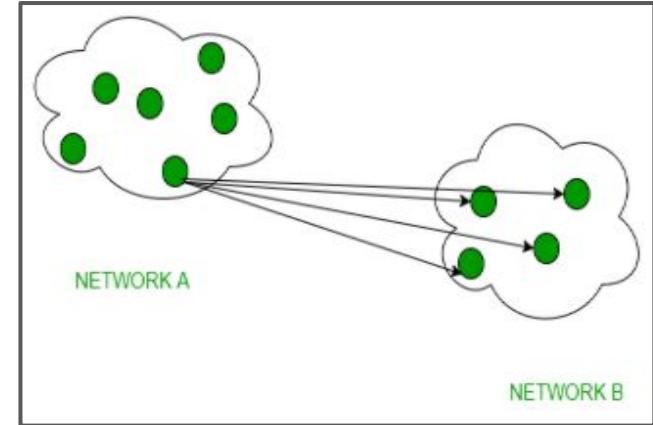


Broadcast Routing:

[1]



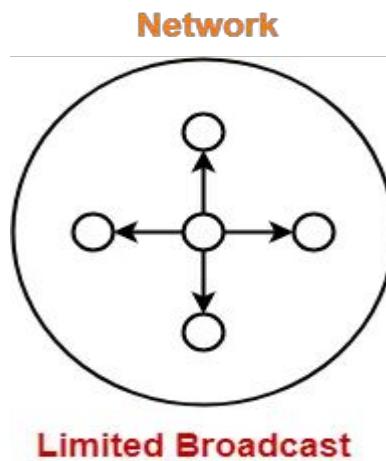
[2]



- In broadcast routing, **packets are sent to all nodes even if they do not want it.**
- Broadcast transfer uses **one-to-all** technique and can be classified into two types :
 - **Limited Broadcasting**
 - **Direct Broadcasting**
- In broadcasting mode, transmission happens from one host to all the other hosts connected on the LAN.

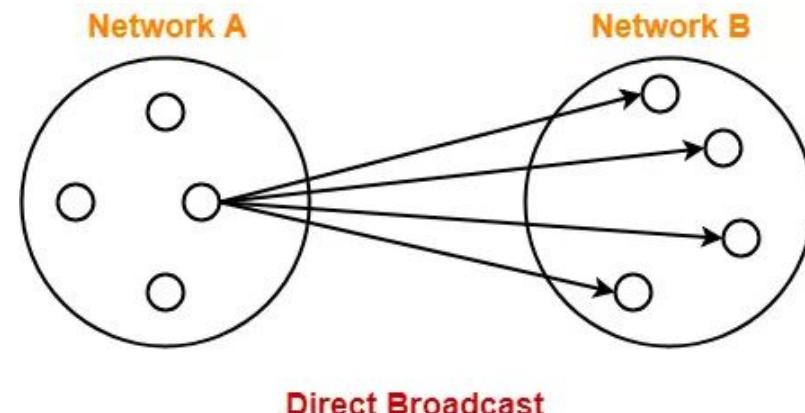
Limited Broadcasting:

- Transmitting data from one source host to all other hosts **residing in the same network** is called as limited broadcast.



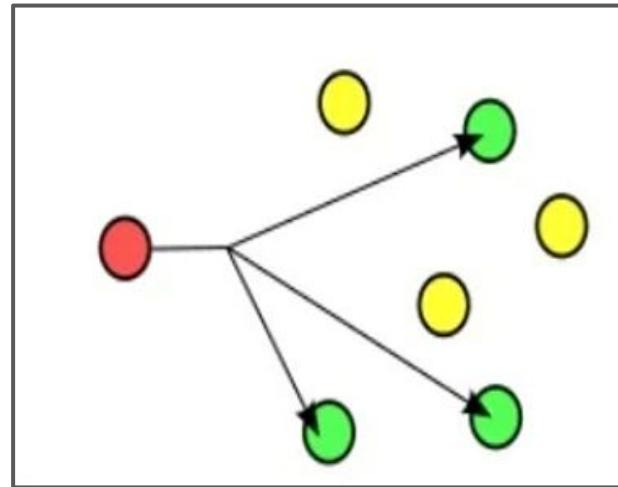
Directed Broadcasting:

- Transmitting data from one source host to all other hosts **residing in some other network** is called as direct broadcast.



Multicast Routing

- In Multicast routing, **the data is sent to only nodes which wants to receive the packets.**
- Transmitting data from one source host to a particular group of hosts having interest in receiving the data is called as multicast.
- Multicast transfer uses **one-to-many** technique.



Broadcast Routing vs Multicast Routing

Broadcast	Multicast
It scales well across large networks.	It does not scale well across large networks.
Its bandwidth is wasted.	It utilizes bandwidth efficiently.
It has one-to-all mapping.	It has one-to-many mapping.
Hub is an example of a broadcast device.	Switch is an example of a multicast device.
It works on star and bus topology	It works on star, mesh, tree and hybrid topology.

Congestion Control Algorithms

- Congestion in a computer network refers to **a situation in which there is an excessive amount of data traffic on a network than it can effectively handle**, which leads to
 - Performance degradation
 - Packet loss
 - Reduced Quality of Service
 - Network instability
- It's a common problem in networks, particularly in situations where **the demand for network resources exceeds their availability or capacity**.
- Congestion can occur in both **wired and wireless networks** and can have various causes and consequences.

Congestion-The situation in which too many packets are present in the subnet.

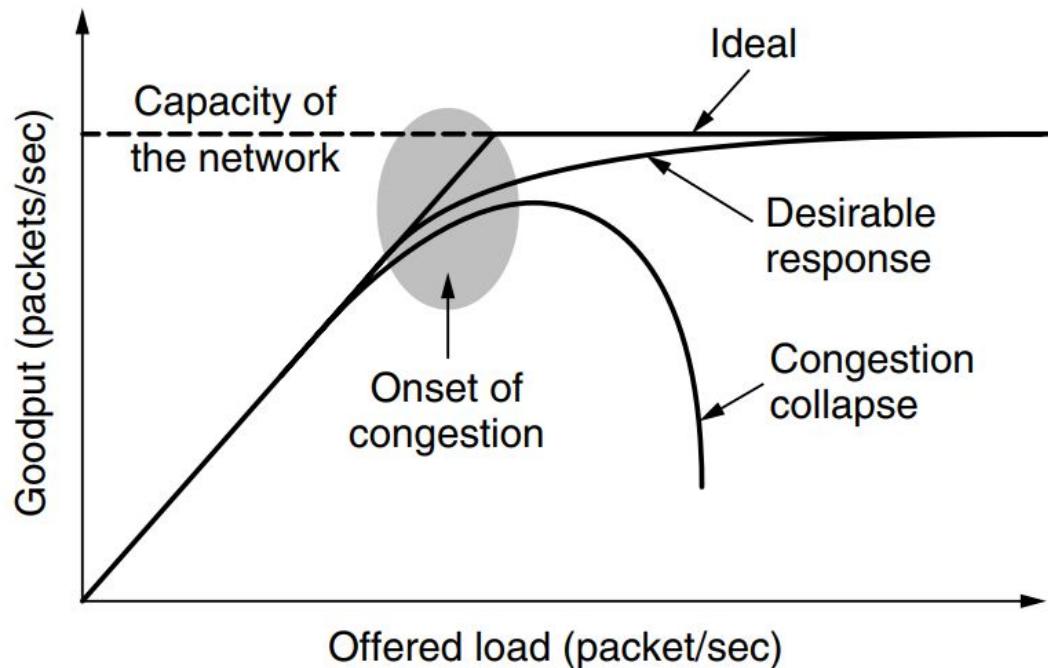


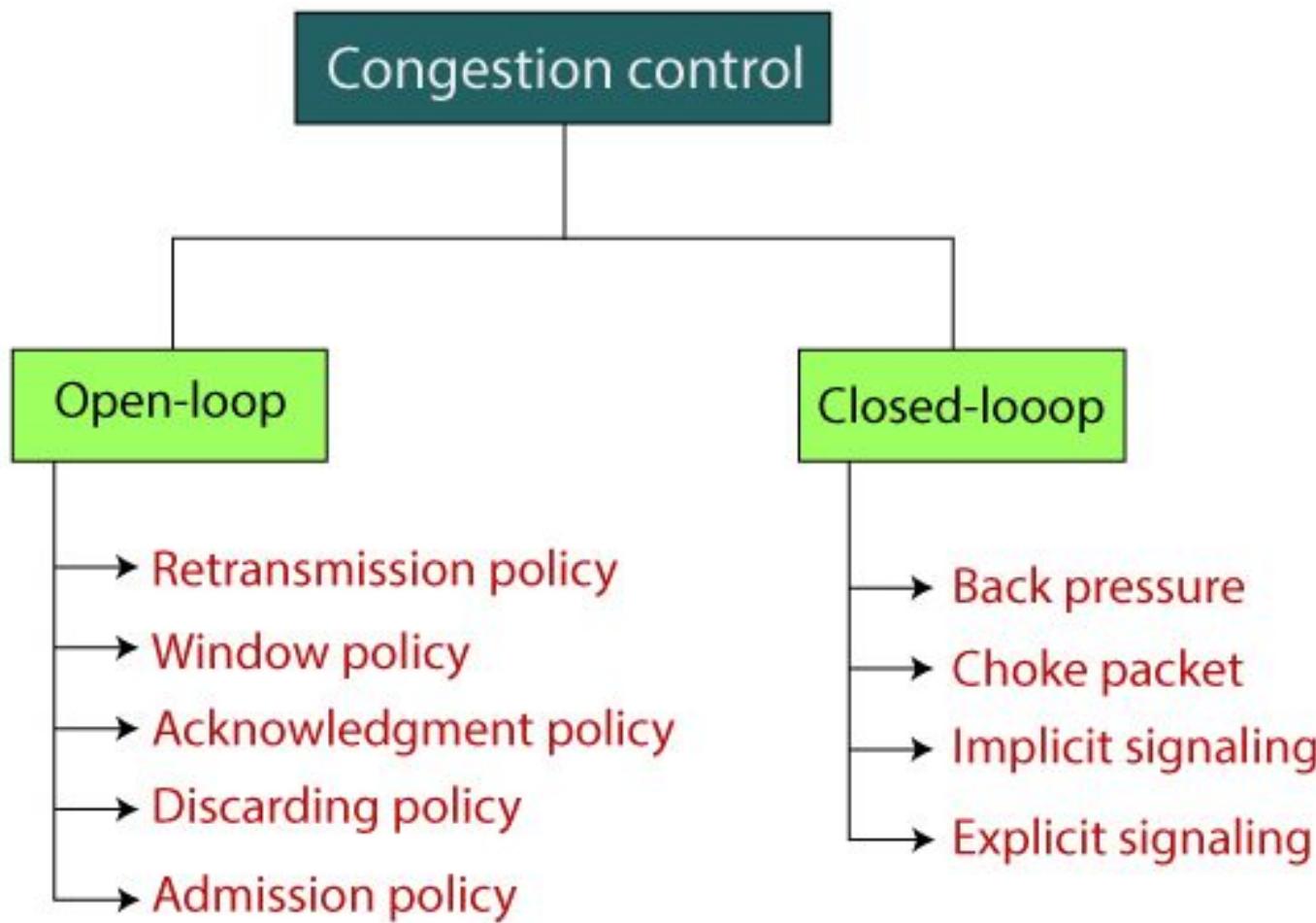
Figure 5-21. With too much traffic, performance drops sharply.

- Figure 5-21 depicts the onset of congestion.
- When the number of packets hosts send into the network is well **within its carrying capacity**, **the number delivered is proportional to the number sent**.
- If twice as many are sent, twice as many are delivered.
- However, as **the offered load approaches the carrying capacity**, bursts of traffic occasionally fill up the buffers inside routers and **some packets are lost**.
- These lost packets consume some of the capacity, so the number of delivered packets falls below the ideal curve. The network is now **congested**.

Congestion Control Techniques

- Congestion control refers to techniques and mechanisms that can **either prevent congestion, before it happens, or remove congestion, after it has happened.**
- In general, we can divide congestion control mechanisms into two broad categories:
 - **Open-loop congestion control (Prevention)**
 - **Closed-loop congestion control (Removal)**

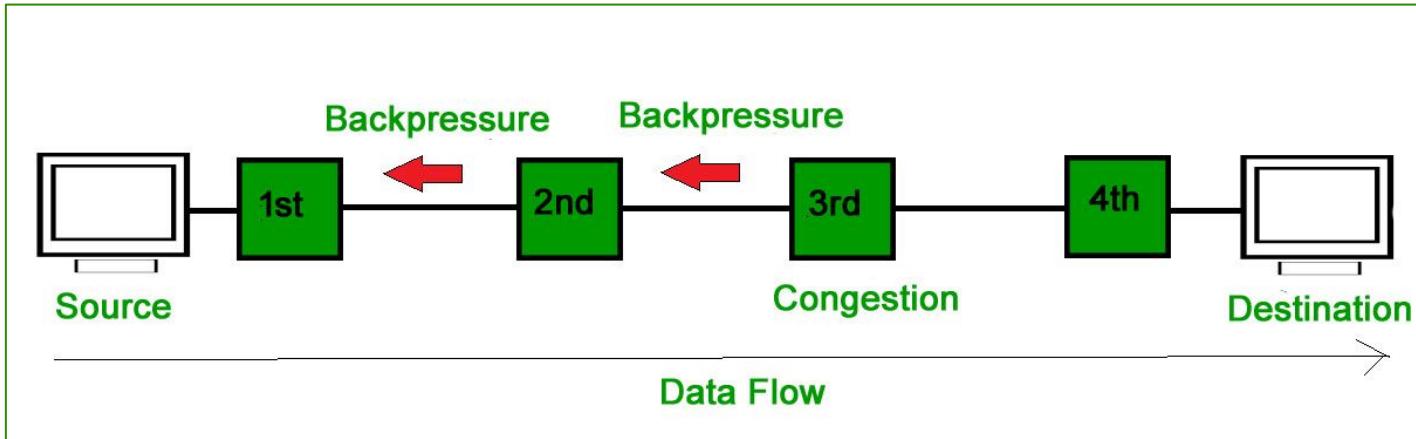
Open-Loop Congestion Control (Prevention)	Closed-Loop Congestion Control (Removal)
<ul style="list-style-type: none">• Open-loop congestion control focuses on preventing congestion from occurring in the first place by adjusting the traffic before it enters the network.• This approach is proactive and relies on various techniques to manage the flow of traffic to avoid congestion.	<ul style="list-style-type: none">• Closed-loop congestion control, also known as feedback-based congestion control, focuses on detecting and reacting to congestion after it has occurred.• It dynamically adjusts the network traffic based on real-time feedback to alleviate congestion.



Warning Bit (or) Back Pressure

- ★ Backpressure is a technique in which a congested node stops receiving packets from upstream node.
- ★ This may cause the upstream node or nodes to become congested and reject receiving data from above nodes.
- ★ Backpressure is a node-to-node congestion control technique that propagate in the opposite direction of data flow
- ★ **A special bit in the packet header is set by the router to warn the source when congestion is detected.**
- ★ The bit is copied and piggy-backed on the ACK and sent to the sender.
- ★ The sender monitors the number of ACK packets it receives with the warning bit set and adjusts its transmission rate accordingly.

Warning Bit (or) Back Pressure

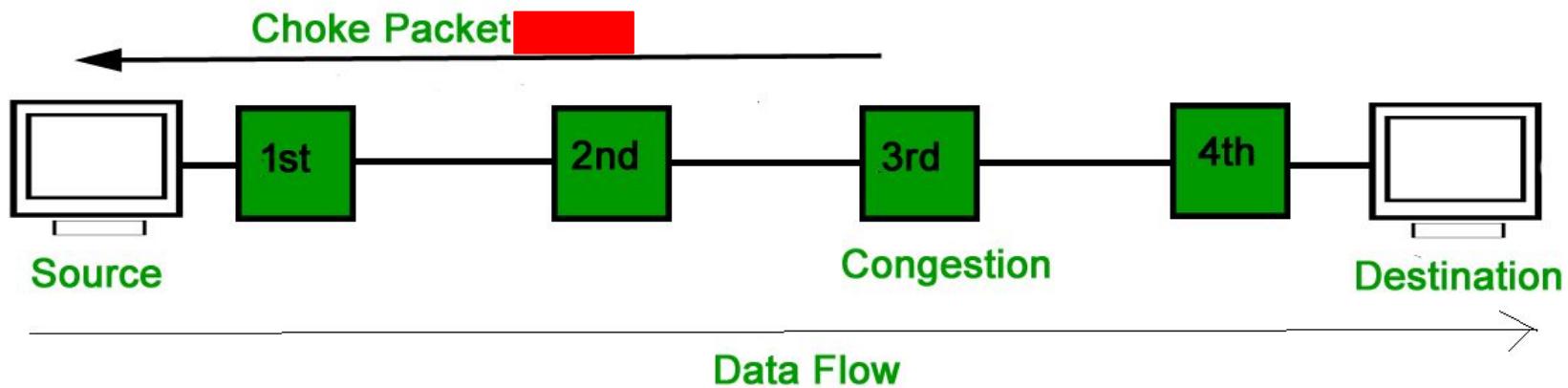


In above diagram the 3rd node is congested and stops receiving packets as a result 2nd node may be get congested due to slowing down of the output data flow. Similarly 1st node may get congested and inform the source to slow down.

Choke Packet Technique

- ★ Choke packet technique is applicable to both **virtual networks as well as datagram subnets**.
- ★ A choke packet is a packet sent by a node to the source to inform it of congestion.
- ★ Each router monitors its resources and the utilization at each of its output lines.
- ★ Whenever the resource utilization exceeds the threshold value which is set by the administrator, the router directly sends a choke packet to the source giving it a feedback to reduce the traffic.
- ★ The intermediate nodes through which the packets has traveled are not warned about congestion.

Choke Packet Technique



Implicit Signaling

- ★ In implicit signaling, **there is no communication between the congested node or nodes and the source.**
- ★ The source guesses that there is congestion somewhere in the network from other symptoms. **For example**, when a source sends several packets and there is **no acknowledgment for a while**, one assumption is that the network is congested.
- ★ The delay in receiving an acknowledgment is interpreted as congestion in the network; **the source should slow down.**

Explicit Signaling

- ★ In explicit signaling, if a node experiences congestion **it can explicitly sends a packet to the source or destination to inform about congestion.**
- ★ In the choke packet method, a separate packet is used for this purpose; in the explicit signaling method, the signal is included in the packets that carry data.

Explicit signaling can occur in either **forward or backward direction**.

- ★ **Forward Signaling** : In forward signaling, a signal is sent in the direction of the congestion. **The destination is warned about congestion.** The receiver in this case adopt policies to prevent further congestion.
- ★ **Backward Signaling** : In backward signaling, a signal is sent in the opposite direction of the congestion. **The source is warned about congestion** and it needs to slow down.

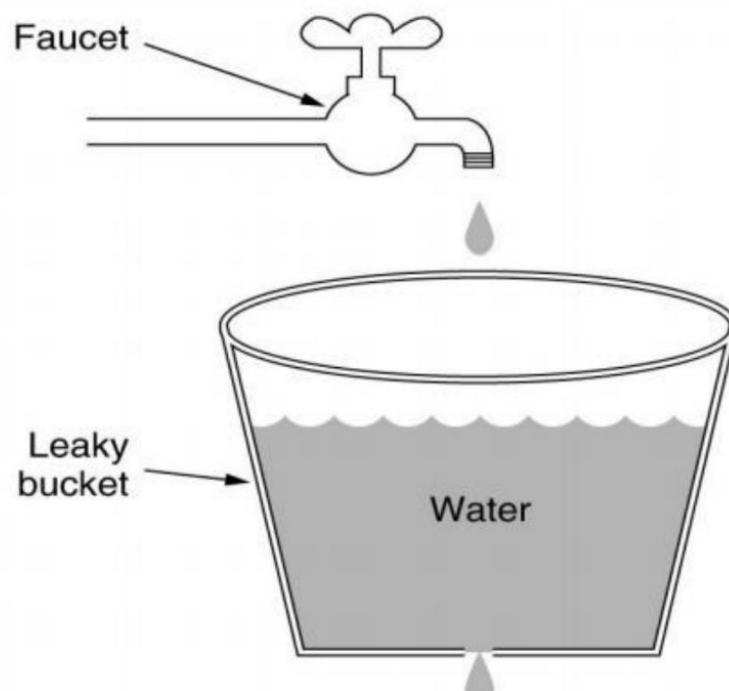
Traffic Shaping

- ★ Another method of congestion control is **to shape the traffic before it enters the network.**
- ★ In the network layer, before the network can make Quality of service guarantees, it must know what traffic is being guaranteed.
- ★ One of the main causes of congestion is that traffic is often bursty.
- ★ **Traffic Shaping** is a mechanism to *control the rate of traffic sent to the network.*
- ★ Traffic shaping **helps to regulate the rate of data transmission** and **reduces congestion.**

There are 2 types of traffic shaping algorithms:

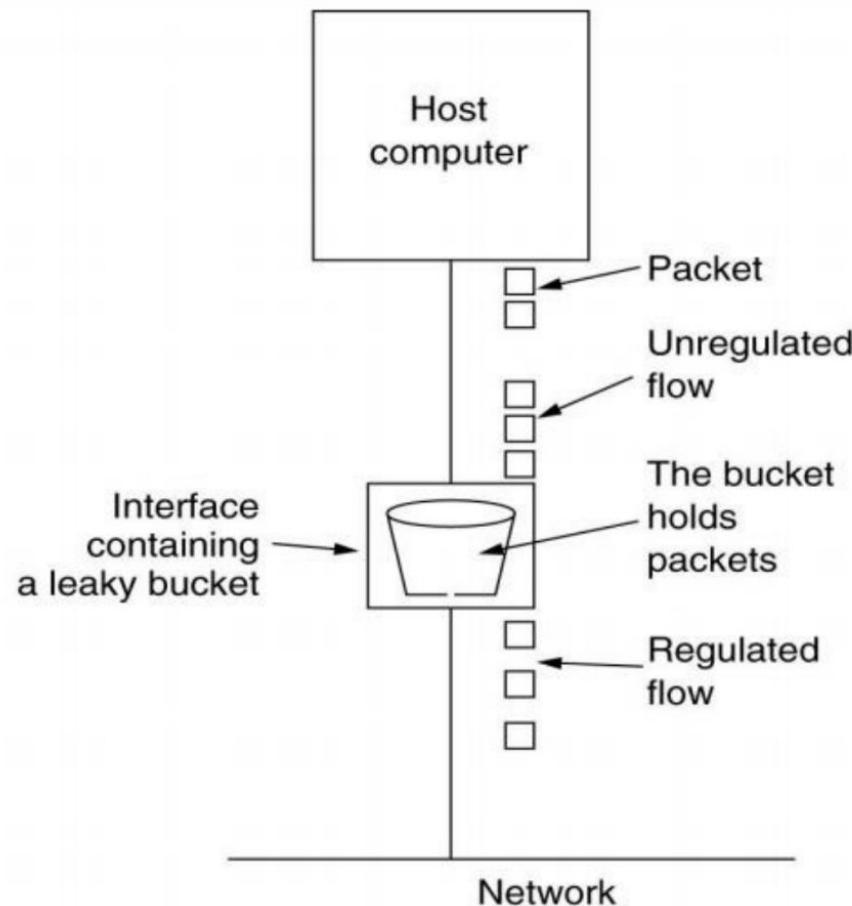
- **Leaky Bucket**
- **Token Bucket**





Water drips out of the
hole at a constant rate

(a)



(b)

- ★ The leaky bucket algorithm is ideal for **smoothing out bursty traffic**.
- ★ Just like a hole at the bottom of a water bucket leaks water out at a fixed rate, the leaky bucket algorithm does the same with network traffic.
- ★ Bursty chunks of traffic are stored in a "bucket" with a "hole" and **sent out at a controlled, average rate**.
- ★ The **hole represents the network's commitment to a particular bandwidth**.
- ★ The leaky bucket shapes the incoming traffic to ensure it conforms to the commitment.
- ★ Thus, regardless of how much data traffic enters the bucket, it always leaves at a constant output rate (the commitment).
- ★ This mechanism regulates the packet flow in the network and helps to prevent congestion that leads to performance deterioration and traffic delays.

- ★ Suppose data enters the network from various sources at different speeds.
- ★ Consider **one bursty source** that sends data at **20 Mbps for 2 seconds** for a total of **40 Mbps**.
- ★ Then it is silent, sending **no data for 5 seconds**. Then it again transmits data at a rate of **10 Mbps for 3 seconds**, thus sending a total of **30 Mbps**.
- ★ So, in a time span of **10 seconds** the source sends **70 Mb data**.
- ★ **However, the network has only committed a bandwidth of 5 Mbps for this source.**
- ★ Therefore, it uses **the leaky bucket algorithm** to output traffic at the rate of **5 Mbps** during the same time period of 10 seconds, which smooths out the network traffic.
- ★ Without the leaky bucket algorithm in place, the initial burst of 20 Mbps would have consumed a lot more bandwidth than the network had reserved (committed) for the source, which would have caused **congestion and a slowdown in the network**.

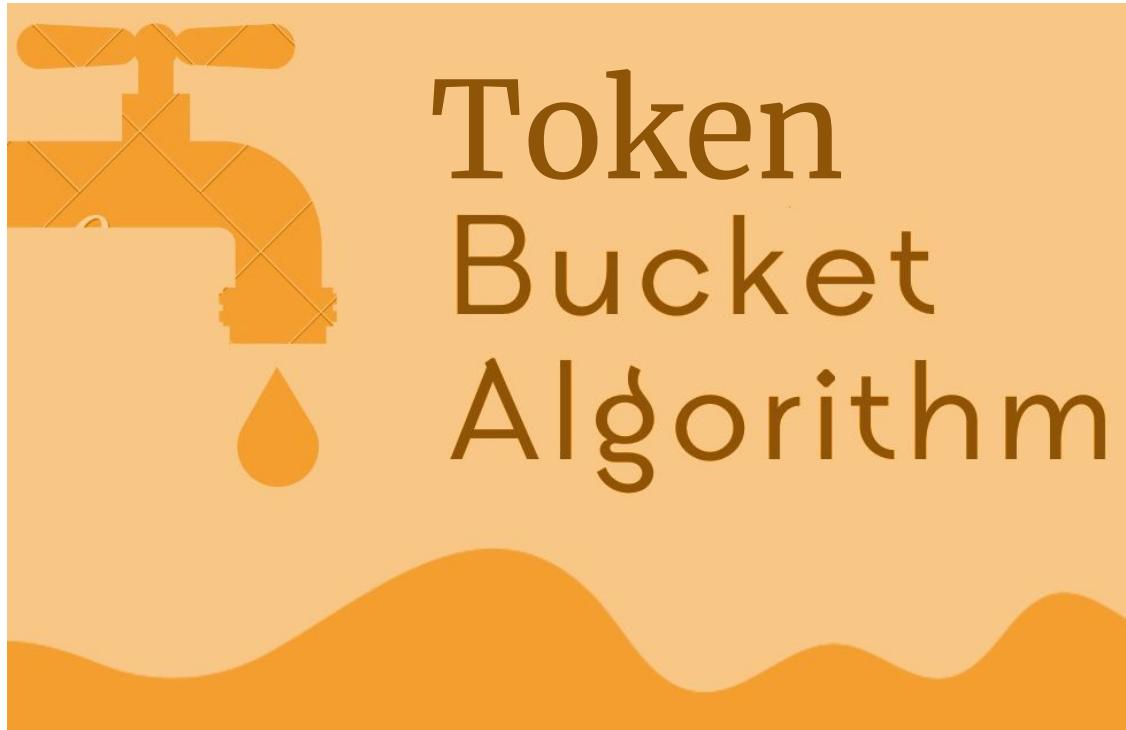
Problems in Leaky bucket Algorithm

[1] Fixed Bucket Size:

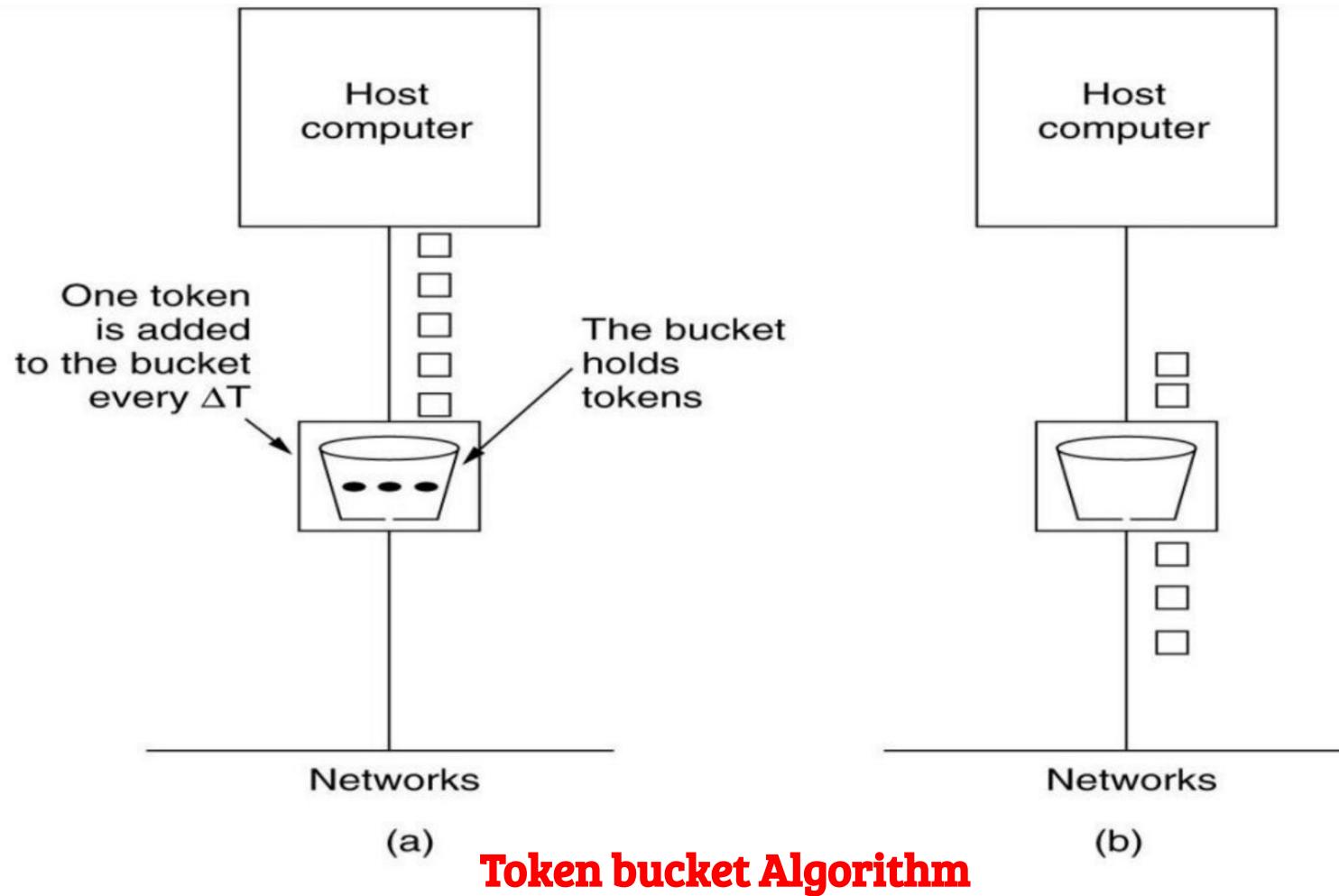
- ★ The leaky bucket algorithm is **primarily designed to regulate the average data rate over time.**
- ★ **It does not handle bursts of traffic well.**
- ★ If traffic arrives in bursts that **exceed the bucket's capacity**, it may result in **packet loss**.

[2] Not Suitable for Real-Time Traffic:

- ★ For real-time applications like **voice and video streaming**, the fixed-rate nature of the leaky bucket algorithm may not be ideal. These applications often require strict Quality of Service (QoS) guarantees, which the leaky bucket cannot provide on its own.



Token Bucket Algorithm



The token bucket algorithm works as follows:

- ★ **Token Generation:** A bucket is used to store tokens, which represent the permission to transmit a packet or request. Tokens are generated at a fixed rate and are added to the bucket at regular intervals. The rate at which tokens are added to the bucket is often referred to as the "token generation rate" or "token arrival rate."
- ★ **Token Consumption:** When a packet or request needs to be transmitted, it must first obtain a token from the bucket. If there are tokens available in the bucket, it is allowed to proceed. If there are no tokens available, the packet or request must wait until tokens become available.
- ★ **Bucket Capacity:** The bucket has a maximum capacity, and tokens cannot accumulate beyond this capacity. If the bucket is already full and new tokens arrive, they are simply discarded.

IP Address

IP address:

- ★ An IP address is **a unique address** that identifies a device on the internet or a local network.
- ★ An IP address, or Internet Protocol address, is **a numerical label** assigned to each device connected to a computer network that uses the Internet Protocol for communication.
- ★ **It serves two main purposes:**
 - *Identifying the host*
 - *Providing the location of the host in the network*
- ★ Computers that communicate over the internet or via local networks share information to a specific location using IP addresses.
- ★ The internet needs a way to differentiate between different computers, routers, and websites.

Here's a simplified explanation of how IP addresses work with a real-time example:

Device Connection:

- ★ When you connect your device (such as a computer, smartphone, or tablet) to the internet, it is assigned an IP address. This can happen through various means, such as a local router assigning a private IP address or your Internet Service Provider (ISP) assigning a public IP address.

Sending Data:

- ★ When you want to send or receive data over the internet, your device breaks down the data into packets. Each packet is a unit of data that includes the sender's IP address, the recipient's IP address, and the actual data being sent.

Here's a simplified explanation of how IP addresses work with a real-time example:

Routing:

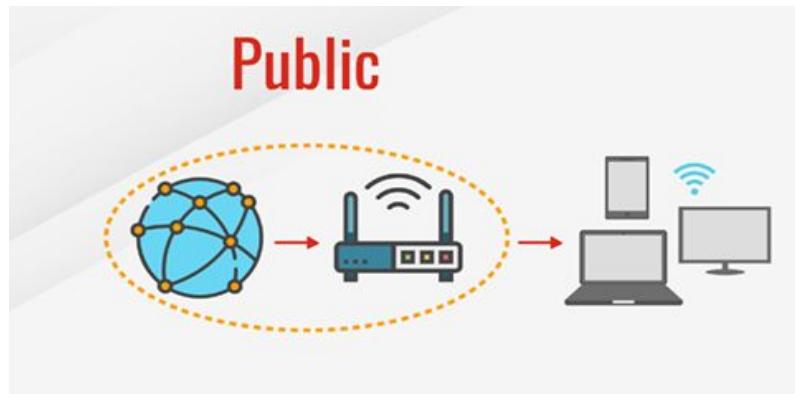
- ★ The packets travel through various routers and switches on the internet to reach their destination. Routers use the IP addresses to determine the best path for each packet to take to reach its destination.

Receiving Data:

- ★ When the packets arrive at their destination, the recipient's device uses its IP address to reassemble the packets into the original data.

Public IP Address:

- ★ A public IP address is an address assigned to your device by your **Internet Service Provider (ISP)** and is **visible** on the internet.
- ★ It uniquely identifies your device on **the global network**.
- ★ Public IP addresses are used for communication between devices on the internet.



Private IP Address:

- ★ Devices within your home network are assigned private IP addresses. These addresses are not directly accessible from the internet.
- ★ Example: Your computer might have a private IP address like 192.168.1.2, your smartphone might be assigned 192.168.1.3, and so on.
- ★ Devices communicate with each other using these private IP addresses within the local network, but when they access the internet, the router translates their requests into the public IP address.



Types of IP Address

IPv6

128-bit address

340 undecillion
possible addresses

Example:

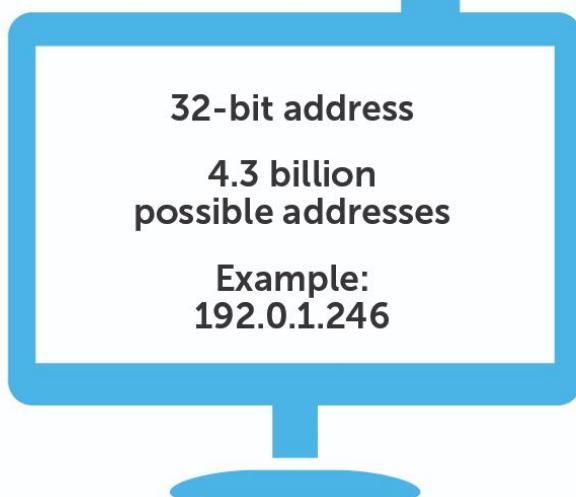
2002:db8::8a3f:362:7897

IPv4

32-bit address

4.3 billion
possible addresses

Example:
192.0.1.246

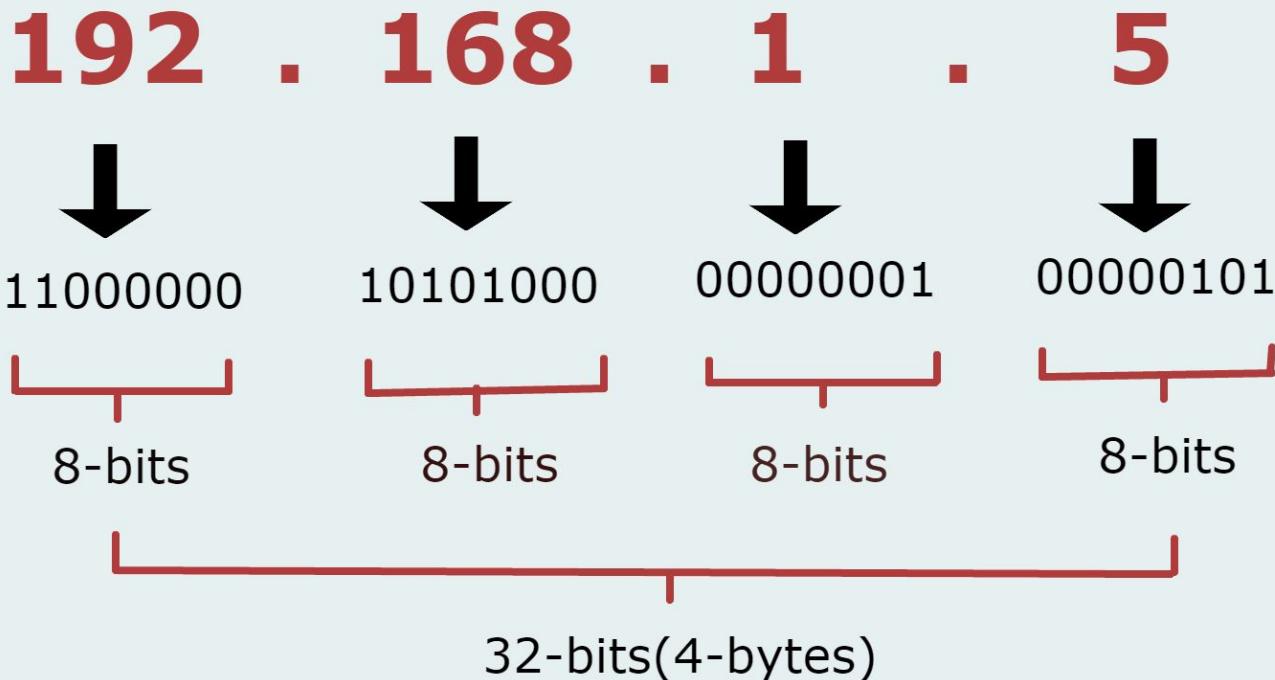


IPv4 Address

- ★ IPv4 stands for **Internet Protocol version four**, It is **a current version** and the most commonly used IP address.
- ★ It was introduced in 1981 by DARPA and was the first deployed version in 1982 for production on SATNET and on the ARPANET in January 1983.
- ★ IPv4 addresses are **32-bit integers** that have to be expressed in **Dotted Decimal Notation**.
- ★ It is represented by **4 numbers** separated by **dots** in the range of **0-255**, which have to be converted to 0 and 1, to be understood by Computers.
 - For Example, An IPv4 Address can be written as **189.123.123.90**.
- ★ **Each number in an octet** is in the range from **0-255**. This address can produce **4,294,967,296** possible unique addresses.i.e **4.29×10^9 address space**

IPV4 Address Format

IPV4 address represented in dotted-decimal notation



IPV6 Address

- ★ IPv6 is based on IPv6 and stands for **Internet Protocol version 6**.
- ★ It was first introduced in **December 1995** by **Internet Engineering Task Force**.
- ★ IP version 6 is the new version of Internet Protocol, which is way better than IP version 4 in terms of **complexity and efficiency**.
- ★ IPv6 is a **128-bit hexadecimal address**, which is written as a group of **8 hexadecimal numbers separated by colon (:)**.
- ★ This **hexadecimal address** contains both **numbers and alphabets**.
- ★ Due to the usage of both the numbers and alphabets, IPv6 is capable of producing over **340 undecillion (3.4×10^{38}) addresses**.
- ★ IPv6 provides a **large address space**, and it contains a **simple header** as compared to IPv4.

IPV6 Address Format

- IPv6 is a **128-bit hexadecimal address** made up of **8 sets of 16 bits each**, and these 8 sets are separated by a colon.
- In IPv6, each hexadecimal character represents 4 bits. So, we need to convert 4 bits to a hexadecimal number at a time.

2001:23ab:7612:0000:0000:**aaaa**:ac61:fde2

The diagram shows the 128-bit IPv6 address 2001:23ab:7612:0000:0000:**aaaa**:ac61:fde2. Below the address, there are eight pairs of brackets, each bracketing a 16-bit segment. The segments are: 2001, 23ab, 7612, 0000, 0000, **aaaa**, ac61, and fde2. The word "16 bit" is written under each of the first seven brackets. The eighth bracket is highlighted in yellow, and the word "16 bit" is written in green under it.

16 bit 16 bit 16 bit 16 bit 16 bit **16 bit** 16 bit 16 bit

Difference Between IPv4 and IPv6

IPV4	IPV6
IPv4 has a 32-bit address length	IPv6 has a 128-bit address length
It Supports Manual and DHCP address configuration	It supports Auto and renumbering address configuration
It can generate 4.29×10^9 address space	The address space of IPv6 is quite large it can produce 3.4×10^{38} address space
Address representation of IPv4 is in decimal	Address Representation of IPv6 is in hexadecimal
The Security feature is dependent on the application	IPSEC is an inbuilt security feature in the IPv6 protocol
In IPv4 Encryption and Authentication facility not provided	In IPv6 Encryption and Authentication are provided
IPv4's IP addresses are divided into five different classes. Class A , Class B, Class C, Class D , Class E.	IPv6 does not have any classes of the IP address

Classes of IP Address

IPv4's IP addresses are divided into five different classes. Class A , Class B, Class C, Class D , Class E.

- An IP (Internet Protocol) address is a numerical label assigned to the devices connected to a computer network that uses the IP for communication.
- IP address act as an identifier for a specific machine on a particular network.
- An IP address is a 32-bit unique address.
- The 32-bit IP address is divided into **five sub-classes**
 1. **Class A**
 2. **Class B**
 3. **Class C**
 4. **Class D**
 5. **Class E**

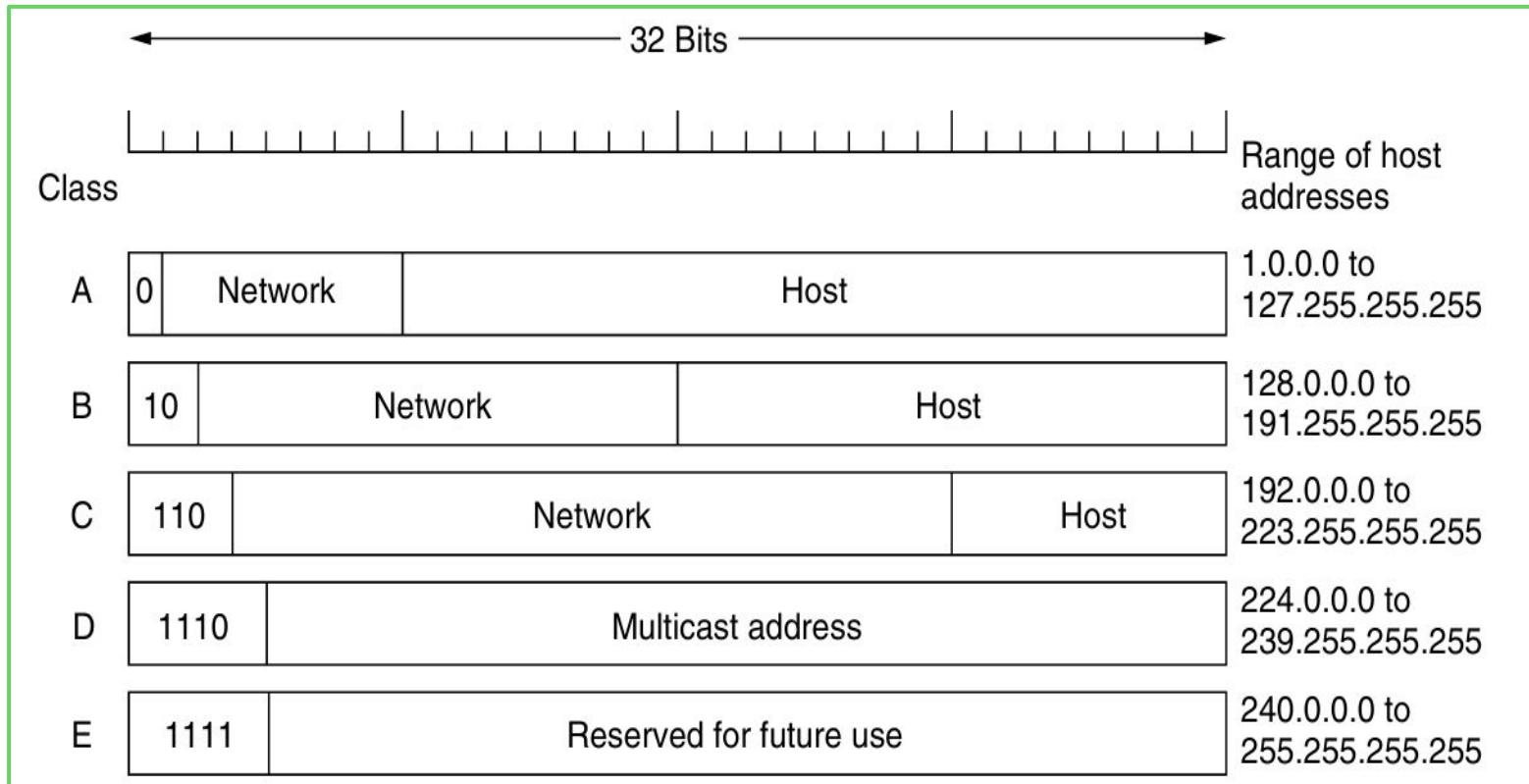
Parts of IP Address



IP Address is divided into two parts:

- **Prefix:** The prefix part of IP address *identifies the physical network to which the computer is attached*. Prefix is also known as **a network address**.
- **Suffix:** The suffix part *identifies the individual computer on the network*. The suffix is also called **the host address**

IP Address Formats



Classes A, B, C offers addresses for networks of three distinct network sizes. Class D is only used for multicast, and class E reserved exclusively for experimental purposes.

Class A Address

- IP addresses belonging to class A are assigned to the networks that contain a large number of hosts.
 - **The network ID is 8 bits long.**
 - **The host ID is 24 bits long.**
- In a Class A type of network,
 - The **higher-order bit** of the first octet in class A is always set to **0**.
 - The **remaining 7 bits** in the first octet are used **to determine network ID**.
 - The **remaining have 24 bits** used to determine the **host in the network**.
- An example of a Class A address is 102.168.212.226. Here, “102” helps you identify the network and 168.212.226 identify the host.

Class B Address

- IP address belonging to class B is assigned to networks that range from medium-sized to large-sized networks.
 - **The network ID is 16 bits long.**
 - **The host ID is 16 bits long.**
- In a Class B type of network,
 - The **higher-order bit** of the first octet in class B is always set to **10**.
 - The **remaining 14 bits** in the first octet are used **to determine network ID**.
 - The **remaining have 16 bits** used to determine the **host in the network**.
- An example of Class B IP address is 168.212.226.204, where "168 212" identifies the network and "226.204" helps you identify the Host.

Class C Address

- IP addresses belonging to class C are assigned to small-sized networks such as home or small business networks
 - **The network ID is 24 bits long.**
 - **The host ID is 8 bits long.**
- In a Class C type of network,
 - The **higher-order bit** of the first octet in class C is always set to **110**.
 - The **remaining 21 bits** in the first octet are used **to determine network ID**.
 - The **remaining have 8 bits** used to determine the **host in the network**.
- An example of Class C IP address is 192.168.178.1, where "192.168.178." identifies the network and "1" helps you identify the Host.

Class D Address

- The **higher-order bits** of the first octet of IP addresses belonging to class D is always set to **1110**.
- Class D address range from 224.0.0.0 to 239.255.255.255.
- **Reserved for multicast groups** and is **not used for unicast communication**.
- Multicast addresses are used to **send data to multiple hosts simultaneously**.

Class E Address

- The **higher-order bits** of the first octet of class E are always set to **1111**.
- Class E address range from 240.0.0.0 to 255.255.255.255)
- Reserved for **experimental and research purposes**.
- **Not used for general IP communication**.

Activity Time:**GATE****Q1**

Identify the **valid and invalid IP address** in the following set, If **invalid** write the reason

- A. 24.25.26.8
- B. 10.3.156.256
- C. 0.0.0.0
- D. 255.255.255.255
- E. 100.2.3.345.456
- F. 16.2e.54.67
- G. 111.064.25.4
- H. 10111010.2.24.36

Activity Time:

Q2

The Dotted Decimal Notation(DDN) format for the given Hexadecimal Notation(HDN) C22F1582 is

- A. 194.50.21.145
- B. 194.47.21.130
- C. 194.45.21.120
- D. 194.47.20.130

Activity Time:

Q2

The Dotted Decimal Notation(DDN) format for the given Hexadecimal Notation(HDN) C22F1582 is

- A. 194.50.21.145
- B. 194.47.21.130**
- C. 194.45.21.120
- D. 194.47.20.130

Activity Time:

Q3

Identify the no.of Networks, no.of Hosts per network in Class B IP addressing format.

- A. $2^{16}, 2^{16}$
- B. $2^{14}, 2^{16}$
- C. $2^{16}, 2^{14}$
- D. $2^{14}, 2^{16}-2$

Activity Time:

Q3

Identify the no.of Networks, no.of Hosts per network in Class B IP addressing format.

- A. $2^{16}, 2^{16}$
- B. $2^{14}, 2^{16}$
- C. $2^{16}, 2^{14}$
- D. $2^{14}, 2^{16-2}$

Activity Time:

Q4

In IPV4 addressing format, the number of networks allowed under Class C is ?

- A. 2^{14}
- B. 2^7
- C. 2^{21}
- D. 2^{24}

Activity Time:

Q4

In IPV4 addressing format, the number of networks allowed under Class C is ?

- A. 2^{14}
- B. 2^7
- C. 2^{21}
- D. 2^{24}

Activity Time:

Q5

Suppose, instead of using 16 bits for network part of a Class B, 20 bits had been used. Then the number of Networks and Hosts per Network are

- A. $2^{10}, 2^{12}$
- B. $2^{18}, 2^{12}$
- C. $2^{18}, 2^{12} - 2$
- D. $2^{10}, 2^{12}-2$

Activity Time:

Q5

Suppose, instead of using 16 bits for network part of a Class B, 20 bits had been used. Then the number of Networks and Hosts per Network are

- A. $2^{10}, 2^{12}$
- B. $2^{18}, 2^{12}$
- C. **$2^{18}, 2^{12} - 2$**
- D. $2^{10}, 2^{12}-2$

UNIT-4

Transport Layer



Syllabus

★ The Transport Layer

- **Transport service**
- **Elements of transport protocol**
- **Simple Transport Protocol**
- **Internet transport layer protocols: UDP and TCP**

THE TRANSPORT LAYER SERVICES

1. Services Provided to the Upper Layers

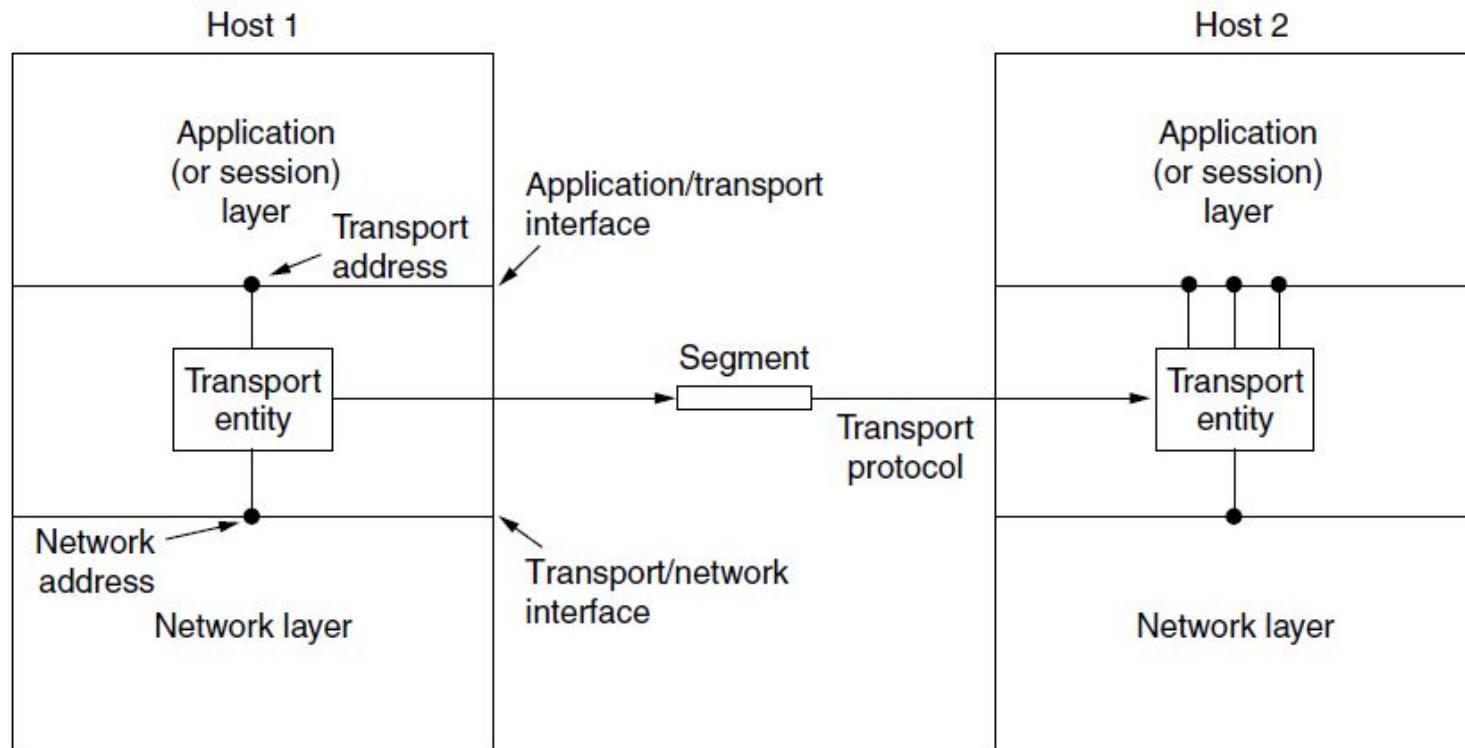


Figure 6-1. The network, transport, and application layers.

- The ultimate goal of the transport layer is to provide **efficient, reliable, and cost-effective data transmission service** to its users, normally processes in the application layer.
- To achieve this, the transport layer makes use of the services provided by the network layer.
- The software and/or hardware within the transport layer that does the work is called **the transport entity**.
- The transport entity can be located in the operating system kernel, in a library package bound into network applications, in a separate user process, or even on the network interface card.
- The (logical) relationship of the network, transport, and application layers is illustrated in Fig. 6-1.

Working of Transport Layer:

The transport layer **takes services from the Application layer** and **provides services to the Network layer** .

At the sender's side	At the receiver's side
<ul style="list-style-type: none">• The transport layer receives data (message) from the Application layer and then performs Segmentation, divides the actual message into segments, adds the source and destination's port numbers into the header of the segment, and transfers the message to the Network layer.	<ul style="list-style-type: none">• The transport layer receives data from the Network layer, reassembles the segmented data, reads its header, identifies the port number, and forwards the message to the appropriate port in the Application layer.

Responsibilities of a Transport Layer:

- The Process to Process Delivery
- End-to-End Connection between Hosts
- Multiplexing and Demultiplexing
- Congestion Control
- Error Control
- Flow control

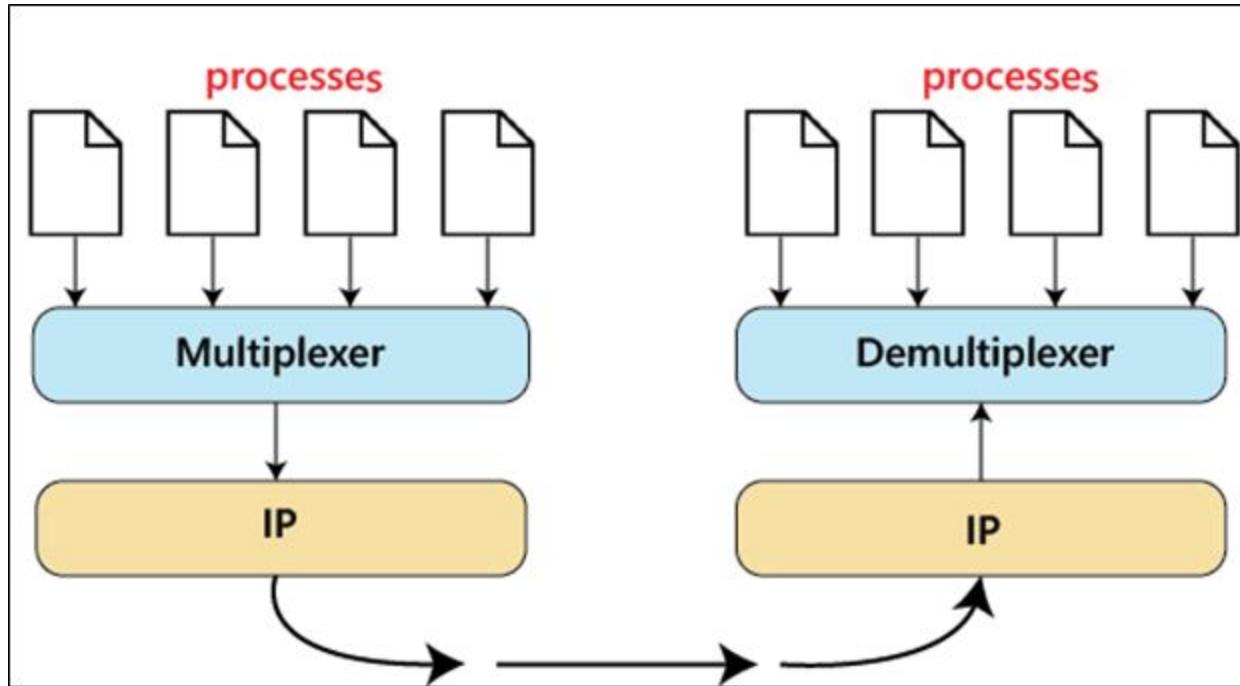
The Process to Process Delivery



1. The Process to Process Delivery

- While
 - **Data Link Layer** requires the **MAC address** of source-destination hosts to correctly deliver **a frame**,
 - **The Network layer** requires **the IP address** for appropriate routing of packets,
 - In a similar way **Transport Layer** requires **a Port number** to **correctly deliver the segments of data to the correct process** amongst the multiple processes running on a particular host.
 - **A port number** is a **16-bit address** used to identify any client-server program uniquely.

Process to Process Delivery



PORT?

- The transport layer is responsible for ensuring the reliable transfer of data between two devices in a network.
- **Ports** play a crucial role in the transport layer, and **they are used to facilitate communication between applications or services running on different devices.**

Port Definition:

- ★ A port is a **16-bit unsigned integer (ranging from 0 to 65535)** that serves as an endpoint for communication.
- ★ Ports are used to distinguish between different services or applications running on the same device, allowing multiple services to operate concurrently.
- ★ For example, a **web server** typically listens on **port 80** for **HTTP requests**, while an **email server** might use **port 25** for **SMTP** (Simple Mail Transfer Protocol) communication.
- ★ A combination of an **IP address and a port number** is referred to as **a socket**.

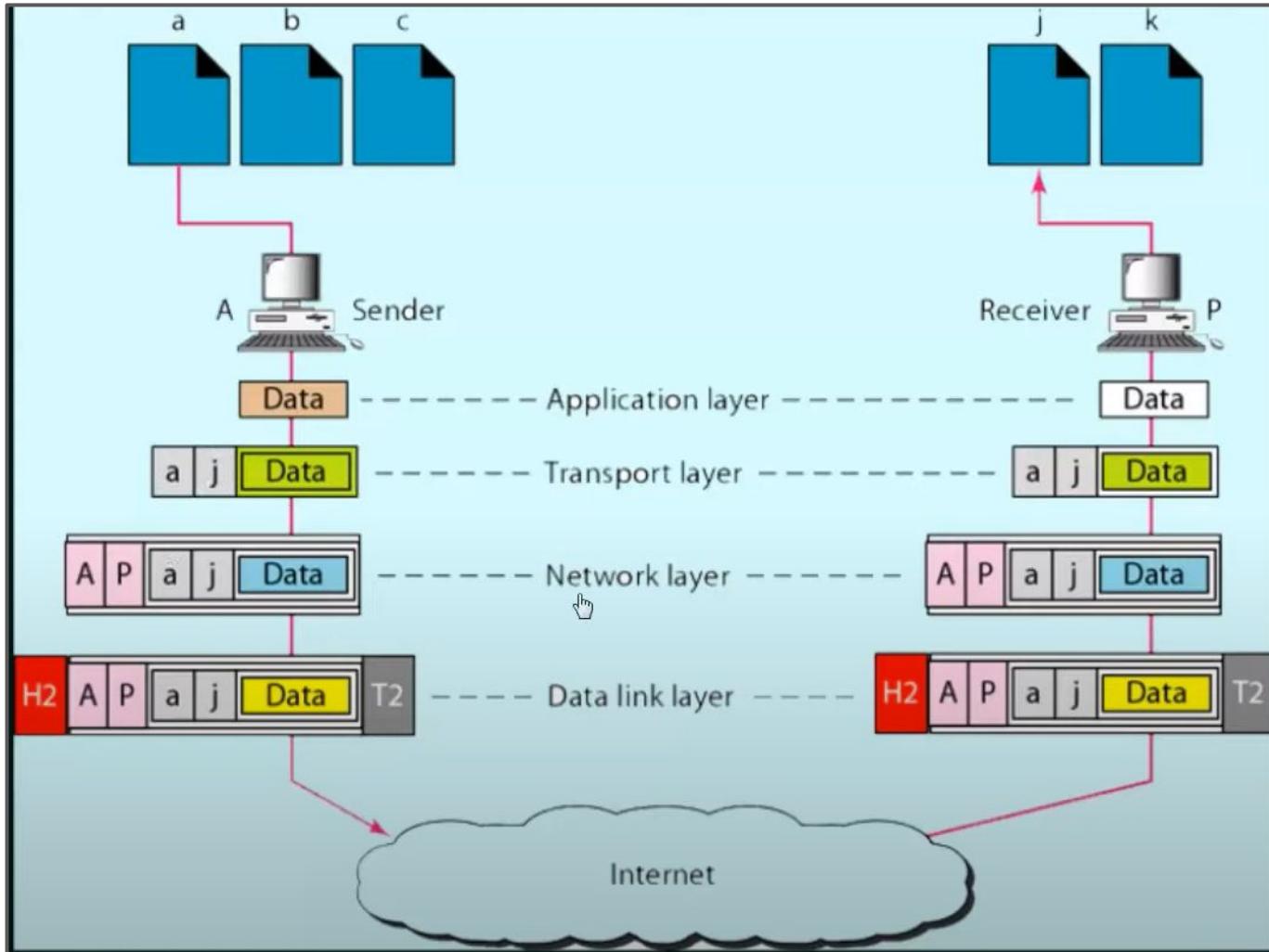
Difference between PORT,IP and MAC Address

Suppose your friends want to send a parcel and he/she is in America and you are in Bhimavaram.

- ★ Reaching your CITY = Reaching Your NETWORK (IP Address)
- ★ Reaching Your APARTMENT = Reaching the HOST (MAC Address)
- ★ Reaching the RIGHT PERSON = Reaching the right PROCESS (PORT Address)



Relation between PORT, IP, MAC Address



2. End-to-end Connection between Hosts

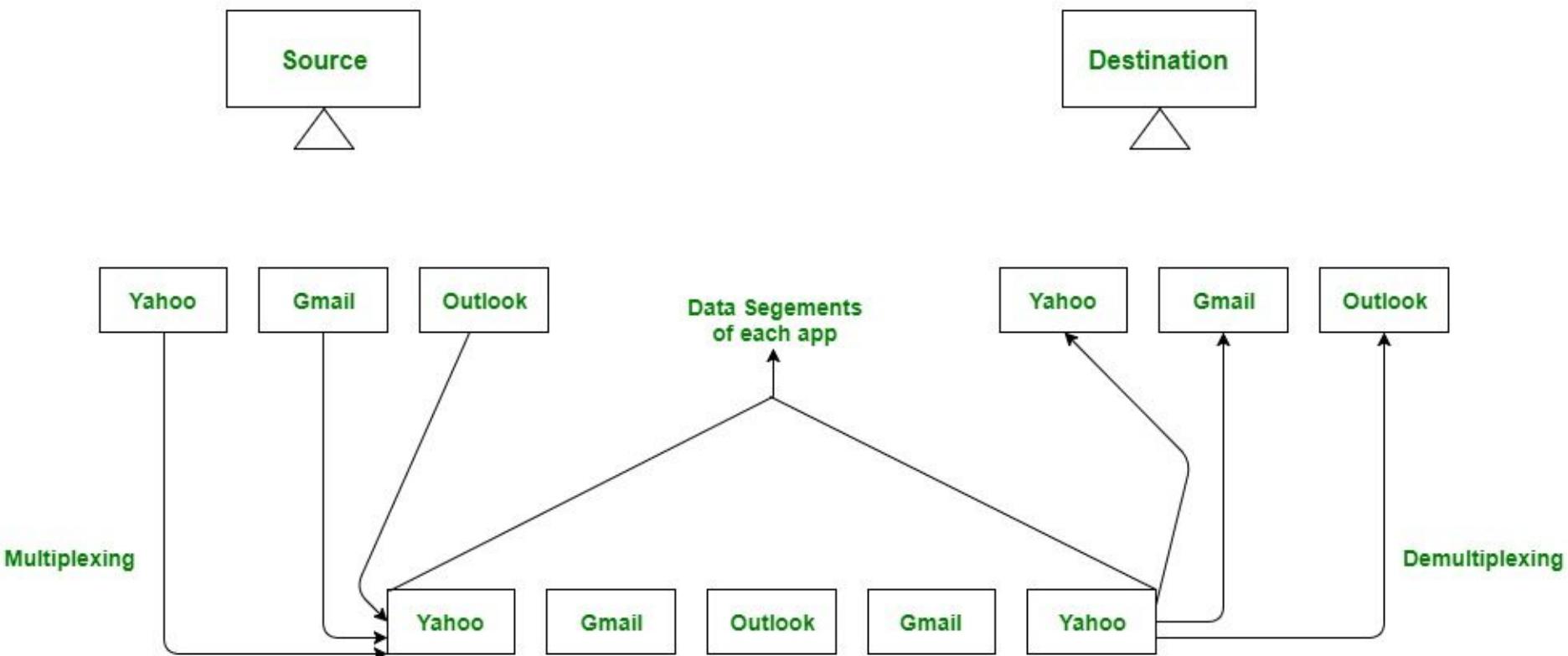
- The transport layer is also responsible for **creating the end-to-end Connection between hosts** for which it mainly uses **TCP and UDP**.
- **TCP** is a secure, connection-oriented protocol that uses a handshake protocol to establish a robust connection between two end hosts.
 - TCP ensures **the reliable** delivery of messages and is used in various applications.
- **UDP**, on the other hand, is a stateless and **unreliable protocol** that ensures best-effort delivery.
 - It is suitable for applications that have little concern with flow or error control and requires sending the bulk of data like video conferencing.
 - It is often used in multicasting protocols.

3. Multiplexing and Demultiplexing



“Multiplexing and Demultiplexing, enable the transport layer to manage multiple communication streams over a single network connection, ensuring that data from different applications or sources can be sent and received correctly.”

Multiplexing / Demultiplexing



1. Multiplexing (Muxing):

Definition: Multiplexing is the process of combining multiple data streams or communication channels into a single data stream for transmission over a network.

Purpose: It allows multiple applications or communication sources to share the same network connection efficiently.

How it works: When data is sent from different applications or sources on a device, the transport layer adds headers or tags to each data segment. These headers contain information that helps identify the source or destination of the data. The transport layer then combines these data segments into a single stream, often referred to as a "multiplexed stream" or "multiplexed connection," before transmitting it over the network.

2. Demultiplexing (Demuxing):

Definition: Demultiplexing is the process of separating the combined multiplexed data stream into individual data streams or communication channels upon reception.

Purpose: It ensures that data from different sources can be correctly routed to their respective applications or destinations.

How it works: When data is received over the network, the transport layer examines the headers or tags in the incoming data to determine which application or source each segment belongs to. Based on this information, it separates the multiplexed data stream into individual data streams, each destined for a specific application or source. These individual data streams are then delivered to the appropriate higher-layer protocols or applications.

Elements of Transport Protocol

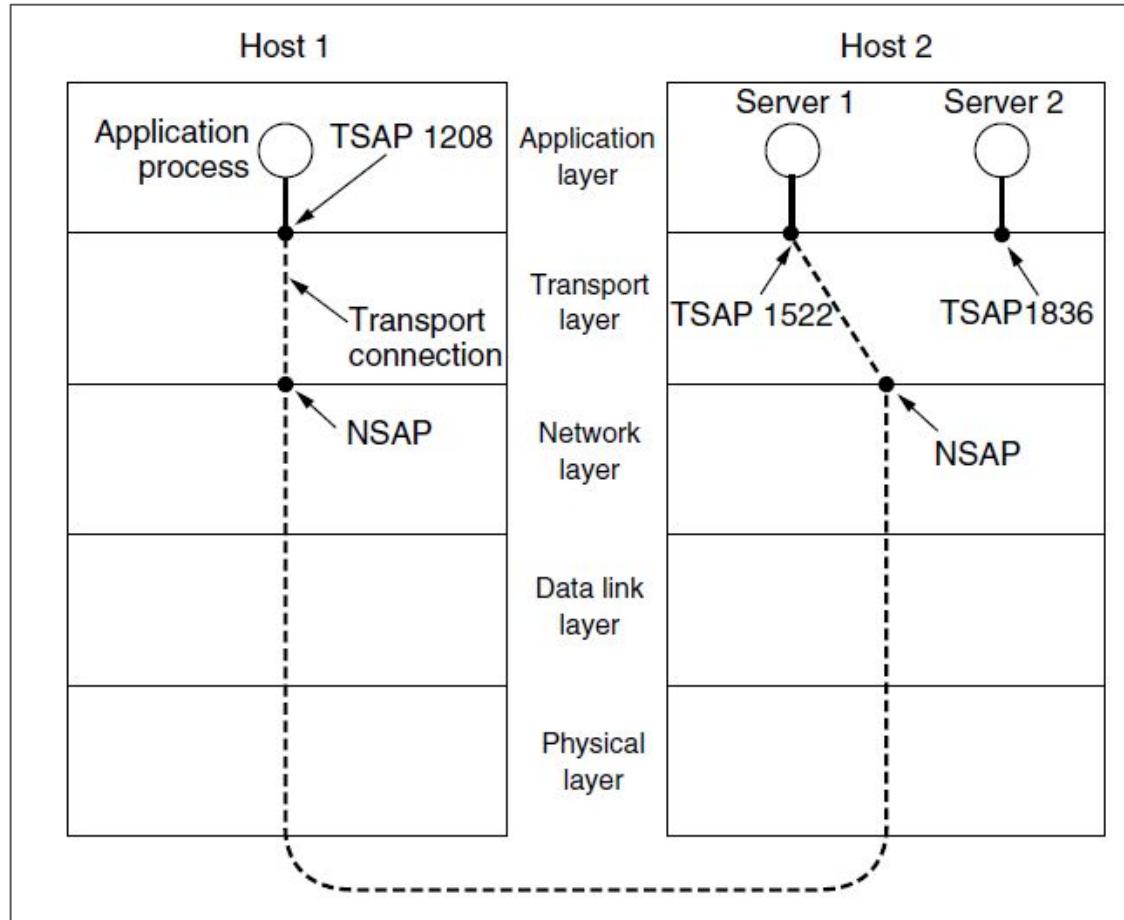
Elements of Transport Protocol

- **Addressing**
- **Connection Establishment**
- **Connection Release**
- **Flow Control and Buffering**
- **Multiplexing**

Addressing

- When an application (e.g., a user) process wishes to set up a connection to a remote application process, **it must specify which one to connect to.**
- The method normally used is **to define transport addresses to which processes can listen for connection requests.** In the Internet, these endpoints are called **ports.**
- We will use the generic term **TSAP** (Transport Service Access Point) to mean a specific endpoint in the transport layer.
- The analogous endpoints in the network layer (i.e., network layer addresses) are called NSAPs (Network Service Access Points). **IP addresses** are examples of **NSAPs.**
- Figure 6-8 illustrates the relationship between the NSAPs, the TSAPs, and a transport connection.
- Application processes, both clients and servers, can attach themselves to a local TSAP to establish a connection to a remote TSAP.
- These connections run through NSAPs on each host, as shown. The purpose of having TSAPs is that in some networks, each computer has a single NSAP, so some way is needed to distinguish multiple transport endpoints that share that NSAP.

The Relationship Between TSAPs, NSAPs and Transport Connections



A possible scenario for a transport connection is as follows:

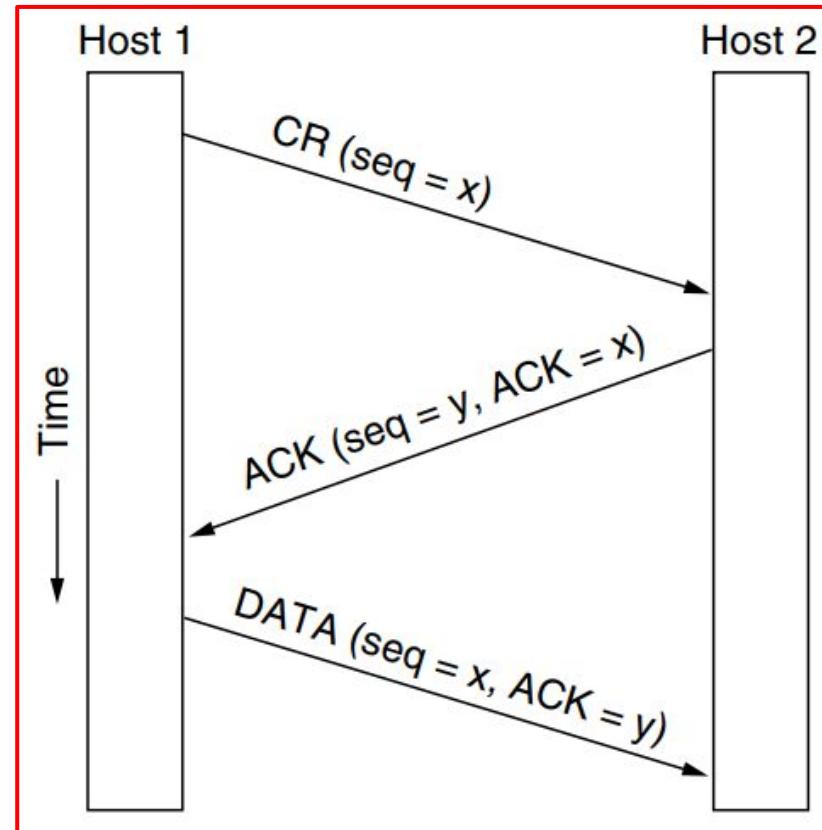
1. A **mail server** process attaches itself to **TSAP 1522** on host 2 to wait for an incoming call.
2. An **application process** on host 1 wants to send an email message, so it attaches itself to **TSAP 1208** and issues a **CONNECT request**. The request specifies **TSAP 1208 on host 1 as the source and TSAP 1522 on host 2 as the destination**. This action ultimately results in a transport connection being established between the application process and the server.
3. The application process sends over the mail message.
4. The mail server responds to say that it will deliver the message.
5. The transport connection is released.

Connection Establishment

- Establishing a connection sounds easy, but it is actually surprisingly tricky.
- At first glance, it would seem sufficient for one transport entity to just send a **CONNECTION REQUEST** segment to **the destination** and wait for a **CONNECTION ACCEPTED** reply.
- **The problem occurs** when the network can lose, delay, corrupt, and duplicate packets. This behavior causes serious complications.
- Imagine a subnet that is so congested that acknowledgements hardly ever get back in time and each packet times out is retransmitted two or more times
- **Tomlinson (1975) introduced the three-way handshake,** to handle **duplications** in transmission process.

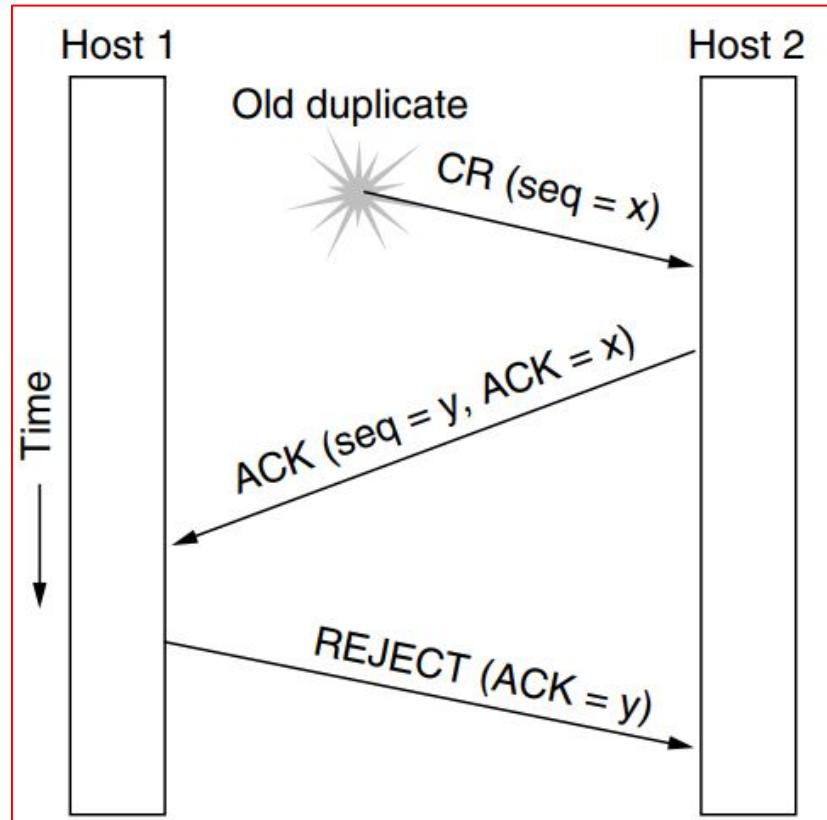
Normal Procedure

- Host 1 chooses a sequence number x , and sends a CONNECTION REQUEST segment containing it to host 2.
- Host 2 replies with an ACK segment acknowledging x and announcing its own initial sequence number, y .
- Finally, host 1 acknowledges host 2's choice of an initial sequence number in the first data segment that it sends.



Abnormal situation: Old duplicate CONNECTION REQUEST appearing out of nowhere.

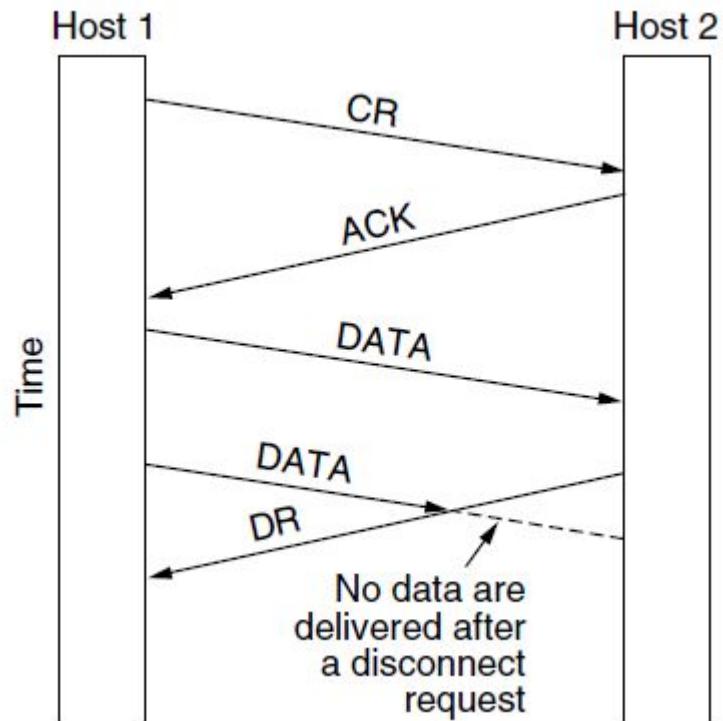
- The first segment is a **delayed duplicate CONNECTION REQUEST** from **an old connection**.
- This segment arrives at host 2 **without host 1's knowledge**.
- Host 2 reacts to this segment by sending host 1 an ACK segment, in effect asking for verification that host 1 was indeed trying to set up a new connection.
- When host 1 rejects host 2's attempt to establish a connection, host 2 realizes that it was tricked by a delayed duplicate and abandons the connection.
- In this way, a delayed duplicate does no damage.



Connection Release

- Releasing a connection is easier than establishing one.
- There are two styles of terminating a connection:
 - **Asymmetric release**
 - **Symmetric release**
- **Asymmetric release** is a method of terminating a network connection **where one side of the connection initiates the release without waiting for the other side to confirm or acknowledge the release.** It may result in **data loss**.
- **Symmetric release**, on the other hand, is a method **where both endpoints of a network connection agree to release the connection simultaneously.**

Asymmetric Release



Abrupt disconnection with loss of data

Symmetric Release

One way to **avoid data loss** is to use symmetric release, in which **each direction** is released independently of the other one.

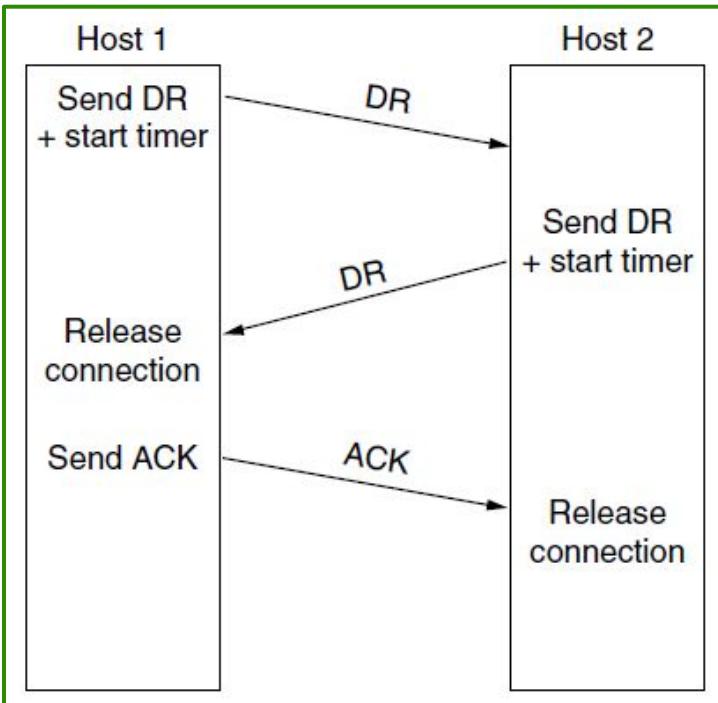
Eg:

- Host 1: I am done, are you done too?
- Host 2: I am done too, goodbye

Figures: Illustrates four scenarios of releasing using a three-way handshake.

- (a) Normal case of three-way handshake**
- (b) Final ACK lost**
- (c) Response lost**
- (d) Response lost and subsequent DRs lost**

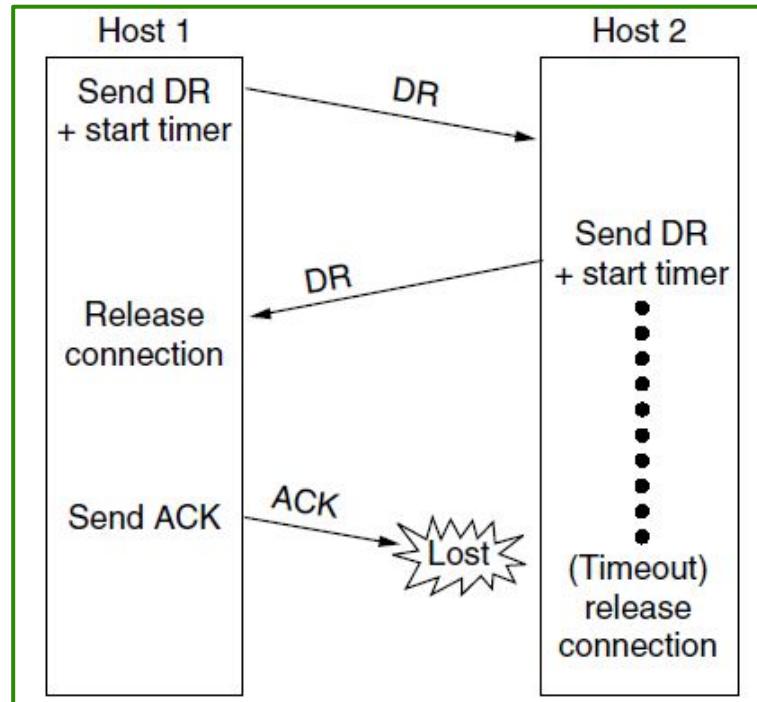
Normal case



[a]

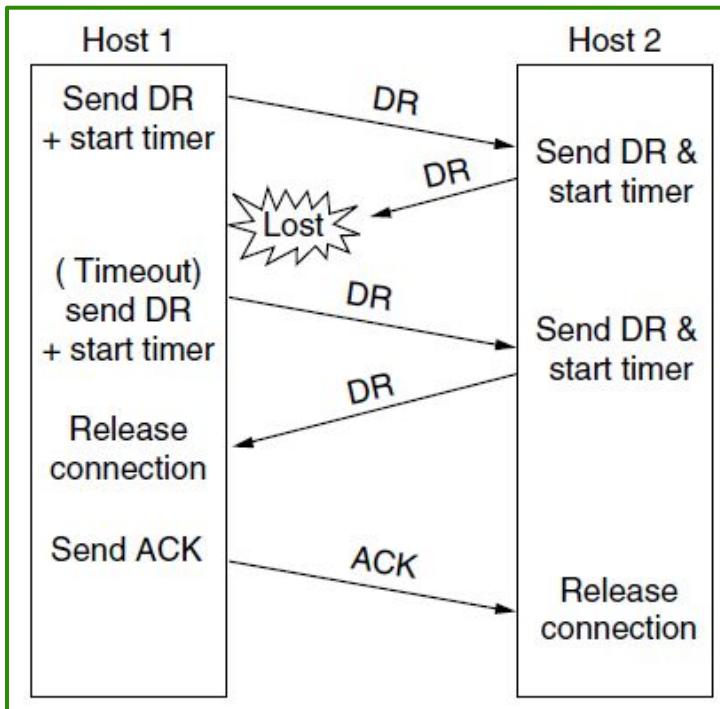
DR-DISCONNECTION REQUEST

Final ACK lost



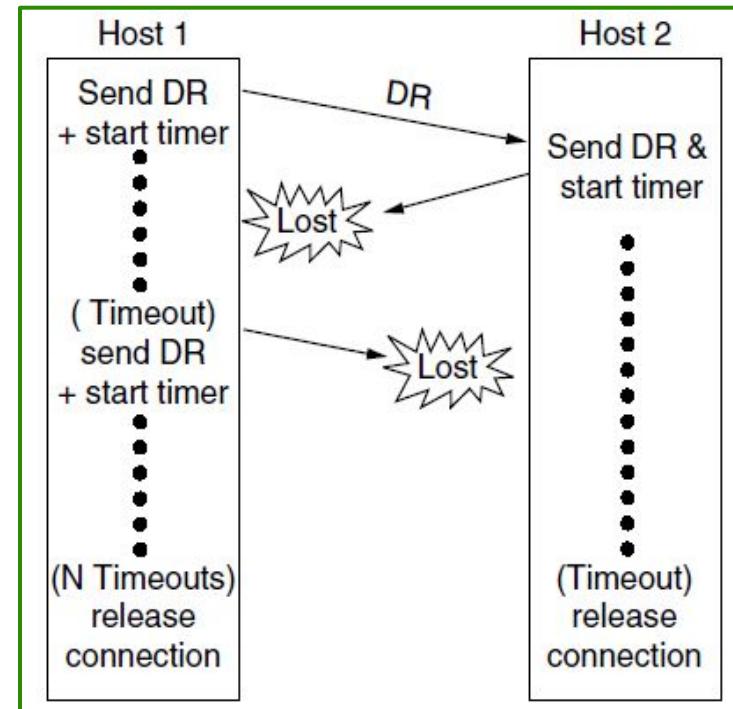
[b]

Response lost



[c]

Response lost and subsequent DRs lost

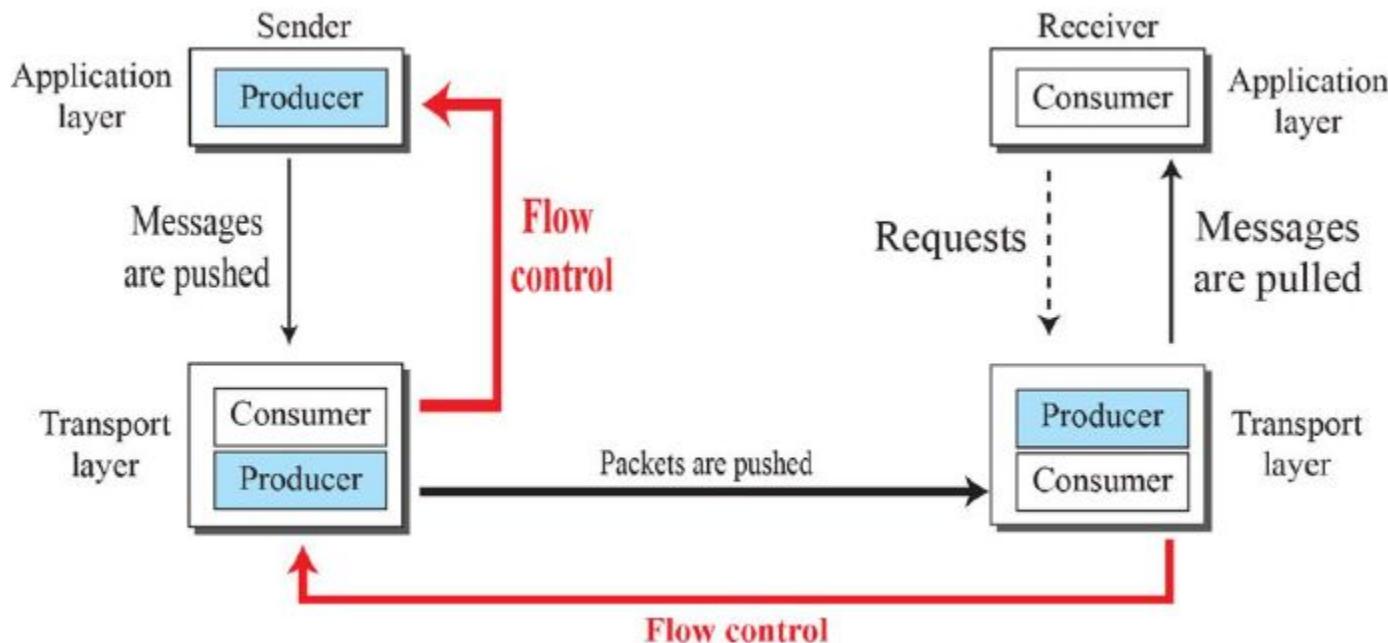


[d]

- In Fig. 6-14(a), we see the normal case in which one of the users sends a DR (DISCONNECTION REQUEST) segment to initiate the connection release. When it arrives, the recipient sends back a DR segment and starts a timer, just in case its DR is lost. When this DR arrives, the original sender sends back an ACK segment and releases the connection. Finally, when the ACK segment arrives, the receiver also releases the connection. Releasing a connection means that the transport entity removes the information about the connection from its table of currently open connections and signals the connection's owner (the transport user) somehow. This action is different from a transport user issuing a DISCONNECT primitive.
- If the final ACK segment is lost, as shown in Fig. 6-14(b), the situation is saved by the timer. When the timer expires, the connection is released anyway. Now consider the case of the second DR being lost. The user initiating the disconnection will not receive the expected response, will time out, and will start all over again.
- In Fig. 6-14(c), we see how this works, assuming that the second time no segments are lost and all segments are delivered correctly and on time.
- Our last scenario, Fig. 6-14(d), is the same as Fig. 6-14(c) except that now we assume all the repeated attempts to retransmit the DR also fail due to lost segments. After N retries, the sender just gives up and releases the connection. Meanwhile, the receiver times out and also exits.

Flow Control and Buffering

Flow control at the transport layer



Flow Control at Transport Layer:

- In communication at the transport layer, we are dealing with **four entities**:
 - Sender Process,
 - Sender Transport Layer,
 - Receiver Transport Layer
 - Receiver Process
- The sending process at the application layer is only **a producer**. It produces message chunks and pushes them to the transport layer. The sending transport layer has a double role: it is both **a consumer and a producer**.
- It consumes the messages pushed by the producer. It encapsulates the messages in packets and pushes them to the receiving transport layer.
- The receiving transport layer also has a double role: it is the consumer for the packets received from the sender and the producer that decapsulates the messages and delivers them to the application layer.

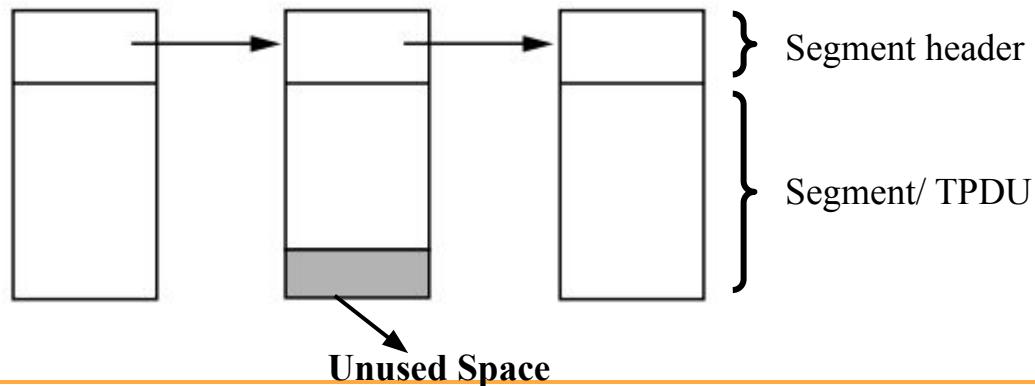
Buffers

- Flow control in the transport layer is a mechanism used to **manage the rate of data transmission** between two devices to ensure that the sender does not overwhelm the receiver with data.
- Buffering is one of the methods employed in flow control to achieve this goal. Buffering involves the use of temporary storage, called buffers, at various points in the communication process to hold data temporarily until it can be transmitted or processed.
- **A buffer** is a set of memory locations that can hold segments at the sender side and receiver side.
- **Sender Buffer:** The sender maintains a sender buffer that stores the data it wants to transmit. When the sender application generates data, it places it into the sender buffer.
- **Receiver Buffer:** The receiver maintains a receiver buffer to temporarily hold incoming data. As data arrives at the receiver, it is placed into the receiver buffer. The receiver then processes the data from the buffer at its own pace.

- The sender buffers all the TPDUs sent to the receiver. The buffer size varies for different TPDUs. The buffer size varies for different TPDUs.
- They are:
 - **Chained Fixed-size Buffers**
 - **Chained Variable-size Buffers**
 - **One large Circular Buffer per Connection**

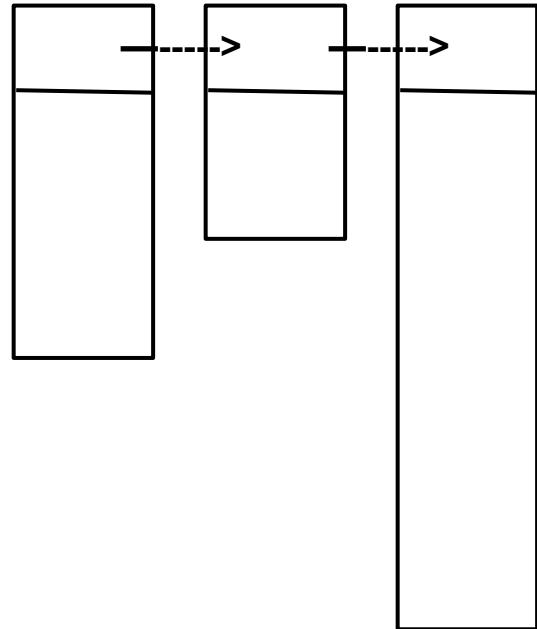
Chained Fixed-size Buffers

- If most segments are nearly **the same size**, it is natural to organize the buffers as a pool of identically sized buffers, with one segment per buffer



Chained Variable-size Buffers:

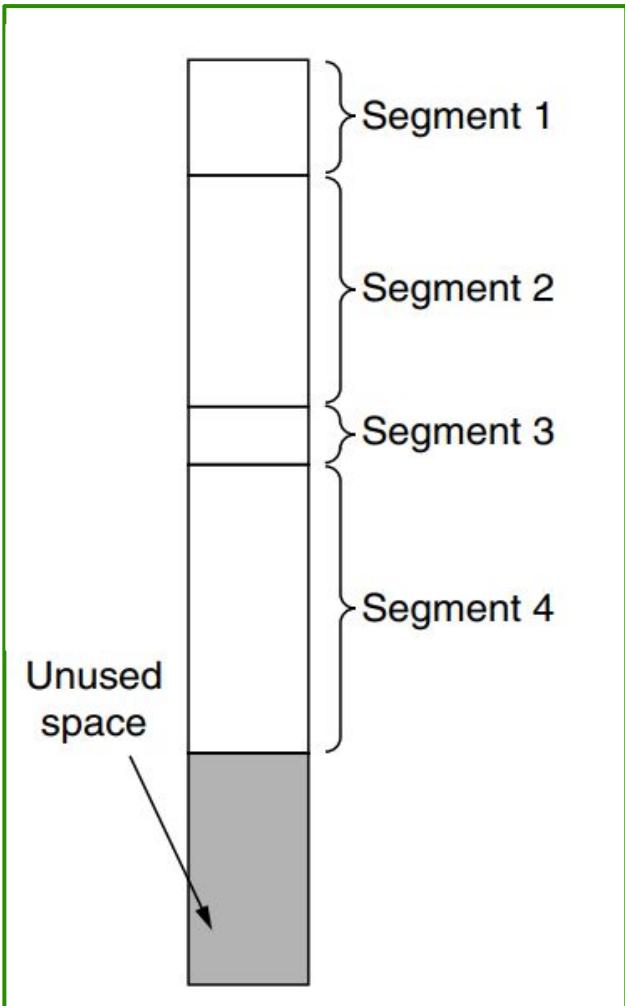
- Chained variable-size buffers extend the concept of chained fixed size buffers but allow for buffers of varying sizes to be linked together.
- Unlike fixed-size buffers, variable-size buffers can adapt to the size of the data being processed.
- This can be more **memory-efficient** when dealing with data of varying sizes.



Chained Variable-size Buffers

One large Circular Buffer per Connection:

- A single large circular buffer per connection is dedicated when all connections are heavily loaded.
- This system is simple and elegant and does not depend on segment sizes, but makes good use of memory only when the connections are heavily loaded.



Multiplexing

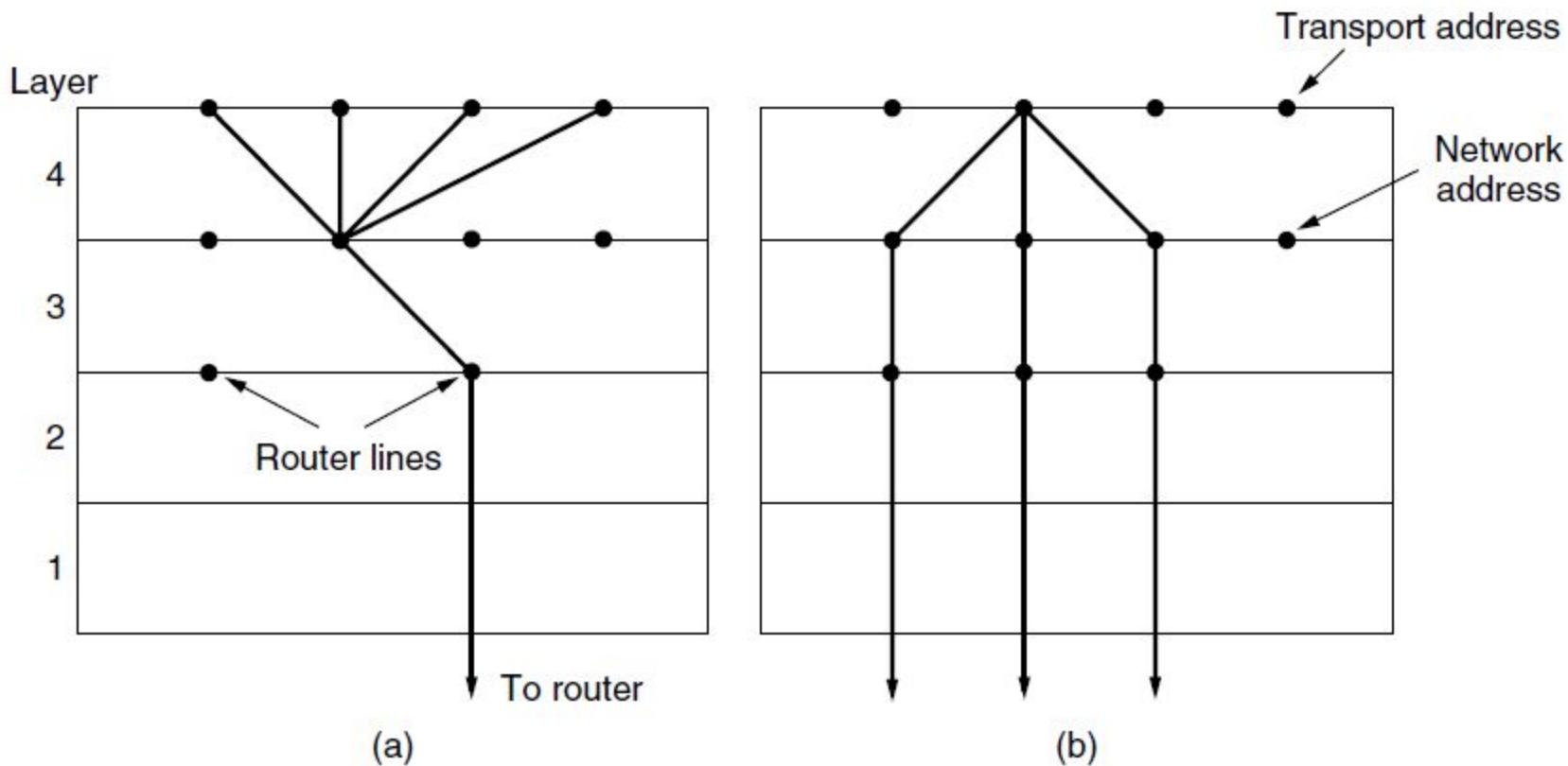
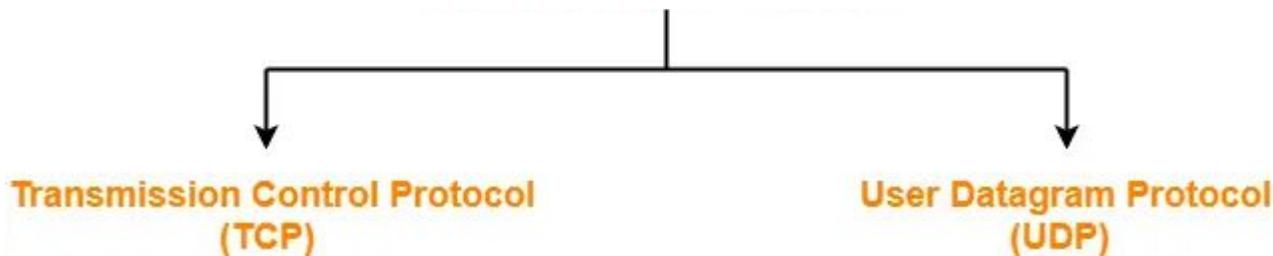


Figure 6-17. (a) Multiplexing. (b) Inverse multiplexing.

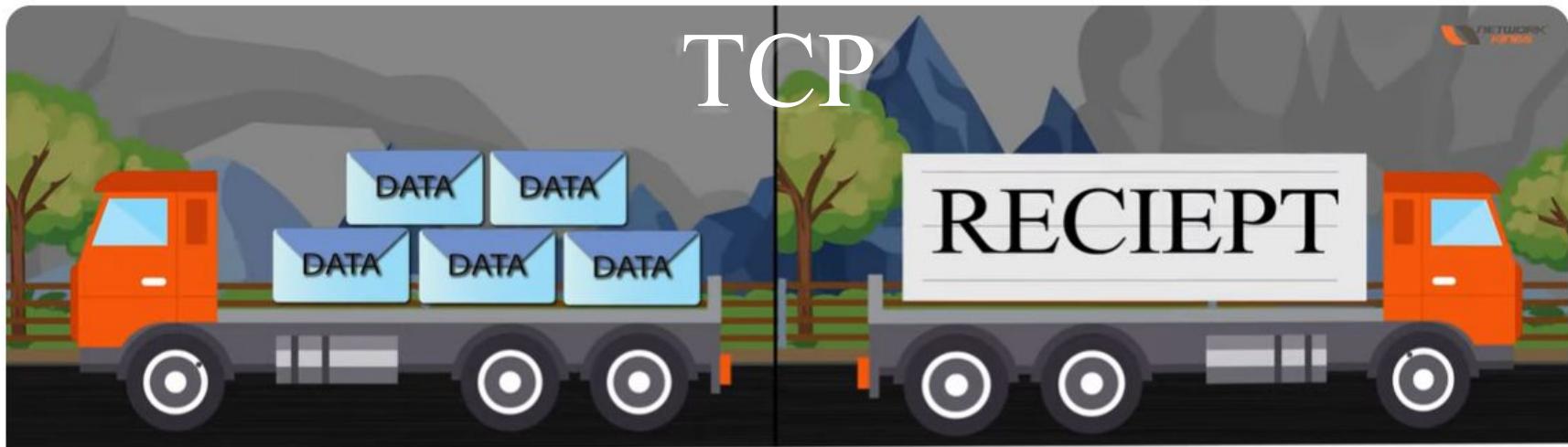
In the transport layer, the need for multiplexing can arise in a number of ways.

- **Case-I** : If only one network address is available on a host, all transport connections on that machine have to use it. When a segment comes in, some way is needed to tell which process to give it to. This situation, called multiplexing, is shown in Fig. 6-17(a).
- In this figure, four distinct transport connections all use the same network connection (e.g., IP address) to the remote host.
- **Case-II** : A host may have access to multiple network paths or links, each with different characteristics, such as bandwidth or reliability. **If a user requires more bandwidth** than a single network path can offer, a technique called inverse multiplexing can be employed.
- Inverse multiplexing involves distributing network traffic among multiple network paths on **a round-robin or load-balancing basis**, effectively combining the resources of these paths.
- Fig. 6-17(b) illustrates **inverse multiplexing**, where multiple network paths are used to enhance the overall performance.

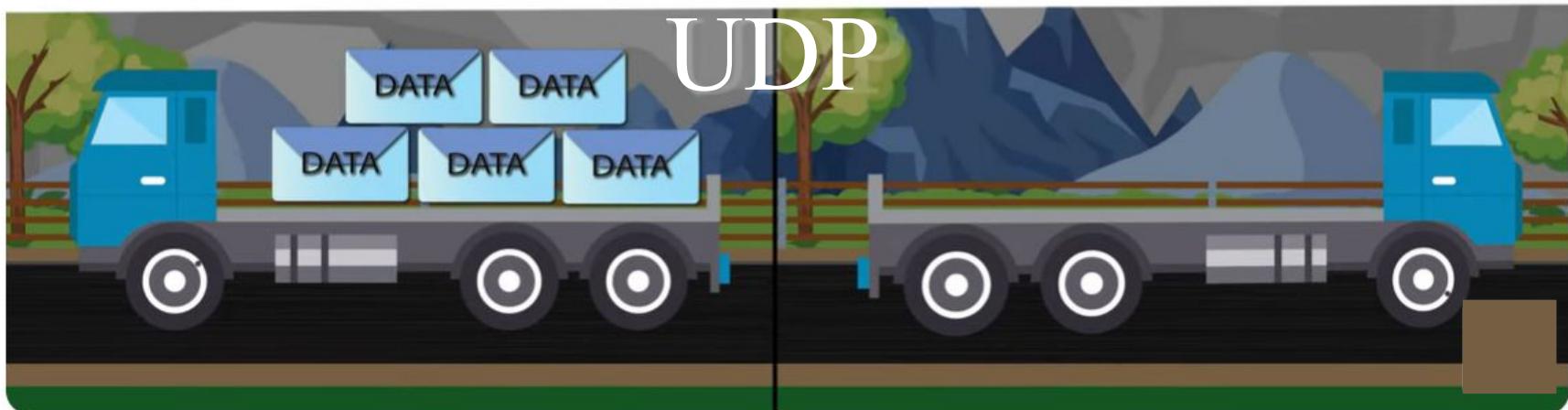
Transport Layer Protocols



TCP



UDP



UDP: User Datagram Protocol

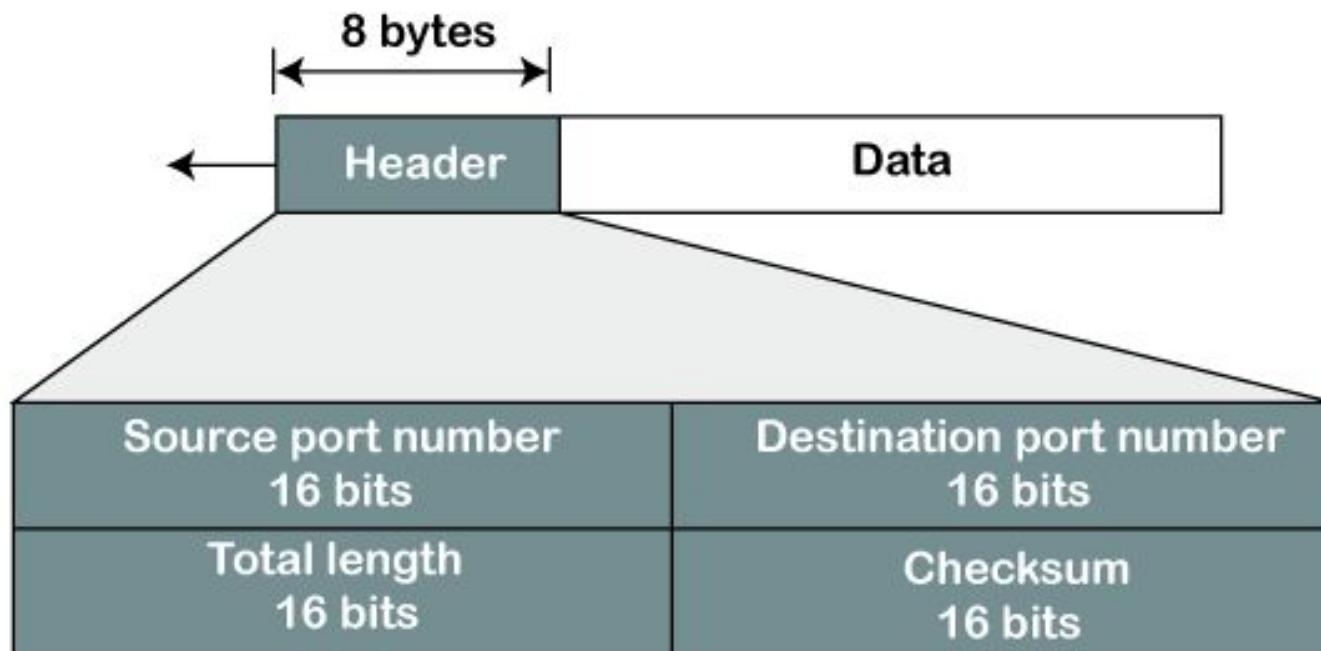
- UDP is a part of the **Internet Protocol suite**, referred to as UDP/IP suite.
- Both UDP and TCP are transport layers protocols which are used on the Internet or run on the top of the Internet Protocol (IP) and commonly known as **UDP/IP** and **TCP/IP**, respectively.
- Unlike TCP, it is **an unreliable** and **connectionless protocol**. So, there is **no need to establish a connection** prior to data transfer.
- Compared to TCP, the UDP is the **simplest transport layer protocol** designed to send data over the Internet.
- It picks the datagram from the network layer and attaches the header then forwards it to the user.

Characteristics of UDP:

- It is **a fast, unreliable, and stateless protocol** that makes it suitable for use with applications that can tolerate lost data.
- It can be used for transaction-based protocols, such as **DNS ,VoIP**(Voice over Internet Protocol) etc..
- UDP supports **broadcast and multicast communication**, allowing a single packet to be sent to multiple recipients simultaneously.
- For real-time services like **computer gaming, voice or video communication, live conferences we need UDP, Since high performance is needed.**
- There is **no error checking** in UDP.
- It is a connectionless protocol as **it doesn't need a virtual circuit before transferring the data.**

UDP Header Format:

UDP Header Format



IPv4 Pseudo Header

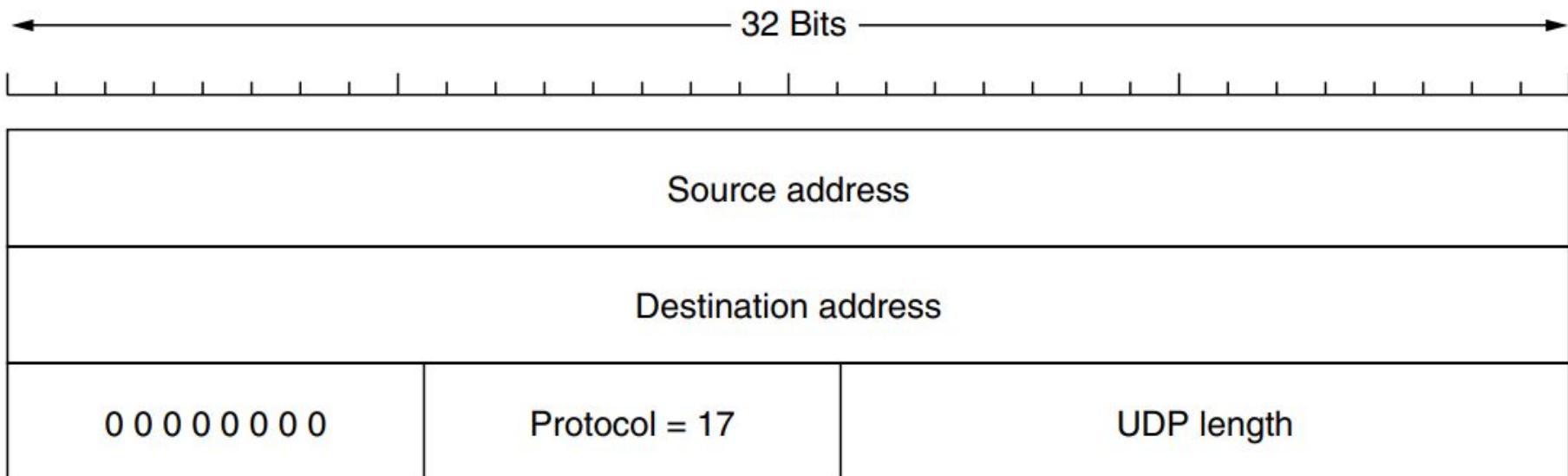


Figure 6-28. The IPv4 pseudoheader included in the UDP checksum.

UDP Header Format:

- **Source Port:** Source Port is a **2 Byte long field** used to identify the port number of the source.
- **Destination Port:** It is a **2 Byte long field**, used to identify the port number of the destination
- **Length:** Length is a **16 bit field/ 2 Bytes.**
 - It identifies the combined length of UDP Header and Encapsulated data.
 - $\text{Length} = \text{Length of UDP Header} + \text{Length of data}$
- **Checksum:** Checksum is 2 Bytes long field.

Checksum:

The UDP checksum is a 16-bit field that provides a form of error detection for the data contained in the UDP datagram. Here's how the UDP checksum field works:

- **Error Detection:** The primary purpose of the UDP checksum is to detect errors in the UDP datagram during transmission. It helps ensure the integrity of the data being sent.
- **Calculation:** The checksum is calculated based on **the contents of the UDP datagram, including the UDP header, UDP data, and a pseudo-header**. The pseudo-header includes the source and destination IP addresses, the protocol number (which is 17 for UDP), and the UDP length.
- **Algorithm:** The calculation of the checksum uses a simple mathematical algorithm called **the one's complement sum**. It involves summing up all 16-bit words in the data, taking the one's complement of the sum, and placing it in the checksum field.

- **Verification:** Upon receiving a UDP datagram, the recipient recalculates the checksum based on the received data, including the UDP header, data, and pseudo-header. It then compares the calculated checksum with the checksum value included in the received UDP header.
- **Discard on Error:** If the calculated checksum does not match the value in the checksum field of the received UDP datagram, the datagram is considered to have an error, and it may be discarded. This indicates that the data may have been corrupted during transmission.

TCP(Transmission Control Protocol)

- ★ **TCP (Transmission Control Protocol)** is one of the main protocols of the Internet protocol suite. It lies between the Application and Network Layers which are used in providing **reliable delivery services**.
- ★ It is **a connection-oriented protocol** that means it establishes the connection prior to the communication that occurs between the computing devices in a network.
- ★ This protocol is used with an IP protocol, so together, they are referred to as a TCP/IP.

Features of TCP protocol:

1. Reliable :

That is, the receiver always sends either **positive or negative acknowledgement** about the data packet to the sender, so that the sender always has bright clue about whether the data packet is reached the destination or it needs to resend it.

2. Order of the data is maintained :

TCP ensures that the data reaches intended destination in the same order it was sent.

3. Connection-oriented:

It is a connection-oriented service that means the data exchange occurs only after the connection establishment. When the data transfer is completed, then the connection will get terminated.

4. Full duplex:

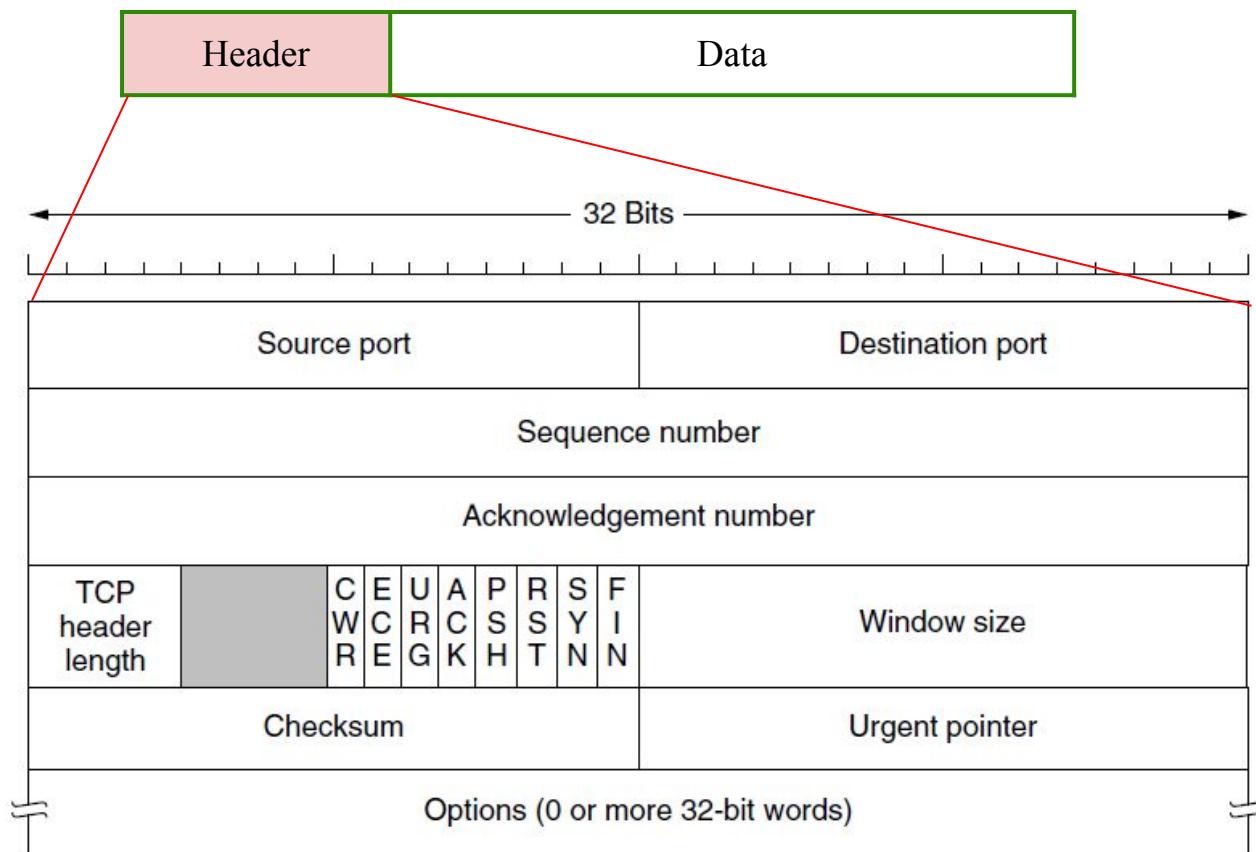
In TCP data can be transmitted from receiver to the sender or vice versa at the same time.

5.TCP provides error-checking and recovery mechanism.

6. TCP provides congestion control.

7. TCP provides flow control and quality of service.

The TCP Header Format

TCP Header Format:**< 20-60 Bytes >**

1. Source Port-

- Source Port is a 16 bit field.
- It identifies the port of the sending application.

2. Destination Port-

- Destination Port is a 16 bit field.
- It identifies the port of the receiving application.

3. Sequence Number-

- Sequence number is a 32 bit field.
- TCP assigns a unique sequence number to each byte of data contained in the TCP segment.
- This field contains the sequence number of the first data byte.
- It is used to reassemble the message at the receiving end of the segments that are received out of order.

4. Acknowledgement Number-

- Acknowledgment number is a 32 bit field.
- It contains sequence number of the data byte that receiver **expects to receive next from the sender.**
- **It is always sequence number of the last received data byte incremented by 1.**

5.TCP Header Length:

- Header length is a 4 bit field.
- This is a 4-bit field that indicates the length of the TCP header by **a number of 4-byte words** in the header, i.e if the header is 20 bytes(min length of TCP header), then this field will hold 5 (because $5 \times 4 = 20$) and the maximum length: 60 bytes, then it'll hold the value 15(because $15 \times 4 = 60$). Hence, the value of this field is always between 5 and 15.

Eight 1-bit flags:(control connection establishment, connection termination, connection abortion, flow control, mode of transfer etc)

C	E	U	A	P	R	S	F
W	C	R	C	S	S	Y	I
R	E	G	K	H	T	N	N

1. ECE(Explicit Congestion Notification)

- ECE is set to signal an ECN-Echo to a TCP sender to tell it to slow down when the TCP receiver gets a congestion indication from the network.

2. CWR(Congestion Window Reduced)

- CWR is set to signal Congestion Window Reduced from the TCP sender to the TCP receiver so that it knows the sender has slowed down and can stop sending the ECN-Echo.

3. URG(Urgent pointer):

- URG is set to 1 if the Urgent pointer is in use.
- The Urgent pointer is used to indicate a byte offset from the current sequence number at which urgent data are to be found.

4. ACK(Acknowledgement number)

- The ACK bit is set to 1 to indicate that the Acknowledgement number is valid.

5. PSH(Push)

When PSH bit is set to 1,

- All the segments in the buffer are immediately pushed to the receiving application.
- No wait is done for filling the entire buffer.
- This makes the entire buffer to free up immediately.

6. RST(Reset):

RST bit is used to reset the TCP connection.

7. SYN :

It synchronizes the sequence number. It's the first bit

8.FIN :

FIN bit is used to terminate the TCP connection.

C	E	U	A	P	R	S	F
W	C	R	C	S	S	Y	I
R	E	G	K	H	T	N	N

Window Size-

- Window size is a 16 bit field.
- It contains the size of the receiving window of the sender.
- It advertises how much data (in bytes) the sender can receive without acknowledgement.
- Thus, window size is used for Flow Control.

Checksum-

- Checksum is a 16 bit field used for error control.
- It verifies the integrity of data in the TCP payload.
- Sender adds CRC checksum to the checksum field before sending the data.
- Receiver rejects the data that fails the CRC check.

Urgent pointer –

- This field (valid only if the URG control flag is set) is used to point to data that is urgently required that needs to reach the receiving process at the earliest. The value of this field is added to the sequence number to get the byte number of the last urgent byte.

Checksum-

- Checksum is a 16 bit field used for error control.
- It verifies the integrity of data in the TCP payload.
- Sender adds CRC checksum to the checksum field before sending the data.
- Receiver rejects the data that fails the CRC check.

Options -

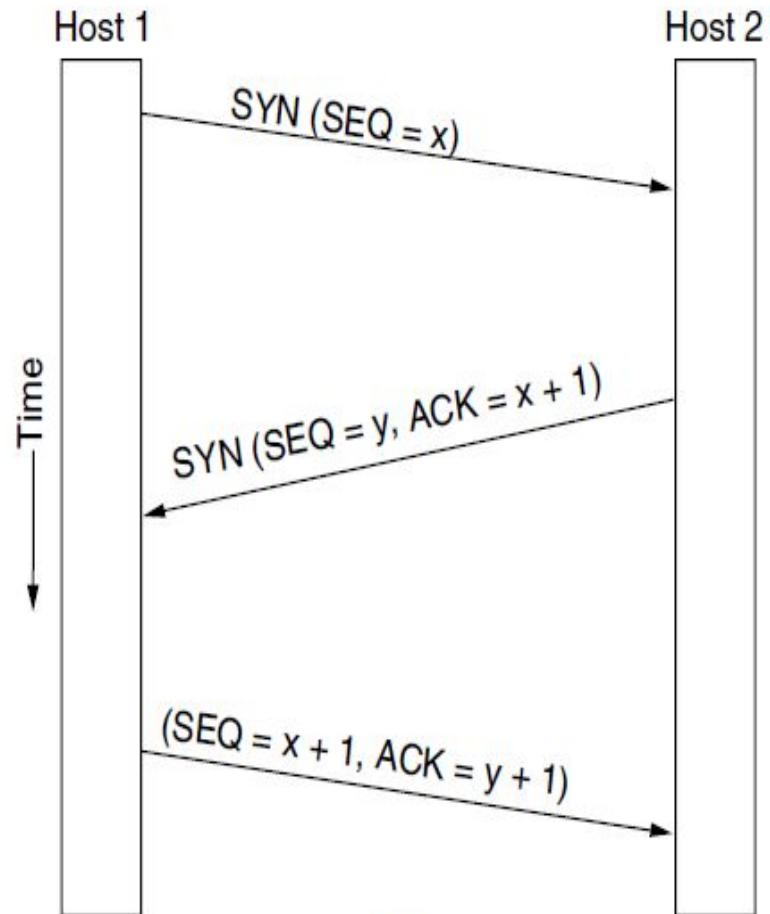
Option field is always described in 32-bit words.

- It facilitates additional options which are not covered by the regular header
- Options field is generally used for the following purposes-
 - Time stamp
 - Window size extension

TCP Connection Establishment

- To make the transport services reliable, TCP hosts must establish a connection-oriented session with one another.
- Connection establishment is performed by using **the three-way handshake mechanism**.
- A three-way handshake synchronizes both ends of a network by enabling both sides to agree upon original sequence numbers.
- This mechanism also provides that both sides are ready to transmit data and learn that the other side is available to communicate.
- This is essential so that packets are not shared or retransmitted during session establishment or after session termination.
- Each host randomly selects a sequence number used to track bytes within the stream it is sending and receiving.

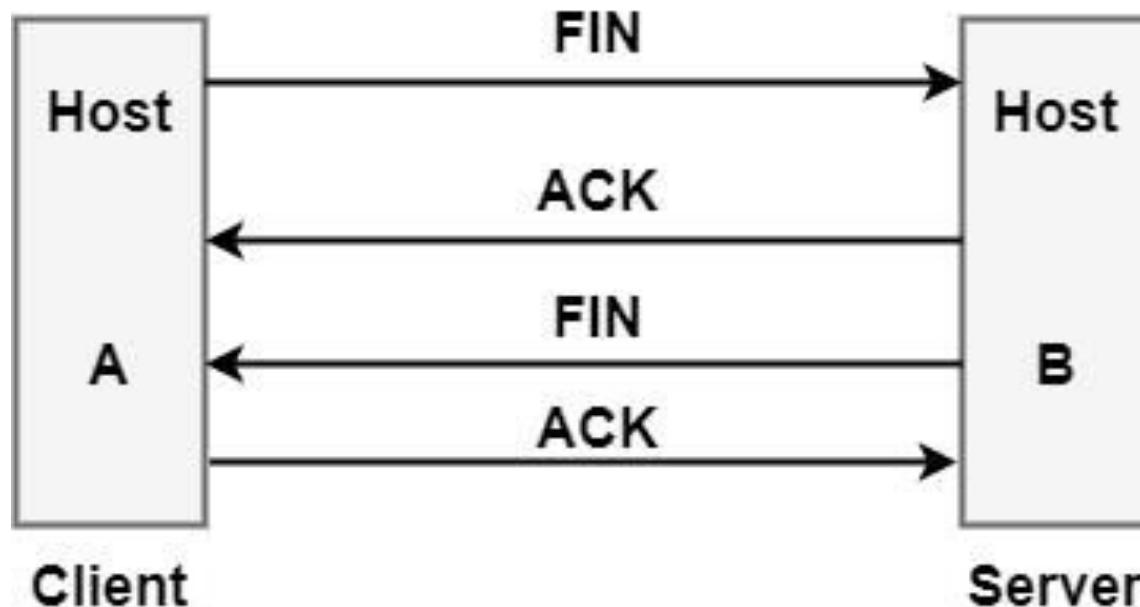
- The first device sends a SYN (synchronize) packet to the second device, indicating its intent to establish a connection.
- The second device receives the SYN packet and sends a SYN-ACK (synchronize-acknowledge) packet back to the first device, indicating its agreement to establish a connection.
- The first device receives the SYN-ACK packet and sends an ACK (acknowledge) packet back to the second device, indicating that it has received the second device's response and is ready to communicate.
- Once the **three-way handshake** is complete, the devices are connected, and data can be exchanged.



TCP connection establishment

TCP Connection Release

- While it creates three segments to establish a connection, it takes **four segments to terminate a connection.**
- During a TCP connection is full-duplex (that is, data flows in each direction independently of the other direction), each direction should be shut down alone.
- The termination procedure for each host is shown in the figure. The rule is that either end can share a FIN when it has finished sending data.
- When a TCP receives a FIN, it should notify the application that the other end has terminated that data flow direction.
- The sending of a FIN is usually the result of the application issuing a close.
- The receipt of a FIN only means that there will be no more data flowing in that direction. A TCP can send data after receiving a FIN.
- The end that first issues the close (example, send the first FIN) executes the active close. The other end (that receives this FIN) manages the passive close.



TCP Termination

UNIT-5

Application Layer



Syllabus

★ The Application Layer:

- Domain name system
- Electronic mail
- World Wide Web: architectural overview
- Dynamic web document

★ Application Layer Protocols

- Http Protocol
- Simple Mail Transfer Protocol
- Simple Network Management Protocol
- File Transfer Protocol
- Telnet

Domain Name System(DNS)

- The DNS Name Space
- Name Servers

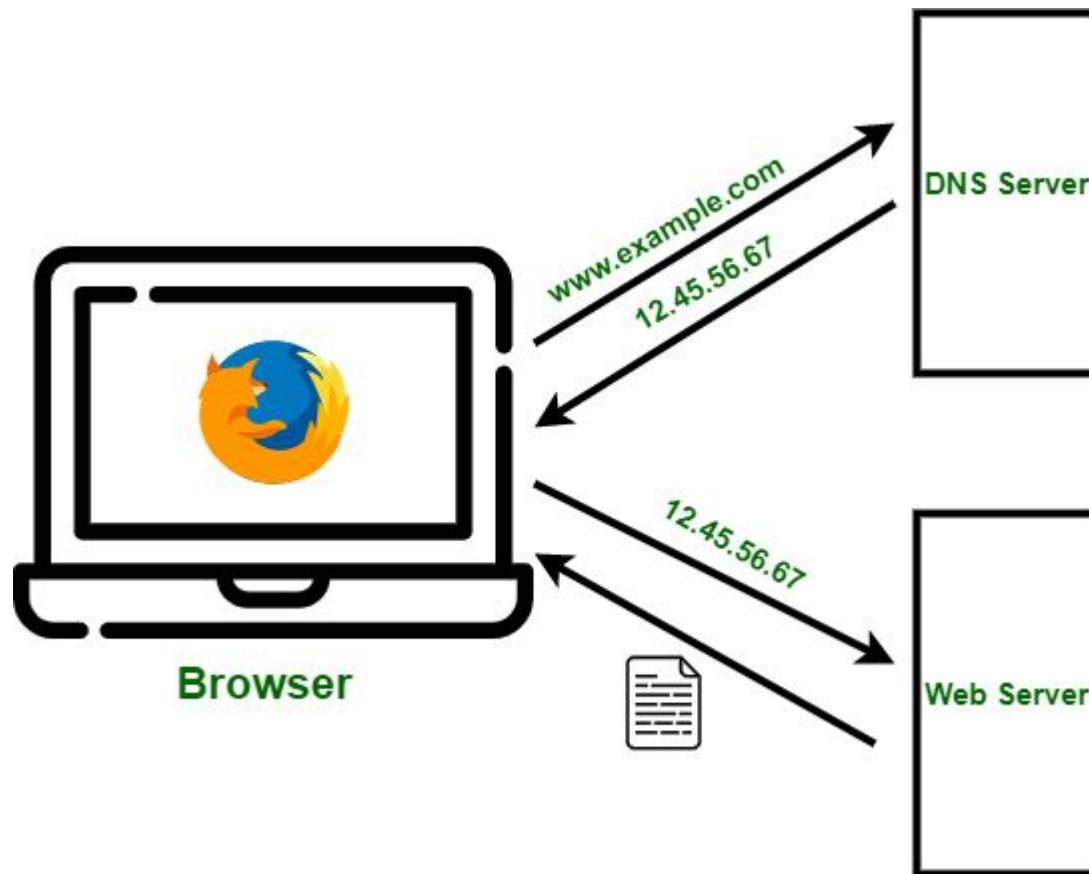
- ★ DNS is an abbreviation of **Domain Name System** (or) **Domain Name Service**. It is an application layer protocol.
- ★ Translates internet **domain names to their unique IP addresses**.



What is the Need of DNS?

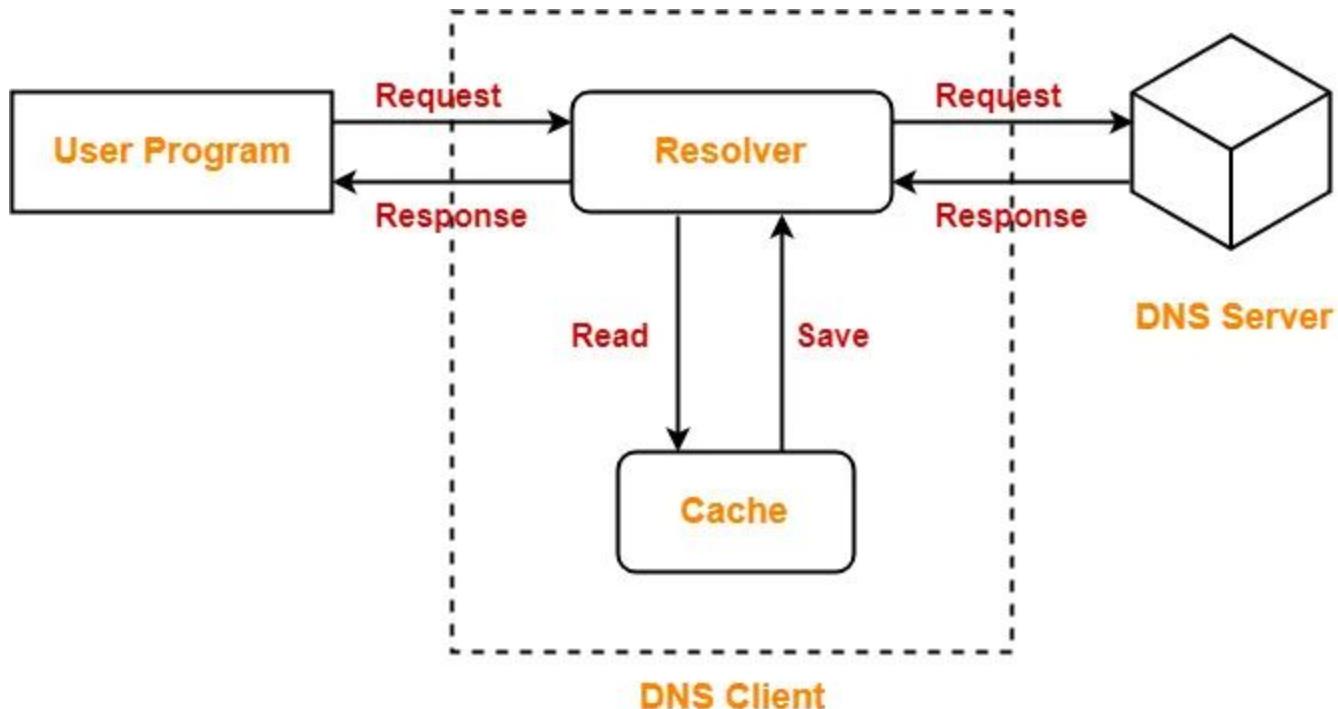
- ★ Every host is identified by the IP address but remembering numbers is very difficult for people.
- ★ So DNS is used to convert the domain name of the websites to their numerical IP address.

Working of Domain Name System (DNS) Server



DNS Resolver?

- To map a host_name onto an IP address, an application program calls a library procedure called **the resolver**, passing it the name as a parameter.



The DNS Name Space

- Managing a large and constantly changing set of names is a nontrivial problem.
- In the postal system, name management is done by requiring letters to specify the country, state, city, street address, and name of the addressee.DNS works the same way.
- For the Internet, the top of the **naming hierarchy** is managed by an organization called **ICANN (Internet Corporation for Assigned Names and Numbers)**.
- Conceptually, the Internet is divided into over **250 top-level domains**, where each domain covers many hosts.
- **Each domain** is partitioned into **subdomains**, and these are further partitioned, and so on.
- All these domains can be represented by a tree, as shown in Fig. 7-1.
- The **leaves** of the tree represent domains that have no subdomains
- A leaf domain may contain a single host, or it may represent a company and contain thousands of hosts.

The Internet Domain Name Space

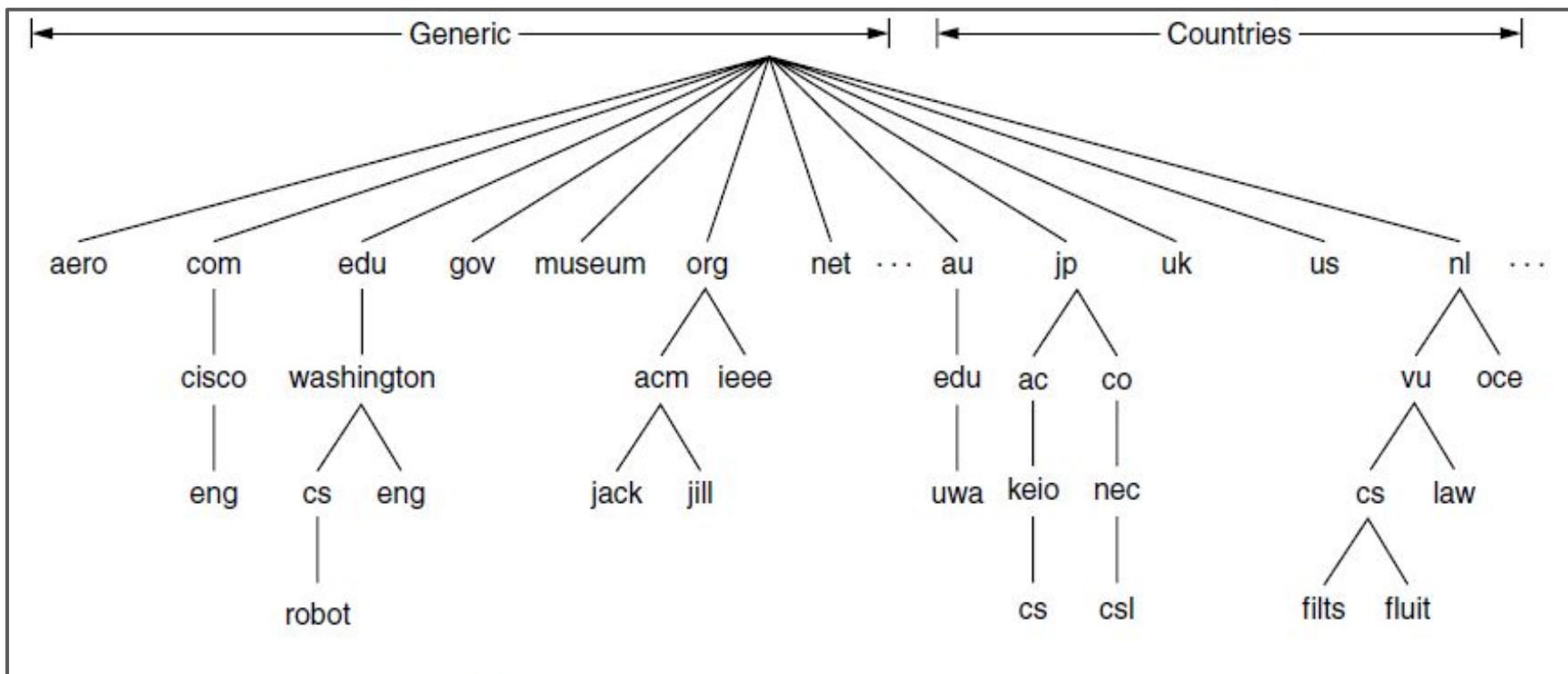


Figure 7-1

The DNS Name Space

- The name of a host is divided into various pieces called **domains**. These domains are structured in a hierarchical structure so that top-level domains are listed at the top of the hierarchy and low levels are listed at the bottom.
- When **searching for a host**, we start our searching in **ascending order**, i.e., **from leaf nodes to root nodes**.

Types of Domain:

1. **Generic domains**
2. **Country domain**

- **Generic domains:**

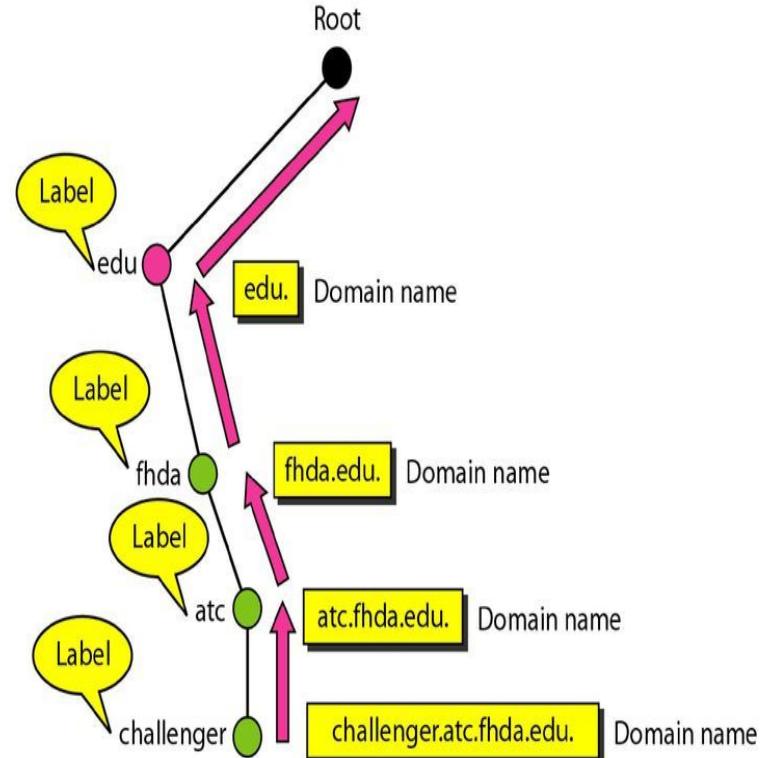
- .com(commercial),
- .edu(educational),
- .mil(military),
- .org(nonprofit organization),
- .net(similar to commercial) all these are generic domains.

- **Country domain:**

- .au (Australia)
- .in (India)
- .us (United States)
- .uk (United Kingdom)
- .nl (Netherlands)
- .jp (Japan)

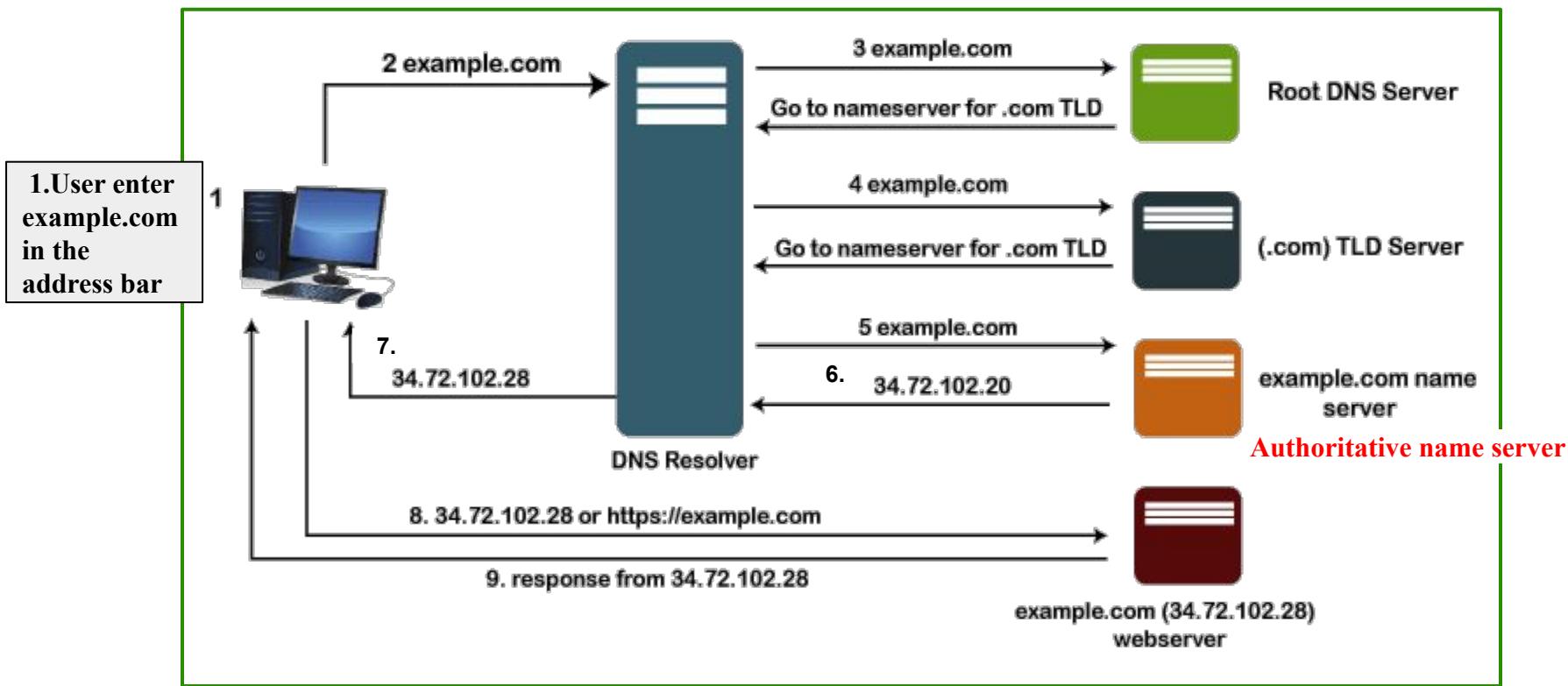
Domain Name and labels

- Each node of the tree has a **domain name**.
- A Full domain name is basically a sequence of labels that are usually **separated by dots(.)**
- The domain name is **always read from the leaf node to the root node.**
- The last label is the label of the root that is always null.
- All this means that the full domain name always ends in the null label, which means that the last character is always a dot because the null string is nothing.



Name Servers

Example of a resolver looking up a remote name in 9 steps



1. Users open a web browser, **enter example.com** in the address bar, and presses Enter button.
2. The request, for example.com is routed to a DNS resolver, which manages by the user's Internet service provider.
3. The **DNS resolver** forwards the request, for example.com to **a root DNS server**.
4. The DNS resolver again forwards the request, for example.com, this time to one of the **TLD(Top Level Domain)name servers for .com domains**. The name server for .com domains responds to the request with the 2 or 4 name servers associated with the example.com domain.
5. The DNS resolver chooses an example.com, **the authoritative name server**, and forwards the request for example.com to that name server.
6. The website's name server looks in the example.com hosted zone for the example.com a record, to get the associated value, such as the IP address for a web server, 34.72.102.28, and returns the IP address to the DNS resolver.

7. Finally, The DNS resolver for the ISP has the IP address that the user needs. The resolver returns that value to the web browser. The DNS resolver can store the IP address, for example.com.
8. The web browser sends a request, for example.com using the IP address that it got from the DNS resolver. This is where the actual content is.
9. The web server or other resource at 34.72.102.28 returns the web page, for example.com to the web browser, and the web browser displays the page.

Electronic mail(e-mail)

- ★ Electronic mail, or more commonly email, is a method of **exchanging messages** over the internet.
- ★ Electronic mail allows a message to **include text, audio, and video**. It also allows one message to be sent to one or more recipients.
- ★ Faster and cheaper than paper mail, email has been a popular application since the early days of the Internet.
- ★ **E-mail systems support 5 basic functions:-**
 - **Composition**
 - **Transfer**
 - **Reporting**
 - **Displaying**
 - **Disposition**

(a) Composition:

- It refers to the process of creating messages and answers. Any text editor is used for body of the message.
- While the system itself can provide assistance with addressing and numerous header fields attached to each message.

(b) Reporting:

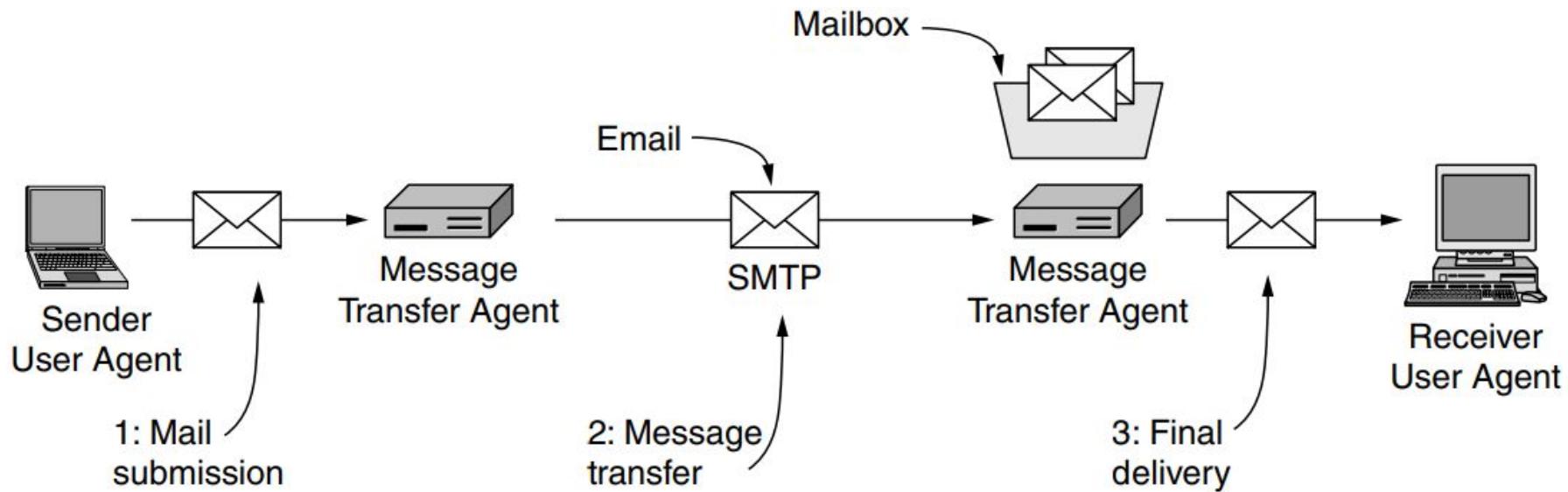
- It has to do with telling the originator what happened to the message that is, whether it was delivered, rejected (or) lost.

(c) Transfer: It refers to moving messages from originator to the recipient.

(d) Displaying: Incoming messages are to be displayed so that people can read their email.

(e) Disposition: It concerns what the recipient dose with the message after receiving it.

Architecture of the email system



Architecture of the email system :-

- The architecture of the email system is shown in Fig. 7-7.
- It consists of two kinds of subsystems:
 - **The UserAgent** : which allow people to read and send email
 - **The Message Transfer Agents** : which move the messages from the source to the destination. We will also refer to message transfer agents informally as **mail servers**.

User Agent:

- The user agent is a program that provides **a graphical interface, or sometimes a text- and command-based interface** that lets users interact with the email system.
- It includes a means to compose messages and replies to messages, display incoming messages, and organize messages by filing, searching, and discarding them.
- The act of sending new messages into the mail system for delivery is called mail submission.

Message Transfer Agent:

- The message transfer agents are typically system processes. They run in the background on mail server machines and are intended to be always available.
- Their job is to **automatically move email through the system from the originator to the recipient with SMTP**. This is the message transfer step.

SMTP (Simple Mail Transfer Protocol):

- It is a set of rules and conventions used to transmit email messages between servers.
- SMTP is an essential component of email communication and **is responsible for routing and delivering outgoing email messages from the sender's email client or server to the recipient's email server.**
- **It reports back the delivery status and any errors.**

Message Format of an email :

→ Email messages follow a specific format that includes various components to ensure proper communication and presentation. Here's an overview of the key message formats in an email system.

1. Header Field:

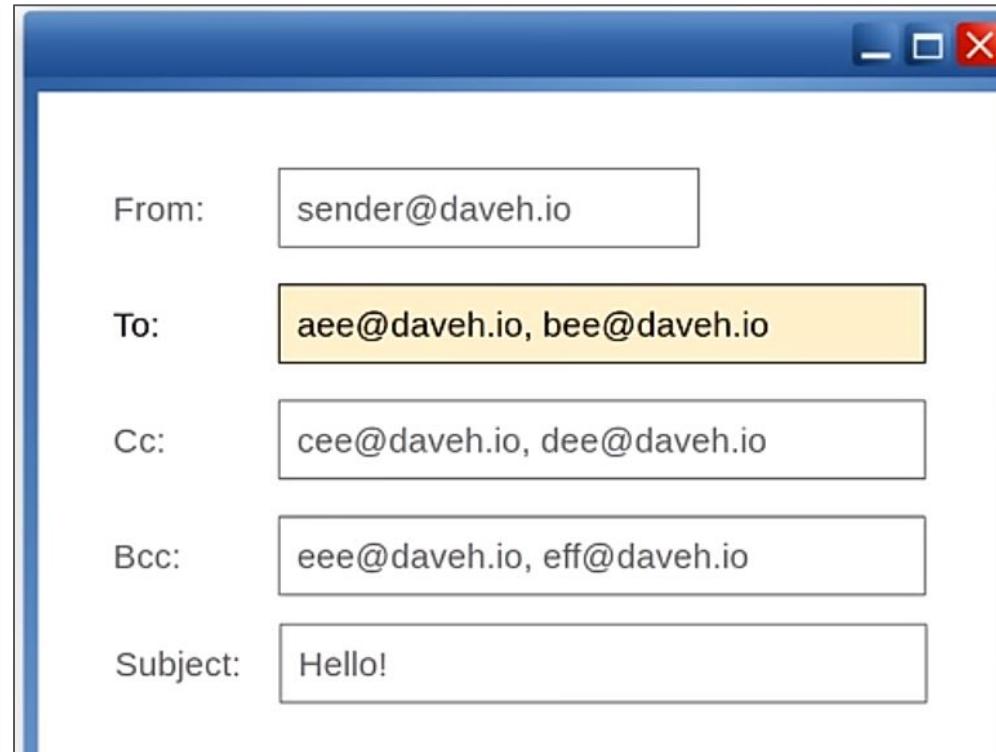
Header	Meaning
To:	Email address(es) of primary recipient(s)
Cc:	Email address(es) of secondary recipient(s)
Bcc:	Email address(es) for blind carbon copies
From:	Person or people who created the message

CC (Carbon Copy):

- When you CC someone on an email, all recipients can see who else received the email.
- It's commonly used when you want to keep others informed about the email conversation but they are not directly involved in it.

BCC (Blind Carbon Copy):

- When you BCC someone on an email, the recipients in the "To" and "CC" fields can't see that person's email address.
- It's used when you want to include someone in the conversation without letting other recipients know.



CC (Carbon Copy) and BCC (Blind Carbon Copy) are both used in email to include additional recipients beyond the primary recipient.

2. Message Body:

Text Content: This is the main part of the message where the sender writes the actual text. It can include plain text or formatted text (HTML).

3. Multipurpose Internet Mail Extensions (MIME):

Multipurpose Internet Mail Extensions (MIME) is an Internet standard that extends the format of email messages to support text in character sets other than ASCII, as well as attachments of **audio, video, images, and application programs.**

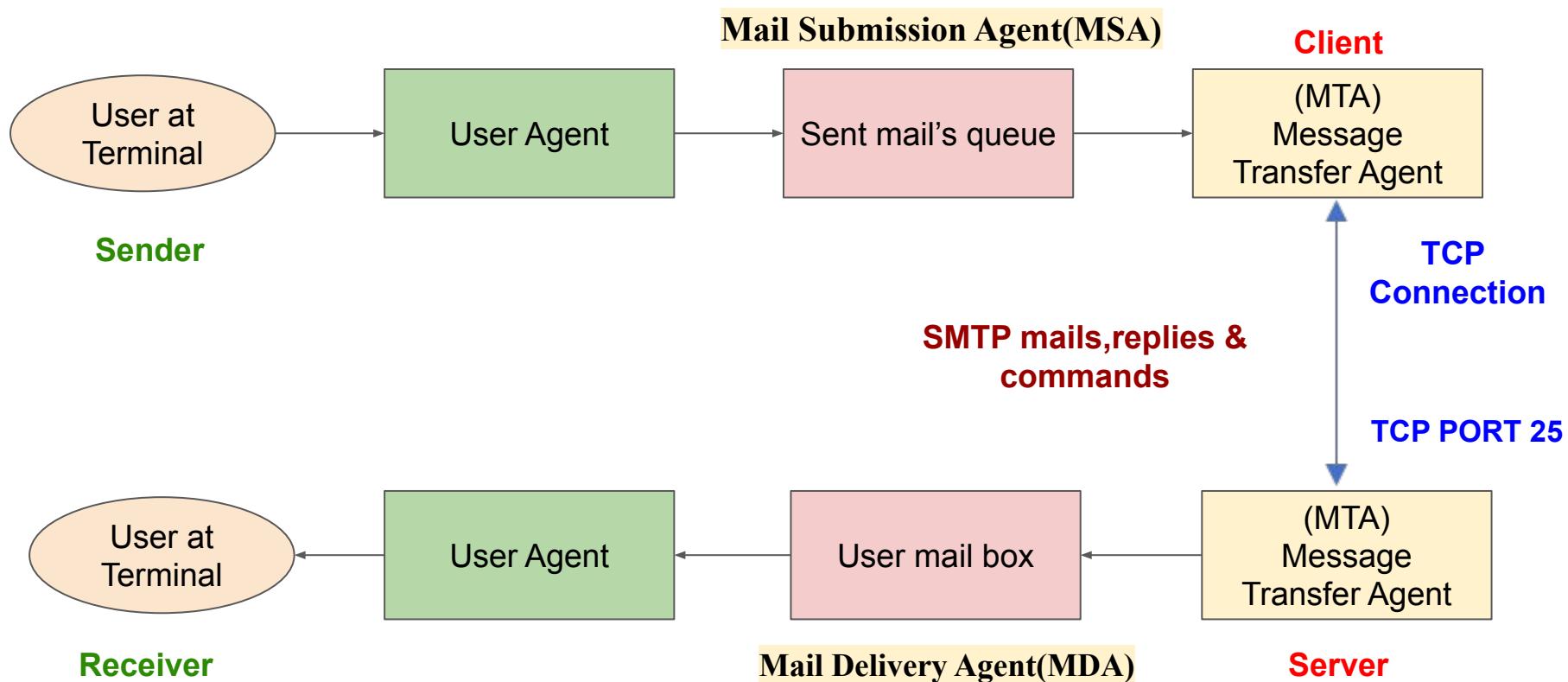
SMTP(Simple Mail Transfer Protocol)

- ★ **SMTP (Simple Mail Transfer Protocol)** is an application layer used in sending and receiving email.
- ★ SMTP is used most commonly by email clients, including Gmail, Outlook, Apple Mail and Yahoo Mail.

SMTP Fundamentals

- ★ The client who wants to send the mail opens **a TCP connection** to the SMTP server and then sends the mail across the connection. The SMTP server is an always-on listening mode. As soon as it listens for a TCP connection from any client, the SMTP process initiates a connection through **port 25**. After successfully establishing a TCP connection the client process sends the mail instantly.
- ★ **If a message cannot be delivered**, an error report containing the first part of the undeliverable message is returned to the sender.

SMTP Model



Components of SMTP

1. User Agent (MUA)
2. Mail Submission Agent (MSA)
3. Mail Transfer Agent (MTA)
4. Mail Delivery Agent (MDA)

1. Mail User Agent (MUA):

- ★ It is a computer application that helps you in **sending and retrieving mail**.
- ★ It is responsible for creating email messages for transfer to the mail transfer agent(MTA).

2. Mail Submission Agent (MSA):

- It is a computer program that basically receives mail from User Agent(MUA) and interacts with the Mail Transfer Agent(MTA) for the transfer of the mail.

3. Mail Transfer Agent(MTA):

- ★ It is basically software that has the work to transfer mail from one system to another with the help of **SMTP**.

4. Mail Delivery Agent(MDA):

- A mail Delivery agent or Local Delivery Agent is basically a system that helps in the delivery of mail to the local system.

Some SMTP Commands

1. HELO/EHLO

- The HELO command **initiates the SMTP session conversation.**
- The client greets the server and introduces itself. As a rule, HELO is attributed with an argument that specifies the domain name or IP address of the SMTP client.

2. MAIL FROM

- The MAIL FROM command **initiates a mail transfer.**

3. RCPT TO

- The RCPT TO command **specifies the recipient.**

4. DATA

- With the DATA command, the client asks the server for permission to **transfer the mail data.**

POP3 & IMAP Protocols



POP3 & IMAP Protocols

1. POP3 (Post Office Protocol, version 3)

- POP3 is a simple and older **email retrieval protocol**.
- It is primarily designed for **downloading email messages to a local device** (e.g., a **computer or a mobile device**).
- POP3 operates over non-secure **port 110**
- POP3 doesn't support advanced mailbox management features like organizing emails into folders on the server.

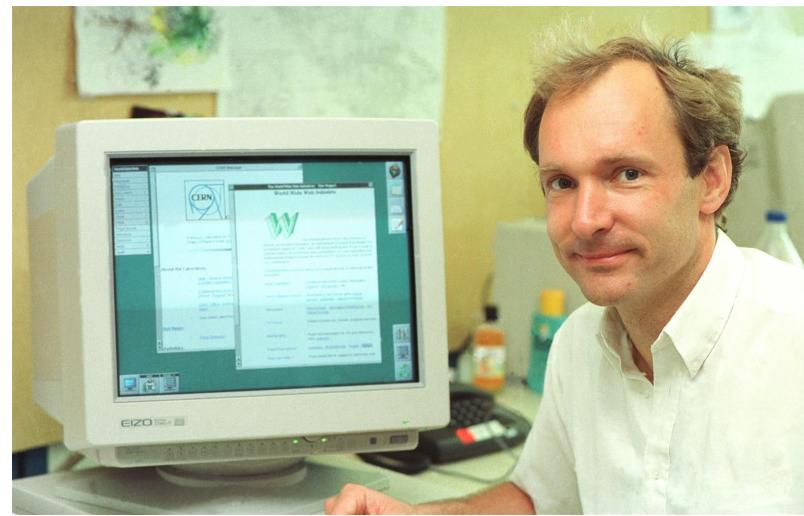
2. IMAP (The Internet Message Access Protocol)

- IMAP is a **much more advanced email retrieval protocol** compared to POP3.
- It allows users to access and manage their email messages stored on mail server.
- The IMAP operates on **port 143**.

World Wide Web: Architectural overview

“ *World Wide Web*, which is also known as *a Web*, is a **collection of websites or web pages stored in web servers** and connected to local computers through the internet. These websites contain text pages, digital images, audios, videos, etc. “

- **Tim Berners-Lee**, a British scientist, invented the World Wide Web (WWW) in 1989, while working at CERN.
- The Web was originally conceived and developed to meet the demand for automated information-sharing between scientists in universities and institutes around the world.
- CERN, where Tim Berners worked, is a community of more than 1700 scientists from more than 100 countries.
- So there was a need for reliable communication tools so that they can exchange information.



Tim Berners-Lee

Architectural Overview

Web page:

- The **Web** consists of a vast, worldwide collection of content in the form of **Web pages**, often just called **pages** for short. Each

Hypertext:

- In the context of the web, hypertext is text that contains links (hyperlinks) to other web pages. These links are usually displayed as underlined or differently colored text and are clickable.
- The idea of having one page point to another, now called **hypertext**.

Web browser:

- Pages are generally viewed with a program called **a browser**. Firefox, Internet Explorer, and Chrome are examples of popular browsers.
- The browser fetches the page requested, interprets the content, and displays the page, properly formatted, on the screen.

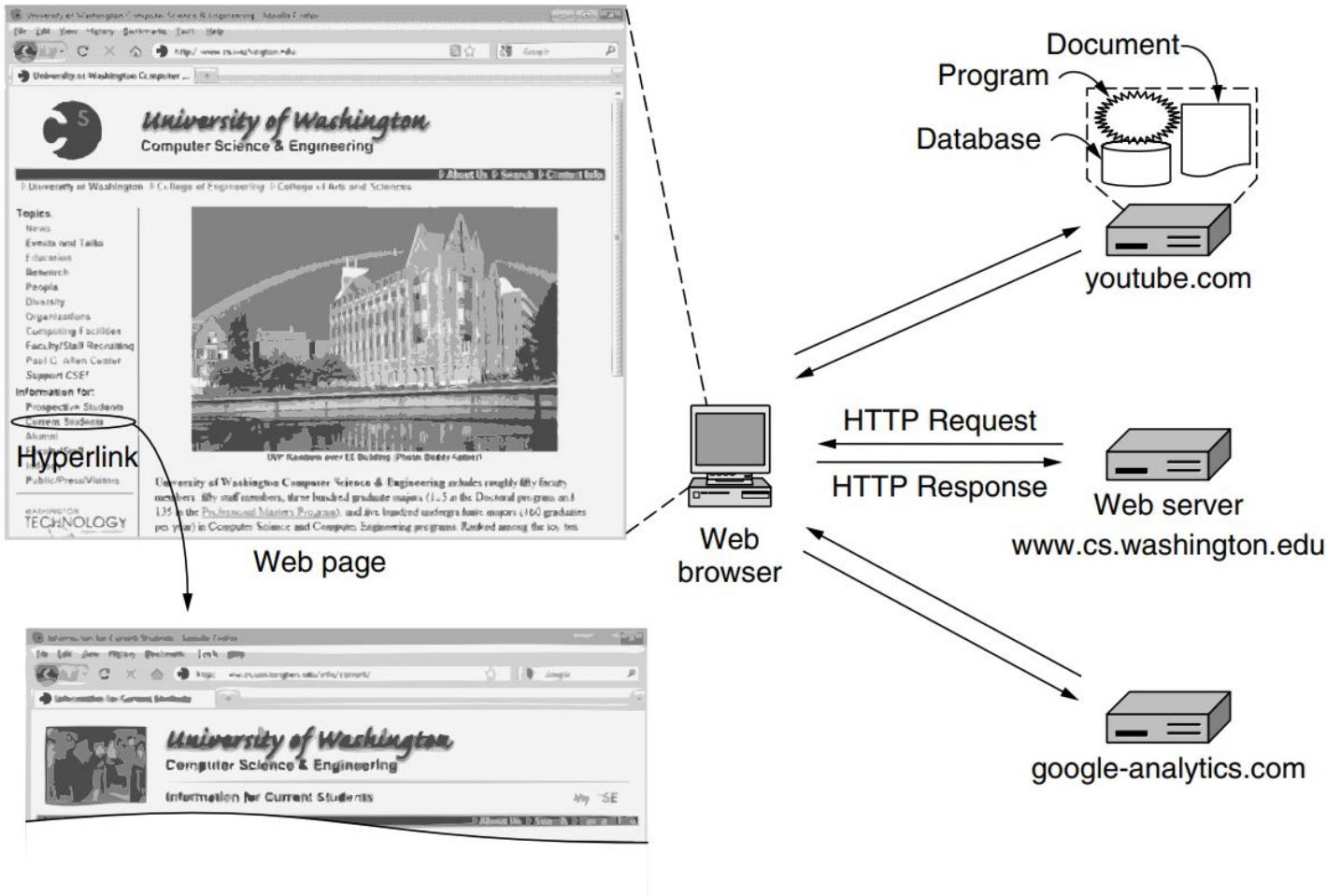


Figure 7-18. Architecture of the Web.

- The basic model behind the display of pages is also shown in Fig. 7-18.
- The browser is displaying a Web page on the client machine.
- Each page is fetched by sending a request to one or more servers, which respond with the contents of the page.
- **The request-response protocol for fetching pages is a simple text-based protocol , called **HTTP(Hypertext Transfer Protocol)**that runs over TCP.**
- Each page is assigned a **URL (Uniform Resource Locator)** that effectively serves as the page's worldwide name.
- **URLs have three parts: (Eg: <http://www.cs.washington.edu/index.html>)**
 - a. The protocol (**http**),
 - b. The DNS name of the machine on which the page is located, (**www.cs.washington.edu**)
 - c. The path uniquely indicating the specific page (**index.html**)

As an example, the URL of the page shown in Fig. 7-18 is
<http://www.cs.washington.edu/index.html>

Client Side:

1. The browser determines the **URL** (by seeing what was selected).
2. The browser asks DNS for the IP address of the server **www.cs.washington.edu**.
3. DNS replies with **128.208.3.88**.
4. The browser makes **a TCP connection to 128.208.3.88 on port 80**, the well-known **port** for the **HTTP protocol**.
5. It sends over an **HTTP request** asking for the page **/index.html**.
6. **The www.cs.washington.edu server** sends the page as an HTTP response, for example, by sending the file **/index.html**

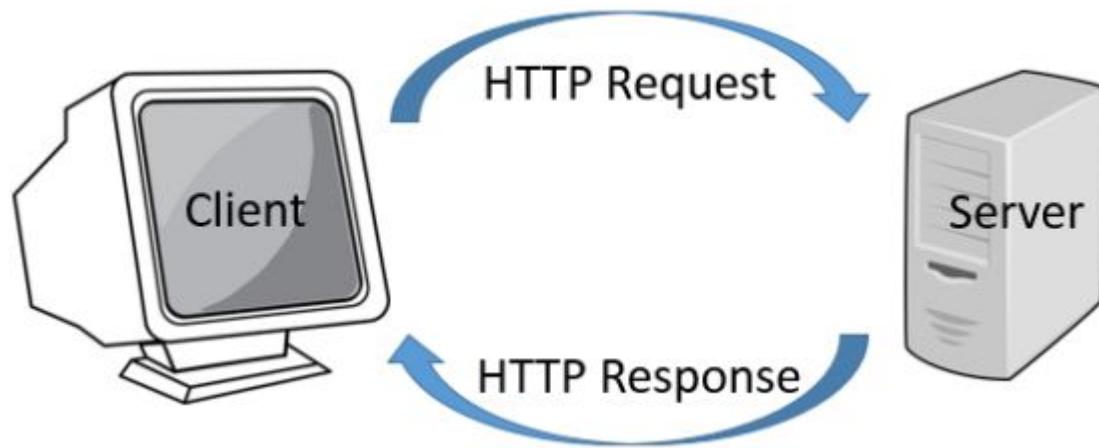
7. The browser displays the page /index.html as it appears in Fig. 7-18.
8. The TCP connections are released if there are no other requests to the same servers for a short period.

The Server Side:

1. Accept a TCP connection from a client (a browser).
2. Get the path to the page, which is the name of the file requested.
3. Get the file (from disk).
4. Send the contents of the file to the client.
5. Release the TCP connection.

HTTP Protocol(Hypertext Transfer Protocol)

- HTTP (HyperText Transfer Protocol), the protocol that is **used to transport all this information between Web Servers and Clients.**
- HTTP is a simple **request-response protocol** that normally **runs over TCP**.
- It specifies what messages clients may send to servers and what responses they get back in return.



Types of HTTP Connections

- 
- 1. Non-Persistent Connection**
 - 2. Persistent Connection**
 - 3. Pipelined Connection**

1. Non-Persistent Connection [HTTP/1.0 connection]

- Also known as "HTTP/1.0" connection.
- In a non-persistent connection, a new TCP connection is established for each HTTP request/response exchange.
- In non-persistent connection HTTP, there can be **at most one webpage that can be sent over a single TCP connection**. This means that for each webpage that is to be sent from source to destination, a new connection will be created
- After each request-response cycle, the connection is closed.
- This approach was **used in early versions of HTTP** and is relatively **inefficient** because it incurs the overhead of establishing and closing a new connection for each resource request, leading to slower page loading times.

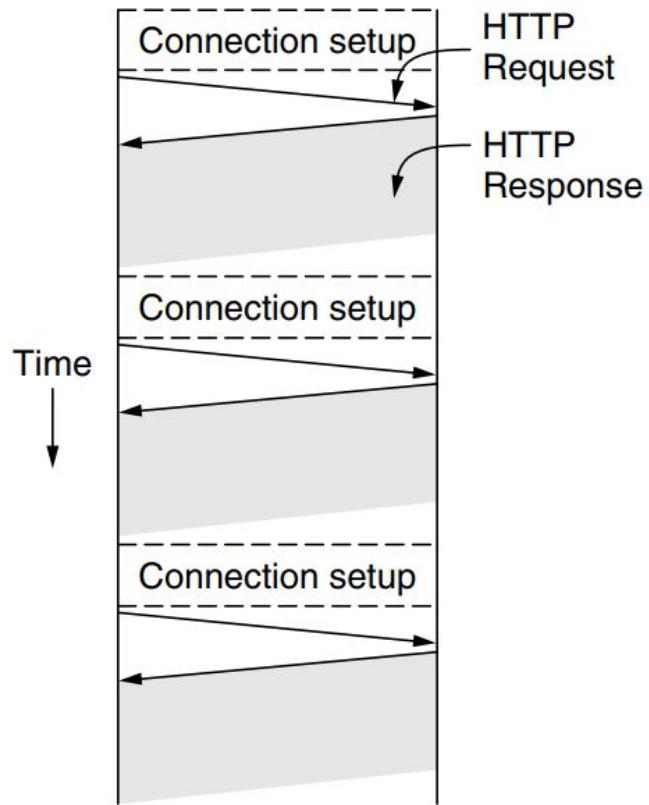


Fig:
Multiple connections and sequential requests.

2. Persistent Connection [HTTP/1.1 connection]

- Also known as a "keep-alive" or "HTTP/1.1" connection.
- In a persistent connection, the TCP connection between the client and the server remains open after the initial request and response, allowing **multiple HTTP requests and responses** to be sent over **the same connection**.
- This means that multiple web pages can be transmitted from source to destination on a single HTTP connection
- This reduces the overhead associated with opening and closing connections for each resource, leading to faster loading times and improved performance.
- All modern web browsers like **Mozilla Firefox and Google Chrome** use persistent HTTP connections.

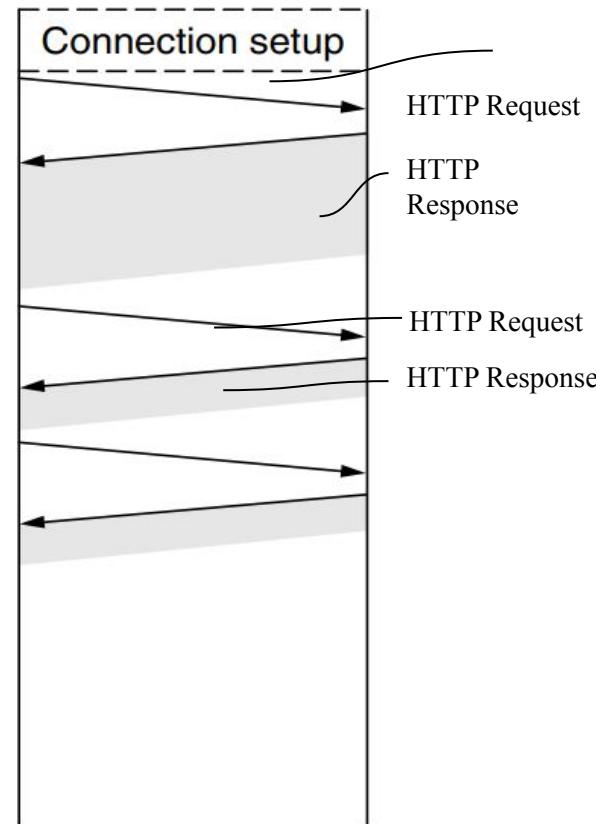


Fig:
A persistent connection and sequential requests

3. Pipelined Connection:

- HTTP pipelining is a feature that allows **multiple HTTP requests** to be sent to the server **without waiting for each response to arrive before sending the next request.**
- With pipelining, the client can send a series of requests in rapid succession without waiting for the responses, which can help reduce latency.
- However, pipelining is not always well-supported by all servers and intermediaries, and careful response handling is required to ensure that responses are processed in the correct order.

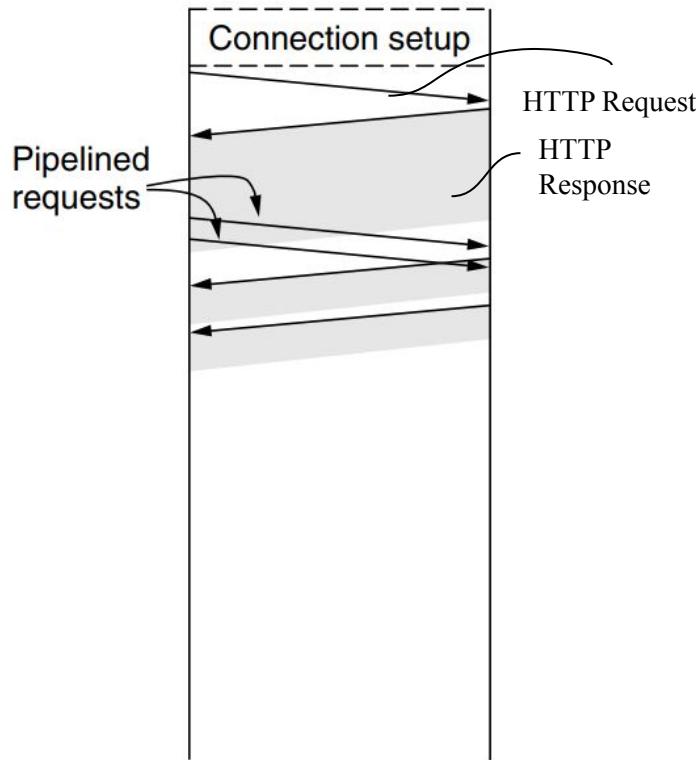


Fig:
A persistent connection and pipelined request

Simple Network Management Protocol(**SNMP**)

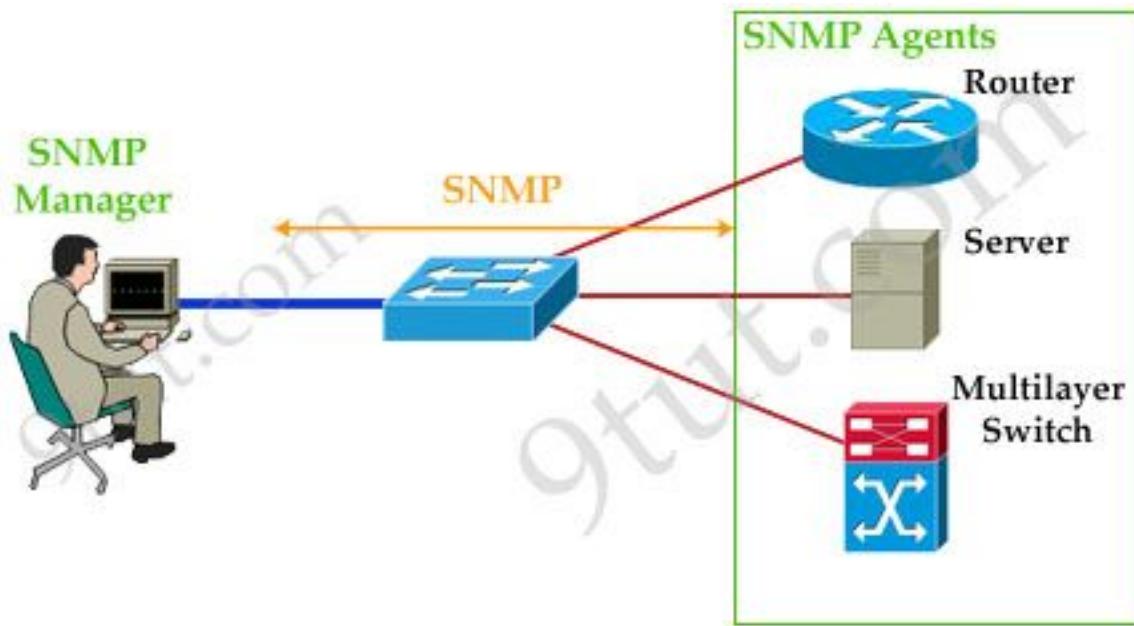
- **SNMP (Simple Network Management Protocol)** is an application layer protocol that utilizes **the UDP protocol** to manage routers, hubs, and switches on an IP network.
- Simple Network Management Protocol (SNMP) is a protocol used in **network management systems** to monitor and manage network devices and their functions.
- SNMP is a widely used protocol that is enabled on a wide range of operating systems, including Windows Server, Linux servers, and network devices such as routers and switches.
- SNMP is an essential part of network management and plays a crucial role in ensuring the reliability and performance of networked systems.
- On a target system, SNMP enumeration is used to list user accounts, passwords, groups, system names, and devices.

Understand SNMP

SNMP consists of 3 items:

- 1. SNMP Manager** (sometimes called Network Management System – NMS): a software runs on the device of the network administrator (in most case, a computer) to monitor the network.
- 2. SNMP Agent:** a software runs on network devices that we want to monitor (router, switch, server...).
- 3. Management Information Base (MIB):** is the collection of managed objects. This components makes sure that the data exchange between the manager and the agent remains structured. In other words, MIB contains a set of questions that the SNMP Manager can ask the Agent (and the Agent can understand them).

MIB is commonly shared between the Agent and Manager.



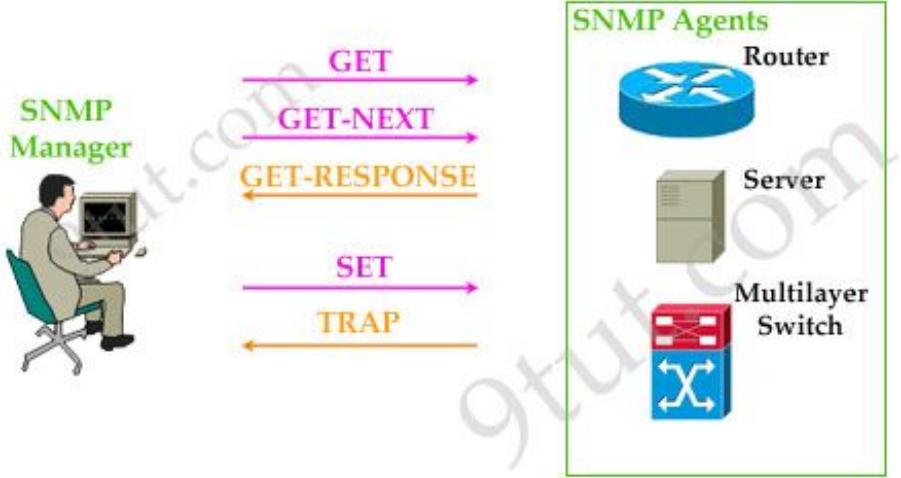
For example, in the topology above you want to monitor a router, a server and a Multilayer Switch. You can run SNMP Agent on all of them. Then on a PC you install a SNMP Manager software to receive monitoring information. SNMP is the protocol running between the Manager and Agent. SNMP communication between Manager and Agent takes place in form of messages. The monitoring process must be done via a MIB which is a standardized database and it contains parameters/objects to describe these networking devices (like IP addresses, interfaces, CPU utilization, ...). Therefore the monitoring process now becomes the process of GET and SET the information from the MIB.

SNMP Messages

SNMP Messages are used to communicate between **the SNMP Manager and Agents.**

SNMP supports five basic SNMP messages:

- 1. SNMP GET**
- 2. SNMP GET-NEXT**
- 3. SNMP GET-RESPONSE**
- 4. SNMP SET**
- 5. SNMP TRAP**



- **The GET messages** are sent by the SNMP Manager to retrieve information from the SNMP Agents .
- **The SET messages** are used by the SNMP Manager to modify or assign the value to the SNMP Agents.
- **GET-NEXT** retrieves the value of the next object in the MIB.
- **The GET-RESPONSE message** is used by the SNMP Agents to reply to GET and GET-NEXT messages.
- **TRAP messages** are initiated from the SNMP Agents to inform the SNMP Manager on the occurrence of an event.

File Transfer Protocol(FTP)

- File transfer protocol (FTP) is an Internet tool provided by TCP/IP.
- It helps to transfer files from one computer to another by providing access to directories or folders on remote computers and allows software, data, text file to be transferred between different kinds of computers.
- The end-user in the connection is known as localhost and the server which provides data is known as the remote host.

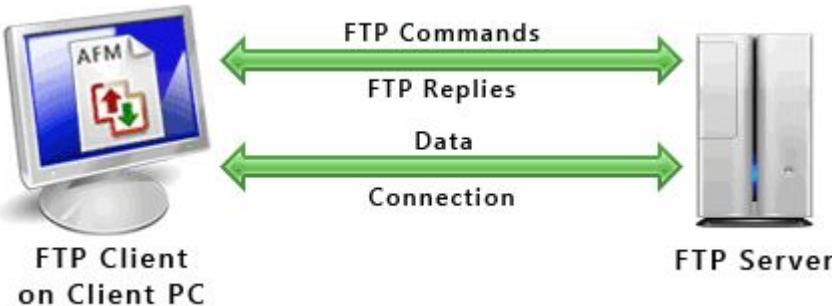
The goals of FTP are:

- It encourages the direct use of remote computers.
- It shields users from system variations (operating system, directory structures, file structures, etc.)
- It promotes sharing of files and other types of data.

FTP protocol works on Client-Server Model:

Client-Server Model:

- A client initiates a connection to a remote server to transfer files. The client is responsible for sending commands, and the server responds to these commands.
- **Some of the commands are:**
 - **get filename** (retrieve the file from server)
 - **mget filename** (retrieve multiple files from the server)
 - **ls** (lists files available in the current directory of the server)



FTP operates in two primary modes: **Active and Passive**

Active Mode:

In this mode, **the client opens** a random port (known as the "data port") for data transfer, and the server connects to this port. The client's command port remains the same.

Passive Mode:

In passive mode, **the server opens** a random data port, and the client connects to it for data transfer. This mode is often used when the client is behind a firewall or NAT, making it difficult for the server to initiate a connection.

Port Numbers:

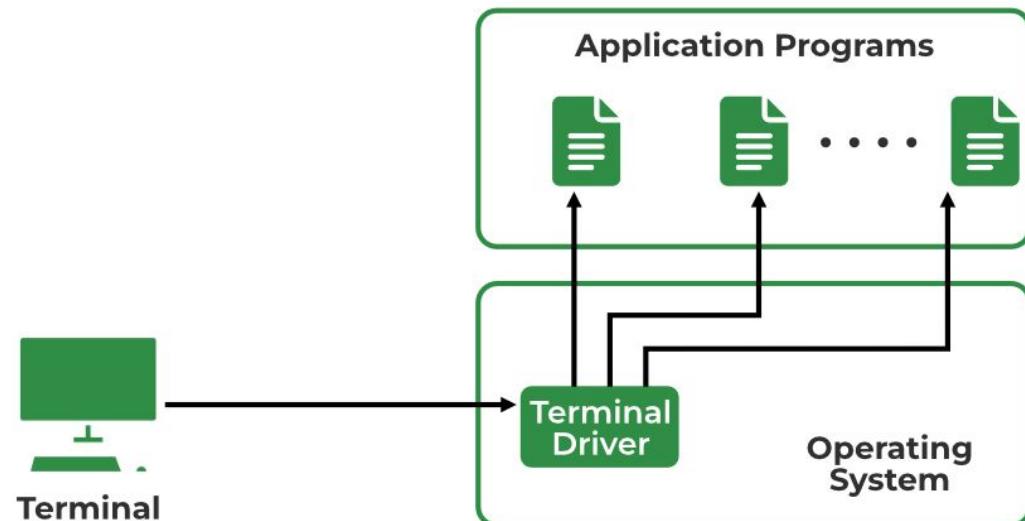
FTP uses well-defined port numbers for communication. The default FTP control **port is 21**, and data transfers occur on various ports depending on whether active or passive mode is used.

TELNET

- TELNET is basically the short form for TErminal NETwork. It is basically a TCP/IP protocol that is used for virtual terminal services and was mainly proposed by International Organization for Standards(ISO).
- It is a general-purpose client/server application program.
- This program enables the establishment of the connection to the remote system in such a way that the local system starts to appear as a terminal at the remote system.
- It is a standard TCP/IP protocol that is used for virtual terminal service.
- **In simple words, we can say that the telnet allows the user to log on to a remote computer. After logging on the user can use the services of the remote computer and then can transfer the results back to the local computer.**
- The TELNET was mainly designed at the time when most operating systems operate in the time-sharing environment. And in this type of environment, a large computer can support multiple users.
- Usually, the interaction between the computer and user occurs via terminal (It is a combination of keyboard, mouse, and monitor).
- TELNET makes the use of only one TCP/IP connection.

Logging

- The logging process can be further categorized into two parts:
 - Local Login
 - Remote Login
- **Local Login:** Whenever a user logs into its local system, it is known as local login.



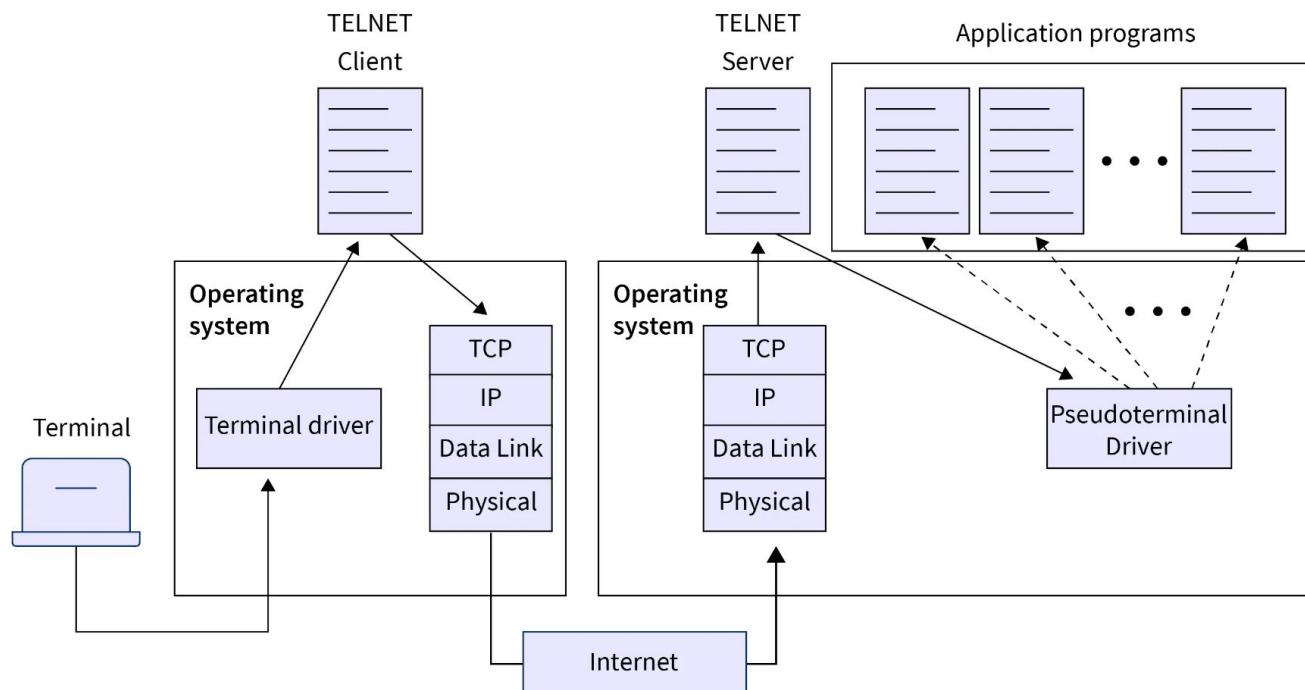
Local Login

The Procedure of Local Login

- Keystrokes are accepted by the terminal driver when the user types at the terminal.
- Terminal Driver passes these characters to OS.
- Now, OS validates the combination of characters and opens the required application.

2. Remote Login:

- Remote Login is a process in which users can log in to a remote site i.e. computer and use services that are available on the remote computer.
- With the help of remote login, a user is able to understand the result of transferring the result of processing from the remote computer to the local computer.



Remote Login in Logging

The Procedure of Remote Login

- When the user types something on the local computer, the local operating system accepts the character.
- The local computer does not interpret the characters, it will send them to the TELNET client.
- TELNET client transforms these characters to a universal character set called Network Virtual Terminal (NVT) characters and it will pass them to the local TCP/IP protocol Stack.
- Commands or text which are in the form of NVT, travel through the Internet and it will arrive at the TCP/IP stack at the remote computer.
- Characters are then delivered to the operating system and later on passed to the TELNET server.
- Then TELNET server changes those characters to characters that can be understandable by a remote computer.
- The remote operating system receives characters from a pseudo-terminal driver, which is a piece of software that pretends that characters are coming from a terminal.
- The operating system then passes the character to the appropriate application program.